COMPUTING SYSTEMS MULTI-OBJECTIVE OPTIMIZATION USING DOMAIN-KNOWLEDGE

Lucian N. VINȚAN[1,2]

[1]Full Member of the Academy of Technical Sciences of Romania
[2]„Lucian Blaga" University of Sibiu

Abstract: The main aim of this short paper is to point out the advanced computing systems' optimization research developed by *The Advanced Computer Architecture and Processing Systems Research Centre* (ACAPS) from "Lucian Blaga" University of Sibiu, Romania. Multi-objective optimization (performance, power consumption, temperatures, complexity…) of computing systems having many parameters is a very complex problem. Not only the hardware needs to be simulated and evaluated; frequently 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 NP-hard problem. In our optimization research we used a Pareto-based approach by implementing some multi-objective evolutionary algorithms and bio-inspired algorithms belonging to the Particle Swarm Optimization class. These briefly presented Design Space Exploration (DSE) characteristics were implemented into a dedicated software product entitled *Framework for Automatic Design Space Exploration*. 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 rules or other specific restrictions. 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.
Keywords: computing systems, hardware and software co-optimization, multi-objective optimization, genetic algorithms, domain-knowledge, meta-optimization

1. INTRODUCTION

The main aim of this short paper is to point out the advanced computing systems' optimization research developed by The Advanced Computer Architecture and Processing Systems Research Centre from "Lucian Blaga" University of Sibiu (LBUS), Romania, led by the author (see http://acaps.ulbsibiu.ro/index.php/en/). The scope of our present-day Computer Architecture research consists in developing innovative computer architectures and optimizing them. Some of the subsequent research objectives are the following:

- Developing some novel effective micro-architectures (microarchitectures with predictive-speculative processing, Network on Chip, etc.) [1];
- Develop a robust and fast automatic design space exploration framework for *hardware-software* optimization of complex computer systems [2];
- Research how domain-knowledge could be represented and integrated into the *Design Space Exploration* algorithms [3, 4];
- Quantify domain-knowledge impact on the DSE process;
- Evaluate & compare different multi-objective DSE algorithms [5];
- Meta-optimization (adaptive selection of DSE algorithms), etc.

2 COMPUTING SYSTEMS MULTI-OBJECTIVE OPTIMIZATION IMPLEMENTED METHODS

Most real-world scenarios that require optimizations do need more than one objective to be maximized or minimized. Often these objectives come into conflict with each other and the optimization of these problems poses difficulties. Evolutionary algorithms have been used in this field for more than a decade because they have the characteristics of being able to overcome the shortcomings of the conflicting objectives. Evolutionary multi-objective optimization is nowadays a topic that combines classical multiple criteria decision making and the field of evolutionary computing. In recent years almost all micro-architectures are multicore because they are better than single core architectures in the performance/power ratio. (Even if the instruction level parallelism – exploited by out of order superscalar and Simultaneous Multithreaded Processors - is quite neglected by the present-day researches in author's opinion.) New GPUs integrate hundreds and thousands of processors (NVIDIA GeForce GTX 690 has CUDA cores). These new architectures bring new research challenges. Hardware architectures have also to be optimized in parallel with the target applications (hardware-software co-design). An extremely huge search space is being created by all these processor configurations and evaluating the entire space is not feasible. In the current processor design methodology an experienced designer manually chooses parameters for the specific architecture. A better solution could be using the experience of the designer and guiding an automated design space exploration software towards the best solution.

Multi-objective optimization (performance, power consumption, temperatures, complexity…) of computing systems having many parameters is a very complex problem (NP-hard). Not only the hardware needs to be simulated and evaluated; frequently 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 complex problem. In our optimization research we used a Pareto-based approach by implementing some multi-objective evolutionary algorithms and bio-inspired algorithms belonging to the Particle Swarm Optimization class.The solutions' quality is usually evaluated through some 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 multi - dimensional space), etc.

These briefly presented Design Space Exploration characteristics were implemented by us in adedicated software product entitled*Framework for Automatic Design Space Exploration* (*FADSE*). The software tool is publicly available (see https://code.google.com/p/fadse/). It includes many state-of-the-art multi-objective DSE algorithms. 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 computer simulators.

The search time is reduced through a database integration (we achieved up to 67% simulations results' reuse). The DSE multi-objective implemented evolutionary algorithms were adapted in order to allow distributed evaluations on LANs or High Performance Computers. Also FADSE implements some reliability techniques: if clients do not respond, networks fail, 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.

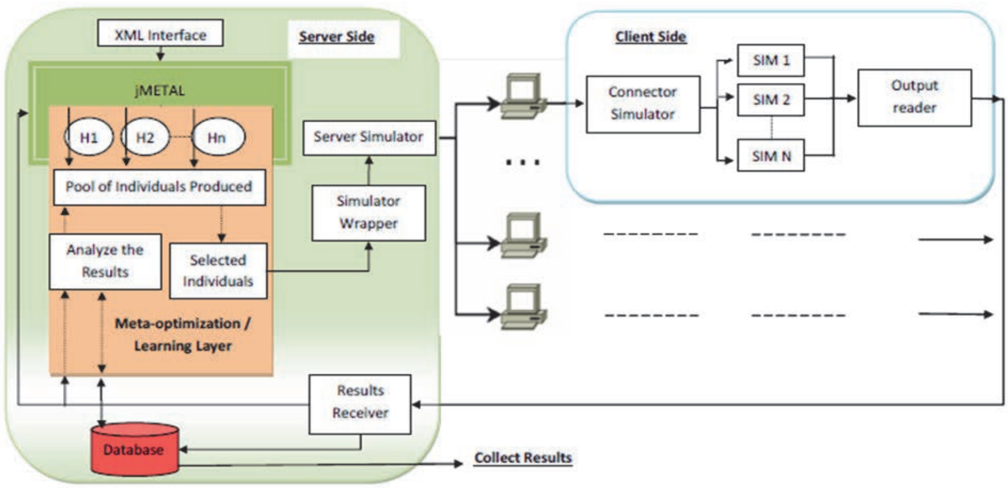

Fig. 1. Structure of FADSE with meta-optimization layer (distributed version).

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 rules or other specific restrictions. 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.Some CPU Fuzzy Logic Rules examples, used by us in optimizing a superscalar architecture augmented through Load value Prediction Methods, are the following:

IF IL1Cache_Size IS *small* AND DL1Cache_Size IS *small* THEN UL2Cache_size IS *big*,
IF IL1Cache_Size IS *big* AND DL1Cache_Size IS *big* THEN UL2Cache_size IS *small*, etc.

For each of the fuzzy logic rules 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);*
   1.1. *If a fuzzy rule exists for the current parameter, having it as a consequent;*
      1.1.1. *Compute COG of this parameter taking into consideration the current values of the other parameters;*
      1.1.2. *Compute the membership* $\mu(COG)$ *value of the COG;*
      1.1.3. *Generate a pseudo-random number between 0 and 1;*
      1.1.4. *If the previously generated pseudo-random number is smaller than "fuzzy probability";*
         1.1.4.1. Current parameter is set to a value equal with COG;
      1.1.5. *Jump to next iteration;*
   1.2. *Otherwise (do bit flip mutation);*
      1.2.1. *Generate a pseudo-random number between 0 and 1;*
      1.2.2. *If the previously generated pseudo-random number is smaller than the probability of mutation;*
         1.2.2.1. Change the current parameter to a random value;
      1.2.3. *Jump to next iteration;*
2. *STOP.*

Fig. 2. The New Mutation Genetic Operator.

Automatically calculating degrees of contradiction between fuzzy logic rules is another scientific objective for us [6]. 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. 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 fuzzy logic rules would be of interest, too. Particularly we are interested to develop and integrate such a tool in our developed tool called*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 Computer Architecture domain micro-ontology. Considering a set of fuzzy logic rules implementing a specific (computer architecture) domain-knowledge it would be useful to automatically reduce the degree of contradiction, if it is over a certain threshold by deleting/modifying some rules, etc.

The Pareto set might be considered as a fuzzy set rather than a classical set. The "equality" between Pareto set individuals is, however, debatable: one individual might have a "superiority degree" compared to another one. How could we calculate such a "superiority degree" represents an interesting open question, at least for us. Based on this measure it would be possible to compute the membership of a Pareto set individual. Other methods - in order to compute the membership degree - would be possible, too. How would influence such an approach the multi-objective genetic optimization algorithms? A first idea is to eliminate, during the optimization process iterations, the individuals having a relative small membership degree to the Pareto set. This might accelerate the optimization algorithm's convergence but might restrict the individuals' diversity during the successive generations, and, as a consequence, it mightinvolve solutions quality degradation. An optimal trade-off would be found.

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). As a consequence, through a joint research with Dr. Muhammad Ali Ismail fromNED University of Engineering and

Technology, Pakistan, during his post-docstage at ACAPShaving the author as scientific supervisor (2014), we developed a new abstraction level in FADSE, called meta-optimization. It acts over the domain-knowledge and DSE algorithms (meta-heuristic) levels. Meta-optimizationsearches within a 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. Developing some effective adaptive meta-optimization algorithms is an important scientific challenge for us.According to our first experiences meta-optimization is 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 meta-optimization algorithm. The adaptive algorithm's flow is presented in the next figure. With the proposed improvements, as a first experiment, we ran FADSE in order to optimize the performance parameters of the Grid ALU Processor (GAP) micro-architecture [7]. Figure 4 shows, based on the hypervolume metric, that the meta-optimization algorithm (Algorithm 2) outperforms both NSGA-II and SPEA2multi-objective genetic algorithms, validating therefore the effectivenes of the meta-optimization method.

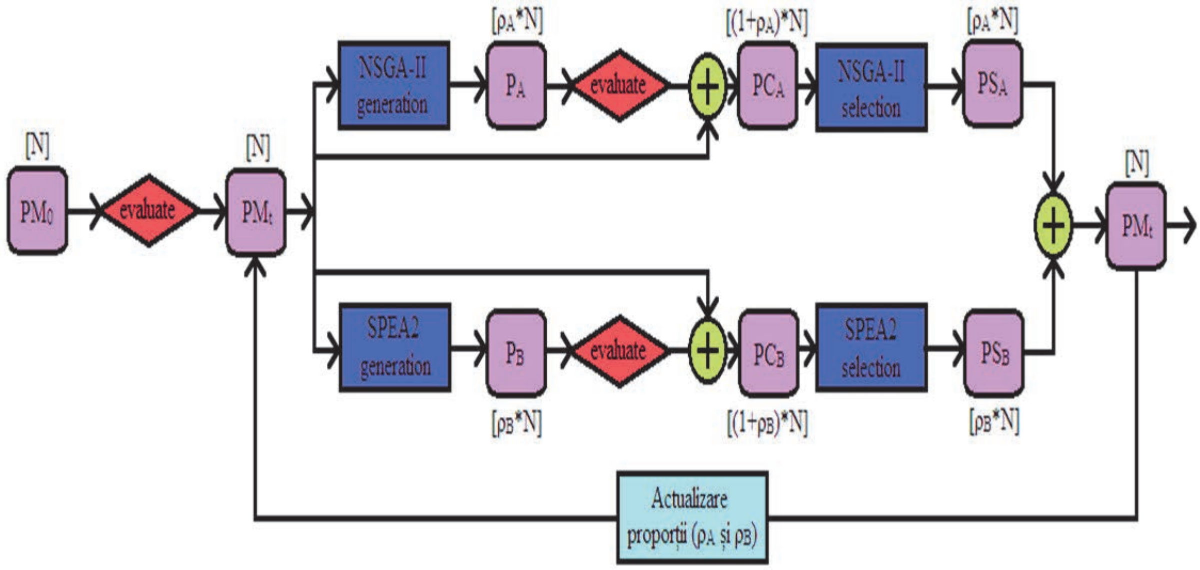

Fig. 3. Adaptive Meta-Optimization Algorithm.

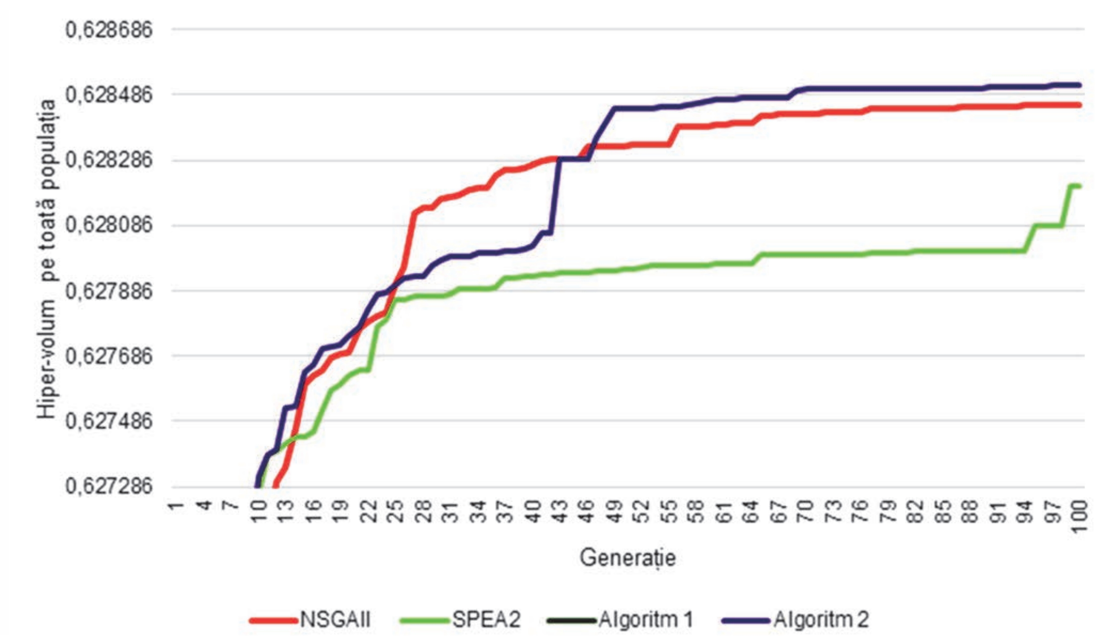

Fig. 4. Hyper-Volume Comparisons for GAP (meta)-optimizations.

3. CONCLUSIONS

We already successfully optimized with FADSE some computer simulators like GAP/GAPtimize (developed at Augsburg University by Professor Theo Ungerer's research group), M-Sim 2 simulator improved by us with a Selective Load Value Predictor, UniMap (our developed NoC simulator), M-Sim 3 multicore simulator, Sniper multicore simulator (with objectives Energy, CPI, Integration Area, Temperature) [8], etc. For the first three projects we already developed and implemented specific effective domain-knowledge in FADSE. Using this knowledge, both the solutions' quality and the convergence speed were improved. Now we are thinking about some multicore systems specific domain-knowledge (for example related to Sniper Multicore Simulator) using advanced knowledge representation methods(fuzzy logic rules, semantic nets, conceptual graphs, etc.). Also we intend to integrate Response Surface Models in FADSE for the acceleration of the DSE process (linear regression, spline regression, radial basis function networks, Shepard interpolation, artificial neural networks, etc.).Developing new effective meta-optimization algorithms is another exciting scientific challenge. For example, integrating in the meta-optimization process a DSE algorithm belonging to the Particle Swarm Intelligence class would be useful. Also integrating in FADSE a parameters feature selection methodology, in order to reduce the search space, might be a useful research. We already integrated Hotspot simulator for temperature computations in Sniper simulator [9]. This allows us a 4D space optimization for Sniper Multicore Simulator rather than a 3D space optimization as it is only possible with the standard Sniper simulator. We also successfully used FADSE for optimization of a motor in the acceleration pedal of a car based on a model implemented in Comsol with Matlab, developed by an automotive company; we looked here for the biggest constant force.

REFERENCES

[1] C. Radu, Md. S. Mahbub, L. Vintan, *Developing Domain-Knowledge Evolutionary Algorithms for Network-on-Chip Application Mapping*, Microprocessors and Microsystems, vol. 37, issue 1, pp. 65-78, ISSN: 0141-9331, Elsevier, February 2013

[2] Calborean H., Vintan L.,*An Automatic Design Space Exploration Framework for Multicore Architecture Optimizations*, Proceedings of The 9-th IEEE RoEduNet International Conference, pp. 202-207, Sibiu, June 24-26, 2010

[3] Jahr R., Calborean H., Vintan 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

[4] Á. Gellert, H. Calborean, L. Vintan, A. Florea, *Multi-Objective Optimizations for a Superscalar Architecture with Selective Value Prediction*, IET Computers & Digital Techniques, United Kingdom, Vol. 6, Issue 4, pp. 205-213, 2012

[5] R. Chis, M. Vintan, L. Vintan, *Multi-objective DSE Algorithms' Evaluations on Processor Optimization,* Proceedings of 9th International Conference on Intelligent Computer Communication and Processing, pp. 27-34, IEEE Computer Society Press, Cluj-Napoca, September 5 – 7, 2013

[6] L. Vintan,*Degrees of Contradiction for Fuzzy Logic Rules implementing Computer Architecture Ontologies (Grade de contradictiepentruontologii de domeniureprezentateprinlogici fuzzy),* RevistaRomână de InformaticășiAutomatică, Editura ICI, Bucuresti, vol. 23, nr. 3, pp. 23-26, 2013

[7] Uhrig, B. Shehan, R. Jahr, and T. Ungerer, *A Two-dimensional Superscalar Processor Architecture*,The First International Conference on Future Computational Technologies and Applications, Athens, Greece, 2009

[8] R. Chis, L. Vintan, *Multi-Objective Hardware-Software Co-Optimization for the SNIPER Multi-Core Simulator*, Proceedings of 10th International Conference on Intelligent Computer Communication and Processing, pp. 3-9, IEEE Computer Society Press, Cluj-Napoca, September 4 – 6, 2014

[9] A. Florea, C. R. Buduleci, R. Chis, Á. Gellert, L. Vintan, *Enhancing the Sniper Simulator with Thermal Measurement*, Proceedings of The 18-th International Conference on System Theory, Control and Computing, Sinaia,October 17 - 19, 2014

OPTIMIZAREA MULTI-OBIECTIV A SISTEMELOR DE CALCUL UTILIZÂND CUNOŞTINŢE DE DOMENIU

Lucian N. VINȚAN[1,2]

[1]Membru titular al Academiei de Ştiinţe Tehnice din România
[2]Universitatea ”Lucian Blaga” din Sibiu

Rezumat: Scopul principal al acestei lucrari este de a prezenta in mod succint cercetarile referitoare la dezvoltarea unor arhitecturi de calcul novatoare precum si cele referitoare la optimizarea multi-obiectiv a acestora, desfasurate in *Centrul de cercetari pentru sisteme avansate de calcul* din cadrul Universitatii „Lucian Blaga” din Sibiu. Se insista pe cercetarile referitoare la metodele euristice dezvoltate si implementate in vederea optimizarii multi-obiectiv a unor sisteme avansate de calcul. In acest sens s-a implementat un cadru software complex, care efectueaza optimizarile sistemelor de calcul vizate, in mod automat. Noutatea consta in faptul ca acest sistem este imbunatatit prin cunostinte de domeniu, exprimate inclusiv prin reguli in logici *fuzzy*. Aceste cunostinte imbunatatesc atat convergenta algoritmilor de optimizare de tip *Pareto*, cat si calitatea solutiilor obtinute. De asemenea s-au dezvoltat si implementat algoritmi de meta-optimizare. Acestia au in vedere rularea simultana, pe baze adaptive, a mai multor algoritmi euristici de optimizare. Primele experimente in acest sens sunt optimiste, dovedind cantitativ si calitativ exploatarea sinergica a algoritmilor de optimizare componenti.

Cuvinte cheie: sisteme de calcul, optimizare hardware-software, optimizare multi-obiectiv, algoritmi genetici, meta-optimizare.