Multi-objective design space exploration of embedded systems

Gianluca Palermoa,⁎, Cristina Silvanoa and Vittorio Zaccariab

aPolitecnico di Milano, Pza. L. Da Vinci 32, I-20133 Milano, Italy
bAdvanced Computing Group, AST – STMicroelectronics, Via Cantonale 16/E, Manno Lugano, Switzerland

Abstract. In this paper, we address the problem of the efficient exploration of the architectural design space for parameterized embedded systems. The exploration problem is multi-objective (e.g., energy and delay), so the main goal of this work is to find a good approximation of the Pareto-optimal configurations representing the best energy/delay trade-offs by varying the architectural parameters of the target system. In particular, the paper presents a Design Space Exploration (DSE) framework to simulate the target system and to dynamically profile the target applications. In the proposed DSE framework, a set of heuristic algorithms have been analyzed to reduce the overall exploration time by computing an approximated Pareto set of configurations with respect to the selected figures of merit. Once the approximated Pareto set has been built, the designer can quickly select the best system configuration satisfying the constraints. Experimental results, derived from the application of the proposed DSE framework to a superscalar architecture, show that the exploration time can be reduced by three orders of magnitude with respect to the full search approach, while maintaining a good level of accuracy.

Keywords: Design space exploration, low-power design, platform-based design, multi-objective optimisation

1. Introduction

Platform-based design is becoming the de-facto approach to the design of modern System-On-Chip (SOC) applications [1]. 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 and delay) for the given class of applications. This tuning process is also called the 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.

In general, the optimization problem involves the minimization (maximization) of multiple objectives making the definition of optimality not unique. The solution of multi-objective or constrained optimization problems consists of finding the points of the Pareto curve [2]. For example, the optimization problem is to minimize the power consumption under a delay constraint or vice versa. The solution of this problem is straightforward if the Pareto curve is available. However, a Pareto curve for a specific platform is available only when all the points in the design space have been explored and characterized in terms of objective functions. This full search approach is often unfeasible due to the cardinality of the design space and to the long simulation time needed for computing the evaluation functions.

Platform-based approach can be considered as a new system-level design paradigm, in which components belonging to a specific library are instantiated and sized to meet specific constraints on the target application domain. However, two fundamental problems must be faced:

1. The overall space of configurations is given by the cartesian product of the configuration spaces for each component and it can become very wide.
2. The figures of merit associated with each system configuration cannot be evaluated statically, unless some specific higher-level models are built.

⁎Corresponding author. E-mail: gpalermo@elet.polimi.it.
To explore the large design space for the target architecture, an approach based on the full search of the optimal architectural parameters at the system-level with respect to the multi-objective cost functions can be computationally very costly due to the long simulation time required to explore the wide space of parameters.

In [3], we addressed the multi-objective exploration problem by introducing a flexible and modular DSE framework to simulate the embedded system architecture and to dynamically profile the target applications. In the present paper, we analyze a set of heuristic exploration algorithms, that have been plugged-in the proposed DSE framework for efficiently pruning the design space. The main goal is to reduce the overall analysis time by computing an approximated Pareto set of configurations with respect to the selected figures of merit (e.g., energy and delay). Once the approximated Pareto set has been built, the designer can quickly select the best system configuration satisfying the constraints. In this way, only a subset of system configurations must be analyzed to identify near-optimal configurations. The DSE framework includes the simulation models of the target system modules to profile dynamically the target applications to evaluate power and performance figures.

The heuristic techniques proposed in the paper are the Random Search Pareto (RSP), the Pareto Simulated Annealing (PSA), and the Pareto Reactive Tabu Search (PRTS). These techniques represent an adaptation and tuning of some classical optimization algorithms to solve the problem of deriving a good approximation of Pareto sets.

Experimental results have then been obtained by comparing efficiency/accuracy of the three presented optimization algorithms by using a selected set of target applications executed on a flexible superscalar platform. Each parameterized component of the target architecture is provided with the corresponding system-level executable model to dynamically profile the given application to collect the information related to the figures of merit. The accuracy and efficiency of the proposed approaches have been compared with the Full Search approach. The comparisons are based on the Average Distance from the Reference Set (ADRS) and Two Set Coverage (C) metrics, coming from the multi-objective optimization domain.

The reported results show a reduction of the simulation time of up to three orders of magnitude with respect to the full search strategy, while maintaining a good level of accuracy (under 8% on average). Furthermore, the proposed methodology has been applied to a case study, the GSM encoder application, in order to find the optimal configuration for a constrained design problem.

The rest of the paper is organized as follows. A review of the most significant works appeared in literature concerning the DSE problem is reported in Section 2. The basic concepts and definitions of the DSE problem are stated in Section 3, while the proposed DSE framework as well as the algorithms analyzed in this paper are described in Section 4. Section 5 discusses the experimental results carried out to evaluate the efficiency with respect to accuracy of the algorithms plugged in the proposed framework for a superscalar configurable target architecture. Finally, some concluding remarks have been reported in Section 6.

2. Background

Several system-level estimation and exploration methods have been recently proposed in literature targeting power-performance tradeoffs from the system-level standpoint [4–10].

The SimpleScalar toolset [8] is based on a set of MIPS-based architectural simulators focusing on different abstraction levels to evaluate the effects of high-level algorithmic, architectural and compilation tradeoffs. The SimpleScalar framework provides the basic simulation-based infrastructure to explore both processor architectures and memory subsystems. However, SimpleScalar does not support power analysis. Based on the SimpleScalar simulators, SimplePower [11] can be considered one of the first efforts to evaluate the different contributions to the energy budget at the system-level. The SimplePower energy estimation environment consists of a compilation framework and an energy simulator that captures the cycle-accurate energy consumed by the SimpleScalar architecture, the memory system and the buses. In particular, the energy estimates related to the memory system include the energy consumed by the I-cache and D-cache, the address and data buses, the address and data pads and the off-chip main memory. Concerning the processor’s energy model, SimplePower considers only the energy consumed by the core’s data path, neglecting the control path. Although the proposed system-level framework is quite general, the exploration methodology reported in [11] is limited to a search over the space of the following parameters: cache size, block buffering, isolated sense amplifiers, pulsed word lines and eight different compilation optimizations (such as loop unrolling). 

The authors of [11] have considered the tradeoff between accuracy and complexity, proposing a series of different design constraints. Among these, the authors have chosen a specific processor, the Avalon, and a set of memory subsystems, the Avalon memory controller and bus. This work is quite different from most of the other works described in this paper, as it is focused on a specific processor and memory subsystem configuration. However, the authors of [11] have shown how their methodology can be applied to other processor and memory subsystem configurations. This is important because it shows that the methodology can be used for designing systems with different characteristics, such as different processor architectures and memory subsystems. This is a significant advantage because it allows designers to choose the most appropriate configuration for their specific application.

In general, the use of multi-objective optimization techniques in the design of embedded systems is becoming more common. These techniques allow designers to explore the design space more efficiently and to find designs that are optimal with respect to multiple objectives. However, the use of these techniques can be challenging, as they require careful consideration of the tradeoffs between accuracy and complexity. The authors of [11] have shown how this can be done in practice, and this work is an important contribution to the field of multi-objective optimization for embedded systems.
More recently, the Watch architecture is a representative of the architectural-level framework proposed in [9] to analyze power with respect to performance tradeoffs with a good level of accuracy with respect to lower-level estimation approaches. Watch represents an extension of the SimpleScalar simulator to support power analysis at the architectural level. Watch provides a framework to explore different system configurations and optimization strategies to save power, in particular focusing on processor and memory subsystems.

The Avalanche framework [5] evaluates simultaneously the energy-performance tradeoffs for software, memory, and hardware for embedded systems. The Avalanche framework mainly focuses on the processor and memory subsystems. The work in [7] proposes a system-level technique to find low-power high-performance superscalar processors tailored to specific user applications. Low-power design optimization techniques for high-performance processors have been investigated in [10] from the architectural and compiler viewpoints. A trade-off analysis of power/performance effects of SOC (System-On-Chip) architectures has been recently presented in [12], where the authors propose a simulation-based approach to configure the parameters related to the caches and the buses.

Among DSE frameworks, there have been a few approaches that have been introduced recently to approximate Pareto-curves for computer architecture design [12,13].

In general, the most trivial approach to determine the Pareto-optimal configurations into a large design space with respect to a multi-objective design optimization criteria consists of the comprehensive exploration of the configuration space. This brute force approach can be feasible only if the number of parameters in the configuration space is very limited. On the contrary, it is quite common to find a design space composed of tens of parameters, leading to an exponential analysis time. Thus, traditional heuristic techniques must be used. When the design space is too large to be exhaustively explored, heuristic methods must be adopted to find acceptable near-optimal solutions. The problem of the efficient construction of Pareto curves has been often addressed by using domain-specific algorithms. For example, the high-level synthesis benchmark problem (minimization of latency with area constraints) implicitly needs to consider an approximation of the Pareto curve of the area/latency design evaluation space. For this problem, specific approaches have been proposed in the past [14]. However, due to the high generality of the design space of a platform design, domain-specific algorithms are very difficult to find, thus one should resort to traditional heuristics.

Platonic [12] is an optimization framework that exploits the concept of parameter independence to derive an approximate Pareto curve without performing the exhaustive search over the whole design space. The authors define two parameters as interdependent if changing the value of one of them impacts the optimal parameter value of the other. In this case, all the combinations of these two parameters must be analyzed to find an optimal configuration. However, if the parameters are independent, the subspaces can be analyzed separately, leading to a reduced simulation time. The main drawback of this approach is that parameter independence must be specified by the user by means of a dependency graph since no automatic methods are proposed for such tasks. Platonic is not a modular framework since it allows only the exploration of a MIPS-based system and its goodness has not been compared with simpler approaches such as random search or full parameter space exploration.

More recently, Palesi et al. [13] extended Platonic by applying genetic algorithms to optimize dependent parameters, resorting to the default Platonic policy when independent parameters are specified by the user. However, their approach is always based on the a-priori parameter dependency graph to be given by the user and it is compared only with the default Platonic policy.

3. System level multi-objective optimization: definitions and performance indices

As said before, IP reuse methodologies and platform-reconfigurability are converging into a new design paradigm [1], which is strongly influencing today's automatic system synthesis. In this context, a stable 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. In fact, for a microprocessor-based system, a reasonable set of parameters is the following:

- Number of levels in the memory hierarchy and positioning (on-chip, off-chip);
3.3. Pareto dominance

In single-objective optimization, the feasible set is totally ordered according to the objective function \( f \). When several objectives are involved, the situation changes and the feasible set is partially ordered. Formally, the relations =, <, \( \leq \), \( \geq \) and > can be extended to objective vectors in the following way.

For any two objective vectors of \( k \) elements \( \vec{g} \) and \( \vec{h} \), we have:

\[
\vec{g} = \vec{h} \quad \text{if} \quad f_i(\vec{x}) = g_i = h_i \\
\vec{g} \leq \vec{h} \quad \iff \quad \forall i \in \{1, 2, \ldots, k\} : g_i \leq h_i \quad (1) \\
\vec{g} < \vec{h} \quad \iff \quad (\vec{g} \leq \vec{h}) \land (\vec{g} \neq \vec{h})
\]

The relations \( \geq \) and > are defined similarly.

The previous notions are useful to introduce the concept of Pareto dominance. In fact, for the minimization problem, for any two design vectors \( \vec{a} \) and \( \vec{b} \), the following concepts can be defined:

- Dominance (\( \succ \))
  \[
  \vec{a} \succ \vec{b} \quad \iff \quad f(\vec{a}) < f(\vec{b})
  \]
- Weak Dominance (\( \succeq \))
  \[
  \vec{a} \succeq \vec{b} \quad \iff \quad f(\vec{a}) \leq f(\vec{b})
  \]
- Indifference (\( \sim \))
  \[
  \vec{a} \sim \vec{b} \quad \iff \quad (f(\vec{a}) \notin f(\vec{b})) \land (f(\vec{b}) \notin f(\vec{a}))
  \]

The definitions for the maximization problem are similar.

3.4. Pareto optimality

Based on the above concept of Pareto Dominance, the optimality criterion for the MOO problem can be introduced.

For each design vector \( \vec{x} \in \Omega \) given a set \( A \subseteq \Omega \), the vector \( \vec{x} \) is said to be nondominated with respect to the set \( A \) if \( \vec{x} \notin A \) : \( \vec{x} \succ \vec{a} \). Moreover, \( \vec{x} \) is said to be Pareto Optimal if \( \vec{x} \) is nondominated with respect to \( \Omega \). That is, \( \vec{x} \) is optimal in the sense that it cannot be improved in any objective without causing a degradation of at least one of the other objectives.
3.5. Nondominated sets and fronts

Let us assume \( A \subseteq \Omega \). The function \( p(A) \) provides the set of nondominated design vectors in \( A \):

\[
p(A) = \{ a \in A \mid \exists a' \in A : a' \succ a \}
\]

The set \( p(A) \) is the nondominated set with respect to \( A \), while the corresponding set of objective vectors \( f(p(A)) \) is said to be the nondominated front with respect to \( A \). Furthermore, the set \( X_p = p(\Omega) \) is called Pareto-optimal Set and the set \( Y_p = f(X_p) \) is named as Pareto-optimal Front.

3.6. Performance indices

In literature, many performance indices have been used for evaluating the quality of the nondominated fronts for multi-objective optimization problems [16–18]. In this section, two metrics are introduced to evaluate the quality of the methods used to approximate the Pareto-optimal front. The first metric is mainly used to measure the distance between the Pareto-optimal front and a front generated by using an algorithm to approximate the Pareto front. The second metric is typically used to compare different methods to approximate the Pareto curve. By using the first metric, an absolute measure can be derived, while the second metric provides a comparison between approximate measures.

3.6.1. Average distance from reference set

To measure the distance between the Pareto-optimal front and a front generated by using an algorithm to approximate the Pareto front, the metric used is the Average Distance from Reference Set [19], \( ADRS \). Assuming that \( A \subseteq \Omega \) is a set of design vectors, while \( p(a) \) is the relative nondominated set. The function \( ADRS(\cdot,\cdot) \) measures the distance between the \( p(A) \) set and the Pareto-optimal set \( X_p = p(\Omega) \) as follows:

\[
ADRS(X_p, p(A)) = \frac{1}{|X_p|} \sum_{x_p \in X_p} \left( \min_{\tilde{a} \in p(A)} \{ d(x_p, \tilde{a}) \} \right)
\]

(2)

where

\[
d(x_p, \tilde{a}) = \max_{j=1,\ldots,M} \left( 0, \frac{f_j(\tilde{a}) - f_j(x_p)}{f_j(x_p)} \right)
\]

(3)

and \( M \) is the number of objective functions.

3.6.2. Two-set coverage

To compare nondominated fronts, the metric used is the Two-set Coverage [20], \( C \). Let \( A, B \subseteq \Omega \) be two sets of design vectors and \( p(A), p(B) \) be the corresponding nondominated set. The function \( C(\cdot, \cdot) \) maps the ordered pairs \((p(A), p(B))\) to the \([0,1]\) interval.

\[
C(p(A), p(B)) = \frac{|\{ \tilde{b} \in p(B) : \exists \tilde{a} \in p(A) : \tilde{a} \succ \tilde{b} \}|}{|p(B)|}
\]

(4)

The value \( C(p(A), p(B)) \) equal to 0 means that all the design vectors in \( p(B) \) are weakly dominated by the design vectors of \( p(A) \). As opposite, \( C(p(A), p(B)) \) equal to 1 represents the situation when none of the points in \( p(B) \) are weakly dominated by the set \( p(A) \). Both directions must always be considered, since \( C(p(A), p(B)) \) is not necessarily equal to \( 1 - C(p(B), p(A)) \).

The above defined metrics are used in Section 5 to compare the results obtained from the proposed DSE framework by using three different methods to approximate the Pareto front.

4. Proposed design space exploration framework

The overall goal of this work aims at providing a methodology and a retargetable tool to drive the designer towards near-optimal solutions, with the given multiple constraints, in a cost-effective fashion. The final product of the framework is a Pareto curve of configurations within the design evaluation space of the given application. To meet our goal we implemented a skeleton for an extendible and easy to use framework for multi-objective exploration. The proposed DSE framework is flexible and modular in terms of:

- Target architecture and related system-level executable models;
- Exploration algorithms;
- System-level metrics.

Given a new target architecture, the user can plug in the new architecture module in the framework, while the other modules can be considered as ready-to-use black boxes. Similarly, given a new exploration algorithm or a new executable module, the user can plug-in the new module in the framework.

The proposed framework is shown in Fig. 1. It is mainly composed of two modules: System Description Module (SDM) and Design Space Exploration Module (DSEM). The DSEM receives as input the description
of the possible design configurations (i.e. the target design space) and the application for which the system has been designed and for which the optimal configuration must be found (SDM). The framework explores the design space by using an iterative optimization technique with respect to different metrics.

The **Optimizer** module is responsible for choosing, from the design space, a set of candidate optimal points to be evaluated in terms of objective functions. Once selected, each point is mapped into a specific instance of the target architecture by the "Architecture Mapping" module. Providing a 'mapping' between the high-level description interface used by the Optimizer and the actual specification of the target architecture, this module enables the evaluation of each point by means of simulation based on the executable model of the target system. In the presented framework, the model of the target architecture can be described at different abstraction levels, being available the corresponding simulator.

In our experimental environment, we used an instruction-level simulator, but the framework can accept also lower level specifications (such as either RT or gate level) as well as higher level specifications, without changing the implementation of the optimizer module. This is done by means of the "Architecture Mapping".

Up to now, the approximation of the Pareto curve could be performed by a set of plugged-in exploration algorithms:

- Random Search Pareto (RSP) [21]. RSP algorithm is derived from Monte Carlo methods. In general, the main characteristic of Monte Carlo methods is the use of random sampling techniques to come up with a solution of the target problem. The random sampling technique has been proved to be one of the best techniques to avoid falling into local minima.

- Pareto Simulated Annealing (PSA) [19]. Simulated annealing is a Monte Carlo approach for minimizing multivariate functions. The term simulated annealing derives from the analogy with the physical process of heating and then slowly cooling a substance to obtain a strong crystalline structure. In the Simulated Annealing algorithm a new configuration is constructed by imposing a random displacement. If the cost function of this new state is less than the previous one, the change is accepted unconditionally and the system is updated. If the cost function is greater, the new configuration is accepted probabilistically; the acceptance possibility decreases with the temperature. This procedure allows the system to move consistently towards lower cost function states, thus 'jumping' out of local minima due to the probabilistic acceptance of some upward moves. The PSA is an evolution of SA for multi-objective optimization. At each step of PSA, the starting point is not a single configuration but a set of configurations (Partial Pareto Set).
Table 1
Description of the selected benchmarks and corresponding full search simulation time

<table>
<thead>
<tr>
<th>Benchmark</th>
<th>Description</th>
<th>Full Search Simulation Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>DCT IDCT UNOPT</td>
<td>Unoptimized Discrete Cosine and Inverse Discrete Cosine Transform</td>
<td>37h</td>
</tr>
<tr>
<td>Fast DCT</td>
<td>Fast Discrete Cosine Transform</td>
<td>34h</td>
</tr>
<tr>
<td>Fast IDCT</td>
<td>Fast Inverse Discrete Cosine Transform</td>
<td>32h</td>
</tr>
<tr>
<td>FIR 1, FIR 2</td>
<td>Linear-phase FIR digital filter</td>
<td>191h, 111h</td>
</tr>
<tr>
<td>Gamma</td>
<td>Gamma coefficients computation</td>
<td>35h</td>
</tr>
<tr>
<td>Gauss</td>
<td>Gaussian elimination</td>
<td>29h</td>
</tr>
<tr>
<td>Quarcube</td>
<td>Quadratic and cubic solving routine</td>
<td>97h</td>
</tr>
</tbody>
</table>

Fig. 2. Comparison between nondominated fronts generated by using the PSA algorithm (dark gray circles), by varying the number of simulations, with respect to the Pareto front (light gray stars) for Gauss benchmark. Figures a, b, c and d show the nondominated fronts for $PSA_{1000}$, $PSA_{10k}$ and $PSA_{100k}$ respectively.

- Pareto Reactive Tabu Search (PRTS) [22]. The Tabu Search (TS) is an iterative algorithm that explores the design space by means of 'moves'. The key concept behind the algorithm is the tabu list, i.e., a list containing prohibited moves that, usually, consist of the most recently visited points. The reason of the tabu list is to avoid to stuck into local minima. Recent studies [22] have demonstrated that the length of this list is a determining factor to reduce the possibility to stuck into local
algorithm. The Reactive Tabu Search is an evolution of the Tabu Search algorithm that exploits an adaptive prohibition period, paired with an escape mechanism, to afford the tuning problem. In RTS, the prohibition period of a specific solution increases with the frequency of the visits to that solution. Moreover, to avoid the possibility of a cyclic exploration, an escape mechanism is used to escape from local minimum. The escape is usually implemented by generating a random walk.

The main effort of our work has been directed toward the tuning phase of the parameters requested by the selected algorithms to achieve a good trade-off of efficiency with respect to accuracy.

5. Experimental results

In this section, we present the experimental results obtained by applying the proposed DSE framework to optimize a superscalar microprocessor-based system. In particular, the first subsection describes the target architecture and the related design evaluation space; the second subsection presents the application of the methodology to a set of target applications and it reports the results obtained by the comparison of the three algorithms presented in Section 4. The last subsection discusses the application of the DSE framework to a case study represented by the optimization of the GSM encoding application.

5.1. Target system architecture

In general, a superscalar architecture 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 and the energy consumption at the system-level. Each instance of the virtual architecture has been described in terms of the following parameters:

- $S_{b1}, S_{b3}, S_{b2}$ are the ordered sets of the possible sizes of the I/D L1 caches (from 2 KByte to 16 KByte) and unified L2 cache (from 16 KByte to 128 KByte),
- $S_{b1}, S_{b4}, S_{b0}$ are the ordered sets of the possible block sizes of the I/D L1 caches (from 16 Byte to 32 Byte) and unified L2 cache (from 32 Byte to 64 Byte).

![Fig. 3. Minimum, average and maximum values of the Average Distance from Reference Set (ADRS) obtained by RSP (a), PSA (b) and PRTS (c) to approximate the Pareto front for all the target applications.](image-url)
ch is an evolution that exploits the problem with an existential solution. In specific solution visits to that possibility of a simulation is used to handle is usually made.

Directed tests requested by the trade-off of the simulation framework to the target evaluation space; plication of the GMS and it reports on of the three last subsection framework to a division of the GSM.

Fig. 4. Box plots representing the Two-set Coverage metric to compare the three proposed exploration algorithms.

- $S_{a1}, S_{a2}, S_{a3}$ are the ordered sets of the possible associativity values of the I/D L1 caches (from 1 way to 2 ways for the I-cache and from 2 ways to 4 ways for the D-cache) and unified L2 cache (from 4 ways to 8 ways).
- $S_{ib}, S_{im}$ are the ordered sets of the possible number of integer ALUs and multipliers (from 1 to 2).
- $S_{fp}, S_{fpm}$ are the ordered sets of the possible number of floating point ALUs and multipliers (from 1 to 2).
- $S_{iu}$ are the ordered sets of the possible issue width sizes (from 2 to 8).

Watch simulator [9] has been used as an architectural simulator providing a dynamic profiling of energy and delay.

5.2. Comparison results

In this subsection, we report the results in terms of efficiency and accuracy obtained by the application of our DSE methodology to a set of benchmarks composed of DCT transforms, FIR filters and other numerical algorithms. Table 1 describes in detail the set of benchmarks chosen as target applications for the validation: all the applications are open-source programs written in C language. The last column of Table 1 reports the simulation time required by the full search for each benchmark executed on a Linux Workstation provided with the Intel Pentium 4 at 1.7 GHz clock speed.

To validate our exploration methodology we carried out two parallel exploration flows: the first flow is based on the Full Search, while the second one is based on one of the proposed heuristic algorithms. In the Full Search, the optimizer analyzes the global design space, so the number of simulations to be executed is 196608 corresponding to approximately 370 hours of simulations for the selected set of benchmarks.

Figure 2 shows four different approximated Pareto fronts generated by varying the number of simulations. To exemplify our methodology, the Gauss benchmark has been selected and the PSA algorithm has been used to approximate the optimal Pareto front. In Fig. 2, the average distance from the nondominated front generated by PSA (represented by dark gray circles) to the optimal Pareto front (represented by light gray stars) decreases by increasing the number of simulations. Formally, indicating as $\Omega_{A_n}$ the subset of $n$ points in the feasible region $\Omega$ visited by the algo-
Fig. 5. Energy-delay exploration results for GSM-encoding benchmark. Light gray points have been generated by the RSP algorithm, while dark gray points are the corresponding nondominated points.

Fig. 2 shows the four fronts composed of the nondominated points: $p(\Omega_{PSA^{100}})$, $p(\Omega_{PSA^{1k}})$, $p(\Omega_{PSA^{100k}})$ and $p(\Omega_{PSA^{100k}})$. As shown in Fig. 2, the $PSA^{1k}$ approximated front is very close to the optimal Pareto front, while the $PSA^{10k}$ and $PSA^{100k}$ approximated fronts are almost totally overlapped to the optimal Pareto front.

Figure 3(a) shows the minimum, average and maximum errors (measured in terms of the Average Distance from Reference Set – ADRS metric) to approximate the full search Pareto front by using the nondominated fronts generated by the RSP ($p(\Omega_{RSP})$) for all the chosen target applications. Similarly, Fig. 3(b) and (c) show the minimum, average and maximum errors obtained by the nondominated fronts respectively generated by PSA ($p(\Omega_{PSA})$) and PRTS ($p(\Omega_{PRTS})$) to approximate the full search Pareto curves.

For each algorithm, fixed the number of simulations, the values have been obtained by repeating the same exploration five times for each benchmark in the target suite. In Fig. 3(a), the average error for $RSP^{100}$ is under 1%, for $RSP^{1k}$ under 7%, and for $RSP^{10k}$ under 1%. Similar trends are presented in Fig. 3(b) and in Fig. 3(c), where for $PSA^{100}$ and $PRTS^{100}$ respectively, the average error is maintained under 7% and 8%, while for $PSA^{1k}$ and $PRTS^{1k}$ the error is under 2% and 3% respectively. For $PSA^{10k}$ and $PRTS^{10k}$ the average error is under 1%. A decreasing trend can be noted in the curves describing the minimum, average and maximum errors by increasing the number of simulations, while the oscillations are due to the random nature of the search.

The comparison of the proposed heuristic algorithms based on the two-set coverage metric is presented in Fig. 4. To compare the exploration algorithms, we selected a testbench composed of 1000 simulations and we repeated the same exploration five times for each benchmark. In Fig. 4, a graphical method, called box plot [23], has been used to represent the spread and the shape of the distribution for the values obtained by the comparison of the proposed exploration algorithms. The upper and the lower lines respectively represent the best and the worst case for the two-set coverage metric. The upper (lower) box includes the 25% of the samples...
slightly above (below) the average value represented by the bold line.

Figure 4 is composed of six box plots. The first two box plots correspond to the comparison between RSP and PRTS algorithms and they represent $C(p(\Omega_{RSP}), p(\Omega_{PRTS}))$ and $C(p(\Omega_{PRTS}), p(\Omega_{RSP}))$ respectively. The third and fourth box plots represent the comparison between RSP and PSA, in terms of $C(p(\Omega_{RSP}), p(\Omega_{PSA}))$ and $C(p(\Omega_{PSA}), p(\Omega_{RSP}))$ respectively. The last two box plots represent $C(p(\Omega_{PRTS}), p(\Omega_{PSA}))$ and $C(p(\Omega_{PSA}), p(\Omega_{PRTS}))$ to compare PRTS and PSA algorithms. Figure 4 shows that the nondominated front generated by PSA covers more than 80% of the PRTS on average, while only the 15% of the PRST front covers the PSA front. Furthermore, both PRTS and PSA algorithms produce better nondominated fronts with respect to RSP.

5.3. Case study

This section presents the results obtained by applying the RSP algorithm to the optimization of the GSM encoding benchmark. The DSE problem consists of finding the optimal configuration in terms of energy with a constraint on the delay of 50 millions of cycles for the encoding of 73760 Byte of data. To solve this problem, we derived the RSP curve of the benchmark for the Energy and Delay design evaluation space by using the RSP $^{10k}$ algorithm. Figure 5 shows the scatter plot of all the points generated by the RSP algorithm $\Omega_{RSP}$ (light gray circles) and the corresponding non-dominated points $p(\Omega_{RSP})$ (dark gray circles). Once the approximation of the Pareto curve has been found, the constrained design space exploration problem consists of selecting among the non-dominated points. After exploration, we found the near-optimal RSP point characterized by the following values: $\text{Energy} = 1.97 \times 10^3$ [J] and Delay = $48 \times 10^2$ [cycle]. This point fulfills the constraints imposed by the problem in terms of delay and it minimizes the energy. Based on the analysis of the accuracy of the algorithms performed in the previous section, we expect that this point is very close to the full search corresponding point. The RSP analysis requires a simulation time of three days, while the full search analysis would have required approximately 516 days of simulation.

6. Conclusions

The paper presents an analysis of heuristic techniques for efficiently synthesizing an optimal hardware/software platform configuration given a set of system-level constraints. The heuristic techniques reduce the overall analysis time by building an approximate Pareto set of configurations with respect to the given figures of merit. Experimental results have shown that the proposed techniques can reduce the time needed to find the near optimal system configuration of up to three orders of magnitude with respect to a full search strategy, while maintaining a good level of accuracy.

References


---

1. Introduction

Embedding by the present application is critical, that is the design of the component is based on measurement and analysis of methods for consumption energy in an application.

Power analysis that employs a model in minimization techniques in [1–4] is developed. This work is funded by the European Free Trade Agreement. Correspondence 998018; E-mail: INTRACOM

ISSN 1740-446