ROMANIAN JOURNAL OF INFORMATION SCIENCE AND TECHNOLOGY
Volume **17**, Number 2, 2014, 121–133

Multi-Objective Optimization of Advanced Computing Systems: Some Achievements and Fertile Work Directions

Lucian N. VINŢAN

Computer Science and Electrical Engineering Department
"*Lucian Blaga*" University of Sibiu, Sibiu, Romania
E-mail: lucian.vintan@ulbsibiu.ro

Abstract. The main aim of this work is to synthetically point out a state of the art in developing automatic design space exploration methods dedicated to Computer Architecture multi-objective optimizations, as they were developed during the time by the author and his research group. Also the paper points out some of the author's envisaged further work directions that would hopefully advance this important research field. So, in order to achieve better convergence speed and solutions' quality, specific domain-knowledge for each of the target computing system is necessary to be developed and integrated into the DSE tools. Adequately representing domain-knowledge is a significant scientific challenge. Also, automatically calculating the degrees of logical contradiction for a certain domain-knowledge represented using fuzzy logic rules would be of certain scientific interest. Effective meta-optimization methods are needed. Integrating some advanced Response Surface Models to accelerate the automatic DSE processes represents another fertile research challenge, too.

Key-words: Computing Systems, Design Space Exploration, Multi-Objective Optimization Methods, Meta-Optimization, Domain-Knowledge, Response Surface Models.

1. Introduction

The main aim of this short paper is to synthetically point out a state of the art in developing automatic design space exploration methods dedicated to Computer Architecture multi-objective optimizations, as they were developed in *"The*

Advanced Computer Architecture and Processing Systems" Research Centre (ACAPS) from "Lucian Blaga" University of Sibiu (LBUS), Romania, led by the author – see `http://acaps.ulbsibiu.ro/index.php/en/`. Also the paper points out some of the author's envisaged potential fertile work directions that would hopefully improve and enlarge our previous achievements in this important research field.

The main scope of the present-day Computer Architecture research developed in the ACAPS Research Group that the author led at LBUS consists in developing innovative computer architectures and optimizing them. Some of the subsequent research objectives are the followings:

- Develop some novel effective micro-architectures (micro-architectures with predictive-speculative processing, Network on Chip, etc.)
- Improve and enlarge an already implemented robust and fast automatic design space exploration framework for hardware-software co-optimization of complex computing systems
- Evaluate & compare different multi-objective Design Space Exploration (DSE) algorithms
- Research how domain-knowledge could be represented and integrated into the DSE algorithms
- Quantify domain-knowledge impact, with corresponding advantages and even disadvantages, on the DSE processes
- Develop meta-optimization methods (adaptive selection of DSE algorithms during the multi-objective optimization process)
- Compute Degrees of Contradiction for Fuzzy Logic Rules implementing Computer Architecture Domain-Knowledge
- Develop of an new approach based on the "Pareto-fuzzy set" concept
- Integrate Response Surface Models in our developed DSE tool, etc.

Multi-objective optimization (performance, power consumption, temperatures, complexity...) of computing systems having many parameters is for sure a very complex problem. Not only the hardware needs to be simulated and evaluated; usually we need hardware and software co-optimization (cross-layer optimization). Usually, exhaustive search is prohibited due to the enormous design space. The solution consists in developing and implementing some advanced heuristic algorithms in order to solve this problem. In our optimization research we used a Pareto-based approach by implementing some multi-objective evolutionary (genetic) algorithms and bio-inspired algorithms belonging to the Particle Swarm Optimization class. The solutions' quality is usually evaluated through some well-known implemented metrics like hyper-volume (also used as a stop condition), 7 point average distance, coverage (for comparing two DSE algorithms), Two Set Hyper-volume Difference (for better understanding

about how much one DSE algorithm dominates the other one in a multidimensional space), etc.

All these briefly presented DSE characteristics were implemented by us in a dedicated complex effective software product entitled *Framework for Automatic Design Space Exploration (FADSE)*. It was initially designed and implemented by the author's former PhD student Horia Calborean under the author's scientific coordination and it is further developed by some of the ACAPS research team members (Radu Chis, Cristian Cotofana *et al.*). The software tool is publicly available (see `https://code.google.com/p/fadse/`). It includes many state-of-the-art multi-objective DSE algorithms (integrated with *jMetal* library). FADSE can be connected to any existing (architectural) simulator (connector needed). It is easily to use XML configuration interface. It was developed an easily to use interface for connectors to the computing systems simulators, too [1], [2].

The search time is reduced through database integration (according to our experiences we achieved up to 67% simulations results' reuse). The implemented DSE multi-objective evolutionary algorithms were adapted in order to allow distributed evaluations on LANs or High Performance Computers. Also FADSE implements some reliability techniques: if a client does not respond the simulation is resubmitted to another client. For problems like: power loss or if the server stops responding, it was implemented a check pointing mechanism. This allows us to restart the DSE process from an intermediate state [2].

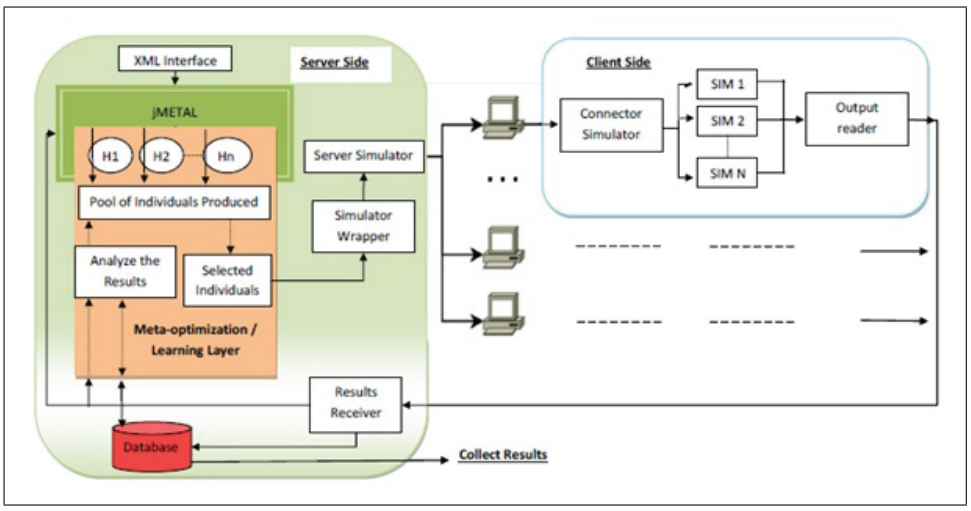

Fig. 1. Structure of FADSE (distributed version).

2. Some Computing Systems Optimizations

We successfully optimized with FADSE tool some advanced computing systems implemented through software simulators like GAP/GAPtimize (developed at Augs-

burg University by Professor Theo Ungerer's research group) [3], [4], M-Sim 2 superscalar/SMT simulator improved by us with a Selective Load Value Predictor [5], UniMap (our developed Network on Chip simulator) [6], Sniper multicore simulator (with objectives Energy, CPI, Integration Area) [7], etc. For the first three projects we already developed and implemented specific effective domain-knowledge in FADSE. Using them, both the solutions' quality and the convergence speed were improved. Now we are thinking about some multicore systems domain-knowledge to be developed and implemented in our DSE tool (for example related to Sniper Multicore Simulator). All the experimental results focused on the above architectures show that our automatic design space exploration provides significantly better configurations than previous manual design space explorations. Using our developed FADSE tool we were able to find very good configurations by evaluating a very small percentage of the total search space (involving thus reasonable time of computation on the available resources – LANs and a HPC system with about 120 cores).

We already integrated Hotspot simulator for temperature computations in Sniper simulator, implementing automatic floorplan generation facilities [8]. This will allow us a further 4D space optimization for Sniper Multicore Simulator rather than a 3D space optimization, as it is only possible with the actual standard Sniper simulator. We already optimized the hardware and software parameters' values of the multicore Sniper simulator running SPLASH-2 benchmarks suite. We had shown the impact of the parameters on the optimization's multi-objectives (performance, area and power consumption). Furthermore, for the best found Pareto configurations the temperatures were computed [7]. We also successfully used FADSE for optimization of an electrical motor in the acceleration pedal of a car based on a model implemented in Comsol with Matlab, developed by an automotive company; in this case we looked for the biggest constant force.

3. Implementing Domain-Knowledge in DSE Algorithms

In order to achieve better convergence speed and solutions' quality we implemented specific domain-knowledge for each of the target computer architecture to be optimized. Domain-knowledge is represented by a system of complete, non-redundant and non-contradictory fuzzy logic rules and/or other specific restrictions. As far as we know, we are the first ones using fuzzy logic as a method to express computer architecture knowledge into a DSE tool. Some common sense CPU Fuzzy Logic Rules Examples: IF IL1Cache_Size IS *small* AND DL1Cache_Size IS *small* THEN UL2Cache_size IS *big* or IF IL1Cache_Size IS *big* AND DL1Cache_Size IS *big* THEN UL2Cache_size IS *small*, etc.

Particularly, calculating degrees of contradiction between fuzzy logic rules is another scientific objective for us (see below). For each of the fuzzy logic rule a Fuzzification → Inference → Defuzzification (Crisp Value) process is computed. As a consequence, Mutation Genetic Operator in our implemented DSE algorithms was essentially modified as follows:

1. *For all the parameters (genes) in the individual (chromosome);*
   (a) *If a fuzzy rule exists for the current parameter, having it as a consequent;*
      i. *Compute Center Of Gravity (COG) of this parameter taking into consideration the current values of the other parameters;*
      ii. *Compute the membership value of the COG;*
      iii. *Generate a pseudo-random number between 0 and 1;*
      iv. *If the previously generated pseudo-random number is smaller than "fuzzy probability";*
         A. *Current parameter is set to a value equal with COG;*
      v. *Jump to next iteration;*
   (b) *Otherwise (do bit flip mutation);*
      i. *Generate a pseudo-random number between 0 and 1;*
      ii. *If the previously generated pseudo-random number is smaller than the probability of mutation;*
         A. *Change the current parameter to a random value;*
      iii. *Jump to next iteration;*
2. *STOP.*

Fig. 2. The Fuzzy Logic Mutation Genetic Operator.

For the defuzzification process we have used the center of gravity (COG) method. Center of gravity is one of the most used methods and it is computed using the following equation:

$$COG = \frac{\int x\mu(x)dx}{\int \mu(x)dx} \tag{1}$$

where $\mu(x)$ is the membership function.

We have proven in our optimization research focused on a Superscalar/SMT Microarchitecture improved by us with a Selective Load Value Predictor [5] that the explorations with fuzzy logic rules expressing processor design-knowledge are involving a noticeable solutions quality gain. Thus, the added fuzzy logic rules are useful, involving a better Pareto front than without using them. Also the spread of the solutions found by the run with fuzzy logic rules was better, providing more choices for computer architects in selecting one of the Pareto optimal configurations discovered by the DSE meta-heuristic algorithm. The DSE algorithm's convergence is significantly faster by running with adequate fuzzy logic rules, too. In fact, fuzzy logic rules are not the single way for describing such domain-knowledge; it would be useful to have a look on semantic nets, knowledge representation methods and on other methods that describe domain-ontologies in semantic web, etc. Anyway, based on our experience in developing Computer Architecture domain-knowledge, if the restrictions are too hard the design space might be not uniformly searched (involving thus a biased search) with a negative influence on the final results. On the other hand, weak domain-knowledge might not effectively reduce the convergence time of the DSE process. An optimal

trade-off would be tuned, based on designer's experience and talent. In our further research we'll adapt some state of the art evolutionary/bio-inspired algorithms using specific domain-knowledge related to the target architecture and applications. Our explorations will be focused on superscalar/Simultaneous Multi-Threading processors (SMT), multicore/manycore architectures (Sniper), Network on Chip, High Performance Computers (HPC), etc. As applications we'll use sequential programs but also parallel programs (both shared memory and message passing paradigms). The automatic DSE process will consider domain-knowledge represented through fuzzy logic rules. Heuristic algorithms from the fields of Machine Learning, Evolutionary Computing and Bio-inspired Computing will be employed to perform a multi-objective search for the best design (Pareto surface).

4. Degrees of Contradiction for Fuzzy Logic Rules

In [9] we analyzed some degrees of logical contradiction for two Conjunctive Normal Form (CNF) fuzzy logic rules and suggested some possible useful software implementations, especially for Multi-Objective Optimization through Automatic Design Space Exploration applied in Advanced Computer Architecture research. Particularly, automatically calculating the degrees of logical contradiction for a certain domain-knowledge represented using fuzzy logic rules would be of certain scientific interest. Automatically computing the contradiction degrees for a set of N>2 fuzzy logic rules (thus not for just two fuzzy logic rules!), using the corresponding concrete defined membership functions, would be helpful in order to avoid solutions quality degradation due to these – possible too contradictory – rules. As far as we know this represents an open problem. Also, we need in our further developed mono-core and multi-core domain-ontologies (example: for Sniper multicore simulator) to be sure that the contradiction degrees in a complex set of fuzzy logic rules are quite "acceptable" in order to maintain the optimization effectiveness. If not, the set of fuzzy logic rules must be automatically modified accordingly (implementing thus a "polymorphic domain-knowledge"), that would be a significant scientific challenge, too. (What certain fuzzy logic rule(s) would be eliminated in order to optimally decrease the global contradiction degree?) Finding some thresholds for acceptable degrees of contradiction is problem dependent. Finally, an optimal set of such fuzzy logic rules, for a certain optimization problem, is envisaged. More general, developing a software tool capable to automatically calculate the contradiction degrees of a set of (CNF) fuzzy rules would be of great interest, too. Particularly we are interested to develop and integrate such a tool in our *Framework for Automatic Design Space Exploration* in order to improve the optimization process through an adequate set of fuzzy logic rules that are implementing a certain Computer Architecture domain-knowledge.

5. Meta-Optimization

Based on our experience in Computer Architecture multi-objective optimization, there is not a general optimal multi-objective DSE algorithm (one algorithm might

converge fastest and other one provides best solution for a certain architecture; for other one might be different) [10], [11]. As a consequence, at the suggestion and with the significant help of Dr. Muhammad Ali Ismail from NED University of Engineering and Technology, Pakistan, we implemented a new abstraction level in FADSE called meta-optimization. It acts over the domain-knowledge and DSE algorithms (meta-heuristic) levels. Meta-optimization searches within a search space of heuristic methods. Meta-optimizations are concerned with intelligently choosing the right heuristic DSE algorithm in a given situation in order to handle a wide range of problem domains rather than current meta-heuristic (DSE algorithm). Developing some effective adaptive meta-optimization algorithms is an important scientific challenge for us. According to our first experiences meta-optimization was used to optimize the performance of design space explorations, driving two different multi-objective DSE algorithms concurrently. More precisely, we selected two genetic algorithms, NSGA-II and SPEA2. In this connection, we developed an elitist evolutionary meta-optimization algorithm.

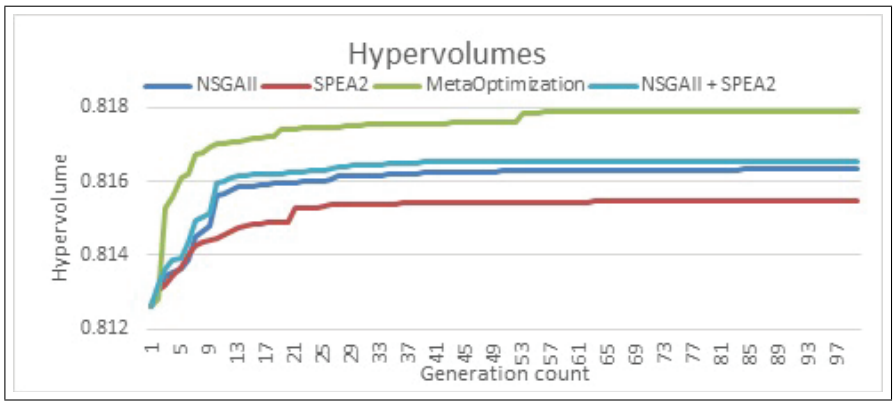

Fig. 3. HV(Meta-Algorithm)>HV(NSGA-II U SPEA2).

In other words, instead of using a single DSE algorithm at a time and, after that, successively repeating the same experiment with other DSE algorithms for finding better possible configurations, a meta-optimization function is introduced that exploits multiple DSE algorithms concurrently, to find Pareto optimal solutions with better convergence speed and solutions' diversity. As we already pointed out, at this moment we have used two multi-objective optimization evolutionary algorithms: NSGA-II and SPEA2. One advantage of our meta-optimization approach consists in the fact that simulation time is approximately equal to the time required for the simulation using just a single DSE algorithm. For meta-optimization implementation, a layer was introduced in FADSE, without making many changes in its existing architecture, which takes care of the meta-optimization (see Fig. 1). Based on the two DSE algorithms run-time effectiveness, the weights of the two algorithms (percentages of generated individuals in a certain generation) are dynamically changing during the optimization process. The performance score for each DSE algorithm is computed from the aggregated results of multiple quality indicators, like hyper-volume, coverage, etc. In this

way, for every generation, after the initial random one, the DSE algorithm performing better will offer more individuals to the current master population. This adaptive behavior drives the optimizations towards better solutions.

With the proposed improvements, as a first experiment, we ran FADSE in order to optimize the performance parameters' values of the Grid ALU Processor (GAP) micro-architecture [12]. According to our first results the proposed evolutionary elitist meta-optimization algorithm generates better results than each of the two used DSE algorithms, showing thus the effectiveness of our proposal. A dedicated paper detailing this new original approach will be developed soon. Figure 3 shows, based on the hyper-volume metric (HV), that the meta-optimization algorithm outperforms even the super positioned NSGA-II and SPEA2 multi-objective genetic algorithms', validating therefore the effectiveness of the meta-optimization method.

6. Integrating Response Surface Models

In order to speed up the DSE evaluation process for a given (computer) architecture, researchers developed so called Response Surface Models (RSMs). These models evaluate the architecture's output metrics based on some analytical expressions derived from a training set of simulations (input-parameters values/output values). The obtained analytical expressions (predictors) must approximate in an acceptable manner the input/output mapping function implemented by the architectural simulator. The main advantage of using RSMs is based on the fact that it is more effective to compute an analytical model than accurately simulating a "use-case" (architectural instance + benchmark). Developing some acceptable approximation functions needs a training process (model development) followed by an evaluation process (how accurate are the models). Each RSM algorithm is characterized by two parameters: accuracy and the corresponding number of training examples. It would be ideal if maximum approximation accuracy would be achieved by using a minimum set of training examples. Therefore the main challenge is how we could effectively approximate a discrete function, implemented by an architectural simulator, defined on a vector set (input parameters) and taking values in another vector set (outputs). In order to do this some methods were used in the literature (linear/polynomial regression, interpolation/approximation methods, etc.). A straightforward solution implements a neural network (NN) approach that is effective in such approximation, like it is shown in [13]. NNs represent powerful methods for generalized non-linear regression. The authors are using fully connected feed-forward multi-layer perceptrons with back-propagation learning algorithm.

In another interesting paper [14] the authors use Radial Basis Function (RBF) networks to model and interpolate the simulation output data to unexplored design points for a simulated superscalar micro-architecture having several hardware parameters. The RBF network aggregates the corresponding RBF functions based on a weighted sum of all RBFs responses. Each RBF receives the entire input vector and generates a response. Using data generated by the architectural simulator helps determine the RBF networks' parameters. According to the authors the proposed

non-linear model gives better results than a previously developed linear regression model.

Both these briefly commented works are not developing a multi-objective approach for the RSM problem. Finding RSMs for multi-objective functions is clearly more difficult than from a single output function (a good interpolation for one component might not be good for others). Also they are not considering compiler/application parameter that is an important further challenge.

In [15] the authors develop regression modeling to derive statistical inference models, requiring a small number of sampled design points in a joint microarchitecture-application design space for initial simulation. The so called application-specific developed models predict performance while the regional models predict power consumption. The authors develop some spline functions (piecewise polynomials) for modeling non-linearity. However a native multi-objective approach (Pareto) is missing.

Therefore, integrating advanced Response Surface Models to accelerate the automatic DSE process in our FADSE tool represents another important research challenge for the ACAPS Research Group led by the author.

7. Pareto Set might be seen as a Fuzzy Set

Actually the Pareto set might be considered as a fuzzy set rather than a classical (crisp) set. Each individual would have a certain membership degree to this set. The perfect "equality" (non-dominance) between Pareto set individuals is debatable: however, one individual might have a "superiority degree" compared to another one. How could we calculate such a (normalized) "superiority degree" between any two individuals belonging to Pareto set represents an interesting open question. Based on this measure it would be possible to compute the membership of an individual belonging to the Pareto fuzzy set. Other methods – in order to compute the membership degree – would be possible. How would influence such an approach the multi-objective evolutionary optimization algorithms? A straightforward idea is to eliminate, during the optimization process iterations, the individuals having a "small" membership degree to the Pareto set. This would accelerate the optimization algorithm's convergence but might restrict the individuals' diversity, potentially involving the solutions quality degradation. An optimal trade-off would be found.

8. Related Works

We present here some of the best known design space exploration tools used in Computer Architecture research. M3Explorer tool [16] is a DSE framework that includes many design space exploration algorithms. M3Explorer can use response surface models to accelerate the design space exploration. Another DSE tool is implemented in a form of a website: archexplorer.org [17]. The users can upload their component on the website where it is integrated into a computer system simulator. So, the design might be compared against other designs focused on the same target architecture. Unfortunately, the users do not have any control on the algorithm being

used. NASA [18] is a similar optimization tool. It allows the user to easily integrate his/hers DSE algorithm and offers the possibility to connect to any simulator. Magellan [19] is a DSE tool which is bounded to a certain simulator (SMTSIM). Magellan can perform only single objective DSEs.

There are some works focused on different multi-objective optimization algorithms on computer architecture simulators. An interesting comparison between DSE algorithms is presented by Silvano et al. [16] with the M3Explorer on the NSGA-II and a Particle Swarm Optimization algorithm. Because the design space of their exploration consists of only 9126 possible configurations, they were able to perform an exhaustive evaluation, too. Other researchers did some design space exploration with single objective optimizations: for example in [19] the authors reached good solutions in a short amount of time. Most of the comparisons cited above have been made using metrics that require the knowledge of the optimal solutions. If the design space is small, such an approach is feasible. Unfortunately, in real-world problems the exhaustive simulation of the design space is not possible.

In [20] it is presented an useful methodology for Design Space Exploration in the context of High Level Synthesis for High Performance Computing and embedded systems targeting FPGAs, providing quickly an RTL description of the design under resources constraints. The proposed flow strictly respects user constraints about resource usage and target frequency. Each feasible transformation is considered only once, thus this Design Space Exploration methodology has a low algorithmic complexity. By handling profiling annotations, the DSE focuses on the most critical sections, and the flow presents a good scalability. The developed demonstration tool was able to transparently explore a wide variety of implementations and to converge towards a satisfactory solution in a short time. However, it is acknowledged that the simplicity and straightforwardness of the proposed DSE methodology can lead to sub-optimal designs.

In [21] the authors are focusing on developing a DSE method for mapping of application tasks onto architectural resources of an MPSoC. There are presented two pruning techniques by exploiting specific domain-knowledge related to tasks allocation through a Genetic Algorithm. The domain knowledge envisages to reduce the redundancy present in chromosome representations by removing the symmetry from the design space (1) and to use a new crossover operator that is based on a mapping distance metric that provides a measure of similarity between design points (2). It is shown that the proposed approach is more effective comparing with a classical Genetic Algorithm solution. The implemented domain-knowledge is not considering hardware constraints, hierarchical parameters and rules expressed in fuzzy logic as we already implemented in [1], [2], [10].

9. Conclusions and Further Work Ideas

Developing automatic design space exploration methods dedicated to Computer Architecture multi-objective hardware-software co-optimizations represents a very important present-day research challenge in Computer Engineering domain. Each new

computing system needs such an optimization before being implemented. According to our view especially the following aspects are extremely important and we'll further focus on them:

- In order to achieve better convergence speed and solutions' quality, specific domain-knowledge – for each of the target computer system – is necessary to be developed and integrated into the DSE tools. Adequately representing such domain-knowledge is a significant scientific challenge. As far as we know, we were the first ones using fuzzy logic as a method to express computer architecture knowledge into a DSE tool [4], [5], [6].

- Automatically calculating the degrees of logical contradiction for a certain (computing system) domain-knowledge represented using (many) fuzzy logic rules would be of certain scientific interest. This would be helpful in order to avoid solutions quality degradation during the optimization process, due to these (possible too contradictory) design rules.

- Taking into account that there is not a best multi-objective optimization algorithm and due to the enormous time consuming for doing an optimization for a complex system using a certain DSE algorithm, effective meta-optimization methods are needed. We just open this field of research in Computer Architecture domain.

- Integrating some advanced effective Response Surface Models to accelerate the automatic DSE processes represents another important fertile research challenge.

- An original approach based on the "Pareto-fuzzy set" concept might be an interesting rationale further idea. How would influence such an approach the multi-objective evolutionary optimization algorithms represents an important research question.

References

[1] CALBOREAN H., VINŢAN L., *An Automatic Design Space Exploration Framework for Multicore Architecture Optimizations*, Proceedings of The 9th IEEE RoEduNet International Conference, pp. 202–207, ISSN 2068-1046, Sibiu, June 24–26, 2010.

[2] CALBOREAN H., *Multi-Objective Optimization of Advanced Computer Architectures using Domain-Knowledge*, PhD Thesis, "L. Blaga" University of Sibiu (PhD Supervisor: Prof. L. VINŢAN, PhD), 2011.

[3] JAHR R., CALBOREAN H., VINŢAN L., UNGERER T., *Boosting Design Space Explorations with Existing or Automatically Learned Knowledge*, The 16-th International GI/ITG Conference on Measurement, Modelling and Evaluation of Computing Systems and Dependability and Fault Tolerance (MMB/DFT 2012), Kaiserslautern, Germany, March 19–21, 2012.

[4] JAHR R., CALBOREAN H., VINŢAN L., UNGERER T., *Finding Near-Perfect Parameters for Hardware and Code Optimizations with Automatic Multi-Objective Design Space Explorations*, Concurrency and Computation: Practice and Experience, doi: 10.1002/cpe.2975, John Wiley & Sons, 2012.

[5] GELLERT A., CALBOREAN H., VINŢAN L., FLOREA A., *Multi-Objective Optimizations for a Superscalar Architecture with Selective Value Prediction*, IET Computers & Digital Techniques, United Kingdom, Vol. **6**, Issue 4, 205–213, 2012.

[6] RADU C., MAHBUB MD. S., VINŢAN L., *Developing Domain-Knowledge Evolutionary Algorithms for Network-on-Chip Application Mapping*, Microprocessors and Microsystems, Vol. **37**, Issue 1, pp. 65–78, Elsevier, 2013.

[7] CHIS R., VINŢAN L., *Multi-Objective Hardware-Software Co-Optimization for the SNIPER Multi-Core Simulator*, Proceedings of 10^{th} International Conference on Intelligent Computer Communication and Processing (ICCP 2014), IEEE Computer Society Press, Cluj-Napoca, September 4–6, 2014.

[8] FLOREA A., BUDULECI C. R., CHIS R., GELLERT A., VINŢAN L., *Enhancing the Sniper Simulator with Thermal Measurement*, Proceedings of The 18^{th} International Conference on System Theory, Control and Computing, Sinaia (Romania), October 17–19, 2014.

[9] VINŢAN L., *Degrees of Contradiction for Fuzzy Logic Rules implementing Computer Architecture Ontologies (Grade de contradictie pentru ontologii de domeniu reprezentate prin logici fuzzy)*, Revista Romana de Informatica si Automatica, Editura ICI, Bucuresti, vol. **23**, nr. 3, pp. 23–26, 2013 (available at `http://rria.ici.ro/ria2013_3/art02.pdf`).

[10] CALBOREAN H., JAHR R., UNGERER T., VINŢAN L., *A Comparison of Multi-Objective Algorithms for the Automatic Design Space Exploration of a Superscalar System*, *Advances in Intelligent Control Systems and Computer Science* (Book title). *Advances in Intelligent Systems and Computing* (Series title), Volume **187**, pp. 489–502, Springer Berlin Heidelberg, 2013.

[11] CHIS R., VINŢAN M., VINŢAN L., *Multi-objective DSE Algorithms' Evaluations on Processor Optimization*, Proceedings of 9^{th} International Conference on Intelligent Computer Communication and Processing (ICCP 2013), ISBN 978-1-4799-1493-7, pp. 27–34, IEEE Computer Society Press, Cluj-Napoca, September 5–7, 2013.

[12] UHRIG S., SHEHAN B., JAHR R., UNGERER T., *The Two-dimensional Superscalar GAP Processor Architecture*, International Journal on Advances in Systems and Measurements, **3**, pp. 71–81, 2010.

[13] IPEK E. *et al.*, *Efficiently Exploring Architectural Design Spaces via Predictive Modeling*, Proceedings of ASPLOS – XII, October 2006.

[14] JOSEPH P. J. *et al.*, *A Predictive Performance Model for Superscalar Processors*,Proceedings of the 39^{th} Annual IEEE/ACM International Symposium on Microarchitecture, pp. 161–170, Washington DC, USA, 2006.

[15] LEE B. C., BROOKS D. M., *Accurate and effcient regression modeling for microarchitectural performance and power prediction*, Proceedings of the 12^{th} International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS), **40**(5):185–194, 2006.

[16] SILVANO C. *et al.*, *MULTICUBE: Multi-Objective Design Space Exploration of Multi-Core Architectures*, Proceedings of the 2010 IEEE Annual Symposium on VLSI, 2010, pp. 488–493.

[17] DESMET V., GIRBAL S., TEMAM O., FRANCE B. F., *Archexplorer. org: Joint compiler/hardware exploration for fair comparison of architectures*, Proc. of the 6^{th} HiPEAC Industrial Workshop, Paris, France, November 2008.

[18] JIA Z. J. *et al.*, *NASA: A generic infrastructure for system-level MP-SoC design space exploration*, Embedded Systems for Real-Time Multimedia (ESTIMedia), 2010 8^{th} IEEE Workshop on, 2010, pp. 41–50.

[19] KANG S., KUMAR R., *Magellan: a search and machine learning-based framework for fast multi-core design space exploration and optimization*, in *Proceedings of the conference on Design, automation and test in Europe*, Munich, Germany, 2008, pp. 1432–1437.

[20] PROST-BOUCLE A., MULLER O., ROUSSEAU F., *Fast and Standalone Design Space Exploration for High-Level Synthesis under Resource Constraints*, Journal of Systems Architecture (2013), doi: http://dx.doi.org/10.1016/j.sysarc.2013.10.002

[21] THOMPSON M., PIMENTEL A.D., *Exploiting domain knowledge in system-level MPSoC design space exploration*, J. Syst. Architect. (2013), http://dx.doi.org/10.1016/j.sysarc.2013.05.023