Doktorandské dny ’12

Ústav informatiky AV ČR, v.v.i.

Jizerka

24. - 26. září 2012

vydavatelství Matematicko-fyzikální fakulty
Univerzity Karlovy v Praze

Ústav informatiky AV ČR, v.v.i., Pod Vodárenskou věží 2, 182 07 Praha 8

ISBN *– not yet –*

Doktorandské dny Ústavu informatiky AV ČR, v. v. i., se konají nepřetržitě od roku 1996 a poskytují doktorandům, podílejícím se na odborných aktivitách Ústavu informatiky, možnost prezentovat výsledky jejich odborného studia. Současně poskytuje prostor pro oponentní připomínky k přednášené tematice a použité metodologii práce ze strany přítomné odborné komunity.

Z jiného úhlu pohledu, toto setkání doktorandů podává průřezovou informaci o odborném rozsahu pedagogických aktivit, které jsou realizovány na pracovištích či za spoluúčasti Ústavu informatiky.

Jednotlivé příspěvky sborníku jsou uspořádány podle jmen autorů. Uspořádání podle tematického zaměření nepovažujeme vzhledem k rozmanitosti jednotlivých témat za účelné.

Vedení Ústavu informatiky jakožto organizátor doktorandských dnů věří, že toto setkání mladých doktorandů, jejich školitelů a ostatní odborné veřejnosti povede ke zkvalitnění celého procesu doktorandského studia zajišťovaného v součinnosti s Ústavem informatiky a v neposlední řadě k navázání a vyhledání nových odborných kontaktů.

1. září 2012

Obsah / Contents

Towards Surrogate Assisted Estimation of Distribution Algorithms

Post-Graduate Student:
MGR. LUKÁŠ BAJER

Faculty of Mathematics and Physics
Charles University in Prague
Malostranské náměstí 25
118 00 Prague 1, CZ

bajer@cs.cas.cz

Supervisor:
DOC. RNDR. ING. MARTIN HOLEŇA, CSC.

Institute of Computer Science of the ASCR, v. v. i.
Pod Vodárenskou věží 2
182 07 Prague 8, CZ

martin@cs.cas.cz

Field of Study:
Theoretical Computer Science

This work was supported by the Czech Science Foundation (GAČR), grant No. P202/11/1368, and Grant Agency of the Charles University (GA UK), grant No. 278511/2011.

Abstract

Estimation of distribution algorithms (EDAs) have become promising kinds of evolutionary algorithms. They are based on sampling an estimated distribution of the better of the solutions from the last generation – iteration of the algorithm. Learning the distribution and sampling is used instead of reproduction common in traditional evolutionary algorithms. In many real-world optimization tasks, objective-function-evaluation of any solution is very expensive, so the lowest possible number of such evaluations is desired. We propose using surrogate models in combination with EDAs as a method of reducing the evaluation costs.

1. Introduction

Probabilistic or linkage models describing relationships between explanatory variables appeared in combination with evolutionary algorithms (EAs) roughly fifteen years ago. Probabilistic models of distribution of promising solutions are built according to the chosen individuals from the current population and then randomly sampled forming a new generation. These algorithms are called Estimation of Distribution Algorithms (EDAs) [1] or Probabilistic Model Building Genetic Algorithms (PMBGAs).

Even though the number of objective function evaluations during EDAs optimization can be lower than in the cases when simpler evolutionary algorithms are used, it is usually still rather high. Moreover, in many real-world applications every fitness evaluation often means substantial amount of time or money. Therefore, we propose using surrogate models of the fitness function in combination with EDAs. Using such models is a well-known technique which can help to decrease the number of fitness evaluations during optimization process.

Surrogate models are trained on gathered data from already evaluated individuals forming an approximating model of the fitness function, so they can be used as a substitution of the original costly function. Since such models are not accurate, the total number of generations needed to get sufficiently near-optimum solution is generally higher. However, as most of the fitness evaluations use a surrogate model, the total number of *original* fitness evaluations very often decreases.

Using surrogate models in EDAs to speed up the convergence has already been proposed by Sastry [2, 3]. In this article, this approach is summarized and further research directions are outlined. The paper is divided in following sections. In the next two sections, the general concept of EDAs and surrogate modeling is briefly presented. The fourth section then combines these approaches together and proposes further steps.

2. Basic Principles of EDAs

The rough structure of the most of the evolutionary algorithms and EDAs are rather similar. The general pseudocode of EDAs is outlined in Figure 1. Here, steps (1), (2) and (3) are the same as in many evolutionary algorithms while steps (4) and (5) are typical particularly for EDAs.

1: $P_0 \leftarrow$ randomly generate m individuals
2: **for** $k = 1, 2, \ldots$ until a stopping criterion is met **do**

3: $pool \leftarrow$ select $n \leq m$ individuals from P_{k-1} according to the selection method
4: $p_k(\mathbf{x}) = p(\mathbf{x} \mid pool) \leftarrow$ estimate the probability distribution of an individual based on the selected individuals (in $pool$)
5: $P_k \leftarrow$ sample new population from $p_k(\mathbf{x})$
6: **end for**

Figure 1: Estimation of distribution algorithm

The main difference between EDAs and EAs lies in the method how they generate new individuals according to the previous generation. Whereas traditional EAs, for example genetic algorithms, try to implicitly combine building blocks representing promising parts of genetic code of already found good solutions by genetic operations (crossover, mutation) [4], EDAs try to find correlations among variables in an explicit way.

The joint probabilistic distribution of the input variables is estimated, forming a model of distribution. Having this model, generating new individuals is relatively easy. However, estimating of the distribution with the model is often a bottleneck of EDAs; especially when the problem being solved is hard and complex dependencies among variables have to be determined.

2.1. Probabilistic Graphical Models

The majority of present EDAs estimate the probability distribution with probabilistic graphical models [1, 5]. These models make use of directed acyclic graphs (DAG) where each node corresponds to one input variable X_i, and the arcs define dependencies between variables. From the conditional (in)dependence defined by the DAG, the factorization of the joint probability distribution of the variables can be expressed as

$$\rho(x_1, \ldots, x_n \mid \boldsymbol{\theta}_S) = \prod_{i=1}^{n} \rho(x_i \mid \mathbf{pa}_i^S, \boldsymbol{\theta}_i). \qquad (1)$$

Here, ρ denotes generalized probability distribution which stands for mass probability $p(X_i = x_i^k)$ for discrete random variables and density function $f(x_i)$ for continuous X_i.

The most frequent representatives of probabilistic graphical models are Bayesian networks for discrete variables and Gaussian networks for continuous variables.

2.2. Current Variants of EDAs

Todays variants of EDAs can be distinguished according to complexness of interactions among variables, and different variants for discrete and continuous variables have been developed.

The simplest algorithms consider all the variables independent. For discrete dimensions PBIL [6], UMDA [7] and cGA [8] exist, UMDA$_c$ is a continuous variant of the second one.

Algorithms whose variables are able to depend on one predecessor are, for example, MIMIC [9], COMIT [10], or BMDA [11].

Multiple dependencies are able to be expressed by BOA (Bayesian Estimation Algorithm) and its variants [12] – probably the most vividly developing discrete EDA today. Other multiple-dependencies-EDAs are, for example, EBNA [13] or FDA [14]. Continuous versions are rather few but some of them exist: EGNA [15] or rBOA [16].

Recently, copulas have been proposed as a model of joint probability distribution for EDAs [17, 18].

3. Surrogate Modelling

Approximation of the fitness function with some regression model is a common cure for tasks when empirical objective function has to be used. These *surrogate models* simulate behaviour of the original function while being much cheaper and much less time consuming to evaluate. As a surrogate model, any suitable regression model can be used [19, 20, 21].

In connection with evolutionary optimization, artificial neural networks of the type multilayer perceptrons [22] and networks with radial basis functions [23, 24] have been particularly popular. We have already used the latter kind of networks in connection with genetic algorithms [25].

3.1. Evolution Control

Evolution control (EC) determines how the original fitness function and the surrogate model are combined during the optimization. In the literature [22], individual and generation based approaches are distinguished.

At the beginning of each generation, *individual-based* EC evaluates all the individuals with the approximating model at first. Then, some of these individuals are chosen for re-evaluation with the original fitness function.

The second type is *generation-based* EC. The basic idea of this approach is rather simple: generations are grouped into cycles of a fixed length λ. In each cycle, η of the generations are controlled by the original fitness function and the rest by the approximate model.

4. Surrogate-Assisted EDAs

Speeding up convergence of the estimation of distribution algorithms by usage of surrogate models has been first suggested in the last decade. In the work of Sastry, Pelikan and Goldberg [2] (p. 6), the term *evaluation relaxation* is used with the same idea: less accurate, but inexpensive objective function is used instead of some of the costly original fitness function evaluations.

The basic pseudocode of EDAs was already described above. Here in Figure 2, the code is supplemented with fitness evaluation of the population (lines 2 and 10) and surrogate model building (lines 7–9).

1: $P_0 \leftarrow$ randomly generate m individuals
2: evaluate the population P_0 with the *original fitness*
3: **for** $k = 1, 2, \ldots$ until a stopping criterion is met **do**
4: $pool_k \leftarrow$ select $n \leq m$ individuals from P_{k-1} according to the selection method
5: $p_k(\mathbf{x}) = p(\mathbf{x} \mid pool_k) \leftarrow$ estimate the probability distribution of an individual based on the selected individuals (in $pool_k$)
6: $P_k \leftarrow$ sample new population from $p_k(\mathbf{x})$
7: **if** enough original evaluated individuals **then**
8: build or upgrade the *surrogate model*
9: **end if**
10: evaluate the individuals in P_k – either with the *original fitness*, or with the *surrogate model*
11: **end for**

Figure 2: Surrogate-assisted estimation of distribution algorithm

The line 10 of the pseudocode forms a crucial part of the surrogate-assisted EDA. At this point, evaluation of much more individuals is possible with the (inaccurate) surrogate model without additional original fitness evaluations.

Both kinds of evolution control – individual and generation based – can be used to combine the original fitness function and the surrogate model. Our former studies [25] showed that individual-based EC can save (at least in the provided applications) more original evaluations than generation-based EC which is probably caused by more frequent updates of the surrogate model in terms of the generations of the evolution algorithm. Nevertheless, detailed research of both approaches in combination with EDAs will form a basic step of our further research.

4.1. Surrogates Using Models of Distributions

The authors of [2] distinguish between so called *exogenous* and *endogenous* surrogates. While the first type embraces general approximate models common in other kinds of evolutionary algorithms, such as neural networks or polynomials, the latter surrogate models make use of the structure of the estimated model of distribution and relationships between variables.

For example, eCGA [26] splits the explanatory variables into disjoint groups during its model-building phase. A simple example of endogenous surrogate model would build surrogate submodels on each such a group of variables, and combine these submodels together.

Utilizing the models of distribution for the construction of surrogate models of the fitness function will be studied further. Different combinations of surrogate models and models of distribution will be examined; for example, usage of Gaussian processes as a surrogate model might be particularly beneficial since distribution of subsets of variables might be possible to compare with distributions from the internal models of distribution from EDAs.

5. Conclusion

This brief paper has introduced estimation of distribution algorithms – novel kinds of evolutionary algorithms from the last fifteen years – and the concept of surrogate modelling which use approximation models of objective function to speed up optimization of costly functions. Usage of these models in combination with EDAs has not yet been fully studied, so directions for future work are outlined in this article.

References

[1] P. Larrañaga and J. A. Lozano, *Estimation of distribution algorithms: A new tool for evolutionary computation*. Kluwer, 2002.

[2] K. Sastry, M. Pelikan, and D. Goldberg, "Efficiency enhancement of estimation of distribution algorithms," *Studies in Computational Intelligence (SCI)*, vol. 33, pp. 161–185, 2006.

[3] K. Sastry, C. F. Lima, and D. E. Goldberg, "Evaluation relaxation using substructural information and linear estimation," in *Proceedings of the 8th annual conference on Genetic and evolutionary computation*, pp. 419–426, ACM, 2006.

[4] D. E. Goldberg, *Genetic algorithms in search, optimization, and machine learning*. Addison-Wesley, Jan. 1989.

[5] S. L. Lauritzen, *Graphical models*. Oxford University Press, 1996.

[6] S. Baluja, "Population-based incremental learning: A method for integrating genetic search based function optimization and competitive learning," Tech. Rep. Tech. Rep. No. CMU-CS-94-163, Carnegie Mellon University, Pittsburgh, 1994.

[7] H. Mühlenbein and G. Paass, "From recombination of genes to the estimation of distributions I. binary parameters," in *Parallel Problem Solving from Nature IV*, pp. 178–187, 1996.

[8] G. R. Harik, F. G. Lobo, and D. E. Goldberg, "The compact genetic algorithm," in *Proceeding of the International Conference on Evolutionary Computation (IECE)*, pp. 523–528, IEEE New York, 1998.

[9] J. S. De Bonet, C. L. Isbell, and P. Viola, "MIMIC: finding optima by estimating probability densities," *Advances in neural information processing systems*, pp. 424–430, 1997.

[10] S. Baluja and S. Davies, "Combining multiple optimization runs with optimal dependency trees," Tech. Rep. Tech. Report CMU-CS-97-157, Carnegie Mellon University, School of Computer Science, 1997.

[11] M. Pelikan and H. Mühlenbein, "The bivariate marginal distribution algorithm," *Advances in Soft Computing-Engineering Design and Manufacturing*, pp. 521–535, 1999.

[12] M. Pelikan and D. Goldberg, "Hierarchical bayesian optimization algorithm," in *Scalable Optimization via Probabilistic Modeling*, pp. 63–90, 2006.

[13] R. Etxeberria and P. Larrañaga, "Global optimization with bayesian networks," in *Second Symposium on Artificial Intelligence (CIMAF-99)*, pp. 332–339, 1999.

[14] H. Mühlenbein, T. Mahnig, and A. O. Rodriguez, "Schemata, distributions and graphical models in evolutionary optimization," *Journal of Heuristics*, vol. 5, no. 2, pp. 215–247, 1999.

[15] P. Larrañaga, R. Etxeberria, J. A. Lozano, and J. M. Pena, "Optimization by learning and simulation of bayesian and gaussian networks," Tech. Rep. EHU-KZAAIK-4, University of the Basque Country, 1999.

[16] C. W. Ahn, "Real-coded bayesian optimization algorithm," in *Advances in Evolutionary Algorithms*, pp. 85–124, 2004.

[17] A. Cuesta-Infante, R. Santana, J. Hidalgo, C. Bielza, and P. Larrañaga, "Bivariate empirical and n-variate archimedean copulas in estimation of distribution algorithms," in *Evolutionary Computation (CEC), 2010 IEEE Congress on*, pp. 1–8, 2010.

[18] R. Salinas-Gutiérrez, A. Hernández-Aguirre, and E. R. Villa-Diharce, "Dependence trees with copula selection for continuous estimation of distribution algorithms," in *Proc. of the Genetic and Evolutionary Computation Conference (GECCO '11)*, (Dublin, Ireland), July 2011.

[19] S. Hosder, L. Watson, and B. Grossman, "Polynomial response surface approximations for the multidisciplinary design optimization of a high speed civil transport," *Optimization and Engineering*, vol. 2, no. 4, pp. 431–452, 2001.

[20] D. Buche, N. Schraudolph, and P. Koumoutsakos, "Accelerating evolutionary algorithms with gaussian process fitness function models," *IEEE Trans. on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, vol. 35, no. 2, pp. 183–194, 2005.

[21] H. Ulmer, F. Streichert, and A. Zell, "Model-assisted Evolution Strategies," *Knowledge Incorporation in Evolutionary Computation*, p. 333, 2005.

[22] Y. Jin, M. Hüsken, M. Olhofer, and B. Sendhoff, "Neural networks for fitness approximation in evolutionary optimization," *Knowledge Incorporation in Evolutionary Computation*, p. 281, 2005.

[23] Z. Zhou, Y. Ong, P. Nair, and A. Keane, "Combining global and local surrogate models to accelerate evolutionary optimization," *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, vol. 37, no. 1, pp. 66–76, 2007.

[24] Y. S. Ong, P. B. Nair, A. J. Keane, and K. W. Wong, "Surrogate-assisted evolutionary optimization frameworks for high-fidelity engineering design problems," *Knowledge Incorporation in Evolutionary Computation*, pp. 307–332, 2004.

[25] L. Bajer and M. Holeňa, "Surrogate model for continuous and discrete genetic optimization based on rbf networks," in *Intelligent Data Engineering and Automated Learning – IDEAL 2010* (C. Fyfe, ed.), vol. 6283 of *Lecture Notes in Computer Science*, pp. 251–258, Springer, Sep 2010.

[26] G. Harik, "Linkage learning via probabilistic modeling in the ecga," Tech. Rep. IlliGAL Report No. 99010, Urbana, 1999.

On Online Labeling with Polynomially Many Labels

Post-Graduate Student:
MGR. VLADIMÍR ČUNÁT

Faculty of Mathematics and Physics
Charles University in Prague
Malostranské náměstí 25
118 00 Prague 1, CZ

vcunat@gmail.com

Supervisor:
PROF. RNDR. VÁCLAV KOUBEK, DRSC.

Faculty of Mathematics and Physics
Charles University in Prague
Malostranské náměstí 25
118 00 Prague 1, CZ

koubek@ktiml.mff.cuni.cz

Field of Study:
Theoretical Computer Science

This is a joint work with Martin Babka, Jan Bulánek, Kichal Koucký, and Michael Saks; accepted into ESA'12.

Abstract

In the online labeling problem with parameters n and m we are presented with a sequence of n *keys* from a totally ordered universe U and must assign each arriving key a label from the label set $\{1, 2, \ldots, m\}$ so that the order of labels (strictly) respects the ordering on U. As new keys arrive it may be necessary to change the labels of some items; such changes may be done at any time at unit cost for each change.The goal is to minimize the total cost. An alternative formulation of this problem is the *file maintenance problem*, in which the items, instead of being labeled, are maintained in sorted order in an array of length m, and we pay unit cost for moving an item.

For the case $m = cn$ for constant $c > 1$, there are known algorithms that use at most $O(n \log(n)^2)$ relabelings in total [3], and it was shown recently that this is asymptotically optimal [1]. For the case of $m = \theta(n^C)$ for $C > 1$, algorithms are known that use $O(n \log n)$ relabelings. A matching lower bound was claimed in [2]. That proof involved two distinct steps: a lower bound for a problem they call *prefix bucketing* and a reduction from prefix bucketing to online labeling. The reduction seems to be incorrect, leaving a (seemingly significant) gap in the proof. In this paper we close the gap by presenting a correct reduction to prefix bucketing. Furthermore we give a simplified and improved analysis of the prefix bucketing lower bound. This improvement allows us to extend thelower bounds for online labeling to the case where the number m of labels is superpolynomial in n. In particular, for superpolynomial m we get an asymptotically optimal lower bound $\Omega((n \log n)/(\log \log m - \log \log n))$.

References

[1] J. Bulánek, M. Koucký, and M. Saks, "Tight lower bounds for the online labeling problem". In *Proc. of 66th Symp. of Theory of Computation*, *(STOC'12)*, H.J. Karloff and T. Pitassi, eds., pp. 1185–1198. ACM, 2012.

[2] P. F. Dietz, J. I. Seiferas, and J. Zhang, "A tight lower bound for online monotonic list labeling". *SIAM J. Discrete Mathematics*, 18(3):626–637, 2004.

[3] A. Itai, A. G. Konheim, and M. Rodeh, "A sparse table implementation of priority queues". In *Proc. of 8th International Colloquium on Automata, Languages and Programming*, *(ICALP'81)* S. Even and O. Kariv, eds., vol. 115 of *LNCS*, pp. 417–431. Springer, 1981.

Conformal Sets in Neural Network Regression

Post-Graduate Student:
ING. RADIM DEMUT

Department of Mathematics
Faculty of Nuclear Science and Physical Engineering
Czech Technical University
Trojanova 13

120 00 Prague 2, CZ

demut@seznam.cz

Supervisor:
DOC. ING. RNDR. MARTIN HOLEŇA, CSC.

Institute of Computer Science of the ASCR, v. v. i.
Pod Vodárenskou věží 2

182 07 Prague 8, CZ

martin@cs.cas.cz

Field of Study:
Mathematical Engineering

The paper will be presented at the ITAT 2012 conference and will be published in the conference proceedings.

Abstract

This paper is concerned with predictive regions in regression models, especially neural networks. We use the concept of conformal prediction (CP) to construct regions which satisfy given confidence level. Conformal prediction outputs regions, which are automatically valid, but their width and therefore usefulness depends on the used nonconformity measure. A nonconformity measure should tell us how different a given example is with respect to other examples. We define nonconformity measures based on some reliability estimates such as variance of a bagged model or local modeling of prediction error. We also present results of testing CP based on different nonconformity measures showing their usefulness and comparing them to traditional confidence intervals.

References

[1] Z. Bosnic and I. Kononenko, "Comparison of approaches for estimating reliability of individual regression predictions", *Data & Knowledge Engineering*, pp. 504–516, 2008.

[2] A. Gammerman, G. Shafer, and V. Vovk, "Algorithmic learning in a random world", *Springer Science+Business Media*, 2005.

[3] E. Uusipaikka, "Confidence Intervals in Generalized Regression Models", *Chapman & Hall*, 2009.

[4] H. Papadopoulos, V. Vovk, and A. Gammerman, "Regression conformal prediction with nearest neighbours", *Journal of Artificial Intelligence Research*, vol. 40, pp. 815–840, 2011.

[5] S. Valero, E. Argente, et al., "DoE framework for catalyst development based on soft computing techniques", *Computers and Chemical Engineering*, vol. 33, No. 1, pp. 225–238, 2009.

Model-Assisted Evolutionary Optimization with Active Learning and Fixed Evaluation Batch Size

Post-Graduate Student:
ING. VIKTOR CHARYPAR

Department of Mathematics
Faculty of Nuclear Science and Physical Engineering
Czech Technical University
Trojanova 13

120 00 Prague 2, CZ

charypar@gmail.com

Supervisor:
DOC. ING. RNDR. MARTIN HOLEŇA, CSC.

Institute of Computer Science of the ASCR, v. v. i.
Pod Vodárenskou věží 2

182 07 Prague 8, CZ

martin@cs.cas.cz

Field of Study:
Mathematical Engineering

This work was supported by the Grant Agency of the Czech Technical University in Prague, grant No. SGS12/196/OHK3/3T/14 as well as the Czech Science Foundation grant No. 201/08/0802.

Abstract

Some black-box optimization problems involve long-running simulations or expensive experiments as the goal function. To enable use of evolutionary algorithms, surrogate models are used to reduce the number of function evaluations. In adaptive model building strategies, some individuals are selected for true function evaluation in order to improve the model. When the experiment or simulation requires a fixed size batch of solutions to evaluate, traditional selection strategies either cannot be used or couple the batch size with the EA generation size. We propose a queue based method for for model-assisted optimization using active learning of a kriging model, where individuals are selected based on the model predictor error estimate. The method was tested on standard benchmark problems and the effects of batch size was studied. Results indicate that the proposed method significantly reduces the number of true fitness evaluation compared to a traditional EA.

1. Introduction

Evolutionary optimization algorithms are a popular class of optimization techniques suitable for various optimization problems. One of their main advantages is the ability to find optima of black-box functions – functions that are not explicitly defined and only their input/output behavior is known from previous evaluations of a finite number of points in the input space. This is typical for applications in engineering, chemistry or biology, where the evaluation is performed in a form of computer simulation or physical experiment.

The main disadvantage for such applications is the very high number of evaluations of the objective function (called fitness function in the evolutionary optimization context) needed for an evolutionary algorithm (EA) to reach the optimum, which makes them impractical for many applications. For example, in the evolutionary optimization of catalytic materials [1], an evaluation for one generation of the algorithm takes between several days and several weeks and costs thousands of euros.

The typical solution to this problem is performing only a part of all evaluations using the true fitness function and using a response-surface model as its replacement for the rest. This approach is called surrogate modeling. When using a surrogate model, only a small portion of all the points that need to be evaluated is evaluated using the true objective function (simulation or experiment) and for the rest, the model prediction is assigned as the fitness value. The model is built using the information from the true fitness evaluations.

Since the fitness function is assumed to be highly non-linear the modeling methods used are non-linear as well. Some of the commonly used methods include artificial neural networks, radial basis functions, regression trees, support vector machines or Gaussian processes [2].

Furthermore, some experiments require a fixed number of samples to be processed at one time. This presents its own set of challenges for adaptive sampling and is the main concern of this paper. We present an evolutionary optimization method assisted by a variant of a Gaussian-process-based interpolating model called kriging. In order to best use the evaluation budget, our approach uses active learning methods in selecting individuals to eva-

luate using the true fitness function. The key feature of the approach is support for online and offline batch evaluation with arbitrary batch size independent of the generation size of the EA.

2. Model-Assisted Evolutionary Optimization

Since the surrogate model used as a replacement for the fitness function in the EA is built using the results of the true fitness function evaluations, there are two competing objectives. First, we need to get the most information about the underlying relations in the data, in order to build a precise model of the fitness function. If the model does not capture the features of the fitness function correctly, the optimization can get stuck in a fake optimum or generally fail to converge to a global one. Second, we have a limited budget for the true fitness function evaluations. Using many points from the input space to build a perfect model can require more true fitness evaluations than not employing a model at all.

In the general use of surrogate modeling, such as design space exploration, the process of selecting points from the input space to evaluate and build the model upon is called sampling [2]. Instead of a traditional upfront sampling schemes based on the theory of design of experiments (DoE), adaptive sampling strategies are used, where a model is improved during the course of the optimization based on previous fitness function evaluations [2]. In an model-assisted evolutionary optimization algorithm, the adaptive sampling decisions change from selecting which points to evaluate to whether to evaluate a given point selected by the EA with the true fitness function or not. There are two general approaches to this choice: the generation-based approach and the individual-based approach. We will discuss both, with emphasis on the latter, a variant of which is used in the method we propose in Section 4.

2.1. Generation-Based Approach

In the generation-based approach the decision whether to evaluate an individual point with the true fitness function is made for the whole generation of the evolutionary algorithm. The optimization takes the following steps.

1. An initial N_i generations of the EA is performed, yielding sets $\mathcal{G}_1, \ldots, \mathcal{G}_{N_i}$ of individuals $(\mathbf{x}, f_t(\mathbf{x}))$, f_t being the true fitness function.
2. The model M is trained on the individuals $(\mathbf{x}, f_t(\mathbf{x})) \in \bigcup_{i=1}^{N_i} \mathcal{G}_i$.
3. The fitness function f_t is replaced by a model prediction f_M.
4. T generations are performed evaluating f_M as the fitness function.
5. One generation is performed using f_t yielding a set $\mathcal{G}_j$ of individuals. (initially $j = N_i + 1$)
6. The model is retrained on the individuals $(\mathbf{x}, f_t(\mathbf{x})) \in \bigcup_{i=1}^{j} \mathcal{G}_i$
7. Steps 4–6 are repeated until the optimum is reached.

The amount of true fitness evaluations in this approach is dependent on the population size of the EA and the frequency of control generations T, which can be fixed or adaptively changed during the course of the optimization [3]. For problems requiring batched evaluation this approach has the advantage of evaluating the whole generation, the size of which can be set to the size of the evaluation batch. The main disadvantage of the generation-based strategy is that not all individuals in the control generation are necessarily beneficial to the model quality and the expensive true fitness evaluations are wasted.

2.2. Individual-Based Approach

As opposed to the generation-based approach, in the individual-based strategy, the decision whether to evaluate a given point using the true fitness function or the surrogate model is made for each individual separately. There are several possible approaches to individual-based sampling, the most used of which is pre-selection. In each generation of the EA, number of points, which is a multiple of the population size, is generated and evaluated using the model prediction. The best of these individuals form the next generation of the algorithm. The optimization is performed as follows.

1. An initial set of points $\mathcal{S}$ is chosen and evaluated using the true fitness function f_t.
2. Model M is trained using the pairs $(\mathbf{x}, f_t(\mathbf{x})) \in \mathcal{S}$
3. A generation of the EA is run with the fitness function replaced by the model prediction f_M and a population $\mathcal{O}_i$ of size qp is generated and evaluated with f_M, where p is the desired population size for the EA and q is the pre-screening ratio. Initially, $i = 1$.
4. A subset $\mathcal{P} \subset \mathcal{O}$ is selected according to a selection criterion.

5. Individuals from $\mathcal{P}$ are evaluated using the true fitness function f_t.
6. The model M is retrained using $\mathcal{S} \cup \mathcal{P}$, the set $\mathcal{S}$ is replaced with $\mathcal{S} \cup \mathcal{P}$, and the EA resumes from step 3.

Another possibility, called the best strategy [4], is to replace $\mathcal{S}$ with $\mathcal{S} \cup \mathcal{O}$ instead of just $\mathcal{P}$ in step 6 after re-evaluating the set $\mathcal{O} \setminus \mathcal{P}$ with f_M (after the model M has been re-trained). This also means using the population size qp in the EA.

The key piece of this approach is the selection criterion (or criteria) used to determine which individuals from set $\mathcal{O}$ should be used in the following generation of the algorithm. There are a number of possibilities, let us discuss the most common.

An obvious choice is selecting the best individuals based on the fitness value. This results in the region of the optimum being sampled thoroughly, which helps finding the true optimum. On the other hand, the regions far from the current optimum are neglected and a possible better optimum can be missed. To sample the areas of the fitness landscape that were not explored yet, space-filling criteria are used, either alone or in combination with the best fitness selection or other criteria.

All the previous criteria have the fact that they are concerned with the optimization itself in common. A different approach is to use the information about the model, most importantly its accuracy, to decide which points of the input space to evaluate with the true fitness function in order to most improve it. This approach is sometimes called active learning.

2.3. Active Learning

Active learning is an approach that tries to maximize the amount of insight about the modeled function gained from its evaluation while minimizing the number of evaluations necessary. The methods are used in the general field of surrogate modeling as efficient adaptive sampling strategies. The terms adaptive sampling and active learning are often used interchangeably. We will use the term active learning for the methods based on the characteristics of the surrogate model itself, such as accuracy.

The active learning methods are most often based on the local model prediction error, such as cross-validation error. Although some methods are independent of the model, for example the LOLA-Voronoi method [5], most of them depend on the model used. The kriging model used in our proposed method offers an estimate of the local model accuracy by giving an error estimate of its prediction.

3. Kriging Meta-Models

The kriging method is an interpolation method originating in geostatistics [6], based on modeling the function as a realization of a stochastic process [7].

In the ordinary kriging, which we use, the function is modeled as a realization of a stochastic process

$$Y(\mathbf{x}) = \mu_0 + Z(\mathbf{x}) \tag{1}$$

where $Z(\mathbf{x})$ is a stochastic process with mean 0 and covariance function $\sigma^2\psi$ given by

$$cov\{Y(\mathbf{x}+\mathbf{h}), Y(\mathbf{x})\} = \sigma^2\psi(\mathbf{h}), \tag{2}$$

where σ^2 is the process variance for all $\mathbf{x}$. The correlation function $\psi(\mathbf{h})$ is then assumed to have the form

$$\psi(\mathbf{h}) = \exp\left[-\sum_{l=1}^{d} \theta_l |\mathbf{h}_l|^{p_l}\right], \tag{3}$$

where $\theta_l, l = 1, \ldots, d$, where d is the number of dimensions, are the correlation parameters. The correlation function depends on the difference of the two points and has the intuitive property of being equal to 1 if $\mathbf{h} = \mathbf{0}$ and tending to 0 when $\mathbf{h} \to \infty$. The θ_l parameters determine how fast the correlation tends to zero in each coordinate direction and the p_l determines the smoothness of the function.

The ordinary kriging predictor based on n sample points $\{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$ with values $\mathbf{y} = (y_1, \ldots, y_n)'$ is then given by

$$\hat{y}(\mathbf{x}) = \hat{\mu_0} + \psi(\mathbf{x})'\mathbf{\Psi}^{-1}(\mathbf{y} - \hat{\mu_0}\mathbf{1}), \tag{4}$$

where $\psi(\mathbf{x})' = (\psi(\mathbf{x} - \mathbf{x}_1), \ldots, \psi(\mathbf{x} - \mathbf{x}_n))$, $\mathbf{\Psi}$ is an $n \times n$ matrix with elements $\psi(\mathbf{x}_i - \mathbf{x}_j)$, and

$$\hat{\mu_0} = \frac{\mathbf{1}'\mathbf{\Psi}^{-1}\mathbf{y}}{\mathbf{1}'\mathbf{\Psi}^{-1}\mathbf{1}}. \tag{5}$$

An important feature of the kriging model is that apart from the prediction value it can estimate the prediction error as well. The kriging predictor error in point $\mathbf{x}$ is given by

$$s^2(\mathbf{x}) = \hat{\sigma}^2\left[1 - \psi'\mathbf{\Psi}^{-1}\psi + \frac{(1 - \psi'\mathbf{\Psi}^{-1}\psi)^2}{\mathbf{1}'\mathbf{\Psi}^{-1}\mathbf{1}}\right] \tag{6}$$

where the kriging variance is estimated as

$$\hat{\sigma}^2 = \frac{(\mathbf{y} - \hat{\mu_0}\mathbf{1})\mathbf{\Psi}^{-1}(\mathbf{y} - \hat{\mu_0}\mathbf{1})}{n}. \tag{7}$$

The parameters θ_l and p_l can be estimated by maximizing the likelihood function of the observed data.

For the derivation of the Equations 4 - 7 as well as the MLE estimation of the parameters the reader may consult a standard stochastic process based derivation by Sacks et al. in [7] or a different approach given by Jones in [8].

4. Method Description

In this section we will describe the proposed method for kriging-model-assisted evolutionary optimization with batch fitness evaluation. Our main goal was to decouple the true fitness function sampling from the EA iterations based on an assumption that requiring a specific number of true fitness evaluations in every generations of the EA forces unnecessary sampling.

In the generation-based approach, some of the points may be unnecessary to evaluate, as they will not bring any new information to the surrogate model. The individual-based approach is better suited for the task, as it chooses those points from each generation, which are estimated to be the most valuable for the model. There is still the problem of performing a given number of evaluations in every generation, although there might not be enough valuable points to select from.

The method we propose achieves the desired decoupling by introducing an evaluation queue. The evolutionary algorithm uses the model prediction at all times and when a point, in which the model's confidence in its prediction is low, is encountered, it is added to the evaluation queue. Once there are enough points in the queue, all the points in it are evaluated and the model is re-trained using the results. The optimization takes the following course.

1. Initial set $\mathcal{S}$ of b samples is selected using a chosen initial design strategy and evaluated using the true fitness function f_t
2. An initial kriging model M is trained using pairs $(\mathbf{x}, f_t(\mathbf{x})) \in \mathcal{S}$.
3. The evolutionary algorithm is started, with the model prediction f_M as the fitness function.
4. For every prediction $f_M(\mathbf{x}) = \hat{y}_M(\mathbf{x})$, an estimated improvement measure $c(s^2_M(\mathbf{x}))$ is computed from the error estimate $s^2_M(\mathbf{x})$. If $c(s^2_M(\mathbf{x})) > t$, an improvement threshold, the point is added to the evaluation queue $\mathcal{Q}$.
5. If the queue size $|\mathcal{Q}| \geq b$, the batch size, all points $\mathbf{x} \in \mathcal{Q}$ are evaluated, the set $\mathcal{S}$ is replaced by $\mathcal{S} \cup \{(\mathbf{x}, f_t(\mathbf{x})\}$ and the EA is resumed.
6. Steps 4 and 5 are repeated until the goal is reached, or a stall condition is fulfilled.

The b and t parameters, as well as the function $c(s^2)$, are chosen before running the optimization. Note that the evaluation in step 5 can be performed either immediately, i.e. online, or offline. In offline evaluation, after filling the evaluation queue, the EA is stopped when the current iteration is finished and the control is returned to the user. After obtaining the fitness values for the samples in the sample queue (e.g. by performing an experiment), the user can manually add the samples and resume the EA from the last generation.

To estimate the improvement, which evaluation of a given point will bring, we use a simple measure of estimated improvement – standard deviation (STD) – based on the kriging predictor error estimate, computed directly as its square root

$$STD(x) = \sqrt{\hat{s^2_M}(\mathbf{x})}. \tag{8}$$

The measure captures only the model's estimate of the error of its own prediction (based on the distance from the known samples). As such, it does not take into account the value of the prediction itself and can be considered a measure of the model accuracy.

An important weakness of the measure is that it is based on the model prediction. If the modeled function is deceptive, the model can be very inaccurate while estimating a low variance. A good initial sampling of the fitness function is therefore very important.

5. Results and Discussion

The proposed method was tested using simulations on three standard benchmark functions. We studied the model evolution during the course of the optimization and investigated the optimal choice of batch size for problems where such a choice is possible.

For testing, we used the genetic algorithm implementation from the global optimization toolbox for the Matlab environment and the implementation of an ordinary kriging model from the SUMO Toolbox [9]. The parameters of the supporting methods, e.g. the genetic algorithm itself, were kept on their default values provided by the implementation.

Because the EA itself is not deterministic, each test was performed 20 times and the results we present are statistical measures of this sample. As a performance measure we use the number of true fitness evaluations used to reach a set goal in all tests. We also track the proportion of the 20 runs that reached the goal before various limits (time, stall, etc.) took effect.

5.1. Benchmark Functions

Since the evolutionary algorithms and optimization heuristics in general are often used on black-box optimization, where the properties of the objective function are unknown, it is not straightforward to asses their quality on real world problems. It has therefore become a standard practice to test optimization algorithms and their modifications on specially designed testing problems.

These benchmark functions are explicitly defined and their properties and optima are known. They are often designed to exploit typical weaknesses of optimization algorithms in finding the global optimum. We used three functions found in literature [10]. Although we performed our tests in two dimensions we give general multidimensional definitions of the functions.

First of the functions used is the De Jong's function. It is one of the simplest benchmarks, it is continuous, convex and unimodal and is defined as

$$f(\mathbf{x}) = \sum_{i=1}^{n} x_i^2 \tag{9}$$

The domain is restricted to a hypercube $-10 \leq x_i \leq 10, i = 1, \ldots, n$. The function has one global optimum $f(\mathbf{x}) = 0$ in point $\mathbf{x} = \mathbf{0}$. The De Jong's function was primarily used as a proof of concept test.

As a second benchmark, we used the Rosenbrock's function, also called Rosenbrock's valley. The global optimum is inside a long parabolic shaped valley, which is easy to find. Finding the global optimum in that valley however is difficult [10]. The function has the following definition

$$f(\mathbf{x}) = \sum_{i=1}^{n} [100(x_{i+1} + x_i^2)^2 + (1 - x_i)^2] \tag{10}$$

The domain of the function is restricted to a hypercube $-2 \leq x_i \leq 2, i = 1, \ldots, n$. It has one global optimum $f(\mathbf{x}) = 0$ in $\mathbf{x} = \mathbf{1}$.

Finally, the third function used as a benchmark is the Rastrigin's function. It is based on the De Jong's function with addition of cosine modulation, which produces a high number of regularly distributed local minima and makes the function highly multimodal. The function is defined as

$$f(\mathbf{x}) = 10n + \sum_{i=1}^{n} [x_i^2 - 10\cos(2\pi x_i)] \tag{11}$$

The domain is restricted to $-5 \leq x_i \leq -5, i = 1, \ldots, n$. The global optimum $f(\mathbf{x}) = 0$ is in $\mathbf{x} = \mathbf{1}$.

5.2. Model Evolution

As the basic illustration of how the model evolves during the course of the EA, let us consider an example test run using the Rosenbrock's function. For this experiment we set the batch size of 15, estimated improvement threshold of 0.001 and the target fitness value of 0.001 as well. The target was reached at the point $(0.9909, 0.9824)$ using 90 true fitness evaluations. A genetic algorithm without a surrogate model needed approximately 3000 evaluations to reach the goal in several test runs.

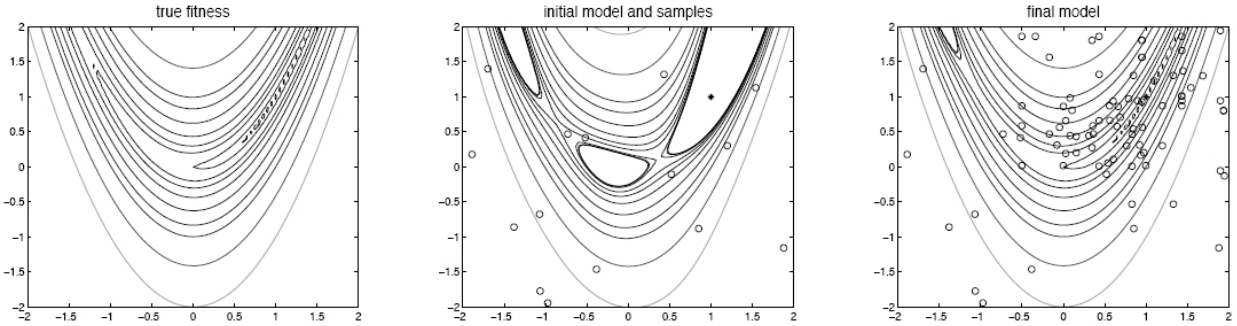


Figure 1: The original fitness function, the initial model and the final model

The model evolution is shown in Figure 1. The true fitness function is shown on the left, the initial model is in the middle and the final model on the right. The points where the true fitness function was sampled are denoted with circles an the optimum is marked with a star.

5.3. Batch Size

In order to study the batch size effect on the optimization, a number of experiments were performed with different batch sizes. The only option to achieve a given batch size is to set the population size in a standard GA, in our method however, the settings are independent so a population size of 30, which proved efficient, was used in all of the tests.

For comparison, we also performed tests with the standard genetic algorithm without a model. Results of these simulations are shown in Table 1.

function	evals (1q)	evals (med)	evals (3q)	goal	reached
De Jong	60	60	120	0.01	1
Rosenbrock	60	125	310	0.1	1
Rastrigin	260	370	580	0.1	0.85

Table 1: GA performance on benchmark functions without a model - number of evaluations to reach the goal and a proportion of 20 runs in which the goal was reached

The results on the De Jong's functions show that apart from small batch sizes (up to 10), the optimization is successful in all runs. Our method helps stabilize the EA for small batch sizes and for batch sizes above 15 the algorithm finds the optimum using a single batch. For a standard GA this strong dependence arises for batch sizes above 40 and the algorithm reaches the goal in the second generation, evaluating twice as many points.

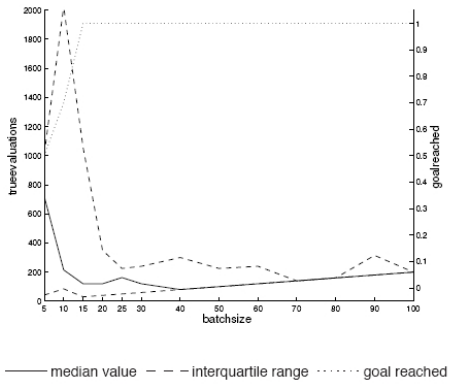


Figure 2: batch size effect on Rosenbrock's function optimization using standard GA - fitness evaluations and proportion of runs reaching the goal

For the Rosenbrock's function we get the intuitive result that setting the batch size too low leads to more evaluations or a failure to reach the goal, while large batch sizes do not improve the results and waste true fitness evaluations. The comparison is shown in Figures 2 and 3 (note the different scales). Overall the method reduces the number of true evaluations from hundreds to tens for the Rosenbrock's function, while slightly reducing the success rate of the computation.

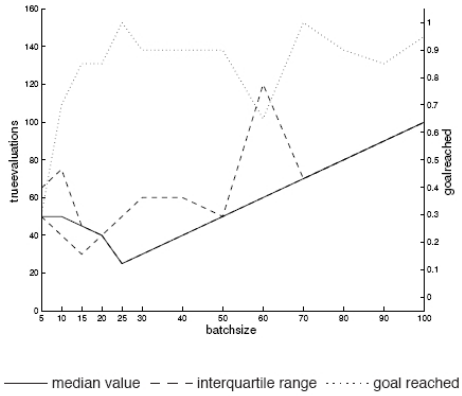


Figure 3: batch size effect on Rosenbrock's function optimization with a surrogate model - true fitness evaluations and proportion of runs reaching the goal

The Rastrigin's function proved difficult to optimize even without a surrogate model. The number of true fitness evaluations was reduced approximately three times in the area of the highest success rate with batch size of 70 (Figure 4). We attribute the method's difficulty optimizing the Rastrigin's function to the fact that the kriging model is local and thus it requires a large number of samples to capture the function's complicated behavior in the whole input space. When the initial sampling is misleading, which is more likely for the Rastrigin's function, both the model prediction and estimated improvement are wrong.

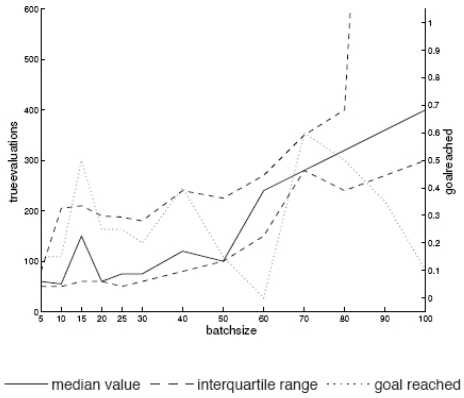


Figure 4: batch size effect on Rastrigin's function optimization with a surrogate model - true fitness evaluations and proportion of runs reaching the goal

In order to prevent bad initial sampling a subset of tests was conducted using an integer multiple of batch size. Figure 5 shows results for Rastrigin function with double initial batch size. Larger initial batch size stabilizes the method. Success rate increased from around 30% to 60% even for smaller batch sizes, which is close to what a simple GA achieved, while maintaining the number of true evaluations low. The fact that a larger initial batch will be evaluated even in cases where a small batch would suffice can be considered a disadvantage of this approach.

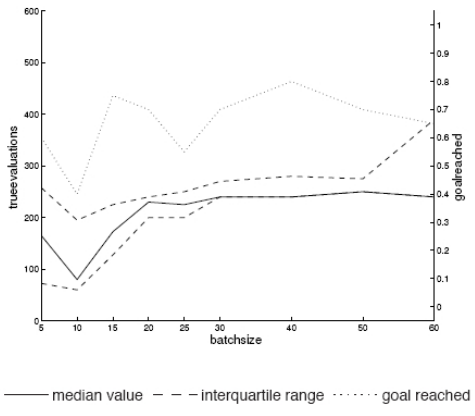


Figure 5: batch size effect on Rastrigin's function optimization with a surrogate model and double initial batch size - true fitness evaluations and proportion of runs reaching the goal

The results suggest that the best batch size is highly problem-dependent. The experimental results support the intuition that batches too small are bad for the initial sampling of the model and batches too large slow down the model improvement by evaluating points that it would not be necessary to evaluate with smaller batches. The proposed method is also very sensitive to good initial sample selection, which is the most usual reason for it to fail to find the optimum. Combining a larger initial batch with a smaller batch during the optimization helps alleviate the problem.

6. Conclusions

In this paper we presented a method for model-assisted evolutionary optimization with a fixed batch size requirement. To decouple the sampling from the EA iterations and support an individual-based approach while keeping a fixed evaluation batch size, the method uses an evaluation queue. The candidates for true fitness evaluations are selected by an active learning method using a measure of estimated improvement of the model quality based on the model prediction error estimate.

The results suggest that small batch sizes perform better when the objective function is simple, while causing bad initial sampling, which can be successfully solved using a larger initial batch. The future development of this work should include experiments with a different initial sample distribution than random as well as comparison of the method with other ways of employing a surrogate model in the optimization and other model-assisted optimization methods.

The method brings promising results, reducing the number of true fitness evaluations to a large degree for some of the benchmark functions, however its success is highly dependent on the optimized function and its initial sampling.

References

[1] M. Baerns and M. Holeňa, *Combinatorial Development of Solid Catalytic Materials: Design of High-Throughput Experiments, Data Analysis, Data Mining.*, Catalytic science series. Imperial College Press, 2009.

[2] D. Gorissen, *Grid-enabled Adaptive Surrogate Modeling for Computer Aided Engineering.* PhD thesis, Ghent University, University of Antwerp, 2009.

[3] Y. Jin, M. Olhofer, and B. Sendhoff, "Managing approximate models in evolutionary aerodynamic design optimization". In *Evolutionary Computation, 2001. Proceedings of the 2001 Congress on*, volume 1, pp. 592– 599. Ieee, 2001.

[4] L. Gräning, Y. Jin, and B. Sendhoff, "Efficient evolutionary optimization using individual-based evo-

lution control and neural networks: A comparative study". In *ESANN*, pp. 273–278, 2005.

[5] K. Crombecq, L. De Tommasi, D. Gorissen, and T. Dhaene, " A novel sequential design strategy for global surrogate modeling". In *Winter Simulation Conference*, WSC '09, pp. 731–742. Winter Simulation Conference, 2009.

[6] G. Matheron, "Principles of geostatistics". *Economic geology*, 58(8): 1246–1266, 1963.

[7] J. Sacks, W.J. Welch, T.J. Mitchell, and H.P. Wynn, "Design and analysis of computer experiments". *Statistical science*, 4(4): 409–423, 1989.

[8] D.R. Jones, "A taxonomy of global optimization methods based on response surfaces". *Journal of Global Optimization*, 21: 345—383, 2001.

[9] D. Gorissen, I. Couckuyt, P. Demeester, T. Dhaene, and K. Crombecq, "A surrogate modeling and adaptive sampling toolbox for computer based design". *The Journal of Machine Learning Research*, 11:2051–2055, 2010.

[10] M. Molga and C. Smutnicki, *Test functions for optimization needs*, 2005.

Linearization of Proofs in Propositional Hilbert Systems

Post-Graduate Student:
MGR. KAREL CHVALOVSKÝ
Institute of Computer Science of the ASCR, v. v. i.
Pod Vodárenskou věží 2
182 07 Prague 8, CZ
Department of Logic, Charles University
Celetná 20
116 42 Prague 1, CZ
chvalovsky@cs.cas.cz

Supervisor:
MGR. MARTA BÍLKOVÁ, PH.D.
Institute of Computer Science of the ASCR, v. v. i.
Pod Vodárenskou věží 2
182 07 Prague 8, CZ
Department of Logic, Charles University
Celetná 20
116 42 Prague 1, CZ
marta.bilkova@ff.cuni.cz

Field of Study:
Logic

The work was supported by grant P202/10/1826 of the Czech Science Foundation and by the long-term strategic development financing of the Institute of Computer Science (RVO 67985807). These results were presented at CiE 2012 in Cambridge and submitted to a journal.

Abstract

Let us have a propositional Hilbert-style proof system containing axioms (strictly speaking schemata of axioms) B (prefixing) and B′ (suffixing)

(B) $(\varphi \to \psi) \to ((\chi \to \varphi) \to (\chi \to \psi))$

(B′) $(\varphi \to \psi) \to ((\psi \to \chi) \to (\varphi \to \chi))$

with implicit substitution and modus ponens as the only rule. We prove that any proof in such a proof system can be transformed into a linear proof. A proof is linear if it uses only a modified version of modus ponens: from φ and $\varphi \to \psi$ derive ψ, where φ can only be an instance of an axiom or assumption.

As prefixing and suffixing are provable in many propositional logics we can obtain similar property for many sets of axioms by adding B and B′. However, a new linear proof can be significantly longer than the original proof. It means that this result is unlikely to be used for the actual proof search, but it can be used for some metamathematical purposes.

Detekce fotovoltaických zdrojů s nestandardním chováním

doktorand:
MGR. IVAN KASANICKÝ

Ústav informatiky AV ČR, v. v. i.
Pod Vodárenskou věží 2
182 07 Praha 8

Matematicko-fyzikální fakulta
Karlova univerzita
Sokolovská 83
186 75 Praha 8

kasanicky@cs.cas.cz

školitel:
RNDR. KRYŠTOF EBEN, CSC.

Ústav informatiky AV ČR, v. v. i.
Pod Vodárenskou věží 2
182 07 Praha 8

eben@cs.cas.cz

obor studia:
Pravděpodobnost a matematická statistika

Abstrakt

Elektřina z fotovoltaických elektráren (FVE) patří mezi neregulované zdroje a proto musí být plně vyčerpána dříve než se začne využívat energie z ostatních (neobnovitelných) zdrojů. Pro úspěšné regulováni distribuční sítě a trhu s elektřinou je tak potřeba v každé chvíli vědět co nejpřesněji, kolik energie fotovoltaické zdroje právě dodávají do sítě. Z různých důvodů však nejsou zdaleka všechny FVE osazeny dálkově odečítaným průběhovým měřením, a tak se celková výroba všech fotovoltaických zdrojů v ČR počítá pomocí extrapolace výroby měřených zdrojů.

Proto bylo v posledné době nutno pracovat na metodách identifikace případného netypické chováni některé z měřených FVE, jako je například částečná odstávka elektrárny, aby mohla být tato informace zohledněna při výpočtu celkové výroby. Tento úkol je však ztížený faktem, že výroba FVE je určena intenzitou slunečního záření, oblačností a dalšími meteorologickými veličinami. Zejména sluneční záření a oblačnost přitom mají velkou volatilitu.

V příspěvku je představena jedna z možných metod detekce, založená na celkové denní výrobě. Dále je naznačen další směr výzkumu v této oblasti, který je zaměřen na detekci na základě funkcionálního vztahu mezi vyrobenou energií a celkovým slunečním zářením měřeným satelity nebo pozemními monitorovacími stanicemi.

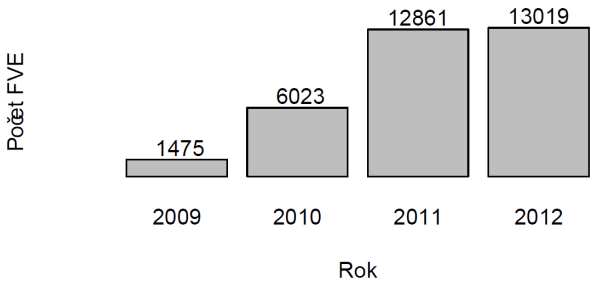
1. Úvod

V průběhu posledních let došlo v České republice a rovněž v téměř celé Evropě k velkému nárůstu počtu fotovoltaických elekráren. Tento nárůst měl dva hlavní důvody:

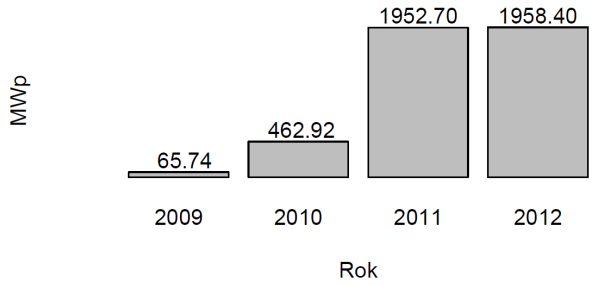
1. legislativní garance vysokých (dotovaných) výkupních cen elektřiny pocházející z obnovitelných zdrojů,

2. neustálé snižování cen fotovoltaických článků.

Zejména první důvod pak zapříčinil tzv. solární boom v letech 2009 a 2010, který je ilustrován na obrázcích 1 a 2. Tento boom se zastavil v roce 2011, kdy došlo k výraznému snížení výkupní ceny elektrické energie pocházející z obnovitelných zdrojů. Nicméně díky neustálému snižování ceny fotovoltaických panelů se objevuje stále více názorů, například v článku [2], že dojde k takzvané paritě s elektrickou sítí, tj. k vyrovnání (nedotované) ceny energie z FV systémů s cenou konvenčních energií. Při takovémto scénáři technologického a energetického vývoje lze samozřejmě očekávat další nárůst počtu a celkového instalovaného výkonu fotovoltaických zdrojů.



Obrázek 1: Celkový počet fotovoltaických elektráren s licencí na provoz v České republice k prvnímu dnu daného roku.



Obrázek 2: Součet instalovaných výkonů všech fotovoltaických zdrojů v České republice k prvnímu dnu daného roku.

Zvyšování podílu elektřiny vyrobené zejména fotovoltaickými zdroji má však řadu problémů, které vyplývají z vlastností těchto zdrojů. Fotovoltaické zdroje totiž tvoří neregulovaný systém a jejich výroba závisí především na obtížně predikovatelných meteorologických podmínkách. Okamžitý výkon jednotlivého zdroje je totiž určen především slunečním zářením a oblačností v dané lokalitě. Obě tyto veličiny pak v sobě obsahují značnou volatilitu, která činí predikci i odhad skutečné výroby jednotlivého zdroje ještě náročnější. Vysoký podíl energie z fotovoltaických zdrojů tak přináší zvýšené nároky na regulaci distribuční soustavy, která musí přenášet energii z vysoce volatilních zdrojů, jejichž výkon se v čase dramaticky mění. Dále přináší také vysoké nároky na řízení regulovaných zdrojů, tak aby byla udržována rovnováha mezi výrobou, spotřebou a ztrátami. V neposlední řadě jsou pak kladeny vysoké nároky na zajištění finančních toků, tak aby reflektovaly složitost systému obchodování s energií, který je určován a ovlivňován volným trhem s energií, povinností vykupovat obnovitelné zdroje energie, smlouvami o odběru a dodávkách a státními dotacemi na některé typy zdrojů.

Průběhově měřeny mohou být jen větší zdroje, zatímco velké množství malých zdrojů, umístěných typicky na střechách rodinných domů, průběhově měřeno není. Celkový výkon všech fotovoltaických zdrojů, potřebný pro vytváření bilancí, tak musí být odhadován jen z průběhově měřených zdrojů a z celkového instalovaného výkonu všech zdrojů. Proto je velmi důležité, aby byla data z průběhových měření kvalitní a aby odrážela typické chování všech FVE. Pro stanovení skutečné denní bilance je pak ještě potřeba detekovat případnou změnu chování jednotlivého zdroje ovlivňující množství vyrobené energie (například částečná či úplná odstávka elektrárny), aby mohla být tato informace zohledněna při výpočtu celkové bilance.

2. Popis vstupních dat

Pro účely této publikace byla k dispozici hodinová měření za rok 2011 ze 387 fotovoltaických elektráren, které představují necelých 40% celkového instalovaného výkonu v České republice.

Výroba jednotlivé farmy je popsána souborem veličin

$$\{Y_{idh}\}_{i,d,h}, \tag{1}$$

kde

i je jednoznačný identifikátor zdroje $(i = 1, \ldots, 387)$,

d označuje den měření $(d = 1, \ldots, 365)$ a

h označuje hodinu měření $(h = 0, \ldots, 23)$.

Zároveň pro každou farmu i máme k dispozici hodnotu instalovaného výkonu IV_i. Tato hodnota se v čase nemění.

3. Detekce založená pouze na datech o výrobě

Z provozního hlediska by bylo nejlepší, kdyby bylo možno detekovat nestandardně se chovající zdroje jen s pomocí dat z měřených FVE a do výpočtu by nevstupovala žádná jiná data (např. měření meteorologických veličin). Je zřejmé, že veličiny Y_{idh} a $Y_{id(h+1)}$, to jest hodnoty výroby jedné farmy ve dvou po sobě jdoucích hodinách, budou silně korelované. Z tohoto důvodu se jeví jako jeden z možných přístupů založení detekce na vhodné veličině reprezentující denní výrobu elektrárny.

3.1. Koeficient denní výroby

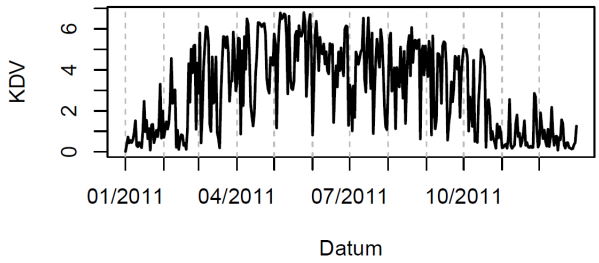
Při tomto přístupu je zdroj reprezentován pomocí koeficientu denní výroby, který se vypočte pro každou farmu $i = 1, \ldots, 387$ a každý den $d = 1, \ldots, 365$ pomocí vzorce

$$KDV_{id} = \sum_{h=0}^{23} \frac{Y_{idh}}{IV_i}, \tag{2}$$

kde Y_{idh} je množství energie vyrobené zdrojem i, ve dnu d v průběhu hodiny h a IV_i je hodnota instalovaného výkonu této FVE.

Koeficient denní výroby KDV tedy může nabývat hodnoty z intervalu $[0, 24]$, přičemž nulová hodnota znamená že daná elektrárna v daný den nevyrobila žádnou energii. Naopak pozorovaná hodnota koeficientu 24 a více znamená, že by výroba tohoto zdroje byla rovna instalovanému výkonu celých 24 hodin (včetně noci), popřípadě by mohla elektrárna vyrábět dlouhodobě více než je její instalovaný výkon, což v praxi není možné.

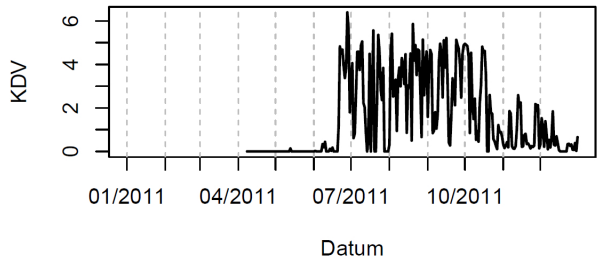
Obě tyto hodnoty tedy znamenají, že se daný zdroj chová výrazně nestandardně anebo že naměřená či přenesená data nejsou validní. U našich dat se hodnota KDV pohybovala v intervalu $(0, 8)$. Průběh KDV pro jednu vybranou, na první pohled standardní farmu, je zobrazen na obrázku 3.



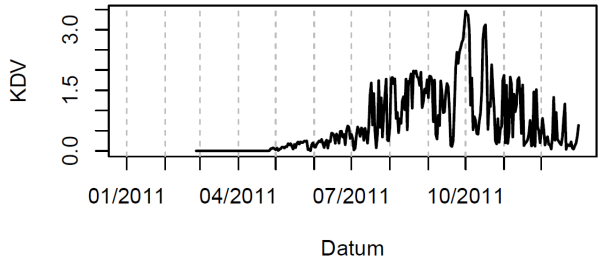
Obrázek 3: Typický průběh koeficientu denní výroby pro fotovoltaickou elektrárnu.

3.2. Chování netypické "na první pohled"

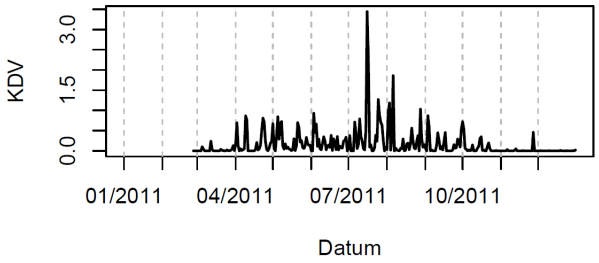
Při detailním zkoumání KDV jsme narazili na několik zjevných případů nestandardního chováni některých elektráren. Jejich příklady jsou uvedeny na obrázcích 4 až 8.



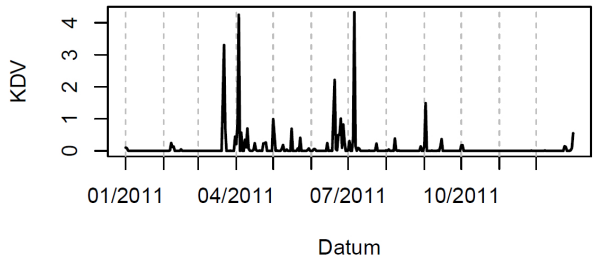
Obrázek 4: Nestandardní FVE – zdroj v prvních týdnech nedodává žádnou energii, přestože byl již veden jako zapojený.



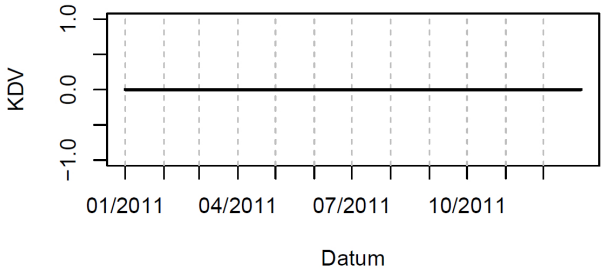
Obrázek 5: Nestandardní FVE – zdroj v prvních týdnech nedodává žádnou energii, pak několik týdnů výroba neodpovída rezervovanému výkonu, což se jeví jako postupný náběh elektrárny



Obrázek 6: Nestandardní FVE – velmi častá nulová výroba, zároveň nízké hodnoty KDV vedou k podezření z nesouladu mezi skutečným instalovaným výkonem a instalovaných výkonem uvedeným v databázi.



Obrázek 7: Nestandardní FVE – většinu roku nulová výroba.



Obrázek 8: Nestandardní FVE – nulová výroba počas celého roku.

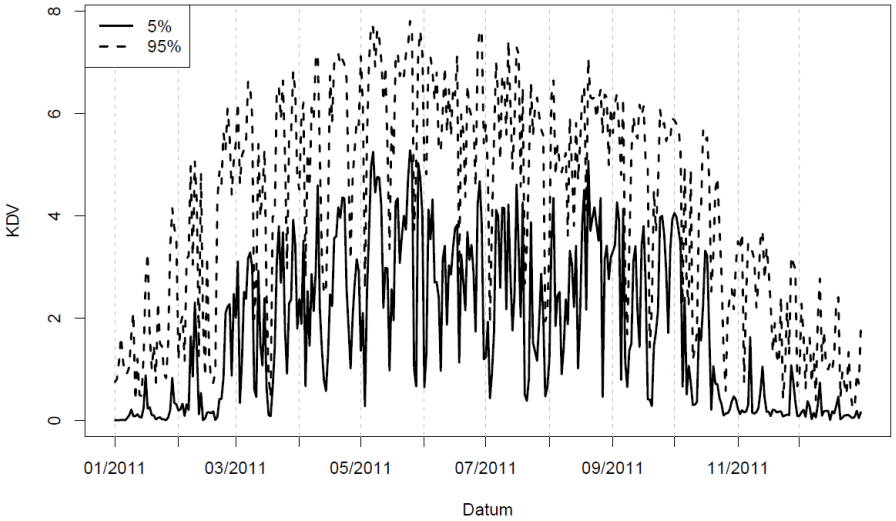
Jednoduché příklady nestandardního chování uvedené výše jsou detekovatelné heuristicky. Pro účely návrhu detekce nestandardního chování v méně jasných případech je třeba tyto případy vyloučit ze vstupních dat. Proto byla vstupní data filtrována a byly odstraněny pro jednotlivé farmy dny s celkovou nulovou výrobou. Tři FVE s extrémním chováním jako je zobrazeno obrázku 8, byly ze zkoumaného souboru úplně vyloučeny.

3.3. Detekce netypických zdrojů

Za ideálních podmínek, kdyby všechny zdroje byly technologicky identické a kdyby na území celé České republiky bylo úplně identické počasí, by hodnota KDV měla být stejná pro všechny zdroje. Rozdíly hodnot

KDV v jeden den pro různé zdroje jsou způsobeny především volatilitou počasí v ČR a různorodostí technických řešení jednotlivých FVE.

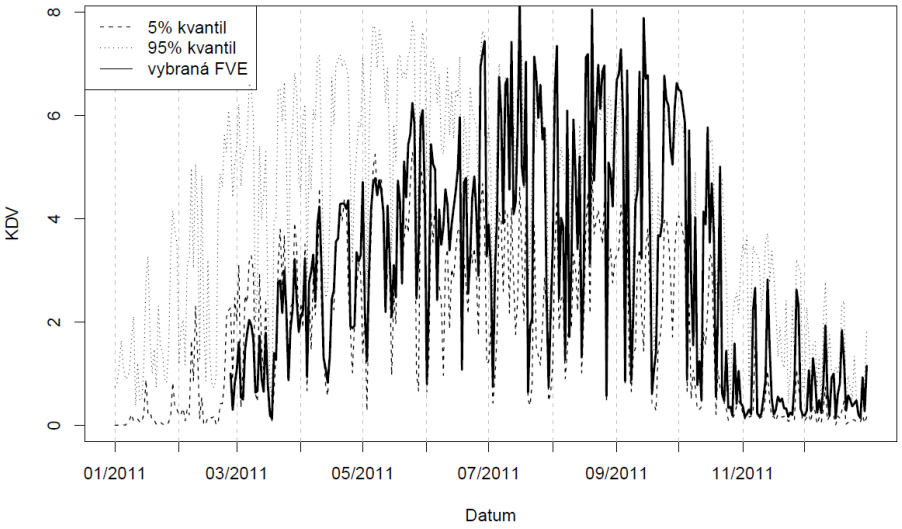
Pro každý den byly z dostupných hodnot KDV spočteny výběrové kvantily. Na základě expertního posouzení byly nakonec zvoleny 5% a 95% kvantily, tyto kvantily jsou znázorněny na obrázku 9. Následně byl pro každý zdroj stanoven počet případů, kdy se hodnota jeho KDV ocitla mimo interval tvořený těmito výběrovými kvantily.



Obrázek 9: Kvantily KDV.

3.4. Detekované zdroje

Použitím popsané metody bylo identifikováno 39 zdrojů, pro které byla hodnota KDV mimo interval ohraničený výběrovými kvantily více než 75 krát. U většiny zdrojů se jednalo buď o systematické překračování horního kvantilu nebo naopak o dlouhodobé nedosahování spodního kvantilu. Největší podezření v těchto případech padalo na nesprávnou hodnotu instalovaného výkonu. Nicméně našly se i zajímavější případy, jeden je uveden na obrázku 10. V tomto případě bylo zjištěno, z 309 hodnot KDV pro tento zdroj se jich 46 vyskytovalo nad hodnotou horního kvantilu a 47 pod hodnotou dolního kvantilu.



Obrázek 10: Jeden z identifikovaných netypických zdrojů spolu s odhadnutými kvantily. Všimněme si, že zatímco v první polovině roku FVE patřila ke zdrojům vyrábějícím velmi málo, v druhé polovině roku patřila naopak k nejefektivnějším zdrojům.

U zdrojů vytipovaných tímto postupem následně dochází k postupnému ověření správnosti informací o nich evidovaných. U některých FVE to vede k výraznému zpřesnění informací o jejích činnosti.

4. Detekce nestandardního chování s využitím závislosti výroby FVE na sluneční radiaci

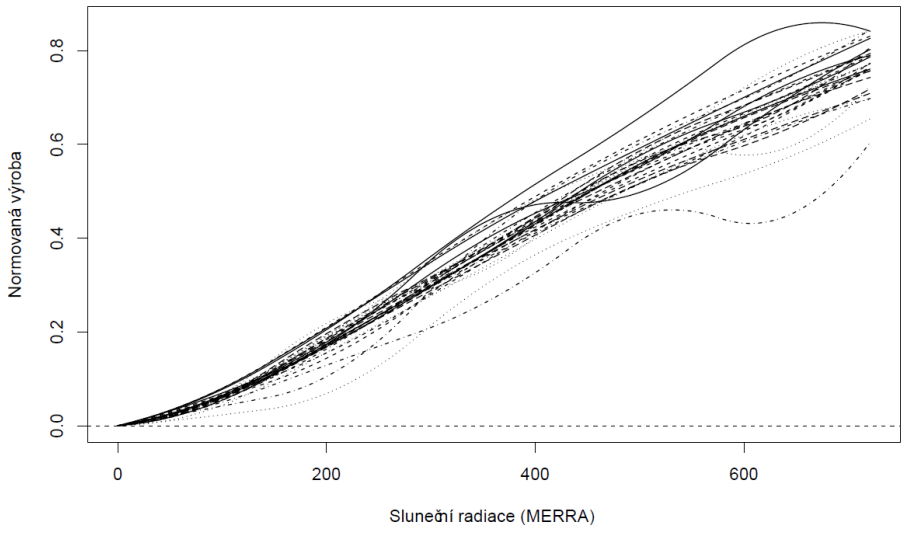
Jednou z nevýhod postupu popsaného v předchozí kapitole, to jest detekce založené pouze na porovnávaní celkové denní výroby, je fakt, že tento postup nijak nezohledňuje průběh výroby uvnitř dne. Proto byla v dalším kroku vyzkoušena možnost klasifikovat jednotlivé zdroje podle jejich vztahu mezi výrobou a celkovou sluneční radiací v dané oblasti. Problémem tohoto přístupu je neexistence měření slunečního záření přímo z místa kde se nachází daná elektrárna. Jedním z řešení tohoto problému je použít měření slunečního svitu z automatických měřícím stanic AIM Českého hydrometeorologického ústavu, další možností je pak použít hodnoty slunečného záření pocházející z meteorologických reanalýz. Protože měření pocházející z AIM stanic nejsou nijak validována a stanice jsou rozmístěny nerovnoměrně po území ČR, byla zvolena druhá možnost.

Tzv. analýza meteorologické situace je odhadem plošného rozložení meteorologických veličin na daném území. Tento odhad vychází jak z předchozího běhu numerického předpovědního modelu počasí, tak i z dostupných měření (pozorování na pozemních stanicích, sondáže ve vyšších vrstvách atmosféry, družicové snímky apod.). Měření jsou do modelového běhu asimilována nejnovějšími metodami, které se poměrně rychle vyvíjejí a mění. Termínem reanalýza se rozumí zejména provedení analýzy za delší časové období jednotnými metodami (na rozdíl od operativních analýz, kde se metody i modely mohou měnit). Pro účely této publikace byla zvolena reanalýza MERRA (Modern-Era Retrospective analysis for Research and Applications) vytvořená v americké NASA. Tato reanalýza je vytvářena na mřížce s velikostí čtverce $1/2 \times 2/3$ stupně.

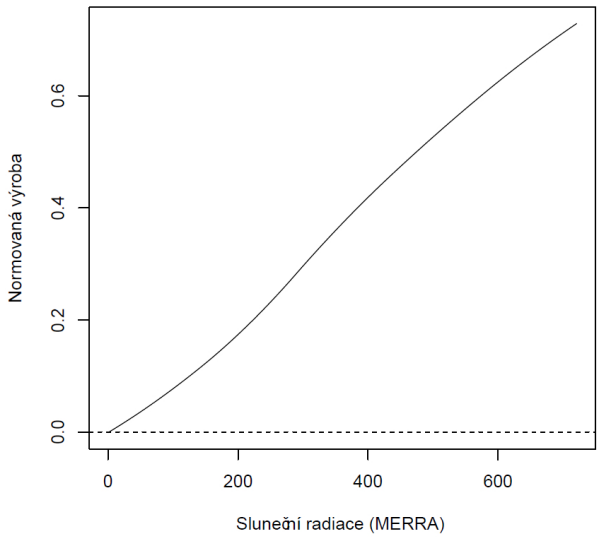
Vztah mezi výrobou i-té elektrárny a hodnotou sluneční radiace pro každý den d a každou farmu i je pak možno napsat ve tvaru

$$NV_{id} = f_{id}(SWD_{id}), \tag{3}$$

kde NV_{id} je normovaná výroba jednotlivé FVE (tj. podíl skutečné hodnoty výroby a instalovaného výkonu dané farmy), SWD_{id} je hodnota sluneční radiace v nejbližším bodě mřížky reanalýzy vzhledem ke skutečné poloze zdroje a funkce f_{id} vyjadřuje vztah mezi sluneční zářením a výrobou FVE. Tyto funkce jsou různé pro různé FVE a také se liší pro jednu FVE v různých dnech. V případě, že by hodnoty sluneční radiace z reanalýzy přesně odpovídaly skutečné radiaci přepočtené na úhel naklonění panelů FVE, měly by být funkce f_{id} téměř lineární (při zanedbání vlivu teploty panelu). Protože ve skutečnosti máme jen hodinová měření, je nutno funkce odhadnout, např. pomocí B-splinů a metody nejmenších čtverců. Detailně je tento postup popsán například v knize [3]. Odhady těchto funkcí, pozorovaných 3.9.2011, pro prvních 30 FVE jsou znázorněny na obrázku 11.

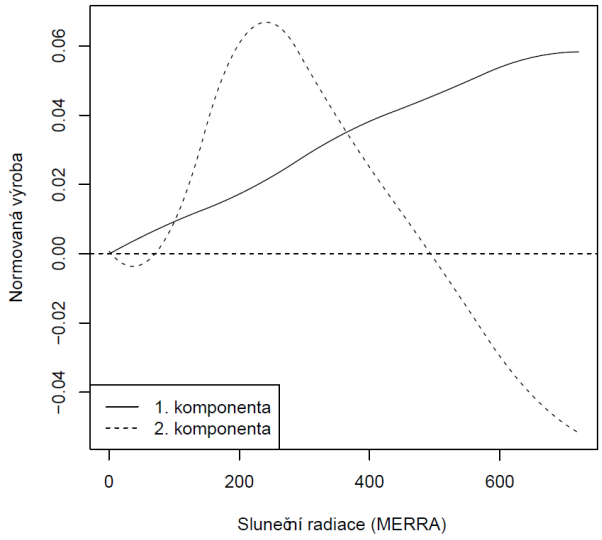


Obrázek 11: Funkce vyjadřující vztah mezi hodnotou sluneční radiace z reanalýz a výroby 30 FVE s největším instalovaným výkonem, dne 3.9.2011.



Obrázek 12: Střední hodnota normované výroby FVE.

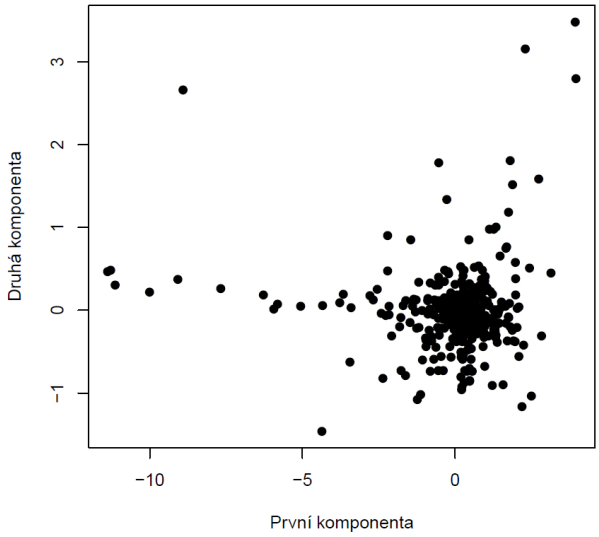
Již na první pohled je vidět, že zdaleka ne všechny FVE vykazují stejné chování vůči sluneční radiaci pocházející z reanalýz MERRA. Tyto rozdíly mohou být způsobeny rozdílnou technologií FVE, různými vzdálenostmi FVE od nejbližšího bodu mřížky, různým typem povrchu v místě instalace FVE apod.



Obrázek 13: První dvě komponenty vysvětlující více než 90% procent rozptylu.

Na funkce odhadnuté tímto způsobem byla aplikována metoda funkcionální analýzy hlavních komponent, podrobně popsaná v [3], která je založena na Karhunen–Loèveho rozkladu autokovariančního operátoru, viz například [1]. Pro každý den pak byly identifikovány dvě hlavní komponenty, které ve většině případů vysvětlovaly více než 90% rozptylu. Na obrázcích 12 a 13 jsou uvedeny střední hodnota a první dvě komponenty pozorované dne 3.9.2011.

Pro detekci nestandardně se chovajících FVE se v tomto případě použijí odhadnuté skóry, nakreslené na obrázku 14. Nicméně odvození pravidel pro detekci je podmíněno správnou interpretací jednotlivých komponent. Proto se v současné době autorova pozornost zaměřuje zejména na tyto dva úkoly. Pro jejich úspěšné vyřešení ale bude nutné provést ještě další experimenty a simulace.



Obrázek 14: Skóry jednotlivých FVE pro první dvě komponenty.

Literatura

[1] R.B. Ash and M.F. Gardner, " Topics in Stochastics Processes", Academic Press, INC., New York, 1975.

[2] K. Branker, M.J.M. Pathak, and J.M. Pearce, "A Review of Solar Photovoltaic Levelized Cost of Electricity", Renewable & Sustainable Energy Reviews, vol. 15, pp. 4470–4482, 2011.

[3] J.O. Ramsay and B.W. Silverman, "Functional Data Analysis", Springer-Verlag, 2005.

Engineering Distributed Adaptive Systems Using Components

Post-Graduate Student:
MGR. JAROSLAV KEZNIKL
Institute of Computer Science of the ASCR, v. v. i.
Pod Vodárenskou věží 2
182 07 Prague 8, CZ
keznikl@cs.cas.cz

Supervisor:
RNDR. TOMÁŠ BUREŠ, PH.D.
Institute of Computer Science of the ASCR, v. v. i.
Pod Vodárenskou věží 2
182 07 Prague 8, CZ
bures@cs.cas.cz

Field of Study:
Software Systems

The work was partially supported by the EU project ASCENS 257414, the Grant Agency of the Czech Republic project P202/11/0312. The work was partially supported by Charles University institutional funding SVV-2012-265312.

Abstract

One of the major issues in the domain of dynamically evolving distributed systems composed of autonomous and (self-) adaptive components is the task of systematically addressing the design complexity of their communication and composition. This is caused mainly by the inherent dynamism of such systems, where components may appear and disappear without anticipation. Addressing this issue, we employ separation of concerns by introducing a mechanism of implicit communication over implicit bindings, enabling components to dynamically form implicitly interacting groups – ensembles. Specifically, we present the DEECo component model, which based on this mechanism.

1. Introduction

Traditional software engineering methodologies together with related programming paradigms have long been guiding the procedure of building software systems through the requirements and design phase to testing and deployment. In particular, engineering paradigms based on the notion of components [1] have gained a lot of popularity as they support separation of concerns – extremely valuable when dealing with systems of high complexity.

It seems, though, that these traditional methodologies and paradigms are not sufficient when exploited in the domain of continuously changing, massively distributed and dynamic systems, such as the ones we explore in the ASCENS project [2]. These systems need to adjust to changes in their architecture and environment seamlessly or, even better, acknowledge the absence of absolute certainty over their (constantly changing) architecture and environment. An appealing research direction seems to be the decomposition of such systems into components able to operate upon temporary and volatile information in an autonomous [3] and self-adaptive fashion [4]. From the software engineering perspective, two main challenges arise:

- What are the correct *low-level abstractions* (models, resp. paradigms) that will allow for separation of concerns?
- How can we devise a systematic approach for *designing* such systems, exploiting the above abstractions?

In response, we propose the *DEECo component model* (stands for Dependable Emergent Ensembles of Components) [5]. The goal of the component model is to allow for designing systems consisting of autonomous, self-aware, and adaptable components, which are implicitly organized in groups called *ensembles*. To this end, we propose a slightly different way of perceiving a component; i.e., as a self-aware unit of computation, relying solely on its local data that are subject to external modification during the execution time. The whole communication process relies on automatic data exchange among components, entirely externalized and automated within the DEECo runtime framework. This way, the components have to be programmed as autonomous units, without relying on whether/how the distributed communication is performed, which makes them very robust and suitable for rapidly-changing environments.

The rest of the paper is organized as follows. In Section 2 the main concepts of the DEECo component model are presented. Section 3 evaluates the presented concepts by giving an example based on the ASCENS

cloud case study. Section 4 discusses the related work, while Section 5 concludes the paper and presents future work ideas.

2. DEECo Component Model

DEECo is based on two concepts: *component* and *ensemble*. Stemming from the ASCENS project, these concepts closely reflect fundamentals of the SCEL specification language [6] and are in detail elaborated in the rest of this section.

2.1. Component

A component is an autonomous unit of deployment and computation. Similar to SCEL, it consists of:

- Knowledge
- Processes

Knowledge contains all the data and functions of the component. It is a hierarchical data structure mapping identifiers to (potentially structured) values. Values are either statically typed data or functions. Thus DEECo employs statically-typed data and functions as first-class entities. We assume pure functions without side effects.

Processes, each of them being essentially a "thread", operate upon the knowledge of the component. A process employs a function from the knowledge of the component to perform its task. As any function is assumed to have no side effects, a process defines mapping of the knowledge to the actual parameters of the employed function (*input knowledge*), as well as mapping of the return value back to the knowledge (*output knowledge*). A process can be either periodic or triggered. A process can be triggered when its input knowledge changes or when a given condition on the component's knowledge (*guard*) is satisfied.

2.2. Component Composition

In DEECo, component composition is captured by means of ensembles. Composition is flat, expressed implicitly via a dynamic involvement in an ensemble. An ensemble consists of multiple member components and a single coordinator component. The only allowed form of communication among components is communication between a member and the coordinator in an ensemble. This allows the coordinator to apply various communication policies.

Thus, an ensemble is described pair-wise, defining the couples coordinator – member. An ensemble definition consists of:

- Required interface of the coordinator and a member
- Membership function
- Mapping function

Interface is a structural prescription for a view on a part of the component's knowledge. An interface is associated with a component's knowledge by means of *duck typing*; i.e., if the component's knowledge has the structure prescribed by the interface, then the component reifies the interface. In other words, an interface represents a partial view on a component's knowledge.

Membership function declaratively expresses the condition, under which two components represent the pair coordinator-member of an ensemble. The condition is defined upon the knowledge of the components. In the situation where a component satisfies the membership functions of multiple ensembles, we envision a mechanism for deciding whether all or only a subset of the candidate ensembles should be applied. Currently, we employ a simple mechanism of a partial order over the ensembles for this purpose (the "maximal" ensemble of the comparable ones is selected, the ensembles which are incomparable are applied simultaneously).

Mapping function expresses the implicit distributed knowledge exchange between the coordinator and a member of an ensemble. It ensures that the relevant changes in knowledge of one component get propagated to the other component. However, it is up to the DEECo runtime framework when/how often the mapping function is invoked. We assume a separate mapping for each of the directions coordinator-member, member-coordinator.

The important idea is that the components do not perceive the existence of ensembles (including their membership in an ensemble). They operate only upon their own local knowledge, which might get implicitly updated by the DEECo runtime framework whenever the component is part of an ensemble.

2.3. Execution Model

The DEECo execution model is based on asynchronous knowledge exchange and process execution, stemming from the asynchronous nature of the target dynamic distributed systems. Specifically, the component processes execute in parallel as independent threads either periodically, when triggered by modification of (a part of) their input knowledge, or whenever the process guard is satisfied. Similarly, a component binding of component forming an ensemble is accomplished by a separate

activity, evaluating the mapping function (again either periodically or when triggered).

Due to the asynchrony, it is necessary to ensure that knowledge is accessed consistently. Thus, at its start, a process is atomically provided with a copy of its input knowledge so that its computation is not affected by later-occurring knowledge modifications. When finishing, the process atomically updates its output knowledge. The same atomic copy-on-start and update-on-return semantics also applies to the membership and mapping functions of ensembles. Technically, this semantics can be implemented for instance via messaging.

Consequently, based on the computational model, an ensemble is created when the ensemble condition starts to hold, and is discarded when the condition gets violated. Technically, as the whole system is asynchronous and potentially distributed, techniques for handling inherent delays, while creating/discarding ensembles, have to be carefully chosen.

3. Evaluation

To evaluate and illustrate the above-described concepts, we'll give an example from the Science Cloud case-study [12]. In this scenario, several interconnected heterogeneous network nodes (execution nodes, storage nodes) run a cloud platform, on which 3rd-party services are being executed. Moreover, the nodes can dynamically enter/leave the network. Provided an external mechanism for migrating a service from one (execution) node to another, the goal is to "cooperatively distribute the load of the overloaded (execution) nodes in the network".

3.1. Solution in a Nutshell

Before describing the solution in DEECo concepts, we will give an outline of the final result. Basically, for the purpose of this evaluation, we consider a simple solution, where each of the nodes tracks its own load and if the load is higher than a fixed threshold, it selects a set of services to be migrated out. Consequently, all the nodes with low-enough load (determined by another fixed threshold) are given information about the services selected for migration, pick some of them and migrate them in using the external migration mechanism.

The challenge here is to decide, which of the nodes the service information should be given to and when, since the nodes join and leave the network dynamically. In DEECo, this is solved by describing such a node interaction declaratively, so that it can be carried out in an automated way by the runtime framework when appropriate.

3.2. Realization in DEECo

Specifically, we first identify the components in the system and their internal knowledge. In this example, the components will be all the different nodes (execution/storage nodes) running the cloud platform (Figure 1). The inherent knowledge of execution nodes is their current `load`, information about running services (`serviceInfo`), etc. We expect an execution node component to have a process, which determines the services to be migrated in case of overload. Similarly, the inherent knowledge of the storage nodes is their current `capacity`, `filesystem`, etc.

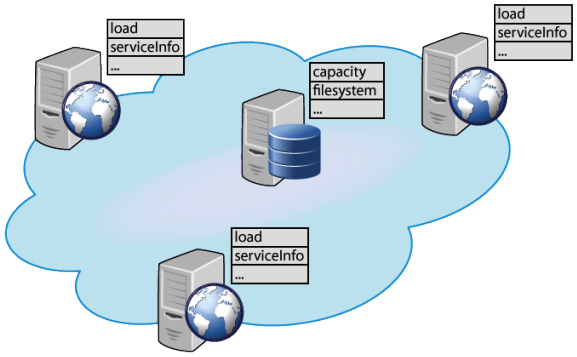


Figure 1: Components representing the cloud nodes and their inherent knowledge.

The second step is to define the actual component interaction and exchange of their knowledge. In this example, only the transfer of the information about services to be migrated from the overloaded nodes to the idle nodes is to be defined. The interaction is captured in a form of an ensemble definition (Figure 2), thus representing

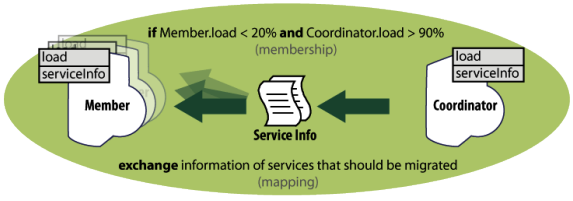


Figure 2: Definition of an ensemble that ensures exchange of the services to be migrated.

a "template" for interaction. Here, the coordinator, as well as the members, has to be an execution node providing the `load` and `serviceInfo` knowledge entries.

Having such nodes, whenever the (potential) coordinator has the load above 90% and a (potential) member has the load below 20% (i.e., the membership function returns true), the ensemble is established and its mapping function is executed (possibly in a periodic manner). The mapping function in this case ensures exchanging the information about the services to be migrated from the coordinator to the members of the ensemble.

When applied to the current state of the components in the system, an ensemble – established according to the above-described definition – ensures an exchange of the service-to-be-migrated information among exactly the pairs of components meeting the membership condition of the ensemble (Figure 3).

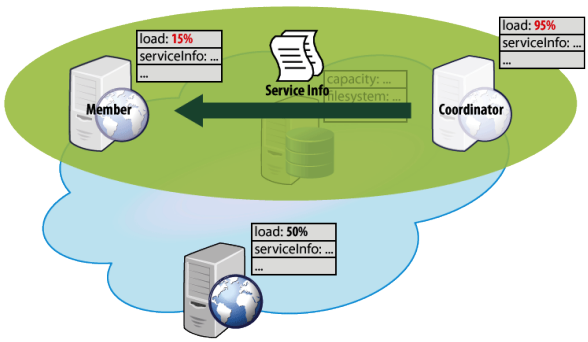


Figure 3: Application of the ensemble definition from Figure 2 to the components from Figure 1.

According to the exchange of service information, the member nodes then individually perform service migration via the external (i.e., outside of DEECo) migration mechanism.

3.3. Runtime Framework Implementation

Currently, we work on a prototype of the DEECo runtime framework implementation, based on distributed tuple spaces and implemented in Java. The sources, as well as documentation and examples, can be found at https://github.com/d3scomp/JDEECo.

4. Related Work

The task of achieving autonomy and (self-) adaptation has been partially addressed by agent-based approaches [7, 8], where actors leveraging on messaging establish explicit bindings for data and code exchange. Jade [7] is a complete framework for building, running and managing distributed multi-agent systems. Similarly, X-Klaim [8] is a complete framework based on a domain-specific language for capturing agent-based systems, where both data and processes are subjects to mobility. In general, agent-based frameworks themselves do not provide any higher abstractions for implicit grouping of components; however, due to their relative maturity, such frameworks could represent a suitable middleware for implementing the knowledge exchange in the DEECo runtime framework (as a replacement of distributed tuple spaces).

As for coping with dynamism of component bindings, techniques utilizing implicit bindings while focusing on explicit communication have been proposed. In iPojo [9] – a component system built upon the Felix OSGi implementation – each component binding is determined by a declarative specification associated with component interfaces. In [10], the idea of agent self-organization based on declarative conditions is presented. Specifically, the agents organize themselves into groups according to their spatial distribution and reorganization rules, communicating explicitly via a shared tuple space.

Finally, separation of concerns was to some extent achieved by introducing implicit communication (driven by a third-party entity) [11]. However, the communication is usually carried-out via explicit bindings.

5. Conclusion and Future Work

We assume that DEECo will be employed in the design of systems of autonomous self-adaptive components, such as a self-managing cloud platform and self-organizing car sharing [12], where it aims at simplifying the design process. Specifically, we expect DEECo to effectively handle knowledge exchange among the components, emphasizing separation of concerns. Although similar to software connectors [13], DEECo ensembles capture component composition implicitly and thus allow for handling of dynamic changes in an automated way. Similar benefits result from the implicit knowledge exchange.

We envision that the component model outlined here will serve as the basis for a design methodology that will exploit the presented abstractions and help in building long-lasting systems of autonomous components and component ensembles. Further, in order to support controlled architecture evolution, we aim to incorporate mechanisms for dynamic addition, modification, and removal of ensemble prescriptions. In addition, we envision supporting formal verification of DEECo applications. As for model checking of temporal properties, we assume a mapping of applications to SCEL [6] and intend to exploit its means [14] for this purpose. Moreover, we anticipate also employing stochastic model

checking [15, 16] for quantitative verification. Finally, inspired by the cloud and e-mobility case studies, we intend to introduce, in addition to abstractions for performance awareness, other forms of implicit knowledge-based communication such as distributed consensus.

References

[1] C. Szyperski, "Component Software: Beyond Object-Oriented Programming" (2nd Edition) (Hardcover), Addison-Wesley Professional, 2002.

[2] ASCENS [Online], http://www.ascens-ist.eu.

[3] J. O. Kephart, and D. M. Chess, "The vision of autonomic computing", *Computer*, vol. 36, IEEE CS, 2003, pp. 41–50.

[4] R. N. Taylor, N. Medvidovic, and P. Oreizy, "Architectural styles for runtime software adaptation", *Joint Working IEEE/IFIP Conference on Software Architecture & European Conference on Software Architecture (WICSA/ECSA 2009)*, 2009, pp. 171–180.

[5] J. Keznikl, T. Bures, F. Plasil, and M. Kit, "Towards Dependable Emergent Ensembles of Components: The DEECo Component Model", *Joint Working IEEE/IFIP Conference on Software Architecture & European Conference on Software Architecture (WICSA/ECSA 2012)*, Aug, 2012.

[6] R. De Nicola, G. Ferrari, M. Loreti , and R.Pugliese, "Languages primitives for coordination, resource negotiation, and task description", ASCENS Deliv. D1.1, 2011, .

[7] F. Bellifemine, G. Caire, and D. Greenwood, "Developing multi-agent systems with Jade", John Wiley & Sons, 2007.

[8] E. Gjondrekaj, M. Loreti, R. Pugliese, and F. Tiezzi, "Modeling adaptation with a tuple-based coordination language", Proc. of *27th Symposium on Applied Computing (SAC 2012)*, 2012.

[9] C. Escoffier and R. S. Hall, "Dynamically adaptable applications with iPOJO service", *Software Composition*, 2007.

[10] C. Villalba, M. Mamei, and F. Zambonelli, "A self-organizing architecture for pervasive ecosystems", *Self-Organizing Architectures*, volume 6090 of LNCS, pp. 275–300, 2010.

[11] A. Basu, M. Bozga, and J. Sifakis, "Modeling heterogeneous real-time components in BIP", Proc. of *Fourth IEEE International Conference on Software Engineering and Formal Methods (SEFM'06)*, 2006, pp. 3-12.

[12] N Serbedzija, S. Reiter, M. Ahrens, J. Velasco, C. Pinciroli, N. Hoch, and B. Werther, "Requirement specification and scenario description", ASCENS Deliv. D7.1, November 2011.

[13] R.N. Taylor, N. Medvidovic, and E.M. Dashofy: "Software architecture: foundations, theory, and practice", Wiley, 2010.

[14] L. Bettini et al., In global computing. Programming Environments, Languages, Security, and Analysis of Systems, volume 2874 of LNCS, 2003, pp. 88–150. "The Klaim project: theory and practice", *Global Computing: Programming Environments, Languages, Security, and Analysis of Systems*, volume 2874 of LNCS, 2003, pp. 88–150.

[15] M. Z. Kwiatkowska, G. Norman, D. Parker, and H. Qu, "Assume-guarantee verification for probabilistic systems", Proc. of *Tools and Algorithms for Construction and Analysis of Systems (TACAS 2010)*, Springer, 2010, pp. 23–37.

[16] J. Barnat, L. Brim, I. Cerna, M. Ceska, and J. Tumova: "ProbDiVinE, a parallel qualitative LTL model checker", *Quantitative Evaluation of Systems (QEST 07)*, IEEE, 2007.

Optimalizace osazování odběrných míst inteligentními plynoměry

doktorand:
MGR. ONDŘEJ KONÁR
Ústav informatiky AV ČR, v. v. i.
Pod Vodárenskou věží 2
182 07 Praha 8
konar@cs.cas.cz

školitel:
DOC. ING. EMIL PELIKÁN, CSC.
Ústav informatiky AV ČR, v. v. i.
Pod Vodárenskou věží 2
182 07 Praha 8
pelikan@cs.cas.cz

obor studia:
Inženýrská informatika v dopravě a spojích

Abstrakt

Celosvětovým trendem v oblasti měření spotřeby plynu je postupné osazování odběrných míst tzv. inteligentními měřidly. Tyto přístroje jednak měří ve vysokém časovém rozlišení a jednak umožňují on-line přenos naměřených dat ke zpracování v informačním systému distributora nebo obchodníka s plynem. Ačkoli cena těchto přístrojů postupně klesá, vzhledem k velmi vysokému počtu odběrných míst (např. v ČR je jich přes milion) je nutné osazování provádět postupně v průběhu několika let.

Ideální je rozmisťovat měřidla tak, aby byla naměřená data využita s maximální efektivitou. Svou roli však hrají také technicko-ekonomická omezení dané vlastním procesem osazování (například je výhodné, aby byly osazované přístroje v geografické blízkosti).

V tomto článku je představena metodika výběru vhodných odběrných míst k osazení inteligentním měřením. Metodika je založena na statistickém zpracování fakturačních dat odběrných míst ze zákaznického kmene distribuční společnosti RWE GasNet, s.r.o. Navržená metodika bude v uvedené distribuční společnosti v následujícím roce provozně testována.

Úvod

Měření odběru je pro distribuci a obchod s energiemi klíčovou záležitostí. Na základě výsledků měření se fakturuje odebraná energie, predikuje odebraná energie v budoucnu, určuje cena atd. Distribuční síť, ať již plynárenská či elektroenergetická, je osazena v různých bodech měřidly s různou přesností a různým časovým rozlišením. Obecně platí, že čím větší množství plynu (nebo elektrického proudu) daným místem proteče, tím větší časové rozlišení se používá při archivaci naměřených hodnot. V uzlových bodech distribuční sítě proto bývá zpravidla velmi podrobné měření (s hodinovou nebo denní frekvencí), zatímco u koncových zákazníků bývá měření méně podrobné. Vyhláška [2] o pravidlech trhu s plynem uvádí tři typy měření odběru:

měření typu A – průběhové měření s dálkovým přenosem dat,

měření typu B – průběhové měření bez dálkového přenosu dat,

měření typu C – kumulativní měření.

Zatímco v prvních dvou případech jsou naměřené hodnoty ukládány v pravidelných časových intervalech (typicky hodina nebo den), v případě měření typu C je k dispozici pouze aktuální stav spotřebované energie. Při tom však chybí dálkový přenos údajů a tudíž je spotřeba známa pouze za delší časové období. V ČR je toto období typicky jeden měsíc pro firemní zákazníky s vyšší spotřebou a jeden rok až 18 měsíců pro domácnosti a malé firemní zákazníky.

Je celkem pochopitelné, že většina zákazníků je vzhledem k malému odběru osazena měřením typu C. Tato skutečnost je však příčinou určitých provozních problémů. V určitých situacích je totiž třeba znát údaje s vyšším časovým rozlišením, než je k dispozici. Jedná se například o tyto situace:

- Změna ceny plynu – v takovém případě je třeba znát k datu změny spotřebu všech zákazníků, aby jim bylo možné fakturovat spotřebu dle platného ceníku.
- Určování hodnoty akcií společnosti – zde je nutné znát přibližně hodnotu majetku distributora či obchodníka s energií, do celkové bilance však chybí

hodnota tzv. nevyfakturované energie, tedy energie, která již byla spotřebována, ale nebyla ještě fakturována.

- Zúčtování odchylek – v každé distribuční síti nutně dochází ke ztrátám. Vzhledem k tomu, že ani u elektrické energie, ani u plynu není možné objektivně určit, kterému obchodníkovi ztráta v daném dni vznikla, jsou ztráty rozpočítávány mezi jednotlivé obchodníky přímo úměrně množství energie, kterou prodali. Toto množství však není z důvodů neprůběhového měření známo.

Existují v zásadě dva hlavní přístupy k řešení výše uvedeného problému:

- osazení všech odběrných míst měřením typu A,
- odhad spotřeby pomocí náhradních metod.

Ačkoli v ČR i ve světě stále z ekonomických důvodů převládá zejména druhý přístup (na vývoji modelů pro odhad spotřeby plynu v ČR se významně podílí i řešitelský tým z ÚI [1, 4]), je postupné osazování zákazníků průběhovým měřením v dlouhodobém plánu mnoha distribučních společností.

Plán počítá s osazováním tzv. inteligentními měřidly (smart meters [5]). Konvenční průběhové měření je realizováno pomocí přepočítávače s datovým úložištěm a případným dálkovým přenosem (v případě měření typu A), který je připojen k neprůběhovému měřidlu (elektroměr, plynoměr). Naproti tomu inteligentní měřidlo je měřící zařízení, které přímo umožňuje ukládání dat s vysokým časovým rozlišením – jedná se prakticky o měření v reálném čase – a jejich odesílání (s maximálně denním intervalem) provozovateli distribuční sítě. Kromě toho umožňují inteligentní měřidla například upozornění na výpadky, monitoring kvality dodávané energie či obousměrnou komunikaci (lze tedy navázat i spojení od provozovatele sítě k měřidlu). Paradoxně (možná právě kvůli zamýšlenému masovému nasazení) jsou inteligentní měřidla levnějším řešením než konvenční průběhové měření, které se již stává morálně zastaralým.

I přes relativně nízkou cenu je však nemožné osadit všechna odběrná místa najednou. Odběrných míst bez průběhového měření je totiž velké množství, v ČR je například více než milion takových zákazníků. I při ceně jednoho měřidla v rámci tisíců korun se jedná o vysoké náklady a tudíž je nutné osazování rozložit do více let. Orientačně se počítá s minimálně desetiletým obdobím osazování. Aby bylo osazování co nejefektivnější, je nutno jej provádět nikoli nahodile, ale podle předem připravené metodiky. V následujících odstavců bude problematika osazování měřidel rozebrána z několika úhlů pohledu. Dále bude prezentován návrh metodiky osazování, který vznikl v rámci spolupráce mezi ÚI a distribuční společností RWE GasNet, s.r.o. Tento návrh zohledňuje všechny níže uvedené úhly pohledu. V následujícím roce se plánuje experimentální ověření efektivity této metodiky ze strany RWE.

1. Možnosti využití dat z inteligentního měření

Inteligentní měřidla poskytují data o spotřebě téměř v reálném čase. Takováto data jsou velmi cenná a mají více možností využití. V některých případech je ke způsobu využití naměřených dat vhodné přihlédnout při návrhu odběrných míst k postupnému osazování měřidly. Vybrané možnosti využití jsou popsány v následujících odstavcích.

S ohledem na všechny níže uvedené aspekty byla vytvořena metodika identifikace obcí vhodných pro osazení inteligentním měřením tak, aby v dané obci byli vysoce zastoupeni problematičtí zákazníci. Definice problematického zákazníka bude diskutována níže v odstavci 2, kde jsou zároveň diskutovány různé varianty navržené metodiky. Vlastní použití pak závisí na konkrétní situaci a na prioritách zadavatele.

1.1. Zpřesnění odhadu spotřeby

Odhad spotřeby plynu v situacích diskutovaných v úvodu tohoto příspěvku probíhá typicky pomocí různě kvalitních matematických modelů. Jedním z nejpoužívanějších modelů v rámci ČR je tzv. model TDD [1] (zkratka TDD odpovídá pojmu “typové diagramy dodávky”), vyvinutý ve spolupráci ÚI a RWE Plynoprojektu pro účely zúčtování odchylek. Jeho použití přitom nemusí být ve všech případech vhodné vzhledem k tomu, že je model optimalizován pro co nejpřesnější odhad celkové spotřeby větších zákaznických skupin. Pro optimalizaci modelu k odhadu individuální spotřeby, případně pro konstrukci zcela nového modelu šitého na míru dané problematice, je zapotřebí značného počtu průběhově naměřených údajů o spotřebě. Data z inteligentních měřidel jsou přitom nejsnadněji získatelnými údaji s dostatečným časovým rozlišením.

Za tímto účelem je vhodné mít k dispozici reprezentativní náhodný vzorek odběratelů. Vzhledem k tomu, že pro optimální klasifikaci zákazníků, na jejímž základě by se prováděl výběr do vzorku, není dostatek informací (dostupná fakturační data neposkytují informaci

v dostatečném časovém rozlišení a data z průběhových měření v rámci projektu TDD pravděpodobně nejsou zcela reprezentativní), bylo by z pohledu zpřesnění odhadu spotřeby vhodné část měření v každém kole instalace rozmístit zcela náhodně.

1.2. Zpřesnění informace u obtížně modelovatelných odběrných míst

Doposud používané modely pro odhad spotřeby jsou založeny na statistickém přístupu. Kriteriální funkce pro optimalizaci jsou tudíž založeny na statistice počítané přes zvolené zákaznické třídy. Z principu tedy tyto metody nemohou dobře fungovat pro odhad individuálních spotřeb netypicky se chovajících zákazníků. Osazení problematických zákazníků inteligentním měřením proto může v důsledku zlepšit přesnost informace o celkovém odběru za předpokladu, že měření budou (relativně) přesná a přesnost odhadu se na neměřeném kmeni zvýší v důsledku větší stability.

1.3. Využití měření pro odhad lokálních ztrát

Inteligentní měření (za předpokladu, že jsou úplná a kvalitní) mohou poskytnout cennou informaci o průběhu ztrát v uzavřené lokalitě. Tato informace je doposud neznámá a ztráty jsou zpravidla odhadovány pevným podílem celkového nátoku do lokality. Podmínkou využití inteligentních měření pro měření ztrát je však osazení všech odběrných míst v dané lokalitě. Předpokládá se totiž, že všechny další vstupy a výstupy již průběhově měřené jsou a tak lze ztráty vypočíst prostým odečtením příslušných naměřených hodnot.

1.4. Jednoduchost osazování

Pro zvýšení efektivity při fyzickém osazování je vhodné, aby osazovaná odběrná místa byla geograficky blízko sebe. Geografická blízkost osazovaných odběrných míst minimalizuje personální náklady a náklady na dopravu. Tento požadavek je v souladu s potřebou osazení celé uzavřené lokality za účelem měření ztrát.

2. Identifikace lokalit vhodných pro osazení inteligentním měřením

Uvažujeme-li všechny možnosti využití naměřených údajů uvedené v odstavci 1 a chceme-li ke všem přihlédnout při plánování postupného osazování odběrných míst inteligentními měřidly, jeví se jako nejvhodnější následující postup:

- část měření umístit zcela náhodně,
- částí měření pokrýt celé uzavřené lokality tak, aby byly vybrány lokality s největším zastoupením obtížně modelovatelných zákazníků.

Vzhledem k tomu, že první bod je z pohledu metodiky poměrně jednoduchý, věnujeme se v následujícím textu výhradně druhé části, tj. identifikaci lokalit vhodných pro osazení. Informace o tom, ke které uzavřené lokalitě odběrné místo přísluší, jsou bohužel nedostupné v dostatečné kvalitě. Za jednotku pro identifikaci byla proto zvolena obec. Hlavním hlediskem je pochopitelně využití měření ke zpřesnění informace u obtížně modelovatelných zákazníků. Vedlejším efektem je však splnění dalších kritérií, jako je geografická blízkost osazovaných odběrných míst a v některých případech také uzavřenost dané lokality (v případě, že se jedná o lokalitu pokrývající jednu obec).

Hlavní myšlenka se dá v bodech popsat takto:

- Nalezneme kritérium kvantifikující problémovost zákazníka. Typicky takovým kritériem bude nějaká číselná charakteristika založená na fakturační historii a nějaká kritická mez.
- Pro všechny obce napočítáme zastoupení (nebo jinou charakteristiku) problematických zákazníků.
- Zvolíme obce s nejhoršími hodnotami výše uvedených charakteristik a ty pak kompletně osadíme inteligentním měřením.

Princip metodiky je formulován relativně obecným způsobem. Vhodnou volbou parametrů lze přiblížit výsledek požadovanému cíli. Jednotlivé volitelné parametry jsou podrobněji rozebrány v následujících odstavcích.

2.1. Kritérium problematického zákazníka

Kritérium problematického zákazníka by mělo být založeno na jeho fakturačních datech. Pro návrh metodiky byly k dispozici fakturační údaje všech zákazníků distribuční společnosti RWE GasNet. Pro každého zákazníka jsou dostupné údaje z jednoho nebo více fakturačních období. V každém období máme kromě naměřené spotřeby také určité detailní informace o odběrném místě – údaje o obci, v níž se odběrné místo nachází, typ a způsob odečtu (řádný/mimořádný, resp. jakým způsobem byl odečet proveden) a dále údaje o spotřebičích, které zákazník využívá. Pro analýzu byla využita pouze data zákazníků, kteří měli alespoň 4 platné odečty. Jedná se přibližně o 80% zákazníků.

Pro návrh kritérií jsme uvažovali tyto základní veličiny:

Plánovaná roční spotřeba (PRS) – hodnota založená na historických spotřebách (tj. spotřebách za období až 3 roky před daným fakturačním obdobím), výpočet probíhá v souladu s vyhláškou [2] podle vzorce

$$PRS = \frac{\sum_{\tau\in\Omega} S_\tau}{\sum_{\tau\in\Omega}\sum_{d\in\tau} TDD_d}, \qquad (1)$$

kde

PRS značí plánovanou roční spotřebu daného zákazníka,

Ω je sjednocení jednoho nebo více historických fakturačních období pokrývající 3 roky zpětně od data výpočtu PRS,

S_τ je fakturovaná energie za období τ,

TDD_d je tzv. "přepočtený typový diagram dodávky"pro den d definovaný vyhláškou [2] a zveřejněný na webové stránce operátora trhu s energiemi OTE, a.s. [3]. Hodnoty typových diagramů jsou vypočteny pomocí modelu TDD [1], pro účely analýzy byly typové diagramy napočítány znovu, aby byla zajištěna jednotná verze modelu TDD po celé testované období.

Normalizovaná aktuální spotřeba (IPRS) – vypočte se tak, že se aktuální fakturovaná spotřeba v kWh vydělí součtem přepočtených TDD za dané fakturační období, tj. podle vzorce

$$PRS = \frac{S_\tau}{\sum_{d\in\tau} TDD_d}, \qquad (2)$$

kde τ je tentokrát poslední známé fakturační období. IPRS v postatě odpovídá "ideální"hodnotě plánované roční spotřeby, tedy tomu, co je odhadováno plánovanou roční spotřebou. Z toho důvodu používáme zkratku IPRS.

Přepočet fakturovaných spotřeb na IPRS, resp. PRS se provádí z toho důvodu, aby byly jednotlivé údaje v rámci historie daného zákazníka porovnatelné. Přepočtem se eliminuje vliv teploty a také vliv různé délky fakturačních období (spotřeba je normovaná na rok).

Při dalších analýzách vycházíme z myšlenky, že zákazník je obtížně modelovatelný, jestliže vykazuje příliš velkou variabilitu spotřeby (přesněji IPRS). Nestabilní chování se dá charakterizovat např. následujícími kritérii:

1. poměr aktuální a historické spotřeby v posledním fakturačním období, tj.

$$K_i^{(1)} = \frac{IPRS_{in_i}}{PRS_{in_i}}, \qquad (3)$$

2. směrodatná odchylka iPRS v rámci historie zákazníka, tj.

$$K_i^{(2)} = \sqrt{\frac{1}{n_i - 1}\sum_{t=1}^{n_i}(IPRS_{it} - \overline{IPRS}_i)^2}, \qquad (4)$$

kde

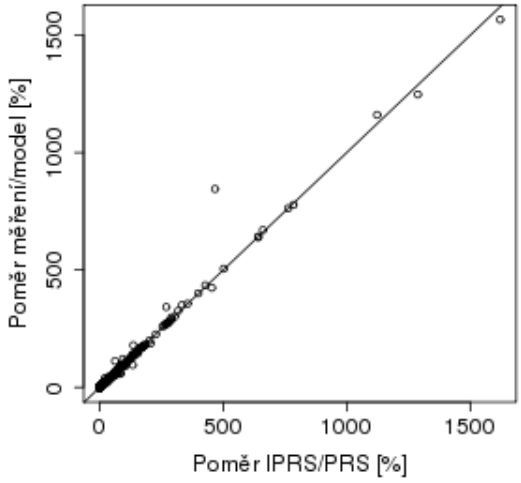
$IPRS_{it}$ je hodnota IPRS zákazníka i za fakturační období t,

$\overline{IPRS}_i$ je průměrná hodnota IPRS zákazníka i za celou jeho fakturační historii,

PRS_{it} je hodnota PRS zákazníka i za fakturační období t,

n_i je počet odečtů zákazníka i.

Kritérium (3) — poměr IPRS/PRS — zohledňuje poslední vývoj. Stabilnímu chování odpovídají hodnoty blízké 1. Vliv různé délky historie pro výpočet PRS byl částečně eliminován omezením se na zákazníky s delší historií (4 a více odečtů). Obrázek 1 navíc ukazuje, že



Obrázek 1: Vztah relativní chyby odhadu modelem TDD a poměrem IPRS/PRS.

je toto kritérium velmi úzce spjaté s obtížnou odhadnutelností spotřeby v daném fakturačním období. Obrázek

ukazuje vztah mezi relativní chybou odhadu modelem TDD a hodnotou kritéria (3). Je vidět, že se jedná téměř o přímou úměrnost.

Kritérium (4) — směrodatná odchylka — je jakousi globální mírou variability daného zákazníka. Obecně menší hodnota značí stabilnější odběry zákazníka, avšak při interpretaci je třeba vzít v úvahu skutečnost, že výše směrodatné odchylky rovněž závisí na celkové hladině spotřeby daného zákazníka.

2.2. Identifikované jednotky

Jako jednotka pro identifikaci vhodnosti osazování inteligentním měřením byla zvolena obec. Důvodem bylo, že obec může být dostatečně malá územní jednotka na to, aby bylo možno osadit měřením všechna přítomná odběrná místa. Z analýz byly vyloučeny obce, které mají 25 a méně odběrných míst (z důvodu zvýšeného rizika "planého poplachu"způsobeného např. jedním vysoce nestabilním zákazníkem).

Pochopitelně je teoreticky možné volit i jiné jednotky, jako je např. příslušnost k uzavřené lokalitě (za podmínky, že tato data jsou k dispozici – v současně používaném souboru nebyla), případně pokud netrváme na osazení celé lokality, lze volit větší územní celky jako např. okres, nebo volit i jiné kategorie (jako je např. nadmořská výška nebo počet zákazníků v obci).

2.3. Ohodnocení zvolených jednotek

Máme-li zvoleno kritérium pro ohodnocení zákazníka, např. jedno z kritérií (3), (4), a máme-li stanovenou jednotku k identifikaci, např. obec, zbývá zvolit kritérium pro ohodnocení těchto jednotek.

Hledáme tedy například obce, které jsou nějakým způsobem nejhorší dle zvoleného zákaznického kritéria. V případě použití kritéria (4), tj. směrodatné odchylky spotřeb daného zákazníka, můžeme pro ohodnocení obce použít např. medián směrodatných odchylek zákazníků v dané obci. Pochopitelně lze použít i jiné statistiky (průměr, maximum apod.), ale výhodou mediánu je určité potlačení vlivu extrémních jednotlivců.

V případě kritéria (3), tj. poměru IPRS za poslední fakturační období a PRS za totéž období, je situace poněkud komplikovanější. Neplatí zde totiž jednoduchá úměra "čím větší, tím hůře". Proto navrhujeme pro použití kritéria (3) následující postup:

1. Odstraníme zákazníky, kteří mají poslední IPRS a poslední PRS nižší než 7 620 kWh[1] (tyto zákazníky považujeme z hlediska osazování inteligentním měřením za nezajímavé, i kdyby měli nestabilní chování).
2. Vypočteme 10. a 90. percentil z hodnot $K_i^{(2)}$, tj. z poměrů IPRS/PRS pro (velké) zákazníky.
3. Jako "problematické"označíme zákazníky, pro něž hodnota $K_i^{(2)}$ leží nad 90. nebo pod 10. percentilem.
4. Pro každou obec spočítáme počet a podíl těchto "problematických"zákazníků a pro osazení volíme obce s nejvyšším podílem.

Přitom vyhodnocujeme pouze obce, které mají alespoň 25 "velkých"zákazníků (tj. zákazníků, jejichž poslední PRS nebo IPRS je větší než 7 620 kWh). Celkem se jedná o cca 75% všech obcí.

Každopádně je při použití jakékoli kombinace uvedených volitelných parametrů vhodné identifikované obce před rutinním osazováním inteligentními měřidly podrobit hlubší analýze za účelem ověření, zda byl identifikován opravdu hledaný problém, či zda byla obec vybrána v důsledku jiného jevu, který je z hlediska osazení měřením nezajímavý. V krajním případě může být příčinou výběru obce například jeden silně netypicky se chovající zákazník.

3. Příklady

V tomto odstavci uvedeme několik příkladů identifikovaných obcí pomocí výše uvedené metodiky. Dle postupu popsaného v odstavci 2.3 byly s využitím kritéria (3) vypočteny podíly "problematických"zákazníků v jednotlivých obcích. Uvažovány byly pouze obce, které mají alespoň 25 "velkých"zákazníků (tj. s poslední PRS a zároveň IPRS větší než 7 620 kWh). Tyto obce byly seřazeny podle zastoupení "problematických"zákazníků, přičemž jako základ pro podíl posloužil počet zákazníků s dlouhou historií v dané obci (zákazníci s třemi a méně odečty tedy do vyhodnocení vůbec nevstupují). Z důvodu důvěrnosti zpracovávaných údajů budou identifikované obce v následujícím textu uváděny pod kódovým označením.

V tabulce 1 je uvedeno deset "nejhorších"obcí, tj. obcí s největším zastoupením "problematických"zákazníků (dle definice v odstavci 2.3). Obec je identifikována názvem obce a okresem, pro úplnost je pro každou obec uveden celkový počet zákazníků, dále počet zákazníků s dlouhou historií, počet "velkých"zákazníků a počet problematických zákazníků. V posledním sloupci je

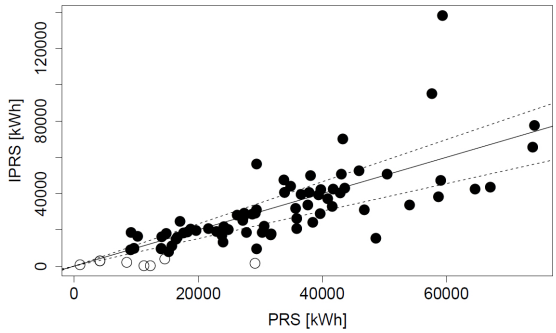
[1] Hodnota 7 620 kWh je definovaná vyhláškou [2] jakožto hranice dělící odběratele s teplotně závislým a teplotně nezávislým odběrem.

uveden procentní podíl problematických zákazníků na celkovém počtu zákazníků s dlouhou historií.

Obec	Počet odběrných míst celkem	dl. hist.	probl.	Podíl prob. [%]
1	430	82	36	43,90
2	106	30	11	36,67
3	127	34	12	35,29
4	285	40	14	35,00
5	116	29	10	34,48
6	65	51	17	33,33
7	46	36	12	33,33
8	159	32	10	31,25
9	137	78	36	29,51
10	191	27	10	29,41

Tabulka 1: Deset „nejhorších“ obcí z pohledu zastoupení zákazníků s vybočujícím poměrem IPRS a PRS.

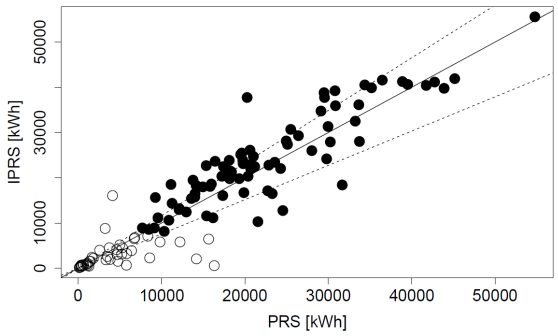
Obrázek 2 ilustruje situaci v „nejhorší“ obci 1. Na ose x jsou vyneseny hodnoty poslední PRS pro každého zá-



Obrázek 2: Porovnání poslední IPRS a poslední PRS v kWh pro jednotlivé zákazníky – obec 1.

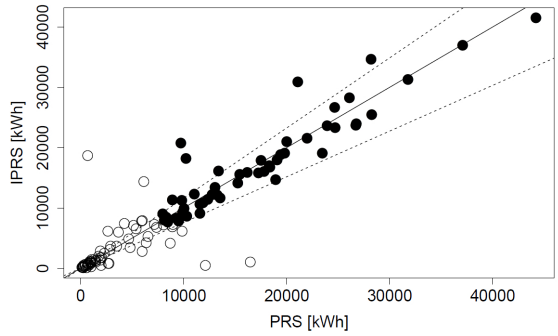
kazníka, na ose y pak hodnoty IPRS. V ideálním případě (tj. v případě stabilního odběru všech zákazníků) by se body pohybovaly okolo střední přímky. Čárkované přímky vyznačují hranice 10. a 90. percentilu. Body ležící mimo ohraničenou oblast reprezentují “problematické”zákazníky. Pro ilustraci obrázek obsahuje i zákazníky se spotřebou menší než 7 620 kWh. Ti jsou vyznačeni nevyplněnými kroužky.

Na obrázku 3 je vykreslena situace v další vybrané obci z tabulky 1, obci 9 z okresu Plzeň-Jih. Situace je poněkud jiná (např. většina odlehlostí leží v opačné polorovině, tzn. zákazníci jako by spíše zvyšovali spotřebu, taktéž vidíme nezanedbatelný počet odlehlostí i mezi zákazníky s malou spotřebou.

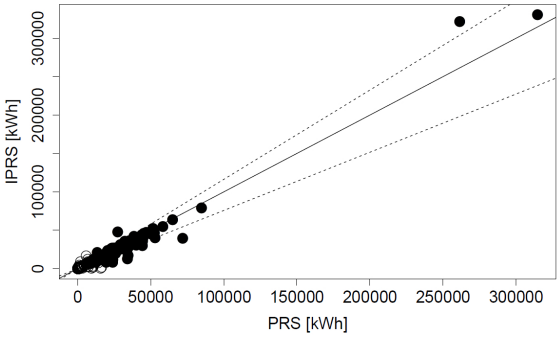


Obrázek 3: Porovnání poslední IPRS a poslední PRS v kWh pro jednotlivé zákazníky – obec 9.

Na obrázku 4 je pro porovnání zobrazena situace v obci z opačného konce žebříčku, tj. obce relativně stabilní, označené jako “obec A”. Obec má celkem 122 zákazníků s dlouhou historií, přičemž zastoupení problematických zákazníků je cca 5%. Na obrázku je vidět, že zákazníků mimo vymezenou oblast je poměrně dost, nicméně se jedná o zákazníky s malou spotřebou, kteří nebyli do kritéria zahrnuti.



Obrázek 4: Porovnání poslední IPRS a poslední PRS v kWh pro jednotlivé zákazníky – obec A.



Obrázek 5: Porovnání poslední IPRS a poslední PRS v kWh pro jednotlivé zákazníky – obec B.

Posledním příkladem je na obrázku 5 obec B, taktéž málo problematická obec. Obec má 1443 zákazníků s dlouhou historií, "problematičtí"zákazníci z toho tvoří méně než 1%. Z obrázku je patrné, že počet zákazníků mimo oblast vymezenou zvolenými kvantily je srovnatelný s ostatními uvedenými obcemi, ale celkový počet zákazníků uvnitř oblasti je řádově vyšší.

4. Závěr

Bylo diskutováno několik různých možností využití dat z inteligentních měřidel v případě jejich osazení. Ke všem těmto možnostem bylo přihlíženo při návrhu metodiky pro osazování měření. Navržená metodika je tedy jakýmsi kompromisem mezi protichůdnými požadavky vzhledem k různým možnostem využití.

Navržená metodika identifikuje obce s vysokým relativním výskytem problematických zákazníků, přičemž problematickým se rozumí takový zákazník, který má vysokou variabilitu spotřeby v čase a zároveň se jedná o zákazníka s teplotně závislou spotřebou.

V následujícím roce bude probíhat ze strany distribuční společnosti RWE GasNet ověření, zda navržená metodika opravdu postihuje problematické zákazníky a na základě výsledků této analýzy bude rozhodnuto o praktické aplikaci.

Poděkování

Autor děkuje svému školiteli Emilu Pelikánovi za revizi textu a věcné i formální připomínky, dále pak celé řešitelské skupině projektu GAMMA, bez níž by tato práce nevznikla.

Literatura

[1] M. Brabec, O. Konár, M. Malý, E. Pelikán, and J. Vondráček, "A Statistical Model for Natural Gas Standardized Load Profiles". *Journal of the Royal Statistical Society Series C – Applied Statistics*. 58 (1), pp. 123–139, 2009.

[2] Energetický regulační úřad, Vyhláška č. 365/2009 Sb. o Pravidlech trhu s plynem. In: *Sbírka zákonů České republiky.* částka 117, s. 5174–5236, 2009.

[3] OTE, a.s. Přepočtené TDD. *Ote-cr.cz* [online]. ©2010 [cit. 2012-08-06]. Dostupné z: http://www.ote-cr.cz/statistika/typove-diagramy-dodavek-plynu/ prepoctene-tdd.

[4] J. Vondráček, E. Pelikán, O. Konár, J. Čermánková, K. Eben, M. Malý, and M. Brabec, "A Statistical Model for the Estimation of Natural Gas Consumption". *Applied Energy*. 85 (5), pp. 362–370, 2008.

[5] WHICH? Energy monitors: Smart meters and energy monitors explained. *Which.co.uk* [online]. 2012 [cit. 2012-08-06]. Dostupné z: http://www.which.co.uk/energy/ creating-an-energy-saving-home/ guides/smart-meters-and -energy- monitors-explained/what-is-a-smart-meter/.

Weather Classification with Respect to NWP Model Output Precision

Post-Graduate Student:
MGR. PAVEL KRČ
Institute of Computer Science of the ASCR, v. v. i.
Pod Vodárenskou věží 2
182 07 Prague 8, CZ
krc@cs.cas.cz

Supervisor:
DOC. ING. EMIL PELIKÁN, CSC.
Institute of Computer Science of the ASCR, v. v. i.
Pod Vodárenskou věží 2
182 07 Prague 8, CZ
pelikan@cs.cas.cz

Field of Study:
Engineer Informatics in Transportation

Abstract

Numerical weather prediction (NWP) models can be a valuable source of weather data for simulating and predicting processes that have some weather dependency, for example predicting electricity production from renewable sources or modeling road weather. The reason for this is that most NWP models can directly export various meteorological variables on grid level, while some of these variables can not be obtained from traditional sources of weather data (point measurements and expert forecasts) with sufficient level of spatial and/or temporal detail. An example to this is predicting power output from photovoltaic (PV) power plant: while traditional weather forecast might provide only diurnal cloudiness prediction for larger regions, to reliably model PV power output one needs a fine time series of solar irradiance prediction for the exact location of the PV plant along with other variables. Obtaining these data from NWP models is straightforward, but it has some limitations. Some of the meteorological variables available from NWP models may not have been primarily designed as output variables and may be tuned for model's internal balance rather than as a direct predictor of the respective physical value. It can further be shown that even though different NWP models may predict the atmospherical conditions as a whole with comparable precision, they might vary significantly in precision, bias and stability of these internal variables. This work will focus on finding suitable methods for classifying different weather situations for the purpose of finding bias corrections and interval and probability forecasts of some of the meteorological variables that are useful for the above mentioned purposes. With respect to different NWP models, it will also try to find optimum model combinations when multimodel forecast is used.

Falsification of Hybrid Systems

Post-Graduate Student:
JAN KUŘÁTKO
Institute of Computer Science of the ASCR, v. v. i.
Pod Vodárenskou věží 2
182 07 Prague 8, CZ
kuratko@cs.cas.cz

Supervisor:
STEFAN RATSCHAN
Institute of Computer Science of the ASCR, v. v. i.
Pod Vodárenskou věží 2
182 07 Prague 8, CZ
stefan.ratschan@cs.cas.cz

Field of Study:
Scientific Computations

Abstract

The aim of this paper is to briefly introduce the concept of a hybrid system and explain the term of an error trajectory. Furthermore, the general problem of falsification of hybrid systems is stated with a basic approach for solving it. This approach combines results from formal verification of hybrid systems and optimization.

1. Introduction

Hybrid systems are dynamical systems with continuous states as well as discrete states. These systems are important since they can function as models of embedded systems such as aircraft and cars. Falsification of hybrid systems is the problem of finding an error trajectory that originates in a set of initial states and enters a set of unsafe states. If at least one such trajectory is found, then the system is said to be unsafe.

Here is an example. Suppose we have a heated room and the heating in it set to turn on when the temperature is below $10°\mathrm{C}$ and to turn off when the temperature is $22°\mathrm{C}$. The heating is a hybrid system H, which has discrete states *on* and *off*, respectively. In the continuous part the temperature in the room is governed by differential equations. When the heating is on, then the temperature is increasing and when the heating is off, the temperature is decreasing. An initial set can be an interval $I = [0°\mathrm{C}, 15°\mathrm{C}]$ and the set of unsafe states can be $U = [22.5°\mathrm{C}, 30°\mathrm{C}]$. Then the error trajectory of this system starts in I and reaches U.

Previous research was done in [1], however, only linear systems were considered. Non-linear systems were dealt with in [2], where `HSOLVER` [3], software for verification of hybrid systems, was used extensively. We aim to interconnect approaches from [2] and [1], exploiting the continuous nature of hybrid systems, now, with non-linear dynamics. By that we mean to do a local search in a part of an initial set, from which an unsafe set can be reached, to obtain a desired trajectory.

2. Problem formulation and basic approach

Let H be a hybrid system and I, U be two sets. Set I contains the initial states of the system and U is the set of unsafe states. Our task is to find any trajectory which originates in I and, if possible, reaches set U.

As suggested before, we approach this task as an optimization problem which is solved by local search in I. Assuming the simulation runs for a given time $T < \infty$, we seek a state in I for which the trajectory *minimizes* an appropriate objective function. The most convenient formulation of an objective function and algorithm itself are now explored.

3. Acknowledgement

This work was supported by Czech Science Foundation GAČR grant P202/12/J060 and long-term financing of the Institute of Computer Science (RVO 67985807).

References

[1] H. Abbas and G. Fainekos, "Linear Hybrid System Falsification With Descent", *eprint arXiv:1105.1733*, tech report, 2011.

[2] S. Ratschan and J.-G. Smaus, "Verification-Integrated Falsification of Non-Deterministic Hybrid Systems", *Analysis and Design of Hybrid Systems 2006*, pp. 371–376, 2006.

[3] S. Ratschan, Z. She, and T. Dzetkulič, "HSOLVER", http://www.hsolver.sourceforge.net.

The Selection of Surrogate Models in Evolutionary Algorithms

Post-Graduate Student:
MGR. MARTIN PILÁT

Faculty of Mathematics and Physics
Charles University in Prague
Malostranské náměstí 25

118 00 Prague 1, CZ

Martin.Pilat@mff.cuni.cz

Supervisor:
MGR. ROMAN NERUDA, CSC.

Institute of Computer Science of the ASCR, v. v. i.
Pod Vodárenskou věží 2

182 07 Prague 8, CZ

roman@cs.cas.cz

Field of Study:
Theoretical Computer Science

This research was partially supported by SVV project No. 265 314 and by Czech Science Foundation project No. 201/09/H057.

Abstract

In this paper we discuss the problem of the selection of surrogate models for the use with evolutionary algorithms. We compare the types of models selected by two different model selectors and discuss how the models affect the performance of the evolutionary optimizer and how the types of selected models change during the run of the evolution.

1. Introduction

In the recent years there has been an increasing interest to create new surrogate based evolutionary algorithms. The main motivation is that evolutionary algorithms use a large number of objective function evaluations which may be costly in practice – either in terms of computing power, or even money (e.g. when a real-life experiments needs to be made to evaluate the individual).

The goal of surrogate modeling is to create a cheaper approximation of the costly objective function and use it instead (or together with) the real objective. The idea of surrogate modeling is quite old in the field of single-objective optimization, however it is relatively new in the field of multiobjective optimization.

The choice of the model is another rarely studied question. Usually the model is selected based on its mean square error on a training set. However, it may be beneficial to choose another metric for evaluation of the models. For example the relation preservation (i.e. how well the model preserves the ordering among individuals) may be a more natural selection criterion as most of the evolutionary algorithms use comparison of individuals during their run [1].

We have recently presented a meta-learning based framework for surrogate modeling in evolutionary algorithms. The framework is able to deal with all the problems mentioned above: it can theoretically recommend models based on the type of problem studied and choose among the recommended models the one which shall be used.

We focus on the way how the model is selected, and how it affects the performance of the resulting algorithm. Namely, the mean square error and relation preservation are considered and compared in a multiobjective setting. Both when the model is used in a kind of local search to optimize the generated individuals, and when the model is used to pre-select the promising individuals from those generated.

In this paper we further analyze results we have presented earlier and discus which types of model got selected by the different selectors and how the selected models changed during the run of the evolution. This should lead to deeper understanding of the proposed meta-learning framework.

The rest of the paper is organized as follows: in the next section we briefly describe the problem of multiobjective optimization and related work from the literature. Section 3 contains the description of the system in the most general case, and Section 4 describes the settings used to test the framework and the effect of the way, how the models are selected. Section 5 provides the results and Section 6 discusses the different types of models used during the evolution. Finally, Section 7 concludes the paper and provides ideas for future research.

2. Preliminaries and Related Work

Contrary to single-objective optimization, in multiobjective optimization there are more objective functions, which shall be optimized simultaneously. These

objective functions are usually conflicting, and thus there is not a single solution, which would be optimal for all of them. This leads to a set of so called Pareto optimal solutions.

The following definitions introduce the multiobjective optimization problem and the Pareto dominance relation, which is used to compare two potential solutions to the problem.

Definition 1 *The* multiobjective optimization problem (MOP) *is a quadruple* $\langle D, O, \vec{f}, C\rangle$*, where*

- D *is the* decision space
- $O \subseteq \mathbb{R}^n$ *is the* objective space
- $C = \{g_1, \ldots, g_m\}$*, where* $g_i : D \to \mathbb{R}$ *is the set of* constraint functions (constraints) *defining the* feasible space $\Phi = \{\vec{x} \in D | g_i(\vec{x}) \leq 0\}$
- $\vec{f} : \Phi \to O$ *is the vector of* n objective functions (objectives)*,* $\vec{f} = (f_1, \ldots, f_n), f_i : \Phi \to \mathbb{R}$

$\vec{x} \in D$ *is called the* decision vector *and* $\vec{y} \in O$ *is denoted as the* objective vector.

Only minimization problems are usually considered. Maximization and mixed problems may be transformed to minimization problems by multiplying the functions, which shall be maximized by -1.

In the field of multiobjective optimization, problems with more than 4 objectives are often called *many-objective*, as this higher number of objectives poses another challenges for the MOEAs (e.g. the dominance relation defined in the next paragraph loses its power to discriminate between good and bed individuals as most of them are mutually incomparable).

To compare two decision vectors, we define so called *Pareto dominance relation*. If one vector is better (has lower objective values) for all of the objective functions, we say it dominates the other vector. This is formally stated in the following definition.

Definition 2 *Given decision vectors* $\vec{x}, \vec{y} \in D$ *we say*

- $\vec{x}$ weakly dominates $\vec{y}$ *(*$\vec{x} \preceq \vec{y}$*) if* $\forall i \in \{1 \ldots n\} : f_i(\vec{x}) \leq f_i(\vec{y})$.
- $\vec{x}$ does not dominate $\vec{y}$ *(*$\vec{x} \not\preceq \vec{y}$*) if* $\vec{y} \preceq \vec{x}$ *or* $\vec{x}$ *and* $\vec{y}$ *are incomparable*

Now, we can state the goal of the multiobjective optimization, it is to find those decision vectors, which are minimal in the Pareto dominance relation.

Definition 3 *The* solution of a MOP *is the* Pareto (optimal) set

$$P^* = \{\vec{x} \in \Phi | \, \forall \vec{y} \in \Phi : \vec{y} \not\preceq \vec{x}\}$$

The projection of P^* *under* $\vec{f}$ *is called the* Pareto optimal front.

The Pareto optimal set is usually infinite for continuous optimization and thus we usually seek a finite approximation of this set. This approximation should be close to the Pareto set (ideally it is a subset of it) and should also be evenly distributed along the Pareto front.

We can extend the Pareto dominance relation to such approximations and compare them with this relation, however, as the ordering is only partial, there would be pairs of approximations which are mutually incomparable (in fact, most of such pair would be incomparable). As we want to compare approximations, which are solutions found by a multiobjective optimizer, we need a way to compare any two sets.

During past years, many measures were proposed to compare such Pareto set approximation and one of the most often used is the hypervolume indicator [2]. This indicator expresses the hypervolume of the objective space, which is dominated by the solutions.

Definition 4 *Let* $R \subset O$ *be a reference set. The* hypervolume metric $\mathcal{S}$ *is defined as*

$$\mathcal{S}(A) = \lambda(H(A, R))$$

where

- $H(A, R) = \{x \in O | \, \exists \vec{a} \in A \, \exists \vec{r} \in R : \forall i \in \{1, \ldots, n\} : f_i(\vec{a}) \preceq \vec{x}_i \preceq \vec{r}_i\}$ *where* f_i *is the* i*-th objective function*
- λ *is Lebesgue measure with* $\lambda(H(A, R)) = \int_O \vec{1}_{H(A,R)}(z)dz$ *and* $\vec{1}_{H(A,R)}$ *is the characteristic function of the set* $H(A, R)$

The reference set bounds the hypervolume from above. It usually contains only a single reference point. We should note here that although the definition of the hypervolume indicator is quite simple, its computation is known to be #P-complete [3] and its complexity grows exponentially with the number of objectives.

Various algorithms were proposed during the last years to deal with the problem of multiobjective optimization. Among them, NSGA-II [4] became one of the most popular. NSGA-II is based on the Pareto dominance relation, which is central to most of the multiobjective evolutionary algorithms (MOEA). In this case, the population is divided into non-dominated fronts, and fitness is assigned to the individuals based on the front they belong to, and on a crowding distance which should promote diversity and ensure the a well-spread set of Pareto optimal solutions. However, NSGA-II has problems when the number of objectives rises, as the dominance relation fails to discriminate between the individuals and most of them are assigned to the same non-dominated front [5].

One of the approaches which at least partially reduces the problem is the use of indicators, e.g. in the ϵ-IBEA algorithm [6]. The indicators are in fact generalizations of the Pareto dominance relation with more discriminative power (i.e. they are able to compare even individuals which are mutually non-dominated). The values of the indicators are then used for the fitness assignment in IBEA.

As the MOEAs develop, new topics come to the focus of the community. Recently, the use of surrogate models has gained a lot of attention. The goal is to reduce the number of objective function evaluations needed to gain a good (preferably optimal) solution. There are at least two different approaches: the more simple one uses a surrogate for each of the objective functions separately (e.g. the surrogate version of NSGA-II presented in [7]), the other approach uses so called aggregate surrogate models. These provide one value which expresses the quality of the particular individual. An example may be found in [8], where the authors use a special kind of support vector machines to predict whether an individual is dominated or not.

Another popular MOEA – the MOEA/D [9] also has a surrogate variant [10]. MOEA/D is based on decomposition – it decomposes the multiobjective problem into several single-objective ones. These are than optimized at once. The surrogate version uses Gaussian processes to model each of the functions and derives models for each of the decompositions of the original multiobjective problem. The optimum of the model is found using the non-surrogate MOEA/D and selected individuals are added back to the population.

However, there are not many references to algorithms which would deal with more than one model. One of the exceptions is [11], where the authors use two different local meta-models. Both are trained to approximate a weighted sum of the objectives. One is an ensemble model, the other is a low order polynomial. Two single-objective algorithms are run to find optima of the respective models, which are then precisely evaluated. A selection procedure is used to decide which of the individuals (if any) is added to the population.

Another work was published by [1], and the approach used there is in a sense complementary to the one presented here. This paper compares different models and monitors various features of the models during the evolutionary search. One of the results of the work is, that mean square error might not be the best measure of the suitability of the model for the use with evolutionary algorithms. The authors also argue, that even model with large mean square error performs well when they preserve the ordering of the individuals well. In contrast, in this paper we will study some of the measures to select the model which is used during the evolution.

3. Meta-Learning for Surrogate Modeling

Usually, when a surrogate model is used in an evolutionary algorithms its parameters are manually assigned, it is trained and used to predict the values which are modeled.

Here, we present a framework, which automatically selects a model from a set of models, either chosen manually, or recommended by a meta-learning engine. The framework uses a meta-model recommender to select some models from the set of all available regression models. These models are uninitialized, which means that their internal parameters (e.g. the weights of synapses of a neural network) are utilized. Only their external parameters (e.g. the number of neurons in the hidden layer) are specified. Later, these models are trained using a model trainer, which sets its internal parameters.

The following definitions provide more accurate expression of these ideas.

Definition 5 *An* n-ary regression model *is a function* $f_w : \mathbb{R}^n \to \mathbb{R}$ *where* w *is the vector of internal parameters of the model.*

By the internal parameters we mean the parameters which are tuned during model training, e.g. the weights of synapses of a neural network.

Definition 6 *Let* $\mathcal{R}_U^n$ *is the class of all* n*-ary regression models viewed as function of their internal parameters. We will call* $\mathcal{M} : \mathcal{R}_U \to \mathcal{P}(\mathcal{R}_U)$ *a* meta-learning recommender.

Thus, the meta-learning recommender selects some of the models (and their parameters) from the class of all available regression models. By regression model we mean e.g. multilayered perceptrons, support vector regression, Gaussian processes etc. By their parameters we mean e.g. the number of neurons in a hidden layer of a multilayer perceptron, or the type of kernel in the case of support vector regression. In fact, the recommender selects classes of models with their internal parameters (e.g. the weights of synapses in the neural network) left uninitialized.

After the models have been selected, we need to train them. To this end we will utilize a model trainer.

Definition 7 *Let $\mathcal{R}_U$ is as above and $\mathcal{R}_I$ is the class of all regression models with all their internal parameters set. A function $\mathcal{T} : \mathcal{R}_U \to \mathcal{R}_I$ is a* model trainer.

The training function usually minimizes the mean square error of the model, however the framework allows for the optimization of other different measures.

Now, we have several trained models and we need to select one to use it as the surrogate for the modeled function.

Definition 8 *Let $\mathcal{R}_I$ is as above. Function $\mathcal{S} : \mathcal{P}(\mathcal{R}_I) \to \mathcal{R}_I$ is a* model selector.

Most often, the model with the minimal mean square error on a validation set is selected by the selector. In this work we consider two different selectors: one which selects the model with the lowest MSE, and one which selects the model which has the best relation preservation.

Definition 9 *Let I be the indicator function (1 if its argument is true and 0 if it is false). Let $F : \mathbb{R}^n \to \mathbb{R}$ is a modeled function, and let $M \in \mathcal{R}_I$ is its regression model. Let $V \subseteq \mathbb{R}^n$ is a finite validation set. The* relation preservation measure *is defined as*

$$RP(M) = \frac{1}{|V|(|V|-1)} \sum_{x,y \in V} I(F(x) \leq F(y) \& M(x) \leq M(y)).$$

The relation preservation measure might be more important when the use of a model as a surrogate function is considered in evolutionary algorithms, as most EAs compare two individuals instead of using the value of the function directly.

For the sake of completeness we also define the well known mean square error here.

Definition 10 *Let V, F, and M are defined as above. The* mean square error *is defined as*

$$\frac{1}{|V|} \sum_{x \in V} (F(x) - M(x))^2.$$

Now, we can define two types of selectors, the mean square selector and the relation preservation selector.

Definition 11 *Let R is a set of trained regression models. Than $\mathcal{S}_M(R) = \arg\min_{m \in R} MSE(m)$ is the* mean square selector *and $\mathcal{S}_P(R) = \arg\max_{m \in R} RP(m)$ is the* relation preservation selector.

We could also define a „model combinator“, which would combine the trained models and create an ensemble model. However, this is not needed as ensemble models can be considered, as any other models, and thus the described framework is general enough to include them. It still might be may be advantageous to define such a combinator in the future, if more finer details are studied.

4. Surrogate Evolutionary Algorithms

There are two main ways, how to use a surrogate model to augment an evolutionary algorithm: to pre-select individuals – individuals are first evaluated using the surrogate model, and only those promising are evaluated using the real (and presumably expensive) objective function, or in a kind of local search – some of the individuals are selected, and the surrogate model is used to lead local search and improve the individuals.

In this work we deal with both of these approaches, and test the effect of the type of model selector on the performance of evolutionary algorithm.

The pre-selection approach is tested on a surrogate version of NSGA-II [4] and ϵ-IBEA [6] algorithm. The model is used to pre-screen the newly generated offspring in such a way, that only non-dominated offspring (according to the model) are considered during the regular selection (and only those are evaluated). To this end each of the objectives is modeled separately, and the predicted values of the objectives are used to judge the dominance and non-dominance of the individuals. This corresponds to the pre-selection scheme we used in our previous work [12].

The local search approach is tested in a different way. In this case we use our aggregate surrogate model which we proposed earlier. This model is trained to predict the distance of each individual to the currently known set of Pareto optimal individuals. More specifically, there is an archive A of previously evaluated individuals, and it is used to create a training set T for the surrogate model in the following way:

$$T = \{(x_i, y_i) | y_i = -d(x_i, P)\}$$

where $d(x, y)$ is the Euclidean distance of individuals x and y in the decision space, P is the set of non-dominated individuals in the archive, and $d(x, P)$ is the distance of individual x to the closest point in the set P.

The model is trained to predict the values as specified in the training set and than an evolutionary algorithm (this time only a single objective one) is used to find the optima of this model in the surroundings of a selected locally improved individual. To this end, the variables of the selected individual are perturbed to create the initial population of the internal evolutionary algorithm. This corresponds to the way the surrogate model is used in ASM-MOMA [13].

5. Experiments

We use the H_{ratio} metric introduced in our previous work [13, 14] to compare the numbers of objective function evaluations needed to get a solution of pre-specified quality. The metric is defined as

$$H_{ratio} = \frac{H(P)}{H(P^*)}$$

where $H(P)$ is the hypervolume [15] of the set of individuals found by the algorithm, and the $H(P^*)$ is the hypervolume of the globally optimal Pareto front. We use $\vec{2} = (2, 2)$ as the reference point in the hypervolume computation.

To assess the effect of the different selectors on the performance of the multiobjective evolutionary algorithm, we selected ten regression models (thus simulating the meta-learning recommender), trained them using their respective training functions (which minimize the mean square error) and then used the selectors to select the best one among them to use as the surrogate model during the evolution. In all cases one third of the training set as specified above was used as the validation set in the selection process, while the remaining two thirds were used for the training.

The ten types of models were selected to represent the most commonly used models in the field of surrogate evolutionary computation, these are:

- linear regression (later reffered to as LINEAR),
- 3 variants of support vector regression – with polynomial kernels of degrees one and two (SVR-Poly-1 and SVR-Poly-2, respectively), and with the RBF kernel (SVR-RBF),
- 3 architectures of multilayer perceptrons – with 2, 5, and 10 neurons in the single hidden layer (MLP-2, MLP-5, and MLP-10 respectively),
- 2 variants of RBF networks – with 2 and 5 RBF units (RBF-2 and RBF-5),
- Gaussian processes (GAUSS).

5.1. Results

In this section we present results of the algorithm with both local search and pre-selection used. This version most closely resembles our previous algorithm [12], which also uses the surrogate model in both phases. We do not discuss these number in great detail, they are provided mostly to show how the different types of model selectors affect the convergence speed of the algorithm. More detailed discussion (together with more results) were presented in [16].

The results are summarized in Table 1. In this case the differences between the two types of selectors are most pronounced. We can see that in most cases the relation preservation selector is better than the MSE selector. The only exception being the ZDT3 test where both selectors perform almost the same.

The largest difference can be observed on the ZDT6 test, where the number of evaluations needed to attain the $H_{ratio} = 0.99$ with the RP selector is about a third lower than with the MSE selector.

6. Detailed Analysis

Although it is obvious that the relation preservation can dramatically improve the results of the optimizer, the question, which models are selected using both of the selectors remains. Do the selectors indeed select different models? Is the same model used during the whole run of the evolution, or do the models change as the evolution proceeds?

Tables 2 and 3 show the relative frequencies with which each of the types of models was selected and used during the evolution for different problem, their objectives

Table 1: Median number (20 runs) of function evaluations needed to reach the specified H_{ratio} on ZDT1, ZDT2, ZDT3 and ZDT6 test problems in the scenario, where both local search and pre-selection were used. The name of the method is encoded as follows: NSGA and IBEA indicate the type of the external evolutionary algorithm (NSGA-II and ϵ-IBEA respectively) and MSE and RP encode the type of selector used to select the best model from the set.

	ZDT1					**ZDT2**				
H_{ratio}	0.5	0.75	0.9	0.95	0.99	0.5	0.75	0.9	0.95	0.99
NSGA-MSE	544	816	1207	1300	6089	209	267	367	483	866
NSGA-RP	485	1069	1228	1389	4843	229	268	342	417	794
IBEA-MSE	487	1194	1575	1660	2499	196	236	332	429	990
IBEA-RP	514	1541	1799	1842	2507	204	259	335	410	983
	ZDT3					**ZDT6**				
H_{ratio}	0.5	0.75	0.9	0.95	0.99	0.5	0.75	0.9	0.95	0.99
NSGA-MSE	202	265	366	402	578	1658	3079	5280	7847	14889
NSGA-RP	221	301	413	490	612	1626	2602	5023	7202	12756
IBEA-MSE	212	284	349	402	549	1952	3961	7053	9522	18208
IBEA-RP	236	303	419	469	632	1555	3123	5599	7213	11773

Table 2: Relative frequencies of selected models when ϵ-IBEA is used as the external algorithm.

Problem	ZDT1						ZDT2					
Function	AGR		F0		F1		AGR		F0		F1	
Model sel.	MSE	RP	MSE	RP	MSE	RP	MSE	RP	MSE	RP	MSE	RP
LINEAR	0.00	0.01	1.00	1.00	0.25	0.38	0.02	0.17	1.00	1.00	0.13	0.07
SVR-Poly-1	0.02	0.03	0.00	0.00	0.37	0.33	0.06	0.05	0.00	0.00	0.09	0.02
SVR-Poly-2	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
SVR-RBF	0.84	0.90	0.00	0.00	0.31	0.21	0.67	0.64	0.00	0.00	0.11	0.20
MLP-2	0.05	0.00	0.00	0.00	0.01	0.00	0.15	0.03	0.00	0.00	0.07	0.02
MLP-5	0.03	0.01	0.00	0.00	0.02	0.00	0.03	0.01	0.00	0.00	0.16	0.13
MLP-10	0.02	0.02	0.00	0.00	0.01	0.05	0.01	0.03	0.00	0.00	0.42	0.53
RBF-2	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
RBF-5	0.00	0.00	0.00	0.00	0.00	0.00	0.01	0.00	0.00	0.00	0.00	0.00
GAUSS	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.02	0.00	0.00	0.00	0.00
Problem	ZDT3						ZDT6					
Function	AGR		F0		F1		AGR		F0		F1	
Model sel.	MSE	RP	MSE	RP	MSE	RP	MSE	RP	MSE	RP	MSE	RP
LINEAR	0.09	0.12	1.00	1.00	0.24	0.14	0.04	0.06	0.00	0.01	0.00	0.02
SVR-Poly-1	0.12	0.18	0.00	0.00	0.30	0.22	0.01	0.07	0.00	0.00	0.00	0.01
SVR-Poly-2	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
SVR-RBF	0.50	0.54	0.00	0.00	0.39	0.53	0.18	0.19	0.00	0.00	0.01	0.76
MLP-2	0.14	0.04	0.00	0.00	0.00	0.02	0.28	0.08	0.24	0.09	0.21	0.06
MLP-5	0.05	0.01	0.00	0.00	0.01	0.01	0.24	0.15	0.30	0.27	0.10	0.01
MLP-10	0.03	0.04	0.00	0.00	0.01	0.01	0.21	0.13	0.30	0.35	0.04	0.01
RBF-2	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.04	0.00	0.00	0.04	0.00
RBF-5	0.02	0.00	0.00	0.00	0.00	0.01	0.00	0.22	0.02	0.02	0.47	0.04
GAUSS	0.01	0.04	0.00	0.00	0.01	0.02	0.00	0.02	0.10	0.21	0.11	0.05

Table 3: Relative frequencies of selected models when NSGA-II is used as the external algorithm.

Problem	ZDT1						ZDT2					
Function	AGR		F0		F1		AGR		F0		F1	
Model sel.	MSE	RP	MSE	RP	MSE	RP	MSE	RP	MSE	RP	MSE	RP
LINEAR	0.15	0.16	1.00	1.00	0.20	0.27	0.15	0.16	1.00	1.00	0.22	0.21
SVR-Poly-1	0.17	0.33	0.00	0.00	0.18	0.13	0.11	0.35	0.00	0.00	0.14	0.13
SVR-Poly-2	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
SVR-RBF	0.20	0.23	0.00	0.00	0.01	0.00	0.30	0.29	0.00	0.00	0.09	0.08
MLP-2	0.15	0.00	0.00	0.00	0.24	0.24	0.06	0.04	0.00	0.00	0.10	0.13
MLP-5	0.15	0.03	0.00	0.00	0.20	0.17	0.11	0.02	0.00	0.00	0.17	0.12
MLP-10	0.14	0.15	0.00	0.00	0.14	0.15	0.22	0.08	0.00	0.00	0.25	0.30
RBF-2	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
RBF-5	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
GAUSS	0.00	0.06	0.00	0.00	0.00	0.00	0.01	0.01	0.00	0.00	0.00	0.00
Problem	ZDT3						ZDT6					
Function	AGR		F0		F1		AGR		F0		F1	
Model sel.	MSE	RP	MSE	RP	MSE	RP	MSE	RP	MSE	RP	MSE	RP
LINEAR	0.14	0.17	1.00	1.00	0.22	0.30	0.14	0.05	0.00	0.10	0.01	0.27
SVR-Poly-1	0.17	0.28	0.00	0.00	0.16	0.39	0.31	0.16	0.00	0.03	0.00	0.12
SVR-Poly-2	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
SVR-RBF	0.24	0.31	0.00	0.00	0.32	0.15	0.08	0.22	0.00	0.05	0.09	0.31
MLP-2	0.14	0.02	0.00	0.00	0.10	0.05	0.15	0.06	0.35	0.15	0.30	0.04
MLP-5	0.18	0.03	0.00	0.00	0.11	0.02	0.12	0.13	0.24	0.23	0.06	0.02
MLP-10	0.08	0.12	0.00	0.00	0.04	0.02	0.13	0.17	0.17	0.24	0.00	0.02
RBF-2	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.03	0.01	0.00	0.05	0.02
RBF-5	0.00	0.00	0.00	0.00	0.00	0.00	0.01	0.06	0.02	0.01	0.32	0.01
GAUSS	0.01	0.04	0.00	0.00	0.01	0.03	0.01	0.08	0.18	0.16	0.12	0.14

(F0 and F2) as well as for the aggregate model used during the local search phase (AGR).

There are some obvious observations: As the first objective of the ZDT1, ZDT2, and ZDT3 problems is a simple linear function, one would expect that the linear regression is the model, which is chosen most often. Regardless of the type of selector, when this function is modeled linear regression is always chosen. It matches the function precisely and thus its MSE is equal to 0 (almost), and all the relations are preserved. This is confirmed by the tables, which indicate that linear regression was indeed always used as to model for this function.

We can also make more detailed observations regarding individual models: the most interesting one being, that (except for the ZDT6 test problem) Gaussian processes and RBF networks are rarely used (i.e. other model have better relation preservation and lower MSE). The same holds for support vector regression with polynomial kernel of degree 2.

Now, let us compare the types of models selected by the two selectors. For the two problems with little differences in overall results (i.e. ZDT2 and ZDT3), as well as ZDT1 with the ϵ-IBEA we can see that both the selectors select the same surrogate models with similar frequencies. In the case of the other problems, where the differences between the two selectors are larger, the MSE selector tends to select the multilayer perceptron based models, while the relation preservation selector selects linear regression and support vector regression based models more often.

Although the results presented in the Tables provide nice overview of the types of models, which are better for both of the selectors, they do not express the dynamic nature of the evolutionary algorithms. In the next few paragraphs, we show, how the frequencies change during the run of the evolution. To this end we use pictures, which present the relative frequencies the different models were selected with during the 20 runs of the algorithms we made. The lines are symmetric moving averages with window size of 15 generations. The aggregate model differs from the individual objectives. It is based on the distance from the currently known Pareto front and thus should be more simple to model.

Figure 1 compares the relative frequency of the types of models selected using the RP and MSE selectors respectively on the ZDT6 problem and its aggregate surrogate function. We can see, that in the beginning, both

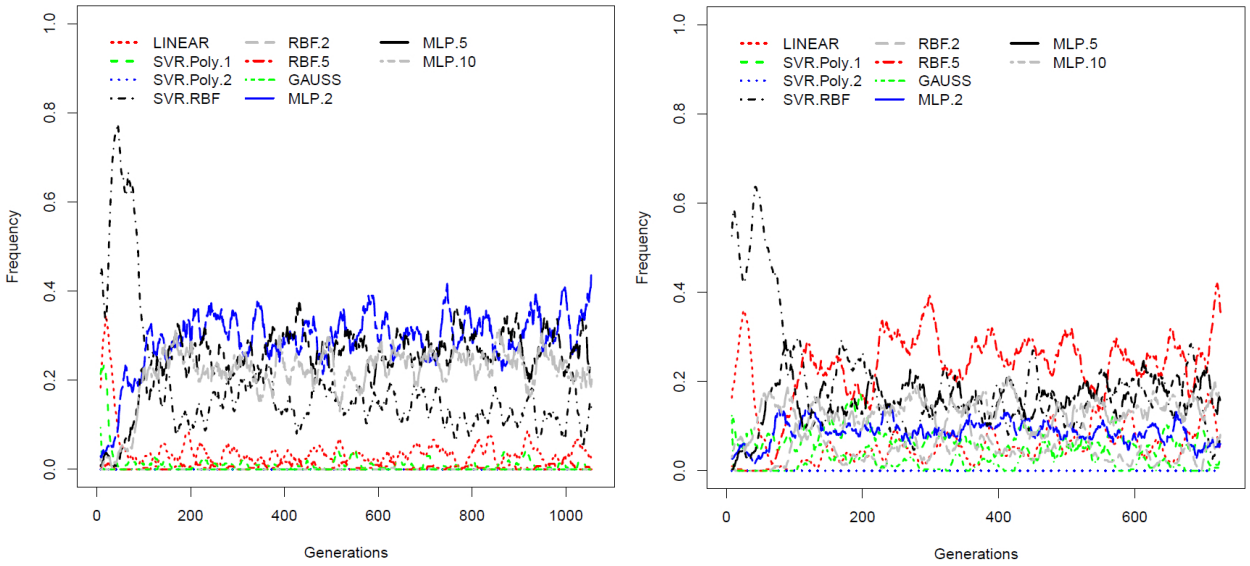


Figure 1: The relative frequency of the types of models selected based on the MSE (left) and RP (right) selector. The ZDT6 test problem with ϵ-IBEA fitness. Aggregate surrogate model.

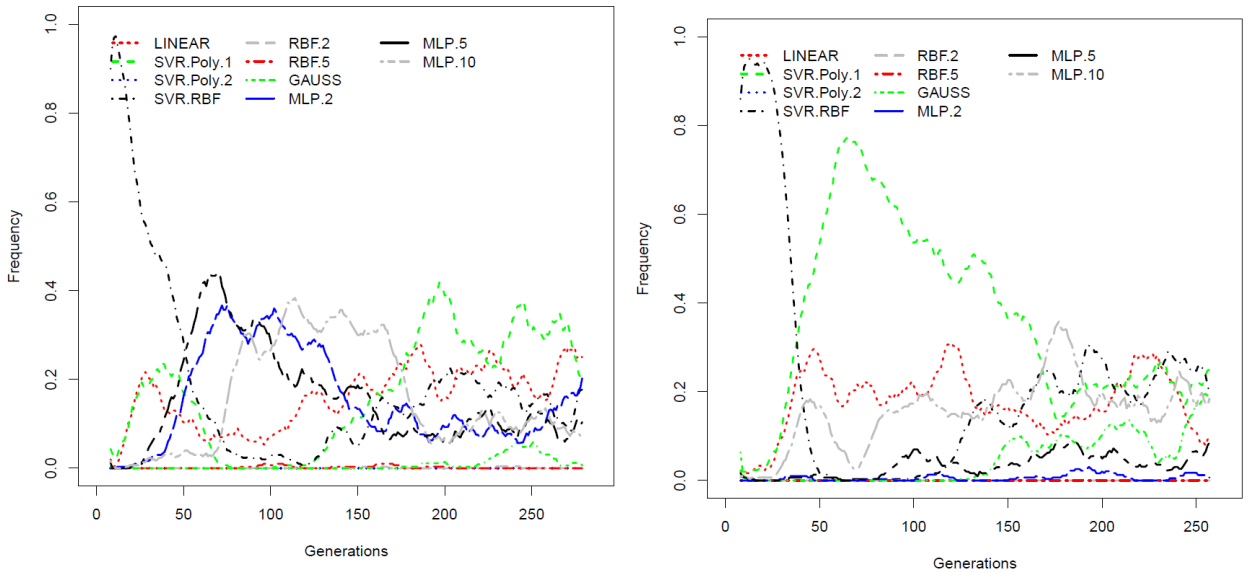


Figure 2: The relative frequency of the types of models selected based on the MSE (left) and RP (right) selector. The ZDT1 test problem with NSGA-II fitness, aggregate surrogate function.

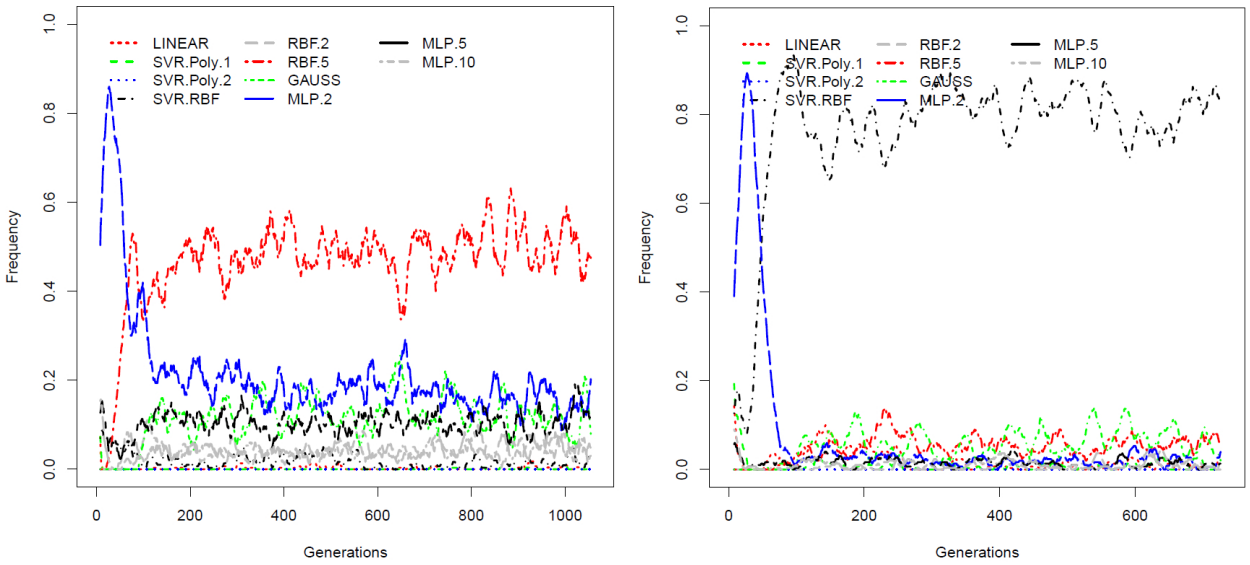


Figure 3: The relative frequency of the types of models selected based on the MSE (left) and RP (right) selector. The first objective function of the ZDT6 test problem with ϵ-IBEA fitness

the selectors tend to select the support vector regression with RBF kernels, while later the RP selector tends to select the RBF network based models, while the MSE selector selects mostly the multilayered perceptron based models. It means, that RBF networks in this case provide better relation preservation than the MLP models, although both the models are trained to minimize the MSE (and MLPs are better in this metric, as they are more often selected by the MSE selector). This also shows, that while both the metrics are strongly correlated, as discussed earlier, the ranking of the models may differ significantly.

Another interesting development may be observed (Figure 2) on the ZDT1 test problem with the NSGA-II based selection (aggregate surrogate model). Again, in the beginning both the selectors prefer the SVR based models – namely the in the beginning one with RBF kernel. However, later, the RP selector starts to select the SVR with polynomial kernel and the MSE selector selects (again) the MLP based models. In this case, we can see large differences between the models used in the beginning and later in the evolution. This emphasizes the importance of dynamic model selection, as the modeled environment changes during the run of the evolution.

The situation is easy for the first objective function on all the test problems but ZDT6, so we do not present any pictures here.

For the second objective function, the situation gets interesting again. For example in the ZDT6 test problem with ϵ-IBEA as the external evolutionary algorithm we can see (Figure 3), that in the first approximately 50 generation both the selectors use multilayered perceptron with 2 neurons in the hidden layer, but later, the RP selector uses support vector regression with RBF kernels, while the MSE selector switches to RBF networks with 5 RBF units.

7. Conclusions

Earlier, we have shown that relation preservation is an interesting measure, which helps to improve the convergence speed of the evolutionary optimizers. In this paper we have analyzed its performance further and we have discussed the different types of models selected based on this measure.

We have shown that the MSE selector tends to select multilayer perceptron based models, while the RP selector tends to select more simple support vector regression based models. The better results of the SVR models correspond to our previous findings that SVR models work usually better than the MLP based ones.

The dynamic nature of evolutionary algorithm causes the fact that during the run of the evolution different models are the best in each of the metric. This encourages the use of automated model selectors or even meta-learning techniques, which could recommend suitable model to use.

The mentioned meta-learning recommender is one of our future steps. We are currently working on a system [17], which should be able to recommend suitable types of models for the tasks at hand and thus should remove the manual selection of the set of models we had to make in this work.

References

[1] A. Diaz-Manriquez, G. Toscano-Pulido, and W. Gomez-Flores, "On the selection of surrogate models in evolutionary optimization algorithms," in *Evolutionary Computation (CEC), 2011 IEEE Congress on*, pp. 2155 –2162, June 2011.

[2] A. Auger, J. Bader, D. Brockhoff, and E. Zitzler, "Theory of the Hypervolume Indicator: Optimal μ-Distributions and the Choice of the Reference Point," in *Foundations of Genetic Algorithms (FOGA 2009)*, 2009. workshop version.

[3] K. Bringmann and T. Friedrich, "Approximating the volume of unions and intersections of high-dimensional geometric objects," *CoRR*, vol. abs/0809.0835, 2008.

[4] K. Deb, S. Agrawal, A. Pratap, and T. Meyarivan, "A fast elitist non-dominated sorting genetic algorithm for multi-objective optimisation: NSGA-II," in *PPSN* (M. Schoenauer, K. Deb, G. Rudolph, X. Yao, E. Lutton, J. J. M. Guervós, and H.-P. Schwefel, eds.), vol. 1917 of *Lecture Notes in Computer Science*, pp. 849–858, Springer, 2000.

[5] H. Ishibuchi, N. Tsukamoto, and Y. Nojima, "Evolutionary many-objective optimization: A short review," in *Proceedings of 2008 IEEE Congress on Evolutionary Computation*, pp. 2424–2431, 2008.

[6] E. Zitzler and S. Künzli, "Indicator-Based Selection in Multiobjective Search," in *Conference on Parallel Problem Solving from Nature (PPSN VIII)* (X. Yao *et al.*, eds.), vol. 3242 of *LNCS*, pp. 832–842, Springer, 2004.

[7] I. Voutchkov and A. Keane, "Multiobjective optimization using surrogates," in *Presented on Adaptive Computing in Design and Manufacture (ACDM 06)*, 2006.

[8] I. Loshchilov, M. Schoenauer, and M. Sebag, "A mono surrogate for multiobjective optimization," in *GECCO* (M. Pelikan and J. Branke, eds.), pp. 471–478, ACM, 2010.

[9] Q. Zhang and H. Li, "Moea/d: A multiobjective evolutionary algorithm based on decomposition," *Evolutionary Computation, IEEE Transactions on*, vol. 11, pp. 712 –731, Dec. 2007.

[10] Q. Zhang, W. Liu, E. Tsang, and B. Virginas, "Expensive multiobjective optimization by moea/d with gaussian process model," *Evolutionary Computation, IEEE Transactions on*, vol. 14, pp. 456 –474, June 2010.

[11] D. Lim, Y. Jin, Y.-S. Ong, and B. Sendhoff, "Generalizing surrogate-assisted evolutionary computation," *Trans. Evol. Comp*, vol. 14, pp. 329–355, June 2010.

[12] M. Pilát and R. Neruda, "A surrogate multiobjective evolutionary strategy with local search and pre-selection," in *GECCO (Companion)* (T. Soule and J. H. Moore, eds.), pp. 633–634, ACM, 2012.

[13] M. Pilát and R. Neruda, "ASM-MOMA: Multiobjective memetic algorithm with aggregate surrogate model," in *Proceedings of the IEEE Congress on Evolutionary Computation, CEC 2011*, pp. 1202–1208, IEEE, 2011.

[14] M. Pilát and R. Neruda, "LAMM-MMA: Multiobjective memetic algorithm with local aggregate meta-model," in *GECCO (Companion)* (N. Krasnogor and P. L. Lanzi, eds.), pp. 79–80, ACM, 2011.

[15] E. Zitzler and L. Thiele, "Multiobjective Optimization Using Evolutionary Algorithms - A Comparative Case Study," in *Conference on Parallel Problem Solving from Nature (PPSN V)*, (Amsterdam), pp. 292–301, 1998.

[16] M. Pilát and R. Neruda, "Meta-learning and model selection in multiobjective evolutionary algorithms," *Machine Learning and Applications, Fourth International Conference on*, 2012. submitted.

[17] O. Kazík, K. Pešková, M. Pilát, and R. Neruda, "Implementation of parameter space search for meta learning in a data-mining multi-agent system," *Machine Learning and Applications, Fourth International Conference on*, vol. 2, pp. 366–369, 2011.

Keystroke Dynamics for Authentication in Biomedicine

Post-Graduate Student:
ING. ANNA SCHLENKER
Institute of Computer Science of the ASCR, v. v. i.
Pod Vodárenskou věží 2
182 07 Prague 8, CZ
schlenker.anna@gmail.com

Supervisor:
ING. MILAN ŠÁREK, CSC.
CESNET
Zikova 4
160 00 Prague 6, CZ
ms@cesnet.cz

Field of Study:
Biomedical Informatics

This work has been supported by "Projects of Large Infrastructure for Research, Development, and Innovations (LM2010005)".

Abstract

This paper analyzes current state of use of behavioral biometrics in authentication. It provides a brief definition of identification and authentication and biometric characteristics. The main part of the work deals with keystroke dynamics, its advantages and disadvantages and applications in biomedicine. Keystroke dynamics could be an interesting behavioral biometric characteristic for use in computer security not being widely used so far. The result of the work will be a new set of methods, which allows multi-factor authentication in the most comfortable and cheaper way.

1. Introduction

When choosing a security strategy, it is interesting to realize the principles of methods, which accompanies us for the whole existence of human society.

On the one hand, we can think of methods that are directly associated with human physiognomy. This corresponds to the initial recognition of persons by body, face, eyes or voice. It was a system that allowed the detection of people in a relatively narrow group, where everyone knows each other. This method obviously has its weaknesses, for example fake wigs and beards or double. When compared only one physiological character, the mistake may occur in simple characters such as face shape. In the case of scanning more than one character or complex characters (iris or retina), the processing may be slow and bothering users.

On the other hand, we can use some external attributes, whether it is formal clothing (uniforms), seal rings or passwords. This system has one major weakness that external attribute may by stolen by unauthorized person. And it is no matter whether it is a seal ring or token.

Only with multi-factor authentication we can eliminate unauthorized access. It can be for example combination of anatomical or behavioral features with external attribute or password.

2. Identification and Authentication

In biomedicine there is a need to protect informations and data. There are two necessary conditions to assure that only authorised person can access or modify the data [2]:

1. identification and
2. personal authentication,

which both together assure the control of the access to the information.

The process of *identification* establishes, who the person is. It happens during the initial login to the system, while the *authentication* confirms or denies the personal identity. It also demands the same proof of identity to obtain the certainty that the person is really who is affirming to be [2].

Basically, there are three ways in which person can be authenticated to the system [7, 9]:

1. The first method of authentication is based on something that the person knows, e.g. password or Personal Identification Number (PIN), called a *knowledge factor*.
2. The second method of authentication is based on something that the person has, e.g., a magnetic strip card or a secret key stored on a smart card, called a *possession factor*.

3. The third method of authentication is based on that the person is, such as a measurable biological or behavioural characteristic, that reliably distinguishes one person from another and that can be used to verify or recognize the claimed identity of the person, called a *biometric factor*.

Security measures which fall under first two categories are inadequate because possession or knowledge may be compromised without discovery – the information or article may be extorted from its rightful owner. Increasingly, attention is shifting to positive identification by biometric techniques that encompass the third class of identification (i.e., biometrics) as a solution for more foolproof methods of identification. For the foreseeable future, these biometric solutions will not eliminate the need for I.D. cards, passwords and PINs. Rather, the use of biometric technologies will provide a significantly higher level of identification and accountability than passwords and cards alone, especially in situations where security is paramount [9].

3. Biometric Characteristics

Biometrics, the physical traits and behavioral characteristics that make each of us unique, are a natural choice for identity verification. Biometrics are excellent candidates for identity verification because unlike keys or passwords, biometrics cannot be lost, stolen, or overheard, and in the absence of physical damage they offer a potentially foolproof way of determining someone's identity. Physiological (i.e., static) characteristics, such as fingerprints, are good candidates for verification because they are unique across a large section of the population [9].

Indispensable to all biometric systems is that they recognize a living person (see [10]) and encompass both physiological and behavioral characteristics. Physiological characteristics such as fingerprints are relatively stable physical features that are unalterable without causing trauma to the individual (see [10]). Behavioral traits, on the other hand, have some physiological basis, but also reflect a person's psychological makeup. Unique behavioral characteristics such as the pitch and amplitude in our voice, the way we sign our names, and even the way we type, form the basis of non-static biometric systems [9].

Biometric technologies are defined as "automated methods of verifying or recognizing the identity of a living person based on a physiological or behavioral characteristic"[8]. Biometric technologies are gaining popularity because when used in conjunction with traditional methods for authentication they provide an extra level of security.

3.1. Anatomical-Physiological Biometric Characteristics

Some examples of identifying biometric features being used for identification based systems include fingerprints, palm prints, hand geometry, blood vessel patterns in the hand, thermal patterns in the face, patterns in the iris or retina (see [10]). Today, a few devices based on these biometric techniques are commercially available. However, some of the techniques being deployed are easy to fool, while others like iris pattern recognition, are too expensive and invasive [10].

3.2. Behavioral Biometric Characteristics

In contrast, behavioral biometrics can be cheaper and easier to use. This group can include signature dynamics, voice verification and mouse or keystroke dynamics.

Mouse dynamics is a measurement of distance, speed and angle during the work with it.

Keystroke dynamics is the duration of each key-press and the time between keystrokes.

4. Keystroke Dynamics

Keystroke dynamics is the process of analyzing the way a user types at a terminal by monitoring the keyboard inputs thousands of times per second in an attempt to identify users based on habitual typing rhythm patterns [9]. It has already been shown that keystroke rhythm is a good sign of identity [6].

Moreover, unlike other biometric systems which may be expensive to implement, keystroke dynamics is almost free – the only hardware required is the keyboard [9, 5].

The application of keystroke rhythm to computer access security is relatively new. There has been some sporadic work done in this area. Joyce and Gupta [6] present a comprehensive literature review of work related to keystroke dynamics prior to 1990. The brief summary of these efforts and examination of the research, that has been undertaken since then, can be found in [9].

Keystroke verification techniques can be classified as either *static* or *continuous* [9].

- *Static verification* approaches analyze keystroke verification characteristics only at specific times,

for example, during the login sequence. Static approaches provide more robust user verification than simple passwords, but do not provide continuous security – they can not detect a substitution of the user after the initial verification.

- *Continuous verification*, on the contrary, monitors the user's typing behavior throughout the course of the interaction.

Keystroke dynamics allows so-called continuous (dynamic) verification, which is based on the use of keyboard as a medium of continuous interaction between user and computer [1]. This offers a possibility of continuous control over the whole time the computer is being used. This method is useful in situations when there is a risk of leaving a computer without control for a while [3].

Some features can be extracted of the keystroke rhythm as [2, 10]:

- the time that a key is pressed (keystroke duration),
- the time of pressing individual keys (keystroke latency),
- speed of the keystroke,
- frequency of errors,
- style of writing capital letters,
- placement of the fingers and
- pressure that the person applies when pressing a key (pressure keystroke).

This latter type requires a special keyboard that allows the force of the push to be measured. All other methods can be evaluated by a special program without any modification of hardware [9, 5].

The history of keystroke dynamics can be found in [9, 6] or in [2].

4.1. Advantages and Disadvantages of this Method

Advantages of technology [11]:

1. The ultimate goal is ability to continually checking the identity of a person as they type at a keyboard [9, 1].
2. Neither enrolment nor verification affect the regular work flow because the user would be typing needed text anyway. Easy to use for example with login and password during logon process.
3. Unlike other biometrics system, keystroke dynamics is almost free. The only hardware required is the keyboard [9, 5].
4. Time to training of users is minimal and ease of use is very high.
5. Public acceptability is very high. There are no prejudices such in case of criminal pattern in fingerprint verification or discomfort such as retina pattern scanning [10].
6. Keystroke dynamics is ideal also for network users.

Disadvantages of technology [11]:

1. Keystroke dynamics are non-static biometrics same as for example voice. This can change quite fast during time, also one-hand typing (due to injury), etc. can influent typing rhythm [9].
2. Low accuracy – keystroke dynamics is one less unique biometrics.
3. Small commercial widespread of technology.

5. Applications in Biomedicine

Keystroke dynamics can be used very well in cooperation with other authentication methods, especially with login and password (structured text), which gain good security results [11]. Now only one company, Net Nanny, works on commercial release of their product BioPassword [4].

There are many potential areas for this technology, especially for its low cost and feature of continuous checking. Limitations are mainly non-consist typists [11].

Monrose [9] also believes that keystroke dynamics can be theoretical used as possible attack to PGP[1], because random seed collected during key generation is calculated from user's typing. This can be weakness, if users typing characteristics are known [11].

Monrose [9] also reports, that there can be some differences between left-handed and right-handed users, but he has only small part of left-handed users in testing group to give some useful results [11].

[1] Pretty Good Privacy (PGP) is a computer program that provides cryptographic privacy and authentication. PGP is often used for signing, encrypting and decrypting electronic mails (e-mails) to increase the security of e-mail communications (see [12]).

Alternatively, dynamic or continuous monitoring of the interaction of users while accessing highly restricted documents or executing tasks in environments where the user must be ”alert”at all times (for example air traffic control), is a ideal scenario for the application of a keystroke authentication system. Keystroke dynamics may be used to detect uncharacteristic typing rhythm (brought on by drowsiness, fatigue etc.) in the user and notify third parties [9].

6. Conclusion

For centuries the handwritten signature is maintained as one of the important identification data. This is a unique expression of human brain. The signature is formed already in school and influenced further by personality and health of individual.

We have to accept that a new generation of students is gradually replacing handwriting by typing on keyboard. So it is appropriate to deal with this new way of human signing.

The purpose of this paper is to concentrate the available information about this new phenomenon. We can assume that typing has its own specifics, which can be in use similar to written text.

References

[1] F. Bergadano, D. Gunett, and C. Picardi, “ User authentication through Keystroke Dynamics:”ACM Transactions on Information and System Security. 5(4):367–397, 2002.

[2] G.C. Boechat, J.C. Ferreira, and E.C.B. Carvalho, “Using the Keystrokes Dynamic for Systems of Personal Security”. Proceedings Of World Academy Of Science, Engineering And Technology. 24(18):61–66, 2006.

[3] D. Gunett and C. Pikardi, “Keystroke analysis of free text”. ACM Transactions on Information and System Security. 8(3):312–347, 2005.

[4] Identity Assurance as a Service: AdmitOne Security [Internet] 2010 [cited 2012 Aug 4] Available from: http://www.biopassword.com/

[5] J. Ilonen, “Keystroke Dynamics”. Advanced Topics in Information Processing. Lappeenranta University of Technology. [Internet] 2003 [cited 2011 Aug 22]. Available from: http://www2.it.lut.fi /kurssit/03-04/010970000/seminars/Ilonen.pdf

[6] R. Joyce and G. Gupta, “Identity authorization based on keystroke latencies”. Communications of the ACM. 33(2):168–176, 1990.

[7] S.M. Matyas and J. Stapleton, “A Biometric Standard for Information Management and Security”. Computers & Security.19(2):428–441, 2000.

[8] B. Miller, “Vital sings of identity”. IEEE Spectrum. 31(2):20–30, 1994.

[9] F. Monrose and D. Rubin, “Keystroke dynamics as a biometric for authentication”. Future Generation Computer Systems. 16(4):351–359, 2002.

[10] A. Schlenker and M. Sarek, “Biometric Methods for Applications in Biomedicine”. EJBI. 7(1):37–43, 2011.

[11] P. Svenda, Keystroke Dynamics. [Internet] 2001. [cited 2012 Jul 28] Available from: http://www.svenda.com /petr /docs /KeystrokeDynamics2001.pdf

[12] P. Zimmermann, “PGP Source Code and Internals”. MIT Press; 1995.

Incorporating Population Structure into Individual Identification Process

Post-Graduate Student:
MGR. DALIBOR SLOVÁK
Department of Medical Informatics
Instutite of Computer Science of the ASCR, v. v. i.
Pod Vodárenskou věží 2
182 07 Prague 8, CZ
slovak@cs.cas.cz

Supervisor:
PROF. RNDR. JANA ZVÁROVÁ, DRSC.
Department of Medical Informatics
Instutite of Computer Science of the ASCR, v. v. i.
Pod Vodárenskou věží 2
182 07 Prague 8, CZ
zvarova@cs.cas.cz

Field of Study:
Biomedical Informatics

This work was presented at the conference Forensica 2012 and was submitted to Forensic Science International: Genetics.

Abstract

Various stochastic models are used in forensic practice more and more often. Balding and Nichols [1] proposed a method how to take the population structure into account using likelihood ratio approach which is currently predominant in assessing weight of evidence against suspect. We elaborated a detailed mathematical derivation of the Balding and Nichols method. It turns out that the inclusion of the population structure in the original work leads to systematic bias. To correct it, we introduce a new parameter which expresses the proportion of the subpopulation in the general population. Using the Dirichlet distribution and the method of moments, we derived generalized sample formula which exploits this parameter of the subpopulation proportion. Calculations show that our new method decreases the assumed number of homozygous genotypes and increases number of heterozygous genotypes in the subpopulation compared to standard methods. For certain values of parameters in our model, the new method yields a very strong improvement over standard methods.

References

[1] D.J. Balding and R.A. Nichols, "DNA profile match probability calculation: how to allow for population stratification, relatedness, database selection and single bands", *Forensic Science International*, vol. 64, pp. 125–140, 1994.

Comparison of CPU and CUDA Implementation of Matrix Multiplication

Post-Graduate Student:
ING. VLADIMÍR ŠPANIHEL

Department of Mathematics
Faculty of Nuclear Science and Physical Engineering
Czech Technical University
Trojanova 13

120 00 Prague 2, CZ

vladimir.spanihel@seznam.cz

Supervisor:
ING. FRANTIŠEK HAKL, CSC.

Institute of Computer Science of the ASCR, v. v. i.
Pod Vodárenskou věží 2

182 07 Prague 8, CZ

hakl@cs.cas.cz

Field of Study:
Mathematical Engineering

This work has been supported by the grant No. LG12020 of the Czech Ministry of Education, Youth and Sport.

Abstract

This paper deals with a comparison of different kinds of matrix-matrix multiplication. Two main approaches are investigate: nonparallel implementation on CPU vs. massively parallel implementation on GPU using NVIDIA CUDA architecture.

On CPU a naive algorithm and a function from scientific library GSL (GNU Scientific Library) are considered against three algorithms on GPU, namely a simple kernel not using shared memory, a kernel using shared memory, and a function from library CUBLAS (CUDA Basic Linear Algebra Subroutines).

It is supposed that the function from CUBLAS will have best performace.

1. Introduction

Our research is targeted on optimization of the NNSU (neural network with switching units) architecture. This network has a highly sophisticated structure, and it is used for high-dimensional physical data separation. Its structure could be simply descibed as a network of networks (each node of the global network represents another self standing network). Detailed description of this type of artificial neural network can be found in [1]. There were two botlenecks in the network learning on different levels.

On the first level, there is a search for the global network architecture using a parallel implementation of a genetic optimization. This learning algorithm has been succesfuly implemented by Jedek [2]. On the second level, there is a search for the optimal architecture of each subnetwork. Nowdays, the nonparallel implementation of linear least squares method `dgels`, used for subnetworks learning, causes the biggest loss of performance. Therefore, design and implementation of least squares method on GPU is the first step of our work. Detailed description of `dgels` function is presented in LAPACK (Linear Algebra PACKage) users' guide [3].

During linear least squares method a system of linear equations has to be solved. For this purpose, an easily parallelized algorithm for solving such system of equations is needed. The QR factorization via the Householder transformations (see [4]) has been chosen. This algorithm requires many matrix-matrix multiplications. As mentioned earlier, input data are high-dimensional; concretely, its dimension is about 120000×150. In this case, the most time consuming operation is matrix multiplication, and this paper will compare the time necessary to compute a product of two matrices of various dimensions depending on CPU (Central Processing Unit) or GPU use. The main purpose of this paper is to determine if computation of the matrix product will be significantly faster on multicore GPU than on CPU.

2. CPU Matrix Multiplication

There are many options for programming such a function. In this section two nonparallel types of matrix multiplication algorithm on CPU will be introduced: at first, a naive algorithm and the second, the optimized function `cblas_dgemm` from GSL (GNU Scientific Library [5]).

2.1. Naive Algorithm

The first method, how to find the result of matrix multiplication, is to write own function. The naive algori-

thm is based on the common matrix multiplication algorithm. It does absolutely the same as any student who solves this type of task. That means, the algorithm computes each element of the resulting matrix according to the well known relation (1).

$$C_{i,j} = \sum_{k=0}^{n-1} A_{i,k} \cdot B_{k,j}. \quad (1)$$

Straightforward C++ function may look like this:

```
void multiplyMatrixCPU(double *aM, double *bM,
    double *cM, int rowA, int rowB, int colA,
    int colB) {
  for (int i = 0; i < rowA; i++) {
    for (int j = 0; j < colB; j++) {
      int idx = j * rowA + i;
      for (int h = 0; h < colA; h++) {
        int idxA = h * rowA + i;
        int idxB = j * rowB + h;
        cM[idx] += aM[idxA] * bM[idxB];
      }
    }
  }
}
```

The input parameters `rowA`, `colA`, `rowB`, and `colB` specify size of matrices `aM` and `bM`. The first two loops specify indices in the resulting matrix, and the third loop computes the value of its appropriate element. Due to this three for loops, the complexity of naive algorithm is $\mathcal{O}(rowA \cdot colB \cdot colA)$.

The next part presents a much better way to compute a product of two matrices.

2.2. CBLAS Implementation

The second option to get easily the product can be made by using an optimized library function. Concretely, the function `cblas_dgemm` from GSL library was used as mentioned above. In comparison with the naive algorithm described above, this function is more sophisticated. This fact can be seen from its header presented in the reference manual [5].

```
void cblas_dgemm (
  const enum CBLAS_ORDER Order,
  const enum CBLAS_TRANSPOSE TransA,
  const enum CBLAS_TRANSPOSE TransB,
  const int M,
  const int N,
  const int K,
  const double alpha,
  const double *A,
  const int lda,
  const double *B,
  const int ldb,
  const double beta,
  double *C,
  const int ldc
);
```

The function `cblas_dgemm` expects 14 arguments as input, and has been designed to be able to compute the result of the folowing form

$$C = \alpha AB + \beta C, \quad (2)$$

where A, B, and C are matrices of dimensions $M \times K$, $K \times N$, and $M \times N$. The coefficients α and β are constants. Since this paper deals with a common matrix product $C = AB$, the parameters `alpha` and `beta` should be set to 1.0 and 0.0 respectively. The function returns its result in the matrix C.

The next difference between this fucntion and the one mentioned in the previous section is in the three first parameters of `cblas_dgemm` function. The first one specifies the data ordering. That means if the matrix is serialized into array row-by-row (row-major order) or column-by-column (column-major order). The next two parameters denote if the matrix A or B should be transposed before the multiplication is performed. Please note that the naive algorithm does not take these settings into account. So that it does not contain additional conditional statements which are necessery in the `cblas_dgemm` function.

The remaining three parameters named like `ld*` denote so called leading dimension i.e. the row count of each input matrix.

The following part is concerned with the paralel GPU implementation of matrix multiplication which was not mentioned yet.

3. GPU Implementation

Similarly to the CPU methods, there are many variants how to implement matrix multiplication on GPU. In this part of text the thread and memory hierarchy of CUDA architecture will be disscussed. Furthermore, at the end of this section, three different approaches to obtain the product of two matrices will be shown. More information about NVIDIA CUDA architecture can be found in [6, 7].

Massively parallel application based on NVIDIA CUDA architecture can be written by everyone who owns a graphics card containing a GPU supporting CUDA. The list of all such cards is provided on CUDA Zone websites [8].

3.1. Thread and Memory Hierarchy

Besides relatively low price, a large number of cores (also referred to as streaming multiprocessor) is the gre-

atest advantage of GPUs. Whereas, it is possible to run concurrently about tens of threads on CPU, on GPU it is in hundreds of thousands. As a result of this fact, massively parallel algorithms may be written on GPU, but programmers require at least a minimal knowledge about the CUDA architecture which depicts Figure 1.

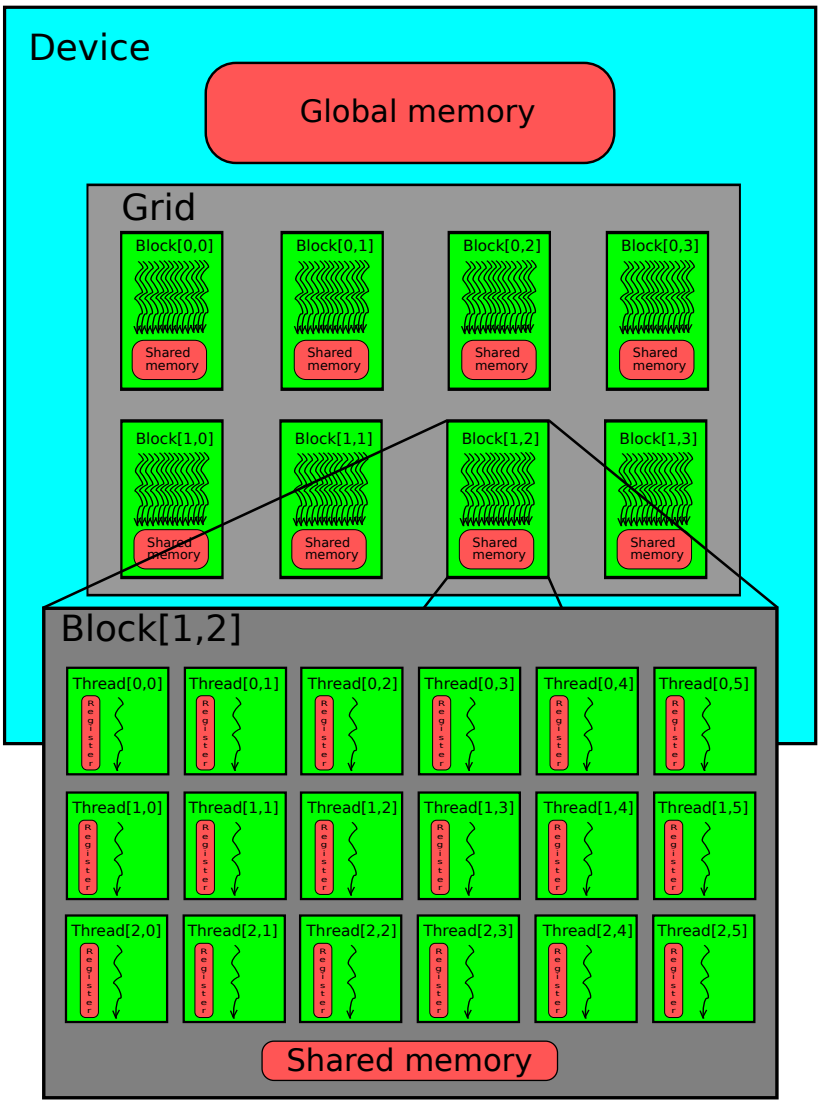


Figure 1: Architecture of a grid is divided into blocks and threads.

Functions performed on GPU are called *kernels*; kernel is invoked from a CPU code, but the body of kernel runs concurrently on GPU threads. The threads are grouped into thread blocks and the blocks are held within a grid (see Figure 1). This structure has to be specified in a kernel call. For this purpose NVIDIA introduced a special notation; variables describing the structure of threads are inserted in triple angle brakets between the kernel identifier and the list of parameters. Then a kernel call could look like this `kernel<<<dimGrid, dimBlock>>>(...);` where variables `dimGrid` and `dimBlock` of type `int` or `dim3` describe the count and layout of threads. Moreover, the grid and blocks may be multidimensional (up to three dimensions). Thus, indexing in matrices is very simple. Even if each thread performs the same code, threads have its own unique id within one block as well as a block within the grid, so build-in variables like `threadIdx.x` or `blockIdx.y` are used for their identification.

Computing on GPU has also a disadvantage, i.e. both CPU referred to as *host* and GPU referred to as *device* have its own memory, and the host cannot access directly

the device memory (also called global memory) and vice versa. Therefore the device memory has to be allocated and all necessary data copied into it before the matrix multiplication starts on GPU. After all, the results must be copied back from the device to the host memory. The memory management makes the computation time longer, and this delay should be included in the computation duration on GPU.

In Figure 1 can be seen three most important memory types of NVIDIA graphics cards: global memory, shared memory, and registers. The global memory is the largest, but also the slowes one. Its size is measured in gigabytes, its content persists while the application is running, and all threads have access to all data stored in it. The second mentioned type is shared memory which should be much faster than the global memory. The lifetime of shared memory is the same as the lifefime of thread block. Therefore each thread block has its own shared memory accessible by only threads inside the block. A size of shared memory is stated in kilobytes. The fastest and smallest memory id register. One register is connected only with one thread. Inside registers are stored for example the build-in variables mentioned above.

The folowing parts will briefly describe three various implementations of matrix multiplication on GPU.

3.2. Simple CUDA Implementation

At first a very straightforward kernel implementation can be seen in next few rows.

```
__global__ void multiplyMatrixGPU(double *aM,
    double *bM, double *cM, int rowA, int rowB,
    int colA, int colB) {
  int idRow = blockIdx.x * blockDim.x
              + threadIdx.x;
  int idCol = blockIdx.y * blockDim.y
              + threadIdx.y;

  // Check if index is in bound of matrix C
  if ((idRow >= rowA) || (idCol >= colB)) {
    return;
  }

  // Perform computation
  int idx = idCol * rowA + idRow;
  float r = 0.0;
  for (int i = 0; i < colA; i++) {
    int idA = i * rowA + idRow;
    int idB = idCol * rowB + i;
    r += aM[idA] * bM[idB];
  }
  cM[idx] = r;
}
```

Actually, this code is very similar to the naive algorithm; however, the first two for loops were replaced by concurrency. This detail has a great effect on computation efficiency, as shown in the next section (Table 1). Please note that this algorithm does not take advantage of fast shared memory.

3.3. CUDA Implementation with Shared Memory

The original version of square matrix multiplication algorithm with shared memory, mentioned here, comes from [9] and assumes row major matrices. It was modified to accept column major matrices, and the source code is here.

```
__global__ void matrixMulSh( double* C,
    double* A, double* B, int hA, int hB) {

  // Block index
  int bx = blockIdx.x;
  int by = blockIdx.y;

  // Thread index
  int tx = threadIdx.x;
  int ty = threadIdx.y;

  // Index of the first sub-matrix of A
  // processed by the block
  int aBegin = BLOCK_SIZE * bx;

  // Step size used to iterate through the
  // sub-matrices of A
  int aStep  = BLOCK_SIZE * hA;

  // Index of the first sub-matrix of B
  // processed by the block
  int bBegin = hB * BLOCK_SIZE * by;

  // Index of the last sub-matrix of B
  // processed by the block
  int bEnd   = bBegin + hB -1;

  // Step size used to iterate through the
  // sub-matrices of B
  int bStep  = BLOCK_SIZE;

  double Csub = 0.0;

  // Loop over all the sub-matrices of A and B
  // required to compute the block sub-matrix
  for (int a = aBegin, b = bBegin;
           b <= bEnd;
           a += aStep, b += bStep) {

    // Declaration of the shared memory array As
    // used to store the sub-matrix of A
    __shared__ double As[BLOCK_SIZE][BLOCK_SIZE];

    // Declaration of the shared memory array Bs
    // used to store the sub-matrix of B
    __shared__ double Bs[BLOCK_SIZE][BLOCK_SIZE];

    // Load the matrices from global memory
    // to shared memory; each thread loads
    // one element of each matrix
    As[tx][ty] = A[a + hA * ty + tx];
    Bs[tx][ty] = B[b + hB * ty + tx];

    // Synchronize to make sure the matrices
    // are loaded
    __syncthreads();

    // Multiply the two matrices together;
    // each thread computes one element
    // of the block sub-matrix
    for (int k = 0; k < BLOCK_SIZE; ++k)
      Csub += As[tx][k] * Bs[k][ty];

    // Synchronize to make sure that
    // the preceding computation is done
    // before loading two new sub-matrices
    // of A and B in the next iteration
    __syncthreads();
  }
```

```
  // Write the block sub-matrix to device
  // memory; each thread writes one element
  int c = hB * BLOCK_SIZE*by + BLOCK_SIZE*bx;
  C[c + hB * ty + tx] = Csub;
}
```

The algorithm is more complex than the first one. For detailed description of this code, see [9].

Again, it could be interesting to compare the throughput of the two described kernels with a library function.

3.4. CUBLAS Implementation

The CUBLAS library is a CUDA alternative to BLAS (Basic Linear Algebra Subroutines). This library contains function called `cublasDgemm`. Its name, as well as its parameter set , is very similar to the `cblas_dgemm` function from GSL. The header of this function can be seen in [10].

4. Results

A simple application was written. It returns an average computation time in milliseconds of each method and matrix dimension. All obtained values are based on fifty observations and are summarized in Table 1 and Figure 2. The GPU results include all memory management.

n	4	8	16	32	64	128	256	512	1024
naive	0	0	0.08	0.12	1.56	12.54	130.26	1850.04	27589.84
cblas_dgemm	0	0	0.02	0.06	0.26	2.14	16.56	170.42	2061.86
GPU	0.14	0.06	0.08	0.16	0.46	2.54	19.34	143.50	1138.90
GPU_shared	0.18	0.06	0.22	0.12	0.6	4.04	30.62	239.70	1903.36
cublasDgemm	0.2	0.14	0.2	0.16	0.34	0.8	3.8	24.68	182.32

Table 1: Time in milliseconds needed to compute product of two $n \times n$ matrices for methods: the naive algorithm, cblas_dgemm function, simple CUDA kernel, CUDA kernel with shared memory, and cublasDgemm function.

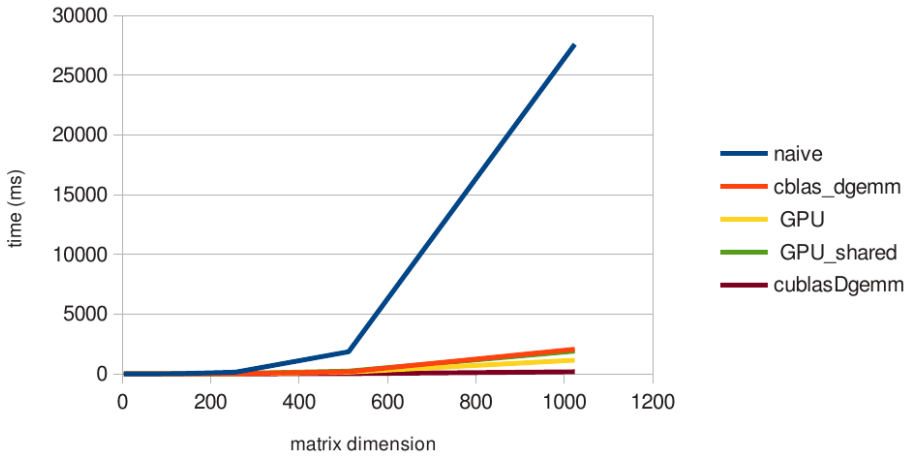


Figure 2: Relation between each method average computation time and matrix dimension.

In Figure 3 the same relations as in Figure 2 are shown except the curve for naive algorithm. Thus, it provides detailed view to throughput of all efficient method.

4.1. System configuration

The application was performed on laptop Lenovo T420 with dedicated NVIDIA NVS 4200M graphics card:

Operating system:	Archlinux (3.4.7-1) 64bit
CPU:	Intel Core i5 2.5 GHz
RAM:	4 GB
GPU:	1 GB
Cuda compilation tools:	release 4.2, V0.2.1221
C++ compiler:	GCC 4.7.1

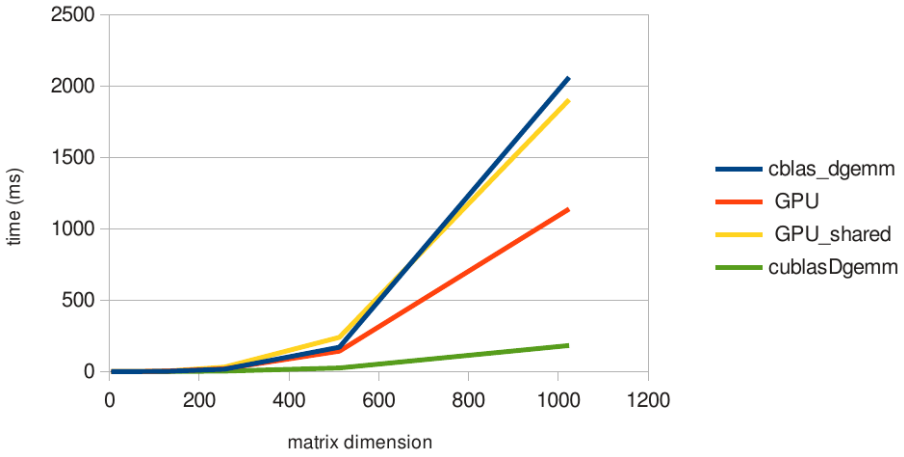


Figure 3: Detailed comparison of efficient methods.

5. Discussion

Here is an interesting question; What is the optimization of the library function `cblas_dgemm`? Thanks to openness of GSL library, source code of this method was inspected. The main difference between library function and naive algorithm is work with memory. Whereas in naive algorithm only one index is changed continuously and remaining two are jumping, the implementation of `cblas_dgemm` always changes two indices continuously and only one index is jumping. It looks like a triviality, but the difference in throughput is huge.

6. Conclusion

The performance of the naive algorithm is very poor, whereas the optimized `cblas_dgemm` function is comparable with the GPU kernel, even though for matrices 1024×1024 the result time of kernel not using shared memory is about two times shorter than on CPU.

The results of kernel with shared memory are comparable with `cblas_dgemm` function. Measured times should be significantly better for matrices of higher dimensions.

With regard to the results, the parallel matrix multiplication on GPU from CUBLAS library will bring the largest increase in performance of the linear least squares method, considering the matrix dimension 120000×150 as mentioned at the beginning of this paper.

References

[1] R. Kalous, *Optimization of Neural Networks Architectures*. PhD thesis, CTU Prague, 2009.

[2] V. Jedek, *Paralelizace distribuovaného výpočtu geneticky optimalizovaných neuronových sítí*. Master's degree thesis, CTU Prague, 2008.

[3] E. Anderson, Z. Bai, et al., *LAPACK Users' Guide Third Edition*. http://www.netlib.org/lapack/lug/ 1999-08-22, [cit. 2012–04–27]

[4] P. Tichý, *3. Ortogonální transformace a QR rozklady*. home.zcu.cz/ ptichy/kma/download/texts/ 03.pdf 2011-10-12, [cit. 2012–04–27].

[5] M. Galassi et al., *GNU Scientific Library Reference Manual (3rd ed.)*. http://www.gnu.org/software/gsl/

[6] R. Farber, *CUDA Application Design and Development*. Waltham USA: Elsevier 2011.

[7] NVIDIA, *NVIDIA CUDA C Programming Guide*. http://developer.download.nvidia.com/compute/ DevZone/docs/html/C/doc/CUDA_C_ Programming _Guide.pdf [cit. 2012–08–06]

[8] CUDA Zone, *CUDA GPUs*. http://developer. nvidia.com/cuda-gpus [cit. 2012–04–27]

[9] Confessions of a Speed Junkie, *Matrix Multiplication 3 (CUDA)*. http://gpgpu-computing4. blogspot.cz/2009/09/ matrix-multiplication-3.html [cit. 2012–08–06]

[10] NVIDIA, *CUDA Toolkit 4.2, CUBLAS Library*. http://developer.download.nvidia.com/compute/ DevZone/docs/html/CUDALibraries/doc/ CUBLAS_Library.pdf [cit. 2012–08–06]

Russian 'Facebook' Problem: Public Announcements and Privacy

Post-Graduate Student:
MGR. PETR ŠVARNÝ

Department of Information Technologies
University of Economics, Prague
Nám. W. Churchilla 4
130 67 Praha 3, CR

svarnypetr@gmail.com

Supervisor:
PROF. ING. VLADIMÍR MAŘÍK, DRSC.

Department of Cybernetics
Faculty of Electrical Engineering
Czech Technical University
Technická 2
166 27 Prague 6, CR

marik@labe.felk.cvut.cz

Field of Study:
Cognitive Informatics

Abstract

The paper investigates some possibilities how to generalize the so called Russian cards problem and use it to analyse private public communication. It shows different ways how this can be done, lists some possible outcomes of these generalizations and suggests a logical language to capture the properties of such generalized problems.

1. Introduction

Multi-agent systems in the framework of dynamic epistemic logic help to analyse all sorts of problems in communication and information spreading in a given system. The usual approach, however, investigates how to share and distribute information. The approach in this article goes in the opposite direction. We investigate the idea how a group of agents can communicate via public announcements without sharing too much with an adversary group. In the article, we focus on the combination of the so called Russian cards problem and some public announcement logics. We present briefly the tools, continue with a case study of scenarios, and finish with a suggestion of a logic capable of capturing the situation.[1]

2. Motivation

Our motivation is quite mundane and we do not have to imagine any elaborate games of spies or intelligence agencies. It is especially on social networks that information sharing has a public character but the user would like to limit the audience in some way. We can do so in a restrictive way, as it is done in the article of Jakob et al. [5] where a mechanism is introduced that suggests the user with whom he can share a given message. This type of mechanism, however, needs vigilance to prevent unwanted sharing and the user needs to be aware of what can be deduced from the published information. Therefore, we want see if there is a safe way of communicating in public and still maintaining the privacy of the transmitted message.

3. The Basic Idea

First we explain our goals with informal explanations. These are followed by formal definitions and theorems in the next section.

3.1. Russian Cards Problem

An important inspiration for this project is the so called Russian cards problem(RCP). The article that introduced the author to the problem comes from van Ditmarsch [2]. The problem for three players, usually called Anne, Bill, and Crow, goes as follows:

> *From a pack of seven known cards two players each draw three cards and a third player gets the remaining card. How can the players with three cards openly (publicly) inform each other about their cards, without the third player learning from any of their cards who holds it?*

[1] The visit of A. Baltag and S. Smets in Prague that took place two years ago lead in the end to the ideas and work presented in this article. Therefore it is only correct to thank them and also to thank L. Běhounek, as he came up with the idea that the framework of merges could be used to investigate some kind of secret service communication. These ideas stayed dormant until the moment when they were combined with the article of van Ditmarsch[2]. Preliminary ideas were already presented at the Prague workshop of Non-classical dynamic epistemic logics.

We use an example to explain how RCP works and introduce a possible solution for the problem.

Example 1 *Let* $012|345|6$ *be an instance of RCP, i.e. Anne has the first three cards, Bill has the next three, and Crow has the last card. In that case a good announcement of Anne is* $[K_a(012_a \vee 034_a \vee 056_a \vee 135_a \vee 246_a)]$. [2] *Bill knows Anne's cards (therefore also Crow's card), and he announces Crow's card (6). Thanks to Bill's announcement Anne also knows Bill's cards but Crow remains ignorant of their respective hands. However, Crow knows both players know their hands.*

Card games represent a great modelling tool for information sharing. Although only few people would publicly announce a card they supposedly hold (and usually these people are playing cards at that moment), card games allow us to grasp the main concepts involved in information sharing or information containing. The seven cards represent a finite discrete set of possible information bits. Public announcements (PA) concern only the cooperating players' hands. The third player is the vile attacker trying to find out more than he is meant to.

Card games are sufficient to capture the main topics that we need to investigate while studying PA. We can, however, identify some possible pitfalls when generalizing RCP:

- It uses cards instead of more elaborate information.
- It has limited options.
- It has a visible and clear distribution of cards.
- There are only three players.

We attempt to present interpretations of the card model in comparison to real life communication situations to see what the model can or cannot deliver for answers and how does it present these answers. We address all the mentioned possible pitfalls one by one.

First, it does not cause any trouble to use cards instead of propositions. The cards are soon replaced by natural numbers for their use in formal approaches (we have seven cards called 0, 1, 2 ... 6). Thus if one would have a problem with using cards, he can shift his disgust to numbers. We take the number labelling as simply a labelling of seven distinct and independent propositions.

What remains a slight problem in comparison to the real world is the number of options. We are faced with only seven possibilities while in a real communication the options are, if not limitless, numerous. These limited options would in real life suggest a context or situation where the speakers either know the topic very well or they have established a set of possible answers. Formally such limitation of options could be inspired by approaches from erotetic logics. However, it is unnecessary at this point. If the reader would like to develop this particular limitation further, Majer and Peliš [6] can be recommended.

The distribution of cards is visible and clear. In RCP all the players know that Anne and Bill have three cards each and Crow has only one card. This is also non-realistic as the distribution of information is often hidden from players in day-to-day conversational games. The second trouble is that in our card game all the cards are distinct and distributed amongst the players, whereas in real life we can have shared knowledge and we even do not have to know that we have shared knowledge. Therefore we certainly need to address the topic of information distribution and knowledge thereof.

One could argue that the number of three players is an important limitation. This should not be taken as an obstacle as we can take the three players as representatives of three groups of interest given by their goals. Every conversation would then be divided up into separate triads for furhter analysis.

Now, with some ground established on one side of the river, let us see if we can create a foundation also on the other side and start building a connecting bridge.

3.2. Logics for PA

The other side of our river of challenges is the logical shore. We rely on the article of Baltag and Smets [1] as a source for dynamic logic ideas. In brief, the presented logic describes doxastic preferences of agents on states and allows them to learn from sources of various levels of trust. These levels are learning certain information, learning from a strongly trusted source, or barely believing the announced information. Although there exist also other approaches to the topic of PA and information sharing, we use this one as a starting point.

3.3. BKontakte

The motivation why to combine the idea of a richer language and a formal (card-like) game is to model attempts of agents to secretly transfer some knowledge while paying attention to their trust towards other agents. This would be in essence very close to the en-

[2] Or in layman words "I hold 012 or 034 or 056 or 135 or 246".

[3] Bkontakte is a Russian version of Facebook. This explains how the subsection gained its name.

vironment of a social network. [3] Before we would enter the formal part of this paper, we revisit our main idea and prepare it for a precise explanation.

We are interested in the case when a group of users interacts on a social network. The group can have only three members for now - Anne, Bill, Crow. The first two agents try to communicate some information without Crow learning this information and we set the goals for our agents accordingly. The agents may communicate only via public updates. So far it was the same as RCP, but our agents also can have preferences on the states of the world. We generalize the card deal to a so called gossip deal. This deal addresses some of the issues mentioned already in the first section and presents a more general idea than the use of a deck of cards. Agents will be communicating actually about the types of information they know. Based on these we investigate what are the necessary presuppositions to allow safe public communication.

4. The Formal Approach

The logic used here will be called Privacy Modal Logic or PriMoLo. We use some parts from the original articles these then have a bibliographic reference number next to their name accompanied by an apostrophe if they were altered in any way. We first generalize the topic of card games in order to allow their thorough study and then we introduce this new logic.

4.1. All Sorts of Games

A RCP game can be described by a set of parameters. Their names are quite self-explanatory, but we leave RCP examples to assure the correct understanding.

Number of players: three - Anne, Bill, Crow.

Relation of players: Anne and Bill want to share their information. Crow listens in.

Positions: $3|3|1$ (A gets three, B gets three and C gets one, seven positions).

Cards: seven different cards.

Card deal type: $012|345|6$ (everyone gets different cards, all cards are dealt).

Knowledge: players know all the above, they only remain ignorant of the exact deal.

We see that some of the formalisms allow us to tell a great deal of information in a short statement. By changing these parameters we can differentiate many situations and even take into account some of the pitfalls from 3.1. Social networks would be on one end of the scale as the most chaotic system and RCP would be on the other as the wll documented and understood situation. Let us first note that not every change of parameters means we left the safe waters of RCP. A generalized approach to RCP is investigated in [3] and it shows some specific rules for changes of the positions parameter that leave the system as a version of RCP. Therefore we can focuse on changes of other parameters.

The following list gives some results of changes in comparison to the original RCP. [4]

1. Less cards than positions: Crow can guess cards because there are less options.

2. More cards than positions: limit the ability of Anne and Bill to communicate their cards.

3. A limited number of players greater than 3: situation can be treated as RCP when using triads.

4. Anne wants to withhold information can be achieved by introducing superfluous information.

5. Agents have various degrees of trust introduction of new operators (based on Baltag and Smets) gives a broader possibility of opinions of agents.

If we combined all the changes from this list, we would get the description of a real life social network accurate enough to support tests of all our possible protocols concerned with privacy and public announcements but at the same time too close to the complexity of a real life social network.

4.2. PriMoLo

Our logic is created as a merge between Baltag and Smets' approach and van Ditmarsch's cards problem borrowing some ideas from Hommersom et al.[4]. We generalize some of the notions and reintroduce them so that they can coexist in one system.

Definition 1 (Plausibility frames [1]') *For a set $\mathcal{A}$ a* plausibility frame *is a finite, multi-agent Kripke frame $(S, \leq_a)_{a\in\mathcal{A}}$. The set S are states and $\leq_a \subseteq S \times S$ is an accessibility relation and stands for the preferences of*

[4] A detailed explanation of these results can be found in the author's thesis[7].

the agents about the possible states S. The relation $\leq_a$ is assumed to be a reflexive, transitive, and euclidean relation.

Definition 2 (Model PriMoLo [1]) *A PriMoLo model is the structure $\boldsymbol{S} = (S, \leq_a, \pi, s_0)_{a\in\mathcal{A}}$, where we gave the frame $(S, \leq_a)_a$ a designated state $s_0 \in S$, and a valuation on states $p \in \Psi \to \pi(p) \subseteq S$.*

We can also use the language from [1], i.e. the relations $s \sim_a t$, $s <_a t$, $s \cong_a t$, and the equivalence class $s(a)$. Also the operators K_a, B_a, B_a^Q, and Ek_G.[5] There are also parts that did not come from [1]: $UPDATE_{(\dagger,\varphi,\mathcal{B})}$, $SIDE-EFFECT_{(\dagger,\varphi,\mathcal{B})}$.[6] The symbol † is replaced by any of the three possible updates known from Baltag and Smets - definitely true (!), strongly believed (⇑) and believed(↑). These operations have their origin in a different article - [4]. Although the article itself focuses on protocols for cryptographic communication, it already uses a dynamic epistemic logic. The formal definitions of these notions, for a given set of states S, a set of agents $\mathcal{A}$, the subset $\mathcal{B} \subseteq \mathcal{A}$, formulae φ, ψ, and propositional atoms $\mathcal{P}$, go as follows:

Definition 3 *Relations:*

$$\begin{aligned} s \sim_a t &\equiv s \leq_a t \vee t \leq_a s \\ s <_a t &\equiv s \leq_a t \wedge s \neq t \\ s \cong_a t &\equiv s \leq_a t \wedge t \leq_a s \\ s(a) &\equiv \{t | t \in S \wedge s \sim_a t\} \end{aligned}$$

Operators:

$$\begin{aligned} K_a\varphi &\equiv \{s \in S | s(a) \subseteq \varphi\} \\ B_a\varphi &\equiv \{s \in S | Max_{\leq_a} \subseteq \varphi\} \\ B_a^\psi\varphi &\equiv \{s \in S | Max_{\leq_a \cap \psi} \subseteq \varphi\} \\ Ek_{\mathcal{B}}\varphi &\equiv \bigcap_{a\in\mathcal{B}} K_a\varphi \end{aligned}$$

Updates:

$$\begin{aligned} !\varphi\, \boldsymbol{S} \equiv \boldsymbol{S}' : S' &= \{s \in S : s \models \varphi\} \wedge \\ &(s \leq'_a t \leftrightarrow s \leq_a t \wedge s,t \in S') \\ \Uparrow \varphi\, \boldsymbol{S} \equiv \boldsymbol{S}' : S' &= S \wedge (s \leq'_a t \leftrightarrow \\ &(s \notin \varphi_{\boldsymbol{S}} \wedge t \in s(a) \bigcap \varphi_{\boldsymbol{S}}) \vee s \leq_a t) \\ \uparrow \varphi\, \boldsymbol{S} \equiv \boldsymbol{S}' : S' &= S \wedge (s \leq'_a t \leftrightarrow \\ &(t \in max_a(s(a) \bigcap \varphi_{\boldsymbol{S}}) \vee s \leq_a t) \end{aligned}$$

$U_{(\dagger,\varphi,\mathcal{B})}\boldsymbol{S} \equiv \boldsymbol{S}'$ *where:*

- $S' = \{old(s), new(s) | s \in S\}$
- $s'_0 = new(s_0)$
- $\forall p \in \mathcal{P} : \pi'(old(u))(p) = \pi'(new(u))(p) = \pi(u)(p)$
- $\forall a \in (A)$, *is* $\leq'_a$ *minimal in* S' *such that:*

$$\begin{aligned} \leq'_a (old(s), old(t)) &\Leftrightarrow \leq_a (s,t) \\ \leq'_a (new(s), old(t)) &\Leftrightarrow a \notin \mathcal{B} \to \leq_a (s,t) \\ \leq'_a (new(s), new(t)) &\Leftrightarrow \end{aligned}$$

  - *if* $a \in \mathcal{B} \wedge \dagger =!$ *then* $\leq_a (s,t) \wedge t \in \varphi_{\boldsymbol{S}}$
  - *if* $a \in \mathcal{B} \wedge \dagger =\Uparrow$ *then* $\leq_a (s,t) \vee s \notin \varphi_{\boldsymbol{S}} \wedge t \in s(a) \cap \varphi_{\boldsymbol{S}}$
  - $a \in \mathcal{B} \wedge \dagger =\uparrow$ *then* $\leq_a (s,t) \vee t \in max_a(s(a) \cap \varphi_{\boldsymbol{S}})$

$ATOMSPLIT_{(\dagger,P,\mathcal{B})} \equiv \boldsymbol{S}'$ *where:*

- $S = S'$
- $\forall s \in S', p \in \mathcal{P}\ (\pi'(s)(p) \Leftrightarrow \pi(s)(p))$
- *if* $a \in \mathcal{B}$ *and*

$$\begin{aligned} \dagger =! &: \leq'_a (s,t) \Leftrightarrow \leq_a (s,t) \wedge s \in P_{\boldsymbol{S}} \leftrightarrow t \in P_{\boldsymbol{S}} \\ \dagger =\Uparrow &: \leq'_a (s,t) \Leftrightarrow \leq_a (s,t) \vee s \notin P_{\boldsymbol{S}} \wedge \\ &\qquad \wedge t \in s(a) \cap P_{\boldsymbol{S}} \\ \dagger =\uparrow &: \leq'_a (s,t) \Leftrightarrow \leq_a (s,t) \vee t \in max_a(s(a) \cap P_{\boldsymbol{S}}) \end{aligned}$$

- *if* $a \notin \mathcal{B}$ *then* $\leq'_a (s,t) \Leftrightarrow \leq_a (s,t)$

$S - E_{(\dagger,P,\mathcal{B},\mathcal{C})}(\boldsymbol{S}, s)$ *for* $N = SUB_{\mathcal{B}}(\boldsymbol{S}')$, $(\boldsymbol{S}', s'_0) = UNFOLD_{(\mathcal{B},\mathcal{A}/\mathcal{B})}(\boldsymbol{S}, s_0)$*:*

$$\begin{aligned} &SIDE-EFFECT_{(P,\mathcal{B},\mathcal{C})}(\boldsymbol{S}, s_0) = \\ &= (REPLACE_N(ATOMSPLIT_{(\dagger,P,\mathcal{C})}(N), \boldsymbol{S}'), s'_0) \end{aligned}$$

[5] For this article we omit Ck_G, $\Box_a$, and Sb_a.

[6] We abbreviate $UPDATE$ as U and $SIDE-EFFECT$ as $S-E$. Also please remark, that the operation $UNFOLD$ isn't defined in this article, it can be regarded as making a bisimilar model that separates states for two groups of agents. The same holds for $REPLACE$, this operation replaces a given submodel with another.

An example serves to familiarise the reader with some of these technicalities.

Example 2 *Let there a PriMoLo model* $\boldsymbol{S}$ *as in the Figure 1, i.e. agents Anne and Bill with different knowledge about the topic of* p *and* q *with the state* $(-q, -p)$ *being the designated state.*

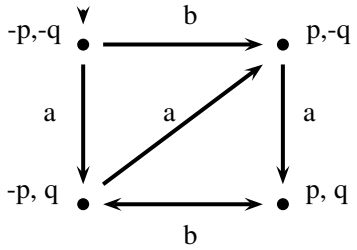


Figure 1: A PriMoLo model.

We need to keep in mind that the preference relation, expressed by arrows, is reflexive, transitive, and euclidean. [7] *It holds that every two states are comparable by* $\leq_a$*, hence every two states are in the relation* $\sim_a$*. Bill does not distinquish between* $(p, -q)$ *and* (p, q)*, in other symbols* $(p, -q) \cong_b (p, q)$*. The evaluation of operators is based on the use of relations and the designated state, in our case* $(-p, -q)$*. Bill knows that* $-q$*,* $K_b(-q)$*, because in all states that are linked by* $\sim_b$ *to* $(-p, -q)$ *it holds that* $-q$*. Bill simply does not see any other options. In this model it also holds that Anne knows that* $(K_b(-q) \vee K_b(q))$*, she doesn't know if* q *or* $-q$ *holds.*

We could use $UPDATE_{(\Uparrow, -q, \mathcal{A})}$ *to convey Bills knowledge to Anne. The formula captures that Bill shouts out that* $-q$ *holds to all the agents, but he does so in some peculiar way as the announcement is only strongly believed by the recipients. It results in the addition of arrows for Anne which point upwards on both vertical sides of the square. Hence Bill's announcement spoils Anne's linear preference* $(-p, -q) \leq_a \leq_a (-p, q) \leq_a (p, -q) \leq_a (p, q)$ *and leaves Anne uncertain whether it holds that* p *or* $-p$*.*

It is useful to distinguish between U and $S - E$ in order to be able to inform agents that another agent has learnt something without letting the agents know what was the content of the newly gained knowledge.

Example 3 *Bill announces* $!(K_b(-q) \vee K_b(q))$*. We use the example from [4]. The result of such action can be seen in Figure 3 for an* $U_{(!,p,\{b\})}\boldsymbol{S}$ *and Figure 4 for a* $S - E_{(!,K_b(-p),\{a\},\{b\})}(\boldsymbol{S}, (-p, -q))$*.*

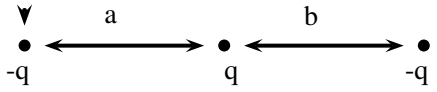


Figure 2: The original model.

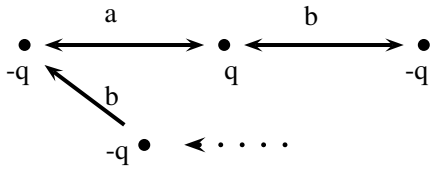


Figure 3: Model after $U_{(!,K_b(-q)\vee K_b(q),\{a\})}\mathbf{S}$.

Announcement of $K_b(-q) \vee K_b(q)$ *to Anne informs her also about* $-q$ *itself. In order to avoid this side-effect we use* $S - E$ *on the model from Figure 2.*

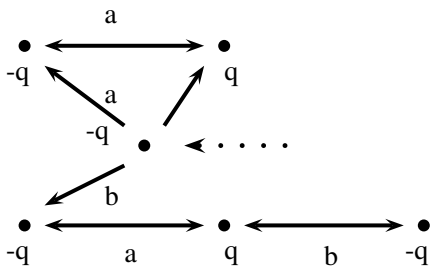


Figure 4: *After* $S - E_{(!,K_b(-q)\vee K_b(q),\{a\},\{b\})}(\boldsymbol{S}, (-p, -q))$.

Anne knows only that Bill knows whether $-q$ *or* q *holds.*

Please observe one important property of knowledge and communication in these models. We assume sincere communication and true knowledge. In other words agents do not lie, nor can they have contrafactual knowledge. [8]

Definition 4 (Gossip deal[2']) ***Gossip deal*** d *is a function:* $Q \rightarrow N$*, where* Q *is a set of gossip and* N *is the set of agents. The sign* $\#d$ *gives us a random distribution of gossip among adversaries. Two gossip deals are indistinquishable for an agent if he holds the same gossip and either there is no difference in positions or he does not know about a difference in positions.*

So far we did capture information relevant to gossip but we need to take into account also the relations between agents.

[7] Arrows resulting from these properties are omitted in the figures for clarity.
[8] For more detailed analysis see [1].

Definition 5 *Agent relation structure is a triplet $\mathcal{V} = (N, C, T)$. N is the set of agents. C is the collaboration relation between agents, a binary relation capturing if two agents cooperate. T is a trust function: $N \times N \to \dagger$.*

As all communication is public, we do not need to capture any kind of communication channels, it is enough to capture the cooperative pairs of agents. Based on these definitions we can describe the communication model itself:

Definition 6 *The state of the communication is for an interview $P = Q \times N$ and a gossip q_n the model $(D, \leq_n, V, d)$. Where q_n means that the information q is held by te agent n, $D \subseteq (Q \to N)$, $\forall q_n \in P : V_{q_n} = \{d \in D | d(q) = n\}$ and $\leq_n$ the preference relation of agent n on D.*

A special kind of state is the initial state of the interview where the agent knows only information held by him (i.e. every $q \in Q$ with the subscript of the given agent), but does not know any other agents information. At the same time, the agents have some preference on the gossip, they suppose some information to be more plausible than other. We can see that a state of communication is a whole epistemic model in the sense Baltag and Smets used it and the whole communication is then a succession of such models. We can conclude, based on definition 2 that we have PriMoLo models with $S = D$, $s_0 = d$ and valuation V based on definition 6, only $\leq_a$ remains unchanged. We need to distribute in some way information and this is given by a definition based again on the card game.

Definition 7 *For the deal $d : Q \to N$, we define $d(q_n) = q_n \Leftrightarrow d(q) = n$ a $d(q_n) = \neg q_n \Leftrightarrow d(q) \neq n$. The description of the deal d is then a conjuction of atoms and their negations: $\delta_d := \wedge_{q \in Q, n \in N} d(q_n)$. Gossip for agent n is $\delta_d^n := \wedge_{q \in Q} d(q_n)$*

We might have already thought about the interpretation of $K_a 012_b$. It would be undesirable if it meant that Anne knows that Bill has the cards 0,1,2. If it would be the case, Anne knows the information although it is not necessary that Bill holds this information. We need to assure that Anne knows this only if Bill holds the given kind of information. This is captured by so called postconditions. Ditmarsch uses postconditions tailored for RCP but we need to generalize them and for this purpose we return to $\mathcal{V} = (N, C, T)$.

Definition 8 *A set of postconditions is a set of formulae that hold in the model $(\boldsymbol{D}, d)$ if it is solved. Every agent has his own conditions depending on his relations according to $\mathcal{V} = (N, C, T)$. For any three agents a, b, c,*

$$\textit{aknowsb} \wedge_{e \in \mathcal{D}(D)} (\delta_e^b \to K_a \delta_e^b)$$
$$\textit{cignorant} \wedge_{q \in Q} \wedge_{n-a,b} (\neg Q_c \to \neg K_c \neg q_n)$$

Based on the relations of the agents, $aknowsb$ should be true if an agent knows the cards of his collaborating partner. On the other hand $cignorat$ should hold for any agent that is not a collaborative member of the given pair. In our model we do not ask for common knowledge of a succesful exchange of information as opposed to RCP. [9]

This concludes the basic formal tools that can be used to capture gossip deals and investigate possible communication protocols in generalized RCPs.

4.3. Conslusions and Possibilities

Let us shortly demonstrate what can be told with these tools and couldn't by the others. Let there be Anne, Bill, and Crow. They need to decide about an issue given by the set $\{0, \ldots, 6\}$ of propositions. These can be different kinds of information about the topic 'who let the dogs out'. The proposition labeled 0 could signify information about the person that last saw them, 1 could be about the state of their ability to jump over the fence, etc. We can also adopt another view that they present yes-no questions relevant to the topic. Although in the end it does not make a difference, the second view can be more intuitive when combined with truth values. The preferences of agents can be captured in two figures, one captures how they think about the holders of information,i.e. who does know, and the second shows their personal preferences about the gossip bits themselves, i.e. what is true. By combining these two we can even add to the expressivity as agents can reason not only about the distribution of knowledge but they can speak also about the information itself.

Thanks to the RCP approach from van Ditmarsch[2] we can quite easily work with assumptions about the distribution of knowledge amongst themselves. However, due to our use of Baltag and Smets-like updates[1], we can talk about different relations of players and introduce the ability of deceit while maintaining sincerity. And as last, Hommersom's treatment of models[4] allows for updates without the risk of unrealistic information spreading.

There could be even more elaborate models enhanced with performatives. Human communication, even on so-

[9] Although this condition could be simply added by Caknowsb, i.e. it is common knowledge that Anne knows Bill's gossips.

cial networks, is filled with signals that indicate how the update should be understood. Performatives are part also of the FIPA Agent Communication specifications and therefore could be incorporated in our framework quite easily. They could, for example, dynamically change levels of trust, truth values, etc. However, all these effects could be modelled with constructs from our current language. An interesting question, on the other hand, is the creation of simpler models. For example, the agent relation structure could be in some way incorporated into the models instead of standing aside.

5. Summary

The article presented some basic ideas of our approach to find a public communication protocol that would be capable of protecting the privacy of transfered messages for a given group of agents. We presented some results from the analysis of the Russian cards problem and parts of the formal apparatus that allows us a richer description of communication in social situations. The presented findings suggest that there is a possibility of creating a functioning protocol of the desired type. However, it certainly will be burdened with strict limitations.

References

[1] A. Baltag and S. Smets, "Talking Your Way into Agreement: Belief Merge by Persuasive Communication", *Proceeding of the Second Multi-Agent Logics, Languages, and Organisations Federated Workshops*, vol. 494, pp. 129–141, 2009.

[2] H. van Ditmarsch, "The Russian cards problem: a case study in cryptography with public announcements", University of Otago, Otago, Technical OUCS-2002-08, Oct. 2002.

[3] H. van Ditmarsch, W. Van Der Hoek, R. Van Der Meyden, and J. Ruan, "Model checking Russian cards", *Electronic Notes in Theoretical Computer Science*, vol. 149, no. 2, pp. 105–123, 2006.

[4] A. Hommersom, et al., "Update semantics of security models", *Information, interaction, and agency*, Kluwer Academic Pub, pp. 289–327, 2005.

[5] M. Jakob, Z. Moler, M. Pěchouček, and R. Vaculín, "Content-Based Privacy Management on the Social Web", *Web Intelligence and Intelligent Agent Technology (WI-IAT)*, 2011 IEEE/WIC/ACM International Conference on, vol. 3, pp. 277–280, 2011.

[6] O. Majer and M. Peliš, "Logic of Questions and Public Announcements", *Eighth International Tbilisi Symposium on Logic, Language and Computation 2009*, Lecture Notes in Computer Science, Springer, pp. 145–157, 2011.

[7] P. Švarný, "Využití logiky v bezpečnosti IT." Thesis. University of Economics, Prague. To be published in 2013.

t-Filters and Fuzzy t-Filters and Their Properties

Post-Graduate Student:
MGR. MARTIN VÍTA
Institute of Computer Science of the ASCR, v. v. i.
Pod Vodárenskou věží 2
182 07 Prague 8, CZ
vita@cs.cas.cz

Supervisor:
ING. PETR CINTULA, PH.D.
Institute of Computer Science of the ASCR, v. v. i.
Pod Vodárenskou věží 2
182 07 Prague 8, CZ
cintula@cs.cas.cz

Field of Study:
Algebra, Theory of Numbers, and Mathematical Logic

The work of Martin Víta was supported by grants GD401/09/H007 and P202/10/1826 of the Grant Agency of the Czech Republic.

Abstract

Theory of special types of (fuzzy) filters on different algebras of non-classical logics has been intensively studied in the last decade. This contribution provides a generalization which covers many particular results and allows us to deal with special types of (fuzzy) filters in a uniform way. Our approach is based on simple definitions of a t-filter and a fuzzy t-filter. We are going to state and prove some basic properties of (fuzzy) t-filters and formulate generalizations of the most typical kinds of results occurring in the literature. We show that these results in this field can be generated via simple principles described in this paper.

1. Preliminaries and Basic Definitions

In this section we are going to recall the notion of a filter on a residuated lattice (more preciously a *bounded pointed commutative integral residuated lattice*).

In the whole text we are going to use often the following comfortable convention: the symbol $\overline{x}$ will be used as an abbreviation of $x, y, \ldots$ i. e. for a listing of variables (neither a sequence, nor a vector) – therefore we can correctly write $\overline{x} \in L$ instead of $x, y, \cdots \in L$.

At the very beginning we recall the definition of a residuated lattice.

Definition 1 *A* residuated lattice *is a structure*

$$\boldsymbol{L} = (L, \&, \to, \wedge, \vee, \overline{0}, \overline{1})$$

of type $(2, 2, 2, 2, 2, 0, 0)$ *satisfying the following conditions:*

1. $(L, \wedge, \vee, \overline{0}, \overline{1})$ *is a bounded lattice.*
2. $(L, \&, \overline{1})$ *is commutative semigroup with the unit element* $\overline{1}$.
3. $(\&, \to)$ *form an adjoint pair, i.e.* $x \,\&\, z \leq y$ *if and only if* $z \leq x \to y$ *for all* $x, y, z \in L$.

A comprehensive overview on residuated lattices and their subvarieties is provided by [1].

Since now we assume that $\boldsymbol{L}$ is a residuated lattice and L its domain.

Definition 2 *A non-empty subset* F *of* $\boldsymbol{L}$ *is called a* filter *on* $\boldsymbol{L}$ *if it satisfies these two conditions:*

1. *if* $x, y \in F$, *then* $x \,\&\, y \in F$,
2. *if* $x \in F$, $x \leq y$, *then* $y \in F$,

for all $x, y \in L$.

The equivalent definition of a filter on a residuated lattice is presented in the following theorem:

Theorem 3 ([2]) *A non-empty subset* F *of* $\boldsymbol{L}$ *is a filter on* $\boldsymbol{L}$ *if and only if it satisfies this following conditions:*

1. $\overline{1} \in F$,
2. *if* $x \in F$ *and* $x \to y \in F$, *then* $y \in F$.

for all $x, y \in L$.

Roughly said, this theorem shows that filters are just 'deductively closed' subsets of L. Therefore some authors uses the name 'deductive systems' – [3]. The connection between the notion of a filter in logic and the notion of a filter in algebra is described in [4], [5] or [6].

Filters can be also defined by many other equivalent ways, for example as a subsets of L containing $\overline{1}$ and satisfying one of these following conditions:

1. if $x \to y \in F$ and $y \to z \in F$, then $x \to z \in F$,
2. if $x \to y \in F$ and $x \,\&\, z \in F$, then $y \,\&\, z \in F$,
3. if $x, y \in F$ and $x \leq y \to z$, then $z \in F$.

for all $x, y, z \in L$ (see [2]).

2. Notion of a t-Filter

In the literature there is a great amount of papers concerning different types of filters on residuated lattices (or subvarieties of residuated lattices, such as BL-algebras, MTL-algebras, etc.). The notion of a t-filter on a residuated lattice was set up in order to generalize these particular results about special types of filters (implicative, boolean, etc.).

Definition 4 ([7]) *Let t be an arbitrary term in the language of residuated lattices. A filter F on $\boldsymbol{L}$ is a t-filter if $t(\overline{x}) \in F$ for all $\overline{x} \in L$.*

The definition in the submitted paper [7] uses slightly more general underlying structure which does not require $\overline{1}$ to be the greatest element, but for the purposes of this contribution we can conveniently restrict ourselves on residuated lattices.

It can be shown that many special types of filters are just t-filters for suitably chosen term t: in BL-algebras implicative filters are just t-filters for

$$t = x \to x \,\&\, x,$$

positive implicative filters are t-filters for

$$t = (\neg x \to x) \to x,$$

and fantastic filters are t-filters for

$$t = \neg\neg x \to x.$$

Recall that in BL-algebras $\neg x$ is defined as $x \to \overline{0}$.

This follows from the corresponding (technical) results in [8]. These special types of filters are defined by condition '$\overline{1}$ is in the filter' and some additional specific condition. After stating this definition, authors prove that defined special type of filter is a filter. This approach is in some sense a bit unnatural: we usually expect that a special type of filter is a filter having some additional properties. This intuition is rendered in the definition of a t-filter.

Remark: the question whether we can find the corresponding term t for a given special type of filter is closely related to the question of axiomatizability of the logics involved.

Note that the answer of the question whether the class of t_1-filters and the class of t_2-filters (for different terms t_1 and t_2) are equal, depends on the algebra we are working on.

3. Fuzzy Case

Definition 5 *Let X be an arbitrary non-empty set. A function $\mu : X \longrightarrow [0,1]$ is called a* fuzzy set *on X. If μ is a fuzzy set on the set X, then for any $\alpha \in [0,1]$ we denote the set $\{x \in X \mid \mu(x) \geq \alpha\}$ by the symbol μ_α.*

Note that a characteristic function of an arbitrary set A can be viewed as a fuzzy set on A.

Definition 6 *A fuzzy set μ of L is a* fuzzy filter *on $\boldsymbol{L}$ if and only if it satisfies the following two conditions:*

1. $\mu(x \,\&\, y) \geq \min\{\mu(x), \mu(y)\}$,
2. *if $x \leq y$, then $\mu(x) \leq \mu(y)$,*

for all $x, y \in L$.

As presented in [2], this definition can be alternatively formulated in the following way:

Theorem 7 *A fuzzy set μ of L is a fuzzy filter on $\boldsymbol{L}$ if and only if it satisfies the following two conditions:*

1. $\mu(y) \geq \min\{\mu(x), \mu(x \to y)\}$,
2. $\mu(x) \leq \mu(\overline{1})$,

for all $x, y \in L$.

The first condition in this theorem is in fact a 'fuzzy version of modus ponens'.

The relationship between fuzzy filters and filters on $\boldsymbol{L}$ is illustrated by the next theorem:

Theorem 8 ([2]) *A fuzzy set μ on $\boldsymbol{L}$ is a fuzzy filter if and only if for each $\alpha \in [0,1]$ the (crisp) set μ_α is either empty or a filter on $\boldsymbol{L}$.*

One of the key notions of this contribution is a notion of a fuzzy t-filter, which is a natural fuzzification of the concept of a t-filter.

Definition 9 *Let t be an arbitrary term in the language of residuated lattices. A fuzzy filter μ on $\boldsymbol{L}$ is called a* fuzzy t-filter *on $\boldsymbol{L}$, if it satisfies $\mu(t(\overline{x})) = \mu(\overline{1})$ for all $\overline{x} \in L$.*

According to the previous definition, we can see that fuzzy boolean filters ([2] and [9]) are just fuzzy t-filters for

$$t = x \vee \neg x.$$

Analogously, regular fuzzy filters ([2]) are fuzzy t-filters for

$$t = \neg\neg x \to x.$$

These filters on MTL-algebras (in the crisp case) are known as IMTL-filters (see [10]).

Similarly as in the crisp case, special types of fuzzy filters are usually presented in a slightly different way – for example fuzzy fantastic filters (fuzzy MV-filters) are defined as a fuzzy filters satisfying

$$\mu(((x \to y) \to y) \to x) \geq \mu(y \to x).$$

However, as shown in [2] again, fuzzy fantastic filters are just fuzzy t-filters where condition

$$\mu(((x \to y) \to y) \to ((y \to x) \to x)) = \mu(\overline{1})$$

holds for all $x, y \in L$.

There is a very close relationship between t-filters and fuzzy t-filters. It can be described in the terms of 'cut-consistency', which is the content of the next theorem.

Theorem 10 *A fuzzy filter μ on $\boldsymbol{L}$ is a fuzzy t-filter if and only if for each $\alpha \in [0, 1]$ the (crisp) set μ_α is either empty or a t-filter on $\boldsymbol{L}$.*

Proof: Let μ be a fuzzy t-filter on $\boldsymbol{L}$, $\alpha \in [0, 1]$. If $\alpha > \mu(x)$ for all $x \in L$, then μ_α is obviously empty. Otherwise let $z \in \mu_\alpha$. Thus $\mu(z) \geq \alpha$. From Theorem 8 we already know that μ_α is a filter on $\boldsymbol{L}$.

Since μ is a fuzzy t-filter, then $\mu(t(\overline{x})) = \mu(\overline{1})$ for all $\overline{x} \in L$, so

$$\mu(t(\overline{x})) = \mu(\overline{1}) \geq \mu(z) \geq \alpha$$

for all $\overline{x} \in L$, hence $\mu(t(\overline{x})) \geq \alpha$, so $t(\overline{x}) \in \mu_\alpha$ for all $\overline{x} \in L$. Thus μ_α is a t-filter.

Conversely, we assume that μ_α is a t-filter or an empty set for each $\alpha \in [0, 1]$. Let us choose $\mu(\overline{1})$ as α. Since $\overline{1} \in \mu_{\mu(\overline{1})}$, then $\mu_{\mu(\overline{1})}$ is non-empty. So $\mu_{\mu(\overline{1})}$ is – due to the assumption – a t-filter, thus $t(\overline{x}) \in \mu_{\mu(\overline{1})}$ for all $\overline{x} \in L$. Hence $\mu(t(\overline{x})) \geq \mu(\overline{1})$ and therefore $\mu(t(\overline{x})) = \mu(\overline{1})$ for all $\overline{x} \in L$. ∎

As the corollary we obtain a relationship between characteristic functions of t-filters and fuzzy t-filters.

Theorem 11 *Let F be a filter of $\boldsymbol{L}$. Then F is a t-filter if and only if χ_F is a fuzzy t-filter of $\boldsymbol{L}$.*

Proof: Straightforward consequence of the previous theorem. ∎

There is one more simple theorem showing the relationship between t-filters and fuzzy t-filters generalizing Theorem 5.20 in [2].

Theorem 12 *Let F be a t-filter on $\boldsymbol{L}$. Then there exists a fuzzy t-filter μ on $\boldsymbol{L}$ such that $\mu_\alpha = F$ for some $\alpha \in (0, 1)$.*

Proof: Let us define μ on L by cases:

$$\mu(x) = \begin{cases} \alpha & \text{if } x \in F \\ 0 & \text{if } x \notin F, \end{cases}$$

where α is an arbitrary number ($0 < \alpha < 1$). Clearly, $\mu_\alpha = F$. Now we simply apply Theorem 10. ∎

4. Core Results about t-Filters and Fuzzy t-Filters

In this section we are going to present generalizations of many statemements about special types of filters via our notion of a (fuzzy) t-filter.

Let us start with a crisp case.

Theorem 13 ([7]) *Let F and G be filters on a residuated lattice $\boldsymbol{L}$ such that $G \supseteq F$. If F is a t-filter, then so is G.*

Proof: Thanks to the definition of a t-filter the proof is obvious. ∎

This theorem is often referred as an 'extension theorem'. Example of a particular result is provided in the next theorem.

Theorem 14 ([11]) *If F is a positive implicative filter, then every filter G containing F is also a positive implicative filter in any* BL*-algebra.*

Theorem 15 ([7]) *Let $\mathbb{B}$ be a subvariety of residuated lattices and $\boldsymbol{L} \in \mathbb{B}$. Moreover let $\mathbb{C}$ be a subvariety of $\mathbb{B}$ such that in all $\boldsymbol{C} \in \mathbb{C}$ the equation $t = \overline{1}$ holds. Then the following statements are equivalent:*

1. *Every filter on $\boldsymbol{L}$ is a t-filter.*
2. *$\{\overline{1}\}$ is a t-filter.*
3. *$\boldsymbol{L} \in \mathbb{C}$.*

This theorem generalizes many particular theorems like the following one:

Theorem 16 ([11]) *In any* BL*-algebra $\boldsymbol{A}$, the following conditions are equivalent:*

1. *Every filter on $\boldsymbol{A}$ is an implicative filter.*
2. *$\{\overline{1}\}$ is an implicative filter.*
3. *$\boldsymbol{A}$ is a Gödel algebra.*

It can be proved easily by the fact that implicative filters on BL-algebras are just t-filters for $t = x \to x \,\&\, x$, the fact that Gödel algebras are just BL-algebras satisfying $x \to x \,\&\, x = \overline{1}$ and our theorem about t-filters.

Now we are going to state and prove fuzzy counterparts of mentioned theorems.

Theorem 17 *Let μ, γ be fuzzy filters on $\boldsymbol{L}$, $\mu(x) \leq \gamma(x)$ for all $x \in L$, and moreover, $\mu(\overline{1}) = 1$. If μ is a fuzzy t-filter, then γ is also a fuzzy t-filter.*

Proof: μ is a fuzzy t-filter, hence $\mu(t(\overline{x}) = \mu(\overline{1})$ for all $\overline{x} \in L$. Since $\mu \leq \gamma$ (pointwise) and also $\mu(\overline{1}) = \gamma(\overline{1}) = 1$, we directly obtain

$$\gamma(t(\overline{x})) = 1 = \gamma(\overline{1}),$$

for all $\overline{x} \in L$, thus γ is a fuzzy t-filter. ∎

Application of this theorem is straightforward again. If we we choose $t = \neg\neg x \to x$ for example, we get the following particular result taken from [2]:

Corollary 18 ([2]) *Let μ and ν be fuzzy filters of $\boldsymbol{L}$ with $\mu \leq \nu$ and $\mu(\overline{1}) = \nu(\overline{1})$. If μ is a fuzzy regular filter of $\boldsymbol{L}$, then so is ν.*

The set of all fuzzy filters on $\boldsymbol{L}$ is denoted by the symbol $FF(\boldsymbol{L})$. The next theorem shows how can we describe a subvariety of residuated lattices via the properties of (all/certain important) filters on $\boldsymbol{L}$.

Theorem 19 *Let $\mathbb{B}$ be a subvariety of residuated lattices and $\boldsymbol{B} \in \mathbb{B}$. Moreover let $\mathbb{C}$ be a subvariety of $\mathbb{B}$ such that in all $\boldsymbol{C} \in \mathbb{C}$ the equation $t = \overline{1}$ holds. Then the following statements are equivalent:*

1. *Every fuzzy filter on $\boldsymbol{B}$ is a fuzzy t-filter.*
2. *$\chi_{\{\overline{1}\}}$ is a fuzzy t-filter.*
3. *$\mu_{\mu(\overline{1})}$ is a t-filter for any $\mu \in FF(\boldsymbol{L})$.*
4. *$\boldsymbol{B} \in \mathbb{C}$.*

Proof: At first we are going to prove the 'circle' 1., 2., 4. and then we are going to connect the third statement.

1.⇒2.: $\chi_{\{\overline{1}\}}$ is a fuzzy filter (consequence of Theorem 8) and thanks to the assumption $\chi_{\{\overline{1}\}}$ is also a fuzzy t-filter.

2.⇒4.: If $\chi_{\{\overline{1}\}}$ is a fuzzy t-filter, then $\{\overline{1}\}$ is a t-filter (using Theorem 10). Therefore $t(\overline{x}) \in \{\overline{1}\}$ for all $\overline{x} \in L$, hence $t = \overline{1}$, so $\boldsymbol{B} \in \mathbb{C}$.

4.⇒1.: If $\boldsymbol{B} \in \mathbb{C}$, then $t(\overline{x}) = \overline{1}$ for all $\overline{x} \in L$, and hence also $\mu(t(\overline{x})) = \mu(\overline{1})$. Thus fuzzy filter μ is a fuzzy t-filter.

1.⇒3.: If $F \in FF(\boldsymbol{L})$, then by the assumption μ is a fuzzy t-filter. We apply Theorem 10 with $\alpha = \mu(\overline{1})$.

3.⇒2.: $\chi_{\{\overline{1}\}}$ is a fuzzy filter, so we choose $\chi_{\{\overline{1}\}}$ as μ in 3. We obtain that $\mu_{\mu(\overline{1})} = \{\overline{1}\}$ is a t-filter an also — by Theorem 11 — that $\chi_{\{\overline{1}\}}$ is a fuzzy t-filter. ∎

Again we are going to show an application of this theorem on concrete particular example concerning fuzzy Boolean filters (Theorem 4.15 from [2]).

Theorem 20 *In any residuated lattice $\boldsymbol{L}$, the following assertions are equivalent, for all $x, y \in L$:*

1. *$\boldsymbol{L}$ is a Boolean algebra.*
2. *Every fuzzy filter of $\boldsymbol{L}$ is a fuzzy Boolean filter of $\boldsymbol{L}$.*
3. *$\chi_{\{\overline{1}\}}$ is a fuzzy Boolean filter of $\boldsymbol{L}$.*

Proof: Let t be the term $\neg x \vee x$. Recall that 'residuated lattices where $t = \overline{1}$ holds are just Boolean algebras'. The rest is provided by Theorem 19. ∎

5. Application of the Theory and Obtaining 'New' Results

One of the main aims of this paper is to illuminate that many published particular results can be generated (using the theory presented in this paper) in an 'automatic' way.

We are going to demonstrate this fact on the examples arising in residuated lattices, resp. MTL-algebras. We are going to define a new special (and artificial!) kind of filter, called *prelinear filter*, as a t-filter for certain t, analogously also *fuzzy prelinear filter*.

Recall that MTL-algebras are just residuated lattices satisfying $(x \to y) \vee (y \to x) = \overline{1}$ (axiom of prelinearity).

Definition 21 *A filter F on $\mathbf{L}$ is called a* prelinear filter *on $\mathbf{L}$, if it satisfies $((x \to y) \vee (y \to x)) = \overline{1}$ for all $x, y \in L$.*

Definition 22 *A fuzzy filter μ on $\mathbf{L}$ is called a* fuzzy prelinear filter *on $\mathbf{L}$, if it satisfies*

$$\mu((x \to y) \vee (y \to x)) = \mu(\overline{1})$$

for all $x, y \in L$.

Theorem 23 *The following statements are equivalent for each residuated lattice $\mathbf{L}$:*

1. *Every filter on $\mathbf{L}$ is a prelinear filter.*
2. *$\{\overline{1}\}$ is a prelinear filter.*
3. *$\mathbf{L}$ is an* MTL*-algebra.*

Theorem 24 *Let μ and γ be fuzzy filters of $\mathbf{L}$ satisfying $\mu(x) \leq \gamma(x)$ for all $x \in L$ and $\mu(\overline{1}) = \gamma(\overline{1})$. If μ is a fuzzy prelinear filter on $\mathbf{L}$, then so is γ.*

Theorem 25 *The following assertions are equivalent, for all $x, y \in L$:*

1. *Every fuzzy filter of $\mathbf{L}$ is a fuzzy prelinear filter of $\mathbf{L}$.*
2. *$\chi_{\{\overline{1}\}}$ is a fuzzy prelinear filter of $\mathbf{L}$.*
3. *$\mathbf{L}$ is an* MTL*-algebra.*

Proof: All these theorems are simple consequences of theorems in the previous sections for

$$t = (x \to y) \vee (y \to x).$$

∎

6. Conclusion and Final Remarks

This work summarizes some results about generalizations of special types of filters. As we can see, provided proofs are simple — the significance of this work comes not from the complexity of the theory, but rather from the amount of particular results which it covers.

The submitted paper [7] contains also generalization of results about the relationship of special types of filters and quotient algebras. In one of the prepared papers we are going to state and prove analogous result about fuzzy quotients.

There is one more way of generalizations: the most widely used definition of a fuzzy filter is closely bind to the Gödel min-conjunction. In this context it is possible to investigate such generalizations where this min-conjunction is replaced by any t-norm.

References

[1] P. Jipsen and C. Tsinakis, "A survey of residuated lattices," in *Ordered Algebraic Structures* (J. Martínez, ed.), pp. 19–56, Dordrecht: Kluwer Academic Publishers, 2002.

[2] Y. Zhu and Y. Xu, "On filter theory of residuated lattices.," *Information Sciences*, vol. 180, no. 19, pp. 3614–3632, 2010.

[3] E. Turunen, "Boolean deductive systems of BL-algebras," *Archive for Mathematical Logic*, vol. 40, no. 1, pp. 467–473, 2001.

[4] P. Cintula, P. Hájek, and C. Noguera, eds., *Handbook of Mathematical Fuzzy Logic (in 2 volumes)*, vol. 37, 38 of *Studies in Logic, Mathematical Logic and Foundations*. London: College Publications, 2011.

[5] J. Czelakowski, *Protoalgebraic Logics*, vol. 10 of *Trends in Logic*. Dordrecht: Kluwer, 2001.

[6] J. M. Font, R. Jansana, and D. L. Pigozzi, "A survey of Abstract Algebraic Logic," *Studia Logica*, vol. 74, no. 1–2, Special Issue on Abstract Algebraic Logic II, pp. 13–97, 2003.

[7] M. Víta and P. Cintula, "Short note: on special types of filters on subvarieties of commutative residuated lattices," *Soft Computing*. submitted.

[8] M. Kondo and W. A. Dudek, "Filter theory of BL-algebras," *Soft Computing*, vol. 12, no. 5, pp. 419–423, 2008.

[9] L. Lianzhen and L. Kaitai, "Fuzzy boolean and positive implicative filters of BL-algebras.," *Fuzzy Sets and Systems*, vol. 152, no. 2, pp. 333–348, 2005.

[10] R. A. Borzooei, S. K. Shoar, and R. Ameri, "Some types of filters in MTL-algebras," *Fuzzy Sets and Systems*, vol. 187, no. 1, pp. 92–102, 2012.

[11] M. Haveshki, A. Saeid, and E. Eslami, "Some types of filters in BL-algebras," *Soft Computing*, vol. 10, no. 8, pp. 657–664, 2006.

Ústav informatiky AV ČR, v.v.i.
DOKTORANDSKÉ DNY '12

Vydal
MATFYZPRESS
vydavatelství
Matematicko-fyzikální fakulty
Univerzity Karlovy
Sokolovská 83, 186 75 Praha 8
jako svou *– not yet –* . publikaci

Obálku navrhl František Hakl

Z předloh připravených v systému LaTeX
vytisklo Reprostředisko MFF UK
Sokolovská 83, 186 75 Praha 8

Vydání první

Praha 2012

ISBN *– not yet –*