INRIA

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

Project-Team MODBIO

Computational Models in Molecular Biology

Lorraine

THEME BIO

Activity Report

2006

Table of contents

1. Team

Team Leader
Eric Domenjoud [Research Associate (CR) CNRS]

Administrative Assistants
Aurélie Prévost Aubry [Secretary (SAR) INRIA, until 01/2006]
Sophie Drouot [Secretary (SAR) INRIA, since 02/2006]

Staff members
Yann Guermeur [Research Associate (CR) CNRS]
Fabienne Thomarat [Associate Professor, INPL]

Ph. D. student
Yannick Darcy [UHP, Grant MENRT]

Post-doctoral Fellows
Khalid Benabdeslem [From 03/2006 until 05/2006]
Emmanuel Monfrini [UHP]

Technical staff
Delphine Autard [UHP, until 03/2006]

Student intern
Levoly Fani [ENSEM, from 02/2006]

External Collaborator
Alexander Bockmayr [Professor, Freie Universität Berlin]

2. Overall Objectives

2.1. Introduction

The aim of the project MODBIO is to develop computational models for molecular and cell biology. We are focusing on two types of problems:

- Determining the structure of biological macromolecules,
- Discovering and understanding the function of biological systems.

We approach these questions by combining techniques from constraint programming, combinatorial optimization, hybrid systems, and machine learning.

2.2. Research themes

- Sequence and structural alignment, phylogeny.
- Protein structure prediction and protein docking.
- Modeling alternative splicing.

2.3. Scientific and industrial relations

- Participation in the "Génopole Strasbourg Alsace-Lorraine"
- Participation in the Bioinformatics project of the Région Lorraine
- Participation in the ACI project GENOTO3D
- Participation in the "Décrypthon" programme
- Various national and international collaborations

- Laboratoire "Maturation des ARN et Enzymologie Moléculaire" (MAEM), UMR 7567, Nancy
- Laboratoire de Cristallographie, LCM3B, Nancy
- Institut de Biologie et Chimie des Protéines, IBCP, Lyon
- DFG Research Center Matheon, Berlin, Germany
- Institute for Genomics and Bioinformatics, University of California, Irvine, USA

3. Scientific Foundations

3.1. Constraint programming

Constraint programming [32] is a declarative programming language paradigm that appeared in the late 80's, and which has become more and more popular since then. A *constraint* is a logical formula that defines a relation to be satisfied by the values of the variables the formula contains. For instance, the formula $x + y \leq 1$ expresses that the sum of the values of the variables x and y must be less than or equal to 1.

In *constraint programming*, the user programs with constraints, i.e., he or she describes a problem by a set of constraints, which are connected by *combinators* such as conjunction, disjunction, or temporal operators (`always`). Each constraint gives some *partial* information about the state of the system to be studied. Constraint programming systems allow one to deduce new constraints from the given ones and to compute *solutions*, i.e., values for the variables that satisfy all constraints simultaneously.

One of the main goals of constraint programming is to develop programming languages that allow one to express constraint problems in a natural way, and to solve them efficiently.

3.1.1. Finite domain constraint programming

In our work, we are first interested in constraint problems over finite domains. In this case, the domain of each variable (the set of values it may take) is a finite set of integer numbers. Theory tells us that most constraint problems over finite domains are NP-hard, which means that there is little hope to solve them by algorithms polynomial in the size of the input. In practice, these problems are handled by tree search methods which try successively different valuations of the variables until a solution is found. Because of the exponential number of possible combinations, it is crucial to reduce the search space as much as possible, i.e., to eliminate *a priori* as many valuations as possible.

There exist two generic methods to solve such problems. The first one is classical *integer linear programming* (see also Sect. 3.3), which has been studied in mathematical programming and operations research for more than 40 years. Here, constraints are linear equations and inequalities over the integer numbers. In order to reduce the search space, one typically uses the linear relaxation of the constraint set. Equations and inequalities are first solved over the real numbers, which is much easier; then the information obtained is used to prune the search tree.

The second method is *finite domain constraint programming* which arose in the last 15 years by combining ideas from declarative programming languages and constraint satisfaction techniques in artificial intelligence. In contrast to integer linear optimization one uses, in addition to simple arithmetic constraints, more complex constraints, which are called *symbolic constraints*. For instance, the symbolic constraint `alldifferent`$(x_1, ..., x_n)$ expresses that the values of the variables $x_1, ..., x_n$ must be pairwise distinct. Such a constraint is difficult to express in a compact way using only linear equations and inequalities. Symbolic constraints are handled individually by specific filtering algorithms that reduce the domain of the variables. This information is propagated to other constraints which may further reduce the domains.

A state-of-the-art survey of finite domain constraint programming, with special emphasis on its relation to integer linear programming can be found in [5].

3.1.2. Concurrent constraint programming

In *concurrent* constraint programming (cc) [29], different computation processes may run concurrently. Interaction is possible via the *constraint store*. The store contains all the constraints currently known about the system. A process may *tell* the store a new constraint, or *ask* the store whether some constraint is entailed by the information currently available, in which case further action is taken.

Hybrid concurrent constraint programming (`Hybrid cc`) [28] is an extension of concurrent constraint programming which allows one to model and to simulate the temporal evolution of *hybrid systems*, i.e., systems that exhibit both discrete and continuous state changes. Constraints in `Hybrid cc` may be both algebraic and differential equations. State changes can be specified using the combinators of concurrent constraint programming and default logic. `Hybrid cc` is well-suited to model dynamic biological systems, as shown in [4].

3.2. Statistical learning

Statistical learning theory [31] is one of the fields of inferential statistics the bases of which have been established by V.N. Vapnik in the late 1960s. The goal of this theory is to specify the conditions under which it is possible to "learn" from empirical data obtained by random sampling. Learning amounts to solving a problem of function or model selection. Basically, given a task characterized by a joint probability distribution on pairs made up of observations and labels, and a class of functions, of cardinality ordinarily infinite, the goal is to find in the class a function with optimal performance. Training can thus be reformulated as an optimization problem. In many cases, the objective function is related to the capacity of the class of functions [17]. The learning tasks considered belong to one of the three following areas: pattern recognition (discriminant analysis), function approximation (regression) and density estimation.

This theory considers more specifically two inductive principles. The first one, named empirical risk minimization (ERM) principle, consists in minimizing the training error. If the sample is small, one substitutes to this the structural risk minimization (SRM) principle. It consists in minimizing an upper bound on the expected risk (generalization error), a bound sometimes called a guaranteed risk. This latter principle is implemented in the training algorithms of the support vector machines (SVMs), which currently constitute the state-of-the-art for numerous problems of pattern recognition.

SVMs are connectionist models conceived to compute indicator functions, to perform regression or to estimate densities. They have been introduced during the last decade by Vapnik and co-workers [26], as nonlinear extensions of the maximal margin hyperplane [30]. Their main advantage is that they can avoid overfitting in the case where the size of the sample is small [31], [24].

3.3. Combinatorial optimization and integer programming

"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" [25]. A combinatorial optimization problem can be defined as follows: we are given a ground set N and consider a finite collection of subsets, say $\{S_1, S_2, \cdots, S_m\}$. For each subset S_k there is an objective function value, $f(S_k)$, typically a linear function over the elements in S_k. The task is to find the subset S_k that minimizes $f(S_k)$. Typically, the feasible subsets are represented by inclusion or exclusion of members such that they satisfy certain conditions. Well known examples of combinatorial optimization problems are assignment, covering, cutting stock, knapsack, matching, packing, partitioning, routing, sequencing, scheduling (jobs), shortest path, spanning tree, and traveling salesman problems.

This then becomes a special class of integer programs (IP) whose decision variables are binary valued: $x_i = 1$ if the i-th element is in the optimal solution; otherwise, $x_i = 0$. In this case, feasible subsets have to be expressed by linear constraints. IP formulations are not always easy, and often there is more than one formulation, some better than others. Many good formulations have exponential size.

4. Application Domains

4.1. Molecular biology

Keywords: *Biology*, *biological sequence processing*, *structure prediction*.

Participants: Delphine Autard, Khalid Benabdeslem, Alexander Bockmayr, Yannick Darcy, Levoly Fani, Yann Guermeur, Emmanuel Monfrini, Fabienne Thomarat.

Molecular biology is concerned with the study of three types of biological macromolecules: DNA, RNA, and proteins. Each of these molecules can initially be viewed as a string on a finite alphabet: DNA and RNA are nucleic acids made up of nucleotides A,C,G,T and A,C,G,U, respectively. Proteins are sequences of amino acids, which may be represented by an alphabet of 20 letters.

Molecular biology studies the information flow from DNA to RNA, and from RNA to proteins. In a first step, called *transcription*, a DNA string ("gene") is transcribed into messenger RNA (mRNA). In the second step, called *translation*, the mRNA is translated into a protein: each triplet of nucleotides encodes one amino acid according to the genetic code. The genes of eukaryotic cells are mostly composed of a succession of coding regions, called exons, and non-coding regions, called introns. During transcription, an intermediate step, the *splicing* process, is then necessary to remove the introns from the premessenger RNA. The remaining exons are concatenated yielding the mature RNA molecule. *Alternative splicing* is a regulatory mechanism by which variations in the incorporation of the exons into mRNA leads to the production of different forms of mature mRNAs and consequently to more than one related protein, or isoform.

Biological macromolecules are not just sequences of nucleotides or amino acids. Actually, they are complex three-dimensional objects. DNA shows the famous double-helix structure. RNA and proteins fold into complex three-dimensional structures, which depend on the underlying sequence. RNA is a single-stranded chain of nucleotides. However, a nucleotide in one part of the molecule can base-pair with a nucleotide in another part, following the Watson-Crick complementarity rules. This results in a folding of the molecule. The *secondary structure* of RNA indicates the set of base pairings in the three dimensional structure of the molecule. This information can be represented by a graph.

Proteins have several levels of structure. Above the primary structure (i.e. the sequence) is the *secondary structure*, which involves three basic types: α*-helices*, β*-sheets*, and aperiodic structure elements called *loops*. The spatial relationship of the secondary structures froms the tertiary structure. Several proteins can function together in a protein complex whose structure is referred to as the quaternary structure. A *domain* of a protein is a combination of secondary structure elements with some specific function. It contains an *active site* where an interaction with an external molecule may happen. A protein may have one or several domains.

The ultimate goal of molecular biology is to understand the *function* of biological macromolecules in the life of the cell. Function results from the *interaction* between different macromolecules, and depends on their structure. The overall challenge is to make the leap from sequence to function, through structure: the prediction of structure will help to predict the function.

Thanks to the huge number of gene and protein sequences available in the sequence databases, molecular phylogenetic analyses multiplied since a few decades. Molecular phylogeny [27] is the use of genes or protein sequences to gain information on the evoluationary history of organisms. By comparison of the sequence of a gene in different organisms, the evolutionary history of these sequences can be inferred. Based on the hypothesis that these sequences are orthologs (i.e. come from a same ancestral sequence by speciation events), the evolutionary history of the organisms can also be inferred and be represented by a tree.

5. Software

5.1. M-SVM: Multi-class Support Vector Machine

Participant: Yann Guermeur [correspondent].

We have extended the functionalities and optimized the code of the application devoted to the standard M-SVM (M-SVM1 in [9]), and its variant dedicated to protein sequence processing [10].

5.2. DSVM: Dendogram based Support Vector Machine

Participant: Khalid Benabdeslem [correspondent].

We have developed a first version of a new multi-class discriminant model based on modular task decomposition and bi-class SVMs (DSVM in [14]). This version is available at the following address: http://www710.univ-lyon1.fr/~kbenabde/index_fichiers/Page902.htm.

6. New Results

6.1. Structural risk minimization inductive principle for multi-class discriminant analysis

Keywords: *Statistical learning theory*, *model selection*, *support vector machines*.

Participants: Yannick Darcy, Yann Guermeur, Emmanuel Monfrini.

We have continued our study of the generalization error of large margin multi-class discriminant models, such as MLPs or M-SVMs, laying emphasis on the use of bounds for model selection. The introduction of new generalizations of the notion of Vapnik-Chervonenkis (VC) dimension, generalizations called margin Ψ-dimensions, has enabled us to complete the VC theory of large margin multi-class discriminant models [23]. In parallel, the work on the computation of upper bounds on the empirical risk of M-SVMs based on the leave-one-out procedure has given birth to new results exposed in [22]. All the aforementioned bounds are progressively incorporated in our M-SVM software, where they can be used to select the "soft margin" (regularization) parameter C.

6.2. Multi-class SVMs

Keywords: *Hierarchical clustering*, *multi-class problems*, *support vector machines*.

Participants: Khalid Benabdeslem, Yann Guermeur.

Khalid Benabdeslem has developed a new type of multi-class discriminant model called Dendogram based SVM (DSVM) [14]. Its principle rests on modular task decomposition. It combines a hierarchical clustering with a set of bi-class SVMs. This model proceeds in two phases. First, it consists in building a taxonomy of classes in an ascendant manner thanks to the Ascendant Hierarchical Clustering (AHC) method. Second, each internal node of the taxonomy is endowed with a SVM, in order to separate the two corresponding subsets of categories. A pattern query to be classified is presented to the "root" SVM, and then, according to the output of this machine, to one of the two SVMs of the subsets, and so on down to a "leaf" corresponding to one of the classes. AHC decomposition uses distance measures to investigate the class grouping in binary form at each level of the hierarchy. SVMs require little tuning and yield both high accuracy levels and good generalization for binary classification. As a consequence, the DSVM represents an advance for multi-class problems by both involving an optimal number of SVMs taking part in the decision and rapidly classifying patterns.

We have also written a survey on multi-class SVMs [16].

6.3. Protein structure prediction

Keywords: *Protein secondary and tertiary structures*, *SVMs*, *classification*, *clustering*, *kernel engineering*, *protein cores*, *structural alignment*.

Participants: Khalid Benabdeslem, Levoly Fani, Yann Guermeur.

Knowing the three-dimensional structure of a protein can greatly help to infer its function. Predicting this *tertiary structure* from the sequence of amino acids (or *primary structure*), remains one of the central open problems in structural biology. This is the subject of the "GENOTO3D" project that we coordinate. This year, our main efforts have been concentrated on identifying the fold class of protein sequences of unknown structure. To that end, in collaboration with Gilbert Deléage and Christophe Geourjon, at IBCP, in Lyon, Khalid Benabdeslem has developed an original approach for treating 3D structures of proteins. This approach consists in generating a significant data set for automatic learning and processing a modeling system for fold recognition. In the first step, the method consists in computing a structural alignment for each class (family) of structures using the Combinational Extension (CE) alignment methodology. In the second step, the method derives from each alignment matrix a taxonomy created with AHC. Then, structural cores are extracted from all families and each core represents a prototype for each class of structures [21]. Finally, a neural network is built from a matricial modeling based sequence coding for protein fold recognition [19], [20]. Subsequently, with the help of cores extraction, a data set is generated to build a strong fold recognition system with accuracy exceeding 72% over 100 CATH families [15].

Our collaboration with Nicolas Sapay and Gilbert Deléage on the prediction of amphipathic in-plane membrane (IPM) anchors in motopic proteins has given birth to a new prediction method, "AmphipaSeek" [18], which is available from the website of the PBIL, at the following address : http://npsa-pbil.ibcp.fr/cgi-bin/npsa_automat.pl?page=/NPSA/npsa_amphipaseek.html.

6.4. Alternative splicing

Keywords: *Alternative splicing*, *kernel design*, *model selection*, *multi-class SVMs*.

Participants: Delphine Autard, Yann Guermeur, Emmanuel Monfrini.

In the framework of a project of the Décrypthon programme (see Sect. 7.2 for details), the contribution of MODBIO consists in trying to identify alternative splicing sites. This implies the capability of identifying exons, introns, sequences of splicing machinery binding sites and those of the splice-regulatory element binding sites. We decided to tackle this multi-class pattern recognition problem with a dedicated multi-class SVM. First, we generated three learning databases of annotated human genes thanks to the generic database AltSplice. The huge size of those bases and the technical constraint inherent to the Décrypthon programme (use of the grid provided by IBM) called for a reprogramming of our M-SVM software. We have started analysing the question of the parallelization with Frédéric Desprez at the ENS of Lyon. Meanwhile, we work on original ways of modifying the M-SVM algorithm to make it more efficient on our sequential machines.

6.5. Molecular phylogeny

Keywords: *Molecular phylogeny*, *kernel methods*, *multiple alignments*.

Participant: Fabienne Thomarat.

The first step to infer the evolutionary history of gene or protein sequences consists in building an alignment of all these sequences, i.e. to determine the homology (common ancestry) at each site of the sequences. To that end, biologists generally use the algorithm provided by the clustalw programme. This algorithm is based on the computation of a distance between each pair of sequences, distance which makes use of statistical models of DNA/protein evolution. We have started investigating the interest to substitute a kernel to this distance.

7. Other Grants and Activities

7.1. Regional projects

We participate in the "Génopole Strasbourg Alsace-Lorraine" together with the laboratory MAEM and the IGBMC in Strasbourg.

In the framework of the CPER Lorraine 2000-2006, we participate in the project "Bioinformatics and Applications to Genomics" of the PRST "Intelligence Logicielle". Our partners here are the Laboratory of Crystallography LCM3B (UMR 7036), the "équipe de Dynamique des Assemblages Membranaires" (eDAM, UMR 7565) and the MAEM (UMR 7567) at the University Henri Poincaré, Nancy 1.

7.2. National projects

Since September 2003, we are coordinating a project called GENOTO3D, which is funded by the "Action Concertée Incitative" (ACI) "Masses de Données". The aim of this project is to apply machine learning approaches to the prediction of the tertiary structure of globular proteins. Our partners are the IBCP in Lyon, the LIF in Marseille, the project team SYMBIOSE from IRISA, the LIRMM in Montpellier, and the MIG unit of INRA in Jouy-en-Josas.

Since October 2005, we participate in the project "Développement et utilisation d'approches informatiques et théoriques pour l'analyse des liens existant entre défauts d'épissage et maladies génétiques" funded by the Décrypthon programme: http://www.decrypthon.fr/. Our partner in this project is the MAEM laboratory.

8. Dissemination

8.1. Serving the scientific community

Yann Guermeur has been a member of the program committee of CAp'06. He is an expert for the ANR.

8.2. Teaching

Fabienne Thomarat is Associate Professor at the Ecole Nationale Supérieure des Mines de Nancy / Institut National Polytechnique de Lorraine (engineering school, master of engineering school). She is in charge of one option (Bioinformatics) at the Department of Computer Science.

Yann Guermeur has been teaching bioinformatics in the M2P speciality "Génomique et Informatique" of the Master "Sciences de la Vie et de la Santé" (SVS), at the UHP.

9. Bibliography

Major publications by the team in recent years

[1] E. Althaus, A. Caprara, H.-P. Lenhof, K. Reinert. *Multiple sequence alignment with arbitrary gap costs: Computing an optimal solution using polyhedral combinatorics.*, in "Proc. European Conference on Computational Biology", Bioinformatics, vol. 18, n^o Supplement 2, October 2002, p. S4–S16.

[2] E. Althaus, K. Mehlhorn. *Traveling Salesman-Based Curve Reconstruction in Polynomial Time*, in "SIAM Journal on Computing", vol. 31, n^o 1, 2001, p. 27–66.

[3] E. Balas, A. Bockmayr, N. Pisaruk, L. Wolsey. *On unions and dominants of polytopes*, in "Mathematical Programming, Ser. A", vol. 299, 2004, p. 223-239.

[4] A. Bockmayr, A. Courtois. *Using hybrid concurrent constraint programming to model dynamic biological systems*, in "18th International Conference on Logic Programming, ICLP'02, Copenhagen", Springer, LNCS 2401, 2002, p. 85-99.

[5] A. BOCKMAYR, J. N. HOOKER. *Constraint Programming*, in "12: Discrete Optimization", K. AARDAL, G. NEMHAUSER, R. WEISMANTEL (editors). , Handbooks in Operations Research and Management Science, chap. 10, Elsevier, 2005, p. 559–600.

[6] A. BOCKMAYR, V. WEISPFENNING. *Solving numerical constraints*, in "Handbook of Automated Reasoning", A. ROBINSON, A. VORONKOV (editors). , vol. 1, chap. 12, Elsevier, 2001, p. 751-842.

[7] Y. GUERMEUR, A. ELISSEEFF, D. ZELUS. *Bound on the risk for M-SVMs*, in "Statistical Learning, Theory and Applications", 2002, p. 48–52.

[8] Y. GUERMEUR, C. GEOURJON, P. GALLINARI, G. DELÉAGE. *Improved performance in protein secondary structure prediction by inhomogeneous score combination*, in "Bioinformatics", vol. 15, n^{o} 5, 1999, p. 413–421.

[9] Y. GUERMEUR. *Combining discriminant models with new multi-class SVMs*, in "Pattern Analysis and Applications", vol. 5, n^{o} 2, 2002, p. 168–179.

[10] Y. GUERMEUR, A. LIFCHITZ, R. VERT. *A kernel for protein secondary structure prediction*, in "Kernel Methods in Computational Biology", B. SCHÖLKOPF, K. TSUDA, J.-P. VERT (editors). , The MIT Press, 2004, p. 193–206.

[11] Y. GUERMEUR, H. PAUGAM-MOISY. *Théorie de l'apprentissage de Vapnik et SVM, Support Vector Machines*, in "Apprentissage Automatique", M. SEBBAN, G. VENTURINI (editors). , Hermès, 1999, p. 109–138.

[12] V. Y. LUNIN, A. URZHUMTSEV, A. BOCKMAYR. *Direct phasing by binary integer programming*, in "Acta Crystallographica Section A", vol. 58, 2002, p. 283-291.

[13] P. MUSÉ, F. SUR, F. CAO, Y. GOUSSEAU, J.-M. MOREL. *An a contrario decision method for shape element recognition*, Technical report, n^{o} 2004-16, CMLA, ENS Cachan, 2004.

Year Publications

Articles in refereed journals and book chapters

[14] K. BENABDESLEM, Y. BENNANI. *Dendogram based SVM for multi-class classification*, in "Journal of Computing and Information Technology (CIT)", vol. 14, n^{o} 4, 2006, p. 291–297.

[15] K. BENABDESLEM, G. DELÉAGE, C. GEOURJON. *Dendogram based SVM for multi-class classification*, in "Bioinformatics", (submitted).

[16] Y. GUERMEUR. *SVM multiclasses*, in "Support Vector Machines et autres méthodes à noyau", S. CANU, C. RICHARD, M. DAVY, A. RAKOTOMAMONJY (editors). , (to appear), Hermès, 2007.

[17] Y. GUERMEUR, O. TEYTAUD. *Estimation et contrôle des performances en généralisation des réseaux de neurones*, in "Apprentissage Connexionniste", Y. BENNANI (editor). , chap. 10, Hermès, 2006, p. 279–342.

[18] N. SAPAY, Y. GUERMEUR, G. DELÉAGE. *Prediction of amphipathic in-plane membrane anchors in monotopic proteins using a SVM classifier*, in "BMC Bioinformatics", vol. 7, n^{o} 255, 2006.

Publications in Conferences and Workshops

[19] K. Benabdeslem, G. Deléage, C. Geourjon. *A Neural Network System based on Structural Alignment and Clustering for Proteins Fold Recognition*, in "ECCB'05", 2005.

[20] K. Benabdeslem, G. Deléage, C. Geourjon. *Cores extraction based neural network for proteins fold recognition*, in "JOBIM'05", 2005.

[21] K. Benabdeslem, G. Deléage, C. Geourjon. *Alignement structural et classification hiérarchique pour l'extraction des cœurs structuraux*, in "BIO-EGC'06", 2006.

[22] Y. Darcy, E. Monfrini, Y. Guermeur. *Borne "rayon-marge" sur l'erreur "leave-one-out" des SVM multi-classes*, in "JdS'06", 2006.

Internal Reports

[23] Y. Guermeur. *Large margin multi-category discriminant models and scale-sensitive Ψ-dimensions*, revised in 2006, Research Report RR-5314, INRIA, September 2004, https://hal.inria.fr/inria-00070686/en/.

References in notes

[24] C. Burges. *A tutorial on support vector machines for pattern recognition*, in "Data Mining and Knowledge Discovery", vol. 2, n^{o} 2, June 1998, p. 121–167.

[25] W. J. Cook, W. H. Cunningham, W. R. Pulleyblank, A. Schrijver. *Combinatorial Optimization*, Wiley, 1998.

[26] C. Cortes, V. Vapnik. *Support-Vector Networks*, in "Machine Learning", vol. 20, 1995, p. 273–297.

[27] J. Felsenstein. *Inferring Phylogenies*, Sinauer, 2004.

[28] V. Gupta, R. Jagadeesan, V. Saraswat. *Computing with Continuous Change*, in "Science of computer programming", vol. 30, n^{o} 1-2, 1998, p. 3-49.

[29] V. A. Saraswat. *Concurrent constraint programming*, ACM Doctoral Dissertation Awards, MIT Press, 1993.

[30] V. Vapnik. *Estimation of Dependences Based on Empirical Data.*, Springer-Verlag, N.Y., 1982.

[31] V. Vapnik. *Statistical learning theory*, John Wiley & Sons, Inc., N.Y., 1998.

[32] P. van Hentenryck, V. Saraswat. *Strategic directions in constraint programming*, in "ACM Computing Surveys", vol. 28, n^{o} 4, 1996, p. 701–726.