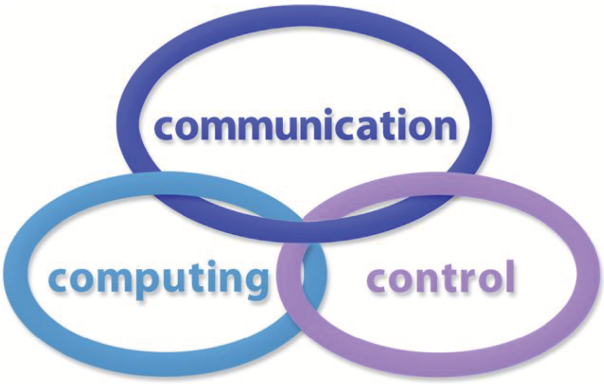
INTERNATIONAL JOURNAL

of

COMPUTERS, COMMUNICATIONS & CONTROL

With Emphasis on the Integration of Three Technologies

IJCCC



Year: 2011 Volume: 6 Number: 1 (March)

Agora University Editing House

CCC Publications

www.journal.univagora.ro

International Journal of Computers, Communications & Control

EDITORIAL ADDRESS:
R&D Agora Ltd. / S.C. Cercetare Dezvoltare Agora S.R.L.
Piaţa Tineretului 8, Oradea, jud. Bihor, Romania, Zip Code 410526
Tel./ Fax: +40 359101032
E-mail: ijccc@univagora.ro, rd.agora@univagora.ro, ccc.journal@gmail.com
Journal website: www.journal.univagora.ro

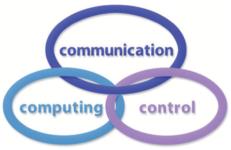
DATA FOR SUBSCRIBERS
Supplier: Cercetare Dezvoltare Agora Srl (Research & Development Agora Ltd.)
Fiscal code: RO24747462
Headquarter: Oradea, Piata Tineretului Nr.8, Bihor, Romania, Zip code 410526
Bank: MILLENNIUM BANK, Bank address: Piata Unirii, str. Primariei, 2, Oradea, Romania
IBAN Account for EURO: RO73MILB0000000000932235
SWIFT CODE (eq.BIC): MILBROBU

International Journal of Computers, Communications & Control

International Journal of Computers, Communications & Control

Short Description of IJCCC

Title of journal: International Journal of Computers, Communications & Control
Acronym: IJCCC
International Standard Serial Number: ISSN 1841-9836, E-ISSN 1841-9844
Publisher: CCC Publications - Agora University
Starting year of IJCCC: 2006
Founders of IJCCC: Ioan Dzitac, Florin Gheorghe Filip and Mişu-Jan Manolescu
Logo:



Number of issues/year: IJCCC has 4 issues/odd year (March, June, September, December) and 5 issues/even year (March, September, June, November, December). Every even year IJCCC will publish a supplementary issue with selected papers from the International Conference on Computers, Communications and Control.

Coverage:

- Beginning with Vol. 1 (2006), Supplementary issue: S, IJCCC is covered by Thomson Reuters - SCI Expanded and is indexed in ISI Web of Science.
- Journal Citation Reports/Science Edition 2009:
  - Impact factor = 0.373
  - Immediacy index = 0.205
- Beginning with Vol. 2 (2007), No.1, IJCCC is covered in EBSCO.
- Beginning with Vol. 3 (2008), No.1, IJCCC, is covered in SCOPUS.

Scope: IJCCC is directed to the international communities of scientific researchers in universities, research units and industry. IJCCC publishes original and recent scientific contributions in the following fields: Computing & Computational Mathematics; Information Technology & Communications; Computer-based Control.

Unique features distinguishing IJCCC: To differentiate from other similar journals, the editorial policy of IJCCC encourages especially the publishing of scientific papers that focus on the convergence of the 3 "C" (Computing, Communication, Control).

Policy: The articles submitted to IJCCC must be original and previously unpublished in other journals. The submissions will be revised independently by at least two reviewers and will be published only after completion of the editorial workflow.

Contents

Int. J. of Computers, Communications & Control, ISSN 1841-9836, E-ISSN 1841-9844
Vol. VI (2011), No. 1 (March), pp. 8-20

Genetic Algorithm for Fuzzy Neural Networks using Locally Crossover

D. Arotaritei

Dragos Arotaritei
"Gr. T Popa" University of Medicine and Pharmacy
Romania, 700115 Iasi
E-mail: dragos_aro@yahoo.com

Abstract: Fuzzy feed-forward (FFNR) and fuzzy recurrent networks (FRNN) proved to be solutions for "real world problems". In the most cases, the learning algorithms are based on gradient techniques adapted for fuzzy logic with heuristic rules in the case of fuzzy numbers. In this paper we propose a learning mechanism based on genetic algorithms (GA) with locally crossover that can be applied to various topologies of fuzzy neural networks with fuzzy numbers. The mechanism is applied to FFNR and FRNN with L-R fuzzy numbers as inputs, outputs and weights and fuzzy arithmetic as forward signal propagation. The α-cuts and fuzzy biases are also taken into account. The effectiveness of the proposed method is proven in two applications: the mapping a vector of triangular fuzzy numbers into another vector of triangular fuzzy numbers for FFNR and the dynamic capture of fuzzy sinusoidal oscillations for FRNN.

Keywords: rules, figures, citation of papers, citation of books, examples.

1 Introduction

Fuzzy neural networks are usually based on neural network architecture [1] with fuzzification of inputs, outputs, weights, or rules that are applied using fuzzy systems [2]. GA is one of the alternatives to gradient techniques (see [3], [4] and [5]) that can be used to develop effective end efficient learning algorithm for fuzzy neural networks.

In [6], the authors proposed a learning algorithm for FFNN with fuzzy inputs and weights called "fuzzy delta rule". This algorithm is derived by replacing fuzzy differentiation with real differentiation, this fact being point out by the authors.

Architecture of multilayer feedforward algebraic neural network for fuzzy input vectors, fuzzy outputs and fuzzy weights is proposed in (see [7] and [8]). The input-output relations in this fuzzy neural network, using *max/min* operators, are defined by the extension principle of Zadeh. In these papers, the authors used only symmetric triangular fuzzy numbers (see [7] and [8]). A cost function for the level sets (α-cuts) of fuzzy outputs and fuzzy target is defined. The fuzzy output from each unit in the fuzzy neural network is computed for level sets of fuzzy inputs, fuzzy weights and fuzzy biases. The learning algorithm, based on the cost function, uses a gradient descent method. A heuristics method, in order to respect the shape of triangular fuzzy weights, during the learning stage is proposed (see [7] and [8]).

The same architecture but with non-symmetric and trapezoidal fuzzy numbers has been used in [9]. The authors proposed and "trail-and-error" algorithm using adaptive updated based on gradient techniques developed in the frame of fuzzy arithmetic [9]. The α-cuts and fuzzy biases are also taken also into account [9]. A learning algorithm to fully connected fuzzy recurrent neural network (FCFRNN) has been developed in [10]. The algorithm is developed for non-symmetric triangular fuzzy numbers and gradient techniques developed in the frame of fuzzy arithmetic. The symmetric fuzzy numbers and crisp numbers are considered particularly cases [10].

Few papers refer to GA for adjusting fuzzy weights. In [11] the authors present a GA mechanism for adaptation of fuzzy weighs as LR-fuzzy numbers for FFNR. The single chromosome includes all the fuzzy weights in binary format. The single point crossover (split point - SP) is applied to entire chromosome that represent the coded binary value of al the triangular fuzzy weights expressed as L-R fuzzy number with n parameters [11].

A structure of fuzzy recurrent neural network with GA learning algorithm is proposed in [12]. All the fuzzy weights are coded in a genome, a single string of bits that represent L-R fuzzy numbers. This string represents an individual,a potential solution of the problem. The connections in the recurrent structure follow a limited connection that is the structure is not fully recurrent structure in the sense of [14]. In both papers (see [11] and [12]) the fitness is the performance index and the objective is to minimize this error.

This paper presents a GA algorithm based on learning mechanism that adjusts the fuzzy weights represented as L-R fuzzy numbers. The fuzzy neural networks computing operations are described in the fuzzy arithmetic framework. The proposed method is applied to two structures: feedforward structure and fully recurrent structure.

The proposed method proved to be better than the existing algorithms in term of number of generations until the optimal solution has been reached.

2 The Fuzzy Arithmetic Framework and the Fuzzy Neuron

In this paper we used the input-output relations fuzzy neural network,using *max-min* operators, defined by the extension principle of Zadeh, and applied in previously papers (see [9] and [10]). All the nodes have a *semilinear (sigmoidal)* transfer function. We denote a triangular fuzzy number as follow:

$$\tilde{X} = (x^L, x^C, x^R) \tag{1}$$

A non-standard algebraic operation between two TFN,$\tilde{A}$ and $\tilde{B}$ is defined, as in (see [9] and [10]), by:

$$\tilde{C} = \tilde{A}\tilde{\Diamond}\tilde{B} = \tilde{X} = (c^L, c^C, c^R) = (min(P), a^C \Diamond b^C, max(P)) \tag{2}$$

$$P = a^L \Diamond b^L, a^L \Diamond b^R, a^R \Diamond b^L, a^R \Diamond b^R \tag{3}$$

The h-level set of a fuzzy number $\tilde{X}$ is defined, as:

$$[\tilde{X_h}] = \{x|\mu_x \geq h, h \in \mathbb{R}\} \quad for \quad 0 < h \leq 1 \tag{4}$$

In the equations (2)-(3), $\Diamond \in \{+, -, \cdot, /\}$ are the operations in classic arithmetic, and $\tilde{\Diamond} \in \{\tilde{+}, \tilde{-}, \tilde{\cdot}, \tilde{/}\}$ are the corresponding modified operations in fuzzy arithmetic, defined according to [7]. The division is restricted to division by a non-zero numbers, that is $\{b^L, b^C, b^R\} \neq 0$ Similar relations can be used to get the α-cuts.

We consider a fuzzy neuron (FN) that perform input-output operations. The FN get an input vector fuzzy signal $\tilde{x} = (\tilde{x_1}, \tilde{x_2}, ..., \tilde{x_N}\}$ nd transfer it after multiplication with fuzzy weights $\tilde{w} = (\tilde{w_1}, \tilde{w_2}, ..., \tilde{w_N}\}$ by activation function to output. The total input of the fuzzy neuron is given by (see [9] and [10]):

$$\tilde{s} = \tilde{w_1}\tilde{\cdot}\tilde{x_1}\tilde{+}\tilde{w_2}\tilde{\cdot}\tilde{x_2}...\tilde{+}\tilde{w_N}\tilde{\cdot}\tilde{x_N} = \sum_{i=1,N}^{fuzz} \tilde{w_i}\tilde{\cdot}\tilde{x_i}\tilde{+}\tilde{b} \tag{5}$$

$$\tilde{y} = f(\tilde{s}) \tag{6}$$

In the equations above the sum is in the fuzzy arithmetic framework (2)-(3) and $\tilde{b}$ is the fuzzy bias. The output is computed by (5) and the extension principle of Zadeh [13]. We used sigmoidal function:

$$f(x) = \frac{1}{1 + e^{-x}} \tag{7}$$

Due to monotonic increasing property of the sigmoidal function *f*, the TFN shape is preserved for the output of the neurons. In section 3 and 4 we will apply these assertions to two types of neural networks architectures: feedforward neural networks [1] and fully connected recurrent neural networks [14].

3 Feed-Forward Fuzzy Neural Networks

There are three basically types of fuzzy neural networks depending on the type of fuzzification of inputs, outputs and weights (including biases): fuzzy weights and crisp inputs, crisp weights and fuzzy inputs and fuzzy weights and fuzzy inputs [11]. In what follows we consider the most complete fuzzification of neural networks: fuzzy inputs, fuzzy weights and fuzzy outputs.

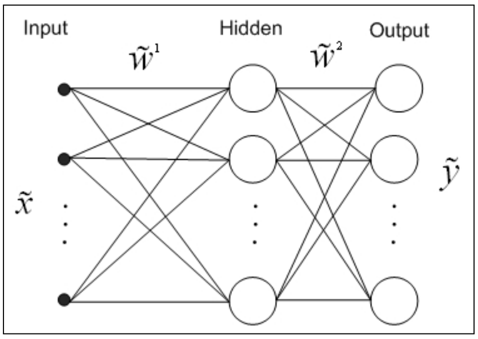


Figure 1: A three layered FFNN

In Fig. 1 we showed an example of FFNN, a three layer FFNN. FFNN propagate the signal layer by layer until the signal reach the output layer. For each *k* layer we have:

$$\tilde{s}_j^k = \sum_{i=1,N_k}^{fuzz} \tilde{w_{ji}^k}\tilde{\cdot}\tilde{x}_i\tilde{+}\tilde{b}_i \tag{8}$$

$$f(x) = f(\tilde{s}_k^j) \tag{9}$$

In the equations above, *j* is the fuzzy neuron from *k* layer and *i*-th is the fuzzy neuron from *k*-1 layer. The training of FFNN is to adapt the network in order to mapping the known inputs to target outputs. The objective of the learning algorithm is to minimize the error measure between the desired outputs and the real outputs. The learning algorithm must adjust the fuzzy weights based of error measure in order to achieve this objective.

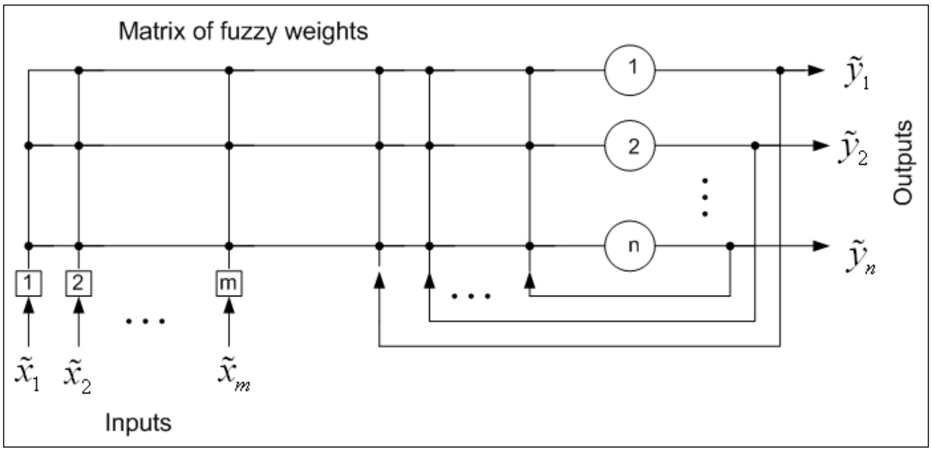


Figure 2: Fully connected FRNN

4 Fuzzy Recurrent Neural Networks

Let the FRNN network have n units and m inputs. Each bias allocation will be seeing as an input line whose value is always $\tilde{1} = (1, 1, 1,)$.

Let the FRNN network have n units and m inputs. Each bias allocation will be seeing as an input line whose value is always $\tilde{1} = (1, 1, 1,)$. We denote by (I, U, T) the set of indices for input units, output units and target units. Lets $\tilde{x}(t)$, $\tilde{y}(t)$ and $\tilde{d}(t)$ denote the inputs, the outputs and the targets (if exists) of the units in RNN at time t, respectively. We denote, similar to [14], a generalized $\tilde{z}(t)$ as follow:

$$\tilde{z}_k = \begin{cases} \tilde{x}_k(t) & \text{if } k \in I \\ \tilde{d}_k(t) & \text{if } k \in T(t) \\ \tilde{y}_k(t) & \text{if } k \in U - T(t) \end{cases} \tag{10}$$

The basic algebraic fuzzy neuron is based on the operations defined in (2)-(5), where the sum above denoted by $\sum^{fuzz}$ is the sense of the algebraic sum (4) and $\tilde{b}_k$ is the bias of the k-th unit.

$$\tilde{s}_k(t) = \sum_{r \in U \cup I}^{fuzz} \tilde{w}_{kr} \tilde{\cdot} \tilde{z}_r(t) = \tilde{w}_{k1} \tilde{\cdot} \tilde{z}_1 \tilde{+} ... \tilde{+} \tilde{w}_{k,m+n} \tilde{\cdot} \tilde{z}_{m+n}(t) \tilde{+} \tilde{b} \tag{11}$$

$$\tilde{y}_k(t+1) = f(\tilde{s}_k(t)) = f(s_k^L, s_k^C, s_k^R) \tag{12}$$

The missing connection in the RAFNN architecture is simply represented by a zero weight value (fuzzy number zero).

5 GA with Locally Crossover (GALC)

GA are optimization techniques based on principles of mechanism of natural evolution (see [3] and [4]). GA are working on possible space of solutions (usually codified by chromosomes) in order to find the best candidate suitable for a particularly problem. The fitness is a measure of performance of the individual in order to achieve to desired objectives. The objective function can be the maximization of fitness or minimization of fitness.

Two problems arise from GA usage for solving the optimization problems: the choice of the fitness function and the codification the values of parameters that must optimized into an individual chromosome. The range of values that are part of this codification play an important role related to both convergence speed of GA and the accuracy of the results.

In our approach we will use the minimization of objective function that is a measure of output errors related to fuzzy numbers. This function is usually named the fitness function.

The general fitness function can be defined based distance among desired fuzzy numbers and actual fuzzy numbers at outputs [11]. However, in our application we will use a more practical measure based on Hamming distance. By this objective function we estimate more intuitive the difference between desired and real outputs.

$$D_{fitness} = min \sum_{i=1}^{K} (|\tilde{y}_i^d - \tilde{y}_i|) \tag{13}$$

GA uses binary encoded in order to represent genes or chromosomes. Each binary value (0 or 1) is named allele [3]. The value of solutions is mapped as binary string in the process of encoding using linear of nonlinear functions. We used the representation of TFN as LR-type fuzzy numbers with n parameters [13]. Each parameter is coded as binary string of a specified length.

$$\tilde{A} = (m, dL, dR) \tag{14}$$

In the case of no α-cuts, each fuzzy weight is represented by three chromosomes corresponding to central value w^k_{jiC} , and L, R values that represents the left and the right spread are denoted by dw^k_{jiL} and dw^k_{jiR} (Fig. 3). The values are coded into binary string in a predefined range.

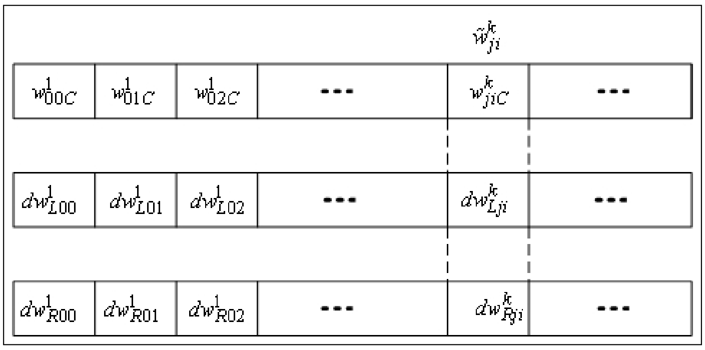


Figure 3: Chromosomes of one individual

In the case of FFNN, $\tilde{w}^k_{ji}$ is fuzzy connection weight between the j-th neuron from k layer with i-th neuron from k-1 layer.

Each individual has three chromosomes that represent the strings for the three types of values. We must remark that in TFN the order must be preserved:

$$w^k_{ji} - dw^k_{jiL} \leq w^k_{ji} \leq w^k_{ji} + dw^k_{jiR}, \quad dw^k_{jiL} \geq 0, dw^k_{jiR} \geq 0 \tag{15}$$

The selection process has two stages. The first stage is selection of two parents for crossover using a known schema: tournament, roulette or stochastic sampling. After this global selection, we perform a locally selection inside of the chromosome. This selection is inspired by the fact observed in crisp feedforward neural networks that not all the weights have the same contribution to forward propagation of the signal. Usually, the weights from k layer has a different contribution than the weights from k+1 layer. Conversely, it is natural to suppose that some parameters give a more contribution to best solution than others, so a fine locally adjustment can improve the global solution.

The locally selection selects the weights, the locations in the chromosome where the corresponded weight is coded in binary string for locally crossover (Fig 4).

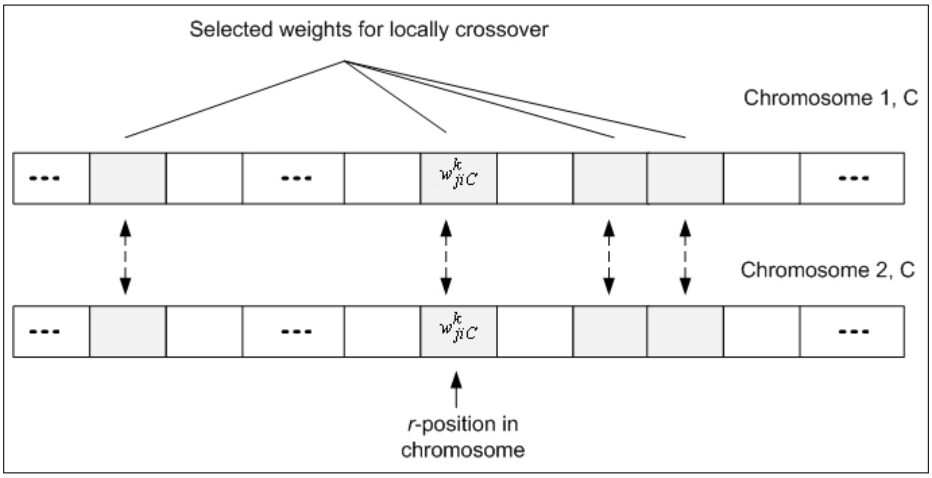


Figure 4: Selection of genes (weights) for locally crossover for C value

The same selected location are taken into account in one step for all three values, as in (13). The representation is given in Fig. 5.

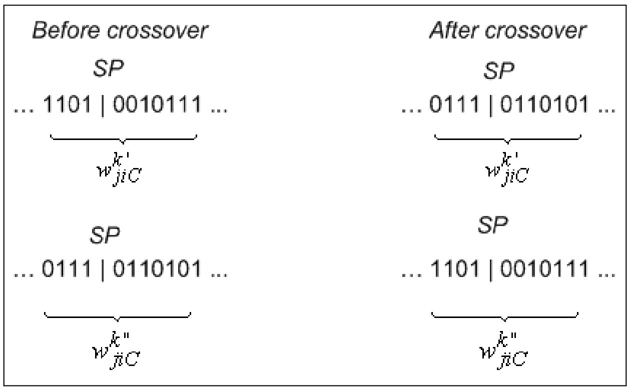


Figure 5: Selection of genes (weights) for locally crossover for dL, dR values

The crossover operation pick the selected genes and mate them in order to produce two offspring genes. The split point (*SP*) is chosen randomly, the same for both parents. The example from Fig. 6 illustrate the locally crossover. The same *SP* is applied to corresponding gene that code the left and right spread of the selected weights.

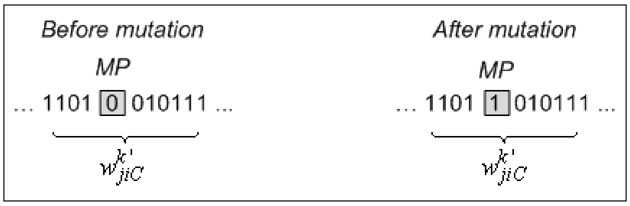


Figure 6: Locally crossover between selected genes

Mutation is made globally al level of entire chromosomes. The mutation pick and bit (*allele*) with a probability p_m and flip this value to 1 if the value is 0 and to 0 if the value is 1. Mutation implements a random search at global level. By p_m, the level of search can be modified, the greater is p_m, the level of search in the solution space is greater.

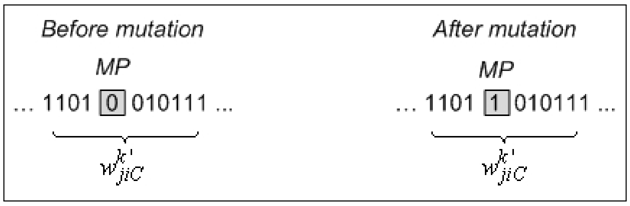


Figure 7: Global mutation

The GALC algorithm is based on the practical following observation. Due to nonlinear aspect of the transfer function of the neuron, the changes in some weights can have a greater contribution to result than the changes in the other weights. Also, the impact of this changes for FFNN can decrease or increase in the forward process (Fig. 8).

GALC propose to use this observation in order to "encourage" the more contributing weights to be selected in order to do crossover and evolve faster meanwhile the rest of weights will evolve slowly, based only on mutations. For the same dx ($dx_1 = dx_2$), we can have $dy_2 > dy_1$ in same cases with one or two magnitude order.

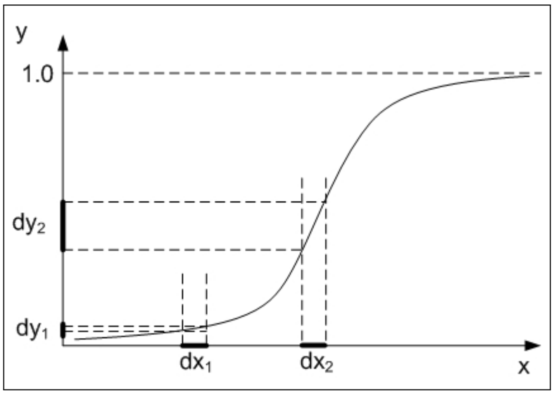


Figure 8: The influence of function of activation over quantitative adaptation of the weight

Let denote by *NN* the total number of neurons in the FFNN. The proposed algorithm based on basic GA procedures is summarized below.

Step 1: Linear mapping of the solution space (fuzzy weights)into chromosomes (binary string) that represent individuals of the population *P*.

Step 2: Initialize the population P with random solutions (uniform distribution). The population has *N* individuals that represents 3**N* set of chromosomes for TFN with no fuzzy bias and 3**N* + 3**NN* set of chromosomes for TFN with fuzzy bias.

Step 3: Evaluate the fitness of the population *P*.

Step 4: Evaluate the stop criteria. If the stop criteria is fulfilled, go to *Step 10*.

Step 5: Store the best individual of the current generation and the best individual of all the generations.

Step 6: Generate a new population using selection operator according to selection schema based on fitness values of the chromosomes.

Step 7: Random selection of genes for locally crossover. The selected genes are a percent pr from total number of the genes that represent a chromosome.

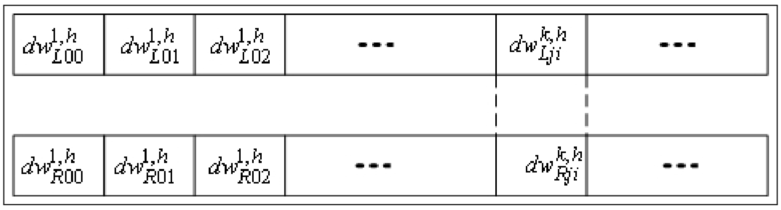
Step 8: Locally crossover for selected genes and creation of new generation of the population. The corresponding genes are mated with probability pc and the results offspring replace the parents in the new population.

Step 9: Global mutation over chromosomes with random probability p_m.

Step 10: Map the best individual of all the generations into solution, the fuzzy weights of the FFNN.

The stop criterion can have different forms. One of the most common is a predefined number of generations. Another one can be the stop of the process, before a maximum number of generations if no significant improvement has been made in the best individual. In our application we used the first criterion.

The GALC algorithm can be easily extended to α-cuts. For each h-level we must allocate 2 chromosomes that correspond to dL and dR values al level h (Fig. 9). The central value is unchanged from one level to next level. We start with the base level (h=0) where we adapt the L, C and R values using L-R type representation (12). Next we consider the C value and we must adapt the left and right spread at the level h=1,2,3,... n.



$dw_{L00}^{1,h}$	$dw_{L01}^{1,h}$	$dw_{L02}^{1,h}$	---	$dw_{Lji}^{k,h}$	---
$dw_{R00}^{1,h}$	$dw_{R01}^{1,h}$	$dw_{R02}^{1,h}$	---	$dw_{Rji}^{k,h}$	---

Figure 9: Chromosomes of one individual for h-level

Because of *max/min* operators and multiplication operation, the shape of membership for fuzzy weights (and biases) is a curved triangle. Restrictions must be made in order to avoid non desired occurrence in the weights adaptation process.

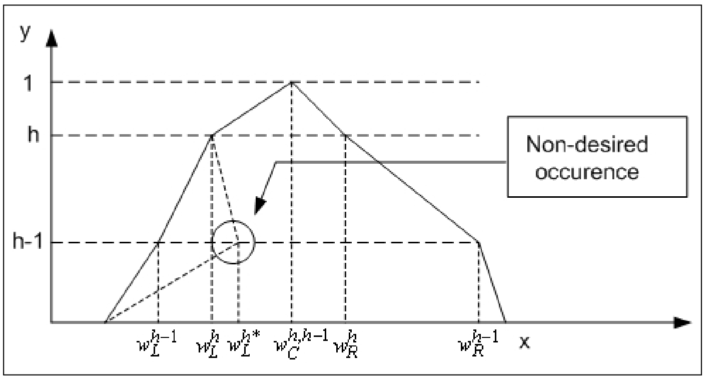


Figure 10: A non-permitted occurrence in the weights adaptation process

The restrictions are:

$$dw_{jiL}^{k,h} \leq dw_{jiL}^{k,h-1}, \quad dw_{jiR}^{k,h} \leq dw_{jiR}^{k,h-1} \tag{16}$$

These restrictions are included in the algorithm in the following way. If the after the crossover, the children accomplish the rules (15) the local crossover operation is validated and the children are selected in the next generation depending on the fitting. If the children doesn't accomplish the rules (15) the local crossover operation is canceled and the selected weights remain unchanged.

6 Applications

6.1 A mapping of non-symmetric triangular fuzzy numbers with FFNN

We apply the proposed method to approximate realisation of non-linear mapping of fuzzy numbers. The TFN from input space have the bases inside the interval [0, 1] and are mapping to TFN which have the bases in [0,+1]. The FFNN has 3 inputs, 6 neurons in the hidden layer and 3 neurons in the output layer.

The FFNNs fuzzy error can be defined as a distance between fuzzy desired output and the fuzzy real output [15]. Because we are interested in accuracy of outputs for all the points that represents the fuzzy number, we express the error measure:

$$e_i^{L,C,R} = \left| d_i^{L,C,R} - y_i^{L,C,R} \right| \tag{17}$$

$$J_{total}(t) = \sum_{q=L,C,R;i=1\ldots N} e_i^q \tag{18}$$

The inputs and the target (desired outputs are):

$\tilde{x} = (0.2,0.1,0.1),(0.3,0.1,0.2),(0.6,0.2,0.1)$
$\tilde{d} = (0.4,0.2,0.2),(0.8,0.2,0.1)$

We generate an initial population of random binary strings that represent the chromosomes. The population is set to P=100 individuals, $p_c = 0.7$, $p_m = 0.3$, $gen = 1000$ (the maximum number of generations). We set $p_r = 20\%$, that is from $6 \times 3 + 6\times\ 2 = 30$ weights, and at each generation we select random a number of $0.2{*}30 = 6$ weights.

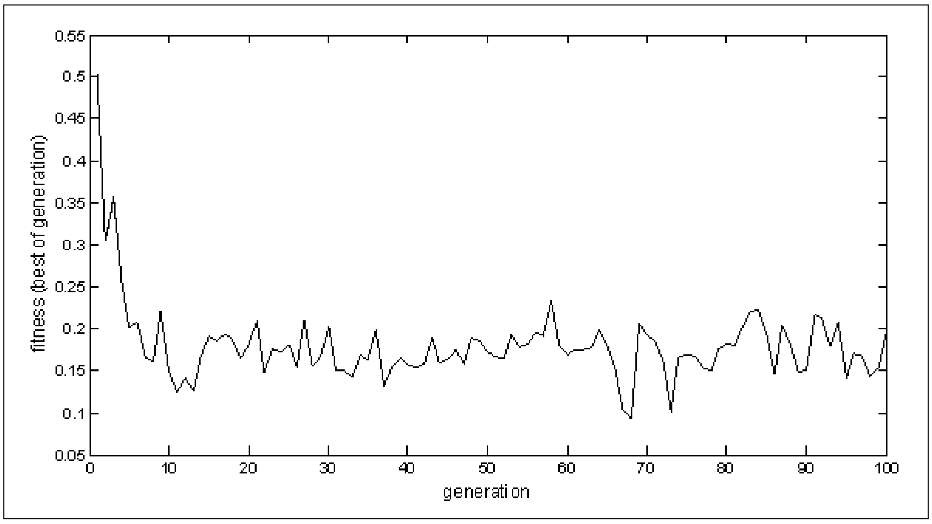


Figure 11: The error evolution (FFNN example)

The best result is obtained in *gen*= 68, when $J = 0.0927$ (Fig. 11). The best fitness individual has the lowest value of fitness (we used the best individual - minimum fitness approach). The desired and the real output values are:

$\tilde{d} = (0.4,0.2,0.2),(0.8,0.2,0.1)$
$\tilde{x} = (0.4055,0.1945,0.2176),(0.7714,0.1636,0.1119)$

We made experiments with α-cuts, also. In our experiment we used three α-cuts, at levels h=0, h= 0.33, and h = 0.66. The experimental results for one of the fuzzy weights (k=2, j =1, i=2) and the levels of the α-cuts mentioned above is showed in Fig. 12.

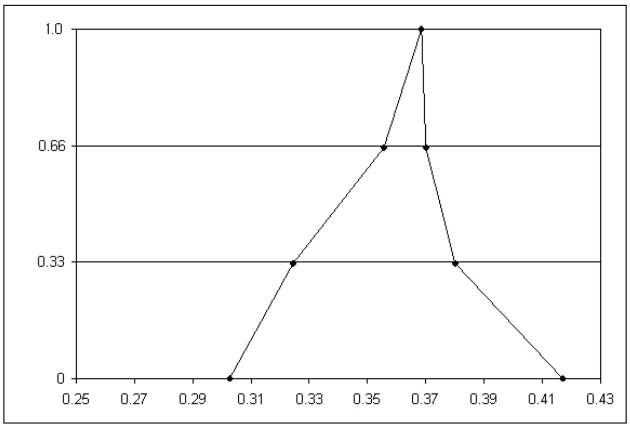


Figure 12: The fuzzy weight $\tilde{w}_{12}^2$ with three α-cuts)

6.2 FRNN - Learning of defined dynamic

The algorithm designed to train an arbitrary network dynamics can be tested using an interesting class of behaviors, the oscillations [14]. A pair of logistic units (Fig. 13) was used to learn a trajectory of a TFN, composed by three sine waves with the same frequencies and the same phase but different range corresponding to L, C and R values of the TFN (Fig. 14).

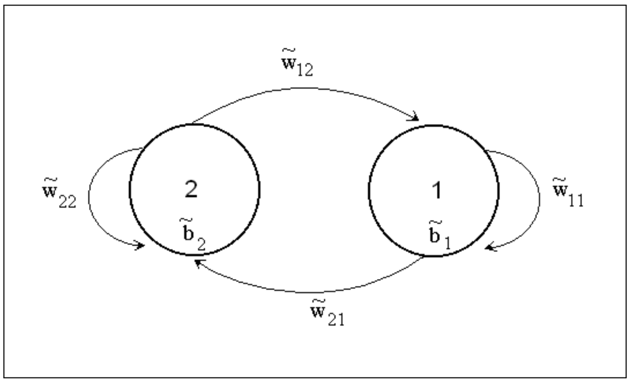


Figure 13: Fuzzy sine wave network.

The overall error measure at each sample time t is given by:

$$J(t) = \sum_{k \in U} \left[\left[\left| e_k^L(t) \right| + \left| e_k^R(t) \right| \right] + \left| e_k^C(t) \right| \right] \tag{19}$$

$$J_{total}^{fuzz}(t_0, t_1) = \begin{cases} d_k^{L,C,R}(t) - y_k^{L,C,R}(t) & \text{if } k \in T(t) \\ 0 \quad otherwise \end{cases} \tag{20}$$

The network performance measure over trajectory is defined by:

$$J_{total}^{fuzz}(t_0, t_1) = \sum_{t=t_0+1}^{t_1} J(t) \tag{21}$$

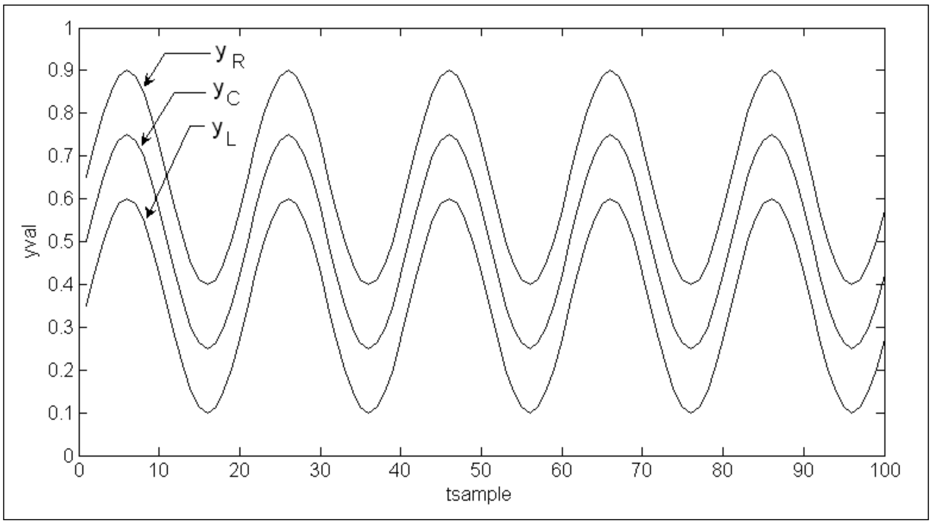


Figure 14: The training sine waves (targets).

In our particular case we consider a complete trajectory a sinus period (2π). The best individual is obtained after 176 generation after 10 starting new generations and epochs.

Usually, the stable solution is obtained in less than 500 generation with uniform random distribution of initial population made from 100 individuals.

Table 1: RAFNN parameter for sine waves

	L	C	R
$\tilde{w}_{11}$	0.3026	0.3687	0.4169
$\tilde{w}_{12}$	-0.4405	-0.2509	-0.0286
$\tilde{w}_{21}$	-0.2506	-0.0076	-0.0035
$\tilde{w}_{22}$	0.2214	0.4054	0.5207
$\tilde{b}_1$	-0.3521	-0.3028	-0.2499
$\tilde{b}_2$	-0.1499	-0.0085	0.0772

Stable, the sine like oscillation are obtained for sine frequencies above 20 network ticks per cycle with the 20,000 ticks or less. The output of the unit 1 in the absence of a teacher imposed to unit 2, after stable oscillation had been established is shown in Fig. 15

As we are the expectations, taking into account the results from [14], the frequency of the free-running logistic is around 8% lower than the trained frequency. The amplitude is smaller and it is diminishing in time.

This example proved that the FRNN is able to learn a dynamics that is the sinusoidal waves. The output of the unit 1 has at each step a triangular fuzzy number, and values y^L, y^C and y^R have a sinusoidal shape. It is clear from results that the output error is important, but out target was to learn predefined dynamic using minimization of error. The algorithm stopped when the performance not improve during the last S generations (in our case, S=3).

7 Conclusions

We proposed to use a novel method and a new learning algorithm, GALC for fuzzy neural networks with feedforward and fully recurrent architecture. The method and a learning algorithm proved by theoretical and experimental approach to be effective with acceptable stable solution.

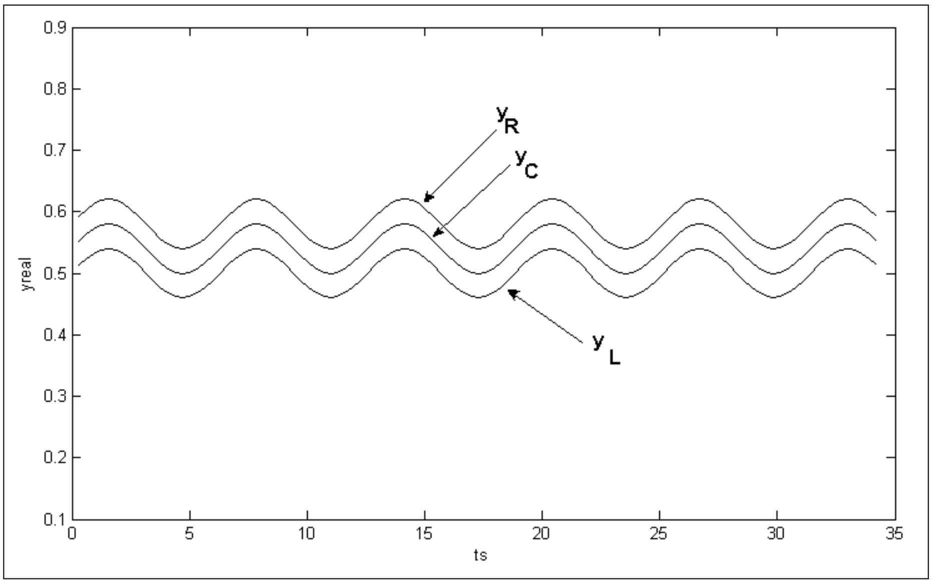


Figure 15: The output of the unit 1.

The solution is usually obtained in a shorter time (in terms of number of generation) than the actually GA based learning algorithms. Moreover, the algorithm is effective also for fully fuzzy recurrent neural networks as they has been defined in section 4.

Different from (see [11] and [12]) we proposed a complete fuzzification of the neural network in the sense that the sum of the fuzzy weighted inputs are also in the fuzzy arithmetic framework (see [11] and [12]).

The proposed method offer a possibility to work with long strings. It is know that in the case of very long chromosome, the global crossover can slow considerably the algorithm convergence. That is a very large number of generation is required in order to achieve the global solutions. The proposed method that start with a larger number of individual in the initial population and selective locally adaptation could improve the performance of convergence by pressure of these parameters that has a greater importance in the best solution.

Bibliography

[1] S. Haykin, *Neural Networks: A Comprehensive Foundation*, Prentice Hall, 1998.

[2] D.J. Dubois, H. Prade, *Fuzzy Sets and Systems: Theory and Applications*, Academic Press, 1980.

[3] D. E. Goldberg, *Genetic Algorithms in Search, Optimization, and Machine Learning*, Addison-Wesley Professional, 1989.

[4] J. R .Koza, *Genetic Programming: On the Programming of Computers by Means of Natural Selection*, The MIT Press, 1992.

[5] O. Cordon, V. Herrera, M. Lozano, On the Combination of Fuzzy Logic and Evolutionary Computation: a short review and Bibliography, in: W. Pedrycz (Ed.), *Evolutionary Computation*, Kluwer Academic Publishers, Dordrecht, pp.33-56, 1997.

[6] Y. Hayashi et. al., Fuzzy Control Rules in Convex Optimization, *Fuzzy Neural Networks with Fuzzy Signals and Weights IJCNN'92*, Vol. 2, pp. 165-195, 1992.

[7] H. Ishibuchi, K. Kwon, H.Tanaka, A learning algorithm on fuzzy neural networks with triangular fuzzy weights, *Fuzzy Sets and Systems*, Vol. 72, No. 3, pp. 257-264, 1995.

[8] H. Ishibuchi, R. Fujioka, H.Tanaka, An Architecture of Neural Networks for Input Vectors of Fuzzy Numbers, *Proc. FUZZ-IEEE '92*, San Diego, USA, pp. 643-650, 1992.

[9] H.N. Teodorescu, D. Arotaritei, E. Lopez Gonzales, A General Trail-and-Error Algorithm for Algebraic Fuzzy Neural Networks, *Proceedings of the Fourth European Congress on Intelligent Techniques and Soft Computing*, Aachen, Germany, September 2-5, Vol. 1, pp. 8-12, 1996.

[10] D. Arotaritei, Recurrent Algebraic Fuzzy Neural Networks based on Fuzzy Numbers, *Joint 9th IFSA World Congress and 20th NAFIPS International Conference*, Vol. 5, pp. 2676 - 2680, 2001.

[11] R.A. Aliev, B. Fazlollahi, R.M. Vahidov, Genetic algorithm-based learning of fuzzy neural network, Part 1: feed-forward fuzzy neural networks, *Fuzzy Sets and Systems*, Vol. 118, Issue 2, pp. 351-358, 2001.

[12] R. A. Aliev, R. R. Aliev, B. G. Guirimov, K. Uyar, Recurrent Fuzzy Neural Network Based System for Battery Charging, *Lecture Notes in Computer Science*, Vol. 4492, pp. 307-316, 2007.

[13] L.A. Zadeh, Fuzzy Sets, *Inform. Control 8*, pp. 338-353, 1965.

[14] R.J. Williams, D. Zipser, A learning algorithm for continually running fully recurrent neural networks, *Neural Computation*, Vol. 1, Issue 2, pp. 270-280, 1989.

[15] C. Chakraborty, D. Chakraborty, A theoretical development on a fuzzy distance measure for fuzzy numbers, *Mathematical and Computer Modelling*, Nr. 43, pp. 254-261, 2006.

Int. J. of Computers, Communications & Control, ISSN 1841-9836, E-ISSN 1841-9844
Vol. VI (2011), No. 1 (March), pp. 21-32

An Automatic Face Detection System for RGB Images

T. Barbu

Tudor Barbu
Institute of Computer Science, Romanian Academy, Iaşi branch, Iaşi, Romania
E-mail: tudbar@iit.tuiasi.ro, www.iit.tuiasi.ro/ tudbar

Abstract: We propose a robust face detection approach that works for digital color images. Our automatic detection method is based on image skin regions, therefore a skin-based segmentation of RGB images is provided first. Then, we decide for each skin region if it represents a human face or not, using a set of candidate criteria, an edge detection process, a correlation based technique and a threshold-based method. A high face detection rate is obtained using the proposed method.

Keywords: color image, color space, RGB, HSV, skin region, face detection, cross-correlation coefficient, edge detection, template matching, threshold.

1 Introduction

This paper approaches an important digital image analysis domain. *Face detection* represents a computer technology that determines the locations and sizes of human faces in arbitrary digital images.

Face detection can be regarded as a specific case of *object-class detection.* In object-class detection, the task is to find the positions and sizes of all objects in an image that belong to a given class [1]. While being a sub-domain of the object detection field, face detection represents a generalization of *face localization.* In face localization, the task is to find the location and size of a given input image, while in face detection one does not have any information about the human faces [2].

The most important application area of face detection is *biometrics.* Face finding is often considered the first step of the *face recognition process* [3, 4]. Thus, most facial recognition systems, and the more complex biometric systems including face recognition components, use face detection techniques. *Video surveillance* represents another important application domain of face detection.

A robust face detection task consists of identifying and locating all the faces in an image, regardless of their position, scale, pose, orientation and illumination [2, 3]. Early face-detection techniques focused on the detection of frontal human faces only, and did not consider the *rotation* problem. The newer methods attempt to solve the more general and difficult problem of multi-view face detection. These algorithms take into consideration the two types of face rotation: *pose*, representing the out-of-plane rotation, and *orientation*, representing the in-plane rotation [2].

Also, there are several factors which could transform the human face finding process into a difficult task, such as: the *structural components* (presence or absence of beards, moustaches, glasses or other elements), the *facial expressions* (smiling, laughing, crying and others), *occlusions* (the faces can be occluded by other objects) and *imaging conditions* (lighting, camera characteristics) [2]. A robust face detection approach must take into consideration the presence of these factors.

There are several known categories of face detection approaches: *knowledge-based* techniques [5], *feature-based* methods [2,6,7], *appearance-based* approaches [8–13] *template matching* methods

[14]. The knowledge-based methods encode human knowledge of what constitutes a typical face, usually the relationships between facial features. A face is represented using a set of human-coded rules. These rules are then used to guide the face search process.

The advantages of the knowledge-based techniques are: the easy rules to describe the face features and their relationships, and the good results obtained for face localization in uncluttered background. Their disadvantages are: the difficulty to translate the human knowledge in rules precisely and the difficulty to extend these methods to detect faces in different poses, respectively [5].

The feature based approaches aim to detect invariant face features. These are structural features of a face that exist even when the pose, viewpoint or lighting conditions vary. We could mention here the *Random Graph Matching* based approaches [6] and the *Feature Grouping* techniques [7]. The main advantage of the feature oriented face detection approaches consists in the fact that these features are invariant to rotation changes. Their main drawback is the difficulty to located facial features in a complex background.

Appearance-based techniques train a classifier using various examples of faces. The classifiers which can be used in the training process include: Neural Networks (Multilayer Perceptrons) [8], Hidden Markov Models [9], Bayes classifiers [10], Support Vector Machines (SVM) [11], Sparse Network of Winnows (SNoW) [12], Principal Component Analysis (PCA) [3] and Boosting algorithms (Ada-Boost) [13].

The template matching based techniques use stored face templates [2, 14]. Usually, these approaches use correlation operations to locate faces in images [15]. The templates are handcoded, not learned. Also, these templates have to be created for different poses.

We propose a template matching based face detection method in this paper, too. Our detection technique works for RGB color images only and it is based on the skin regions of the image. Thus, in the first stage, our approach performs a skin segmentation process, extracting the human skin regions from the analyzed image. The proposed skin detection technique is described in the next section.

Next, the technique identifies the human faces, by performing an analysis of the previously obtained skin segments. The face identification method is provided in the third section. In the fourth section, the experiments performed using the proposed human face detection system, are discussed. The paper ends with a conclusions section and the references section.

2 A Skin Detection Approach for RGB Images

Human skin color is proven to represent a very useful face detection and localization tool [2, 14, 16, 17]. A skin-based face finding approach identifies the skin regions of the image, then determine those of them which represent human faces. Besides face detection, there exist other important application areas of skin detection, such as image content filtering and finding illegal internet content [16], content-aware video compression or image color balancing.

Many skin color localization techniques have been developed in recent years. A robust and very known skin finding method is the algorithm proposed by Fleck and Forsyth in 1996, that uses a skin filter [16].

We are interested in color images only and do not perform skin and face detection in grayscale images. Obviously, the color images are usually in the RGB format. While it is one of the most used color spaces for processing and storing of digital image data, RGB is not a favorable choice for skin color analysis, because of the high correlation of its three channels and the mixing of luminance and chrominance data [17]. For this reason, most skin segmentation algorithms work with other color spaces, such as the *normalized* RGB, HSV (and other Hue saturated based spaces) and $\mathrm{YC_rC_b}$ formats [2, 17].

We propose a skin detection technique using the HSV and $\mathrm{YC_rC_b}$ color spaces. First, a denoising process should be performed on the input RGB image, I. Usually, these images are affected by Gaussian noise. Therefore, a 2-D Gaussian smoothing filter has to be applied to them, to remove detail and noise [18].

Then, the smoothed image is converted into the Hue Saturation Value format, by computing the three components using the known conversion equations. We obtain the components, H, S and V, as three matrices with coefficients in the $[0, 1]$ interval. We are interested mainly in the hue value, H.

The $\mathrm{YC_rC_b}$ color model represents a family of color spaces [17]. In fact, it is not an absolute color space, but a way of encoding the RGB information. In this format, Y represents the luminance, while C_r and C_b are the blue-difference and red-difference chroma components. These three components of the color space are computed as linear combinations of R, G and B components of the image.

Thus, the computation formulas of the chroma components, C_r and C_b, have the general form $\alpha \cdot R + \beta \cdot G + \gamma \cdot B + 128$, where coefficients $\alpha, \beta, \gamma \in [-0.5, 0.5]$. We choose empirically some proper values for these coefficients and get the following components:

$$\begin{cases} C_r &= 0.15 \cdot R - 0.3 \cdot G + 0.45 \cdot B + 128 \\ C_b &= 0.45 \cdot R - 0.35 \cdot G - 0.07 \cdot B + 128 \end{cases} \tag{1}$$

Then, we apply a set of restrictions on these two components and on the hue, to identify the skin regions. Thus, we have determined a skin related interval for each component. In our approach, each pixel of the image I belongs to a human skin segment if the corresponding values in C_r, C_b and H are situated in those intervals.

We create a binary image Sk, having the same size as I, whose white regions correspond to the skin segments. The proposed skin segmentation process is modeled by the following relation:

$$Sk = \begin{cases} 1, & \textit{if}\, C_r(i, ju) \in [150, 165] \propto C_b(i, j) \in [145, 190] \propto H(i, ju) \in [0.02, 0.1] \\ 0, & \textit{otherwise} \end{cases} \tag{2}$$

where $i, j \in [1, M]$ and $j \in [1, N]$, I representing an $[M \times N]$ image.

The connected components from image Sk, computed by (2), represent the detected skin regions. The proposed detection method provides good results, although some skin identification errors could appear. That means some non-skin image regions could be detected as skin segments, but this fact will not affect the final goal, human face detection. Therefore, we are satisfied with the obtained skin finding results.

In Figure 1 (a), there is displayed an RGB image depicting human persons. The result of the HSV conversion is displayed in Figure 1 (b).

The skin detection process is performed by applying the equations (1) and (2). The resulted skin regions are those depicted in Figure 2.

3 A Face Finding Technique

The skin regions detected in the previous section are used in the human face identification process. For each skin segment we have to decide if it represents a face, or it is a non-facial skin region. We will propose an automatic template matching scheme for face detection. Before applying the matching procedure, our face finding approach performs several necessary pre-processing steps.

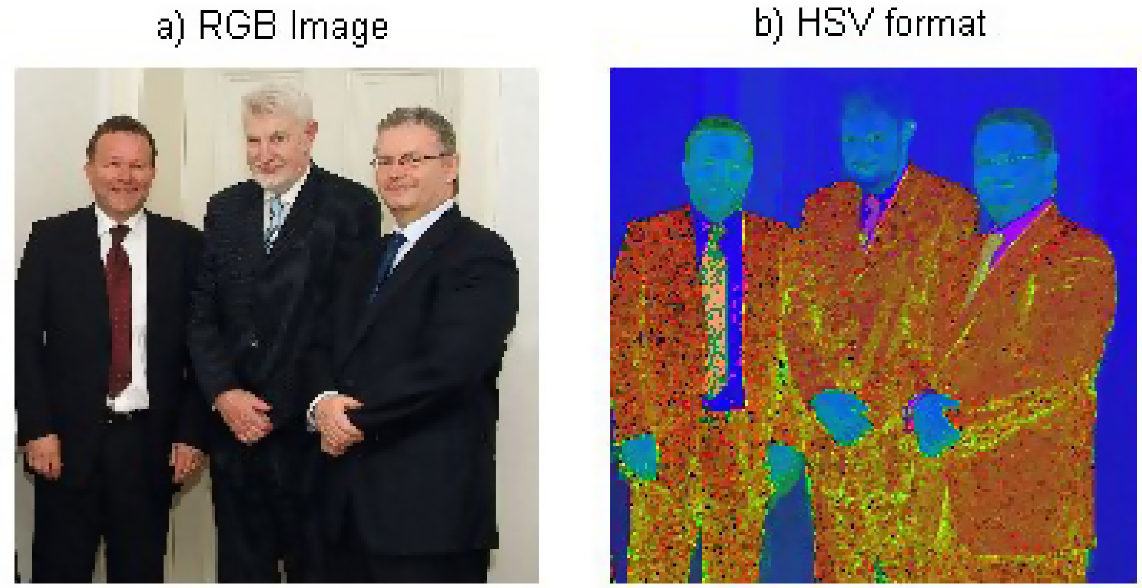


Figure 1: Digital color image conversion: RGB to HSV

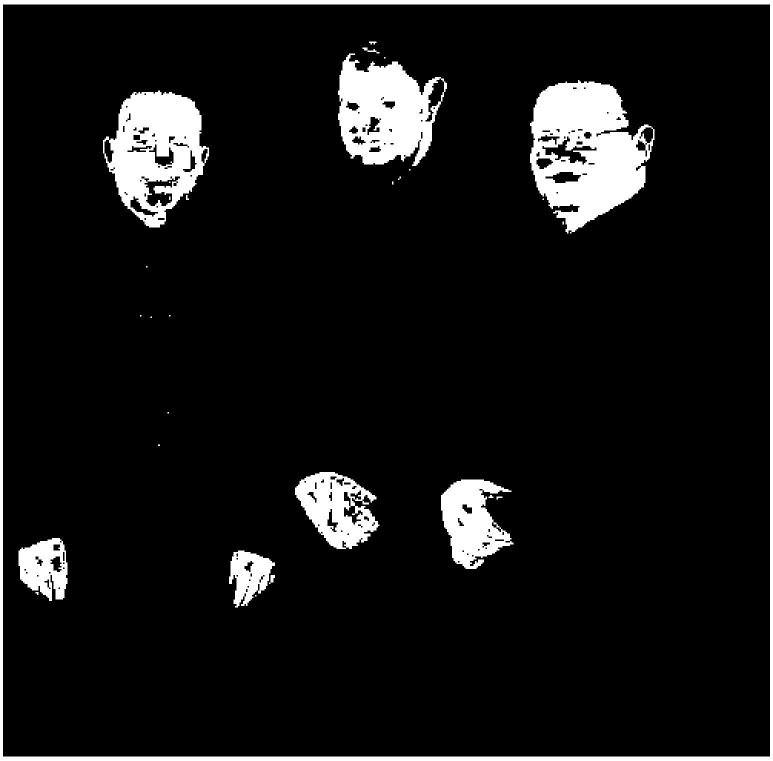


Figure 2: Skin detection result

3.1 Skin region pre-processing

A task we try to solve is the separation of human faces from adiacent or occluding skin regions. Our detection task cannot identify properly the faces which are occluded by other skinlike *objects* in the images. Usually, this situation appears in group photos, like that displayed in Figure 3 (a).

Therefore, we provide a separation technique involving some morphological operations [19], performed on the corresponding binary image, Sk. Thus, we will apply two successive *erosions* on it. First, the binary image is eroded with a structuring element L, representing a vertical line, having a length of 5 pixels:

$$Sk' = Sk \ominus L = \bigcap_{\ell \in L} Sk_{-\ell} \qquad (3)$$

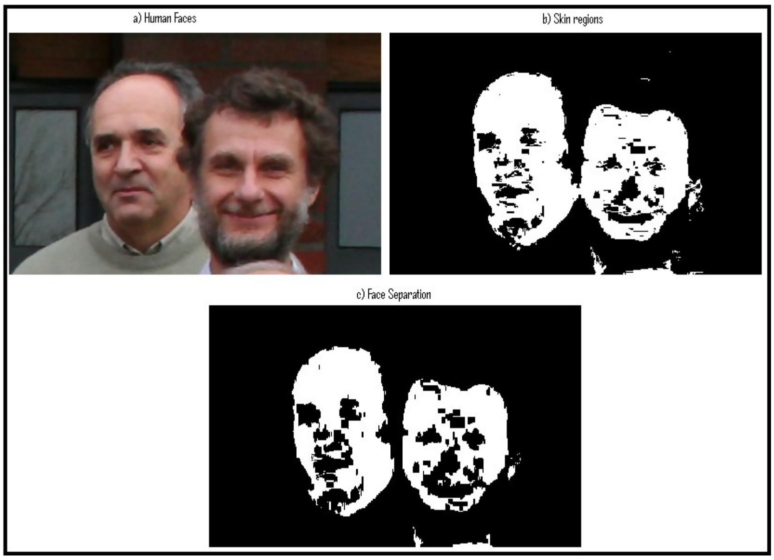


Figure 3: Face Separation

Then, another erosion operation is performed on Sk', using a structuring element Sq, representing a small square area (for example containing a single pixel):

$$Sk'' = Sk' \ominus Sq = \bigcap_{p \in Sq} Sl'_{-p} \qquad (4)$$

In the figure above, one can see a skin separation example using the proposed method. In Figure 3 (b), the skin segments corresponding to the faces from Figure 3 (a) are conjoined into a single region. The result of the morphology-based process, given by the relations (3) and (4), is displayed on Figure 3 (c). The two greatest skin regions are clearly separated in the final binary image, Sk''. Obviously, there could be situations, provided by big occlusions for example, when the face separation is not possible in the binary image.

The binary image Sk'' contains a set of skin segments, which represent connected sequences of white pixels. Let this set of regions be $\{S_1, ..., S_n\}$. Now, we have to decide which of these regions could qualify as *face candidates* for the template matching process. So, we have established a set of candidate criteria for S_i segments.

First condition is size related. The skin region set is usually very large because of the many small white regions which could be present in image Sk''. We have decided to not take into consideration these small area white spots, because they cannot represent serious face candidates.

It is still possible to exist very small faces in an image, like those in a very large crowd, but we consider them irrelevant and do not try to detect them. So, if a white region area (the number of pixels) is below a given threshold value, that region is labeled as a non-facial segment.

Another condition is related to the shape of the skin regions. Obviously, a face region should have a rectangular-like or ellipse-like shape. Thus, a rigorous shape analysis can determine which skin segments cannot represent human faces.

We propose a less complex approach to this task. A connected component S_i has to be rejected as non-face, having a non-facial shape, if its *solidity*, representing the ratio between its area and its *bounding box* area is below an established threshold. A facial skin-region is usually characterized by a high ratio, close to 1, but the solidity value may become lower because the region's area is affected by the presence of many black holes in the face region. These holes could represent human face components, such as eyes, eyebrows, mouth, nose, ears and wrinkles, or some skin detection errors. Therefore, we perform a black hole filling process on the binary image Sk'', first, then compute and use the areas of the filled S_i segments.

Also, a human face is characterized by some limits of its *width to height ratio*. None of the two dimensions of a face, the width and the height, can be *much* larger than the other. For this reason, we set another condition, requiring the width to height ratio of the face candidates to be restricted to a certain interval. The ratio between the width of the region's bounding box and the height of the bounding box should be between two properly chosen threshold values.

The white regions satisfying the proposed restrictions represent the face candidates. For each S_i, $i \in [1, n]$, the described face candidate identification process is formally expressed as follows:

$$Area(S_i) \geq T_i \alpha \frac{Area(Fill(S_i))}{Area(Box(S_i))} \geq T_2 \alpha \frac{Width(Box(S_i))}{Height(Box(S_i))} \in [T_3, T_4] \Longrightarrow S_i = candidate \qquad (5)$$

where $Area(\ )$ computes the number of white pixels of the region received as argument, $Fill(\ )$ performs the filling process, $Box(\ )$ returns the bounding rectangle, $Width(\ )$ and $Height(\ )$ returning the dimensions of a rectangle. We have considered the following proper values for the thresholds in equation (5): the area threshold $T_1 = 130$, the solidity threshold $T_2 = 0.65$, and the width to height thresholds $T_3 = 0.6$ and $T_4 = 1.8$.

A face candidate identification example is described in Figure 4. In the RGB image from Figure 4 (a) one can see a boy flexing his muscles. Figure 4 (b) represents the corresponding binary image resulted after skin detection, erosion operations, small region removing and hole filling process. The bounding boxes of the three remaining skin regions are depicted in pictures c), d) and e). The one representing the right arm is rejected because of its low solidity percent (meaning a wrong shape), the skin segment of the left arm is rejected because a wrong width to height ratio, and the one representing the skin of head and neck is accepted as a right face candidate.

3.2 Template matching process

In its next stage, our facial detection approach determines which of the face candidates represent human faces. First, one converts the denoised RGB image I into a $2D$ grayscale form, let it be I'. If there is a set of K face candidates, where $K \leq n$, then we determine the set of the sub-images of I' corresponding to the bounding boxes of these candidates.

The face detection process can be affected by the head hairline of the persons and by the skin zone corresponding to the neck and the upper chest. Therefore, a narrow upper zone and a narrow bottom zone from each image are removed. In our tests, the height of each removed zone represents one eleventh of the bounding box height.

Let the set of the truncated *skin images* be $\{I_1, ..., I_K\}$, where $I_i \subset I'$, $\forall i \leq K$. Then, we perform a correlation-based template matching process on this set. Our template-based approach

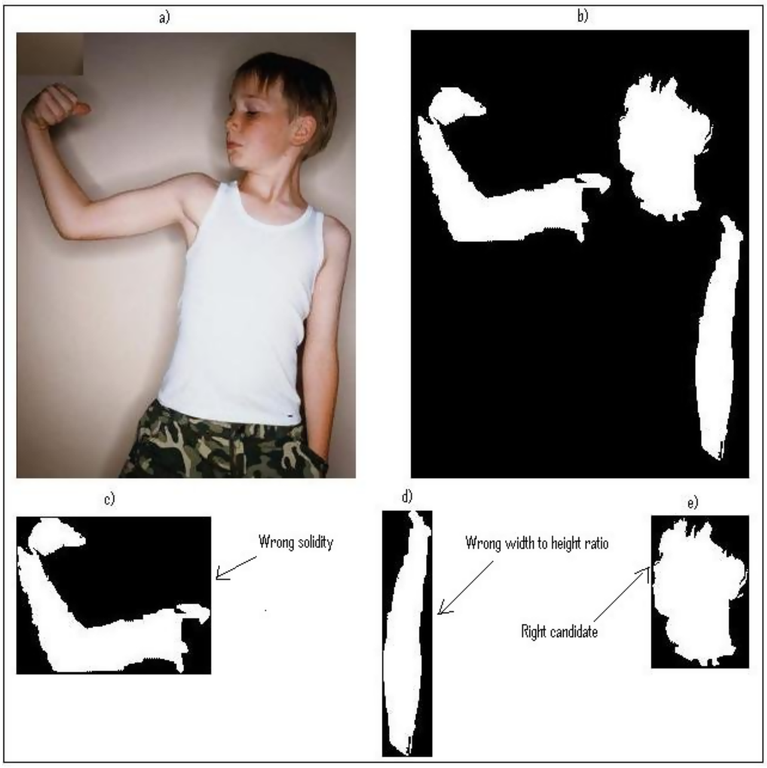


Figure 4: Face candidate identification example

works like a supervised classification algorithm. We create a face template set, containing human faces of various sizes, orientations and poses, and representing both male and female people, of various ages and races. Let the template set be $\{F_1, ..., F_N\}$, with N large enough, where each F_i represents a grayscale image.

Next, an edge detection operation is performed on both the skin image set and the face template set. A Canny filtering technique is used for image edge extraction, because this detector is less likely than the others to be affected by noise [20]. First, it computes the gradient of the image, using the derivative of a Gaussian filter. Then, it finds edges by looking for local maxima of this gradient. The Canny method uses two thresholds, to detect strong and weak edges. It takes into consideration only the truly weak edges, representing those connected to the strong ones. Thus, for each skin image I_i and each face image F_j, a binary image representing its edges is determined. Let us note I_i^e and F_j^e the *edge images* corresponding to images I_i and F_j respectively.

Then, for each candidate (skin image), one computes the $2D$ *cross-correlation coefficients* [21] between its edge image and the edge images of the templates, and the average value of this sequence of coefficients. Let us note $v(I_i)$, this mean value corresponding to image I_i. Each time a correlation operation is performed the edge image of the candidate has to be resized to the size of the template. The best solution to the face detection task is a threshold-based one. The computed two-dimensional mean correlation coefficient corresponding to a facial skin image, must exceed a properly chosen threshold value. The face identification process is expressed mathematically as follows:

$$\forall i \in [1, K], \quad I_i = face \Longleftrightarrow v(I_i) \geq T \tag{6}$$

where T represents the chosen threshold and

$$v(I_i) = \frac{1}{N}\sum_{j=1}^{N}\frac{\Sigma_x\Sigma_y(I_i^e(x,y)-\mu(I_i^e))(F_j^e(x,y)-\mu(F_j^e))}{(\Sigma_x\Sigma_y(I_i^e(x,y)-\mu(I_i^e))^2)(\Sigma_x\Sigma_y(F_j^e(x,y)-\mu(F_j^e))^2)} \tag{7}$$

where $\mu(\ )$ computes the mean of a matrix. The threshold value is determined empirically. From the performed experiments, we have got a satisfactory threshold value, $T = 0.185$. If T is not exceeded for any I_i, then the color image I contains no human faces. We propose a no-threshold automatic face finding approach, too. The threshold can be replaced with a clustering procedure that uses the values $v(I_i)$ computed by (7) as feature vectors. Thus, the value set $\{v(I_1), ..., v(I_k)\}$ is divided into two classes. A region-growing algorithm can be used in this case.

There is also a more simple way to perform the clustering: to sort the set in ascending order, then to find the greatest difference between two successive values. That pair of successive correlation-based values marks the dividing point between the two clusters. Obviously, the cluster containing the high values is the one corresponding to the facial images. This method works satisfactory when the values $v(I_i)$ related to faces are *much* greater than those corresponding to non-facial images. Some face detection errors could be produced. For this reason, the threshold based approach is preferred by us.

After determining those I_i images representing faces, if such images exist, the corresponding facial sub-images of the RGB image I are provided to the output of our face detection system. Let us return to the example described in the first two figures. If the described face finding technique is applied to the skin detection results depicted in Figure 2, we get the results displayed in Figure 5.

In Figure 5 (a) there are displayed the main skin regions of the image, obtained after performing the morphological operations, the hole filling and the small region removing processes on the binary image depicted in Figure 2. One of these skin regions is rejected because of its low solidity, the remaining regions being accepted as face candidates, as one can see in Figure 5 (b).

The template matching process is performed using the template face set represented in Figure 6. One can see the resulted average correlation coefficient values in Figure 5 (c), those greater than 0.185 corresponding to the detected faces, surrounded by black rectangles in that grayscale image. The final face detection result for the RGB image is displayed in Figure 5 (d), the human faces being marked by red bounding boxes.

4 Experiments

We performed a lot of face detection experiments using the described system. Our tests involved tens of RGB images containing human faces and produced satisfactory results. A high face detection rate is obtained.

We created a template face set that contains 25 grayscale images of various scales and imaging conditions, for our experiments. As one can see in the pictures below, these templates represent both male and female faces, and people of various ages and races. These faces are also characterized by various orientations and poses, some of them have structural elements, too. The template set can be extended, by adding new faces, but although a large set could improve the detection results, it also produces a high computation complexity.

As mentioned in the previous section, the threshold-based face finding approach provides much better results than the clustering-based one. Our face detection technique is characterized not only by a high detection rate, that is approximately 90% and indicates a low number of *false negatives* (missed faces), but also by a low number of *false positive* (non-facial image regions declared to be faces).

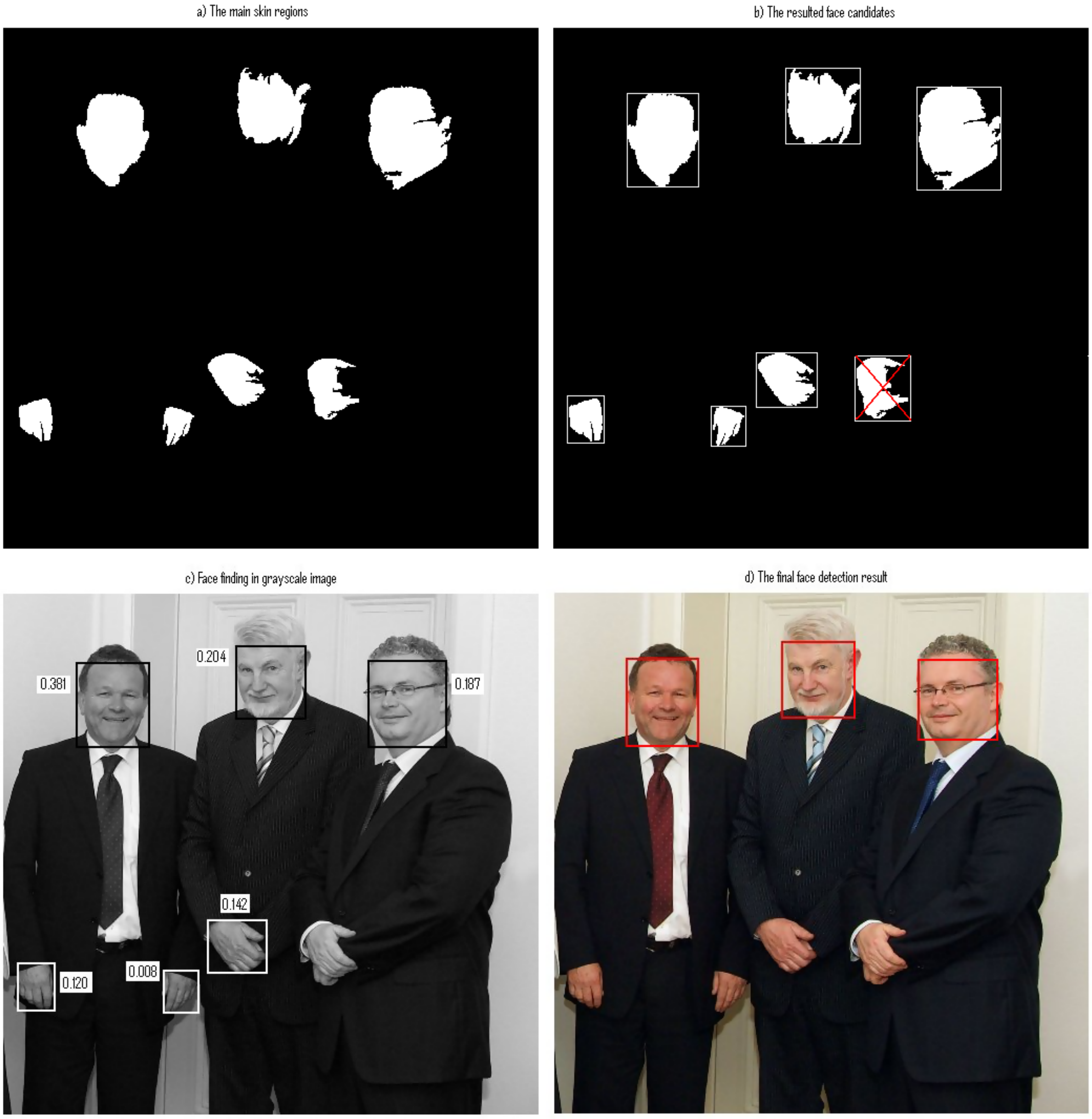

a) The main skin regions
b) The resulted face candidates
c) Face finding in grayscale image
0.381
0.204
0.187
0.142
0.120
0.008
d) The final face detection result

Figure 5: Face detection example

The performances of the proposed face detection system are comparable with those of the face detection approaches mentioned in the introduction. It achieves better detection results for the frontal faces, than for faces characterized by various orientations.

5 Conclusions

A skin segmentation based face detection system for RGB color images has been proposed in this paper. The main contributions of this work are the proposed skin detection approach and face identification technique.

The skin regions resulted from the skin segmentation process are analyzed to determine which of them could represent human faces. A set of face candidate criteria was proposed by us, to reduce the set of face candidates and the computation volume and complexity.

We used a template matching method for face detection and provided a cross-correlation

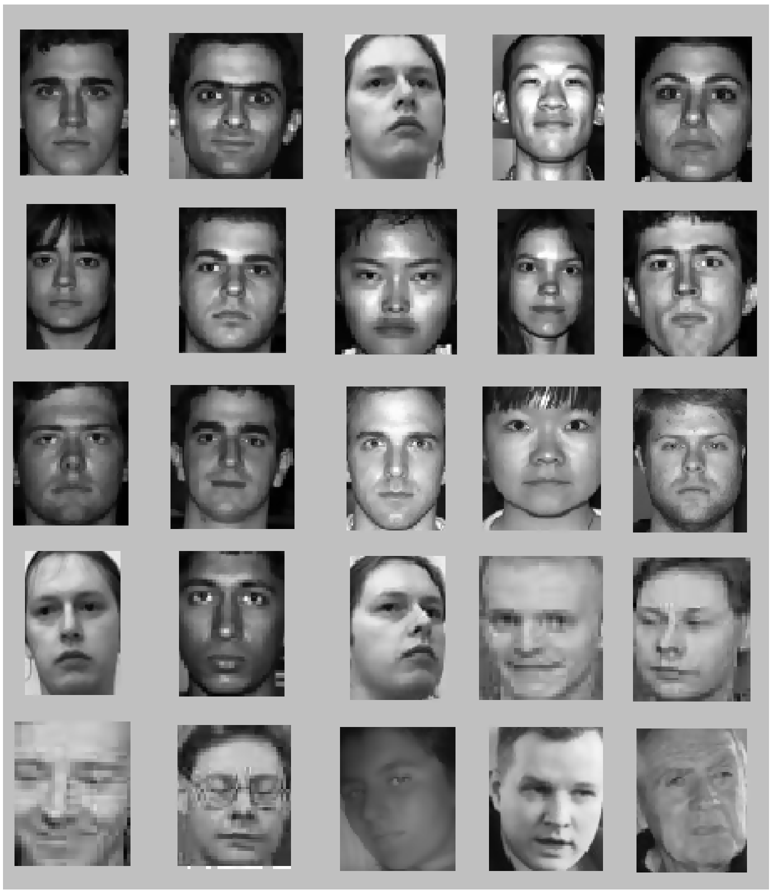


Figure 6: Template face set

based skin region discrimination approach. Unlike other template matching algorithms, our procedure uses the edges of the skin region and not the region itself. We choose to perform an edge detection first, because we consider that face features are contained mainly in the edge image.

The good face detection results obtained in our experiments prove the effectiveness of our technique. Our future work will focus on developing robust face recognition and more complex biometric systems, using the face detector proposed in this paper. We approached the face recognition domain in our previous works [4], and we want to unify the research in these two areas.

Acknowledgment

The research described here has been supported by the grant PNCDI II, IDEI program having the CNCSIS Code 70/2008.

Bibliography

[1] C. Papageorgiou, M. Oren, T. Poggio. A General Framework for Object Detection, *International Conference on Computer Vision*, Bombay, India, pp. 555-562, Jan. 1998.

[2] M.H. Yang, D. Kriegman, N. Ahuja. Detecting Faces in Images: A Survey, *IEEE Transactions on Pattern Analysis and Machine Intelligence* (*PAMI*), Vol. 24, no. 1, pp. 34-58, Jan. 2002.

[3] S. Atsushi, I. Hitoshi, S. Tetsuaki, H. Toshinori. Advances in face detection and recognition technologies, *NEC Journal of Advanced Technology*, Vol. 2, no. 1, pp. 28-34, 2005.

[4] T. Barbu. Eigenimage-based face recognition approach using gradient covariance, *Numerical Functional Analysis and Optimization*, Volume 28, pp. 591 . 601, Issue 5 & 6, May 2007.

[5] G. Yang, T.S. Huang. Human face detection in a complex background. *Pattern Recognition*, Vol. 27, no. 1, pp. 53-63, 1994.

[6] T.K. Leung, M.C. Burl, P. Perona. Finding Faces in Cluttered Scenes Using Random Labeled Graph Matching, *Proceedings of the 5th International Conference on Computer Vision*, pp. 637-644, Cambridge, Mass., June 1995.

[7] K.C. Yow, R. Cipolla. A probabilistic framework for perceptual grouping of features for human face detection, *Second IEEE International Conference on Automatic Face and Gesture Recognition* (FG '96), pp. 16, 1996.

[8] H.A. Rowley, S. Baluja, T. Kanade. Neural Network-Based Face Detection, *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 203-208, 1996.

[9] A.V. Nefian. An embedded HMM-based approach for face detection and recognition, *Proceedings of the Acoustics, Speech, and Signal Processing ďž˝e99 on 1999 IEEE International Conference*, Vol. 6, pp. 3553-3556, 1999.

[10] T.V. Pham, M. Worring, A.W.M. Smeulders. Face Detection by Aggregated Bayesian Network Classifiers, *Machine Learning and Data Mining in Pattern Recognition*, Book Series Lecture Notes in Computer Science, Volume 2123, pp. 249-262, 2001.

[11] E. Osuna, R. Freund, F. Girosi. An improved training algorithm for support vector machines, In *Proceedings of IEEE NNSP'97*, pp. 276-285, Amelia Island, Florida, 1997 (a).

[12] M. Nilsson, J. Nordberg, I. Claesson. Face Detection using Local SMQT Features and Split Up SNoW Classifier, *IEEE International Conference on Acoustics, Speech, and Signal Processing* (*ICASSP*), Vol. 2, pp. 589-592, April 2007.

[13] K. Ichikawa, T. Mita, O. Hori. Component-based robust face detection using AdaBoost and decision tree, *Proc. of the 7th Int. Conference on Automatic Face and Gesture Recognition*, pp. 413-420, 2006.

[14] Z. Jin, Z. Lou, J. Yang, Q. Sun. Face detection using template matching and skin-color information, *Advanced Neurocomputing Theory and Methodology*, Vol. 70, Issues 4-6, pp. 794-800, Jan. 2007.

[15] S. Majed, H. Arof. Pattern correlation approach towards face detection system framework, *Information Technology, 2008. ITSim 2008. International Symposium on*, Vol. 4, pp. 1-5, Aug. 2008.

[16] D. A. Forsyth, M. M. Fleck. Identifying nude pictures, *IEEE Workshop on the Applications of Computer Vision '96*, pp. 103-108, 1996.

[17] V. Vezhnevets, V. Sazonov, A. Andreeva. A Survey on Pixel-Based Skin Color Detection Techniques, In *Proceedings of the GraphiCon 2003*, pp. 85-92, 2003.

[18] L.G. Shapiro, G. C. Stockman. *Computer Vision*, pp. 137- 150, Prentince Hall, 2001.

[19] H.J.A.M. Heijmans. Morphological Image Operators, *Advances in Electronics and Electron Physics*, Boston: Academic Press, 1994.

[20] J. Canny, A Computational Approach To Edge Detection, *IEEE Trans. Pattern Analysis and Machine Intelligence*, Vol. 88, pp. 679-714, 1986.

[21] A.L. Edwards, *An Introduction to Linear Regression and Correlation*, San Francisco, CA: W.H. Freeman, pp. 33-46, 1976.

Int. J. of Computers, Communications & Control, ISSN 1841-9836, E-ISSN 1841-9844
Vol. VI (2011), No. 1 (March), pp. 33-44

The Flag-based Algorithm - A Novel Greedy Method that Optimizes Protein Communities Detection

R. Bocu, S. Tabirca

Razvan Bocu
National University of Ireland, Cork
Department of Computer Science
E-mail: razvan.bocu@cs.ucc.ie

Sabin Tabirca
National University of Ireland, Cork
Department of Computer Science
E-mail: tabirca@cs.ucc.ie

Abstract: Proteins and the networks they determine, called interactome networks, have received attention at an important degree during the last years, because they have been discovered to have an influence on some complex biological phenomena, such as problematic disorders like cancer. This paper presents a contribution that aims to optimize the detection of protein communities through a greedy algorithm that is implemented in the C programming language. The optimization involves a double improvement in relation to protein communities detection, which is accomplished both at the algorithmic and programming level. The resulting implementation's performance was carefully tested on real biological data and the results acknowledge the relevant speedup that the optimization determines. Moreover, the results are in line with the previous findings that our current research produced, as it reveals and confirms the existence of some important properties of those proteins that participate in the carcinogenesis process. Apart from being particularly useful for research purposes, the novel community detection algorithm also dramatically speeds up the proteomic databases analysis process, as compared to some other sequential community detection approaches, and also to the sequential algorithm of Newman and Girvan.

Keywords: Interactome networks, protein-protein interactions, protein communities, cancer, greedy algorithm.

1 Introduction

1.1 Basic Considerations on Protein Networks and Their Importance

Interactome networks, or, more specifically, networks of proteins, determine a fundamental biological theoretical entity. Theoretical and practical endeavours often use interactome networks-related formalisms in order to analyze the protein interactions that determine a biological network, which is essential for the proper organization and function of a biological organism. These networks exhibit a complex structure, which implies that any research activity in the field is handled with inherent theoretical and technical difficulties. Nevertheless, the dynamics and the structure of these biological networks have to be accurately understood, as they play an important role on the function of a biological organism seen as a whole, regardless their degree of structural complexity. As a consequence, it is highly required to design and implement efficient proteomic data analysis techniques that can be integrated in any research framework that study the structure and properties of the interactome networks.

The aim of this paper is to present a novel and faster algorithm that performs the detection of communities in the interactome networks, based on a computationally-effective greedy technique.

The significant influence that proteins exercise on fundamental physiological processes has been demonstrated in a series of recent contributions. In this respect, this paper re-states our research's previous developments, apart from the algorithmic optimization itself, that cancer affects the most important proteins in the interactome network and, as a consequence, the normal function of the organism is greatly disturbed. An accurate understanding of the structure and importance of proteins requires the usage of efficient analysis techniques. In this context, the proper detection of protein communities is of utmost importance, because it is one of the fewer methods that allows an in-depth and informative analysis of protein networks, through the isolation of functionally-related protein communities.

The paper will briefly enumerate the most relevant existing works regarding the community detection. Furthermore, the novel algorithm will be described and analyzed. Also, its practical usability is assessed on real proteomic data.

1.2 Essential Previous Work

The Newman-Girvan algorithm is one of the methods used to detect communities in complex systems. A community is built up by a subset of nodes within which the node-node connections are dense, and the edges to nodes in other communities are less dense. It is important to note that there are a number of alternative algorithmic techniques for the detection of communities in networks. They include hierarchical clustering, partitioning graphs to maximize quality functions such as network modularity, k-clique percolation, and some other interesting algorithmic methods [1,2]. Nevertheless, the Newman and Girvan conceptual system is often chosen as a pretext for scientific contributions due to its structural articulation and its ability to be used in a wide range of practical situations [3]. The Newman-Girvan algorithm is particularly used to compute betweenness for edges (biological links) that connect the nodes (proteins) in a network.

The algorithm of Blondel et al. [26] is able to find high modularity partitions of large networks in a timely manner and, thus, to unfold a complete hierarchical community structure for the analyzed network. Contrary to some other community detection algorithms, the network size limits that this algorithm faces are mainly generated by the limited storage capacity on the processing machines, rather than by the computational complexity. The algorithm of Blondel et al. represents a different greedy approach to community detection that can be successfully applied to protein networks. The algorithm assigns proteins to their respective communities following a particular-to-general approach. Thus, proteins are progressively added to their correct clusters considering only the existence of inter-protein links. This way, at some point the modularity cannot increase any more and thus, the algorithm stops. This approach has the significant advantage of avoiding the usage of computationally expensive data structures, as it is the case with the algorithm of Clauset et al. Therefore, it is rather limited by storage (memory) requirements than by computational time.

The algorithm of Clauset et al. [25] has the merit to point out that the updates that are performed by the algorithm of Newman and Girvan involve a significant number of pointless operations, as a consequence of the sparse nature of the adjacency matrix. As a consequence, the algorithm is optimized in order to properly handle the sparsity of the input data sets. Thus, the three data structures that the algorithm uses contribute to optimizing the analysis process of protein networks. In this respect, it can be stated that the modularity can be updated in a sensibly quicker way using these three data structures, as compared to the algorithm of Newman and Girvan. Consequently, it can be inferred it is one of the few algorithms that can be used to determine community structure in the case of protein networks.

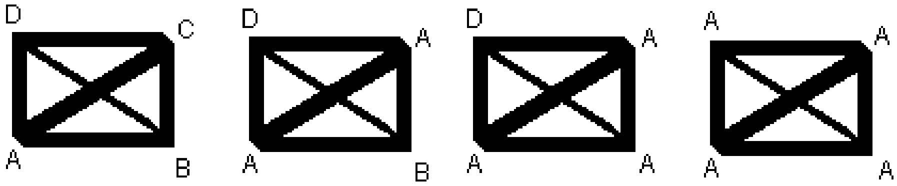


Figure 1: The flag values are updated one by one, starting with the leftmost representation. As a consequence of the complete nature of this network, all the nodes are flagged identically, thus constituting the only and network-wide community.

The following sections will describe the improved algorithm in more detail, along with some theoretical considerations that are mandatory for a good understanding of the new approach's structure and strengths.

2 General Presentation of the New Approach

2.1 Description of the Algorithm

The algorithm is designed according to a local-to-general approach. The iterative process starts with all the nodes considered as members of their own one-node community. Then, each node assigns itself to one of the neighbouring communities considering local information about its proximity. In order to accomplish this, we attached to each node in the network an additional data structure called *flag*, which holds information about the community to which a node currently belongs.

The mechanism that coordinates the flag-based community algorithm is the following. Let us suppose that a certain node *node* has the set of neighbours $\{node_1, node_2, node_3, ..., node_k\}$, with $k \leq n$, where n represents the number of nodes in the network. Each of these neighbours has a flag attached to it, which offers information about the community to which it belongs at a given time during the iterative process. In these conditions, *node* assigns itself to the optimal community, considering the community information offered by the flags of its neighbours. The algorithm is designed in such a way that it chooses the neighbouring community to which most of *node's* neighbours belong. At the beginning of the first iteration, each node is initialized with unique community information contained in its own flag. During the course of the algorithm's iterations, accurate information describing the community structure spreads network-wide. Thus, functionally or otherwise related nodes feature, in their flag field, similar community information, after a fairly reduced number of iterations. As many such consensus-based groups of nodes build, they continue to attract nodes from smaller neighbouring communities, while it is possible to do so. When the algorithm completes its last iteration, nodes that are characterized by the same flag information are grouped in the same community. The flag information updating process is illustrated in *Figure 1* on a 4-node *complete* network configuration. We say that a network configuration is complete if, for any pair of vertices (u, v), there is an edge that connects them.

The algorithm conducts the flags updating process in an iterative manner. At every iteration, each node updates its flag considering the community information that is stored in the flags of its neighbours. Formally, this idea can be expressed by the following function: $Flag_{current\_node}(t) = update\_flag(Flag_{node_1}(t-1), Flag_{node_2}(t-1), Flag_{node_3}(t-1), ..., Flag_{node_k}(t-1))$. Here, $Flag_{current\_node}(t)$ represents the flag of the currently analyzed node at time t, which is computed through the function $update\_flag$, which takes as arguments the flag values of its neighbours, $(node_1, node_2, node_3, ..., node_k)$. Following the node processing suggestion that is introduced by the algorithm of Blondel et al., which is presented in a previous section, the order according to which nodes are selected to have their flag updated, is established in a random manner. It is immediate to note that, while there are n flags at the beginning of the first iteration, the number reduces over the course of the algorithm, resulting in the end in as many unique flag values as there are communities.

The algorithm should continue to run until no node changes the information in its flag anymore. However, there could be nodes in the network that are featured by an equal maximum number of neighbors in two or even more neighbouring communities. We addressed this potential shortcoming by instructing the algorithm to stop when each node in the network is described by a flag value that the maximum of its neighbours have. Let us consider $\{Flag_1, Flag_2, ..., Flag_s\}$ to be the set of the flags that exist in the network at a given moment. Moreover, let us note by $number\_neighbours(current\_node, Flag_{certain\_flag})$, the number of vicinities $current\_node$ has with nodes described by a $certain\_flag$. In these conditions, it can be stated that the algorithm is stopped when, considering each $current\_node$ in the network, the following condition is true:

$number\_neighbours(current\_node, Flag_{certain\_flag})$
is less or equal than
$number\_neighbours(current\_node, Flag_{some\_flag})$,

for all possible selections of $certain\_flag$, and considering that $current\_node$ is characterized by the information stored in $Flag_{some\_flag}$.

The algorithm configures the community structure in the network by grouping together nodes with similar flags. Let us note that, obeying this stop criterion, the algorithm establishes a community structure that ensures each node is connected to, at least, as many other nodes from its own community than it is with each other extra-community node.

We are able now to state the algorithm's execution pertains to the following general steps:

1. The flag that is attached to each node in the network is initialized with the appropriate value, which respects the following rule: $Flag_{current\_node}(0) \leftarrow current\_node\_identifier$. In other words, each node's flag initially stores the identifier of the vertex, for example the number that denotes the node's rank.

2. Following, the first effective processing iteration of the algorithm is initiated. It implies the arrangement of the nodes in a random manner, let us note the resulting vertex set with $\mathcal{R}$.

3. Then, for each $current\_node \in \mathcal{R}$, and respecting the order of the elements, we have that $Flag_{current\_node}(t) = update\_flag(Flag_{node_1}(t-1), Flag_{node_2}(t-1), Flag_{node_3}(t-1), ..., Flag_{node_k}(t-1))$. Let us recall that the function $update\_flag$ updates the information stored by the flag of $current\_node$, considering the flag that appears with the highest frequency among its neighbours.

4. The algorithm continues to iterate until each node stores in its flag the same information as the majority of its neighbours do. Otherwise, $t \leftarrow t+1$, and the next iteration is initiated from step 2.

The main steps of the algorithm are presented, in a very brief manner, in *Algorithm 1*.

The actual C implementation of the algorithm is sensibly more complex than it apparently seems to be, considering the pseudo code in *Algorithm 1*, as it includes some auxiliary decision

Algorithm 1 The flag based community detection algorithm - general structure

```
//Perform some initial checks
if $checkDataSet() = FALSE$ then
  DisplayErrorMessage("It is a protein network, it should be un-weighted!")
  return errorCode
end if
DO Create adjacency list and store edges
DO Create storage space to store and count flags
DO Create the node ordering vector nodeOrder[]
DO Arrange nodes in random order relative to nodeOrder[]
DO Considering the random order, process all nodes and assign flags accordingly
for $i = 0; i < numberOfNodes; i++$ do
  $unprocessed[i] \leftarrow TRUE$
  $noOfFlags[i] \leftarrow 0$
end for
for $i = 0; i < numberOfNodes; i++$ do
  $currentFlag \leftarrow getFlagOfNode(i)$
  $numberNeighboursCurrentNode \leftarrow getNumberNeighboursOfNode(current\_flag)$
  $unprocessed[i] \leftarrow FALSE$
  for $j = 0; j < numberNeighboursCurrentNode; j++$ do
    $currentFlagNeighbour \leftarrow getFlagOfNode(j)$
    if $unprocessed[j] = TRUE$ then
      DO Continue
    else
      $noOfFlags[currentFlagNeighbour] \leftarrow noOfFlags[currentFlagNeighbour] + 1$
      if $numberNodesWithFlag(currentFlag) <$
      $numberNodesWithFlag(currentFlagNeighbour)$ then
        DO $setFlagOfNode(i, getFlagOfNode(j))$
      end if
    end if
  end for
end for
//At this point, nodes are properly flagged
DO Revert nodes ranks in nodeOrder[] back to their un-randomized state, preserving flag
information
```

statements and processing blocks of instructions. Nevertheless, the pseudo code summarizes the core of the process that performs the proper assignment of values to each node's flag and, as a consequence, the detection of the community structure.

2.2 Remarks Regarding the Complexity and Correctness

In spite of the *double for loop* structure that is also displayed in *Algorithm 1*, in practice, the flag based algorithm exhibits a near linear time complexity. There are several reasons that explain this efficient behaviour.

The initial allocation of flag values to each node, which is performed in the first part of the algorithm's execution, takes $O(n)$ time to complete. Considering each *current_node*, its neighbours are initially grouped according to the information stored in their flags, which generates a worst-case time complexity of $O(number\_neighbours(current\_node))$. Then, in order to conduct all subsequent iterations, the algorithm selects the flag-related group that has the maximum size, and consequently assigns the relevant flag to *current_node*. This selective flag filtering process is repeated for all the nodes in the network. Therefore, an overall run time of $O(m)$ is required to complete the main processing loop. The global time complexity results following a simple join operation, which takes into account the complexity required to perform the initial initializations, together with the time complexity that is generated by the main processing loop. As a consequence, it can be stated that the global time complexity of the algorithm is $O(n + m)$, where n denotes the number of nodes (proteins) in the processed network, while m represents the number of edges (biological links).

Another factor that explains the efficient behaviour of the algorithm resides in the fact the algorithm is able to correctly fill in the flags, and therefore assigns the nodes to the appropriate community, after only a few iterations. According to the experimental results of the tests we performed, the algorithm manages to fill approximately 80% of the flags with appropriate data after only three or four iterations. Furthermore, it can be stated that the community structure is fully determined after only six or seven iterations, in most cases. The next section, which offers a detailed account on all the algorithms' performance, will fully present the performance assessment process.

We have performed an assessment of the algorithm's correctness following the same empirical method with three stages, which has also been used to prove the correctness of the algorithm designed by Clauset et al. First, we determined the community structure of a 9000-node subset that was extracted from the aggregated protein data set. The output that the flag-based algorithm produced was compared to that generated by the parallel version of the Newman-Girvan algorithm. We found the two community partitions to be similar. Furthermore, we performed a comparative assessment considering the flag-based algorithm, together with Blondel's algorithm. The analysis took into account the *Amazon.com purchasing network* [22], and the aggregated protein data set. Let us recall that the Blondel's algorithm correctness has also been evaluated against the parallel version of the Newman-Girvan algorithm. We found that the flag-based algorithm produces a community structure that is identical to that generated by both reference algorithms, the Newman-Girvan parallel version and the approach of Blondel et al. Therefore, the algorithm's correctness is sufficiently demonstrated through this empirical method. Nevertheless, it is worth noting that we have also conducted a more formal verification regarding the algorithm's correctness, by going through the main steps suggested in *Algorithm 1*, and considering five randomized 20-node sample networks. The random networks were generated with Network Workbench Tool [23], and following the suggestions contained in [24]. The step by step execution of the algorithm on these networks revealed that it is able to correctly assign the flag values to the networks' nodes in all situations.

Table 1: Execution times - comparative analysis

Test number	Algorithm	Execution time
1	Flag-based	502
2	Blondel et al.	749

3 Comparative Analysis Regarding the Algorithm's Performance on Interactome Networks

The algorithm was implemented in C and run, as a sequential process, on a Beowulf class cluster of computers. Each node in the cluster provides 12 MB of cache memory for the sequential or parallel processes that run on it. Additionally, each node in the parallel world is powered up by an Intel Xeon 5420 processor, clocked at 2.5 GHz. The C language was selected as the backbone of all our implementations, because it is the closest-to-machine medium level programming language and, as a consequence, the gain in performance is noticeable.

The comparative performance testing procedure made use of an aggregated biological data set that features $22,573$ proteins and $1,886,753$ biological links. The size of this input data set determines a problem space that can be hardly handled properly in terms of execution times through a sub-optimal sequential approach.

The testing procedure conducted a comparative performance assessment that made use of the biological data provided by the aggregated protein data set. The flag-based community detection algorithm is assessed against the optimal community detection algorithm that has already been introduced, the construct of Blondel et al. Let us recall that the algorithms were implemented in C and run on the same Beowulf class cluster of computers, which has been standard during the course of our research, as sequential processes.

In *Table 1*, durations are expressed in seconds. Additionally, the speedup generated by the flag-based algorithm is suggestively displayed in *Figure 2.* It can be noticed that, compared to the fastest community algorithm, which has been introduced in the previous chapter, the flag-based algorithm performs noticeably faster. Furthermore, seven smaller protein datasets (1000, 2000, 2500, 3000, 7000, 8000, 9000 proteins) have been considered. In this respect, *Figure 3* contains a visualization of a comparative analysis concerning the algorithms' performance.

4 Remarks Concerning the Accuracy of the Algorithms' Output

Community detection algorithms determine the community structure with various degrees of accuracy. In this respect, the main problem that may impede the output of such an algorithm is represented by the possibly inaccurate allocation of nodes (proteins) to their respective communities. We have consistently faced this problem during the current phase of our research. When speaking about protein data sets that require a proper community structure detection, the accuracy of the algorithm's output is mandatory, as even the slightest community structure misconfiguration may lead to incorrect deductions and conclusions. We made use of a measure called *modularity*, which assesses the quality of the community structure determined by the algorithm.

Suppose there are k clusters in the current iteration of the algorithm. A symmetric matrix E of size $k \times k$ is constructed according to the following procedure. An element e_{ij} in E represents the fraction of all edges that link the vertices in cluster i to the vertices in cluster j and e_{ii} represents the fraction of edges that connect vertices within cluster i. Thus, summation of row (or column) elements $c_i = \sum_{i=1}^{k} e_{ij}$ represents the fraction of all edges that connect vertices to and

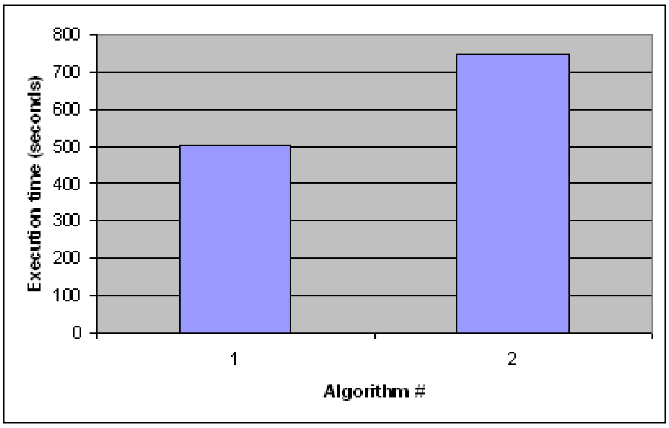


Figure 2: Community detection optimization in protein networks - the descending trend of the execution time induced by the flag-based community detection algorithm, as compared to the algorithm of Blondel et al. The algorithm numbers denote the following: 1 - Flag-based algorithm, 2 - Blondel et al.

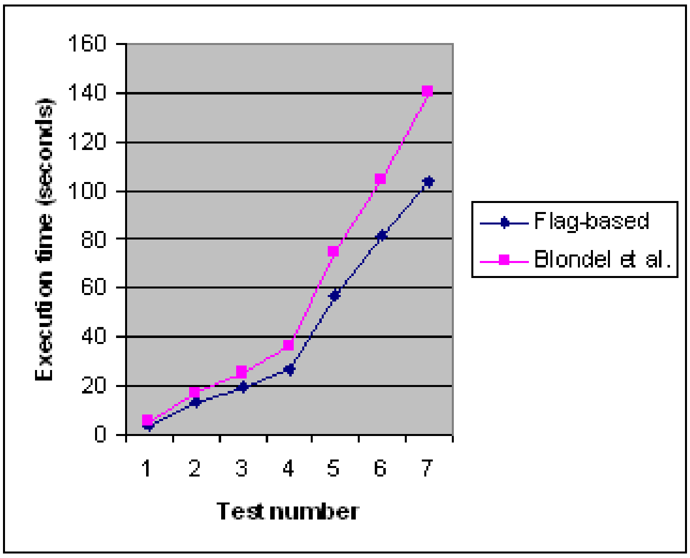


Figure 3: The flag-based community detection algorithm - comparative performance analysis on seven smaller protein datasets.

within cluster i. In these conditions, modularity is defined as $Q = \sigma_{i=1}^{k}(e_{ii} - c_i^2)$, which measures the fraction of the edges that connect vertices within the same cluster minus the expected value of the same quantity in the network [25]. For a random network with random decomposition, Q approaches 0. Values approaching $Q = 1$, which is the maximum, indicate strong clustering structure. The higher is the value, the stronger is the clustering structure in the network. The inclusion of modularity as a community assessment measure significantly improved the accuracy of the community detection process output.

In this context, it can be stated that, taking into consideration some further comparative tests we conducted, the accuracy of the community structure generated by the algorithm is comparable to that produced by the algorithm of Newman and Girvan, both in its sequential and parallel version. Therefore, the tradeoff created by the utilization of this greedy-based optimization strategy is, in practice, nonexistent. Thus, while the sequential algorithm of Newman and Girvan generates, considering the aggregated protein data set, a community structure that is characterized by a modularity of 0.837, the flag-based algorithm builds a community structure whose modularity is 0.835. Additionally, the algorithms of Clauset et al. and Blondel et al. determine, each of them, a protein community structure whose accuracy is characterized by a modularity of 0.821 and 0.811, respectively. It is immediate to note that the flag-based algorithm is not only the fastest community detection algorithm that has been used, but also the most accurate greedy algorithm, according to the modularity of the community structure that has been determined.

5 The Algorithm and Its Suitability for Protein Networks Analysis

The flag-based community detection algorithm features a simple and intuitive structure that works around all the overheads that are inherent, in connection with sub-optimal algorithms, when a complex operation like community detection is performed. The degree of complexity and the associated overhead dramatically increase when a large networked structure, like the interactome, is processed. Therefore, suitable algorithms have to avoid the usage of less efficient data structures. Additionally, a proper balance between output's accuracy and computational time has to be sought and implemented accordingly.

The flag-based community detection algorithm has been designed with all the above principles in mind. Thus, the algorithm uses only simple array structures. Moreover, the community discovery method proposed by this algorithm determines a straightforward detection of protein communities. Thus, more than 50% of all the proteins in the aggregated data set are assigned to their respective communities after only a few iterations. This behaviour suggests that the algorithm is able to process any existing biological data set, regardless the degree of enrichment with new data in the foreseeable future.

As a final remark in this section, it is important to note that this gain in performance is not accomplished in the detriment of the output's accuracy. In this respect, it can be stated that the value of modularity demonstrates that the algorithm assigns proteins to correct communities with almost the same accuracy as the algorithm of Newman and Girvan, which is an exact construct that needs significantly more computational time. Considering all the greedy community detection algorithms that have been analyzed, the flag-based algorithm generates the most accurate proteomic community structure.

6 Analysis of Cancer-related Protein Communities

In order to extract the data that is relevant to cancer, we used the valuable data on protein families that is made available in the Pfam database [16].

Let us recall that the testing procedure made use of a compiled biological data set that features $22,573$ proteins and $1,886,753$ biological links. This size of the input data set determines a problem space that can be hardly handled properly in terms of execution times through a sequential approach.

We examined the protein communities our method determined and some interesting differences in the community sizes were noticed. Cancer proteins belong to more highly populated communities compared to non-cancer proteins. The explanation may reside in the fact that cancer proteins take part in more complex cellular (carcinogenic) processes than those proteins that are of lower importance in the interactome network and, consequently, have less influence on the carcinogenesis. It can also be asserted that larger protein communities feature a larger or more complicated cellular mechanism, in which cancer proteins play an important role.

Proteins identified as members of more than one protein community are of particular interest. In general, each protein community represents and determines a distinct cellular process. Therefore, proteins that are part of multiple communities may generate multiple cellular processes, and can be considered to be at the intersection of distinct but adjacent cellular processes that are determined by particular protein communities, which are isolated by our community detection technique. The comparison between the cancer proteins population and the non-cancer proteins population reveals that cancer proteins reside at community junctions at a sensibly greater extent than their non-carcinogenic siblings. This particular feature of cancer proteins enforces their special importance in the interactome network seen as a whole and, as a consequence, their influence on all the physiological processes and related disorders.

Existing contributions distinguish between highly connected domains in peripheral cores (locally central) and highly connected domains in central cores (globally central). We noticed that globally central proteins represent an essential backbone of the proteome, exhibit at a high degree evolutionary conservation, and are essential for the organism. It is important to note that cancerous disease provokes mutations exactly to these globally central proteins. This observation supports and extends the findings of Wachi et al. (2005), who showed that differentially expressed proteins in squamous cell carcinoma of the lung tend to be global hubs [18]. Moreover, the findings reported in this paper support and extend the results generated by our research's previous stage. Practically, the above findings reveal the topological features of cancer proteins that are primarily displayed for cancer mutated proteins in exhibiting the highest betweenness centrality compared to the proteins that didn't loose their normal function. In other words, the carcinogenic process is generated by clusters of proteins that feature a central position in the protein network. As a consequence, the high adverse impact of any cancer form is, in our opinion, determined by the way the disease affects the fundamental proteins that coordinate the most essential processes in the metabolic and physiological chains.

The already gathered experimental information can be summed up into the following conclusions:

- The novel protein communities detection algorithm was designed and implemented and was found to accurately determine the functionally-related communities of proteins.
- We practically assessed the suitability and performance of the new approach on real proteomic data related to cancer and the interesting properties of the determined protein communities allowed us to infer an explanation regarding cancer evolution.
- The algorithm performs faster than the algorithms of Blondel et al. and Clauset et al., which represent two of the most efficient community detection solutions that have been designed by now.

6.1 Conclusions and Future Developments

The most important property of cancer proteins is their importance at the scale of the whole interactome. The flag-based algorithm was used to show that the globally central proteins are the ones that are the most affected in a carcinogenic process and are also located at the junction of the most important protein communities.

The resulting clustering algorithm allows us to explore protein-protein connectivity in a more informative way than is possible by just counting the interaction partners for each protein. It allows us to distinguish between central and peripheral hubs of highly connecting proteins, revealing proteins that form the backbone of the proteome. The fact that we observe an enrichment of cancer proteins in this group and also their highest betweenness centrality values indicate the central role of these proteins. The domain composition of cancer proteins indicates the explanation for this topological feature: we have shown, based on our experiments' results, that cancer proteins contain a high ratio of highly malign domains. Therefore, all cancer drugs should be designed in such a way to prevent possible mutations to these highly-important proteins or, if the disease is already on the way, to contribute to reverting back to the original proteomic structure.

Moreover, it is important to note that the scientific presentation in this paper demonstrates that cautiously-designed greedy algorithms can produce an output whose accuracy is on par with that of similar conventional (exact) approaches.

The next stages of our research will involve further optimizations of the algorithms that are used for an efficient community structure detection in protein networks. Also, we intend to analyze even more biological data sets related to cancer and, possibly, other high-impact contemporary diseases.

Acknowledgment

This work is supported by the Irish Research Council for Science, Engineering and Technology, under the Embark Initiative program.

Bibliography

[1] J. Yoon, A. Blumer and K. Lee, *An algorithm for modularity analysis of directed and weighted biological networks based on edge-betweenness centrality*: Bioinformatics, 2006.

[2] M. Girvan and M.E.J. Newman, *Community structure in social and biological networks*: State University of New Jersey, 2002.

[3] D. Ucar et al., *Improving functional modularity in protein-protein interactions graphs using hub-induced subgraphs*: Ohio State University, 2007.

[4] K. Lehmann and M. Kaufmann, *Decentralized algorithms for evaluating centrality in complex networks*: IEEE, 2002.

[5] J. Griebsch et al., *A fast algorithm for the iterative calculation of betweenness centrality*: Technical University of Munchen, 2004.

[6] G.H. Traver et al., *How complete are current yeast and human protein-interaction networks?*: Genome biology, 2006.

[7] R. Bunescu et al., *Consolidating the set of known human protein-protein interactions in preparation for large-scale mapping of the human interactome*: Genome biology, 2005.

[8] U. Brandes, *A faster algorithm for betweenness centrality*: University of Konstanz, 2001.

[9] B. Preiss, *Data structures and algorithms with object-oriented design patterns in C++*: John Wiley and sons, 1998.

[10] EMBL-EBI, *The IntAct protein interactions database.* URL: http://www.ebi.ac.uk/intact/site/index.jsf, 2009.

[11] A. Grama et al., *Introduction to parallel computing, second edition*: Addison-Wesley, 2003.

[12] University of California, *The DIP protein interactions database.* URL: http://dip.doe-mbi.ucla.edu/, 2009.

[13] R. Bocu and S. Tabirca, *Betweenness Centrality Computation - A New Way for Analyzing the Biological Systems*: Proceedings of the BSB 2009 conference, Leipzig, Germany, 2009.

[14] L.C. Freeman, *A set of measures of centrality based on betweenness*: Sociometry, Vol. 40, 35-41, 1977.

[15] P.F. Jonsson and P.A. Bates, *Global topological features of cancer proteins in the human interactome*: Bioinformatics Advance Access, 2006.

[16] Wellcome Trust Sanger Institute, *The Pfam protein families database.* URL: http://pfam.sanger.ac.uk/, 2009.

[17] R. Bocu and S. Tabirca, *Sparse networks-based speedup technique for proteins betweenness centrality computation*: International Journal of Biological and Life Sciences, 2009.

[18] S. Wachi et al., *Interactome-transcriptome analysis reveals the high centrality of genes differentially expressed in lung cancer tissues*: Bioinformatics, 21, 4205-4208, 2005.

[19] G. Palla et al., *Uncovering the overlapping community structure of complex networks in nature and society*: Nature, 435, 814-818, 2005.

[20] P.F. Jonsson et al., *Cluster analysis of networks generated through homology: automatic identification of important protein communities involved in cancer metastasis*: BMC Bioinformatics, 7, 2, 2006.

[21] R. Bocu and S. Tabirca, *Proteomic Data Analysis Optimization Using a Parallel MPI C Approach*: IEEE Computer Society, The First International Conference on Advances in Bioinformatics and Applications, 2010.

[22] A. Clauset, M.E.J. Newman and Ch. Moore, *Finding community structure in very large networks*: Phys. Rev. E 70, 066111, 2004.

[23] S. Schnell, S. Fortunato and R. Sourav, *Is the intrinsic disorder of proteins the cause of the scale-free architecture of protein-protein interaction networks?*: Proteomics 7 no. 6, 961-964, 2007.

[24] V. Batagelj and U. Brandes, *Efficient generation of large random networks*: Physical Review E. 71:036113-036118, 2005.

[25] R. Bocu, *Detecting community structure in networks*: Eur. Phys. J. B 38, 321-330, 2004.

[26] V.D. Blondel, J.-L. Guillaume, R. Lambiotte and E. Lefebvre, *Fast unfolding of communities in large networks*: Journal of Statistical Mechanics, arXiv:0803.0476v2 [physics.soc-ph], 2008.

Int. J. of Computers, Communications & Control, ISSN 1841-9836, E-ISSN 1841-9844
Vol. VI (2011), No. 1 (March), pp. 45-52

Domain Ontology of the VirDenT System

C.M. Bogdan

Crenguţa Mădălina Bogdan
Ovidius University of Constanta, Romania
Numerical Methods and Computer Science Department
124 Mamaia Blvd., 900527
E-mail: cbogdan@univ-ovidius.ro

Abstract: The goal of using virtual and augmented reality technologies in therapeutic interventions simulation, in the fixed prosthodontics (VirDenT) project, is to increase the quality of the educational process in dental faculties, by assisting students in learning how to prepare teeth for all-ceramic restorations. Its main component is an e-learning virtual reality-based software system that will be used for the developing skills in grinding teeth, needed in all-ceramic restorations. This paper presents a domain ontology that formally describes the knowledge of the domain problem that the VirDenT e-learning system dealt with. The ontology was developed based on the UML models of the VirDenT information system, making sure in this way the ontology captures knowledge identified and described in the analysis of the information system. At first, we constructed the taxonomy of these concepts, using the DOLCE ontology and its modules. Then, we defined the conceptual relations between the concepts. We also added SWRL rules that formally describe the business rules and knowledge previously identified. Finally, with the assistance of the Pellet reasoner system, we checked the ontology consistency.

Keywords: knowledge, ontology, taxonomy, information system, UML diagram.

1 Introduction

The goal of using virtual and augmented reality technologies in therapeutic interventions simulation in the fixed prosthodontics (VirDenT) project is to increase the quality of the educational process in the dental faculties, by helping students to learn how to prepare teeth for ceramic crowns.

The VirDenT system is an e-learning system that will be used as a software tool by the students of the dental medicine faculty, to developing their skills in grinding teeth during the dental laboratories of fixed prosthetic department.

These skills are required in making one of the crown types: all-ceramic. All-ceramic restorations are one of the most successfully restorations available from the point of view of aesthetics and biocompatibility. The first criterion is fulfilled by the crown translucent close to the dental hard tissues allowing light to pass through, at gingival level. Biocompatibility refers to the adhesion degree of the ceramic with the surrounding tissue of the teeth. It also refers to how much gingivae tolerate the crown. At present, ceramic appears to be more biocompatible than other dental material [15].

However, this kind of restorations is not widely used because ceramic crowns preparation requires strict and delicate precision. In addition, the bonding techniques employed are somewhat complex and adverse effects could appear, such as pulp inflammation. That is why, the VirDenT system will be a useful tool to the students of dental medicine faculties.

The first phase of the developing process of the VirDenT software system consisted of the analysis of the information system supported by VirDenT. During the analysis a series of models have been created such context diagram, domain model, business process model, and so on [2]. All these models have been created using the Unified Modelling Language (UML) [9] which is a standard language and allows creating graphic models of the information or software system. The ontology construction method is based on these models and therefore we give further two of the models: domain model and business process model.

Domain model describes structurally the VirDenT information system, i.e. considers only persistent information used by the system. Information is classified in classes together with their relationships which form the UML class diagram of the model domain. The classes emerged from the concepts of the problem domain such as Tooth, Root, Gingiva, Enamel, DiamondTool, and so on (Figure 1). The classes are linked together by generalization/specialization relations or different kinds of associations such as aggregations or compositions.

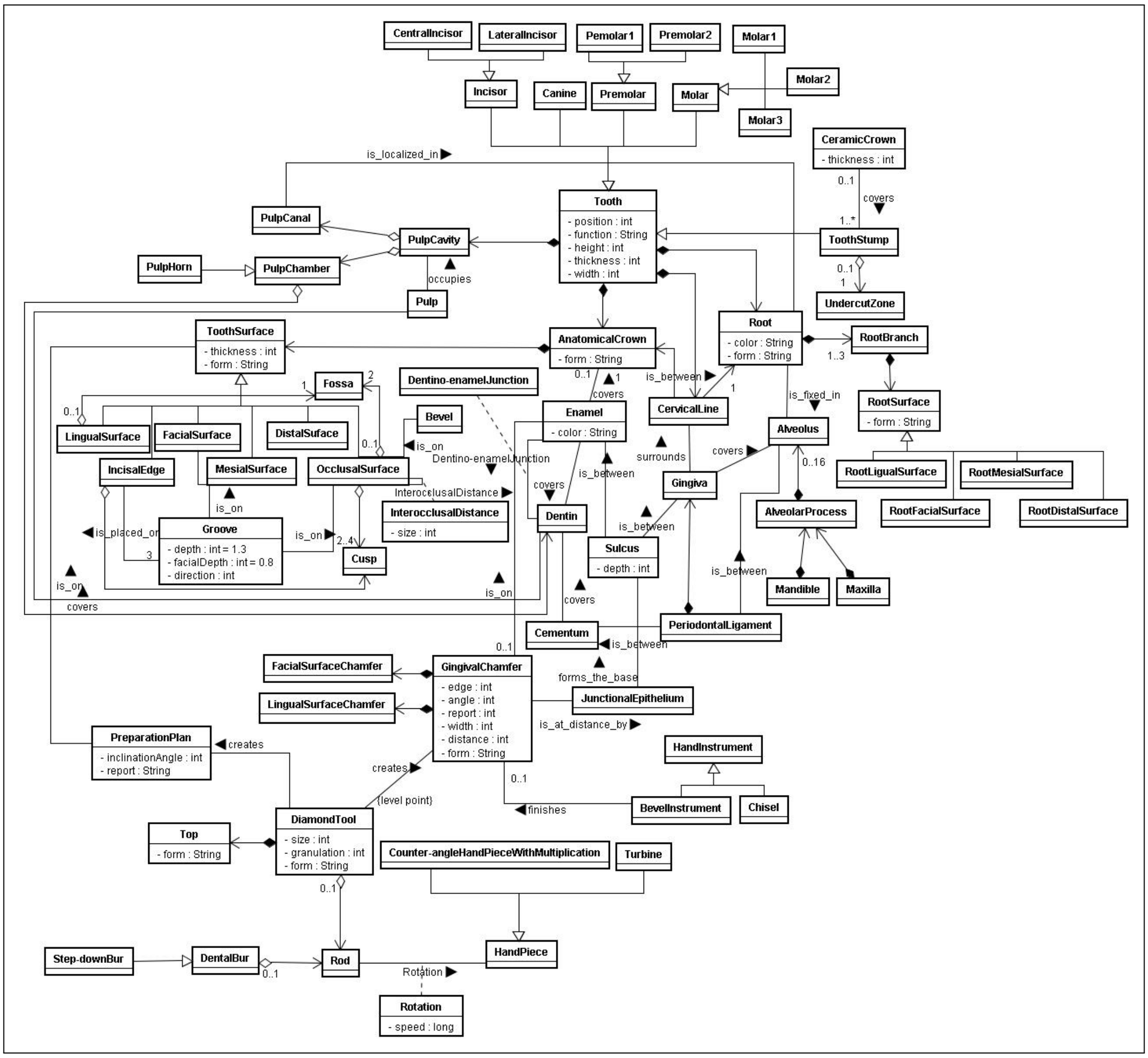


Figure 1: VirDenT domain model

To describe the system from the behavioral point of view we created the business use case diagram which form the business model business processes. The diagram contains eight business use cases that describe the preparation of teeth for all-ceramic crowns. One of the business use cases is "Central incisor preparation" whose UML business activity diagram is presented in Figure 2. The diagram contains activities that are part of the preparation techniques of the central incisors, their execution order and the business objects used as inputs or created or modified as outputs by the activities. Business objects are central incisors, grooves, incisal edges, and so on, and the tools used during execution of the activities, such as diamond tools and step-down burs.

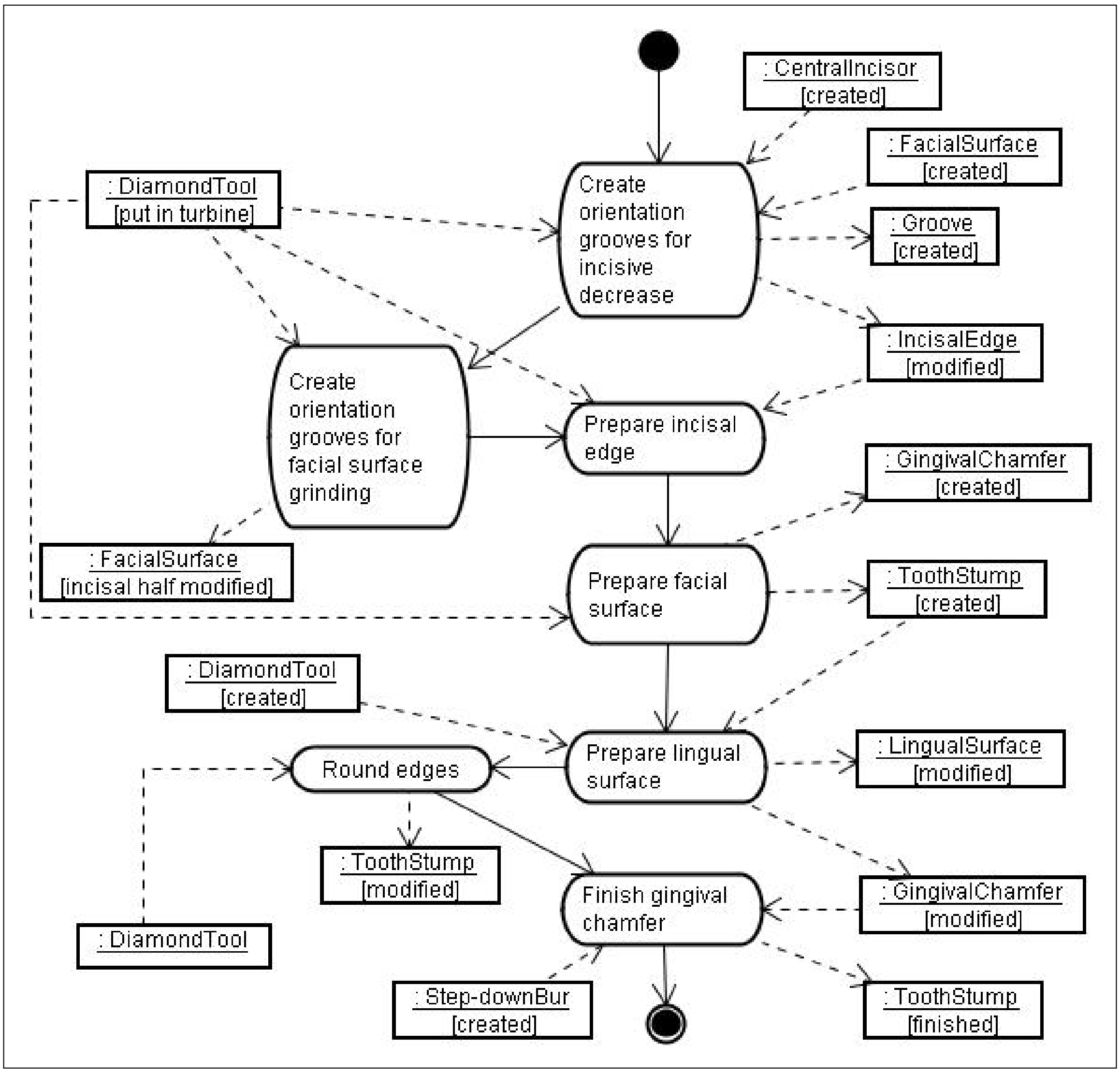

: CentralIncisor [created]
: DiamondTool [put in turbine]
Create orientation grooves for incisive decrease
: FacialSurface [created]
: Groove [created]
: IncisalEdge [modified]
Create orientation grooves for facial surface grinding
Prepare incisal edge
: GingivalChamfer [created]
: FacialSurface [incisal half modified]
Prepare facial surface
: ToothStump [created]
: DiamondTool [created]
Round edges
Prepare lingual surface
: LingualSurface [modified]
: ToothStump [modified]
Finish gingival chamfer
: GingivalChamfer [modified]
: DiamondTool
: Step-downBur [created]
: ToothStump [finished]

Figure 2: Business activity diagram of the VirDenT system

An ontology is a formal specification of the concepts intension and the intensional relationships that can exist between concepts. According to Guarino's definition, *"an ontology is a logical theory accounting for the intended meaning of a formal vocabulary, i.e. its ontological commitment to a particular conceptualization of the world*" [6]. In this paper, we present a domain ontology of the VirDenT system.

2 Method of construction of ontology

The methodology of ontology construction is based on the few existent methodologies, like ontology development 101 [8], concern and business rule-oriented [1] and other ones. Our method performs a translation of the knowledge semantics described semi-formally in the UML models of the VirDenT information system, making sure in this way the ontology captures knowledge identified and described in the analysis of the information system. The method consists of the following steps: a) define the classes and the class hierarchy; b) define the relations of classes; c) ontologically describe the business rules; d) write additional constraints; e) verify the ontology. In addition, we mention that we used to build ontology language OWL DL (Web Ontology Language Description Logic) [16] and editor Protégé [5].

3 VirDenT Ontology Construction

Nowadays, there are some top-level ontologies (such as DOLCE, SUMO and BFO) which describe very general concepts like space, time, matter, object, event, etc., i.e. independent concepts by a particular domain or problem. Among these, we used the DOLCE ontology [7] and its modules such as D&S [4], Temporal Relations, and so on. DOLCE is an ontology of particulars, in the sense that its domain of discourse is restricted to particulars. Other top-level ontologies might be used.

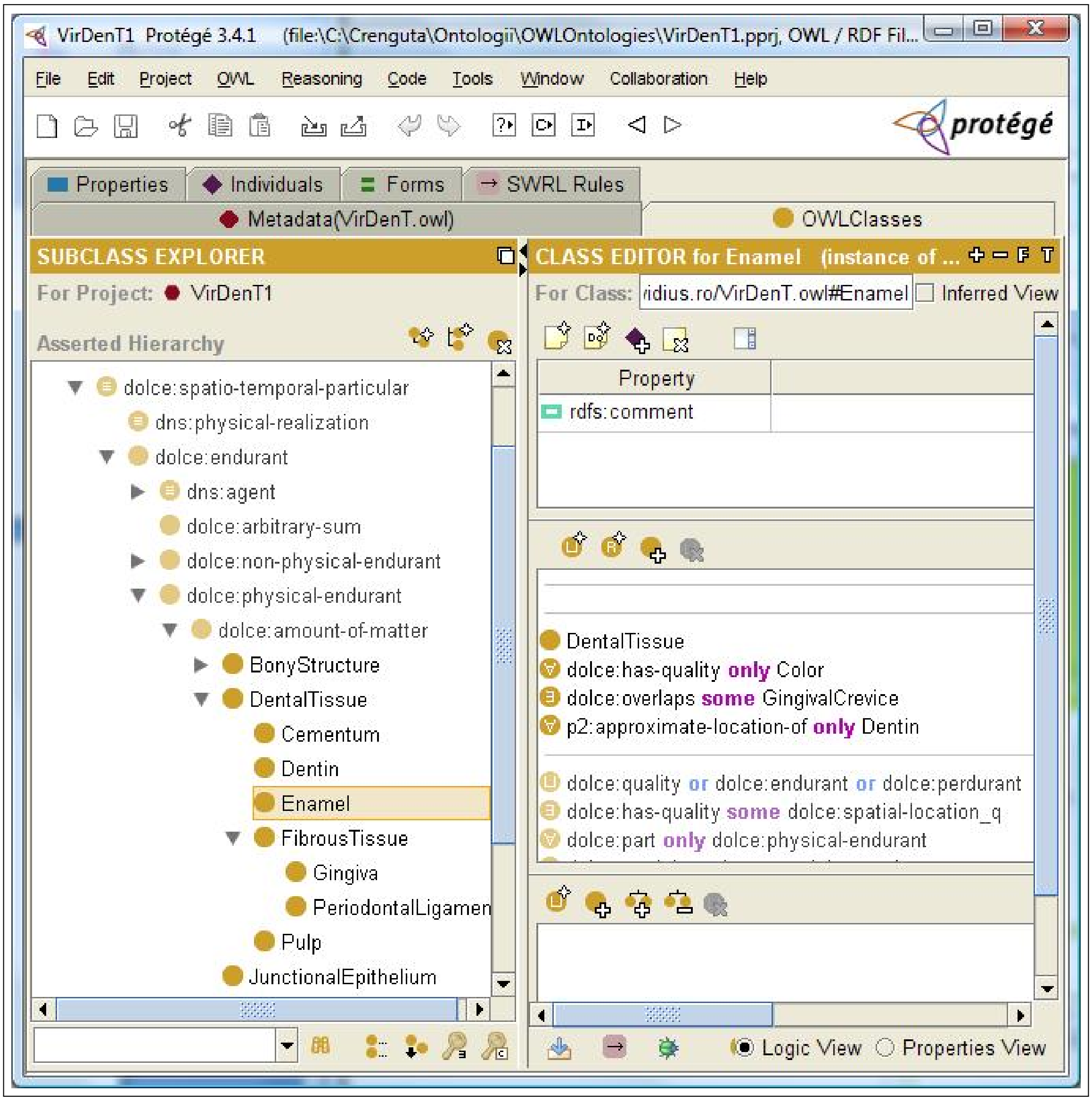
3.1 Identify Classes and Creating Taxonomy

First, classes are identified based on the UML class diagram of the model domain VirDenT system. So, each class in the class diagram has a corresponding OWL class, since an OWL class represents a set of individuals that form the extension of the concept mapped by class. Second, the most of the attributes of the classes have been transformed in physical, temporal and abstract qualities which are subclasses of the quality class of the DOLCE ontology. For example, Thickness and Form are two physical qualities, i.e. subclasses of the *dolce:physical-quality* class. Third, the activities of the business use cases are described ontologically as DOLCE processes, i.e. perdurants that fulfill the properties of cumulativity and homeomericity. For example, reduction is a process that executes during the ”Create orientation grooves for incisive decrease” business activity (Figure 2). We classified reduction as a process, because the sum of two reductions is still a reduction occurrence and there are temporal parts of a reduction that are not reductions. These classes form the basis of the VirDenT ontology as during construction of taxonomy and ontology appear other new OWL classes.

The classes are organized in a taxonomy created on the basis of the subsumption relation. Two classes A and B are linked by subsumption relation if and only if every individual (instance) of the B class is also an individual of the A class [7]. In this case, we can say that A subsumes B, or A is the superclass of the B class or B is a subclass of A. For example, the taxonomy of the amount of matters of the VirDenT ontology is presented in Figure 3.

3.2 Defining the Conceptual Relationships

Most of the conceptual relations of our ontology map the relations found in the DOLCE ontology and its sub modules, such as *inherent-in*, *generic-constituent-of*, *participate-in*, *generically-dependent-on* and so on. So, the qualities are linked by the concepts which they inhere in through the *has-quality* relation. For example, between the Enamel and Color concepts there is the *has-quality* relation (see Figure 3).



VirDenT1 Protégé 3.4.1 (file:\C:\Crenguta\Ontologii\OWLOntologies\VirDenT1.pprj, OWL / RDF Fil...
File Edit Project OWL Reasoning Code Tools Window Collaboration Help
protégé
Properties Individuals Forms SWRL Rules
Metadata(VirDenT.owl) OWLClasses
SUBCLASS EXPLORER
CLASS EDITOR for Enamel (instance of ...
Asserted Hierarchy
Logic View Properties View
Inferred View

For Project: VirDenT1

For Class: ıidius.ro/VirDenT.owl#Enamel

dolce:spatio-temporal-particular
- dns:physical-realization
- dolce:endurant
  - dns:agent
  - dolce:arbitrary-sum
  - dolce:non-physical-endurant
  - dolce:physical-endurant
    - dolce:amount-of-matter
      - BonyStructure
      - DentalTissue
        - Cementum
        - Dentin
        - Enamel
        - FibrousTissue
          - Gingiva
          - PeriodontalLigamen
        - Pulp
      - JunctionalEpithelium

Property	
rdfs:comment	

DentalTissue
dolce:has-quality **only** Color
dolce:overlaps **some** GingivalCrevice
p2:approximate-location-of **only** Dentin

dolce:quality **or** dolce:endurant **or** dolce:perdurant
dolce:has-quality **some** dolce:spatial-location_q
dolce:part **only** dolce:physical-endurant

Figure 3: A subtaxonomy of our ontology

The aggregation relations from the UML class diagram (Figure 1) have been transformed into the *temporary-component* D&S relation between the aggregate class and the class aggregated. For example, between the pair of concepts DiamondTool-Rod and DentalBur-Rod there is the *temporary-component* relation.

The composition relations from the class diagram (Figure 1) have been transformed into the *specific-constant-constituent* DOLCE relation which states that an entity consists of another entity for a period of time. For example, every individual of the AlveolarProcess concept is specific and constant constituent by individuals of the Alveolus concept.

Most of the associations of the UML class diagram (Figure 1) have correspondent imported ontological relations. For example, between the Enamel and Dentin concepts we have the *approximate-location-of* relation of the imported Spatial Relations ontology (Figure 3).

Observe in Figure 1 that each attribute has an UML data type, i.e. a classifier whose instances are values that attribute can take. Ontological modeling of data types of attributes is by datatype properties whose domains are the OWL classes associated with attributes and ranges are XML predefined data types. For example, the ThicknessValue is a datatype property that describes the data type of the thickness attribute of the Tooth class (Figure 1). The property domain is the Thickness class and range is the *nonNegativeInteger* predefined data type.

Furthermore, according to DOLCE ontology in any perdurant (process, state, etc.) at least one endurant participates in. For example, in the MakeGrooves process there are three participants: a groove, the incisal edge and a cylindrical-conical diamond tool with a flat top. So, any individual of the Groove concept participates in the MakeGrooves process for a period of time. This piece of knowledge is ontologically described introducing the DOLCE *participant-in* relation between the Groove and MakeGrooves concepts.

3.3 Ontological Description of the Business Rules

Business rules were identified during analysis of the VirDenT information system [2]. They describe knowledge that can be expressed ontologically using OWL DL or SWRL language. SWRL (Semantic Web Rule Language) allows us writing rules expressed in terms of OWL concepts: classes, properties and individuals [14]. For instance, the SWRL rule that expresses the piece of knowledge: "The incisal edge is decreased by 2 mm" is shown in the Figure 4.

Finally, we checked the ontology consistency with the help of the Protégé tool [5] version 3.4 and the Pellet reasoner system [11].

4 Related Work

There are few medical ontologies within dentistry. Park et al [10] created an OWL tooth ontology that embeds spatial relations. These relations describe the position of the tooth compared to other teeth. The relations derive from the formal relations in biomedical ontologies [12].

Pathogenic Pathway Database for Periodontitis was created based on an ontology that describes the molecular pathology of periodontal disease [13]. The database is used by programs which provide the following functionalities: displaying of the taxonomy in a tree-like view, key word search, showing causal relationships associated to a concept and a pathway browser.

SOMWeb is a semantic web-based system that provides IT support for clinicians and researchers in oral medicine to meet to review patient cases, establish a diagnosis, and decide on the most appropriate treatment plan for the patient. OWL ontologies are used in SOMWeb to represent oral medicine templates and knowledge, as well as to represent community models and data [3].

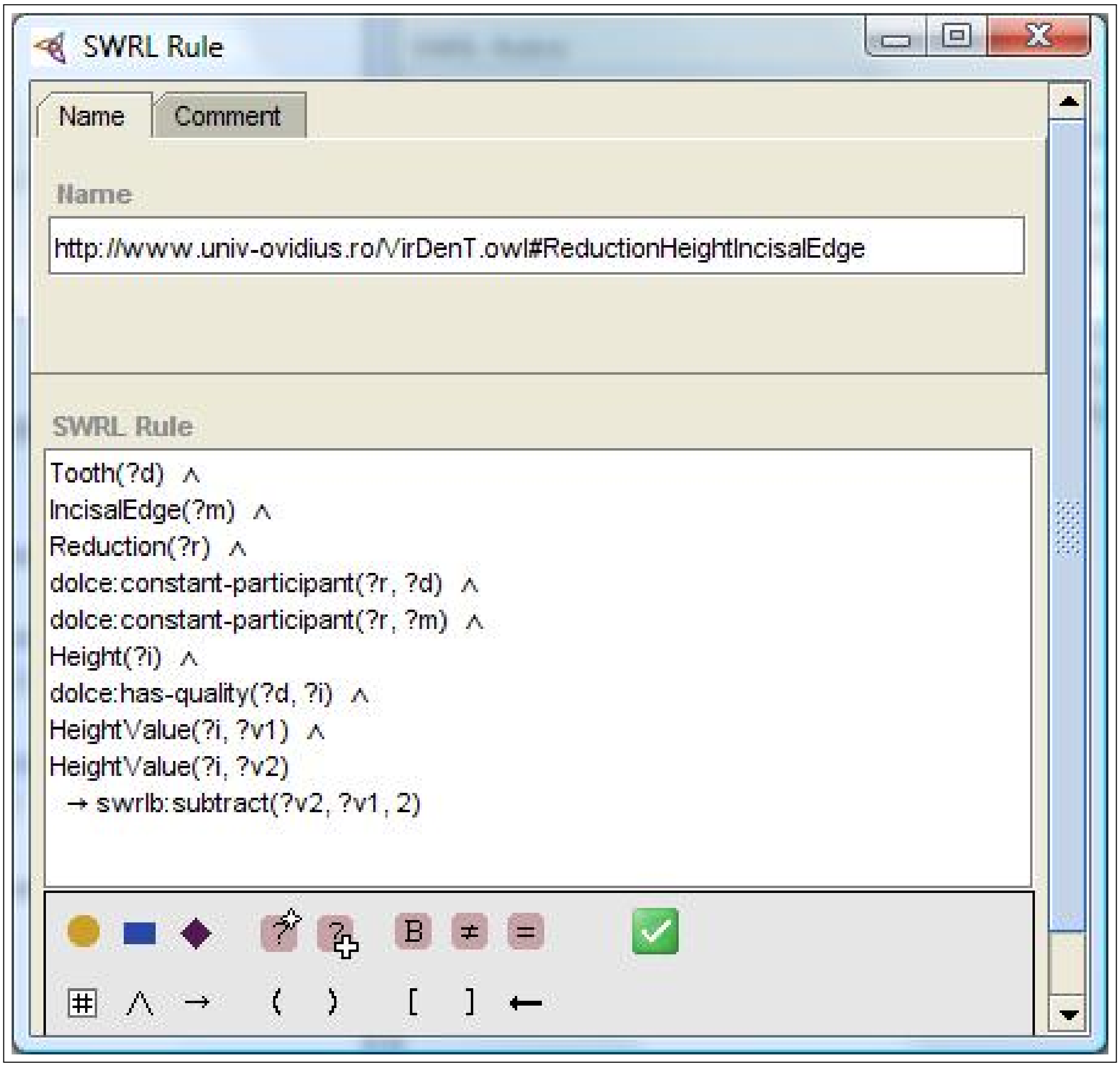
5 Conclusions and Future Works

We presented in this paper an ontology of the protocols for preparation of teeth for all-ceramic crowns. The ontology was developed based on the VirDenT information system.

Furthermore, we intend to use ontology to design the VirDenT software architecture and for the construction of a knowledge base that will form the persistence layer of the e-learning system.

Acknowledgements

We would like to thank Dr. C. Amariei, Dr. C. Ştefănescu, Dr. A. Zaharia and Dr. L. Petcu, professors at Faculty of Dental Medicine, Ovidius University of Constanta, Romania, for valuable comments to the study that forms the base of this paper. This research was supported by the Romanian Ministry of Education, Research and Innovation, National Centre of Programs Management, inside PNCDI-2 Partnerships program.



http://www.univ-ovidius.ro/VirDenT.owl#ReductionHeightIncisalEdge

```
Tooth(?d) ∧
IncisalEdge(?m) ∧
Reduction(?r) ∧
dolce:constant-participant(?r, ?d) ∧
dolce:constant-participant(?r, ?m) ∧
Height(?i) ∧
dolce:has-quality(?d, ?i) ∧
HeightValue(?i, ?v1) ∧
HeightValue(?i, ?v2)
 → swrlb:subtract(?v2, ?v1, 2)
```

Figure 4: The SWRL rule of a piece of knowledge

Bibliography

[1] C. M. Bogdan, A. Păduraru, Concern and Business Rule-Oriented Approach to Construction of Domain Ontologies, *International Journal of Computers, Communications and Control*, Vol III (2008), Suppl. Issue: S, pp. 185-189.

[2] C. M. Bogdan, D. M. Popovici, Information System Analysis of an E-Learning System Used for Dental Restorations Simulation, submitted to *Computer Methods and Programs in Biomedicine Journal*, 2009.

[3] G. Falkman, M. Gustafsson, M. Jontell, O. Torgersson, SOMWeb: A Semantic Web-Based System for Supporting Collaboration of Distributed Medical Communities of Practice, *Journal of Medical Internet Research*, 10(3):e25, 2008.

[4] A. Gangemi, P. Mika, Understanding the Semantic Web through Descriptions and Situations. In *Proceedings of the International Conference ODBASE03*, Italy, Springer, pp. 689-706, 2003.

[5] J. Gennari, M. Musen, R. Fergerson, W. Grosso, M. Crubzy, H. Eriksson, N. Noy, S. Tu, The evolution of Protégé-2000: An environment for knowledge-based systems development. *International Journal of Human-Computer Studies*, 58(1):89-123, 2003.

[6] N. Guarino, Formal Ontology and Information Systems. In *Proceedings of FOIS'98*, Trento, Italy, IOS Press, pp. 3-15, 1998.

[7] C. Masolo, S. Borgo, A. Gangemi, N. Guarino, A. Oltramari, WonderWeb Deliverable D18. Ontology Library. *IST Project 2001-33052 WonderWeb: Ontology Infrastructure for the Semantic Web*, 2003.

[8] N. F. Noy, D. McGuinness, A Guide to building ontologies: Ontology Development 101. A Guide to Creating Your First Ontology, March, 2001 at http://www.ksl.stanford.edu/people/dlm/papers/ontology101/ontology101-noy-mcguinness.html

[9] OMG, Unified Modeling Language Superstructure, version 2.0, ptc/03-0802, 2003.

[10] S. G. Park, H. G. Kim, M. K. Kim, Tooth Positional Ontology Represented in OWL, In *Medinfo 2007: Proceedings of the 12th World Congress on Health (Medical) Informatics; Building Sustainable Health Systems*, pp. 2288-2289. Kuhn, Klaus A (Editor); Warren, James R (Editor); Leong, Tze-Yun (Editor). Amsterdam: IOS Press, 2007.

[11] Pellet Reasoner, http://clarkparsia.com/pellet

[12] B. Smith, W. Ceuster, B. Klagges, J. Kohler, A. Kumar, J. Lomax, C. Mungall, F. Neuhaus, A. Rector, C. Rosse, Relations in biomedical ontologies, *Genome Biology*, 6(5):R46, 2005.

[13] A. Suzuki, T. Takai-Igarashi, Y. Numabe, Hiroshi Tanaka, Development of a database and ontology for pathogenic pathways in periodontitis, In *Silico Biology* 9, 0020, 2009, pp. 233 - 244.

[14] SRWL W3C Submission, http://www.w3.org/Submission/2004/SUBM-SWRL-20040521/.

[15] B. Touati, P. Miara, D. Nathanson, Esthetic Dentistry and Ceramic Restorations, Martin Dunitz Ltd., London, 1999.

[16] World Wide Web Consortium. OWL Web Ontology Language Reference. W3C Recommendation, 2004.

Int. J. of Computers, Communications & Control, ISSN 1841-9836, E-ISSN 1841-9844
Vol. VI (2011), No. 1 (March), pp. 53-62

Effective Retransmission in Network Coding for TCP

J. Chen, L.X. Liu, X.H. Hu, W. Tan

Jing Chen
Information Center
The Ministry of Science and Technology of the People's Republic of China
15#B, Fuxing Road, Beijing, 100862, P.R. China
E-mail: dongdcj@gmail.com

Lixiang Liu, Xiaohui Hu
Institute of Software, Chinese Academy of Sciences
4# South Fourth Street, Zhong Guan Cun Street, Beijing, China, 100190

Wei Tan
Baidu, Inc
E-mail: tanweildd@gmail.com

Abstract:
Incorporating network coding into TCP has the advantage of masking packet losses from the congestion control algorithm. It could make a lossy channel appear as a lossless channel for TCP, therefore the transport protocol can only focus on handling congestion. However, most schemes do not consider the decoding delay, thus are not suitable to be implemented in practical systems. We propose a novel feedback based network coding (FNC) retransmission scheme which has high throughput and quite low decoding delay without sacrificing throughput. It uses the implicit information of the seen scheme to acquire the exact number of packets the receiver needs for decoding all packets based on feedback. We also change the encoding rules of retransmission, so as to decode part of packets in advance. The scheme can work well on handling not only random losses but also bursty losses. Our scheme also keeps the end-to-end philosophy of TCP that the coding operations are only performed at the end hosts. Thus it is easier to be implemented in practical systems. Simulation results show that our scheme significantly outperforms the previous coding approach in reducing decoding delay, and obtains the throughput which is close to the scenarios where there is zero error loss. It is particularly useful for streaming applications.

Keywords: network coding, TCP, decoding delay, retransmission.

1 Introduction

It is well known that TCP suffers poor performance in lossy wireless networks. Even if a perfect congestion control algorithm can avoid congestion loss in transmission, there are still non-congestion losses (including random losses with a fixed bit error rate and bursty losses due to bad weather or signal shadowing, etc.), which necessarily degrade the transmission performance. Network coding has emerged as an important potential approach in the operation of communication networks [1]. The core idea is that the sender transmits coded packets combined with unacknowledged original packets rather than transmitting individual packets. Thus sending a packet can be seen as adding a packet to the pool and acknowledgement as removing the received packets from the pool. Each transmission is not affected by any other losses. Thus, incorporating network coding with TCP is a natural way to enhance the robustness and effectiveness of data transmission in lossy channels.

The ARQ for network coding (ANC) scheme presented in [2] defines the seen packet as an abstraction for the case in which a packet has not yet been decoded, but can be safely removed from the sender's buffer. The expected queue size of the scheme is reduced from the traditional length $\Omega((1-\varepsilon)^{-2})$, to $\Omega((1-\varepsilon)^{-1})$ where is the erasure probability. However, the weak receivers are unlikely to recover from erasures in reasonable time, since the decoding delay becomes very large.

TCP/NC [3] uses seen scheme to mask losses from the congestion control algorithm. It aims to make a lossy channel appear as a lossless channel to TCP, so the congestion control protocol does not need to pay attention to the error losses and thus can focus solely on the congestion. In fact, masking losses from TCP was considered earlier by using link layer retransmission [4]. Yet it has been noted in [5], [6] that the interaction between link layer retransmission and TCP's retransmission is complicated and the performance may suffer due to independent retransmission protocols at different layers. TCP/NC uses a constant redundancy factor for retransmission in order to compensate for the loss rate of the channel and match TCP's sending rate. However, if it suffers bursty losses due to bad weather or signal shadowing, the algorithm must wait a long time to decode all packets. In fact, most coding schemes do not take decoding delay into consideration. The receiver has to wait for a considerable number of packets before it can decode the data. Consequently, it is hard for them to deploy in a real system in spite of the benefits in terms of throughput and robustness the network coding can bring [7], [8].

The work by Barros et al. [9] redesigns the encoding rules and stages of the ANC scheme to decrease decoding delay, which is useful for streaming applications with special delay requirements. However, it reduces delay by sacrificing some of the throughput. And its expected queue size increases from $\Omega((1-\varepsilon)^{-1})$ in ANC to $\Omega((1-\varepsilon)^{-2})$, since the sender cannot discard a packet after it is being seen by all senders.

Therefore, when incorporating the existing coding mechanisms with TCP in wireless environment, we faced two main problems: (1) retransmission schemes which are triggered by a static parameter such as a fixed delay threshold or a redundancy factor, can not balance both throughput and decoding delay. This is shown in Sec V., Fig. 3 [9], and Sec 3 of our paper. (2) The retransmission schemes require the fixed loss rate and are deeply influenced by it. As a result it cannot handle bursty losses well.

To overcome the disadvantages in existing approaches, we provide the end-to-end Feedback based Network Coding (FNC) retransmission scheme which makes use of the implicit information of the seen scheme to obtain the exact number of packets needed by the receiver to decode all data as soon as possible. We also change the encoding rules to decode part of the packets in advance. Our effective retransmission scheme can mask losses better than the previous scheme, and hence significantly improve the performance of decoding delay and throughput under both random losses and bursty losses. Simulation results show that our scheme significantly outperforms the previous coding approach in reducing decoding delay while increasing throughput. It obtains the throughput which is close to the scenarios where there is zero error loss.

The remainder of the paper is organized as follows. Section 2 introduces the terminology and describes basic ideas of network coding in TCP with the seen scheme. Section 3 proposes our new network coding retransmission scheme FNC. The corresponding simulation results are presented in Section 4. We conclude the paper in Section 5.

2 Essential Background

We treat packets as vectors over a finite field F_q of size q. The k^{th} packet generated by the source has an index k and is denoted as p_k. When the source is allowed to transmit, it sends a random linear combination of all packets instead of the original packet. We will firstly explain the seen scheme with logical description and then illustrate its implementation in the existing protocol stacks.

2.1 ARQ for Network Coding

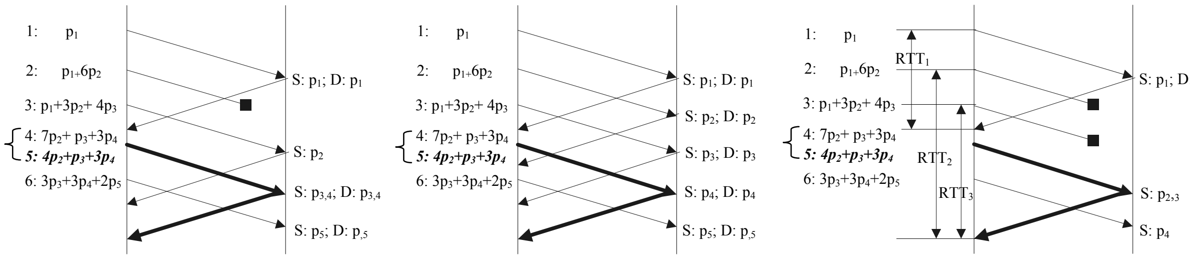
The ARQ for network coding (ANC) scheme presented in [2], is designed to handle loss in a multicast environment. In the scheme, the decision of which packets to combine relies on the concept of the seen packets. A packet p_k is said to be seen by a receiver if it has enough information to compute a linear combination in the form of p_k + q, in which q is a linear combination of packets that are newer than p_k, i.e. q = $\sum_{l>k} \alpha_l \cdot p_l$, with $\alpha_l \in F_q$ for all $l > k$. The receiver acknowledges the oldest unseen packet, so the sender always transmits a packet that is a combination of the oldest unseen packets of each receiver. A packet can be dropped from the sender queue when it is seen by all receivers. This provides an efficient method to keep the sender's queue sizes small, although the receiver may not decode all packets. It is demonstrated to be throughput optimal for the case of Poisson arrivals, perfect feedback, and identical erasure probabilities on all channels, because each reception is innovative.

2.2 Network Coding for TCP

The reference system for our scheme is the TCP/NC protocol [3], which is designed with respect to a single source that generates a stream of packets to one sink. It incorporates the seen scheme with congestion control and introduces a new network coding layer between the transport layer and the network layer in the protocol stack, which masks packet losses from congestion control algorithm.

There are several modifications on ANC to be fit for end-to-end connection. First, the sender transmits random linear combinations of packets in the coding window, instead of combinations of the oldest unseen packets of each receiver. The receiver acknowledges the oldest unseen packet although it may not be decoded yet. There will never be any duplicate ACKs as each reception is innovative. Every ACK will cause the congestion window to advance, so it is not proper to apply fast retransmit/recovery algorithms which use three duplicate ACKs as the packet loss indication. Therefore TCP/NC chooses TCP Vegas [12] as the congestion control approach and introduces a novel RTT estimation algorithm. It matches the newly arrived ACK with the transmission that occurred after the one that triggered the previous ACK, rather than the transmission that triggered this ACK. For example, in Figure 1(c), RTT_2 matches the 2^{nd} transmission, rather than the 4^{th} transmission.

In the implementation, TCP/NC embeds the network coding operations in a separate layer below TCP and above IP on two end nodes in order to naturally add network coding to the current protocol stack. In the source side, when a packet arrives at the coding layer from the transport layer, the coding layer generates a random linear combination of the packets in the coding window and sends it to the sink. In order to compensate for the loss rate of the channel and to match TCP's sending rate, for every packet from TCP, R linear combinations are sent to IP on average, where R is the constant redundancy factor equal to the reciprocal of the probability of successful reception. As an example shown in Figure 1(a), assuming packet loss rate e to be 20%, hence R is 1/0.8=1.25, which means that the sender makes a retransmission after sending four combinations triggered by TCP. The retransmission does not include any new



(a) proper retransmission (b) redundant retransmission (c)insufficient retransmission

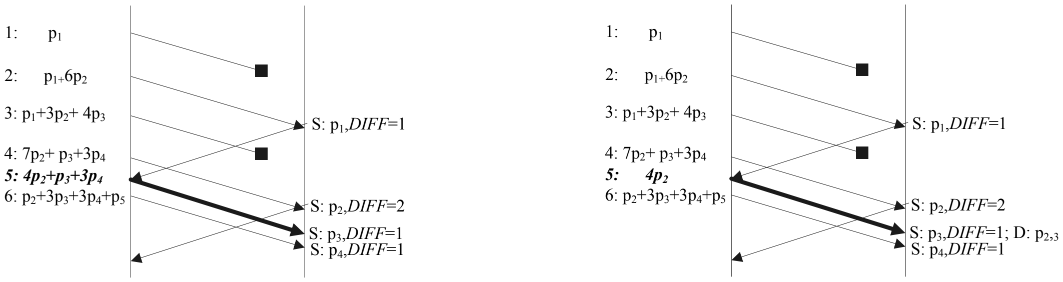
Figure 1: Example of the transmission with a redundancy factor

original packet from TCP. In Figure 1(a), the 5^{th} transmission is a retransmission which only encodes p_2, p_3 and p_4, without p_5. It compensates the loss of the 2^{nd} transmission, makes p_3, p_4 to be seen and p_3, p_4, p_5 decodable. However, the retransmission with a fixed rate may result in the following two problems: 1) the retransmission may be useless. For example, in Figure 1(b), the 5^{th} retransmission is useless since no losses happened before it. We call this redundancy or redundant retransmission. 2) The retransmission may be helpless. In Figure 1(c), the 5^{th} retransmission is valid but still can not help the receiver to decode all packets as there are two combinations lost before. Therefore, it is natural to think about using a feedback based retransmission scheme to replace the scheme with a fixed retransmission rate.

3 Network Coding Retransmission Scheme

To reduce the decoding delay and redundancy, we propose the feedback based network coding (FNC) retransmission scheme. It acknowledges the total number of coding packets required to repair a loss and then decodes all packets. In the receiver, the difference between the number of seen packets and the largest packet index in the coefficient matrix implies the number of packets that the source needs to retransmit. So we maintain two variables in the receiver side: one is the number of seen packets $SEEN\_CNT$; the other is the largest index of the received packets MAX_SEQ. For each ACK, the receiver embeds in the header not only the sequence number that equals the oldest unseen packet, but also the difference between MAX_SEQ and $SEEN\_CNT$, which is called $DIFF$. For example, suppose the source transmits the following linear combinations: $x = p_1, y = (p_1 + p_2)$ and $z = (p_1 + p_2 + p_3)$. The second transmission y is lost. So the sink only receives the linear combinations x and z. As p_1 and p_2 have been seen, $SEEN\_CNT$ is 2, the largest index MAX_SEQ is 3, and hence $DIFF = 3 - 2 = 1$. Thus $DIFF$ indicates the number of packets the receiver needs to change into the decodable state which means the receiver can decode all packets. In the sender side, when an ACK arrives, if the $DIFF$ is larger than zero, the sender uses this value to decide to retransmit in the following two steps: First, it checks to see if the difference between the current time T_{now} and the time of the last retransmission T_{last} is greater than the timeout value. If it is, the sender retransmits $DIFF$ linear combinations of first $DIFF$ packets in the coding window. This avoids retransmission more than once for losses that occurred during one RTT interval. If it is not, the sender compares the $DIFF$ received this time with the previous retransmission $LAST\_DIFF$. If the current $DIFF$ is greater than the $LAST\_DIFF$, there is new packet loss and it retransmits $(DIFF - LAST\_DIFF)$ random linear combinations of the first $(DIFF - LAST\_DIFF)$ packets in coding window.

If we try to retransmit the combinations of all packets in the coding windows, as in TCP/NC [3], then the average decoding delay of our scheme is indeed lower than NC. However, it is still



(a) coding with all packets (b) coding with the first several packets

Figure 2: Retransmission based on feedback

much higher than our expectation. Given an example in Figure 2(a), if the sender retransmits the coded packet combined with p_2, p_3 and p_4 in the current coding window in the 5^{th} transmission, the receiver is still in undecodable state and cannot decode any packets, because the $DIFF$ has changed to 2 due to the loss of the 3^{rd} transmission. Therefore, we improve the encoding rule in which retransmission packets are only combined with first $DIFF$ (or $DIFF - LAST\_DIFF$) packets in the coding window in order to make a part of packets decodable. Although we do not decode all packets together, we significantly reduce the number of undecodable packets as well as the decoding delay. Additionally, encoding less packets can reduce the overhead and complexity of encoding and decoding operations. Consequently, in the example of FNC in Figure 2(b), we only retransmit the combination containing p_2 (as $DIFF$ equals to 1) in the 5^th transmission, and thus p_1 and p_2 are decodable.

Additionally, TCP/NC simply retransmits packets through redundancy factor to compensate for error losses, rather than for congestion losses. The retransmission in the coding layer does not consume the congestion window. However, our previously mentioned algorithm treats congestion losses and error losses the same, which may bring or exacerbate congestion. We can solve this problem by limiting the current total number of retransmission packets in the coding layer with no more than the product of the total number of transmitted packets N and the loss rate e. In contrast with TCP/NC that retransmits dispersedly, our policy can retransmit the appropriate number of packets together while they are required, and does not mask congestion losses at the same time. In other words, NC (i.e. TCP/NC) is a static retransmission scheme whereas FNC is a dynamic one. However, regarding those near-zero congestion loss algorithms such as VCP [10] and MLCP [11], it is unnecessary to adopt this policy. Thus, it can handle such losses much better, when it encounters burst or inconstant loss rate conditions.

The improved algorithm is specified below using pseudo-code:

```
Source side:
Initialization:
    Set $LAST\_DIFF$ and $T_{last}$ to 0.
ACK arrives from receiver:
    The source side algorithm removes the seen packet from the buffer and retrieves $DIFF$
    from ACK header. If this is the fourth uninterrupted time that $DIFF$ is larger than 0:
    1) If $(T_{now} - T_{last} > RTT)$
        a) Set $LAST\_DIFF$=0;
        b) goto 3);
    2) If $DIFF <= LAST\_DIFF$ goto 5).
    3) Repeat the following $(DIFF - LAST\_DIFF)$ times:
        a) Generate a random linear combination of the packets in the coding window.
        b) Deliver the packet to the IP layer.
```

4) Update T_{last} to the current time;
5) Update $LAST\_DIFF$ to the $DIFF$ of the new arrival.

Receiver side:

Initialization:

Set $SEEN\_CNT$ and MAX_SEQ to 0.

Packet arrives from source side:

1) Performs Gaussian elimination to update the set of seen packets.
2) Update $SEEN\_CNT$ and MAX_SEQ.
3) Add the network coding ACK header to TCP ACK, consisting of the value of $DIFF$ which is the difference between MAX_SEQ and $SEEN\_CNT$.

The algorithm not only avoids the redundant transmissions when the receiver is in the decodable state, but also retransmits the appropriate number of packets so as to make a part of packets decoded in advance. It significantly reduces the decoding delay, which is particularly useful for streaming applications with stringent delay requirement. Also, as we do not change the acknowledgment scheme of seen packet, the expected queue size for our scheme remains $\Omega((1-\varepsilon)^{-1})$ rather than $\Omega((1-\varepsilon)^{-2})$ in SNC [9] and TCP.

In contrast with traditional retransmission schemes, such as SACK etc., our scheme obtains the total number of packets for decoding. It significantly reduces overheads in the ACK header. It is also more robust and effective. Compared with TCP/NC [3], FNC has a relative small decoding delay. It can handle both random losses and unknown bursty losses. Furthermore, our scheme respects the end-to-end philosophy of TCP that coding operations are only performed at the end hosts while achieving the aim of masking losses from congestion control algorithm at the same time.

4 Simulation Results

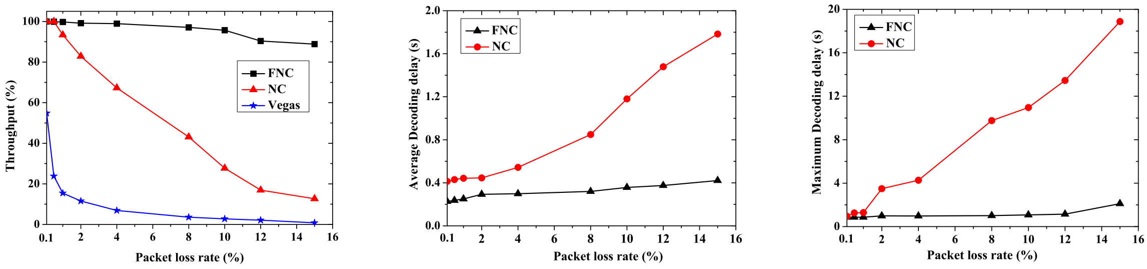
We evaluate the performance of different coding algorithms by means of the network simulator "ns-2" [13]. The basic setting is a tandem network consisting of 4 hops. The source and sink nodes are at opposite ends of the chain. The packet size is 1000 bytes. We incorporate network coding with TCP Vegas. The Vegas parameters are set as $\alpha = 28, \beta = 30, \gamma = 2$. The performance metrics are the throughput, the average decoding delay and the maximum decoding delay. The throughput with network coding is calculated as the total number of seen packets, rather than the decoded packets, divided by the simulation time. All simulations are run for at least 200s to ensure that the system reaches its steady state.

4.1 Random Losses

We first evaluate the performance of FNC under the case of a fixed packet loss rate. The basic setting is a 10Mbps link capacity, an 80ms round-trip time, and a 5% packet loss rate. We compare the throughput of NC, FNC and Vegas under a fixed loss rate with the standard Vegas under no loss rate to evaluate their performance on masking losses. Each scenario runs 20 times and we get the mean value.

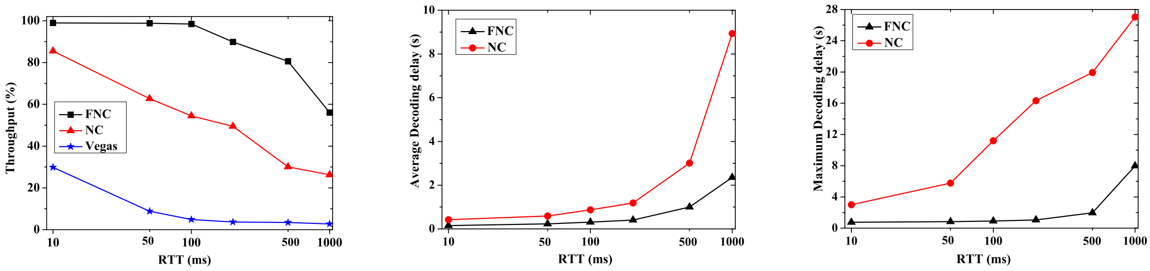
4.2 Impact of Packet Loss Rate

We first study the variation of throughput with loss rate from 0% to 15%. Figure 3(a) shows that the throughput of Vegas falls rapidly as losses increase. NC performs better than Vegas but it is worse than FNC since it only successfully masks part of losses. In contrast, FNC is very robust to losses. It maintains over 88% throughput as if there is no error loss. Our



(a) Throughput (b) Average decoding delay (c) Maximum decoding delay

Figure 3: Performance as a function of the loss rate variation



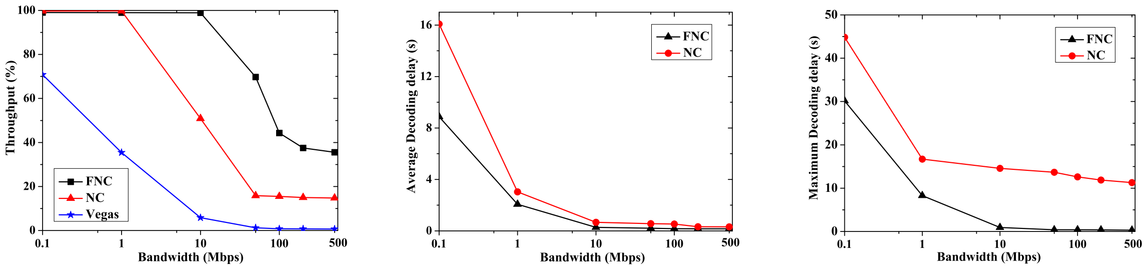
(a) Throughput (b) Average decoding delay (c) Maximum decoding delay

Figure 4: Performance as a function of RTT variation

scheme also keeps quite a low decoding delay and the average value does not increase as the loss rate increases. This is because FNC is according to the actual losses to retransmit packets dynamically. In contrast, NC is deeply affected by the loss rate since its retransmission is static. In some cases, the average decoding delay of NC is even higher than the maximum decoding delay of FNC.

4.3 Impact of Feedback Delay

Next, we evaluate the performance across a wide range of propagation delay from 10ms to 1000ms. As illustrated in Figure 4(a), our scheme performs much better than NC and Vegas in every case. The throughput in FNC is at least 2 times more than in NC. However, the FNC's throughput degrades significantly when RTT increases to 1000ms. This is because our retransmission scheme is based on feedback. NC may perform better than FNC when RTT becomes much larger. Thus our retransmission scheme can not be applied in deep space communication of which RTT may be hundreds of seconds, whereas in most real cases, RTT is no more than 1000ms. Figure 4(b) shows that the average decoding delay of NC is 2 to 10 times of FNC. The decoding delay for both schemes increases quickly when RTT grows beyond 500ms. When RTT is 500 ms, FNC's average decoding delay is 1.00165s, and when RTT is 1000ms, its average decoding delay is 2.35303s. They are approximately 2 times that of RTT, which is in accordance with our expectations. The average decoding delay of NC is 3.0087s and 8.93286s respectively, much larger than FNC.



(a) Throughput (b) Average decoding delay (c) Maximum decoding delay

Figure 5: Performance as a function of the bandwidth variation

4.4 Impact of Link Capacity

We fix the round-trip time to 100ms with a packet loss rate of 5%, and vary the link capacity from 0.1Mbps to 1000Mbps. As shown in Figure 5(b) and (c), the maximum decoding delay for NC is very high in every case, whereas both the maximum and the average decoding delay for FNC are quite low. When the bandwidth is lower than 1Mbps, the congestion control algorithm itself leads to congestion or even loss, which increases the delay. If the bandwidth is very high, one loss can lead to great throughput decrease. It is shown in Figure 5(a) that due to the limitation of Vegas itself, the performance of all three schemes is not very high when the bandwidth is larger than 100Mbps. Since our scheme can mask packet losses more effectively, it has the highest throughput in contrast with NC and Vegas without incorporating with network coding.

Overall, the decoding delay for FNC depends on the RTT, the overhead of operation of encoding and decoding rules, and the congestion control algorithm, but it is not deeply affected by the loss rate. It seldom has a long undecodable chain thus its maximum decoding delay is endurable. Furthermore, FNC masks random losses more effectively than NC does, and hence obtains higher throughput in most cases.

4.5 Bursty Loss

All previous simulations focus on the behavior of FNC under the fixed loss rate. Now, we investigate its performance in an unknown environment with unfixed loss rate. The setting of this scenario is a 10Mbps link capacity with 80 ms RTT, where the background loss rate is 0.1%. At t=40s, the loss rate is changed to 50%, and lasts for 5 seconds until t=45s. At t=60s, the loss rate is changed to 100% which means the signal is fully shadowed by obstacles, and it lasts for 2 seconds until t=62s. The tested protocols Vegas and FNC do not know these loss rate changes. Figure 6 clearly shows that FNC can quickly and effectively handle the sudden bursty losses. Before t=40s, FNC almost masks all random losses whereas Vegas without network coding is seriously affected by random losses. When the loss rate is changed to 50%, Vegas could hardly work whereas FNC still has a relatively high throughput. FNC does not mask all losses because the retransmission packet also has a 50% probability to be lost. After around t=63s, the sending rate of FNC has a sudden increase because the sender retransmits all lost packets together. Due to the limitation of the congestion control algorithm Vegas itself, the stable sending rate after this burst does not recover as before. The maximum decoding delay of FNC is 3.33053s, which contains 2 seconds during which the connection is broken. The average decoding delay is 0.0839042s, a little more than one RTT. We do not test NC because its retransmission scheme depends on the estimation of the loss rate, and it is hard to give an effective estimation algorithm.

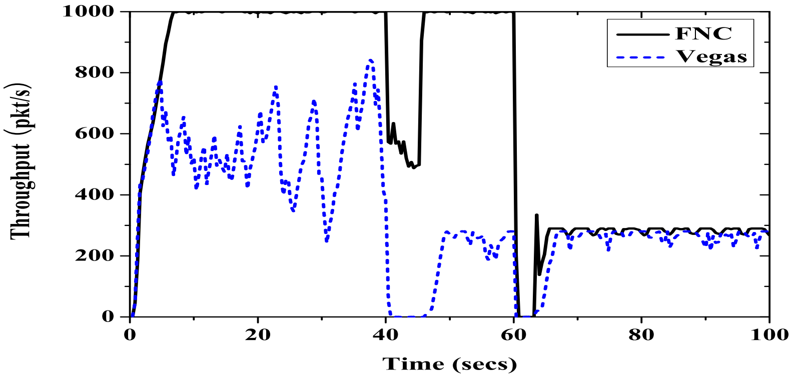


Figure 6: Performance under unknown bursty losses

5 Conclusions

Network coding is a powerful tool in fighting against non-congestion losses. However, its redundancy and decoding delay can significantly impair transmission performance so that most schemes cannot be implemented in practical systems. In our work, we propose a novel dynamic network coding retransmission scheme which makes use of the information implied in the seen scheme to acquire the exact number of packets the receiver wants instantly. As we do not retransmit packets with a stable rate or a constant redundancy factor, our approach can handle not only random losses, but also unknown bursty losses. Simulation results show that our scheme significantly outperforms the previous coding approach in reducing decoding delay and masking losses. It obtains the throughput which is close to the scenarios where there is zero error loss.

The remaining issue in our research is to evaluate the performance of incorporating network coding with other congestion control algorithms, such as those load factor-based algorithms, VCP and MLCP, etc. Furthermore, we intend to implement our algorithm in a Linux protocol stack to asses its strengths and limitations in practice.

Bibliography

[1] T. Ho, Networking from a network coding perspective, PhD Thesis, Massachusetts Institute of Technology, Dept. of EECS, May 2004.

[2] J. K. Sundararajan, D. Shah, M. Medard, ARQ for network coding, *in IEEE ISIT 2008*, Toronto, Canada, Jul, 2008.

[3] J. K. Sundararajan, D. Shah, M. Medard, M. Mitzenmacher, and J. Barros, Network Coding Meets TCP, *in IEEE INFOCOM, 2009*, San Francisco, USA, Apr 2009.

[4] S. Paul, E. Ayanoglu, T. F. L. Porta, K.-W. H. Chen, K. E. Sabnani, and R. D. Gitlin, An asymmetric protocol for digital cellular communications, *in Proceedings of INFOCOM, 1995.*

[5] A. DeSimone, M. C. Chuah, and O.-C. Yue, Throughput performance of transport-layer protocols over wireless LANs, *IEEE Global Telecommunications Conference (GLOBECOM '93)*, pp. 542-549 Vol. 1, 1993.

[6] H. Balakrishnan, S. Seshan, and R. H. Katz, Improving reliable transport and handoff performance in cellular wireless networks, *ACM Wireless Networks*, vol. 1, no. 4, pp. 469-481, 1995.

[7] S. Katti, H. Rahul, W Hu, Databi, M. Mcdard, and J. Crowcrofg, XORs in the Air: Practical Wireless Network Coding, *in IEEE/ACM Transactions on Networking*, 16(3): 497-510, 2008.

[8] C. Fragouli, J.-Y. Le Boudec, and J. Widmer, Network coding: Aninstant primer, *ACM Computer Communication Review*, Jan. 2006.

[9] J. Barros, R. A. Costa, D. Munuaretto, and J. Widmer, Effective Delay Control in Online Network Coding, *in IEEE INFOCOM, 2009*, San Francisco, USA, Apr 2009.

[10] Yong Xia, L. Subramanian, I. Stoica, S. Kalyanaraman, One More Bit is Enough, *in IEEE/ACM Trans. Networking*, 16(6):1281-1294, Dec 2008.

[11] I. A. Qazi, and T. Zuai, On the Design of Load Factor based Congestion Control Protocols for Next-Generation Networks, *in IEEE INFOCOM*, Apr 2008.

[12] L. S. Bramko, S. W. O'Malley, and L. L. Peterson, TCP Vegas: New Technichques for Congestion Detection and Avoidance, *in Proceedings of the SIGCOMM '94 Symposium*, August 1994.

[13] ns-2 Network Simulator, http://www.isi.edu/nsnam/ns/

Int. J. of Computers, Communications & Control, ISSN 1841-9836, E-ISSN 1841-9844
Vol. VI (2011), No. 1 (March), pp. 63-71

Software Components for Signal Fishing based on GA Element Position Optimizer

N. Crisan, L.C. Cremene, M. Cremene

Nicolae Crisan, Ligia Chira Cremene,
Marcel Cremene
Technical University of Cluj-Napoca Romania,
Cluj-Napoca, Memorandumului nr. 28, 400114
E-mail(s): Nicolae.Crisan@com.utcluj.ro,
Ligia.Cremene@com.utcluj.ro,
Marcel.Cremene@com.utcluj.ro

Abstract: Long-term adaptation solutions do not receive much attention in the design phase of a wireless system. A new approach is proposed, where the antenna takes an active role in characterising and learning the operation environment. The proposed solution is based on a signal fishing mechanism. Several software components, among which a genetic optimizer, implement the processing stages of autonomous design of the antenna array during operation.

Keywords: smart antennas, signal fishing, long-term dynamic adaptation, GA optimizer

1 Introduction

Wireless receivers incorporate more and more adaptive techniques in the attempt to compensate the effects of the radio channel. Technical challenges for current wireless technologies include limited bandwidth and transmit power, interference, signal fading and other propagation impairments.

A multitude of adaptive techniques are nowadays implemented at different levels of a wireless transmission chain, as depicted in figure 1. At the physical layer, data link and network layers there are many adaptive techniques that deal with modulation and coding schemes, equalization, filtering, ARQ mechanism, fragmentation, routing [9], [10]. These techniques have more or less independent approaches, and their joint operation is not much studied. Adaptive solutions are usually specific solutions, having a local effect. Most of the existing adaptive solutions are hard-coded and therefore not able to evolve.

The main concerns regarding wireless receiver performance are link availability and link capacity. Tens of adaptive techniques are trying to dynamically adapt the transmission to the changes in the wireless environment. These are short-term, instant adaptation methods/mechanisms. Long-term adaptation solutions are more complex and their design and operation are time consuming.

The majority of the adaptation methods considering the spatial component of the propagation channel use highly-parameterized, yet simplified channel models [3], [4]. Usually fixed parameters, such as mean values of statistical channel parameters are involved [5]. Also, processing methods that consider time and frequency variability of the channel exploit only partially the spatial component information.

An approach for developing a long-term dynamic adaptation solution for wireless receiver chains is proposed. Unlike current approaches the antenna takes an active role in characterising and learning the operation environment, namely the wireless propagation channel.

The proposed solution is based on a spatial characterization of the fading allowing the detection of signal maxima in the antenna environment. The antenna is described as an *MxN* rectangular array. Only two elements in the array are active at a given time. In order to search for the best positions of the active elements a genetic optimizer is used.

This paper is organized as follows: Section 2 discuses the interdisciplinary aspect of the smart antenna field. In section 3 we describe several software components that implement the processing stages of the autonomous design of the antenna array during operation. The proposed solution is based on a signal fishing mechanism [6] and uses a genetic optimizer. Experimental results are presented in section 4 together with our conclusions.

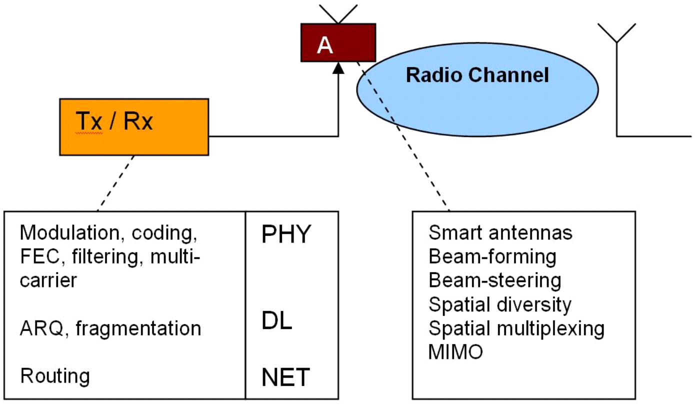


Figure 1: Adaptive techniques operating at different levels of the wireless Tx-Rx chain

2 Smart antenna interdisciplinary aspect

Classical smart antennas are based on antenna arrays of different configurations and controlled by means of a specialized signal processor. Their beamforming and beamsteering mechanisms address capacity and reliability issues of wireless systems [14], [15]. The subject of smart antennas, also known as adaptive antennas, becomes a widely interdisciplinary field as it bridges several disciplines: electronics, electromagnetics, channel&propagation modelling, communication techniques – signal processing, control&adaptive systems, random processes, and evolutionary computing.

Antenna, communications, and control engineers tend to view adaptive antennas from quite different perspectives. Some are inclined to focus on electromagnetic features (e.g. radiation patterns and levels), others on communication-link parameters like SNR (Signal-to-Noise Ratio) and BER (Bit Error Rate), and others on various control algorithms.

There are at least three issues that computational intelligence (especially evolutionary computing) can help solve in smart antenna systems. The first direction is generating new, optimized antenna geometries, capable of multi-mode operation. The other two are dynamic optimization issues that concern signal detection and tracking and statistical signal processing.

We identify the antenna as a pivotal element in determining and assessing quality in wireless communication systems. Actually, the user-perceived network or equipment quality relies, ultimately, on antenna performance. Moreover, until recently, the antenna influence on channel measurements for channel modelling was considered a bias, now it can be used to a benefit by integrating it into the channel analysis [3], [4].

The distance between transmitter and receiver, their speed, their transmit powers, obstacles, all have an impact on the channel matrix [7], [2], [1]. These are difficult to control factors. Yet, there is a system that has a great influence on the channel matrix, a system that is under the control of the receiver: it is the receiver antenna array. The position of the antenna array in the field and the spacing between array elements are the main parameters that can influence the channel matrix.

3 Proposed solution: Signal fishing based on GA element position optimizer

Signal fishing (SF) is a new concept [6] that addresses fading mitigation in adaptive multiple-antenna receivers. The SF mechanism is based on spatial fading characterization relaying on the antenna array itself. The main idea is to detect and exploit channel signal maxima.

Fading is a random fluctuation of amplitude and phase of the propagated signal. Amplitude fluctuation in a multipath environment is usually described by a Rayleigh distribution. A useful fading characterization will turn the selective-fading channel problem into a flat-fading problem, and will enable the receiver signal processing part to perform better [6].

Antenna selection techniques generally consider a fixed number of elements in the antenna array allowing only regular configurations. Signal fishing allows a random activation of a variable number of array elements which is equivalent to dynamically modifying the element spacing d and the orientation angle Ω.

Figure 2 a) illustrates the description of the receive rectangular array which is part of a transmit-receive system operating in a MIMO 2x2 configuration. Each element is a Sierpinski gasket (Fig. 2b). Two elements are selected at a specific moment of time and six wave-fronts are considered. The two array elements operate as receive antennas; the other elements are used for channel estimation and spatial fading characterization and are uniformly spaced (d is constant only for these *Np-2* elements). Spatial fading characterization involves estimation or measuring of the angle of arrival θi,, phase shift φi, and amplitude ai of the incoming signal which is performed during the training sequence. When the training sequence elapses, the channel is considered quasi-static for about 448 symbols. By changing the element displacement (element perturbation) we can introduce a phase difference between arriving signals *Rx1* and *Rx2*.

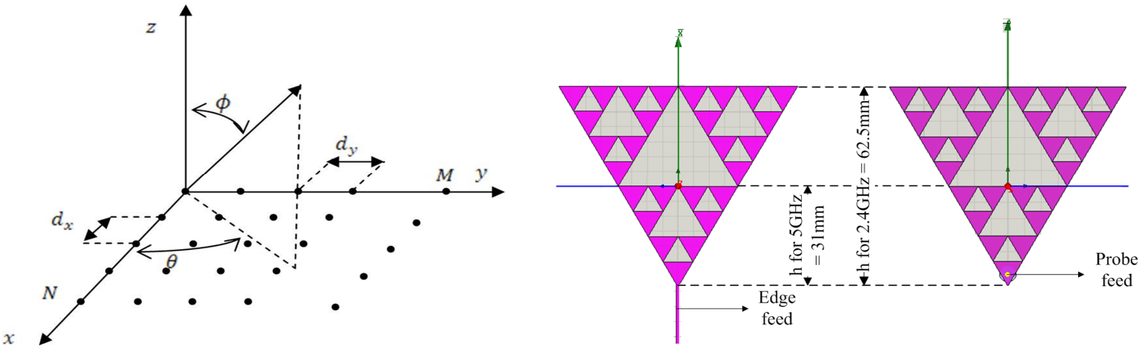


Figure 2: a) 32x32 rectangular antenna array $\phi = 90^0$, $0 \leq \theta \leq 90^0$, $d_x = d_y = \frac{\lambda}{4}$, $M = N = 32$. b) Sierpinski gasket multi-band antenna

The signal equations for each of the two receive elements, *Rx1* and *Rx2*, are:

$$R_{x1}\left(a,b\right)=\sum_{i=1}^{n_p}\rho_{\left(\theta_i,90^0\right)}f_i exp\left[j\left(\left(a-1\right)\left(d_x\cos\theta_i\ \right)\right)+j\left(b-1\right)\left(d_y\sin\theta_i\ \right)\right] \tag{1}$$

$$R_{x2}\left(c,d\right)=\sum_{i=1}^{n_p}\rho_{\left(\theta_i,90^0\right)}f_i exp\left[j\left(\left(c-1\right)\left(d_x\cos\theta_i\ \right)\right)+j\left(d-1\right)\left(d_y\sin\theta_i\ \right)\right]$$

where: ρ is the antenna element factor
$f_i = |f_i| \exp(j\phi_i)$ is the complex value associated to wavefront i.
a, b, c, and d are the coordinates of the activated receive elements *Rx1* and *Rx2*
β is the phase constant
θi is the angle of arrival (AoA) of the *ith* front.

Figure 3 presents an illustration of the signal fishing concept – exploiting the spatial component information of the channel in order to detect and use the signal maxima. Signal maxima are estimated based on the optimum element spacing

$$d_{opt} = \lambda \frac{\sum_{i=1}^{N_p} |h_i| \frac{4K \pm 1}{4 sin\theta_i}}{\sum_{i=1}^{N_p} |h_i|} \tag{2}$$

where K is a positive integer; the array grows wider with K.

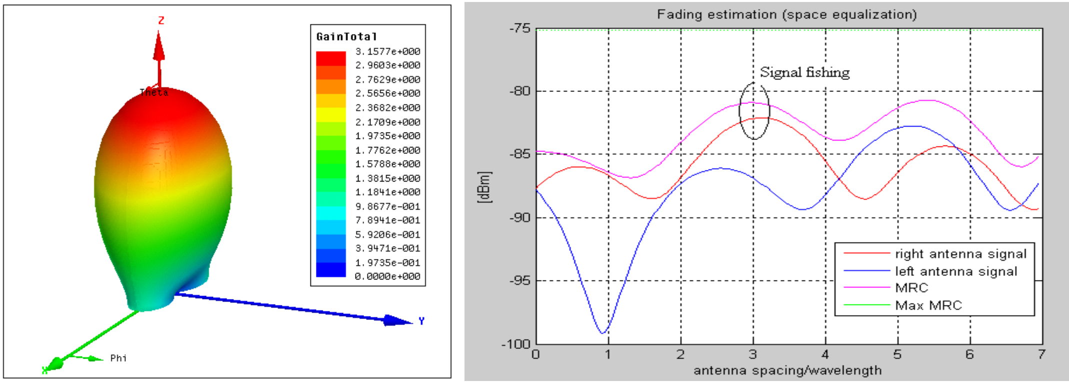


Figure 3: Signal fishing with two antennas

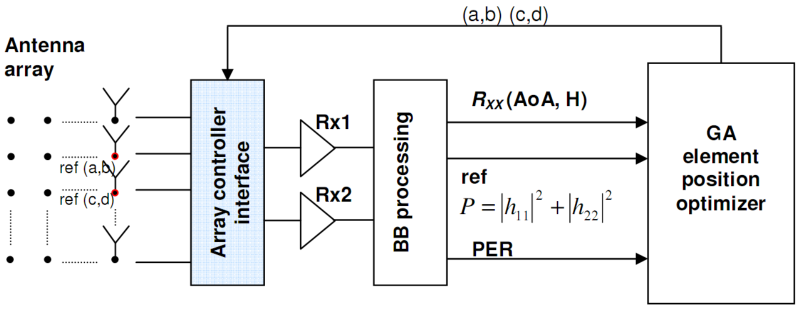

(a,b) (c,d)
Antenna array
ref (a,b)
ref (c,d)
Array controller interface
Rx1
Rx2
BB processing
R_{XX}(AoA, H)
ref
$P = |h_{11}|^2 + |h_{22}|^2$
PER
GA element position optimizer

Figure 4: Signal fishing based on GA element position optimizer

Figure 4 shows the block diagram of the signal fishing mechanism realization. The main components are: an *Np*-element antenna array, two receiver chains, an array controller, a baseband processing component and a genetic optimizer.

The *array controller* plays a key role in the antenna selection mechanism. The array controller can be located between the antenna array and the receiver RF chain. Antenna selection control is based on a switching algorithm that activates two elements at a time. The array controller is the software component that translates the coordinates a, b, c, d into a binary signal suited for on/off element switching. This is done via a microcontroller located on a Software Defined Radio board. This algorithm can be implemented in the context of the SDAA (Software Defined Autonomous Antenna), described by the authors in [11].

Benefits of adaptive antenna spacing and angle diversity are discussed in [16]. Passive antenna diversity alone is not enough to ensure significant capacity improvements in MIMO systems. However, spacing and angle reconfiguration of the antenna array is enough to significantly increase the system data rate [8]. Classical digital beamforming and beamsteering algorithms can achieve this but with limited degrees of freedom and introducing noise. Activating the appropriate two array elements results in maximization of *Rx1* and *Rx2* signals.

Activating certain elements in the array is equivalent to complex coefficient multiplication (complex weighting) in baseband processing [2]. This spatial method is meant to maximize the SNR by spatial decorrelation of the incoming fronts.

From the adaptive mechanism point of view, the *baseband processing component* includes two important software components: the PER (Packet Error Rate) calculation component and the *Rxx* correlation matrix estimation component.

The *PER component* counts the dropped packets and statistically evaluates the packet error rate after thousands of received data packets. This software component is available in most wireless equipments and its value is readable using a software register of the SDR. The PER value is used as a trigger for the genetic search of array element coordinates *a,b,c* and *d.*

The *Rxx component* provides the correlation matrix at the receiver; the matrix is not readily available and is computed based on pilot signal information. The *Rxx* correlation matrix of the receiver (MxM) offers the mean power delay profile and power angular profile. The *H* channel matrix coefficients and angles of arrival (AoA) can be further extracted.

$$R_{xx} = E\left\{\bar{x}\,\bar{x}^{H}\right\} = AR_{SS}A^{H} + R_{nn} \tag{3}$$

where E is the expectation function and

$\bar{x}^{H}$ is the Hermitian of the received signal matrix $\bar{x}$ (*Mx1*)

M is the number of elements that are used for power and angle profile estimation

A is the array of the steering vectors (*MxD*) (one vector for each angle of arrival, D propagation paths are considered and $D<M$).

$R_{SS}$is the corellation matrix of the source (*DxD*) and

R_{nn} is the correlation matrix of the noise (*MxM*).

Computation of the H matrix is implemented in many wireless standards, such as IEEE 802.11n, 802.16e. The H matrix represents the average energy of the channel between receive antenna j and transmit antenna i. The angles of arrival are estimated based on the available *Rxx* matrix, using the ESPRIT method [1]. The computation time is not critical in this case because Rxx computation is also a statistical problem. Rxx computation is an ergodic process that depends on the signal-to-noise ratio.

GA element-position optimizer. Genetic Algorithms (GAs) entered the electromagnetic domain in the 1990s, and have been successfully used in antenna design to evolve new antenna geometries [12], [13]. Unlike most approaches, where GAs are involved, our proposal takes into account solution search during device operation. The GA search is based on spatial fading characterisation data.

The signal fishing (SF) adaptive mechanism needs to find the maximum of the mean received signal power $P = |h_{11}|^2 + |h_{22}|^2$, after the MRC signal processing. Therefore the SF mechanism problem reduces to the maximization of the objective function $SF : (MxN)^2 -> R^+$, where:

$$SF(a,b,c,d) = |R_{x1}(a,b)|^2 + |R_{x2}(c,d)|^2 \tag{4}$$

$$SF(a,b,c,d) \to \max$$

Solution encoding. Candidate solutions are encoded as x = *(a,b,c,d)*, where *a, b ,c ,d* are expressed as binary strings.
Fitness assignment. The fitness of solution x is:

$$eval(x) = 1/SF(a, b, c, d) \tag{5}$$

Population model. A generational model where the offspring completely replace the parent population at each generation is considered.
Stopping criterion. The search process stops when the maximum number of generations is reached.
Search operators. Recombination is based on uniform crossover operator. A mask is randomly generated for each pair. One bit mutation is considered.
Parameter setting. Population size – we consider the initial population of 100 chromosomes randomly generated for the first generation. Number of generations – between 100 and 300 generations. Mutation rate – 0.01. The number of mutations is 19 for each generation and is computed [13] with:

$$nrmut = mutrate * nbits * nvar * (npop - 1) \tag{6}$$

where: *mutrate* is the mutation rate (0,01),
nbits is the number of bits/chromosome (5),
nvar is the number of genes for each chromosome (4 - a,b,c,d) and
npop is the total population number (100 chromosomes).
nrmut is the number of mutations per generation.

The search process follows the flowchart described in figure 5.

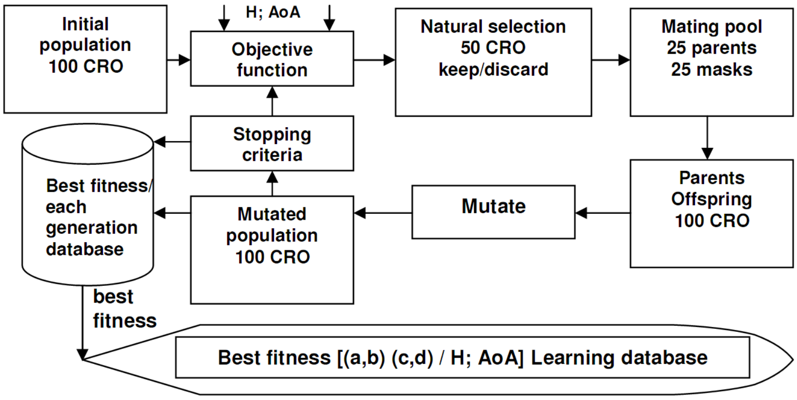

Initial population 100 CRO
H; AoA
Objective function
Natural selection 50 CRO keep/discard
Mating pool 25 parents 25 masks
Stopping criteria
Best fitness/ each generation database
Mutated population 100 CRO
Mutate
Parents Offspring 100 CRO
best fitness
Best fitness [(a,b) (c,d) / H; AoA] Learning database

Figure 5: GA optimizer flowchart

4 Experimental results

Figure 6 a) illustrates the GA search process. The solution pair is circled in red - Rx1 and Rx2. It indicates the best coordinates *(a,b)* for receiving element *Rx1* and *(c,d)* for *Rx2*. The corresponding array elements to be activated are shown in figure 6 b). The evolution of the objective function for each generation is tracked in Figure 7 a), revealing many local maxima. The global maximum is marked in figure 7 a).

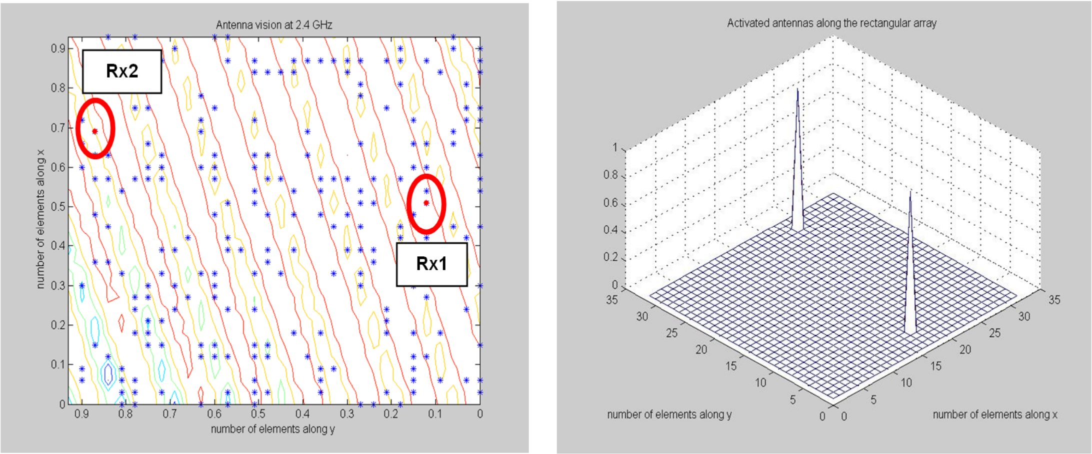
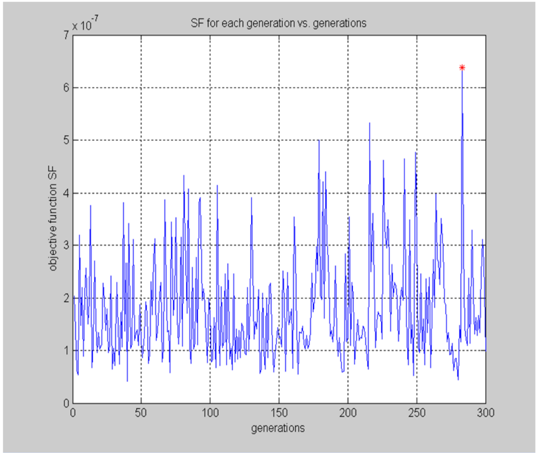
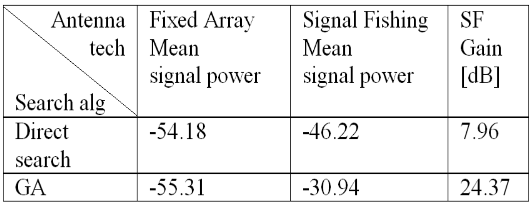


Figure 6: a) GA search process and solution, b) corresponding activated array elements





Antenna tech / Search alg	Fixed Array Mean signal power	Signal Fishing Mean signal power	SF Gain [dB]
Direct search	-54.18	-46.22	7.96
GA	-55.31	-30.94	24.37

Figure 7: a) Objective function over generations, b) Signal-fishing mechanism gain

From the antenna point of view the local maxima are also important because each can result in a signal gain. So whenever a tradeoff between speed and gain is necessary these local maxima can be used. The gain is calculated as the difference between the mean signal power for a fixed two-element array and the mean signal power for the signal fishing mechanism.

5 Conclusions

A long-term adaptation solution for wireless receivers is proposed. It is based on a systemic approach of the wireless receiver chain and gives the antenna array a central role in learning the environment, namely the wireless propagation channel. An implementation of the proposed signal-fishing concept using a GA optimizer for element positioning shows that the maximum signal levels of the channel can be detected and used to increase the mean received signal power. The channel fading effects are mitigated starting at the antenna level, an approach that does not introduce additional noise like current baseband processing techniques.

Acknowledgments

This work was supported by CNCSIS-UEFISCSU, PNII-IDEI, project number 1062/2007.

Bibliography

[1] F. Gross, *Smart antennas for wireless communications*, McGraw Hill, 2005

[2] A.B. Gershman, N.D. Sidiropoulos, *Space-Time Processing for MIMO Communications*, Wiley&Sons, 2005

[3] 3GPP, "SpatialChannel Model for Multiple Input Multiple Output MIMO simulations," Technical Specification Group Radio Access Network TR 25.996 v6.1.0, 3GPP, Sept. 2003.

[4] J. Salo, G. Del Galdo, J. Salmi, et al., "MATLAB implementation of the 3GPP spatial channel model," Tech. Rep. TR 25.996, 3GPP, Jan. 2005.

[5] L. Mucchi, Claudia Staderini, J. Ylitalo, P. Kyosti, "Modified Spatial Channel Model for MIMO Wireless Systems", *EURASIP J. on Wireless Comm. and Netw.*, vol. 2007, pp.1-7.

[6] N. Crisan, Ligia C. Cremene, "Antenna-based Signal Fishing", The Fifth Int. Conf. on Wireless and Mobile Communications – ICWMC'09, pp.152-156, IEEE Computer Society Press, Cannes, 2009.

[7] G. Tsoulos, *MIMO System Technology for Wireless Communications,* CRC Press Cityplace-Boca Raton, StateFlorida, 2006.

[8] N. Crisan, Ligia C. Cremene, "A Novel Combining Technique for Adaptive Antenna Arrays", *ACTA TECHNICA NAPOCENSIS Electronics and Telecommunications*, vol. 49/2, pp.27-34, Mediamira Science Publisher, Cluj-Napoca, 2008.

[9] M-S. Alouini, *Adaptive and Diversity Techniques for Wireless Digital Communications over Fading Channels*, Ph.D. Thesis, California Institute of Technology, 1998.

[10] L. Hanzo, C. H. Wong, M. S. Yee, *Adaptive Wireless Transceivers*, John Wiley & Sons, NY, 2002.

[11] Ligia C. Cremene, N. Crisan, "Towards Cognitive Antenna Systems based on Antenna-Channel Co-evolution" 3rd Int. Workshop on Soft Computing Applications - SOFA2009, pp.165-169, Szeged, 2009.

[12] D. Dumitrescu, B. Lazzerini, L. C. Jain, A. Dumitrescu, *Evolutionary computation*, CRC Press, NY, 2000.

[13] R. L. Haupt, Sue Ellen Haupt, *Practical genetic algorithms*, John Wiley & Sons, Inc., 2004.

[14] S. D. Blostein, H. Leib, "Multiple Antenna Systems: Their Role and Impact în Future Wireless Access", *IEEE Communications Magazine*, vol.41, no.7, pp.94-101, July 2003.

[15] Angeliki Alexiou, M. Haardt, "Smart Antenna Technologies for Future Wireless Systems: Trends and Challenges", *IEEE Communications Magazine*, vol.42, no.9, pp. 90-97, 2004.

[16] J. D. Boerman, J. T. Bernhard, "Performance Study of Pattern Reconfigurable Antennas in MIMO Communication Systems", *IEEE Trans. on Ant. and Propag.*, Vol.56, No.1, Jan.2008.

Int. J. of Computers, Communications & Control, ISSN 1841-9836, E-ISSN 1841-9844
Vol. VI (2011), No. 1 (March), pp. 72-80

Evaluation Measures for Partitioning based Aspect Mining Techniques

G. Czibula, G. S. Cojocar, I. G. Czibula

Gabriela Czibula, Grigoreta Sofia Cojocar, Istvan Gergely Czibula
Babeş-Bolyai University
1, M. Kogălniceanu Street, 400084, Cluj-Napoca, Romania
E-mail: {gabis, grigo, istvanc}@cs.ubbcluj.ro

Abstract: *Aspect mining* is a research direction that tries to identify crosscutting concerns in already developed software systems, without using aspect oriented programming. The goal is to identify them and then to refactor them to aspects, to achieve a system that can be easily understood, maintained and modified. In this paper we propose two new evaluation measures for evaluating the results of partitioning based aspect mining techniques. A small example on how to compute them is provided. The applicability of these measures to different aspect mining techniques is also discussed.

Keywords: partitioning, aspect mining, crosscutting concern, evaluation.

1 Introduction

Nowadays, software systems have become more and more complex and large. A software system is usually composed of many core concerns and (some) crosscutting concerns (like logging, exception handling). If core concerns can be cleanly separated and implemented using existing programming paradigms, this is not true for crosscutting concerns, as a crosscutting concern has a more system-wide behaviour that cuts across many of the core concerns implementation modules. The aspect oriented paradigm is one of the approaches proposed, so far, for the design and implementation of crosscutting concerns. Aspect oriented techniques allow crosscutting concerns to be implemented in a new kind of module called *aspect*, by introducing new language constructs like pointcuts and advices [13].

Kiczales et al. introduce for the first time aspect oriented programming (AOP) in [11]. Since 1997 the aspect oriented paradigm has been slowly adopted by the industry, too, leading to the appearance of new research problems like software reverse engineering, reengineering, and refactoring to use the aspect-oriented paradigm in order to benefit from the advantages it brings.

Aspect mining is a research direction that tries to identify crosscutting concerns in already developed software systems, without using AOP. The goal is to identify them and then to refactor them to aspects, to achieve a system that can be easily understood, maintained and modified. The task of crosscutting concerns identification cannot be successfully done using only a manual approach, as it is a difficult and error-prone process due to the complexity of software systems, their size, the lack of documentation, etc. As a consequence researchers have focused on developing tools and techniques that help developers to identify the crosscutting concerns in already developed software systems. The tools and techniques proposed, so far, try to discover the symptoms that an inadequate solution for a crosscutting concern implementation has over a software system: duplicated code, *scattering* of concerns throughout the entire system and *tangling* of concern-specific code with that of other concerns.

Although aspect mining is a relatively new research domain, many aspect mining techniques have been proposed. Some use metrics [14], some use formal concept analysis [2, 24, 25], or execution relations [1]. There are also a few approaches that use clone detection techniques

[3,22] or natural language processing [18]. A few techniques use clustering in order to identify crosscutting concerns [8,15,20,23].

There are very few comparisons made between the aspect mining techniques proposed so far [4,10,16,17], and even less comparisons based on the obtained results [4,17]. One important cause is the lack of measures for evaluating the results obtained and the quality of the results (i.e. how well did the technique manage in separating crosscutting concerns from non-crosscutting concerns, and in separating one crosscutting concern from other crosscutting concerns).

The main contribution of this paper is to propose two new evaluation measures for comparing partitioning based aspect mining techniques.

The paper is structured as follows. The context in which the measures are defined is introduced in Section 2. The new evaluation measures are defined in Section 3. A small example on how two compute the measure is given in Section 4. The applicability of the newly introduced measures is studied in Section 5. Some conclusions and further work are given in Section 6.

2 Formal Model

In [5] Cojocar and Şerban have proposed a formal model for partitioning based aspect mining. The measures that we proposed in this paper are based on their model. In the following we briefly introduce this model.

A software system S is viewed as a set of elements from the system: $S = \{s_1, s_2, \ldots, s_n\}$, where $s_i, 1 \leq i \leq n$. An *element* can be a statement, a method, a class, a module, etc. The number of elements of the system is denoted by n ($n = |S|$).

A crosscutting concern is considered as a set of elements $C \subset S$, $C = \{c_1, c_2, ..., c_m\}$, elements that implement this concern. CCC denotes the set of all crosscutting concerns that exist in the system S, $CCC = \{C_1, C_2, ..., C_q\}$, and q denotes the number of crosscutting concerns in the system S, $q = |CCC|$. It is considered that two different crosscutting concerns do not have elements in common, meaning that $C_i \cap C_j = \emptyset, \forall i, j, 1 \leq i, j \leq q, i \neq j$.

The problem of aspect mining is viewed as the problem of identifying a partition $\mathcal{K}$ of the software system S, such that $CCC \subset \mathcal{K}$.

Definition 1. Optimal partition of a system S.
Being given a partition $\mathcal{K} = \{K_1, K_2, ..., K_p\}$ of the system S, $\mathcal{K}$ is called an **optimal partition** of the system S with respect to the set $CCC = \{C_1, C_2, ..., C_q\}$ of all crosscutting concerns, iff:
(1) $p \geq q$;
(2) $\forall C \in CCC,\ \exists K_C \in \mathcal{K}$ such that $C = K_C$.

From the aspect mining point of view, $\mathcal{K}$ is an optimal partition of the system S if and only if the components of each crosscutting concern $C \in CCC$ are in the cluster K_C and K_C contains only the elements of C.

3 Evaluation measures

In this subsection we propose two measures for evaluating a partition of a software system from the aspect mining point of view. Such a partition can be obtained using a partitioning algorithm, such as a clustering algorithm.

In the following, let us consider a partition $\mathcal{K} = \{K_1, \ldots, K_p\}$ of a software system S and $CCC = \{C_1, C_2, ..., C_q\}$ the set of all crosscutting concerns from S. We assume that each crosscutting concern consists of a set of elements, i.e., $C_i = \{c_{i_1}, c_{i_2}, \ldots, c_{i_{m_i}}\}$.

Definitions 2 and 6 introduce evaluation measures for a partition of a software system from the aspect mining point of view.

Definition 2. Cohesion of Recovered Crosscutting Concerns - CORE.

Let $\mathcal{K}$ be a partition of a software system identified by a partitioning based aspect mining technique.

The cohesion of crosscutting concerns CCC recovered in partition $\mathcal{K}$, denoted by $CORE(CCC, \mathcal{K})$, is defined as: $CORE(CCC, \mathcal{K}) = \frac{1}{q}\sum\limits_{i=1}^{q} core(C_i, \mathcal{K})$. $core(C_i, \mathcal{K})$ is the cohesion of crosscutting concern C_i in partition $\mathcal{K}$ and is defined as: $core(C_i, \mathcal{K}) = \frac{\sum\limits_{k \in M_{C_i}} \dfrac{|C_i \cap k|}{|C_i \cup k|}}{|M_{C_i}|}$, where M_{C_i} is defined as: $M_{C_i} = \{k \mid k \in \mathcal{K},\ C_i \cap k \neq \emptyset\}$.

For a given crosscutting concern $C \in CCC$, $core(C, \mathcal{K})$ defines the degree to which the components of C belong together.

Lemma 3. *If $\mathcal{K}$ is a partition of the software system S and CCC is the set of crosscutting concerns in S, then the following inequality holds:* $0 \leq CORE(CCC, \mathcal{K}) \leq 1$.

For lack of space, we will not give the proof of Lemma 3.

Remark 4. Larger values for $CORE$ indicate better partitions with respect to CCC, meaning that $CORE$ has to be maximized.

In the following we give a necessary and sufficient condition for a partition $\mathcal{K}$ to be an **optimal partition**, with respect to the set of crosscutting concerns from the software system S.

Lemma 5. *If $\mathcal{K} = \{K_1, K_2, \ldots, K_p\}$ is a partition of the software system S, and CCC is the set of crosscutting concerns in S, then $\mathcal{K}$ is an **optimal partition** iff* $CORE(CCC, \mathcal{K}) = 1$.

For lack of space, we will not give the proof of Lemma 5.

The next measure determines the percentage of elements that must be analyzed in order to discover all the crosscutting concerns from the system. Usually, partitioning based aspect mining techniques return the clusters to be analyzed in a specific order.

Let σ be a permutation of the set $\{1,\ 2, \ldots\ p\}$. σ denotes the order in which the clusters from a partition of the software system are analyzed: $K_{\sigma(1)}$ is the first analyzed cluster, $K_{\sigma(2)}$ is the second, etc. The permutation σ is particular to each partitioning based aspect mining technique.

Definition 6. Complexity of Crosscutting Concerns Discovery - CODI. Let σ be a permutation of the set $\{1,\ 2, \ldots\ p\}$. The complexity of crosscutting concerns CCC discovery in partition K, denoted by $CODI(CCC, \mathcal{K}, \sigma)$, is defined as: $CODI(CCC, \mathcal{K}, \sigma) = \frac{1}{q}\sum\limits_{i=1}^{q} Codi(C_i, \mathcal{K}, \sigma)$. $Codi(C_i, \mathcal{K}, \sigma)$ is the percentage of the elements that need to be analyzed in the partition $\mathcal{K}$ in order to discover the crosscutting concern C_i, and it is defined as: $Codi(C_i, \mathcal{K}, \sigma) = \frac{1}{m_i}\sum\limits_{j=1}^{m_i} codi(c_{ij}, \mathcal{K}, \sigma)$. $codi(c_{ij}, \mathcal{K}, \sigma)$ is the percentage of the elements that need to be analyzed in the partition $\mathcal{K}$ in order to discover the element c_{ij} of crosscutting concern C_i, and it is defined as: $codi(c_{ij}, \mathcal{K}, \sigma) = \frac{1}{n}\sum\limits_{l=1}^{r} |K_{\sigma(l)}|.$, where $r \in \{1,\ ,2, \ldots, p\}$ and it has the property that $c_{ij} \cap K_{\sigma(r)} \neq \emptyset$.

In our view $CODI(CCC, \mathcal{K}, \sigma)$ gives the complexity of crosscutting concerns discovery, and defines the percentage of the number of elements that need to be analyzed in the partition in order to discover all the crosscutting concerns that are in the system S. We consider that a crosscutting concern was discovered when all the elements that implement it were analyzed.

Lemma 7. *If $\mathcal{K}$ is a partition of the software system S, σ is a permutation of $\mathcal{K}$ and CCC is the set of crosscutting concerns in S, then the following inequality holds:* $0 < CODI(CCC, \mathcal{K}, \sigma) \leq 1$.

For lack of space, we will not give the proof of Lemma 7.

Remark 8. Smaller values for $CODI$ indicate shorter time for analysis, meaning that $CODI$ has to be minimized.

Based on the evaluation measures defined above, the comparison of the results obtained by different aspect mining techniques can be made from two different criteria:

1. **Partitioning**. The degree to which each crosscutting concern is well placed in the partition (using measure $CORE$).

2. **Ordering**. How relevant is the order in which the clusters are analyzed (using measure $CODI$).

In order to compare two partitions obtained by partitioning based aspect mining techniques, we introduce Definition 9. The definition is based on the properties of the evaluation measures defined above and considers both criteria presented above.

Definition 9. If $\mathcal{K}_1$ and $\mathcal{K}_2$ are two partitions of the software system S, CCC is the set of crosscutting concerns in S, and σ_1 and σ_2 are permutations of $\mathcal{K}_1$ and $\mathcal{K}_2$ respectively, then $\mathcal{K}_1$ is **better** than $\mathcal{K}_2$ iff the following inequalities hold: $CORE(CCC, \mathcal{K}_1) \geq CORE(CCC, \mathcal{K}_2)$, $CODI(CCC, \mathcal{K}_1, \sigma_1) \leq CODI(CCC, \mathcal{K}_2, \sigma_2)$.

For the above definition we can remark the following:

Remark 10. If at least one of the inequalities from Definition 9 is not satisfied, we cannot decide which of the partitions $\mathcal{K}_1$ or $\mathcal{K}_2$ is better from the aspect mining point of view (considering simultaneously both criteria).

Remark 11. However, the importance of the above mentioned comparison criteria may depend on the user of the aspect mining technique. In our view, the most important criterion is **Partitioning** (how well the crosscutting concerns are grouped) and the last one is **Ordering** (how quickly the crosscutting concerns are discovered).

4 Example

In the following, a small example showing how to compute $CORE$ and $CODI$ measures is presented.

Let $S = \{s_1, s_2, ..., s_{17}\}$ be a software system with 17 elements, and let $C_1 = \{s_2, s_3, s_{17}\}$ and $C_2 = \{s_1, s_4, s_8\}$ be the crosscutting concerns that exist in the system S ($CCC = \{C_1, C_2\}$).

Let $\mathcal{K} = \{K_1, K_2, K_3, K_4, K_5\}$ be a partition of the software system S, where: $K_1 = \{s_2, s_{16}\}$; $K_2 = \{s_3, s_7, s_8, s_9, s_{17}\}$; $K_3 = \{s_1, s_4, s_5, s_{12}\}$; $K_4 = \{s_6, s_{10}, s_{13}\}$; $K_5 = \{s_{11}, s_{14}, s_{15}\}$.

CORE

Using Definition 2, we have to compute $core(C, \mathcal{K})$ for each $C \in CCC$.

C_1 First we have to determine the set M_{C_1}. It consists of two elements: $M_{C_1} = \{K_1, K_2\}$. The value of $core(C_1, \mathcal{K}) = \frac{1}{2} \sum\limits_{k \in M_{C_1}} \frac{|C_1 \cap k|}{|C_1 \cup k|}$. We obtain that $core(C_1, \mathcal{K}) = \frac{1}{2}(\frac{|C_1 \cap K_1|}{|C_1 \cup K_1|} + \frac{|C_1 \cap K_2|}{|C_1 \cup K_2|}) = \frac{1}{2}(\frac{1}{4} + \frac{2}{6}) = \frac{7}{24}$.

C_2 The set M_{C_2} also consists of two elements: $M_{C_2} = \{K_2,\ K_3\}$. Based on Definition 2 $core(C_2, \mathcal{K}) = \frac{1}{2} \sum\limits_{k \in M_{C_2}} \frac{|C_2 \cap k|}{|C_2 \cup k|}$. We obtain that $core(C_2, \mathcal{K}) = \frac{1}{2}(\frac{|C_2 \cap K_2|}{|C_2 \cup K_2|} + \frac{|C_2 \cap K_3|}{|C_2 \cup K_3|}) = \frac{1}{2}(\frac{1}{7} + \frac{2}{5}) = \frac{19}{70}$

Based on the Definition 2, $CORE(CCC, \mathcal{K}) = \frac{1}{2}(core(C_1, \mathcal{K}) + core(C_2, \mathcal{K})) = \frac{1}{2}(\frac{7}{24} + \frac{19}{70}) = \frac{473}{1680}$.

CODI

We consider the permutation σ in which the partition $\mathcal{K}$ is analyzed to be the identity permutation.

Using Definition 6, we have to compute $Codi(C, \mathcal{K}, \sigma)$ for each $C \in CCC$.

C_1 Based on the definition, $Codi(C_1, \mathcal{K}, \sigma) = \frac{1}{3}[codi(s_2, \mathcal{K}, \sigma) + codi(s_3, \mathcal{K}, \sigma) + codi(s_{17}, \mathcal{K}, \sigma)] = \frac{1}{3}[\frac{|K_1|}{17} + \frac{|K_1| + |K_2|}{17} + \frac{|K_1| + |K_2|}{17}] = \frac{1}{3}(\frac{2}{17} + \frac{2+5}{17} + \frac{2+5}{17}) = \frac{16}{51}$

C_2 Based on the definition, $Codi(C_2, \mathcal{K}, \sigma) = \frac{1}{3}[codi(s_1, \mathcal{K}, \sigma) + codi(s_4, \mathcal{K}, \sigma) + codi(s_8, \mathcal{K}, \sigma)] = \frac{1}{3}[\frac{|K_1| + |K_2| + |K_3|}{17} + \frac{|K_1| + |K_2| + |K_3|}{17} + \frac{|K_1| + |K_2|}{17}] = \frac{1}{3}(\frac{2+5+4}{17} + \frac{2+5+4}{17} + \frac{2+5}{17}) = \frac{29}{51}$

Based on the definition, $CODI(CCC, \mathcal{K}, \sigma) = \frac{1}{2}[Codi(C_1, \mathcal{K}, \sigma) + Codi(C_2, \mathcal{K}, \sigma)] = \frac{1}{2}(\frac{16}{51} + \frac{29}{51}) = \frac{45}{102}$.

For this example, an **optimal partition** is: $K_1 = \{s_6, s_9, s_{10}\}$; $K_2 = \{s_2, s_3, s_{17}\}$; $K_3 = \{s_{11}, s_{12}, s_{14}, s_{15}\}$; $K_4 = \{s_5, s_7, s_{13}, s_{16}\}$; $K_5 = \{s_1, s_4, s_8\}$.

5 Applicability

In this section we present some of the existing approaches in aspect mining. For each approach we analyze the applicability of the evaluation measures proposed in this paper.

Marin et al [14] have proposed an aspect mining technique that uses the *fanin* metric [9]. Their idea is to search for crosscutting concerns among the methods that have the value of the fanin metric greater than a given threshold. The result obtained by this technique can be viewed as a *partition* of the software system to be mined. The partition contains two clusters: the first one contains the methods that have the fanin greater than the given threshold and the second contains the remaining methods. So, the evaluation measures $CORE$ and $CODI$ can be applied for the result of this technique.

A graph based approach in Aspect Mining is introduced in [19]. The basic idea of this technique is to determine methods that are similar. The approach is to construct a graph between the methods of the software system, to determine the connex components of this graph, called *clusters*, and then to identify crosscutting concerns in the obtained clusters. As the set of connex components determined by this technique represents a *partition* of the analyzed software system, the evaluation measures $CORE$ and $CODI$ can also be applied for the result of this technique.

There are a few aspect mining techniques proposed in the literature that use *clustering* in order to identify crosscutting concerns [8, 20, 23].

He and Bai [8] have proposed an aspect mining technique based on dynamic analysis. They obtain execution traces for each use case, but they apply clustering and association rules to discover aspect candidates.

Shepherd and Pollock [23] have proposed an aspect mining tool based on clustering. They use hierarchical clustering to find methods that have common substrings in their names. The obtained clusters are then manually analyzed to discover crosscutting concerns.

A clustering approach for identifying crosscutting concerns is proposed and a partitional clustering algorithm named kAM is introduced in [20].

An *evolutionary* approach in aspect mining is introduced in [21] and two *genetic clustering* algorithms used to identify crosscutting concerns are proposed. The clustering approach proposed in [21] is based on the use of genetic algorithms [6].

As all the above presented clustering techniques provide a *partition* of the software system, the applicability of $CORE$ and $CODI$ evaluation measures is assured.

There are in the literature some aspect mining techniques, briefly presented in the following, that do not provide a partition of the entire analyzed software system, but a subset of it. $CORE$ and $CODI$ measures cannot be applied for these techniques, as they require a partition of the entire system. However, the proposed measures can be extended in order to also consider these kind of situations. In the future we plan to tackle these particular cases.

Breu and Krinke [1] have proposed an aspect mining technique based on dynamic analysis. The mined software system is run and program traces are generated. From program traces, recurring execution relations that satisfy some constraints are selected. Among these recurring execution relations they search for aspect candidates. This approach is adapted to static analysis in [12]. In this approach the recurring execution relations are obtained from the control flow graph of the program.

Tonella and Ceccato [24] have also proposed an aspect mining technique based on dynamic analysis. An instrumented version of the mined software system is run and execution traces for each use case are obtained. Formal concept analysis [7] is applied on these execution traces and the concepts that satisfy some constraints are considered as aspect candidates.

Tourwé and Mens [25] have proposed an aspect mining technique based on identifier analysis. The identifiers associated with a method or class are computed by splitting up its name based on where capitals appear in it. They apply formal concept analysis on the identifiers to group entities with the same identifiers. The groups that satisfy some constraints and that contain a number of elements larger than a given threshold are considered as aspect candidates.

Bruntink et al [3] have studied the effectiveness of clone detection techniques in aspect mining. They did not propose a new aspect mining technique, but they tried to evaluate how useful clone detection techniques are in aspect mining.

Shepherd et al [22] have proposed an aspect mining technique based on clone detection. They search for code duplication in the source code using the program dependency graph. The obtained results are further analyzed to discover crosscutting concerns.

Breu and Zimmermann [2] have proposed an history based aspect mining technique. They mine CVS repositories for add-call transactions on which they apply formal concept analysis. Concepts that satisfy some constraints are considered aspect candidates.

Sampaio et al [18] have proposed an aspect mining technique to discover aspect candidates early in the development lifecycle. They use natural language processing techniques on different documents (requirements, interviews, etc.) to discover words that are used in many sentences.

The words that have a high frequency and have the same meaning in all the sentences are considered aspect candidates.

6 Conclusions and further work

In this paper we have proposed two new measures (*CORE* and *CODI*) for evaluating the results of partitioning based aspect mining techniques. We have also proved three important lemmas related to the proposed evaluation measures. A small example on how to compute these measures was provided. We have also discussed the applicability of these measures to different aspect mining techniques.

Further work may be done in the following directions: to adapt the measures to consider the case in which the crosscutting concerns have overlapping elements (the same element belongs to different crosscutting concerns); to evaluate some of the existing aspect mining techniques using the proposed measures; to extend the proposed evaluation measures in order to consider the situation in which the result obtained by an aspect mining technique is not a partition of the entire software system.

Acknowledgements

This work was supported by CNCSIS -UEFISCSU, project number PNII - IDEI 2286/2008.

Bibliography

[1] S. Breu and J. Krinke. Aspect Mining Using Event Traces. In *Proceedings of International Conference on Automated Software Engineering (ASE)*, pages 310–315, 2004.

[2] S. Breu and T. Zimmermann. Mining Aspects from Version History. In S. Uchitel and S. Easterbrook, editors, *21st IEEE/ACM International Conference on Automated Software Engineering (ASE 2006).* ACM Press, September 2006.

[3] M. Bruntink, A. van Deursen, R. van Engelen, and T. Tourwé. On the use of clone detection for identifying crosscutting concern code. *IEEE Transactions on Software Engineering*, 31(10):804–818, 2005.

[4] M. Ceccato, M. Marin, K. Mens, L. Moonen, P. Tonella, and T. Tourwé. A Qualitative Comparison of Three Aspect Mining Techniques. In *IWPC '05: Proceedings of the 13th International Workshop on Program Comprehension*, pages 13–22. IEEE Computer Society, 2005.

[5] G. S. Cojocar(Moldovan) and G. Serban. A Formal Model for Partitioning based Aspect Mining. *INFOCOMP Journal of Computer Science, Brazil*, 6(3):19–26, 2007.

[6] C. Cubillos, E. Urra and N. Rodríguez. Application of Genetic Algorithms for the DARPTW Problem. *International Journal of Computers, Communication and Control*, Vol. IV, No. 2:127-136, 2009.

[7] B. Ganter and R. Wille. *Formal Concept Analysis.* Springer-Verlag, Berlin, Heidelberg, New York, 1996.

[8] L. He and H. Bai. Aspect Mining using Clustering and Association Rule Method. *International Journal of Computer Science and Network Security*, 6(2):247–251, February 2006.

[9] B. Henderson-Sellers. *Object-Oriented Metrics: Measures of Complexity.* Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1996.

[10] A. Kellens, K. Mens, and P. Tonella. A Survey of Automated Code-level Aspect Mining Techniques. *Transactions on Aspect-Oriented Software Development, Special Issue on Software Evolution*, VI(LNCS 4640):145–164, 2007.

[11] G. Kiczales, J. Lamping, A. Menhdhekar, C. Maeda, C. Lopes, J.-M. Loingtier, and J. Irwin. Aspect-Oriented Programming. In *Proceedings European Conference on Object-Oriented Programming*, volume LNCS 1241, pages 220–242. Springer-Verlag, 1997.

[12] J. Krinke. Mining control flow graphs for crosscutting concerns. In *13th Working Conference on Reverse Engineering: IEEE International Astrenet Aspect Analysis (AAA) Workshop*, pages 334–342, 2006.

[13] R. Laddad. *AspectJ in Action: Practical Aspect-Oriented Programming.* Manning Publications Co., 2003.

[14] M. Marin, A. van, Deursen, and L. Moonen. Identifying Aspects Using Fan-in Analysis. In *Proceedings of the 11th Working Conference on Reverse Engineering (WCRE2004).*, pages 132–141. IEEE Computer Society, 2004.

[15] G. S. Moldovan and G. Serban. Aspect Mining using a Vector-Space Model Based Clustering Approach. In *Proceedings of Linking Aspect Technology and Evolution (LATE) Workshop*, pages 36–40, Bonn, Germany, March, 20 2006. AOSD'06.

[16] B. Nora, G. Said, and A. Fadila. A Comparative Classification of Aspect Mining Approaches. *Journal of Computer Science*, 2(4):322–325, 2006.

[17] C. K. Roy, M. G. Uddin, B. Roy, and T. R. Dean. Evaluating Aspect Mining Techniques: A Case Study. In *ICPC '07: Proceedings of the 15th IEEE International Conference on Program Comprehension*, pages 167–176, Washington, DC, USA, 2007. IEEE Computer Society.

[18] A. Sampaio, N. Loughran, A. Rashid, and P. Rayson. Mining Aspects in Requirements. In *Early Aspects 2005: Aspect-Oriented Requirements Engineering and Architecture Design Workshop (held with AOSD 2005)*, Chicago, Illinois, USA, 2005.

[19] G. Serban and G. S. Moldovan. A Graph Algorithm for Identification of Crosscutting Concerns. *Studia Universitatis Babes-Bolyai, Informatica*, LI(2):53–60, 2006.

[20] G. Serban and G. S. Moldovan. A New k-means Based Clustering Algorithm in Aspect Mining. In *Proceedings of 8th International Symposium on Symbolic and Numeric Algorithms for Scientific Computing (SYNASC'06)*, pages 69–74, Timisoara, Romania, September, 26-29 2006. IEEE Computer Society.

[21] G. Serban and G. S. Moldovan. Aspect Mining using an Evolutionary Approach. *WSEAS Transactions on Computers*, 6(2):298–305, 2007.

[22] D. Shepherd, E. Gibson, and L. Pollock. Design and Evaluation of an Automated Aspect Mining Tool. In *2004 International Conference on Software Engineering and Practice*, pages 601–607. IEEE, June 2004.

[23] D. Shepherd and L. Pollock. Interfaces, Aspects, and Views. In *Proceedings of Linking Aspect Technology and Evolution Workshop(LATE 2005)*, March 2005.

[24] P. Tonella and M. Ceccato. Aspect Mining through the Formal Concept Analysis of Execution Traces. In *Proceedings of the IEEE Eleventh Working Conference on Reverse Engineering (WCRE 2004)*, pages 112–121, November 2004.

[25] T. Tourwé and K. Mens. Mining Aspectual Views using Formal Concept Analysis. In *SCAM '04: Proceedings of the Source Code Analysis and Manipulation, Fourth IEEE International Workshop on (SCAM'04)*, pages 97–106, Washington, DC, USA, 2004. IEEE Computer Society.

Int. J. of Computers, Communications & Control, ISSN 1841-9836, E-ISSN 1841-9844
Vol. VI (2011), No. 1 (March), pp. 81-89

Architecting Robotics and Automation Societies over Reusable Software Frameworks: the Case of the G++ Agent Platform

F. Guidi-Polanco, C. Cubillos

Franco Guidi-Polanco, Claudio Cubillos
Pontificia Universidad Católica de Valparaíso
Av. Brasil 2241, Valparaíso, Chile.
E-mail: {franco.guidi,claudio.cubillos}@ucv.cl

Abstract: This work presents a software framework that allows the implementation of societies made-up by autonomous collaborative devices. The framework is structured as a multi-layer reusable architecture, adopting the software agent as the design paradigm. In this paper, an overview of its main features is offered, and an example of its adoption in the design of a robot colony is provided.

Keywords: Global Automation, Robots colony, Agent Architecture, Agent-Based Framework.

1 Introduction

In de last decades, robotics and automation systems have evolved both in "body" and in "soul": physiological functions of robots (like sensing, walking, etc.) have reached a degree of maturity, and control systems allow programming robots as entities that can perform autonomous and proactive behaviours. Today, the research is focused in architecting models of societies made-up by autonomous mobile cooperating devices, performing common goal-seeking tasks. Building software for such kind of systems, however, encompasses several challenges, such as how to increase efficiency in the development of control modules (avoid building always from scratch or unnecessary replicating code), how to the deploy software updates in geographically distributed devices, or how to provide data interaction between mobile robots over unreliable communication channels, among other common development problems.

In our vision, the agent paradigm is envisioned as the software abstraction for development of societies of distributed autonomous collaborative systems. Under such paradigm we have designed an agent-based framework providing a reusable architecture that addresses common issues in robotics and automation systems.

This paper offers a description of the state of the art in software architectures for robotics and automation systems, and describes our agent-based framework architecture (the G++ Agent Platform).

2 Related Work

The multi-agent system (MAS) paradigm is being adopted to implement control and communication in distributed automation and robotic societies. In such systems, the modelling paradigm is centred in the concept of agent. An agent is a software entity capable to perceive its environment, to evaluate these perceptions against some given design objectives, and to perform some activity in order to reach them, interacting with other similar entities, and acting over its environment. Agents should be designed to exhibit robust operation, even if they are immersed in an open or unpredictably changing environment [18].

In recent years the literature offers several examples of multi-agent architectures and organizations created for domain-specific applications (see [8], [11], [15] for some examples). These architectures accent the identification of agent's roles and responsibilities, and the description of their interactions and communications. As expected, due to their ad-hoc nature, these architectures are hardly reusable outside their original domains.

In order to improve the reuse of design, some studies establish the convenience of identifying and separating domain-specific aspects from those generic aspects that are common in families of systems. One example is the orientation followed in [16] that proposes the reuse of organizational coordination mechanisms across different problem domains and environmental situations. Nevertheless, their work just emphasizes organization and distribution of tasks and goals, while the system's structure is not deeply treated. An important contribution, in accordance with the latter approach, is the *holonic* paradigm [17]. This approach, offers an organizational model highly reusable, which can be applied at diverse abstraction levels and replicable in different domains [10]. However, it is a conceptual model that does not specify implementation of concrete services that can be required and reused when developing such systems. On the other hand, models of agent societies and agent platforms implementations play insufficient attention to the agent's environment, which is an essential part in robotic system's structure. In practice agent architectures fail to adequately identify and consider its role. As indicated in [19], popular frameworks minimize the environment reducing it just to a message transport system or to a brokering infrastructure. In terms of structure and services, the development of generic agent platforms (e.g. Jade [2]) presents concrete architectures with high degree of reusability, but made-up by low-granularity components (commonly, basic communication and directory services), that implement commonly agreed abstract models (e.g. FIPA). Also, these platforms are not designed to satisfy security, connectivity, and scalability requirements originated in the robotic and automation domain [7]. The adoption of agent systems as enabling technologies for the development of distributed organizations' infrastructures is currently matter of research. In particular, the agent technology seems not only to satisfy the demand for high flexibility requested by enterprise-wide integration, but also to provide approaches to support autonomous self-configuration and self-adaptability of their activities in their operational environment.

3 An abstract model for agent-based robotics societies

We envision the *agent paradigm* as the software engineering approach to model entities (control modules) in robotic architectures. The arguments in favour of an agent-oriented approach in software engineering for modelling a system can be summarized in the three ideas indicated in [9]: (1) Agent oriented decompositions are an effective way of partitioning the problem space of a complex system; (2) The key abstractions of the agent-oriented mindset are natural means of modelling complex systems; and (3) The agent-oriented philosophy for modelling and managing organizational relationships is appropriate for dealing with the dependencies and interactions that exist in complex systems. Our approach introduces a layered model that identifies and classifies system's components (i.e. agents and services) accordingly with different granularities. Those components and services that share similar levels of reuse from both, the structural and the organizational point of view, are grouped together. The model is build recognizing at its basis the physical environment, which is virtualized in superior levels, making explicit the way in which agents will interact with it.

This vision is constructed as the abstract model depicted in Figure 1. The abstract model is divided by the following five layers:

a) *Environment*: it is composed by physical objects pertaining or observed in the real world (e.g. objects in mobile robot's environment, wired or wireless communication networks, computa-

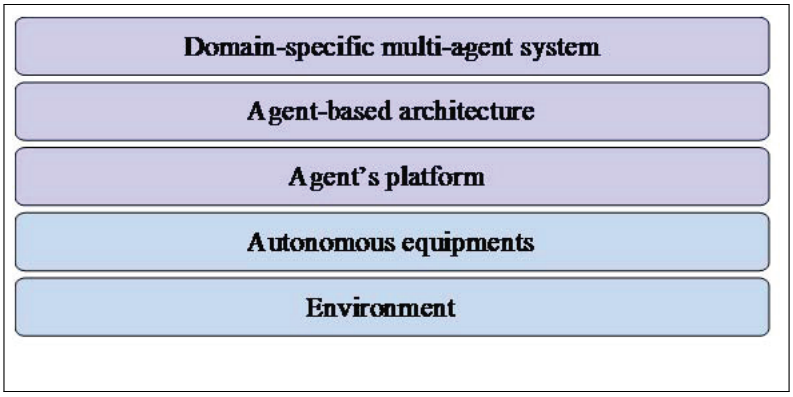


Figure 1: The layers in a robotic society

tional systems in organizations, human operators, etc.), and concepts conventionally adopted for its characterization (e.g. geographical coordinates obtained from a GPS service, temperatures, data transmission latency, water flows measurements, etc.). The physical world is conceptualized as a multidimensional space surrounding agents accomplishing physical-related tasks.

b) *Autonomous Equipments*: represent computing-enabled platforms, such as mobile robots, automated factory machines, or computing devices, which has to be programmed in order to act proactively in the robotic community. Such systems, can offer a wide range of capabilities expressed in terms of CPU, runtime memory, data storage, data communication, or operating system. These equipments are usually provided with sensors that allow the perception of surrounding relevant variables, actuators to interact with the environment (changing their own position, taking objects, etc.), and communications devices to interchange messages with other equipments. Also, the autonomous equipments provide the runtime environment for the agents, so they must satisfy a set of minimum hardware/software requirements imposed by the agent platform's software (or in an opposite point of view, the agent platforms must be designed to be executed in specific categories of devices).

c) *Agent platform*: corresponds to the software that offers the base classes to build agents, and to virtualize environment-dependent services (e.g. interfaces to peripheral devices, motors, databases, network communication, etc.). It also offers the execution environment that controls the entire agent's life-cycle, and regulates its interactions with other agents and resources. A known example of agent platform is JADE [2]. A comprehensive list of agent platforms can be found in [1].

d) *Agent-based architecture*: represents a reusable architecture to support the development of different kinds of agent-based systems. The architecture specifies a set of common services (e.g. directory facilitator, yellow pages, etc.), and a framework of communication/content languages (e.g. ACL [6], KQML [4], etc.) and interaction protocols, necessary to achieve interoperability among agents. An example of a particular agent-based architecture is specified by FIPA standards, which was conceived to obtain interoperability between different and generic agent systems (e.g. FIPA Request Interaction Protocol [5]).

e) *Domain-specific multi agent system* (DSMAS): corresponds to a concrete instance of a multi-agent system, where domain-dependent agents are designed to represent real-world services and systems, and interactions among them are well defined. At this level, agents are often abstractions of real entities pertaining to the application domain. DSMAS architectures can be reused within the scope of the context they were created for. Examples of DSMAS could be a colony of exploration robots, an automated work cell, or a domotic network of devices.

4 The G++ Agent Platform

The G++ Agent Platform runs over a Java Virtual Machine (JVM) hosted in a computational device. The execution environment of the G++ agent platform provides connectivity services, being responsible for the interactions among all agents. It is also responsible for the virtualization of the physical environment, through the implementation of sensors and actuators interfaces that agents can access. The platform provides a Container, which is the environment for the execution of agents. A container runs over a Java Runtime Environment that allows the access to the resources offered by the host. The container presents to the contained agents common services, such as messaging transport, local event communication, and support for access to external data repositories. Containers implement connectivity services among them for message interchange, and for agent and services migration. They also provide connectivity and state monitoring of external agents, and they instantiate proxies to make transparent the communication between external and internal agents.

4.1 The communication infrastructure

Since early stages of the design, this agent platform has been envisioned as the cornerstone of the distributed architecture for automation systems. In particular, under our conception this environment not only corresponds to the space where agents can perform their duties (as all platforms do), it is also aimed to provide a reliable communication infrastructure that agents can (and should) exploit to interact among themselves in a distributed application. As result, the G++ Agent Platform is able to offer an implementation of a robotic and automation system that will delegate to the own agent's container the conduction of the major communication traffic. So agents can communicate among themselves asking their own container to deliver the message to its destination. Messages are delivered following the best effort policy (i.e. no unnecessary delays are introduced in their expedition), but it is not guaranteed their reception in the right order. This can happen for two main reasons: (1) the latency of the Internet, plus costs incurred in retransmissions of packets naturally tends to increase the time required to transmit a message over long distances, and (2) the interconnections between containers define the paths that messages have to follow from the source to the target, each node acting as a router (the processing time on each container has to be added to the network delays described above). The platform, however, can guarantee the delivery of messages, detecting and informing the sender when they are not arrived within the pre-established time. A time window and a timestamp message field are used in the message for this scope. The time window value can also be infinite, which means no time window is specified. The message timestamp can also be useful to the message receiver, to determine the exact sequence of messages.

4.2 Virtual mobility

Virtual Mobility allows agents to be suspended, transported and restored in diverse containers. Mobility can be decided autonomously by the agent, in terms of the moment and destination in which it will be done, or can be enforced by the agent's owner, or by another agent. Mobility is implemented through the serialization of the state of the agent, the transport together with the code (if necessary), and the de-serialization at the destination container. The platform does not provides support for the serialization of the stack of calling methods, so when this procedure is activated, the agent has to be suspended. When an agent is moved from its home container to a foreign container, its original agent management system together with the mobility service is responsible to keep trace of the new position of the agent. In such a way, it is possible to implement automatic roaming in the communication to the agent.

4.3 Interaction with the environment

The structure of an agent considers a subsystem responsible for achieving information from its environment, where the environment can be virtual, composed by software processes or systems running in a computing device, or physical, as the real world is. For example, a virtual agent can be able to listen to keystrokes, listen to messages sent by other agents, receive network information, or perceive events from the operating system; a robotic agent can be enabled with sonars, infrared range sensors, accelerometers or gyroscopes to perceive the physical environment and its own relationship with it. On the other hand, agents must be able to act over its environment in order to achieve their goals. The actions can result in a virtual effect, such as the creation of files, the communication of messages to other agents, or physical, as commands over the engine in a wheel-enabled robot. Sensors and actuators are closely related to the environment because their functionality depends directly on the aspects that they have to detect. In this way, sensors and actuators are device-dependent. However, enabling software agents with specific sensors and actuators can limit their mobility in virtual spaces. The G++ Agent Platform manages sensors and actuators through interface objects that can be attached to agents in runtime. This allows a migrating agent to get access to the specific sensors and actuators offered in each container/platform. This flexibility is obtained providing a common interface for all sensors and actuators that the agent must use to interact with. Also is supported the definition of descriptors to recognize sensors and actuators that the agent could access. The independence between the agent implementation and its environment makes it possible to follow an evolutionary approach in the development of software agents. Portability of agents allows new agents to be tested in simulation environments before they are deployed in the real world (e.g. a mobile robot controller). On the other hand, the model well suits for agents based on learning architectures or requiring initial training (such as those based on neural networks), because they can be conditioned for final execution in a simulated environment.

5 The Agent-Based architecture for automation and robotics

This agent-based architecture introduces a communication standard and a set of services to build global automation systems in different domains. The former defines the languages that will be used for exchange of information between entities participating in automation systems. The latter, the set of services that are available for supporting their activity. Three services are offered at this level:

a) *Messaging*: it provides persistence and reliability in direct messaging between senders and well-defined receivers. It is based on based on persistent messages queues, which allows time-decoupled communications among participants b) *Event distribution*: it implements the asynchronous publish/subscribe communication model. Each container provides local event publication and notification services. The architecture for global automation systems includes agents for the management of distributed subscriptions and notifications. c) *Service brokering*: it supports dynamic reconfiguration of the relationships between service providers and consumers. Each container provides local event publication and notification services. The architecture for global automation systems includes agents for the management of distributed subscriptions and notifications (that is, among different service points). The design of the services has explicitly considered the problem of distribution, particularly the unreliability of network connections, which makes indistinguishable crashed components from slow components. This problem, common to all implemented services, was addresses through a mechanism of registration and renewal of the registration with the service provider, that interested users must perform during their lifecycle.

6 An example of domain-specific multi agent system

For the domain-specific application example, let us consider a colony of mobile robots that have to explore a surface, and cooperate synchronously collecting certain objects, that must be carried to the robots' base. This colony requires the participation of different kinds of autonomous devices:

- Explorers, which are autonomous mobile robots provided with telecameras and grippers, that recognize objects to be collected, and carry them to the nearest transport robot. - Transport robots: they are autonomous mobile robots that receive the objects collected by explorers and transport them in batches to the colony nest. - Coordinator: is the autonomous system that receives communication from carriers and coordinates the operation of explorer and transport robots. It operates in a fixed computing device, which is located outside the exploring area, and connected through satellite to the Communicator robot. - Communicator: is a mobile device that acts as a gateway between the Coordinator and the robots located in the surface to explore. - Colony Nest: is the computing device that runs messaging and brokering services of the nest.

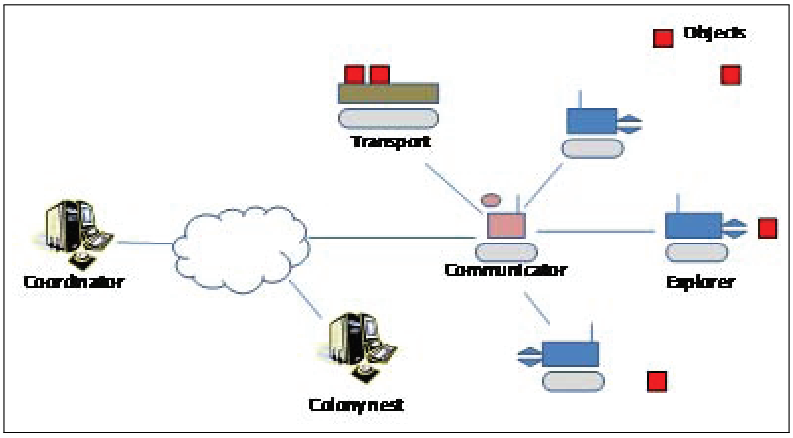


Figure 2: A colony of explorers.

In this system all the members run a container where operates the agent that implements the own control module. At startup, the containers pertaining to all devices in the surface to explore establish communication among themselves, using a peer-to-peer discovering protocol. Then they register their service capabilities in the ServiceBroker running in the Colony Nest device. The coordinator, remotely located, creates a data channel with the Colony Nest, using a known IP direction to locate it. Thus, the coordinator asks the service broker located in the Colony Nest for the suitable robotics devices to accomplish the task. From the answer provided by the service broker, the coordinator selects a Communicator device, some explorers and transport robots. The Coordinator transmits the mission to each robotic device. The explorers move across the surface detecting objects to be collected, and transmit their positions to the Coordinator. With this information, the coordinator assigns routes to the transport robots, in order to pick up the objects collected by the explorers. If an explorer or transport robot goes out of the communication range, it starts a "turn back home" procedure in order to reestablish communication. If messages had been sent to that robot during the "blackout", they are kept in the messages queue of the MessagingService and transmitted when the device is "visible" again. The Coordinator could be subscribed in the event broker to listen for certain events, such as mechanical failures. Thus, if a failure message is received, the coordinator can take measures like a replacement for the defective robot. In the following subsections, the system is described using the model presented in this chapter.

6.1 Environment and autonomous equipments (levels 1 and 2)

The environment is mainly composed by the physical surface that have to be explored, the objects that have to be collected, the obstacles in the route of each robot, etc. Also, available communication networks or global positioning systems that can be accessed by devices are considered part of the environment. In terms of autonomous equipments we have all the devices participating in the colony, including the Coordinator and the Colony Nest devices. Robots are mobile vehicles enabled with sensors (cameras, GPS, encoders, etc.), actuators (engines, grippers, etc.) and communication interfaces (Wi-Fi, satellite, etc.) that must be available for local control modules (robot controllers).

6.2 The agent platform (level 3)

The G++ Agent Platform is used to provide the underlying software infrastructure for the implementation control modules. Each device must provide a Java Runtime Environment where a G++ Container will run. Also, in each robotic device the agent platform must have access to the API (application programming interface) of the available sensors and actuators, in order to build the access to physical components of the robots. The access to the communication stack is obtained, in general cases, through the standard TCP/IP interface provided by the device's operating system.

6.3 Agent-based architecture (level 4)

The agent based-architecture is made-up by all the components required to achieve interoperability among the different participants. For example, the ServiceBroker agent, which is responsible for maintaining the network location (IP address) of each robotic device, and the list of services that them are capable to provide. Another agent that participates in the architecture is the MessengerAgent that supports message queues for reliable delivery of message to mobile robots.

6.4 The domain-specific multi-agent system (level 5)

Each participant in the colony is conceptualized as a software agent, programmed to accomplish its own mission. The implementation requires providing every one of the devices with an agent/control. The control modules can be implemented using different artificial intelligence model. At this level must be also programmed the standard interfaces to the sensors and actuators, and registered locally as service objects in the device's container.

7 Conclusion

In this work we have described our framework for the implementation of distributed robotics and automation systems. Its design was driven by the interest to obtain a decoupled and scalable infrastructure in different application scenarios. This approach emphasizes software engineering aspects of agency, which is a differentiating point when comparing it with other architectures, whose functionalities are more focused in distributed artificial intelligence. Currently we are developing some case of studies that can help us to test the framework in systems offering different complexity levels.

8 Acknowledgement

This work has been partially funded by CONICYT through Fondecyt Project No. 11080284 and the Pontificia Universidad Católica de Valparaíso (www.pucv.cl), through Nucleus Project No. 037.115/2008 "Collaborative Systems".

Bibliography

[1] AgentLink. *Software Products for MultiAgent Systems.* Technical Report. Europe's Network of Excellence for Agent-Based Computing, 2002.

[2] F. Bellifemine, A. Poggi, G. Rimassa. Jade, a FIPA-Compliant Agent Framework. Proceedings of the 4th Int. Conference on Practical Applications of Intelligent Agents and Multi-Agent Technology, 1999.

[3] P. Eugster, P. Felber, R. Guerraoui, A. Kermarrec. The Many Faces of Publish/Subscribe. ACM Computing Surveys, Vol. 35, No. 2 (June 2003), p. 114-131.

[4] T. Finin, D. McKay, R. Fritzson, R. McEntire. KQML: An Information and Knowledge Exchange Protocol.Proceedings of the Int. Conference on Building and Sharing of Very Large-Scale Kniowledge Bases, December 1993.

[5] FIPA (2002). FIPA ACL Message Structure Specification. Standard N. SC00061G, December 2002.

[6] M. Genesereth, S. Ketchpel. Software Agents. Communications of the ACM, Vol 37, No. 7, 1994, p. 48-53.

[7] F. Guidi-Polanco, C. Cubillos, G. Menga. The Agent-Based GAP: A Framework for Global Automation Systems, Proceedings of the 13th IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises, p.53-58, June 14-16, 2004

[8] Z. Haibin. A Role-Based Approach to Robot Agent Team Design. Proceedings of the IEEE International Conference on Systems, Man and Cybernetics SMC, 2006. Vol.6 pp.4861-4866, October 2006.

[9] N. Jennings. An Agent-Based Approach for Building Complex Software Systems. Communications of the ACM, Vol 55 No. 4 (April 2001), p. 35-41.

[10] L. Jianhui, W. Kesheng, G. Hang, Q. Ligang. An OOT-supported migration approach to holonic robot assembly cell. Proceedings of the 8th Int. Conference on Computer Supported Cooperative Work in Design, 2004. Vol.2 pp. 498-501.

[11] C. Lim, R. Mamat, T. Braunl. Market-based approach for multi-team robot cooperation. 4th International Conference on Autonomous Robots and Agents, ICARA 2009, pp.62-67, Feb. 2009.

[12] D. Mamady, G. Tan, M. Toure. An artificial immune system based multi-agent model and its application to robot cooperation problem. Proceedings of the 7th World Congress on Intelligent Control and Automation, WCICA 2008, pp.3033-3039, June 2008.

[13] G. Moro, A. Natali. On the Event Coordination in Multi-Component Systems. Proceedings of SEKE 2002, Ischia, Italy.

[14] J. Odell, H. Van Dyke Parunak, B. Bauer. Extending UML for Agents. Proceedings of the Agent-Oriented Information Systems Workshop at the 17th National Conference on Artificial Intelligence 2000.

[15] T. Rogers, A. Sekmen, J. Peng. Attention Mechanisms for Social Engagements of Robots with Multiple People. The 15th IEEE International Symposium on Robot and Human Interactive Communication, ROMAN 2006, pp.605-610, Sept. 2006.

[16] M. Sims, D. Corkill, V. Lesser. Separating Domain and Coordination in Multi-Agent Organizational Design and Instantiation, Proceedings of the International Conference on Intelligent Agent Technology, IAT 2004.

[17] P. Valckenaers, H. Van Brussel, T. Holvoet. Fundamentals of Holonic Systems and Their Implications for Self-Adaptive and Self-Organizing Systems. Proceedings of the Second IEEE International Conference on Self-Adaptive and Self-Organizing Systems Workshops, SASOW 2008., pp.168-173, Oct. 2008.

[18] G. Weiss. Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence. MIT Press, Cambridge, Massachusetts, 1999

[19] D. Weyns, M. Schumacher, A. Ricci, M. Viroli, T. Holvoet. Environments for multiagent systems, State-of-the-art and research challenges. In Lecture Notes in Computer Science, vol 3374 (2005), Berlin, Heidelberg, Germany.

Int. J. of Computers, Communications & Control, ISSN 1841-9836, E-ISSN 1841-9844
Vol. VI (2011), No. 1 (March), pp. 90-100

Speed Control of a Permanent Magnet Synchronous Machine Powered by an Inverter Voltage Moment Approach

J. Khedri, M. Chaabane, M. Souissi

Jamel Khedri, Mohamed Chaabane (corresponding author), Mansour Souissi
Research Unit of Industrial Processes Control
National School of Engineers of Sfax
ENIS, Route soukra, km 3.5 - BP.W, 3038, Sfax, Tunisia
E-mail: {khedrijamel,chaabane_uca}@yahoo.fr, mansour.souissi@ipeis.rnu.tn

Abstract: In this paper, the method of moments is presented in order to synthesize controllers for current and speed of a Magnet Permanent Synchronous Machine (PMSM). The controller's dynamics are chosen in respect with the time domain constraints, which have equality relationships between the moments of the closed-loop system and those of the reference model. The Linear Matrix Inequality (LMI) formalism is used for the controller synthesis. The proposed controller is applied for the speed tracking of a PMSM to illustrate the performances of the method.

Keywords: Times Specifications, method of moments, Reference Model, LMI, PMSM.

1 Introduction

New industrial applications require variable speed drives with high dynamic performance. In recent years several techniques of control have been developed, allowing PMSM with variable speed to achieve these performances. However, vector control, which allows decoupling between control variables remains the most widely used [1–3]. The main advantages of this configuration are that the regulatory cascade is a very industrial responded [4, 5]. It gives high dynamic performance for a wide range of applications. The complexity of the dynamic model of the permanent magnet synchronous variable speed and the presence of external disturbances and parametric variations limit the performance of the control law [6]. Thus, our main contribution in this paper consist in introducing a time constraint with the help of a reference model, simple to operate and which characterizes the dynamics of the closed loop system. This time constraint is formulated by the equality of the first time moments of the closed-loop system and those of the reference model [7–11].

Practically, a first LMI is used to insure closed-loop stability, a second one permits, by minimizing the norm 2 of time moments cost, to identify time specifications (overshoot, response time ...).

Without loss of generality, the methodology has been restricted in this paper to the case of square, invertible, MIMO (multi inputs, multi outputs) systems.

The paper is organised as follows: in section 2, we define the state space representations of the systems and those of the controller; moreover, the problem of the design of a controller verifying stability and time performances is stated. In section 3, we propose a solution of this problem, owing to the demonstration of a specific theorem. Experimental results applied to a permanent magnet synchronous machine (PMSM) are presented in section 4.

2 Problem statement

Consider a Linear Time Invariant (LTI) square invertible system $H(s)$ with m inputs and m outputs:

$$\begin{cases} \dot{x}_{sys}(t) = A_{sys}x_c(t) + B_{sys}u(t) \\ y(t) = C_{sys}x(t) \end{cases} \tag{1}$$

The controller is modeled in the state space by:

$$\begin{cases} \dot{x}_c(t) = A_c x_c(t) + B_c e(t) \\ u(t) = C_c x_c(t) + D_c e(t) \end{cases} \tag{2}$$

with

$$e(t) = r(t) - y(t) \tag{3}$$

where: $x_{sys}(t) \in \mathbb{R}^l$ is the state vector of the system,$u(t) \in \mathbb{R}^m$ is the control input,$y(t) \in \mathbb{R}^m$ is the output,$r(t) \in \mathbb{R}^m$ is the reference of closed-loop system, Figure 1. Matrices A_{sys}, B_{sys}, C_{sys} are supposed to be known with appropriate dimensions. The controller model matrices A_c, B_c, C_c and D_c are to be computed.

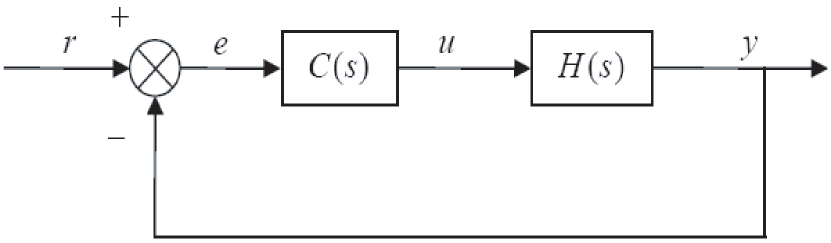


Figure 1: Closed-Loop System

The synthesis of controller requires the construction of the augmented model which regroups the system and the controller model. Consider the augmented state vector $z(t)$ defined as:

$$z(t) = \begin{bmatrix} x_{sys}(t) \\ x_c(t) \end{bmatrix}. \tag{4}$$

Then, the closed-loop system, Figure 1, can be represented by the following model:

$$\begin{cases} \dot{z}(t) = A_{bf}z(t) + B_{bf}r(t) \\ y(t) = C_{bf}z(t) \end{cases} \tag{5}$$

with:

$$A_{bf} = \begin{bmatrix} A_{sys} - B_{sys}D_cC_{sys} & B_{sys}C_c \\ -B_cC_{sys} & A_c \end{bmatrix}; \quad B_{bf} = \begin{bmatrix} B_{sys}D_c \\ B_c \end{bmatrix}; \quad C_{bf} = \begin{bmatrix} C_{sys} & 0 \end{bmatrix} \tag{6}$$

Let's note that the closed-loop stability is ensured if and only if there exist a positive definite matrix P with $P = P^T$, such as the LMI

$$PA_{bf} + A_{bf}^T P < 0 \tag{7}$$

is satisfied.

To facilitate the study, our work is restricted to the case where the controller $C(s)$ is to be synthesised. The controller design can be considered as a static feedback gain K for the augmented system.

The controller synthesis is ensured by the fact that matrices A_c and B_c are defined a priori and only the matrices C_c and D_c are to be computed. For this purpose, we consider the following hypothesis:

The matrix A_{bf} of augmented system can be written as follows:

$$A_{bf} = \tilde{A} + \tilde{B}K\tilde{C} \tag{8}$$

where:

$$\tilde{A} = \begin{bmatrix} A_{sys} & 0 \\ -B_c C_{sys} & A_c \end{bmatrix}; \quad \tilde{B} = \begin{bmatrix} B_{sys} \\ 0 \end{bmatrix}; \quad \tilde{C} = \begin{bmatrix} -C_{sys} & 0 \\ 0 & I \end{bmatrix} \tag{9}$$

It is to notice that the gain K is defined as:

$$K = \begin{bmatrix} D_c & C_c \end{bmatrix} \tag{10}$$

The extended system is then characterized by $(\tilde{A}, \tilde{B}, \tilde{C})$. The objective of this work is to ensure closed-loop stability and to specify transient performances which are characterized by a reference model $T_{ref}(S)$. The Dynamics of the reference model are determined by the use of the time moments approach. In this case, the time moments of transfer $W(S)$, which are represented in state space by (A_W, B_W, C_W, D_W) (see [4]), are given by the following relations:

$$\begin{aligned} M_{W,0} &= -C_W A_W^{-1} B_W + D_W \\ M_{W,j} &= (-1)^{j+1} C_W A_W^{-(j+1)} B_W + D_W \end{aligned} \tag{11}$$

$(j = 1...\infty)$
$M_{W,0}$: is the 0^{th} order moment.

The exposing j refers to the order of moments. For applying the proposed moment approach, we will define briefly the first three time moments.

The 0^{th} order moment M_0 is equal to the static gain of the system and represents the area of its impulse response. The first order moment M_1 characterizes the average time t_m , the same as the average value of a probability density: $tm = \frac{M_1(g)}{M_0(g)}$

$g(t)$ is the impulse response of the system, represented by Figure 2.

For a first order system, we have $M_1 = \tau M_0$ where represents the time constant of the system. More generally, M_1 is used to characterize the time response of the considered system.

The second order moment M_2, characterizes the variance σ (or dispersion $\Delta\tau$) of the impulse response around its mean time tm.

In practice, the time moments M_0, M_1, M_2 carry sufficient information on the impulse response: static gain, mean time and dispersion.

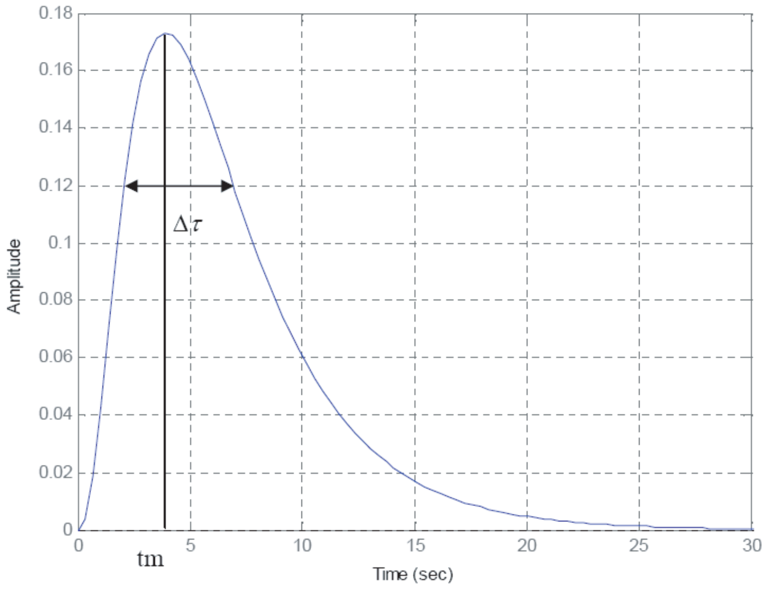


Figure 2: Characterization of an impulse response

The controller has to be designed such that the closed-loop transfer $T(S)$ is equal to the reference model $T_{ref}(s)$.

$$(I + H(s)C(s))^{-1}H(s)C(S) = T_{ref}(s) \tag{12}$$

or

$$C(s)(I - T_{ref}(s)) = H^{-1}(s)T_{ref}(s) \tag{13}$$

It is known that such control law $u(t)$ is in the form: $u(t) = Ke(t)$ Using state space representation of $C(s)$, we can write:

$$u(s) = [C_c(sI - A_c)^{-1}B_c + D_c]e(s) \tag{14}$$

While referring to considered hypothesis given by relation (9), equation (13) can be written as:

$$K \begin{bmatrix} I \\ (sI - A_c)^{-1}B_c \end{bmatrix} (I - T_{ref}(s)) = H^{-1}(s)T_{ref}(s) \tag{15}$$

If we set:

$$F(s) = H^{-1}(s)T_{ref}(s) \tag{16}$$

$$W(s) = \begin{bmatrix} I \\ (sI - A_c)^{-1}B_c \end{bmatrix} (I - T_{ref}(s)) \tag{17}$$

Then, the time constraint (15) can be expressed by:

$$KW(s) = F(s) \tag{18}$$

Notice that the last relation (18) is linear, then the formalism LMI can be applied. In fact, an ideal equality in the form (18) between the closed-loop $T(s)$ and $T_{ref}(s)$ is not possible to obtain.

Therefore, it is recommended to replace an ideal equality by an approximation between $T(s)$ and $T_{ref}(s)$. In order to achieve this condition, we use the time moments approach, as mentioned previously.

Therefore, when the objective is to characterize the time response, the equality (18) is transformed to a minimization of the quadratic distance between the first $(n+1)$ time moments of the closed-loop system and its reference model, where the minimization criterion is defined as:

$$(KM_{W,j} - M_{F,j})^T(KM_{W,j} - M_{F,j}) < \gamma I \tag{19}$$

$j = 0, 1, ..., n$.
Using this criterion, we can notice that the transient responses of $T(s)$ and $T_{ref}(s)$ become approximately similar (they have approximately the same overshoot, the same response time, etc.)

The overall objective is achieved if there exist a symmetric and positive defined matrix P and a gain K given by relation (10). In such case, the closed-loop stability condition (7) and the minimization criterion condition (19) are satisfied. In the next section, we present a solution to this problem, which can be formulated by the theorem 1.

3 Mains results

The objective of this section is to give solution of the previous problem.

3.1 Theorem 1.

Consider a LTI system $(A_{sys}, B_{sys}, C_{sys})$ (1) and the corresponding augmented system $(\tilde{A}, \tilde{B}, \tilde{C})$ defined in (9). Consider a static state feedback K_e stabilizing the pair $(\tilde{A}, \tilde{B})$.

The stability of the closed-loop system and the time constraints imposed by the reference model $T_{ref}(s)$ are guaranteed, if the optimisation problem:
Minimum γ
(L, G, P) with the constraints
$P > 0$

$$\begin{bmatrix} Sym\{P(\tilde{A} + \tilde{B}K_e)\} & (L\tilde{C} - GK_e)^T + P\tilde{B} \\ (L\tilde{C} - GK_e) + \tilde{B}^T P & -(G + G^T) \end{bmatrix} < 0 \tag{20}$$

and

$$\begin{bmatrix} -I & (LM_{W,j} - GM_{F,j}) \\ (LM_{W,j} - GM_{F,j})^T & -\gamma I \end{bmatrix} < 0 j = 0, 1, ..., n. \tag{21}$$

has a solution, then the gain of dynamic controller $C(s)$ is given by :

$$K = G_{opt}^{-1} L_{opt} \tag{22}$$

Notice that L_{opt}, G_{opt} correspond to the minimal value of γ denoted γ_{opt}. Before demonstration of Theorem 1, let us recall the following result [6, 7].

3.2 Lemma 1.

Consider the augmented system (5), where the state matrix A_{bf} is defined by (8). The following statements are equivalent:

1) There exist a symmetric and positive defined matrix $P = P^T > 0$ and two matrices K_e and K such that:

$$P(\tilde{A} + \tilde{B}K_e) + (\tilde{A} + \tilde{B}K_e)^T P < 0 \tag{23}$$

$$P(\tilde{A} + \tilde{B}K\tilde{C}) + (\tilde{A} + \tilde{B}K\tilde{C})^T P < 0 \tag{24}$$

2) There exist a symmetric and positive defined matrix $P = P^T > 0$, a non singular matrix G and two matrices K_e and K such that:

$$\begin{bmatrix} Sym\{(\tilde{A} + \tilde{B}K_e)^T P\} & P\tilde{B} \\ \tilde{B}^T P & 0 \end{bmatrix} + sym\{\begin{bmatrix} 0 \\ I \end{bmatrix} G[(K\tilde{C} - K_e) - I]\} \tag{25}$$

3.3 Proof of lemma 1

First of all, notice that the orthogonal complements of the second term of (25) are:
$V = \begin{bmatrix} I & 0 \end{bmatrix}$, the orthogonal complement of V, such that $VV^{\perp} = 0$, is $V^{\perp} = \begin{bmatrix} 0 \\ I \end{bmatrix}$;
and $U = \begin{bmatrix} I & (K\tilde{C} - K_e)^T \end{bmatrix}$, the orthogonal complement of U, such that $UU^{\perp} = 0$, is:
$U^{\perp} = \begin{bmatrix} (K\tilde{C} - K_e)^T \\ -I \end{bmatrix}$
Applications of elimination lemma [9] lead to left and right multiplication of (25) by V and $V^{\perp}$ respectively, which gives inequality (23):

$$\begin{bmatrix} I & 0 \end{bmatrix} sym\{\begin{bmatrix} 0 \\ I \end{bmatrix} G \begin{bmatrix} (K\tilde{C} - K_e) - I \end{bmatrix}\} \begin{bmatrix} I \\ 0 \end{bmatrix} = P(\tilde{A} + \tilde{B}K_e) + (\tilde{A} + \tilde{B}K_e)^T P < 0$$

Even, if (25) is left and right multiplied respectively by U and $U^{\perp}$, we get inequality (24).

3.4 Proof of Theorem 1

By using Shur complement, the condition (21) can be expressed as:

$$(LM_{W,j} - GM_{F,j})(LM_{W,j} - GM_{F,j})^T < \gamma I \tag{26}$$

$j = 0, 1, ..., n$.
Besides, if one multiplies equation (18) by G:

$$LW(s) = GF(s) \tag{27}$$

equation (19) becomes:

$$(LM_{W,j} - GM_{F,j})(LM_{W,j} - GM_{F,j})^T < \gamma I \tag{28}$$

$j = 0, 1, ..., n$.
One demonstrates that (26) and then the condition (21) correspond to the minimization of the 2 norm of the error between the $(n+1)$ first moments of the closed-loop and those of the reference model. If the minimization of γ_{opt} is satisfied, then the two transfers $T(s)$ and $T_{ref}(s)$ are very close in low frequencies, which implies the desired transients for the closed-loop system. The stability is also insured due to the fact that condition (20) implies condition (24), equivalent to condition (7) which expresses closed-loop stability.

4 Application to the Permanent Magnet Synchronous Machine (PMSM)

Our Permanent Magnet Synchronous Machine (PMSM) is powered by a two levels voltage inverter. To obtain non reciprocating quantities, our model is done in the Park landmark [10]. Thus we have the following equations in the synchronously d-q reference frame [11, 12]:

$$\left\{\begin{array}{c} \dot{I}_d = \frac{-R_s}{L_d} I_d + \frac{L_q}{L_d} w_r I_q + \frac{1}{L_d} v_d \\ \dot{I}_q = \frac{-R_s}{L_q} I_q - \frac{L_d}{L_q} w_r I_d - \frac{\phi}{L_q} w_r + \frac{1}{L_q} v_q \\ \dot{w}_r = p\frac{(c_e - c_r)}{J} - \frac{f_c}{J} w_r \\ c_e = p[(L_d - Lq)I_d + \phi]I_q - \frac{f_c}{J} w_r - \frac{f_c}{J} w_r \\ \dot{w}_r = \frac{p^2}{J}[(L_d - Lq)I_d + \phi]I_q - \frac{f_c}{J} w_r - \frac{p}{J} c_r \end{array}\right. \tag{29}$$

where v_d and v_q are the stator voltages, I_d and I_q are the stator currents, $L_d$and L_q are the inductances, R_s is the stator winding resistance, ϕ is the flux linkage of the permanent magnets, w_r is the angular velocity of the motor shaft, f_c is the friction coefficient relating to the rotor speed; J is the moment of inertia of the rotor , c_e is the electromagnetic torque, c_r is the load torque and p is the number of pole. In an equivalent manner, these equations can be written in the form of a general transfer function $H(s)$ who contains the open-loop transfer of variables to control namely $H_{id}(s)$ and $H_{wr}(s)$.

$$H(s) = \left[\begin{array}{cc} H_{id}(s) & 0 \\ 0 & H_{wr}(s) \end{array}\right] \tag{30}$$

where:

$$H_{id}(s) = \frac{\frac{1}{R_s}}{1 + \tau_{id}(s)}, \quad H_{wr}(s) = \frac{H_{bfIq}\frac{p^2\phi}{f_c}}{1 + \tau(s)} \tag{31}$$

where $\tau_{id} = \frac{L_d}{R_s}$; $\tau = \frac{J}{f_c}$
Let $H_{bfIq}(s)$,is the closed-loop transfer of the current I_q: $H_{bfIq} = \frac{1}{1+\tau_{bfiq}(s)}$ τ_{bfiq} is chosen equal to 7.0175 10-4 s in order to perform a time response of the closed-loop system ten times faster than open-loop.
Notice that: $\tau_{iq} = \frac{L_q}{R_s}$; $H_{iq}(s) = \frac{\frac{1}{R_s}}{1+\tau_{iq}(s)}$
The objective is to reduce the time response of the system, to keep the overshoot of the closed-loop inferior to 5% while decoupling the two outputs. These specifications can be carried out by a second order reference model with static unity gain for the speed, and a first order one for the current I_d:

$$T_{ref}(s) = \left[\begin{array}{cc} \frac{1}{1+\tau_{idref}s)} & 0 \\ 0 & \frac{1}{1+\frac{2\zeta}{w}(s)+\frac{s^2}{w^2}} \end{array}\right] \tag{32}$$

The design choices correspond to $\tau_{idref} = 0.1\tau_{id}$ in order to perform a time response of the closed-loop current ten times faster than the open-loop.
$\zeta = 7.6205$, $w = 93.906 rds{-}1$ in order to obtain a closed-loop speed 3.3 times faster than the open-loop one. Let's note that H_{wr} has two constants time:
$\tau = \frac{J}{f_c} = 0.5333s$, $\tau_1 = 7.047210^{-4}s^{-1}$ The two constants time of the closed-loop speed are $\tau bf = 0.1618s$, $\tau_{1bf} = 7.017510^{-4}s$ Notice that the decoupling specification is insured by a diagonal reference model. The d-axis current Id is regulated to follow a zero setpoint. The static

state feedback K_e stabilizing the pair $(\tilde{A}, \tilde{B})$ such as (23) feasible has been computed:

$$K_e = \begin{bmatrix} -79.1667 & 0 & 0 & 722 & 0 \\ 0 & 0 & -2.1433 & 0 & 0.0939 \end{bmatrix} \tag{33}$$

Solving LMIs (20) and (21) by LMI optimization algorithm, the matrices P, G and L can be obtained:

$$G = 10^8 \begin{bmatrix} 0.0065 & 0 \\ 0 & 4.5881 \end{bmatrix}; \gamma_{opt} = 6.609610^{-14} \tag{34}$$

The obtained static gain is:

$$K = \begin{bmatrix} 5.7 & 0 & 722 & 0 \\ 0 & 0.5006 & 0 & 0.09386 \end{bmatrix} \tag{35}$$

5 Experimental results and interpretations

The implementation of our model was performed on a test benchmark consisting of two Permanent Magnet Synchronous Machines (figure 3) fitted by the same encoder and powered by a voltage inverter; one was used as a motor and the other one as a load. The proposed control algorithm was executed by the use of Matlab/ Simulink software. Then it was compiled and implemented on Dspace 1104. The digital sampling period was taken equal to $0.1ms$.

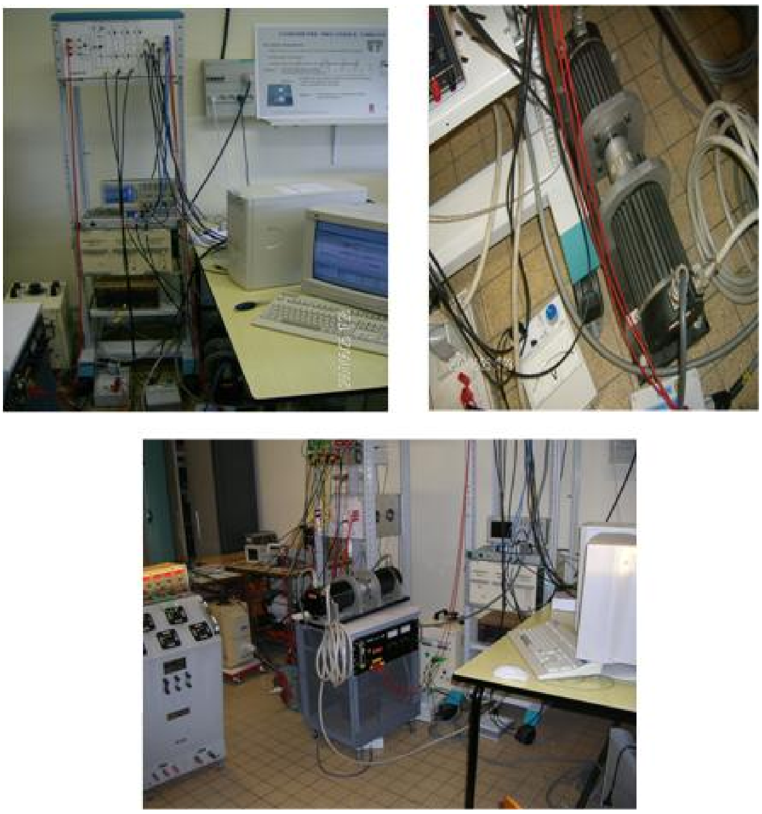


Figure 3: Photo of the test bench.

Figures (4)-(6) illustrate the experimental results of the dynamic behaviour of PMSM for speed operating without charge.

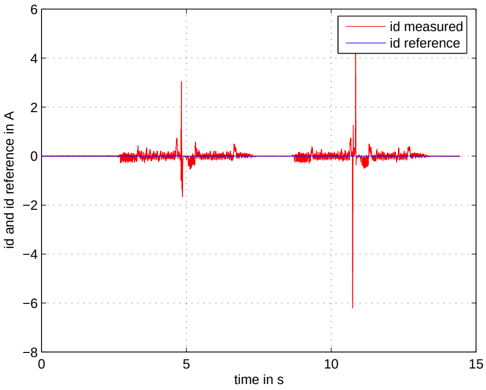


Figure 4: d-axis current and its reference

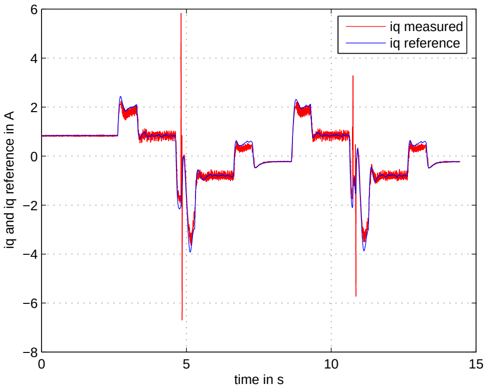


Figure 5: q-axis current and its reference

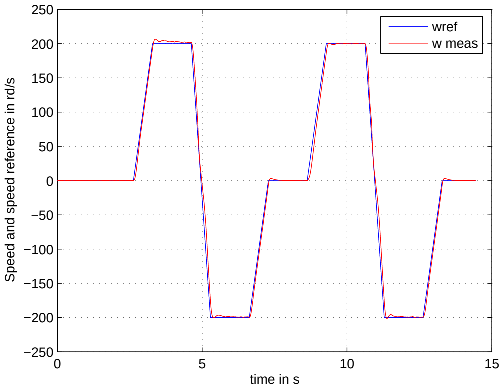


Figure 6: Speed and its reference

Whose data are listed in the Appendix. The results show that the closed loop system with the synthesized controller have a good behaviour: indeed, the measured speed and dq-axis currents track well the trajectory of reference one with good accuracy over the whole speed range, moreover, the time constraint imposed with the help of time moments has permitted to keep the closed loop outputs very close to those of the reference model.

6 Conclusion

In this paper, a controller design, for current and speed tracking of a PMSM has been developed by using the method of moments. The controller's dynamics are chosen in respect with the time domain constraints. These constraints have equality relationships between the moments of the closed-loop system and those of the reference model. The proposed controller has been achieved in order to ensure the closed-loop stability and to specify transient performances in regard with a reference model. In this study, the LMI formalism has been used for the controller synthesis. Finally, we have applied the proposed controller for the speed tracking of a PMSM which has given satisfactory results.

7 Acknowledgements

The authors wish to thank all the team of the Laboratory of Automatic and Industrial Informatics of the University of Poitiers FRANCE and special thanks to Pr. Gerard Champenois for the aid he afforded us for the realization of our experimental work validation.

8 Appendix : Motor parameters

Motor rated power	1 KW
Rated current	6.5 A
Pole pair number (p)	2
d-axis inductance Ld	4.5mH
Stator resistance	0.56 Ω
Motor inertia J	$2.08.10 - 3Kg.m^2$
Friction coefficient fc	$3.9.10^{-3}Nm.s.rad^{-1}$
Magnet flux constant ϕ	0.064 wb

Bibliography

[1] Vas, P., Sensorless vector and direct torque control, *Oxford, U. K, Oxford Univ. Press*, 1998.

[2] Bose, B. K., Power electronics and AC drives, *Englewood Cliffs, NJ, Prentice-Hall*,1986.

[3] Bose, B. K., Power electronics and variable frequency Drives, *Technology and Appliquation*,IEEE press 1997.

[4] Flaus, J. M, Industrial control, Hermes, Paris 1994.

[5] Maret, L., automatic control, Polytechnic Press romandes, 1987.

[6] Lin, F-J, Real time position controller design with torque feedforward control for a PM synchronous motor , *IEEE Trans. On Industrial Electronics*, Vol.44, N°.3, June 1997, pp. 398-407.

[7] Bentayeb, A., N. Maamri and J-C Trigeassou, The Moments in Control: a tool for Analysis, Reduction and Design,*International Journal of Computers, Communications & Control (IJCCC).* Vol. 2, num. 1, pp. 82-103, 2007.

[8] Maamri, N. and Trigeassou, J. C, PID design for time delayed systems by the method of moments ,*European Control Conference*, Groningen Holland (1993).

[9] Maamri, N. and Trigeassou, J. C, Controllers design by moments placement ,*European Control Conference*, Rome Italy (1995).

[10] Maamri, N., Bentayeb, A. and Trigeassou, J. C, Design and Iterative optimization of Reduced Robust Controllers with Equality Constraints , RECOND-Milan (2003).

[11] Trigeassou, J. C., Method of moments in automatic . *Conférence Internationale Francophone d'Automatique*, CIFA, Lille (2000).

[12] Mehdi, D., Control theory some insights . Report Automatic Control UCA-ENIS Sfax Tunisia Avril (2004).

[13] Peaucelle, D. and Arzelier, D., An efficient numerical solution for H2 static output feedback synthesis . *European Control Conference*, 2001, pp.3800-3805.

[14] Skelton, R. E., Iwasaki, T. and Grigoriadis, K., a unified algebraic approach to linear control design , Taylor & Francis, (1998).

[15] Bose, B. K., Power electronics and AC drive . New york, Prince Hall, first Edition, (1986), chapter 2, page 95.

[16] Sturtzer, G. and Sanigiel, E., Modeling and Control of three-phase motors .

[17] Hassaine, S., Applications of new technology command to permanent magnet synchronous machine. Thesis, ESIP, Poitiers, March 2008.

Int. J. of Computers, Communications & Control, ISSN 1841-9836, E-ISSN 1841-9844
Vol. VI (2011), No. 1 (March), pp. 101-112

An Optimal Task Scheduling Algorithm in Wireless Sensor Networks

L. Dai, Y. Chang, Z. Shen

Liang Dai, Yilin Chang, Zhong Shen
State Key Laboratory of Integrated Service Networks
Xidian University
Xi'an 710071, China
E-mail: ldai1981@gmail.com, ylchang@xidian.edu.cn, zhshen@mail.xidian.edu.cn

Abstract: Sensing tasks should be allocated and processed among sensor nodes in minimum times so that users can draw useful conclusions through analyzing sensed data. Furthermore, finishing sensing task faster will benefit energy saving, which is critical in system design of wireless sensor networks. To minimize the execution time (makespan) of a given task, an optimal task scheduling algorithm (OTSA-WSN) in a clustered wireless sensor network is proposed based on divisible load theory. The algorithm consists of two phases: intra-cluster task scheduling and inter-cluster task scheduling. Intra-cluster task scheduling deals with allocating different fractions of sensing tasks among sensor nodes in each cluster; inter-cluster task scheduling involves the assignment of sensing tasks among all clusters in multiple rounds to improve overlap of communication with computation. OTSA-WSN builds from eliminating transmission collisions and idle gaps between two successive data transmissions. By removing performance degradation caused by communication interference and idle, the reduced finish time and improved network resource utilization can be achieved. With the proposed algorithm, the optimal number of rounds and the most reasonable load allocation ratio on each node could be derived. Finally, simulation results are presented to demonstrate the impacts of different network parameters such as the number of clusters, computation/communication latency, and measurement/communication speed, on the number of rounds, makespan and energy consumption.

Keywords: wireless sensor networks; divisible load theory; multi-round task scheduling.

1 Introduction & Motivation

Owing to the wireless sensor network node with limited energy, the task should be completed within the shortest possible amount of time. Divisible load theory [1] provides an effective solution to wireless sensor networks for task scheduling [2-5]. Different from other heuristic solutions of task scheduling problem in wireless sensor networks [6, 7], this scheme can get not only the optimal solution, but also the analytic solution, thus ensuring the consistency of the results of scheduling.

Divisible load theory has been intensively studied in the past decades. It is mainly concerned with obtaining an optimal partitioning and scheduling strategy for a given task such that it can be processed in the shortest amount of time. Divisible load scheduling algorithm can be divided into single-round scheduling algorithms [2-5] and multi-round scheduling algorithms [8, 9]. Single-round scheduling algorithm is relatively simple, but its computation and communication overlap are rather poor, and the extra overhead is relatively large. Single-round scheduling algorithms were applied to wireless sensor networks in [2-5]. Although the authors derived closed-form solutions to obtain the optimal finish time, the network topology discussed in those papers is single-level tree structure. While in wireless sensor networks, as compared with the single-level tree structure, clustered structure (multi-level tree structure) has a great of advantages [10]. Multi-round scheduling algorithm has the characteristics of better computation and communication overlap, thus properly reducing the scheduling overhead. However, it is more difficult to analyze, so fewer results of multi-round scheduling algorithm are available and the existing multi-round scheduling algorithms are designed based on grid computing environments.

Therefore, we present a multi-round task scheduling algorithm (OTSA-WSN) in clustered wireless sensor networks. The goal of this algorithm is to minimize the overall execution time (hereafter called makespan) and fully utilize network resources, by finding an optimal strategy of splitting the original tasks received by SINK into a number of sub-tasks as well as distributing these sub-tasks to the clusters in the right order.

2 Optimal Scheduling Algorithm

Wireless sensor networks construct clusters several times in its life cycle. Each cluster will have a set-up phase and a steady-state phase[10]. We discuss our multi-round task scheduling algorithm in a steady-phase phase.

The original tasks received by SINK are divided into two stages: inter-cluster task scheduling and intra-cluster task scheduling. First, inter-cluster task scheduling partitions the entire tasks into each cluster, and then the sub-tasks in a cluster is assigned to each intra-cluster sensor node by intra-cluster task scheduling. To improve overlap of communication with computation, inter-cluster task scheduling assigned sensing tasks among all clusters in multiple rounds.

According to divisible load theory, to remove performance degradation caused by communications interference, SINK sends each round's tasks to cluster heads sequentially. After each cluster finishing its tasks and fusing the data, the cluster heads also send this round's results to SINK sequentially. That in every moment only allows SINK node sends sub-tasks to a cluster head, or a cluster head return fusion data to the SINK.

2.1 Intra-cluster task scheduling

In order to ensure that the tasks are processed orderly, SINK allocates the tasks to each cluster according to the task-processing rate of each cluster, which guarantees that the task execution time of all clusters in each round remains the same.

Definition: The task-processing rate of a cluster is the average rate the cluster takes to complete the intra-cluster tasks, that is the number of tasks dealt (measurement and reporting data) per unit of time.

Assuming there are k nodes in a cluster, according to divisible load theory, the cluster's task-processing rate is as follows:

$$S = (1 + \sum_{i=2}^{k} \prod_{j=2}^{i} h_j)/(1/s_1 + 1/b_1) \tag{1}$$

where $h_i = (1/s_{i-1})/(1/s_i + 1/b_i)$, $i = 2, \cdots, k$, where s_i is node i 's measuring rate, in this case, the number of tasks completed per unit of time, b_i states node i 's transmitting rate to cluster head, in this case, the number of tasks transmitted per unit of time.

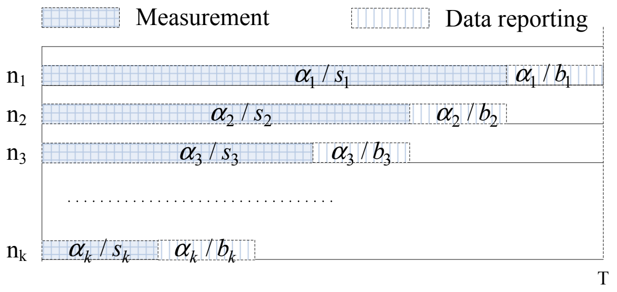


Figure 1: Timing diagram for in-cluster task-processing

Proof: α_i is defined as the fraction of sensing task assigned to node n_i by the cluster head. It is assumed that every node will be assigned non-zero task, i.e., $0 < \alpha_i < 1$, and the task for all nodes in this cluster sums to 1.

By definition we can see:

$$\sum_{i=1}^{k} \alpha_i = 1 \tag{2}$$

So, the time for node n_i measuring its tasks and reporting results to cluster head are α_i/s_i and α_i/b_i, respectively.

One cluster head and a set of sensor nodes constitute a cluster where each node is able to communicate with the cluster head directly. Sensor nodes measure data from surroundings related to the given task, and then report these data to the cluster head. To complete certain amount of sensor readings in minimum finishing time, the sensing task should be allocated to each sensor node and scheduled to avoid transmission conflicts and idle time on the cluster head.

Fig.1 illustrates the timing diagram for a set of sensor nodes, indexed from n_1 to n_k, in one cluster. From Fig.1, it can be observed that there is no time gap between every two successive nodes because the divisible workload can be transferred in the cluster. All sensor nodes start to measure data at the same time. Once the previous node finishes transmitting data, the other one completes its measuring task and starts to report its data. As a result, the proposed timing diagram minimizes the finish time by scheduling the measuring time and reporting time of each senor node. Moreover, since the intra-cluster scheduling tries to avoid the transmission conflicts at the cluster head, energy spent on retransmission are conserved.

The working time of a sensor node can be divided into two parts: measuring time and reporting time.

In Fig.1, one can set up the following corresponding recursive load distribution equations:

$$\alpha_{i-1}/s_{i-1} = \alpha_i/s_i + \alpha_i/b_i, i = 2, 3, ...k \tag{3}$$

Rewriting the above set of equations as:

$$\alpha_i = h_i \alpha_{i-1} \tag{4}$$

where $h_i = (1/s_{i-1})/(1/s_i + 1/b_i)$, $i = 2, 3, ...k$

Using Fig.1, Eq.(2) and Eq.(4), the largest workload for the sensor node can be solved as:

$$\alpha_1 = 1/(1 + \sum_{i=2}^{k} \prod_{j=2}^{i} h_j) \tag{5}$$

Similarly, the workloads of other sensor nodes given by Eq.(4) can be obtained:

$$\alpha_i = \prod_{j=2}^{i}(h_j)/(1+\sum_{i=2}^{k}\prod_{j=2}^{i}h_j) \tag{6}$$

Fig.1 indicates that when the node with the largest measuring data finishes transmission, the local cluster completes its assigned sensing task. Then, the finish time of measuring and reporting data for the cluster is:

$$T = (1/s_1 + 1/b_1)/(1+\sum_{i=2}^{k}\prod_{j=2}^{i}h_j) \tag{7}$$

We can get the task-processing rate of the cluster:

$$S = (1+\sum_{i=2}^{k}\prod_{j=2}^{i}h_j)/(1/s_1 + 1/b_1) \tag{8}$$

It's not difficult to see that in the homogeneous network environment, every cluster have the same parameters, and their task-processing rate is:

$$S = (1-h^k)/(1/s+1/b)(1-h) \tag{9}$$

where $h = (1/s)/(1/s+1/b)$

Theoretical analysis and simulation in this paper are based on LEACH protocol[10] family (inter-cluster one-hop topology). For the multi-hop networks, namely, the multi-layer tree structure, we can do the calculations on parent node in the tree using Eq.(3) Eq.(8). According to divisible load theory, the conflict-free scheduling strategy for each parent node is calculated in order to save energy and prolong network lifetime.

In wireless sensor networks, cluster head is responsible for data exchange for SINK and in-cluster nodes. In order to reduce energy consumption caused by transmitting redundant data, lower latency and prolong the survival period, cluster head needs fuse the data [11]. A new estimation method for data fusion information utilization constant is introduced [5] in this paper. Information utilization constant is based on a technique of information accuracy estimation. Through estimating accuracy of information, cluster head can know the approximate percentage of data fusion.

2.2 Inter-cluster task scheduling

The following notations will be used throughout this paper:

- W_{total}: total amount of workload that resides at SINK;
- W_{ji} :number of tasks assigned to cluster i in round j.
- S_i: rate of cluster i's task-processing, that is, the tasks processed in a unit time.
- B_i: downlink communication speed SINK to cluster head i; B_i': uplink transmission rate cluster head i to SINK, that is, the number of tasks transmitted per unit time.
- t_j: processing time of round j.
- W_j: size of the total load dispatched during round j.

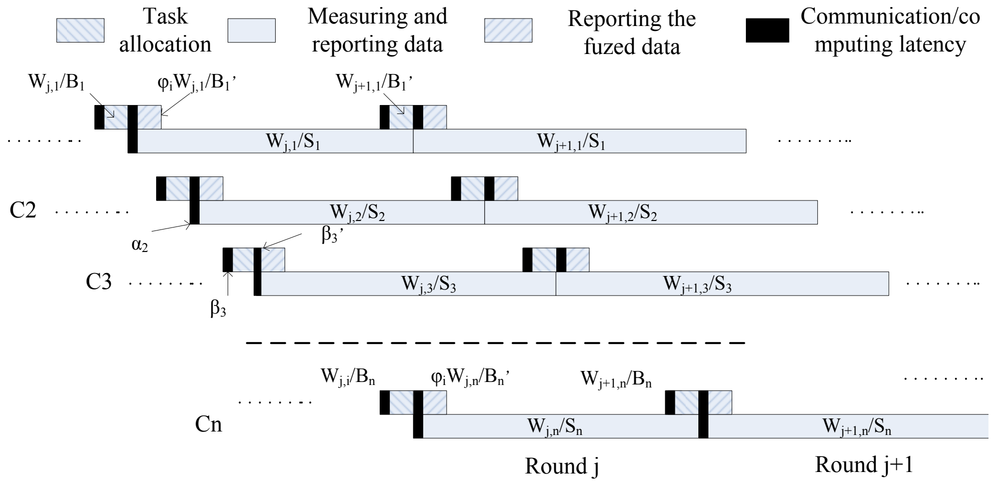


Figure 2: Process of multi-round scheduling algorithm

- φ_i: information utility constant of cluster head i.

Thus, the entire load for cluster i in round j can be sensed (measured, transmitted) is W_{ji}/S_i. In round j, the time for SINK sending tasks to cluster i and for cluster head i sending the fused data are W_{ji}/B_i and $\varphi_i W_{ji}/B_i'$, respectively.

In practical wireless sensor network environment, communication and computing latency caused by pre-initialization are inevitable.

Suppose affine cost parameters are as follows:

- α_i: computing latency of cluster i, that is, the time taken for initialization.
- β_i(resp. $\beta_i{}'$): communication latency incurred by SINK to initiate a data transferring to cluster head i. (resp. start-up time for communication from the cluster head i to SINK).

Fig.2 describes the procedure of SINK dispatching the tasks to each cluster, each cluster measuring and reporting data, as well as cluster heads reporting the fused data to SINK. In this paper, we assume that there are total n clusters in a stable stage, where C_i, $i = 1, \cdot\cdot\cdot, n$ represents each cluster.

As the computational cost of each cluster remains the same, so there are:

$$\alpha_i + W_{ji}/S_i = t_j, j = 1, \cdot\cdot\cdot, M-1 \tag{10}$$

where t_j is only related to the number of rounds j, and M is the optimal scheduling round.

The sum of tasks allocated to every cluster in round j is equal to the tasks in round j:

$$W_j = \sum_{i=1}^{n} W_{ji} \tag{11}$$

From the Eq.(10) and Eq.(11) we can compute:

$$W_{ji} = a_i W_j + b_i \tag{12}$$

where $a_i = S_i / \sum\limits_{k=1}^{n} S_k$, $b_i = (S_i / \sum\limits_{k=1}^{n} S_k) \sum\limits_{k=1}^{n} (S_k \alpha_k) - S_i \alpha_i$.

As shown in Fig.2, in order to fully utilize the bandwidth and avoid cluster waiting among different rounds, SINK must send the tasks allocated in round $j+1$ to all the cluster heads and receive the fused data from all the cluster heads in the round j, before cluster n finished the tasks in round j. When the time for intra-cluster nodes processing the tasks in round j is exactly equal to the sum of the time for SINK sending sub-tasks to all cluster heads in round $j+1$ and receiving the fused data from all the cluster heads in round j, then, the best bandwidth utilization is achieved, that is:

$$\sum_{i=1}^{n}[(W_{j+1,i}/B_i)+\beta_i+(\varphi_i W_{j,i}/B'_i)+\beta_i{}'] = t_j \tag{13}$$

Utilizing Eq.(10), Eq.(12) and Eq.(13), we have:

$$W_{j+1} = W_j * \frac{1-\varphi_i\sum\limits_{i=1}^{n}(S_i/B'_i)}{\sum\limits_{i=1}^{n}(S_i/B_i)} + \frac{\sum\limits_{i=1}^{n}(S_i\alpha_i)}{\sum\limits_{i=1}^{n}S_i\sum\limits_{i=1}^{n}(a_i/B_i)} - \frac{\sum\limits_{i=1}^{n}(\beta_i+b_i/B_i+\beta_i{}'+\varphi_i b_i/B'_i)}{\sum\limits_{i=1}^{n}(a_i/B_i)} \tag{14}$$

Simplify the Eq.(14) as follows:

$$W_j = \theta^j(W_0-\eta)+\eta \tag{15}$$

where $\theta = [1-\varphi_i\sum\limits_{i=1}^{n}(S_i/B'_i)]/\sum\limits_{i=1}^{n}(S_i/B_i)$,

$$\eta = \frac{\sum\limits_{i=1}^{n}(S_i\alpha_i)}{[\sum\limits_{i=1}^{n}(S_i/B_i)+\varphi_i\sum\limits_{i=1}^{n}(S_i/B'_i)]-1} - \frac{\sum\limits_{i=1}^{n}S_i\sum\limits_{i=1}^{n}(\beta_i+b_i/B_i+\beta_i{}'+\varphi_i b_i/B'_i)}{[\sum\limits_{i=1}^{n}(S_i/B_i)+\varphi_i\sum\limits_{i=1}^{n}(S_i/B'_i)]-1}$$

Also the total load is equal to the sum of the tasks allocated in all rounds:

$$\sum_{j=0}^{M-1}W_j = W_{total} \tag{16}$$

The following constraint relations can be obtained:

$$G(M,W_0) = (W_0-\eta)(1-\theta^M)/(1-\theta)+M\eta-W_{total} = 0 \tag{17}$$

The problem of minimizing the total task finish time in scheduling algorithm[8] is described below:

$$EX(M,W_0) = \sum_{j=0}^{M-1}t_j + \frac{1}{2}\sum_{i=1}^{n}[(W_{0,i}/B_i)+\beta_i+(\varphi_i W_{M-1,i}/B'_i)+\beta_i{}'] \tag{18}$$

The above minimization problem can be solved through the Lagrange multiplication.

$$W_0 = (1-\theta)/(1-\theta^M)(W_{total}-M\eta)+\eta \tag{19}$$

After solving W_0 and M, the sizes of all the chunks $W_{j,i}$ can be obtained using Eq.(12) and Eq.(15).

3 Wireless Energy Use

In this section, the energy model of the OTSA-WSN algorithm is presented in detail and the equations of energy consumption of individual sensor nodes are derived. The model is based on first-order radio model [10]. There are three kinds of energy consumption in the wireless sensor network: measurement, data fusion, and communication. Because nodes in the sensor network cooperate with each other via data transmission. Energy consumption of communications exist in sensor nodes, cluster heads and SINK. It is not necessary for cluster heads and SINK to perform any sensing task. Thus, there is no energy cost for cluster heads due to the measurement of these nodes, while the additional energy cost of cluster heads attributes to data fusion. The energy to sense, fuse, and transmit a unit sensory data are denoted by e_s, e_p, and e_{tx}, respectively. Sensor nodes also consume the energy of e_{rx} to receive one unit of data. The distance between the sender and the receiver is d.

The energy use for each kind of nodes is outlined as follows:

Energy use for individual sensor nodes j in cluster i:

$$E_{i,j} = \alpha_{i,j}(e_s + e_{tx}d^2), i = 1, \cdots, k\,, j = 1, \cdots, n_i \tag{20}$$

Energy use for individual cluster head:

$$E_i = \alpha_i(e_{rx} + e_p + \varphi_i e_{tx}d^2), i = 1, \cdots, k \tag{21}$$

Energy use for SINK:

$$E_{SINK} = \sum_{i=1}^{k} \alpha_i \varphi_i e_{tx} \tag{22}$$

4 Performance Evaluation

In the above sections, we have obtained the optimal number of rounds M for a given sensing task, and energy use for individual sensor nodes. In this section, we investigate the effects of three network parameters, such as the number of clusters, computation/communication latency, and measurement/communication speed, on the number of rounds, makespan and energy consumption in the homogeneous network environment.

In the simulation, the following energy parameters are adopted: transmitting a unit of sensor reading over a unit distance takes e_{tx}=200nJ, receiving one unit of sensor reading consumes e_{rx}=150nJ, measuring one unit of sensor reading needs e_s=100nJ, fusing one unit of observation consumes e_p=20nJ and the distance between the sender and the receiver is d=100m. There are 20 sensor nodes in each cluster.

The simulation results are shown in Fig.3 to Fig.6.

Firstly, Fig.3 plots the M values computed by OTSA-WSN versus computation/communication latency when they vary among 0 and 1.0. Assume the communication speeds of the uplink and downlink between SINK and cluster head are identical, namely, β=β'. As can be seen from Fig.3, M decreases with either the communication or computation latency increasing, owing to the reason that fewer rounds may result in less overhead. Because when the communication latency and computation latency increase, the tasks allocated for each round will increase to meet the requirement, which makes full use of bandwidth. Therefore, the number of rounds will be reduced.

Next, the makesapn against the number of clusters are plotted in Fig.4. Assume that transmission rate of all links between nodes is the same, that is, B=$B' = b$. In Fig.4(a), the value of s

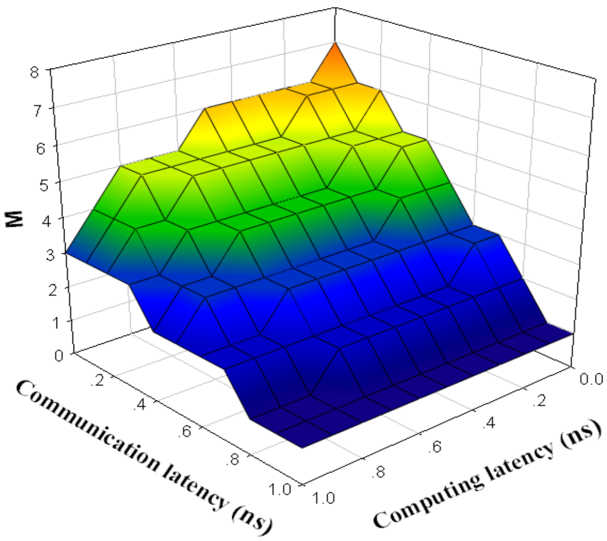
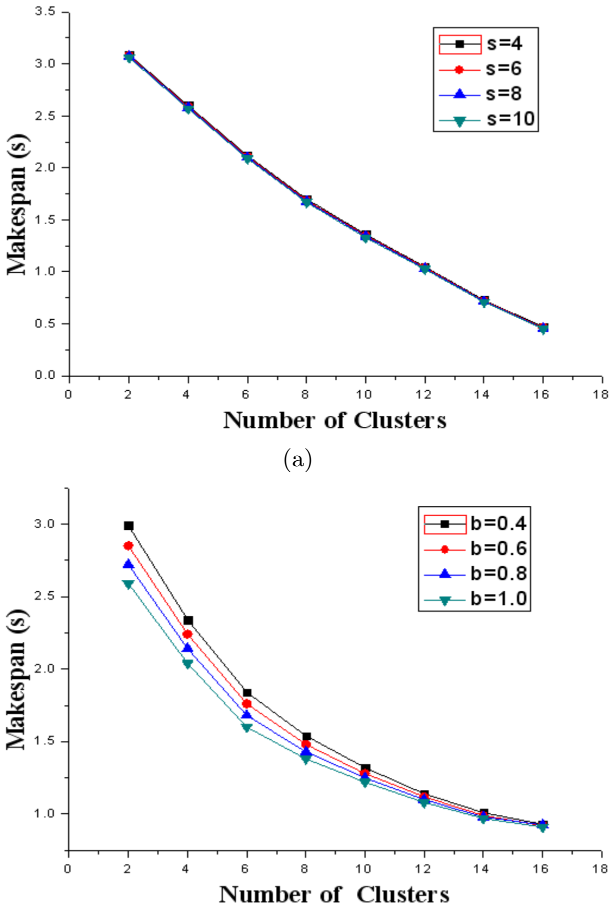


Figure 3: Impact of communications and computing latency on the number of rounds

is chosen from 4 to 10, while *b* is fixed to 1.0. This figure shows that measurement speed almost does not affect the makespan because sensing takes a small fraction of the entire execution time. Fig.4(b) shows that when the communication speed of sensor nodes increases, the makespan of a given task is reduced. It can be found that the four lines in Fig.4(b) converge when the number of clusters becomes large.

Then, the third simulation is about the energy consumption of intra-cluster sensor nodes. SINK and cluster heads are not taken into account because generally, SINK has no energy constraint and the chosen cluster heads have the possibly enough energy. The network is configured with 20 clusters. Without loss of generality, the intra-cluster sensor nodes in the first cluster are chosen to study the energy consumption, as shown in Fig.5. Fig.5 presents the energy consumption of all the nodes in the first cluster as given by Eq.(20), where the intra-cluster nodes are indexed from 1 to 20. In each case, the energy consumption of sensor nodes monotonically decreases due to the reduced workload. Fig.5(a) shows the higher the in-cluster node's measuring speed, the more evenly the tasks allocated to each nodes, hence the smaller the energy consumption of the nodes. Fig.5(b) presents the larger communication speed between nodes, the smaller the energy consumption of the in-cluster nodes.

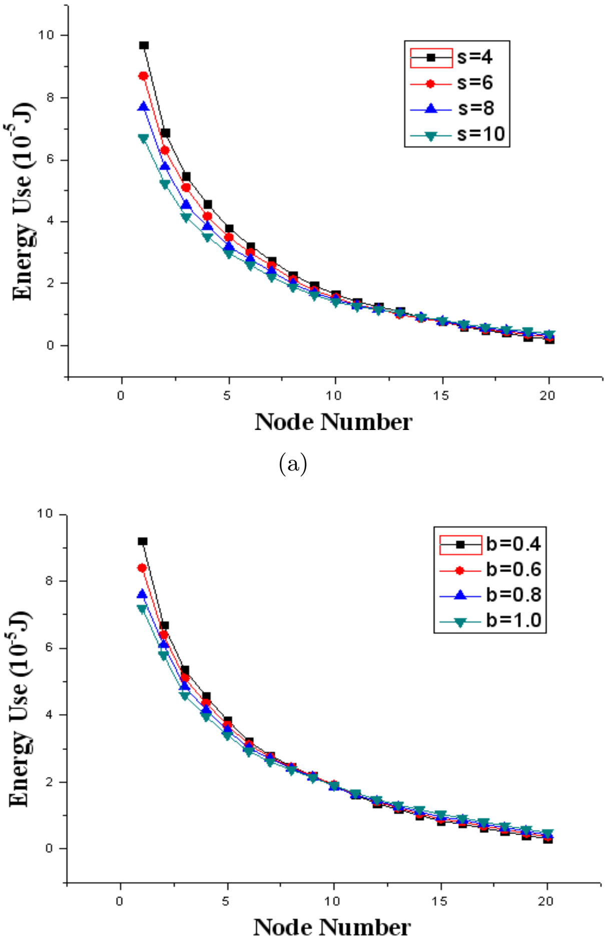
Finally, to simulate the OTSA-WSN algorithm in the heterogeneous network environments, the measuring speed of intra-cluster nodes is set as the random numbers vary from 4 to 10, and the communication speed of links from 0.4 to 1.0. Through 8 experiments, the performance of scheduling algorithms in the heterogeneous network environment is analyzed. Fig.6(a) shows the impact of random measuring speed and communication speed on the makespan with the increasing number of clusters in the heterogeneous network environment. Fig.6(b) presents the impact of random measuring speed and communication speed on energy-consuming in the order of tasks allocation. Compared with Fig.4 and Fig.5, it can be seen that, the impact of measuring speed on the makespan and energy consumption is less than communication speed. The makespan decreases with the increasing number of clusters, and the energy consumption are reduced in the order of tasks allocation.



(a)

(b)

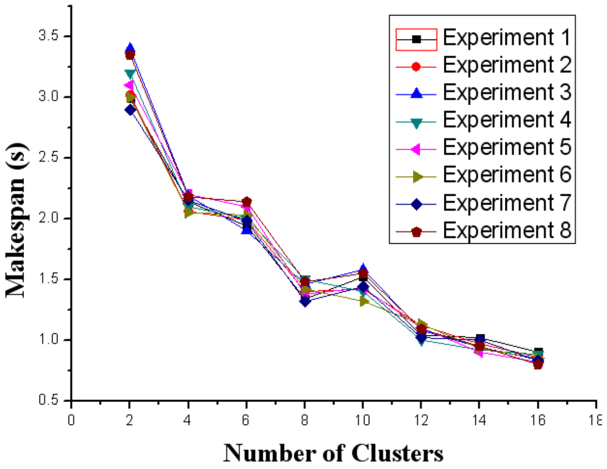
Figure 4: Impact of measuring speed and bandwidth on the makespan



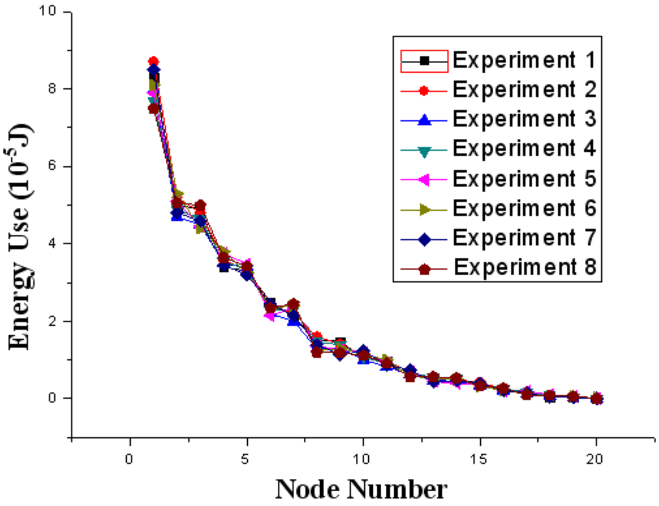
(a)

(b)

Figure 5: The impact of measuring speed and bandwidth on the energy consumption in in-cluster nodes



(a)



(b)

Figure 6: Impact of random measuring speed and bandwidth on the makespan and energy consumption

5 Conclusions

As the wireless sensor network node with limited energy, so the tasks should be completed as quickly as possible, and the network resources should be fully utilized. In this paper, we present a multi-round task scheduling algorithm (OTSA-WSN) in clustered wireless sensor networks. The goal of this algorithm is to minimize the makespan and fully utilize network resources, by finding an optimal strategy of splitting the original load received by SINK into a number of chunks as well as distributing these chunks to the clusters in the right order.

Bibliography

[1] V. Bharadwaj, D. Ghose, T. G.ROBERTAZZI, Divisible Load Theory: A New Paradigm for Load Scheduling in Distributed Systems. *Cluster Computing*, vol.6, No.1, pp.7-18, 2003.

[2] M. Moges, T.G. Robertazzi, Wireless Sensor Networks: Scheduling for Measurement and Data Reporting. *IEEE Transactions on Aerospace and Electronic Systems*, Vol.42, No.1, pp.327-340, 2006.

[3] H. LIU, X. YUAN, M. Moges, An Efficient Task Scheduling Method for Improved Network Delay in Distributed Sensor Networks. *In Proceedings of TridentCom 2007*, Orlando, FL, USA, 1-8, 2007.

[4] H. LIU, J. SHEN, X. YUAN, M. Moges, Performance Analysis of Data Aggregation in Wireless Sensor Mesh Networks, *In Proceedings of Earth & Space 2008*, Akron, OH, USA, 1-8, 2008.

[5] C. Kijeung , T. G. Robertazzi, Divisible Load Scheduling in Wireless Sensor Networks with Information Utility Performance. *In Proceedings of IPCCC 2008*, Austin, Texas, USA, 9-17, 2008.

[6] Z. ZENG, A. LIU, D. LI, A Highly Efficient DAG Task Scheduling Algorithm for Wireless Sensor Networks, *In Proceedings of ICYCS 2008*,Zhang Jia Jie , Hunan , China, 570-575, 2008.

[7] J. LIN, W. XIAO, F. L. Lewis, Energy-Efficient Distributed Adaptive Multisensor Scheduling for Target Tracking in Wireless Sensor Networks. *IEEE Transactions on Instrumentation and Measurement*, Vol.58, No.6, pp.1886 - 1896, 2009.

[8] Y. Yang, R. Van, Der, K, H. Casanova, Multiround Algorithms for Scheduling Divisible Loads. *IEEE Trans on Parallel and Distributed Systems*, Vol.16, No.11, pp.1092-1102, 2005.

[9] C. Yeim-Kuan, W. Jia-Hwa, C. Chi-Yeh, C. Chih-Ping, Improved Methods for Divisible Load Distribution on k-Dimensional Meshes Using Multi-Installment. *IEEE Transactions on Parallel and Distributed Systems*, Vol.18, No.11, pp. 1618-1629,2007.

[10] W. Heinzelman, A. Chandrakasan, An Application-specifid Protocol Architecture for Wireless Microsensor Networks. *IEEE Transaction on Wireless Communications*, Vol.1, No.4, pp. 660-670, 2002.

[11] X. Tang, J. Xu, Optimizing Lifetime for Continuous Data Aggregation With Precision Guarantees in Wireless Sensor Networks. *IEEE/ACM Transactions on Networking*, Vol.16, No.4, pp. 904 - 917,2008.

Int. J. of Computers, Communications & Control, ISSN 1841-9836, E-ISSN 1841-9844
Vol. VI (2011), No. 1 (March), pp. 113-124

EECDA: Energy Efficient Clustering and Data Aggregation Protocol for Heterogeneous Wireless Sensor Networks

D. Kumar, T.C. Aseri, R.B. Patel

Dilip Kumar
Centre for Development of Advanced Computing, Mohali, India
E-mail: dilip.k78@gmail.com

Trilok C. Aseri
PEC University of Technology, Chandigarh, India
E-mail: a_trilok_chand@yahoo.com

R.B. Patel
M.M. University, Mullana, India
E-mail: patel_r_b@yahoo.com

Abstract: In recent years, energy efficiency and data gathering is a major concern in many applications of Wireless Sensor Networks (WSNs). One of the important issues in WSNs is how to save the energy consumption for prolonging the network lifetime. For this purpose, many novel innovative techniques are required to improve the energy efficiency and lifetime of the network. In this paper, we propose a novel Energy Efficient Clustering and Data Aggregation (EECDA) protocol for the heterogeneous WSNs which combines the ideas of energy efficient cluster based routing and data aggregation to achieve a better performance in terms of lifetime and stability. EECDA protocol includes a novel cluster head election technique and a path would be selected with maximum sum of energy residues for data transmission instead of the path with minimum energy consumption. Simulation results show that EECDA balances the energy consumption and prolongs the network lifetime by a factor of 51%, 35% and 10% when compared with Low-Energy Adaptive Clustering Hierarchy (LEACH), Energy Efficient Hierarchical Clustering Algorithm (EEHCA) and Effective Data Gathering Algorithm (EDGA), respectively.

Keywords: clustering; data aggregation; lifetime; heterogeneous wireless sensor networks.

1 Introduction

For past few years, Wireless Sensor Networks (WSNs) attracted lots of researchers because of its potential wide applications and many research challenges. Early study on WSNs mainly focused on technologies based on the homogeneous WSN in which all nodes have same system resources. However, heterogeneous WSN is becoming more and more popular because the benefits of using heterogeneous WSNs with different capabilities in order to meet the demands of various applications have been presented in recent literature [1], [2].

One of the crucial challenges in the organization of the WSNs is energy efficiency and stability because battery capacities of sensor nodes are limited and replacing them are impractical. Since, sensor nodes use a large amount of energy for data transmission and aggregation. Therefore, new energy efficient routing protocols are required to save energy consumption. In this paper, we propose a novel Energy-Efficient Clustering and Data Aggregation (EECDA) protocol for heterogeneous WSN. In this approach, a new Cluster Head (CH) election and data communication mechanism is presented to extend the lifetime and stability of the network. After the

CHs election, a path with maximum sum of residual energy would be selected for data communication instead of the path with minimum energy consumption. Therefore, each CH first aggregates the received data and then transmits the aggregated data to the Base Station (BS). The main contributions of EECDA protocol is to provide longest stability (when the first node is dead) and improves the network lifetime in comparison to Low-Energy Adaptive Clustering Hierarchy (LEACH), Energy-Efficient Hierarchical Clustering Algorithm (EEHCA) and Effective Data Gathering Algorithm (EDGA).

The rest of this paper is organized as follows. Section 2 presents related works. Section 3 describes the EECDA protocol. Section 4 explores on simulation results, and finally paper is concluded in Section 5.

2 Related Work

Many recent research works in the area of cluster-based WSNs have extensively focussed on energy efficiency, lifetime, stability and scalability. In the past few years, numerous clustering algorithms have been proposed for a wide range of applications [3], [4], [5].

Data aggregation and hierarchical mechanism are commonly used in many critical applications of WSNs. It reduces the data redundancy and communication load [6]. LEACH [7] is the first clustering protocol based on single-hop communication model. In LEACH, during the setup phase, each node generates a random number between 0 and 1. If this random number is smaller than the threshold value, $T(s)$, which is given by Equation (1), then the node becomes a CH for the current round. During each round, new CHs are elected and as a result balanced load energy is distributed among the CHs and other nodes of the network.

$$T(s) = \left(\begin{array}{cc} \frac{p_{opt}}{1-p_{opt}\times(r \quad mod \quad \frac{1}{p_{opt}})} & if \quad s \in G \\ \\ 0 & otherwise \end{array} \right) \tag{1}$$

where p_{opt} is the desired percentage of CHs, r is the count of current round, G is the set of sensor nodes that have not been CHs in the last $\frac{1}{p_{opt}}$ rounds. In this paper, we refer round, $\frac{1}{p_{opt}}$, as epoch of the heterogeneous WSN.

Power-Efficient Gathering in Sensor Information Systems (PEGASIS) [8] is a chain-based power efficient protocol based on LEACH. It assumes that each node must know the location of all other nodes. It starts with the farthest node and the chain is constructed by using a greedy algorithm. The chain leader aggregates data and forwards it to the BS. In order to balance the overhead involved in communication between the chain leader and the BS, each node in the chain takes turn to be the leader. In [9], the authors described a heuristic approach to solve the data-gathering problem with aggregation in sensor networks. In this scheme, the data is collected in an efficient manner from all the sensor nodes and transmitted to the BS to maximize the lifetime of the network.

In [10], the authors have studied the impact of heterogeneity of sensor nodes in terms of their energy and proposed a heterogeneous-aware protocol to prolong the time interval before the death of the first node. In [11] a cost-based comparative study between homogeneous and heterogeneous clustered WSNs is proposed to estimate the optimal distribution among different types of sensors, but this result is hard to use if the heterogeneity is due to the operation of the network. In [12], authors have developed energy efficient clustering protocol in WSN which is more suitable for periodical data gathering applications. A survey on many ad hoc and mobile ad hoc network clustering schemes are presented in [13]. In this article authors observed that new clustering schemes are required to handle the topology maintenance and managing node movement in the network. In [14], the authors have proposed a new data gathering approach for single-hop transmission wherein both the data gathering and the aggregation are performed by the same sensor in a cluster but the report to the BS may be done by a different sensor.

In [15], authors have investigated the problem of cluster formation for data fusion by focusing on two aspects: (i) how does one can estimate the number of clusters needed to utilize efficiently data correlation of sensors for a sensor network, and (ii), how does one can pick the CHs to cover the whole network more efficiently. In [16], the authors have analyzed the strengths and weaknesses of many existing clustering algorithms and observed many solutions of appropriate aggregation metrics those have been recently proposed in the literature. Energy-Efficient Protocol with Static Clustering (EEPSC) which partitions the network into static clusters and utilizes CHs to distribute the energy load among high power sensor nodes and extends the network lifetime [17].

A distributed energy saving clustering algorithm called BPEC has been proposed in [18]. In this algorithm, CHs are selected by two probabilities. First is based on the ratio between average residual energy of neighbor nodes and its residual energy and second is the node's degree. By using this algorithm, the entire network broadcasting complexity is $O(n)$, the entire network computing complexity is $O(1)$. The results show that when the network has a higher communication coverage density, analytical and experimental results are very close. Energy-Efficient Hierarchical Clustering Algorithm (EEHCA) [19] has adopted a new method for CH election, which can avoid the frequent election of CHs. A new concept of backup CHs is introduced which improves the performance over LEACH and Hybrid Energy-Efficient Distributed clustering (HEED), in terms of network lifetime. An energy efficient hierarchical data gathering protocol, called EDGA adopts weighted election probabilities of each heterogeneous sensor node to become a CH which better handle heterogeneous energy circumstances [20]. The results demonstrate that EDGA significantly outperforms LEACH and HEED in terms of network lifetime.

The authors in [21] have discussed a new CH election problem based on a set of coverage-aware cost metrics which favor nodes deployed in densely populated network areas. The coverage-aware election of CH nodes, active sensor nodes and routers in clustered WSN increases the lifetime as compared with traditional energy based election methods. In [22], the authors have presented an important corona model to maximize the network lifetime by using maximal transmission range of sensors into different levels. The sensor nodes belong to the same corona have the same transmission range, whereas different coronas have different transmission ranges. In [23] authors have presented a short survey on the main techniques used for energy conservation in WSNs. The main focus is primarily on duty cycle scheme which represents the most suitable technique for energy saving. In [24], the authors reviewed many existing definitions of network lifetime and discussed about the merits and demerits of these definitions.

3 EECDA Protocol

The main goal of EECDA protocol is to maintain efficiently the energy consumption of sensor nodes by involving them in a single-hop communication within a cluster. The data aggregation and fusion technique is used to reduce the number of transmitted messages to the BS to save the energy and prevent the congestion. To make the protocol implementation, we have adopted a few reasonable assumptions as follows: (i) n sensor nodes are uniformly dispersed within a square field; (ii) All sensor nodes and the BS are stationary after deployment; (iii) The WSN consists of heterogeneous nodes in terms of node energy; (iv) CHs perform data aggregation; (v) The BS is not energy limited in comparison with the energy of other nodes in the network.

We use the same radio model defined in [7]. The amount of energy required to transmit a L bit packet over a distance, d, is given by Equation (2).

$$E_{Tx}(L,d) = \begin{pmatrix} L \times E_{elec} + L \times \epsilon_{fs} \times d^2 & if \;\; d <= d_0 \\ \\ L \times E_{elec} + L \times \epsilon_{mp} \times d^4 & if \;\; d >= d_0 \end{pmatrix} \tag{2}$$

E_{elec} is the energy being dissipated to run the transmitter or receiver circuitry. The parameters ϵ_{mp} and ϵ_{fs} is the amount of energy dissipates per bit in the radio frequency amplifier according to the distance d_0, which is given by Equation (3).

$$d_0 = \sqrt{\frac{\epsilon_{fs}}{\epsilon_{mp}}} \tag{3}$$

The amount of energy required to receive a packet is given by Equation (4).

$$E_{Rx}(L) = L \times E_{elec} \tag{4}$$

3.1 Impacts of heterogeneity on network performance

By placing few heterogeneous nodes in the network can bring three main benefits: (i) Extending network lifetime: the average energy consumption for forwarding a packet from the heterogeneous nodes to a BS will be much less than the energy consumed in homogeneous sensor networks, (ii) Improving reliability of data communication: the heterogeneous sensor network can get much higher end-to-end delivery rate than the homogeneous sensor network and (iii) Decreasing latency of data transmission: the heterogeneous nodes can decrease the forwarding latency by using fewer hops to the BS.

3.2 Optimal number of clusters

The optimal probability of a node to become a CH is very important in WSNs. This clustering is optimal in the sense that energy consumption is well distributed among all the sensor nodes and the total energy consumption should be minimum. Such optimal clustering highly depends on the energy model. For EECDA, we have used similar energy model as discussed in [7]. Let us assume an area $A = M \times M$ square meters over which n nodes are uniformly distributed. For simplicity, assume the BS is located in the center of the field, and the distance of any node to the BS or its CH is d_0. Therefore, the energy dissipated by the CH node during a round is given by the Equation (5).

$$E_{CH} = (\frac{n}{k}) \times L \times (E_{elec} + E_{DA}) + L \times \epsilon_{fs} \times d_{BS}^2 \tag{5}$$

where k is the number of clusters, E_{DA} is the data aggregation and d_{BS} is the average distance between a CH and the BS which is given by Equation (6).

$$d_{BS}^2 = \int \sqrt{(x^2 + y^2)} \times \frac{1}{A} = 0.765 \times \frac{M}{2} \tag{6}$$

The energy dissipated by a non-CH node is given by Equation (7).

$$E_{NCH} = L \times (E_{elec} + \epsilon_{fs} \times d_{CH}^2) \tag{7}$$

where d_{CH} is the average distance between a non-CH node and its associated CH, which is given by Equation (8) [10].

$$d_{CH}^2 = \int \int (x^2 + y^2) \times \rho(x,y) dx dy = \frac{M^2}{2\pi k} \tag{8}$$

where $\rho(x, y)$ is the node distribution and M is the area of monitoring field. The total energy dissipated in a cluster per round is given by Equation (9).

$$E_T = E_{CH} + E_{NCH} \tag{9}$$

By substituting Equation (5) and Equation (7) in Equation (9), we obtain energy dissipating during a round which is given by Equation (10).

$$E_T = L \times (2 \times n \times E_{elec} + n \times E_{DA} + \epsilon_{fs} \times (k \times d_{BS}^2 + n \times \frac{M^2}{2\pi k}) \tag{10}$$

By setting the derivative E_T with respect to k to zero, we derive the optimal number of clusters which is given by Equation (11).

$$k_{opt} = \sqrt{\frac{n}{2\pi}} \times \sqrt{\frac{\epsilon_{fs}}{\epsilon_{mp}}} \times \frac{M}{d_{BS}^2} \tag{11}$$

Using Equation (6) and Equation (11), the optimal probability of a node to become a CH, p_{opt}, can be computed by Equation (12)

$$p_{opt} = \frac{1}{0.765} \times \sqrt{\frac{2}{n\pi}} \times \sqrt{\frac{\epsilon_{fs}}{\epsilon_{mp}}} \tag{12}$$

If the clusters are not constructed in an optimal way, the total energy dissipated per round is increased exponentially either when the number of clusters is greater or less than the optimal value.

3.3 CH election phase

EECDA considers three types of nodes (i.e., normal, advanced and super) which have deployed in a harsh wireless environment where battery replacement is impossible. Nodes with higher battery energy are advanced and super nodes and the remaining nodes are normal nodes. The main aim of EECDA is to increase the energy efficiency, lifetime and stability of the network in the presence of heterogeneous nodes. Let m be the fraction of advanced nodes among the normal nodes and (m_o) be the proportion of super nodes among the advanced nodes. Let us assume the initial energy of each normal node is E_0. The initial energy of each advanced and super node is $E_0 \times (1 + \alpha)$ and $E_0 \times (1 + \beta)$, where both α and β means the advanced and super node have α and β times more energy than the normal node. Intuitively, advanced and super nodes have to become CHs more often than the normal nodes, which is equivalent to a fairness constraint on energy consumption. The new heterogeneous setting has no affect on the spatial density of the network so the priori setting of, p_{opt}, does not change but the total energy of the network will be changed. The total initial energy of the new heterogeneous network setting is given by Equation (13).

$$n \times E_0 \times \{(1-m) + m \times ((1-m_o) \times (1+\alpha) + m_o \times (1+\beta)\}) = n \times E_0 \times (1 + m \times (\alpha - m_o \times (\alpha - \beta))) \tag{13}$$

The first improvement to the existing protocols is to increase the epoch of the sensor network in proportion to the energy increment. In order to optimize the stable region of the system, the new epoch must become equal to $(\frac{1}{p_{opt}}) \times (1 + m \times (\alpha - m_o \times (\alpha - \beta)))$ because the system has $m \times (\alpha - m_o \times (\alpha - \beta))$ times more energy due to heterogeneous nodes.

If we set the same threshold value for super, advanced and normal nodes with the difference that each normal node ϵG becomes a CH once every $(\frac{1}{p_{opt}}) \times (1 + m \times (\alpha - m_o \times (\alpha - \beta)))$ rounds per epoch, and each advanced and super node ϵG becomes a CH $(1 + \alpha)$ and $(1 + \beta)$ times every $(\frac{1}{p_{opt}}) \times (1 + m \times (\alpha - m_o \times (\alpha - \beta)))$ rounds per epoch, then there is no guarantee that the number of CHs per round per epoch will be $p_{opt} \times n$. This problem can be overcome by modifying the threshold Equation (1).

In EECDA, we assign a weight to the optimal probability p_{opt}. This weight must be equal to the initial energy of each node divided by the initial energy of the normal node. Let us define p_n, p_a, and p_s are the weighted election probabilities for normal, advanced and super nodes. Virtually there are $n \times (1 + m \times (\alpha - m_o \times (\alpha - \beta)))$ nodes with energy equal to the initial energy of a normal node. In order to maintain the minimum energy consumption in each round within an epoch, the average number of CHs per round per epoch must be constant and equal to $p_{opt} \times n$. In the heterogeneous scenario, the average number of CHs per round per epoch is equal to $(1 + m \times (\alpha - m_o \times (\alpha - \beta))) \times n \times p_n$ because each virual node has the initial energy of a normal node. Therefore, the weighed probabilities for normal, advanced and super nodes are respectively given by Equations (14-16).

$$p_n = \frac{p_{opt}}{(1 + m \times (\alpha - m_o \times (\alpha - \beta))} \tag{14}$$

$$p_a = \frac{p_{opt}}{(1 + m \times (\alpha - m_o \times (\alpha - \beta))} \times (1 + \alpha) \tag{15}$$

$$p_s = \frac{p_{opt}}{(1 + m \times (\alpha - m_o \times (\alpha - \beta))} \times (1 + \beta) \tag{16}$$

By substituting Equation (14) in Equation (1) and a new threshold is derived for normal nodes which is given by Equation (17).

$$T(s_n) = \begin{pmatrix} \frac{p_n}{1 - p_n \times (r \quad mod \quad \frac{1}{p_n})} & if \quad s_n \ \epsilon \ G' \\ & \\ 0 & otherwise \end{pmatrix} \tag{17}$$

where r is the current round, G' is the set of normal nodes that have not become CHs within the last, $\frac{1}{p_n}$, rounds of the epoch and $T(sn)$ is the threshold applied to a population of $n \times (1 - m)$ normal nodes. This guarantees that each normal node will become a CH exactly once every $(\frac{1}{p_{opt}}) \times (1 + m \times (\alpha - m_o \times (\alpha - \beta)))$ rounds per epoch, and that the average number of CHs of normal nodes per round per epoch is equal to $(n \times (1 - m) \times p_n)$. Similarly, new thresholds for advanced and super nodes can be derived by substituting Equation (15) and (16) into Equation (1), which are given by Equation (18) and Equation (19).

$$T(s_a) = \begin{pmatrix} \frac{p_a}{1 - p_a \times (r \quad mod \quad \frac{1}{p_a})} & if \quad s_a \ \epsilon \ G'' \\ & \\ 0 & otherwise \end{pmatrix} \tag{18}$$

$$T(s_s) = \begin{pmatrix} \frac{p_s}{1 - p_s \times (r \quad mod \quad \frac{1}{p_s})} & if \quad s_s \ \epsilon \ G''' \\ & \\ 0 & otherwise \end{pmatrix} \tag{19}$$

Route selection phase

Once all CHs are elected in a specific round by using weighted election probability, each CH first estimates its energy residue $E_{(CHR)_s}$ and broadcast this information with its CH role to the neighboring nodes. The value of $E_{(CHR)_s}$ can be calculated by Equation (20).

$$E_{(CHR)_s} = (E_{(CHrem)_s} - (E_{(BS)_s}) \qquad s \ \epsilon \ G_c \tag{20}$$

where G_c is the set of elected CHs per round. $(E_{(CHrem)_s}$ indicates the remaining energy of CH_s in current round and $(E_{(BS)_s})$ indicates the communication energy dissipated from CH_s to the BS. Then, each CH records the value of $(E_{(CHR)_s})$ in advertisement message and broadcasts advertisement message to the rest of the nodes in the WSN. During the CH election phase, each non-CH node receives all advertisement messages, and extracts all of energy residue data of CH_s from advertisement messages.

Moreover, each non-CH node also calculates energy residues $(E_{(NCHR)_i})$ to every CH respectively which is given by Equation (21).

$$E_{(NCHR)_i} = (E_{(NCHrem)_i} - (E_{(CH)_{is}}) \qquad i \ \epsilon \ G_n \tag{21}$$

where G_n is the set of non-CH nodes. $(E_{(NCHrem)_i})$ indicates the residual energy of non-CH node i in the current round and $(E_{(CH)_{is}})$ indicates the communication energy from non-CH node i to CH node s. Finally, each non-CH node would associate one of the existing CH according to maximum energy residue which is given by Equation (22). Therfore, a path with a maximum sum of energy residues would be selected for data transmission in spite of that path with minimum energy consumption.

$$Max\{E_{(CHR)_s} + E_{(NCHR)_i}\} \qquad s \ \epsilon \ G_c, i \ \epsilon \ G_n \tag{22}$$

Data communication phase

In data communication phase, each non-CH node transmits its data to the associated CH. Each CH will receive all sensed data from its associated non-CH nodes and sends it to the BS.

4 Simulation Results and Discussion

To evaluate and compare the performance of EECDA with EEHCA, EDGA and LEACH in the heterogeneous WSN, we have conducted simulations for two scenarios: first, a network with *100* nodes deployed over an area of size *100* × *100* square meter, and second, a network with *200* nodes deployed over an area of size *200* × *200* square meter as shown in Figure 1 and Figure 2, we denote a normal node with (o), an advanced node with (+), a super node with (*) and the BS with (x). The simulation parameters are summarized in Table 1.

The performance metrics used for these protocols are: (i) Network Lifetime: this is the time interval from the start of the operation until the first and last node dies; (ii) Stability Period: this is the time interval from the start of the operation until the death of the first alive node; (iii) Instability Period: this is the time interval from the death of the first alive node until the death of the last alive node and (iv) Number of alive nodes per round: this is the instantaneous measure reflects that the total number of alive nodes per round that have not yet expended all of their energy.

Figure 3 and Figure 4 show that both LEACH and EEHCA fails to take the full advantage of heterogeneity in both the scenarios where the first and last node dies earlier as compared to EDGA and EECDA. Therefore, EECDA prolongs the network lifetime by 51% , 35% and 10%

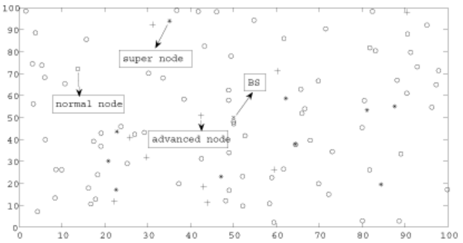


Figure 1: Random deployment of 100 nodes over an area 100 × 100 m^2.

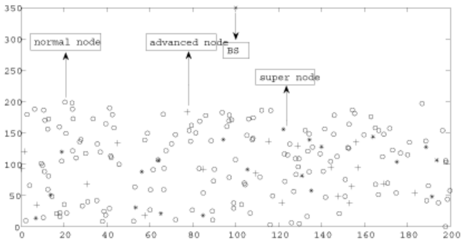


Figure 2: Random deployment of 200 nodes over an area 200 × 200 m^2.

Table 1: Simulation parameters

Parameter	Scenario I and II
Network area	$100 \times 100m^2$, $200 \times 200m^2$
BS location	$(50, 50), (100, 350)$
n	$100, 200$
E_{DA}	$5nJ/bit/report$
Packet size	$50bytes$
ϵ_{mp}	$0.0013pJ/bit/m^4$
ϵ_{fs}	$10pJ/bit/m^2$
E_{elec}	$50nJ/bit$

when compared with LEACH, EEHCA and EDGA protocols, respectively. Figures 3, 4, 5 and 6 present that the unstable region for EECDA is shorter than that of LEACH, EEHCA and EDGA because the normal nodes die in both the scenarios very fast in case of LEACH, EEHCA and EDGA that result in the sensing field it become sparse very fast. On the other hand, advanced and super nodes die in a very slow fashion, because they are not selected as CHs very often after the death of the normal nodes, which again affects the election process of CHs and makes the network unstable. It is quite evident that the stable region of EECDA is extended as compared with LEACH, EEHCA and EDGA for both the scenarios. Figure 5 and Figure 6 indicate that the number of alive nodes are more per round in case of EECDA as compared with EDGA, EEHCA and LEACH because a path with a maximum sum of energy residual would be selected for data transmission in spite of that path with minimum energy in case of EECDA.

Figures 7, 8, 9, 10, 11 and 12 illustrate the performance of residual energy of normal, advanced and super nodes under the heterogeneous settings of EECDA, EDGA, EEHCA and LEACH. Initially, EECDA has the same initial energy as EDGA, LEACH and EEHCA, but gradually it decreases in EDGA, EEHCA and LEACH over rounds. So, EDGA, EEHCA and LEACH have less residual energy left after certain rounds for both the scenarios. Therefore, more the residual energy more efficient is the system.

5 Conclusion

Most existing research considers homogeneous sensor networks. However, a homogeneous sensor network suffers from poor performance and scalability. In this paper, we have developed

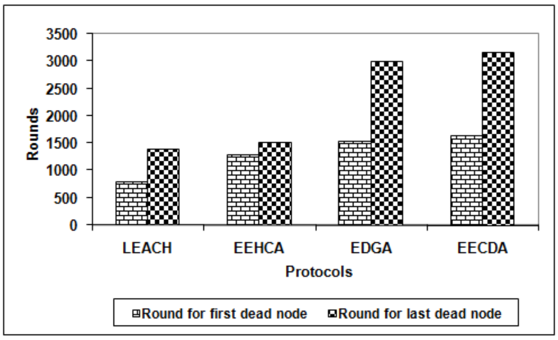


Figure 3: Network lifetime as a function of first and last dead nodes over an area 100 × 100 m^2.

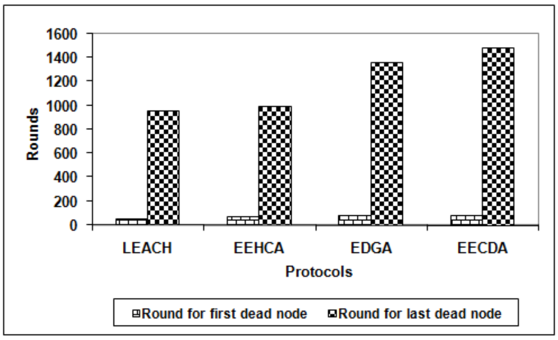


Figure 4: Network lifetime as a function of first and last dead nodes over an area 200 × 200 m^2.

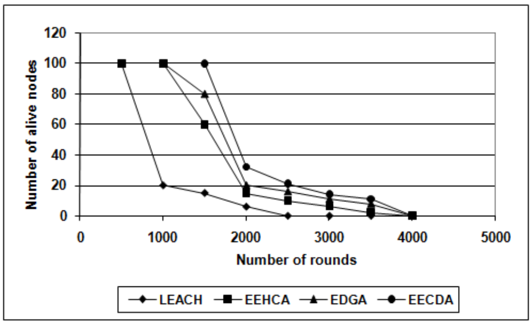


Figure 5: Stability as a function of number of alive nodes per round over an area 100×100 m^2.

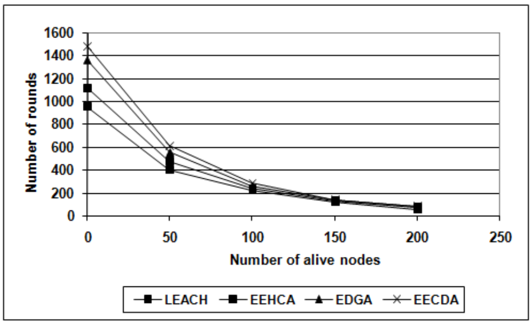


Figure 6: Stability as a function of number of alive nodes per round over an area 200×200 m^2.

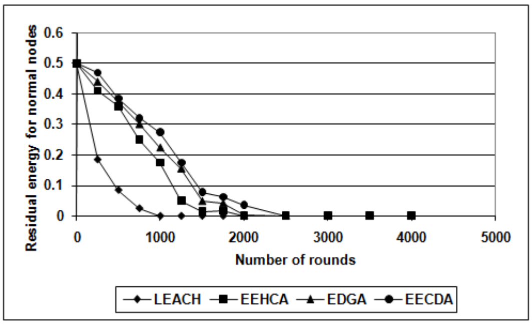


Figure 7: Residual energy of normal nodes per round over an area 100 × 100 m^2.

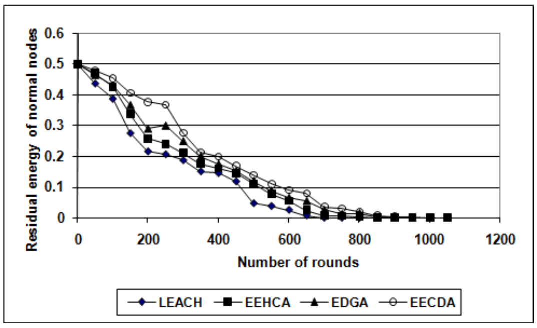


Figure 8: Residual energy of normal nodes per round over an area 200 × 200 m^2.

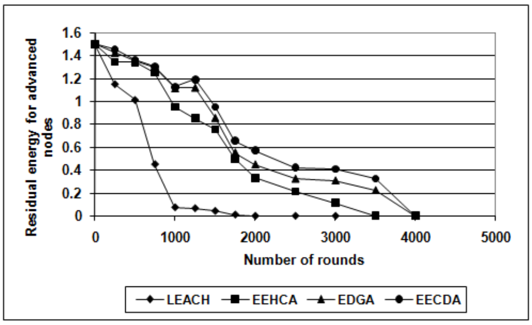


Figure 9: Residual energy of advanced nodes per round overan area 100 × 100 m^2.

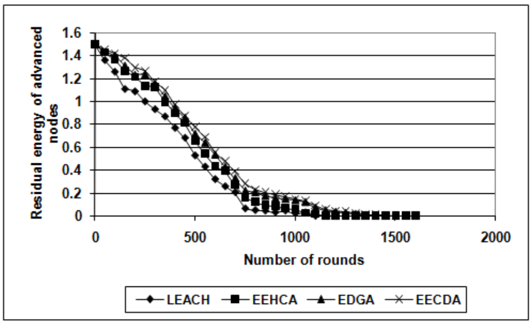


Figure 10: Residual energy of advanced nodes per round over an area 200 × 200 m^2.

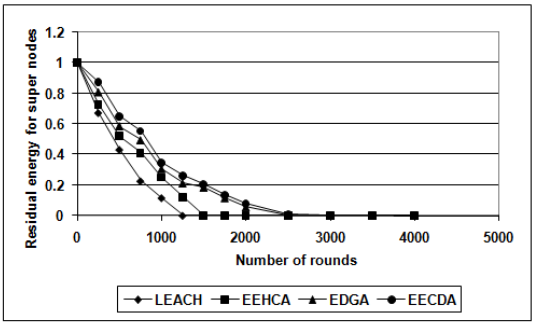


Figure 11: Residual energy of super nodes per round overan area $100 \times 100 \ m^2$.

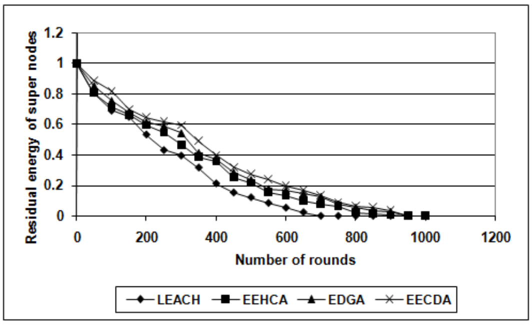


Figure 12: Residual energy of super nodes per round over an area $200 \times 200 \ m^2$.

a novel Energy Efficient Clustering and Data Aggregation (EECDA) protocol to improve the network performance by using some heterogeneous nodes in the network. A novel cluster head election technique and a path with maximum sum of energy residual for data transmission can maintain the balance of energy consumption in the network. Simulation results show that EECDA has better network lifetime, stability and energy efficiency when compared with EDGA, EEHCA and LEACH protocols. The future work includes more levels of hierarchy with some mobility in the network.

Bibliography

[1] R Kumar, V. Tsiatsis, M.B. Srivastava, Computation Hierarchy for In-Network Processing, *Proceedings of 2nd ACM International Workshop on Wireless Networks and Applications, San Diego, CA*, 68-77, 2003.

[2] D. Kumar, T. C. Aseri, R.B. Patel, HCEE: Hierarchical clustered energy efficient protocol for heterogeneous wireless sensor networks, *International Journal of Electronics Engineering*, 1(1):123-126, 2009.

[3] J. Yu, P. Chong, A Survey of Clustering Schemes for Mobile Ad Hoc Networks, *IEEE Communication Surveys*,7(1): 32-48, 2005.

[4] N. Vlajic, D. Xia, Wireless Sensor Networks: To Cluster or Not to Cluster?, *Proceedings of International Symposium on a Wireless of Wireless , Mobileand Multimedia Networks (WoWMoM'06)*, 259-268, 2006.

[5] W.S. Jang, W.M. Heley, M.J. Skibniewsk, Wireless Sensor Networks as a Part of Web Based Building Environment Monitoring System, *Automation in Construction Journal*, 17: 729-736, 2008.

[6] B. Krishnamachari, D. Estrin, S.B. Wicker, The Impact of Data Aggregation in Wireless Sensor Networks, *Proceedings of 22nd International Conference on Distributed Computing Systems Workshops (ICDCSW'02)*, 575-578, 2002.

[7] W.B. Heinzelman, A.P. Chandrakasan, H. Balakrishnan, An Application-Specific Protocol Architecture for Wireless Microsensor Networks, *IEEE Transactions on Wireless Communications*, 1(4):660-669, 2002.

[8] S. Lindesy, C. Raghavendra, PEGASIS: Power-Efficient Gathering in Sensor Information System, *Proceedings of IEEE Aerospace Conference*, 1-6, 2002.

[9] K. Dasgupta, K. Kalpakis, P. Namjoshi, An Efficient Clustering-Based Heuristic for Data Gathering and Aggregation in Sensor Networks, *Proceedings of Wireless Communication and Networking (WCNC'03)*, 3: 1948-1953, 2003.

[10] G. Smaragdakis, I. Matta, A. Bestavros, SEP: A Stable Election Protocol For Clustered Heterogeneous Wireless Sensor Networks, *Proceedings of 2nd International Workshop on Sensor and Actor Network Protocols and Applications (SANPA'04),Boston, MA*, 660-670, 2004.

[11] V. Mhatre, C. Rosenberg, Homogeneous vs. Heterogeneous Clustered Sensor Networks: A Comparative Study, *Proceedings of IEEE International Conference on Communications (ICC'04)*, 2004.

[12] M. Ye, C. Li, G. Chen, J. Wu, EECS: An Energy Efficient Clustering Scheme in Wireless Sensor Networks, *Proceedings of 24th IEEE International Performance, Computing and Communications Conference (IPCCC'05)*, 535- 540, 2005.

[13] D. Wei, H. A. Chan, Clustering Ad Hoc Networks: Schemes and Classifications, *Proceedings of 3rd Annual IEEE Communication Society on Sensors and Ad hoc Communication and Networks (SECON'06),*, 3: 920-926, 2006.

[14] H. Yuning, Z. Yongbing, J. Yusheng, S. Xuemin, A New Energy Efficient Approach by Separating Data Collection and Data Report in Wireless Sensor Networks, *Proceedings of International Conference on Communication and Mobile Computing (IWCMC'06)*, 1165-1170, 2006.

[15] H. Chen , S. Megerian, Cluster Sizing and Head Selection for Efficient Data Aggregation and Routing in Sensor Networks, *Proceedings of Wireless Communication and Networking (WCNC'06)*, 4: 2318-2323,2006.

[16] S. Xun, A Combinatorial Algorithmic Approach To Energy Efficient Information Collection In Wireless Sensor Networks, *ACM Transactions on Sensor Networks*, 3(1), 2007.

[17] A. S. Zahmati, B. Abolhassani, A. A. B. Shirazi, A. S. Bakhtiari, An Energy-Efficient Protocol with Static Clustering for Wireless Sensor Networks, *International Journal of Electronics, Circuits and Systems*, 3(2): 135-138, 2007.

[18] X. Jianbo, H. Yong, L. I. Renfa, An Energy-Aware Distributed Clustering Algorithm in Wireless Sensor Networks, *Proceedings of International Conference on Computer Science and Software Engineering*, 528-531, 2008.

[19] G. Xin, W.H. Yang, D. DeGang, EEHCA: An Energy-Efficient clustering Algorithm for Wireless Sensor Networks, *Information Technology Journal*, 7(2):245-252, 2008.

[20] Y. Mao, Z. Liu, L. Zhang, X. Li, An Effective Data Gathering Scheme in Heterogeneous Energy Wireless Sensor Networks, *Proceedings of International Conference on Computational Science and Engineering* , 338-343, 2009.

[21] A. Bari, A. Jaekel, S. Bandyopadhyay, Clustering Strategies for Improving the Lifetime of Two-Tiered Sensor Networks, *Elsevier, Computer Communication Journal*, 31(14): 3451-3459, 2008.

[22] C. Song, M. Liu, J. Cao, Y. Zheng et. al, Maximizing Network Lifetime Based on Transmission Range Adjustment in Wireless Sensor Networks, *Elsevier, Computer Communication Journal* , 32(11): 1316-1325, July 2009.

[23] H.Y. Shiue, J.X Lieo-hong, S. Horijuchi, Energy Saving in Wireless Sensor Networks, *Journal of Communication and Computing* , 6(5):20-28, 2009.

[24] I. Dietrich, F. Dressler, On the Lifetime of Wireless Sensor Networks, *ACM Transactions on Sensor Networks* , 5(1): 5:1-5:39, 2009.

Int. J. of Computers, Communications & Control, ISSN 1841-9836, E-ISSN 1841-9844
Vol. VI (2011), No. 1 (March), pp. 125-133

Ontology Model of a Robotics Agents Community

H. Latorre, K. Harispe, R. Salinas, G. Lefranc

Homero Latorre Karina Harispe
Universidad Tecnológica Metropolitana de Chile
Departamento de Informática y Computación
José Pedro Alessandri 1242, Ñuñoa. Santiago – Chile
E-mail: helatorre@gmail.com

Renato Salinas
Universidad de Santiago de Chile
Departamento de Ingenieria Mecánica
Avda. Bernardo O'Higgins 3363. Estación Central. Santiago – Chile
E-mail: renato.salinas@usach.cl

Gastón Lefranc
Pontificia Universidad Católica de Valparaíso
Escuela de Ingenieria Eléctrica
Avda. Brasil 2147. Valparaiso – Chile
E-mail: glefranc@ucv.cl

Abstract: This paper presents an Ontology Model of necessities and decisions for a cooperative community of heterogeneous robotic agents. Based on an ant community, it defines the characteristics of a collaborative and cooperative community of robots, using multi agent theory where each robot shares information with others robots in the community, to accomplish a common objective.

Keywords: Robotics, Intelligent Agents, Multi agents, Collaborative and Cooperative Robotics, Long-range-dependent, network layer, network traffic, self-similar process.

1 Introduction

Collaborative and Cooperative tasks are used in robotic agents, where the knowledge is shared; there exist communication to talk to the others robots where the area to the tasks is located and which decision that they have make, according to the necessities. These works have the hierarchy characteristics similar to that presented in nature, especially in the animal kingdom, where the most interesting are the behavior of bees, ants and termites, communicating where are the foods and the decisions that they have to take, according to the necessities. Based on this group behavior, these insects are intelligent agents designed by nature.

The agents can be simulated in a computer and put in group of robots that will be able to do tasks in coordinating way, where the decisions are taken with the group characteristics. The tasks in a community of collaborative and cooperative robots have increased in complex way, meaning the tasks cannot be done with only one robot, but needing a heterogeneous community of robots.

This heterogeneity of collective robots produces new challenges, adds a new difficulty to collective robotics, such as the aptitude to coordinate individuals with multiple qualities of an intelligent way; and to solve a specific goal working together and taking into account all the characteristics. To solve a certain task, each of them fulfills a specific function with which the rest must be able to coexist, the Satisfactory results by means of the union of their qualities, and work together [1].

There exist several methodologies for building multi agent systems. Few of them address the information domain of the system. It is as important as the representation of the system in the information domain is the various agents' information domain view. Heterogeneous systems can contain agents with differing data models, a case that can occur when reusing previously built agents or integrating legacy components into the system. Most existing methodologies lack specific guidance on the development of the information domain specification for a multi agent system and for the agents in the system. An appropriate methodology for developing ontology must be defined for designers to use for specifying domain representations in multi agent systems. The existing methodologies for designing domain ontologies are built to describe everything about a specific domain; however, this is not appropriate for multi agent systems because the system ontology should only specify the information required for proper system execution. The system ontology acts as a prerequisite for future reuse of the system, as the ontology specifies the view of the information domain used by the multi agent system. Any system that reuses the developed multi agent system must ensure that the previously developed system ontology does not conflict with the ontology being used in the new system [2]. Once the system ontology is constructed, a multi agent system design methodology should allow the analyst to specify objects from the data model as parameters in the conversations between the agents. To ensure the proper functionality of the multi agent system, the designer must be able to verify that the agents have the necessary information required for system execution. Since the information is represented in the classes of the data model, the design of the methodology must show the classes passed between agents [3,4].

In this paper is presented an Ontology Model of necessities and decisions for a cooperative community of heterogeneous robotic agents based on an ant community. It defines the characteristics of collaborative and cooperative community of robots, using multi agent theory where each robot shares information with others robots in the community, to accomplish a common objective.

2 Description of the problem

In general, the complexity of the tasks added to the lack of constant structure in the environment, does not allow a homogeneous and structured community should be capable to fulfill the aims raised of an ideal way. Nevertheless, these limitations of capacities can be covered if different technologies are integrated; applying the heterogeneity in the architecture added to intelligent behaviors handled by means of Multi Agent Systems – MAS.

It is important to create MAS capable of controlling a heterogeneous community of robots in order to exhibit cooperation and collaboration among them, reaching the proposed aim. There are two concepts that have to be clarified, how the members of the community must be interrelated to the moment to make the work; and cooperation and collaboration.

Cooperation and collaboration is not the same thing. To collaborate is to contribute, is to give help (in knowledge or work) to do something, is to share the same proposition and the common goals. To cooperate is a collective work with a common goals, it means working together simultaneously, to have interactions of the collective work and interchange of ideas [5].

The cooperative and collaborative environments seek to transform heterogeneous groups into intelligent, flexible and autonomous communities. The community does not need having subordination; this means that the community has no teachers and apprentices nor masters and slaves; the agency is a mere way of communication towards the exterior, to send of results or requests of support [6, 7].

In some communities, the robots communicate to a robot agency, to send reports outside of the community. In a heterogeneous community of robots, the communication between the robots must be in an egalitarian form, without intermediaries. Each of them has to have the capacity of direct connection with its partners, without establishing first the communication with the agency.

Briefly, it is important to have a MAS capable of provoking cooperation and collaboration, generating intelligence, flexibility and autonomy, without subordination, to arrive to results obtained from actions realized as consequence of consensus decisions [1, 8–10].

3 Description of the organization of the community

The community is formed by heterogeneous robots; each one is specializing in specific functions, which it helps in the achievement of the common goals.

Inside the community, every robot must report its characteristics to the rest of the community, in order to generate knowledge that it allows to determine which of its qualities are adapted to a specific works. Every robot has responsibilities, which will be resolved by consensus of the community, and it will have to realize its actions without being an obstacle to its companions, and giving any help that is necessary in unforeseen moments.

To realize the work, and unlike the existing robotic communities, a leader does not exist hence, the communication, transfer of information, and capture of decisions is of "all with all", where every contribution has the same value for the community.

If we base the study on a community of ants, these have behaviors that lead them to achieving its goals, though an ant alone is not capable of feeding and defending its anthill, millions of them can do it. In turn, a robot is not capable of realizing tasks that need major capacities that those that it possesses, but a group of them that brings together as a whole all the characteristics necessary, it can realize it.

Considering a group of agents, natural or artificial, which must come to an aim it can say that: (a) The ants (natural agents) possess intrinsic behaviors of communication, work and capture of decisions. (b) The robots (artificial agents) possess behavior learned or incorporated by means of programming which is based on predefined ontologies.

To manage to obtain a "natural behavior" in a robotic community modularization of every action realized by the natural agents in an artificial agent is based in the observation of the nature.

4 Systems of Ontologies of Communication

Every robot must know the directions of its community, or detect what signs are available, to be able to send the requests of connection and initiate the communicative act that will coordinate the execution of the tasks to realize. For this connection, three agents are needed: one agent capable of detecting signs, and to check the existing directions, one agent to request connection and other one to accept the request (Fig. 1).

On the other hand, whenever a new member joins the community, it has to report to its companions which are its characteristics and the functions for those that it was designed. For that purpose, it needs to have an agent that it should allow it to deliver its "curriculum" to the rest of the community, and in turn, it needs an agent capable of catching the information relating to each of its new collaborators. (Fig. 1).

During the accomplishment of the tasks, certain needs arise when in an individual is not capable of solving the task. In this case, it requests help to others. This implies that every individual has the aptitude to communicate its needs and in turn to process requests raised from others. A way of solving this problem is by means of agents in charge of realizing these tasks (Fig. 2).

Besides, it is necessary to have an agent capable of trying the need and to generate an action that could cover this need, based on the knowledge that it has, or to declare itself unable to realize something of usefulness. In brief, there needs an agent capable of making an individual decision.

5 Ontology System of Making of Decisions

In the execution of a cooperative and collaborative work, the communities of robotic agents face problems that need of rapid solution, even if its behavior is intrinsic in them and present certain characteristics of collective memory. They have to take decisions that in some cases involve to sacrifice some members of the colony. For it, when a need is communicated and it receives several individual answers to that aptitude to cover the need, the community must decide which of the offers is better, or how to coordinate them. For it, there can be applied different concepts based on the ontologies of decision in which the robotic MAS is sustained.

The last instance, before realizing the pertinent assignments to cover a need, is the taking of Decisions (D), where the conducts of the community meet reflected with major force, from the coordination up to the cooperation. Unlike the traditional focus in the collective robotics, in the community of robotic agents there appears a new concept, the consensus. Any decision taken must be realized by mutual agreement for the whole community that it is directly faced to a problem – Fig. 3.

The Making of Decisions carries out three different forms, commonly used among human groups:

Decision for Similarity (DS): This one is taken when more than one individual presents the same solution, expressed of different way; the community may take both as different, and nevertheless it has to have the aptitude to detect the similarities among them and to obtain the unique solution. Yet, when there are significant differences between the offers realized, it must be feasible to unite the similarities among them and to optimize the differences to obtain the final solution that will be applied.

Decision for Quality (DC): This one turns out to be one of the most complexes since the robots must be capable of determining which of all the offers is the optimal, or the best solution. This requires that to community possesses another human characteristic, the

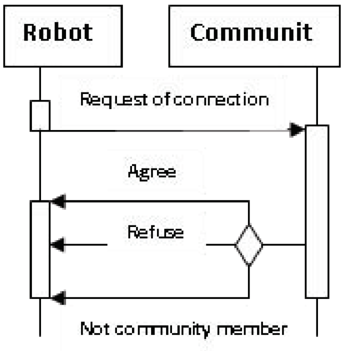


Figure 1: FIPA Protocol for community of Robotic Agents.

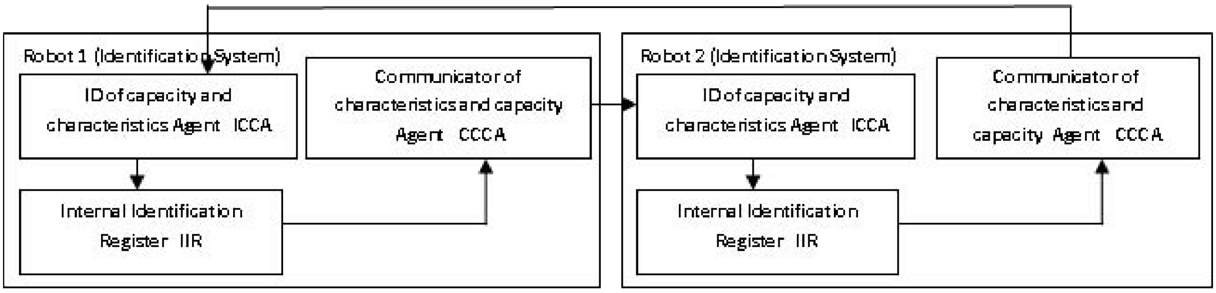

Robot 1 (Identification System)
ID of capacity and characteristics Agent ICCA
Communicator of characteristics and capacity Agent CCCA
Internal Identification Register IIR
Robot 2 (Identification System)
ID of capacity and characteristics Agent ICCA
Communicator of characteristics and capacity Agent CCCA
Internal Identification Register IIR

Figure 2: Architecture of knowledge behaviors for community of Robotic Agents.

intelligence, which brings the artificial intelligence, besides needing a major quantity of resources assigned to the Making decisions.

Decision for Majority (DM): This one refers to the choice of a solution from the generated offer more time, this solution goes of the hand in the majority of the times of the decision for similarity, since it turns out slightly probable that two or more agent components of robots with different characteristics and particular knowledge generate exactly equal solutions. The decision for majority implies reducing the number of offers by means of the assimilation, then there determining which is the accepted solution by means of the acceptance of the members' major quantity.

The behavior (similar to the human being) assigned to the community of robotic agents, goes directly related to the solution of problems. It is for it that the actions to take realize in a sequenced way being careful not to omit any of them in order not to present faults to the moment of it determines the suitable way of handling a certain situation arisen during the palliation of pertinent labors for the fulfillment of an aim.

On the other hand, the Satisfactories generally they turn out to be inactive individuals of the colony, already be that they are in the nest, or that have stopped its labors inside the community. These Agents Robots are capable of solving problems and of covering the needs of its companions as they were described in the previous point.

The general way of obtaining the Satisfactory determined by means of the process that culminates in the Making of Decisions, is by means of communicative acts which allow to deliver information to other members of the colony on what it is looked, in order that these could provide it. In a graphical form this sequence of actions is represented in Fig. 4.

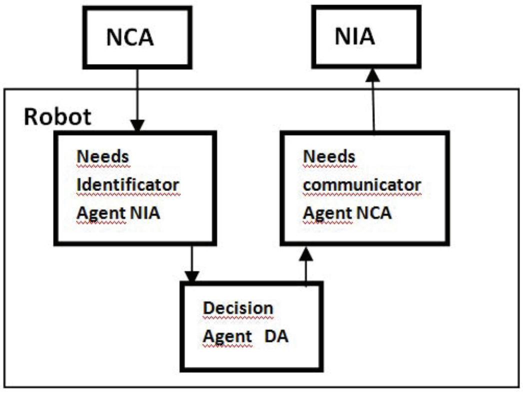


Figure 3: Architecture for making decisions in a community of Robotic Agents.

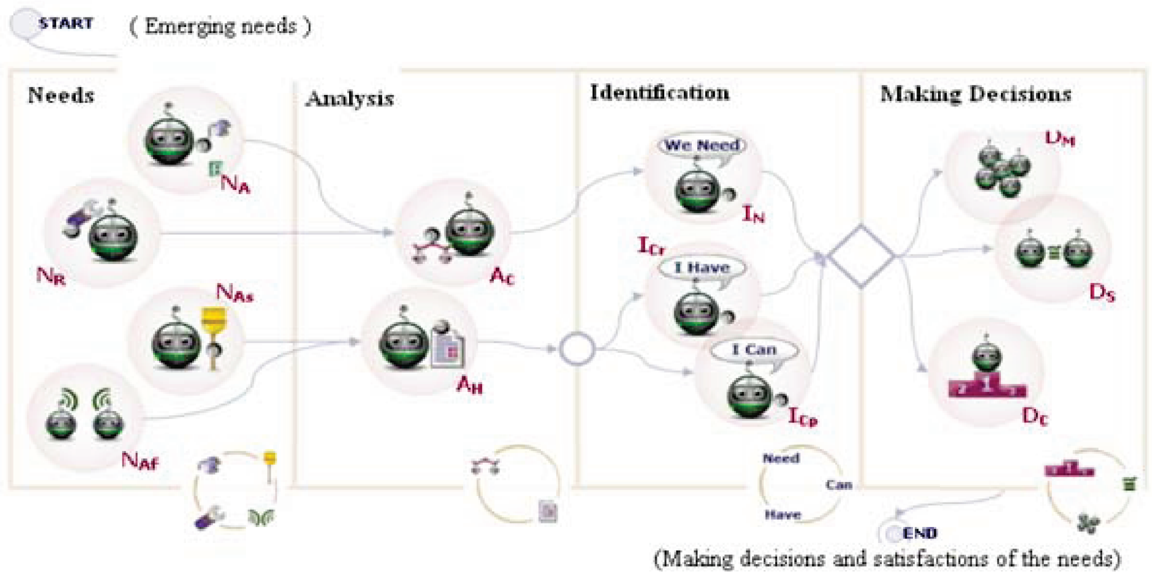


Figure 4: Sequence of actions for the search of a Satisfactory decision.

In a robotic community, every individual must have a small base of knowledge handled by an agent which will determine those actions can be used in the future and which not, beside to determine which are those situations that must be reported to the rest of the community to create a collective memory.

Likewise, the robots must possess an agent capable of Making a record of mistakes and correct them, registering also the above mentioned alteration. Also it has to have the aptitude to eliminate the knowledge that is unnecessary or is obsolete, to avoid information garbage inside the robot and its community.

6 Ontologies for Needs

These types of Ontologies define the particularities of every type of existing need. Nowadays, there are four basic needs, the nourishment, affiliation, assignment and repair. Each of them possesses certain points by means of which there can be associated an individual and to a labor, bequeathing this way to facilitating the take of decisions in the active community. Each of these needs possesses a particular class, these are:

Class Nourishment: This one describes the levels of discharge of energy that produce some problem in a robot, and defines the possible solution to the above mentioned mishap. The table 1 shows one of the subclasses that shape the class Nourishment.

Class Affiliation: This one describes the most particular need of all, the affiliation. This one takes place at the arrival of an individual to the active community and defines what must be fulfilled in order that this one is accepted by who will be its companions of Making place the affiliation of the new individual to the group. The table 2 shows one of the subclasses that shape the class Affiliation.

Class Assignment: This one describes the need of union between an individual and a labor. When it arises a certain task must be looked the one who realizes it, of compatibility exists with the Agents Robots inactive these they will be assigned, of not being like that a request will realize to the Agency Robots. They can be given in combination when more than one individual needs, and the total coverage of the Satisfactory does not exist in the active community. The table 3 shows one of the subclasses that shape the class Assignment.

Class Repair: This one describes possible damages somewhere or a piece of an individual, defining possibilities of repair, change or to send to workshop, depending on the severity of the fault. The table 4 shows one of the subclasses that shape the class Repair.

Subclass	Definition
Level of Hunger	Measure in per cent. It refers at the level of discharge that presents the battery of nourishment of the robot.
Intermediate State	It adduces to the state to which there will change the individual while it is recharged (if it is left or it is still active or in wait)
Satisfactory	It will determine depending on the unload if a loader or a derrick is necessary.
Way of Connection	It determines the type of connector between the individual uncharged and the individual charger.

Table 1: Subclass of the class Nourishment

Subclass	Definition
Community Activates	It determines the identificator of the active community to which an individual is sent.
Individual	It determines the identificator of the individual who is sent.
Minimal percentage of acceptance	It determines the minimal percentage of approval that must exist in order that the individual be accepted as member of the community.

Table 2: Subclasses of the class Affiliation.

Subclass	Definition
Labor	Definition of the labor to realize.
Satisfactory	Determination of the type of suitable individual.
Compatibility with O_I	Existence of compatibility with the Agents Robots inactive inside the active community. (Alphanumeric chain which first character is the number of Agents compatible Robots, followed by its identificator).
Request to Agency Robots	Package of request of compatible individuals to send to the Agency (chain of characters where the first character defines the quantity followed by the description of the Satisfactory).

Table 3: Subclasses of the class Assignment

In the communities of robotic agents, it is necessary to define each and every of the ontologies that there are forming the base of knowledge of the robotic agents, which are fundamentals to define the task that the community is capable of realizing.

In this work, it presents a proposition of the ontologies of connection, of decision and of needs, for a community of robotic agents in particular, which joined the development of intelligent agents, manages to form a type of social behavior very similar to the behaviors in the conducts presented by social groups such as the ants, which possess a highly developed instinct of collaboration and cooperation.

Subclass	Definition
Damaged Piece	It determines which is the piece that presents damage.
Level of damage	It determines of percentage form the severity of the fault.
Capacity of change	It determines if it is possible to replace the damaged piece or not.
Intermediate State	It adduces to the state to which there will change the individual while it is repaired (if it is left or it is still active or in wait)
Satisfactory	It determines that repairer is the most suitable, or if it turns out to be more suitable if a derrick is requested.

Table 4: Subclasses of the class Repair

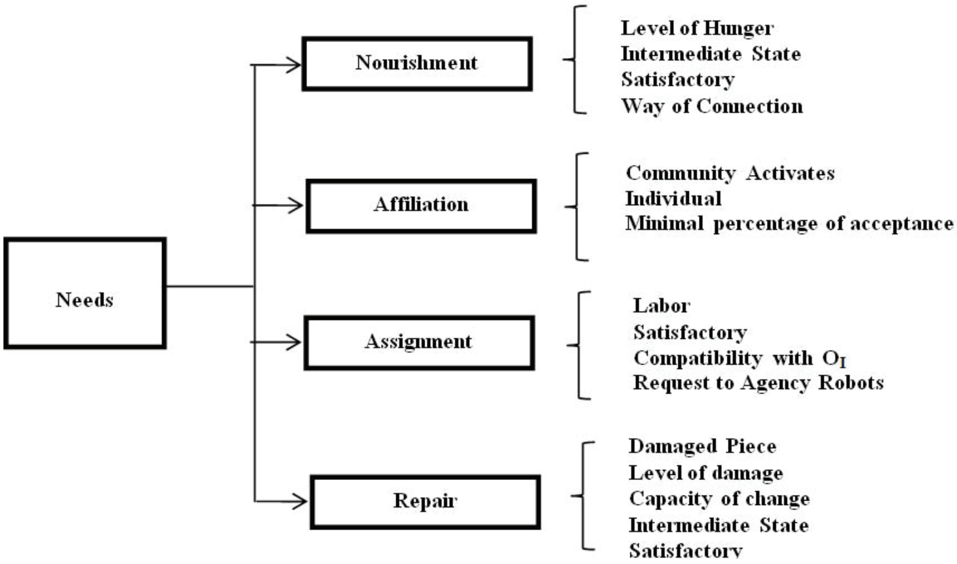


Figure 5: Ontology of Needs

The following image (Fig 5) presents a graph of the components of the classes of the Ontology of Needs.

7 Conclusions

In the communities of robotic agents, it is necessary to define each and every one of the ontologies that there are forming the base of knowledge of the robotic agents, which are fundamental to define the task that the community is capable of realizing.

In this work, it presents a proposition of an Ontology Model for a cooperative community of heterogeneous robotics agents. It defines the characteristics of collaborative and cooperative community of robots, using multi agent theory where each robot shares information with others robots in the community, to accomplish a common objective. The ontologies are for connection, for decisions and for needs, for a community of robotic agents in particular, which joined the development of intelligent agents, manages to form a type of social behavior very similar to the behavior by social groups such as the ants.

Bibliography

[1] Lefranc, G. "Colony of robots: New Challenge", Int. J. of Computers, Communications & Control, ISSN 1841-9836, E-ISSN 1841-9844, Vol. III (2008), Suppl. issue: IJCCC 2008, pp.

92-107.

[2] Ellips Masehian, and Davoud Sedighizade, "Classic and Heuristic Approaches in Robot Motion Planning – A Chronological Review". Proceedings of World Academy of Science, Engineering and Technology Volume 23, August 2007.

[3] McGuinness, D.L., Fikes, R., Rice, J. and Wilder, S. An Environment for Merging and Testing Large Ontologies. Principles of Knowledge Representation and Reasoning: proceedings of the Seventh International Conference. A. G. Cohn, F. Giunchiglia and B. Selman, editors. San Francisco, CA, Morgan Kaufmann Publishers. 2000.

[4] Supnithi T., Inaba A., Ikeda M., Toyoda J., Mizoguchi R., 1999. "Learning goal ontology supported by learning theories for Opportunistic Group Formation". *Proc. of AIED99*, pp. 67-74.

[5] Brophy S., Biswas G., Katzlberger T., Bransford J. and Schwartz D., 1999. "Teachable agents: Combining Insights from Learning Theory and Computer Science". In S. P. Lajoie and M. Vivet (Eds.), Artificial Intelligence in Education, pp. 21-28.

[6] Sloczinski, H. Lucila Maria Costi Santarosa, "Aprendizagem Coletiva em Curso Mediado Pela Web", VII Congresso Iberoamericano de Informática Educativa, 2003.

[7] Nwana H., 1996. "Software Agents: An Overview". Vol. **11**, No 3, pp. 205-244. Cambridge University Press.

[8] Paquette G., 1999. "Meta-Knowledge representation for learning scenarios engineering". Proceedings of AI-Ed, 1999.

[9] Posadas J., 2003. "Arquitectura para el control de robots móviles mediante delegación de códigos y agentes". Tesis Doctoral. Universidad Politécnica de Valencia.

[10] Soller, A. Linton, F., Goodman, B., and Lesgold, A., 1999. "Toward Intelligent analysis and support of Collaborative learning interaction". Proceedings of the Ninth International Conference on Artificial Intelligence in Education, Le Mans, France, 75-82.

[11] Russell, S., Norvig, P., 1995. "Artificial Intelligence: A Modern Approach". Ed. Prentice Hall.

[12] Wooldridge M. and Jennings R., 1995. "Intelligent agents: Theory and practice". The Knowledge Engineering Review, 10(2): pp. 115–152.

Int. J. of Computers, Communications & Control, ISSN 1841-9836, E-ISSN 1841-9844
Vol. VI (2011), No. 1 (March), pp. 134-149

Load Balancing by Network Curvature Control

M. Lou, E. Jonckheere, F. Bonahon, Y. Baryshnikov, B. Krishnamachari

Mingji Lou
Western Digital
Lake Forest, CA 92692, USA
E-mail: mingjilou@gmail.com

Edmond Jonckheere, Francis Bonahon, Bhaskar Krishnamachari
University of Southern California
Los Angeles, CA 90089-2563, USA
E-mail: jonckhee@usc.edu, fbonahon@math.usc.edu, bkrishna@usc.edu

Yuliy Baryshnikov
Alcatel-Lucent
Murray Hill, NJ 07974-0636, USA
E-mail: yuliy.baryshnikov@alcatel-lucent.com

Abstract: The traditional heavy-tailed interpretation of congestion is challenged in this paper. A counter example shows that a network with uniform degree can have significant traffic congestion when the degree is larger than 6. A profound understanding of what causes congestion is reestablished, based on the network curvature theorem. A load balancing algorithm based on curvature control is presented with network applications.

Keywords: network congestion, curvature, inertia, Gromov hyperbolic graphs, Poincaré disk, load balancing, Yamabe flow.

1 Introduction

One of the most important challenges in networking systems, especially in large and wide area networks, is the traffic congestion problem. The queuing feature between two routers can create a logical bottleneck between two users. Correspondingly, insufficient bandwidth on the physical links between routers is a contributor to congestion. The current congestion control technologies in communication networks are based on the feedback from the congested node to the source to slow down the packet flow rate, such as bidirectional congestion control and Random Early Detection (RED). However, these technologies can only be applied once the congestion has happened to some degree, and it is only based on the *local* point of view of some queue overflow along the source to target path. This paper investigates the deeper reason behind the congestion *in the large scale*, and will challenge the current least-cost-path algorithms, such as Dijkstra's shortest path algorithm. It will indeed be established that these least-cost-path algorithms aggravate the congestion, especially in negatively curved networks.

A fundamental ingredient in this paper is that, in order to get a large-scale view on the congestion problem, we utilize the coarse approximation of a network graph by a Riemannian manifold. A graph as a mathematical idealization of a network is completely different than a Riemannian manifold; however, the recent development of the so-called coarse geometry under the leadership of Mikhael Gromov has given the two mathematical structures—graphs and manifolds—the unifying framework of geodesic spaces. As a corollary, the concept of curvature has become applicable to graphs [2, 8–10]. The fundamental mathematical idea behind this unification is to realize that the traditional Riemannian curvature, which relies on the differentiable structure of the manifold, can be rephrased in terms of the more primitive concept of distance [3, 6]. Since a communication network can be endowed with a distance, which represents communication cost, delay, outage, etc., its curvature can be defined. The positively curved versus negatively curved network dichotomy roughly corresponds to the more traditional meshed (decentralized) versus core-concentric (centralized) network dichotomy [14].

It has been experimentally observed that, on the Internet and other networks, traffic seems to concentrate quite heavily on some very small subsets. As shown in Fig. 1, congestion could occur at the “core” through an easy mechanism. However, one extremely important point that will be made in this paper is that, contrary to traditional belief, congestion is not necessarily a manifestation of the heavy-tailed behavior related to high node degree, but is a manifestation of a more subtle process that can be traced back to the curvature.

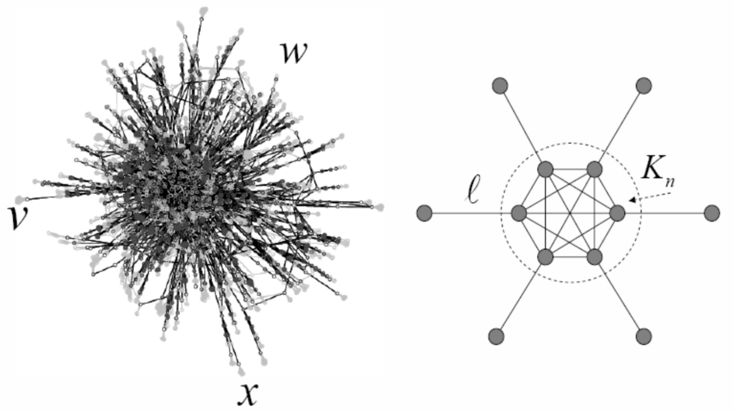


Figure 1: Traditional understanding of congestion occurring at the “core:” left, Internet Service Provider (ISP) graph; right: idealized model.

The mathematical apparatus and computer simulations will unveil this striking traffic pattern in negatively curved networks from both a theoretical and a practical point of view. In further, this paper studies another fundamental question: if congestion does not necessarily occur at vertices of high degree, nor at the so-called highly connected “core,” then what are the congestion points? This paper shows that congestion occurs at the points of least inertia of the network. Last but not least, a curvature-based load-balancing routing is proposed, and its performance is compared with a shortest-path routing in multicast communication network.

2 General Concepts and Conjectures

2.1 Curvature, Traffic, Betweenness, Inertia, and Centroid

Let $G = (V, E)$ be a graph specified by its vertex set V and its edge set E and endowed with a (symmetric) distance function $d : G \times G \to \mathbb{R}^+$. A path $p(s, t)$ from s to t is a continuous map $[0, l] \to G$ such that $p(s, t)(0) = s$ and $p(s, t)(l) = t$. The weight of an edge $e = xy$ is defined as $w(e) = d(x, y)$. The ***length*** of the path is defined as $\ell(p(s, t)) = \sum_{e \subseteq p(s,t)} w(e)$. A ***geodesic*** $[s, t]$ is a path such that $\ell\left([s, t]\right) \leq \ell\left(p\left(s, t\right)\right)$, $\forall p(s, t)$. A geodesic ***triangle*** is defined as $\Delta abc = [a, b] \cup [b, c] \cup [c, a]$.

For the sake of simplicity, the network curvature concept is restricted to planar communication graphs and is based on Alexandrov angles [1,3,6]. Let $(ab_1 = ab_{\deg(a)+1}, ab_2, ..., ab_{\deg(a)})$ be a cyclic ordering of the set of edges attached to the vertex a. The Alexandrov angle α_k at the vertex a of the geodesic triangle $\Delta ab_k b_{k+1}$ is defined as $\alpha_k = \cos^{-1} \frac{w(ab_k)^2 + w(ab_{k+1})^2 - w(b_k b_{k+1})^2}{2w(ab_k)w(ab_{k+1})}$

and the (Gauss) ***curvature*** at the vertex a is defined as

$$\kappa(a)=\frac{2\pi-\sum_{i=1}^{\deg(a)}\alpha_k}{\sum_{k=1}^{\deg(a)}A(\Delta ab_kb_{k+1})}=\frac{K(a)}{\sum_{k=1}^{\deg(a)}A(\Delta ab_kb_{k+1})} \tag{1}$$

where $A(\Delta ab_kb_{k+1})$ denotes the area of the geodesic triangle Δab_kb_{k+1} easily computable via Heron's formula. It is easily seen that, for the number of hops metric ($w(e)=1$), $\alpha_k=\pi/6$; therefore, $\kappa(a)<0$, $\kappa(a)=0$, or $\kappa(a)>0$ depending on whether $\deg(a)>6$, $\deg(a)=6$, or $\deg(a)<6$, respectively.

An infinite negatively curved graph has the property that it is isometric to a negatively curved manifold up to a bounded distortion (see [5] for precise statement). Since this graph-manifold identification entails a bounded error, large scale problems on graphs can be mapped to more manageable continuous geometry problems on manifolds (see Sec. 4).

The traffic on the graph is driven by a ***demand measure*** $\Lambda_d : V\times V\to\mathbb{R}^+$, where the demand $\Lambda_d(s,t)$ is the traffic rate (e.g., number of packets per second) to be transmitted from the source s to the destination target t. Assume that the routing protocol sends the packets from the source s to the target t along the path $p(s,t)$ with probability $\pi(p(s,t))$. As such, the path $p(s,t)$ inherits a traffic rate measure $\tau(p(s,t))=\Lambda_d(s,t)\pi(p(s,t))$. An edge e laying on the path $p(s,t)$ inherits from that path a traffic $\tau(p(s,t))$. Aggregating this traffic over all source-target pairs and all paths traversing the edge e yields the traffic rate sustained by the edge e, $\tau(e)=\sum_{(s,t)\in V\times V}\sum_{p(s,t)\supseteq e}\tau(p(s,t))$. The ***traffic rate*** at a vertex v is defined as

$$\begin{array}{rcl} b(v) & = & \frac{1}{2}\sum_{e\supseteq v}\tau(e)+\sum_{s\neq v}\Lambda_d(s,v)+\sum_{t\neq v}\Lambda_d(v,t)\\ & = & |\{[s,t]:v\in[s,t]\}| \quad (\text{if } \Lambda_d(x,y)=1, \forall x\neq y)\end{array}$$

The notation $b(\cdot)$ stands for ***betweenness*** [3], as for a uniformly distributed demand, the traffic at v is the number of geodesics passing through v. Given a connected subgraph $X\subseteq G$, we define its ***traffic load*** to be representative of the number of packets in it:

$$\Lambda_t(X)=\sum_{s,t\in V}\left(\sum_{e\in p(s,t)\cap X}w(e)\right)\Lambda_d(s,t)\pi(p(s,t)) \tag{2}$$

The ***inertia*** of a (connected) graph G with respect to a vertex v is defined as $\phi_G(v)=\sum_{v^i\in V}d^2(v,v^i)$. Observe that this inertia may be infinite. A ***center of mass*** or ***centroid*** of the graph G is defined as a vertex relative to which the inertia is minimum: $\mathrm{cm}(G)=\arg\min_{v\in V}\phi_G(v)$. The global minimum need not be unique.

2.2 Conjectures

Conjecture 1. G$^-$. *If the graph $G=(V,E)$ is negatively curved along with a demand measure Λ_d uniformly distributed over $V\times V$, the least-cost protocol that sends packets over optimal routes leads to a very high traffic rate $b(v)$ (traffic load $\Lambda_t(X)$) over a very small number of vertices v (over a very small subset X).*

Conjecture 2. G$^+$. *If the graph $G=(V,E)$ is nonnegatively curved along with a demand measure Λ_d uniformly distributed over $V\times V$, the protocol that sends packets over optimal routes leads to a nearly uniform traffic rate $b(v)$.*

We can now formulate our third conjecture, saying that the maximum traffic load for uniformly distributed demand occurs near the center of mass of the network relative to uniformly distributed weight:

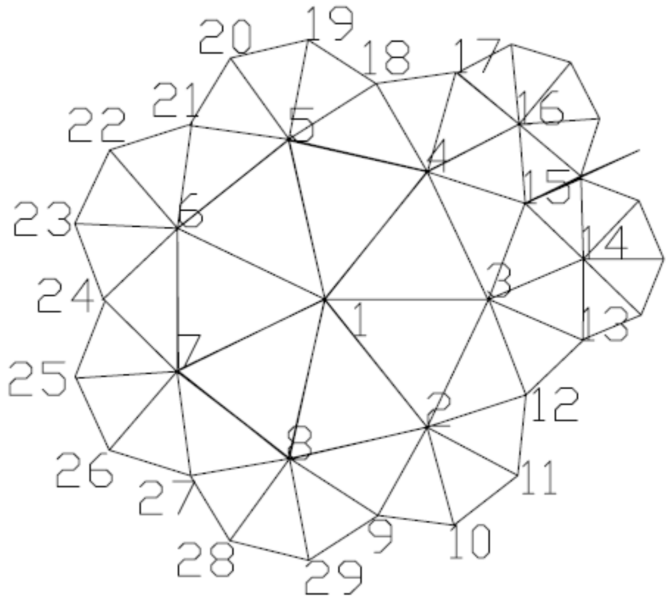


Figure 2: Simple hyperbolic graph with node degree seven and uniform edge length.

Conjecture 3. **G**. $\arg\max_v b(v) \approx \mathrm{cm}(G)$. *(Equality failed only in one positively curved example.)*

Conjecture 4. $\mathbf{M}^-$. *If the graph G is negatively curved, the inertia $\phi_G(v)$ has a unique global minimum and* $\mathrm{cm}(G)$ *is unique. (This result is already known for global Busemann nonpositively curved spaces [12].)*

Conjecture 5. $\mathbf{M}^+$. *If G is nonnegatively curved,* $\mathrm{cm}(G)$ *is not uniquely defined. (In a real-life, massive, nonnegatively curved network, the inertia $\phi_G(v)$ is nearly constant with v and* $\mathrm{cm}(G)$ *might be hard to identify.)*

Clearly, Conjecture $M^\pm$ along with Conjecture G would yield Conjecture $G^\pm$.

3 Benchmark Examples

Several benchmark examples in here will provide support for these conjectures: a set of planar graphs in which the curvature is dictated by the valence (degree) of the nodes, as shown in Fig. 2. As shown in Fig. 3, we examine the negatively curved cases of valence 7, 8 and 9, hence of curvature $\frac{1}{2\pi}\left(2\pi - 7\frac{\pi}{3}\right) = -\frac{1}{6}$, $-\frac{1}{3}$ and$-\frac{1}{2}$, respectively, in which significant traffic congestion occurs at the *centroid* of the graph when least cost routing (Dijkstra's) algorithm is applied. Then we contrast the results with those of a vanishing curvature graph of valence 6 (curvature=0), in which the congestion is more smoothly distributed over all nodes. Towards a more realistic situation, we then look at a case of mixed valence. We then proceed to positively curved graph of valence <6, in which the situation is drastically different than in negative curvature, as the traffic is uniformly distributed!

Clearly, as shown in Fig. 4, as the curvature becomes more and more negative, the vertex carrying the heaviest traffic in the graph becomes more and more congested relative to the other nodes, consistently with the inertia at the center of mass becoming smaller and smaller By increasing the node degree, with the same number of nodes and traffic demand, there are more

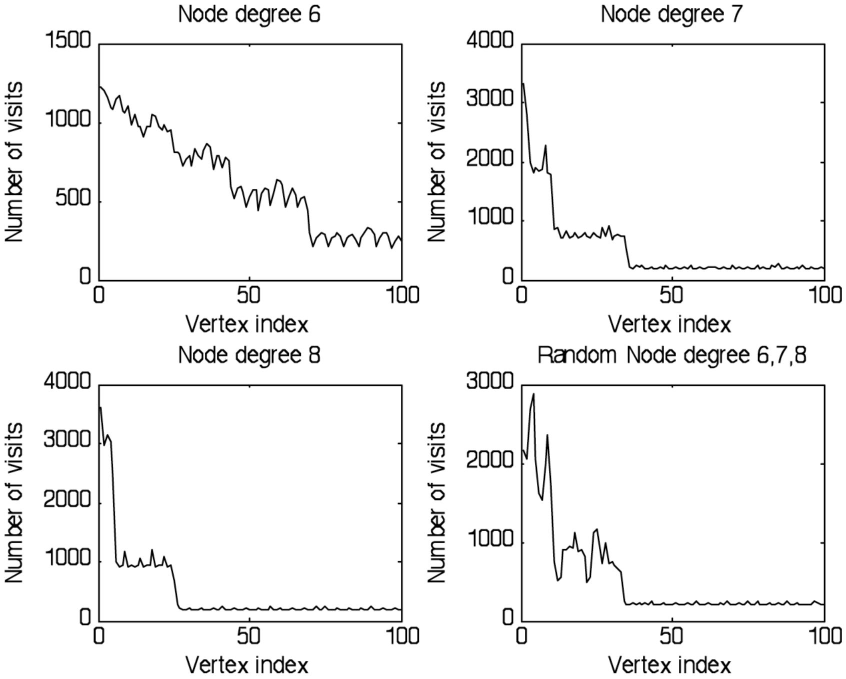


Figure 3: Traffic distribution in different planar networks: degree 6 everywhere, degree 7 everywhere, degree 8 everywhere, and random degree 6/7/8. The routing algorithm implements random pickup of equal cost paths.

connections in the network, so that the total traffic in the network decreases, but the traffic is heavier in the congestion center.

To further test Conjecture $\mathbf{M}^-$, especially to eliminate a possible contribution of the symmetrical structure of the graph to the congestion cases of the previous paragraph, we simulate the traffic and inertia distribution in a highly unsymmetrical network as shown in Fig. 5. The position of the heaviest traffic congestion point matches the node with minimum inertia. These results, in further, confirm our conjecture $\mathbf{M}^-$.

4 Proofs of Conjectures

4.1 Proof Conjecture $\mathbf{G}^+$: Traffic in Positive Curvature

Consider the Platonic solids. All of these Platonic solids have their symmetry group Γ. This symmetry group Γ acts on the vertex set V_P of the Platonic solid P as a map $\Gamma \times V_P \to V_P, (g, v) \mapsto g(v)$, where g is an element of the symmetry group. Recall that the action of a symmetry group on a space is ***vertex-transitive*** if $\forall v, w \in V_P$ there exists a $g \in \Gamma$ such that $w = g(v)$.

It is easy to see that the action of the symmetry groups on the Platonic solids is vertex-transitive. We prove this as follows: Observe that all Platonic solids except the tetrahedron have dihedral (rotation) symmetries about axes joining the centers of pairs of opposite faces, while the tetrahedron has D_3 symmetry about the axis joining a vertex to the center of the opposite face. Then consider two vertices v, w on a Platonic solid. Join them by a sequence of consecutive edges. It is easy to see that the beginning vertex of an edge can be moved to the end vertex by a symmetry about the axis perpendicular to the center of a face comprising the edge.

With the above concept, it is easy to prove that the betweenness is uniform. Let $b_G(w)$

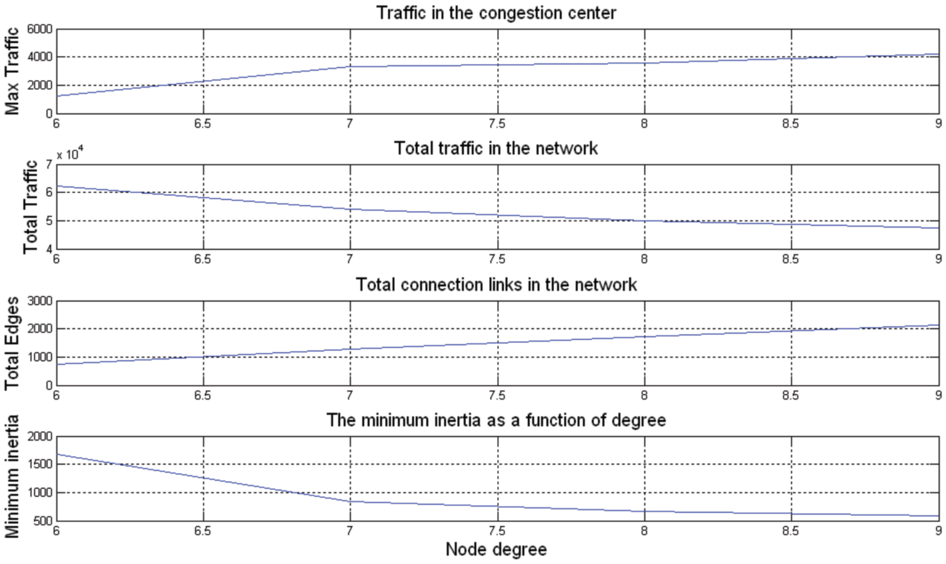


Figure 4: Traffic at the center of mass, total traffic in the network, and the total connection in the network.

denote the betweenness of the node v *in the graph* G. Then we have

$$b_G(w) = b_G(gv) = b_{g^{-1}G}(v) = b_G(v)$$

The only nontrivial part in the above string of equalities is the second one, where it is essential that the edge length be uniform. Indeed if (s,t) is a pair communicating via gv, we have $d(s, g(v)) + d(g(v), t) = d(s, t)$, from which it follows that $d(g^{-1}s, v) + d(v, g^{-1}t) = d(g^{-1}s, g^{-1}t)$, hence there is a pair $(g^{-1}s, g^{-1}t)$ communicating via v.

The proof that the inertia is uniform is essentially the same:

$$\phi_G(w) = \phi_G(gv) = \phi_{g^{-1}G}(v) = \phi_G(v)$$

The proof that $\tau(v)$ is uniform involves the edge-transitivity of the symmetry group. Let $\tau_G(e)$ be the traffic rate on edge e *in the graph* G. Then

$$\tau_G(e_2) = \tau_G(ge_1) = \tau_{g^{-1}G}(e_1) = \tau_G(e_1)$$

Again, in the second inequality, it is essential that the demand be uniform. From the above, it easily follows that $\tau_G(e_1) = \tau_G(e_2)$.

Hence we have the following result: For a uniformly distributed demand measure $\Lambda_d : V_P \times V_P \to \mathbb{R}^+$ on the squared power of the vertex set of one of the 5 Platonic solids, the traffic load $b : V_P \to \mathbb{R}^+$ is uniform, for a geodesic routing and provided the traffic is equally distributed among pairs of nodes.

More generally, by Higuchi's theorem [17], positively curved graphs are finite; next, the valence can only take values 3, 4, 5; therefore, the inertia and the congestion remain bounded from above and from below.

4.2 Quantitative Measure of Traffic in Disks

In general, the ***traffic load density*** in a convex subset X of a surface Σ is defined in a way inspired from (2), except for some normalization,

$$\lambda_t(X) = \frac{1}{A(\Sigma)^2 A(X)} \iint_{(s,t)\in\Sigma\times\Sigma} \ell\left(X \cap [s,t]\right) d\Lambda_d(s,t)$$

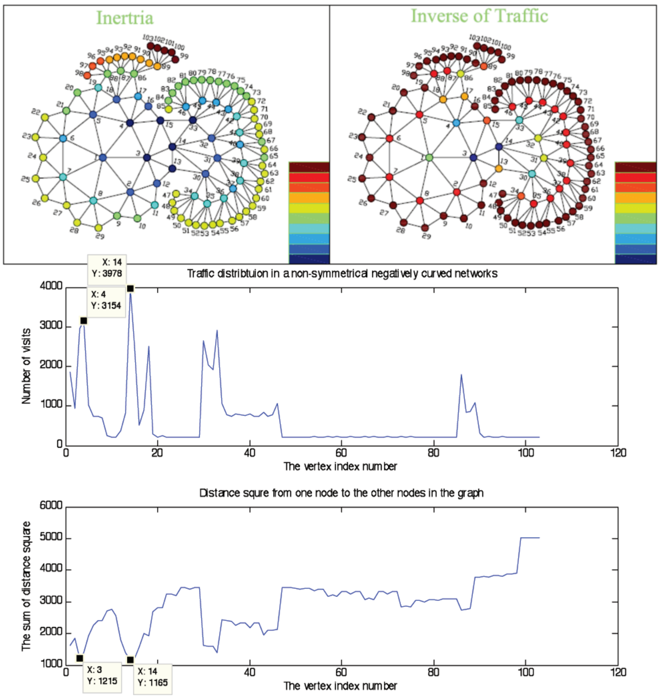


Figure 5: Traffic and inertia distribution in an unsymmetrical network. (Vertex #3 has maximum traffic and minimum inertia.)

Here Σ is the "network" that comprises all sources s and all targets t; the normalization by the squared area $A(\Sigma)^2$ is justified by $d\Lambda_d(s,t) = A(ds)A(dt)$, and the normalization $A(X)$ is for obvious reasons. From here on, we specialize the computations to $\Sigma = B(R)$ and $X = B(r)$, a large ball and a small ball ($R >> r$), respectively, with their common center at the origin of the Euclidean space $\mathbb{E}^2$ or the hyperbolic space $\mathbb{H}^2$.

4.3 Conjecture G$^+$: Traffic at the Center of a Euclidean Disk

The above double integral can be rewritten as the following:

$$\lambda_t(X) \quad = \quad \frac{1}{A\left(B(R)\right)^2 A(B(r))} \int_0^{\sqrt{R^2-u^2}-\sqrt{r^2-u^2}} \int_0^{\sqrt{R^2-u^2}-\sqrt{r^2-u^2}} \int_0^{2\pi} \int_0^{r} \ell\left\{X \cap [s,t]\right\} \ldots$$
$$\ldots \times |\text{Jacobian}|\; du\, d\theta\, dl\, dl$$

We first compute the Jacobian relative to the change of variables from Cartesian coordinates to polar coordinates. Assume the points s and t are at (x,y) and (x',y') in Cartesian coordinates. As shown in Fig. 6, their corresponding representations in polar coordinates with (u,θ,l,l') are the following:

$$\begin{aligned} x &= u\cos\theta + (l+\sqrt{r^2-u^2})\cos\left(\theta+\tfrac{\pi}{2}\right) = u\cos\theta - (l+\sqrt{r^2-u^2})\sin\theta \\ y &= u\sin\theta + (l+\sqrt{r^2-u^2})\sin\left(\theta+\tfrac{\pi}{2}\right) = u\sin\theta + (l+\sqrt{r^2-u^2})\cos\theta \\ x' &= u\cos\theta + (l'+\sqrt{r^2-u^2})\cos\left(\theta-\tfrac{\pi}{2}\right) = u\cos\theta + (l'+\sqrt{r^2-u^2})\sin(\theta) \\ y' &= u\sin\theta + (l'+\sqrt{r^2-u^2})\sin\left(\theta-\tfrac{\pi}{2}\right) = u\sin\theta - (l'+\sqrt{r^2-u^2})\cos(\theta) \end{aligned}$$

Long but elementary calculations show that $|\text{Jacobian}| = l + l'$. Then

$$\begin{aligned} \lambda_t(X) &= \lim_{R\to\infty} \frac{\int_0^{\sqrt{R^2-u^2}-\sqrt{r^2-u^2}} \int_0^{\sqrt{R^2-u^2}-\sqrt{r^2-u^2}} \int_0^{2\pi} \int_0^{r} 2\sqrt{r^2-u^2}(l+l')du\, d\theta\, dl\, dl'}{(\pi R^2)^2(\pi r^2)} \\ &= \frac{1}{\pi R} \end{aligned}$$

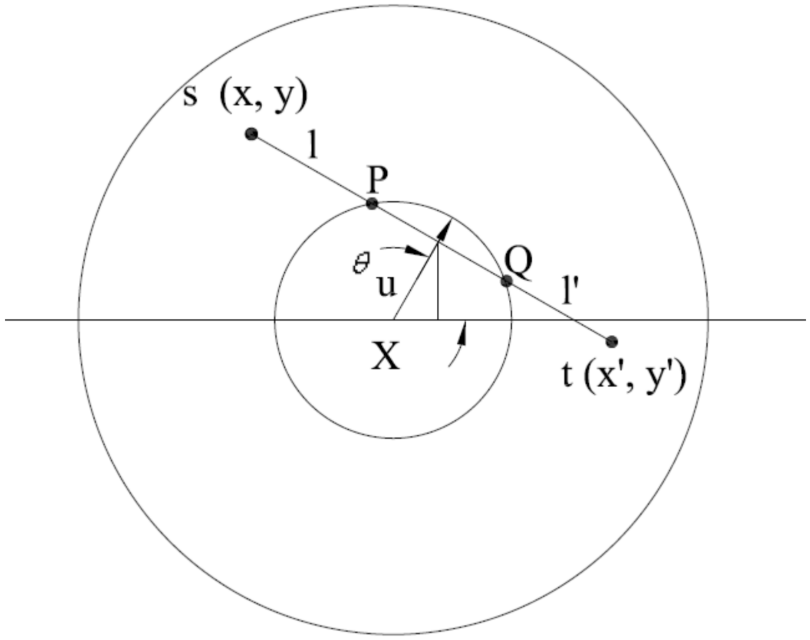


Figure 6: Traffic in Euclidean disk.

4.4 Conjecture G$^-$: Traffic at the Center of a Hyperbolic Disk

As shown in Fig. 7, the points s and t are at (x, y) and (x', y'), respectively, in the Cartesian coordinates of the Poincaré disk, $D = \{x + jy \in C : |x + jy| = r < 1\}$. It is, however, useful to parameterize the source and target by their representations in "polar" coordinates $(u, \theta, \overline{r}, \overline{r}')$, where the distances $u, \overline{r}, \overline{r}'$ are hyperbolic. If $r, \overline{r}$ represent the Euclidean and hyperbolic measurements, respectively, of the radius, then $r = \tanh\left(\frac{1}{2}\overline{r}\right)$. As is well known, the area element is given by

$$dA = \frac{4dxdy}{\left(1 - |r|^2\right)^2} = \frac{4dxdy}{\left(1 - \tanh^2 \frac{1}{2}\overline{r}\right)^2} = 4\cosh^4\left(\frac{1}{2}\overline{r}\right) dx\, dy$$

The Cartesian (x, y, x', y') versus polar-hyperbolic $(u, \theta, \overline{r}, \overline{r}')$ coordinate transformation is the following:

$$\begin{aligned} x &= \cos(\lambda + \theta) \cdot \tanh\left(\tfrac{1}{2}\overline{r}\right), & x' &= \cos(\theta - \lambda') \cdot \tanh\left(\tfrac{1}{2}\overline{r'}\right) \\ y &= \sin(\lambda + \theta) \cdot \tanh\left(\tfrac{1}{2}\overline{r}\right), & y' &= \sin(\theta - \lambda') \cdot \tanh\left(\tfrac{1}{2}\overline{r'}\right) \end{aligned}$$

where $\cos\lambda = \frac{\tanh u}{\tanh \overline{r}}$ and $\cos\lambda' = \frac{\tanh u}{\tanh \overline{r'}}$, per hyperbolic trigonometry in square angle triangles. Next,

$$|\text{Jacobian}| \overset{\bar{r}, \bar{r}' \to \infty}{=} O\left(\frac{1}{\cosh^2\left(\frac{1}{2}\overline{r}\right)\cosh^2\left(\frac{1}{2}\overline{r'}\right)}\right)$$

$$\begin{aligned} \lambda_t(X) &= \tfrac{1}{A(B(R))^2 A(B(r))} \iint_{(s,t)\in B(R)\times B(R)} \ell\left\{X \cap [s,t]\right\} A(ds)\, A(dt) \\ &= \tfrac{1}{A(B(R))^2 A(B(r))} \iint \iint_{B(R)\times B(R)} \ell\left\{X \cap [s,t]\right\} \times \left(16\cosh^4 \overline{r} \cdot \cosh^4 \overline{r'}\right) dx\, dx'\, dy\, dy' \\ &= \int_r^{\bar{R}} \int_r^{\bar{R}} \int_0^{2\pi} \int_0^{\bar{r}} \ell\left\{X \cap [s,t]\right\} \times \left(16\cosh^4 \overline{r} \cdot \cosh^4 \overline{r'}\right) \times |\text{Jacobian}|\, du\, d\theta\, d\overline{r}\, d\overline{r}' \\ &\approx O\left[\frac{\sinh^2\left(\frac{\bar{R}}{2}\right)\cosh^2\left(\frac{\bar{R}}{2}\right)}{\sinh^2\left(\frac{\bar{R}}{2}\right)\sinh^2\left(\frac{\bar{R}}{2}\right)}\right] \qquad (\bar{R} \to \infty) \\ &= O(\text{constant}) \end{aligned}$$

(The reader is referred to [15] for the details.)

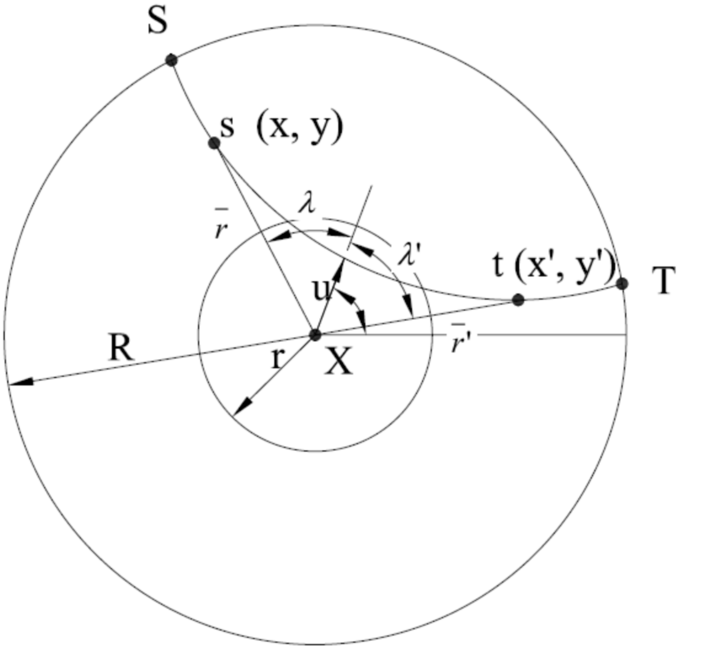


Figure 7: Traffic in hyperbolic disk.

The conclusion is that, in the hyperbolic case, the normalized traffic transiting through the small ball remains bounded from below as $R \to \infty$. This *strongly contrasts* with the Euclidean case, where the normalized traffic goes to zero as $R \to \infty$. In other words, in the hyperbolic case, the traffic density $\Lambda_t(X)/A(X) \asymp \text{cst}A(B(R))^2$, which is *worse* than the asymptotic estimate of $\text{cst}A(B(R))^{1.5}$ in the Euclidean case. (See [4] for traffic in scale-free rather than hyperbolic spaces.)

4.5 Conjecture $\mathrm{M}^{\pm}$: Minimum Intertia

In the Poincaré disk, the Laplacian operator is $\Delta = \left(1-|z|^2\right)^2 \frac{\partial}{\partial z}\frac{\partial}{\partial \bar{z}} = \frac{\left(1-|z|^2\right)^2}{4}\left(\frac{\partial^2}{\partial x^2}+\frac{\partial^2}{\partial y^2}\right)$. A twice continuously differentiable function f such that $\Delta f = 0$ is said to be *harmonic.* If $\Delta f \geq 0$, then the function is said to be *subharmonic.* What motivates the utilization of (sub)harmonic functions is that they reach their maxima on the boundary of analyticity.

Theorem 6. *The inertia of $B(R)$ in the Poincaré disk relative to the point v,*

$$\phi(v) = \iint_{B(R)} d(v,z)^2 dA(z)$$

reaches its minimum at $v = 0$.

Proof: We first show that $d^2(v,z)$ is subharmonic in v. Indeed, obviously, the Poincaré disk is a complete Riemannian manifold of nonpositive curvature, and hence it is a Busemann Non Positively Curved (NPC) space [12, page 45]. But in a Busemann NPC space, the distance squared is strictly convex [12, page 61]. A strictly convex function has positive definite Hessian [13, page 395]. Hence the trace of the Hessian, Δd^2, is (strictly) positive. Next, we prove that $\phi(v)$ is subharmonic; indeed

$$\Delta\phi(v) = \Delta \iint_{B(R)} d(v,z)^2 dA(z) = \iint_{B(R)} \left(\Delta d(v,z)^2\right) dA(z) \geq 0$$

Moreover, since $\Delta d^2(\cdot, z) > 0$, it follows that ϕ is subharmonic in the strong sense that $\Delta\phi > 0$. By rotational symmetry, $\phi(v)$ is constant on $v \in \partial B(r)$, $r < R$. Write this value as $\phi(\partial B(r))$. By the subharmonic property, it follows that $\phi(0) \leq \phi(\partial B(r)), \forall r < R$. For any point $v \in B(R)$, we obviously have $v \in \partial B(|v|)$, with $|v| < R$, so that $\phi(0) \leq \phi(\partial B(|v|)) = \phi(v)$. Hence the minimum is reached at $v = 0$. It remains to show that the minimum is unique, that is, to show that the preceding inequality can be strengthened to a strict inequality. From the strengthened subharmonic property $\Delta\phi > 0$ and the Green function argument of [11, page 9], it follows that $\phi(\partial B(|v|))$ is *strictly* increasing with $|v|$. Hence $\phi(v)$ reaches its (unique) minimum at $v = 0$.

□

5 Shortest-Path Routing vs. Curvature Based Load Balancing

5.1 Traditional Shortest-Path Routing

In the previous sections, we have shown, from a theoretical point of view, that for uniformly distributed demand the shortest path length routing in negatively curved networks causes congestion over a small number of vertices; moreover, these vertices with heavy traffic rate occur near the center of mass of the network. In this section, we more specifically look at this congestion phenomenon in the practical setting of traffic overload in communication network. To make this problem more specific and straightforward, we focus our attention on multicasting traffic, even though our theorem can be applied to more general communication network paradigms, such as VoIP and multimedia networking, mobile Ad-Hoc networks, wireless sensor networks, etc., where traffic congestion and routing algorithms are the big concerns in the design of those communication networks.

Multicasting could involve almost all layers of a communication network. A multicast task can be performed at the application layer, where a hybrid network is a good model for this application, as will be presented later. A multicast task can also go systematically through the physical, link, and network layers. The increasing popularity of group communication applications such as teleconference and information dissemination services has led to an increasing interest for the development of multicast transport protocols. However, these transport protocols could cause congestion collapse if they are widely used, as they ignore the curvature and are hence prone to the related congestion problems discussed above.

Two basic multicast tree algorithms are currently available in the industry: one is the dense-mode algorithm; the other is the sparse-mode algorithm. Both multicast tree algorithms are at the heart of the multicast protocols, such as the Distance Vector Multicast Routing Protocol (DVMRP) in dense-mode, and the Protocol-Independent Multicast (PIM) operating in both dense mode and sparse mode. As shown in Fig. 8a, the dense-mode uses the source-based tree. It determines a shortest-path tree to all destinations first, and then uses a reverse shortest-path tree rooted at a source. So the spanning tree starts at the source and guarantees the lowest cost from a source to all leaves of the trees. The sparse-mode algorithm uses a shared-tree technique which uses a rendezvous point (RP) to connect sources and receivers. This rendezvous point acts as the core or root to coordinate forwarding packets from source to receivers under its distribution subtrees, as shown in Fig. 8b.

We used the Network Simulator (`ns-2`) to build up the traffic congestion environment in multicast communication. To make our simulation straightforward, we focus our attention on the congestion versus network curvature issue by ignoring the dynamic change in the group membership and using User Datagram Protocol (UDP) as the sources. In further, we apply the same topology structure (uniform node degree 6, 7 or 8) into the `ns-2` simulation as the one already used in Section 3. A snapshot of the visualization with `ns-2` NAm (the Network

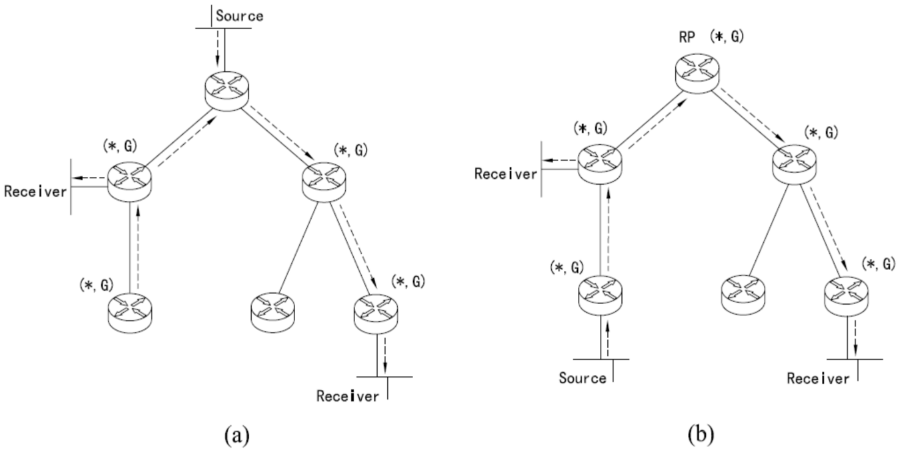


Figure 8: Two methods for constructing multicast algorithms: (a) dense mode, using a source-based tree; (b) sparse-mode, using a shared tree.

Animator) is shown in Fig. 9. In this figure, the node degree is 8 with a total of 100 nodes in the graph with node #0 at the centroid (a similar layout as the one shown in Fig. 2).

Other important networking settings are the followings:

- Every node in this graph is multicasting to all the other nodes in the network.
- The maximum buffer size of the queue in every link between two nodes is 1000 bytes pks, and every link is a duplex-link with 1Mb bandwidth, with a response time of 2ms.
- The size of every file is fixed to 2000 bytes.
- The start time of every UDP source is an exponential random variable with average value 0.01, and the interval time between two successive UDP packets for the source is 2.0 seconds.

As shown in Fig. 9, with the above setting, the network with 100 nodes and node degree 8 has congestion at nodes #0, #2, #3, #4, #5, as revealed by heavy packet drops. There is no such congestion for the network with node degree 6.

5.2 Load Balancing Routing

Load balancing algorithms are widely used to curb the congestion. For example, Cisco IOS router software has built-in load balancing functionality, and is available across all router platforms. It allows a router to use multiple equal cost paths to a destination when forwarding packets. The fundamental mechanism is as follows: When the router must select a route from many with the same administrative distance, the routers choose the path with the lowest congestion cost to the destination. In further, one can select load-balancing to work per-destination or per-packet. As shown in Table 1, from [7], the position of the asterisk (*) points to the interface over which the next packet/destination-based flow is sent; and the asterisk (*) keeps rotating among the equal cost paths each time a packet/flow is served.

However, in most cases, negatively curved networks have worst congestion problem, and the current load-balancing algorithms cannot alleviate it. The reason is this: first of all, there are not many multiple paths with the same administrative distance, since negatively curved manifolds have no conjugate points as the positively curved manifolds have. Second, even if we allow for quasi-optimal paths, there are still too close to the optimal one to bypass the congestion points.

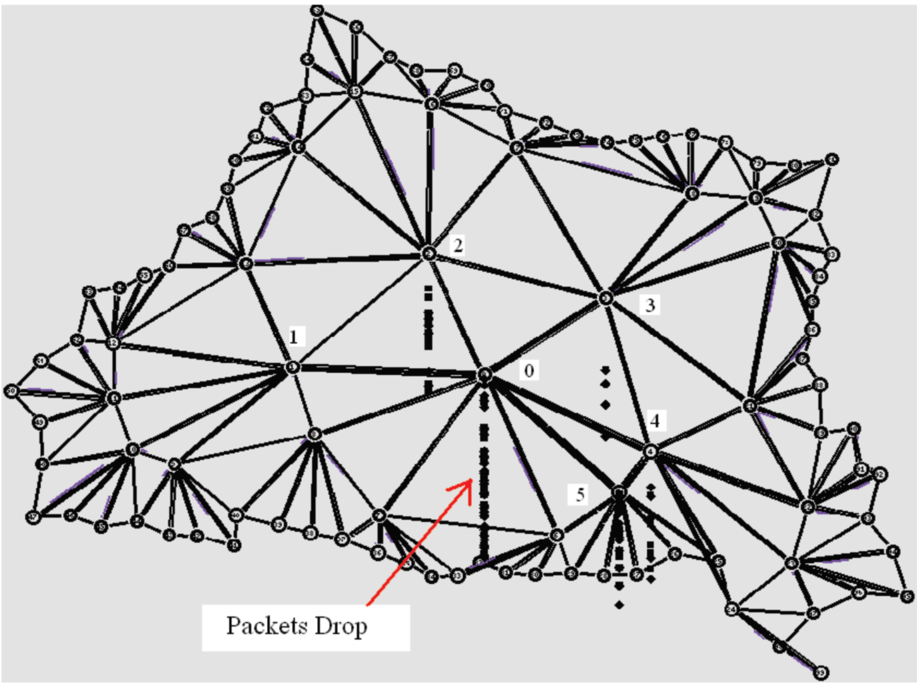


Figure 9: Snapshot of the visualization of network with node degree 8 and its packet loss using `ns-2` Nam.

So, in here, we propose a curvature based load-balancing algorithm. The system diagram is shown in Fig. 10. The curvature κ of the network is used to control a switch. If the network is nonnegatively curved, $\kappa \geq 0$, the weight of the edges in the network is the administrative distance; and the shortest path is calculated based on that. Therefore, traditional load-balancing is used as we mentioned above. If the network is negatively curved, the weight of the edge between two directly connected vertex v_i and vertex v_j is reassigned to be:

$$\tilde{w}(v_iv_j) = \underbrace{\left(\sum_k d^2(v_k, v_i)\right)^{-1}}_{u(v_i)} w(v_iv_j) \underbrace{\left(\sum_j d^2(v_j, v_k)\right)^{-1}}_{u(v_j)} \tag{3}$$

where d is the administrative distance.

A modified graph is generated with these edge weights instead of the administrative distances. The curvature will be positive in this modified graph since $\chi = 2 > 0$. The inertia distribution will be tend to be flat since the edges close to minimum inertia vertices (with heaviest traffic) of the original graph are assigned larger weights to increase the inertia so that the routing curbs the traffic along those edges. Fig. 11 compares the traffic distribution with and without the curvature based load-balancing. In this experiment, we use the node degree 7 network. The heaviest traffic drops from 3340 to 1756 after the curvature based load-balancing. It is a 47% decrease. Since the paths have to be detoured from the congestion vertex through extra routers, it will cause an increase of the total traffic in the network. The total traffic with the load-balancing is 69126 compared with 53964 without the load-balancing. It is a 28% increase. Fig. 12 compares the typical routings with and without load-balancing.

Table 1: In load balancing [7], the asterisk (*) keeps rotating among equal cost paths.

```
M2515-B# show ip route 1.0.0.0
Routing entry for 1.0.0.0/8
   Known via "rip", distance 120, metric 1
   Redistributing via rip
   Advertised by rip (self originated)
   Last update from 192.168.75.7 on Serial1, 00:00:00 ago
   Routing Descriptor Blocks:
   *  192.168.57.7, from 192.168.57.7, 00:00:18 ago, via Serial0
         Route metric is 1, traffic share count is 1
       192.168.75.7, from 192.168.75.7, 00:00:00 ago, via Serial1
         Route metric is 1, traffic share count is 1
```

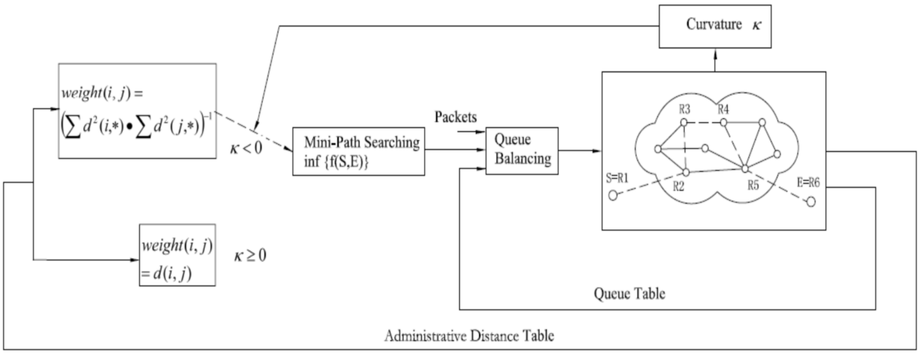


Figure 10: The system level diagram of curvature based load balancing.

5.3 Load Balancing by Yamabe Flow

The link weight reassignment (3) fundamentally smoothes over the inertia of the graph and, as a corollary of the various conjectures and results, alleviates the congestion by distributing the traffic more uniformly. From a deeper mathematical viewpoint, the new link weight $\tilde{w}$ is in fact a *conformal transformation* [16] of the original weight w. The *combinatorial Yamabe flow* [16] is a refined procedure that iterates on the conformal factor $u : V \to \mathbb{R}^+$ to produce, subject to no obstruction, a metric of uniformly positive curvature. More specifically, the ***combinatorial Yamabe flow*** on a triangulated surface is the system of ODE's

$$\frac{du(v_i,t)}{dt} = -K_{u*d}(v_i)u(v_i,t), \quad u(v_i,0) = 1$$

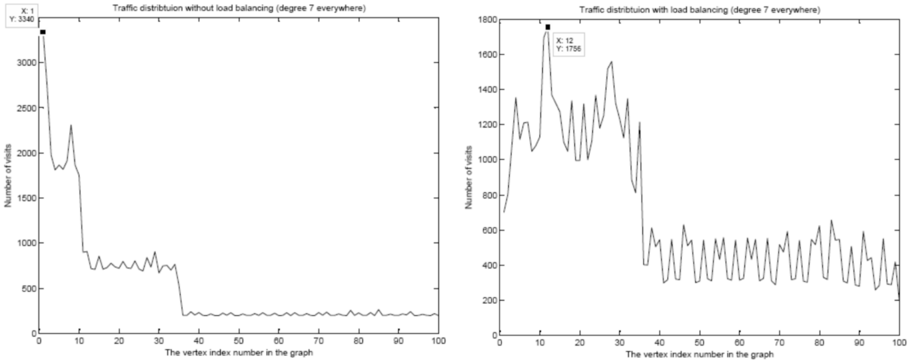


Figure 11: Traffic distribution of node degree 7 network; left: without load-balancing; right: with load-balancing.

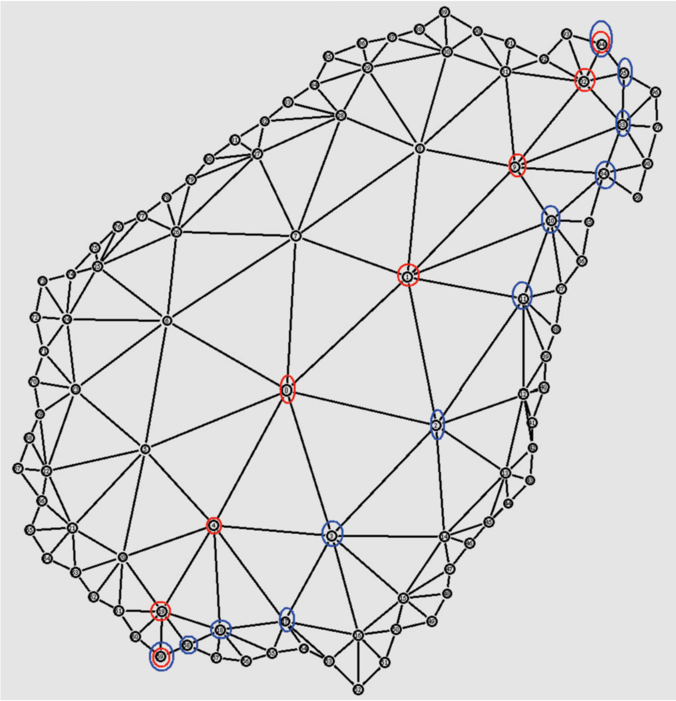


Figure 12: Routing with and without the curvature-based load balancing (Red circle: without load balancing; Blue circle: with load balancing.)

where $u(v_i, t)$ is the *conformal factor* associated with the vertex v_i at time t, $u * d$ is the *conformally modified administrative distance* defined as $u * w(v_i v_j) = u(v_i)w(v_i v_j)u(v_j)$, and $K_{u*d}(v_i)$ is the combinatorial curvature (the numerator of (1)) at v_i for the metric $u * w$. By the piecewise linear (PL) Gauss-Bonnet theorem, $\sum_i K(v_i) = 2\pi\chi$, where $\chi = |F| - |E| + |V|$ is the Euler characteristic. Thus a metric of uniformly positive curvature exists only if $\chi > 0$, which is the case for the triangulation of Fig. 12. On such a triangulation, the Yamabe flow will converge to a metric of constant positive curvature, unless it reaches a removable singularity. The latter is a degenerate triangle $\Delta v_i v_j v_k$ of the triangulation, that is, $u * w(v_i v_j) + u * w(v_j v_k) = u * w(v_i v_k)$. This singularity is easily removed by deleting the link $v_i v_k$. Thus, to alleviate congestion, some links might have to be removed, a phenomenon otherwise referred to as *Braess paradox*.

6 Conclusion and Future Work

We have proposed to utilize coarse geometry concepts to analyze the traffic pattern in networks, especially in negatively curved networks. We have found that the Alexandrov angles provide relevant curvature information, consistently with the Gauss concept. The latter provides the quintessence of the topological structure of a network. Networks with different curvatures have drastically different behaviors as far as traffic, random walk, percolation processes, etc. are concerned. Negatively curved networks are prone to congestion. Because of the pervasive implication of negative curvature, the congestion cannot be really alleviated, unless more drastic action—curvature control—is implemented by a Yamabe-like scheme. The remaining challenge is to implement the Yamabe flow in some flooding scheme.

7 Acknowledgment

This research was partially supported by National Science Foundation grant NetSE 1017881.

Bibliography

[1] F. Ariaei and E. Jonckheere, Cooperative 'curvature-driven' control of mobile autonomous sensor agent network, *IEEE Conference on Decision and Control*, New Orleans, LA, December 2007, pp. 1453-1458.

[2] Y. Baryshnikov, On the curvature of the Internet, in *Workshop on Stochastic Geometry and Teletraffic*, Eindhoven, The Netherlands, April 2002.

[3] L. M. Blumenthal, *Theory and Applications of Distance Geometry,* Oxford at the Clarendon Press, London, 1953.

[4] M. Boguna, F. Papadopoulos and D. Krioukov, Sustaining the Internet with hyperbolic mapping, *Nature Communications*, volume 1, article number 62, September 2010.

[5] M. Bonk and O. Schramm, Embeddings of Gromov hyperbolic spaces, *Geom. Funct. Analysis*, volume 10, pp. 266-306, 2000.

[6] D. Burago, Y. Burago, and S. Ivanov, *A Course in Metric Geometry*, volume 33 of Graduate Study in Mathematics, American Mathematical Society, Providence, Rhode Island, 2001.

[7] Cisco, How Does Load Balancing Work? *Cisco Document* ID: 5212,2005. http://www.cisco.com/en/US/tech/tk365/technologies_tech_note09186a0080094820.shtml.

[8] E. Ghys and P. de la Harpe, editors. Sur les groupes hyperboliques d'après Mikhael Gromov, Number 83 in *Progress in Mathematics*, Birkhauser, Boston, MA, 1990. (Papers from the Swiss Seminar on Hyperbolic Groups held in Bern, 1988.)

[9] M. Gromov, Hyperbolic groups, in S. M. Gersten, editor, *Essays in Group Theory,* volume 8 of Mathematical Sciences Research Institute Publication, pages 75-263, Springer-Verlag, New York, 1987.

[10] M. Gromov, *Metric Structures for Riemannian and Non-Riemannian Spaces,* volume 152 of Progress in Mathematics, Springer-Verlag, 1999.

[11] S. D. Fisher, *Function Theory on Planar Domains: A Second Course in Complex Analysis,* John Wiley & Sons, New York, 1983.

[12] J. Jost, *Nonpositive Curvature: Geometric and Analytic Aspects,* Birkhauser, Lectures in Mathematics, Basel-Boston-Berlin, 1997.

[13] J. Jost, *Riemannian Geometry and Geometric Analysis,* Second Edition, Springer, Universitext, Berlin, Heidelberg, New York, 1998.

[14] M. Lou and E. A. Jonckheere, Tracking and mitigating piracy, *American Control Conference,* Minneapolis, MN, June 14-16, 2006, paper WeA20.2, Session: Networks and Control, pp. 656–661.

[15] M. Lou, Traffic pattern analysis in negatively curved network, Ph.D. dissertation, Department of Electrical Engineering, University of Southern California, Los Angeles, Calif, USA, 2008.

[16] F. Luo, Combinatorial Yamabe flow on surfaces, *Communications in Contemporary Mathematics,* volume 6, number 5, pp. 765-780, 2004.

[17] L. Sun and X. Yu, Positively curved cubic plane graphs are finite, *Journal of Graph Theory,* volume 47, number 4, pp. 241-274, 2004.

Int. J. of Computers, Communications & Control, ISSN 1841-9836, E-ISSN 1841-9844
Vol. VI (2011), No. 1 (March), pp. 150-157

Intelligent Management of the Cryptographic Keys

G. Moise, O. Cangea

Gabriela Moise, Otilia Cangea
Petroleum-Gas University of Ploiesti
Romania, 100680 Ploiesti, 39 Bvd. Bucuresti
E-mail: {gmoise,ocangea}@upg-ploiesti.ro

Abstract: With the continuous development of the computers networks, new problems have been posed in the process of keys management in the cryptographic systems. The main element in the cryptographic technologies is the keys management, as the cryptographic algorithms are known, while the keys have to be either secret (for unauthorized users that do not need them), or public (for users that need them). With an efficient cryptographic keys management system and the existing encryption techniques, there may be implemented a proper security system in the informational systems of the organizations. The process of cryptographic keys management consists in the following operations: keys generation, distribution, update, revocation, storage, backup/ recovery, import and export, usage control, expiration, and destruction. The cryptographic keys management techniques depend on the type of the keys, i.e. symmetric or public. Nowadays, the efforts of the researches in the cryptographic keys management are focused on the standardization and interoperability of the keys management. In this paper, the authors analyze the existing keys management systems and standards available for the keys management techniques, emphasizing the advantages and disadvantages of different systems. They also propose a cryptographic keys management model based on the ideas and principles of the INTERRAP architecture (a conceptual model developed by Jőrg Műller for intelligent agents). Also, there are incorporated some intelligent techniques to manage emergency situations, such as keys losing or their improper usage.

Keywords: cryptographic key management, intelligent agents, key management model.

1 Introduction

The key management is the core of a cryptographic system. The processes related to the key management consist in generation, distribution, update, revocation, storage, backup/recovery, import and export, usage control, expiration, and destruction of the cryptographic keys. Practically, the security of the information is assured by keeping secret the private cryptographic keys. Key management consists in a set of protocols that enable to establish and maintain the keying relationships between the entities of a network [6]. The concept of "keying relationship"is defined in [6] as the state wherein parties of the cryptosystems share keying material. According to the type of cryptographic algorithm used in a cryptosystem, there are two situations: key management used in a symmetric cryptosystem and key management used in an asymmetric cryptosystem. In the former case, the sender and the receiver share the same secret key or two keys computationally feasible and in the latter case, there are involved two transformations: one to generate the public key and the other to generate the private key. [6] [11] The techniques used to distribute confidential keys are: key layering, key translation center, and symmetric key certificate techniques. Key layering comprises the following techniques: master key, key encrypting keys and data keys. *Key Translation Center* (*KTC*) consists in a trusted server, which allows two entities to establish a secure communication using long-term keys. Techniques used to distribute public keys are: point to point delivery over a trusted channel, direct access to a trusted public file (public-key registry), use of an online trusted server, use of an off-line server and certificates, and use of systems implicitly guaranteeing authenticity of public parameters [6]. The advantages of using the keys management in the situation of a public key are: use of a simple key management, on-line trusted server not-required and enhancing the functionality of the system.

In this paper, there is studied the problem of the cryptographic key management in large distributed systems, more specifically, the problems of keys distribution and generation. The main concepts used in this paper are the security domain and the keys graph. The concept of security domain is defined in [7] as "a collection of systems (servers, devices, and so on) that share a common set of keys and are attached to an administered network". In this paper the concept is used in the sense of a collection of entities (to allow an abstractive interpretation) which share a private key. The concept of key graph was introduced in [10] as an arrangement of the keys into a hierarchy and a key server manages all keys. A particular keys hierarchy is the keys tree, which enables to define key management scheme. In this paper it is proposed an intelligent key management model suited to the structure of the network.

The new ideas introduced in this paper are: combining the behaviour agent architecture into the key distribution problem and distribution based on *CRT* (*Chinese Reminder Theorem* [13]).

The paper is structured as follows:

- formalism of the key management problem, that studies the existing key management systems and introduces the concept of security domain graph;

- backgrounds of the intelligent key management model, referring to the *Chinese Remainder Theorem*, as an important calculation algorithm used in order to generate the key management model;

- intelligent key management, proposing a model of the cryptographic key management that may be used in a *SDG* type architecture, based on the principles of the INTERRAP architecture, introducing intelligent agents responsible with the key management in the cryptographic system;

- conclusions, that emphasize the importance of the key management and the advantages offered by the proposed model.

2 Formalism of the key management problem

To formalize the problem of keys management in a computers communication system, there are defined the security domain and a partially ordered relation between security domains. A security domain is a set of the entities (users, data, hardware devices, etc.) which share the same secret key. So, it can be established an equivalence relation between two entities e_1, e_2, according to the following statement:

$e_1 \equiv e_2$ if e_1, e_2 share the same secret key.

The equivalence relation between two entities produces equivalence classes, called security domains.

Let us consider n security domains $SD=\{SD_1,SD_2, ..., SD_n\}$. The set SD contains all the entities of the computer communications system. On the set of the security domains, it can be defined

a binary partially-ordered relationship, using the operator $\prec$.
The relation $SD_j \prec SD_i$ means that the entities of the security domain SD_i have a security clearance higher or equal than that of the entities of the security domain SD_j. For example, the entities from SD_i can decrypt the message received from the entities that belong to SD_j. Also, it is used the expression that the security domain SD_i dominates the security domain SD_j. In this way, one may determine a partially ordered set $(SD, \prec)$, shortly called poset. Messages (data, plain texts) from the security class SD_i are encrypted with the cryptographic key sk_i and data from the security domain SD_j are encrypted with the cryptographic key sk_j. If there is the relation $SD_j \prec SD_i$, the entities of the security domain SD_i have the right to decrypt the cipher text using the cryptographic key sk_i. In contrast, the entities which are parts of the security domains, SD_j , cannot decrypt the messages received from the entities of the security domain SD_i . Also, if the following relations between three security domains $SD_k \prec SD_j$ and $SD_j \prec SD_i$, are true, that means that the entities of SD_j can decrypt the cipher texts received from the entities of SD_k and the entities of SD_i can decrypt the cipher texts received from the entities of SD_j . Consequently, the entities of SD_i can decrypt the cipher texts received directly from the entities of SD_k.
In this manner, there are generated domains hierarchies. A Hasse diagram can represent a poset. This diagram is called in [1] a security class privilege graph *(SCPG)*, in this paper it is used the term of security domain graph *(SDG)*. An example of *SDG* is shown in figure 1 (a - security domains tree *(SDT)*, b - a general security domain graph *(GSDG))*.

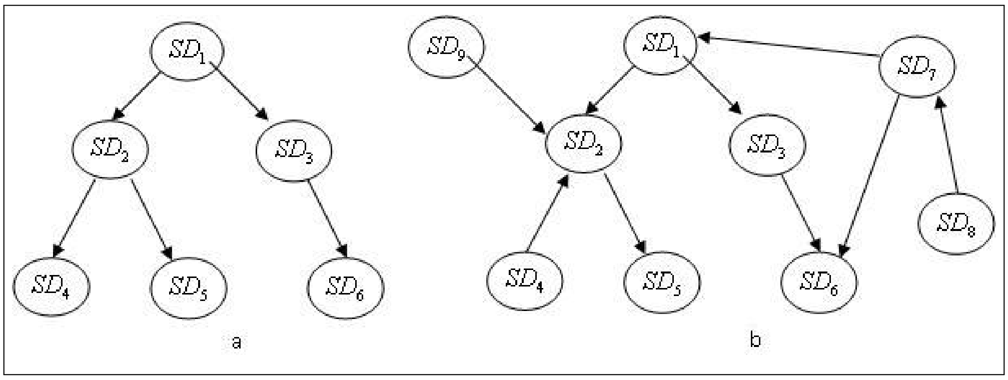


Figure 1: Security Domain Graph

where the following representations have the similar sense (figure 2)
The most simple keys management model assumes the existence of a keys server and if an entity

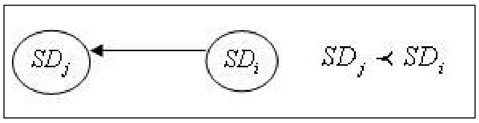


Figure 2: Graphical representation of a partially-ordered relationship between two security domains

needs to decrypt a message, it has to ask the proper key over a secure channel, or each security domain has to store all successors secret keys. These models are not really feasible, because the tendencies of increasing the networks dimensions require large storage spaces and a lot of secure channels. In this way, there is quite a challenge to define a keys management model properly adjusted to the security domain graph.

3 Backgrounds of the Intelligent Key Management Model

There were proposed a lot of cryptographic keys management architecture. To implement Data Encryption Standard, IBM proposed a Key Management Scheme for DES in the 70' years. The cryptographic keys management architecture consists in cryptographic systems connected via a communications network. Each cryptographic system has a cryptographic facility, a cryptographic key data set, a cryptographic facility access program and is using application programs. One solution based on control vector is proposed in [5]. The scheme uses a control vectors which facilities the implementation of owner key management policy and rules. This technique enables key distribution in different environments: peer-to-peer distribution, key distribution center, and key translation center. A list of keys management architecture can be found at [8].

Akl and Taylor proposed the first cryptographic keys assignment scheme to solve problems related to access control in hierarchies (AT scheme). According to AT scheme [1], each security class (the security class contains data and users with the same rights) has associated a secret key and a public parameter. For a relation $SC_j \prec SC_i$, SC_i use the public parameter T_j and the secret key k_i to derive the secret key k_j. The secret key k_i is computed according the formula $k_i = k_0^{T_i}(modM)$, where k_0 is the secret key of the *Central Authority* and M is the product of two secret large prime numbers. T_i is a public parameter with the following property: $SC_j \prec SC_i$, if T_j is a multiple of T_i. In order to generate the public parameters, for T_i it is used the formula $T_i = \prod\limits_{SC_i NOT \succ SC_k} p_k$, where p_k are prime numbers related to each security class.The major inconvenient of the method is the fact that the value of T_i increases and will become impractical (figure 3). Another problem is represented by the question: is the arrangement of the security domains according to the *SDG*? (that is, when a security domain has more than one parent).

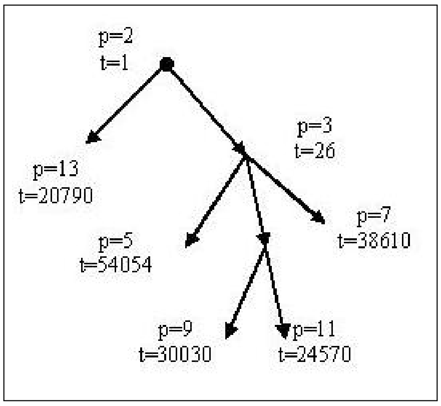


Figure 3: The diagram of generating the T parameter in an AT scheme

In order to generate a model for the key management, the authors studied the *Chinese Remainder Theorem* that is an ancient, but important calculation algorithm in modular arithmetic. It enables one to solve simultaneous equations with respect to different moduli in a considerable generality. Here is the statement of the problem that the *Chinese Remainder Theorem* solves.

Theorem 1. *Chinese Remainder Theorem [12]. Let*

m_1, m_2, ..., $m_k \in Z$ with $gcd(m_i, m_j)=1$,any $i,j=\overline{1,k}$, $i \neq j$. Let m be the product
$m=m_1 \times m_2 \times ... m_k$. Let a_1, a_2, ...,$a_k \in Z$. Consider the system:

$$x \equiv a_1 (mod\ m_1)$$
$$x \equiv a_2 (mod\ m_2)$$
$$\ldots$$
$$x \equiv a_k (mod\ m_k)$$

Then there exists exactly one $x \in Z_m$ *solution of the system.*

The solution to the system above may be obtained using the following algorithm:
Step 1 For each to i=1 to k calculate $z_i = m_1 \times m_2 \times \ldots \times m_{i-1} \times m_{i+1} \times \ldots m_k$
Step 2 For each to i=1 to k calculate $y_i = z_i^{-1} (mod\ m_i)$
Step 3 Calculate $x = a_1 \times y_1 \times z_1 + \ldots + a_k \times y_k \times z_k$, Return x.

4 Intelligent Key Management

In this paper it is proposed a model of cryptographic keys management that may be used in the architecture of *SDG* type. Each entity of the security domains has associated an intelligent agent *(SDKMA)* responsible with the cryptographic key management in the system (figure 4). where $SD_i = e_{i1} \mathrm{U} e_{i2} \mathrm{U} e_{i3} \ldots$, where e_{ij} use k_i and each SD_i is organized according to the schema shown in figure 5.

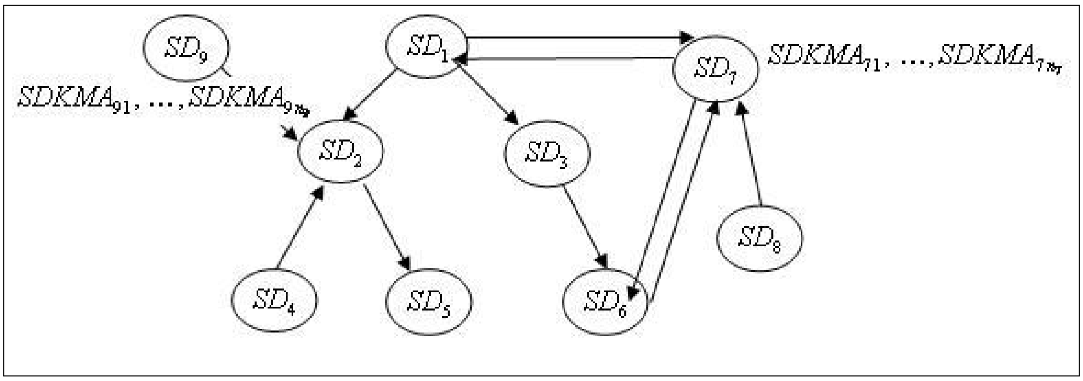

SD_9
SD_1
SD_7
$SDKMA_{71}, \ldots, SDKMA_{7n_7}$
$SDKMA_{91}, \ldots, SDKMA_{9n_9}$
SD_2
SD_3
SD_8
SD_4
SD_5
SD_6

Figure 4: Security Domains Graph and Intelligent Key Management Agents

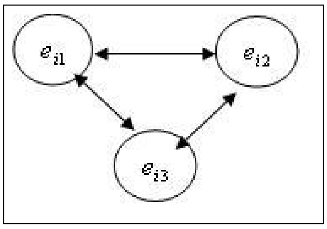


Figure 5: The entities relations in a security domain

Each *SDKMA* is structured according the INTERRAP intelligent agent architecture defined by Műller [9]. INTERRAP architecture is a layered BDI model (belief-desire-intention model) with three layers: behavior based layer, local planning layer, and cooperative planning layer. The behavior based layer contains the reactivity and procedural knowledge used in routine tasks, the local planning layer provides reasoning to realize the local tasks and to produce goal oriented behaviors, and the cooperative planning layer enables and facilities collaborative work with other agents. The structure of *SDKMA* is presented in figure 6.

Behavior planning layer acts in the emergency situations (renewal keys, delete entity, add entity, destroy keys). Local planning layer manages the cryptographic keys within the framework of the security domain. Cooperative planning layer manages the cryptographic keys between security domains.
World KB contains the procedures and functions used in the emergency situations (structure

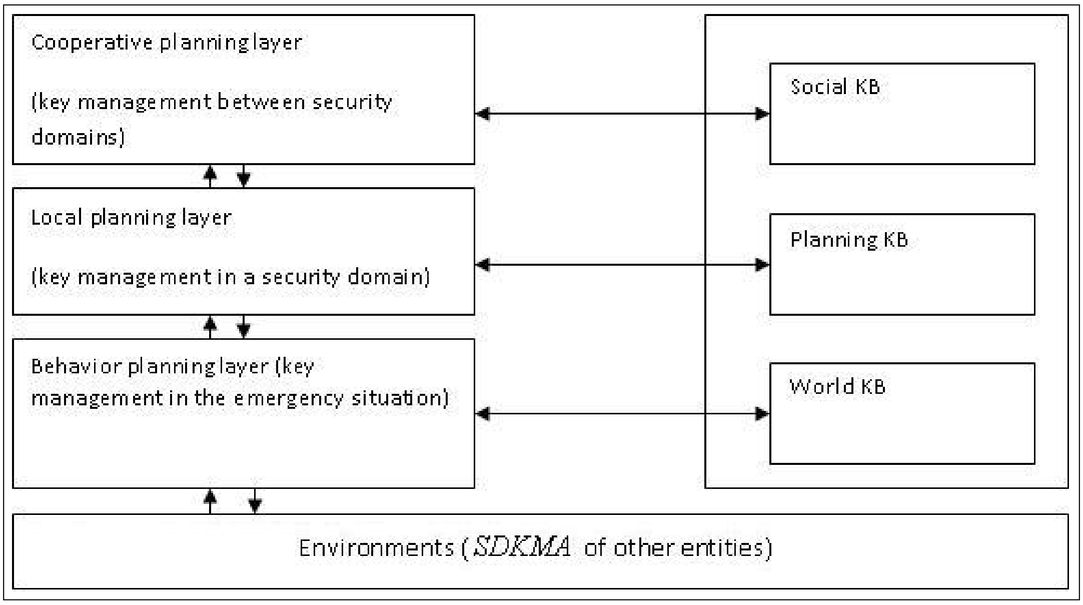

Cooperative planning layer
(key management between security domains)
Local planning layer
(key management in a security domain)
Behavior planning layer (key management in the emergency situation)
Social KB
Planning KB
World KB
Environments (SDKMA of other entities)

Figure 6: The structure of SDKMA

of the networks and security domains, new arrivals and leaving of the entities). Planning KB contains procedures and functions responsible with the key management within the framework of the security domain and social KB contains procedures and functions used to key management between security domains. These knowledge bases are self-updated; therefore, the keys management model has the property of flexibility that is modifications in the structure of the network cause modifications in the key management model. Keys management within the framework of security domains can be realized according to key server schema. For each security domains, it is selected, in a randomized way, an entity. Its *SDKMA* will play the role of key server. The server stores the key and the associated list of the entities from the security domain. Entities request the key to the server. Upon entity authentication, the server sends the keying material if the entity is authorized. Key management between security domains can be realized according toAT scheme if the node corresponding to the security domain has one parent. The arrangement in security domains offers the advantage of calculating not greater values of parameters T. There were provided solutions to optimize the AT schema in order to compute smaller values of parameters T [4]. If the relations between security domains are according to the scheme presented in figure 7, then one has the following situation: a node has more than one parents and a node has more than two ancestors.

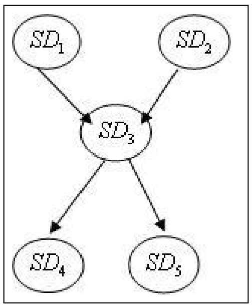


Figure 7: Security domain SD3 with two parents and two ancestors

If an entity of SD_3 broadcasts a message m, than the entities of SD_1 and SD_2 can decrypt the

cipher text c. So, the entities have to know a single decryption key. The asymmetric keys generator has to send to each security domains a private key or each *SDKMA* can compute (cooperative planning layer) the private key using the *Chinese Remainder Theorem.* For each entity (SD_1 and SD_2) it will select a large secret prime number n_1, respectively n_2. The secret key is the solution (in the space $\mathrm{Z}_{n1\times n2}$) of the system:
$x\equiv a_1(mod\ n_1)$
$x\equiv a_2(mod\ n_2)$, where a_1 , a_2 are public parameters.
To calculate the solution of the system, it can be used the Lagrange method:
Step 1 Find cofactors u_{12}, u_{21} using Extended Euclidian algorithm
Step 2 Calculate $l_1=u_{21}\times m_2$, $l_2=u_{12}\times m_1$
Step 3 Calculate the solution $x=a_1\times l_2+a_2\times l_2$

If one entity of SD_4 and one entity of SD_5 broadcast in the same time two encrypted messages c_4 and c_5, the *SDKMA* (the cooperative planning layer) of each entity of SD_3 manage a waiting queue and decrypt the messages using randomized priorities.
The advantages offered by the proposed model are: more securely, because the intelligent property changes any time the parameters used in keys generation, and more efficiently, through the use of a security domain based structure of the network. The dynamism of the network can be easily implemented because the knowledge bases related to each layer of the *SDKMA* are updated with actual information in any moment. Also, in order to generate the keys, the procedures and functions may contain different algorithms and the *SDKMA* can change any time the key distribution algorithms according to the dimensions of the networks or the necessary security level.

5 Conclusions

Starting from the analysis of the existing key management systems and standards available for the keys management techniques, in this paper it is proposed an intelligent cryptographic key management model between security domains, *SDKMA* type, based on the ideas and principles of the INTERRAP architecture, emphasizing the advantages referring to security, efficiency and feasibility. As future directions, one may consider designing a fuzzy model of the cryptographic key management system, but only if this approach would bring important improvements for the model, regarding to the procedures and functions used to generate the keys. Nevertheless, the most secure situation is to hold the keys in secure hardware and perform all processing there [3], being impossible to achieve this goal in large scale networks.

Bibliography

[1] Akl, S.G., Taylor, P.D., Cryptographic solution to a problem of access control in a hierarchy,*ACM Transactions on Computer System*,3 (1), 1983.

[2] Hassen, R. H., Bouabdallah A., Bettahar, H., Challal, Y., Key management for content access control in a hierarchy, *Computer Networks*, 51 3197-3219, 2007.

[3] Lin, J. C., Huang, K. H., Lai, L., Lee, H. C., Secure and efficient group key management with shared key derivation, it Computer Standards and Interfaces, 31, 2009.

[4] MacKinnon, S. , Taylor, P., Meijer, H., Akl, S., An optimal algorithm for assigning cryptographic keys to control access in a hierarchy, *IEEE Transactions on Computers*, C-34 (9), 1985.

[5] Matyas, S. M., Le, A.V. Abraham, D. G., A Key-Management Scheme Based on Control Vectors, *IBM Systems Journal*, Vol. 2, Issue 3, 1991.

[6] Menezes, A., van Oorschot, P., Vanstone, S., Handbook of Applied Cryptography, *CRC Press*, 1996.

[7] Michener, J. R., Acar, T., Security Domains: Key Management in Large-Scale Systems,*IEEE SOFTWARE*, 2000.

[8] Savard, J. J. G., A Cryptographic Compendium, http://www.quadibloc.com/crypto/jscrypt.htm, accessed on the December 5th, 2009.

[9] Müller, J. P., The Design of Intelligent Agents: A Layered Approach. Lecture notes in computer science, *Lecture notes in artificial intelligence*, 1177, Springer-Verlag, 1996.

[10] Wong, C.K., Gouda, M., Lam, S., Secure groups communication using key graphs,*Proceedings of the ACM SIGCOMM'98*, 1998.

[11] Key management in cryptography, http://www.netlab.tkk.fi/opetus/s38153/k2003/Lectures/g33keymgmt.ppt, accessed on December 10, 2009.

[12] Chinese Remainder Theorem, http://www.math.tamu.edu/~jon.pitts/courses/2005c/470/supplements/chinese.pdf, accessed on December 10, 2009.

[13] Zhou, J., Ou, O. H., Key Tree and Chinese Remainder Theorem Based Group Key Distribution Scheme, *Proceedings of the 9th International Conference on Algorithms and Architectures for Parallel Processing*, ISBN:978-3-642-03094-9, 2009.

Int. J. of Computers, Communications & Control, ISSN 1841-9836, E-ISSN 1841-9844
Vol. VI (2011), No. 1 (March), pp. 158-165

Heuristic Algorithms for Solving the Generalized Vehicle Routing Problem

P.C. Pop, C. Pop Sitar, I. Zelina, V. Lupşe, C. Chira

Petrică Claudiu Pop, Ioana Zelina, Vasile Lupşe
North University of Baia Mare, Department of Mathematics and Informatics
Romania, V. Babeş , 430083, Baia Mare
E-mail: petrica.pop@ubm.ro, ioanazelina@yahoo.com, vasilelupse@yahoo.co.uk

Corina Pop Sitar
North University of Baia Mare, Department of Economics
Romania, V. Babeş , 430083, Baia Mare
E-mail: sitarcorina@yahoo.com

Camelia Chira
"Babes-Bolyai" University of Cluj-Napoca
Romania, M. Kogalniceanu, 400084, Cluj-Napoca
E-mail: cchira@cs.ubbcluj.ro

Abstract: The vehicle routing problem (VRP) is one of the most famous combinatorial optimization problems and has been intensively studied due to the many practical applications in the field of distribution, collection, logistics, etc.

We study a generalization of the VRP called the generalized vehicle routing problem (GVRP) where given a partition of the nodes of the graph into node sets we want to find the optimal routes from the given depot to the number of predefined clusters which include exactly one node from each cluster. The purpose of this paper is to present heuristic algorithms to solve this problem approximately. We present constructive algorithms and local search algorithms for solving the generalized vehicle routing problem.

Keywords: network design, combinatorial optimization, generalized vehicle routing problem, heuristic algorithms.

1 Introduction

Combinatorial optimization is a lively field of applied mathematics, combining techniques from combinatorics, linear programming, and the theory of algorithms, to solve optimization problems over discrete structures. The study of combinatorial optimization owes its existence to the advent of modern digital computer. Most currently accepted methods of solution to combinatorial optimization problems would hardly have been taking seriously 30 years ago, for the simple reason that no one could have carried out the computations involved. Moreover, the existence of digital computers has also created a multitude of technical problems of a combinatorial character.

Combinatorial optimization problems can be generalized in a natural way by considering a related problem relative to a given partition of the nodes of the graph into node sets, while the feasibility constraints are expressed in terms of the clusters. In this way, it is introduced the class of generalized combinatorial optimization problems. In the literature one finds generalized problems such as the generalized minimum spanning tree problem [15], the generalized traveling salesman problem, the generalized vehicle routing problem, the generalized (subset) assignment problem, etc. These generalized problems belong to the class of NP-complete problems, are harder than the classical ones and nowadays are intensively studied due to the interesting properties and applications in the real world, even though many practitioners are reluctant to use them for practical modeling problems because of the complexity of finding optimal or near-optimal solutions.

The Generalized Vehicle Routing Problem (GVRP) is an extension of the Vehicle Routing Problem (VRP) and was introduced by Ghiani and Improta [4]. The GVRP is the problem of designing optimal delivery or collection routes, subject to capacity restrictions, from a given depot to a number of predefined, mutually exclusive and exhaustive node-sets (clusters). The GVRP can be viewed as a particular type of location-routing problem (see, e.g. Laporte [7], Nagy and Salhi [10]) for which several algorithms, mostly heuristics, exist.

Ghiani and Improta [4] showed that the problem can be transformed into a capacitated arc routing problem (CARP) and Baldacci et al. [1] proved that the reverse transformation is valid. Recently, Pop [14] provided a new efficient transformation of the GVRP into the classical vehicle routing problem (VRP). In 2003, Kara and Bektas [5] proposed an integer programming formulation for GVRP with a polynomially increasing number of binary variables and constraints and in 2008 Kara and Pop [6] presented two integer linear programming formulations for GVRP with $O(n^2)$ binary variables and $O(n^2)$ constraints, where n is the number of customers which are partitioned into a given number of clusters. As far as we know, the only specific algorithm for solving the GVRP was developed by Pop et al. [13] and was based on ant colony optimization.

The complexity of obtaining optimum or even near-optimal solutions for the generalized combinatorial optimization problems may lead to the development of:

- efficient transformations of the generalized combinatorial optimization problems into classical combinatorial optimization problems [4, 13];
- heuristic and metaheuristic algorithms [11].

The aim of this paper is to describe three classes of heuristic algorithms for solving approximately the generalized vehicle routing problem.

2 Definition of the GVRP

Let $G = (V, A)$ be a directed graph with $V = \{0, 1, 2, ..., n\}$ as the set of vertices and the set of arcs $A = \{(i, j) \mid i, j \in V, i \neq j\}$. A nonnegative cost c_{ij} associated with each arc $(i, j) \in A$. The set of vertices (nodes) is partitioned into $k+1$ mutually exclusive nonempty subsets, called clusters, $V_0, V_1, ..., V_k$ (i.e. $V = V_0 \cup V_1 \cup ... \cup V_k$ and $V_l \cap V_p = \emptyset$ for all $l, p \in \{0, 1, ..., k\}$ and $l \neq p$). The cluster V_0 has only one vertex 0, which represents the depot, and remaining n nodes belonging to the remaining k clusters represent geographically dispersed customers. Each customer has a certain amount of demand and the total demand of each cluster can be satisfied via any of its nodes. There exist m identical vehicles, each with a capacity Q.

The generalized vehicle routing problem (GVRP) consists in finding the minimum total cost tours of starting and ending at the depot, such that each cluster should be visited by exactly once, the entering and leaving nodes of each cluster is the same and the sum of all the demands of any tour (route) does not exceed the capacity of the vehicle Q. An illustrative scheme of the GVRP and a feasible tour is shown in the next figure.

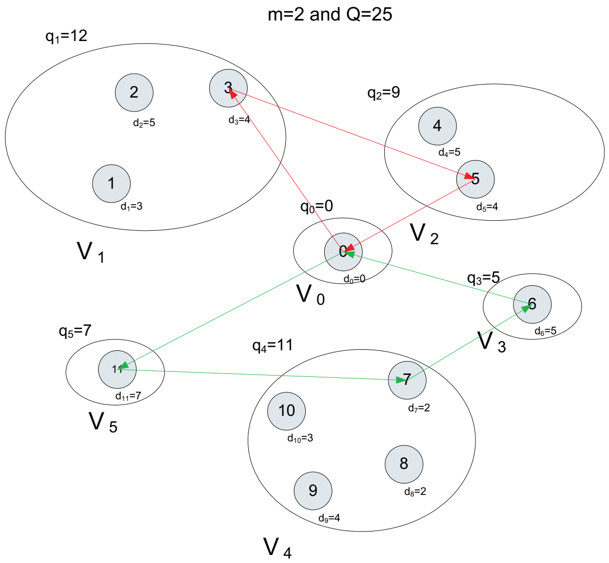


Figure 1 An example of a feasible solution of the GVRP

In Figure 1, it is presented a feasible solution consisting of a collection of two tours (routes): 0-3-5-0 and 0-11-7-6-0 satisfying the capacity restrictions and the condition that from each cluster is visited exactly one node. The cost of this feasible solution is obtained summing the costs of the arcs belonging to the selected tours.

The GVRP reduces to the classical Vehicle Routing Problem (VRP) when all the clusters are singletons and to the Generalized Traveling Salesman Problem (GTSP) when $m = 1$ and $Q = \infty$. The GVRP is NP-hard because it includes the generalized traveling salesman problem as a special case when $m = 1$ and $Q = \infty$.

Several real-world situations can be modeled as a GVRP. The post-box collection problem described in Laporte et al. [8] becomes an asymmetric GVRP if more than one vehicle is required. Furthermore, the GVRP is able to model the distribution of goods by sea to a number of customers situated in an archipelago as in Philippines, New Zeeland, Indonesia, Italy, Greece and Croatia. In this application, a number of potential harbours is selected for every island and a fleet of ships is required to visit exactly one harbour for every island. Several applications of the GTSP may be extended naturally to GVRP.

3 Heuristic Algorithms for Solving the GVRP

Two fundamental goals in computer science are finding algorithms with provably good run times and with provably good or optimal solution quality. A heuristic is an algorithm that abandons one or both of these goals; for example, it usually finds pretty good solutions, but there is no proof the solutions could not get arbitrarily bad; or it usually runs reasonably quickly, but there is no argument that this will always be the case. Heuristics are typically used when there is no known method to find an optimal solution, under the given constraints (of time, space etc.) or at all.

Several families of heuristic algorithms have been proposed for the classical VRP, see for example Laporte *et al.* [9]. These can be classified into two main classes: classical heuristics and metaheuristics. Most standard construction and improvement procedures in use belong to the first class. These methods performs a relatively limited exploration of the solution space and generally produce good quality solutions in reasonable computational times.

In what it follows we will provide three classes of heuristic algorithms for solving the GVRP:

- constructive heuristics: Nearest Neighbour and a Clarke-Wright based heuristic;
- improvement heuristics: String Cross (SC), String Exchange (SE), String Relocation (SR) and String Mix (SM);
- a local-global heuristic.

3.1 Constructive heuristics

Nearest Neighbour

Perhaps the most natural heuristic for the GVRP is the famous *Nearest Neighbour* algorithm (NN). In this algorithm the rule is always to go next to the nearest as-yet-unvisited customer subject to the following restrictions: we start from the depot, from each cluster is visited exactly one vertex (customer) and the sum of all the demands of the current tour (route) does not exceed the capacity of the vehicle Q. If the sum of all the demands of a current tour (route) exceeds the capacity of the vehicle then we start again from the depot and visit next the nearest customer from an unvisited yet cluster. If all the clusters are visited, then the algorithm terminates. A collection of routes traversing exactly one city from each cluster in the constructed order represents the output of the algorithm.

The nearest neighbour algorithm is easy to implement and executes quickly, but it can sometimes miss shorter routes, due to its greedy nature. The running time of the described nearest neighbour algorithm is $O(n^2)$.

A Clarke-Wright based heuristic algorithm

The Clarke and Wright [2] savings algorithm is perhaps the most well known heuristic for the VRP. It applies for the problems for which the number of vehicles is a decision variable, and works in the case of directed and undirected problems.

The algorithm in the case of the GVRP works as follows:

Step 1 (Savings computation). For each $i \in V_l$ and $j \in V_p$, where $l \neq p$ and $l, p \in \{1, ..., k\}$ compute the savings:

$$s_{ij} = c_{i0} + c_{0j} - c_{ij}.$$

It is obviously that $s_{ij} \geq 0$ and $s_{ij} = s_{ji}$. We order the savings in a nonincresing fashion.

At the beginning we create k routes denoted $(0, i_l, 0)$, $l \in 1, ..., k$ as follows for each cluster V_l we define $c_{0i_l} = \min\{c_{0j} \mid j \in V_l\}$.

There will be as many routes as the number of clusters and total distance of the routes is:

$$d = c_{0i_1} + c_{0i_2} + ... + c_{0i_k}.$$

Step 2 (Route extension). Consider in turn each route $(0, i, ..., j, 0)$. Determine the first saving s_{ui} or s_{jv} that can feasibly be used to merge the current route with another route ending with $(u, 0)$ or starting with $(0, v)$, for any $u \in V_l$ and $v \in V_p$, where $l \neq p$ and $l, p \in \{1, ..., k\}$ and V_l and V_p are clusters not visited by the route $(0, i, ..., j, 0)$.

Because at a given moment there can exist more feasible route extensions, the priority will have that one that produces the biggest reduction of the total distance of the route.

We implement the merge and repeat this operation to the current route. If no feasible merge exists, consider the next route and reapply the same operations.

Stop when no route merge is feasible.

The Clarke-Wright based algorithm for solving the GVRP is easy to implement and its running time is $O(n^2 \log n)$.

3.2 Improvement heuristics

The improvement heuristics algorithms for the GVRP are based on simple routes modifications and may operate on each vehicle route taken separately, or on a several routes at a time. In the first case, any improvement heuristic for Traveling Salesman Problem (TSP) can be applied, such as 2-OPt, 3-Opt, etc. In the second case, procedures that exploit the multi-route structure of the GVRP can be developed. We can see these improvements as a neighbourhood search process, where each route has an associated neighborhood of adjacent routes.

The heuristics algorithms for the GVRP that we are going to describe are based on the classification of the Van Breedam [16] of the improvement operations as string cross, string exchange, string relocation and string mix.

a) *String cross (SC)*: two strings of vertices are exchanged by crossing two edges of two different routes.

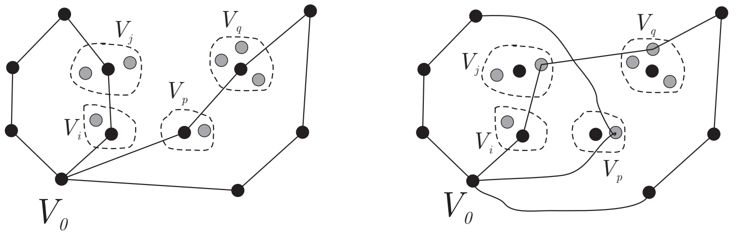


Figure 2 An example of a possible string cross. In the left side are presented the routes before the exchange of the vertices strings and in the right side the routes after the exchange

In the above picture there were presented just the clusters of the string of vertices that are exchanged in order to have a clearer figure. It is important to mention that we investigate all the possible connections of the exchanged vertices within the clusters in order to get improved routes, as is shown in Figure 2: the nodes belonging to the marked clusters after the exchange may be diferrent.

b) *String exchange (SE)*: two strings of at most r vertices are exchanged between two routes.

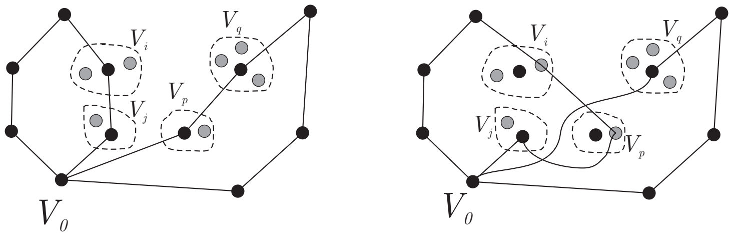


Figure 3 An example of a possible string exchange

c) *String relocation (SR)*: a string of at most k vertices is moved from one route to another ($k = 1$ or $k = 2$).

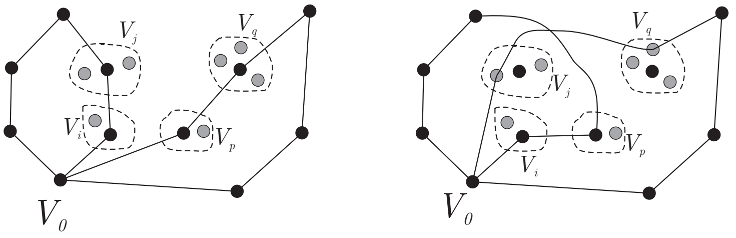


Figure 4 An example of a possible string relocation

d) *String mix (SM)*: consists in selecting the best move between the string exchange and string relocation.

3.3 A local-global heuristic for GVRP

The last heuristic algorithm for solving the GVRP that we are going to describe is based on local-global approach and it aims at distinguishing between *global connections* (connections between clusters) and *local connections* (connections between nodes from different clusters). As we will see, having a global collection of routes connecting the clusters it is rather easy to find the corresponding best (w.r.t. cost minimization) solution of the GVRP.

There are several generalized collection of routes, i.e. routes containing exactly one node from a cluster, corresponding to a global collection of routes. Between these generalized collection of routes there exist one called the best generalized collection of routes (w.r.t. cost minimization) that can be determined either by dynamic programming or by solving an linear integer program.

The local-global approach was applied succesfully to other generalized combinatorial optimization problems such as: generalized minimum spanning tree problem (GMSTP) and generalized traveling salesman problem (GTSP) in order to provide exact exponential time algorithms, strong mixed-integer programming formulations, solution procedures based on these mixed-integer programming formulations and a heuristic algorithm for solving the GMSTP, see [**?**,12].

Let G' be the graph obtained from G after replacing all nodes of a cluster V_i with a supernode representing V_i. We will call the graph G' the global graph. For convenience, we identify V_i with the supernode representing it. Edges of the graph G' are defined between each pair of the graph vertices $V_1, \dots, V_k$.

In the next figure we present the collection of generalized routes corresponding to the a global collection of routes.

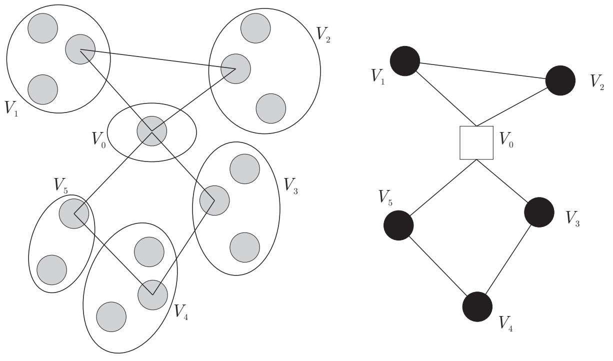


Figure 5 Example showing a generalized collection of routes corresponding to a global collection of routes

Given a global route and a sequence $(V_0, V_{k_1}, ..., V_{k_p})$ in which the clusters are visited, we want to find the best feasible route R^* (w.r.t cost minimization), visiting the clusters according to the given sequence. This can be done in polynomial time, by solving the following shortest path problem as we will describe below.

We construct a layered network, denoted by LN, having $p + 2$ layers corresponding to the clusters $V_0, V_{k_1}, ..., V_{k_p}$ and in addition we duplicate the cluster V_0, containing the vertex denoted 0 and representing the depot. The layered network contains in addition the extra node denoted by $0'$ for each. There is an arc (i, j) for each $i \in V_{k_l}$ and $j \in V_{k_{l+1}}$ $(l = 1, ..., p - 1)$, having the cost c_{ij} and an arc (i, h), $i, h \in V_{k_l}$, $(l = 2, ..., p)$ having cost c_{ih}. Moreover, there is an arc $(i, 0')$ for each $i \in V_{k_p}$ having cost $c_{i0'}$.

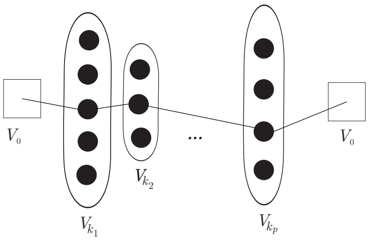


Figure 6 Example showing a route in the constructed layered network LN

We consider paths from 0 to $0'$, that visits exactly one node from each cluster $V_{k_1}, V_{k_2}, ..., V_{k_p}$, hence it gives a feasible route.

Conversely, every route visiting the clusters according to the sequence $(V_0, V_{k_1}, ..., V_{k_p})$ corresponds to a path in the layered network from 0 to $0'$.

Therefore, it follows that the best (w.r.t cost minimization) route R^* visiting the clusters in a given sequence can be found by determining all the shortest paths from 0 to the corresponding $0'$ with the property that visits exactly one node from each of the clusters $(V_{k_1}, V_{k_2}, ..., V_{k_p})$.

The overall time complexity of the above procedure is $O(m + \log n)$, where by m we denoted the number of edges and n number of nodes.

Therefore, given a global collection of routes connecting the clusters we can find efficiently the best corresponding collection of generalized routes. In order to provide global collections of routes we may use any improvement heuristics for the classical VRP.

4 Conclusion and future work

The Generalized Vehicle Routing Problem is an extension of the Vehicle Routing Problem (VRP) and consists in designing optimal delivery or collection routes, subject to capacity restrictions, from a given depot to a number of predefined, mutually exclusive and exhaustive node-sets (clusters). The GVRP is an NP-hard problem and finds many interesting real-world applications.

The aim of this paper was to present three classes of heuristic algorithms: constructive heuristics including Nearest Neighbour and a Clarke-Wright based heuristic; improvement heuristics including String Cross (SC), String Exchange (SE), String Relocation (SR) and String Mix (SM) and a local-global heuristic for the GVRP.

Computational results are planned in order to assess the effectiveness of the proposed heuristic algorithms.

Acknowledgement. This work was partially supported by CNCSIS-UEFISCSU, project number PNII - IDEI 508/2007.

Bibliography

[1] R. Baldacci, E. Bartolini and G. Laporte, *Some applications of the Generalized Vehicle Routing Problem*, Le Cahiers du GERAD, G-2008-82, 2008.

[2] G. Clarke and J.W. Wright, *Scheduling of Vehicles From a Central Depot to a Number of Delivery Points*, Operations Research, Vol. 12, pp. 568-581, 1964.

[3] M. Fischetti, J.J. Salazar and P. Toth, *A branch-and-cut algorithm for the symmetric generalized traveling salesman problem*, Operations Research, 45:378-394, 1997.

[4] G. Ghiani, G. Improta, *An efficient transformation of the generalized vehicle routing problem*, European Journal of Operational Research, Vol. 122, pp. 11-17, 2000.

[5] I. Kara and T. Bektas, *Integer linear programming formulation of the generalized vehicle routing problem*, in Proc. of the 5-th EURO/INFORMS Joint International Meeting, 2003.

[6] I. Kara and P.C. Pop, *New Mathematical Models of the Generalized Vehicle Routing Problem and Extensions*, in Proc. of the International Conference on Applied Mathematical Programming and Modelling, Bratislava, Slovakia, May 27-30, 2008.

[7] G. Laporte, *Location-routing problems*, in B.L. Golden and A.A. Assad (eds.), Vehicle Routing: Methods and Studies, North-Holland, Amsterdam, pp. 163-197, 1988.

[8] G. Laporte, S. Chapleau, P.E. Landry and H. Mercure, *An algorithm for the design of mail box collection routes in urban areas*, Transportation Research B, Vol. 23, pp. 271-280, 1989.

[9] G. Laporte, M. Gendreau, J-Y. Potvin and F. Semet, *Classical and modern heuristics for the vehicle routing problem*, International Transactions in Operational Research, Volume 7, Issue 4-5, pp. 285 - 300, 2006.

[10] G. Nagy and S. Salhi, Location-routing issues: Models and methods, *European Journal of Operational Research*, Vol. 177, pp. 649-672, 2007.

[11] C.M. Pintea, D. Dumitrescu and P.C. Pop, *Combining heuristics and modifying local information to guide ant-based search*, Carpathian Journal of Mathematics, Vol. 24, No. 1, pp. 94-103, 2008.

[12] P.C. Pop, C. Pop Sitar, I. Zelina and I. Tascu, Exact algorithms for generalized combinatorial optimization problems, Lecture Notes in Computer Science, Vol. 4616, pp. 154-162, 2007.

[13] P.C. Pop, C.M. Pintea, I. Zelina and D. Dumitrescu, *Solving the Generalized Vehicle Routing Problem with an ACS-based Algorithm*, American Institute of Physics (AIP), Conference Proceedings: BICS 2008, Vol.1117, No.1, 157–162, 2009.

[14] P.C. Pop, *Efficient Transformations of the Generalized Combinatorial Optimization Problems into Classical Variants*, Proceedings of the 9-th Balkan Conference on Operational Research, Constanta, Romania, 2-6 September 2009.

[15] P.C. Pop, *A survey of different integer programming formulations of the generalized minimum spanning tree problem*, Carpathian Journal of Mathematics, Vol. 25, No. 1, pp. 104-118, 2009.

[16] A. van Breedam, An analysis of the behavior of the heuristics of the vehicle routing problem for a selection of problems, with vehicle-related, customer-related and time-related constraints, Ph.D. Dissertation, University of Antwerp, Belgium, 1994.

Int. J. of Computers, Communications & Control, ISSN 1841-9836, E-ISSN 1841-9844
Vol. VI (2011), No. 1 (March), pp. 166-174

Human Intervention and Interface Design in Automation Systems

P. Ponsa, R. Vilanova, B. Amante

Pere Ponsa (corresponding author)
Automatic Control Department
Technical University of Catalonia
EPSEVG, Av. Víctor Balaguer,
08800 Vilanova i la Geltrú, Barcelona, Spain
E-mail: pedro.ponsa@upc.edu

Ramón Vilanova
Department de Telecomunicació i Enginyeria de Sistemes
Universitat Autònoma de Barcelona
08193, Bellaterra, Spain
E-mail: ramon.vilanova@uab.cat

Beatriz Amante
Project Department
Technical University of Catalonia
ETSEIAT Edifici TR5, C. Colom 11,
08222, Barcelona, Spain
E-mail: beatriz.amante@upc.edu

Abstract: Human-Machine-Interfaces are with no doubt one of the constitutive parts of an automation system. However, it is not till recently that they have received appropriate attention. It is because of a major concern about aspects related to maintenance, safety, achieve operator awareness, etc has been gained. Even there are in the market software solutions that allow for the design of efficient and complex interaction systems, it is not widespread the use of a rational design of the overall interface system, especially for large scale systems where the monitoring and supervision systems may include hundreds of interfacing screens. It is on this respect hat this communication provides an example of such development also by showing how to include the automation level operational modes into the interfacing system. Another important aspect is how the human operator can enter the control loop in different ways, and such interaction needs to be considered as an integral part of the automation procedure as well as the communication of the automation device.In this paper the application of design and operational modes guidelines in automation are presented inside an educational flexible manufacturing system case study.

Keywords: human-automation interaction, process control, display design.

1 Introduction

Automation can refer to open-loop operation on the environment or closed-loop control. And the human intervention adopts diverse possibilities: human in the loop (human intervention), human out of the loop (controller intervention) and human on the loop (supervisory control mode over the controlled process). The basic automation replaces the human manual control by an automatic controller; however in highly automated systems is necessary human beings for supervision, adjustment, maintenance, expansion and improvement. Automation increases complexity. It is difficult to maintain operational skills in a automated environment with the presence of an abnormal situation when the operator is required to intervene. The complexity of industrial human process supervision makes it necessary to supplement the Human-Computer Interaction approach with a cross-disciplinary cooperation in order to integrate knowledge and methods from other fields, especially Automation and Artificial Intelligence [1].

Our view is that complete control systems engineering must encompass all these approaches. This increasing complexity of production systems has also translated to the automation level. The need to face for larger amounts of information and the capability to interact with other subsystems of the production chain requires of the application of appropriate modelling methodologies. On that respect it is worth to mention that, on the academic side, the authors have developed different tools to tackle such problems.

What is proposed in this paper, and presented by its application to a laboratory scale Flexible Manufacturing System (FMS), is a complete integrated approach for the design of the HMI (Human Machine Interface) system. The development follows a top-down approach where the different screens that constitute the overall system are conceived and particular methods are used to ensure, within each one of the designed interfaces, appropriate ergonomic usage. One of the most interesting points is the introduction, at the automation level, of considerations related to the different start and stop modes of the process units. The inclusion of such considerations at HMI level will help in achieving a more solid and helpful interaction system on its relation with the safety and maintenance operations.

The structure of the paper is as follows. The second section is a brief introduction about the computer architecture, the plant layout in the flexible manufacturing system and brief comments about human-systems interaction. In the next section, the translation from the physical layout to the global design of the HMI is proposed: the human can use the HMI inside a control room or near the machine/process. Section four shows explain the relationship between automatic control and human intervention in manual mode. The paper ends by drawing some concluding remarks.

2 Human role in the Flexible manufacturing system

The application of the proposed integrated approach for HMI design is peformed within a concrete educational frame whose base focuses on two subjects from the Automatics and Electronics Engineering degree program: *Modelling and Simulating Systems* and *Integrated Production Systems*, at the Technical University of Catalonia UPC, Spain, and *Automatic Control* and *Industrial Informatics* at the "Universitat Autňnoma de Barcelona" UAB, Spain. The subjects from the technical engineering program make a special point on technical topics of programmable logic controllers (PLC) and industrial handler robots. In the remaining subjects of the engineering program, Petri Net modelling, simulation by means of the ARENAŠ discrete-even simulation software [2], physical distribution of flexible manufacturing systems and production system modelling and simulation are the main topics. Here, we focus on training the students with system behaviour analysis and simulation software so that they would therefore be able to apply all these techniques in their professional life. This software allows Flow Shop, Job Shop and Flexible Manufacturing Systems simulation [3]. Practical exercises in laboratory complement the theory and, with this purpose, a laboratory scale FMS has been built up, configured and being operative during the last two years which allows the comparison between simulated models and real plant performance. This paper does not describe more technical details of the role playing methodology and the project-based learning approach, readers interested in more detail about these techniques are directed to prior work Ponsa et al. [4].

Another step in a foreseeable future is the integration of supervision tools (supervisory control and data acquisition;SCADA software), management tools (Manufacturing Execution Systems; MES; and Enterprise Resource Planning; ERP) as well as production planning tools (planning policy analysis). In this sense it would also be advisable a new subject on automated production management within the study program which integrates MES and ERP systems at the highest level of the Computer Integrated Manufacturing CIM pyramid.

The Figure 1 shows the physical distribution of stations in the above referred educational FMS, the computers architecture and the human intervention with the FMS system: at the top level (human-computer interaction and human in the supervisory task; in the bottom level (human-machine interaction and human in manual mode task). The FMS is composed of electro-pneumatic units controlled by PLC's and PC's. The main purpose here consists of emulating current manufacturing systems: object mechanization and supply, transfer, product assembling, quality control, checking and storage; and technologies, such as, pneumatics, robotics, PLC, monitoring and production supervision come together. A total of 5 stations constitute the FMS.

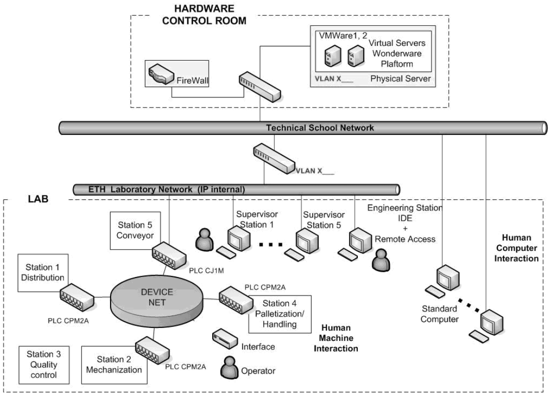


Figure 1: Human intervention in the flexible manufacturing system: high level (supervisory task), low level (machine operation

The cell manufactures two types of products: Product 1 goes across stations 1, 2, 3 and 4 (Palletization); and Product 2 goes from station 4 (Assembling) to 3 and comes back to station 4 (Palletization). The FMS system has product and process flexibility. The Product 1 has 9 different variations (size, color); the Product 2 has 8 different variations (size, color). The FMS can produce 17 different products. From the point of view of process flexibility, the system can produce only the Product 1, only the Product 2 or both; in this last case Station 4 performs two tasks (Assembling and Palletization) but only has a robot so that production planning policy is important in order to prevent an excessive workload in this station.

In contrast with other educational manufacturing systems which use pallets as functional trays for the same purposes, piece transport is carried out directly on an item by item way over the conveyor system from one station to the next one,. At cell control level, we make use of different programmable logic controllers(PLC's) such as the CP2MA, CJ1M models from OMRON; industrial communication protocols (RS-232, RS-422, PC-link, Device Net, Ethernet), and a PLC industrial Network with S7-314 from SIEMENS and communication protocols (RS-485, Profibus DP). Regarding programming software which enables local control level and monitoring-supervision linking we are testing products, such as In TouchŠ software and the technological system platform from Wonderware [5]. In a future, the authors want to increase the numbers of the stations and add a storage AS/RS station after the Station 4 [6].

3 Human machine interface design

The structure of the HMI application is a distributed interface with six parts (the Main application, and one application for each station) [7]. The Main application has information about the behavior of the flexible manufacturing system: (type of items, Petri Net algorithm, and coordination between stations, maintenance, and safety). The programmable controllers of each station have Device net modules and send this information to the master PLC.

Each station has an autonomous mode of functioning or an integrated mode of functioning. In the first mode, the global design of the HMI is conformed as a set of five autonomous HMI applications. In a near future, one of the problems to solve is the connection between the HMI single applications and works all the FMS in an integrated mode.

The full number of screens of the HMI application for the station1 is 17 screens, while the total number for the global HMI design is of more than 100 screens. Control engineering students have been working in the development of the HMI application for each station. In this sense, expert students (today are working in air traffic control) are the tutors of the novice students. Through the use of project based learning and cooperative learning, the students have made focus groups in order to define a clear and a global structure for all the HMI applications.

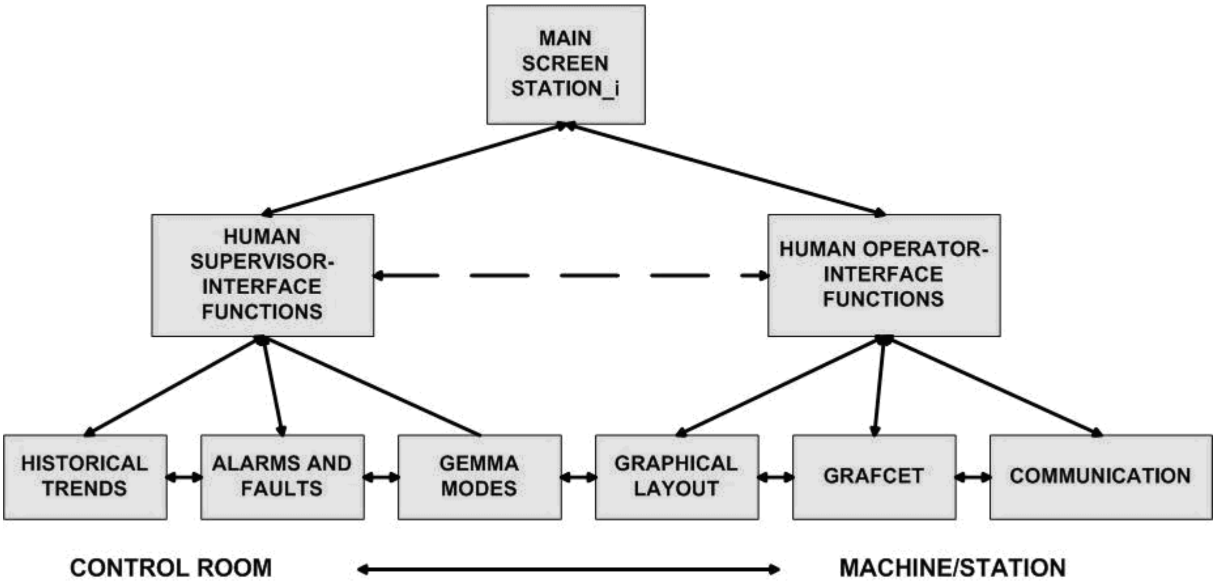


Figure 2: Station_i Display design. Example of a cyclic menu structure. From the main screen to monitoring and supervisory control screens.

In order to achieve an effective HMI application some display design guidelines need to be used. Here the ergonomic display design guideline called GEDIS has been employed [8]. The

GEDIS guide is a method that seeks to cover all the aspects of the interface design. From the initial point of view of strategies for effective human- computer interaction applied to supervision tasks in an industrial control room [9]. The GEDIS guide offers design recommendations at the time of creating the interface as well as recommendations for improvement of already created interfaces. The GEDIS guide is composed of two parts: description of ten indicators and measure of ten indicators. The indicators have been defined from extracted concepts of other generic human factors guidelines and from aspects of human interface design in human-computer interaction. The method to continue for the use of the GEDIS guide is: analyze the indicator, measure the indicator, obtain the global evaluation index and finally offer recommendations for improvement. For the correct use of the GEDIS guide the collaboration between human factor technicians is necessary since in some cases the expert's opinion is needed to analyze the indicator. This paper does not describe more details of the GEDIS, readers interested in more detail about this guideline are directed to prior work Ponsa and Díaz [10].

For each station we have developed an interface with the same template. The architecture of this interface has two parts: monitoring and supervisory control (see Fig. 2). For monitoring a set of screens (graphical layout of the station, automatic control, communication with the programmable controller PLC) have been defined. The aim of this design is to consider the automatic control loop and the human control. In human supervisory control a set of high level screens (start and stop modes guideline, alarm system, fault detection, diagnostic) has been defined. The aim of this design is to consider the supervisory control loop, a high level over the automatic control loop.

Often, the monitoring interface is associated with the activities of the human operator near the plant, or near the industrial machine (automation level). This task is related to the allocation of physical interface functions and human sensory-motor functions (choose manual or automatic control, activate or deactivate devices, use of the teach pendant of the robot, etc.). With the animating objects properties of the SCADA it is possible to develop a screen with the devices of the station (pneumatic actuators, motors, conveyors and sensors, etc). In this interface it is important the knowledge of the behavior of the devices. In this sense it is important the monitoring interface of station 4. It is a complex interface because it is necessary to develop an animation screen with the activities of the industrial robot (palletization or handling).

Another important screen within the monitoring interface is the panel screen when the human operator can select manual mode or automatic mode. This screen is the one that permits a clear interaction between the human and the automatic controller PLC.

On the other hand the supervisory control interface is associated with the activities of the human operator inside the industrial control room (supervisory control level). The human supervisory task is related to the allocation of interface functions and human cognitive functions (information processing, planning). Respect to the interface this part develops a set of screens: historical trends, alarm screen, fault detection and diagnosis screen. It is necessary to apply a general framework to define which is the correct interface representation of a risky situation. However, each station has specific devices so a specific framework is necessary in order to define a possible anomaly of the station.

4 Human intervention in the automation cycle

Petri Nets (PN) has proved to be a successful approach on a broad range of applications [11], [12], [13]. There is however one point that it is not clear how to deal with by using the PN formalism: the introduction of the human operator. Effectively, in every automation problem the fully automated part is just one part of the solution. It is customary that the operator can enter the loop in different ways, for example by tuning a PID controller [14] and such interaction needs to be considered as an integral part of the automation procedure as well as the communication of the automation device - usually a (PLC) - with the operator. A Human Machine Interface (either as a PC display, Industrial Panel, etc) provides the connection between the human operator action and the input to the control algorithm inside the controller device (control based PC, control based PLC).

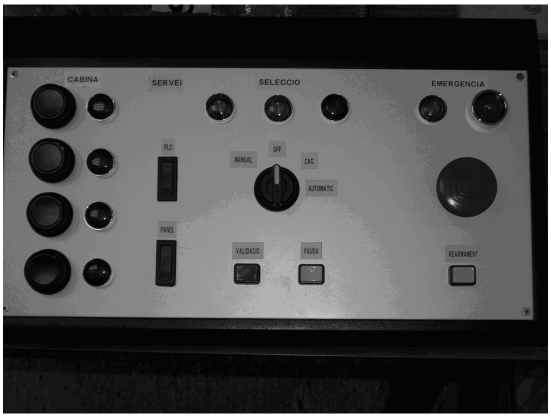


Figure 3: The human operator can intervene inside the automation cycle (GRAFCET transition) with the use of an industrial panel. In this panel the human operator follow the functional operational modes of the GEMMA guide.

On the other side, the industrial counterpart said that other approaches than PN are currently in use. Effectively, even PN allow tackling really large and complex problems, other approaches like Sequential Flow Charts; SFC; or State-Transition Graph; GRAFCET; can be considered the primary automation design tool found on the industry. In fact, some of the PLCs allow direct programming by using GRAFCET [15]. Therefore it seems there is a gap between both of these approaches. What we would like to point out here is that both disciplines should be combined and used. The important point with GRAFCET; a simpler approach if compared with PN; is that it has a close and clear connection with the design guide for start and stop modes called GEMMA [16]. Even the GEMMA guide was started to be used twenty-eight years ago and it was introduced to the engineering students in several places, in our opinion we don‘t paid sufficiently attention to the GEMMA guide. This is the reason this section would like to focus on the advantages of using the GEMMA guide and using it inside the HMI application explained in the previous section. Although GRAFCET is very useful in describing the detailed operation of a sequential control performed by a PLC it does not provide a general approach for the operation of an automated machine. It is still necessary to define general operational modes and conditions. This is usually done at the specifications definition stage. In this sense, GEMMA is a recommended tool for this task [17].

On the other hand, it has always been difficult in terms of vocabulary to clearly, and precisely, explain how to start with manual mode or a semi-automatic mode or an automatic mode. Also what are the consequences of an emergency stop, a safety stop or a fast stop for a production machine? Usually, these questions are answered by each designer in relation to his own experience and knowledge but can seldom be related to a systematical analysis, except in large companies which have been able to define design guide and standards for control and instrumentation.

From the previous observations, the GEMMA guide is a valuable tool to introduce automation in a more general setting than only for the automation of the process production cycle. It is really important to understand the role played by the operator and how he interacts with the automated system [18]. The different operation modes need to be established and be interrelated in a clear and well defined way.

For example in normal conditions the automation cycle is represented by three GEMMA modes: from A1 Initial State Stop, then F1 Normal Production, and A2 Requested Stop at the End of the Cycle and finally comeback to A1. In abnormal situation the automation cycle include the management of an emergency situation: from A1 Initial State Stop, then F1 Normal Production, D1 Emergency Stop, A5 Preparation to Restart after a Failure, A6 Production Reset to the Initital State and finally A1 again.

The main contribution of this section is to advance one step into an integral conception of the automation process: the design of the automation system has to include considerations on operational modes and these have to be reflected in the human interface. With this aim, Fig. 4 shows the GEMMA graphical representation inside the HMI application. When the system is in normal state or in anomaly, the human operator can see an active mode on screen, this facilities the situation awareness.

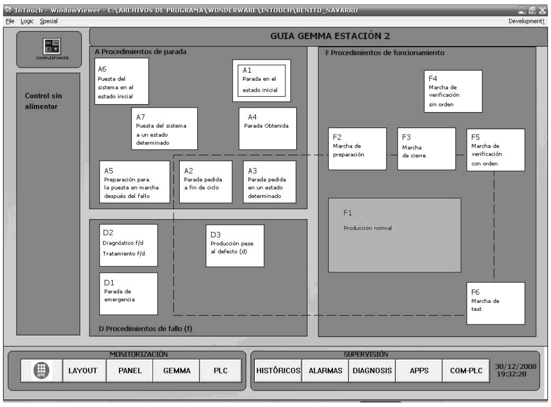

InTouch - WindowViewer - C:\ARCHIVOS DE PROGRAMA\WONDERWARE\INTOUCH\BENITO_NAVARRO
File Logic Special
Development!
COMPLEXFUNCFB
GUIA GEMMA ESTACIÓN 2
Control sin alimentar
A Procedimientos de parada
A6 Puesta del sistema en el estado inicial
A1 Parada en el estado inicial
A7 Puesta del sistema a un estado determinado
A4 Parada Obtenida
A5 Preparación para la puesta en marcha después del fallo
A2 Parada pedida a fin de ciclo
A3 Parada pedida en un estado determinado
D2 Diagnóstico f/d Tratamiento f/d
D3 Producción pese al defecto (d)
D1 Parada de emergencia
D Procedimientos de fallo (f)
F Procedimientos de funcionamiento
F4 Marcha de verificación sin orden
F2 Marcha de preparación
F3 Marcha de cierre
F5 Marcha de verificación con orden
F1 Producción normal
F6 Marcha de test
MONITORIZACIÓN
UPC
LAYOUT
PANEL
GEMMA
PLC
SUPERVISIÓN
HISTÓRICOS
ALARMAS
DIAGNOSIS
APPS
COM-PLC
30/12/2008 19:32:28

Figure 4: GEMMA graphical representation inside the HMI application. The human operator can observe the changes from Production, Stop or Failure procedures.

When the system is in normal state or in anomaly, the human operator can see an active mode on screen, this facilities the situation awareness.

In the HMI application there is a clear relationship between maintenance and safety services of the system. Inside the PLC an emergency GRAFCET has been designed with instructions in case of emergency. The maintenance recommendations are used in order to reduce the stop of the station, and the stop of the production. The translation of the GEMMA guide inside the PLC needs the use of the GRAFCET representation. When a problem appears in station3, for example, the human operator can activate the emergency stop: in this moment stations 1, 2 and

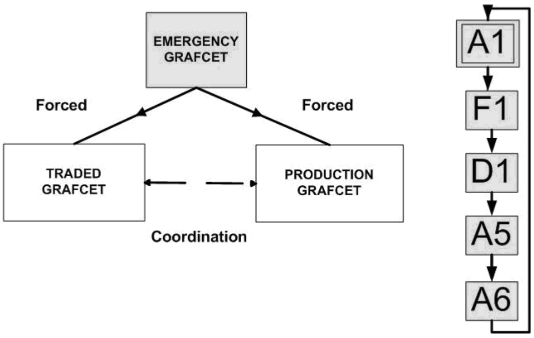


Figure 5: When the emergency stop is activated the Emergency GRAFCET forced the stop of the other GRAFCET. In the Traded GRAFCET we have automatic control or human manual control. The Production GRAFCET is the set of machine operations.

3 stop their production while station 4 and transport unit will still be functioning. When the human operator solves the problem, can procede to activate the reset of the system, drive the system to a safety state and begin again the production in stations 1, 2 and 3.

5 Conclusions

In this work we raised the need for a top-down integrated design for Human Automation systems, with special emphasis on the use of appropriate design and operational guidelines (GEDIS, GEMMA). Different actuation levels are first identified going from the Flexible Manufacturing System level to the automation level at each station. While the interfacing system at the station or global level has to cover, mostly, high level monitoring actions, it is at the automation level where more complex situations may arise. At this point the authors propose to base the use of the start and stop modes guide, GEMMA guide, to build up a convenient representation of the operational stages of the production machines. By designing the automation level HMI along these lines its integration with the automation control system becomes almost natural. The GEMMA guide approach is therefore a recommended taxonomy approach for the introduction of the human operator into the automation cycle in complex academic/industrial domains. It provides a natural relationship between the design of the automation system and the operational modes that are to be considered from the industrial panel point of view.

Acknowledgment

This work has received financial support from an internal grant of the Technical University of Catalonia (Project: Human centred design in human supervisory control, 2008) and from the Spanish CICYT program under grant DPI2007.63356.

Bibliography

[1] T. Sheridan and P. Parasuraman, “Human automation interaction,” *Reviews of Human Factors and Ergonomics*, vol. 1, pp. 89–129, 2006.

[2] Rockwell, “Arena simulation software,” *In URL: http://www.arenasimulation.com*, 2009.

[3] H. Boukef, M. Benrejeb, and P. Borne, "A proposed genetic algorithm coding for flow-shop scheduling problems," *International Journal of Computers, Communications and Control*, no. 3, pp. 229–240, 2007.

[4] P. Ponsa, B. Amante, J. Roman, S. Oliver, M.Díaz, and J. Vives, "Higher education challenges: introduction of active methodologies in engineering curricula," *International Journal of Engineering Education*, vol. 25, no. 4, pp. 799–813, 2009.

[5] Wonderware, "Wonderware intouch hmi," *In URL: http://global.wonderware.com*, 2009.

[6] F. Véliz and G. Lefranc, "Improvement and extension of virtual reality for flexible systems of manufacture," *International Journal of Computers, Communications and Control*, no. 2, pp. 95–101, 2006.

[7] P. Ponsa, R. Vilanova, and B. Amante, "Towards integral human-machine system conception: from automation design to usability concerns," *HSI 2009 - Human System Interaction 2nd International Conference, Catania, Italy*, 2009.

[8] P. Ponsa, R. Vilanova, M. Díaz, and A. Gomŕ, "Application of a guideline for design improvement in supervisory control," *International Journal on Human.Computer Interaction*, vol. 1, pp. 21–36, 2007.

[9] B. Schneiderman, *Designing the user interface. Strategies for effective human.computer interaction.* Addison-Wesley, third edition, 1997.

[10] P. Ponsa and M. Díaz, "Application of an ergonomic guideline for sugar mill control room interface design," *FAIM , 17th International Conference on Factory Automation and Intelligent Manufacturing, Philadelphia, USA*, vol. 1, pp. 89–129, 2007.

[11] K. Kurihara, S. Takigawa, N. Nishiuchi, and M. Kitaoka, "Factory automation control software designing method based on petri nets," *International Journal of Production Research*, vol. 40, no. 15, pp. 3605–3625, 2002.

[12] E. Gutuleac, "Descriptive time membrane petri nets for modeling of parallel computing," *International Journal of Computers, Communications and Control*, no. 3, pp. 33–39, 2006.

[13] F. Córdova, L. Canete, and L. Q. F. Yanine, "An intelligent supervising system for the operation of an underground mine," *International Journal of Computers, Communications and Control*, no. 4, pp. 259–269, 2008.

[14] R. Vilanova and O. Arrieta, "Balanced pid tuning application to series cascade control systems," *International Journal of Computers, Communications and Control*, no. 4, pp. 521–525, 2008.

[15] AFCET-ADEPA, "Le grafcet," *Second Edition, Toulouse, Opadus*, 1995.

[16] ADEPA, "Le gemma. guide d'etude des modes de marches et d'arrets," *Montrouge, ADEPA*, 1981.

[17] P. Ponsa and R. Vilanova, *Automatización de procesos mediante la guía GEMMA*. Edicions UPC, no. 105, 2005.

[18] K. Li, S. Thompson, P. Wieringa, and J. Xeng, "A study on the role on human operators in supervisory automation system and its implications," *Proceedings of the 4th World Intelligent Control and Automation*, pp. 3288–3293, 2002.

Int. J. of Computers, Communications & Control, ISSN 1841-9836, E-ISSN 1841-9844
Vol. VI (2011), No. 1 (March), pp. 175-186

A New Linear Classifier Based on Combining Supervised and Unsupervised Techniques

L. State, I. Paraschiv-Munteanu

Luminiţa State
University of Piteşti
Faculty of Mathematics and Computer Science
Romania, 110040 Piteşti, 1 Târgu din Vale St.
E-mail: lstate@clicknet.ro

Iuliana Paraschiv-Munteanu
University of Bucharest
Faculty of Mathematics and Computer Science
Romania, 010014 Bucharest, 14 Academiei St.
E-mail: pmiulia@fmi.unibuc.ro

Abstract: The aim of the research reported in the paper is to obtain an alternative approach in using Support Vector Machine (SVM) in case of non-linearly separable data based on using the k-means algorithm instead of the standard kernel based approach.

The SVM is a relatively new concept in machine learning and it was introduced by Vapnik in 1995. In designing a classifier, two main problems have to be solved, on one hand the option concerning a suitable structure and on the other hand the selection of an algorithm for parameter estimation.

The algorithm for parameter estimation performs the optimization of a convenable selected cost function with respect to the empirical risk which is directly related to the representativeness of the available learning sequence. The choice of the structure is made such that to maximize the generalization capacity, that is to assure good performance in classifying new data coming from the same classes. In solving these problems one has to establish a balance between the accuracy in encoding the learning sequence and the generalization capacities because usually the over-fitting prevents the minimization of the empirical risk.

Keywords: support vector machine, classification, unsupervised learning, supervised learning, k-means algorithm.

1 Introduction

In addition to its solid mathematical foundation in statistical learning theory, SVM's have demonstrated highly competitive performance in numerous real-world applications, such as bioinformatics, text mining, face recognition, and image processing, which has established SVM's as one of the state-of-the-art tools for machine learning and data mining, along with other soft computing techniques. In training support vector machines the decision boundaries are determined directly from the training data so that the separating margins of decision boundaries are maximized in the high-dimensional space called feature space. This learning strategy, based on statistical learning of the training data and the unknown data.

The method based on support vectors aims to increase the efficiency in approximating multidimensional functions. The basic idea in a SVM approach is twofold. On one hand it aims to determine a classifier that minimizes the empirical risk, that is to encode the learning sequence as good as possible with respect to a certain architecture, and the other hand to improve the generalization capacity by minimizing the generalization error. In case of non-linear separable data the SVM is combined with kernel based technique which transforms the data in a linear separable data by mapping the initial data on to higher dimensional space of features. This mapping is performed in terms of special tailored kernels that allow to keep the computations at a reasonable complexity level.

The SVM approach proves useful in classifying linear separable data as well as non-linear separable data because the mapping of the initial data onto a higher dimensional space of features determines that these classifiers behave as non-linear classifiers.

The basic idea of a SVM approach is to obtain higher dimensional representations of the initial data by mapping them using a technique based on kernels in a feature space, such that for a non-linear separable learning sequence, its representation in the feature space becomes linearly separable. Being given that the representation of the learning sequence in the feature space is linearly separable, several techniques can be applied to determine in this space a separating hyperplane. Obviously, in case of linearly separable learning sequence the set of solutions is infinite, different algorithms yielding to different hyperplane solutions. Since a solution that keeps at distance as much as possible all examples assures good generalization capacities, this can be taken as a criterion in selecting a best solution among the available solutions.

At first sight, it seems unreasonable to combine a supervised technique to an unsupervised one, mainly because they refer to totally different situations. On one hand, the supervised techniques are applied in case the data set consists of correctly labeled objects, and on the other hand the unsupervised methods deal with unlabeled objects. However our point is to combine the SVM and k-means algorithms, in order to obtain a new design of a linear classifier.

A new linear classifier resulted as a combination of a supervised SVM method and 2-means algorithm is proposed in the paper, and its efficiency is evaluated on experimental basis in the final part of the paper.

The tests were performed on simulated data generated from multi-dimensional normal repartitions yielding to linearly separable and non-linearly separable samples respectively.

2 Supervised Learning Using SVM

Let us assume that the data is represented by examples coming from two categories or classes such that the true provenance class for each example is known. We refer to such a collection of individuals as a supervised learning sequence, and it is represented as

$$\mathcal{S}=\left\{ (x_i,y_i) \mid x_i \in \mathbf{R}^d \, , \; y_i \in \{-1,1\} \, , \; i=\overline{1,N} \right\} \, . \tag{1}$$

The values 1, -1 are taken as labels corresponding to the classes. We say that the data are linearly separable if there exists a linear discriminant function $g : \mathbf{R}^d \longrightarrow \mathbf{R}$,

$$g(u) = b + w_1 u_1 + \ldots + w_d u_d \, , \tag{2}$$

where $u = (u_1, \ldots, u_d) \in \mathbf{R}^d$, such that $y_i g\left(x_i\right) > 0, \;\; \forall \; (x_i, y_i) \in \mathcal{S}$.

Denoting by $w = (w_1, \ldots, w_d)^T$ the vector whose entries are the coefficients of g, we say that $\mathcal{S}$ is separated without errors by the hyperplane

$$H_{w,b} \; : \quad w^T u + b = 0 \, , \tag{3}$$

and $H_{w,b}$ is called a solution of the separating problem because all examples coming from the class of label 1 belong to the positive semi-space, and all examples coming from the class of label

-1 belong to the negative semi-space defined by $H_{w,b}$. Obviously a hyperplane is a solution of separating problem if the functional margin $\min\left\{y_i\left(w^T x_i + b\right),\, 1 \le i \le N\right\} > 0$.

In a SVM-based approach, by imposing that the functional margin is 1, the search for a solution yields to a constrained quadratic programming problem imposed on the objective function $\Phi(w) = \frac{1}{2}\left\|w\right\|^2$,

$$\begin{cases} \min \Phi(w) \\ y_i\left(w^T x_i + b\right) \ge 1\,, \quad i = \overline{1,N}\,. \end{cases} \tag{4}$$

If w^* is a solution of (4), then H_{w^*,b^*} is called an optimal separating hyperplane. The computation of w^* and b^* is carried out using the $SVM1$ algorithm.

Algorithm $SVM1$ ([9])

Input: *The learning sequence* $\mathcal{S}$*;*

Step 1. *Compute the matrix* $D = (d_{ik})$ *of entries* $d_{ik} = y_i y_k \left(x_i\right)^T x_k\,,\ i,k = \overline{1,N}$*;*

Step 2. *Solve the constrained optimization problem*

$$\begin{cases} \alpha^* = arg\left(\max\limits_{\alpha \in \mathbf{R}^N}\left(\alpha^T \mathbf{1} - \frac{1}{2}\,\alpha^T D \alpha\right)\right)\,, \\ \alpha_i \ge 0\,, \quad \forall\, 1 \le i \le N\,, \quad \sum\limits_{i=1}^{N} \alpha_i y_i = 0\,, \end{cases} \tag{5}$$

If $\alpha_i^* > 0$ *then* x_i *is called the support vector.*

Step 3. *Select two support vectors* x_r*,* x_s *such that* $\alpha_r^* > 0\,,\ \alpha_s^* > 0\,,\ y_r = -1\,,\ y_s = 1$.

Step 4. *Compute the parameters* w^*, b^* *of the optimal separating hyperplane, and the width of the separating area* $\rho\left(w^*, b^*\right)$*,*

$$\begin{cases} w^* = \sum\limits_{i=1}^{N} \alpha_i^* y_i x_i\,,\ b^* = -\frac{1}{2}\left(w^*\right)^T\left(x_r + x_s\right)\,, \\ \rho\left(w^*, b^*\right) = \frac{2}{\left\|w^*\right\|} \end{cases} \tag{6}$$

Output: $w^*, b^*, \rho\left(w^*, b^*\right)$.

A linear separable sample is represented in figure 1*a*. The straight lines d_1, d_2, d_3 and d_4 are solutions for the separating problem of $\mathcal{S}$, d_4 corresponds to the optimal separating hyperplane. The examples placed at the minimum distance to the optimum separating hyperplane are the support vectors.

In case of non-linearly separable samples the idea is to determine a separating hyperplane that minimizes the number of misclassified examples.

The problem of finding a optimal hyperplane in case of non-linearly separable samples has been approached several ways. The approach introduced by Cortes and Vapnik ([3]) uses the error function

$$\Phi_\sigma(\xi) = \sum_{i=1}^{N} \xi_i^\sigma\,, \tag{7}$$

where the slack variables $\xi_i\,, 1 \le i \le N$, are taken as indicators for the classification errors (see figure 1*b*), and σ is a positive real number.

The optimality is expressed in terms of the objective function $\Phi : \mathbf{R}^d \times \mathbf{R}^N \longrightarrow [0, +\infty)$

$$\Phi(w, \xi) = \frac{1}{2}\left\|w\right\|^2 + c\, F\left(\sum_{i=1}^{N} \xi_i^\sigma\right)\,, \tag{8}$$

where $c > 0$ is a given constant, $\xi = (\xi_1, \ldots, \xi_N)$, and F is a monotone convex function, $F(0) = 0$.

The idea is to compute a subset of $\mathcal{S}$, say $\{(x_{i_1}, y_{i_1}), \ \ldots, (x_{i_k}, y_{i_k})\}$, by minimizing $\Phi_\sigma(\xi)$, such that there exists an optimal hyperplane for $\mathcal{S} \setminus \{(x_{i_1}, y_{i_1}), \ldots, \ (y_{i_k}, y_{i_k})\}$. Such an optimal hyperpl perpla

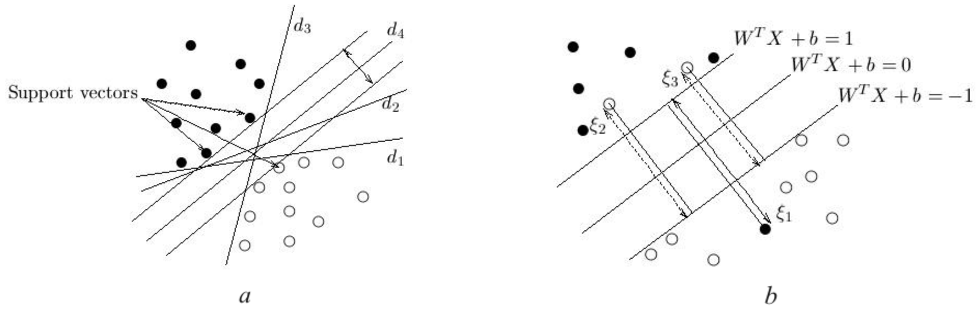


Figure 1: *a) Optimal separating hyperplane; b)Classification errors.*

A soft margin hyperplane is a solution of the constrained optimization problem

$$\begin{cases} \arg\left(\min\limits_{w\in\mathbf{R}^d,\, b\in\mathbf{R},\, \xi\in\mathbf{R}^N} \left(\Phi(w,\xi)\right)\right) \\ \\ y_i\left(w^T x_i + b\right) \geq 1 - \xi_i\,, \ \forall\, 1 \leq i \leq N\,, \\ \xi_i \geq 0\,, \ \forall\, 1 \leq i \leq N\,, \end{cases} \tag{9}$$

The samples represented in figure 1*b*, correspond to the non-linearly separable case. A soft margin hyperplane, the separating area, and the slack variables are indicated in figure 1*b*.

The computation of a soft margin hyperplane is carried out by the algorithm $SVM2$.

Algorithm $SVM2$ ([9])

Input: *The learning sequence* $\mathcal{S}$; $c \in (0, \infty)$.

Step 1. *Compute the matrix* $D = (d_{ik})$ *of entries,* $d_{ik} = y_i y_k (x_i)^T x_k\,,\ i,k = \overline{1,N}$ *;*

Step 2. *Solve the constrained optimization problem*

$$\begin{cases} \alpha^* = arg\left(\max\limits_{\alpha\in\mathbf{R}^N}\left(\alpha^T\mathbf{1} - \frac{1}{2}\,\alpha^T D\alpha - \frac{(\alpha_{max})^2}{4\,c}\right)\right)\,, \\ \alpha_i \geq 0\,, \quad \forall\, 1 \leq i \leq N\,, \quad \sum\limits_{i=1}^{N} \alpha_i y_i = 0\,, \end{cases} \tag{10}$$

where $\alpha_{max} = \max\{\alpha_1, \ldots, \alpha_N\}$

Step 3. *Select two support vectors* x_r, x_s *such that* $\alpha_r^* > 0\,,\ \alpha_s^* > 0\,,\ y_r = -1\,,\ y_s = 1$.

Step 4. *Compute the parameters* w^*, b^* *of the soft margin hyperplane, and the width of the separating area* $\rho(w^*, b^*)$*, according to (6).*

Output: $w^*, b^*, \rho(w^*, b^*)$.

3 Unsupervised Learning using the k-means Method

Center-based clustering algorithms are very efficient for clustering large and high-dimensional databases. They use objective functions to express the quality of any clustering solution, the optimal solution corresponding to a solution to a constrained/unconstrained optimization problem imposed of the particular objective function. Usually the clusters found have convex shapes and a one of more centers are computed for each cluster. The k-means algorithm was introduced by MacQueen ([8]) for clustering numerical data, each of the produced clusters having a center referred as the cluster *mean.*

Let $\mathcal{D} = \{x_1, \ldots, x_N\} \subset \mathbf{R}^d$ be the data, k a given positive integer. The classes of any partition $\{\mathcal{C}_1, \ldots, \mathcal{C}_k\}$ of $\mathcal{D}$ are called *clusters.* For any $\{\mu(\mathcal{C}_1), \ldots, \mu(\mathcal{C}_k)\} \subset \mathbf{R}^d$ where each $\mu(\mathcal{C}_i)$ in taken as the center of $\mathcal{C}_i$, then *the inertia momentum* is,

$$\varepsilon = \sum_{i=1}^{k} \sum_{x \in \mathcal{C}_i} d^2\left(x, \mu\left(\mathcal{C}_i\right)\right) , \tag{11}$$

where d is a convenable distance function on $\mathbf{R}^d$. In the following we take d as being the Euclidean distance, $d(x, y) = \|x - y\|$.

The k-means method proceeds by iteratively allocate the individuals to the nearest clusters and re-computation of the centers is performed to minimize the inertia momentum, the computation ending when non-significant changes of the centers/value of inertia momentum/membership functions of individuals to clusters are obtained.

The k-means algorithm can be treated as an optimization problem where the goal is to minimize a given objective function under certain constraints.

Let $\mathcal{C}$ be the set of all subsets of $\mathbf{R}^d$ of cardinal k, any particular $Q = \{q_1, \ldots, q_k\} \in \mathcal{C}$ is a possible set of cluster centers.

Any partition of $\mathcal{D}$ into k classes can be obviously represented by a $N \times k$ matrix $W = (w_{il})$ where

$$\begin{aligned} &(i) \quad w_{il} \in \{0, 1\}, \quad i = \overline{1, N},\ l = \overline{1, k} \\ &(ii) \quad \sum_{l=1}^{k} w_{il} = 1, \quad i = \overline{1, N}. \end{aligned} \tag{12}$$

The k-means algorithm can be formulated as the constrained optimization problem on the objective function $P(W, Q) = \sum_{i=1}^{N} \sum_{l=1}^{k} w_{il} \|x_i - q_l\|^2$ as follows:

$$\begin{cases} \min\limits_{W \in \mathcal{M}_{N \times k}(\mathbf{R}),\ Q \in \mathcal{C}} P(W, Q) \\ w_{il} \in \{0, 1\}, \quad i = \overline{1, N},\ l = \overline{1, k}, \\ \sum\limits_{l=1}^{k} w_{il} = 1, \quad i = \overline{1, N}, Q = \{q_1, \ldots, q_k\}. \end{cases} \tag{13}$$

The 'hard' problem (13) can be solved by decomposing it into two simpler problems P_1 and P_2, and then iteratively solving them, where

P_1. Fix $Q = \widehat{Q} \in \mathcal{C}$ and solve the reduced constrained optimization problem for $P\left(W, \widehat{Q}\right)$.

P_2. Fix $W = \widehat{W} \in \mathcal{M}_{N \times k}(\mathbf{R})$ and solve the reduced unconstrained optimization problem for $P\left(\widehat{W}, Q\right)$.

The solutions of these problems can be derived by straightforward computations, and they are given by the following theorems:

Theorem 1. *For any fixed set of centers* $\widehat{Q} = \{\widehat{q}_1, \ldots, \widehat{q}_k\}$, *the function* $P\left(W, \widehat{Q}\right)$ *is minimized in* $W^{(0)} = \left(w_{ij}^{(0)}\right)$ *if and only if* $W^{(0)}$ *satisfies the conditions*

$$\begin{aligned} & w_{il}^{(0)} = 0 \Longleftrightarrow \|x_i - \widehat{q}_l\| > \min_{1 \le t \le k} \|x_i - \widehat{q}_t\| \ , \\ & w_{il}^{(0)} = 1 \Longrightarrow \|x_i - \widehat{q}_l\| = \min_{1 \le t \le k} \|x_i - \widehat{q}_t\| \ , \\ & \sum_{j=1}^{k} w_{ij}^{(0)} = 1 \ , \ \textit{ for any } \ i = \overline{1, N} \ , \ l = \overline{1, k}. \end{aligned}$$

Note that in general, for any given $\widehat{Q}$ there are more solutions because in general there exist individuals x_i at minimum distance to more than one center of $\widehat{Q}$.

Theorem 2. *For any fixed* $\widehat{W}$ *satisfying the constraints of (13), the function* $P\left(\widehat{W}, Q\right)$ *is minimized there exist an unique point* $Q^{(0)} = \left\{q_1^{(0)}, \ldots, q_k^{(0)}\right\}$ *if and only if*

$$q_l^{(0)} = \left(\sum_{i=1}^{N} \widehat{w}_{il} x_i\right) \Bigg/ \left(\sum_{i=1}^{N} \widehat{w}_{il}\right) \ , \quad l = \overline{1, k}.$$

The scheme of the k-means algorithm viewed as search method for solving the optimization problem (13) is:

The algorithm k-MOP

```
Input: $\mathcal{D}$ - the data set,
       $k$ - the pre-specified number of clusters,
       $d$ - the data dimensionality,
       $T$ - threshold on the maximum number of iterations.
Initializations: $Q^{(0)}$, $t \longleftarrow 0$
Solve $P\left(W, Q^{(0)}\right)$ and get $W^{(0)}$
$sw \longleftarrow false$
repeat
    $\widehat{W} \longleftarrow W^{(t)}$
    solve $P\left(\widehat{W}, Q\right)$ and get $Q^{(t+1)}$
    if $P\left(\widehat{W}, Q^{(t)}\right) = P\left(\widehat{W}, Q^{(t+1)}\right)$ then
        $sw \longleftarrow true$
        output $\left(\widehat{W}, Q^{(t+1)}\right)$
    else
        $\widehat{Q} \longleftarrow Q^{(t+1)}$
        solve $P\left(W^{(t)}, \widehat{Q}\right)$ and get $W^{(t+1)}$
        if $P\left(W^{(t)}, \widehat{Q}\right) = P\left(W^{(t+1)}, \widehat{Q}\right)$ then
            $sw \longleftarrow true$
            output $\left(W^{(t+1)}, \widehat{Q},\right)$
        endif
    endif
    $t \longleftarrow t + 1$
until $sw$ or $t > T$.
Output: $\widehat{W}$, $\widehat{Q}$.
```

Note that the computational complexity of the algorithm k-MOP is $\mathcal{O}(Nkd)$ per iteration. The sequence of values $P\left(W^{(t)}, Q^{(t)}\right)$ where $W^{(t)}$, $Q^{(t)}$ are computed by k-MOP is strictly decreasing, therefore the algorithm converges to a local minima of the objective function.

4 The Combined Separating Technique based on SVM and the k-means Algorithm

At first sight, it seems unreasonable to compare a supervised technique to an unsupervised one, mainly because they refer to totally different situations. On one hand the supervised techniques are applied in case the data set consists of correctly labeled objects, and on the other hand the unsupervised methods deal with unlabeled objects. However our point is to combine SVM and k-means algorithm, in order to obtain a new design of a linear classifier.

The aim of the experimental analysis is to evaluate the performance of the linear classifier resulted from the combination of the supervised SVM method and the 2-means algorithm.

The method can be applied to data, either linearly separable or non-linearly separable. Obviously in case of non-linearly separable data the classification can not be performed without errors and in this case the number of misclassified examples is the most reasonable criterion for performance evaluation. Of a particular importance is the case of linearly separable data, in this case the performance being evaluated in terms of both, misclassified examples and the generalization capacity expressed in terms of the width of separating area. In real life situations, usually is very difficult or even impossible to established whether the data represents a linearly/non-linearly separable set. In using the $SVM1$ approach we can identify which case the given data set belongs to. For linear separable data, $SVM1$ computes a separation hyperplane optimal from the point of view of the generalization capacity. In case of a non-linearly separable data $SVM2$ computes a linear classifier that minimizes the number of misclassified examples. A series of developments are based on non-linear transforms represented be kernel functions whose range is high dimensional spaces. The increase of dimensionality and the convenable choice of the kernel aim to transform a non-linearly separable problem into a linearly separable one. The computation complexity corresponding to kernel-based approaches is significantly large, therefore in case the performance of the algorithm $SVM1$ proves reasonable good it could be taken as an alternative approach of a kernel-based SVM. We perform a comparative analysis on data consisting of examples generated from two dimensional Gaussian distributions.

In case of a non-linearly separable data set, using the k-means algorithm, we get a system of pairwise disjoint clusters together with the set of their centers representing a local minimum point of the criterion (13), the clusters being linearly separable when $k = 2$. Consequently, the $SVM1$ algorithm computes a linear classifier that separates without errors the resulted clusters.

Our procedure is described as follows

```
Input: $\mathcal{S}$ = the learning sequence;
Step 1. Compute the matrix $D = (d_{ik})$ of entries, $d_{ik} = y_i y_k (x_i)^T x_k$, $i, k = \overline{1, N}$;
        $sh \longleftarrow true$
Step 2. If the constrained optimization problem (5) does not have solution then
          $sh \longleftarrow false$
          input $c \in (0, \infty)$, for soft margin hyperplane
          Solve the constrained optimization problem (10)
        endif
Step 3. Select $x_r$, $x_s$ such that $\alpha_r^* > 0$, $\alpha_s^* > 0$, $y_r = -1$, $y_s = 1$;
```

```
        Compute the parameters $w^*, b^*$ of the separating hyperplane, and the width
        of the separating area, $\rho\,(w^*, b^*)$ according to (6);
Step 4. If not sh then
            compute nr_err1 = the numbers of examples incorrectly classified;
            compute err1 = classification error;
        endif
Step 5. Split the set $\mathcal{D} = \left\{ x_i \mid x_i \in \mathbf{R}^d\,,\ i = \overline{1, N} \right\}$ into two clusters $\mathcal{C}_1$ and $\mathcal{C}_2$ using
        the 2-means algorithm and label the data belonging to $\mathcal{C}_1$ by $y_i' = 1$,
        and label by $y_i' = -1$ the data belonging to $\mathcal{C}_2$.
Step 6. Apply the algorithm SVM1 to $\mathcal{S}' = \left\{ \left(x_i, y_i'\right) \mid x_i \in \mathbf{R}^d,\ y_i' \in \{-1, 1\},\ i = \overline{1, N} \right\}$
        and obtain the parameters of optimal separating hyperplane: $w_1^*, b_1^*, \rho\,(w_1^*, b_1^*)$;
        compute nr_err2 = the number of data incorrectly classified by the algorithm $2-means$;
        compute err2 = classification error resulted after the $2-means$ splitting ;
Output: $w^*, b^*, \rho\,(w^*, b^*)$, nr_err1, err1; $w_1^*, b_1^*, \rho\,(w_1^*, b_1^*)$, nr_err2, err2.
```

5 Comparative Analysis and Experimental Results

The experimental analysis is based on a long series of tests performed on linear/non-linear separable simulated data of different volumes. The analysis aims to derive conclusions concerning:

1. The statistical properties (the empirical means, covariance matrices, eigenvalues, eigenvectors) of the clusters computed by the 2-means algorithm as compared to their counterparts corresponding to the true distributions they come from.
2. The comparison of the performances corresponding to the linear classifier resulted as a combination of SVM and the 2-means algorithm described in section 4 and $SVM2$ in terms of the empirical error.
3. The analysis concerning the influences of the samples sizes on the performance of the procedure described in section 4.
4. The quality of cluster characterization in terms of the principal directions given by a system of unit orthogonal eigenvectors of the sample covariance and empirical covariance matrices of the computed clusters. The analysis aimed to derive conclusions concerning the contribution of each principal direction, and for this reason, some tests were performed on data whose first principal component is strongly dominant, and when the principal directions are of the same importance respectively.

The tests were performed on data generated from two-dimensional normal distributions $\mathcal{N}\left(\mu_i, \Sigma_i\right)$, $i = 1, 2$ of volumes N_1 and N_2, respectively. The sample covariance matrices are denoted by $\widehat{\mu}_i, \widehat{\Sigma}_i$, $i = 1, 2$. The centers and the empirical covariance matrices corresponding to the clusters computed by the 2-means algorithm are denoted by $\overline{\mu}_i, \overline{\Sigma}_i$, $i = 1, 2$. We denote by Z_i , $\widehat{Z}_i$, $\overline{Z}_i$, $i = 1, 2$ orthogonal matrices having as columns unit eigenvectors of Σ_i , $\widehat{\Sigma}_i$, $\overline{\Sigma}_i$, $i = 1, 2$ respectively.

Test 1: $N_1 = N_2 = 50$, $\mu_1 = \begin{pmatrix} 1 \\ 1 \end{pmatrix}$, $\Sigma_1 = \begin{pmatrix} 1 & 0 \\ 0 & 0.25 \end{pmatrix}$, $\mu_2 = \begin{pmatrix} 2 \\ 3 \end{pmatrix}$, $\Sigma_2 = \begin{pmatrix} 0.5 & 0 \\ 0 & 0.5 \end{pmatrix}$.
The matrices Z_1 , Z_2 and their eigenvalues are

$$\lambda_1^{(1)} = 0.25\,, \quad \lambda_2^{(1)} = 1, \quad Z_1 = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \lambda_1^{(2)} = 0.5\,, \quad \lambda_2^{(2)} = 0.5, \quad Z_2 = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}.$$

The set is non-linear separable and it is represented in figure $2i)a$. In this case we get

$$\widehat{\mu}_1=\begin{pmatrix}0.92\\1.0004\end{pmatrix},\ \widehat{\Sigma}_1=\begin{pmatrix}0.85&0.08\\0.08&0.25\end{pmatrix},\ \widehat{\mu}_2=\begin{pmatrix}1.98\\2.87\end{pmatrix},\ \widehat{\Sigma}_2=\begin{pmatrix}0.44&0.09\\0.09&0.63\end{pmatrix}.$$

the matrices $\widehat{Z}_1$, $\widehat{Z}_2$ and their eigenvalues being

$$\widehat{\lambda}_1^{(1)}=0.24\,,\ \widehat{\lambda}_2^{(1)}=0.86,\ \widehat{Z}_1=\begin{pmatrix}0.14&-0.98\\-0.98&-0.14\end{pmatrix},\ \widehat{\lambda}_1^{(2)}=0.40\,,\ \widehat{\lambda}_2^{(2)}=0.67,\ \widehat{Z}_2=\begin{pmatrix}-0.92&0.38\\0.38&0.92\end{pmatrix}.$$

Using the $SVM2$ with $c=70$ we get the classification error $class\_error=14.70$, the number of misclassified samples $n\_errors=13$ and the width of separating area is $\rho=0.61$. The value of the error coefficient defined as the ratio of the number of misclassified samples and total volume of the data is $c\_error=0.13\%$. The soft margin line d_1 is represented in figure $2i)b$.

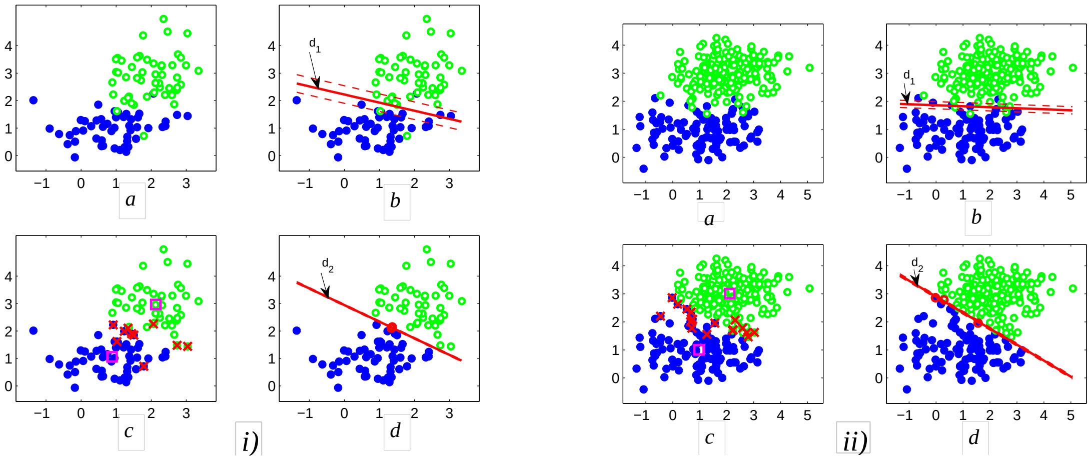


Figure 2: *i) The classification of the data set in test 1; ii) The classification of the data set in test 2.*

By applying the 2-means algorithm we get clusters whose sample means and covariance matrices are

$$\overline{\mu}_1=\begin{pmatrix}0.88\\1.06\end{pmatrix},\ \overline{\Sigma}_1=\begin{pmatrix}0.64&0.05\\0.05&0.30\end{pmatrix},\ \overline{\mu}_2=\begin{pmatrix}2.13\\2.96\end{pmatrix},\ \overline{\Sigma}_2=\begin{pmatrix}0.41&-0.06\\-0.06&0.56\end{pmatrix}.$$

The matrices $\overline{Z}_1$, $\overline{Z}_2$ and their eigenvalues are

$$\overline{\lambda}_1^{(1)}=0.29\,,\ \overline{\lambda}_2^{(1)}=0.65,\ \overline{Z}_1=\begin{pmatrix}0.14&-0.98\\-0.98&-0.14\end{pmatrix},\ \overline{\lambda}_1^{(2)}=0.39\,,\ \overline{\lambda}_2^{(2)}=0.58,\ \overline{Z}_2=\begin{pmatrix}-0.92&-0.37\\-0.37&0.92\end{pmatrix},$$

the number of misclassified samples is 10 and the clusters are represented in figure $2i)c$.

Note that the computed centers and clusters are not influenced by the choice of the initial centers. The clusters computed by the 2-means algorithm for randomly selected initial centers are reprezented in figure $2i)c$. The separating line d_2 resulted by applying the $SVM1$ algorithm to the data represented by the clusters computed by the 2-means algorithm is represented in figure $2i)d$.

Test 2: $N_1=100\,,\quad N_2=200,\ \mu_1=\begin{pmatrix}1\\1\end{pmatrix},\ \Sigma_1=\begin{pmatrix}1&0\\0&0.25\end{pmatrix},\ \mu_2=\begin{pmatrix}2\\3\end{pmatrix},\ \Sigma_2=\begin{pmatrix}1&0\\0&0.25\end{pmatrix},$

$$\lambda_1^{(1)}=0.25\,,\quad \lambda_2^{(1)}=1,\quad Z_1=\begin{pmatrix}0&1\\1&0\end{pmatrix},\ \lambda_1^{(2)}=0.25\,,\quad \lambda_2^{(2)}=1,\quad Z_2=\begin{pmatrix}0&1\\1&0\end{pmatrix},$$

$$\widehat{\mu}_1=\begin{pmatrix}1.12\\0.92\end{pmatrix},\ \widehat{\Sigma}_1=\begin{pmatrix}1.35&0.04\\0.04&0.26\end{pmatrix},\ \widehat{\mu}_2=\begin{pmatrix}2.01\\3.00\end{pmatrix},\ \widehat{\Sigma}_2=\begin{pmatrix}0.86&0.05\\0.05&0.25\end{pmatrix},$$

$\widehat{\lambda}_1^{(1)}=0.26\,,\ \widehat{\lambda}_2^{(1)}=1.35,\ \widehat{Z}_1=\begin{pmatrix}0.03 & -0.99\\ -0.99 & -0.03\end{pmatrix},\ \widehat{\lambda}_1^{(2)}=0.25\,,\ \widehat{\lambda}_2^{(2)}=0.87,\ \widehat{Z}_2=\begin{pmatrix}0.09 & -0.99\\ -0.99 & -0.09\end{pmatrix}.$

The data set is non-linear separable and it is represented in figure $2ii)a$. Applying the $SVM2$ for $c=5$ we obtain the soft margin line d_1 represented in figure $2ii)b$ and $class\_error=19.12$, $n\_errors=13$, $\rho=0.25$, $c\_error=0.043\%$.

The clusters computed by the 2-means algorithm are represented in figure $2ii)c$ and their statistical characteristics are

$\overline{\mu}_1=\begin{pmatrix}0.96\\ 1.004\end{pmatrix},\ \overline{\Sigma}_1=\begin{pmatrix}1.19 & -0.10\\ -0.10 & 0.38\end{pmatrix},\ \overline{\mu}_2=\begin{pmatrix}2.10\\ 3.007\end{pmatrix},\ \overline{\Sigma}_2=\begin{pmatrix}0.76 & -0.02\\ -0.02 & 0.28\end{pmatrix},$

$\overline{\lambda}_1^{(1)}=0.37\,,\ \overline{\lambda}_2^{(1)}=1.20,\ \overline{Z}_1=\begin{pmatrix}-0.12 & -0.99\\ -0.99 & 0.12\end{pmatrix},\ \overline{\lambda}_1^{(2)}=0.27\,,\ \overline{\lambda}_2^{(2)}=0.76,\ \overline{Z}_2=\begin{pmatrix}-0.05 & -0.99\\ -0.99 & 0.05\end{pmatrix}.$

In this case the number of misclassified samples is 18. Note that the initial choice of the centers does not influence significantly the computed centers and clusters. For instance in figure $2ii)c$ are represented the resulted clusters in case of randomly selected initial centers.

The separating line d_2 computed by the algorithm $SVM1$ applied to the data represented by these clusters is represented in figure $2ii)d$.

Test 3: $N_1=N_2=50,\ \mu_1=\begin{pmatrix}1\\ 1\end{pmatrix},\ \Sigma_1=\begin{pmatrix}1 & 0\\ 0 & 0.25\end{pmatrix},\ \mu_2=\begin{pmatrix}3\\ 4\end{pmatrix},\ \Sigma_2=\begin{pmatrix}0.5 & 0\\ 0 & 0.5\end{pmatrix},$

$\lambda_1^{(1)}=0.25\,,\quad \lambda_2^{(1)}=1,\quad Z_1=\begin{pmatrix}0 & 1\\ 1 & 0\end{pmatrix},\ \lambda_1^{(2)}=0.5\,,\quad \lambda_2^{(2)}=0.5,\quad Z_2=\begin{pmatrix}1 & 0\\ 0 & 1\end{pmatrix},$

$\widehat{\mu}_1=\begin{pmatrix}0.76\\ 1.008\end{pmatrix},\ \widehat{\Sigma}_1=\begin{pmatrix}1.17 & -0.06\\ -0.06 & 0.21\end{pmatrix},\ \widehat{\mu}_2=\begin{pmatrix}2.87\\ 4.03\end{pmatrix},\ \widehat{\Sigma}_2=\begin{pmatrix}0.56 & 0.009\\ 0.009 & 0.31\end{pmatrix},$

$\widehat{\lambda}_1^{(1)}=0.214\,,\ \widehat{\lambda}_2^{(1)}=1.180,\ \widehat{Z}_1=\begin{pmatrix}-0.07 & -0.99\\ -0.99 & 0.07\end{pmatrix},\ \widehat{\lambda}_1^{(2)}=0.31\,,\ \widehat{\lambda}_2^{(2)}=0.56,\ \widehat{Z}_2=\begin{pmatrix}0.03 & -0.99\\ -0.99 & -0.03\end{pmatrix}.$

The data set is linearly separable and it is represented in figure $3i)a$. The soft margin line d_1 computed by the $SVM1$ algorithm is represented in figure $3i)b$, the value of the resulted margin being $\rho=1.196429$.

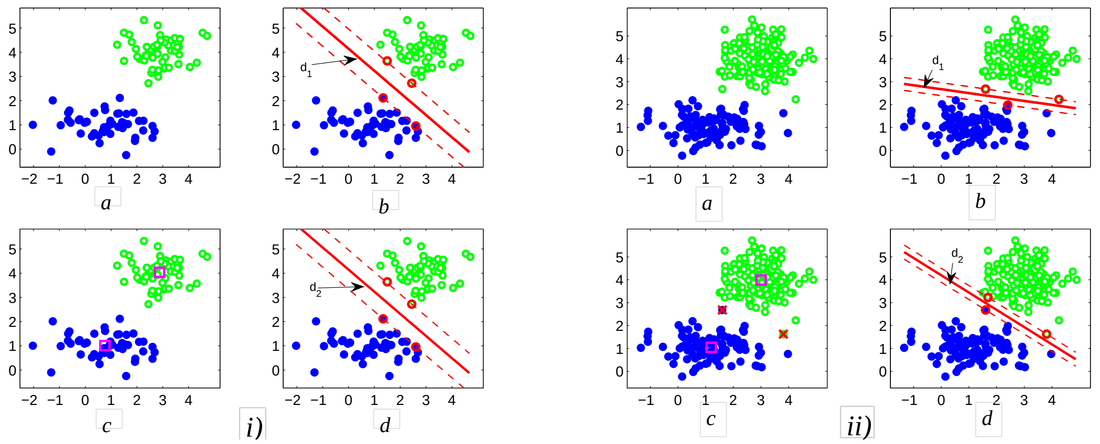


Figure 3: *i) The classification of the data set in test 3; ii) The classification of the data set in test 4.*

The clusters computed by the 2-means algorithm are represented in figure $3i)c$ and they are the same as in initial data set whatever the initial choice of the centers is. So, the statistical characteristics are

$\overline{\mu}_1 = \widehat{\mu}_1,\ \overline{\Sigma}_1 = \widehat{\Sigma}_1,\ \overline{\mu}_2 = \widehat{\mu}_2,\ ,\ \overline{\Sigma}_2 = \widehat{\Sigma}_2,$

$\overline{\lambda}_1^{(1)} = \widehat{\lambda}_1^{(1)},\ \overline{\lambda}_2^{(1)} = \widehat{\lambda}_2^{(1)},\ \overline{Z}_1 = \widehat{Z}_1,\ \overline{\lambda}_1^{(2)} = \widehat{\lambda}_1^{(2)},\ \overline{\lambda}_2^{(2)} = \widehat{\lambda}_2^{(2)},\ \overline{Z}_2 = \widehat{Z}_2,$

and the separating line d_2 computed by the algorithm $SVM1$ and represented in figure $3i)d$ coincides with d_1.

Test 4: $N_1 = 100,\quad N_2 = 150,\ \mu_1 = \begin{pmatrix} 1 \\ 1 \end{pmatrix},\ \Sigma_1 = \begin{pmatrix} 1 & 0 \\ 0 & 0.25 \end{pmatrix},\ \mu_2 = \begin{pmatrix} 3 \\ 4 \end{pmatrix},\ \Sigma_2 = \begin{pmatrix} 0.5 & 0 \\ 0 & 0.5 \end{pmatrix}.$

$\lambda_1^{(1)} = 0.25,\quad \lambda_2^{(1)} = 1,\quad Z_1 = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix},\ \lambda_1^{(2)} = 0.5,\quad \lambda_2^{(2)} = 0.5,\quad Z_2 = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}.$

$\widehat{\mu}_1 = \begin{pmatrix} 1.22 \\ 1.03 \end{pmatrix},\ \widehat{\Sigma}_1 = \begin{pmatrix} 1.04 & -0.03 \\ -0.03 & 0.24 \end{pmatrix},\ \widehat{\mu}_2 = \begin{pmatrix} 2.98 \\ 3.99 \end{pmatrix},\ \widehat{\Sigma}_2 = \begin{pmatrix} 0.48 & -0.01 \\ -0.01 & 0.43 \end{pmatrix}.$

$\widehat{\lambda}_1^{(1)} = 0.24,\ \widehat{\lambda}_2^{(1)} = 1.04,\ \widehat{Z}_1 = \begin{pmatrix} -0.04 & -0.99 \\ -0.99 & 0.04 \end{pmatrix},\ \widehat{\lambda}_1^{(2)} = 0.42,\ \widehat{\lambda}_2^{(2)} = 0.49,\ \widehat{Z}_2 = \begin{pmatrix} -0.27 & -0.96 \\ -0.96 & 0.27 \end{pmatrix}.$

The data set is linear separable and it is represented in figure $3ii)a$. Applying the $SVM1$ we obtain the soft margin line d_1 represented in $3ii)b$ and $\rho = 0.552508$.

The clusters computed by the 2-means algorithm are represented in figure $3ii)c$ and their statistical characteristics are

$\overline{\mu}_1 = \begin{pmatrix} 1.20 \\ 1.04 \end{pmatrix}, \overline{\Sigma}_1 = \begin{pmatrix} 0.98 & -0.04 \\ -0.04 & 0.26 \end{pmatrix},\ \overline{\mu}_2 = \begin{pmatrix} 3.00 \\ 3.98 \end{pmatrix},\ \overline{\Sigma}_2 = \begin{pmatrix} 0.48 & -0.04 \\ -0.04 & 0.45 \end{pmatrix},$

$\overline{\lambda}_1^{(1)} = 0.26,\ \overline{\lambda}_2^{(1)} = 0.98,\ \overline{Z}_1 = \begin{pmatrix} -0.05 & -0.99 \\ -0.99 & 0.05 \end{pmatrix},\ \overline{\lambda}_1^{(2)} = 0.42,\ \overline{\lambda}_2^{(2)} = 0.51,\ \overline{Z}_2 = \begin{pmatrix} -0.60 & -0.79 \\ -0.79 & 0.60 \end{pmatrix}.$

In this case the number of misclassified samples is 2 and the initial centers are randomly selected. The separating line d_2 computed by the algorithm $SVM1$ applied to the data represented by these clusters is represented in figure $3ii)d$.

6 Conclusions and future work

Although to combine a supervised technique to an unsupervised one seems to be meaningless, mainly because they refer to totally different situations, the combined methodology resulted by putting together k-means and SVM methods yielded to an improved classifier. The experimental results point out good performance of the proposed method from both points of view, accuracy and computational complexity. We are optimistic that the research aiming to obtain refined methods by combining supervised, unsupervised and semi-supervised technics has good chances to provide a class of new powerful classification schemes.

It has been already performed a series of tests on different types of classifiers obtained by combining PCA (Principal Component Analysis), ICA (Independent Component Analysis) and kernel based SVM's the results being quite encouraging.

Bibliography

[1] S. Abe, *Support Vector Machines for Pattern Classification*, Springer-Verlag, 2005.

[2] C.J.C. Burges, A Tutorial on Support Vector Machines for Pattern Recognition, *Data Mining and Knowledge Discovery*, 2, pp. 121-167, 1998.

[3] C. Cortes, V. Vapnik, Support-vector networks, *Machine Learning*, 20(3):273-297, 1995.

[4] N. Cristianini, J. Shawe-Taylor, *An Introduction to Support Vector Machines and Other Kernel-based Learning Methods*, Cambridge University Press, 2000.

[5] G. Gan, C. Ma, J. Wu, *Data Clustering: Theory, Algorithms and Applications*, SIAM, 2007.

[6] S.R. Gunn, *Support Vector Machines for Classification and Regression*, University of Southampton, Technical Report, 1998.

[7] L. State, I. Paraschiv-Munteanu, I., *Introducere in teoria statistică a recunoaşterii formelor*, Editura Universitaţii din Piteşti, 2009.

[8] J.B. MacQueen, Some Methods for classification and Analysis of Multivariate Observations, *Proceedings of 5-th Berkeley Symposium on Mathematical Statistics and Probability*, Berkeley, University of California Press, pp. 281-297, 1967.

[9] I. Paraschiv-Munteanu, Support Vector Machine in solving pattern recognition tasks, *Proceedings of First Doctoral Student Workshop in Computer Science*, University of Piteşti, May 2009.

[10] I. Paraschiv-Munteanu, Theoretical approach in performance evaluation of a classsification system, *Scientific Bulletin of the University of Piteşti*, Series Mathematics and Informatics, No. 14, pp. 203-218, 2008.

[11] N. Popescu-Bodorin, Fast K-Means Image Quantization Algorithm and Its Application to Iris Segmentation", *Scientific Bulletin of the University of Piteşti*, Series Mathematics and Informatics, No. 14, 2008.

[12] R. Stoean, C. Stoean, M. Preuss, D. Dumitrescu, Evolutionary multi-class support vector machines for classification, *International Journal of Computers Communications & Control*, Vol.1 Supplement: Suppl. S, pp. 423-428, 2006.

[13] V.N. Vapnik, *The Nature of Statistical Learning Theory*, New York, Springer Verlag, 1995.

[14] V.N. Vapnik, *Statistical Learning Theory*, New York, Wiley-Interscience, 1998.

[15] R. Xu, D.C.II Wunsch, *Clustering*, Wiley&Sons, 2009.

Int. J. of Computers, Communications & Control, ISSN 1841-9836, E-ISSN 1841-9844
Vol. VI (2011), No. 1 (March), pp. 187-195

Using Fixed Priority Pre-emptive Scheduling in Real-Time Systems

D. Zmaranda, G. Gabor, D.E. Popescu, C. Vancea, F. Vancea

Doina Zmaranda, Gianina Gabor, Daniela Elena Popescu
Codruta Vancea, Florin Vancea
University of Oradea
Romania, 410087 Oradea, 1 Universitatii St.
E-mail: {zdoina,gianina,depopescu,cvancea,fvancea}@uoradea.ro

Abstract: For real-time applications, task scheduling is a problem of paramount importance. Several scheduling algorithms were proposed in the literature, starting from static scheduling or cyclic executives which provide very deterministic yet inflexible behaviour, to the so called best-effort scheduling, which facilitates maximum run-time flexibility but allows only probabilistic predictions of run-time performance presenting a non-predictable and non-deterministic solution. Between these two extremes lies fixed priority scheduling algorithms, such as Rate Monotonic, that is not so efficient for real-time purposes but exhibits a predictable approach because scheduling is doing offline and guarantees regarding process deadlines could be obtained using appropriate analysis methods. This paper investigates the use of Rate Monotonic algorithm by making adjustments in order to make it more suitable for real-time applications. The factors that motivate the interest for fixed priority scheduling algorithms such Rate Monotonic when doing with real-time systems lies in its associated analysis that could be oriented in two directions: schedulability analysis and analysis of process interactions. The analyzing process is carried out using a previously implemented framework that allows modelling, simulation and schedulability analysis for a set of real-time system tasks, and some of the results obtained are presented.

Keywords: real-time systems, fixed priority preemptive scheduling.

1 Introduction

Real-time systems are often safety critical and require a high quality design in order to obtain and guarantee the requested properties. The design process consists in building models on which the required system properties are assessed; based on this previously developed models an implementation that preserves these properties is further developed.

In order to develop a large-scale real-time system we must be able to manage both the logical complexity and timing complexity using a highly disciplined approach [10].The problem of dealing with logical complexity is addressed by the several existing software engineering general methodologies [11] while timing complexity represents an issue that is addressed by specific scheduling algorithms.

For real-time systems, two modelling approaches are known in the literature: first of it allows handling of traditional, periodically sampled control systems, and is represented by the so called timed-triggered approach; the second type of model deals with discrete event systems, and it is known as the event-triggered approach. If the main advantage of event-driven approach is flexibility and better resource utilization, the main advantage of time-driven approach is predictability.

In the time-triggered approach, all communication and processing activities are initiated at predetermined points in time: there is only one interrupt and that is the periodic clock interrupt, which partitions the continuum time into sequences of equally spaced granules. Timed-triggered tasks are characterized by a period and a deadline; also, knowledge about task's Worst case Execution Time (WCET) is generally assumed.

On the other side, event-triggered approach is dictated by the external environment: all communication and processing activities are initiated whenever a significant change of state, i.e., an event other than regular event of a clock tick, is noted. The signalling of significant events is realized by the well-known interrupt mechanism. The event-triggered based systems require a scheduling strategy to achieve the appropriate software task that services the event.

In practice, for several real-time applications event-triggered tasks are sporadic, and exhibit a predictable inter-arrival time. Thus, this time could be seen as task period, deadline being smaller or equal with this [15]. In practical situations, when dealing with hard real-time issues, mixed systems are often encountered [13]. For these, a common approach is to model the system as a timed-triggered one, and deal with events as periodic tasks with inter-arrival times considered as their period. Of course, if the system requires handling of urgent events, that implies a pre-emptive scheduling strategy to be able to meet those deadlines.

Consequently, when we are modelling such real-time mixed systems, choosing the right scheduling strategy is an important aspect, and several issues has to be considered [17]: if we are considering time-triggered tasks, a static scheduling proves to be efficient both for scheduling and for the communications purposes. It is important into the design phase to correctly divide system functionality into tasks and further, tasks with long periods should be statically divided into subtasks with shorter periods; if sporadic event-triggered tasks occur and they have shorter deadline that execution time of another task, allowing pre-emption is mandatory and therefore leading to a pre-emptive scheduling approach.

Therefore, using a static and pre-emptive scheduling strategy together with some initial reasonable assumptions when constructing task's model, could provide a solution for analysing and developing mixed systems [8]. In this case, as the static approach, we consider that fixed priority assignment can be adopted without loosing the benefits of the fully static approach.

2 Fixed priority pre-emptive scheduling and real-time systems

Fixed priority scheduling algorithms exhibit a predictable approach: because scheduling is doing offline, guarantees regarding process deadlines could be obtained using appropriate analysis methods [5]. Fixed priority scheduling has been often criticized as being too static by the supporters of best effort scheduling and too dynamic by the supporters of cyclic executives. For building a mixed real-time system from a number of periodic tasks and several sporadic tasks, static priority pre-emptive scheduler implies that at run time the highest priority task is run, this pre-empting other lower priority tasks [6].

From a historical perspective, Rate Monotonic scheduling algorithm is the most appropriate in this sense, because it is pre-emptive and because it is known to be the optimal in their respective classes [14].

Rate Monotonic scheduling algorithm is an example of a priority driven algorithm with static priority assignment [2], in the sense that the priorities of all requests are known before their arrival, the priorities for each task being the same and known a-priori (they are determined only by the period of task).

Therefore, for the time being, Rate Monotonic is used in most practical applications [3]. The reasons for this choice are the following: easy to implement an to analyze; however, the schedulability assessment given by Liu and Layland [12] is sufficient (all task sets that pass the test are guaranteed to be schedulable) but not necessary (a task set that fails to pass the test is not necessarily unschedulable) [12]; more predictable, especially in high overloaded conditions; this prediction is linked to several initial simplifying assumptions and restrictions that Rate Monotonic has: all tasks are independent and periodic, deadline is equal to their period.

The factors that motivated the interest for fixed priority scheduling algorithms such Rate Monotonic when doing with real-time systems lies in its associated analysis, which could be oriented in two directions [3]: *schedulability analysis* based on worst case execution times of the processes; one property of the early utilization schedulability analysis is its simplicity both in concept and in computational complexity. This simplicity comes from the initial assumptions that are based on constraints upon characteristics of all processes (all processes are periodic and must have deadline equal to their period) and assumption that process priorities given by their period (according to rate monotonic policy) are correctly assigned, otherwise analysis is not efficient and *inclusion of aperiodic processes* by making them periodic based on estimated inter-arrival times.

Rate Monotonic priority assignment policy [12] states that process deadlines must be equal with their respective periods $D_i = T_i$. This assignment could be restrictive, especially for hard real-time sporadic tasks that have deadlines not related to their inter-arrival times, and hence they cannot be modelled as simple periodic tasks with period equal with deadline. For this case the following relation holds:

$$C_i \leq D_i \leq T_i \tag{1}$$

where: C_i represents the computation time for task i D_i represents task i deadline T_i represents task i period

A variation of original Rate Monotonic, called Deadline Monotonic assign priorities in inverse order to the task deadlines. Deadline Monotonic algorithm is equivalent with Rate Monotonic when, for all processes $D_i = T_i$. Deadline Monotonic priority assignment is optimal in a similar manner to Rate Monotonic if there is a feasible priority ordering over a set of processes, a deadline monotonic priority ordering over those processes will be also feasible [4]. Both Rate Monotonic and Deadline Monotonic approaches assume that all processes have a common release time: if processes are permitted to have arbitrary offsets, then optimality could be affected [8,14]. Under these circumstances, neither priority assignment is optimal [9]; but, if controlled offset is allowed this property may be not significant affected.

3 Considering the overheads

Even if in most models overhead induced by scheduling is neglected and considered 0, this is not the case in reality. Generally, implementation scheme for fixed priority schedulers implies maintaining at least two queues: a ready queue and a waiting queue. The ready queue contains the tasks that are ready for execution and the waiting queue contains tasks that have already been executed and they are waiting for the next period. The queues are ordered in different ways: the ready queue is ordered based on task priority (the most priority task first - for Rate Monotonic the task with lower period T_i is the most priority task) and the waiting queue is ordered based on the starting time of the task R_i.

If a task from the waiting queue becomes ready for execution, then it is moved to the ready queue according to its priority; if the priority of the first task from the ready queue becomes bigger than the priority of the current task, then a context switch will occur [4]. This leads to two kinds of overheads: *context switching overhead* - being the time needed to pre-empt a task, save its context and load the context of another task and *scheduling overhead* - the time taken to move newly arrived or pre-empted tasks between the two queues.

The common ways to consider these times into a system model implies adding the overhead times to the known times according to the following [6]: for *context switching times*, the simplest way to do it is to increase task computation time C_i of all tasks with the double of estimated time needed to do of doing the context switch C_{sw} :

$$C_i = C_i + 2 * C_{sw} \tag{2}$$

and *for scheduling overhead* the same approach could be used, this time being added the scheduling overhead time C_{sch} :

$$C_i = C_i + C_{sch} \tag{3}$$

Generally, when considering the total of all overheads, an average could be calculated over all tasks, denoted by C_{ov}, and could be added to each computation time:

$$C_i = C_i + C_{ov} \tag{4}$$

This approach of measuring the total system overhead, averaging it and including in all computation times is simple to be modelled and used [16]. Another way to include overheads into the model could be considered and modelled as additional tasks, but this complicates it too much and sometimes is unnecessarily.

When constructing a real-time system model, from the accuracy point of view, it is important to take into consideration these overheads. But, an important aspect is represented also by the possibilities of reducing the overheads impact on the overall system's performance [20]. Because most of these overheads are linked with pre-emption (both context switching and scheduling overheads occur after a task pre-emption), a possibility of doing this could be to reduce the number of pre-emption over system's task set. Generally, one of the weaknesses of Rate Monotonic algorithm comes from the high level of overhead that results due to high pre-emptions. Therefore, reducing unnecessary pre-emptions could have a significant impact on algorithm performance, in the same time by keeping its simplicity [7].

4 Reducing the number of pre-emptions

The idea of the proposed method is based on the observations derived from Rate Monotonic Algorithm usage for scheduling real-time tasks (tasks with deadlines), from which a high number of pre-emptions were noticed. As it is well known, every pre-emption induce a run-time overhead and we consider that, by reducing the number of pre-emptions in the resulting scheduling scheme, the overall runtime overhead decreases, and this could result in an increasing efficiency when we speak about real-time applications.In order to avoid pre-emptions, it is important to know when they occur: generally, they take place when a higher priority task is activated during the execution of a lower priority task. Lower priority tasks would experience more pre-emption than high priority ones, as they stay longer in the ready queue.

To reduce the chance for pre-emption, one possible method implies to reduce the period of time while a task stays into the ready queue. It should be possible to do this, and one method is to delay the activation of the task, by setting R_i to a value > 0; of course, this delay should not have implications to the overall schedulability of the task set, and, consequently, must be done under strict control. In order to derive such a method, based on an algorithm that delays start time for a set of tasks, the following task model is considered: every task is denoted by ζ_i; each task is periodic and the method applies itself only for periodic tasks, the period of task ζ_i is denoted by T_i; C_i represents the worst case execution time (WCET) for task ζ_i; P_i represents the priority of task and the priority of each task is fixed; the ratio C_i/T_i represents the utilization factor of the task ζ_i and represents the fraction of processor time that is used by task ζ_i; the deadline of a task D_i represents a typical task constraint in real-time systems and represents the time before which the task must complete its execution - usually, the deadline of the task is relative, meaning that, from the moment when a task ζ_i arrives, it should finish within D_i time units and in particular, when using Rate Monotonic algorithm, the deadline is considered equal with task period: $D_i = T_i$; task ready (arrival) time denoted by R_i which represents the moment of time when the task ζ_i is ready for execution.

Consequently, the algorithm for ready time modification calculates first the maximum delay time for each task, in order not to affect the task deadline. For task ζ_i that is characterized by a period (and deadline) T_i and has a worst case execution time C_i, the following relation is used:

$$\text{max\_delay}(\zeta_i) = T_i - C_i. \tag{5}$$

Let's denote with S a task set of size n and consider that tasks are ordered by their priority (as they appear in the ready queue):

$$S = \{\zeta_1\ \zeta_2\ .\ .\ .\ \zeta_{n-1}\ \zeta_n\} \tag{6}$$

where ζ_1 has the highest priority and ζ_n the lower one.

The algorithm picks every task from S, in the decreasing order of their priority, and verifies if it is possible to change the ready time R_i; in order to reduce the possibility of pre-emption for the higher priority task (ζ_i), the algorithm starts by delaying it as much as possible [1]. Verification is based on the maximum delay that is possible for each task, calculated as presented in (5) and take into consideration the computation time for tasks that were already delayed in order not to compromise the schedulability for the given task. The algorithm tries to delay as much as possible tasks with high priority, given to the tasks with low priority the chance to be less pre-empted. The set of tasks that have been delayed during execution of algorithm are included in DELAYED (corresponding to the waiting queue) and corresponding delay is calculated (R_i). This will be used further into the simulation tool for modifying ready times. Consequently, we obtained the following structure of the algorithm:

```
$R_1 = max\_delay(\zeta_1) = T_1 - C_1$;
include $(\zeta_1)$ in DELAYED ;
for$(i = 2;\ i \le n;\ i{+}{+})$
 {
   if $((max\_delay(\zeta_i) - \Sigma_{j\ in\ DELAYED}\ C_j) > C_i)$
   {
        $R_i = max\_delay(\zeta_i) - \Sigma_{j\ in\ DELAYED}\ C_j$;
        include$(\zeta_j)$ in DELAYED;
   }
   else
    {
   $R_i = max\_delay(\zeta_i)$;
    }
 }
```

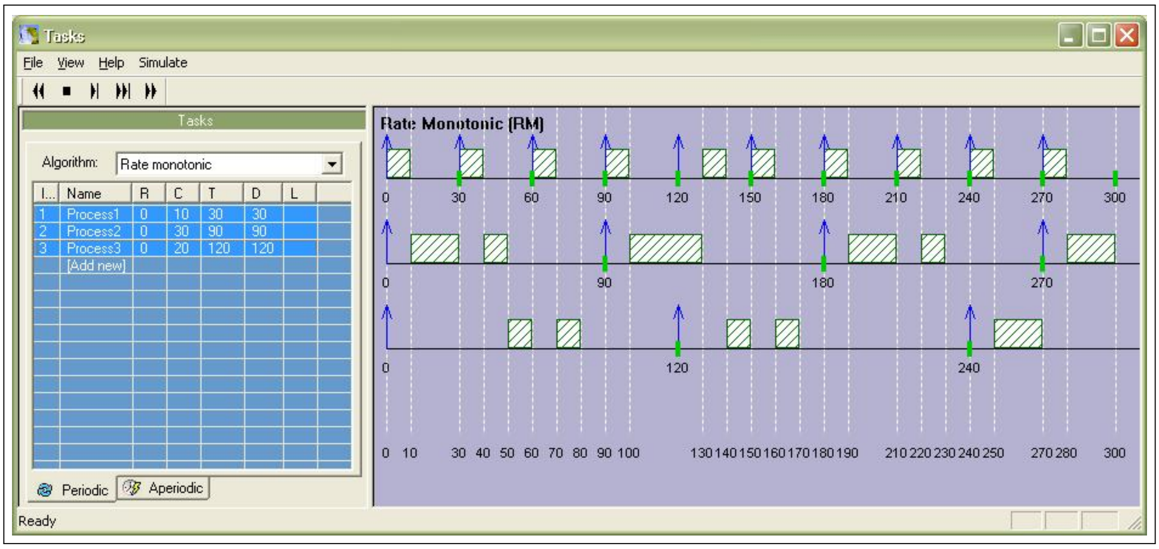
5 Case study analysis and results

The idea of the above algorithm is illustrated considering the case study shown in Figure 1. Each task is represented by its computation time C, period T and deadline D, and it is assumed that release time R for all tasks is null. So, we considered a task set consisting of three tasks ζ_1, ζ_2, and ζ_3 with the following characteristics:

$C_1 = 10;\ T_1 = 30;\ C_2 = 30;\ T_2 = 90;\ C_3 = 20;\ T_3 = 120.$

We simulated the execution of these tasks using the framework developed in [18, 19] and the results obtained are presented in Figure 1. The simulation is carried out using a developed framework which allows modelling, simulation and schedulability analysis for a set of real-time systems tasks. The tool has a graphical user interface for introducing tasks parameters, such as: deadline, execution time, priority (as presented in the left panel) permitting that all these parameters to be saved in a (text) file for each given task, in a specific format. Also, several scheduling algorithms are implemented into the tool, and the user could choose from a list of implemented algorithms. These algorithms are grouped into two categories: for periodic and non-periodic tasks.

For our case study, we consider to work only with Rate Monotonic. The framework uses a built-in simulator that illustrates a graphical representation of the generated trace of execution for the set of tasks, according to the chosen scheduling algorithm (as presented in the right panel). From these results we noticed that task ζ_1 exhibits no pre-emption, ζ_2 exhibits maximum one pre-emption and task ζ_3 maximum 2 pre-emptions. By applying start times modifications according to the proposed algorithm, the following release times for the considered tasks were calculated:



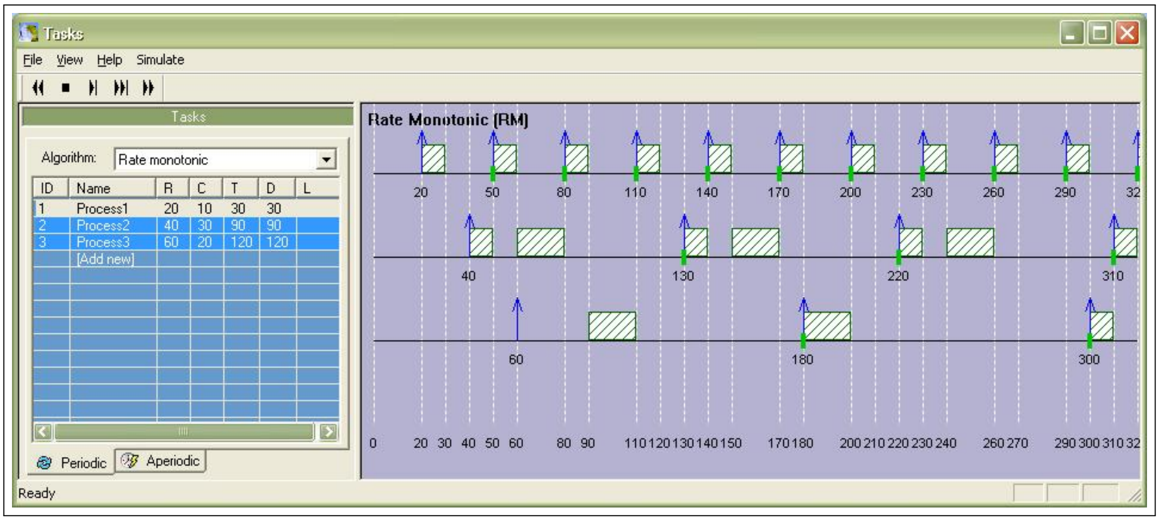
Algorithm: Rate monotonic

I...	Name	R	C	T	D	L
1	Process1	0	10	30	30	
2	Process2	0	30	90	90	
3	Process3	0	20	120	120	
	[Add new]					

Figure 1: Set of 3 tasks scheduled with Rate Monotonic without modifying start (ready) times

$i = 1;$
$R_1 = T_1 - C_1 = 30 - 10 = 20;$
$S = \{\zeta_1\};$
$i = 2;\ \text{max\_delay}(\zeta_2) = T_2 - C_2 = 90 - 30 = 60;$
$60 - 20 = 40 \Rightarrow R_2 = 40;$
$S = \{\zeta_1\ \zeta_2\};$
$i = 3;\ \text{max\_delay}(\zeta_3) = T_3 - C_3 = 120 - 20 = 100;$
$100 - 40 = 60 => R_3 = 60;$
$S = \{\zeta_1\ \zeta_2\ \zeta_3\};$

We modified the release times for our tasks accordingly; consequently, as it is shown into the new simulation presented in Figure 2, we observed that the number of pre-emption for task ζ_3 is reduced to maximum one. It is obvious that the number of pre-emptions for task ζ_3 is reduced by this modification.



Algorithm: Rate monotonic

ID	Name	R	C	T	D	L
1	Process1	20	10	30	30	
2	Process2	40	30	90	90	
3	Process3	60	20	120	120	
	[Add new]					

Figure 2: Set of 3 tasks scheduled with Rate Monotonic with modifying start (ready) times

One thing about our resulting model is no concluding is the delay of the least priority task: based on the algorithm idea, it should not be delayed at all, because so it has a chance to run when other tasks have not, due to their delays, and, therefore, the chance for pre-empting it decreases in these conditions. But, in many cases, an overall lower pre-emption rate is achieved.

6 Conclusions

Starting from the premises that Rate Monotonic algorithm is simpler to implement and exhibits a predictable behaviour resulted from its associated analysis, in this paper, a possible adaptation of Rate Monotonic algorithm is proposed, in order to overcome some of its disadvantages when using it in real-time applications.

One aspect that has impact on the model overall accuracy takes into consideration the overheads implied by the scheduling process by including them into the task's computation times, and a possible way of considering this is proposed. Another aspect, with impact on scheduling overall performance focuses on reducing the number of pre-emptions, accrediting the idea that by doing this, the performance of the algorithm increases. Pre-emptions were reduced based on tasks start times modifications, and an algorithm that controls these start time adjustments was proposed.

The algorithm was tested using several use cases (one example presented in the paper), pseudo-randomly generated and the same conclusion was reached: by controlling tasks release times according to the proposed algorithm, in most cases the number of pre-emptions decreases. Thus, the proposed algorithm together with the implemented tool provides a very powerful analysis framework that can be used in real-time application modelling and further development.

Bibliography

[1] A. Aravind and J. Chelladurai, Activation Adjusted Scheduling Algorithms for Real-Time Systems, *Advances in Systems, Computing Sciences and Software Engineering*, pp. 425-432,

Springer 2006.

[2] N. Audsley, On priority assignment in fixed priority scheduling, Fuzzy Control Rules in Convex Optimization, *Inf. Process. Lett.*, 79(1), pp.39-44, 2001.

[3] N. Audsey, A. Burns, R. Davis, K. Tindell, A. Wellings, *Fixed Priority Preemptive Scheduling: An Historical Perspective, Real Time Systems,* vol. 8, pp. 173-198, 1995.

[4] R.J. Bril, P.J.L. Cuijpers, *Analysis of hierarchical fixed-priority pre-emptive scheduling revisited,* TU/e CS-Report 06-36, 2006.

[5] G. C. Butazzo, Rate Monotonic vs. EDF: Judgement Day, *Real-Time Systems,* 2005.

[6] R.I. Davis, A. Burns, Hierarchical Fixed Priority Pre-Emptive Scheduling, *Proceedings of the 26th IEEE Real Time System Symposium,* IEEE Computer Society, pp. 389-398,2005.

[7] R. Dobrin and G. Fohler, Reducing the Number of Preemptions in Fixed Priority Scheduling, *Proceedings of Euromicro Conference on Real Time Systems,* pp. 144-152, 2004.

[8] J. Goossens, Scheduling of Offset Free Systems, *Real-Time Systems,* 24(2), pp. 239-258, 2003.

[9] J. Goossens, R. Devillers, The no-optimality fo the monotonic priority assignments for hard real-time systems, *Real-Time Systems,* 13(2), pp. 107-126, 1997.

[10] C. Kirch, Principles of Real-Time Programming, *EMSOFT02, LNCS 2491,* Springer-Verlag Berlin, 2002.

[11] J. Kollar, J. Poruban, P. Vaclavik, Evolutionary Nature of Crosscutting Modularity, *Proceedings of the 9th International Conference of Modern Electric Systems,* EMES'07, pp. 43 - 48, 2007.

[12] C. L. Liu and J. W. Layland, Scheduling Algorithms for Multiprogramming in a Hard real Time Environment, *Journal of the ACM,* vol. 20(1), pp. 46-61, 1973.

[13] C. L. Liu, *Real-Time Systems*, Prentice Hall, 2000.

[14] M. Naghibzadeh and K. H. Kim, A modified Version of Rate Monotonic Scheduling Algorithm and its Eficiency Assessment, *Proceedings of the Seventh IEEE Internation Workshop on Object Oriented Real Time Dependent Systems,* pp. 289-294, 2002.

[15] I.Shin, I. Lee, Periodic resource model for compositional real-time guarantees , *Proceedings of 24th IEEE Real Time System Symposium,* (RTSS), pp.2-13, 2003.

[16] S. Saewong, R. Rajkumar, J.P. Lohoczky, M.H. Klein, Analysis of Hierarchical Fixed-Priority Scheduling, *Proceedings of 14th Euromicro Conference on Real Time Systems,* (ECRTS), pp. 152-160, 2002.

[17] K. Somasundaram; S. Radhakrishnan, Task Resource Allocation in Grid using Swift Scheduler, *International Journal of Computer, Communication and Control,* ISSN 1841-9836, E-ISSN 1841-9844, vol. IV, no.2, pp. 158-166, 2009.

[18] D. Zmaranda, G. Gabor, Tool for Modeling and Simulation of Real-Time Systems Behavior, *Proceedings of the 2nd IEEE International Workshop on Soft Computing Applications,* SOFA 2007, Gyula, Hungary - Oradea, Romania, ISBN: 978-1-4244-1608-0, pp. 211-215, 2007.

[19] D. Zmaranda, C. Rusu and M. Gligor, A Framework for Modeling and Evaluating Timing Behaviour for Real-Time Systems, *Proceedings of the International Symposium on Systems Theory - Software Engineering,* SINTES vol III, pp. 514-520, ISBN 973-742-148-5, 2005.

[20] C. Gyorodi, R. Gyorodi, M. Dersidan, L. Bandici, Applying a pattern length constraint on the FP-Growth algorithm, Proceedings of the International Workshop on Soft Computing Applications SOFA 2009, IEEE - Computational Intelligent Society, 29 July - 1 August 2009, Szeged-Hungary, Arad - Romania, IEEE Catalog number CFP0928D-PRT, ISBN 987-1-4244-5054-1, pp. 181-185, 2009

Author index