

超大规模集成电路布图规划方法研究综述

史梓慧, 欧阳丹彤, 张立明

(吉林大学 计算机科学与技术学院, 长春 130012;
吉林大学 符号计算与知识工程教育部重点实验室, 长春 130012)

摘要: 综述超大规模集成电路(VLSI)布图规划方法, 探讨布图规划在集成电路设计中的重要性, 以及其对芯片面积、互连线长和设计周期的影响。首先, 回顾集成电路技术的发展历程, 强调布图规划在确定模块位置、尺寸和旋转角度中的作用。其次, 详细介绍4类主要的VLSI布图规划方法: 直观构造方法、分析法、迭代法和基于机器学习的方法。再次, 讨论两个VLSI设计领域中常用的基准数据集MCNC和GSRC对测试和评估布图设计方法的重要性。最后, 总结布图规划领域的研究进展, 并指出未来的研究方向。

关键词: 超大规模集成电路; 布图规划; 布局; 构造法; 分析法; 迭代法; 机器学习方法

中图分类号: TP391; TN47 **文献标志码:** A **文章编号:** 1671-5489(2025)01-0139-12

Research Review of Floorplanning Methods for Very Large Scale Integration

SHI Zihui, OUYANG Dantong, ZHANG Liming

(College of Computer Science and Technology, Jilin University, Changchun 130012, China;
Key Laboratory of Symbolic Computation and Knowledge Engineering of Ministry of Education,
Jilin University, Changchun 130012, China)

Abstract: We review the floorplanning methods for very large scale integration (VLSI), explore the significance of floorplanning in integrated circuit design, and its impact on chip area, interconnect length, and design cycle. Firstly, we review the development history of integrated circuit technology, emphasize the role of floorplanning in determining the position, size, and rotation angles of modules. Secondly, we provide a detailed introduction to four main categories of VLSI floorplanning methods: intuitive construction methods, analytical methods, iterative methods and machine learning methods. Thirdly, we discuss two commonly used MCNC and GSRC benchmark datasets, which are crucial for testing and evaluating floorplanning methods in the VLSI design field. Finally, we summarize the research progress in the field of floorplanning and point out future research directions.

Keywords: very large scale integration; floorplanning; placement; construction method; analysis method; iterative method; machine learning method

集成电路技术的发展极大缩小了电子器件的尺寸, 降低了功耗, 提升了性能, 并简化了器件间的连接。在集成电路早期研究中, 主要采用小规模集成电路(SSI)。随着集成电路技术的进步和制造工艺

收稿日期: 2024-12-05.

第一作者简介: 史梓慧(2001—), 女, 汉族, 硕士研究生, 从事布图规划的研究, E-mail: shizh23@ mails. jlu. edu. cn. **通信作者简介:** 欧阳丹彤(1968—), 女, 满族, 博士, 教授, 博士生导师, 从事基于模型诊断和语义网的研究, E-mail: ouyd@jlu. edu. cn.

基金项目: 国家自然科学基金(批准号: 62076108; 61872159; 61672261).

的改进,集成度显著提升,从早期的小规模集成电路逐步过渡到中规模集成电路(MSI)和大规模集成电路(LSI),最终发展至超大规模集成电路(VLSI)^[1]. VLSI 技术使单个芯片上能集成数百万个晶体管甚至更多,极大增强了电子设备的性能和复杂性.这种集成度的飞跃不仅极大推动了计算机和电子设备性能的显著提升,也加速了信息技术的发展,为现代电子工程和计算机科学的进步奠定了坚实的基础.

布图规划是集成电路设计中的重要环节之一,其目的是确定模块在芯片版图中的位置、尺寸和旋转角度,从而生成一个合理的布图,最小化芯片的面积和模块间的互连线长^[2].一个好的布图可提升芯片的质量,还能缩短芯片的设计周期;而一个效率高、效果好的布图规划算法,可降低芯片的设计成本,提高生产效益,推动 VLSI 和电子设计自动化(EDA)相关产业发展.

从计算角度,布图规划是一个 NP-难组合优化问题^[3].目前,超大规模集成电路的布图规划和布局问题主要有 4 类方法:1) 直观构造方法,这类算法是使用分而治之策略的经典布图算法;2) 分析方法,将布图和布局问题转化为数学规划问题再求解;3) 迭代法;4) 基于机器学习的方法.在超大规模集成电路布图规划领域,迭代方法因其在寻找最优解过程中的高效性而受到广泛关注,其通常结合元启发式策略,如模拟退火、粒子群优化和遗传算法等,可在巨大的搜索空间中探索高质量的解决方案.而基于学习的方法以其速度快、泛化能力强等优势,同样在该领域展现出巨大的潜力.

1 基于构造的布图规划方法

在集成电路设计问题中,构造法以其简单直观的特性成为经典方法之一.该类算法通常采用分而治之的策略,通过直接的启发式规则或递归式分割生成初始布局.尽管构造算法在实现效率和初步布局生成方面性能优异,但也存在局限性,例如缺乏全局视野和深度优化能力.因此,许多研究在构造算法的基础上进行改进,以提高布局效果和适应性.下面介绍几种典型的构造算法,包括 Bottom-Left (BL)^[4]及其改进算法 Bottom-Left-Fill (BLF)^[5]、Bottom-Left-Decreasing (BLD)^[6]和 BLD*^[7],以及基于角点占据的 COA 算法,并分析其特点、应用场景及局限性.

1.1 BL 布图规划算法

BL 算法由 Baker 等^[4]提出,该算法将每个矩形物体尽可能地移动到左下角位置,具有简单直观、易实现、执行速度较快等特点.通过改变输入列表顺序,可显著影响算法的性能.例如,按宽度递减顺序排列矩形可显著改善包装效果,而按宽度递增排序方式则会导致效果不佳.该算法缺乏全局视野,无法有效优化复杂目标.对某些随机顺序的矩形排列性能较差,无法保证生成的解达到全局最优,适用于初始布局生成.

1.2 BLF 布图规划算法

BLF 算法^[5]在 BL 算法的基础上进行了改进,不仅将矩形物体移向左下角,还尝试填补已放置物体周围的空白区域.与 BL 算法相比,能更有效地填补空隙,从而实现更密集的填充.该算法的计算成本比 BL 算法高.但 BLF 算法能更有效地利用空间,因此在需要高效材料利用和精确布局的应用中有一定价值.

1.3 BLD 和 BLD* 布图规划算法

BLD 算法^[6]是一种用于二维矩形包装问题的启发式算法.该算法先将矩形按高度、宽度、面积和周长的降序进行排序,然后尝试将每个矩形放置当前可用的最底部和最左侧位置.这种方法通常返回 4 种排序中的最优包装结果.在电子设计自动化中,矩形打包用于布图规划,目的是最小化芯片面积,同时减少布线长度.尽管 BLD 算法在某些情况下表现良好,但它在随机排序的场景中性能通常不佳.

BLD* 算法^[7]是 BLD 算法的一种随机搜索改进版,引入了随机扰动原始排序的概念.使用随机扰动排序法从 BLD 的固定排序中生成排列,并评估其性能.通过随机扰动扩展搜索范围,有效避免陷入局部最优,可在较短时间内显著提高打包效果.相比传统的 BLD 算法,BLD* 算法在精度和灵活性上都有显著提升.

BLD* 算法虽然在优化问题中有一定的优势, 但其局限性也不容忽视. 首先, 由于需要多次尝试随机扰动排序, 该算法的计算复杂度显著提高, 导致在处理大规模数据集时产生较高的时间成本. 其次, 随机性虽然有助于提升搜索空间的覆盖率, 但也会引入性能的不稳定性, 使不同运行结果之间存在较大差异, 缺乏一致性. 此外, 该算法对初始排序质量依赖较高, 如果初始排序较差, 随机扰动的优化效果会受到限制, 进而降低算法的整体性能. 最后, 对一些规模较大的问题, 随机搜索策略无法有效覆盖全部优选解, 导致搜索效果的局限性. 因此, 在实际应用中, 应充分考虑这些限制并与具体问题需求相结合, 以优化算法的适用性和效率.

1.4 角点占据布图规划算法

角点占据(corner-occupying action, COA)算法^[8]是用于解决二维矩形打包问题的启发式算法, 基于“角点占据行为”和“凹陷度优化”两个核心概念设计, 旨在通过合理放置矩形提高空间利用率, 减少未利用的碎片空间. 应用场景包括制造业的钢板、玻璃等材料的切割优化, 通过减少材料浪费降低成本; 物流领域货物装载和仓储规划, 提高空间利用率; 超大规模集成电路设计中的模块布局, 布局紧凑和高效利用空间.

COA 算法的核心思想是通过优先占据角点^[9]放置矩形, 同时利用凹陷度指标对放置位置进行优化. 角点是指由已放置矩形或容器边界形成的交点, 这些点通常是打包中潜在的最佳位置, 因为占据这些位置可显著减少剩余空隙, 从而提高打包密度. 凹陷度则用来衡量矩形与周围环境的接近程度, 凹陷度值越高, 说明放置后的矩形与周围矩形贴合越紧密, 打包方案越紧凑.

角点占据行为是该算法的基础, 通过优先将待放置矩形安排在由已放置矩形或容器边界形成的角点位置, 显著减少空隙, 提高打包密度. 角点占据能最大限度地利用现有空间, 使矩形尽可能靠近边界或其他矩形, 从而形成紧凑的布局. 此外, 当矩形不仅占据角点, 还与其他已放置矩形直接接触时, 这种放置被称为洞穴占据行为. 洞穴占据进一步优化了矩形与周围矩形的贴合程度, 减少了未利用空间, 是提高空间利用率的重要策略.

凹陷度则是 COA 算法中的另一个关键指标, 用于衡量矩形放置的紧密程度. 凹陷度大小反映了矩形与其邻近矩形或边界的接触程度, 值越高, 说明矩形放置越紧凑. 凹陷度 C_i 的计算通过评估矩形与周围环境的距离实现:

$$C_i = 1 - \frac{d_i}{\sqrt{w_i \cdot h_i}}, \quad (1)$$

其中 d_i 为待放置矩形与最近邻矩形或边界的距离, w_i 为待放置矩形的宽, h_i 为待放置矩形的高. 通过凹陷度的优化, 算法能优先选择使打包密度最大的角点位置进行放置. 此外, 凹陷度还结合了矩形的形状特性, 通过考虑长宽比等因素进一步提升算法的选择性.

在具体实现中, COA 算法通过一套优先级规则指导放置决策. 在所有可行的角点中, 首先计算每个角点与所有候选矩形的凹陷度, 并选择凹陷度最大的角点作为目标位置; 若多个角点凹陷度相同, 则优先选择能提升长宽比的矩形放置方案; 若仍有多种选择, 则根据矩形排列顺序最终决定放置位置. 这一基于凹陷度和优先级规则的决策过程, 使 COA 算法能在打包过程中不断优化空间利用率. 通过角点占据与凹陷度优化的结合, COA 算法为高效解决二维矩形打包问题提供了一种可靠且简洁的策略.

COA 算法能显著提高空间利用率, 减少未利用的碎片空间, 同时算法设计简洁高效, 计算复杂度不高, 便于实现. 该算法能优先选择空间利用率最高的放置方式, 从而在局部最优驱动下实现较紧凑的全局布局. 但 COA 算法也存在一定局限性, 其基于局部优化的特性可能导致陷入局部最优解, 错失全局最优方案; 同时, 算法对初始角点和矩形排列顺序较敏感, 初始条件选择不当会影响最终结果. 此外, 在问题规模显著扩大时, 计算凹陷度和更新角点列表的开销会随之增加, 导致运行效率下降. 这些因素在一定程度上限制了 COA 算法在超大规模打包问题中的性能, 但并不削弱其在小到中规模问题中的有效性和实用价值.

2 基于分析的布图规划方法

分析方法通过将布局问题转化为具有线性约束的数学规划问题,如二次规划问题,这类方法提供了一种高效且有确定性的解决方案,将模块的位置坐标作为自变量,根据面积、线长等优化目标建立线性或非线性(不)等式方程组进行求解,最终获得各模块在版图中的最优位置.这类算法因其确定性,对特定布局问题会生成一致的结果. mPL6 (multilevel placement version 6)^[10], APlace^[11] 和 NTUplace3^[12] 是此类分析方法的经典代表,下面介绍这 3 种算法的工作原理、特点、应用场景、优势和局限性.

2.1 多层次非线性编程布局方法 mPL6

多层次非线性编程的混合尺寸布局 mPL6 方法^[10] 是一种适用于复杂集成电路布局的高效算法,其工作流程包括全局布局、合法化和详细布局 3 个阶段.

在全局布局阶段,算法基于多层次优化思想,通过最佳选择聚类法递归地聚合与细化问题规模,并在每一层中采用力导向布局平衡线长与密度,确保模块分布均匀.在合法化阶段,利用约束图方法处理宏观模块的重叠问题,通过构建水平和垂直邻接图调整模块位置,同时通过动态权重调整优化关键路径,降低最终布线长度.详细布局阶段则采用窗口交换算法,在局部区域内对模块位置进行微调,通过最佳排列进一步减少布线长度.其核心特点是多层次优化增强了算法的可扩展性和鲁棒性,提出了密度敏感的布局方法,能逐步确定大型模块的位置,从而降低合法化阶段的计算复杂度,并在优化线长和密度平衡的同时满足复杂设计约束.

该方法尤其适用于高精度要求的混合尺寸设计,在低白空间或高密度场景下表现尤为出色.但其算法复杂度较高,在处理超大规模设计时耗时较长.

2.2 基于分析优化的布局方法 APlace

APlace^[11] 是一种基于分析优化的通用布局工具,旨在通过数学分析技术实现高效的集成电路布局.这是一种通过优化线长、密度和平衡设计约束的数学分析技术实现高效布局的工具,具有高灵活性和广泛的适用性,其工作流程包括全局布局、密度与布线优化以及特殊问题优化 3 个阶段.

在全局布局阶段, APlace 采用平滑的线长函数如 $\log\text{-sum-exp}$ ^[13-14] 和多层次框架,将问题分解为若干层次的松弛优化问题,并通过结合梯度下降方法进行求解,同时动态调整平滑参数以加速收敛.在密度与布线优化中,提出混合使用不同线长与密度函数的策略,结合改进的惩罚函数控制密度约束,有效平衡布局密度和线长优化.针对特殊问题,如电源退化等, APlace 通过重新分布电路单元优化供电稳定性,并结合模拟和时序分析进行位置感知的布局优化.其核心特点是灵活性强,可扩展至时序驱动、混合尺寸设计等不同场景,采用动态密度调整策略在速度和质量之间取得良好平衡.

APlace 适用于需要处理多样化约束的布局问题,以及对速度要求较高的中等规模问题设计.尽管其速度较早期版本有所提升,但在超大规模问题中仍落后于专门的快速工具.

2.3 大规模混合尺寸布局方法 NTUplace3

NTUplace3^[12] 是一种高效的大规模混合尺寸布局工具,专为现代集成电路设计中的复杂约束问题而开发.其核心思想是基于 $\log\text{-sum-exp}$ 函数的线长模型,通过动态调整密度分布和平滑技术,在优化线长、密度均衡和模块位置合法性之间实现良好平衡.

NTUplace3 的布局流程包括全局布局、合法化和详细布局 3 个阶段:在全局布局中,采用 $\log\text{-sum-exp}$ 模型结合两阶段平滑技术控制模块密度,并通过动态可用空间分配减少拥塞;在合法化阶段,使用基于邻接图的模块调整策略,同时引入宏块平移和预见合法化技术,确保无重叠布局;在详细布局阶段,通过单元匹配和滑动优化技术进一步优化线长和局部密度分布.

在性能方面, NTUplace3 展现了强大的大规模设计处理能力,能在包含大量模块和预放置块的复杂场景中实现高效布局.此外,共轭梯度法结合动态步长调整的优化方法显著加快了全局布局的收敛速度,显著提升了大规模设计的效率.但对包含大量预放置块的设计,其计算复杂度增加,且在时序或温度等多目标优化任务中适用性有限. NTUplace3 凭借其先进的数学模型和优化技术,为混合尺寸

和高密度集成电路设计提供了有效的解决方案。

3 基于迭代的布图规划方法

迭代法求解 VLSI 布图规划和布局问题时, 首先需要将布图编码。根据平面图是否可沿一条直线进行一次或多次水平或垂直切割, 将其分为可切片平面结构和不可切片平面结构^[15-16], 二者均需通过数据结构进行编码。对于可切片平面结构, 常采用波兰表示法处理^[17]; 对于不可切片平面结构, 可采用序列对(sequence pair)^[18]、角块列表(corner block list)^[19]、有界切片网格(bounded slicing grid, BSG)^[20-21]、传递闭包图(transitive closure graph, TCG)^[22-23]及其改进版本 TCG-S^[24]、O-Tree^[15]和 B*-Tree^[25-27]等表示方法。然后需要选择一个元启发式搜索的框架, 例如遗传算法^[28]、模拟退火算法^[29-30]、蚁群算法^[31]、粒子群优化算法^[32]、模因算法^[33-34]和混合算法^[35-37]等。

3.1 基于遗传算法的布图规划方法

遗传算法(genetic algorithm, GA)是根据大自然中生物体进化规律而设计的, 是以达尔文生物进化论的自然选择和遗传学理论为基础模拟生物进化过程的计算模型, 通过模拟自然进化过程搜索最优解。该算法将问题的求解过程转换成类似生物进化中染色体基因的交叉、变异等过程。

算法首先随机生成一组解构成初始种群, 然后根据平面图的各种参数, 如块的高度、模块的数量等, 计算出适应度函数, 根据适应度选择优良个体作为父代, 通过交叉操作生成子代以产生新的解, 并对子代进行随机变异操作, 以增加种群的多样性, 从而避免陷入局部最优解。最后, 通过不断更新种群并重复该过程, 直到满足停止条件, 从而获得优化结果。

遗传算法在布局设计中的交叉算子常存在不能有效继承父代优良特性的问题。因此, Lin 等^[38]提出了改进遗传算法(GFA)用于优化可切片结构的布局面积, 该算法融合了切片树编码方案的特性和遗传算法的进化机制。GFA 算法用二叉树结构表示可切片结构, 并提出了一种新的遗传算子, 该算子在算法中始终继承父代的优良特性, 从而有效探索解空间。算法首先随机生成字符串组成种群, 计算适应度, 然后进入包括交叉和变异操作的进化阶段, 设置交叉率大于变异率, 若在给定时间内无改进则调整阈值, 当满足停止条件时算法结束。该方法能避免遗传算法在布局设计中交叉算子不能有效继承父代优良特性的问题, 弥补了传统进化算法在布局设计上的不足。

3.2 基于模拟退火算法的布图规划方法

模拟退火算法来源于固体退火原理, 是一种基于概率的算法, 将固体升温至充分高, 再将其慢慢冷却, 升温时, 固体内部粒子随升温变为无序状, 内能增大, 而慢慢冷却时粒子渐趋有序, 在每个温度都达到平衡态, 最后在常温时达到基态, 内能减为最小。模拟退火算法从某一较高初始温度开始, 伴随温度参数的不断下降, 结合概率特性在解空间中随机寻找目标函数的全局最优解, 即在局部最优解能概率性地跳出并最终趋于全局最优。模拟退火算法的主要缺点是它给出了近似最优解, 但不能保证稳定^[39]。模拟退火算法是一种通用的优化算法, 理论上算法具有概率的全局优化性能, 目前已在工程中得到了广泛应用, 如 VLSI^[40]和信号处理等领域。

Chang 等^[25]提出了 B*-Tree 的布图表示法并使用模拟退火算法求解, 用很短的时间得到了较优的布图解。Chen 等^[29]提出了一种基于 B*-Tree 表示的快速三阶段模拟退火 FastSA 算法, 并对面积和长度进行了优化。FastSA 算法可以通过动态调整成本函数中的权值优化线长, 三阶段温度控制可以有效避免陷入局部最优。但由于 FastSA 算法在每一步中只能获得一个邻居解, 因此无法尽可能多地探索搜索空间。

3.3 基于蚁群算法的布图规划方法

蚁群算法是一种基于自然界蚂蚁行为模式的优化算法^[41], 体现了了一种正反馈机制: 选择某条路

径的蚂蚁越多,这条路径对其他蚂蚁就越有吸引力,即更倾向于选择吸引力高的路径。

Hoo 等^[31]提出了一种针对 VLSI 多目标布局优化的变量顺序蚁群系统(VOAS)。该系统使用角列表模型^[38]表示芯片布局,并通过 VOAS 蚂蚁和侦查蚂蚁两组蚂蚁进行合作,优化芯片区域和连线长度。侦查蚂蚁搜集局部信息,而 VOAS 蚂蚁则根据局部和全局信息决定模块的放置。这种变量顺序属性使蚂蚁在选择模块位置时能根据它们在布局过程中的不同阶段优先考虑全局或局部信息。尽管 VOAS 通过在路径选择中使用变量顺序属性和修改信息素更新规则改进传统的蚁群算法,但在算法的早期阶段,所有蚂蚁会跟随相同的路径并构建相同的解决方案,从而导致产生次优解。

3.4 基于粒子群优化的布图规划方法

粒子群优化(particle swarm optimisation, PSO)算法是一种群体智能算法,属于元启发式算法的一种。灵感来源于鸟群觅食和鱼群游动的社会行为^[42]。粒子群优化算法中每个个体称为粒子,粒子组成群体。粒子群优化算法从随机粒子开始,每个粒子在解空间内以一定速度飞行,并根据自身和邻居的经验进行更新。粒子群优化算法在处理连续优化问题时特别有效,常被用于各种工程和科学问题的求解,如函数最小化和其他多目标优化问题等。

Chen 等^[43]提出了一种基于粒子群优化的算法处理 VLSI 布图规划问题,采用基于模块数的整数编码和新的加速度系数推荐值对粒子群优化算法进行优化布置。受遗传算法物理特性的启发,将遗传算法中的突变算子和交叉算子的原理融入到 PSO 算法中,使该算法摆脱了局部最优,实现了较好的多样性。由于各变量之间相互影响,而变量的赋值通常是主观的,将导致赋值困难。因此,为避免这些缺点,未来的工作可侧重于提高算法的效率。

3.5 基于模因算法的布图规划方法

模因算法(memetic algorithm, MA)^[44]是一种结合了群体智能和局部搜索策略的优化技术。它的核心思想是模拟自然和文化进化过程,通过全局搜索和局部精细搜索相结合,以提高搜索效率和解的质量。模因算法一般是进化算法和局部搜索的结合。进化算法用于寻找全局最优,局部搜索有助于进化计算的收敛速度。

Tang 等^[33]引入了一种基于 O-tree 表示的模因算法 MA,通过结合混合遗传算法和新的偏差搜索策略,虽然在面积成本上有所优化,但在面积与线长之间的权衡上存在局限。Shanavas 等^[45]提出的算法结合了遗传算法等分层设计技术和模拟退火等构造技术进行局部搜索,以解决 VLSI 的分区和布局问题。

3.6 基于混合算法的布图规划方法

Chen 等^[46]提出了一种混合遗传算法(hybrid genetic algorithm, HGA),该算法采用有效的遗传搜索方法对搜索空间进行探索,并采用有效的局部搜索方法对搜索区域内的信息进行挖掘。Chen 等^[30]提出了一种采用 B*-Tree 表示的混合模拟退火算法(HSA)优化布图规划的面积和总长度。HSA 有 3 个主要思想:第一个思想是一种新的贪心方法构造初始 B*-Tree;第二个思想是通过交换 B*-Tree 中的两个子树探索搜索空间的新操作;第三个思想是采用一种新的搜索策略,用于平衡全局搜索和局部搜索,但在搜索空间的探索上仅采用交换两个子树的方法仍不够充分,缺乏多样性。Sivaranjani 等^[37]提出了一种基于粒子群优化和遗传算法的混合算法,其结合了粒子群优化算法、萤火虫算法(FF)和修正角表算法(MCL)的混合粒子群优化-萤火虫算法(HPSOFF)。最初,粒子群优化算法利用 MCL 算法进行非切片平面图表示和适应度值评估,由粒子群优化算法得到的解作为 FF 算法的初始解,再使用 MCL 算法对 FF 算法进行适应度函数评估和平面图表示,通过从粒子群优化子系统与遗传算法子系统之间交换一些个体,两个子系统共享布局信息并共同进化。文献[36]提出了智能决

策混合粒子群优化-遗传算法 PGHA, 通过均匀分布温度在芯片上减少面积、带宽. 该算法首先利用 B^* -Tree 生成初始平面图, 然后利用基于 PSO-GA 的混合算法得到最优布局方案. 在控制温度方面效果良好. Xu 等^[47]提出了一种结合蚁群算法和模拟退火算法的两阶段方法处理固定轮廓布图问题. 通过采用序列对表示, 两阶段算法可以很容易地应用于二维布图规划问题, 并可获得很好的布线效果. 受 MA 方法的启发, Chen 等^[34]在 HSA 工作的基础上, 提出了一种自适应混合模拟退火算法 AHMA 解决该问题, 选择遗传算法作为全局搜索方法, 改进的模拟退火算法作为局部搜索方法, 并采用死亡概率策略平衡全局和局部搜索. 尽管在全局和局部搜索上取得了平衡, 但其复杂的交叉操作和静态成本函数限制了其多目标优化能力.

在多目标优化方面, Srinivasan 等^[48]提出了基于多目标萤火虫优化的布图规划技术 MOFO-FP, 针对固定轮廓约束问题, 利用能量和热感知萤火虫优化(EHAFO)算法进行优化. 在萤火虫优化算法中, 为在超大规模集成电路设计中进行高效的布局规划, 计算了发热量、占用空间和导线长度等多目标函数. 根据目标函数, 计算出每只萤火虫的光强, 将低亮度的萤火虫向高亮度的萤火虫移动, 以确定最佳位置. 最后, 根据萤火虫的强度对其进行排序, 找到最优的 VLSI 平面规划树结构. MOFO-FP 技术减少了电路占用的空间, 并降低了产生的热量, 同时缩短了导线长度. MOFO-FP 具有灵活性, 可以进一步根据特殊目标或更多目标进行调整和扩展, 以适应不同的布局规划需求. 但在设计大型电路时会出现时间复杂度问题, 随着电路规模增大, 算法执行所需时间也会显著增加, 影响算法的效率和实用性.

Jiang 等^[49]提出了 DPAHMA 算法, 是一种基于 AHMA 算法的改进算法. 通过引入候选种群机制、动态自适应目标函数和改进的遗传算子, 进一步提高算法的性能和搜索能力, 以更有效地处理非切片 VLSI 布图规划问题. 自适应目标函数能找到更适合用户指定权重的解; 全局搜索阶段引入候选种群辅助正常种群的主辅种群机制. 为充分利用局部搜索方法获得的信息, 候选种群保留其高质量的解. 在每次迭代结束前, 候选种群中的个体与正常种群中的个体交叉, 以尽可能获得更高质量的解.

4 基于机器学习的布图规划方法

在超大规模集成电路布图规划领域, 机器学习方法展现出独特的优势和潜力, 其中基于深度学习的布图规划方法已取得了一定成果. 而除深度学习外, 强化学习同样在布图规划中有重要的应用与探索, 其通过智能体与环境的交互寻求最优策略, 为布图规划提供了新的思路和方法. 此外, 将深度学习与强化学习相结合的布图规划方法在近年研究中也备受关注.

4.1 基于深度学习的布图规划方法

深度学习是基于人工神经网络的机器学习技术, 其通过构建多层神经网络模型, 使计算机从海量数据中自动学习特征与模式. 其优势显著, 具有强大的自动特征学习能力, 具有准确性高和可扩展性佳的特点, 可处理大规模数据与复杂模型结构, 能随数据量与计算资源提升而优化性能, 在多领域都有广泛应用.

Lin 等^[50]提出了称为 DREAMPlace 的 GPU 加速布局框架, 通过将解析布局问题等效转换为训练神经网络实现. 该框架基于深度学习, 将解析布局问题类比为神经网络训练问题, 把单元位置视为权重, 线长成本对应预测误差, 密度成本对应正则化项. 在实现过程中, 从随机初始布局开始, 通过高效的全局布局迭代, 结合 Nesterov 方法^[51]优化, 在收敛后进行合法化操作, 最后依赖外部工具进行详细布局, 从而实现了在全局布局和合法化过程中的显著加速且保持质量, 展现出良好稳定性和扩展性. 该框架在超大规模集成电路布局中具有重要意义, 能显著提高布局速度并保持质量. 但 DREAMPlace 的详细布局仍依赖于外部框架, 若外部工具未进行 GPU 加速或与 DREAMPlace 的集成不佳, 则将无

法充分发挥 GPU 加速的潜力,影响整个布局流程的效率.此外,该方法效率受硬件性能影响,设备配置不足时无法达到最佳效果.

4.2 基于强化学习的布图规划方法

强化学习是一种机器学习范式,智能体在环境中采取一系列行动,根据环境反馈的奖励信号学习最优的行为策略.智能体的目标是最大化长期累积奖励,环境会在智能体行动后返回一个奖励信号,用于评估行动的好坏.智能体在与环境的交互过程中通过学习策略达成回报最大化或实现特定目标.

He 等^[52]提出了一种方法,旨在利用强化学习获取有效局部搜索启发式算法,从而解决布图规划问题.该方法将布图规划中的局部搜索问题转化为强化学习问题,详细定义了包含状态、动作、转移和奖励的 Markov 决策过程.选取能量相关统计信息、扰动影响和搜索进度等合适特征,构建以多层感知器为策略近似器的神经网络作为智能体,利用 DQN(deep Q-network)算法应对大动作空间带来的挑战,实现智能体的有效训练.该方法具有在智能体搜索过程更平滑、生成布图死区空间更小等显著优势.但该方法的训练样本是随机生成的网表,无法保证完全覆盖实际电路布局中的各种复杂情况,导致训练出的模型在面对某些特定类型或极端情况下的电路布局时,泛化能力不足.

4.3 基于深度强化学习的布图规划方法

深度强化学习是一种机器学习方法,它将深度学习的感知能力与强化学习的决策能力相结合.智能体通过与布局环境的交互学习最优策略.

Yu 等^[53]提出了一种基于序列对的深度强化学习布局规划算法,该算法利用序列对表示模块间的几何关系,序列对由正负序列组成,能有效描述模块间的相对位置关系.在强化学习框架中,状态空间定义为包含完整门序列及模块方向的布局方案,动作空间通过预定义的 5 种扰动操作生成相邻布局方案,状态转移模型根据扰动使智能体在不同状态间转换,奖励函数则根据布局面积和线长的优化情况给予智能体相应奖励,以激励其寻找最优布局.采用基于 Actor-Critic 架构的策略梯度算法训练智能体,借助深度学习学习参数,根据奖励反馈优化策略,其训练过程遵循特定步骤,并设置了如折扣因子、学习率等超参数.该算法在布局面积和线长优化等方面性能优异,尤其在大规模电路中优势更明显.但目前该算法存在优化时间长的问题亟待解决.

VLSI 是芯片实现的关键技术,而芯片则是 VLSI 技术的物理载体,二者相互促进、共同发展.在芯片布局设计领域, Mirhoseini 等^[54]设计了基于深度强化学习的布局设计方法,该方法将芯片布局问题定义为一个强化学习问题,采用一种新的基于边的图卷积神经网络(edge-based graph convolutional neural network, Edge-GNN)架构学习芯片的丰富表示.在该方法中,芯片布局被视为一个顺序的决策过程,包括状态、动作、状态转移和奖励 4 个关键要素.状态包含网表信息、节点信息等;动作是宏模块的合法放置位置.通过预测奖励训练网络,然后将其作为策略网络的编码器,以实现跨芯片的泛化.此外,该方法使用近端策略优化(PPO)算法^[55]更新策略网络的参数,以最大化累积奖励.在训练过程中,强化学习代理依次放置宏模块,然后使用力导向方法放置标准单元集群.该方法在满足设计标准方面均表现优秀,且在面积、功耗和线长等指标上与手动布局相当或更优,目前已应用于谷歌 TPU 设计.但该方法对计算资源要求很高,在资源受限的环境下不适用.

5 常用数据集

基准数据集 MCNC (<http://vlsicad.eecs.umich.edu/BK/MCNCbench/>) 和 GSRC (<http://vlsicad.eecs.umich.edu/BK/GSRCbench/>) 是超大规模集成电路设计领域中常用的两个关键数据集,用于测试和评估布图设计算法.布图设计是 VLSI 设计中的关键步骤,其中不同功能块的芯片在硅片上进行布局,以优化面积和互连长度等多种目标.

5.1 基准数据集 MCNC

基准数据集 MCNC 广泛用于评估 VLSI CAD 工具. 这些基准数据集包括来自 MCNC 研究项目的各种实际芯片布局的集合. 芯片布局从小到大, 涵盖了广泛的设计类型和复杂性.

5.2 基准数据集 GSRC

基准数据集 GSRC 为研究人员和设计师提供一组标准化的测试集, 帮助他们评估其版图设计方法和工具的效率和有效性. 这些基准数据集包括各种合成和现实世界的问题, 代表了 VLSI 设计的不同方面, 从简单到复杂的电路布局. 其用于模拟设计时可面临的不同场景, 测试算法在各种复杂度和设计约束条件下的性能.

数据集 GSRC 和 MCNC 内各电路属性列于表 1. 通过提供标准化测试, 这些基准数据集有助于促进版图设计技术和 VLSI 设计方法的进步, 从而推动更高效、更紧凑、功耗更低芯片的开发.

表 1 数据集 MCNC 和 GSRC 内各电路的属性

Table 1 Attributes of each circuit in MCNC and GSRC datasets

电路	来源	模块数	线网数	引脚数	端子数	总面积/mm ²
apte	MCNC	9	97	287	73	46.561 628
xerox	MCNC	10	203	698	2	19.350 296
hp	MCNC	11	83	309	45	8.830 584
ami33	MCNC	33	123	522	42	1.156 449
ami49	MCNC	49	408	953	22	35.445 424
n10	GSRC	10	118	248	69	0.221 679
n30	GSRC	30	349	723	212	0.208 591
n50	GSRC	50	485	1 050	209	0.198 579
n100	GSRC	100	885	1 873	334	0.179 501
n200	GSRC	200	1 585	3 599	564	0.175 696
n300	GSRC	300	1 893	4 358	569	0.273 170

6 未来研究方向

在超大规模集成电路布图规划领域, 目前研究主要集中在开发各种算法以优化布局的质量和效率. 研究涵盖了直观构造方法、分析方法、迭代方法及基于机器学习的方法. 由近年的研究成果可见, 迭代方法和基于机器学习的方法成为研究重点, 迭代方法在寻找全局最优解方面表现出强大的能力, 机器学习方法则在处理复杂数据和特征学习方面具有优势. 综合本文的研究进展介绍及分析, 未来研究可以围绕以下几个方面进行.

多目标优化: 当前的超大规模集成电路布图规划方法大多数只考虑面积和线长这两个优化目标, 除面积和线长这两个关键指标外, 温度、功耗等指标也是集成电路设计中影响芯片性能和可靠性的重要因素. 温度可以影响芯片的性能稳定性和寿命, 功耗是衡量其能效的关键指标, 尤其是在移动设备和数据中心的应用中尤为重要. 因此未来研究可以开发出能同时考虑面积、线长、温度和功耗等多个目标的布图规划算法, 以满足现代集成电路设计中日益复杂的需求.

更高效的算法开发: 虽然目前已有算法在超大规模集成电路布图规划中取得了一定成效, 但随着集成电路设计复杂性的增加, 需要更高效的算法应对更复杂的布局挑战. 当开发需要基于以往方法进行改进时, 需关注可供使用的计算资源条件, 若计算资源不足, 可考虑基于迭代方法的算法展开优化和改进.

算法融合: 未来研究可以探索将多种布图规划方法进行融合, 以提高算法的效率和优化质量, 进一步优化现有的布图规划算法, 例如结合机器学习技术与传统的迭代法, 以发挥各自的优势, 提升布图规划的效率和效果.

新型数据集和评估方法: 尽管基准数据集 GSRC 和 MCNC 包含了对芯片块的详细描述, 并被广

泛用于 VLSI 布图规划等问题的测试和评估,但超大规模集成电路的实际设计具有多样性,现有的基准数据集和评估方法无法保证完全反映实际设计中的复杂情况。未来可以开发新的数据集和评估方法,以更好地测试和评估布图规划方法的效果和性能。

参 考 文 献

- [1] KAHNG A B, LIENIG J, MARKOV I L, et al. Chip Planning [C]//VLSI Physical Design: From Graph Partitioning to Timing Closure. Berlin: Springer International Publishing, 2022: 53-93.
- [2] RINI D P, SHAMSUDDIN S. Minimization of Floorplanning Area and Wire Length Interconnection Using Particle Swarm Optimization [EB/OL]. (2013-08-01)[2024-11-20]. <https://www.researchgate.net/publication/326804232>.
- [3] SINGH R B, BAGHEL A S, AGARWAL A. A Review on VLSI Floorplanning Optimization Using Metaheuristic Algorithms [C]//2016 International Conference on Electrical, Electronics, and Optimization Techniques (ICEEOT). Piscataway, NJ: IEEE, 2016: 4198-4202.
- [4] BAKER B S, COFFMAN E G, Jr, RIVEST R L. Orthogonal Packings in Two Dimensions [J]. SIAM Journal on Computing, 1980, 9(4): 846-855.
- [5] CHAZELLE B. The Bottomn-Left Bin-Packing Heuristic: An Efficient Implementation [J]. IEEE Transactions on Computers, 1983, 32(8): 697-707.
- [6] LESH N, MARKS J, McMAHON A, et al. New Heuristic and Interactive Approaches to 2D Rectangular Strip Packing [J]. ACM Journal of Experimental Algorithmics, 2005, 10: 12-1-12-18.
- [7] HOPPER E, TURTON B C H. An Empirical Investigation of Meta-Heuristic and Heuristic Algorithms for a 2D Packing Problem [J]. European Journal of Operational Research, 2001, 128(1): 34-57.
- [8] HUANG W Q, CHEN D B, XU R C. A New Heuristic Algorithm for Rectangle Packing [J]. Computers & Operations Research, 2007, 34(11): 3270-3280.
- [9] WU Y L, HUANG W, LAU S C, et al. An Effective Quasi-human Based Heuristic for Solving the Rectangle Packing Problem [J]. European Journal of Operational Research, 2002, 141(2): 341-358.
- [10] CHAN T F, CONG J, SHINNERL J R, et al. mPL6: Enhanced Multilevel Mixed-Size Placement [C]//Proceedings of the 2006 International Symposium on Physical Design. New York: ACM, 2006: 212-214.
- [11] KAHNG A B, WANG Q K. A Faster Implementation of APlace [C]//Proceedings of the 2006 International Symposium on Physical Design. New York: ACM, 2006: 218-220.
- [12] CHEN T C, JIANG Z W, HSU T C, et al. NTUplace3: An Analytical Placer for Large-Scale Mixed-Size Designs with Preplaced Blocks and Density Constraints [J]. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 2008, 27(7): 1228-1240.
- [13] NAYLOR W C, DONNELLY R, SHA L. Non-linear Optimization System and Method for Wire Length and Delay Optimization for an Automatic Electric Circuit Placer: US, 6301693B1 [P]. 2001-01-24.
- [14] KAHNG A B, WANG Q K. Implementation and Extensibility of an Analytic Placer [C]//Proceedings of the 2004 International Symposium on Physical Design. New York: ACM, 2004: 18-25.
- [15] GUO P N, CHENG C K, YOSHIMURA T. An O-Tree Representation of Non-slicing Floorplan and Its Applications [C]//Proceedings 1999 Design Automation Conference. Piscataway, NJ: IEEE, 1999: 268-273.
- [16] JAIN L, SINGH A. Non Slicing Floorplan Representations in VLSI Floorplanning: A Summary [J]. International Journal of Computer Applications, 2013, 71(15): 12-20.
- [17] WONG D F, LIU C L. A New Algorithm for Floorplan Design [C]//23rd ACM/IEEE Design Automation Conference. New York: ACM, 1986: 101-107.
- [18] MURATA H, FUJIYOSHI K, NAKATAKE S, et al. VLSI Module Placement Based on Rectangle-Packing by the Sequence-Pair [J]. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 1996, 15(12): 1518-1524.
- [19] HONG X L, HUANG G, CAI Y C, et al. Corner Block List: An Effective and Efficient Topological

- Representation of Non-slicing Floorplan [C]//IEEE/ACM International Conference on Computer Aided Design. Piscataway, NJ: IEEE, 2000: 8-12.
- [20] NAKATAKE S, FUJIYOSHI K, MURATA H, et al. Module Placement on BSG-Structure and IC Layout Applications [C]//Proceedings of International Conference on Computer Aided Design. Piscataway, NJ: IEEE, 1996: 484-491.
- [21] KANG M, DAI W W M. General Floorplanning with L-Shaped, T-Shaped and Soft Blocks Based on Bounded Slicing Grid Structure [C]//Proceedings of Asia and South Pacific Design Automation Conference. Piscataway, NJ: IEEE, 1997: 265-270.
- [22] LIN J M, CHANG Y W. TCG: A Transitive Closure Graph-Based Representation for General Floorplans [J]. IEEE Transactions on Very Large Scale Integration (VLSI) Systems, 2005, 13(2): 288-292.
- [23] LIN J M, CHANG Y W. TCG: A Transitive Closure Graph-Based Representation for Non-slicing Floorplans [C]//Proceedings of the 38th Annual Design Automation Conference. New York: ACM, 2001: 764-769.
- [24] LIN J M, CHANG Y W. TCG-S: Orthogonal Coupling of P^* -Admissible Representations for General Floorplans [C]//Proceedings of the 39th Annual Design Automation Conference. New York: ACM, 2002: 842-847.
- [25] CHANG Y C, CHANG Y W, WU G M, et al. B^* -Trees: A New Representation for Non-slicing Floorplans [C]//Proceedings of the 37th Annual Design Automation Conference. New York: ACM, 2000: 458-463.
- [26] SUN T Y, HSIEH S T, WANG H M, et al. Floorplanning Based on Particle Swarm Optimization [C]//IEEE Computer Society Annual Symposium on Emerging VLSI Technologies and Architectures (ISVLSI'06). Piscataway, NJ: IEEE, 2006: 1-5.
- [27] LIN J M, HUNG Z X. SKB-Tree: A Fixed-Outline Driven Representation for Modern Floorplanning Problems [J]. IEEE Transactions on Very Large Scale Integration (VLSI) Systems, 2012, 20(3): 473-484.
- [28] LIN C T, CHEN D S, WANG Y W. An Efficient Genetic Algorithm for Slicing Floorplan Area Optimization [C]//2002 IEEE International Symposium on Circuits and Systems (ISCAS). Piscataway, NJ: IEEE, 2002: 879-882.
- [29] CHEN T C, CHANG Y W. Modern Floorplanning Based on B^* -Tree and Fast Simulated Annealing [J]. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 2006, 25(4): 637-650.
- [30] CHEN J L, ZHU W X, ALI M M. A Hybrid Simulated Annealing Algorithm for Nonslicing VLSI Floorplanning [J]. IEEE Transactions on Systems, Man, and Cybernetics, 2011, 41(4): 544-553.
- [31] HOO C S, JEEVAN K, GANAPATHY V, et al. Variable-Order Ant System for VLSI Multiobjective Floorplanning [J]. Applied Soft Computing, 2013, 13(7): 3285-3297.
- [32] 陈锦珠. 适用于求解 VLSI 布图规划问题的多目标 PSO 算法研究 [D]. 福州: 福州大学, 2011. (CHEN J Z. Research on Multi-objective PSO Algorithm for Solving VLSI Floorplanning Problem [D]. Fuzhou: Fuzhou University, 2011.)
- [33] TANG M L, YAO X. A Memetic Algorithm for VLSI Floorplanning [J]. IEEE Transactions on Systems, Man, and Cybernetics, 2007, 37(1): 62-69.
- [34] CHEN J L, LIU Y, ZHU Z R, et al. An Adaptive Hybrid Memetic Algorithm for Thermal-Aware Non-slicing VLSI Floorplanning [J]. Integration, 2017, 58: 245-252.
- [35] 陈建利, 朱文兴. 求解 VLSI 不可二划分布图规划问题的混合遗传算法 [J]. 福州大学学报(自然科学版), 2014, 42(5): 688-693. (CHEN J L, ZHU W X. Hybrid Genetic Algorithm for Solving the Non-slicing Floorplan Problem in VLSI [J]. Journal of Fuzhou University (Natural Science Edition), 2014, 42(5): 688-693.)
- [36] SIVARANJANI P, SENTHIL K A. Thermal-Aware Non-slicing VLSI Floorplanning Using a Smart Decision-Making PSO-GA Based Hybrid Algorithm [J]. Circuits, Systems, and Signal Processing, 2015, 34(11): 3521-3542.
- [37] SIVARANJANI P, SENTHIL K A. Hybrid Particle Swarm Optimization-Firefly Algorithm (HPSOFF) for Combinatorial Optimization of Non-slicing VLSI Floorplanning [J]. Journal of Intelligent & Fuzzy Systems, 2017, 32(1): 661-669.

- [38] LIN F, HE G M. An Improved Genetic Algorithm for Multi-objective Optimization [C]//Sixth International Conference on Parallel and Distributed Computing Applications and Technologies (PDCAT'05). Piscataway, NJ: IEEE, 2005: 938-940.
- [39] KIYOTA K, FUJIYOSHI K. Simulated Annealing Search through General Structure Floorplans Using Sequence-Pair [C]//2000 IEEE International Symposium on Circuits and Systems (ISCAS). Piscataway, NJ: IEEE, 2000: 77-80.
- [40] 姚新, 陈国良. 模拟退火算法及其应用 [J]. 计算机研究与发展, 1990(7): 1-6. (YAO X, CHEN G L. Simulated Annealing Algorithm and Its Applications [J]. Journal of Computer Research and Development, 1990(7): 1-6.)
- [41] COLORNI A, DORIGO M, MANIEZZO V. Distributed Optimization by Ant Colonies [C]//Proceedings of the First European Conference on Artificial Life. New York: ACM, 1990: 1-12.
- [42] KENNEDY J, EBERHART R. Particle Swarm Optimization [C]//Proceedings of International Conference on Neural Networks. Piscataway, NJ: IEEE, 1995: 1942-1948.
- [43] CHEN G L, GUO W Z, CHEN Y Z. A PSO-Based Intelligent Decision Algorithm for VLSI Floorplanning [J]. Soft Computing, 2010, 14(12): 1329-1337.
- [44] KRASNOGOR N, SMITH J. A Tutorial for Competent Memetic Algorithms: Model, Taxonomy, and Design Issues [J]. IEEE Transactions on Evolutionary Computation, 2005, 9(5): 474-488.
- [45] SHANAVAS I H, GNANAMURTHY R K. Wirelength Minimization in Partitioning and Floorplanning Using Evolutionary Algorithms [J]. VLSI Design, 2011, 2011: 10-1-10-10.
- [46] CHEN J L, ZHU W X. A Hybrid Genetic Algorithm for VLSI Floorplanning [C]//2010 IEEE International Conference on Intelligent Computing and Intelligent Systems. Piscataway, NJ: IEEE, 2010: 128-132.
- [47] XU Q, CHEN S, LI B. Combining the Ant System Algorithm and Simulated Annealing for 3D/2D Fixed-Outline Floorplanning [J]. Applied Soft Computing, 2016, 40: 150-160.
- [48] SRINIVASAN B, VENKATESAN R. Multi-objective Optimization for Energy and Heat-Aware VLSI Floorplanning Using Enhanced Firefly Optimization [J]. Soft Computing, 2021, 25(5): 4159-4174.
- [49] JIANG L Y, OUYANG D T, ZHOU H S, et al. DPAHMA: A Novel Dual-population Adaptive Hybrid Memetic Algorithm for Non-slicing VLSI Floorplans [J]. The Journal of Supercomputing, 2023, 79(14): 15496-15534.
- [50] LIN Y B, DHAR S, LI W X, et al. DREAMPlace: Deep Learning Toolkit-Enabled GPU Acceleration for Modern VLSI Placement [C]//Proceedings of the 56th Annual Design Automation Conference 2019. New York: ACM, 2019: 1-6.
- [51] LU J W, CHEN P W, CHANG C C, et al. ePlace: Electrostatics-Based Placement Using Fast Fourier Transform and Nesterov's Method [J]. ACM Transactions on Design Automation of Electronic Systems, 2015, 20(2): 17-1-17-34.
- [52] HE Z L, MA Y Z, ZHANG L, et al. Learn to Floorplan through Acquisition of Effective Local Search Heuristics [C]//2020 IEEE 38th International Conference on Computer Design (ICCD). Piscataway, NJ: IEEE, 2020: 324-331.
- [53] YU S L, DU S M. VLSI Floorplanning Algorithm Based on Reinforcement Learning with Obstacles [C]//Biologically Inspired Cognitive Architectures 2023. Berlin: Springer, 2024: 1034-1043.
- [54] MIRHOSEINI A, GOLDIE A, YAZGAN M, et al. A Graph Placement Methodology for Fast Chip Design [J]. Nature, 2021, 594(7862): 207-212.
- [55] SCHULMAN J, WOLSKI F, DHARIWAL P, et al. Proximal Policy Optimization Algorithms [EB/OL]. (2017-07-20)[2024-11-25]. <https://arxiv.org/abs/1707.06347>.