Automatic Operating Point Distillation for Hybrid Mapping Methodologies

Behnaz Pourmohseni*, Michael Glaß†, Jürgen Teich*
*Friedrich-Alexander-Universität Erlangen-Nürnberg (FAU), Germany, †Ulm University, Germany
Email: *{behnaz.pourmohseni, juergen.teich}@fau.de, †michael.glass@uni-ulm.de

***Abstract*—Efficient execution of applications on heterogeneous many-core platforms requires mapping solutions that address different aspects of run-time dynamism like resource availability, energy budgets, and timing requirements. Hybrid mapping methodologies employ a static design space exploration (DSE) to obtain a set of mapping alternatives termed operating points that trade off quality properties (compute performance, energy consumption, etc.) and resource requirements (number of allocated resources of each type, etc.) among which one is selected at run-time by a run-time resource manager (RRM). Given multiple quality properties and the presence of heterogeneous resources, the DSE typically delivers a substantially large set of operating points handling of which may impose an intolerable run-time overhead to the RRM. This paper investigates the problem of truncation of operating points termed *operating point distillation*, such that (a) an acceptable run-time overhead is achieved, (b) online quality requirements are met, and (c) dynamic resource constraints are satisfied, i.e., application embeddability is preserved. We propose an automatic design-time distillation methodology that employs a hyper grid-based approach to retain diverse trade-off options wrt. quality properties, while selecting representative operating points based on their resource requirements to achieve a high level of run-time embeddability. Experimental results for a variety of applications show that compared to existing truncation approaches, proposed methodology significantly enhances the run-time embeddability while achieving a competitive and often improved efficiency in the distilled quality properties.**

***Index Terms*—Operating point distillation, hybrid application mapping, design space exploration, run-time management, heterogeneous many-core systems.**

I. Introduction

Future multi-application embedded systems highly depend on many-core platforms comprising an extensive number of processing elements (PEs) interconnected via a network-on-chip (NoC) [1]. Heterogeneous many-core platforms incorporate different types of PEs with distinct features to address the ever increasing diversity in demands of applications. To embed an application with concurrent tasks in a many-core platform, mapping methodologies consider different mapping alternatives to find a configuration that meets application requirements and satisfies system constraints.

In [2], mapping methodologies for multi-/many-core systems are classified into *design-time* and *run-time* strategies. Design-time strategies, e.g. [3], typically target systems with a fixed set of applications and rather static workloads. As they analyze mapping alternatives at design-time, they can afford compute-intensive design space exploration (DSE) approaches to find (near) optimal solutions. Run-time strategies, e.g. [4], consider dynamic application mixes and/or dynamic workload scenarios and are classified in *on-the-fly* and *hybrid* approaches [2]. On-the-fly methodologies derive mapping solutions at run-time, but, subject to the available run-time compute power, they might deliver inefficient mappings or even fail to find an adequate solution. As a remedy, hybrid methodologies, e.g. [5], [6], have recently emerged which (a) employ a design-time DSE to find multiple mapping alternatives termed *operating points* to address various run-time scenarios and (b) provide them to a light-weight run-time resource manager (RRM) to select the most fitting operating point for each run-time scenario.

TABLE I
Average run-time overhead of the feasibility check of one operating point on a desktop computer (Intel i7-4770) in ms.

Many-core NoC size	application size in number of tasks			
	18	14	11	7
6×6	8.07	7.34	1.90	0.79
8×8	25.49	27.70	6.88	2.41
10×10	134.79	115.24	31.99	4.67

The DSE evaluates the solutions wrt. a number of *quality objectives*, e.g. compute performance, energy efficiency, or reliability, and multiple *resource objectives*, e.g. the number of allocated resources of each type, and retains a set of *Pareto-optimal* operating points. Each operating point corresponds to a certain demand of specific resources which may or may not be available at run-time. Therefore, the RRM checks—at run-time—whether a solution meets the platform resource constraints, an NP-complete process called *feasibility check* [5]. Given multiple quality objectives and the presence of heterogeneous resources, the DSE typically delivers a substantially large set of operating points, which may impose an intolerable feasibility check overhead to the RRM. As an example, Table I specifies the average feasibility check run-time overhead of one operating point for a variety of applications and NoC-based platform sizes, performed on a desktop computer (Intel i7-4770). With a feasibility check overhead of dozens to hundreds of milliseconds[1] for medium-sized embedded platforms, checking hundreds of operating points at run-time might render hybrid mapping strategies de-facto impractical due to the immense timing overhead of the RRM.

This paper presents an automatic design-time approach to reduce the number of operating points obtained from the DSE to achieve an acceptable run-time overhead. This reduction step which we call *operating point distillation* can be seamlessly integrated in the standard hybrid mapping flow as illustrated in Fig. 1. The goal of operating point distillation is to reduce a given set of operating points such that the retained solutions (a) deliver *diverse* trade-off alternatives wrt. the quality objectives to address the—at design-time unknown—run-time quality requirements and (b) exhibit efficient resource combinations to enhance the *embeddability* of the application in view of

[1]For *embedded* target platforms, the feasibility check overhead is expected to be one or two orders of magnitude higher than measures reported in Table I.

978-3-9815370-8-6/17/$31.00 ©2017 IEEE

a dynamic run-time resource availability. For the former, various techniques exist in the domain of multi-objective optimization (MOO) while the latter requires special attention: Operating points with (near) optimal quality measures typically exhibit inferior run-time embeddability, i.e., they have lower chances of satisfying the dynamic run-time resource constraints as they depend on relatively large sets of resources and/or engage powerful yet rare resources like hardware accelerators. On the other hand, operating points with *modest* resource requirements exhibit a seamless embeddability, but might have inferior quality measures. While some hybrid mapping approaches, e.g. [6], [7], employ simple distillation techniques, to the best of our knowledge, there exists no automatic and configurable distillation methodology that considers the outlined interdependency of quality- and resource objectives.

In this paper, a two-level distillation methodology is proposed. First, a hyper grid-based approach is employed in the space of quality objectives to ensure that a diverse set of quality trade-off alternatives are retained. Subsequently, from within the grid cells, representative operating points are distilled based on their resource requirements to support run-time embeddability. By configuring the hyper grid, proposed approach can distill operating point sets of any size to achieve acceptable run-time overheads for any many-core system and application domain. Opposed to existing truncation techniques, experimental results show that the proposed approach significantly enhances the run-time embeddability while achieving a competitive and often improved efficiency in distilled quality trade-offs. The rest of the paper is organized as follows: Section II discusses related work in the field. Section III presents the proposed distillation approach. Experimental results are presented in Section IV before the paper is concluded in Section V.

II. Related Work

Some recent hybrid mapping strategies reduce the number of operating points they provide to the RRM: In [6], for each combination of resources, the operating point with the highest throughput and the one with the lowest energy consumption are selected, among which the Pareto-optimal points in the space of energy-throughput-resources are distilled. In [7], for each resource combination, the operating point with the highest $\frac{\text{throughput}}{\text{energy}}$ is retained. Although both approaches reduce the number of operating points, they retain representative solutions for every resource combination which might prove impractical as the number of resource combinations grows with application size (number of tasks) and platform heterogeneity (number of resource types). For instance, for a small 6×6 many-core platform with three resource types of equal numbers of instances, operating points of a small application with four tasks can have 34 distinct resource combination. For medium-sized applications with 10 and 12 tasks, this number grows up to 285 and 454, respectively. While, in an embedded system with an average feasibility check overhead of $100\,\text{ms}$ per operating point, far fewer points can be feasibly checked at run-time. To alleviate the overhead, [7] and [6] provide the RRM with a *presorted* set of operating points to eliminate *thorough* feasibility checks. Nonetheless, still a huge amount of operating points may have to be checked before the first embeddable one is found. Furthermore, a static prioritization of operating points for all possible on-line scenarios might degrade the efficiency

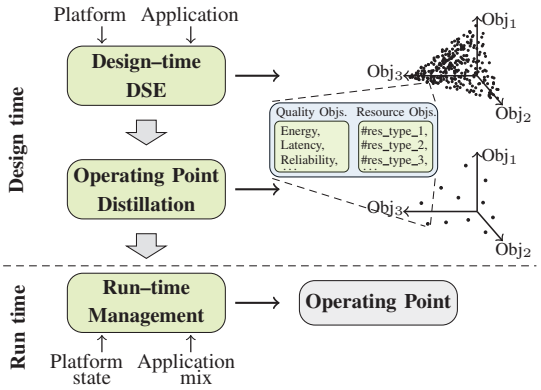

Fig. 1. Integration of a design-time operating point distillation step in the standard hybrid application mapping flow.

of the run-time management, cf. [8]. As a remedy, we propose a *configurable* distillation methodology that adapts the number of retained operating points to a provided distillation capacity.

In the domain of MOO, the *truncation* problem is well-studied [9]. MOO truncation techniques aim at reducing a set of multi-objective solutions such that the *spacing* among the retained ones is maximized [10]. Spacing indicates how evenly the solutions are distributed over the objective space, with a higher spacing implying a better *diversity*. Three prominent approaches from this domain are studied and evaluated in this paper: In [11], an *adaptive grid* is placed in the objective space, and the most crowded grid cells are truncated iteratively to reach a uniform distribution of remaining solutions. In [12], solutions with the smallest distance from their *nearest neighbors* are removed iteratively to maximize the distances among the remaining individuals. In [13], a dispersion metric termed *crowding distance*, calculated as the sum of the normalized distance between the left and the right neighbor of an individual in each objective, is employed to rank solutions. A solution with a low crowding distance exhibits objective values in the crowded regions and is more likely to be eliminated.

The MOO truncation techniques retain a diverse set of solutions. For the distillation problem at hand, however, they exhibit a degraded run-time embeddability, as they regard all objectives similarly and independent from one another. Quality objectives of an operating point address distinct run-time requirements. As different mixes of quality requirements might emerge at run-time, such as application performance demand or available energy budget, quality objectives must be investigated independently to deliver a diverse set of quality trade-off alternatives. Allocated resources of an operating point, however, *jointly* determine embeddability of the point wrt. a set of available resources. Thus, a proper distillation mechanism must consider the resource objectives of each point *collectively* when investigating its embeddability. The correlation among the resource objectives distinguishes the distillation problem at hand from the general MOO truncation problem. Since MOO truncation techniques do not consider this correlation, the high level of diversity they deliver does not necessarily indicate a high level of embeddability. A naïve remedy could be to merge the resource objectives into a single measure, e.g. using a weighted sum, and use it as a measure of embeddability. However, with such an approach, the diversity of the resource combinations cannot be investigated which

might degrade the run-time embeddability when exposed to diverse mixes of available resources at run-time. To investigate resource objectives collectively, we *separate* them from the quality objectives and propose a novel two-level distillation methodology: First, operating points are clustered into groups of comparable trade-offs wrt. quality objectives. Then, from different groups, representative operating points are selected based on their resource requirements to deliver efficient yet diverse resource combinations to enhance the run-time embeddability.

III. Proposed Distillation Approach

Given a set of Pareto-optimal operating points A in the space of m' *quality objectives* and m'' *resource objectives* (i.e., m'' resource types), and a distillation capacity c, the goal is to distill a subset of points B with $|B| = c$ such that diverse quality trade-offs and diverse yet efficient resource combinations are retained. In the following, first our approaches for addressing the distillation challenges are described. Subsequently, the proposed distillation methodology is presented.

A. Distillation Challenges

Configurable Distillation Capacity: Subject to the available run-time compute power, the number of operating points that can feasibly be handled by the RRM varies, which necessitates a configurable distillation capacity. To realize this, first, we divide the objective space into a number of trade-off *regions*, granted that solutions located in the same region exhibit *approximately* the same objective trade-off. *Representative* solution are then distilled from different trade-off regions. Configurable distillation capacity is achieved by adapting the number and size of the regions according to c.

We employ this approach in the space of the quality objectives to classify the operating points into *quality trade-off regions*. To create the regions, first, the m' quality objectives are normalized to the scale of 1 to (a desired value of) K. Then, an exponentially-spaced hyper-grid is placed in the objective space with the coordinates of $\sqrt[\text{div}]{K^i}$, $i \in \{0, ..., \text{div}\}$ in each objective, where div represents the number of regions per objective and is adaptive to c. Each grid cell represents one quality trade-off region. Fig. 2 (left) illustrates a subdivided exemplary space of two quality objectives with div$= 5$. Using such a setup, (a) the granularity of the regions, i.e., the degree of approximation, adapts to c, while (b) the intra-cell maximum ratio of approximation remains constant across the grid. Thus, diverse quality trade-off alternatives are distilled with their *spacing* adapting to the distillation capacity.

Collective Investigation of Resource Objectives: Since allocated resources of an operating point *jointly* contribute to its run-time embeddability, resource objectives must be considered *collectively*. Furthermore, to enhanced run-time embeddability, (a) distilled resource combinations must be *diverse* to satisfy diverse mixes of resource constraints at run-time, while (b) their *cost* must be kept as low as possible, i.e., operating points with smaller resource sets and fewer instances of costly and rare resources must be prioritized.

To investigate resource objectives collectively, we *separate* them from the quality objectives and use *PE-fronts* to consider the diversity and cost of the resource sets, *simultaneously*. The *PE-front* of a set of operating points is defined as the *smallest* subset of them that dominate the rest wrt. the resource objectives. We label the operating points with the index of the PE-front to which they belong and exploit the labels in the distillation step. For this purpose, we use the Pareto-ranking approach [14] in the space of the m'' resource objectives, in which, starting with index $i = 1$: (1) the PE-front of A is determined, (2) corresponding operating points are labeled with i and are eliminated from A, and (3) i is incremented, iteratively, until all points are labeled. Fig. 2 (right) illustrates the PE-front labeling approach in an exemplary space of two resource objectives. Since operating points in the same PE-front are mutually non-dominating wrt. the resource objectives, they exhibit diverse resource combinations. Furthermore, as operating points with lower PE-front indices require smaller resource sets, they have a higher priority of being distilled.

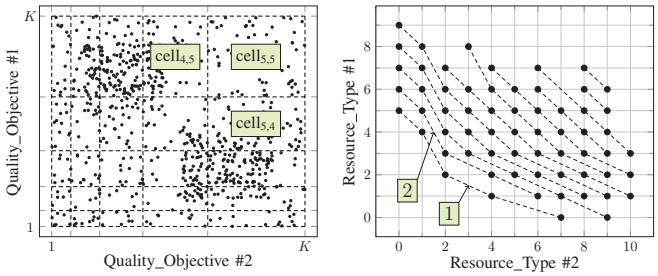

Fig. 2. Proposed approaches to address distillation challenges: Quality space subdivision with div$= 5$ in an exemplary space of two quality objectives (left). Pareto-ranking (PE-front labeling) of operating points wrt. two resource objectives where points belonging to the same PE-front are connected (right).

B. Proposed Distillation Methodology

We propose a two-level distillation methodology, in which, first, operating points are classified into regions of comparable trade-offs in the quality objectives, and then, representative points are distilled from different regions based on their resource requirements to retain diverse yet efficient resource combinations. The proposed distillation methodology is given in Algorithm 1, where—for the sake of simplicity and w.l.o.g.—an exemplar minimization problem with two quality objectives and multiple resource objectives is considered.

In the first step (lines 1–6), quality regions are formed by placing the previously described hyper-grid in the space of the quality objectives, and operating points are added to their corresponding regions. Fig. 3 illustrates the space of quality objectives. The semi-diagonal line divides the space in half: Operating points in the lower half excel in quality trade-offs, i.e., are *quality-dominant*, but have relatively expensive resource requirements. On the other hand, those in the upper half demand modest resource sets, i.e., are *resource-efficient*, but exhibit inferior quality trade-offs. To distill an efficient set of quality trade-offs, it suffices to retain representative solution form quality-dominant regions, i.e., dark-shaded cells in Fig. 3. Nonetheless, to enhance the run-time embeddability in platform states with low resource availability, resource-efficient solutions must be distilled as well. For this purpose, the parameter div (line 1) is derived from $\frac{1}{2}(\text{div} - 1)(\text{div}) < c$ so that, alongside the operating points from quality-dominant regions, at least one resource-efficient point is also distilled.

In the second step, first, operating points are labeled with their respective PE-front index (line 7). Subsequently, representative points are selected form quality-dominant regions (line 8–15), and eventually, resource-efficient points are distilled

Algorithm 1 Proposed two-level automatic distillation approach.

Input: Set of Pareto-optimal operating points A, Distillation capacity c.
Output: Distilled subset of operating points B.

1: div $\coloneqq \lceil \frac{1+\sqrt{1+8c}}{2} \rceil - 1$ ▷ Step 1: Quality space subdivision
2: **for all** $i \in \{1, ..., \text{div}\}$ **do**
3: **for all** $j \in \{1, ..., \text{div}\}$ **do**
4: $\text{cell}_{i,j} \coloneqq \{p \in A \mid \text{CELLID}(p) = (i,j)\}$
5: **end for**
6: **end for**
7: PARETORANK$(A, \{\text{res_objs}\})$ ▷ Step 2: Distillation
8: **for all** $x \in \{1, ..., \text{div} - 1\}$ **do**
9: **for all** $i \in \{1, ..., x\}$ **do**
10: $j \coloneqq x - i + 1$
11: $w \coloneqq \text{DISTILL}(\text{cell}_{i,j})$
12: $B \coloneqq B \cup \{w\}$
13: $A \coloneqq A \setminus \{\{w\} \cup \{p \in A \mid \text{cell}(w) \succ \text{cell}(p) \ \wedge \ w \succ_{PE} p\}\}$
14: **end for**
15: **end for**
16: **while** $|B| < c \ \wedge \ A \neq \varnothing$ **do**
17: $w \coloneqq \text{DISTILL}(A)$
18: $B \coloneqq B \cup \{w\}$
19: $A \coloneqq A \setminus \{\{w\} \cup \{p \in A \mid \text{cell}(w) \succ \text{cell}(p) \ \wedge \ w \succ_{PE} p\}\}$
20: **end while**

(line 16–20). Quality-dominant regions (cells) are explore in the order of dominance $\succ$ given as:

$$\text{cell}_{a,b} \succ \text{cell}_{c,d} \iff a \leq c \ \wedge \ b \leq d$$

and, from each region, one operating point w is selected using function DISTILL (line 11)—will be elaborated on, later—and is added to B (line 12). Since the quality measures are approximated, w might dominate further points $\{p\}$ in A if (1) $\text{cell}(w) \succ \text{cell}(p)$, and (2) w dominates p in resource objectives, denoted by $\succ_{PE}$. Dominated points are eliminated from A before proceeding to next cells (line 13). After distilling quality-dominant regions, resource-efficient points are selected from A (lines 16–20), in a similar procedure.

The function DISTILL employs a heuristic *estimation* of the relative run-time embeddability of a set of operating points and returns the point with the highest embeddability chances. For this purpose, it compares the operating points wrt. their (1) *PE-front* index and (2) resource cost, respectively. The resource cost of an operating point is calculated as its accumulative utilization rate of different resource types, where utilization rate of a resource type is defined as the proportion of its instances allocated by the operating point. For distillation of resource-efficient points, two additional parameters are integrated as well to incorporate quality measures of the points in case of equal run-time embeddability estimations: (3) distance of their quality regions from the origin, calculated as $(i-1)+(j-1)$ for $\text{cell}_{i,j}$, and (4) the number of already distilled points from their respective quality regions.

Pareto-ranking of operating points wrt. m'' resource objectives has a worst case time complexity of $O(m'' |A|^2)$ with $|A|$ being the cardinality of A [15]. Accordingly, Algorithm 1 has a worst case complexity of $O(m'' |A|^2 + (m' + m'' + 1) |A| c)$. Assuming $|m'| \approx |m''| \ll c \ll |A|$, the complexity of Pareto-ranking dominates the other terms, and the overall worst case time complexity becomes $O(m'' |A|^2)$.

IV. EXPERIMENTAL EVALUATION

This section presents the experimental results of evaluating the run-time embeddability and efficiency of the delivered quality properties of the proposed distillation methodology, in comparison with the previously discussed standard MOO truncation approaches: Adaptive Grid (AG) [11], Nearest Neighbor (NN) [12], and Crowding Distance (CD) [13]. We also incorporate the *quality front* (Q-front) which is the smallest subset of the initial set of operating points that dominate the remaining points wrt. the quality objectives.

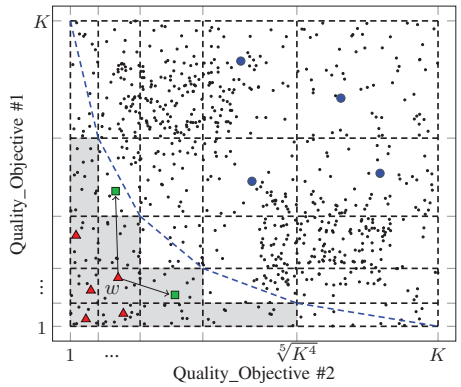

Fig. 3. An exemplary subdivided space of two quality objectives with div=5 in the fifth distillation iteration of the proposed methodology. Dark-shaded cells represent quality-dominant regions. Red triangles indicate distilled operating points, green squares are operating points dominated by w, and blue circles denote potential resource-efficient operating points.

To obtain the initial set of operating points, i.e., the *archive*, the DSE approach from [5] is employed which uses an evolutionary algorithm provided by the OPT4J framework [16]. The benchmark applications are from the domains of automotive industry (18 tasks), telecommunication (14 task), consumer (11 tasks), and networking (7 tasks) and are selected from the Embedded System Synthesis Benchmarks Suite (E3S) [17]. We use a 6×6 NoC-based heterogeneous many-core platform composed of three types of resources from the E3S benchmark suite with diverse numbers of instances. For each application, the DSE runs for 2,000 generations and considers latency and energy consumption as quality objectives, and the number of allocated resource of each type as resource objectives. For latency analysis, the approach from [5] is employed. For energy evaluation, the communication energy is calculated using the energy model from [18] with NoC technology parameters from [19] and a link length of 2 mm, while the processing energy is obtained from [17]. For each application, the DSE returns an unbounded *archive* of Pareto-optimal operating points which is provided as input to the distillation methodologies under study. We perform 10 DSE runs per application to reinforce the evaluations.

For run-time embedding of operating points, we use the backtracking-based constraint solver from [5]. To emulate the run-time dynamism in resources availability, starting with an empty platform, i.e., 100 % resource availability, we successively (1) preoccupy one random resource and (2) perform the corresponding experiment, until all resources are preoccupied. This process is repeated 5 times to incorporate different mixes of preoccupied resources for each level of resource availability.

A. Run-time Embeddability

Run-time embeddability of the distillation methodologies is evaluated using two parameters: *success rate* and *embeddability rate*. For a given set of operating points, success rate indicates whether *at least one* points can be feasibly embedded on a certain set of available resources, whereas embeddability rate denotes the *proportion* of the embeddable points. Fig. 4 presents the corresponding results for the benchmark applications and

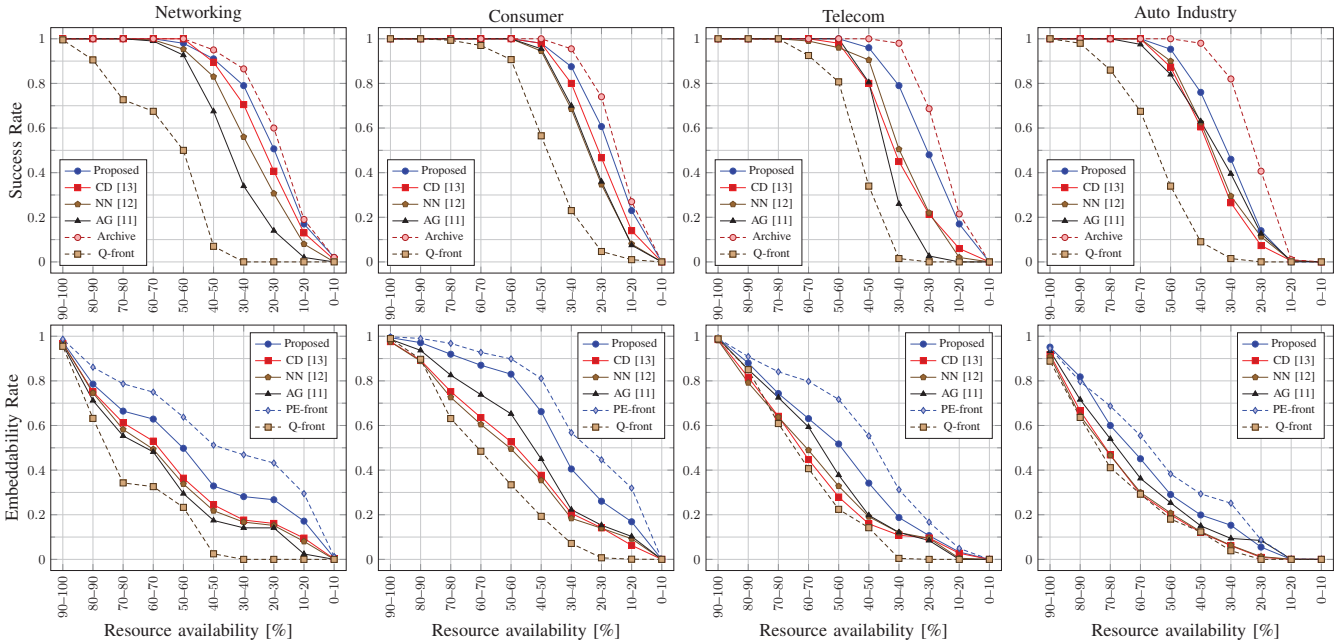

Fig. 4. Success- (top row) and embeddability rate (bottom row) of the distillation approaches vs. resource availability [%] for benchmark applications (columns) with a distillation capacity of $c = 12$. Archives are the best-case references in success rate. PE-fronts are (near) optimal references in embeddability rate.

a distillation capacity of $c = 12$. Archive is plotted as best case reference for success rate, and the first PE-front of the archive is used as a (near) optimal reference for embeddability rate. As illustrated, while the ranking of the three standard truncation approaches varies per application and criterion, proposed distillation methodology—especially when the platform becomes crowded—significantly outperforms them in both, success- and embeddability rate. One can observe that, compared to the standard approach with the *best* average results in the respective application and criterion, proposed methodology improves the success rate by 8 % up to 45 % and the embeddability rate by 16 % up to 45 %. While the enhanced success rate is important to bring applications to *execution*, the enhanced embeddability rate improves the *choices* for the RRM to select between different *embeddable* operating points. Note also that when merely the quality-dominant operating points are retained (Q-front) embeddability is dramatically degraded as those operating points often demand costly resource sets.

B. Distilled Quality Properties

To assess the efficiency of the distilled quality trade-off options, we use the well-established ϵ-*dominance* indicator [20] from the domain of MOO. Fig. 5 presents the ϵ-dominance of the distillation approaches wrt. the quality objectives for a distillation capacity of $c = 12$. A lower ϵ-dominance indicates a more efficient set of quality trade-offs. Q-fronts have an ϵ-dominance value of zero as they comprise quality-dominant operating points (excluded from Fig. 5). Resource-efficient operating points typically exhibit inferior quality measures as reflected in the high ϵ-dominance of PE-fronts. Compared to standard truncation approaches, proposed methodology exhibits an ϵ-dominance reduction of 0.04–0.06 on average over the benchmark applications, which translates into an improvement of 35 % up to 42 %. One can notice that, in case of the Consumer application, proposed approach exhibits a degraded ϵ-dominance. For this application, distribution of the operating points in the space of quality objectives is such that a small number of regions contain a huge number of points while other regions are empty. Hence, the majority of distilled operating points are resource-efficient points, and only a small portion of them belong to the quality-dominant cells which degrades the ϵ-dominance. Note that, for this application, proposed approach significantly improves the embeddability rate, cf. Fig. 4.

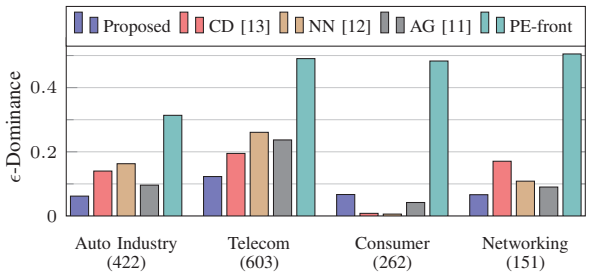

Fig. 5. ϵ-Dominance of the distillation approaches for benchmark applications with a distillation capacity of $c = 12$. The size of the initial archive is noted under the name of respective applications.

To further investigate the efficiency of the quality properties, we perform a run-time embedding experiment and log out the minimum- latency, energy consumption, and energy-latency product (ELP) delivered by the embeddable subset of the distilled points. ELP is used as a measure for the efficiency of the quality trade-offs. Corresponding results[2] for one sample application (Telecom) are illustrated in Fig. 6. Archive is plotted as best case reference. Although Q-front exhibits the best quality measures, as the resource availability decreases, it

[2]The reported results are an average over 10 runs of the DSE. Operating points of different DSE runs turn infeasible to embed at different levels of resource availability. As a result, with very low resource availability, the—otherwise increasing—minimum energy (and consequently, minimum ELP) measures decrease. In case of a single DSE run, the measures monotonically increase. The same argument applies where the minimum quality measure falls below that of the initial archive.

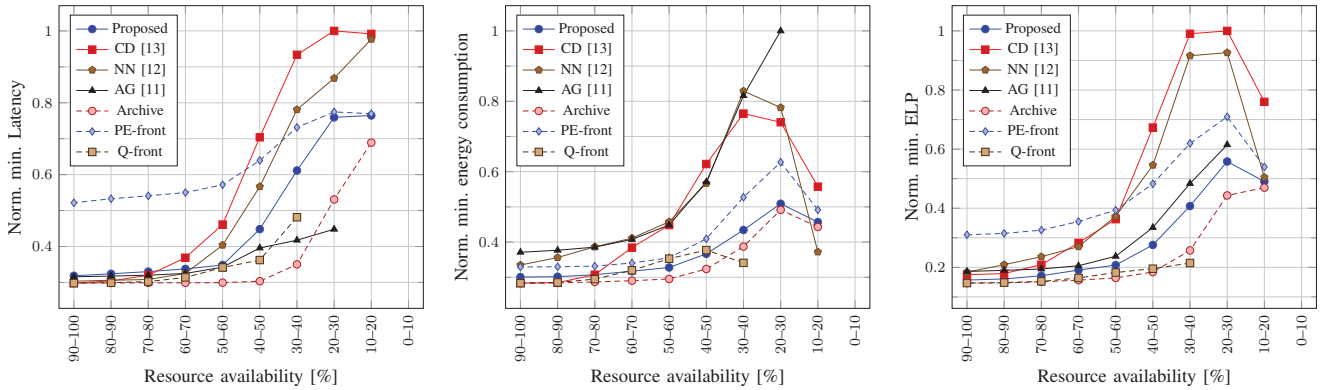

Fig. 6. Normalized minimum achievable latency, energy consumption, and energy-latency product (ELP) by the embeddable subset of distilled operating points vs. resource availability [%] for the Telecom application with a distillation capacity of $c = 12$. The archive represents the best-case reference.

rapidly turns infeasible to embed due to its costly resource requirements. PE-front exhibits low energy consumption but high latency, resulting in a degraded ELP. As illustrated, proposed methodology exhibits more efficient quality trade-offs, indicated by its low ELP measures. One can observe that, compared to the truncation approach with the *best* ELP, proposed methodology achieves an average ELP improvement of 7 % up to 18 % for a resource availability of 20–100 %. Furthermore, considering the quality properties individually, proposed approach achieves efficient measures in both, latency and energy consumption, while standard truncation approaches exhibit degraded measures in either or both criteria.

V. Conclusion

Targeting hybrid mapping strategies, this paper investigates design-time *distillation* of operating points obtained from the static design space exploration (DSE) to achieve an acceptable run-time overhead. Given a huge number of Pareto-optimal operating points in the space of quality objectives (performance, energy efficiency, etc.) and heterogeneous resources, the goal is to retain a specific number of points such that a high level of run-time embeddability is achieved and dynamic run-time quality requirements are addressed. We propose a two-level automatic distillation methodology that can seamlessly be integrated into the standard hybrid mapping flow. First, using a hyper-grid in the space of quality objectives, *quality regions* are formed to enable retaining diverse quality trade-off options. Subsequently, from within the quality regions, representative operating points are distilled wrt. their resource requirements to deliver efficient resource combinations. Experimental results show that, compared to existing truncation approaches, proposed methodology significantly enhances the run-time embeddability while achieving a comparable and often improved efficiency in the distilled quality properties.

Acknowledgment

This work was supported by the German Research Foundation (DFG) as part of the Transregional Collaborative Research Center "Invasive Computing" (CFB/TR 89).

References

[1] S. Borkar, "Thousand core chips: a technology perspective," in *Proc. Design Automation Conf. (DAC)*, 2007, pp. 746–749.

[2] A. K. Singh, M. Shafique, A. Kumar, and J. Henkel, "Mapping on multi/many-core systems: survey of current and emerging trends," in *Proc. Design Automation Conf. (DAC)*, 2013, pp. 1–10.

[3] A. Bonfietti, L. Benini, M. Lombardi, and M. Milano, "An efficient and complete approach for throughput-maximal SDF allocation and scheduling on multi-core platforms," in *Proc. Design, Automation and Test in Europe Conf. and Exhibition (DATE)*, 2010, pp. 897–902.

[4] A. Faruque, M. Abdullah, R. Krist, and J. Henkel, "ADAM: run-time agent-based distributed application mapping for on-chip communication," in *Proc. Design Automation Conf. (DAC)*, 2008, pp. 760–765.

[5] A. Weichslgartner, D. Gangadharan, S. Wildermann *et al.*, "DAARM: Design-time application analysis and run-time mapping for predictable execution in many-core systems," in *Proc. Int. Conf. Hardware/Software Codesign and System Synthesis (CODES+ ISSS)*, 2014, pp. 1–10.

[6] A. K. Singh, A. Kumar, and T. Srikanthan, "Accelerating throughput-aware runtime mapping for heterogeneous mpsocs," *ACM Trans. Design Automation of Electronic Systems (TODAES)*, vol. 18, no. 1, pp. 9:1–9:29, 2013.

[7] P. N. Khanh, A. K. Singh, A. Kumar, and K. M. M. Aung, "Incorporating energy and throughput awareness in design space exploration and run-time mapping for heterogeneous mpsocs," in *Proc. Euromicro Conf. Digital System Design (DSD)*, 2013, pp. 513–521.

[8] S. Wildermann, M. Glaß, and J. Teich, "Multi-objective distributed run-time resource management for many-cores," in *Proc. Design, Automation and Test in Europe Conf. and Exhibition (DATE)*, 2014, pp. 1–6.

[9] V. L. Vachhani, V. K. Dabhi, and H. B. Prajapati, "Survey of multi objective evolutionary algorithms," in *Proc. Int. Conf. Circuit, Power and Computing Technologies (ICCPCT)*, 2015, pp. 1–9.

[10] N. Hallam, P. Blanchfield, and G. Kendall, "Handling diversity in evolutionary multiobjective optimization," in *IEEE Congr. Evolutionary Computation*, vol. 3, 2005, pp. 2233–2240.

[11] J. Knowles and D. Corne, "Properties of an adaptive archiving algorithm for storing nondominated vectors," *IEEE Trans. Evol. Comput. (TEVC)*, vol. 7, no. 2, pp. 100–116, 2003.

[12] E. Zitzler, M. Laumanns, and L. Thiele, "SPEA2: Improving the strength pareto evolutionary algorithm," in *Proc. Conf. Evolutionary Methods for Design, Optimisation and Control with Application to Industrial Problems (EUROGEN)*, 2001, pp. 95–100.

[13] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Trans. Evol. Comput. (TEVC)*, vol. 6, no. 2, pp. 182–197, 2002.

[14] W. Abdou, C. Bloch, D. Charlet, and F. Spies, "Multi-pareto-ranking evolutionary algorithm," in *Proc. European Conf. Evolutionary Computation in Combinatorial Optimization*, 2012, pp. 194–205.

[15] X. Zhang, Y. Tian, R. Cheng, and Y. Jin, "An efficient approach to non-dominated sorting for evolutionary multi-objective optimization," *IEEE Trans. Evol. Comput. (TEVC)*, vol. 19, no. 2, pp. 201–213, 2015.

[16] M. Lukasiewycz, M. Glaß, F. Reimann, and J. Teich, "Opt4J: a modular framework for meta-heuristic optimization," in *Proc. Genetic and Evolutionary Computation Conf. (GECCO)*, 2011, pp. 1723–1730.

[17] R. Dick. (2010) Embedded system synthesis benchmarks suite (E3S). [Online]. Available: http://ziyang.eecs.umich.edu/ dickrp/e3sdd/

[18] J. Hu and R. Marculescu, "Energy-aware mapping for tile-based noc architectures under performance constraints," in *Proc. Asia and South Pacific Design Automation Conf. (ASP-DAC)*, 2003, pp. 233–239.

[19] P. T. Wolkotte, G. J. Smit, N. Kavaldjiev *et al.*, "Energy model of networks-on-chip and a bus," in *Proc. Int. Symp. System-on-Chip (SoC)*, 2005, pp. 82–85.

[20] M. Laumanns, L. Thiele, K. Deb, and E. Zitzler, "Combining convergence and diversity in evolutionary multiobjective optimization," *Evolutionary Computation*, vol. 10, no. 3, pp. 263–282, 2002.