# Tuning DSE for Heterogeneous Multi-Processor Embedded Systems by means of a Self-Equalized Weighted Sum Method

Vittoriano Muttillo University of L'Aquila L'Aquila, italy vittoriano.muttillo@univaq.it Giuseppe Fiorilli University of L'Aquila L'Aquila, italy giuseppe.fiorilli@student.univaq.it Tania Di Mascio University of L'Aquila L'Aquila, italy tania.dimascio@univaq.it

# ABSTRACT

Heterogeneous multi-processor platforms are becoming widely diffused in the embedded system domain, mainly because of the opportunity to improve timing performance and, at the same time, to minimize energy/power consumption and costs. In using such kind of platforms, to be able to consider the trade-offs among different goals, a Design Space Exploration (DSE) is generally adopted. For this, existing DSE approaches typically rely on evolutionary algorithms to solve Multi-Objective Optimization Problems (MOOP) by minimizing a linear combination of weighted objective functions (i.e., Weighted Sum Method, WSM). The problem is then shifted towards the identification of weights able to represent desired tradeoffs. In such a context, this paper focuses on DSE for heterogeneous multi-processor embedded systems and introduces an approach that, while still driven by a "decision maker", is able to self-tune weights to equalize objective functions contribution. In particular, this work presents a self-equalized WSM integrated into a genetic algorithm used to identify sub-optimal implementation alternatives in the context of an Electronic System Level HW/SW Co-Design flow.

#### **CCS CONCEPTS**

Computer systems organization → Embedded systems; Embedded hardware.

## **KEYWORDS**

Design Space Exploration, Heterogeneous Multi-Processor Systems, Embedded Systems, Multi-objective optimization

# **1 INTRODUCTION**

Nowadays, embedded systems are everywhere, as they are widely used in several application domains (e.g., home automation, aerospace, automotive, etc.) imposing different functional and non-functional requirements. As a consequence, their design is an increasingly complex activity since several implementation alternatives, with different HW/SW technologies and constraints (e.g., timing performances, power/energy consumption, cost), need to be considered. To improve timing performances and to minimize energy/power consumption and cost at the same time, heterogeneous platforms are becoming widely diffused as well as *Electronic System Level* (ESL) HW/SW co-design methodologies that support designers in reducing time-to-market and costs. In such a context, one of the main design activity is the *Design Space Exploration* (DSE). In fact, it shall be able to solve a *Multi-Objective Optimization Problem* (MOOP) by optimizing a linear combination of weighted objectives. So, the problem is shifted towards the identification of weights able to represent the desired trade-offs. For this, the present work focuses on DSE for *Heterogeneous Multi-Processor Embedded Systems* (HMPESs) and proposes a "decision maker"-driven method for weights self-equalization.

In this paper, a description of related works considering decision maker suggestions into the DSE activity is reported in Section II, while Section III describes the adopted DSE approach and Section IV presents the main features of the proposed *Weights Linear Equalization* (WLE) method. Section V analyzes experimental results highliting some conclusions and future works.

## 2 LITERATURE OVERVIEW AND MOTIVATIONS

A common classification of decision maker preferences introduction in the resolution of MOOPs [20] is related to the possibility to explicitly indicate preferences at the beginning of the optimization process (a priori method), to make a preferences selection at the end of the MOOP process (a posteriori method), or a combination of both (interactive method). The "a posteriori" methods [11] are based on *Pareto Optimal Set* ( $\rho^*$ ) calculation followed by an evaluation of the worthiness of such solutions starting from Pareto Front ( $\rho F^*$ ). However, frequently, these approaches appear difficult to use since decision makers struggle to process  $\rho^*$  and  $\rho F^*$  values (especially if the number of objective functions is greater than 3). Whereas the "a posteriori" and "interactive" methods consider the possibility to analyze the MOOP results obtained at the end of the optimization process, the "a priori" method introduces the decision maker preferences in the early steps, taking into account the decision maker general knowledge of the MOOP [12] itself. Different "a priori" approaches have been presented in literature, and most of them are related to assign a "relevance value" to each objective function, then, using meta-heuristic algorithms, to find sub-optimal solutions for MOOP [6]. A common meta-heuristic algorithm involves the use of Genetic Algorithms (GAs) and Weighted Sum Method (WSM). GAs are able to capture excellent Pareto solutions, leaving to the decision maker the identification of the best one from a proposed solutions set [9]. As an example of GAs used for DSE of

embedded systems, Multicube [3] aims to realize a tool to support platform-based design, implementing different GAs meta-heuristics and offering different output solutions, as "a posteriori" process for the final decision maker.

Anyway, the main issue with GAs+WSM methods is the decision maker effort. Different methods incorporate preferences and priorities by using weights that represent an indication of how a specific objective function is important in the meta-heuristic analysis, but no unifying analysis of them is present in literature [12]. However, such methods can be classified with respect to the kind of rating assigned to each objective function, where a decision maker can assign values of relative importance to each objective function in a fixed or variable mode. Some works assign random values to weights, or apply periodical changes following specific pattern functions [8]. Another method is a subset of rating methods, called ranking, where the objective functions are ordered by their importance degree ([1 : *K*], i.e., from the lowest to the highest). In a ranking method, it is also possible to apply objective functions classification based on pairwise comparison between objective functions [19] or final utility function values [5]. It is also possible to classify objective functions in groups of broad categories (i.e., high, average, low) [10].

By better analyzing the presented approaches, it emerges the need to simplify as much as possible the decision maker work while also taking into account the contribution of the involved objective functions. In other words, it is needed a method to automatically evaluate weights, but still able to take into account decision maker preferences. So, this work presents an approach that introduces decision maker preferences into "a priori" GA+WSM method, using a "ranking categorization" based on a simple fixed value assigned to each objective function. Then, the GA steps are modified to automatically change, at each iteration, the weights with respect to the average of the objective functions they have been assigned to, so performing their self-equalization.

## 3 REFERENCE DESIGN SPACE EXPLORATION APPROACH

In the context of HMPESs, this work adopts the ESL HW/SW codesign flow presented in [4, 16, 17] where the most critical development step is the semi-automatic *Design Space Exploration* one. It involves several stages, from the definition of the solution space, the encoding with respect to the decision variable space, and the definition of the objective functions and the general MOOP. The main problem is to map application processes onto a set of basic HW components (i.e., *Basic Blocks*, BBs) selected by the decision maker [13]. It is worth noting that the MOOP is modeled as a minimization problem, defined as follows:

Definition 3.1. (Reference Design Space Exploration MOOP).

$$\min_{\bar{x}} \quad \bar{F}(\bar{x}) = [F_1(\bar{x}), \dots, F_k(\bar{x})]^{\intercal}$$
s.t.  $\bar{x} \in \Omega = \{ \bar{x} \in \mathbb{N}_{>0}^n : x_i \le b \}$ 

$$(1)$$

where the value *b* is the total number of considered BBs. Fig. 1 shows the graphical representation of the considered MOOP. The  $\bar{x}$  vector represents application processes and values in the decision variable space are BB instances, so the solution space is bounded

by the total number of BBs. The cost functions depends on different metrics evaluated/estimated during the co-design flow [2, 14]. Then, a multi-objective genetic algorithm is used to individuate an approximation of the  $\rho F^*$  in a single run, where, starting from the phenotype space, the solution has been encoded considering application processes and BBs. Considering the decision variable space size, it is possible to evaluate the number of solutions as the permutations with repetition of *n* application processes, that compose the solution vectors  $\bar{x}$ , allocated on *b* BBs, so the space size is  $b^n$ . Note that the feasible design space  $\Omega$  is convex, but it is not possible to say anything about the feasible criterion space *Z*. As said before, a GA is used to solve the HW/SW partitioning and mapping problem.



Figure 1: Reference MOOP.

Respect to the GA population, each individual is characterized by a "fitness", which is the value of the cost functions calculated in correspondence of each individual. Applying a WSM with respect to the MOOP considered in this work, it is possible to define the utility function that quantifies the quality of each individual of the GA population. The cost functions (called *indexes*) and the methods used to evaluate them at each iteration have been defined in [4, 16, 17]. In this context, the instance of an individual  $\bar{x}$  is defined as a vector where the indexes represents processes and the values represents BB instances.

## 4 PROPOSED METHODOLOGY

The proposed methodology try to respond to several research questions: (1) How is it possible to explicitly introduce decision maker preferences? (2) How is it possible to offer the possibility to correctly understand the cost functions magnitude without a Pareto trade-off analysis? (3) How is it possible to tune the DSE solution to fulfill decision maker preferences?



Figure 2: GA with WLE schema.

To introduce decision maker preferences, in this work a ranking assignment has been used, so it is possible to assign four relative importance values:  $\lambda_i = 0$  - this cost function will not be considered at all;  $\lambda_i = 1$  - low balance, the tool considers cost function *i* with low importance (it will be half weighted in the linear combination);  $\lambda_i = 2$  - normal balance, the tool considers cost function *i* equal to others;  $\lambda_i = 3$  - high balance, the tool considers cost function *i* with high importance (it will be double weighted in the linear combination); Starting from this ranking, the weights assigned to each cost function will change depending on the GA evolution. Fig. 2 shows the proposed GA algorithm extended to incorporate the WLE method. The starting point is the selection of GA parameters. In this work different performance metrics [18] (i.e. Hypervolume, HV, Ratio on Non-dominated Individuals, RNI, Generational Distance, GD, Inverted Generational Distance, IGD, Spread,  $\Delta$ , and Epsilon,  $\epsilon$ ) have been used to find the best parameters related to the GA. Next step involves the initialization activities. Since the starting population depends only on random values, the initial weights vector values are set equal to  $\omega_i(0) = 1/k$ , where k is the number of the cost functions considered by the decision maker. After the evaluation activity (and possible *Elitism*), the tool reaches the WLE step. Then, the classical GA steps are applied and weights are changed at each iteration according to WLE, until the stop criteria is satisfied, as presented in the equations described in the Definition 4.1:

Definition 4.1. (Weights Linear Equalization). The weights  $\omega_i(t)$  associated to each cost function  $F(\bar{x})$  at iteration t > 0 ( $t \in \mathbb{N}$ ) are evaluated by solving the linear system:

$$\begin{cases} \lambda_1 \cdot \mu_1(t) \cdot \omega_1(t) = \lambda_2 \cdot \mu_2(t) \cdot \omega_2(t) \\ \cdots \\ \lambda_{k-1} \cdot \mu_{k-1}(t) \cdot \omega_{k-1}(t) = \lambda_k \cdot \mu_k(t) \cdot \omega_k(t) \\ \omega_1(t) + \omega_2(t) + \cdots + \omega_k(t) = 1 \end{cases}$$
(2)

The values  $\bar{\lambda} = \{\lambda_1, \dots, \lambda_k\}$  are assigned by the decision maker and  $\mu_i(t) = [\sum_{j=1}^{I(t)} F_i(\bar{x}_j(t))]/I(t)$ . I(t) is the population size at iteration *t*, so weights are tuned by the average cost functions values at each step of GA.  $\bar{x}_j(t)$  is the individual *j* in the population at iteration *t*. Using this method it is possible to make explicit the decision maker suggestions (by means of  $\lambda_i$  values), avoid to represents trade-off solution in a *k*-dimensional design space (*k*>3), and to propose a feasible solution tuned with respect to the average cost functions values. Equation 2 admits a solution that can be found with different methods. A possible one is to apply an iterative approach, where the computational cost is  $O(k^2)$ , so it does not take too much time if *k* is small [1], and it is normally the case for the number of considered cost functions.

#### 5 RESULTS AND CONCLUSIONS

This section presents some experimental results related to the DSE step with GA+WLE, and some conclusions. The implemented GA uses a random parent selection, a random one-point crossover, a random mutation step and a fitness-based survivor activity. The maximum number of iterations is fixed to 100, the initial population size is fixed to  $10^4$  individuals, and the maximum population size is fixed to  $10^6$  individuals. In the context of this validation, the reference use case is a synthetic application, called *FirFirGCD*, composed by 8 processes [15]. The available BBs are: *bb*<sub>1</sub>, 16 MHz 8-bit **8051** CISC core with 128 byte of Internal RAM, 64KB of internal ROM (cost 10); *bb*<sub>2</sub>, 16 MHz 16-bit **PIC24** core with 14KB of internal ROM

and 1KB of internal RAM (cost 20);  $bb_3$ , 150 MHz 32-bit *LEON3* soft-processor with 2\*4 KB L1 caches, RAM size of 4096 KB and a ROM of 2048 KB (cost 100);  $bb_4$ , 50 MHz *Spartan3an* (cost 400);  $bb_5$ , 250 MHz *Virtex-7* (cost 900). The maximum number of instances for each  $bb_i$  is 2 (i.e., the total number of different instances of BBs is 10), the maximum number of instances of  $bb_i$  considered into the DSE is equal to the number of application processes (i.e., 8) and  $bb_i$  are supposed to communicate by means of a shared bus. The decision variable space size is  $10^8$ .

The selected cost functions to be taken into account by the DSE are [4, 16, 17]: (1) *Affinity Index* - it indicates how much a process is suitable to be executed by a given processor technology; (2) *Parallelism Index* - it expresses the set of processes pairs that could be potentially working concurrently; (3) *Load Index* - it is the load (i.e. the processor utilization percentage) that each process, when implemented in SW, would impose to the processors in  $BB_i$  to satisfy imposed timing constraints; (4) *Cost Index* - it is related to the cost (monetary cost, design effort, or any other issues of interest for the designer) associated to each  $BB_i$  considered in the specific individual.



Figure 3: Pareto plots w.r.t. cost functions pairs.

Fig. 3 shows the Pareto trade-off analysis for different cost functions pairs. It is worth noting that it is not possible to visually analyze simultaneously all the cost functions with a k>3 design space plot (without some assumption or trying to visualize the design space with different graphs [7]).



Figure 4: Normal GA and WLE-GA average population utility function values.

Fig. 4 presents the weighted utility function trend w.r.t. the different iterations and the best solution found at each iteration. The GA modified with WLE (the violet curves) has an higher average utility function value ( $\approx$  9% higher), and an higher standard deviation value (the curves in the top and bottom that delimit the violet

area,  $\approx 17\%$  higher) respect to normal GA, so the WLE method increases the GA variance introducing more diversity respect to the normal GA population. The GA+WLE drives in a better way the GA evolution since it also has a better performance metrics trend [18], considering the evolution among the initial population (common to all) and the  $100^{th}$  iteration, as shown in Table 1 (e.g. greater values for GD, IGD and  $\epsilon$ , and lower ones for RNI, means that the final  $\rho F^*$  is better as well).

 Table 1: DSE Performance Parameters Analysis (values in bold are the most representative ones).

Туре	HV	RNI	GD	IGD	Δ	e
Normal GA	0.870791	0.000343	0.043285	0.023377	1.073481	0.088622
GA+WLE	0.871635	0.000298	0.047934	0.027974	1.333774	0.103508
GA+Elit.	0.867554	0.000286	0.092763	0.030771	1.680116	0.227712
GA+WLE+Elit.	0.869643	0.000292	0.074056	0.027974	1.575254	0.172427

Instead, the best WLE solution found at each iteration is lower in terms of utility function value for the different weights assigned (Fig. 4). In terms of final best solution (at iteration 100), it is worth noting that the normal solution is an all-HW solution (2 Spartan3an and 1 Virtex-7, where the utility function is 0.212), because the best solution for the Load index is the all-HW implementation, while the WLE best solution is a real equalized trade-off among the different metrics (2 8051, 2 PIC24 and 1 Spartan3an, where the utility function is 0.118).



Figure 5: Execution times comparison.

Finally, Fig. 5 presents the execution time related to 4 different GA implementations: standard GA, GA with elitism, GA with WLE and GA with WLE and elitism. The elitism has been implemented to save the best solution at each iteration, and also to save the solutions with the best single cost function value at each iteration. Unfortunately, the Elitism feature increases the execution time in terms of about 30% respect to the normal one, but behaves better w.r.t other GA performance metrics (as shown in Table 1). Introducing WLE in the GA increases the DSE execution time in terms of about 2-4%, which is acceptable compared to the opportunity of finding a result that better takes into account decision maker preferences. In conclusion, a self-equalization of weights in the utility function guarantees compliance with the decision maker qualitative preferences, in a ranking-based "a priori" method that converges to sub-optimal solutions, while not introducing relevant overhead in term of execution time. However, since the use of elitism (and other methods to increase GA performance) introduces relevant overhead

in term of execution time, future works involves the exploitation of parallel programming techniques, considering the possibility to implement parallel GAs still taking into account decision maker preferences in the whole design flow.

## ACKNOWLEDGMENTS

This work has been partially supported by the ECSEL RIA 2016 MegaM@Rt2 and AQUAS projects.

#### REFERENCES

- A Bakari and I Dahiru. 2018. Comparison of Jacobi and Gauss-Seidel Iterative Methods for the Solution of Systems of Linear Equations. Asian Research Journal of Mathematics 8 (02 2018), 1–7. https://doi.org/10.9734/ARJOM/2018/34769
- [2] D. Ciambrone, V. Muttillo, L. Pomante, and G. Valente. 2018. HEPSIM: An ESL HW/SW co-simulator/analysis tool for heterogeneous parallel embedded systems. In *Mediterranean Conference on Embedded Computing (MECO)*. IEEE, Budva, Montenegro, 1–6.
- [3] Gianluca Palermo Giovanni Mariani Fabrizio Castro Cristina Silvano, Vittorio Zaccaria. 2018. Multicube Explorer.
- [4] Daniele Di Pompeo, Emilio Incerto, Vittoriano Muttillo, Luigi Pomante, and Giacomo Valenete. 2017. An Efficient Performance-Driven Approach for HW/SW Co-Design. In Proceedings of the 8th ACM/SPEC on International Conference on Performance Engineering (ICPE '17). ACM, New York, NY, USA, 323–326. https: //doi.org/10.1145/3030207.3030239
- [5] M. A. Gennert and A. L. Yuille. 1988. Determining The Optimal Weights In Multiple Objective Function Optimization. In [1988 Proceedings] Second International Conference on Computer Vision. IEEE, Tampa, FL, USA, USA, 87–89.
- [6] C.L. Hwang and K. Yoon. 1981. Multiple attribute decision making: methods and applications : a state-of-the-art survey. Springer-Verlag, Berlin Heidelberg.
- [7] A. Ibrahim, S. Rahnamayan, M. V. Martin, and K. Deb. 2016. 3D-RadVis: Visualization of Pareto front in many-objective optimization. In 2016 IEEE Congress on Evolutionary Computation (CEC). IEEE, Vancouver, BC, 736–745.
- [8] Yaochu Jin, Tatsuya Okabe, and Bernhard Sendho. 2001. Adapting Weighted Aggregation for Multiobjective Evolution Strategies. In Evolutionary Multi-Criterion Optimization. Springer, Berlin, Heidelberg, 96–110.
- [9] D.F. Jones, S.K. Mirrazavi, and M. Tamiz. 2002. Multi-objective meta-heuristics: An overview of the current state-of-the-art. European Journal of Operational Research 137, 1 (2002), 1 – 9.
- [10] W.-J. Li K.-S. Zhang, Z.-H. Han and W.-P. Song. 2008. Bilevel Adaptive Weighted Sum Method for Multidisciplinary Multi-Objective Optimization. AIAA Journal 46, 10 (2008), 2611–2622.
- [11] I.Y. Kim and O.L. de Weck. 2005. Adaptive weighted-sum method for bi-objective optimization: Pareto front generation. *Structural and Multidisciplinary Optimization* 29, 2 (01 Feb 2005), 149–158.
- [12] R.T. Marler and J.S. Arora. 2010. The weighted sum method for multi-objective optimization: New insights. *Structural and Multidisciplinary Optimization* 41, 6 (Jan. 2010), 853–862.
- [13] Vittoriano Muttillo, Giacomo Valente, Daniele Ciambrone, Vincenzo Stoico, and Luigi Pomante. 2018. HEPSYCODE-RT: A Real-Time Extension for an ESL HW/SW Co-Design Methodology. In Workshop on Rapid Simulation and Performance Evaluation: Methods and Tools (RAPIDO '18). ACM, New York, NY, USA, Article 6, 6 pages.
- [14] Vittoriano Muttillo, Giacomo Valente, and Luigi Pomante. 2018. Criticalitydriven Design Space Exploration for Mixed-Criticality Heterogeneous Parallel Embedded Systems. In PARMA-DITAM '18. ACM, New York, NY, USA, Article 6, 6 pages.
- [15] V. Muttillo, G. Valente, and L. Pomante. 2018. Design Space Exploration for Mixed-Criticality Embedded Systems Considering Hypervisor-Based SW Partitions. In Euromicro Conference on Digital System Design (DSD). IEEE, Prague, Czech Republic, 740–744.
- [16] L. Pomante. 2011. System-level design space exploration for dedicated heterogeneous multi-processor systems. In *International Conference on Application-specific Systems, Architectures and Processors.* IEEE, Santa Monica, CA, USA, 79–86.
- [17] L. Pomante. 2013. HW/SW co-design of dedicated heterogeneous parallel systems: an extended design space exploration approach. *IET Computers Digital Techniques* 7, 6 (November 2013), 246–254.
- [18] N. Riquelme, C. Von LÄijcken, and B. Baran. 2015. Performance metrics in multiobjective optimization. In 2015 Latin American Computing Conference (CLEI). IEEE, Arequipa, Peru, 1–11.
- [19] Thomas L. S. 2003. Decision-making with the AHP: Why is the principal eigenvector necessary. European Journal of Operational Research 145, 1 (2003), 85–91.
- [20] Eckart Z., Kalyanmoy D., and Lothar T. 2000. Comparison of Multiobjective Evolutionary Algorithms: Empirical Results. *Evol. Comput.* 8, 2 (June 2000).