A Multi-Objective Decision-Theoretic Exploration Algorithm for Platform-based Design

Giovanni Beltrame, Gabriela Nicolescu
École Polytechnique de Montréal
giovanni.beltrame@polymtl.ca, gabriela.nicolescu@polymtl.ca

***Abstract*—This paper presents an efficient technique to perform multi-objective design space exploration of a multi-processor platform. Instead of using semi-random search algorithms (like simulated annealing, tabu search, genetic algorithms, etc.), we use the domain knowledge derived from the platform architecture to set-up the exploration as a discrete-space multi-objective *Markov Decision Process* (MDP). The system walks the design space changing its parameters, performing simulations only when probabilistic information becomes insufficient for a decision. The algorithm employs a novel multi-objective value function and exploration strategy, which guarantees high accuracy and minimizes the number of necessary simulations. The proposed technique has been tested with a small benchmark (to compare the results against exhaustive exploration) and two large applications (to prove effectiveness in a real case), namely the *ffmpeg* transcoder and *pigz* parallel compressor. Results show that the exploration can be performed with 10% of the simulations necessary for state-of-the-art exploration algorithms and with unrivaled accuracy ($0.6 \pm 0.05\%$ error).**

I. Introduction

Parametrized embedded System-on-Chip (SoC) architectures must be optimally tuned (i.e. their configuration parameters must be appropriately chosen) to find the best trade-off in terms of the selected figures of merit (e.g. energy and delay) for a given class of applications. This tuning process is called *Design Space Exploration (DSE)*.

In the past, multi-processor platforms have been explored using either classical heuristic algorithms (such as tabu search, simulated annealing, etc.) [1] or pruning techniques that try to reduce the size of the design space [2]. Both classes of techniques rely on simulation (or estimation) as a means for evaluating the system-level metrics corresponding to a newly found configuration. If system-level simulation (or estimation) can be performed in a reasonable time, these algorithms provide good results. This is generally not true for Multi-Processor Systems-on-Chip (MPSoCs), for which simulations can be rather lengthy and time consuming [3].

This work addresses the problem by exploiting the domain knowledge provided by the definition of a design platform. We extend our previous approach where exploration is modeled as a Markov Decision Process (MDP) [4], to create a multi-objective exploration algorithm referred to as Multi-Objective MDP (MOMDP), with the following contributions:

- A new value function definition
- Improved accuracy: MOMDP sports a six-fold accuracy improvement compared to MDP and surpasses nine other state-of-the-art exploration algorithms. The number of Pareto points discovered is also increased
- High performance: although the number of simulations required by MOMDP is higher than MDP, it remains an order of magnitude lower than any algorithm with comparable accuracy.

In addtition, we envision a trade-off between speed and accuracy combining MDP and MOMDP: the former can be used to obtain a quick overview of the design space, while the latter performs detailed analysis.

The paper is structured as follows: Section II summarizes classical exploration algorithms for multi-variate analysis; our extensions to MDP are introduced in Section III; the application of the methodology to three benchmarks is described in Section IV; finally, Section V draws some concluding remarks.

II. Related work

Three main classes of techniques have been proposed in literature for aiding the design space exploration of system architectures: (a) techniques that try to reduce the design space size [5], [6] (b) techniques that provide exploration heuristics [1](c) techniques based on statistical analysis aimed at guiding the exploration to specific regions of the design space [7]

These algorithms approach the problem as a black-box: they do not take into account the peculiar features, the design constraints and the a-priori knowledge of the target platform. For instance, platform-based design [8] methodologies provide a useful amount of knowledge that can be used to define better exploration strategies, consistently reducing the number of required simulations. We previously introduced the theoretical framework for the use of decision theory to exploit this information in [4]. In this paper we extend our methodology by making it fully multi-objective and by adding a new technique to avoid local minima. The idea is not to improve the performance of the algorithm, but rather to increase its accuracy and guarantee that the largest possible number of Pareto points are found.

III. Proposed Extensions

In this work, we propose two important extensions of the algorithm presented in [4], that increase its accuracy without excessively impacting on its performance.

978-3-9810801-7-9/DATE11/©2011 EDAA

A. The Leap of Faith

In our previous studies, the enabling of forbidden actions in the fourth step of the algorithm was not sufficient to escape from deep minima, especially when this required a large number of actions. To maximize the number of Pareto points discovered by the algorithm and guarantee that all the largest possible number of points is found, in this work we introduce a special action called the "*leap of faith*": if the application of forbidden actions fails to provide a new point with a better value function, the algorithm simulates the best point found on the graph, **disregarding its probability**. This technique allows to escape from local minima by selecting congurations with high but unlikely gains. If the configuration found in this way has a better value function, the algorithm restarts from this point, otherwise it is considered to have converged.

B. Multi-Objective Exploration

The algorithm works with value functions [9] that map states to real numbers; in the context of multi-objective optimization such functions are also called *scalarizing functions*. For the algorithm to generate an approximate Pareto curve, as opposed to generating a single point, the algorithm is applied multiple times, changing the scalarizing function at every application so that it covers the whole span of the metric space of interest. This technique, called *parameter variation* is often used to generate the Pareto curve of multi-objective optimization problems [10].

In this context, we introduce a novel value function to guide the exploration of the algorithm. As opposed to previous works (including our own), the proposed value function does not optimize globally according to a scalarizing function, but gives the best values to points that are likely to be on the Pareto curve. In the following we describe the proposed approach using only two metrics for the sake of simplicity, but it can be easily extended to n dimensions.

The main idea is to have a value function that depends on the number and location of approximate Pareto points found by the algorithm during execution. The algorithm is started by minimizing only one of the considered metrics (e.g. energy), disregarding all the others (e.g. time, area, etc.). This leads to one of the extremes of the Pareto front: from here, we use the proposed value function, which gives best values to points that are close to the starting point, but still minimizing all considered metrics.

The algorithm is iteratively, making the algorithm "follow" the Pareto front, until it is not possible to find any new Pareto-dominating solutions, as shown in Figure 1.

Figure 2 shows how the value function is dened for a generic application of the algorithm for two metrics (energy and time) starting from a point (E_0, T_0). The metrics space is divided into three areas: (1) an area where all the points are not Pareto-covered by already discovered points, (2) an area where all points are already Pareto-covered, and (3) an area where points are not Pareto-covered, but has already been considered in the previous itearations.

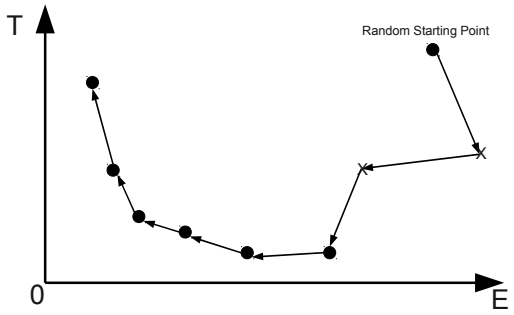


Figure 1. Iterative search of the Pareto front: at each execution the algorithm tries to discover a new point on the Pareto front. Dominated points are discarded.

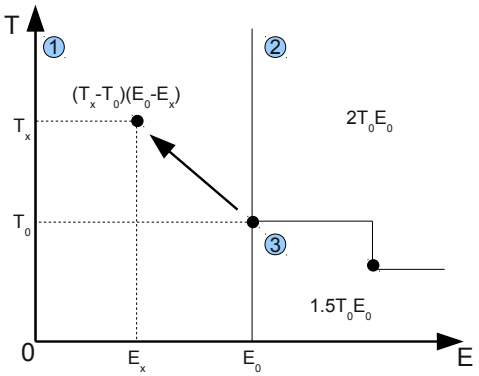


Figure 2. When moving from a point (E_0, T_0) to (E_x, T_X), the proposed value function partitions the space in 3 sectors

The points in areas (2) and (3) are to be avoided, because they are either already covered, or they are likely to have been found in the previous runs. For this reason, they are assigned a high (but not infinite, to still allow for multi-decision solutions to be found) value: $2E_0T_0$ for (2) and $1.5T_0E_0$ for (3). In area (1), the metric for any point (E_x, T_x) is $(T_x - T_0)(E_0 - E_x)$, making the algorithm favor the closest point that is not Pareto-covered and still minimizes the metrics. With repeated applications, the algorithm keeps looking for close Pareto points until no additional solutions are found. During the exploration, all points simulated by the algorithm are added to a set S if and only if they are not Pareto-covered by other points, and points are removed from S if they are found to be Pareto-covered at any time. At the end, S constitutes the estimation of the Pareto curve.

This way, there is no need to define appropriate parameters for the scalarizing function, and no need to determine the number of times the algorithm should be applied. This reduces the burden on the designer, as this number depends on the size and shape of the design space, the chosen algoritm, the chosesn scalarizing function, and other parameters. In addtition, the estimation of this number for a certain Pareto front coverage target usually requires a large number of experiments.

IV. Experimental Results

The proposed methodology has been validated using two large applications and a small benchmark for which exhaustive

Table I
CHOSEN APPLICATIONS

Application	Model	Synchr.	Sim. Time
ffmpeg	pthreads	semaphore	30m
pigz	pthreads	condition	3m
fft6	OpenMP	barrier	30s

search was possible, as listed in Table I. The two applications are *ffmpeg*, a video transcoder used to convert a small clip from MPEG-1 to MPEG-4, and *pigz*, a parallel compression algorithm. The small benchmark consists of an implementation of Bailey's 6-step FFT algorithm (*fft6*). All application are data-parallel and are targeted towards a homogeneous shared-memory multi-processor platform (N processors accessing a common memory via bus). *ffmpeg* and *pigz* are implemented using pthreads and create a set of "worker" threads equal to the number of available processors and dispatch independent data to each worker. *fft6* uses OpenMP instead, with loop parallelization and static scheduling. To guarantee the maximum variability in application behaviour, all applications use a different synchronization mechanism and require very different simulation times.

The ReSP [11] open-source simulation environment was used to perform the exploration of the aforementioned applications on a chip-multiprocessor platform. The platform has been explored using the same parameters as [11], with a resulting design space of 8640 points, comparable with similar works (e.g. 6144 points for [6]). To gather sufficient data for a statistical analysis the benchmarks were run 10 times for each exploration algorithm ($N = 30$ for each algorithm).

The proposed algorithm was compared with nine state-of-the-art multi-objective optimization algorithms implemented in MOMHLib++ [12]. To avoid problems of scale in our graphs (most algorithms perform orders of magnitude worse), we selected only the three **best** to be included here: MOGLS (Multiple Objective Genetic Local Search), IMMOGLS (Ishibuchi's and Murata's Multiple Objective Genetic Local Search), and PMA (Pareto Memetic Algorithm), plus our previously developed algorithm MDP.

A. Estimation of the Number of Evaluations

The number of evaluations needed to obtain an approximate Pareto-set for the proposed algorithm depends on the number of available actions, on the size of their bounds, and on the algorithm parameters l (the event horizon), ϵ (the convergence margin), and λ (the accuracy factor).

The best values for the parameters ($\lambda = 0.3, \epsilon = 10^{-6}, l = 4$), and their effect on the performance of the proposed algorithm was shown in [4], and will not be repeated here. In such previous work, additional parameters and testing were needed since the algorithm was *not multi-objective*: MOMDP does not require such effort to produce high-quality results.

Figure 3(a) shows how the proposed approach (indicated as MOMDP) compares with other algorithms in terms of number of simulations; note that it is possible to use the number of simulations, since all algorithms have negligible execution time with respect to a single simulation time. We tried to our best to find the optimal settings for each algorithm for the design space under consideration, but, still, MOMDP and MDP use ten times fewer evaluations than all other algorithms. The variability of the number of evaluations is very limited, showing stable convergence for all algorithms (with the notable exception of PMA) on all benchmarks.

Figure 3(b) shows the number of Pareto points found by each algorithm, normalized by the average found for each benchmark to allow global comparison. MOMDP provides the highest number of points on average (with a statistically significant difference with MDP), confirming the validity of its novel value function.

Also when comparing the proposed algorithm with recent approaches [13], the number of simulations required is an order of magnitude lower, except for MDP (which is significantly less accurate).

B. Quality of the Resulting Approximate Pareto-set

We use three indicators, presented in [14], to compare the relative quality of the approximate Pareto-Set obtained by MDP and other state-of-the-art algorithms: the *Average Distance from Reference Set* ($ADRS$), and the *non-uniformity* and *concentration* of solution distribution. Figure 4 shows the comparison of the Pareto-set evaluation metrics: concerning non-uniformity, all algorithms provide well-distributed solutions without any significantly different behaviour, and the detailed results are not reported here for the sake of brevity.

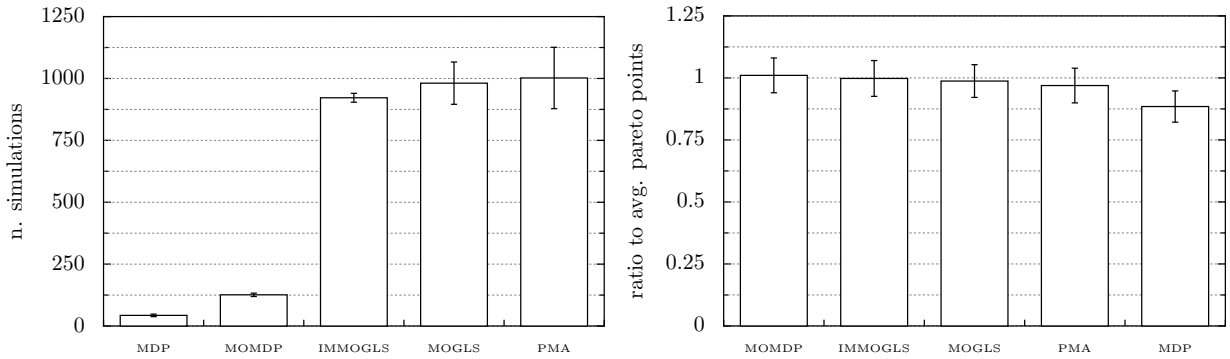
Figure 4(a) shows that MDP and MOMDP have significantly lower concentration, meaning that solutions cover more of the design space for the proposed algorithm. The results are confirmed using repeated t-tests (p-values< 0.02). Note how MDP and MOMDP are not significantly different ($p = 0.5$).

The accuracy of MOMDP was measured as the *Average Distance from a Reference Set (ADRS)*; the reference set consists of the Pareto-Set obtained after collecting all simulation results from all algorithms, or exhaustive simulation in the case of fft6. ANOVA reports that algorithms have different average ADRS with a p-value close to zero, and Figure 4(b) shows the boxplot of ADRS per algorithm, ordered by average. Using repeated t-tests between MDP and all the other algorithms, evidence suggests that MOMDP is the most accurate algorithm.

In conclusion, MOMDP outperforms all other algorithms for accuracy and uniformity, with similar extent, using one order of magnitude fewer simulations than pseudo-random algorithms, and 2.5 times more simulations than MDP.

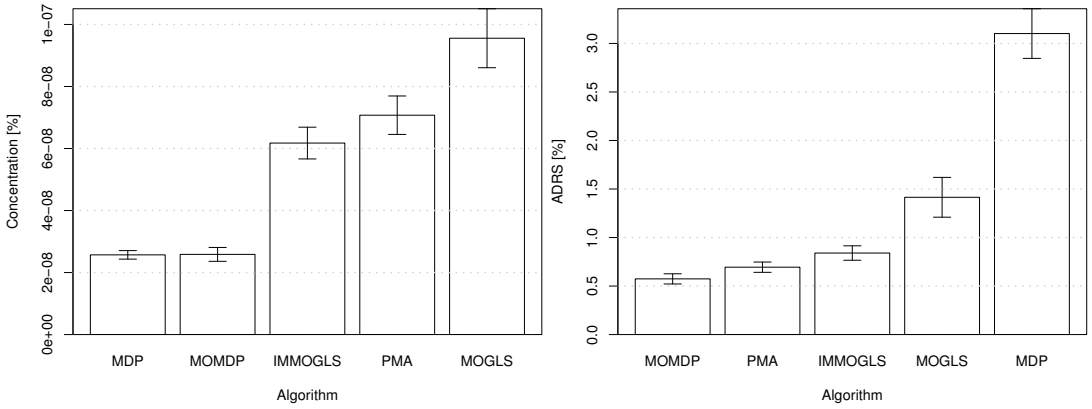
V. CONCLUSIONS

This paper presented a multi-objective design exploration algorithm based on Markov Decision Processes.The proposed algorithm presents a novel Pareto-front discovery technique that guarantees results of the highest accuracy, but still requiring a small number of simulations. In addition, when compared to previous work on MDPs, results show an increase in the number of discovered pareto points and a factor 6



(a) Evaluations (b) Pareto points

Figure 3. Average evaluations for algorithm convergence and number of Pareto points found by each algorithm as a ratio to the global average



(a) Concentration (b) ADRS with confidence intervals

Figure 4. Concentration (a)and ADRS (b) metrics comparing MOMDP with state-of-the-art algorithms (lower is better)

improvement in accuracy. The number of simulations has increased 2.5 times, but still remains one order of magnitude smaller than nine state-of-the-art pseudo-random heuristics. The proposed technique has been applied to three very differnt applications showing consistent behaviour on all three.

REFERENCES

[1] M. Palesi and T. Givargis, "Multi-objective design space exploration using genetic algorithms," in *CODES '02: Proceedings of the tenth international symposium on Hardware/software codesign*. Estes Park, Colorado: ACM, 2002, pp. 67–72.

[2] S. Mohanty, V. K. Prasanna, S. Neema, and J. Davis, "Rapid design space exploration of heterogeneous embedded systems using symbolic search and multi-granular simulation," *SIGPLAN Not.*, vol. 37, no. 7, pp. 18–27, 2002.

[3] J. Yi and D. Lilja, "Simulation of computer architectures: simulators, benchmarks, methodologies, and recommendations," *Computers, IEEE Transactions on*, vol. 55, no. 3, pp. 268–280, 2006.

[4] G. Beltrame, L. Fossati, and D. Sciuto, "Decision-theoretic design space exploration of multiprocessor platforms," *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, vol. 29, no. 7, pp. 1083–1095, 2010.

[5] W. Fornaciari, D. Sciuto, C. Silvano, and V. Zaccaria, "A Sensitivity-Based design space exploration methodology for embedded systems," *Design Automation for Embedded Systems*, vol. 7, no. 1, pp. 7–33, 2002.

[6] D. Sheldon, F. Vahid, and S. Lonardi, "Soft-core processor customization using the design of experiments paradigm," in *DATE Conference, 2007*, 2007, pp. 1–6.

[7] M. Lukasiewycz, M. Glay, C. Haubelt, and J. Teich, "Efficient symbolic multi-objective design space exploration," in *ASP-DAC '08: Proceedings of the 2008 Asia and South Pacific Design Automation Conference*. Seoul, Korea: IEEE Computer Society Press, 2008, pp. 691–696.

[8] A. Sangiovanni-Vincentelli, L. Carloni, F. D. Bernardinis, and M. Sgroi, "Benefits and challenges for platform-based design," in *Design Automation Conference, 2004. Proceedings. 41st*, 2004, pp. 409–414.

[9] T. Murata, H. Ishibuchi, and H. Tanaka, "Multi-objective genetic algorithm and its applications to flowshop scheduling," *Computers & Industrial Engineering*, vol. 30, no. 4, pp. 957–968, Sept. 1996.

[10] M. Laumanns, L. Thiele, and E. Zitzler, "An efficient, adaptive parameter variation scheme for metaheuristics based on the epsilon-constraint method," *European Journal of Operational Research*, vol. 169, no. 3, pp. 932–942, Mar. 2006.

[11] G. Beltrame, L. Fossati, and D. Sciuto, "ReSP: a nonintrusive Transaction-Level reflective MPSoC simulation platform for design space exploration," *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, vol. 28, no. 12, pp. 1857–1869, 2009.

[12] "MOMH: Multiple Objective Meta Heuristics," available at the web site http://home.gna.org/momh/.

[13] G. Palermo, C. Silvano, and V. Zaccaria, "An efficient design space exploration methodology for multiprocessor SoC architectures based on response surface methods," in *Embedded Computer Systems: Architectures, Modeling, and Simulation, 2008. SAMOS 2008. International Conference on*, 2008, pp. 150–157.

[14] C. Erbas, *System-level Modelling and Design Space Exploration for Multiprocessor Embedded System-on-chip Architectures*. Amsterdam University Press, Oct. 2006.