Review Article
Development of Multiobjective High-Level Synthesis for FPGAs

Darian Reyes Fernandez de Bulnes, Yazmin Maldonado 🄸🄳**, and Leonardo Trujillo** 🄸🄳

Tecnológico Nacional de México, IT Tijuana, Posgrado en Ciencias de la Ingenieria, Tijuana, BC, Mexico

Correspondence should be addressed to Yazmin Maldonado; yaz.maldonado@tectijuana.edu.mx

Traditionally, the High-Level Synthesis (HLS) for Field Programmable Gate Array (FPGA) devices is a methodology that transforms a behavioral description, as the timing-independent specification, to an abstraction level that is synthesizable, like the Register Transfer Level. This process can be performed under a framework that is known as Design Space Exploration (DSE), which helps to determine the best design by addressing scheduling, allocation, and binding problems, all three of which are NP-hard problems. In this manner, and due to the increased complexity of modern digital circuit designs and concerns regarding the capacity of the FPGAs, designers are proposing novel HLS techniques capable of performing automatic optimization. HLS has several conflicting metrics or objective functions, such as delay, area, power, wire length, digital noise, reliability, and security. For this reason, it is suitable to apply Multiobjective Optimization Algorithms (MOAs), which can handle the different trade-offs among the objective functions. During the last two decades, several MOAs have been applied to solve this problem. This paper introduces a comprehensive analysis of different MOAs that are suitable to perform HLS for FPGA devices. We highlight significant aspects of MOAs, namely, optimization methods, intermediate structures where the optimizations are performed, HLS techniques that are addressed, and benchmarks and performance assessments employed for experimentation. In addition, we show the analysis of how multiple objectives are optimized currently in the algorithms and which are the objective functions that are optimized. Finally, we provide insights and suggestions to contribute to the solution of major research challenges in this area.

1. Introduction

Field Programmable Gate Arrays (FPGAs) designs are made with High-Level Synthesis (HLS). HLS also is known as behavioral synthesis or architectural synthesis, the process to transform an algorithmic description to a synthesizable Register Transfer Level (RTL) netlist. HLS allows designers to work at a higher level of abstraction by using high-level languages such as C/C++ to define the hardware description. Typically, behavioral description, also known as algorithmic level design or system-level design, defines inputs, outputs, and data flow of the behavior inside the algorithm in terms of operations to be performed. Inwardly, this description is usually represented (as an intermediate structure) with an acyclic directed graph. It establishes the data dependencies indicated in the data flow and input/output relations of the design [1]. For any behavioral description, there may be many possible RTL implementations, each one with its own features.

1.1. High-Level Synthesis. HLS can be performed under a framework that is known as Design Space Exploration (DSE), which helps to compute the best design using scheduling, allocation, and binding techniques. All of these tasks are NP-hard problems [2]. Scheduling defines how the design operations will be scheduled into clock cycles. Allocation determines the type and the number of hardware resources (for instance, Functional Units (FUs), storage, or connectivity components) needed to satisfy the design constraints. Binding, also known as assignment mapping and module selection, determines how each variable (in each clock cycle) will be linked to an FU. As Coussy et al. stated, "allocation, scheduling, and binding can be performed simultaneously or in specific sequence depending on the strategy and algorithms used" ([3]; p. 5).

In [4], the evolution of HLS for FPGAs and the HLS tools with single-objective optimization are discussed; according to this review, it is clear that HLS is important because (i)

software programmers want to use FPGA devices to accelerate tasks, so without having knowledge about Hardware Description Language (HDL)—VHDL (Very High-Speed Integrated Circuit (VHSIC) Hardware Description Language) or Verilog—they can create circuit designs; (ii) designing at a higher level of abstraction leads to increased productivity; for example, software debugging is faster than hardware debugging; and (iii) this process has a lot of potential to perform optimizations. Recently, HLS has been applied to a variety of applications with significant benefits in terms of performance and energy consumption. For instance, [5] presents a case study comparing HLS and handwritten RTL implementations, where HLS achieves a drastic reduction in delay. Another example is a convolutional neural network developed in [6], demonstrating the ability of HLS to support complex algorithms. Additionally, there are many practical applications of HLS where multiobjective optimization was applied, for example, custom processor design to find an optimized architecture [7], watermarking to provide protection of authorship in reusable Intellectual Property (IP) [8], and exploration of a low-cost Trojan security hardware [9].

1.2. Multiobjective Optimization in High-Level Synthesis.

There are several opportunities to perform optimizations in HLS during scheduling, allocation, and binding. These optimizations are highly multiobjective by nature, with conflicting objective functions. To deal with that scenario, it is necessary to apply Multiobjective Optimization Algorithms (MOAs). These algorithms maintain a trade-off between conflicting metrics. Multiobjective optimization is dedicated to solve problems in which a set of objective functions $f_1(x), f_2(x), \ldots, f_m(x)$ must be optimized simultaneously. A multiobjective optimization problem where all objective functions should be minimized can be defined as

$$\arg \min_{x \in D} F(x) = \langle f_1(x), f_2(x), \ldots, f_m(x) \rangle, \quad (1)$$

where D is known as the decision space. The image set O which results from projecting $F: D \longrightarrow O$ is called the objective space, which is the space where the objective vectors belong. An objective vector $v = (v_1, \ldots, v_m)$ dominates $u = (u_1, \ldots, u_m)$ if and only if all the components of v are equal or better than the corresponding components of u and at least one component of v is strictly better. For a multiobjective optimization problem where all objective functions are of minimization, Pareto dominance can be defined as

$$\forall i \in \{1, \ldots, m\}: v_i \leq u_i \wedge \exists i \in \{1, \ldots, m\}: v_i < u_i. \quad (2)$$

A point x^* is Pareto optimal if there is no other solution $x \in D$ that dominates it. The set of optimal Pareto solutions are the Pareto optimal set P^*. The Pareto Front (PF) is the image of the Pareto optimal set in the objective space $PF^* = F(x), x \in P^*$ [10]. Solutions to this problem should approximate the Pareto Front, instead of a single solution. The solution quality is commonly expressed in terms of Pareto dominance.

Always, it is desired to find an approximation with good convergence and diversity. Convergence is the proximity to the set of ideal points. Figure 1 provides two examples of PF approximations (minimization of two objective functions). The first plot (left) contains a set of solutions where some regions are not covered, so this PF is not attractive because the decision maker could lose important information of the PF. The second one (right) shows a front having a very good spread of solutions (diversity).

According to the HLS literature, the authors have tried to optimize the following objective functions, as shown in Figure 2.

(i) *Delay* is the total number of time steps or clock cycles. It is also called control step, timing, latency, or performance. This objective can be replaced by throughput, which is given as the ratio of the operating frequency to the latency multiplied by the input size. These system-level specifications are defined by the behavioral description.

(ii) *Area* is the total of occupied components in the device, i.e., FUs plus registers [11]. It is also called memory or space.

(iii) *Power* is the total power consumption (dynamic power plus static power).

(iv) *Wire length* is the measure of the overall interconnection length plus connectivity components used by the design, which is based on a global routing step. It is also called an interconnection or data path. This measurement must be computed out after binding.

(v) *Digital noise* is an estimation of computational errors plus noise propagation when the design contains real numbers, considering floating-point accuracy. When real numbers are represented by a limited number of bits, this causes a loss of information, which is usually considered as noise. It is also called error propagation.

(vi) *Reliability* refers to the need to avoid the presence of soft error (intermittent failure caused by neutrons and alpha particles). The probability that a soft error will occur depends on which types of FUs are used for the design operations since some FUs are ideal for certain types of operations.

(vii) *Temperature* should be minimized for every design because temperature variations and hotspots inside an FPGA can cause electronic failures.

(viii) *Security* is the protection against attacks, for instance, IP protection and reverse engineering attacks. This objective is also called robustness.

Most of these metrics should be minimized; only reliability and security must be maximized. According to the review of the state of the art that we have made in this survey, we find that multiobjective optimization works assume that the objective functions are in conflict but only one work verifies that some of the objective functions are in conflict [12]. According to the above, Table 1 presents a summary of
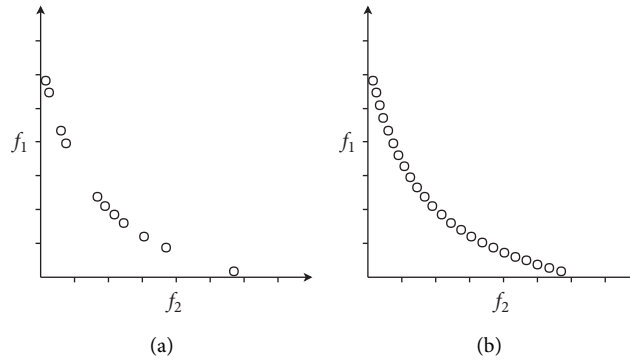
FIGURE 1: Examples of Pareto Front approximations: bad diversity (a) and good diversity (b).
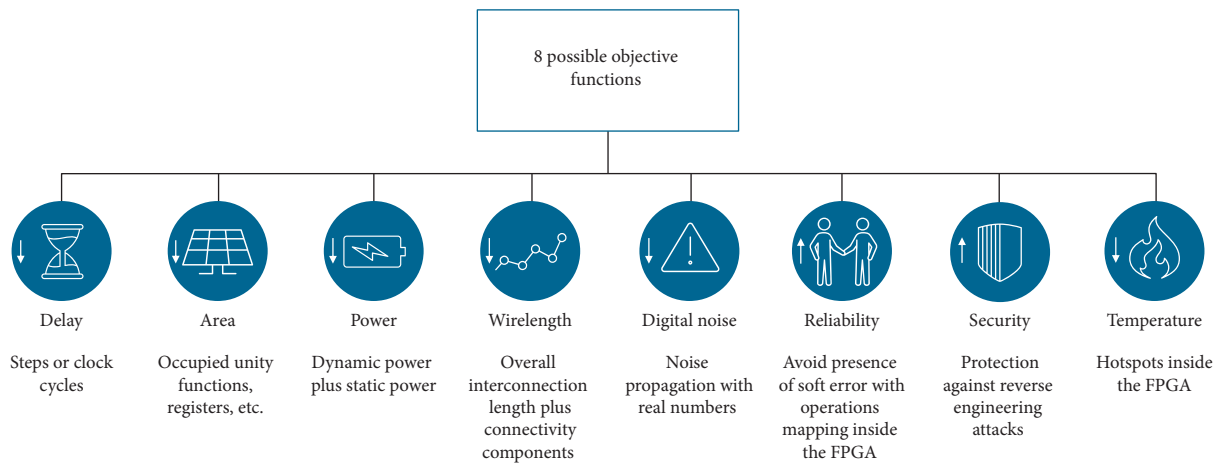

FIGURE 2: The 8 objective functions that have been proposed for the grand challenge optimization problem of HLS.

the possible conflicts between the eight objective functions. With the symbol ✓, we mark the objective functions where a conflict was demonstrated through a payoff matrices [12]. The objective functions that some authors have assumed are in conflict because they use a multiobjective approach are shown with the symbol ✓. Then, with the symbol ≈, we indicate the objective functions that we hypothesize are in conflict, according to what is known of the internal structure of FPGA devices. Finally, the symbol ? means that we do not know if the two objective functions are in conflict.

1.3. Contribution. Figure 3 is an Euler diagram of the optimization methods applied to HLS and highlights in black the subject area of this paper. The intersection of the optimization methods with three main stages (HLS, logic synthesis, and layout synthesis) involved in circuits implementation into FPGA devices is shown in [13]. Considering that multiobjective optimization is a subarea of optimization, this paper focuses on the multiobjective optimization of HLS for FPGA devices. For instance, HLS with single-objective optimization is not considered in this survey.

In summary, the novel contributions of this survey include the following:

(1) A review of the state of the art on HLS techniques with multiobjective optimization

(2) A description and comparisons of MOAs applied to HLS, analyzing optimization methods, HLS techniques, intermediate structures where optimization is performed, objective functions, the cost assignment strategies, and the benchmarks employed for experimentation

(3) Identification of major research challenges in this area that should be studied in the near future and notes on how to tackle them, including a hypothetical grand challenge to carry out HLS as a many-objective optimization problem with eight objective functions

The rest of the paper is organized as follows. Section 2 discusses the related survey, while Section 3 provides an overview of multiobjective optimization techniques in HLS. In Section 4, open issues are presented. Finally, we discuss our conclusions in Section 5 and outline future work in this area.

2. Related Surveys

The origins of HLS can be traced back to the ALERT system [14], developed by IBM at the T. J. Watson Research Center in 1969, but it was not until 2003 [15] that this task was studied as a combinatorial multiobjective problem for FPGA devices. Since then, several surveys concerning

TABLE 1: Possible conflicts of the eight objective functions. In this case, ✓, is the objective functions where a conflict was demonstrated. The symbol ✓ means the objective functions that some authors have assumed are in conflict. The symbol ≈ is the objective functions that we hypothesize are in conflict. And the symbol ? means that we do not know if the two objective functions are in conflict at the moment.

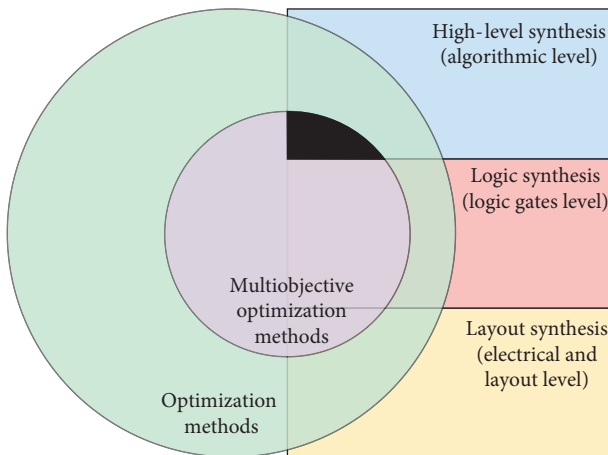| Objective functions	Delay	Area	Power	Wire length	Digital noise	Reliability	Security	Temperature
Delay		✓	✓	✓	✓	✓	≈	≈
Area			✓	✓	✓	✓	≈	≈
Power				?	✓	≈	✓	≈
Wire length					≈	≈	≈	≈
Digital noise						?	?	≈
Reliability							?	?
Security								≈
Temperature								

FIGURE 3: Euler diagram with different areas of knowledge, and the intersection in black is the study area of this survey.

optimizations (regardless of the number of objective functions) in HLS for FPGA devices have been published. The work of [16] provides a taxonomy of optimization in HLS on the basis of the intermediate representation used, such as a Data Flow Graph (DFG) or Control DFG (CDFG), and the performed tasks in HLS, namely, scheduling, allocation, and binding. It also enumerates research based on transformations of initial behavioral descriptions. The survey of [17] includes several approaches and frameworks for HLS optimization. For the first time in this area, a manuscript mentions multiobjective optimization and even explains some objective functions. It also presents details inside the optimization techniques, for example, how different types of internal structures are used to perform the optimizations. Four years later, [18] describes a retrospective of HLS and also explains the algorithms and academic software to apply optimization approaches.

Reference [19] presents a survey of memory, power, and temperature optimization techniques in HLS, explaining how these objective functions had been handled and the importance of analyzing the relationships (trade-off) between them. They also wrote notes about open issues, such as the order of optimization and code generation for low power. The survey presented in [20] deals with the three most popular objective functions: delay, area, and power. The paper also presented methodologies for multiobjective optimization and a classification of metaheuristics that were

used. A review of bioinspired optimization techniques was presented in [21], including a few evolutionary multiobjective approaches. They presented details about using both evolutionary computation and hardware design. The state of the art of HLS software tools is investigated in [22], which includes comparisons and evaluations of some software tools. The authors also present a taxonomy of input languages in software tools. Although the survey provides a comprehensive analysis of HLS software tools (commercial and academic), it does not mention which tools perform multiobjective optimization. The overview presented in [23] mentions strategies to solve the DSE problem by reducing the design space-time. The techniques are compared based on their performance improvement. They also include a few multiobjective approaches and performance metrics formulations.

In summary, it is important to note that none of the previous papers are completely focused on the subject area of MOAs in HLS for FPGA devices, the main contribution of this paper.

3. Multiobjective Approaches in High-Level Synthesis for FPGA Devices

In this section, the state of the art of MOAs in HLS for FPGA devices is presented.

In order to provide a visual representation of this survey, we created an online relational graph available at http://201.174.122.25/moo_hls_fpga [24].

The graph was created with the [25] library and arranged by the *edge-weighted force-directed* algorithm; the graph is shown in Figure 4. This graph allows you to search papers on multiobjective optimization in HLS. Circular gray nodes are papers in the state of the art and the number of citations is calculated using Google Scholar and it is represented by the size of the circle. When a paper is selected, the paper is a blue node. The multiobjective methods are classified in the red box. In the purple box, the MOAs are organized. In the light blue box, the cost assignment strategies are shown; the objective functions are shown in the yellow box. In the green box, the benchmarks are classified; finally, in the blue box, the compiler techniques are shown.

The edges are the connection between the papers with the multiobjective method, the cost assignment, the objective functions, the benchmarks used in the paper, the
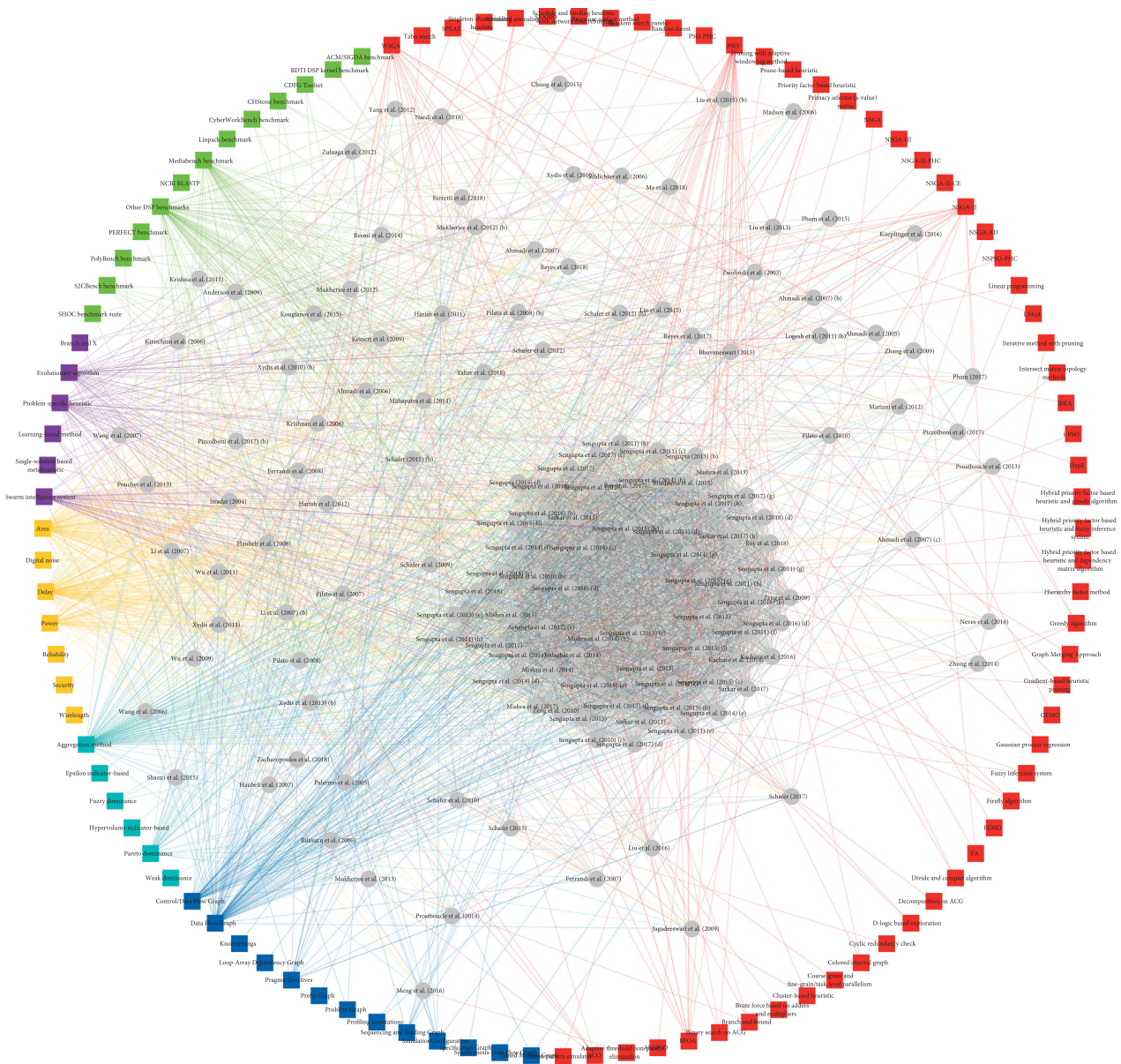
Figure 4: The relational graph of the state of the art of papers that apply MOAs in HLS.

compiler techniques, and the MOA. Figure 5 is an example of the graph, the paper [26] is the gray node, the multi-objective method is a branch and bound, Pareto dominance is the cost assignment strategy, DFG and CDFG are the compiler techniques, while branch and X is the MOA, area and power are the objective functions, and finally, experiments were carried out on the CDFG toolset benchmark. In this case, the gray nodes [27, 28] are the papers that have at least one author in common with respect to the blue node.

3.1. Optimization Approach. The optimization approaches can be classified into the following two categories [29]:

(1) *Compiler Techniques.* Behavioral description is represented by an acyclic directed graph, such as DFG, CDFG, Synchronous-DFG (SDFG), Loop-Array

Dependency Graph (LADG), Timed Marked Graph (TMG), Sequencing and Binding Graph (SBG), Prefix Graph, Problem Graph, and Specification Graph (CDFG and DFG are the most used, see Figure 6). All these intermediate structures have the same intention, to represent the semantics of the behavioral description. This technique requires, before optimization, converting (compilation) the behavioral description into the structure and, after the optimization, converting the optimized structure to RTL (RTL generation). Figure 7 presents the general framework of compiler techniques. The behavioral description and the components library are inputs, where the latter describes the characteristics of the FPGA device. The multiobjective optimization process must perform scheduling, allocation, and binding. Generally, the output is an
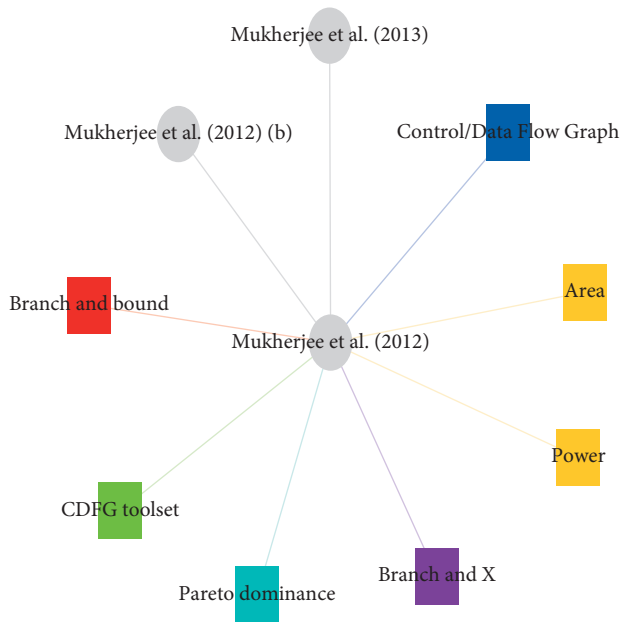
FIGURE 5: The paper [26] as an example query of the online graph developed in this survey.

HDL code that is ready for an EDA software tool to perform logic and layout synthesis.

(2) *HLS Tool as a Black Box*. These are approaches that explore the design space using commercial and academic HLS tools as black boxes. These approaches invoke the software tool to choose the objective functions. This technique is more comfortable to code because there is no need to worry about compilation, RTL generation, and estimations; but it is strongly dependent on the selected software tool. The variations of the scheduling, allocation, and binding are made through simulation tasks, knob settings, pragma directives, or profiling annotations inside the behavioral description. Figure 8 presents the general framework of approaches that use HLS tools as a black box. This methodology has a higher computational cost because in each iteration the selected HLS tool has to recompile the behavioral description and regenerate the RTL.

Figure 9 shows a taxonomy of multiobjecive methods based on [30]. In this survey, we focus on the highlighted boxes which are MOAs in HLS.

Six multiobjective methods have been studied by authors in this domain, organized as exact or approximate methods. According to Figure 9, these methods are *branch and X*, *problem-specific heuristics*, *single-solution-based heuristics*, *learning-based methods*, *evolutionary algorithms*, and *swarm intelligence systems*. Figure 10 shows six multiobjective methods represented in our relational graph in the state of the art. For instance, in the graph, we can see that the *branch and X* method is the least used since it has the least amount of edges. On the other hand, the most used multiobjective method is the swarm intelligence system.

Next, we explain each approach highlighted in Figure 9.

(1) Among exact methods [30], *branch and X* searches over the whole solution space, which is explored by dynamically building a tree whose root node represents the problem being solved. The optimization is performed by subdividing the problem into simpler subproblems.

(2) *Problem-specific heuristics* are, as the name implies, methods that are based specifically on the problem. They can achieve good results but cannot be applied generically to other problems.

(3) *Single-solution based metaheuristics* function as walks through local neighborhoods in the search space [30].

(4) *Learning-based methods* approximate the PF using machine learning models that learn by posing a classification or regression problem using a training set of instances. Then, the model acts over the decision-making process.

(5) *Evolutionary algorithms* are population-based metaheuristics, where solutions are selected and reproduced using variation operators (for instance, mutation and recombination). The main components to design an evolutionary algorithm are the following: representation, selection strategy, reproduction strategy, and replacement strategy. Population-based metaheuristics share common concepts. They start with a random initial population. Later, a new population is created in each generation that replaces the current population. This process iterates until a stop criterion is met.

(6) *Swarm intelligence systems* are another population-based metaheuristic, and these systems are inspired by the collective behavior of species such as ants, bees, and wasps. The key features of these algorithms are simple and nonsophisticated agents; they move in the search space and cooperate with each other by an indirect communication [30].

In [30], it is argued that it is better to use population-based metaheuristics than exact methods for multiobjective optimization problems. The reason is that, with exact methods, if the number of objective functions increases, then the algorithm design is more complex. In the same way, population-based metaheuristics are better than single-solution based metaheuristics because the population of solutions helps with diversity preservation on the PF, and consequently the convergence too.

On the other hand, in multiobjective optimization, to be able to compare solutions, it is necessary to apply cost assignment strategies. For a given solution, a cost assignment strategy maps to a cost vector (several objective functions) into a single value. Figure 11 shows a taxonomy of cost assignment strategies based on [30], where we are highlighting works of HLS in the literature.

Next, we describe in chronological order specific works of HLS for FPGA devices.

3.2. Branch and X Approaches. In [26, 27], a branch and bound algorithm was developed, which is capable of
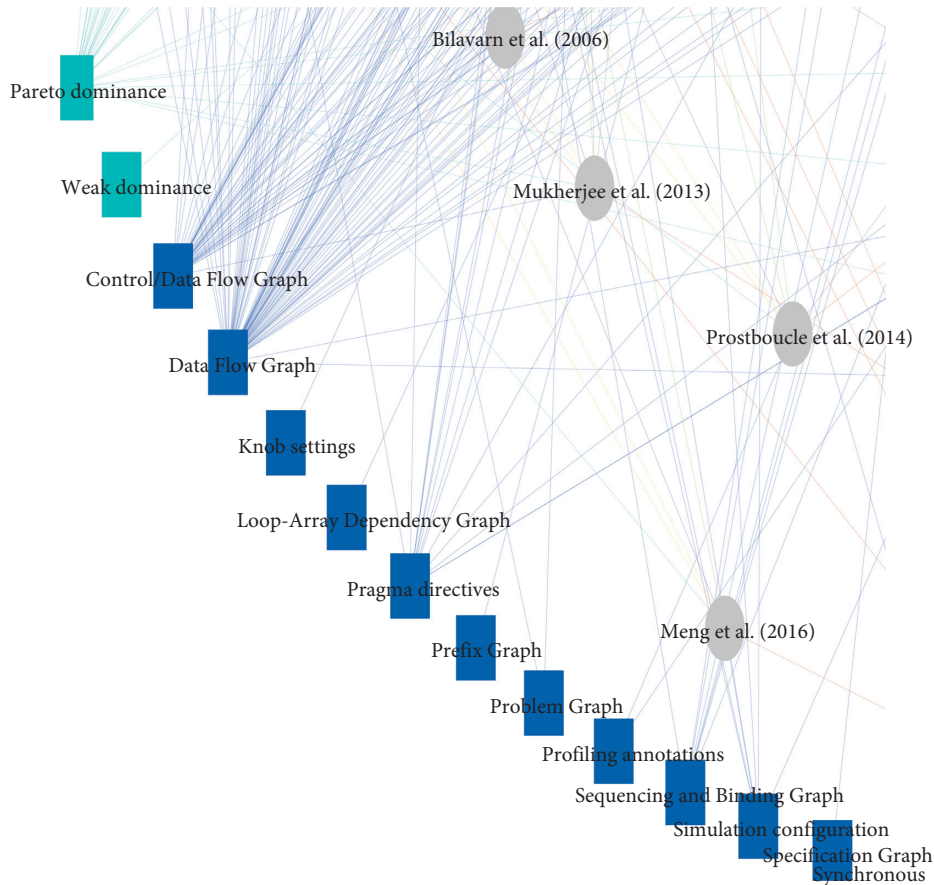
Figure 6: Graph of the intermediate structures used in the state of the art.
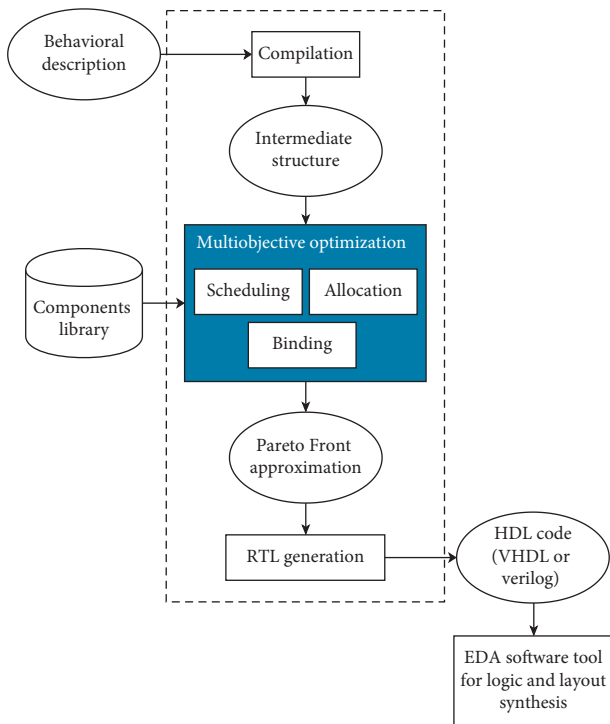

Figure 7: General framework of compiler techniques.

generating a nondominated solution with the CDFG toolset [31]. Over a CDFG, a multiobjective optimization is carried out with a Pareto dominance technique considering area and power as metrics (see Figure 5). One year later, in [28], published by the same authors, a biobjective proposal with similar characteristics was presented, taking into account FUs that support dynamic voltage and frequency. The publications of the branch and X method are scarce, due to their ineffectiveness dealing with multiobjective problems and their high probability of being stuck in a local optimum.

3.3. Problem-Specific Heuristic Approaches. The paper [32] was the first application of a Fuzzy Inference System (FIS) to optimize this problem with a multiobjective focus. Three proposals based on the DFG are presented: a module selection scheme in the HLS using fuzzy logic, an allocation process of DFG, and scheduling of DFG with processing times characterized by fuzzy sets. Two years later, [33] presented another problem-specific heuristic based on the decomposition of an Architecture Configurations Graph (ACG). In [34], a greedy algorithm to optimize delay and area was studied. The authors analyzed in detail the estimations of the objective functions. Two years later, [35] explored power-area trade-offs in HLS through dynamic FU allocation with a network flow rebinding using a DFG
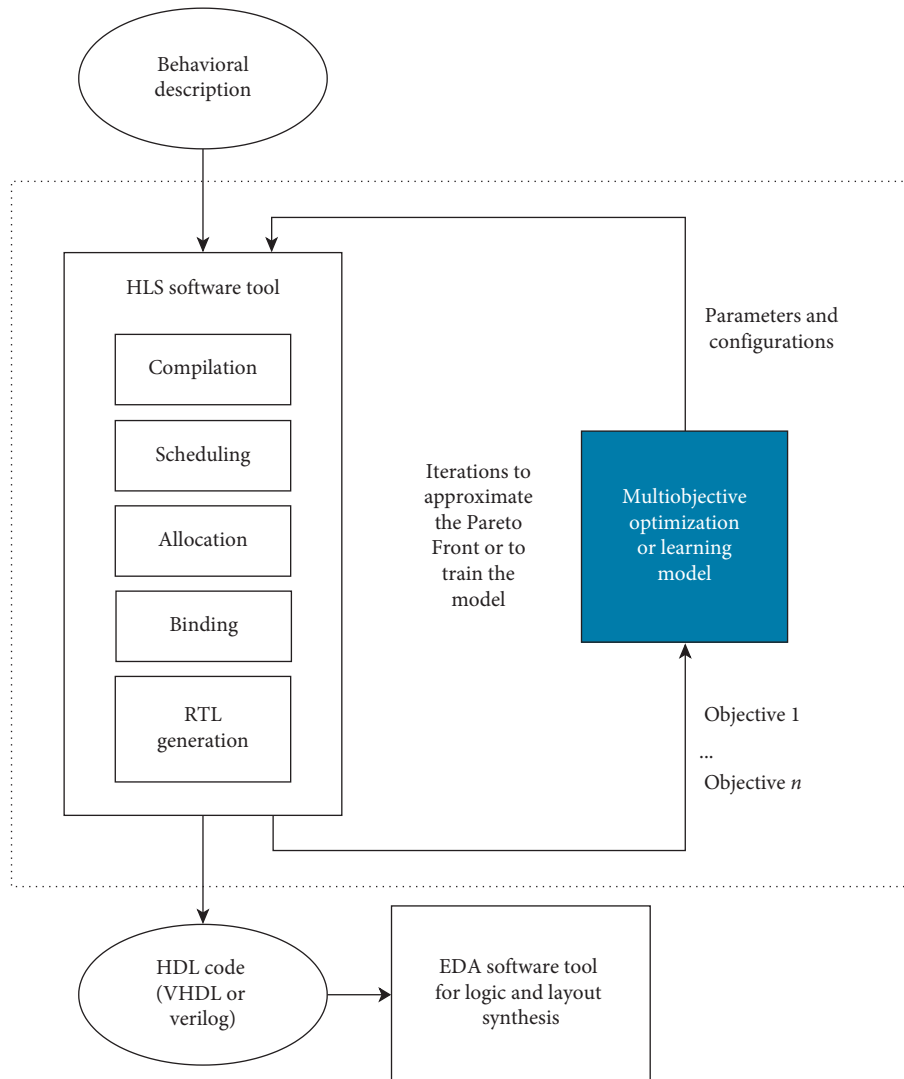
FIGURE 8: General framework of the HLS tool as a black box.

representation. In [36], a hierarchy factor method to simultaneously optimize delay, area, and power was studied. The authors in [37] studied a greedy algorithm to minimize the area, power, and digital noise as objective functions. The authors introduce an analytical precision analysis approach based on a quantization error propagation model.

Sengupta et al. presented several papers with a priority factor-based heuristic [1, 38–41]. The proposed approaches try to resolve several issues related to the DSE, such as the precision from evaluation, when the time is exhausted during the evaluation, and also automation of the exploration process. Furthermore, scheduling, allocation, and binding were tested with several DSP benchmarks and real-world problems. At the same time, [42] introduced a hybrid priority factor-based heuristic and FIS employing an aggregation method and fuzzy dominance to optimize delay, area, and power. The proposed hybrid exploration was applied to different DSP benchmarks, and these methods provide acceleration compared to some DSE approaches. A DSE by hybrid priority factor-based heuristic and FIS is presented in [43]. It is a combination of the priority factor

method and fuzzy search technique that is rapid and accurate, used in the evaluation and selection in the architecture design space. Other hybrid approaches that use an aggregation method are presented in [44], which is a combination of a priority factor-based heuristic and a dependency matrix algorithm. This iterative heuristic method has a considerably good exploration runtime while delay and area are used as the objective functions. Krishna et al. [45] proposed a different hybrid heuristic, which is a combination of a priority factor-based heuristic and greedy algorithm to optimize delay and power. This work also has a design with less execution time, providing increased acceleration when it is compared with other iterative proposals.

Another FIS was presented in [46], this time with the cost assignment strategies of fuzzy dominance. It achieves significant improvement in speedup with a real benchmark. A brute force search based on adders and multipliers is presented by [47]. The authors considered the code level transformations together with the architectural level optimizations and their impact on the scheduled data path. The same authors optimized delay and area again, but this time
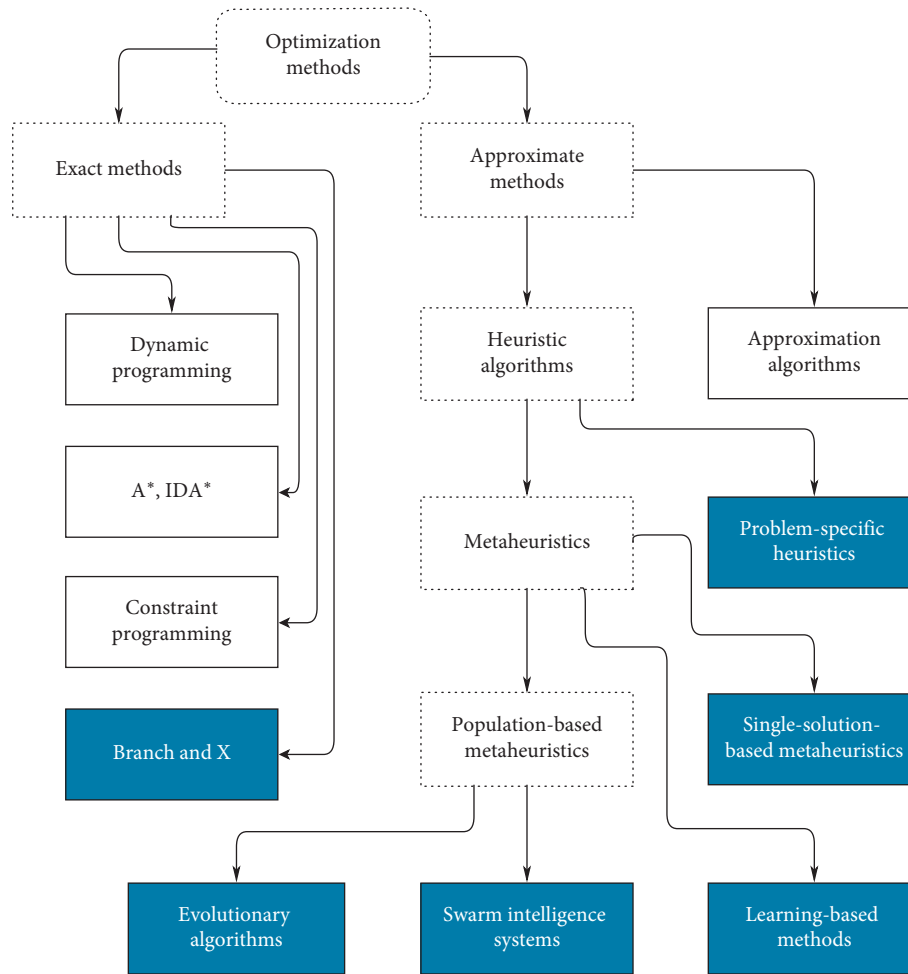
FIGURE 9: Taxonomy of multiobjecive methods in HLS for FPGA devices.

with a gradient-based heuristic pruning [48, 49]. The work [50] presents a clustering method that acts over pragma directives to optimize delay and area using a PF approximation. In [51], a schedule and binding heuristic with network flow rebinding is described. Their work employs a dynamic FU allocation strategy in HLS to achieve a compromise between power and area. References [52–54] proposed an algorithm to explore the design space using binary search employing an ACG. Alternatively, the problem of DSE was addressed in [55, 56] by a D-logic based exploration. These are mathematical models for the power, delay, and area metrics that deterministically prune the vast design space into a subset of valid design variants without compromising the speed and the quality of the design.

The HLS design requires an efficient exploration approach with the ability to determine optimal/near-optimal scheduling solutions and module selection with significant speed and precision. Based on this idea, [57] introduced a heuristic based on the primacy selector (*s*-value) metric which is common with the matrix topology methods. Most of the research has focused on using an HLS tool as a black box with pragma directives, and [58] is another example. In this case, the divide and conquer algorithm with the CHStone benchmark was used [59]. By profiling

annotations, in [60, 61], a greedy algorithm to optimize delay and area with an aggregation method is presented. This methodology is completely autonomous and it incorporates area and frequency like constraints. The work of [62] presented a fully automated C-to-FPGA framework to address this problem. This technique can satisfy hardware resource constraints (scratchpad size) while still aggressively exploiting data reuse. This approach can also be used to reduce the on-chip buffer size subject to bandwidth constraints. In [29], an iterative method with pruning that can deal with the DSE of multiple loops on FPGAs is described.

Many methodologies have been introduced which are capable of drastically reducing the number of variants to be analyzed for the selection of the optimized design using the minimal execution time. The paper [63] presented a problem-specific heuristic based on a graph merging approach to deal with delay, area, and power. Allocation and scheduling of reconfigurable arrays are implemented in Verilog HDL and synthesized by an RTL representation using the Xilinx ISE Design Suite. The graph merging approach is validated by the results which showed that the area allocated is less for the graph merging technique than the reconfigurable array using multiplexers. Concerning the objective
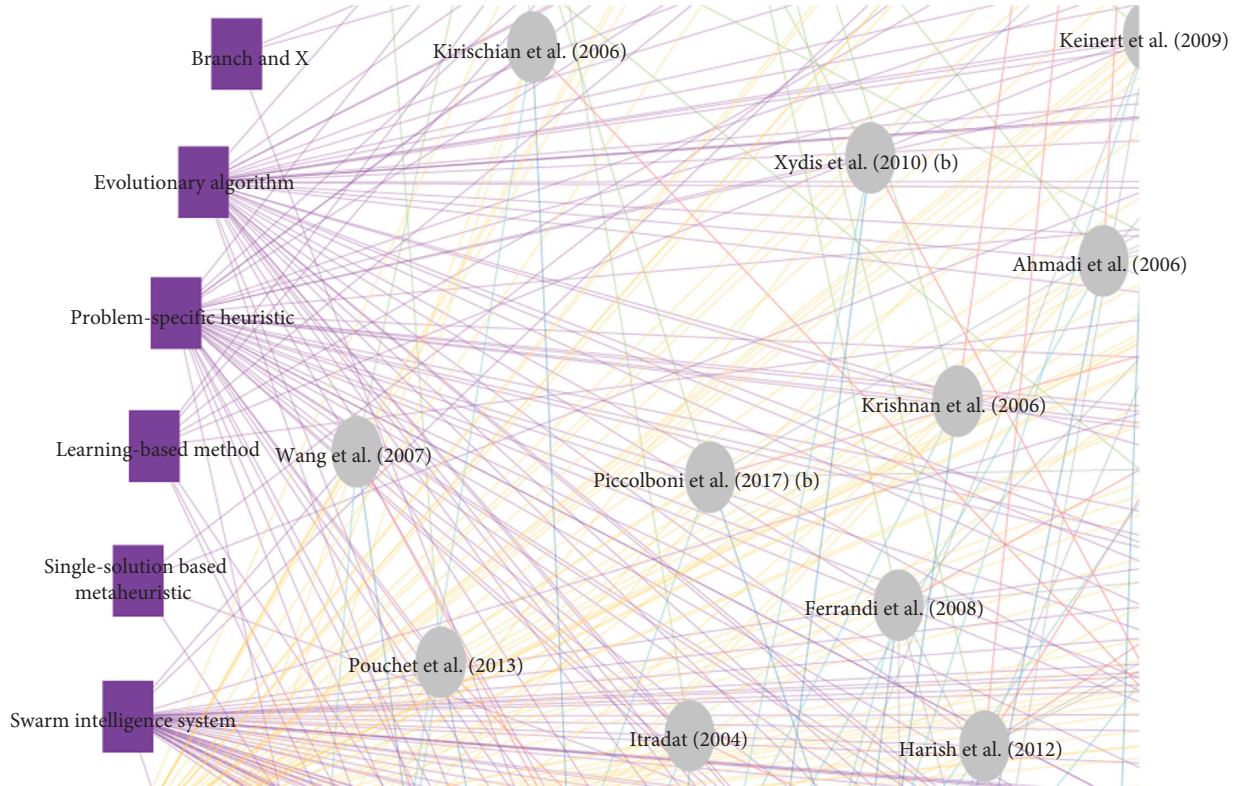
FIGURE 10: State of the art of six MOAs.

function digital noise (as well as area), [64] studied a bit-width optimization by a divide and conquer algorithm for fixed points. In [65], the authors present a hierarchical DSE method that can speed up the exploration and can also perform incremental DSE avoiding rerunning a full exploration—by an HLS tool—each time that the changes in the source are made, a Cyclic Redundancy Check- (CRC-) based method is used to detect changes at the behavioral description (source code).

Pham et al. [66] proposed a heuristic based on an access pattern simulator by a LADG to reduce the dimensions of the design space. A scheduling and binding heuristic for HLS of fault-tolerant FPGA applications is presented in [67]. The authors stated that integrating redundancy into the HLS is an attractive approach that enables synthesis to rapidly explore different trade-offs at no cost to the designer. In [68], the authors present a multiobjective optimization with quick estimates of cycle count and FPGA area usage for designs in the Delite Hardware Definition Language (DHDL). Their estimations take into account available off-chip memory bandwidth and on-chip resources for data path and routing, as well as effects from low-level optimizations like LUT packing and logic duplication. A year later, linear programming for multiobjective optimization is studied by [69, 70] and a colored interval graph approach is studied by [71, 72].

3.4. Single-Solution Based Metaheuristics.
Aggregation method consists of changing a multiobjective optimization problem into a monoobjective one or a set of such problems.

It consists of using an aggregation function combining various objective functions f_i into a single-objective function f generally in a linear way:

$$F(x) = \sum_{i=1}^{n} \lambda_i f_i(x), \quad x \in D, \tag{3}$$

where the weights $\lambda_i \in [0...1]$ and $\sum_{i=1}^{n} \lambda_i = 1$. However, the use of scalarization approaches is only justified when they generate Pareto optimal solutions [30]. Zwolinski and Gaur [15] optimized the delay and area by an aggregation method, scaling from multiobjective to monoobjective with a simple weight vector (see equation (3)). Within the next two years, another three approaches from this emerging research field were published, and one of them was [73] with single-solution based metaheuristics. On this occasion, simulated annealing, random search Pareto, and tabu search algorithms were used, with the peculiarity of selecting weak dominance as the cost assignment strategy. In [74], a similar approach employing simulated annealing was used. This time with pragma directives instead of simulation configurations. The paper [75] studied the trade-offs between power and security estimations on a CDFG. This paper considers IP protection as a new objective function of the DSE.

3.5. Learning-Based Methods.
Machine learning methods have been used in recent years, almost all of them using HLS tools as a black box (see Figure 8). These techniques always perform scheduling, allocation, and binding because HLS tools are responsible for carrying them out. In [76],
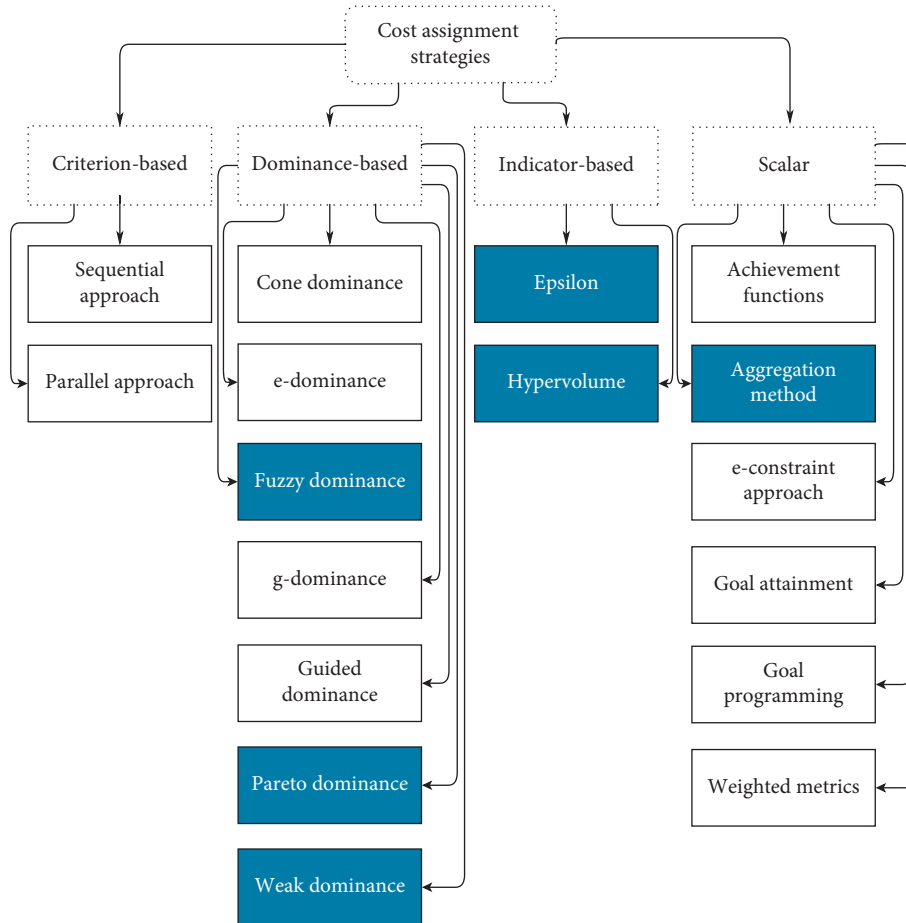
FIGURE 11: Taxonomy of cost assignment strategies in HLS for FPGA devices.

a machine learning algorithm is presented, where the authors determine the PF approximations by only sampling and synthesizing a fraction of the design space. A DSE to derive PF approximations of the design configurations for a set of targeted metrics (in this case delay and area) is developed in [49]. That work used a response surface method with Pareto dominance to perform scheduling, allocation, and binding. In the same year, [77] investigated a methodology based on random forest and results compared favorably with other black box alternatives. This research optimizes simultaneously the same objective functions (delay and area), but this time using knob settings to create variations on the search process. One year later, a machine learning approach based on simulated annealing was created for HLS of the DSE [78] using pragma directives. This approach employs a standard simulated annealer to generate a training set and uses this set to implement a decision tree. The delay and area optimization developed by [79] used an Adaptive Threshold Non-Pareto Elimination (ATNE). This approach focuses on understanding and estimating the inaccuracy, instead of focusing on regression accuracy improvement. They employed five OpenCL applications as behavioral descriptions to perform experiments.

An alternative strategy was proposed in [80], called cluster-based heuristic, an open-source project. The exploration methodology is divided into five steps: initial sampling, clustering, cluster selection, intracluster exploration, and intercluster exploration. Ma et al. [81] presented a Gaussian process regression to optimize simultaneously delay, area, and power. Machine learning is applied to predict the PF approximation of the adders in the physical domain, because it is infeasible to exhaustively run the HLS tools for many architectural solutions. On the other hand, [82] developed HyperMapper 2.0, a methodology and corresponding software framework, which handles multi-objective optimization in the DSE for FPGAs. This methodology also can incorporate prior knowledge from the user in the search. Another random forest approach is presented in [83], which focuses on hardware loop unrolling with an HLS directive.

3.6. Evolutionary Algorithms.

Evolutionary algorithms have been good candidates to tackle DSE. The first one was in [84], making use of a Weighted Sum Genetic Algorithm (WSGA). This is the first proposal where area and digital noise are objective functions. Additionally, the same authors propose an extension with a similar DFG-based methodology [85–88], but with power as an additional objective function. One of the most important contributions in this field was offered in [89], since they explained the use of a multichromosome approach, and consequently, it was

more feasible to represent the scheduling and allocation tasks concurrently.

Strength Pareto Evolutionary Algorithm 2 (SPEA2), using Pareto dominance, is an algorithm that performs a much more intelligent multiobjective search. In 2006, it was used for the first time in this problem by [90] with two objective functions and in [91] with three objective functions. Another evolutionary algorithm that uses Pareto dominance is the Nondominated Sorting Genetic Algorithm II (NSGA-II), which uses the crowding distance as a diversity preservation technique. This algorithm was used for the first time in HLS for FPGA devices in [92–94] to optimize delay and area. One year later, these same proposals were improved, in terms of the representation of the solutions (encoding) in [95–97].

In [98, 99], a dynamic combination of WSGA and Ant Colony Optimization (ACO) is presented. In this method, the initial pheromone distribution is generated with WSGA, and then ACO is used to obtain the solutions. Dynamic switching conditions are also discussed. In [100, 101], the SystemCoDesigner software tool was presented, which offers a fast DSE and rapid prototyping of behavioral SystemC models. The work [102] presents a multiobjective evolutionary algorithm for hardware-software partitioning of embedded systems, and the MediaBench benchmark [103] was selected for testing. Anderson and Khalid [104] applied the Simple Evolutionary Algorithm for Multiobjective Optimization (SEAMO), a genetic-based algorithm to prune the design space of the parametrized core and determine a PF approximation by simulation. Speeding-up expensive evaluations in HLS using solution modeling and fitness (cost) inheritance are presented in [105]. They use NSGA-II with CDFG for delay and area optimization. The works [11, 106, 107] present a different approach with respect to the previous ones. The research employs the multichromosome representation presented in [89] but incorporates an accurate power estimation. The methodology based on NSGA-II was evaluated through the MediaBench benchmark in DFG, and the results indicate that it yields improved solutions with better diversity compared to a WSGA approach.

In [108–111], the authors solved scheduling, allocation, and binding using WSGA. The presented approach incorporates a new seeding process for two special parent chromosomes as well as a load factor heuristic, which guarantees that the final solution will always be near-optimal in terms of the user-specified constraints. In [112], a fully automated design flow that exploits multiobjective DSE to enable runtime resource management is studied. They developed a technique that identifies the most promising operating points by using profiling information coming from both software simulation and hardware synthesis. The optimization is done by using the Greedy Evolutionary Multiobjective Optimization (GEMO) algorithm. Schafer and Wakabayashi [113] demonstrate the feasibility to apply NSGA-II in conjunction with a machine learning-based predictive model. It is an HLS tool based on a black box method that creates a predictive model from a training set until a given error threshold is reached. Then, it continues with the exploration using the predictive model avoiding

time-consuming synthesis and simulations of new configurations. HLS for FPGA devices by Learning Automata Genetic Algorithm (LAGA) is studied in [114]. According to this work, the scheduling and allocation are performed over a DFG, optimizing delay and area simultaneously.

In another example, [115] presented a technique for area-delay trade-off using residual load decoding heuristics with genetic algorithms for integrated DSE of scheduling and allocation. They employed the aggregation method as a cost assignment strategy. The work [116] summarizes a set of techniques that were presented in previous papers, the main one being [11]. This work explains how to deal with the simultaneous optimization of delay, area, and power. In the same year, [117] released another chromosome representation along with a driven integrated exploration of loop unrolling factor and data path by WSGA for scheduling of the CDFG. In [118], a DSE methodology for the optimization of delay and area by an evolutionary algorithm based on pragma directives is presented. One year later, [119] described another application of NSGA-II for the optimization of delay and power with the NCBI BLASTP benchmarks [120]. Other methods use a predictive model to avoid having to resynthesize each new configuration to be explored. In [121], a dedicated DSE for FPGAs is presented that is based on a pruning algorithm with an adaptive windowing method to extract the design candidates to be further (logic) synthesized after HLS. The adaptive windowing is based on a learning method inspired by the Rival Penalized Competitive Learning (RPCL) model in order to classify which designs need to be synthesized.

In [12], an approach to apply two optimizations consecutively is presented. As the first optimization, several metaheuristic algorithms for multiobjective optimization were applied in HLS based on [116]. As a second optimization, reductions of LUTs at the logic synthesis stage were carried out. The paper showed how several optimizations belonging to different design stages can coexist. One year later, as an extension, a many-objective optimization algorithm—Nondominated Sorting Genetic Algorithm III (NSGA-III)—was applied in [122] for the first time to this problem. In [123], a delay and power optimization is proposed. In this case, an SDFG is employed for modeling DSP applications. In [124], the authors focused on finding the smallest microarchitecture for a specific target latency. They used pragma directives with the S2CBench benchmark [125]. The authors of [126] incorporated a new dimension to the multiobjective optimization of this problem, the reliability. This methodology is composed of two main phases. The first one performs HLS for DSE leading to a trade-off curve of designs with delay, area, and reliability. The second phase finds the most reliable system given delay and area constraints by either implementing time or space redundancy, or a mixture of both using any combinations of microarchitectures found by the explorer.

3.7. Swarm Intelligence Systems.
This family of algorithms did not appear in this domain until 2006, when [127, 128] implemented an ACO to perform scheduling and allocation

taking into account the objective functions delay and area. A comparison between Particle Swarm Optimization (PSO) and the evolutionary algorithms NSGA-II and WSGA was made in [11]. According to their work, it is observed that, compared to WSGA, PSO shows considerable improvement in runtime with a comparable solution quality. The proposed integrated approach in [129] comprises a comprehensive mapping process and a sophisticated strategy for evaluating solutions. They introduced a PSO driven DSE methodology for delay and power trade-off over a CDFG.

An adaptive DSE framework called integrated Particle Swarm Optimization (i-PSO) for delay and power as objective functions in HLS is presented in [130], including a sensitivity analysis of the algorithm. The use of PSO for the DSE of data paths in HLS is also proposed in [131–134], and the MediaBench benchmark and another DSP benchmark (the paper does not provide details of the benchmark name) to measure the optimization quality of the simultaneous exploration of data path and loop unrolling factor are used. Other authors published a similar strategy, but delay and area were optimized in [135, 136]. The authors in [137] describe an approach to solve the DSE problem, which is based on the Bacterial Foraging Optimization Algorithm (BFOA). They also study BFOA in a similar way, where delay and power were optimized in [138–144]. The proposed exploration approach is simulated to operate in the feasible temperature range of an *Escherichia coli* bacterium in order to mimic its biological life-cycle. Mishra and Sengupta [7] studied the trade-offs between delay and power proposing MOPSE, an adaptive multiobjective PSO based on DSE. Sengupta and Mishra [145] described an approach to solve the DSE problem based on Weighted Sum Particle Swarm Optimization (WSPSO) with two variants of the acceleration coefficient, a hierarchical time-varying acceleration coefficient and a constant acceleration coefficient.

A compiler approach performing delay, area, and power optimization is presented in [146], where a better behavior of the firefly algorithm over simulated annealing (single-solution based metaheuristic) stands out. This metaheuristic has a competitive execution time, compared, for instance, with an evolutionary algorithm. Research in [136] described a methodology based on automating DSE and loop unrolling factor using the high-level transformation during area-delay trade-off using the PSO. Using CDFG, [147–155] described approaches based on the *k-cycle* transient fault secured data path during the HLS. Bhuvaneswari [116] studied a multichromosome structure on a DFG to optimize delay, area, and power using several algorithms, including swarm intelligence and evolutionary algorithms. Multiobjective optimization is performed in [9, 156], considering an interesting topic, the secure information processing against a hardware Trojan. In [157], a low-cost (delay and area) approach that relies on the PSO metaheuristic to explore the Trojan secured schedule with optimal unrolling is proposed. This paper also provides security against specific Trojans (causing a change in computational output), while the area and delay constraints are provided by the user. In [158], a low-cost optimized Trojan secured schedule at the behavioral level for single and nested loop CDFG was studied.

Other examples within the use of this type of metaheuristic can be found in [8, 159], and a multivariable signature encoding for embedding a dynamic watermark in an IP design was presented. These investigations used the same DSE framework with the PSO optimizing delay and area. The authors of [160, 161] proposed a firefly algorithm for scheduling and allocation on the DFG using the MediaBench benchmark and another DSP benchmark (in the paper, no details of the benchmark name are given). Besides, these papers report a sensitivity analysis that provides a good tuning of the algorithm control parameters for performing the DSE that leads to faster convergence.

Obfuscation is the process of transforming an original application or design into a functionally equivalent form to make the reverse engineering process significantly more complex. The authors in [162] provided a structural obfuscation methodology for protecting IP core at the HLS design stage. The proposed approach specifically targets the protection of IP cores that involve complex loops. The authors of [163, 164] created a multiobjective optimization (delay and area) that can deal with low-cost functional obfuscation of reusable IP cores. The work in [165] was the first to incorporate the switching device and the storage element delay from scheduling during the delay estimation. They provide a BFOA that gives a balanced DSE methodology and includes comprehensive delay estimation by considering the combined delay of FUs, the switching devices, and the storage elements directly from scheduling. Results indicate improvement in achieving a more realistic delay estimation process than previous approaches.

In [166], the authors presented an optimization of delay and area of the obfuscated JPEG CODEC IP core design using particle swarm based on the DSE. And [167] introduced an obfuscation of fault secured design through a hybrid transformation with delay and area objective functions by a PSO. In [168], a BFOA to achieve low-cost (delay and area) IP design is performed. And [169] studied a PSO to achieve delay and power minimization combined with an IP functional locking.

3.8. Analysis, Comparisons, and Main Findings. Figure 12 shows all the MOAs used over the years. This chart evidences, in addition to the increase of papers over the years, that swarm intelligence systems have been the most studied. Evolutionary algorithms have also been used, due to the simple way in which chromosomes can be generated.

Analyzing the cost assignment strategies in HLS, scalar approaches have been the most used. Among these strategies, the aggregation (or weighted) method has been the only one studied, due to it is simplicity. During the last decade, more methods such as dominance-based approaches and indicator-based approaches have been used.

In Figure 13, the cost assignment strategies used over the years are shown. In Figure 14, the cost assignment strategies are shown, in the proposed relational graph, where the aggregation method and Pareto dominance are the most used.
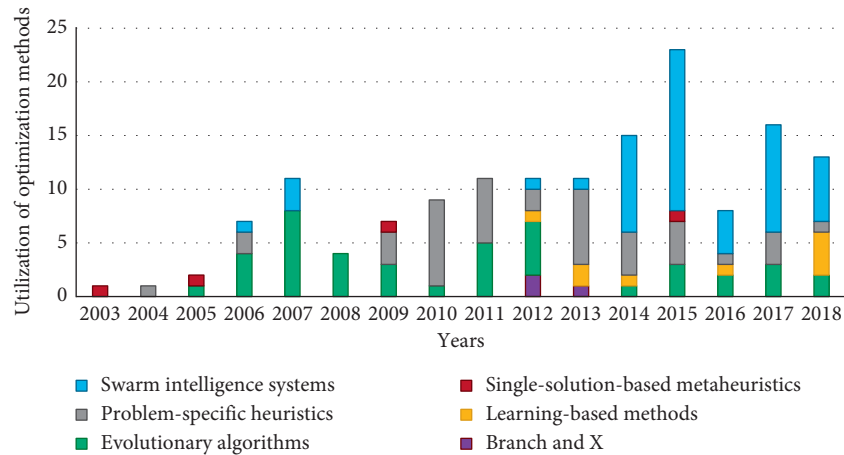
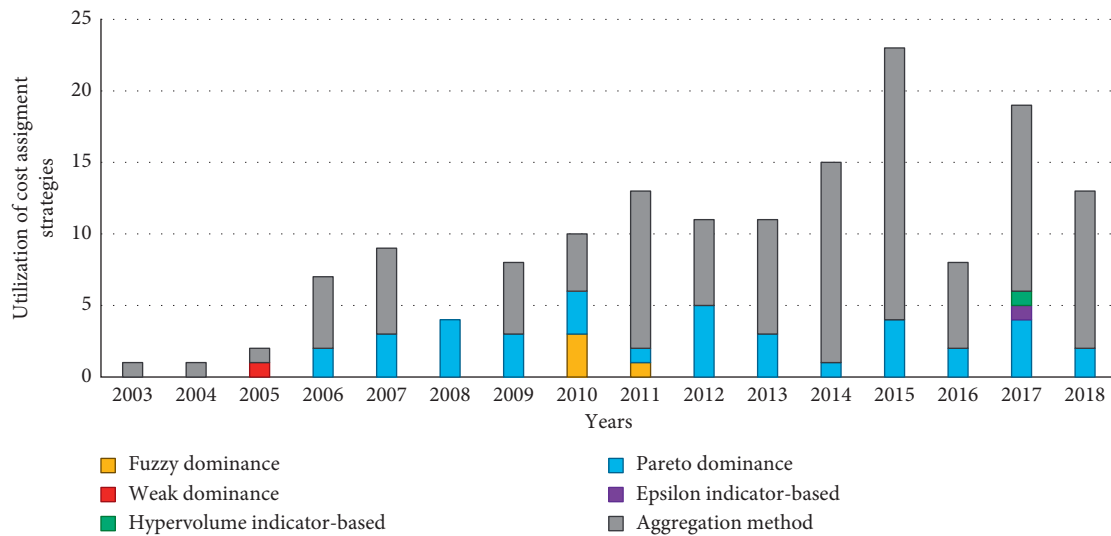Figure 12: Utilization of MOAs over the years in HLS.

Figure 13: Utilization of cost assignment strategies over the years.

Objective functions have been estimated, represented, and calculated in different ways (especially delay, area, and power, as seen in Figure 15).

The authors have proposed many ways to represent the circuit design, and therefore, the estimations have to be coupled to the data structure of the representation (for instance, chromosome representation in evolutionary algorithms). At least one of the delay, area, and power metrics is present in almost all works covered in this survey. For that reason, their estimation methods have become sophisticated over time. In Figures 16 and 17, we can see the objective functions using compiler techniques and an HLS tool as a black box; it is important to note that delay, area, and power stand out from the rest.

Furthermore, thanks to a technique called payoff matrices, [12] showed how the objective functions of delay, area, and power are in conflict. It demonstrates the importance to solve this problem with a multiobjective approach. However, until now, the optimization process has not been solved considering all eight objective functions simultaneously. The papers that used the most objective functions have been [86–88] dealing with four (considered to be a many-objectives optimization problem). In the case of the optimizations with HLS tools as a black box, the objective functions have been delay, area, power, and reliability because those are the ones that can be obtained from the software tools.

Regarding the optimization method, diversity is as important as convergence. Therefore, MOAs should have techniques for diversity preservation with statistical density estimations. In this sense, the following techniques have been applied in HLS for FPGA devices: nearest neighbor and histogram ([30]; p. 343). These techniques are implicit inside many MOAs mechanisms. For example, the NSGA-II uses the nearest neighbor technique (with crowding distance) and the NSGA-III uses the histogram (with reference points).

The benchmarks used to evaluate these techniques are very important for the experiments, comparisons, and validation of the results. In the state of the art, we can find
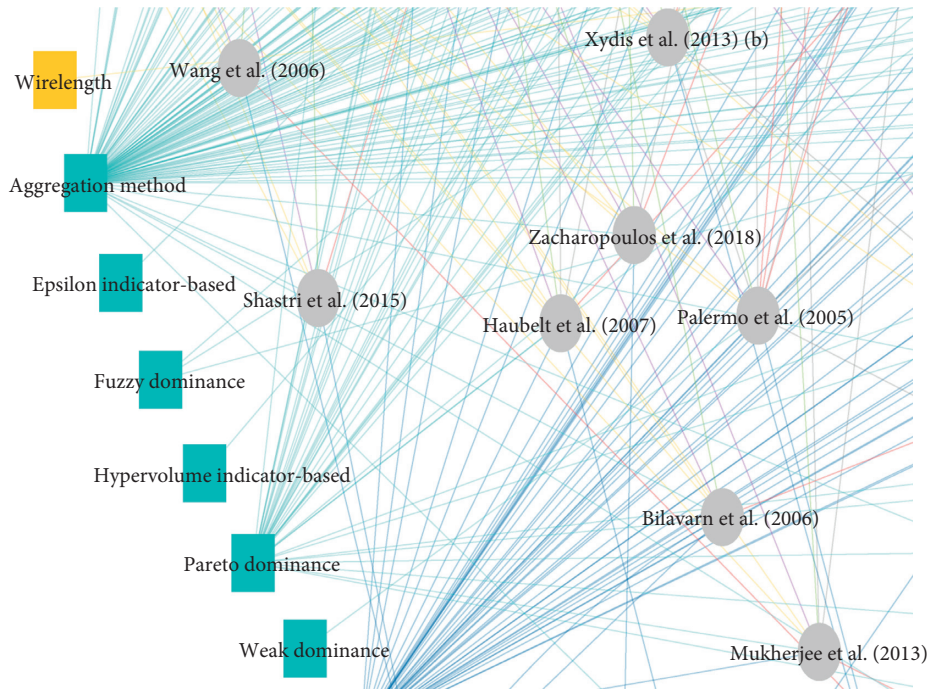
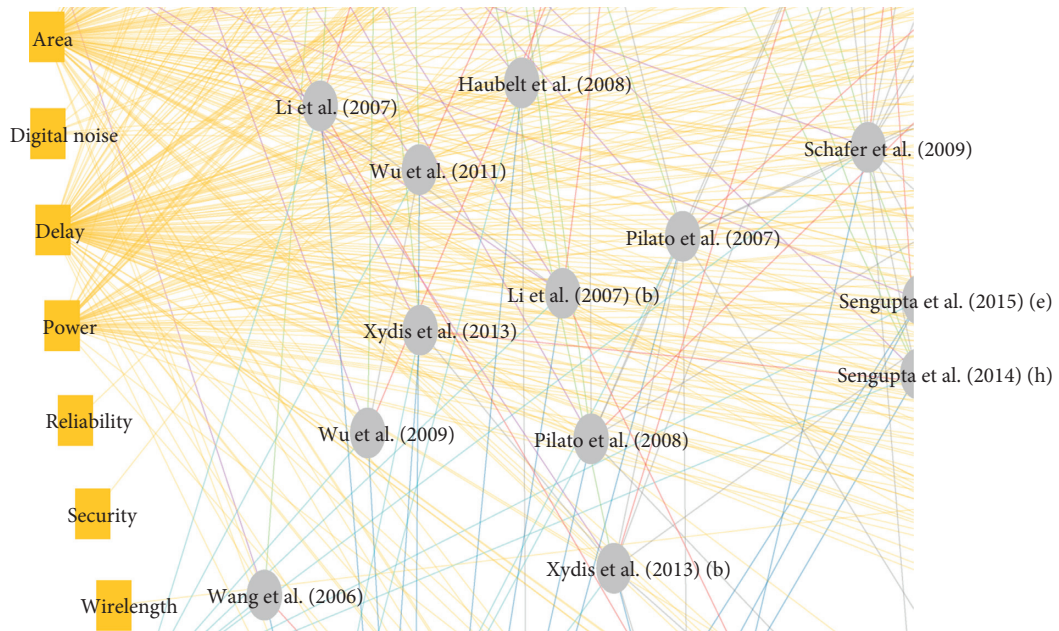FIGURE 14: Subgraph of cost assignment strategies.

FIGURE 15: : Objective functions studied in the literature.

that MediaBench, also called Express benchmark, is the most used. MediaBench was introduced in [103] for performance evaluation of solutions on microprocessor architectures applied to multimedia and communication systems. Figure 18 shows the benchmarks used in the state of the art considered in this survey.

Nonetheless, many papers have used DSP benchmarks like [170] or real-world benchmarks like [33]. The

benchmark proposed in [31] is used by the authors who proposed using the branch and X approach. The S2CBench benchmark [125] was employed for optimization proposals that use pragma directives for the search process. On the other hand, the PERFECT benchmark [171] is referenced in [69, 70] for an accelerator of Wide-Area Motion Imagery (WAMI) applications for SystemC specifications. Schafer et al. have used the [172] in [50, 74, 113] with pragma
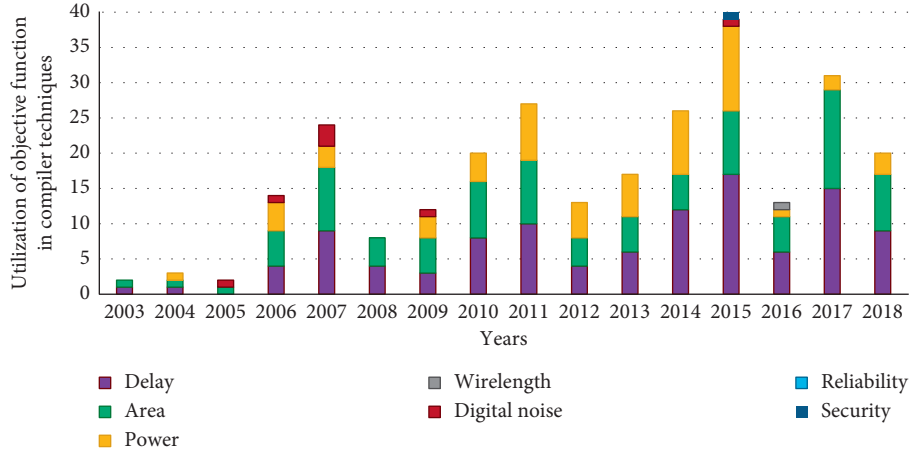
FIGURE 16: Objective functions using compiler techniques considered in the literature.
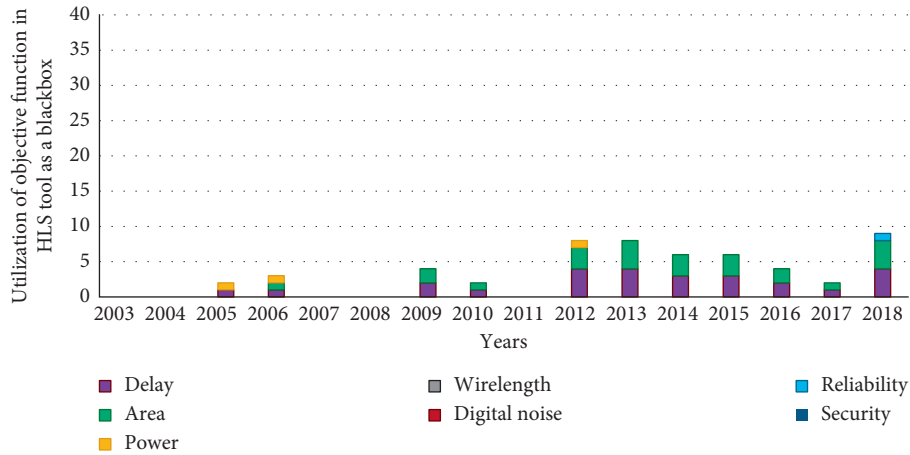

FIGURE 17: Objective functions using HLS tools as a black box considered in the literature.

directives to optimize delay and power. The experiments in [62, 66, 123] are performed for five applications from the polyhedral benchmark suite (PolyBench) [173], a benchmark for testing loop and array related problems. Another benchmark used with pragma directives and profiling annotations was CHStone, a benchmark program suite for practical C-based HLS [59]. It was used by learning-based methods and problem-specific heuristics. Other benchmarks of less use were the ACM/SIGDA benchmarks [174] in [7, 116], the Linpack benchmark [175] in [75], the NCBI BLASTP [120] in [119], the BDTI DSP [176] in [162], and the SHOC benchmark suite [177] in [83].

In addition to the benchmarks, a better way to measure the performance of the optimization method is by quality indicators. However, quality indicators have been studied in a few papers, [12, 29, 49, 58, 65, 66, 73, 77, 79–82, 91, 102, 113, 118, 121–123, 134, 145, 178]. Some quality indicators used are Average Distance from Reference Set (ADRS) in [179], Epsilon in [180], Hypervolume in [181], and R in [182]. ADRS was the most frequent quality indicator and is usually represented by percentage. It is based on the normalized distance between two PF approximations $(\overline{P}, P)$:

$$\text{ADRS}(\overline{P}, P) = \frac{1}{|\overline{P}|} \sum_{\overline{p} \in \overline{P}} \left\{ \min_{p \in P} \gamma(\overline{p}, p) \right\}, \quad (4)$$

where

$$\gamma(\overline{p}, p) = \max_{i=1,\dots,m} \left\{ 0, \frac{f_i(p) - f_i(\overline{p})}{f_i(\overline{p})} \right\}, \quad (5)$$

and m is the number of objective functions. A high value of ADRS reports a low-quality approximation, while a low one indicates that $\overline{P}$ has good approximates to P.

The main findings of these works can be summarized as follows. After reviewing and analyzing the state of the art about MOAs in HLS for FPGAs, we found only one paper that demonstrates that some of the objective functions are in conflict [12]. This is an important aspect; in the rest of the publications, the authors assume that the objective functions are in conflict. On the other hand, with the review of the state of the art, we conclude that there is no survey that allows researchers to contextualize all of the related works related to MOAs in HLS for FPGAs. This paper is intended to be of help to carry out new research in this area. In this survey, we have focused on organizing the papers according to the
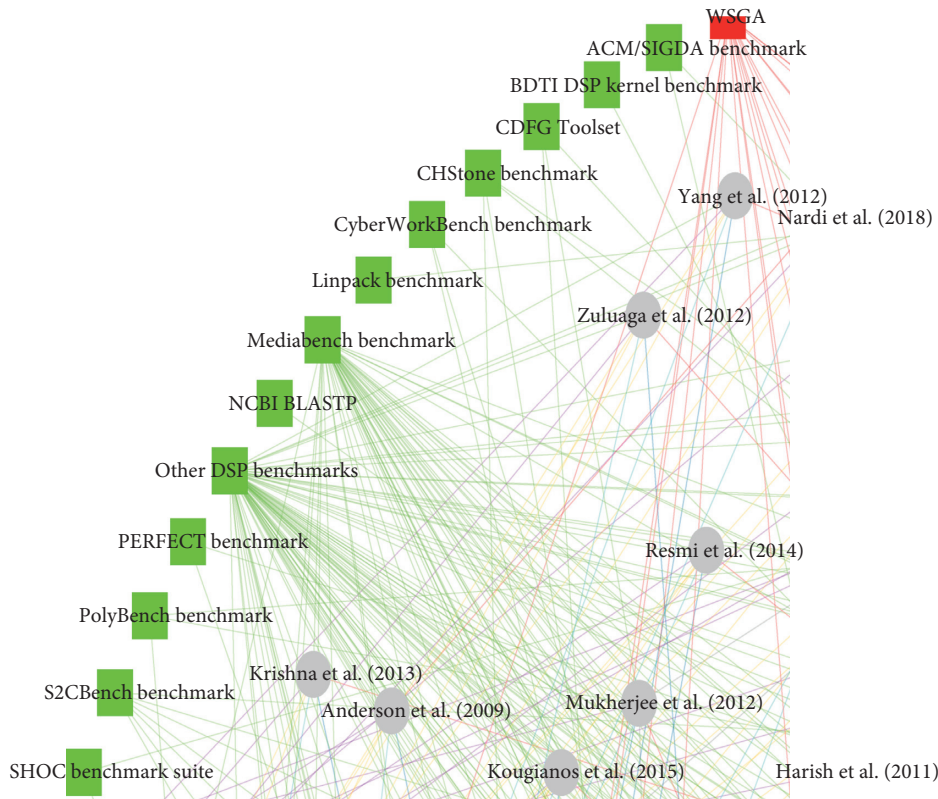
FIGURE 18: Benchmarks used in the state of the art of multiobjective HLS.

MOAs, the cost assignment strategies, the objective functions, the benchmarks, and the compiler techniques. With this analysis, we have detected that swarm intelligence systems and evolutionary algorithms are the most used. The most used graphs of the intermediate structures are DFG and CDFG. The aggregation method and Pareto dominance are the most used cost assignment strategies. Moreover, of the 8 objective functions studied, the most optimized functions are area, power, and delay. Regarding the benchmarks, MediaBench and other DSPs are the most used in these studies.

4. Open Issues

In this section, future challenges are presented. In addition, some ideas about how to approach them are mentioned:

(1) It is important to use quality indicators to measure the convergence and diversity of the PF, instead of observing the convergence of only some solutions within the PF approximation, as is done in most of the papers. The Hypervolume quality indicator is a good option because it measures the volume of the dominated space bounded from below by a reference point and it is capable of measuring convergence and diversity at the same time [183, 184].

(2) Temperature has been studied in a few papers [185, 186] with a single-objective approach. This objective function should be studied with a multi-objective approach, since the temperature is in conflict with the wire length objective function, because if the use of FUs is increased, then more interconnections will be needed.

(3) The grand challenge is the optimization problem of HLS with 8 objective functions: delay, area, power, wire length, digital noise, reliability, security, and temperature (see Figure 2). We want to push the FPGA designers and researchers to create a new representation for solutions that includes scheduling, allocation, and binding, with which all these objective functions can be estimated. Then, we verify by payoff matrices that these eight objective functions are in conflict. Later, it is necessary to use many-objective optimization algorithms like NSGA-III [187, 188] or MOEA/D [189] to solve the problem. Finally, with many-objective optimization, the results obtained can be analyzed.

(4) To develop more estimations methods for the objective functions: wire length, digital noise, reliability, security, and temperature, this is an area of opportunity where researchers can develop estimations of these metrics with the intention of increasing their potential. One possibility is to use machine learning for this task.

(5) The HLS software tools with multiobjective optimizations should show the PF approximation. Also, these tools should let the designer select the optimization method and configure the most important parameters which are most convenient to him, so the

designer can choose which solution will be implemented into the FPGA device. In [190], visualization techniques are presented which can be used to improve the HLS software tools. In this challenge, the runtime of Multiobjective Optimization Algorithms could be considered a weakness relative to modern tools, such as Vivado HLS. Therefore, to improve this point, we pose the following challenge.

(6) Since multiobjective optimizations require large execution times, it is desirable that the executions of the algorithms are performed on a web server with high-performance computing and parallelization potential, instead of the user side. This can be achieved by developing an HLS web-based software tool with a microservices-based architecture or service-oriented architecture, instead of a monolithic application [191] or using cloud computing to streamline the process.

5. Conclusions

This paper presented the state of the art of multiobjective optimization methods in HLS. An online graph was designed with the aim of creating a visual representation of this survey. In summary, an analysis of the convergence of two fields was carried out: HLS and MOAs. The optimizations methods were identified and classified, as well as internal aspects within them, such as the intermediate structures where the optimizations are performed; HLS techniques; and the benchmarks employed for experimentation. Moreover, this work also studied what cost assignment strategies have been used in the algorithms and which are the objective functions to be optimized. In addition, it was demonstrated that multiobjective HLS is a knowledge area that has been in constant growth since 2003, where a wide range of algorithms and specific details in the scheduling, allocation, and binding techniques have been addressed. To finish, we identified open issues and we mentioned some ideas about how to approach them. The main one is that this problem must be visualized as a many-objective optimization problem with eight objective functions to optimize simultaneously.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

References

[1] A. Sengupta, R. Sedaghat, and Z. Zeng, "A high level synthesis design flow with a novel approach for efficient design space exploration in case of multi-parametric optimization objective," *Microelectronics Reliability*, vol. 50, no. 3, pp. 424–437, 2010.

[2] G. D. Micheli, *Synthesis and Optimization of Digital Ciruits*, McGraw-Hill, New York, NY, USA, 1994.

[3] P. Coussy, D. D. Gajski, M. Meredith, and A. Takach, "An introduction to high-level synthesis," *IEEE Design & Test of Computers*, vol. 26, no. 4, pp. 8–17, 2009.

[4] J. Cong, B. Liu, S. Neuendorffer, J. Noguera, K. Vissers, and Z. Zhang, "High-level synthesis for FPGAs: from prototyping to deployment," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 30, no. 4, pp. 473–491, 2011.

[5] F. Winterstein, S. Bayliss, and G. A. Constantinides, "High-level synthesis of dynamic data structures: a case study using vivado HLS," in *Proceedings of the 2013 International Conference on Field-Programmable Technology (FPT)*, pp. 362–365, Kyoto, Japan, December 2013.

[6] J. Zhang and J. Li, "Improving the performance of OpenCL-based FPGA accelerator for convolutional neural network," in *Proceedings of the 2017 ACM/SIGDA International Symposium on Field-Programmable Gate Arrays—FPGA '17*, pp. 25–34, ACM, Monterey, CA, USA, February 2017.

[7] V. K. Mishra and A. Sengupta, "MO-PSE: adaptive multi-objective particle swarm optimization based design space exploration in architectural synthesis for application specific processor design," *Advances in Engineering Software*, vol. 67, pp. 111–124, 2014.

[8] A. Sengupta and S. Bhadauria, "Exploring low cost optimal watermark for reusable IP cores during high level synthesis," *IEEE Access*, vol. 4, pp. 2198–2215, 2016.

[9] A. Sengupta, S. Bhadauria, and S. P. Mohanty, "Low-cost security aware HLS methodology," *IET Computers & Digital Techniques*, vol. 11, no. 2, pp. 68–79, 2017.

[10] C. Coello, G. Lamont, and D. A. V. Veldhuizen, "Evolutionary algorithms for solving multi-objective problems," *Genetic and Evolutionary Computation Series*, Springer US, The Salmon Tower Building, NY, USA, 2007.

[11] D. S. Harish Ram, M. C. Bhuvaneswari, and S. S. Prabhu, "A novel framework for applying multiobjective GA and PSO based approaches for simultaneous area, delay, and power optimization in high level synthesis of datapaths," *VLSI Design*, vol. 2012, pp. 1–12, 2012.

[12] D. R. F. de Bulnes, J. C. D. Simental, Y. Maldonado, and L. Trujillo, "High-level synthesis through metaheuristics and LUTs optimization in FPGA devices," *AI Communications*, vol. 30, no. 2, pp. 151–168, 2017.

[13] A. Sengupta, "Design flow of a digital IC: the role of digital IC\/SOC design in CE products," *IEEE Consumer Electronics Magazine*, vol. 5, no. 2, pp. 58–62, 2016.

[14] T. D. Friedman and S.-C. Yang, "Methods used in an automatic logic design generator (ALERT)," *IEEE Transactions on Computers*, vol. C-18, no. 7, pp. 593–614, 1969.

[15] M. Zwolinski and M. S. Gaur, "Integrating testability with design space exploration," *Microelectronics Reliability*, vol. 43, no. 5, pp. 685–693, 2003.

[16] R. K. Namballa, "CHESS: a tool for CDFG extraction and high-level synthesis of VLSI systems," Master thesis, University of South Florida, Tampa, FL, USA, 2003.

[17] M. Gries, "Methods for evaluating and covering the design space during early design development," *Integration, the VLSI Journal*, vol. 38, no. 2, pp. 131–183, 2004.

[18] P. Coussy and A. Morawiec, *High-Level Synthesis: From Algorithm to Digital Circuit*, Springer, Berlin, Germany, 2008.

[19] A. Punitha and M. Joseph, "Survey of memory, power and temperature optimization techniques in high level synthesis," *International Journal of Recent Trends in Engineering*, vol. 2, no. 8, pp. 22–26, 2009.

[20] S. M. Logesh, R. H. Ram, S. Dasanpotty, and M. C. Bhuvaneswari, "A survey of high-level synthesis techniques for area, delay and power optimization," *International Journal of Computer Applications*, vol. 32, no. 10, pp. 1–6, 2011.

[21] J. Lanchares, O. Garnica, F. Fernández-de-Vega, and J. Ignacio Hidalgo, "A review of bioinspired computer-aided design tools for hardware design," *Concurrency and Computation: Practice and Experience*, vol. 25, no. 8, pp. 1015–1036, 2013.

[22] R. Nane, V.-M. Sima, C. Pilato et al., "A survey and evaluation of FPGA high-level synthesis tools," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 35, no. 10, pp. 1591–1604, 2016.

[23] R. Shathanaa and N. Ramasubramanian, *Design Space Exploration for Architectural Synthesis—A Survey*, Springer, Singapore, 2018.

[24] D. Reyes and Y. Maldonado, "Evolution of multi-objective HLS as graph," 2019, http://201.174.122.25/moo_hls_fpga/.

[25] Cytoscape (library), 2019, http://js.cytoscape.org.

[26] R. Mukherjee, P. Ghosh, N. Kumar, P. Dasgupta, and A. Pal, "Multi-objective low-power CDFG scheduling using fine-grained DVS architecture in distributed framework," in *Proceedings of the 2012 International Symposium on Electronic System Design (ISED)*, pp. 267–271, Kolkata, India, December 2012.

[27] R. Mukherjee, P. Ghosh, and A. Pal, "HotSpot minimization using fine-grained DVS architecture at 90 nm technology," in *Proceedings of the 2012 Asia Pacific Conference on Postgraduate Research in Microelectronics and Electronics*, pp. 13–18, Hyderabad, India, December 2012.

[28] R. Mukherjee, P. Ghosh, P. Dasgupta, and A. Pal, "A multi-objective perspective for operator scheduling using fine-grained dvs architecture," *International Journal of VLSI Design & Communication Systems (VLSICS)*, vol. 4, no. 1, 2013.

[29] G. Zhong, V. Venkataramani, Y. Liang, T. Mitra, and S. Niar, "Design space exploration of multiple loops on FPGAs using high level synthesis," in *Proceedings of the 2014 IEEE 32nd International Conference on Computer Design (ICCD)*, pp. 456–463, Seoul, South Korea, October 2014.

[30] E.-G. Talbi, *Metaheuristics: From Design to Implementation*, John Wiley & Sons, Hoboken, NJ, USA, 2009.

[31] J. Jeon, Y. Ahn, and K. Choi, "Control data flow graph toolset," Technical report, 2002, http://dal.snu.ac.kr/software/cdfg.

[32] A. H. Itradat, "A fuzzy logic based approach for high-level synthesis of DSP data-flow graphs onto multiprocessor systems," Master thesis of Concordia University, Canada, 2004.

[33] L. Kirischian, V. Geurkov, V. Kirischian, and I. Terterian, "Multi-parametric optimisation of the modular computer architecture," *International Journal of Technology, Policy and Management*, vol. 6, no. 3, pp. 327–346, 2006.

[34] S. Bilavarn, G. Gogniat, J.-L. Philippe, and L. Bossuet, "Design space pruning through early estimations of area/delay tradeoffs for FPGA implementations," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 25, no. 10, pp. 1950–1968, 2006.

[35] F. Wu, N. Xu, J. Yu, F. Zheng, and J. Bian, "Exploiting power-area tradeoffs in high-level synthesis through dynamic functional unit allocation," in *Proceedings of the 2009 International Conference on Communications, Circuits and Systems*, pp. 1092–1096, Milpitas, CA, USA, July 2009.

[36] Z. Zeng, R. Sedaghat, and A. Sengupta, "A novel framework of optimizing modular computing architecture for multi objective VLSI designs," in *Proceedings of the 2009 International Conference on Microelectronics—ICM*, pp. 328–331, Marrakech, Morocco, December 2009.

[37] L. Zhang, Y. Zhang, and W. Zhou, "Fast trade-off evaluation for digital signal processing systems during wordlength optimization," in *Proceedings of the 2009 International Conference on Computer-Aided Design—ICCAD '09*, pp. 731–738, IEEE, San Jose, CA, USA, November 2009.

[38] A. Sengupta, R. Sedaghat, and Z. Zeng, "Rapid design space exploration for multi parametric optimization of VLSI designs," in *Proceedings of the 2010 IEEE International Symposium on Circuits and Systems*, pp. 3164–3167, Paris, France, May 2010.

[39] A. Sengupta and R. Sedaghat, "Accelerated exploration of cost-performance tradeoffs for multi objective VLSI designs," in *Proceedings of the 2010 International Conference on Microelectronics*, pp. 100–103, Cairo, Egypt, December 2010.

[40] P. Sarkar, R. Sedaghat, and A. Sengupta, "Power gradient based design space exploration in high level synthesis for DSP kernels," in *Proceedings of the International Conference on Management (ICM 2011)*, pp. 1–6, Hammamet, Tunisia, December 2011.

[41] A. Sengupta, "A methodology for self correction scheme based fast multi criterion exploration and architectual synthesis of data dominated applications," in *Proceedings of the 2013 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, Mysore, India, August 2013.

[42] A. Sengupta and R. Sedaghat, "Rapid exploration of power-delay tradeoffs using hybrid priority factor and fuzzy search," in *Proceedings of the 2010 International Conference on Microelectronics*, pp. 355–358, Cairo, Egypt, December 2010.

[43] A. Sengupta, R. Sedaghat, and Z. Zeng, "Rapid design space exploration by hybrid fuzzy search approach for optimal architecture determination of multi objective computing systems," *Microelectronics Reliability*, vol. 51, no. 2, pp. 502–512, 2011.

[44] A. Sengupta, R. Sedaghat, P. Sarkar, and S. Sehgal, "Priority function based power efficient rapid design space exploration of scheduling and module selection in high level synthesis," in *Proceedings of the 2011 24th Canadian Conference on Electrical and Computer Engineering (CCECE)*, pp. 538–543, Niagara Falls, ON, Canada, May 2011.

[45] T. Krishna, P. Sachdeva, S. Dhanuka, M. Gagrani, and P. Sarkar, "A multi parametric optimization based novel approach for an efficient design space exploration for ASIC design," in *Proceedings of the 2013 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, pp. 868–873, Mysore, India, August 2013.

[46] Z. Zeng, R. Sedaghat, and A. Sengupta, "A framework for fast design space exploration using fuzzy search for VLSI computing architectures," in *Proceedings of the 2010 IEEE*

International Symposium on Circuits and Systems, pp. 3176–3179, Toronto, Canada, May 2010.

[47] S. Xydis, C. Skouroumouni, K. Pekmestzi, D. Soudris, and G. Economakos, "Designing efficient DSP datapaths through compiler-in-the-loop exploration methodology," in *Proceedings of the 2010 IEEE International Symposium on Circuits and Systems*, pp. 2598–2601, Paris, France, May 2010.

[48] S. Xydis, C. Skouroumouni, K. Pekmestzi, D. Soudris, and G. Economakos, "Efficient high level synthesis exploration methodology combining exhaustive and gradient-based pruned searching," in *Proceedings of the 2010 IEEE Computer Society Annual Symposium on VLSI*, pp. 104–109, Lixouri, Kefalonia, Greece, July 2010.

[49] S. Xydis, K. Pekmestzi, D. Soudris, and G. Economakos, "Compiler-in-the-loop exploration during datapath synthesis for higher quality delay-area trade-offs," *ACM Transactions on Design Automation of Electronic Systems*, vol. 18, no. 1, pp. 1–35, 2012.

[50] B. C. Schafer and K. Wakabayashi, "Design space exploration acceleration through operation clustering," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 29, no. 1, pp. 153–157, 2010.

[51] F. Wu and N. Xu, "A new dynamic functional unit allocation strategy in high-level synthesis to achieve power-area trade-offs," *Journal of Circuits, Systems and Computers*, vol. 20, no. 5, pp. 915–925, 2011.

[52] A. Sengupta, R. Sedaghat, and Z. Zeng, "Multi-objective efficient design space exploration and architectural synthesis of an application specific processor (ASP)," *Microprocessors and Microsystems*, vol. 35, no. 4, pp. 392–404, 2011.

[53] A. Sengupta and R. Sedaghat, "A high-level synthesis design flow from ESL to rtl with multi-parametric optimization objective," *IETE Journal of Research, Taylor & Francis*, vol. 57, no. 2, pp. 165–171, 2011.

[54] R. Sedaghat and A. Sengupta, "Rapid exploration of cost-performance tradeoffs using dominance effect during design of hardware accelerators," *Facta Universitatis—Series: Electronics and Energetics*, vol. 27, no. 3, pp. 317–328, 2014.

[55] A. Sengupta, V. K. Mishra, and P. Sarkar, "Rapid search of Pareto fronts using D-logic exploration during multi-objective tradeoff of computation intensive applications," in *Proceedings of the Fifth Asia Symposium on Quality Electronic Design (ASQED 2013)*, pp. 113–122, Penang, Malaysia, August 2013.

[56] A. Sengupta and V. K. Mishra, "D-logic exploration: rapid search of Pareto fronts during architectural synthesis of custom processors," in *Proceedings of the 2013 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, pp. 586–593, Mysore, India, August 2013.

[57] A. Sengupta, R. Sedaghat, and P. Sarkar, "Rapid exploration of integrated scheduling and module selection in high level synthesis for application specific processor design," *Microprocessors and Microsystems*, vol. 36, no. 4, pp. 303–314, 2012.

[58] B. C. Schafer and K. Wakabayashi, "Divide and conquer high-level synthesis design space exploration," *ACM Transactions on Design Automation of Electronic Systems*, vol. 17, no. 3, pp. 1–19, 2012.

[59] Y. Hara, H. Tomiyama, S. Honda, H. Takada, and K. Ishii, "CHStone: a benchmark program suite for practical C-based high-level synthesis," in *Proceedings of the 2008 IEEE International Symposium on Circuits and Systems*, Seattle, WA, USA, May 2008.

[60] A. Prost-Boucle, O. Muller, and F. Rousseau, "A fast and autonomous HLS methodology for hardware accelerator generation under resource constraints," in *Proceedings of the 2013 Euromicro Conference on Digital System Design*, pp. 201–208, Los Alamitos, CA, USA, September 2013.

[61] A. Prost-Boucle, O. Muller, and F. Rousseau, "Fast and standalone design space exploration for high-level synthesis under resource constraints," *Journal of Systems Architecture*, vol. 60, no. 1, pp. 79–93, 2014.

[62] L.-N. Pouchet, P. Zhang, P. Sadayappan, and J. Cong, "Polyhedral-based data reuse optimization for configurable computing," in *Proceedings of the ACM/SIGDA International Symposium on Field Programmable Gate Arrays—FPGA '13*, Monterey, CA, USA, February 2013.

[63] R. Resmi and B. B. Tripura Sundari, "Allocation of optimal reconfigurable array using graph merging technique," in *Proceedings of the 2014 International Conference on Embedded Systems (ICES)*, pp. 49–54, Coimbatore, India, July 2014.

[64] J. Chung and L.-W. Kim, "Bit-width optimization by divide-and-conquer for fixed-point digital signal processing systems," *IEEE Transactions on Computers*, vol. 64, no. 11, pp. 3091–3101, 2015.

[65] B. C. Schafer, "Hierarchical high-level synthesis design space exploration with incremental exploration support," *IEEE Embedded Systems Letters*, vol. 7, no. 2, pp. 51–54, 2015.

[66] N. K. Pham, A. K. Singh, A. Kumar, and M. M. Khin, "Exploiting loop-array dependencies to accelerate the design space exploration with high level synthesis," in *Proceedings of the Design, Automation & Test in Europe Conference & Exhibition (DATE), 2015*, pp. 157–162, EDA Consortium, San Jose, CA, USA, March 2015.

[67] A. Shastri, G. Stitt, and E. Riccio, "A scheduling and binding heuristic for high-level synthesis of fault-tolerant FPGA applications," in *Proceedings of the 2015 IEEE 26th International Conference on Application-specific Systems, Architectures and Processors (ASAP)*, pp. 202–209, Toronto, ON, Canada, July 2015.

[68] D. Koeplinger, R. Prabhakar, Y. Zhang, C. Delimitrou, C. Kozyrakis, and K. Olukotun, "Automatic generation of efficient accelerators for reconfigurable hardware," in *Proceedings of the 2016 ACM/IEEE 43rd Annual International Symposium on Computer Architecture (ISCA)*, pp. 115–127, Seoul, South Korea, June 2016.

[69] L. Piccolboni, P. Mantovani, G. D. Guglielmo, and L. P. Carloni, "COSMOS: coordination of high-level synthesis and memory optimization for hardware accelerators," *ACM Transactions on Design Automation of Electronic Systems*, vol. 16, no. 5s, pp. 1–22, 2017.

[70] L. Piccolboni, P. Mantovani, G. Guglielmo, and L. P. Carloni, "Broadening the exploration of the accelerator design space in embedded scalable platforms," in *Proceedings of the 2017 IEEE High Performance Extreme Computing Conference (HPEC)*, pp. 1–7, Boston, MA, USA, September 2017.

[71] D. Roy and A. Sengupta, "Low overhead symmetrical protection of reusable IP core using robust fingerprinting and watermarking during high level synthesis," *Future Generation Computer Systems*, vol. 71, pp. 89–101, 2017.

[72] A. Sengupta, D. Roy, and S. P. Mohanty, "Triple-phase watermarking for reusable IP core protection during architecture synthesis," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 37, no. 4, pp. 742–755, 2018.

[73] G. Palermo, C. Silvano, and V. Zaccaria, "Multi-objective design space exploration of embedded systems," *Journal of Embedded Computing—Low-Power Embedded Systems*, vol. 1, no. 3, pp. 305–316, 2005.

[74] B. C. Schafer, T. Takenaka, and K. Wakabayashi, "Adaptive simulated annealer for high level synthesis design space exploration," in *Proceedings of the 2009 International Symposium on VLSI Design, Automation and Test*, pp. 106–109, Hsinchu, Taiwan, April 2009.

[75] Q. Liu and H. Li, "Hardware design space exploration with a new dimension—IP protection robustness," in *Proceedings of the 2015 Euromicro Conference on Digital System Design*, pp. 599–605, Funchal, Portugal, August 2015.

[76] M. Zuluaga, A. Krause, P. Milder, and M. Püschel, ""Smart" design space sampling to predict Pareto-optimal solutions," in *Proceedings of the 13th ACM SIGPLAN/SIGBED International Conference on Languages, Compilers, Tools and Theory for Embedded Systems—LCTES '12*, vol. 47, no. 5, pp. 119–128, Beijing, China, June 2012.

[77] H.-Y. Liu and L. P. Carloni, "On learning-based methods for design-space exploration with high-level synthesis," in *Proceedings of the 50th Annual Design Automation Conference on—DAC '13*, pp. 1–7, Austin, TX, USA, May 2013.

[78] A. Mahapatra and B. C. Schafer, "Machine-learning based simulated annealer method for high level synthesis design space exploration," in *Proceedings of the 2014 Electronic System Level Synthesis Conference (ESLsyn)*, pp. 1–6, San Francisco, CA, USA, May 2014.

[79] P. Meng, A. Althoff, Q. Gautier, and R. Kastner, "Adaptive threshold non-Pareto elimination: re-thinking machine learning for system level design space exploration on FPGAs," in *Proceedings of the 2016 Design, Automation & Test in Europe Conference & Exhibition (DATE)*, pp. 918–923, EDA Consortium, Dresden, Germany, March 2016.

[80] L. Ferretti, G. Ansaloni, and L. Pozzi, "Cluster-based heuristic for high level synthesis design space exploration," *IEEE Transactions on Emerging Topics in Computing*, pp. 1–9, 2019.

[81] Y. Ma, S. Roy, J. Miao, J. Chen, and B. Yu, "Cross-layer optimization for high speed adders: a Pareto driven machine learning approach," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 38, no. 12, pp. 2298–2311, 2018.

[82] L. Nardi, D. Koeplinger, and K. Olukotun, "Practical design space exploration," 2018, https://arxiv.org/abs/1810.05236.

[83] G. Zacharopoulos, A. Barbon, G. Ansaloni, and L. Pozzi, "Machine learning approach for loop unrolling factor prediction in high level synthesis," in *Proceedings of the 2018 International Conference on High Performance Computing & Simulation (HPCS)*, pp. 91–97, Orleans, France, July 2018.

[84] A. Ahmadi and M. Zwolinski, "Area word-length trade off in DSP algorithm implementation and optimization," *IET Conference Proceedings*, vol. 1, pp. 1–16, 2005.

[85] A. Ahmadi and M. Zwolinski, "Word-length oriented multiobjective optimization of area and power consumption in DSP algorithm implementation," in *Proceedings of the 25th International Conference on Microelectronics*, pp. 614–617, Belgrade, Serbia, May 2006.

[86] A. Ahmadi and M. Zwolinski, "Multiple-width bus partitioning approach to datapath synthesis," in *Proceedings of the 2007 IEEE International Symposium on Circuits and Systems*, pp. 2994–2997, New Orleans, LA, USA, May 2007.

[87] A. Ahmadi and M. Zwolinski, "A symbolic noise analysis approach to word-length optimization in DSP hardware," in *Proceedings of the 2007 International Symposium on Integrated Circuits*, pp. 457–460, Singapore, September 2007.

[88] A. Ahmadi and M. Zwolinski, MW^2P-*Bus: A New Bus Structure for Datapath Synthesis*, University of Southampton, Southampton, UK, 2007.

[89] V. Krishnan and S. Katkoori, "A genetic algorithm for the design space exploration of datapaths during high-level synthesis," *IEEE Transactions on Evolutionary Computation*, vol. 10, no. 3, pp. 213–229, 2006.

[90] J. Madsen, T. Stidsen, P. Kjaerulf, and S. Mahadevan, "Multi-objective design space exploration of embedded system platforms," *International Federation for Information Processing Digital Library: From Model-Driven Design to Resource Management for Distributed Embedded Systems*, Springer, vol. 225, pp. 185–194, , Berlin, Germany, 2006.

[91] T. Schlichter, M. Lukasiewycz, C. Haubelt, and J. Teich, "Improving system level design space exploration by incorporating SAT-solvers into multi-objective evolutionary algorithms," in *Proceedings of the IEEE Computer Society Annual Symposium on Emerging VLSI Technologies and Architectures (ISVLSI'06)*, Karlsruhe, Germany, March 2006.

[92] C. Haubelt, J. Falk, J. Keinert et al., "A systemC-based design methodology for digital signal processing systems," *EURASIP Journal on Embedded Systems*, vol. 2007, no. 1, Article ID 047580, 2007.

[93] C. Pilato, G. Palermo, A. Tumeo, F. Ferrandi, D. Sciuto, and P. L. Lanzi, "Fitness inheritance in evolutionary and multi-objective high-level synthesis," in *Proceedings of the 2007 IEEE Congress on Evolutionary Computation*, pp. 3459–3466, Singapore, September 2007.

[94] F. Ferrandi, P. L. Lanzi, G. Palermo, C. Pilato, D. Sciuto, and A. Tumeo, "An evolutionary approach to area-time optimization of FPGA designs," in *Proceedings of the 2007 International Conference on Embedded Computer Systems: Architectures, Modeling and Simulation*, pp. 145–152, Samos, Greece, July 2007.

[95] F. Ferrandi, P. L. Lanzi, D. Loiacono, C. Pilato, and D. Sciuto, "A multi-objective genetic algorithm for design space exploration in high-level synthesis," in *Proceedings of the 2008 IEEE Computer Society Annual Symposium on VLSI*, pp. 417–422, Montpellier, France, April 2008.

[96] C. Pilato, D. Loiacono, F. Ferrandi, P. Lanzi, and D. Sciuto, "High-level synthesis with multi-objective genetic algorithm: a comparative encoding analysis," in *Proceedings of the 2008 IEEE Congress on Evolutionary Computation (IEEE World Congress on Computational Intelligence)*, pp. 3334–3341, Hong Kong, June 2008.

[97] C. Pilato, A. Tumeo, G. Palermo, F. Ferrandi, P. L. Lanzi, and D. Sciuto, "Improving evolutionary exploration to area-time optimization of FPGA designs," *Journal of Systems Architecture*, vol. 54, no. 11, pp. 1046–1057, 2008.

[98] G. Li, J. Wu, G. Wang, H. Yu, and G. Ma, "A new resource constrained scheduling method based on dynamic combination of genetic algorithm and ant algorithm," in *Proceedings of the 2007 7th International Conference on ASIC*, pp. 1182–1185, Guilin, China, October 2007.

[99] G. Li, J. Wu, B. Huang, and G. Ma, "A time constrained scheduling method based on dynamic combination of genetic algorithm and ant algorithm," in *Proceedings of the 2007 Internatonal Conference on Microelectronics*, pp. 119–122, Cairo, Egypt, December 2007.

[100] C. Haubelt, T. Schlichter, J. Keinert, and M. Meredith, "SystemCoDesigner: automatic design space exploration and rapid prototyping from behavioral models," in *Proceedings of*

the 45th annual conference on Design automation—DAC '08, pp. 580–585, Anaheim, CA, USA, June 2008.

[101] J. Keinert, M. Streubūhr, T. Schlichter et al., "System-CoDesigner-an automatic esl synthesis approach by design space exploration and behavioral synthesis for streaming applications," *ACM Transactions on Design Automation of Electronic Systems*, vol. 14, no. 1, pp. 1–23, 2009.

[102] M. Jagadeeswari and M. C. Bhuvaneswari, "Efficient multi-objective genetic algorithm for hardware-software partitioning in embedded system design: ENGA," *International Journal of Computer Applications in Technology*, vol. 36, no. 3-4, pp. 181–190, 2009.

[103] C. Lee, M. Potkonjak, and W. Mangione-Smith, "Mediabench: a tool for evaluating and synthesizing multimedia and communications systems," in *Proceedings of the 30th Annual International Symposium on Microarchitecture*, pp. 330–335, North Carolina, USA, December 1997.

[104] I. D. L. Anderson and M. A. S. Khalid, "SC build: a computer-aided design tool for design space exploration of embedded central processing unit cores for field-programmable gate arrays," *IET Computers & Digital Techniques*, vol. 3, no. 1, pp. 24–32, 2009.

[105] C. Pilato, D. Loiacono, A. Tumeo, F. Ferrandi, P. L. Lanzi, and D. Sciuto, "Speeding-up expensive evaluations in high-level synthesis using solution modeling and fitness inheritance," *Computational Intelligence in Expensive Optimization Problems*, Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 701–723, 2010.

[106] S. Logesh, R. D. Harish, and M. C. Bhuvaneswari, "Multi-objective optimization of power, area and delay during high-level synthesis of DFG's—a genetic algorithm approach," in *Proceedings of the 2011 3rd International Conference on Electronics Computer Technology*, pp. 108–112, Kanyakumari, India, April 2011.

[107] S. D. Ram Harish, M. C. Bhuvaneswari, and S. Logesh, "A novel evolutionary technique for multi-objective power, area and delay optimization in high level synthesis of datapaths," in *Proceedings of the 2011 IEEE Computer Society Annual Symposium on VLSI*, pp. 290–295, Chennai, India, July 2011.

[108] A. Sengupta and R. Sedaghat, "Integrated scheduling, allocation and binding in High Level Synthesis using multi structure genetic algorithm based design space exploration," in *Proceedings of the 2011 12th International Symposium on Quality Electronic Design*, pp. 1–9, Santa Clara, CA, USA, March 2011.

[109] A. Sengupta, R. Sedaghat, and P. Sarkar, "Integrated design space exploration based on power-performance trade-off using genetic algorithm," in *Proceedings of the International Conference on Advances in Computing and Artificial Intelligence—ACAI '11*, pp. 77–81, ACM, Rajpura, Punjab, India, July 2011.

[110] A. Sengupta, R. Sedaghat, P. Sarkar, and S. Sehgal, "Integrated scheduling, allocation and binding in high level synthesis for performance-area tradeoff of digital media applications," in *Proceedings of the 2011 24th Canadian Conference on Electrical and Computer Engineering(CCECE)*, pp. 533–537, Niagara Falls, ON, Canada, May 2011.

[111] A. Sengupta, R. Sedaghat, and P. Sarkar, "A multi structure genetic algorithm for integrated design space exploration of scheduling and allocation in high level synthesis for dsp kernels," *Swarm and Evolutionary Computation*, vol. 7, pp. 35–46, 2012.

[112] G. Mariani, V.-M. Sima, G. Palermo, V. Zaccaria, C. Silvano, and K. Bertels, "Using multi-objective design space exploration to enable run-time resource management for reconfigurable architectures," in *Proceedings of the 2012 Design, Automation & Test in Europe Conference & Exhibition (DATE)*, pp. 1379–1384, Dresden, Germany, March 2012.

[113] B. C. Schafer and K. Wakabayashi, "Machine learning predictive modelling high-level synthesis design space exploration," *IET Computers Digital Techniques*, vol. 6, no. 3, pp. 153–159, 2012.

[114] H. Yang, C. Wang, and N. Du, "High level synthesis using learning automata genetic algorithm," *Journal of Computers*, vol. 7, no. 10, pp. 2534–2541, 2012.

[115] A. Sengupta, R. Sedaghat, and V. Mishra, "Execution time—area tradeoff in gausing residual load decoder: integrated exploration of chaining based schedule and allocation in HLS for hardware accelerators," *Facta Universitatis—Series: Electronics and Energetics*, vol. 27, no. 2, pp. 235–249, 2014.

[116] M. C. Bhuvaneswari, *Application of Evolutionary Algorithms for Multi-Objective Optimization in VLSI and Embedded Systems*, Springer, India, 2015.

[117] P. Sarkar, A. Sengupta, and M. K. Naskar, "GA driven integrated exploration of loop unrolling factor and datapath for optimal scheduling of CDFGs during high level synthesis," in *Proceedings of the 2015 IEEE 28th Canadian Conference on Electrical and Computer Engineering (CCECE)*, pp. 75–80, Halifax, New Scotia, Canada, May 2015.

[118] Y. Liu and B. C. Schafer, "Adaptive combined macro and micro-exploration of concurrent applications mapped on shared bus reconfigurable SoC," in *Proceedings of the 2015 Electronic System Level Synthesis Conference (ESLsyn)*, pp. 11–16, San Francisco, CA, USA, June 2015.

[119] N. Neves, R. Neves, N. Horta, P. Tomás, and N. Roma, "Multi-objective kernel mapping and scheduling for morphable many-core architectures," *Expert Systems with Applications*, vol. 45, pp. 385–399, 2016.

[120] A. Mahram and M. C. Herbordt, "NCBI BLASTP on high-performance reconfigurable computing systems," *ACM Transactions on Reconfigurable Technology and Systems*, vol. 7, no. 4, pp. 1–20, 2015.

[121] D. Liu and B. C. Schafer, "Efficient and reliable high-level synthesis design space explorer for FPGAs," in *Proceedings of the 2016 26th International Conference on Field Programmable Logic and Applications (FPL)*, pp. 1–8, Lausanne, Switzerland, August 2016.

[122] D. Reyes and Y. Maldonado, "Comparison of multi-objective evolutionary algorithms for high level synthesis in FPGA devices," *Computación Y Sistemas, Thematic Issue on Numerical and Evolutionary Optimization*, vol. 22, no. 2, pp. 425–437, 2018.

[123] N. K. Pham, *Multi-objective design automation for reconfigurable multi-processor systems*, PhD thesis, National University of Singapore, Singapore, 2017.

[124] B. C. Schafer, "Parallel high-level synthesis design space exploration for behavioral IPs of exact latencies," *ACM Transactions on Design Automation of Electronic Systems*, vol. 22, no. 4, pp. 1–20, 2017.

[125] B. C. Schafer and A. Mahapatra, "S2cbench: synthesizable SystemC benchmark suite for high-level synthesis," *IEEE Embedded Systems Letters*, vol. 6, no. 3, pp. 53–56, 2014.

[126] F. N. Taher, M. Kishani, and B. C. Schafer, "Design and optimization of reliable hardware accelerators: leveraging the advantages of high-level synthesis," in *Proceedings of the 2018 IEEE 24th International Symposium on On-Line Testing and*

Robust System Design (IOLTS), pp. 232–235, Platja d'Aro, Spain, July 2018.

[127] G. Wang, W. Gong, B. DeRenzi, and R. Kastner, "Design space exploration using time and resource duality with the ant colony optimization," in *Proceedings of the 43rd Annual Conference on Design Automation—DAC '06*, pp. 451–454, San Francisco, CA, USA, July 2006.

[128] G. Wang, W. Gong, B. Derenzi, and R. Kastner, "Exploring time/resource trade-offs by solving dual scheduling problems with the ant colony optimization," *ACM Transactions on Design Automation of Electronic Systems*, vol. 12, no. 4, 2007.

[129] V. K. Mishra and A. Sengupta, "Swarm intelligence driven design space exploration: an integrated framework for power-performance trade-off in architectural synthesis," in *Proceedings of the 2013 25th International Conference on Microelectronics (ICM)*, pp. 1–4, Beirut, Lebanon, December 2013.

[130] A. Sengupta and S. Bhadauria, "Exploration of multi-objective tradeoff during high level synthesis using bacterial chemotaxis and dispersal," *Procedia Computer Science*, vol. 35, pp. 63–72, 2014.

[131] A. Sengupta and V. K. Mishra, "Automated exploration of datapath and unrolling factor during power-performance tradeoff in architectural synthesis using multi-dimensional PSO algorithm," *Expert Systems with Applications*, vol. 41, no. 10, pp. 4691–4703, 2014.

[132] V. K. Mishra and A. Sengupta, "PSDSE: particle swarm driven design space exploration of architecture and unrolling factors for nested loops in high level synthesis," in *Proceedings of the 2014 Fifth International Symposium on Electronic System Design*, pp. 10–14, Surathkal, India, December 2014.

[133] V. K. Mishra and A. Sengupta, "Swarm-inspired exploration of architecture and unrolling factors for nested-loop-based application in architectural synthesis," *Electronics Letters*, vol. 51, no. 2, pp. 157–159, 2015.

[134] B. Schafer, "Probabilistic multiknob high-level synthesis design space exploration acceleration," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 35, no. 3, pp. 394–406, 2016.

[135] A. Sengupta and V. K. Mishra, "Swarm intelligence driven simultaneous adaptive exploration of datapath and loop unrolling factor during area-performance tradeoff," in *Proceedings of the 2014 IEEE Computer Society Annual Symposium on VLSI*, pp. 106–111, Tampa, FL, USA, July 2014.

[136] A. Sengupta and V. K. Mishra, "Simultaneous exploration of optimal datapath and loop based high level transformation during area-delay tradeoff in architectural synthesis using swarm intelligence," *International Journal of Knowledge-Based and Intelligent Engineering Systems*, vol. 19, no. 1, pp. 47–61, 2015.

[137] A. Sengupta and S. Bhadauria, "Exploration of multi-objective tradeoff during high level synthesis using bacterial chemotaxis and dispersal," in *Proceedings of the 18th International Conference on Knowledge-Based and Intelligent Information & Engineering Systems (KES 2014)*, pp. 63–72, Gdynia, Poland, September 2014.

[138] A. Sengupta and S. Bhadauria, "Automated exploration of datapath in high level synthesis using temperature dependent bacterial foraging optimization algorithm," in *Proceedings of the 2014 IEEE 27th Canadian Conference on Electrical and Computer Engineering (CCECE)*, Toronto, ON, Canada, May 2014.

[139] A. Sengupta and S. Bhadauria, "Error masking of transient faults: exploration of a fault tolerant datapath based on user specified power and delay budget," in *Proceedings of the 2014 International Conference on Information Technology*, pp. 345–350, Bhubaneswar, India, December 2014.

[140] S. Bhadauria and A. Sengupta, "Adaptive bacterial foraging driven datapath optimization: exploring power-performance tradeoff in high level synthesis," *Applied Mathematics and Computation*, vol. 269, pp. 265–278, 2015.

[141] A. Sengupta and S. Bhadauria, "Bacterial foraging driven exploration of multi cycle fault tolerant datapath based on power-performance tradeoff in high level synthesis," *Expert Systems with Applications*, vol. 42, no. 10, pp. 4719–4732, 2015.

[142] A. Sengupta and S. Bhadauria, "Automated design space exploration of transient fault detectable datapath based on user specified power and delay constraints," in *Proceedings of the VLSI Design, Automation and Test (VLSI-DAT)*, pp. 1–4, Hsinchu, Taiwan, April 2015.

[143] A. Sengupta and S. Bhadauria, "Automated design space exploration of multi-cycle transient fault detectable datapath based on multi-objective user constraints for application specific computing," *Advances in Engineering Software*, vol. 82, pp. 14–24, 2015.

[144] A. Sengupta and S. Bhadauria, "Untrusted third party digital IP cores: power-delay trade-off driven exploration of hardware Trojan secured datapath during high level synthesis," in *Proceedings of the 25th edition on Great Lakes Symposium on VLSI—GLSVLSI '15*, pp. 167–172, Pittsburgh, PA, USA, May 2015.

[145] A. Sengupta and V. K. Mishra, "Time varying vs. fixed acceleration coefficient PSO driven exploration during high level synthesis: performance and quality assessment," in *Proceedings of the 2014 International Conference on Information Technology*, pp. 281–286, Bhubaneswar, India, December 2014.

[146] E. Kougianos and S. P. Mohanty, "A nature-inspired firefly algorithm based approach for nanoscale leakage optimal RTL structure," *Integration, the VLSI Journal*, vol. 51, pp. 46–60, 2015.

[147] A. Sengupta and R. Sedaghat, "Swarm intelligence driven design space exploration of optimal k-cycle transient fault secured datapath during high level synthesis based on user power-delay budget," *Microelectronics Reliability*, vol. 55, no. 6, pp. 990–1004, 2015.

[148] A. Sengupta, "Exploration of k_c-cycle transient fault-secured datapath and loop unrolling factor for control data flow graphs during high-level synthesis," *Electronics Letters*, vol. 51, no. 7, pp. 562–564, 2015.

[149] D. Kachave and A. Sengupta, "Integrating physical level design and high level synthesis for simultaneous multi-cycle transient and multiple transient fault resiliency of application specific datapath processors," *Microelectronics Reliability*, vol. 60, pp. 141–152, 2016.

[150] A. Sengupta and D. Kachave, "Low cost fault tolerance against k_c-cycle and k_m-unit transient for loop based control data flow graphs during physically aware high level synthesis," *Microelectronics Reliability*, vol. 74, pp. 88–99, 2017.

[151] A. Sengupta and D. Kachave, "Particle swarm optimisation driven low cost single event transient fault secured design during architectural synthesis," *The Journal of Engineering*, vol. 2017, no. 6, pp. 184–194, 2017.

[152] A. Sengupta and D. Kachave, "Spatial and temporal redundancy for transient fault-tolerant datapath," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 54, no. 3, pp. 1168–1183, 2018.

[153] A. Sengupta and R. Sedaghat, "Exploration of optimal multi-cycle transient fault secured datapath during high level synthesis based on user area-delay budget," in *Proceedings of the 2015 IEEE 28th Canadian Conference on Electrical and Computer Engineering (CCECE)*, pp. 69–74, Halifax, New Scotia, Canada, May 2015.

[154] A. Sengupta and S. Bhadauria, "User power-delay budget driven PSO based design space exploration of optimal k-cycle transient fault secured datapath during high level synthesis," in *Proceedings of the Sixteenth International Symposium on Quality Electronic Design*, pp. 289–292, Santa Clara, CA, USA, March 2015.

[155] A. Sengupta and D. Kachave, "Generating multi-cycle and multiple transient fault resilient design during physically aware high level synthesis," in *Proceedings of the 2016 IEEE Computer Society Annual Symposium on VLSI (ISVLSI)*, pp. 75–80, Pittsburgh, PA, USA, July 2016.

[156] A. Sengupta and S. Bhadauria, "Secure information processing during system-level: exploration of an optimized trojan secured datapath for CDFGs during HLS based on user constraints," in *Proceedings of the 2015 IEEE International Symposium on Nanoelectronic and Information Systems*, pp. 1–6, Indore, India, December 2015.

[157] A. Sengupta, S. Bhadauria, and S. P. Mohanty, "TL-HLS: methodology for low cost hardware Trojan security aware scheduling with optimal loop unrolling factor during high level synthesis," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 36, no. 4, pp. 655–668, 2017.

[158] A. Sengupta, D. Roy, and S. Bhadauria, "Low cost optimized Trojan secured schedule at behavioral level for single & nested loop control data flow graphs (invited paper)," *Integration*, vol. 58, pp. 378–389, 2017.

[159] A. Sengupta, S. Bhadauria, and S. P. Mohanty, "Embedding low cost optimal watermark during high level synthesis for reusable IP core protection," in *Proceedings of the 2016 IEEE International Symposium on Circuits and Systems (ISCAS)*, pp. 974–977, Montreal, Quebec, Canada, May 2016.

[160] P. Sarkar, A. Sengupta, S. Rathlavat, and M. K. Naskar, "Designing low-cost hardware accelerators for CE devices [hardware matters]," *IEEE Consumer Electronics Magazine*, vol. 6, no. 4, pp. 140–149, 2017.

[161] P. Sarkar, A. Sengupta, S. Rathlavat, and M. K. Naskar, "A firefly algorithm driven approach for high level synthesis," in *Proceedings of the 2017 IEEE International Symposium on Nanoelectronic and Information Systems (iNIS)*, pp. 15–19, Bhopal, India, December 2017.

[162] A. Sengupta, D. Roy, S. P. Mohanty, and P. Corcoran, "DSP design protection in CE through algorithmic transformation based structural obfuscation," *IEEE Transactions on Consumer Electronics*, vol. 63, no. 4, pp. 467–476, 2017.

[163] A. Sengupta and D. Roy, "Protecting IP core during architectural synthesis using HLT-based obfuscation," *Electronics Letters*, vol. 53, no. 13, pp. 849–851, 2017.

[164] A. Sengupta, D. Kachave, and D. Roy, "Low cost functional obfuscation of reusable IP cores used in CE hardware through robust locking," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 38, no. 4, pp. 604–616, 2019.

[165] V. K. Mishra and A. Sengupta, "Comprehensive operation chaining based schedule delay estimation during high level synthesis," in *Proceedings of the 2017 IEEE International Symposium on Nanoelectronic and Information Systems (iNIS)*, pp. 66–68, Bhopal, India, December 2017.

[166] A. Sengupta, D. Roy, S. P. Mohanty, and P. Corcoran, "Low-cost obfuscated JPEG CODEC IP core for secure CE hardware," *IEEE Transactions on Consumer Electronics*, vol. 64, no. 3, pp. 365–374, 2018.

[167] A. Sengupta, S. Neema, P. Sarkar, P. S. Harsha, S. P. Mohanty, and M. K. Naskar, "Obfuscation of fault secured DSP design through hybrid transformation," in *Proceedings of the 2018 IEEE Computer Society Annual Symposium on VLSI (ISVLSI)*, pp. 732–737, Hong Kong, China, July 2018.

[168] D. Roy, P. Sarkar, A. Sengupta, and M. K. Naskar, "Optimizing DSP cores using design transformation [hardware matters]," *IEEE Consumer Electronics Magazine*, vol. 7, no. 4, pp. 91–94, 2018.

[169] D. Kachave and A. Sengupta, "Shielding CE hardware against reverse-engineering attacks through functional locking hardware matters]," *IEEE Consumer Electronics Magazine*, vol. 7, no. 2, pp. 111–114, 2018.

[170] A. Antola, F. Ferrandi, V. Piuri, and M. Sami, "Semi-concurrent error detection in data paths," *IEEE Transactions on Computers*, vol. 50, no. 5, pp. 449–465, 2001.

[171] K. Barker, T. Benson, D. Campbell et al., *PERFECT (Power Efficiency Revolution for Embedded Computing Technologies)*, Pacific Northwest National Laboratory and Georgia Tech Research Institute, Atlanta, GA, USA, 2018.

[172] CyberWorkBench, "Benchmarks," 2018, http://www.cyberworkbench.com.

[173] L. Pouchet, "Polybench: the polyhedral benchmark suite," 2018, http://www.cs.ucla.edu/pouchet/software/.

[174] L. Pouchet, "ACM/SIGDA benchmark," 2018, https://people.engr.ncsu.edu/brglez/cbl/.

[175] A. Petitet, R. Whaley, J. Dongarra, and A. Cleary, "HPL—a portable implementation of the high-performance linpack benchmark for distributed-memory computers, version 2.0," 2018.

[176] BDTI-DSP, "Kernel-benchmarks," 2018, https://www.bdti.com/services/benchmarks/dkb.

[177] A. Danalis, G. Marin, C. McCurdy et al., "The scalable heterogeneous computing (SHOC) benchmark suite," in *Proceedings of the 3rd Workshop on General-Purpose Computation on Graphics Processing Units—GPGPU '10*, vol. 63–74, March 2010.

[178] S. Xydis, G. Palermo, V. Zaccaria, and C. Silvan, "A meta-model assisted coprocessor synthesis framework for compiler/architecture parameters customization," in *Proceedings of the Design, Automation & Test in Europe Conference & Exhibition (DATE), 2013*, pp. 659–664, Grenoble, France, March 2013.

[179] P. Czyżak and A. Jaszkiewicz, "Pareto simulated annealing—a metaheuristic technique for multiple-objective combinatorial optimization," *Journal of Multi-Criteria Decision Analysis*, vol. 7, no. 1, pp. 34–47, 1998.

[180] E. Zitzler, L. Thiele, M. Laumanns, C. M. Fonseca, and V. G. da Fonseca, "Performance assessment of multiobjective optimizers: an analysis and review," *IEEE Transactions on Evolutionary Computation*, vol. 7, no. 2, pp. 117–132, 2003.

[181] A. Auger, J. Bader, D. Brockhoff, and E. Zitzler, "Hyper-volume-based multiobjective optimization: theoretical

foundations and practical implications," *Theoretical Computer Science*, vol. 425, pp. 75–103, 2012.

[182] M. P. Hansen and A. Jaszkiewicz, "Evaluating the quality of approximations to the non-dominated set," Technical report, Technical University of Denmark, Lyngby, Denmark, 1998.

[183] E. Zitzler and L. Thiele, "Multiobjective evolutionary algorithms: a comparative case study and the strength Pareto approach," *IEEE Transactions on Evolutionary Computation*, vol. 3, no. 4, pp. 257–271, 1999.

[184] K. Bringmann and T. Friedrich, "Approximation quality of the hypervolume indicator," *Artificial Intelligence*, vol. 195, pp. 265–290, 2013.

[185] A. Punitha and M. Joseph, "A temperature-aware scheduling with incremental binding and floorplanning for HLS," *Middle-East Journal of Scientific Research*, vol. 22, no. 6, pp. 822–828, 2014.

[186] R. Mukherjee, M. S. Ogrenci, and G. Memik, "Temperature-aware resource allocation and binding in high-level synthesis," in *Proceedings of the 42nd annual conference on Design automation—DAC'05*, pp. 196–201, Anaheim, CA, USA, June 2005.

[187] K. Deb and H. Jain, "An evolutionary many-objective optimization algorithm using reference-point-based non-dominated sorting approach, part I: solving problems with box constraints," *IEEE Transactions on Evolutionary Computation*, vol. 18, no. 4, pp. 577–601, 2014.

[188] H. Jain and K. Deb, "An evolutionary many-objective optimization algorithm using reference-point based non-dominated sorting approach, part II: handling constraints and extending to an adaptive approach," *IEEE Transactions on Evolutionary Computation*, vol. 18, no. 4, pp. 602–622, 2014.

[189] Q. Zhang and H. Li, "MOEA/D: a multiobjective evolutionary algorithm based on decomposition," *IEEE Transactions on Evolutionary Computation*, vol. 11, no. 6, pp. 712–731, 2007.

[190] A. Ibrahim, S. Rahnamayan, M. M. Vargas, and K. Deb, "3D-RadVis: visualization of Pareto front in many-objective optimization," in *Proceedings of the 2016 IEEE Congress on Evolutionary Computation (CEC)*, pp. 736–745, Vancouver, British Columbia, Canada, July 2016.

[191] M. Richards, *Microservices vs. Service-Oriented Architecture*, O'Reilly Media, Inc., Newton, MA, USA, 2016.