

A Correlation-based Design Space Exploration Methodology for Multi-Processor Systems-on-Chip

Giovanni Mariani[†]
Aleksandar Brankovic[†]

Gianluca Palermo[‡]
Jovana Jovic[†]

Vittorio Zaccaria[‡]
Cristina Silvano[‡]

[†]ALaRI - University of Lugano
Lugano, Switzerland
{marianig, brankova, jovicj}@alari.ch

[‡]Politecnico di Milano
Dipartimento di Elettronica e Informazione
{gpalermo, zaccaria, silvano}@elet.polimi.it

ABSTRACT

Given the increasing complexity of multi-processor systems-on-chip, a wide range of parameters must be tuned to find the best trade-offs in terms of the selected system figures of merit (such as energy, delay and area). This optimization phase is called *Design Space Exploration* (DSE) consisting of a *Multi-Objective Optimization* (MOO) problem. In this paper, we propose an iterative design space exploration methodology exploiting the statistical properties of known system configurations to infer, by means of a correlation-based analysis, the next design points to be analyzed with low-level simulations. In fact, the knowledge of few design points is used to predict the *expected improvement* of unknown configurations. We show that the correlation of the configurations within the multi-processor design space can be modeled successfully with analytical functions and, thus, speed up the overall exploration phase. This makes the proposed methodology a *model-assisted* heuristic that, for the first time, exploits the correlation about architectural configurations to converge to the solution of the multi-objective problem[†].

Categories and Subject Descriptors

J.6 [Computer-Aided Engineering]: Computer-aided design (CAD);
C.3 [Special-Purpose and Application-Based Systems]: Real-time and embedded systems

General Terms

Design

Keywords

Multi-Processor Systems-on-Chip, Design Space Exploration, Response Surface, Kriging

[†]This work was supported by the EC under grant MULTICUBE FP7-216693.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

DAC'10, June 13-18, 2010, Anaheim, California, USA.

Copyright 2010 ACM ACM 978-1-4503-0002-5 /10/06 ...\$10.00.

1. INTRODUCTION

In the last decade, Multi-Processor Systems-on-Chip (MPSoC) have become widely used for general-purpose architectures and are increasingly used for the embedded domain. The platform-based design philosophy represents a dominant paradigm shift for designing optimized architectures and meeting time-to-market constraints. In this context, parametric simulation models of the system are used to accurately calibrate the architecture to meet the target application requirements in terms of performance, battery lifetime and area. This tuning phase is called *Design Space Exploration* (DSE) and it typically consists of solving a multi-objective optimization problem by pruning a large design space of parameters at system and micro-architectural levels.

So far, several heuristic methodologies, such as evolutionary or sensitivity-based algorithms, have been proposed to address this problem, but they are characterized by low efficiency to identify the Pareto front since they treat the overall system as a black-box simulation model. Thus, state-of-the-art techniques incur in long simulation time to evaluate a comprehensive subset of the design space. In fact, simulation time can span several days, depending on the application and platform complexity and the system resources dedicated to the simulation.

In this paper, we propose an iterative design space exploration methodology exploiting the statistical properties of the design space to infer, by means of a correlation-based analytic model, the most promising design points to be analyzed with low-level simulations. In fact, the knowledge of few design points is used to predict the *expected improvement* of unknown configurations and borrows some fundamental results from the *efficient global optimization* (EGO) [1] heuristic.

The correlation between the configurations of the design space is indeed very strong for those configurations which are very similar and decreases with their euclidean distance in the design space. We will show that this information can be combined with a linear model to accurately capture both the *global* and *local* behavior of the system metrics over the design space. The resulting *correlation-enhanced* meta-model is used to carefully select promising configurations and, thus, speed up the overall exploration phase. This makes the proposed methodology a *model-assisted* heuristic that, for the first time, exploits the correlation about architectural configurations to converge to the solution of the multi-objective problem.

The rest of the paper is organized as follows. Section 2 introduces the state-of-the art of multi-processor design space exploration techniques. Section 3 describes the proposed methodology and the correlation-based analytic model while Section 4 reports a set of experimental results on the efficiency of the proposed approach. Finally, Section 5 concludes the paper by highlighting the significant results obtained.

2. BACKGROUND

Several methods have been recently proposed to reduce the multi-objective design space exploration time by using traditional statistical techniques and advanced meta-model assisted optimizations [1–12].

A set of relevant statistical approaches to DSE can be found in [2–5]. In [2, 3] the authors analyze the main effect and interaction effects among the processor architectural parameters by using linear regression for both architecture performance prediction and assessment. In [4] and [5] the authors use advanced Design of Experiments techniques (DoEs), such as Plackett-Burman to address DSE for FPGA and super-scalar microprocessor design respectively.

Concerning meta-model assisted optimization for embedded systems EDA, [6] and [7] represent the most significant contributions appeared so far. In [6] the authors apply a combined Genetic-Fuzzy system approach for power/performance architectural exploration; the approach is based on a Strength Pareto Evolutionary Algorithm (SPEA) merged with a fuzzy-logic rule engine. In [7], the authors present an iterative technique that combines DoEs and response surface models to address the optimal tuning of a multi-processor architecture.

Other important contributions to the field of meta-model assisted optimization theory can be found in [1, 8–12]; in particular, the authors of [8–10] manage to combine simple and approximate models of the system with more expensive simulation techniques.

Meta-model uncertainty information is used for optimization in [1, 11, 12]. In particular, the authors of [11] use meta-models as a pre-selection criterion which excludes from the exploration the less promising configurations. In [1] (extended in [12]), meta-models are used to identify the best set of experiments to be performed to improve the accuracy of the model itself. In both cases, the uncertainty measure provided by statistical meta-model is exploited to trade-off optimization and exploration, leading to improved optimization efficiency.

In this paper, we address meta-model assisted MPSoC DSE by leveraging the existing spatial correlation to create an enhanced meta-model. To this end we extend some optimization theory results as obtained in [1] in order to address the optimization of multiple correlated objectives (such as energy consumption and execution time) for the target multi-processor architecture.

3. THE PROPOSED DESIGN SPACE EXPLORATION METHODOLOGY

In this paper we present an iterative *multi-objective optimization* (MOO) strategy for Multi-processor Systems-on-chip (MPSoC) which leverages the knowledge of a set of known system configurations to explore the design space towards optimal solutions. The optimality of a solution is a trade-off of the expected execution time of an application and its energy consumption.

The target architecture. The proposed design space exploration framework is based on a set of state-of-the-art delay and energy models of a MPSoC. The MPSoC platform is composed of a variable number of homogeneous superscalar processors with private L1 and L2 caches, all configurable at design time. To estimate system-level metrics, we leveraged the SESC [13] simulation tool, a MIPS instruction set simulator (ISS) providing the overall energy $\epsilon(x)$ and execution cycles $\delta(x)$ associated with a single architectural configuration x . The design space parameters are presented in Table 1. Each parameter follows a power of two progression; overall, the number of available architecture configurations is 2^{17} .

To perform the analysis presented in the rest of the paper, we used a set of benchmarks derived from the Stanford Parallel Applications for Shared Memory (SPLASH) [14] suite. The SPLASH suite is organized in two sections: kernels and applications. We se-

Table 1: Design space for the shared-memory multi-processor platform

Parameter	Acronym	Min.	Max.
Number of Processors	pn	2	16
Processor Issue Width	iwidth	1	8
L1 Instruction Cache Size	ics	2K	16K
L1 Data Cache Size	dcs	2K	16K
L2 Cache Size	l2cs	32K	256K
L1 Instruction Cache Associativity	icw	1w	8w
L1 Data Cache Associativity	dcw	1w	8w
L2 Cache Associativity	l2cw	1w	8w
I/D/L2 Block Size	cbs	16	32

lected a partial subset of the suite composed of the following applications: Complex 1D FFT (*fft*), Integer Radix Sort (*radix*), Ocean Simulation (*ocean*), Blocked LU Decomposition (*lu*).

Assumptions and Motivation. The approach proposed in this paper is based on the assumption that a strong spatial correlation is present in the design space. More specifically, we assume that the regularity of the configurable platform allows, to some extent, to identify close dependencies in terms of energy and execution time for configurations which are in a neighborhood of the configuration design space (in terms of euclidean distance).

This assumption can be experimentally verified by analyzing the correlation $\tau(x, x_t)$ between close architectural configuration pairs (x, x_t) . As an example, Figure 1 shows the contour map of the average correlation of the execution time δ projected on a single dimension (parameter 'Number of processors') of the target architecture for both x and x_t (X and Y Axis). As can be seen, $\delta(x)$ and $\delta(x_t)$ have a very high correlation ($\sim 90\%$) even when the number of processors is very different (e.g., $pn[x] = 2$ and $pn[x_t] = 16$).

This information is relevant, because it means that the execution time $\delta(x)$ can be used, to a very high extent, to predict the execution time $\delta(x_t)$.

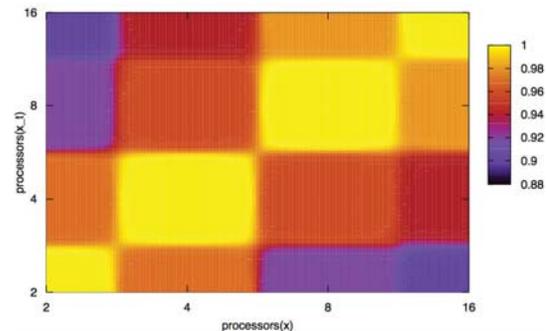


Figure 1: Correlation between configurations with a similar number of processors

In this paper, we will leverage the previous observations to construct an efficient and statistical methodology for identifying optimal configurations with respect to the system level figures of merit $\epsilon(x)$ and $\delta(x)$. The methodology is based on the construction of a *correlation-based* meta-model which is useful for predicting the quality of the configurations without resorting to an architectural simulation. In the proposed meta-model we consider that the euclidean distance on each axis of the design space² has a different influence on the correlation between two design points. Thus the overall correlation between the design points is a weighted function of such a set of distances.

²Every architectural parameter has a corresponding axis in the design space.

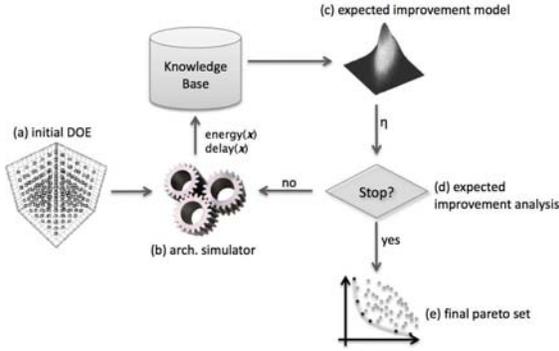


Figure 2: The basic flow of the methodology

3.1 Correlation-based DSE methodology

The main challenge of the design space exploration is the simulation time needed for evaluating a single configuration \mathbf{x} and the large size of the design space. The technique proposed in this paper aims at reducing the overall exploration time by carefully selecting which design point \mathbf{x} to simulate by using correlation information.

The optimization problem. The overall optimization problem is multi-objective and it consists of minimizing the system level energy and execution time simultaneously:

$$\min_{\mathbf{x} \in X} \omega(\mathbf{x}), \quad \omega(\mathbf{x}) = \begin{bmatrix} \epsilon(\mathbf{x}) \\ \delta(\mathbf{x}) \end{bmatrix} \quad (1)$$

where X is the space of parameters as defined in Table 1, while $\epsilon(\mathbf{x})$ and $\delta(\mathbf{x})$ have been defined as the energy consumption and execution time of configuration \mathbf{x} . The solution to this problem is a set of configurations $X_p \subseteq X$ known as Pareto set³.

The proposed optimization flow. Figure 2 shows the basic design flow proposed in this paper. The methodology is, basically, a sequential *design of experiment* heuristic which, starting from an initial random design of experiments (block [a]), interacts iteratively with the architectural simulator (block [b]) to produce the final *approximate* Pareto set (block [e]).

The initial random experiments are useful to perform an initial analysis of both global and local (non-linear) behavior. At each iteration I_i , the algorithm identifies a candidate configuration \mathbf{x}_i and computes its *expected improvement* $\eta(\mathbf{x}_i)$ by means of an *expected improvement analysis* (block [d]). The expected improvement is an aggregate measure of the goodness of the candidate configuration with respect to those visited so far. The expected improvement is computed by means of a correlation-based model (block [c]) which leverages the information contained in the *knowledge-base*, i.e., a database which contains data about the energy and execution time of all the configurations visited so far. All the dominant configurations belonging to the knowledge-base at the end of the iterative process represent the solution to the multi-objective problem as defined in Equation 1.

Reducing the dimension of the problem. In this paper, we address problem of Equation 1 by reducing it to a single objective problem. To this end, we synthesize the vector objective function $\omega(\mathbf{x})$ into a scalar *quality* $\rho(\mathbf{x})$ by defining $\rho(\mathbf{x})$ as the *non-domination rank* of \mathbf{x} . The non-domination rank of a known configuration \mathbf{x} is a measure of how much deeply a configuration \mathbf{x} is nested into sub-optimal (non-Pareto) solutions. Figure 3 shows, for instance, a ranking for a set of known points at iteration i . All the

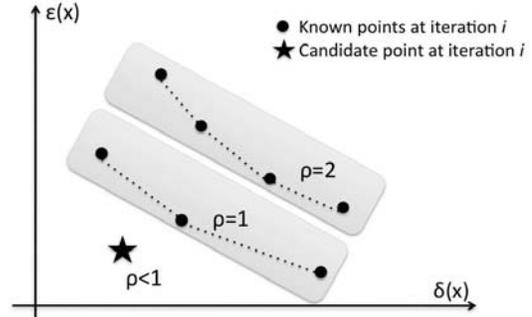


Figure 3: Representation of *domination-rank*

points belonging to the actual Pareto front have $\rho(\mathbf{x}) = 1$. Points with $\rho(\mathbf{x}) = 2$ are those which are dominated only by points with $\rho(\mathbf{x}) = 1$, so the higher $\rho(\mathbf{x})$, the worse is the configuration \mathbf{x} .

In this paper, we assume that a candidate configuration \mathbf{x} to be simulated presents a $\rho(\mathbf{x}) < 1$ with respect to already known points. Basically, the methodology tries to identify the configurations which have $\rho(\mathbf{x}) < 1$ which are likely to belong to the actual Pareto-optimal front when the number of iterations is suitably large.

Dealing with uncertainty. As long as the architecture \mathbf{x} considered at iteration i has not been simulated, its rank $\rho(\mathbf{x})$ can only be predicted with an analytical model $\hat{\rho}(\mathbf{x})$. In particular, in this paper we introduce a *correlation-based* model for $\hat{\rho}(\mathbf{x})$ whose uncertainty is taken into account during the exploration process.

The approach proposed in this paper is inspired by the *efficient global optimization* heuristic as proposed in [1]. The technique builds on the assumption that, even if $\hat{\rho}(\mathbf{x})$ can't be precisely predicted, its probability distribution $\phi_{\hat{\rho}(\mathbf{x})}$ can be approximated. Later on, we will show that *correlation* among configurations can be used to approximate with reasonable accuracy the probability distribution $\phi_{\hat{\rho}(\mathbf{x})}$.

Expected improvement. The target of our methodology is to increase the probability that the selected configuration \mathbf{x} represents an effective improvement with respect to the configurations already simulated in the the previous iterations. We define the improvement as:

$$\chi(\mathbf{x}) = \max[1 - \hat{\rho}(\mathbf{x}), 0] \quad (2)$$

which measures the improvement with respect to the current Pareto front ($\rho=1$) provided by the predicted rank $\hat{\rho}(\mathbf{x})$ at iteration i . The larger $\chi(\mathbf{x})$, the better \mathbf{x} .

Given $\phi_{\hat{\rho}(\mathbf{x})}$, the associated probability distribution $\phi_{\chi(\mathbf{x})}$ can be computed thus we can define the following *expected improvement* measure:

$$\eta(\mathbf{x}) = \mathbf{E}[\chi(\mathbf{x})] \quad (3)$$

which trades off the actual improvement with the uncertainty associated with the distribution $\phi_{\chi(\mathbf{x})}$. More precisely, if the uncertainty is high, $\eta(\mathbf{x})$ is likely to be high and force the proposed heuristic to escape from local minima.

Driving experiments with expected improvement. Assuming that $\phi_{\hat{\rho}(\mathbf{x})}$ is known, the proposed methodology aims at selecting the best simulation candidates \mathbf{x} that possess the highest expected improvement $\eta(\mathbf{x})$. The sequential design of experiments consists of the following steps:

1. Perform an initial random set of simulation. and store simulation results in the *knowledge base* **KB**. The random set of

³We recall that our target problem is a multi-objective problem which has more than a single solution. The set of solutions of a multi-objective problem is called Pareto set.

experiments is used to support both global and local modeling through the target analytic model.

2. Identify the best simulation candidate \mathbf{x} such that its expected improvement is maximum. The identification is performed by using a single objective genetic algorithm by maximizing the expected improvement $\eta(\mathbf{x})$ estimated by using block (c) in Figure 2.
3. Simulate the candidate \mathbf{x} with the architectural simulator and store the result into **KB**.
4. Verify stopping criterion. If the Pareto set contained in **KB** remains unchanged or a maximum number of simulation has been performed the algorithm stops, otherwise restart from step 2.

3.2 Exploiting correlation to model uncertainty

As shown in the previous sections, we assume that a suitable prediction $\hat{\rho}(\mathbf{x})$ can be computed without incurring in the actual simulation of architecture \mathbf{x} . In this paper, we propose to use a *correlation-based* analytic model for our target multi-processor domain. The model (block [c] in Figure 2) takes into account the *correlation* of the unknown point \mathbf{x} with respect to a set of known points (used as a *training set*) to compute the prediction distribution $\phi_{\hat{\rho}(\mathbf{x})}$ and, eventually, the expected improvement $\eta(\mathbf{x})$.

The modeling approach we propose in this paper is based on the *kriging* technique as proposed in [15]. In particular, the prediction $\hat{\rho}(\mathbf{x})$ is modeled as the realization of a random process $\phi_{\hat{\rho}(\mathbf{x})}$ which derives from a Gaussian random field process $\zeta(\mathbf{x})$ and a regression model $\psi(\mathbf{x})$:

$$\hat{\rho}(\mathbf{x}) = \psi(\mathbf{x}) + \zeta(\mathbf{x}) \quad (4)$$

Function $\psi(\mathbf{x})$ is an actual linear polynomial of \mathbf{x} 's elements while the Gaussian process $\zeta(\mathbf{x})$ is a random process with mean zero and a variance $\sigma^2(\mathbf{x})$. However, $\zeta(\mathbf{x})$ can be assumed spatially correlated with the points \mathbf{x}_t already visited by the exploration algorithm.

The literature about *kriging* proposes several correlation functions to choose from; in this paper we adopt the widely used exponential correlation function:

$$\tau(\mathbf{x}, \mathbf{x}_t) = \prod_{i=1}^p e^{(-\theta_i |\mathbf{x}_i - \mathbf{x}_{t,i}|)} \quad (5)$$

where, p is the number of design parameters while θ_i are unknown correlation distance weights. Given a training set of configurations $\langle \rho_t, \mathbf{x}_t \rangle$, both $\psi(\mathbf{x})$ and $\tau(\mathbf{x}, \mathbf{x}_t)$ parameters are estimated with a *maximum likelihood* method. Once $\tau(\mathbf{x}, \mathbf{x}_t)$ is determined, $\zeta(\mathbf{x})$ can be expressed in closed form as a function of \mathbf{x} [1]. The availability of $\hat{\rho}(\mathbf{x})$ allows to compute $\eta(\mathbf{x})$ from Equation 3 by using Bayesian statistic tools.

4. EXPERIMENTAL RESULTS

In this section, we present a set of experimental results obtained by applying the proposed correlation-based design space exploration technique to the problem defined in Section 3. First, we introduce the validation of the rank prediction model and analyze the behavior of the kriging model parameters; we then present a set of optimization results by comparing the Pareto sets found with the proposed methodology with state-of-the-art techniques.

4.1 Rank Model Validation

As described in section 3.2, we use a kriging based model for predicting the *non-dominance rank* $\hat{\rho}(\mathbf{x})$ of non-visited design points.

The model produces an exact interpolation of the training data, thus residual model error for known points is always 0.

Statistical features of the model. To perform the statistical analysis of the prediction error of the adopted kriging model, we use a *leave-one-out cross-validation*: given a set of n simulated architecture configurations and the corresponding *non-dominance ranks* ρ , the model is trained n times over $n - 1$ configurations and the prediction $\hat{\rho}(\mathbf{x})$ for the remaining configuration \mathbf{x} is evaluated against the actual rank $\rho(\mathbf{x})$ to compute the residual $\mathcal{E}(\mathbf{x}) = |\hat{\rho}(\mathbf{x}) - \rho(\mathbf{x})|$.

To provide statistical soundness, the model residual should be normally distributed (i.e., $\mathcal{E}(\mathbf{x}) \sim \mathcal{N}(0, s^2)$). This ensures that the model captures all the systematic behavior of the target architecture by leaving out only *random noise*.

To verify this hypothesis, we compare the distribution of the *studentized residuals* (SR) of the model $\mathcal{E}(\mathbf{x})/s^2$ with an ideal normal distribution $\mathcal{N}(0, 1)$ by means of a Quantile-Quantile (QQ) plot. The rationale behind this kind of plot is that, by construction, all the observations which have a normal distribution are likely to be laid on a straight line.

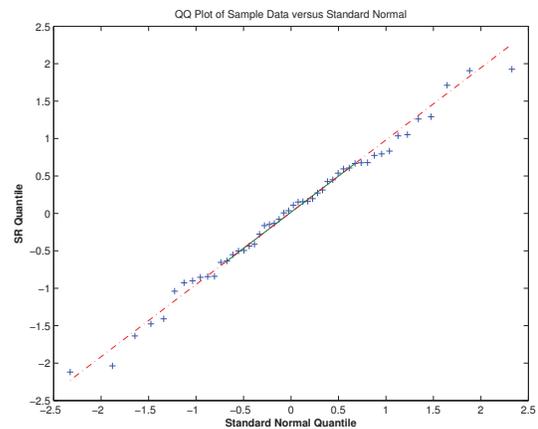


Figure 4: QQ plot for the studentized residuals for the rank predictions.

Figure 4 presents the QQ plot associated with the *radix* use case rank prediction. As can be seen, the residuals of the model are fitted accurately by a straight line, confirming the above hypothesis for this benchmark. The same accuracy has been obtained for the remaining benchmarks, thus we can reasonably assume that the model is statistically sound.

Model Accuracy. To further validate the predictive capabilities of the rank model, we evaluate the average prediction error by varying the number of training samples. To better understand the quality of the kriging model we compared the model with a first order linear regression without interactions, and a three-layer feed-forward artificial neural network (ANN) with 5 neurons in the hidden layers (Figure 5). As can be seen, both the adopted kriging model and the neural model present a better prediction error with respect to the linear model for the considered range of training samples. However, the kriging model error is always lower and improves more rapidly than the neural model error.

4.2 Correlation-Based Analysis

The kriging model of $\hat{\rho}(\mathbf{x})$ (Equation 4) is the superposition of a spatially correlated random process $\zeta(\mathbf{x})$ and a linear regression model $\psi(\mathbf{x})$, thus the fitting process identifies simultaneously the coefficients of both the linear term $\psi(\mathbf{x}) = \alpha_0 + \sum_{i=1, \dots, N} \alpha_i x_i$ and the correlation weights θ_i (Equation 5) associated with $\zeta(\mathbf{x})$.

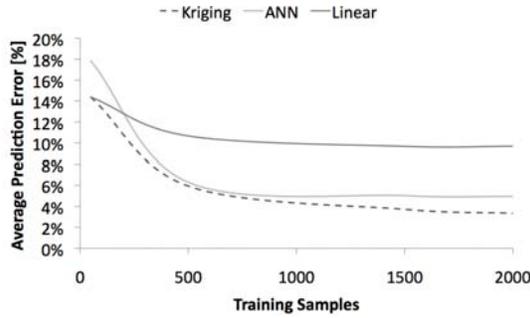


Figure 5: Average prediction error on the ranking model for kriging, artificial neural network and linear regression

While the regression term $\psi(\mathbf{x})$ defines a global trend, the random process $\zeta(\mathbf{x})$ captures local and non-linear behavior of the system by taking into account the distance between points.

Figure 6 presents the linear regression coefficients associated with $\psi(\mathbf{x})$ (a) and correlation weights θ_i (b) for the different design parameters. All the values have been obtained by fitting the kriging model $\hat{\rho}(\mathbf{x})$ with 3000 training configurations from the four different benchmarks defined in Section 3. The fitting has been performed with off-the-shelf statistical tools.

Figure 6(a) presents the regression coefficients for each design parameter (the constant α_0 has been omitted). Since the model is a first order model without interactions, the higher the absolute value of α_i , the higher is the impact of the related design parameter x_i on $\hat{\rho}(\mathbf{x})$. As can be seen, the number of processors (pn), the cache block size (cbs) and the L2 cache size ($l2cs$) represent the design variables with the most important global effects, for all the benchmarks. Parameters with lower main effects are represented by instruction cache size (ics) and the instruction cache associativity (icw). As can be seen, the coefficients related to processor issue width ($iwidth$) and the number of processors (pn) present different signs for the different benchmarks. This implies that the same variation on a design variable may impact differently the linear model $\psi(\mathbf{x})$ (and $\hat{\rho}(\mathbf{x})$) depending on the benchmark used.

Figure 6(b) presents the weights θ_i of the correlation function defined in Equation 5. The higher the values of θ_i , the lower is the contribution to the spatial correlation associated with that design variable. We can note that the most important parameters impacting the spatial correlation of $\hat{\rho}(\mathbf{x})$ are the instruction cache associativity (icw) and the cache block size (cbs). The number of processors (pn) and the instruction cache size (ics) present a lower contribution to the spatial correlation for all the benchmarks. Finally, we can observe a benchmark dependent behavior for the L2 cache size ($l2cs$), data cache associativity (dcw) and processor issue width ($iwidth$).

In summary, we can note that both the linear regression term $\psi(\mathbf{x})$ and the random process $\zeta(\mathbf{x})$ provide non-overlapping information about the system behavior. In fact, Figure 6 shows that some of the design variables may have a big *global* impact but low *local* impact (such as the processor number pn), or low *global* impact and high *local* impact (such as icw).

4.3 Optimization Results

In this section, we compare the performance of the proposed correlation-based algorithm with two state-of-the-art genetic algorithms: a plain NSGA-II [16] and a Meta-model Assisted NSGA-II (MA-NSGA-II), where the basic engine of NSGA-II has been enhanced with a kriging based model [8].

The optimizations have been performed on the target multi-processor platform for the four applications of the SPLASH benchmark suite

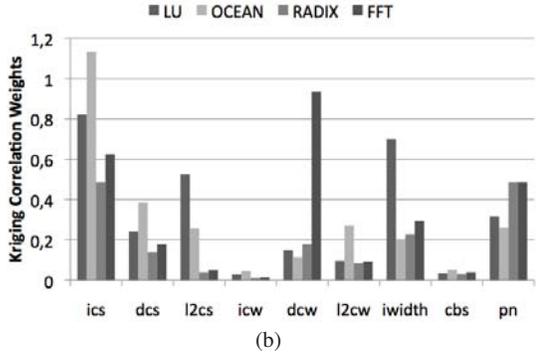
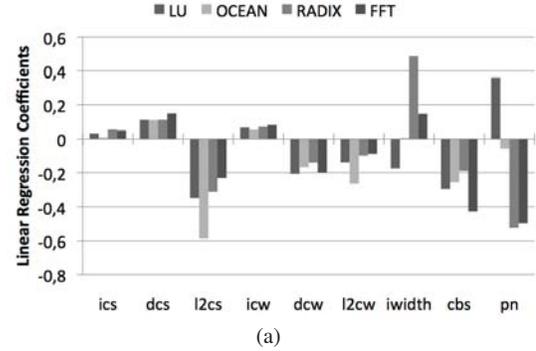


Figure 6: Kriging linear regression coefficients (a) and correlation weights θ_i (b) for the different design parameters

[14]: *fft*, *radix*, *lu*, *ocean*. In this Section we will report the average results.

For fairly compare the different approaches, we first compute the reference exact Pareto solution for all the applications by exhaustively simulating the design space composed of 2^{17} different configurations (Table 1). The performance comparison is then done by following two main assessment criterion:

- **Exploration time.** This criterion corresponds to the evaluation of the total amount of time required by the optimization heuristics to run until a desired quality of the heuristic solution is achieved.
- **Quality of the solution set.** This criterion corresponds to how much the heuristic solutions approximate the exact Pareto set after a fixed amount of time. We measure this figure of merit in terms of *Average Distance from Reference Set* (ADRS) [17], that is essentially a measure of the distance with respect to exact Pareto solution (the lower the ADRS, the better is the approximate solution).

Figure 7(a), shows the exploration speed-up (in terms of simulation time) of the proposed methodology with respect to NSGA-II and MA-NSGA-II, by varying the required ADRS from 0.5 % up to 5 %. Even if a stopping criterion based on an ADRS value is unrealistic, this comparison is enabled thanks to the knowledge of the exact Pareto set. In both cases, the saving is higher than 35% and, over the considered ADRS range, increases as the required ADRS is tightened. The MA-NSGA-II presents a better performance with respect to the plain NSGA because of its kriging based meta-model, but it's still lower than the performance of the proposed heuristic. Figure 7(b) shows the improvement in terms of ADRS when the 3 heuristics are run until a fixed amount of simulations is reached

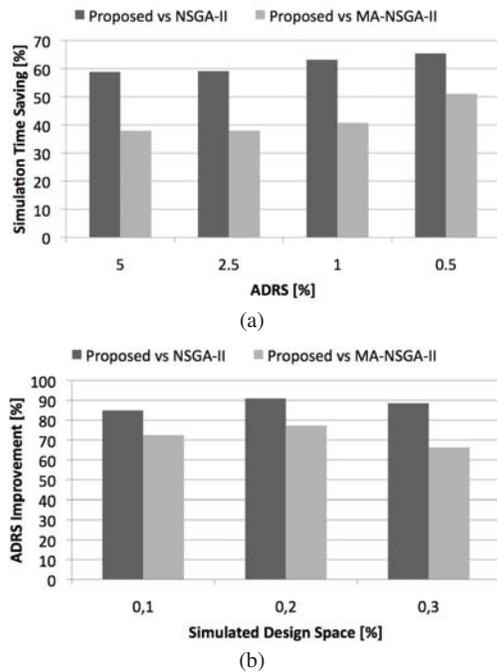


Figure 7: Simulation time saving given fixed values of ADRS (a) and relative ADRS improvement given fixed number of simulations (b): comparison of the proposed method with respect to the plain NSGA-II and MA-NSGA-II

(0.1 % , 0.2 % and 0.3 % of the size of the design space). The ADRS improvement obtained with the proposed solution is almost constant for all the scenarios, being over 80% better than NSGA-II and 60% better than MA-NSGA-II respectively. As a matter of fact, both NSGA-II and MA-NSGA-II suffer from the limited size of simulations used as a constraint. In particular, MA-NSGA-II is better than plain NSGA-II since most of the selected points are predicted without wasting time in simulations. In summary, Figure 7 shows that the proposed method can reduce the exploration effort by managing the meta-model more efficiently than the MA-NSGA-II.

5. CONCLUSIONS

In this paper, we proposed an iterative design space exploration methodology exploiting the correlation properties of multi-processor system configurations to accurately choose the most promising configurations to be analyzed with a low-level simulation. We have shown that the correlation of the configurations within the multi-processor design space can be modeled successfully with analytical functions and, thus, speed up the overall exploration phase up to 65% with respect to state-of-the-art heuristics. This makes the proposed methodology an efficient *model-assisted* heuristic that, for the first time, exploits the correlation about architectural configurations to converge to the solution of the multi-objective problem.

6. REFERENCES

- [1] Donald R. Jones, Matthias Schonlau, and William J. Welch. Efficient global optimization of expensive black-box functions. *J. of Global Optimization*, 13(4):455–492, 1998.
- [2] P. J. Joseph, Kapil Vaswani, and Matthew J. Thazhuthaveetil. Construction and use of linear regression models for

processor performance analysis. In *Symposium on High Performance Computer Architecture*, pages 99–108, Austin, Texas, USA, 2006. IEEE Computer Society.

- [3] Benjamin C. Lee and David M. Brooks. Accurate and efficient regression modeling for microarchitectural performance and power prediction. *SIGOPS Oper. Syst. Rev.*, 40(5):185–194, 2006.
- [4] David Sheldon, Frank Vahid, and Stefano Lonardi. Soft-core processor customization using the design of experiments paradigm. In *DATE '07: Proceedings of the conference on Design, automation and test in Europe*, pages 821–826, 2007.
- [5] Joshua J. Yi, David J. Lilja, and Douglas M. Hawkins. A statistically rigorous approach for improving simulation methodology. In *HPCA '03: Proceedings of the 9th International Symposium on High-Performance Computer Architecture*, page 281, Washington, DC, USA, 2003. IEEE Computer Society.
- [6] Giuseppe Ascia, Vincenzo Catania, Alessandro G. Di Nuovo, Maurizio Palesi, and Davide Patti. Efficient design space exploration for application specific systems-on-a-chip. *Journal of Systems Architecture*, 53(10):733–750, 2007.
- [7] G. Palermo, C. Silvano, and V. Zaccaria. ReSPIR: A response surface-based pareto iterative refinement for application-specific design space exploration. *IEEE Transactions on Computer Aided Design of Integrated Circuits*, 28(12):1816–1829, December 2009.
- [8] Yaochu Jin, M. Olhofer, and B. Sendhoff. Managing approximate models in evolutionary aerodynamic design optimization. In *Proc. Congress on Evolutionary Computation*, volume 1, pages 592–599 vol. 1, 2001.
- [9] Yaochu Jin, M. Olhofer, and B. Sendhoff. A framework for evolutionary optimization with approximate fitness functions. *IEEE Transaction on Evolutionary Computation*, 6(5):481–494, 2002.
- [10] John Dennis, J. E. Dennis, Virginia Torczon, and Virginia Torczon. Managing approximation models in optimization. In *Multidisciplinary Design Optimization: State-of-the-Art*, pages 330–347, 1995.
- [11] M.T.M. Emmerich, K.C. Giannakoglou, and B. Naujoks. Single- and multiobjective evolutionary optimization assisted by gaussian random field metamodels. *Evolutionary Computation, IEEE Transactions on*, 10(4):421–439, Aug. 2006.
- [12] Joshua Knowles. Parego: a hybrid algorithm with on-line landscape approximation for expensive multiobjective optimization problems. *Evolutionary Computation, IEEE Transactions on*, 10(1):50–66, Feb. 2006.
- [13] Jose Renau, Basilio Fraguera, James Tuck, Wei Liu, Milos Prvulovic, Luis Ceze, Smruti Sarangi, Paul Sack, Karin Strauss, and Pablo Montesinos. SESC simulator, January 2005. <http://sesc.sourceforge.net>.
- [14] S.C. Woo, M. Ohara, E. Torrie, J.P. Singh, and A. Gupta. SPLASH-2 programs: characterization and methodological considerations. *Proceedings of the 22th International Symposium on Computer Architecture*, page 2436, 1995.
- [15] T. J. Santner, Williams B., and Notz W. *The Design and Analysis of Computer Experiments*. Springer-Verlag, 2003.
- [16] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan. A fast and elitist multiobjective genetic algorithm: NSGA-II. 6(2):182–197, 2002.
- [17] Jaszkievicz A. Czyzak P. Pareto simulated annealing - a metaheuristic technique for multiple-objective combinatorial optimisation. *Journal of Multi-Criteria Decision Analysis*, (7):34–47, April 1998.