

FPGA设计空间探索综述

甘芷莹

合肥工业大学, 安徽 合肥

收稿日期: 2022年3月16日; 录用日期: 2022年4月15日; 发布日期: 2022年4月22日

摘要

设计空间探索(DSE)问题一直是一个具有挑战性的问题, 因为设计空间大小呈指数倍增长, 仿真时间非常长, 并且DSE存在多个目标。由于人工探索的局限性促使了许多自动探索的算法的提出, 于是设计师们希望通过一种自动化的方法来解决这个问题, 从而加快探索过程。本文旨在通过对设计空间探索技术进行的调查, 以便为研究人员提供有关未来研究方向的见识。

关键词

FPGA, 高层次综合, DSE

A Review of Spatial Exploration of FPGA Design

Zhiying Gan

Hefei University of Technology, Hefei Anhui

Received: Mar. 16th, 2022; accepted: Apr. 15th, 2022; published: Apr. 22nd, 2022

Abstract

The Design Space Exploration (DSE) problem has always been a challenging one, because the size of the design space is exponentially increasing, the simulation time is very long, and DSEs have multiple targets. Because the limitations of human exploration have led to the introduction of many algorithms for automatic exploration, designers hope to solve this problem through an automated method to speed up the exploration process. This paper aims to provide researchers with insights into future research directions through the investigation of spatial exploration techniques for design.

Keywords

FPGA, High-Level Synthesis, DSE

Copyright © 2022 by author(s) and Hans Publishers Inc.

This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

1. 引言

现场可编程门阵列(FPGA)是一种可重新配置的数字逻辑设备,有助于快速进行原型设计和设计验证,从而使设计人员能够开发强大的硬件和软件解决方案。近年来,FPGA 系统已在许多应用中获得普及,例如数字信号处理、高性能计算、生物应用等。FPGA 显示出在硬件专业化方面的巨大潜力,为了获得高质量的结果(QoR),CAD 工具需要大量的人工来配置大量的设计和工具参数。由于 CAD 映射流程的每个步骤通常都是 NP-complete,因此有必要使用试探法在合理的时间内生成 FPGA 比特流。现代 FPGA 支持数百万个 LUT,数千个 DSP 和片上 Block RAM,以及各种奇特的硬件功能,导致编译时间增加到数小时或数天。当处理设计流程中的时序约束时,通常会反复使用 FPGA CAD 工具来提供时序收敛,而长时间运行时间和复杂的配置选项的结合使开发人员越来越难以管理此设计过程。FPGA 编译流程可以分解为一系列连续的 CAD 阶段,比如高层次综合、逻辑综合、工艺映射、逻辑单元打包和布局布线等。在每个阶段进行设计空间探索(DSE)都有可能影响下面阶段的选择。DSE 中存在多个目标,从而产生了一组最优解,也称为 Pareto 最优解,而不是单个最优解。在没有任何进一步信息的情况下,这些 Pareto 最优解都不能被认为比其他解更好。因此,非常需要一种自动方法来尽快找到尽可能多的 Pareto 最优设计。

本文调查发现对相关技术进行分类时,使用不同的技术主要区别所探索的问题不同,导致输入的数据类型不同。输入的数据主要分成无序的,有序的和不规则的。如果探索的是组合优化问题即输入的数据是无序的,通常使用的是传统的机器学习方法或是本地搜索技术,如果输入的数据是有序的(图片类型)通常用深度学习的方法,而不规则结构(比如图)会使用图学习的方法。

本文其余部分安排如下:第二节讨论了先前设计空间探索技术相关的工作,详细的阐述了针对设计空间探索提出的不同类型的算法,第三节对基准进行总结,讲述了高层次综合常见的基准测试,第四节介绍了 DSE 常用的评估指标,第五节是结论。

2. 设计空间探索技术

2.1. 高层次综合

高层次综合(High level Synthesis, HLS)通过允许此类任务以更高的抽象级别进行(以 C/C++或 system C 等高级语言作为输入综合到优化的低级实现(例如 RTL 或门级网表)),HLS 技术已被广泛研究了二十多年,并且到目前为止,不仅在学术界而且在工业中已经开发了许多 HLS 工具,从而简化了硬件设计和验证。在 HLS 中进行有效的设计空间探索是一项重要的研究任务,现代 HLS 工具中会提供一些可用的循环优化的设置,比如展开,流水线,数组拆分,内联等设置,组合的方式呈指数增长。

2.2. 基于学习的方法

基于学习的方法,通过在运行实际模拟/合成之前预测解决方案质量来指导 DSE 过程,与本地搜索技

术相比，基于学习的方法可以产生更好的解决方案质量，并且需要较短的仿真/综合运行时间，使用基于机器学习算法的算法比较如表 1 所示。

Table 1. Comparison of algorithms using machine learning algorithms
表 1. 使用机器学习算法的算法比较

	方法	基准	样本量	评估指标
[1]	基于决策树机器学习算法的快速模拟退火算法	S2CBench	-	ADRS
[2]	随机森林，转导实验设计	DFT	合成了一组受限制的 242 个旋钮设置，其中包括 11 个循环操作，2 个函数内联选择和 11 个可行时钟周期。	ADRS
[3]	随机森林	adpcm, sha, jpeg 和 mpeg2	-	ADRS
[4]	贝叶斯优化	MachSuite	AES: 104,976 Backpropagation: 23,328 Radix Sort: 7776	LUT 利用率与有效等待时间
[5]	多保真度优化方法高斯过程	MachSuite CHStone	AES: 55,566、 Backpropagation: 23,253、 Radix Sort: 4096、 ADPCM: 17,024	LUT 的使用，最坏情况下的估计周期数和时钟周期
[6]	Pyramid 框架：集成任意机器学习技术	Machsuit, S2CBench, CHStone 和 Rosetta	-	时序和资源使用

我们知道机器学习训练模型是一个非常耗时的过程，以如今计算力而言要想将每种可能的参数组合都实验一遍(即网格搜索)明显不现实，所以一般就是事先限定若干种可能，机器学习方法主要理论是通过仅对感兴趣的一部分进行采样，学习了采样组合的特性，企图去减少设计空间。即使不能找到 Pareto 最优解，但是可以从概率上找到一个“相对最优”。比如以下基于传统机器学习算法的模型，Mahapatra 和 Schafer [1]提出了一种基于决策树机器学习算法的快速模拟退火算法(FSA)，在初始阶段生成决策树，以决定哪些属性有助于最大化成本函数，然后固定这些属性，很大程度减少了设计空间，可以在 HLS 中进行有效的多目标设计空间探索。Liu 和 Carloni [2]提出了基于随机森林的学习模型以解决监督设计空间探索，且在初始阶段引入了转导实验设计，它可以对具有代表性和难以预测的微体系结构选择进行抽样，并将其用于训练学习模型。同样是基于随机森林机器学习分类的方法，Zacharopoulos 等[3]提出的方法着眼于 Loop Unrolling 指令，能够预测目标应用程序中所有循环的最佳循环展开因子。Lo 和 Chow 在文章 [4]使用基于贝叶斯优化方法来自动选择 HLS 指令设置，以及在文章[5]采用了多保真度优化方法，对 HLS CAD 流程中可用的多个度量的组合用高斯过程(GP)回归方法进行建模。以上是使用单个机器学习算法建模的，还有集合了多种算法进行建模。比如 Makrani 等[6]采用的方法是基于集成学习提出了一个称为 Pyramid 的框架来准确估计 HLS 设计的最佳性能和资源利用率。

基于神经网络的模型可映射任意复杂的非线性关系，且具有高速寻找优化解的能力。Hsiao 和 Anderson [7]提出了基于卷积神经网络(CNN)的预测器，它可以预测由于减少位宽而带来的合成后区域节省，并为用户提供程序变量及其区域影响的排名，实现较高的区域预测精度，并能够快速探索节省区域的机会。算法比较如表 2 所示。

Table 2. Comparison of algorithms using neural network algorithms
表 2. 使用神经网络算法的算法比较

	方法	基准	样本量	评估指标
[7]	卷积神经网络(CNN)	CHStone	约 7000 个位宽减少的电路	自适应逻辑模块减少的数量

其它基于学习的方法, Zhao 等[8]提出了 COMBA, 这是一个基于模型的综合分析框架, 相当于一个 Vivado HLS 工具。它能够使用可插拔分析模型, 递归数据收集器(RDC)以及度量指导的设计空间探索算法(MGDSE)来分析与设计描述中的功能, 循环和数组有关的多编译指示的影响。Xydis 等[9]开发了一种名为 SPIRIT 的新型 HLS 探索框架, 该框架利用响应面模型(RSMs)来预测设计点的质量, 并且使用频谱技术来指导迭代细化。Liu 等[10]发现原以为当重新合成(HLS)时, 导致 ASIC 情况下导致 Pareto 最优设计的综合指令的相同组合, 当以 FPGA 为目标时, 这也将导致 Pareto 最优设计, 实验结果并非如此。引入基于预测模型的方法以快速进行, 而无需重新探索, 找到给定 ASIC 的探索结果的 FPGA Pareto 最优设计。算法比较如表 3 所示。

Table 3. Comparison of algorithms using other learning-based algorithms
表 3. 使用其他基于学习的算法比较

	方法	基准	样本量	评估指标
[8]	COMBA 综合分析框架	Polybench JCHStone	-	DSP 和 Block-RAM 的使用
[9]	光谱技术	LMS, MPEG IDCT, Gauss Blur, Sobel, DCT-1D, YUV2RGBA, MatMul, JPEG 2D-DCT	-	ADRS
[10]	基于预测模型的方法	S2CBench	-	ADRS

2.2.1. 群智能优化算法

由于解决方案空间很大, 很多的 DSE 问题依赖于本地搜索技术, 例如, 像遗传算法和 PSO 这些经典黑盒优化算法, 我们归类为群智能优化算法。但是, 当与 CAD 工具一起使用时, 这些算法在每一步都需要实际的仿真/综合来获得解决方案的质量, 因此要经历长时间的仿真/综合运行时间。群智能优化算法主要模拟了昆虫、鸟群和鱼群的群集行为, 这些群体按照一种合作的方式寻找食物, 群体中的每个成员通过学习它自身的经验和其他成员的经验来不断地改变搜索的方向。群智能优化算法的突出特点就是利用了种群的群体智慧进行协同搜索, 从而在解空间内找到最优解, 比如[11]-[20]。我们主要研究以下几篇, Ferrandi 等[11]与 Schafer 和 Wakabayashi [12]提出了一种基于遗传算法的高层次综合设计空间探索方法, 而 Liu 等[13]同样提出了基于遗传算法进行搜索, 并且指出了用机器学习进行建模, 该方法首先针对给定的 HLS 多个行为描述, 针对 ASIC 生成各种数据流配置, 然后基于成分预测模型, 映射到 FPGA 时自动重新优化数据流。与以上工作有所差异的是, Sarkar 等[14]提出一种基于自适应遗传算法的解决方案, 用于在 HLS 期间对控制数据流图的数据路径资源配置和循环展开因子(UF)进行综合探索, 以实现最佳调度。Sengupta 和 Bhadauria [15]提出用粒子群优化算法驱动 DSE, 除了在探索过程中将功耗和执行时间(或延迟)作为设计目标之外, 还考虑了最佳数据路径生成过程中的多周期瞬态故障约束。

2.2.2. 其它 DSE 算法

除了以上方法外, 还有以下其他高层次综合 DSE 算法。比如论文[21] [22] [23] [24] [25]主要通过对高级语言程序修改或插入指令实现设计空间探索。Silva 和 Bampi [21]提出了一种迭代, 递归和多平台的

方法, 有效地探索了设计空间。Schafer 在论文[22]中提出了对 knobs 进行分类并顺序进行探索来加快设计空间探索(DSE)的速度, 在论文[23]中介绍了一种分层方法, 分别探索每个功能, 然后执行合并阶段以获得总体优势结果。Gao 等[24]介绍了第一个开源程序优化器, 用于自动重写给定程序以优化延迟, 同时控制准确性和面积。

根据衡量设计空间探索算法性能的常用方法, 可看出时序是评估算法好坏的重要指标, 但是对于精准的时序信息提取存在不小的难度。在 Liu 等[25]提到缓慢的 CAD 运行时间和繁琐的低级寄存器传输级设计一直困扰着 FPGA 开发, 所以设计离线表征工具, 以生成校准路径并使用 RapidWright 提取 FPGA 互连资源的时序延迟。

2.3. RTL

当前的技术主要集中在设计流程的单个阶段, 例如高级综合(HLS), 从而错过了重要的跨界优化机会。现有的方法通常使用 pre-PnR 甚至是 pre-synthesis 报告来评估设计点的质量, 虽然这可以缩短执行时间, 但仅依靠早期设计阶段的粗略估计可能 DSE 无法达到高质量的设计点。为了解决上述缺陷, 以下是基于多阶段设计空间搜索。

这类文章也用了各式各样的方法进行设计空间探索。Kapre 等[26]介绍了一种基于智能, 机器学习和云计算的方法, 可有效地自动选择参数, 以最大程度地降低计时成本。[27] [28] [29]用的都是 InTime 插件。Kapre 等[27]建议在贝叶斯学习和分类框架的支持下在云上并行评估多个 CAD 运行, 以生成最有可能有助于实现时序收敛的多个 CAD 参数组合。虽然 InTime 对于 FPGA 设计的时序收敛很有用, 但是有可能需要在 2-3 天的探索中以完全自动化的方式实现定时收敛, 于是 Yanghua 等[28]人通过将特定于设计的特征选择与分类相结合, 这些分类方法可提高模型质量并减少交付时序收敛所需的迭代次数。同样为了提供具有高分类器准确性的时序收敛, 从而减少时序收敛的迭代次数。在 Yanghua 等[29]又专注于选择合适的机器学习算法和技术, 以识别和调整最有前途的解决方案。此外 Xu 等[30]也专注于选择合适的机器学习算法, 设计了一种高效的并行化方案, 该方案使多个基于 multi-armed bandit (MAB)的自动调谐器能够同时探索设计空间, 具体来说 OpenTuner (一个开源自动调整框架)使用的 MAB 算法将每种搜索方法视为一条手臂。同样 Ustun 等[31]也是基于 OpenTuner 构建学习辅助 FPGA 自动调整框架, 具体来说, LAMDA 选择了 OpenTuner 提出的一组看不见的配置, XGBoost 回归器用于根据工具和特定于设计的功能估算时序。以上的论文多数是针对时序的, Mаметjanov 等[32]提出基于机器学习的方法来自动调整 FPGA 设计参数可以大大降低功率要求, 同时仍然满足时序约束。

同样为了提高预测准确性并降低了过多的仿真成本, 与传统的蛮力 DSE 方法相反, Guo 等[33]提出了 Co-Training Model Tree (COMT)方法, 该方法可以利用未标记的设计配置来改进模型。除了提高了预测精度外, 由于使用了可理解的模型树, COMT 还可以指导微处理器的设计。

我们知道准确地预测任何配置的性能是一个难题, 它需要大量的训练模拟来达到不错的精度, 所以回归模型大都集中于预测任何给定体系结构配置的性能, 而 Chen 等[34]认为架构师在 DSE 流程中最需要的信息是, 给定的配置在存在设计约束的情况下是否会比另一个配置表现更好, 还是比目前为止看到的任何其他配置都更好, 而不是精确地估计该配置的性能。ArchRanker 采用了一种称为 RankBoost 的代表性学习排名算法来构建排名模型, 以预测两种架构配置中哪一种效果最佳。

2.4. 布局布线

在 2.1 和 2.2 中所解决的问题主要是组合优化问题(属于无序的数据), 所以主要使用的是机器学习方法, 人们尝试过用深度学习解决组合优化问题但效果不明显, 但人们仍希望使用深度学习解决问题, 于

是把问题转移到布线性预测上。早期的可布线性预测有助于设计人员和工具执行预防措施，从而可以主动避免违反设计规则的情况。对于大型设计，物理设计过程通常要花费数小时至数天的时间，而布线是最关键的步骤，而布局布线过程不能分开，特定的位置可能根本无法布线，而稍微不同的位置可能会产生高质量的布线。

使用深度学习意味着数据的读取是以图片的形式，即将布局信息认为是一张图片的信息。我们知道，设计一个既准确又快速的布线预测器是一个巨大的挑战，Xie 等[35]利用卷积神经网络来应对这一挑战，提出了称为 RouteNet 的方法可以在不进行全局布线的情况下评估单元布局解决方案的总体可布线性，或者预测 DRG (设计规则检查)热点的位置。Yu 和 Zhang [36]在[35]的基础上提出了一种基于条件生成对抗网络(conditional GAN)的方法，该方法通过给定的解决方案来估计详细的布线拥塞。所提出的方法使用条件生成对抗网络模型完全预测了布线拥堵热图。在 Yu 和 Zhang 的基础之上，Alawieh 等[37]使用高分辨率图像，加快 FPGA 布局和布线的闭合速度。

2.5. IP 设计空间探索

IP 核心是逻辑或数据块，用于制造产品的 FPGA 或 ASIC。设计人员可以通过利用通常由第三方开发的现有 IP 模块来构建整个芯片或其一部分，而不是从头开始设计芯片中的每个组件。这种做法极大地减少了较大芯片中各个子模块的开发时间和成本。作为设计重用的基本要素，IP 内核是不断重复使用以前设计的组件的不断发展的电子设计自动化(EDA)行业趋势的一部分。Papamichael 等[38]提出了一种在 IP 生成器中内置扩展遗传算法(GA)来执行自动 IP 参数调整的方法，并且做了一些扩展，使 IP 作者可以嵌入相关的设计知识，在 IP 作者知识的帮助下，Nautilus 指导的 GA 可以进一步加速 GA 搜索过程。Papamichael 是基于进化算法进行 IP 设计空间搜索，如今 HLS 工具供应商已经开始提供保护行为 IP，允许 BIP 用户以编译指示的形式将综合指令插入源代码，所以 Wang 和 Schafer [39]提出了一种通过对 BIP 的不同部分进行加密来选择性地控制搜索空间的方法。

2.6. 其它

前面讨论过对于有序和无序数据所常用的技术，而对于不规则结构主要使用图学习。我们知道如果使用深度学习可以将输入视为图像，以便在使用 DL 进行图像处理的先前工作的基础上进行构建。但是，当书投入的是一个网表，如果用深度学习处理网表，估计电路中的性能需要考虑很多的偏置，比如图片放大放小或是旋转都会产生不必要的影响，所以 Ma [40]考虑使用图学习。Ma 为了处理逻辑电路的不规则图表示，提出了一种高性能的图卷积网络(GCN)模型，该方法适用于识别网表中难以观察的点，与最新的商业工具相比，基于 GCN 模型和迭代观察点插入流程，在工业设计上实现了更好的可测试性。

除此之外现代 VLSI 同样面临着 DSE 复杂问题，比如[41] [42] [43] [44] [45]。主要详细介绍一下几篇，[41] [42] [43]使用 SynTunSys 使设计人员的决策过程中的关键部分自动化，来控制综合参数调整过程，SynTunSys 包含有决策引擎算法组件，决策引擎解决的一般问题可以描述为黑盒调整或搜索，并且也已使用多种技术来解决，例如机器学习和贝叶斯优化等。此外还有 Kwon 等[45]提出了一种 LSPD 参数推荐器系统，该系统通过张量分解和回归学习协作预测模型。

对于专用多处理器片上系统(MPSoC)的设计，也可以调整各种可自定义参数，以根据选定的品质因数找到最佳折衷方案。比如论文[46]提出了结合了实验设计(DoE)和响应面建模(RSM)技术，用于管理系统级约束。

3. 基准

FPGA 十分依赖于基准来评估其硬件和软件解决方案的性能，并且标准基准套件对于研究人员定量

评估其算法至关重要。因此，必须使用标准和公平的基准测试实践来评估 FPGA 系统并确定其支持目标应用程序的潜力。例如，最终用户可以研究各种 FPGA 供应商发布的基准测试结果，以选择适合于预期应用的 FPGA 器件。因此，FPGA 社区开发了许多基准，以评估实现这些系统的硬件和软件解决方案。这项调查也探索了基准测试的使用，以评估包含 FPGA 器件及其相关软件设计工具链的系统。

这段讨论的是，不同的基准是基于不同的语言的基准程序，针对的目标也不同。虽然有几篇论文选用了相同的基准套件，但是其真正选用的基准或是数量都有所不同，对于评估指标也有差异。比如在论文[5]使用 CHStone 套件中 10 个基准，主要针对减少位宽而带来的合成后区域节省，生成了约 7000 个位宽减少的电路，其目标是估计自适应逻辑模块(ALM)的面积消耗，其中 ALM 是具有两个触发器的双输出 6-LUT。论文[6]只使用 CHStone 套件中的 ADPCM 基准，ADPCM 设计空间大小为 17024，这篇论文将 LUT 的使用，最坏情况下的估计周期数和时钟周期视为相关的设计指标。论文[7]的估计目标为 LUT, FF, DSP 和 BRAM 的资源使用情况，以及 Minerva 报告的最大时钟频率。

总结上述列举的论文，下面介绍几个 DSE 中比较常见的基准。

3.1. CHStone

CHStone [47]是一套基于 C 的高级综合的基准程序，CHStone 套件包含从各种应用程序领域中选择的 12 个程序，包括算术，媒体处理，安全性和微处理器等。CHStone 包含 12 个基准测试 aes, blowfish, dfadd, dfdiv, dfmul, dfsin, gsm, mips, motion, sha, adpcm, jpeg (四个算术程序，四个媒体应用程序，三个加密程序和一个处理器)。CHStone 基准套件非常易于使用，CHStone 基准测试程序使用有限的标准的 C 语言编写，不使用大多数现有 HLS 工具不支持的以下数据类型和构造：浮点数据，诸如 struct 的复合数据类型，动态内存分配和递归。此外 CHStone 基准测试套件未使用此类特定于工具的扩展，因此 CHStone 具有高度的可移植性。使用 CHStone 的论文有[5] [6] [7]。

3.2. S2CBench

S2CBench [48]是用于高层次综合的 SystemC 基准套件，不仅包括不同大小的应用程序，而且这些应用程序通常来自 HLS 中使用的各个领域(例如，加密，图像和 DSP 应用程序)，而且还测试了每种应用程序中的特定优化技术。S2CBench 包括 12+1 针对 HLS 中通常使用的各种应用的程序。12 个基准符合最新的 SystemC 可综合子集，而一种设计(FFT)是不可综合的，因为它包含三角和浮点运算(qsort, Sobel, aes_cipher, kasumi, md5C, snow3G, adpcm, fft, fir, decimation, interpolation, idct, disparity)。CHStone 基准套件与 S2CBench 的工作类似，它以 HLS 为目标，并包含一组 ANSI-C 程序，使用 ANSI-C 的主要缺点是某些主要的商用 HLS 工具不支持 ANSI-C，而所有主要的 HLS 工具都支持一种通用语言：SystemC。使用 S2CBench 的论文有[1] [6] [10] [13] [22] [23] [39]。

3.3. MachSuite

高级综合和与加速器相关的体系结构显示了工作负载选择方面的巨大差异。为了提高加速器研究社区的标准化，MachSuite [49]设计目的便是用于评估高级综合工具和以加速器为中心的体系结构。MachSuite 是一组涵盖 12 个不同内核的 19 个基准测试，旨在涵盖各种应用程序域并结合不同的算法选择(AES/AES, BFS/BULK, BACKPROP/BACKPROP, BFS/QUEUE, FFT/STRIDED, FFT/TRANPOSE, GEMM/NCUBED, GEMM/BLOCKED, KMP/KMP, MD/KNN, MD/GRID, NW/NW, SORT/MERGE, SORT/RADIX, SPMV/CRS, SPMV/ELLPACK, STENCIL/STENCIL2D, STENCIL/STENCIL3D, VITERBI/VITERBI)。与 CHStone 相比，CHStone 被设计为 HLS 工具的评估机制，它专注于少量的低级

计算, 虽然对于评估较旧的 HLS 框架的功能很有用, 但在现代 CAD 工具链所处理的复杂设计以及许多架构师正在考虑和构建的扩展系统设计的背景下, 有些不足。使用 MachSuite 的论文有[4] [5]。

3.4. PolyBench

PolyBench [50]是一个包含 30 个数值计算的基准套件, 具有静态控制流, 并从各个应用程序域的操作中提取(线性代数计算, 图像处理, 物理模拟, 动态规划, 统计信息等)。使用 PolyBench 的论文有[8] [24]。

3.5. IPSD2015

上面提及的都是高层次综合相关的基准, ISPD 2015 布线性预测相关的基准[51]。ISPD 2015 是 Fence Regions 和 Routing Blockages 的基准。时序约束和区域布局约束是实际工业设计中常见的特征。布局质量的最终衡量标准是其时序性能及其详细的可布线性, 尽管这里描述的基准仍然缺乏时序约束, 但它们的可布线性和区域约束相当完整。

4. 评估指标

衡量设计空间探索算法性能除了需要有基准测试外, 还需要有作为评估或比较的标准。比较不同的多目标函数优化方法时的主要问题是如何衡量结果的质量: 与 Pareto front 的接近程度, 下面列出了一些用于衡量设计空间探索算法性能的常用方法: 1) 距参考集的平均距离(ADRS): ADRS 表示 Pareto-front 与 reference-front 的距离。值越小, 说明所获得的 approximate-front 和 reference-front 越接近。2) Cardinality: 基数表示每种方法找到的主要设计的数量。基数高表明有很多可供选择的解决方案, 但是需要与其他数据一起仔细解释。3) IPC (Instruction-per-Cycle): 表示每个时钟周期执行的平均指令数。

5. 结论

本文调查了设计空间探索的不同技术, 以及高层次综合所使用的基准测试以及 DSE 评估指标。可以看出近年来人们对于 DSE 关注度越来越高, 仔细对比各种方法的基准测试或是评估指标会发现有很大的不同, 没有一个通用的基准供各种方法使用, 所以无法做到客观的比较, 以至于现有的技术始终无法达到对实施成本的精确预估。

参考文献

- [1] Mahapatra, A. and Schafer, B.C. (2014) Machine-Learning Based Simulated Annealer Method for High Level Synthesis Design Space Exploration. *Proceedings of the 2014 Electronic System Level Synthesis Conference (ESLsyn)*, San Francisco, 31 May-1 June 2014, 1-6. <https://doi.org/10.1109/ESLsyn.2014.6850383>
- [2] Liu, H.Y. and Carloni, L.P. (2013) On Learning-Based Methods for Design-Space Exploration with High-Level Synthesis. 2013 *50th ACM/EDAC/IEEE Design Automation Conference (DAC)*, Austin, 29 May-7 June 2013, 1-7.
- [3] Zacharopoulos, G., Barbon, A., Ansaloni, G. and Pozzi, L. (2018) Machine Learning Approach for Loop Unrolling Factor Prediction in High Level Synthesis. 2018 *International Conference on High Performance Computing & Simulation (HPCS)*, Orleans, 16-20 July 2018, 91-97. <https://doi.org/10.1109/HPCS.2018.00030>
- [4] Lo, C. and Chow, P. (2016) Model-Based Optimization of High Level Synthesis Directives. 2016 *26th International Conference on Field Programmable Logic and Applications (FPL)*, Lausanne, 29 August-2 September 2016, 1-10. <https://doi.org/10.1109/FPL.2016.7577358>
- [5] Lo, C. and Chow, P. (2018) Multi-Fidelity Optimization for High-Level Synthesis Directives. 2018 *28th International Conference on Field Programmable Logic and Applications (FPL)*, Dublin, 27-31 August 2018, 272-2727. <https://doi.org/10.1109/FPL.2018.00054>
- [6] Makrani, H.M., et al. (2019) Pyramid: Machine Learning Framework to Estimate the Optimal Timing and Resource Usage of a High-Level Synthesis Design. 2019 *29th International Conference on Field Programmable Logic and Applications (FPL)*, Barcelona, 8-12 September 2019, 397-403. <https://doi.org/10.1109/FPL.2019.00069>

- [7] Hsiao, H. and Anderson, J.H. (2018) Sensei: An Area-Reduction Advisor for FPGA High-Level Synthesis. 2018 *Design, Automation & Test in Europe Conference & Exhibition (DATE)*, Dresden, 19-23 March 2018, 25-30. <https://doi.org/10.23919/DATE.2018.8341974>
- [8] Zhao, J., Feng, L., Sinha, S., Zhang, W., Liang, Y. and He, B. (2017) COMBA: A Comprehensive Model-Based Analysis Framework for High Level Synthesis of Real Applications. 2017 *IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, Irvine, 13-16 November 2017, 430-437. <https://doi.org/10.1109/ICCAD.2017.8203809>
- [9] Xydis, S., Palermo, G., Zaccaria, V. and Silvano, C. (2015) SPIRIT: Spectral-Aware Pareto Iterative Refinement Optimization for Supervised High-Level Synthesis. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, **34**, 155-159. <https://doi.org/10.1109/TCAD.2014.2363392>
- [10] Liu, S., Lau, F.C. and Schafer, B.C. (2019) Accelerating FPGA Prototyping through Predictive Model-Based HLS Design Space Exploration. 2019 *56th ACM/IEEE Design Automation Conference (DAC)*, Las Vegas, 2-6 June 2019, 1-6. <https://doi.org/10.1145/3316781.3317754>
- [11] Ferrandi, F., Lanzi, P.L., Loiacono, D., Pilato, C. and Sciuto, D. (2008) A Multi-Objective Genetic Algorithm for Design Space Exploration in High-Level Synthesis. 2008 *IEEE Computer Society Annual Symposium on VLSI*, Montpellier, 7-9 April 2008, 417-422. <https://doi.org/10.1109/ISVLSI.2008.73>
- [12] Schafer, B.C. and Wakabayashi, K. (2012) Machine Learning Predictive Modelling High-Level Synthesis Design Space Exploration. *IET Computers & Digital Techniques*, **6**, 153-159. <https://doi.org/10.1049/iet-cdt.2011.0115>
- [13] Liu, S., Lau, F. and Schafer, B.C. Predictive Compositional Method to Design and Re-Optimize Complex Behavioral Dataflows. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, **39**, 2615-2627.
- [14] Sarkar, P., Sengupta, A. and Naskar, M.K. (2015) GA Driven Integrated Exploration of Loop Unrolling Factor and Datapath for Optimal Scheduling of CDFGs during High Level Synthesis. 2015 *IEEE 28th Canadian Conference on Electrical and Computer Engineering (CCECE)*, Halifax, 3-6 May 2015, 75-80. <https://doi.org/10.1109/CCECE.2015.7129163>
- [15] Sengupta, A. and Bhadauria, S. (2015) User Power-Delay Budget Driven PSO Based Design Space Exploration of Optimal k-cycle Transient Fault Secured Datapath during High Level Synthesis. *Sixteenth International Symposium on Quality Electronic Design*, Santa Clara, 2-4 March 2015, 289-292. <https://doi.org/10.1109/ISQED.2015.7085441>
- [16] Sengupta, A. and Mishra, V.K. (2014) Integrated Particle Swarm Optimization (i-PSO): An Adaptive Design Space Exploration Framework for Power-Performance Tradeoff in Architectural Synthesis. *Fifteenth International Symposium on Quality Electronic Design*, Santa Clara, 3-5 March 2014, 60-67. <https://doi.org/10.1109/ISQED.2014.6783307>
- [17] Sengupta, A., Bhadauria, S. (2014) Automated Exploration of Data Path in High Level Synthesis Using Temperature Dependent Bacterial Foraging Optimization Algorithm. *IEEE 27th Canadian Conference on Electrical and Computer Engineering (CCECE)*, 4-7 May 2014, Toronto, 1-5. <https://doi.org/10.1109/CCECE.2014.6900920>
- [18] Mishra, V.K. and Sengupta, A. (2014) MO-PSE: Adaptive Multi-Objective Particle Swarm Optimization Based Design Space Exploration in Architectural Synthesis for Application Specific Processor Design. *Advances in Engineering Software*, **67**, 111-124. <https://doi.org/10.1016/j.advengsoft.2013.09.001>
- [19] Sengupta, A. and Mishra, V.K. (2014) Swarm Intelligence Driven Simultaneous Adaptive Exploration of Datapath and Loop Unrolling Factor during Area-Performance Tradeoff. *IEEE Computer Society Annual Symposium on VLSI*, Tampa, 9-11 July 2014, 106-111. <https://doi.org/10.1109/ISVLSI.2014.10>
- [20] Sengupta, A. and Bhadauria, S. (2014) Exploration of Multi-Objective Tradeoff during High Level Synthesis Using Bacterial Chemotaxis and Dispersal. *Procedia Computer Science*, **35**, 63-72. <https://doi.org/10.1016/j.procs.2014.08.085>
- [21] da Silva, J.S. and Bampi, S. (2015) Area-Oriented Iterative Method for Design Space Exploration with High-Level Synthesis. 2015 *IEEE 6th Latin American Symposium on Circuits & Systems (LASCAS)*, Montevideo, 24-27 February 2015, 1-4. <https://doi.org/10.1109/LASCAS.2015.7250447>
- [22] Schafer, B.C. (2016) Probabilistic Multiknob High-Level Synthesis Design Space Exploration Acceleration. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, **35**, 394-406. <https://doi.org/10.1109/TCAD.2015.2472007>
- [23] Schafer, B.C. (2015) Hierarchical High-Level Synthesis Design Space Exploration with Incremental Exploration Support. *IEEE Embedded Systems Letters*, **7**, 51-54. <https://doi.org/10.1109/LES.2015.2417216>
- [24] Gao, X.T., Wickerson, J. and Constantinides, G. (2016) Automatically Optimizing the Latency, Area, and Accuracy of C Programs for High-Level Synthesis. *Proceedings of the 2016 ACM/SIGDA International Symposium on Field-Programmable Gate Arrays*, Monterey, 21-23 February 2016, 234-243. <https://doi.org/10.1145/2847263.2847282>
- [25] Liu, L. and Kapre, N. (2019) Timing-Aware Routing in the RapidWright Framework. 2019 *29th International Conference on Field Programmable Logic and Applications (FPL)*, Barcelona, 8-12 September 2019, 24-30. <https://doi.org/10.1109/FPL.2019.00014>

- [26] Kapre, N., Ng, H., Teo, K. and Naude, J. (2015) InTime: A Machine Learning Approach for Efficient Selection of FPGA CAD Tool Parameters. *Proceedings of the 2015 ACM/SIGDA International Symposium on Field-Programmable Gate Arrays (FPGA '15)*, Monterey, 22-24 February 2015, 23-26. <https://doi.org/10.1145/2684746.2689081>
- [27] Kapre, N., Chandrashekar, B., Ng, H. and Teo, K. (2015) Driving Timing Convergence of FPGA Designs through Machine Learning and Cloud Computing. *2015 IEEE 23rd Annual International Symposium on Field-Programmable Custom Computing Machines*, Vancouver, 2-6 May 2015, 119-126. <https://doi.org/10.1109/FCCM.2015.36>
- [28] Que, Y.H., Ng, H. and Kapre, N. (2016) Boosting Convergence of Timing Closure Using Feature Selection in a Learning-Driven Approach. *2016 26th International Conference on Field Programmable Logic and Applications (FPL)*, Lausanne, 29 August-2 September 2016, 1-9. <https://doi.org/10.1109/FPL.2016.7577302>
- [29] Que, Y.H., Raj, C.A., Ng, H., Teo, K. and Kapre, N. (2016) Case for Design-Specific Machine Learning in Timing Closure of FPGA Designs. *Proceedings of the 2016 ACM/SIGDA International Symposium on Field-Programmable Gate Arrays (FPGA '16)*, Monterey, 21-23 February 2016, 169-172. <https://doi.org/10.1145/2847263.2847336>
- [30] Xu, C., Liu, G., Zhao, R., Yang, S., Luo, G.J. and Zhang, Z.R. (2017) A Parallel Bandit-Based Approach for Autotuning FPGA Compilation. *Proceedings of the 2017 ACM/SIGDA International Symposium on Field-Programmable Gate Arrays (FPGA '17)*, Monterey, 22-24 February 2017, 157-166.
- [31] Ustun, E., Xiang, S., Gui, J., Yu, C. and Zhang, Z. (2019) LAMDA: Learning-Assisted Multi-stage Autotuning for FPGA Design Closure. *2019 IEEE 27th Annual International Symposium on Field-Programmable Custom Computing Machines (FCCM)*, San Diego, 28 April-1 May 2019, 74-77. <https://doi.org/10.1109/FCCM.2019.00020>
- [32] Mаметjanov, A., Balaprakash, P., Choudary, C., Hovland, P.D., Wild, S.M. and Sabin, G. (2015) Autotuning FPGA Design Parameters for Performance and Power. *2015 IEEE 23rd Annual International Symposium on Field-Programmable Custom Computing Machines*, Vancouver, 2-6 May 2015, 84-91. <https://doi.org/10.1109/FCCM.2015.54>
- [33] Chen, T.S., Chen, Y.J., Guo, Q., Zhou, Z.H., Li, L. and Xu, Z.W. (2014) Effective and Efficient Microprocessor Design Space Exploration Using Unlabeled Design Configurations. *ACM Transactions on Intelligent Systems and Technology*, **5**, Article No. 20. <https://doi.org/10.1145/2542182.2542202>
- [34] Chen, T., et al. (2014) ArchRanker: A Ranking Approach to Design Space Exploration. *2014 ACM/IEEE 41st International Symposium on Computer Architecture (ISCA)*, Minneapolis, 14-18 June 2014, 85-96. <https://doi.org/10.1109/ISCA.2014.6853198>
- [35] Xie, Z., et al. (2018) RouteNet: Routability Prediction for Mixed-Size Designs Using Convolutional Neural Network. *2018 IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, San Diego, 5-8 November 2018, 1-8. <https://doi.org/10.1145/3240765.3240843>
- [36] Yu, C. and Zhang, Z. (2019) Painting on Placement: Forecasting Routing Congestion Using Conditional Generative Adversarial Nets. *2019 56th ACM/IEEE Design Automation Conference (DAC)*, Las Vegas, 2-6 June 2019, 1-6. <https://doi.org/10.1145/3316781.3317876>
- [37] Alawieh, M.B., Li, W., Lin, Y., Singhal, L., Iyer, M.A. and Pan, D.Z. (2020) High-Definition Routing Congestion Prediction for Large-Scale FPGAs. *2020 25th Asia and South Pacific Design Automation Conference (ASP-DAC)*, Beijing, 13-16 January 2020, 26-31. <https://doi.org/10.1109/ASP-DAC47756.2020.9045178>
- [38] Papamichael, M.K., Milder, P. and Hoe, J.C. (2015) Nautilus: Fast Automated IP Design Space Search Using Guided Genetic Algorithms. *2015 52nd ACM/EDAC/IEEE Design Automation Conference (DAC)*, San Francisco, 7-11 June 2015, 1-6. <https://doi.org/10.1145/2744769.2744875>
- [39] Wang, Z. and Schafer, B.C. (2019) Partial Encryption of Behavioral IPs to Selectively Control the Design Space in High-Level Synthesis. *2019 Design, Automation & Test in Europe Conference & Exhibition (DATE)*, Florence, 25-29 March 2019, 642-645. <https://doi.org/10.23919/DATE.2019.8714920>
- [40] Ma, Y., et al. (2019) High Performance Graph Convolutional Networks with Applications in Testability Analysis. *2019 56th ACM/IEEE Design Automation Conference (DAC)*, Las Vegas, 2-6 June 2019, 1-6. <https://doi.org/10.1145/3316781.3317838>
- [41] Ziegler, M.M., Liu, H., Gristede, G., Owens, B., Nigaglioni, R. and Carloni, L.P. (2016) A Synthesis-Parameter Tuning System for Autonomous Design-Space Exploration. *2016 Design, Automation & Test in Europe Conference & Exhibition (DATE)*, Dresden, 14-18 March 2016, 1148-1151. https://doi.org/10.3850/9783981537079_0690
- [42] Ziegler, M.M., Liu, H.Y. and Carloni, L.P. (2016) Scalable Auto-Tuning of Synthesis Parameters for Optimizing High-Performance Processors. *Proceedings of the 2016 International Symposium on Low Power Electronics and Design (ISLPED '16)*, San Francisco, 8-10 August 2016, 180-185. <https://doi.org/10.1145/2934583.2934620>
- [43] Anwar, M., Saha, S., Ziegler, M.M. and Reddy, L. (2016) Early Scenario Pruning for Efficient Design Space Exploration in Physical Synthesis. *2016 29th International Conference on VLSI Design and 2016 15th International Conference on Embedded Systems (VLSID)*, Kolkata, 4-8 January 2016, 116-121. <https://doi.org/10.1109/VLSID.2016.94>

-
- [44] Ziegler, M.M., Liu, H.Y., Gristede, G., Owens, B., Nigaglioni, R. and Carloni, L.P. (2016) A Scalable Black-Box Optimization System for Auto-Tuning VLSI Synthesis Programs. *Proceedings of 1st Workshop on Resource Awareness and Application Autotuning in Adaptive and Heterogeneous Computing*, Dresden, 18 March 2016, 8-12.
- [45] Kwon, J., Ziegler, M.M. and Carloni, L.P. (2019) A Learning-Based Recommender System for Autotuning Design Flows of Industrial High-Performance Processors. 2019 *56th ACM/IEEE Design Automation Conference (DAC)*, Las Vegas, 2-6 June 2019, 1-6. <https://doi.org/10.1145/3316781.3323919>
- [46] Palermo, G., Silvano, C. and Zaccaria, V. (2009) ReSPIR: A Response Surface-Based Pareto Iterative Refinement for Application-Specific Design Space Exploration. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, **28**, 1816-1829. <https://doi.org/10.1109/TCAD.2009.2028681>
- [47] Hara, Y., *et al.* (2009) Proposal and Quantitative Analysis of the CHStone Benchmark Program Suite for Practical C-Based High-Level Synthesis. *Journal of Information Processing*, **17**, 242-254. <https://doi.org/10.2197/ipsjjip.17.242>
- [48] Schafer, B.C. and Mahapatra, A. (2014) S2CBench: Synthesizable SystemC Benchmark Suite for High-Level Synthesis. *IEEE Embedded Systems Letters*, **6**, 53-56. <https://doi.org/10.1109/LES.2014.2320556>
- [49] Reagen, B., Adolf, R., Shao, Y.S., Wei, G.Y. and Brooks, D. (2014) MachSuite: Benchmarks for Accelerator Design and Customized Architectures. 2014 *IEEE International Symposium on Workload Characterization (IISWC)*, Raleigh, 26-28 October 2014, 110-119. <https://doi.org/10.1109/IISWC.2014.6983050>
- [50] Yuki, T. and Pouchet, L. (2016) PolyBench 4.2. <https://sourceforge.net/projects/polybench/>
- [51] Bustany, I.S., Chinnery, D., Shinnerl, J.R. and Yutsis, V. (2015) ISPD 2015 Benchmarks with Fence Regions and Routing Blockages for Detailed-Routing-Driven Placement. *Proceedings of the 2015 Symposium on International Symposium on Physical Design*, Monterey, 29 March-1 April 2015, 157-164. <https://doi.org/10.1145/2717764.2723572>