Evaluating Design Space Subsetting for Multi-Objective Optimization in Configurable Systems

Mohamad Hammam Alsafrjalani*, Tosiron Adegbija†, and Lokesh Ramamoorthi*
*University of Miami, †University of Arizona
*{alsafrjalani,lokeshr}@miami.edu, †tosiron@email.arizona.edu

***Abstract*—Design space subsetting has been used to select configurations that are suitable for a target design objective. However, given the growing number of design constraints and objectives (energy, performance, EDP, temperature, user expectations, etc.) selecting the best subset for a single objective may no longer satisfy current design requirements. Additionally, the increasing design space sizes in emerging systems, and the variety of configurations that can satisfy multiple objectives, makes design space subsetting very challenging. In this paper, using a configurable cache as a case study, we evaluate the impact of design space subsetting for multi-objective optimization of performance, energy, and temperature. Using a design space of 243 configurations, yielding up to 1.4×10^{73} subsets, we evaluate the quality of the subsets obtained for one design constraint against the complete design space and against the remaining design objectives (e.g., best energy subsets for performance and thermal optimization). Our results reveal that prior subsetting methods are insufficient to meet current design trends due to the correlation between design objectives. Our results also suggest that large subsets of 10 or more configurations are required to maintain multi-objective optimization results that are within 3% of the optimal.**

***Index Terms*—Design space subsetting, configurable caches, temperature reduction, low energy**

I. Introduction

Resource-constrained systems (e.g., mobile phones, tablets, embedded systems) have become ubiquitous and have stringent constraints for high performance, low energy consumption, and low temperature dissipation. Since these constraints are usually conflicting, selecting a single configuration that optimizes multiple design objectives (e.g., energy, performance, or temperature) is very challenging. To adhere to multiple constraints, system architects revert to systems that offer multiple configurations, each of which adheres to a specific constraint during runtime. Heterogeneous systems such as ARM big.LITTLE [1], configurable systems such as those with dynamic voltage-frequency scaling (DVFS), and configurable and heterogeneous systems (HaCS) with configurable caches [2], offer a set of configurations that are selected or tuned dynamically based on applications' runtime requirements.

To determine the system's constituent configurations, designers evaluate the design space at design time. The subset of configurations that most closely satisfies the design constraints is then selected for use in the system. This process is known as *design space subsetting* [3]. Whereas evaluating the full design space allows the system to closely adhere to different design constraints, exhaustively searching the design space for the best subset can be prohibitively time-consuming. Furthermore, given the growing number of design objectives (energy, performance, energy-delay product (EDP), temperature, user expectations, etc.), determining the best subset requires an iterative exploration process for the different objectives, thus exacerbating the time overhead. As a result, current subsetting techniques only target a single optimization objective, limiting their effectiveness for any conflicting objective(s) [3], [4].

Given the variety of design constraints, optimization objectives, and large design spaces, the challenges are then to: (1) determine the minimum subset size that is required to adhere to multiple design objectives, (2) determine the constituent configurations of this subset, with minimal design time overhead, and (3) determine the likelihood that a subset selected for a specific design objective contains the best configuration for the other objectives. Addressing these challenges will enable highly configurable, and heterogeneous systems that can adhere to multiple runtime design objectives.

In this paper, we evaluate the impact of design space subsetting on performance, energy, and temperature. We analyze the impact of the subset size on each of the objectives and how subsets that satisfy the different objectives correlate with each other. To speed up our design space subsetting process, we propose a design-time approach that alleviates the exploration time. Our approach builds on prior algorithms and substantially reduces the design space exploration time (by > 99%) compared to exhaustive search. We analyze the results obtained using our algorithm, and compare these results to prior work [3], [4] and to exhaustive search.

Our work broadens the understanding of design space subsetting for multi-objective optimizations. Our results reveal that, in general, larger subset sizes are required for optimizing multiple design objectives. For our case study of a configurable cache featuring 243 configurations, subset sizes of 10 and 13 provided quality configurations for energy and performance, achieving optimization results within 3% of the complete design space. We observed that the subset sizes determined to be optimal for single objectives, as in prior work [3], [4], degraded multi-objective optimization results by up to 11.8% compared to the optimal. Finally, we also observed that selecting the best subsets for energy or performance also provided good-quality subsets for temperature. However, the best subset for temperature did not necessarily contain high quality configurations for the energy or performance.

II. BACKGROUND AND RELATED WORKS

There has been much prior work on optimizations and design tradeoffs in configurable systems [5]–[8], heterogeneous systems [1], [9], [10], HaCS [2], and design space exploration [3], [4], [11]. For brevity, we focus the background presented herein on work related to design space subsetting.

To mitigate the overheads of design space exploration while achieving near-optimal energy results, Viana et al. [3] argued that it is unnecessary to evaluate all the configurations in the design space. The authors studied the similarity of the energy savings among different configurations, and found that configurations that revealed similar energy savings could be merged and not evaluated on subsequent iterations. For a configurable cache, the authors revealed that a subset of four configurations were representative of the complete design space. Similarly, Palermo et al. [11] proposed a method that iteratively eliminated configurations from the design space. On each iteration, a set of Pareto optimal configurations was selected. The process was repeated until the desired number of iterations was reached, enabling feasible exploration time.

Whereas these prior works sped up the design space exploration, the works required a priori knowledge of the anticipated applications. Alsafrjalani et al. [4] extended the subsetting methods to applications that were unknown a priori. The proposed method evaluated the quality (energy-saving) of the configuration subset using different number of a priori-known applications or application-domains. The authors revealed that complete a priori access to the applications was unnecessary, and a limited knowledge of the anticipated applications was sufficient to obtain high quality subsets.

Despite the provided insights on the impact of design space subsetting on energy, prior works did not account for performance and/or temperature. Additionally, these works (e.g., [3]), focused on a small design space of 18 configurations. In our work, we evaluate the impact of design space subsetting on energy, performance, and temperature, considering a design space of 243 configurations.

III. SUBSET SELECTION AND EVALUATION

This sections motivates the selection of our design space configurations and provides details on the subset selection and evaluation.

A. Complete Design Space

To tractably describe and illustrate our approach, we focus on the L1 cache, due to its high impact on performance, energy, and temperature [4], [5]. We note that our approach can be adapted to other microarchitectural parameters such as instruction window, reorder buffer, and pipeline depth, independently or collectively with the cache.

We considered a design space that comprises of the combinations of cache sizes (8KB, 16KB, and 32KB), line sizes (16B, 32B, and 64B), and associativities (1-, 2-, and 4-way). We also assumed separate instruction (i) and data (d) caches with similar line sizes across both caches. Our design space contained the cross product of these cache configurations. Thus, our design space comprised of 243 configurations $3lineSizes \times (3Sizes \times 3associativities)^2$.

TABLE I: Design Space Layout for a Cache Memory

	Line Size		
Size_Asso	**16B**	**32B**	**64B**
8_1W	c_1	c_2	c_3
8_2W	c_4	c_5	c_6
8_4W	c_7	c_8	c_9
16_1W	c_{10}	c_{11}	c_{12}
.	.	.	.
.	.	.	.
.	.	.	.
32_4W	c_{25}	c_{26}	c_{27}

Table I depicts our design space layout for the instruction cache—for brevity, the data cache is not shown—in a two-dimensional layout. The rows represent cache sizes in KB (K) and associativities in number of ways (W). For example, a 16KB direct-mapped cache is denoted as 16K_1W). The columns represent the different line sizes in bytes (B) (e.g., a 32 byte line size is denoted as 32B).

B. Problem Formulation

For a given design space, S, there exists a best configuration, s_b that most closely satisfies the optimization objective (i.e., lowest energy or temperature, or highest performance) for a given application a. A subset of configurations, C, is a set of configurations such as $1 \leq |C| \leq |S|$. For each subset, C, there exists a best configuration, c_b, that most closely meets the design objective for a. Additionally, for a set of n applications, $A = \{a_1, a_2, \ldots, a_n\}$, we denote the energy, performance, and temperature of application a executing on a given configuration as $e(a_i, c_k)$, $p(a_i, c_k)$, $t(a_i, c_k)$, respectively, for $1 \leq i \leq n$, and $1 \leq k \leq |C|$.

For a given subset size, m, there exists

$$|N| = \frac{m!}{|S|! \times (m - |S|)!} \tag{1}$$

different subset combinations, and the best subset, C_b, is the subset with the configurations that best satisfy the optimization objective, on average, for all A.

The challenge, for each optimization objective, is to determine C_b, for each subset size $1 \leq m \leq 243$. In our case, the following equation applies:

$$W = \sum_{m=1}^{243} N \binom{243}{m} \approx 1.4 \times 10^{73} \tag{2}$$

That is, there are 1.4×10^{73} possible subsets to evaluate and 243 C_b's (one per subset size) to determine. Furthermore, given the disjoint optimization objectives, determining the best subset requires W^3 evaluations, if we must determine a single C_b for all three objectives.

One way to reduce the prohibitive exploration time is to evaluate the possibility of determining a subset which provides best configurations for all of the optimization objective, and thus only one exploration is required. To study this possibility, we obtained the best subsets separately for energy, performance, and temperature, for $1 \leq m \leq 243$. For each C_b of size m and constraint x, denoted C_{bmx}, where x is energy (e), performance (p), or temperature (t) constraint, we evaluated the subset's optimization of the remaining objectives.

C. Selecting and Evaluating Subsets

To speed up our design space exploration process, we developed a design-time algorithm that iteratively eliminates configurations from the design space, thereby exponentially reducing the number of configurations evaluated during subsequent iterations.

Algorithm 1 Design space subsetting algorithm for energy

```
1: Input:
     Complete cache configuration design space, $S$
     Application-configuration matrix, $EPT$
     Design constraint, $e$
2: Output:
     best subset $C_{bme}$ for $1 \leq m \leq |S|$
3: Begin
4: for all $1 < m < |S|$ do          //For all subset sizes
5:   C=S
6:   while $i < m$ do
7:     for all adjacent pairs of i$(cj,ck) do
8:       for all adjacent pairs of d$(cj,ck) do
9:         for all applications ai do
10:          find_$\mu\Delta$
11:        end for
12:      end for
13:    end for
14:    $Min\_pair$ = find_pair_min_$\mu\Delta$
15:    merge_pair($Min\_pair$)
16:    $C = C - c_i$                 //remove merged conf.
17:  end while
18:  Evaluate_quality($C_{bme}, p, t$)
19: end for
```

For illustration, we explain our algorithm for determining the best subset for energy optimization, using our design space (Table I). Algorithm 1 depicts our design space subsetting algorithm for energy optimization. The algorithm takes in as inputs the complete design space S, the three-dimensional configuration energy-performance-temperature EPT matrix, and the target optimization objective, x, where $x = e$ for energy (this can be changed for performance p or temperature t). The outputs of the algorithm are the best subsets, C_{bme}, for each subset size $1 < m < S$, and the quality of C_{bmx} for p and t.

The algorithm contains two nested iterations. The first one explores the design space for each subset size $1 < m < S$ (line 4 to 19) and the second determines the best subset C_{bme} (lines 6 to 17). For each subset size m, the algorithm begins with the complete design space S (line 5) as the starting subset. In each iteration, for adjacent pairs of data cache (d\$) and instruction cache (i\$) configurations, and for all applications, the algorithm evaluates the average energy consumption change ($\mu_\delta(c_j, c_k)$) incurred by executing the applications with configuration c_j instead of configuration c_k:

$$\mu_\delta(c_j, c_k) = \frac{1}{n}\sum_{i=1}^{n} e_\delta(c_j, c_k, a_i) \tag{3}$$

where n is the number of applications, c_j and c_k are row- or column-adjacent configuration pairs in Table I, and:

$$e_\delta(c_j, c_k, a_i) = \frac{e(c_j, a_i) - e(c_k, a_i)}{e(c_k, a_i)} \tag{4}$$

The algorithm searches for and merges the pair of configurations that resulted in the least energy change (line 14 to 16) (i.e., c_k becomes c_j). When $i = (m-1)$, the resulting configurations of the subset are the constituent configurations of C_{bme}, the best subset of size m for the energy constraint.

Finally, in line 18, using this subset, the algorithm evaluates the adherence of this subset to the other design constraints (similar to Equation 4), using the best configurations from the subset (c_b), compared to using the best configuration in the complete design space (s_b), expressed as follows:

$$p_{av\delta}(c_b, s_b, a_i) = \frac{p(c_b, a_i) - p(s_b, a_i)}{p(s_b, a_i)} \tag{5}$$

and

$$t_{av\delta}(c_b, s_b, a_i) = \frac{t(c_b, a_i) - t(s_b, a_i)}{t(s_b, a_i)} \tag{6}$$

.

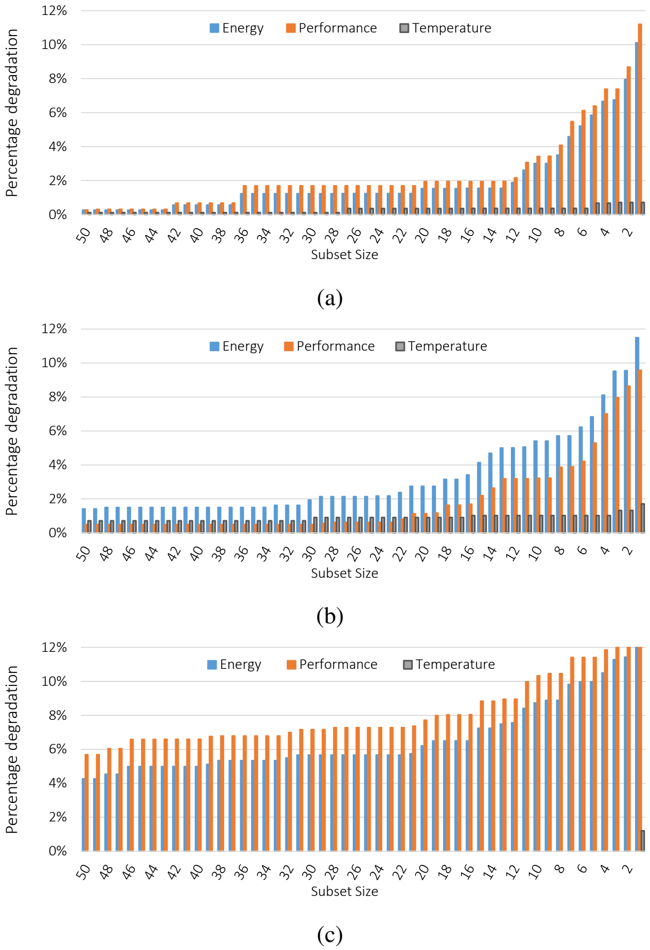
Similarly, we note that when $x = p$ or $x = t$, the algorithm will select the best subsets using Eq. (3) and (4) for performance and temperature, respectively. It will also evaluate the quality of the subsets for the energy constraint using the following equation:

$$t_{av\delta}(c_b, s_b, a_i) = \frac{e(c_b, a_i) - e(s_b, a_i)}{e(s_b, a_i)} \tag{7}$$

IV. EXPERIMENT SETUP

To evaluate the impact of the subset size on a configurable cache's ability to adhere to design objectives, we used our algorithm (Algorithm 1) for the design space, S, (Section III-A). We performed the experiments for $x = e, p$, and t, separately. For each constraint x we determined the quality of C_{bmx}, for $m \in [1, |S|]$. Furthermore, for each x, we evaluated the quality of C_{bmx} for $\{e, p, t\} - x$, using Eq. 5-7.

To obtain application-configuration energy-performance-temperature EPT matrix values, we used a collection of applications that represent different application domains and with our design space $|S|$. In total, we used seventeen applications from the MiBench [12] and the EEMBC [13] Automotive suites. The benchmarks were compute kernels performing specific tasks in different application domains, including networking, image processing, security, etc. We obtained the performance results by executing the applications on the gem5 simulator [14], and modeled the energy and temperature using McPAT [15] and HotSpot [16], respectively. To model fan-less systems, we designed our system with a heat sink of 1mm and spreader 0.1 mm thickness, and set the convection resistance to 4K/W.



(a)

(b)

(c)

Fig. 1: Impact of subset size on (a) energy, (b) performance, and (c) temperature, using Algorithm 1

V. Results and Analysis

In this section, we discuss the results of the impact of the subset size on the design objectives using Algorithm 1 and the qualities of these subset compared to those obtained using the exhaustive method.

A. Impact of subset size on the design objectives

Figure 1(a), (b), and (c) depict the results of our three experiments to select the best subsets for the energy, performance, and temperature, respectively, using Algorithm 1. The results depict the average percent increase in energy, performance, and temperature values for all of the applications using the best subsets, as compared to the complete design space (higher percentage values represent lower subset quality). For each experiment, the figures also depict the quality of the best subsets for the other objectives (e.g., performance and temperature in Figure 1(a)), using (5), (6), and (7) (Section III-C).

In general, as expected, larger subsets improve the energy, performance, and temperature as the subset size tends towards the full design space. However, we observed that subset sizes larger than 50 did not result in any substantial changes for all objectives. Increasing the subset size beyond 50 did not improve any objective by more than 0.49%. Thus, in the depicted results, we only show subset sizes less than 50.

1) Energy: The results revealed that, on average, a smaller subset size decreased the subset quality for energy. However, only subsets smaller than 11 degraded the quality by 2.6% or more. Furthermore, the quality of the subset degrades substantially for subset sizes smaller than 5 (e.g., 10% for size of 1). This increase was due to the variability in application characteristics; subsets smaller than 5 were insufficient to satisfy the different applications' cache requirements.

We also observed that the best energy subsets provided high quality subsets for performance and temperature. The best energy subsets (C_{b10e}) provided quality subset for performance and temperature within 3.1% and 0.4%, respectively, of the complete design space. The smallest best energy subset (C_{b1e}) provided performance and temperature within 11.2% and 0.7%, respectively, of the complete design space. Although performance and temperature are degraded by smaller subsets, the performance degradation was much larger than the temperature degradation. Regardless of the subset size, we observed that when Algorithm 1 selected the subsets for the energy objective, the quality of the subsets for energy were always higher than the quality for performance. Since Algorithm 1 used (3) and (4) to select the subsets, the algorithm prioritized the energy objective over performance and the resulting subsets provided better energy savings than performance. Furthermore, the configurations required to adhere to energy savings were not sufficient for the performance.

On the other hand, the best subset for energy only increased the temperature by 0.4% and 0.7%, for C_{b10e} and C_{b1e}, respectively. Since temperature is a byproduct of energy and high energy input (e.g., large currents) dissipates energy in form of heat, low energy configurations are likely to reduce temperature dissipation. As a result, subsets that prioritized energy also maintained low average temperatures.

2) Performance: Similar to energy, the smaller sizes reduced the quality of the subsets. However, unlike energy, larger subset sizes were required to maintain less than 2% degradation in performance. For instance, to maintain the performance within 2.6% of the optimal, a subset of size 13 (C_{b13p}) was required, whereas a subset of size 10 (C_{b10e}) was required to maintain energy within 2.6% of the the complete design space. These results suggest that maintaining the applications' performance requires more configurations than meeting applications' energy constraints. Due to the higher variation in the hardware requirements for performance, larger subsets, with more configurations, are required to provide high quality c_b's compared to the complete design space. Since our design space considered independent instruction and data caches, more configurations were required to satisfy design objectives for both caches.

Furthermore, C_{b1p} degraded the performance by 9.5% compared to the full design space, since one configuration is insufficient to sustain high performance for all of the applications. However, we note that C_{b1p} provided better performance quality, as compared to C_{b1e} (Section V-A1), since the algorithm obtained C_{b1p} by merging the average performances of the different configurations. Additionally, C_{b1p} incurred 1.7% temperature increase compared to the complete design space, which was more than the temperature increase for C_{b1e}. Since high performance configurations typically consume more energy, and thus dissipate more temperature, the best performance subsets also contained higher temperature configurations.

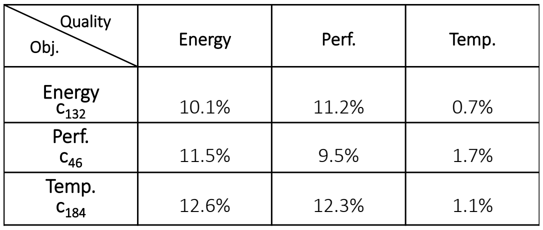
3) Temperature: Unlike for energy and performance, subsetting the design space for temperature did not increase the average temperature of the system. On average, the applications required at least two-configuration subsets to execute with temperatures equivalent to the complete design space. A subset of size one C_{b1t} incurred 1.1% increase in average temperature. We observed that smaller cache sizes always dissipated the least temperature, and there was not much difference between the other cache configurations with respect to temperature. That is, as long as the cache size was kept small (e.g., 8KB), regardless of line size and associativity, average temperature dissipation was low. However, using C_{b1t} significantly degraded the energy and performance. For instance, while C_{b13p} degraded energy by 4.6%, C_{b13t} degraded energy by 7.5%. In order to maintain energy savings within 4.6% of the complete design space, a temperature subset of size 39 was required.

Furthermore, C_{b1t} incurred 12.6% and 12.3% energy and performance overhead, respectively, and increase of 24.7% and 29.4%, compared to C_{b1e} and C_{b1p}, respectively. Since temperature optimization tends to favor configurations with smaller cache sizes, those configurations were not suitable for all of the performance and energy requirements of all of the applications A. In addition, since the applications required caches of disparate sizes, the applications' performance and energy degraded significantly when the design space was reduced to a single cache size.

4) Comparison to prior work: Prior work ([3], [4]) required a subset of 4 configurations (C_{b4e}) to provide energy savings within 3% of the complete design space for a design space of 18 configurations. However, given our design space of 243 configurations, prior work degraded the energy and performance by 6.6% and 8.1%, respectively. Also, prior work required a separate subset for each design objective, whereas our algorithm determined a single subset that simultaneously improved multiple objectives. For instance C_{b10e} (Figure 1 A) shows a subset of ten configurations that achieved energy, performance, and temperature within 3.0%, 3.4%, and 0.4% of the complete design space.

Furthermore, prior work required 22.2% of the complete design space to maintain energy savings within 3% of the optimal. However, our work was able to maintain energy, performance, or temperature within 3% of the optimal using only 4.1% of the complete design space.

5) Broader impact of the best subsets: To determine the best single configuration for multi-objective optimization, if only one configuration could be selected, we analyzed the tradeoffs of a single-configuration best subset C_{b1x}.



Quality / Obj.	Energy	Perf.	Temp.
Energy c_{132}	10.1%	11.2%	0.7%
Perf. c_{46}	11.5%	9.5%	1.7%
Temp. c_{184}	12.6%	12.3%	1.1%

Fig. 2: Summary of the best subset C_{b1x}'s configuration for each design objective and quality of that subset for energy, performance, and temperature

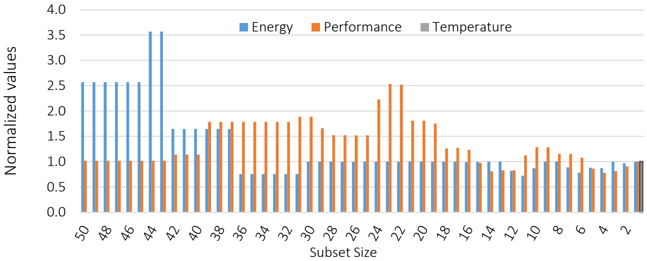


Fig. 3: The results of Algorithm 1 normalized to the result of exhaustive exploration, for energy, Performance, and thermal

Figure 2 depicts a summary of the best subset C_{b1x}'s configuration for each design objective and quality of that subset for energy, performance, and temperature. The rows represent the target objectives and the columns represent the remaining objectives. For example, Perf.C_{46} contains configuration 46 and is the best subset of size 1 (i.e., best single configuration) for performance, resulting in performance within 9.5% of the complete design space.

Figure 2 provides insights on the design opportunities for multi-objective optimizations. For energy-constrained system, a single core with c_{132} will provide energy, performance, and temperature within 10.1%, 11.2%, and 0.7% of the complete design space. However, if the system is also required to perform real-time tasks, the designer can elect to include another core with c_{46}. Alternatively, for systems with relaxed hardware area constraints, the designer could elect to include several cores with all of the c_{132}, c_{46}, and c_{184} configurations, configurable cores with the aforementioned configurations, or replicated cores with similar configurations.

B. Algorithm 1 vs. exhaustive

Compared to exhaustive search, Algorithm 1 reduced the exploration time by $> 99\%$ for each objective. To select the subset Algorithm 1 compares adjacent configurations only, instead of all possible configuration combinations, which results in exponential reduction of time. For instance, to select C_{b121x} (largest number of possible combinations given Equation (2)), Algorithm 1 performs $\approx 1.2 \times 10^7$ comparisons vs. 7.2×10^{71} in exhaustive search.

We also analyzed the quality of subsets achieved by Algorithm 1, compared to exhaustive search. Figure 3 depicts the results of the absolute values of the quality of subsets obtained with Algorithm 1 normalized to the absolute values of those obtained using exhaustive exploration, for energy, performance, and temperature. Values above/below 1.0 represent worse/better qualities. As expected, Algorithm 1 produced lower-quality subsets compared to the exhaustive search. However, the difference in qualities converged for smaller subset sizes.

1) Energy: For the energy objective Algorithm 1 provided lower quality subsets for large subset sizes (e.g., 38). However, large subset sizes did not necessarily result in worse qualities. For instance, Algorithm 1 determined higher quality for C_{b50e} than C_{b36e}. Since Algorithm 1 compared only adjacent configurations, the algorithm merged equally high-quality configurations, whereas the exhaustive search evaluated all possible configuration combinations before eliminating configurations from the design space. Alternatively, for the smallest subset size, both Algorithm 1 and the exhaustive search provided the same subset C_{b1e}. This result revealed that the design space layout (Table I) affects the outcome of Algorithm 1 and that different arrangement of configurations will impact the quality of the selected subset.

2) Performance: Similar to energy, Algorithm 1 met or under performed the exhaustive search method. Also, the quality of the subsets converged with the quality of the subsets obtained with the exhaustive search. However, by examining Figure 3 we observe that lowest divergence occurred for subset sizes 28—19. This result revealed that merging neighboring configurations in the design space degrades the quality of the subsets for intermediate subset sizes, with respect to the complete design space. However, to select a subset of single configuration, the algorithm was on par with exhaustive search.

3) Temperature: Since a very small subset was required to maintain temperature quality, Algorithm 1 did not degrade the subset quality for temperature, compared to exhaustive search. To determine C_{b1p}, both methods provided the same configuration.

In general, for all design objectives, we observed that as the subset size decreased the variance between our algorithm and the exhaustive method converged. Furthermore, both methods resulted in c_{b1e}, c_{b1p}, c_{b1t}, with configurations c_{32}, c_{46}, and c_{184} as the constituent configuration for energy, performance and temperature, respectively.

VI. CONCLUSIONS

In this paper we evaluated the applicability of design space subsetting to the multi-objective optimization of energy, performance, and temperature. As a case study, we used a configurable cache featuring a design space of 243 configurations with 1.4×10^{73} possible subsets. We used our proposed subsetting algorithm to make the subsetting design space more tractable. Our results revealed that the design space subsetting does not need to be repeated for the different design objectives. For our design space, a 10-configuration subset for energy provided high quality configurations, within 3% of exhaustive search, for all of the design objectives. Similarly, a 13-configuration subset for performance provided high quality configurations, within 3% of exhaustive search, for all of the design objectives. However since temperature-specific subsets contained small cache sizes only, temperature-specific subsets resulted in poor quality configurations for energy and performance optimization.

Our future work involves extending our algorithm to other parameters such as clock frequency, reorder buffer, and issue window. We also plan to explore include hardware area as an additional design objective.

REFERENCES

[1] B. Jeff, “Big. little system architecture from arm: saving power through heterogeneous multiprocessing and task context migration,” in *Proceedings of the 49th Annual Design Automation Conference*. ACM, 2012, pp. 1143–1146.

[2] M. H. Alsafrjalani and A. Gordon-Ross, “Scheduling and tuning for low energy in heterogeneous and configurable multicore systems,” *Computers*, vol. 7, no. 2, p. 25, 2018.

[3] P. Viana, A. Gordon-Ross, E. Keogh, E. Barros, and F. Vahid, “Configurable cache subsetting for fast cache tuning,” in *Proceedings of the 43rd annual Design Automation Conference*. ACM, 2006, pp. 695–700.

[4] M. H. Alsafrjalani and A. Gordon-Ross, “Low effort design space exploration methodology for configurable caches,” *Computers*, vol. 7, no. 2, p. 21, 2018.

[5] C. Zhang, F. Vahid, and W. Najjar, “A highly configurable cache architecture for embedded systems,” in *Computer Architecture, 2003. Proceedings. 30th Annual International Symposium on*. IEEE, 2003, pp. 136–146.

[6] Y. Kora, K. Yamaguchi, and H. Ando, “Mlp-aware dynamic instruction window resizing for adaptively exploiting both ilp and mlp,” in *Microarchitecture (MICRO), 2013 46th Annual IEEE/ACM International Symposium on*. IEEE, 2013, pp. 37–48.

[7] G. Semeraro, G. Magklis, R. Balasubramonian, D. H. Albonesi, S. Dwarkadas, and M. L. Scott, “Energy-efficient processor design using multiple clock domains with dynamic voltage and frequency scaling,” in *High-Performance Computer Architecture, 2002. Proceedings. Eighth International Symposium on*. IEEE, 2002, pp. 29–40.

[8] W. Wang, P. Mishra, and A. Gordon-Ross, “Dynamic cache reconfiguration for soft real-time systems,” *ACM Transactions on Embedded Computing Systems (TECS)*, vol. 11, no. 2, p. 28, 2012.

[9] B. de Abreu Silva, L. A. Cuminato, and V. Bonato, “Reducing the overall cache miss rate using different cache sizes for heterogeneous multi-core processors,” in *Reconfigurable Computing and FPGAs (ReConFig), 2012 International Conference on*. IEEE, 2012, pp. 1–6.

[10] R. Kumar, D. M. Tullsen, N. P. Jouppi, and P. Ranganathan, “Heterogeneous chip multiprocessors,” *Computer*, vol. 38, no. 11, pp. 32–38, 2005.

[11] G. Palermo, C. Silvano, and V. Zaccaria, “Respir: a response surface-based pareto iterative refinement for application-specific design space exploration,” *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 28, no. 12, pp. 1816–1829, 2009.

[12] M. R. Guthaus, J. S. Ringenberg, D. Ernst, T. M. Austin, T. Mudge, and R. B. Brown, “Mibench: A free, commercially representative embedded benchmark suite,” in *Workload Characterization, 2001. WWC-4. 2001 IEEE International Workshop on*. IEEE, 2001, pp. 3–14.

[13] J. A. Poovey, T. M. Conte, M. Levy, and S. Gal-On, “A benchmark characterization of the eembc benchmark suite,” *IEEE micro*, vol. 29, no. 5, 2009.

[14] N. Binkert, B. Beckmann, G. Black, S. K. Reinhardt, A. Saidi, A. Basu, J. Hestness, D. R. Hower, T. Krishna, S. Sardashti *et al.*, “The gem5 simulator,” *ACM SIGARCH Computer Architecture News*, vol. 39, no. 2, pp. 1–7, 2011.

[15] S. Li, J. H. Ahn, R. D. Strong, J. B. Brockman, D. M. Tullsen, and N. P. Jouppi, “Mcpat: an integrated power, area, and timing modeling framework for multicore and manycore architectures,” in *Microarchitecture, 2009. MICRO-42. 42nd Annual IEEE/ACM International Symposium on*. IEEE, 2009, pp. 469–480.

[16] K. Skadron, M. R. Stan, K. Sankaranarayanan, W. Huang, S. Velusamy, and D. Tarjan, “Temperature-aware microarchitecture: Modeling and implementation,” *ACM Transactions on Architecture and Code Optimization (TACO)*, vol. 1, no. 1, pp. 94–125, 2004.