

# Robust Optimization of SoC Architectures: A Multi-Scenario Approach

Gianluca Palermo Cristina Silvano Vittorio Zaccaria  
Politecnico di Milano - Dipartimento di Elettronica e Informazione  
{gpalermo, silvano, zaccaria}@elet.polimi.it

## Abstract

*Multimedia and computational intensive applications are more and more pushing towards high-performance, low-power consumption and limited area occupation making the concept of optimality multi-objective. In the context of a design space exploration framework to support the platform-based design approach, we address the problem of robust optimization of a parameterized system-on-chip platform towards a set of objective functions. The concept of robustness deals with the uncertainty of the scenarios for which the system is optimized. The problem of the robust optimization has been treated as a minimization of a geometric mean of the objective functions computed over each uncertainty scenario and it is solved with a multi-criteria decision technique. The resulting robust solution is characterized by a multi-objective optimality over the set of considered scenarios. Experimental results have been gathered from two virtual platforms in a multi-scenario approach.*<sup>1</sup>

## 1 Introduction

Platform-based design represents the most widely used approach to design System-On-Chip (SOC) applications [14]. In this context, parameterized embedded SOC architectures must be optimally tuned to find the best trade-offs in terms of the selected figures of merit (e.g. energy, delay and area) for a given target application. This tuning process is called *Design Space Exploration* (DSE). The overall goal of the DSE phase consists of optimally configure the parameterized SOC platform in terms of system-level requirements depending on the given application.

Generally, the DSE problem can be stated as a multi-objective optimization problem (MOO) which involves the minimization (maximization) of *multiple objectives* making the definition of optimality not unique. The solution of *multi-objective* optimization problems consists of finding the Pareto front associated with the objective functions given in the problem [1].

So far, the literature addresses the DSE minimization problem for a single, specific target application at a time (*reference workload*). However, optimizing the architecture for a single reference application even considering multiple-objectives is often not enough. In fact, the environment in which the architecture is presumed to work can be characterized by several levels of uncertainty in terms of utiliza-

tion scenarios. In this work, we define a *scenario* as the *system workload* consisting of the application and the given input data set. A perturbation of the reference workload can lead to a sub-optimal system behavior. Robust design [8, 6] is a relatively recent discipline which aims at optimizing the target design by taking into account the uncertainty factors to produce designs for which this effect is mitigated, i.e., whose performance is optimal over the set of considered uncertainty scenarios.

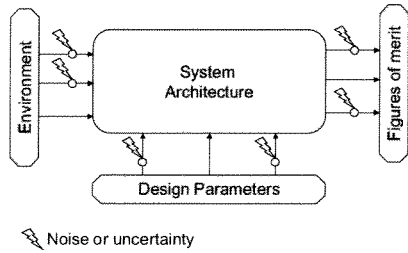
The main contributions of this paper are the following: 1) A robust formulation of the problem of the design space exploration in the context of multiple scenarios/workloads of utilization, 2) A technique for solving the robust optimization problem which aims at finding Pareto solutions, 3) A Multi-Criteria Decision Making strategy to select, among the Pareto solutions, those that are slightly affected by uncertainty factors related to multiple scenarios/workloads, 4) An extended set of experimental results to prove the effectiveness of the proposed techniques on single and multi-processor on-chip architectures. The multi-scenario consists of multimedia kernels (such as DCT, FFT, FIR) and numerical computational kernels.

The paper is structured as follows: Section 2 gives an overview on the DSE problem while Section 3 presents the robust optimization technique proposed in this work. Section 4 shows the experimental results of the robust optimization technique applied to two processor platforms while Section 5 introduces the related work on DSE and robust optimization. Finally, Section 6 highlights the conclusions and gives some insights to the future work.

## 2 The DSE Problem

The IP reuse and platform-reconfigurability approaches are converging into a new design paradigm [14], which is strongly influencing today's automatic system synthesis. In this context, a virtual microprocessor-based architecture can be easily extended and customized for a particular application, enabling a quick, low-risk deployment. More specifically, pre-verified components belonging to a specific library are instantiated and sized to meet specific constraints on the target application domain. However, the space of configurations (or "*design space*") of each of the system components can be very large. Considering a multi-processor SoC, a reasonable set of architectural parameters can be the number of processor cores, the number of parallel issues per core, the number of levels in the memory hierarchy and positioning (on-chip, off-chip), cache configuration parameters (cache size, block size, associativity, unified vs

<sup>1</sup>This work was supported in part by the EC under grant MULTICUBE FP7-216693



**Figure 1. Representation of the optimization scenario with noise/uncertainty.**

split data/instruction caches, at any hierarchy level, etc.), bus/network topologies and channel width. Compiler-level parameters (e.g., loop unrolling, software pipelining etc.) can also be included in the design space definition.

## 2.1 Multiple Objective Optimization

The general DSE problem involves the minimization of *multiple objectives* (such as task latency, task energy consumption, overall system area, etc.) making the definition of optimality not unique [13]. In fact, a system configuration which is the best from the performance point of view, can be the worst in terms of power consumption and vice-versa.

The optimization goal consists of minimizing a target function, i.e.:

$$\min_{\vec{x} \in X} \vec{f} = \vec{f}(\vec{x}) = (f_1(\vec{x}), f_2(\vec{x}), \dots, f_m(\vec{x})) \quad (1)$$

subject to the constraints:

$$\vec{e}(\vec{x}) = (e_1(\vec{x}), e_2(\vec{x}), \dots, e_k(\vec{x})) \leq 0$$

where  $\vec{x}$  is the design vector,  $\vec{f}$  is the objective vector,  $X$  is a subset of  $\mathbb{N}_0^n$  (denoted as  $n$ -dimensional design space).

In this paper, we only consider unconstrained optimization problems, leaving robust constrained multiple-objective optimization as future work.

Given problem 1, to compare two solutions  $\vec{f}$  and  $\vec{g}$  we use the concept of *dominance*. If  $\vec{f}$  is no worse for all objectives than  $\vec{g}$  and better than for at least one objective, we say that  $\vec{f}$  *dominates*  $\vec{g}$  or  $\vec{f} \prec \vec{g}$ .

Formally,  $\vec{f} \prec \vec{g}$  iff:

$$f_i \leq g_i, \forall i = 1, \dots, m \text{ and} \\ f_i < g_i \text{ for at least one } i$$

Dominance in the objective space can be extended to the parameter space that is  $a \prec b$  iff  $\vec{f}(a) \prec \vec{f}(b)$ .

We define the set  $X_p$  of solutions not dominated by any other solution as the *Pareto-optimal Set* while the corresponding set  $Y_p = \vec{f}(X_p)$  is called the *Pareto-optimal Front*. The Pareto-optimal set is the actual solution of the multi-objective optimization problem.

After the Pareto set has been found, a Multi-Criteria Decision Making technique [12] is used to select one of

the Pareto-optimal solutions. According to [10] among the most used techniques in literature we can find Goal programming, weighted objective functions, multi-attribute utility theory and Analytic Hierarchy Procedure (AHP) [12, 2]. The aim of these techniques is to transform the multi-criteria problem into a single objective problem, which is essentially a weighted average of the objective functions or a measure of the distance with respect to the target goals. In this work, we propose a modified weighted geometric average as a criterion for ranking and choosing one of the Pareto-optimal solutions to the architectural multi-objective exploration problem.

## 2.2 Robust optimization

Robust design [8, 6] is a relatively new discipline which aims at optimizing the target design by taking into account the uncertainty factors to produce designs whose performance is optimal over the set of considered uncertainty scenarios. A comprehensive review of robust optimization can be found in [8], where a classification of the uncertainty factors involved into system optimization is presented. According to [8], let us consider the System Architecture shown in Figure 1, where we classify uncertainties appearing during the design process of the target architecture. From the designer point of view, the target architecture should be designed in order to generate the desired *figures of merit* which are controlled by the current configuration of the system (the *Design Parameters*) as well as the current environment conditions. In real-world system design, several uncertainties related to different environmental conditions, experimental errors and implementation errors should be introduced in the optimization problem. Uncertainties can be classified as follows:

- *Type I variations*. These uncertainties come from the environment in which the system operates. In our case these can be due to changing scenario/workloads applied to the system.
- *Type II variations*. These uncertainties come from tolerances in the physical implementation of the current system configurations and are represented by variations  $\delta$  of the design parameters. In this work we will not consider this type of variation, assuming that the production system is able to exactly realize the design configuration chosen by the designer.
- *Variations of the figures of merit*. These uncertainties are currently due to the imprecision in the system evaluation of the figures of merit. This includes simulation accuracy and measurement errors.

In general the real system function  $f$  to be optimized is different from the observed  $f_o$ :

$$f_o = \sigma[f(x + \delta, \xi)]$$

where  $\sigma$  is a functional that includes the variations on the figures of merit,  $\delta$  represents the type II variations and  $\xi$  represents the type I variations.

This paper represents the first step towards the robust optimization for the system level design by addressing type I uncertainties as *scenarios uncertainty*. Scenario-based uncertainties have been introduced in [16] and will be the basis of the work presented here.

### 3 The proposed methodology

Robust Optimization represents a framework for modeling optimization problems in which some of the information is uncertain and it is only known to belong to some uncertainty sets. Studies have revealed [6] that nominal solutions to the problem stated in Equation 1 can be severely affected by small perturbations on the uncertain factors and that Robust Optimization methodologies can be successfully used to overcome this phenomenon. The deterministic minimization problem given by Equation 1 can be expressed in terms of type I variations:

$$\min_{x \in X} \vec{y} = \vec{f}(\vec{x}, \xi), \forall \xi \in U \quad (2)$$

where the set  $U$  is called the *uncertainty set* and represents the possible values of the variation variable  $\xi$ .

In our architectural optimization approach,  $\vec{x}$  is the vector of the architectural parameters to be tuned during the optimization flow, while the *uncertainty set*  $U$  is discrete and composed of several instances  $\xi$  each one defined as *scenario*. More in detail, in this paper we are addressing the robustness of the system with respect to the following case of *scenario uncertainty*, i.e.:

- The workload of the architecture is uncertain or inexact and cannot be represented by a single reference application;
- The data set associated with a single application is variable/uncertain/inexact.

According to [15], problem 2 is often unsolvable and, in general, it is negotiated with less constrained problems such as the *robust counterpart problem* or the minimization of statistical moments such as the *mean value* of  $f$ , over the probability distribution of  $\xi$ .

First of all, we formulate the following robust architectural optimization problem: find a system configuration which minimizes the average value of the objective functions in each single scenario of uncertainty:

$$\min_{x \in X} E[\vec{f}(x, \xi) | \xi] \quad (3)$$

The solution  $x_o$  is *robust* because considering  $x_k$  as the optimal configuration for the specific workload  $\xi_k$ , the following quantity

$$\vec{f}(x_o, \xi_k) - \vec{f}(x_k, \xi_k) \quad (4)$$

is minimal on average for each workload  $\xi_k \in U$ .

It is worth noting that the Problem 3 is a multi-objective problem that generates a Pareto surface of solutions. Given the Pareto surface, a trade-off or constraint-based analysis on the objective functions can be performed to select the most appropriate robust architectural configuration. Often, a Decision Making Criterion (DMC) is applied to select the optimal configuration. One of the contribution of this work is to introduce a DMC which takes into account the essential characteristics of the objective functions related to architectural exploration.

We must note that the simple 'minimization of arithmetic average' has some drawbacks. In fact we are considering

several figures of merit which can differ of some orders of magnitude; if we consider as a figure of merit the *execution cycles* of a task, this can have very different values depending on the task executed. The same issue holds for the energy. One could consider normalized values with respect to the number of instructions (*cycles per instruction* or *CPI*) but these can have even reverse meaning with respect to the real execution time. To solve this problem we focus on the concept of *geometric mean*. A geometric mean, unlike an arithmetic mean, tends to dampen the effect of very high or low values, which might bias the mean if a straight average (arithmetic mean) were calculated.

In order to make the problem as general as possible, we consider the case when each scenario is executed with a different probability. Thus, in the computation of the geometric mean, we introduce the probability  $\alpha_k$  of execution of each scenario  $\xi_k$ .

Based on the previous considerations, we introduce a two-step optimization algorithm which is structured as follows:

#### 1) Formulation of the robust problem as a MOO problem

We perform the minimization of a weighted geometric average of the figures of merit as follows (note that the overall exponent  $1/|U|$  can be neglected from the minimization problem):

$$\min_{x \in X} \prod_{\xi_k \in U} f_i^{\alpha_k}(\vec{x}, \xi_k), 1 \leq i \leq m \quad (5)$$

This problem 5 is in turn a multiple-objective problem whose solution is a Pareto set  $X_p$ . The solution of this problem is computationally unfeasible by using a full search approach since the number of points to be evaluated results equal to the product of the size of the architectural design space ( $|\vec{x}|$ ) and the number of analyzed scenarios ( $|U|$ ). In this paper, we use a Multi-Objective Simulated Annealing heuristic (MOSA) [19] to efficiently derive an approximated Pareto-front for the target problem. As shown in [17], the MOSA meta-heuristic provides a good approximation of the Pareto front (2% of average distance from the actual Pareto front) by evaluating less than 1% of the feasible configurations.

Several DMCs can be applied to the Pareto-front to select one optimal configuration. In the following step, we introduce a DMC which takes into account the variability of the robust objective functions characteristic of architectural exploration.

2) *Selection of the optimum point* A Multi-Criteria Decision Making Criterion (DMC) is defined for selecting, among the Pareto points related to Problem 5, the final robust architecture. To simplify this decision process, we apply again a minimization of the *geometric mean* of the product of the average metrics found for each configuration  $x$ . More formally, we look for configurations  $\vec{x}$  in the Pareto set  $X_p$  which solve the following problem:

$$\min_{\vec{x} \in X_p} \prod_{\xi_k \in U} \prod_{i=1 \dots m} f_i^{\alpha_k * \beta_i}(\vec{x}, \xi_k) \quad (6)$$

where  $\beta_i$  is an inverse measure of the ranking chosen by the Decision Maker [12] for each objective function. As an example, when we consider objective functions like energy and delay spent by an application with a probability

of execution  $\alpha_k = 1$  and equal ranking  $\beta_i = 1$ , Equation 6 becomes a formalization of the minimization of the well known Energy-Delay Product (EDP). We remark that solution of problem 6 is found after the Pareto surface has been generated. Thus there is no need for analyzing the entire design space, but only the Pareto surface.

#### 4 Experimental results

In this section we show some results of the application of our optimization methodology to the design space exploration of two SoC platforms: 1) a virtual superscalar architecture model based on the Watch simulator [9] and 2) a virtual multiprocessor SoC model based on the SESC simulator [18]. The experimental results have been carried out on a multi-scenario approach composed of a mixed workload spanning from multimedia application kernels (DCT, FFT and FIR) and numerical computational intensive kernels.

In general, a multimedia architecture based on a superscalar processor is composed of many parameters, so that the design space to explore is quite large. Our analysis has been focused on those design parameters significantly impacting the performance, the energy consumption and the area at the system-level. In this work, the Watch simulator [9] has been used as target superscalar architectural simulator providing a dynamic profiling of energy and delay. The simulator has been extended for providing also the area estimation in  $0.18\mu\text{m}$  CMOS technology. Each instance of the Watch virtual super-scalar architecture has been described in terms of the parameters listed in Table 1 whose minimum and maximum variations are listed in the column 'min' and 'max'. Steps between the minimum and maximum levels are assumed as a power of two. Globally, the architectural design space is composed of 196608 points to be evaluated.

The benchmark set is composed of five applications: a finite impulse response filter, an algorithm implementing the gamma function, a gaussian elimination algorithm, an equation solver and a DCT transform.

The robust problem has been solved by assuming the probability of execution  $\alpha_k = 1, \forall k$  and a decision maker ranking of  $\beta_{energy} = 2, \beta_{delay} = 3$  and  $\beta_{area} = 1$ .

A *nominal configuration* is defined as a configuration of the virtual architecture which is generated by solving Problem 2 for a specific benchmark/scenario  $\xi_k$ :

$$\min_{x \in X} \vec{y} = \vec{f}(x, \xi_k) \quad (7)$$

This still represents a MOO problem to which we applied the same decision making criteria of the robust problem to find an optimal configuration  $c_{opt,k}$ . Table 1 shows the nominal configurations of the architecture for each given scenario (column 3 to 7) and the robust configuration (last column). The configurations vary mostly in terms of the I-cache size, L2-cache size and associativity. The small values of data cache size are due to the introduction of area and energy as objective functions which tend to bias the solution towards low-cost data-cache solutions. The performance gain of increasing the data-cache has been overwhelmed by its costs during the optimization process, even with  $\beta_{delay} = 3$ .

The analysis of the nominal configurations has been carried out as follows; for each nominal configuration  $c$  and

Parameter	Design Space		FIR	GAMMA	GAUSS	QUARCUBE	DCT	ROBUST
	Min	Max						
Icache size	2K	16K	2K	8K	16K	16K	4K	16K
Icache blk. sz	16	32	32	32	32	32	32	32
Icache assoc.	1	2	2	2	2	2	2	2
Dcache size	2K	16K	2K	2K	2K	2K	2K	2K
Dcache blk. sz	16	32	32	32	32	32	32	32
Dcache assoc.	2	4	2	2	2	2	2	2
L2cache size	16K	128K	16K	32K	16K	64K	32K	64K
L2cache blk. sz	32	64	64	64	64	64	64	64
L2cache assoc.	4	8	4	4	8	8	8	4
Issue w.	2	8	4	2	4	2	4	4
Int. ALUs	1	2	2	2	2	2	2	2
Fp. ALUs	1	2	1	1	1	1	1	1
Int. Multiplrs	1	2	1	1	1	1	1	1
Fp. Multiplrs	1	2	1	1	1	2	1	1

Table 1. Design space and nominal configurations for the superscalar platform

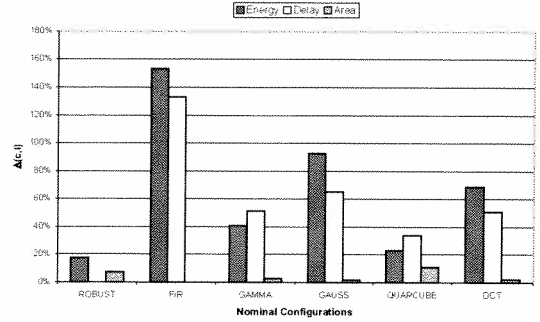


Figure 2.  $\Delta_{c,i}$  for Energy, Delay and Area for the superscalar nominal configurations

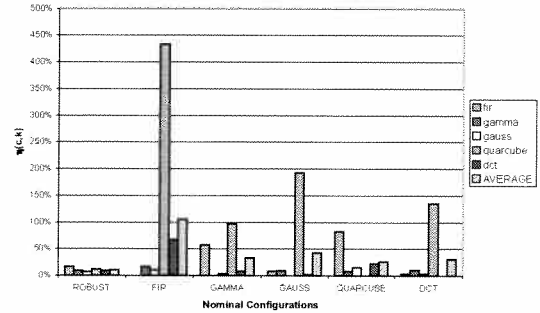


Figure 3. The  $\eta_{ck}$  values for the super-scalar nominal configurations

scenario  $\xi_k$ , we measured the objective function  $f_i(c, \xi_k)$  and we computed the percentage difference:

$$\delta_{ck,i} = \frac{f_i(c, \xi_k) - f_i(c_{opt,k}, \xi_k)}{f_i(c_{opt,k}, \xi_k)} \quad (8)$$

where  $c_{opt,k}$  is the optimal configuration found with Equation 7 for each scenario  $k$ . For each figure of merit  $i$  and nominal configuration  $c$ , Figure 2 shows the maximum percentage  $\Delta_{c,i} = \max_{\forall k} \{\delta_{ck,i}\}$ . The nominal configuration related to the *fir* benchmark is the less robust out of the 6 configurations, presenting a maximum of 150% percentage

Parameter	Design Space		LU			OCEAN			RADIX			FFT			ROBUST
	Min	Max	-n128	-n64	-n32	-n34	-n18	-n10	-n64k	-n32k	-n16k	-m14	-m12	-m10	
Icache size	2K	16K	2K	2K	2K	16K	16K	16K	2K	2K	2K	2K	2K	4K	2K
Dcache size	2K	16K	8K	8K	8K	8K	8K	8K	8K	8K	8K	8K	8K	8K	8K
L2cache size	32K	256K	256K	128K	32K	256K	256K	256K	256K	256K	256K	256K	128K	256K	256K
Icache assoc.	1w	8w	1w	1w	1w	1w	1w	1w	1w	1w	1w	1w	1w	1w	1w
Dcache assoc.	1w	8w	8w	8w	8w	8w	8w	8w	8w	4w	4w	8w	8w	8w	8w
L2cache assoc.	1w	8w	4w	4w	2w	4w	4w	2w	8w	8w	4w	4w	4w	1w	4w
Proc. Issue width	1	8	2	2	2	2	2	2	1	1	1	2	2	2	2
I/D/L2 block size	16B	32B	32B	32B	32B	32B	32B	32B	16B	32B	32B	16B	32B	32B	32B
#Processors	2	16	2	2	2	8	2	2	16	8	8	16	8	4	4

Table 2. Design space and nominal configurations for the multi-processor platform

difference. This maximum corresponds to the case when the *fir* configuration is used in the *quarcube* scenario. The robust configuration found is the one which presents the best  $\Delta_{c,i}$  trade-off among the figures of merit. As a matter of fact, with a slight increase in area, the variation in energy and delay is less than 20% for all the considered scenarios.

Figure 3 shows the percentage differences  $\eta_{ck}$  of the energy-delay-area product defined as follows:

$$\eta_{ck} = \frac{\pi_{ck} - \pi_{c_{opt,k}}}{\pi_{c_{opt,k}}} \quad (9)$$

where  $\pi_{ck} = \prod_i f_i(c, \xi_k)$ . Also in this case, the robust configuration presents the best values of  $\eta$  with an average of 11% and a maximum of 17%. The  $\eta$  values of the *quarcube* configuration are the closest to the robust ones, with a maximum 82%, thus 65% worse than the robust one.

To further validate the methodology, we replicated the analysis performed on a symmetric shared-memory multi-processor architecture with private L2 caches. The simulation tool we used is SESC [18], a fast simulator for chip-multiprocessor architectures with out-of-order processors that is able to provide energy and performance results for a given application. We focus the analysis on the architectural parameters listed in Table 2 providing a design space composed of 131072 points.

The scenarios we used are composed by four applications derived by the SPLASH-2 [21] parallel benchmark suite (FFT, OCEAN, LU, RADIX) with 3 different data-sets for each application.

The robust problem has been solved by assuming the probability of execution  $\alpha_k = 1, \forall k$  and a decision maker ranking of  $\beta_{energy} = 1, \beta_{delay} = 1$  and  $\beta_{area} = 1$ .

Table 2 shows the design space and the nominal configurations for each given workload and the robust configuration. The optimum architectural configurations are different for each application scenarios. Analyzing the architectural parameters, while data cache size and instruction cache associativity are constant respectively to 8KB and 1-way, the number of processors, the instruction/L2 cache size and L2 associativity present a wide variation.

Figure 4 shows the maximum difference  $\Delta_{c,i}$  for each cost function (energy and delay). Since there is a significant variation in the nominal configurations reported in Table 2, the values of  $\Delta_{c,i}$  are bigger than the values found for the super-scalar platform reported in Figure 2. Concerning the overall behavior, while all non-robust configurations (except for FFT -m10) present a small maximum value for only one of the cost functions, the robust one results the best-one in terms of energy-performance trade-off.

Figure 5 shows the  $\eta_{ck}$  for each nominal configuration. The robust architectural configuration shows the best aver-

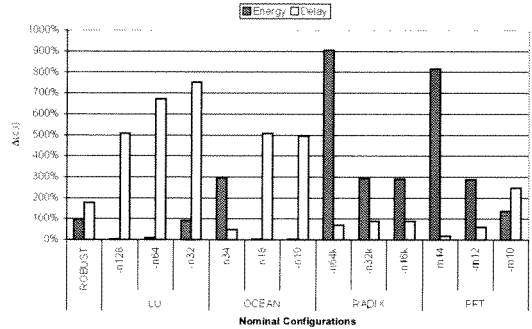


Figure 4.  $\Delta_{c,i}$  for Energy and Delay for the multi-processor nominal configurations

age (53%) and maximum (130%)  $\eta$  values across all the scenarios. Also OCEAN -n34 and FFT -m12-m10 nominal configurations show low average values (respectively 80%, 88% and 104%) but, on the other side, they are characterized by larger maximum values (respectively 303% and 219%).

## 5 Related work

In this section, we introduce the state of the art of design space exploration for processor architectures by highlighting some relevant approaches appeared in the literature so far. We introduce as well some robust optimization techniques.

The traditional approach to DSE consists of formulating it as a multiple-objective combinatorial optimization problem which, when solved, generates a set of Pareto-optimal system configurations. In general, the most trivial approach to determine the Pareto-optimal configurations consists of the comprehensive exploration of the configuration space. However, when the design space is too large, heuristic methods must be adopted to find acceptable near-optimal solutions. Many works have been presented in literature in this direction, evolutionary strategies [3] and decision-theoretic methods [4] are two simple examples.

Concerning the robust optimization, different approaches have been proposed to tackle the uncertainties as presented in Section 2.2. A set of approaches [11, 8, 16, 20] formulate the robust problem in terms of minimization of mean, variance and other higher level moments with respect to the variations involved in the design. The simultaneous minimization is either approached as a multi-objective problem or with aggregating functions combining mean and

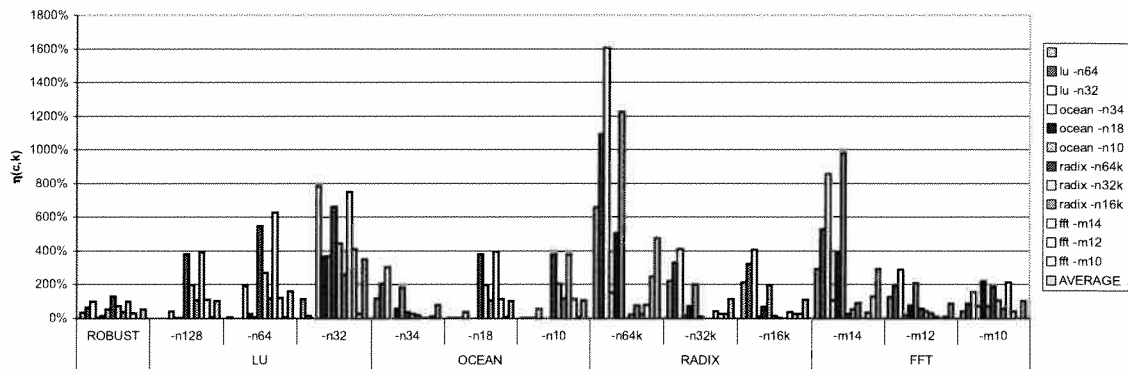


Figure 5. The  $\eta_{ck}$  values for the multi-processor nominal configurations

variance in several ways. Solutions based on the Taguchi method [8, 20] fall into this category, being based on quality function which has a quadratic dependency on both mean and variance.

In [8, 5, 7] a *robust counterpart approach* is proposed where the robust problem is formulated as a minimization of the objective functions in case of maximum variations on the uncertain parameters. This type of problem is also called a *worst-case*, or *regularization* problem.

## 6 Conclusions

In this paper, we proposed a multi-objective design space exploration framework considering a set of different application scenarios. Experimental results on single and multi-processor virtual platforms for multimedia kernels and numerical computational kernels, have shown significant improvements over traditional single scenario approach, therefore proving the effectiveness of the proposed approach.

## References

- [1] A. Aho, J. Hopcroft, and J. Ullman. *Data Structures and Algorithms*. Addison-Wesley, Reading, MA, USA, 1983.
- [2] F. Akhavi and C. Hayes. A comparison of two multi-criteria decision-making techniques. *IEEE International Conference on Systems, Man and Cybernetics*, 1:956–961, Jan 2003.
- [3] G. Ascia, V. Catania, and M. Palesi. A multi-objective genetic approach for system level exploration in parameterized system-on-chip. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 24(4):635–645, 2005.
- [4] G. Beltrame, D. Bruschi, D. Sciuto, and C. Silvano. Decision-theoretic exploration of multiprocessor platforms. In *Proceedings of CODES+ISSS '06*, pages 205–210, 2006.
- [5] A. Ben-Tal and A. Nemirovski. Robust convex optimization. *Mathematics of Operations Research*, 23(4):769–805, Jan 1998.
- [6] A. Ben-Tal and A. Nemirovski. Robust optimization - methodology and applications. *Mathematical Programming*, B92:453–480, 2002.
- [7] D. Bertsimas, O. Nohadani, and K. Teo. Robust nonconvex optimization for simulation-based problems. *Optimization Online*, Jan 2007.
- [8] H. Beyer and B. Sendhoff. Robust optimization—a comprehensive survey. *Computer Methods in Applied Mechanics and Engineering*, 196(33-34):3190–3218, Jan 2007.
- [9] D. Brooks, V. Tiwari, and M. Martonosi. Wattch: a framework for architectural-level power analysis and optimizations. In *Proceedings ISCA 2000: International Symposium on Computer Architecture*, pages 83–94, 2000.
- [10] D. Coit, T. Jin, and N. Wattanapongsakorn. System optimization with component reliability estimation uncertainty: a multi-criteria approach. *Reliability, IEEE Transactions on*, 53(3):369–380, Jan 2004.
- [11] K. Deb and H. Gupta. Introducing robustness in multi-objective optimization. *Evol. Comput.*, 14(4):463–494, 2006.
- [12] J. Dyer, P. Fishburn, R. Steuer, J. Wallenius, and et al. Multiple criteria decision making, multiattribute utility theory: The next ten years. *Management Science*, 38(5), Jan 1992.
- [13] Hwang, C. L. and Masud, A. S. M. . *Multiple Objective Decision Making – Methods and Applications: A State-of-the-Art Survey*, volume 164. Lecture Notes in Economics and Mathematical Systems, Springer-Verlag, 1979.
- [14] K. Keutzer, S. Malik, A. R. Newton, J. Rabaey, and A. Sangiovanni-Vincentelli. System level design: Orthogonalization of concerns and platform-based design. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 19(12):1523–1543, December 2000.
- [15] S. Kugele, L. Watson, and M. Trosset. Interplay of numerical integration with gradient based optimization algorithms for robust design optimization. *SoutheastCon, 2007. Proceedings. IEEE*, pages 472–477, Jan 2007.
- [16] J. Mulvey, R. Vanderbei, and S. Zenios. Robust optimization of large-scale systems. *Operations Research*, 42(2):264–281, Jan 1995.
- [17] G. Palermo, C. Silvano, and V. Zaccaria. An efficient dse methodology for on-chip multiprocessors subject to application-specific constraints. In *Proceedings of SASP, 2008*.
- [18] J. Renau, B. Fraguera, J. Tuck, W. Liu, M. Prvulovic, L. Ceze, S. Sarangi, P. Sack, K. Strauss, and P. Montesinos. SESC simulator, January 2005. <http://sesc.sourceforge.net>.
- [19] K. Smith, R. Everson, and J. Fieldsend. Dominance measures for multi-objective simulated annealing. *Congress on Evolutionary Computation*, 1:23 – 30, May 2004.
- [20] A. Song, A. Mathur, and K. Pattipati. Design of process parameters using robust design techniques and multiple criteria optimization. *IEEE Transactions on Systems, Man and Cybernetics*, 25(11):1437–1446, Jan 1995.
- [21] S. Woo, M. Ohara, E. Torrie, J. Singh, and A. Gupta. Splash-2 programs: characterization and methodological considerations. *Proceedings of the 22th International Symposium on Computer Architecture*, page 2436, 1995.