HypAp: A Hypervolume-Based Approach for Refining the Design of Embedded Systems

Rabeh Ayari, Mahdi Nikdast, Imane Hafnaoui, Giovanni Beltrame, and Gabriela Nicolescu

***Abstract*—Designing complex embedded systems requires simultaneous optimization of multiple system performance metrics that can be addressed by applying Pareto-based multi-objective optimization techniques. At the end of this type of optimization process, designers always face Pareto fronts (PFs) including a large number of near-optimal solutions from which selecting the most proper system implementation is potentially infeasible. In this letter, for the first time, we present HypAp, a hypervolume-based automated approach to systematically help designers efficiently choose their preferred solutions after the optimization process. HypAp is a two-stage approach relying on clustering Pareto optimal solutions and then finding a subset of solutions that maximizes the hypervolume by using a genetic algorithm. The performance of HypAp is evaluated through applying HypAp to the PF by the case study of mapping applications on network-on-chip-based heterogeneous MPSoC.**

***Index Terms*—Design space exploration, designer support, hypervolume, multi-objective optimization, Pareto front.**

I. Introduction

NEW APPLICATIONS, such as Internet-of-things, have driven the integration of a large number of functionalities into modern embedded systems. As a result, more software and hardware components need to be used, considerably enlarging the design space of modern embedded systems [1]. Furthermore, the design of such systems involves the optimization of multiple competing objectives (i.e., multiobjective optimization) in which the preferred system configuration can be realized by finding the best tradeoffs between these objectives (e.g., energy, throughput, etc.) usually expressed using fitness functions [2]. This optimization process is referred to as the design space exploration (DSE), allowing designers to find a near-optimal solution. It is worth mentioning that there is no unique optimal solution but rather a set of efficient solutions, also known as Pareto solutions. The set of all the Pareto solutions constitutes the Pareto front (PF) [3].

A common approach for the DSE is to employ Pareto-based multiobjective optimization techniques [4]. These approaches, such as evolutionary algorithms [5], aims at finding the solutions approximating the PF [6]. The PF represents a range of performance tradeoffs that can be achieved by adjusting different design parameters. As a result, such approaches can simultaneously optimize different competing objectives. However, the large size of the PF constitutes the major shortcoming of such techniques: the main challenge for designers, is how to effectively select their preferred solution(s).

From the designers' perspective, evaluating all the near-optimal solutions (i.e., nondominated solutions) is unrealistic. On the other hand, selecting a preferred solution from a large PF is potentially infeasible. Therefore, a possible solution is to provide a refined representation of the PF optimal solutions (i.e., a subset of solutions belonging to the PF) to which we refer as the reduced Pareto front (RPF). The novel contribution of this letter is in developing HypAp, a hypervolume-based automated approach generating an RPF that is much smaller in size compared to the PF, and yet maintains the main characteristics of the PF. HypAp helps the system designers effectively to choose their preferred solutions, now from an RPF includes fewer solutions.

HypAp is a two-stage approach consisting of clustering PF solutions, and then finding a subset of solutions from each cluster that maximizes the hypervolume by using a genetic algorithm. We define several quality indicators, including hypervolume, nonuniformity, and outer-diameter, to evaluate the similarity and effectiveness of the RPF compared to the PF. As a case study, we apply HypAp to the PF of a network-on-chip (NoC) mapping optimization problem.

The rest of this letter is organized as follows. Section II briefly reviews related work. In Section III, we detail the proposed two-stage approach, called HypAp. Section IV evaluates the effectiveness of the proposed approach and includes the case study of an NoC mapping problem and the results. Finally, Section V concludes this letter.

II. Related Work

High-level design of embedded systems can be performed by employing different automated approaches relying on Pareto-based algorithms. Nevertheless, the major drawback of such works is that designers cannot easily choose their preferred solution(s) from the resulting large-size PF.

Designers' preferences can be ignored [7] or considered in different multiobjective optimization approaches before the optimization (i.e., *a-priori* methods), after the optimization (i.e., *a-posteriori* methods), or interactively during the optimization process. An *a-priori* approach was presented in [8] to guide the search in the design space toward a preferred region. In [9], an *a-priori* method was proposed based on

Manuscript received February 21, 2017; accepted April 3, 2017. Date of publication April 18, 2017; date of current version August 25, 2017. This manuscript was recommended for publication by D. Sciuto. *(Corresponding author: Rabeh Ayari.)*

The authors are with the Department of Computer Engineering, École Polytechnique de Montréal, Montreal, QC H3T 1J4, Canada (e-mail: rabah.ayari@gmail.com).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/LES.2017.2695118

assigning a relative preference factor (i.e., weight) to each design objective considering the designer's preferences. An *a-priori* technique to optimize one objective and assign other objectives with an upper constraint was proposed in [10]. This method can alleviate the difficulties faced by the scalarization approach when solving problems with concave PFs. Such techniques, however, require extensive knowledge of the problem in advance and cannot guarantee the Pareto-optimality of the obtained solutions, while missing some regions including promising solutions.

A-posteriori preference consideration methods are employed when the relative importance of the objectives is unknown. Such techniques guarantee that no superior solution will be missed and the feasible implementations are available for evaluation. In [11], a clustering approach was proposed to find an RPF by applying K-means method and then picking the closest solution to the centroid of each cluster as the candidate of that cluster. A graphical representation, called level diagrams, for n-dimensional PF analysis was proposed in [12]. By clustering the PF and then employing level diagrams to represent and analyze the RPF, [13] proposed a two-stage approach aimed at identifying a limited number of representative solutions to be presented to the designer.

While there is no unique definition of an optimal selection, all the aforementioned *a-posteriori* methods failed to comprehensively evaluate the quality of their RPFs in terms of different Pareto quality indicators. In this letter, an effort is made to find an RPF that highly represents the PF. Particularly, we evaluate the effectiveness of the RPF in terms of several quality indicators, such as the Pareto-coverage, distribution, uniformity, and extent, and hence, quantify the characteristics of the PF that have been maintained in the RPF.

III. HypAp: Our Proposed Approach

This section details our proposed HypAp, developed to systematically help with the design choices obtained after the optimization process (i.e., *a-posteriori* approach). HypAp is a two-stage approach: Pareto optimal solutions will be first clustered and then a subset of solutions that maximizes the hypervolume will be selected. Please note that different performance metrics that need to be optimized during the DSE have a large size or great variability due to their different distributions and orders of magnitudes. Consequently, this kind of variability can lead to a knock-on effect on the clustering result. Therefore, prior to clustering, we normalize the fitness values by adjusting them to notionally averages (also known as feature scaling) which leads to a better symmetry, and hence, more reliable learning.

A. Clustering Pareto Front Optimal Solutions

The first stage of HypAp aims at clustering PF optimal solutions with respect to their similarity, facilitating the application of the genetic algorithm in the next stage. We consider using K-means, which is an unsupervised learning algorithm [14], to perform the clustering. K-means groups a given set of solutions into k different clusters through calculating the centroid for each cluster, and then assigning each solution to the cluster with the nearest centroid. Finding the solutions that belong to the same cluster, K-means considers employing the euclidean distance. As a result, the sum of squared distances, d, between the centroid of each cluster, x_j, and the solutions on the PF, x_i, should be minimized

$$d = \sum_{j=1}^{k} \sum_{i=1}^{n} \left(x_i - \bar{x}_j\right)^2 \tag{1}$$

in which k and n are the number of clusters and solutions, respectively.

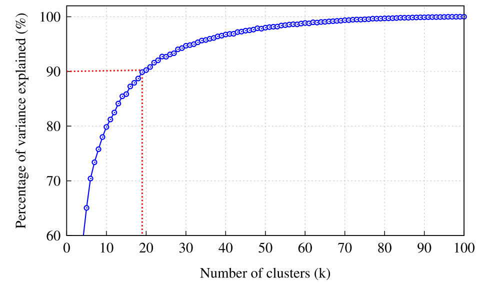


Fig. 1. Using the elbow method to determine the number of clusters: the percentage of variance explained versus the number of clusters.

In any clustering technique, choosing the right number of clusters (i.e., k) is challenging. The best choice for k is often ambiguous as it highly depends on the shape and scale of the PF. In this letter, we consider using the Elbow method to determine k [15]. Using this method, we consider the number of clusters based on the percentage of variance explained[1] (see Fig. 1), which is the ratio of the between-cluster variance to the total variance. As a result, based on Fig. 1, we choose the number of clusters k corresponding to the percentage of variance explained of 90%, after which the variance explained gain marginally drops, and hence, increasing the number of clusters will no longer add significant information.

B. Hypervolume Maximization

Although the resulted clusters from the first stage are informative, the number of solutions existing in each cluster can still be very large for the designer to make right choices. Furthermore, selecting solutions maintaining the information of the PF from each cluster is challenging. We address this problem by employing an optimization approach seeking ideal subsets of the PF that maximizes the hypervolume, as we discuss in the following. As a result, the problem can be seen as a variant of the maximum coverage problem where the RPF Λ', in which $|\Lambda'| = k$ and k is the number of clusters, is obtained in advance. Our aim is to maximize the diversity of the solutions belonging to Λ' and the coverage of the PF Λ.

The hypervolume indicator (also known as Lebesgue measure [16]) is one of the most popular quality indicators for multiobjective optimization. It can solely capture the coverage of the solutions and their distances from the true PF. Therefore, a subset with a larger hypervolume is likely to present a better set of tradeoffs. Considering solutions as points in an objective space, the hypervolume is the n-dimensional space that is contained by a solution relative to a reference point defined

[1] In statistics, variance explained measures the proportion to which a mathematical model accounts for the variation of a given data set.

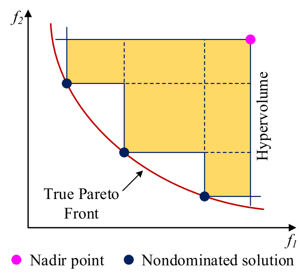


Fig. 2. Hypervolume indicator for the nondominated solutions with respect to the nadir point in a biobjective optimization problem.

as the nadir point (depicted in Fig. 2) in our approach. Nadir point is the worst-known value in each dimension.

The hypervolume indicator for a normalized PF $\bar{\Lambda}$ is defined as [17]

$$H(\bar{\Lambda}) = \int_{\mathcal{Y}} 1\left[\exists x \in X : f(x) \prec y \prec z\right] dy \tag{2}$$

in which $z \in \Re^N$ is the reference point (i.e., the nadir point in our method), f is the vector obtained by stacking the objectives, and $\prec$ is the dominance operator defined as $x \prec y \Leftrightarrow (x_1 < y_1) \wedge \cdots \wedge (x_m < y_m)$.

The application of the hypervolume indicator has been greatly limited by the high computation cost of existing algorithms for the exact hypervolume computation as it is a #P-Hard problem. We consider a methodology based on Monte-Carlo sampling [18] to estimate the hypervolume, and hence improve its computation time, while sacrificing the accuracy within a tolerable limit. Since using the exact hypervolume values are not crucial in our optimization process, a Monte-Carlo estimation is performed in order to compute the percentage of random points in the performance space to be dominated by the PF. The estimation error related to the employed Monte-Carlo approach after considering 10^6 random points is within 0.02% which is highly reasonable.

Finally, due to the increasing number of possible configurations, selecting the RPF is nontrivial and considered as an NP-hard problem. Therefore, one needs to use meta-heuristics to find a good approximation of the optimal configuration. In order to find this configuration, we implement a genetic algorithm aiming on the maximization of the hypervolume. After that, HypAp successfully identifies an RPF that highly represents the PF, as we indicate in the next section.

IV. Evaluations and Results

A. *Quality Indicators*

Several unary quality indicators have been proposed to evaluate the quality of solution sets in terms of convergence, and diversity. In this letter, convergence is ignored since we are assuming that the PF is the input of HypAp. In order to study the quality of the RPF, we opted for the following.

1) Nonuniformity: An RPF with a lower nonuniformity is more evenly distributed and better estimates the PF. Given a normalized PF $\bar{\Lambda}$, nonuniformity is given by [19]

$$\mathrm{NU}(\bar{\Lambda}) = \sum_{i=1}^{|\bar{\Lambda}|-1} \frac{|d_i - \bar{d}|}{\sqrt{m}(|\bar{\Lambda}| - 1)} \tag{3}$$

TABLE I
Comparing the RPF Obtained Using HypAp and K-Means With the PF of the NoC Case Study

Quality indicator	PF	HypAp RPF	K-means RPF [11]
Hypervolume	0.089	0.078	0.051
Non-uniformity	0.031	0.027	0.032
Outer diameter	0.364	0.243	0.261

where d_i is the euclidean distance between two consecutive solutions, and $\bar{d}$ defines the average distance. Also, $|\bar{\Lambda}| = k$ is the number of clusters, and m is the dimension of the design space. Please note that $0 \leq NU(\bar{\Lambda}) \leq 1$, where $NU(\bar{\Lambda}) = 0$ means that the set is uniformly distributed.

2) Outer Diameter: The outer diameter indicator aims at computing the distance between the ideal objective vector and the nadir objective vector of a given PF using a distance metric. This unary indicator is given by [20]

$$\mathrm{OD}(\bar{\Lambda}) = \max_{1 \leq i \leq n} w_i \left(\left(\max_{x \in \bar{\Lambda}} f_i(x) \right) - \left(\min_{x \in \bar{\Lambda}} f_i(x) \right) \right) \tag{4}$$

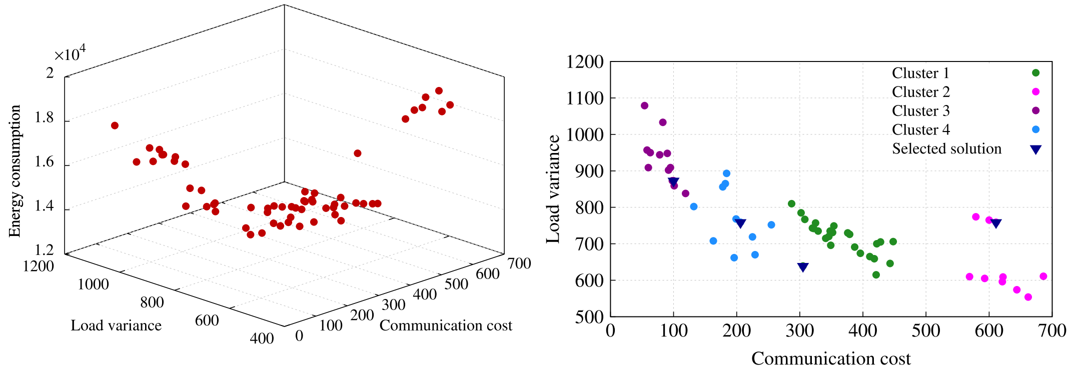
with weights $0 < w_i \leq 1$. If $\forall i \in [2, m]$, $w_i = 1$, then the outer diameter becomes the maximum extent over all the dimensions of the design space. The outer diameter has a low computation cost and it is agnostic to the problem features.

B. *Case Study*

In this section, we present the case study of an NoC mapping problem with three objectives; load variance, communication cost, and energy consumption [21]. In this problem, TGFF [22] is used to generate a set of purely synthetic task graphs with varying number of tasks to model the application. The architecture model consists of p types of processing elements interconnected by a Spidergon NoC topology.

We apply HypAp to the PF, depicted in Fig. 3(a), obtained by mapping the generated task graphs on the NoC architecture. HypAp first classifies the optimal solutions on the PF into four clusters. Solutions in each cluster are indicated using a unique color in Fig. 3(b). As can be seen, clusters represent different aspects of the PF with respect to the cost functions.

After clustering, HypAp selects the representative solutions (i.e., RPF), indicated in Fig. 3(b), based on maximizing the hypervolume. Employing the quality indicators defined before, Table I compares the RPF suggested by HypAP with the PF of the problem. For better comparison, Table I also considers the RPF obtained by applying the K-means method proposed in [11]. As can be seen, HypAp achieves a high hypervolume that is very close to the one for the PF, while it is also higher than the hypervolume obtained using K-means. In other words, HypAp proposes an RPF including only four solutions that roughly represents the same space coverage as the PF. Moreover, higher hypervolume also means that the selected solutions in HypAP are among the closest ones to the true PF. As a result, designers can easily choose from these four solutions based on their preferences. Moreover, HypAp enables the designers to refine their choices through iteratively discovering the neighborhood of the preferred solutions: each iteration can consider a new subset using a predefined radius r, representing the smallest disk containing $|\Lambda'| = k$ solutions.



(a) (b)

Fig. 3. (a) PF obtained by mapping the generated task graphs on the NoC architecture along with its (b) 2-D illustration including the clusters and RPF.

Considering the nonuniformity results in Table I, the RPF obtained using HypAp has a better uniformity compared with that from K-means and even the one from the PF itself. In other words, knowing that the PF is not uniformly distributed, HypAP suggests an RPF that indicates a higher uniformity: HypAp not only maintains the features of the PF in the RPF but also improves them when it is possible. Moreover, considering the outer-diameter results, the first observation is that the extent (i.e., spread of the data) is better using K-means, while HypAp still provides a good extent. Nevertheless, a deeper interpretation of the results obtained using this indicator shows that HypAp avoids selecting the solutions that are in the borders of the PF (i.e., extremity of the PF). It is worth mentioning that such selections are quite poor in terms of representativity: even if they result in a good extent on the PF, they cannot provide a good-enough reference point for the refinement process explained above. Consequently, designers will be guided toward fewer directions as no more improvement can be achieved through the selections made on the extremity of the PF.

V. Conclusion

This letter presents HypAp, which systematically help embedded systems designers choose their preferred solutions after the optimization process. HypAp first clusters the PF solutions, and then seeks an RPF that maximizes the hypervolume. The result of this preliminary work is critical for the design of complex embedded systems, in which designers require to choose from PFs including a large number of near-optimal solutions.

References

[1] T. A. Henzinger and J. Sifakis, "The embedded systems design challenge," in *Proc. Int. Symp. Formal Methods*, Hamilton, ON, Canada, 2006, pp. 1–15.

[2] M. D. Hill and M. Marty, "Amdahl's law in the multicore era," *Computer*, vol. 41, no. 7, pp. 33–38, 2008.

[3] J. Legriel, C. L. Guernic, S. Cotton, and O. Maler, "Approximating the Pareto front of multi-criteria optimization problems," in *Proc. TACAS*, Paphos, Cyprus, 2010, pp. 69–83.

[4] Y. Jin and B. Sendhoff, "Pareto-based multiobjective machine learning: An overview and case studies," *IEEE Trans. Syst., Man, Cybern. C, Appl. Rev.*, vol. 38, no. 3, pp. 397–415, May 2008.

[5] R. Drechsler *et al.*, *Evolutionary Algorithms for Embedded System Design*. Boston, MA, USA: Kluwer, 2002.

[6] T. Givargis, F. Vahid, and J. Henkel, "System-level exploration for Pareto-optimal configurations in parameterized systems-on-a-chip," in *Proc. ICCAD*, 2001, pp. 25–30.

[7] R. Ayari *et al.*, "Multi-objective mapping of full-mission simulators on heterogeneous distributed multi-processor systems," *J. Defense Model. Simulat. Appl. Methodol. Technol.*, 2016, doi: 10.1177/1548512916657907.

[8] R. Ayari, I. Hafnaoui, G. Beltrame, and G. Nicolescu, "Schedulability-guided exploration of multi-core systems," in *Proc. 27th Int. Symp. Rapid Syst. Prototyping Shortening Path Specification Prototype*, Pittsburgh, PA, USA, 2016, pp. 121–127.

[9] K. Miettinen and M. M. Mäkelä, "On scalarizing functions in multiobjective optimization," *OR Spectr.*, vol. 24, no. 2, pp. 193–213, 2002.

[10] Y. Y. Haimes *et al.*, "On a bicriterion formulation of the problems of integrated system identification and system optimization," *IEEE Trans. Syst., Man, Cybern.*, vol. SMC-1, no. 3, pp. 296–297, Jul. 1971.

[11] P. M. Chaudhari, R. V. Dharaskar, and V. M. Thakare, "Computing the most significant solution from Pareto front obtained in multi-objective evolutionary," *Int. J. Adv. Comput. Sci. Appl.*, vol. 1, no. 4, pp. 63–67, 2010.

[12] X. Blasco, J. M. Herrero, J. Sanchis, and M. Martínez, "A new graphical visualization of n-dimensional Pareto front for decision-making in multiobjective optimization," *Inf. Sci.*, vol. 178, no. 20, pp. 3908–3924, 2008.

[13] E. Zio and R. Bazzo, "A clustering procedure for reducing the number of representative solutions in the Pareto front of multiobjective optimization problems," *Eur. J. Oper. Res.*, vol. 210, no. 3, pp. 624–634, 2011.

[14] J. A. Hartigan and M. A. Wong, "Algorithm AS 136: A k-means clustering algorithm," *J. Roy. Stat. Soc. C (Appl. Stat.)*, vol. 28, no. 1, pp. 100–108, 1979.

[15] R. Tibshirani, G. Walther, and T. Hastie, "Estimating the number of clusters in a data set via the gap statistic," *J. Roy. Stat. Soc. B (Stat. Methodol.)*, vol. 63, no. 2, pp. 411–423, 2001.

[16] C. A. C. Coello, G. B. Lamont, and D. A. Van Veldhuizen, *Evolutionary Algorithms for Solving Multi-Objective Problems*, vol. 5. New York, NY, USA: Springer, 2007.

[17] E. Zitzler, D. Brockhoff, and L. Thiele, "The hypervolume indicator revisited: On the design of Pareto-compliant indicators via weighted integration," in *Proc. Int. Conf. Evol. Multi Criterion Optim.*, Münster, Germany, 2007, pp. 862–876.

[18] W. K. Hastings, "Monte Carlo sampling methods using Markov chains and their applications," *Biometrika*, vol. 57, no. 1, pp. 97–109, 1970.

[19] C. Erbas, *System-Level Modelling and Design Space Exploration for Multiprocessor Embedded System-On-Chip Architectures*. Amsterdam, The Netherlands: Amsterdam Univ. Press, 2006.

[20] E. Zitzler, J. Knowles, and L. Thiele, "Quality assessment of Pareto set approximations," in *Multiobjective Optimization*. Heidelberg, Germany: Springer, 2008, pp. 373–404.

[21] D. Belkacemi, Y. Bouchebaba, M. Daoui, and M. Lalam, "Network on chip and parallel computing in embedded systems," in *Proc. IEEE 10th Int. Symp. Embedded Multicore Many Core Syst. Chip (MCSoC)*, 2016, pp. 146–152.

[22] R. P. Dick, D. L. Rhodes, and W. Wolf, "TGFF: Task graphs for free," in *Proc. Workshop Hardw. Softw. Codesign*, Seattle, WA, USA, 1998, pp. 97–101.