Co-Design of Systems-On-Chip for Sustainability

Jan Spieck
Friedrich-Alexander-Universität Erlangen-Nürnberg (FAU), Germany

Dominik Walter
Friedrich-Alexander-Universität Erlangen-Nürnberg (FAU), Germany

Jan Waschkeit
Friedrich-Alexander-Universität Erlangen-Nürnberg (FAU), Germany

Jürgen Teich
Friedrich-Alexander-Universität Erlangen-Nürnberg (FAU), Germany

Abstract

This paper introduces a novel approach to the co-design of sustainable embedded systems through multi-objective design space exploration (DSE). We propose a two-phase methodology that optimizes both the multiprocessor system-on-chip (MPSoC) architecture and application mappings, considering sustainability, reliability, performance, and cost as optimization objectives of equal importance. Unlike existing approaches, our method thereby accounts for both operational and embodied emissions, providing a more comprehensive assessment of sustainability. The first phase employs intra-application DSEs to explore Pareto-optimal constraint graphs for each application. The second phase, an inter-application DSE, combines these results to explore sustainable target architectures and corresponding application mappings. Our approach incorporates detailed models for embodied emissions (scope 1 and scope 2), operational emissions, reliability, performance, and cost. The evaluation demonstrates that our sustainability-aware DSE is able to explore design spaces overlooked by traditional approaches, supported by superior results in four key objectives. This enables the development of more environmentally friendly embedded systems while still achieving high performance and reliability.

2012 ACM Subject Classification Computer systems organization → Embedded and cyber-physical systems; Hardware → Power and energy; Hardware → Chip-level power issues; Computer systems organization → System on a chip; Social and professional topics → Sustainability; Hardware → Impact on the environment

Keywords and phrases System-on-Chip, Sustainability, Multi-objective Optimization, Design Space Exploration, Embedded Systems, Carbon Emissions

Digital Object Identifier 10.4230/OASIcs.NG-RES.2025.3

Funding *Jan Spieck*: Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) – Project number 146371743 – TRR 89: Invasive Computing
Dominik Walter: Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) – Project number 146371743 – TRR 89: Invasive Computing

1 Introduction

The impact of information and communication technology (ICT) on global greenhouse gases (GHGs) emissions is currently estimated to be between 2.1 % and 3.9 %, and is accelerating rapidly [11]. But since global emissions must be reduced significantly over the next years, a fast shift to more sustainable computing systems is inevitable. However, sustainability does not come for free and to design a sustainable system that balances performance, reliability, and at the same time cost poses a complex challenge. This is to our knowledge the first work to provide a solution to this problem by proposing a highly configurable multi-objective sustainability-aware design space exploration (DSE) that explores sustainable embedded system designs automatically. More specifically:

1. We propose a DSE for exploring sustainable MPSoC target architectures, optimizing for sustainability, reliability, performance and cost. It can be fine-tuned to the specific production and deployment environment of the embedded system to accurately model the CO_2 emissions during both production and operation.
2. Besides exploring sustainable target architectures, the proposed approach explores also composable application mapping for real-time applications, that are tailored to these architectures. Besides applications with static execution properties, the approach even supports applications with dynamic workload variations during operation.
3. A comprehensive evaluation of the proposed sustainability-aware DSE is provided to demonstrate the need to consider sustainability during the design.

2 Related Work

Design space exploration (DSE) has become a de-facto standard for simultaneous optimization of various aspects of multiprocessor systems-on-chip (MPSoC) architectures. However, existing approaches overlook the significant impact of the embodied emissions of an embedded system, although [9] predicts that these embodied emissions will dominate the total emissions for future embedded systems. [27] propose a scenario-aware DSE that explores a MPSoC architecture and corresponding scenario-aware mappings using a co-evolutionary algorithm concerning the objectives of execution time and chip area. However, their DSE has scalability issues for large number of applications and big architectures. [1] present a DSE that optimizes the mapping of applications regarding execution time, lifetime, power, and temperature for a given target architecture. Furthermore, it turns the multi-objective optimization problem into a single-objective one, thus losing large parts of the search space. [19] provide an overview of hybrid application mapping techniques for heterogeneous MPSoC target architectures, including approaches for DSE. Instead of a DSE, also heuristic mapping approaches can be used at run time to speedup the exploration [22]. [21] propose a symmetry-eliminating DSE, enabling efficient exploration of so-called constraint graphs that specify an application mapping on a higher abstraction level. By doing so, the exploration can skip the vast inherent redundancy of mappings. [9] and [12] present a detailed analysis of the environmental impact of computing systems, considering both operational and embodied emissions. Their work highlights the importance of a holistic approach to sustainability in computing. [25] present a tool to analyze the carbon footprint of chiplet-based architectures, showing how it can reduce greenhouse gas emissions in manufacturing. [5] propose a co-optimization approach for datacenter sustainability, considering both design technology and system-level factors, thereby balancing performance and sustainability objectives.

3 Fundamentals

In this section, we introduce our application and architecture model, describe the corresponding formal mapping and give a short overview on modeling and evaluating of sustainability and reliability, using the illustrative example in Fig. 1.

An application $a \in A$ is a directed graph of tasks $\tau \in T_a$, communicating over channels. These graphs express data dependencies between the tasks by direct edges; a task can start when the preceding tasks have finished processing. At run time, applications process data $d \in D_a$ periodically, where the processing of each data should finish before an application-specific deadline θ_a. The task graphs of three example applications $a_1, a_2, a_3 \in A$ are depicted at the top of Fig. 1.

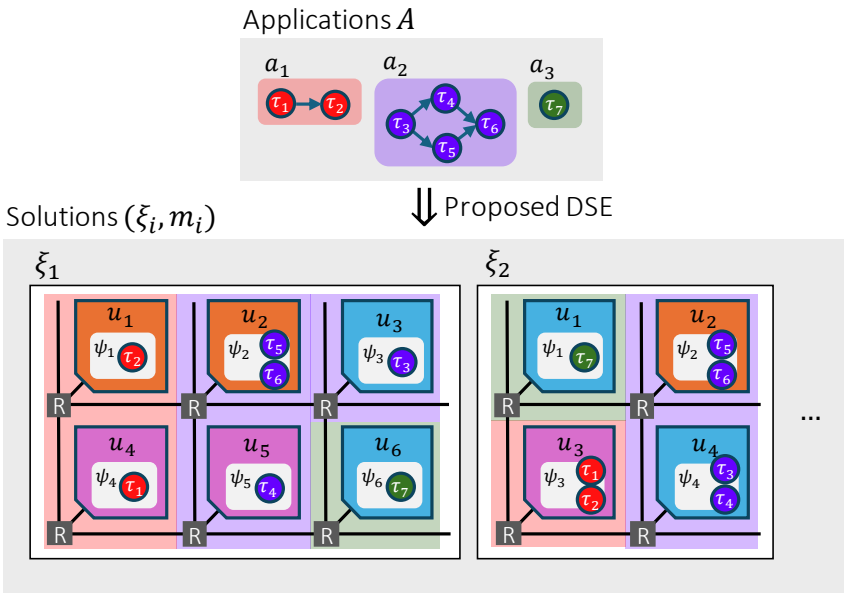

Figure 1 Illustration of application graphs for three example applications $a_1, a_2, a_3 \in A$ and two target MPSoC solutions ξ_1, ξ_2 with two annotated mappings, explored by the proposed DSE. Architecture ξ_1 is equipped with six tiles of three different types, while ξ_2 has four tiles. These tiles are connected by a 2D NoC mesh using routers. Each tile is equipped with a single-core processor of different types, onto which the tasks of the applications are mapped.

This work explores heterogeneous tile-based MPSoC architectures ξ, as presented by [3, 7, 14, 6]. Each tile $u \in U$ is equipped with a set of processing resources $\psi \in u$, e.g., CPUs or accelerators [28], connected to a 2D Network-on-Chip (NoC) mesh, and is of a heterogeneous tile type $v \in V$, differing in the number and type of processors on the tile. Two example target architectures ξ_1 and ξ_2 are depicted in the bottom row of Fig. 1. The MPSoC ξ_1 is equipped with six tiles $u_1, \ldots, u_6$ of three different color-coded tile types v_1, v_2, v_3, defined by the type of the mounted single-core processor ψ_i. ξ_2 contains four tiles $u_1, \ldots, u_4$.

A mapping m maps each application task $\tau \in T$ to exactly one resource ψ of the target architecture ξ. Different applications are always mapped to separate sets of tiles of the target architecture, i.e., there are no tiles with tasks from two different applications. This allows composability [2] which is achieved by applying a temporal isolation scheme [13] to the NoC. As the search space for mappings is polluted by many redundant mappings with a different binding but the same execution characteristics, [21] proposes a symmetry-eliminating DSE. Instead of exploring individual mappings m directly, this approach explores *constraint graphs* $\eta \in H$, which group tasks together, assign these groups to tile types, and specify constraints between tasks, e.g., a maximum hop distance on a NoC. This abstraction captures essential mapping characteristics without tying to a specific chip layout, thus eliminating redundant solutions. As a result, the proposed DSE efficiently considers only meaningfully design alternatives, later transforming these graphs into actual task-to-resource mappings m by a binding algorithm [23]. Fig. 1 illustrates the transformed mappings for the three applications to the two target architectures. On ξ_1, the application a_1 with tasks $\tau_1, \tau_2 \in T_{a_1}$ is mapped to the tiles $u_1, u_4 \in \xi_1$, the application a_2 with tasks $\tau_3, \ldots, \tau_6 \in T_{a_2}$ is mapped to the tiles $u_2, u_3, u_5 \in \xi_1$, and the application a_3 with tasks $\tau_7 \in T_{a_3}$ is mapped to tile $u_6 \in \xi_1$. This mapping also includes access grants to the routers of the NoC, which are necessary for inter-tile communication. Another mapping is illustrated for ξ_2.

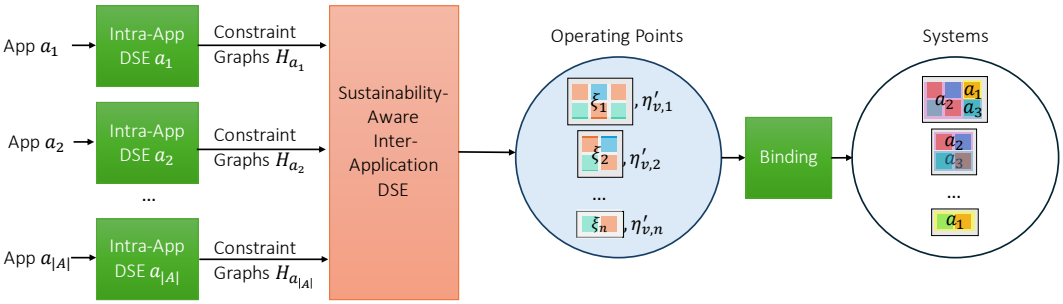

Figure 2 Optimization flow of the proposed sustainability-aware DSE of embedded SoC systems that implement a given set A of applications.

In this work, we analyze and systematically explore the design space of tile-based multiprocessor system-on-chip (MPSoC) implementations of embedded systems with respect to sustainability by considering both their so-called *embodied* and *operational* greenhouse gas (GHG) emissions. *Operational* emissions are the emissions that occur during the operation of the system, while the *embodied* emissions refer to the emissions that occur during production of the chip. These can be differentiated into three *scopes* [29]. *Embodied Scope 1* emissions include chemicals and gases produced directly during chip manufacturing. The emissions are represented in CO_2 equivalents. *Embodied Scope 2* emissions represent the portion of emissions attributed to the energy required during the manufacturing process. Unlike scope 1 emissions which occur directly within the production environment, scope 2 emissions are dependent on the energy mix used to power the production facility. This energy mix varies by location based on the region's reliance on renewable or fossil fuel energy sources. *Embodied Scope 3* emissions include upstream and downstream supply-chain emissions and are omitted in this work due to their high variability and difficulty in estimation.

While operational emissions can be reduced by designing more energy-efficient computer architectures, embodied emissions can only be reduced – from a hardware design perspective – by either building smaller less powerful chips or extending the lifetime of the chip to amortize the fixed emissions during manufacturing. Therefore, the lifetime, or more generally the reliability, of a system plays a critical role in achieving a sustainable design.

4 DSE for sustainable Co-Design

Exploring a sustainable MPSoC architecture with corresponding application mappings poses a complex multi-objective optimization problem. In this work, we propose to employ the two-phase exploration methodology displayed in Fig. 2 to solve this optimization problem. For a given set A of applications to be implemented on the system to be designed, we first perform a set of separate intra-application DSEs for each application $a \in A$ to explore Pareto-optimal *constraint graphs* $\eta_a \in H_a$ for a variety of tile number combinations using [21]. Afterwards, an inter-application DSE explores the space of architectures ξ and constraint graphs $\eta_v = \{\eta_{a_1}, ..., \eta_{a_{|A|}}\}$, maximizing sustainability, performance, and reliability while minimizing cost. Finally, the constraint graphs are bound to the target architecture using the binding algorithm from [23], resulting in a set of target designs with varying application mappings.

4.1 Intra-Application DSE

For each application $a \in A$, a separate intra-application DSE explores so-called *constraint graphs* $\eta_a \in H_a$, which minimize the deadline miss percentage $dm(\eta_a, \xi)$ and energy consumption $eng(\eta_a, \xi)$ when executing on an architecture ξ, constituted by the required number of tiles $r_{v_i}(\eta_a)$ of each tile type $v_i \in V$ for the regarded constraint graph η_a:

$$\underset{\eta_a \in H_a}{\text{minimize}}(dm(\eta_a, \xi), eng(\eta_a, \xi), r_{v_1}(\eta_a), ..., r_{v_{|V|}}(\eta_a)) \tag{1}$$

For applications with constant execution times, $dm(\eta_a, \xi)$ is binary, being 0 if the execution time $l(\eta_a, \xi)$ of the application a under a representative mapping for constraint graph η_a on ξ is smaller than the deadline θ_a and 1 otherwise. For applications with input-dependent execution, the deadline miss percentage can be determined by a representative dataset D:

$$dm(\eta_a, \xi) = \frac{1}{|D|} \cdot \sum_{d \in D} \begin{cases} 0 & \text{if } l(\eta_a, \xi, d) \leq \theta_a \\ 1 & \text{else} \end{cases} \tag{2}$$

Let eng represent the total energy consumption of the chip during its lifetime, which we calculate by the average energy consumption eng_{θ_a} per period (equalling the deadline θ), scaled up to its lifetime ϑ:

$$eng(\eta_a, \xi) = \phi \cdot \frac{\vartheta}{\theta_a} \cdot eng_{\theta_a}(\eta_a, \xi) \tag{3}$$

In this formula, ϕ denotes the utilization of the MPSoC, expressing the percentage the SoC is active during its lifetime. $eng_{\theta_a}(\eta_a, \xi)$ is then given as the sum of the energy consumption per task $\tau \in T_a$, which is the product of the power consumption $P(\psi)$ of a compute resource $\psi = \eta_a(\tau)$ and the average latency of the task τ:

$$eng_{\theta_a}(\eta_a, \xi) = \sum_{\tau \in T_a} \left(P(\eta_a(\tau)) \cdot \frac{1}{|D|} \sum_{d \in D_a} l_\tau(\eta_a, \xi, d) \right) \tag{4}$$

As the intra-application DSE only considers the required tiles of each type to execute a single considered application, it does not yet regard reliability, cost, and the embodied emissions, for which the total target architecture is needed. These missing objectives are included into the sustainability-aware inter-application DSE, described next.

4.2 Inter-Application DSE

The sustainability-aware inter-application DSE receives the optimized sets of constraint graphs H_a per application $a \in A$ and then explores sustainable target architectures and a selection of constraint graphs, which minimize the objectives of the mapping problem. This work assumes that the user specifies the location of the system at use, while the DSE explores the production location $\mu \in W$, incorporating the trade-off between cost and sustainability. Formally, the DSE optimizes a selection of constraint graphs $\eta' = \{\eta_{a_1}, ..., \eta_{a_{|A|}}\}$, and a *production location* μ such that the global goals of sustainability $Su(\eta', \xi, \mu)$ and reliability $Re(\xi)$ are maximized, and the mean deadline miss rate $DM(\eta', \xi)$ and the cost $Co(\xi, \mu)$ are minimized:

$$\underset{\eta', \xi, \mu}{\text{maximize}}(Su(\eta', \xi, \mu), Re(\xi), -DM(\eta', \xi), -Co(\xi, \mu)) \tag{5}$$

The architecture ξ is given implicitly by the union of resources allocated for each application, exploiting spatial isolation and composability [24]. In the following, each objective is discussed in more detail.

4.2.1 Sustainability

We maximize sustainability by minimizing the emissions $Su(\eta'_v, \xi, \mu)$ of the constraint graph vector η'_v on architecture ξ, estimated by the sum of the embodied emissions (scope 1 and scope 2) and the expected operational emissions over the lifetime of the MPSoC:

$$Su(\eta'_v, \xi, \mu) = -F_{\text{Scope1}}(\xi, \mu) - F_{\text{Scope2}}(\xi, \mu) - F_{\text{op}}(\eta'_v, \xi) \tag{6}$$

The *embodied scope 1* emissions F_{Scope1} can be estimated by the chip yield y, number of produced chips n_c, the number of wafers per chip $n_{\text{w/c}}$ for the target architecture ξ and the CO_2 emissions per wafer $E_{\text{CO}_2/\text{w}}$ at location μ (extended from [9]):

$$F_{\text{Scope1}}(\xi, \mu) = \frac{1}{y} \cdot \lceil n_c \cdot n_{\text{w/c}}(\xi) \rceil \cdot E_{\text{CO}_2/\text{w}}(\mu) \tag{7}$$

The chip yield y is the proportion of defect-free chips on a wafer, with defects being caused by contamination and other factors during manufacturing. The emissions per wafer $E_{\text{CO}_2/\text{w}}(\mu)$ depend on factors such as wafer size and semiconductor feature size. [17] reports a CO_2-equivalent GHG emission of 61 kg for a 300 mm wafer based on a 130 nm technology node, which can be reduced to about 13 kg of carbon equivalent by exhaust gas treatment, neutralizing or filtering out most of the NF_3 and CF_4 gases. [5] provides recent data on GHG emissions from wafer production, showing the gas quantities and compositions for different technology nodes (28 to 3 nm) and manufacturing processes. Emissions are greatly reduced by filtering out NF_3. The number of chips that can fit on a wafer depend on the wafer's diameter ζ and die size α_{die}:

$$n_{\text{w/c}}(\xi) = \frac{\pi \cdot \zeta^2}{4 \cdot \alpha_{\text{die}}(\xi)} - 0.58 \frac{\pi \cdot \zeta}{\sqrt{\alpha_{\text{die}}(\xi)}} \tag{8}$$

Since wafers are round and chips are rectangular, not all the wafer's area can be utilized, as estimated by the formula's second term. The die size $\alpha_{\text{die}}(\xi)$ is determined by the core area $\alpha_{\text{core}}(\xi)$, i.e., the area in which the SoC tiles on the architecture ξ are placed. By estimating the number of transistors for logic $n_{\text{tr,log}}$ and memory $n_{\text{tr,mem}}$ with respective transistor density χ_{log} and χ_{mem}, the core chip area amounts to:

$$\alpha_{\text{core}}(\xi) = g \cdot \left(\frac{n_{\text{tr,log}}(\xi)}{\chi_{\text{log}}} + \frac{n_{\text{tr,mem}}(\xi)}{\chi_{\text{mem}}}\right) \tag{9}$$

In this formula, g is the wiring factor, considering the additional space required for the connections between the transistors. However, the core area is surrounded by additional peripheral area containing on-chip components that do not correspond directly to the MPSoC architecture like, for instance, I/O pad cells. With the width of this overhead given as δ, the die area can then be estimated as:

$$\alpha_{\text{die}}(\xi) = (\lceil \sqrt{\alpha_{\text{core}}(\xi)} \rceil + \delta)^2 \tag{10}$$

The *embodied scope 2* emissions F_{Scope2} can be calculated as follows where eng_{w} denotes the needed energy to produce one wafer (extended from [9]):

$$F_{\text{Scope2}}(\xi, \mu) = \frac{1}{y} \cdot \lceil n_c \cdot n_{\text{w/c}}(\xi) \rceil \cdot eng_{\text{w}} \cdot \rho_\mu \tag{11}$$

The factor ρ_μ expresses the CO_2 intensity of the energy mix at location μ during production. [12] present the CO_2 efficiency of energy production in different regions. The energy required to produce a wafer depends on its size and the semiconductor's feature size. According

to [17], this includes energy for processing quartz into silicon, refining silicon into ingots, and further processing into wafers. Additional energy is needed for chemicals, semiconductor fabrication, and maintaining infrastructure. A separate study [10] provides further values across various semiconductor fabs. [5] presents the energy consumption for various semiconductor manufacturing processes (28 nm to 3 nm), assuming that the supporting infrastructure (e.g., cooling, vacuum) accounts for 40 % of total energy use. The *operational emissions* F_{op} can be computed as follows. Let $C \subseteq \mathbb{R} \times W$ be the set of tuples (β, λ) which describes that $\beta \cdot 100\,\%$ of the produced chips are expected to be operated in country λ. By considering both the ratio of chips $n_c \cdot \beta$ that are operated using a local energy mix ρ_λ and the energy consumption eng of the chip during its lifetime, see Eq. (3), the total operational emissions can be estimated as:

$$F_{\text{op}}(\eta'_v, \xi) = \sum_{(\beta,\lambda)\in C} (n_c \cdot \beta \cdot \rho_\lambda \cdot \sum_{\eta_a \in \eta'_v} eng(\eta_a, \xi)) \tag{12}$$

4.2.2 Reliability

The reliability $Re(\xi)$ of the architecture ξ models the mean time to failure (MTTF) of the system. We approximate the reliability as follows[1]:

$$Re(\xi) = \int_0^\infty \prod_{\psi\in\xi} \varsigma_\psi(t)dt = \int_0^\infty \prod_{\psi\in\xi} e^{-\lambda_\psi t}dt = \int_0^\infty e^{-t\sum_{\psi\in\xi}\lambda_\psi}dt = \frac{1}{\sum_{\psi\in\xi}\lambda_\psi} \tag{13}$$

In the above conservative model, it is assumed that the applications allocate all resources of the SoC system, and that the system is considered defective already once a single of those processing resources fails. ς_ψ denotes the reliability function of processing resource ψ and λ_ψ its failure rate. To determine the failure rate λ_ψ in Eq. (13), e.g. [16] can be used.

4.2.3 Performance

The performance $DM(\eta'_v, \xi)$ of the constraint graph vector η'_v on architecture ξ is estimated as follows. Given the set of periodic real-time applications where each application $a \in A$ has a priority p_a and a deadline θ_a, we want to minimize the average deadline misses $dm(\eta_a, \xi)$ of all applications that are mapped to the SoC:

$$DM(\eta'_v, \xi) = \frac{1}{|A|} \cdot \sum_{\eta_a\in\eta'_v} (1 - dm(\eta_a, \xi)) \cdot p_a \tag{14}$$

By dividing the sum of deadline misses of running applications by the total number of applications $|A|$, we reward systems implementing more applications than others with a higher performance $DM(\eta'_v, \xi)$.

4.2.4 Cost

The cost $Co(\xi)$ of an architecture ξ is modeled by the wafer cost κ_{w}, and the electricity cost κ_{el} per wafer:

$$Co(\xi) = \frac{\kappa_{\text{w}} + eng_{\text{w}} \cdot \kappa_{\text{el}}}{n_{\text{w/c}} \cdot y} \tag{15}$$

[1] Our reliability model (Eq. (13)) assumes independent failures of processing resources, equal load distribution, and all applications starting at the same time. While this simplification allows for efficient computation, it may overestimate system-level failure rates for architectures with redundancy. Future work can incorporate more complex reliability models to account for fault tolerance.

In this formula, $n_{\mathrm{w/c}}$ describes the number of chips per wafer (see Eq. (8)). The yield y of defect-free chips on a wafer is modeled using the negative binomial distribution, which accounts for defect clustering with the ϵ:

$$y = \left(1 + \frac{\alpha_{\mathrm{chip}} \cdot \chi_0}{\epsilon}\right)^{-\epsilon} \tag{16}$$

Here, α_{chip} is the critical chip area, χ_0 is the defect density. The defect density χ_0 depends on the production environment and structure size, with smaller structures more prone to defects. The cluster factor ϵ indicates defect distribution: higher values suggest evenly distributed defects (lower yield), while lower values indicate concentrated defects (higher yield). Typical values for ϵ range from 0.3 to 5 [15].

4.3 Optimization Time

The total optimization time of the proposed sustainability-aware DSE is composed of the optimization times of the intra-application DSEs, the inter-application DSE, and the negligible overhead t_{bind} of the binding algorithm. As the separate intra-application DSEs can be executed in parallel, their total optimization time is the maximum across the DSEs, which depends on the number of generations $\#\mathrm{gen}_a$ and the time per generation $t_{\mathrm{DSE_a,ge}}$. The optimization time for the inter-application sustainability-aware DSE is given by the number of generations $\#\mathrm{gen}_{\mathrm{sus}}$ and the time per generation $t_{\mathrm{sus,ge}}$:

$$t_{\mathrm{total}} = \#\mathrm{gen}_{\mathrm{sus}} \cdot t_{\mathrm{sus,ge}} + t_{\mathrm{bind}} + \max_{a \in A} \left(\#\mathrm{gen}_a \cdot t_{\mathrm{DSE_a,ge}}\right) \tag{17}$$

In our experiments, we performed the inter-application DSE for 1,000 generations, with $t_{\mathrm{sus,ge}} = 20\,\mathrm{ms}$. For the intra-application DSE, we also performed 1,000 generations, where the time per generation depends on the number of new individuals created in this generation and the time for evaluating an individual. Assessing the execution time of an individual requires simulating the execution of the application under the individual mapping on the platform, which takes up to half a second. For input-dependent applications, a separate execution is needed for every training data, leading up to even higher execution times. The optimization time is thus typically dominated by the intra-application phase, while the inter-application DSE is highly efficient and only takes around 2 seconds. Hence, the proposed approach is highly scalable and can be used to design large and complex embedded systems.

5 Evaluation

The proposed sustainable-aware DSE is highly customizable and can be fine-tuned by the user. This customization allows a flexible approach across different production scenarios, but requires accurate settings for a reliable result. Since these parameters are often classified and not openly available, we resort in this work to rather limited publicly available data, e.g., for modeling the sustainability objectives, we assumed a lifetime of 5 years, used the emission data from [12, 17, 12, 10, 5], and extracted reliability data from [16]. The lack of accurate data does, however, not invalidate our results, since although different settings may alter the design space, the superiority of our approach in exploring the entire design space remains consistent across various parameter configurations. In the following evaluation, the ray-tracing, Mjpeg, and stitching application from [24], combined with the auto consumer, auto industry, office automation, and telecom applications from [8] serve

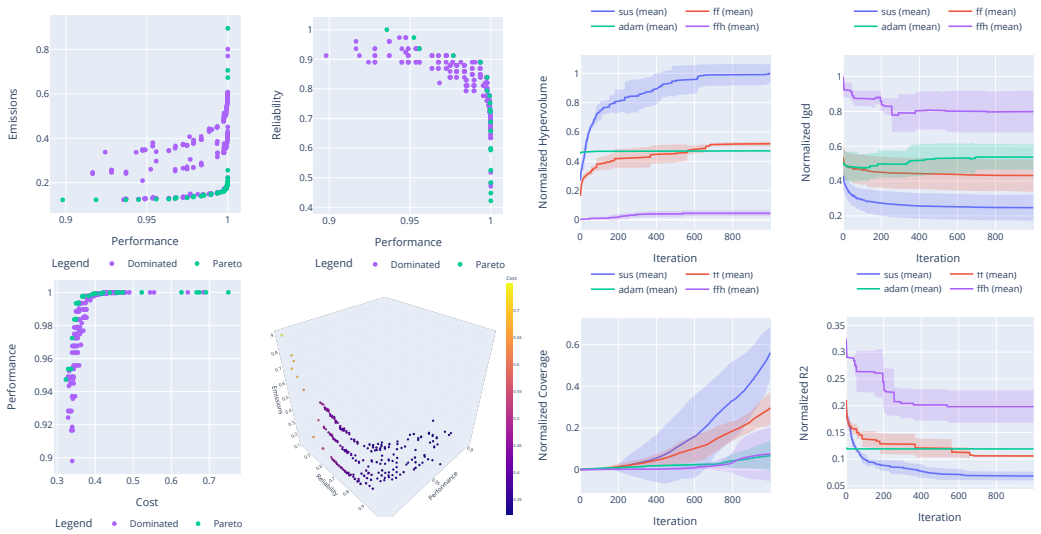

Figure 3 Different objective trade-offs of the obtained approximation of the Pareto front by the proposed sustainability-aware DSE (left) and evolution of the normalized hypervolume [4], IGD [30], R2 [26], and coverage [31] metric for FirstFit (ff), AdaMD+ (adam), Composition (ffh), and the proposed sustainability-aware DSE (sus) (right). Shown are the average results across 5 runs (solid line) with the shaded area representing the ±1 standard deviation from the mean.

as the soft real-time applications A that shall be executed on a SoC target architecture. All DSEs are implemented in Opt4J [18]. Fig. 3 shows on the left the embedded system solutions explored by the proposed multi-objective sustainability-aware DSE concerning the objectives of emissions, reliability, performance and cost. To facilitate comparability, the objectives are normalized in the following. The comparison of emissions and performance shows a clear trade-off, where higher performance correlates with increased emissions. In the mid-performance range, significant performance improvements can be achieved with only small increases in emissions. This finding suggests that it's possible to design systems with high performance without necessarily incurring large environmental costs. However, the graph shows a sharp increase in emissions at the highest performance levels. This steep rise indicates that aiming for maximum performance leads to disproportionately higher emissions. The data suggests that by slightly reducing performance targets from their maximum, designers can potentially achieve large reductions in emissions, leading to more environmentally friendly systems without major performance losses. The normalized reliability versus performance plot demonstrates an inverse relationship. As performance increases, reliability tends to decrease, with the Pareto front showing a steep drop in reliability at the highest performance levels. This suggests a critical trade-off point where marginal performance gains come at

Table 1 Normalized key metrics upon convergence.

Metric	R2 (↓)	Hypervolume (↑)	IGD (↓)	Coverage (↑)
Composition	0.20	0.05	0.80	0.08
AdaMD+	0.12	0.47	0.53	0.07
FirstFit	0.11	0.52	0.43	0.30
Proposed	**0.07**	**1.0**	**0.25**	**0.56**

a substantial cost to reliability. Finally, the performance versus cost plot exhibits a clear positive correlation, showing how higher performance invariably comes at a greater cost. At the highest performance levels, the cost increases sharply. These visualizations indicate the complexity of the design space and the importance of a DSE approach in identifying non-obvious trade-offs that might be overlooked by simpler methods. To further justify the need for a DSE incorporating sustainability aspects, We compare the proposed sustainability-aware DSE against various baselines regarding hypervolume, inverted generational distance (IGD), coverage, and r2 indicator [31]. This includes *FirstFit*, an extension of the multi-objective DSE from [24] with reliability and cost objectives, *AdaMD+*, an extension of the fast multi-application mapping optimization methodology by [20], and *Composition*, which uses *AdaMD* to determine the architecture as a composition of the mappings per application. This evaluation is based on the emission data from [12, 17, 12, 10, 5]. Acc. to Fig. 3, our proposed approach demonstrates superior performance across all metrics. The hypervolume metric rapidly converges to 1.0, while the best baseline, FirstFit, plateaus at 0.5. The IGD plot also shows that our solutions are consistently closer to the true Pareto front. Our method further excels in the coverage metric, finding nearly twice as many non-dominated solutions as FirstFit. Finally, the R2 indicator validates our method's superior convergence and diversity. Table 1 lists the final R2, hypervolume, IGD, and coverage values upon convergence.

6 Summary

As the demand for computing continues to grow, the environmental impact of this development is often neglected, despite the fact that data centers already contribute massively to global emissions. However, designing an environmentally friendly embedded SoC system without compromising performance, reliability, or cost is quite complex, as all objectives must be carefully balanced. As a solution, this work proposes a multi-objective, sustainability-aware two-phase DSE to find Pareto-optimal tile-based MPSoC architectures for a set of given time-critical applications. In an intra-application phase, the DSE first determines optimal mappings for each individual application in the form of constraint graphs, and then explores the combination of these mappings in an inter-application phase. Here, the DSE is sustainability-aware by considering the embodied *scope 1* and *scope 2* emissions of the architecture during chip fabrication and its operation as objective. In addition, the DSE also considers the reliability, manufacturing cost, i.e., the die area, and the performance on the set of applications. In a thorough evaluation against various baselines from the literature, the proposed work dominates over four selected metrics significantly. Particularly, it achieves nearly double the hypervolume than the best baseline, indicating that it explores a wider part of the design space, determines solutions closer to the true Pareto front, and explores a more diverse set of non-dominated solutions.

References

1 Athena Abdi and Hamid R. Zarandi. HYSTERY: A Hybrid Scheduling and Mapping Approach to Optimize Temperature, Energy Consumption and Lifetime Reliability of Heterogeneous Multiprocessor Systems. *J. Supercomput.*, 74(5):2213–2238, 2018. doi:10.1007/s11227-018-2248-2.

2 Benny Akesson, Anca Mariana Molnos, Andreas Hansson, Jude Angelo Ambrose, and Kees Goossens. Composability and Predictability for Independent Application Development, verification, and execution. In Michael Hübner and Jürgen Becker, editors, *Multiprocessor System-on-Chip - Hardware Design and Tool Integration*, pages 25–56. Springer, 2011. doi:10.1007/978-1-4419-6460-1_2.

3 Nidhi Anantharajaiah, Tamim Asfour, Michael Bader, Lars Bauer, Jürgen Becker, Simon Bischof, Marcel Brand, Hans-Joachim Bungartz, Christian Eichler, Khalil Esper, Joachim Falk, Nael Fasfous, Felix Freiling, Andreas Fried, Michael Gerndt, Michael Glaß, Jeferson Gonzalez, Frank Hannig, Christian Heidorn, Jörg Henkel, Andreas Herkersdorf, Benedict Herzog, Jophin John, Timo Hönig, Felix Hundhausen, Heba Khdr, Tobias Langer, Oliver Lenke, Fabian Lesniak, Alexander Lindermayr, Alexandra Listl, Sebastian Maier, Nicole Megow, Marcel Mettler, Daniel Müller-Gritschneder, Hassan Nassar, Fabian Paus, Alexander Pöppl, Behnaz Pourmohseni, Jonas Rabenstein, Phillip Raffeck, Martin Rapp, Santiago Narváez Rivas, Mark Sagi, Franziska Schirrmacher, Ulf Schlichtmann, Florian Schmaus, Wolfgang Schröder-Preikschat, Tobias Schwarzer, Mohammed Bakr Sikal, Bertrand Simon, Gregor Snelting, Jan Spieck, Akshay Srivatsa, Walter Stechele, Jürgen Teich, Isaías A. Comprés Ureña, Ingrid Verbauwhede, Dominik Walter, Thomas Wild, Stefan Wildermann, Mario Wille, Michael Witterauf, and Li Zhang. *Invasive Computing*. FAU University Press, 2022. doi:10.25593/978-3-96147-571-1.

4 Anne Auger, Johannes Bader, Dimo Brockhoff, and Eckart Zitzler. Hypervolume-based multiobjective optimization: Theoretical foundations and practical implications. *Theo. Comp. Sci.*, 425:75–103, 2012. doi:10.1016/J.TCS.2011.03.012.

5 M Garcia Bardon, P Wuytens, L-Å Ragnarsson, G Mirabelli, D Jang, G Willems, A Mallik, A Spessot, J Ryckaert, and B Parvais. DTCO including sustainability: Power-performance-area-cost-environmental score (PPACE) analysis for logic technologies. In *2020 IEDM*, pages 41–4. IEEE, 2020.

6 Tilera Corporation. Tile Processor Architecture Overview for the Tile-Gx Series. https://cdn.manesht.ir/17871___210769647-UG130-ArchOverview-TILE-Gx.pdf, 2012. [Online; accessed 04-July-2023].

7 Benoît Dupont de Dinechin, Renaud Ayrignac, Pierre-Edouard Beaucamps, Patrice Couvert, Benoit Ganne, Pierre Guironnet de Massas, François Jacquet, Samuel Jones, Nicolas Morey Chaisemartin, Frédéric Riss, and Thierry Strudel. A Clustered Manycore Processor Architecture for Embedded and Accelerated Applications. In *HPEC 2013, Waltham, MA, USA, September 10-12, 2013*, pages 1–6. IEEE, 2013. doi:10.1109/HPEC.2013.6670342.

8 Robert P. Dick. Embedded System Synthesis Benchmarks Suite (E3S), 2002. Accessed September 11, 2024. URL: https://ziyang.eecs.umich.edu/~dickrp/e3s/.

9 Lieven Eeckhout. Kaya for Computer Architects: Toward Sustainable Computer Systems. *IEEE Micro*, 43(1):9–18, 2023. doi:10.1109/MM.2022.3218034.

10 Robert Falkner. The Paris Agreement and the new logic of international climate politics. *International Affairs*, 92(5):1107–1125, 2016.

11 Charlotte Freitag, Mike Berners-Lee, Kelly Widdicks, Bran Knowles, Gordon S. Blair, and Adrian Friday. The real climate and transformative impact of ICT: A critique of estimates, trends, and regulations. *Patterns*, 3(8):100576, 2022. doi:10.1016/J.PATTER.2022.100576.

12 Udit Gupta, Young Geun Kim, Sylvia Lee, Jordan Tse, Hsien-Hsin S Lee, Gu-Yeon Wei, David Brooks, and Carole-Jean Wu. Chasing carbon: The elusive environmental footprint of computing. In *HPCA*, pages 854–867. IEEE, 2021. doi:10.1109/HPCA51647.2021.00076.

13 Jan Heisswolf, Ralf König, and Jürgen Becker. A Scalable NoC Router Design Providing QoS Support Using Weighted round robin scheduling. In *ISPA, Leganes, Madrid, Spain, July 10-13, 2012*, pages 625–632. IEEE Computer Society, 2012. doi:10.1109/ISPA.2012.93.

14 Jason Howard, Saurabh Dighe, Sriram R. Vangal, Gregory Ruhl, Nitin Borkar, Shailendra Jain, Vasantha Erraguntla, Michael Konow, Michael Riepen, Matthias Gries, Guido Droege, Tor Lund-Larsen, Sebastian Steibl, Shekhar Borkar, Vivek K. De, and Rob F. Van der Wijngaart. A 48-Core IA-32 Processor in 45 nm CMOS Using On-Die Message-Passing and DVFS for performance and power scaling. *IEEE J. Solid State Circuits*, 46(1):173–183, 2011. doi:10.1109/JSSC.2010.2079450.

15 Jung Yoon Hwang. *Spatial stochastic processes for yield and reliability management with applications to nano electronics*. Texas A&M University, 2004.

16 Texas Instruments. MTBF and FIT Rate Estimator, 2024. URL: https://www.ti.com/quality/docs/estimator.tsp.

17 Nikhil Krishnan, Sarah Boyd, Ajay Somani, Sebastien Raoux, Daniel Clark, and David Dornfeld. A hybrid life cycle inventory of nano-scale semiconductor manufacturing. *Environmental science & technology*, 42(8):3069–3075, 2008.

18 Martin Lukasiewycz, Michael Glaß, Felix Reimann, and Jürgen Teich. Opt4J: A Modular Framework for Meta-Heuristic Optimization. In *GECCO, Dublin, Ireland, July 12-16*, pages 1723–1730. ACM, 2011. doi:10.1145/2001576.2001808.

19 Behnaz Pourmohseni, Michael Glaß, Jörg Henkel, Heba Khdr, Martin Rapp, Valentina Richthammer, Tobias Schwarzer, Fedor Smirnov, Jan Spieck, Jürgen Teich, Andreas Weichslgartner, and Stefan Wildermann. Hybrid Application Mapping for Composable Many-Core Systems: Overview and Future Perspective. *Journal of Low Power Electronics and Applications*, 10:1–37, 2020. doi:10.3390/jlpea10040038.

20 Basireddy Karunakar Reddy, Amit Kumar Singh, Bashir M. Al-Hashimi, and Geoff V. Merrett. AdaMD: Adaptive Mapping and DVFS for Energy-Efficient Heterogeneous multicores. *IEEE Trans. Comput. Aided Des. Integr. Circuits Syst.*, 39(10):2206–2217, 2020. doi:10.1109/TCAD.2019.2935065.

21 Tobias Schwarzer, Andreas Weichslgartner, Michael Glaß, Stefan Wildermann, Peter Brand, and Jürgen Teich. Symmetry-Eliminating Design Space Exploration for Hybrid Application mapping on many-core architectures. *TCAD*, 37(2):297–310, 2018. doi:10.1109/TCAD.2017.2695894.

22 Jan Spieck, Stefan Wildermann, and Jürgen Teich. On Transferring Application Mapping Knowledge Between Differing MPSoC architectures. *IEEE Trans. Comput. Aided Des. Integr. Circuits Syst.*, 41(11):4289–4300, 2022. doi:10.1109/TCAD.2022.3197527.

23 Jan Spieck, Stefan Wildermann, and Jürgen Teich. A Learning-based Methodology for Scenario-aware Mapping of Soft Real-time applications onto heterogeneous mpsocs. *ACM Trans. Design Autom. Electr. Syst.*, 28(1):4:1–4:40, 2023. doi:10.1145/3529230.

24 Jan Spieck, Stefan Wildermann, and Jürgen Teich. A Scenario-Based DVFS-Aware Hybrid Application Mapping Methodology for MPSoCs. *ACM Trans. Des. Autom. Electron. Syst.*, 29(4), June 2024. doi:10.1145/3660633.

25 Chetan Choppali Sudarshan, Nikhil Matkar, Sarma B. K. Vrudhula, Sachin S. Sapatnekar, and Vidya A. Chhabria. ECO-CHIP: Estimation of Carbon Footprint of Chiplet-based Architectures for sustainable VLSI. In *HPCA, Edinburgh, United Kingdom, March 2-6, 2024*, pages 671–685. IEEE, 2024. doi:10.1109/HPCA57654.2024.00058.

26 Heike Trautmann, Tobias Wagner, and Dimo Brockhoff. R2-EMOA: Focused Multiobjective Search Using R2-Indicator-Based selection. In Giuseppe Nicosia and Panos M. Pardalos, editors, *LION 7, Catania, Italy, January 7-11, 2013, Revised Selected Papers*, volume 7997, pages 70–74. Springer, 2013. doi:10.1007/978-3-642-44973-4_8.

27 Peter van Stralen and Andy D. Pimentel. Scenario-Based Design Space Exploration of MPSoCs. In *ICCD 2010, 3-6 October 2010, Amsterdam, The Netherlands, Proceedings*, pages 305–312, 2010. doi:10.1109/ICCD.2010.5647727.

28 Dominik Walter, Marcel Brand, Christian Heidorn, Michael Witterauf, Frank Hannig, and Jürgen Teich. ALPACA: an Accelerator Chip for Nested Loop Programs. In *IEEE International Symposium on Circuits and Systems, ISCAS 2024, Singapore, May 19-22, 2024*, pages 1–5. IEEE, 2024. doi:10.1109/ISCAS58744.2024.10558549.

29 World Resources Institute, C40 Cities Climate Leadership Group, and ICLEI Local Governments for Sustainability. *Global Protocol for Community-Scale Greenhouse Gas Emission Inventories*. World Resources Institute, Washington, DC, 2014. [Online; accessed 2024-09-16].

30 Gary G. Yen and Zhenan He. Performance Metric Ensemble for Multiobjective Evolutionary Algorithms. *IEEE Trans. Evol. Comput.*, 18(1):131–144, 2014. doi:10.1109/TEVC.2013.2240687.

31 Eckart Zitzler and Lothar Thiele. Multiobjective evolutionary algorithms: a comparative case study and the strength Pareto approach. *IEEE Trans. Evol. Comput.*, 3(4):257–271, 1999. doi:10.1109/4235.797969.