

# 芯片智能布线方法研究综述

周展文<sup>1</sup> 卓汉逵<sup>1</sup>

**摘要** 布线是芯片设计自动化流程中至关重要且特别耗时的一环,直接影响最终产品的面积、成本、功耗、速度和可靠性,研究智能布线算法对提高芯片布线效率和优化芯片布线效果具有重要意义.芯片布线问题是一个多目标、多约束的 NP 困难问题.即使已有几十年的研究历史,目前仍存在大量未突破的问题和空间.随着制造工艺的不断发展,布线规则、约束和目标也持续调整和增加,使得布线选择极其困难.因此,对芯片设计自动化中自动布线的前沿研究进行了全面归纳与分析,以帮助科研人员全面了解该领域的研究进展和方向,助力智能布线算法的研究和发展.具体而言,首先阐述芯片布线的问题背景,然后分别介绍全局布线(Global routing, GR)和详细布线(Detailed routing, DR)的任务定义和目标、过程特点、难点和挑战、评估方法;接着详述和分析各布线方法,重点论述基于规划搜索的布线方法和基于机器学习的布线方法的最新研究成果、优缺点及其应用环节;然后介绍公开数据集和开源布线工具;最后总结现有方法在实际应用中存在的局限性,并对自动布线未来的发展趋势和潜在研究方向进行展望.

**关键词** 机器学习, 智能规划, 强化学习, 芯片, 布线, 全局布线, 详细布线

**引用格式** 周展文, 卓汉逵. 芯片智能布线方法研究综述. 自动化学报, 2024, 50(9): 1671-1703

**DOI** 10.16383/j.aas.c230368

## Survey on Intelligent Routing Approaches for Chips

ZHOU Zhan-Wen<sup>1</sup> ZHUO Han-Kui<sup>1</sup>

**Abstract** Routing is a crucial and particularly time-consuming part of the chip design automation process, directly affecting the area, cost, power consumption, speed and reliability of the final product. It is of great significance to study intelligent routing algorithms to improve the efficiency and effectiveness of optimizing chip routing. The chip routing problem is a NP-hard problem with multiple objectives and constraints. Even with decades of research history, there are still a large number of problems and spaces for breakthrough. As manufacturing techniques evolve, design rules, constraints and objectives are continuously adjusted and added, which make wiring selection extremely difficult. This survey aims to provide a comprehensive overview and analysis of the cutting-edge research on automated routing in the chip design automation, in order to help researchers fully understand the research progresses and directions in this field, and to accelerate the research and development of intelligent routing algorithms. Specifically, this paper first describes the background of chip routing, then introduces the definitions and objectives, process characteristics, difficulties and challenges, and evaluation methods of global and detailed routing respectively, followed by a detailed description and analysis of various routing methods, focusing on the latest research results, advantages and disadvantages of planning and search-based routing methods and machine learning-based routing approaches and their application aspects, then introduces public data sets and open-source routing tools, and finally summarizes the limitations of existing methods in practical applications and provides an outlook on future trends and potential research directions for automated routing.

**Key words** Machine learning, AI planning, reinforcement learning, chip, routing, global routing (GR), detailed routing (DR)

**Citation** Zhou Zhan-Wen, Zhuo Han-Kui. Survey on intelligent routing approaches for chips. *Acta Automatica Sinica*, 2024, 50(9): 1671-1703

芯片, 是信息产业的核心部件. 一颗芯片从无到有、从市场需求到最终应用, 大致可以分为设计、制造、封装、测试四个主要环节. 其中, 芯片设计是

将系统、逻辑和性能的设计要求转化为具体物理版图的过程. 芯片设计周期长、过程复杂, 对设计人员专业程度要求高, 往往需要借助电子设计自动化(Electronic design automation, EDA)工具完成. 在芯片设计流程里, 通常只有在最后的物理验证、签核和测试阶段, 才能准确衡量设计在功率、性能和面积(Power, performance and area, PPA)方面的质量. 因此, 在设计的前期阶段对 PPA 的准确估

收稿日期 2023-06-15 录用日期 2023-11-09

Manuscript received June 15, 2023; accepted November 9, 2023

本文责任编辑 段书凯

Recommended by Associate Editor DUAN Shu-Kai

1. 中山大学计算机学院 广州 510006

1. School of Computer Science and Engineering, Sun Yat-sen University, Guangzhou 510006

算可以减少设计迭代次数, 增加设计的可靠性, 提高最终芯片的质量. 其中, 布线是芯片设计自动化流程中至关重要也是特别耗时的一环, 能否按照设计需求将所有线网布通是决定芯片能否流片的首要前提, 布线结果的好坏直接影响最终产品的面积、成本、功耗、速度和可靠性.

芯片布线问题是一个多约束、多目标和多任务的 NP 困难问题, 即使对于相对简单的布线问题, 如斯坦纳最小树问题 (Steiner minimum tree problem, SMTP), 即在没有障碍物和设计规则约束的情况下为层中一个包含多个终端 (Terminal) 的线网 (Net) 寻找连接所有端点的最短路线, 不同于最小生成树 (Minimum spanning tree, MST) 问题可以有  $O(n \log n)$  时间复杂度算法, SMTP 问题如果允许通过任意角度连线, 则为 NP-hard<sup>[1]</sup>; 如果仅允许水平和垂直连线, 则为 NP-complete<sup>[2]</sup>. 除此之外, 布线过程还需要解决拥塞避让、线网排序、引脚连接点分析、轨道分配、设计规则检查 (Design rule checking, DRC) 等各种困难问题.

因布线问题复杂、规模庞大, 布线过程一般分为全局布线 (Global routing, GR) 和详细布线 (Detailed routing, DR)<sup>[3-4]</sup>. 全局布线做全局规划生成布线指导, 在保证关键路径时延前提下, 分散走线, 避免局部拥塞, 从而提高详细布线的成功率; 详细布线在布线指导指引下完成具体走线, 满足设计规则检查, 并使总线长尽可能短, 通孔尽可能少. 虽然布线研究已有几十年历史, 产生了迷宫布线 (Maze routing)、通道布线 (Channel routing)、四边通道布线 (Switchbox routing)、拥塞协商布线 (Negotiated-congestion routing)、并行布线 (Parallel routing)、基于学习的布线 (Learning-based routing) 等大量非常智能且有效的布线算法, 推动着 EDA 自动布线技术的发展. 但现有算法中仍然存在大量可研究问题和未突破空间, 如庞大布线搜索空间下布线时长与布线质量的优化与平衡、线网选取与实际走线之间的微妙关系、布线经验沉淀与数据利用等, 都具有相当大的难度和非常高的研究价值. 随着信息技术和制造工艺的不断发展, 芯片晶体管密度不断提高, 各类应用对芯片的要求也越来越高, 使得布线规则、约束和目标不断调整和增加: 压缩体积、降低功耗、消除拥塞、优化时序、减少耦合效应、消除串扰、保证信号完整性等, 这一长串经常相互冲突的目标使布线选择极其困难. 因此, 研究智能布线算法对提高芯片布线效率和优化芯片布线效果具有重要意义, 部分研究问题的解决方案甚至能够推

广到其他组合优化问题上. 在技术应用上, 研究智能布线算法可沉淀布线知识和经验, 提高布线质量和产品可靠性, 加快芯片和电子产品的上市速度. 特别是在硅材料芯片因工艺制程极限约束逐步往多层架构方向发展, 以及如何将人工智能 (Artificial intelligence, AI) 应用于 AI 芯片设计并加快产品产量上, 智能布线都具有非常高的研究意义和应用价值. 从国家战略上, EDA 工具是芯片设计、制造、封装、测试等阶段的必备工具, 在当前复杂多变的国际国内形势下, 必须拥有一批自主可控的 EDA 工具与企业, 以避免在芯片设计环节被“卡脖子”, 已经成为国家和行业共识. 研究智能布线算法和技术可弥补我国在 EDA 工具研发上的部分空白, 且从布线这一芯片后端设计的核心环节上进行突破能加速竞争力的形成.

目前国内外的有关综述<sup>[5-7]</sup>大多探讨机器学习 (Machine learning, ML) 方法在 EDA 流程各环节中的应用. 部分综述<sup>[8-9]</sup>同时探讨超大规模集成电路 (Very large scale integration, VLSI) 的布局和布线方法, 尚无针对芯片布线全面而细致的综述文献. 文献 [10] 只综述了 VLSI 全局布线和斯坦纳树构建的主要挑战和相关基于规划搜索的布线算法. 文献 [11] 将布线问题抽象为图搜索问题或多商品流问题, 分别介绍了基于迷宫或 A\* 搜索、整数线性规划以及并行加速技术的 VLSI 详细布线算法的最新研究进展, 但缺少对全局布线的探讨, 对机器学习类布线算法的研究和分析亦不够深入. 文献 [12] 仅从布线违例预测、布线优化、智能布线三个维度综述了机器学习类算法在 VLSI 布线中的应用案例, 缺少从全局布线和详细布线的特点和环节出发归纳机器学习算法的应用框架, 缺乏对传统布线工具和机器学习算法的差异对比和融合方式的思考. 文献 [13] 则专注于近期 VLSI 布线中的挑战和布线方法, 包括 GPU (Graphics processing unit) 加速布线、信号速度优化、印制电路板 (Printed circuit board, PCB) 布线和 AI 驱动的模拟电路布线. 本文对芯片设计自动化中布线方法进行综述, 旨在帮助研究人员对该领域的任务设置、主流方法、评价体系、开源工具等方面有系统性的了解, 便于快速进入该领域. 本文关注各布线方法间的共性与不同, 按照问题导向逐步介绍其发展脉络, 着重介绍规划搜索类算法在 VLSI 和现场可编程门阵列 (Field programmable gate array, FPGA) 布线的最新研究成果, 剖析机器学习类算法在元件布局优化、全局布线和详细布线各环节的框架设计、应用

效果、与传统算法的融合思路, 以及现有开源环境的支撑, 并结合领域研究现状对未来发展趋势进行展望, 希望帮助研究者们开拓思路, 助力智能布线算法的研究和发展。

本文主要内容的框架如图 1 所示。

本文后续内容组织如下: 第 1 节首先介绍全局布线和详细布线的问题背景和定义, 分析研究难点与挑战, 然后介绍芯片智能布线方法的发展历程; 第 2 节首先简述基础的单线网和多线网布线方法, 接着面向 VLSI 布线和 FPGA 布线分别介绍基于规划搜索的智能布线方法的最新研究成果; 然后在第 3 节着重讲述近年主流的基于机器学习的智能布线方法, 按照方法的应用环节和设计原理进行归纳与分析; 第 4 节介绍常用的公开数据集与开源布线工具; 第 5 节分析了智能布线研究的现状与局限性, 并对未来发展趋势与潜在研究方向进行展望; 第 6 节是文章的总结与展望。

### 1 问题背景与分析

芯片布线通常在布局完成之后进行, 布局已将各种模块 (Block)、标准单元 (Standard cell) 和输入输出接口单元 (I/O port) 等电路组件安置在芯片上, 布线则对这些组件之间的互连线资源进行合理配置, 按电路逻辑关系进行互连, 保证不同组件之间的连接畅通, 并为满足时序预算 (Time bud-

get)、计规则、布线层数、线宽、线间距限制, 及各种物理、电气、制造约束进行优化。布线可能会因未连接应连接的终端 (开路, Open), 错误连接两个不应连接的终端 (短路, Short), 或设计规则违例 (Design rule violation, DRV) 而失败。芯片布线时, 为满足基本的间距要求, 一般把布线区域划分为固定宽度的布线场所 (Pitch), 场所的中间为走线轨道 (Track), 轨道方向为优选走线方向 (Preferred direction), 多层芯片布线时为尽可能较少信号串扰, 相邻层之间走线方向成正交, 交叉位置可通过通孔 (Via) 连接上下两金属层 (Metal layer), 走线时要避开障碍物 (Obstacle)。如图 2 所示, 在一个三层的布线空间中, 线网包含三个引脚 (Pin) (A, B, C), 橙色走线是其中一种布线连接方案。但在真实的芯片中, 需要布线的线网可能多达百万个。

芯片布线就是根据网表 (Netlist)、关键线网的时序预算、宏单元 (Macro) 和标准单元及引脚的位置等输入信息, 在满足时序要求且不违反设计规则前提下连接所有线网, 并最小化总线长、总通孔数、芯片面积, 最后输出所有线网的具体走线及通孔位置和类型。但多个需要同时优化的目标、千差万别的布线环境、庞大的线网数量、复杂的设计规则和约束、不规范的 3D 网格以及多种形状的通孔选择等都给芯片布线带来非常大的困难和挑战。

鉴于芯片布线问题复杂, 解空间过于庞大, 布

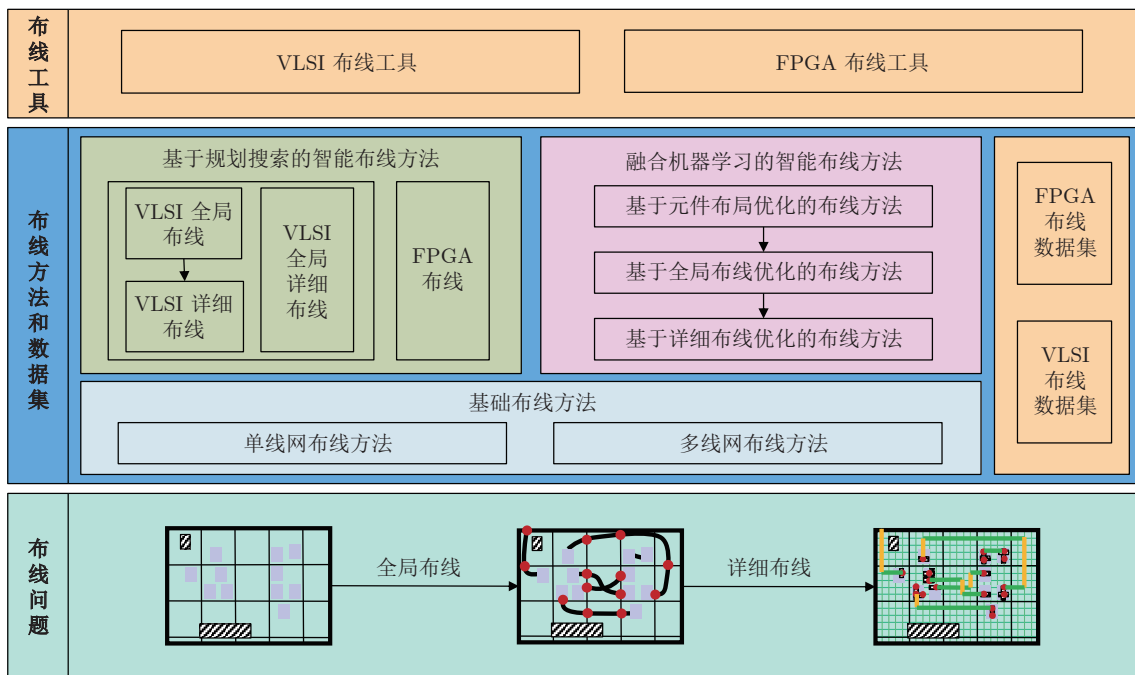


图 1 本文主要内容框架

Fig.1 The main content framework of this paper

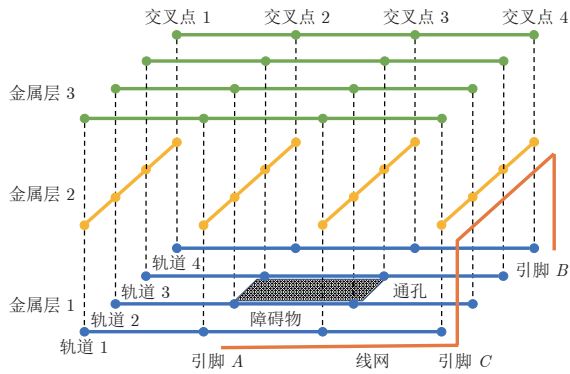


图 2 3D 网格空间与线网布线

Fig.2 3D grid space and net routing

线过程一般分为全局布线和详细布线两个阶段, 这样能够避免拥塞, 优化电路性能, 确保最终所有线网的总线长较低. 当前也有不少论文<sup>[14-15]</sup> 研究用统一布线算法同时完成全局布线和详细布线, 以降低评估误差, 获得更好的布线结果. 但这类方法容易随着电路复杂度的增加变得越发复杂和不可扩展, 因此在大规模电路布线中较少使用.

### 1.1 全局布线

全局布线是指对整个芯片的走线做全局规划, 以便确定即使是很大的芯片设计, 它的最终布线计划也能实现<sup>[16]</sup>. 它的主要目标是保证关键路径延时最小, 分散布线避免引起局部拥塞, 总线长评估尽量短. 全局布线是一种“松散”的布线, 它将芯片预先划分成若干布线区域, 每个区域包含纵横方向上的多条走线轨道, 这样布线时可以同时规划多个走线方案, 加速布线并尽早报告结果, 以便根据拥塞情况进行调整或修改. 全局布线一般分为三个步骤, 如图 3 所示.

1) 区域划分. 按照某种策略将芯片分成若干个方块区域 (Global cell, GCell), 除非处于芯片边界位置, GCell 通常是固定大小的, 相邻 GCell 包含多条连接两个区域的走线轨道. 区域划分的方式和大

小设定至关重要. 区域太小会导致巨大的全局布线空间, 需要更多时间完成全局布线; 区域太大则无法及早暴露拥塞问题, 将布线压力转移到详细布线阶段.

2) 区域分配. 对于跨区域的线网, 按照特定策略选择合适的区域连接线网的所有引脚, 不同线网可以共享连接的区域, 但要尽量分散连接, 避免拥塞 (需求量达到容量的一定百分比), 同时降低总线长估算.

3) 边界接合点分配. 按照特定策略给每个经过区域的线网在区域边界分配接合点, 以用于指导详细布线.

全局布线速度快、时间短, 能快速收敛, 它为用时很长的详细布线做好规划. 目前商用芯片布局布线工具的全局布线运行时间通常约为详细布线运行时间的 10% 或更少. 如果全局布线时发现问题, 设计人员可以及时调整解决, 而不必再花费很长时间去做详细布线及其他后续工作. 虽然全局布线降低了问题求解复杂性, 但全局布线无法看到区域内的障碍物、预布线线网等细节信息, 只能使用一个比较粗的模型来表示区域内的布线资源使用情况, 可能给出错误的估算, 并影响全局布线线路的选择. 在 FPGA 布线时, 全局布线因无法了解各区域中的不同类型的线段和可编程开关的具体分布, 可能给详细布线输出错误指导.

全局布线问题的定义如下: 给定布局结果、芯片技术定义和设计规则, 给定一个网表  $\mathcal{N} = \{N_1, N_2, \dots, N_n\}$ , 布线图  $G = (V, E)$ , 点  $v \in V$  表示区域, 边  $e \in E$  表示相邻两个区域  $(v_i, v_j)$  的连接. 对每个线网  $N_i$ ,  $1 \leq i \leq n$ , 找到一棵斯坦纳树  $T_i$ , 使得对于所有  $e_j \in E$ ,  $d(e_j) \leq c(e_j)$ , 最小化  $\sum_{i=1}^n L(T_i)$ . 其中  $d(e_j) = \sum_{i=1}^n x_{ij}$ , 如果  $e_j$  存在于  $T_i$  中, 则  $x_{ij} = 1$ , 否则  $x_{ij} = 0$ , 表示连线经过边  $e_j$  的线网数量;  $c(e_j)$  是边  $e_j$  的容量;  $L(T_i)$  表示斯坦纳树  $T_i$  的拥塞加权长度.

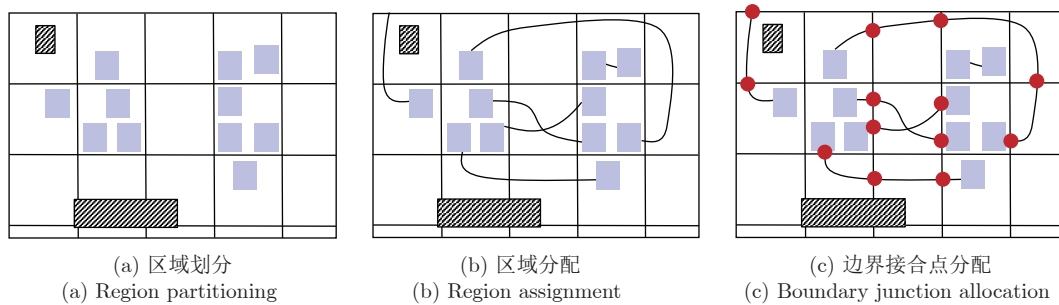


图 3 全局布线的三个步骤

Fig.3 Three steps of global routing



评估全局布线算法优劣的主要指标包括: 1) 总溢出, 即超出给定容量的布线需求量; 2) 最大溢出; 3) 拥塞线网数量; 4) 拥塞加权总线长; 5) 布线时长。

## 1.2 详细布线

详细布线又称最终布线, 它是物理实施的最后步骤, 它所实现的结果将作为参数提取和时序分析<sup>[16]</sup>。相对于全局布线, 详细布线事实上是一种局部布线, 它在全局布线结果指引下, 按生产工艺要求和设计规则约束, 将线网的连线片段分配到具体的金属层轨道和通孔上, 并不断优化总连线长度、通孔数等目标。详细布线一般分为四个步骤, 如图 4 所示。

1) 引脚连接点分析。基于引脚的轮廓, 结合引脚所在层及上层轨道的位置和空间间隔要求, 分析并确定用于连接引脚的网格点和连接方向。

2) 轨道分配和布线。按照特定策略选择轨道片段或者通孔, 串行或并行地完成所有线网的布线连接, 线网之间不得交叉出现短路, 尽可能减少连线长度和通孔数。

3) 设计规则检查。根据设计规则全面检查布线结果, 找出其中的违例位置、类型以及涉及的线网。设计规则包括各种间距 (Spacing) 约束、最小面积

(Min-area) 约束, 以及各种物理、电气、制造约束如串扰、光学邻近效应修正等。间距约束指定了两个对象之间所需的间距, 常见的间距约束包括并行走线长度 (Parallel run length) 间距、线末 (End-of-line, EOL) 间距、邻近切口 (Adjacent cut) 间距和角到角 (Corner-to-corner) 间距, 如图 5 所示。图 5(a) 中的并行走线长度 (即它们之间的投影长度) 间距定义了两个并行运行的金属物体之间的间距要求。线末指的是比给定线宽  $eolWidth$  短的边缘位置, 线末间距要求在线末以外给定区间  $eolWithin$  范围内的任何其他对象与线末的距离不小于给定间距  $eolSpace$ , 如图 5(b) 所示。图 5(c) 中邻近切口间距指定了在同一线网或不同线网上小于  $cutWithin$  距离的通孔之间允许的最小间距。角到角间距指定凸角和任何边之间所需的间距, 当两个对象之间的并行运行长度小于等于 0 时触发, 如图 5(d) 所示。

4) 违例修复。利用拆线重布 (Rip-up and reroute, RRR)、推挤、打补丁线等方式修复设计规则违例。

为满足各种设计规则约束, 详细布线必须记录所有元件的位置、形状、空间距离, 以及其他线网的走线路径, 加上庞大的布线轨道和多层的布线空间, 使得详细布线成为芯片布线过程中最复杂最耗时的

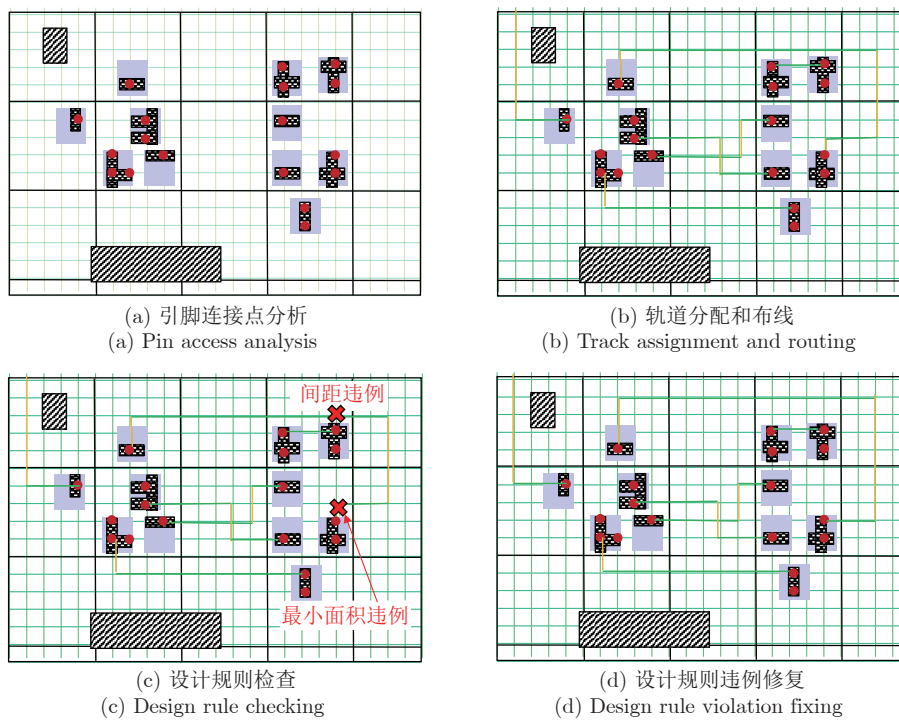


图 4 详细布线的四个步骤  
Fig.4 Four steps of detailed routing

阶段, 通常需要经过多轮迭代才能获得比较好的布线效果.

详细布线问题的定义如下: 给定芯片布局结果、全局布线指导、芯片技术定义和设计规则, 给定一个网表  $\mathcal{N} = \{N_1, N_2, \dots, N_n\}$ , 布线图  $G = (V, E)$ , 点  $v \in V$  表示相邻层轨道交叉点或者引脚连接点, 边  $e \in E$  表示  $G$  中相邻两个交叉点  $(v_i, v_j)$  的连接, 可以是同一轨道或者引脚连接点与轨道的连接片段, 或是连接相邻两层的通孔. 对每个线网  $N_i, 1 \leq i \leq n$ , 找到一棵直线斯坦纳树  $T_i$ , 使得所有线网的斯坦纳树所经过的点和边均不重叠, 最小化  $\sum_{i=1}^n C(T_i)$ , 其中  $C(T_i)$  为斯坦纳树  $T_i$  的成本, 是总长度、通孔数、设计规则违例数量的一种加权.

评估详细布线算法优劣的主要指标包括: 1) 总线长; 2) 总通孔数; 3) 设计规则违例数量; 4) 时序

要求满足情况; 5) 布线计算与内存峰值资源占用; 6) 布线时长.

### 1.3 芯片智能布线方法

鉴于芯片布线问题的规模、复杂度和对高端芯片及高效 EDA 工具的需求, 智能布线一直是芯片设计领域最引人注目的研究课题之一. 经过几十年的布线研究, 产生了迷宫布线、通道布线、拥塞协商布线、并行布线、基于机器学习的布线等针对特定场景和需求的智能布线算法, 这些布线方法推动着 EDA 智能布线技术的持续发展. 芯片智能布线方法的发展历程如图 6 所示. 从图 6 可以发现, 自 1961 年到 2000 年左右, 支撑芯片全局和详细布线的基础布线方法得到充分的研究. 随着 ISPD-2007<sup>[17]</sup> 和 ISPD-2008<sup>[18]</sup> 全局布线比赛及基准测试程序的

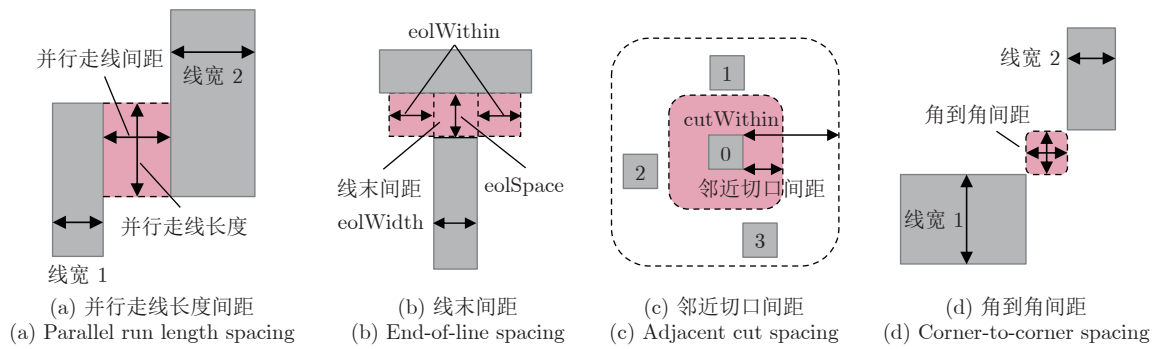


图 5 常见的设计间距约束

Fig.5 Representative design spacing constrains

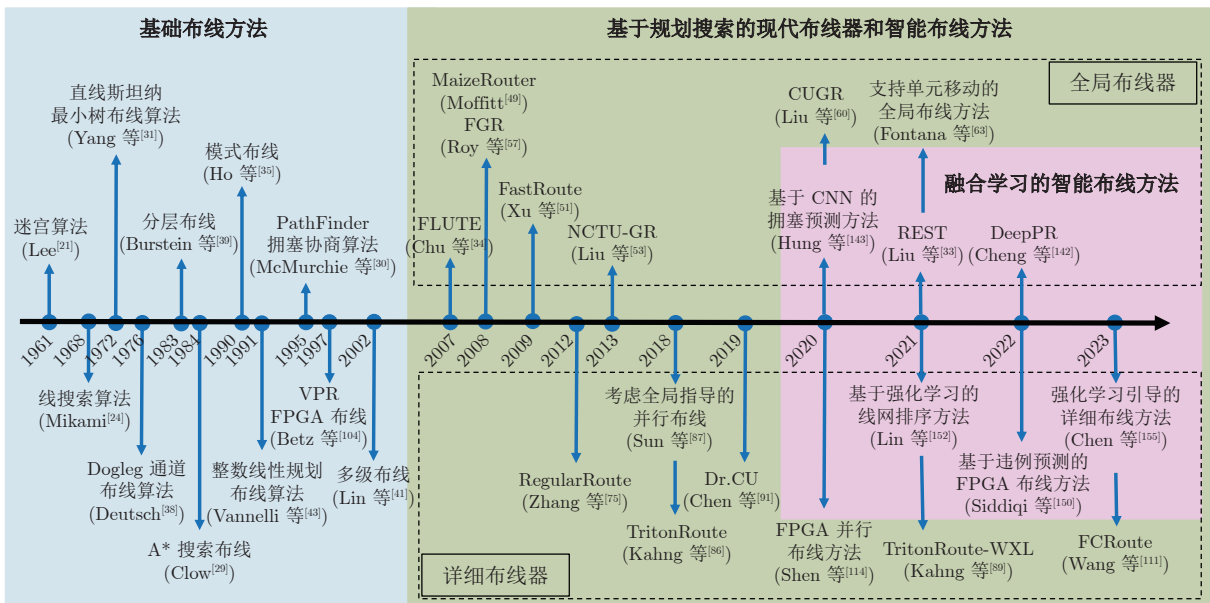


图 6 芯片智能布线方法的发展历程

Fig.6 Evolution of intelligent chip routing approaches

发布, VLSI 全局布线器迎来了快速迭代期. 而 VLSI 详细布线器则在 ISPD-2018<sup>[19]</sup> 和 ISPD-2019<sup>[20]</sup> 详细布线竞赛发布后进入研究的快车道. 最近几年, 随着深度学习和强化学习算法的突破, 融合机器学习的布线方法得到广泛的关注, 已逐步在工业级芯片布线工具中的个别环节中有所应用.

其中主流的芯片智能布线方法的过程特点和应用阶段如表 1 所示.

## 2 基于规划搜索的智能布线方法

本节首先介绍基础的单线网和多线网布线算法, 然后重点阐述基于规划和搜索的智能布线方法在 VLSI 和 FPGA 布线中的最新研究成果. 结合机器学习的智能布线算法及其在元件布局优化、全局

布线和详细布线中的应用将在第 3 节中重点介绍.

### 2.1 基础布线方法

#### 2.1.1 单线网布线

单线网布线中两个引脚的连接是布线中的最基础问题. 早在 1961 年, Lee<sup>[21]</sup> 便提出使用迷宫算法, 采用广度优先搜索技术, 解决存在障碍的网格中两点间的曼哈顿最短路径问题 (Obstacle-avoiding Manhattan shortest path, OAMSP). Hadlock<sup>[22]</sup> 和 Soukup<sup>[23]</sup> 分别从空间和时间效率方面对迷宫算法进行优化. Mikami 等<sup>[24]</sup> 和 Hightower<sup>[25]</sup> 则通过线搜索算法寻找逃逸点, 继而利用逃逸点串连两个端点. Hetzel<sup>[26]</sup> 尝试综合迷宫算法的最优性和线搜索算法的效率, 提出用一组相邻顶点的间隔而非单

表 1 智能布线方法的特点和应用阶段  
Table 1 Characteristics and application phases of intelligent routing approaches

类型	布线方法	方法过程和特点	布线阶段
	迷宫布线	从一个引脚点出发, 采用广度优先搜索技术, 找出存在障碍的网格中另一引脚点的曼哈顿最短路径, 然后以已经连接好的路径为起点, 用同样的方法寻找下一个引脚点, 直到连接完线网的所有引脚	GR/DR
	线搜索布线	首先将源和目标设置为基点, 向四个方向 (两个水平和两个垂直) 执行深度优先搜索, 一直延伸到芯片边界或障碍物, 生成 0 级线段. 然后, 将这些第 $i$ 级线段的每个网格点设置为新的基点, 生成第 $i + 1$ 级的垂直线段. 这个过程不断重复, 直到从源生成的线段与从目标生成的线段相交, 通过交叉点跟踪到源和目标来找到它们的连接路径	GR/DR
	通道布线	在一个包含水平和垂直多条布线轨道的横向布线区域, 线网引脚分布在顶端和底部, 布线时构建水平约束图和垂直有向约束图, 利用贪婪算法和遗传算法等方法将不同线网的横边分配到合适的轨道上, 以减少轨道占用, 压缩通道宽度	DR
	A* 搜索布线算法	一般以连线长度、通孔成本和违例成本为权重的曼哈顿距离作为启发式函数, 利用 A* 算法连接存在障碍的网格中的两个引脚点. 然后以已经连接好的路径为起点, 选择离已连线引脚的中心最近的引脚作为下一个目标点, 直到连接完线网的所有引脚	GR/DR
规划搜索类	模式布线	对每一个双引脚线网, 只考虑 L, Z, U, T, E 等几种形状的连接方式, 能快速找到最短连接方案, 可用于全局布线或者详细布线的首次迭代	GR/DR
	整数线性规划布线	网表中每个线网通过 L 型或 Z 型找出多个可选路径方案, 并按照成本给每个方案一个权重, 在满足每个线网只选择一个方案, 且所有线网的路径方案所使用的边不超过其容量前提下, 最小化所有方案的成本和	GR
	拥塞协商	多线网布线时以协商的方式确定各线网间互联资源的分配, 迭代地进行拆线重布, 初始时所有线网都用最小代价完成布线, 逐渐加大对多个线网重复使用的互连资源或者设计规则违例区域的惩罚力度, 经过多次迭代后, 基于拥塞协商的布线能快速分散拥塞, 避免冲突和溢出	GR/DR
	多级布线	用动态规划的思想将芯片逐级划分到更小区域, 直到能够被算法处理为止, 部分线网因而被切割成多段, 待下级区域内布线完成后再向上串连各段线网, 最终输出完整的布线方案	GR/DR
	查表布线	对于少于一定数量 (如 10 个) 引脚的线网, 先预先构建并存储好直线斯坦纳最小树的所有可能连接方案, 在实际布线时通过查表的方式快速确认方案的可行性; 对于大线网则通过线网拆分技术将其拆成多个小线网, 单独查表布线后再汇总连接	GR
	并行布线	将布线区域进行切割或者将非重叠线网划分到不同的分组, 通过布线任务调度和多线程并行布线实现加速, 提高布线效率	GR/DR
	拥塞与违例预测	利用卷积神经网络等深度学习框架, 基于芯片布局和引脚密度等信息生成拥塞热图, 或者基于全局布线产生的拥塞报告, 预测给定设计的详细布线所产生的设计规则违例位置, 指导布线路径优化	GR/DR
机器学习类	布线路径生成	利用收集的布线方案进行对抗学习, 通过生成模型生成线网的全局布线方案, 并根据预估线长和拥塞结果指导宏布局和线网排序	GR
	强化学习布线	基于强化学习算法框架, 训练布线智能体完成线网排序、路径选择、违例修复等任务	GR/DR
	多智能体布线	将芯片布线转化为多智能体路径规划任务, 基于多智能体深度强化学习框架训练多智能体并行布线, 利用智能体之间的异步通信来避免布线冲突	GR/DR



个顶点来表示布线网格图, 结合面向目标的 Dijkstra 算法<sup>[27]</sup>完成单线网布线, 但 Hetzel 算法削弱了后面介绍的多级布线方法的前提, 因而并没有在学术界流行. Hart 等<sup>[28]</sup>在 1968 年提出 A\* 搜索算法, 因其高效的图搜索能力, 得以迅速应用到布线中, 其中, 基于 A\* 搜索的 VLSI 布线算法<sup>[29]</sup>和 FPGA 布线算法<sup>[30]</sup>直到现在仍然被大量布线器所使用, 作为最基础的布线寻路方案.

对于拥有两个以上引脚的线网的布线问题, Yang 等<sup>[31]</sup>将该问题转化为直线斯坦纳最小树 (Rectilinear Steiner minimum tree, RSMT) 问题, 并通过分支定界法找出次优解. 在此基础上, 很多研究<sup>[32-34]</sup>都尝试在效率和效果上优化 RSMT 的生成. 其中, Chu 等<sup>[34]</sup>给出了一种快速生成 RSMT 的 FLUTE 算法, 对少于 10 个引脚的线网, 先预先构建并存储好 RSMT 的所有可能连接方案, 在实际布线时通过查表的方式快速确认方案的可行性; 对于其余的大线网则通过线网拆分技术将其拆成多个小线网, 单独查表布线后再汇总. 在不考虑障碍物或障碍物不多且比较分散的环境下, 模式布线 (Pattern routing)<sup>[35-37]</sup>对每一个线网, 只考虑 L, Z, U, T, E 等几种形状的连接方式, 能快速找到大部分线网的最短连接方案, 可用于全局布线或者详细布线的首次迭代.

### 2.1.2 多线网布线

多线网布线时如何争夺和协商布线资源, 以确保在全部线网布通的前提下, 实现尽可能低的总线长、通孔数和设计规则违例, 是芯片布线的最大挑战, 特别是现代芯片具有百万级以上线网的情况下, 协商机制变得愈发重要. 通道布线是早期芯片详细布线中非常重要的场景, 通道是指一个横向的布线区域, 包含水平和垂直多条布线轨道, 线网引脚分布在顶端和底部, 布线时构建水平约束图和垂直有向约束图, 利用 Dogleg<sup>[38]</sup>等算法将不同线网的横边分配到合适的轨道上, 以减少轨道占用, 压缩通道宽度. 但当电路复杂性达到一定水平, 通道必须非常大, 所以有必要改变电路布局范式, 删除单元行之间的通道, 并将布线分配到单元之上. 现代 VLSI 芯片线网数量庞大, 布线层数较多, 对于元件布局也没有特别的约束, 是一种全芯片布线 (Full-chip routing), 一般采用多级布线方法, 如自上而下 (Top-down)<sup>[39]</sup>、自底向上 (Bottom-up)<sup>[40]</sup>、粗化-反粗化 (Coarsening-uncoarsening)<sup>[41]</sup>、 $\Lambda$ -型 ( $\Lambda$ -shape)<sup>[42]</sup>等, 将芯片逐级划分到更小区域, 直到能够被算法处理为止, 部分线网因而被切割成多段, 待下级区域内布线完成后向上串连各段线网, 最终输出完整的布线方案. 图 7 是粗化-反粗化多级布线方法

的框架. 在粗化阶段, 为本地连接同时执行全局和详细布线, 然后估计下一级别的布线拥塞; 在反粗化阶段, 使用迷宫布线来重新布线失败的连接并完善解决方案.

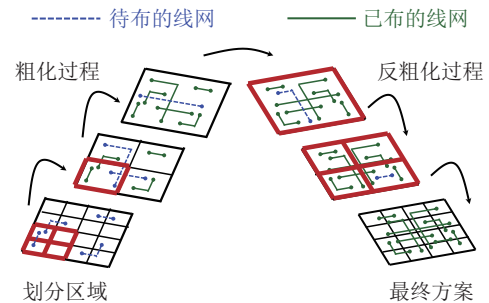


图 7 粗化-反粗化多级布线方法框架  
Fig. 7 Coarsening-uncoarsening multilevel routing framework

对单个区域布线时, 可以采取并行布线或顺序布线. 并行布线可通过整数线性规划 (Integer linear programming, ILP)<sup>[43-45]</sup>求解, 或者利用模拟退火布线 (Simulated annealing routing) 算法<sup>[46]</sup>逐步优化, 但 ILP 方法不太适合应用于具有百万级以上线网的大规模芯片布线中. 顺序布线则是每次只处理一个线网, 通常使用 A\* 搜索或者其他单线网布线算法, 布线时允许线网间临时重叠, 然后对于重叠的线网进行拆线重布<sup>[47]</sup>, 并对重叠位置进行惩罚, 直至所有线网都布通, 且不存在重叠. 顺序布线与拆线重布方案通常效率更高, 但依赖于有效的线网排序、走线策略和拥塞位置的预测, 以及布线资源的重新配置, 否则容易陷入局部最优解. 传统线网排序依据通常包括线网重要性、预估线长 (短则优先)、引脚数 (多则优先) 等; 当前研究则侧重利用机器学习算法学习线网排序策略 (将在后文中介绍). 对于拥塞识别和资源重配, 基于拥塞协商的布线算法<sup>[30]</sup>是一种非常流行的处理方法. 它允许在布线时以协商的方式确定各线网间互联资源的分配, 迭代地进行拆线重布, 初始时所有线网都用最小代价完成布线 (即使导致过度拥塞), 走线时一般使用迷宫算法. 随着迭代次数的增加, 逐渐加大对多个线网重复使用的互连资源或者设计规则违例区域的惩罚力度, 以此来逐步分散拥塞, 降低互连资源的重复利用, 消除违例.

### 2.1.3 现代布线器

现代布线器通常将上述基础布线算法结合起来使用. 如初次布线时使用模式布线或者基于查表的斯坦纳树构建, 然后修改部分区域的布线模式以避免拥塞或冲突, 最后针对还存在拥塞的区域和线网



使用基于拥塞协商的布线算法完成最终布线. 布线总体流程如图 8 所示.

布线器接收到布局结果、技术定义、设计规则等数据输入后, 首先会对数据进行解析, 初始化布线前的各项参数和芯片环境; 然后进入多轮的布线迭代, 在每一轮迭代中, 布线器采用分而治之 (Device-and-conquer) 的策略将整个芯片布线空间划分成一个个区域 (每个区域可能覆盖多个 GCell), 计算好区域间的连接点, 对每个区域单独完成布线后进行汇总形成最终的布线结果. 在每一个区域布线过程中, 首先初始化区域环境, 然后根据布线算法的策略逐个选择未布线线网进行最终布线, 或者选择多个未布线线网进行并行布线, 如果布线空间被投影到二维, 则还需要考虑层分配 (Layer assignment) 问题, 以获得最终三维布线方案. 布线时需要满足设计规则和各种约束、时序要求, 尽可能减少布线长度和通孔数. 线网选择与线网布线是整个布线算法的最核心部分. 布线后计算以连线长度、通孔数、溢出或违例数等指标加权构成的布线成本, 如果出现溢出或违例, 记录该区域和受影响的线网, 尝试进行拆线重布, 最后回写走线路径. 如果完成区域内所有线网的布线, 则进入下一轮迭代. 在下一轮的迭代中, 可通过增加区域偏移以便处理区域交汇处的溢出或违例. 经过多轮迭代后, 整理并输出最终布线结果.

## 2.2 VLSI 智能布线

### 2.2.1 VLSI 全局布线器

随着 ISPD-2007<sup>[17]</sup> 和 ISPD-2008<sup>[18]</sup> 全局布线比赛及基准测试程序的发布, VLSI 全局布线器迎来了快速迭代期. 其中基于整数线性规划的并行布线方法, 对网表  $\mathcal{N} = \{N_1, N_2, \dots, N_n\}$  中的每个线网  $N_i, 1 \leq i \leq n$ , 通过 L 型或 Z 型找出  $k$  个可选路

径方案, 用二元变量  $x_{i1}, \dots, x_{ik}$  表示  $N_i$  是否选择该路径方案, 并给每个方案一个与布线成本相关的权重  $w_{i1}, \dots, w_{ik}$ , 在满足每个线网只选择一个方案 (互斥), 且所有线网的路径方案所使用的边不超过边容量下, 最小化所有方案的成本和  $\sum_{i=1}^n \sum_{j=1}^k w_{ij} x_{ij}$ . 其中, BoxRouter 2.0<sup>[48]</sup> 通过渐进式盒扩展、整数线性规划和自适应迷宫布线, 有效地预测并分散拥堵. GRIP<sup>[45]</sup> 则通过分区策略缩小问题规模, 利用基于线性规划的定价流程给每个线网生成一组有前途的候选方案, 并在三维空间直接优化连线长度和通孔成本.

随着芯片规模逐步增大, 迭代拆线重布的布线方案获得了更多的关注. 全局布线器 MaizeRouter<sup>[49]</sup>、NTUgr<sup>[50]</sup>、FastRoute 4.0<sup>[51]</sup>、NTHU-Route 2.0<sup>[52]</sup>、NCTU-GR 2.0<sup>[53]</sup>、SPRoute<sup>[54]</sup> 等均采用类似的布线流程, 如图 9 所示. 1) 将三维布线问题投影到二维平面上; 2) 分区或分解大范围多引脚线网到可处理规模; 3) 利用基于拥塞协商的拆线重布算法, 或拥塞预测、感知和绕行方法, 或并行处理方法, 完成二维平面布线; 4) 执行层分配获得三维全局布线方案. 这些算法的不同点大多在于分区策略或分解策略、二维平面布线策略和优化方法, 以及后续的层分配方法上. Jiang 等<sup>[55]</sup> 在 2022 年提出了一种并行层分配框架 COALA, 并行考虑线网的所有线段, 并从最低可用层迭代分配到最高可用层, 该方案可以最大限度地利用每一层的布线资源, 从而实现有效的重布线过程, 大大减少了溢出. 刘耿耿等<sup>[56]</sup> 采用基于区域划分的并行策略和基于线网等效布线方案感知的通孔优化策略实现并行层分配, 减少了布线运行时间和通孔数量.

将三维布线空间投影到二维平面固然能够降低问题的处理难度, 提高布线效率, 但同时也丢失了三维空间中的部分特征, 如特定层的阻塞物、不同

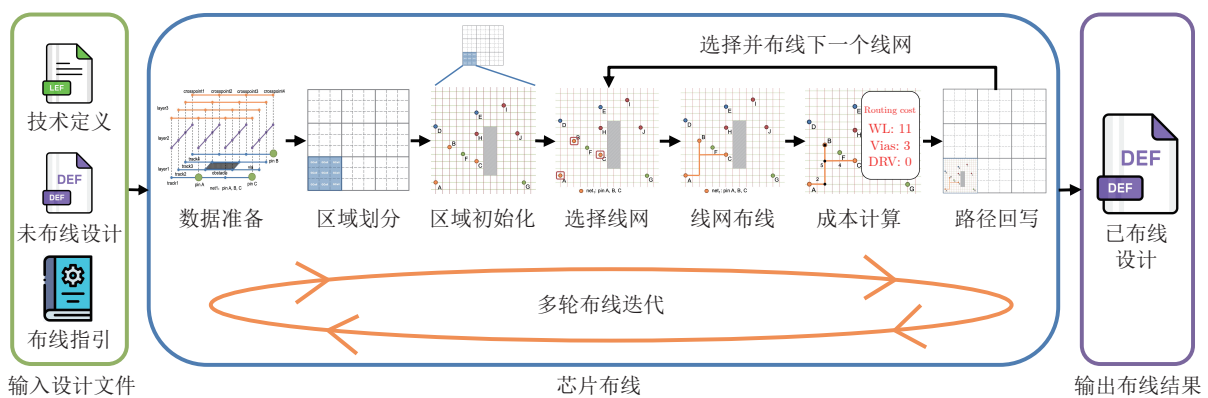


图 8 布线流程图

Fig. 8 Routing procedures

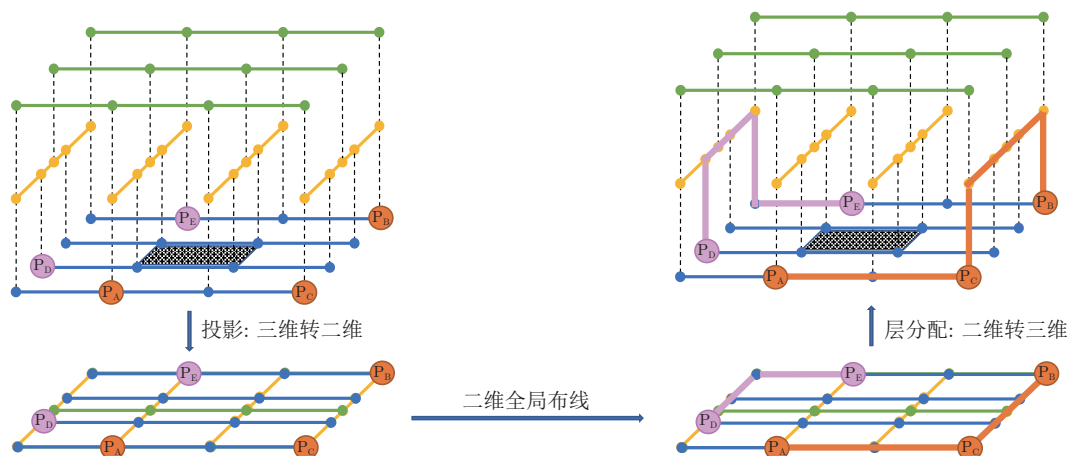


图 9 三维转二维全局布线流程: 投影-布线-层分配

Fig. 9 3D to 2D global routing process: Projection-routing-layer assignment

层的不同线宽和线距等信息, 从而降低了方案质量. 因此, 布线器 FGR<sup>[57]</sup>、ARCHER<sup>[58]</sup>、MGR<sup>[59]</sup> 等直接在三维空间中进行全局布线. CUGR<sup>[60]</sup> 全局布线器设计了结合模式布线和层分配的三维模式布线, 以及具有不同代价函数和目标的两级三维迷宫布线, 并给出多种“补丁”技术以完善布线指引, 布线结果和性能均优于 ICCAD-2019 全局布线竞赛<sup>[61]</sup> 的所有参赛者. 为了充分利用设备的 GPU 资源, Liu 等<sup>[62]</sup> 设计了一个 GPU 加速的 3D 模式布线算法, 并通过高性能的任务图调度器来分配 CPU (Central processing unit) 和 GPU 任务, 以提高布线效率.

布局 and 布线通常定义为两个独立的问题, 以降低设计的复杂性, 但不同步的拥塞分散和布线资源供给目标会导致解决方案质量的下降, 有必要设计一个高效且高度耦合的布局 and 布线联合引擎, 一种支持单元移动 (Cell movement) 的全局布线. Fontana 等<sup>[63-64]</sup> 提出了一种基于整数线性规划的带单元移动的全局布线器 ILP-GRC, 能够同时执行单元移动和线网布线, 并在不损失布线质量的前提下重定位可能导致布线问题的单元. Zou 等<sup>[65]</sup> 则提出了一种考虑单元移动和复杂布线约束的增量式三维全局布线引擎, 以实现单元重定位和线网重布线. Zhu 等<sup>[66]</sup> 为了实现对单元移动的快速高精度位置预测, 构建了一个考虑复杂约束/目标的查找表, 并基于查找表为每个单元生成一个时间驱动的增益图. 最后, 在预测的基础上提出了一种单元移动和集群移动交替进行、部分拆线重布的方案来优化布线方案. 该引擎在 ICCAD-2020<sup>[67]</sup> 和 ICCAD-2021<sup>[68]</sup> 带单元格移动的布线竞赛基准测试中以较少的运行时间获得了更好的求解结果.

### 2.2.2 VLSI 详细布线器

VLSI 详细布线器参考全局布线器输出的布线

指导, 根据生产工艺要求和设计规范, 分析引脚的连接方式, 将走线片段分配到具体的金属层轨道和通孔上, 并不断优化总连线长度、通孔数、DRV 等目标. 下面介绍 VLSI 详细布线中几个主要任务的科研成果, 如表 2 所示.

1) 引脚连接点分析. 引脚访问已成为 14 nm 及以后技术节点详细布线的最大挑战之一, 其中必须使用双重成像光刻技术来制造具有紧密间距的较低金属层. 为了执行正确的引脚连接, Nieberg<sup>[69]</sup> 计算各种引脚访问候选路径, 并选择不违反任何设计标准的从引脚到网格点的最短路径. 在自对准双重成像技术 (Self-aligned double patterning, SADP) 限制下, Xu 等<sup>[70]</sup> 提出了一种动态得分策略以基于引脚访问点数量来动态评估引脚可达性. 在后续文献<sup>[71]</sup> 中, 将并发引脚访问优化问题建模为加权区间分配问题, 采用最优整数线性规划公式和可扩展拉格朗日松弛算法求解. Kahng 等<sup>[72]</sup> 提出了一个完整的设计规则感知的基于动态规划的引脚访问分析框架, 该框架能够同时进行标准的基于单元和基于实例的引脚可达性分析. 并在此基础上, 开发了一个引脚访问驱动的详细布局优化算法以加速布线收敛<sup>[73]</sup>.

2) 轨道分配. 轨道分配将全局布线的连接片段 (特别是横跨 GCell 的线段) 预先分配到具体的布线轨道上, 以减少详细布线时的违例产生, 是衔接全局布线和详细布线的重要一步, 由 Batterywala 等<sup>[74]</sup> 于 2002 年首次提出后, 不断得到优化. RegularRoute<sup>[75]</sup> 将每组布线轨道内的全局线段分配问题构建为最大加权独立集问题, 基于构造修正 (Correct-by-construction) 的方法, 逐步解决因轨道分配不合理导致的间隔违例, 在 ISPD98<sup>[76]</sup> 和 ISPD05/06<sup>[77-78]</sup> 基准测试集上进行了验证. Wong 等<sup>[79]</sup>

表 2 VLSI 详细布线方法  
Table 2 Detailed routing approaches for VLSI

布线任务	文献 (年份)	方法特点	数据集
引脚连接点分析	Nieberg <sup>[60]</sup> (2011)	选择从引脚到网格点不违反设计规则的最短路径	Foundry
	Xu 等 <sup>[70]</sup> (2016)	基于引脚访问点数量来动态评估引脚可达性	OpenSparc T1*
	Kahng 等 <sup>[72]</sup> (2020)	设计规则感知的基于动态规划的引脚连接点分析框架	ISPD18 <sup>[19]</sup>
轨道分配	Zhang 等 <sup>[75]</sup> (2012)	基于最大加权独立集问题启发式求解的轨道分配算法	ISPD98 <sup>[76]</sup> & ISPD05 <sup>[77]</sup> & ISPD06 <sup>[78]</sup>
	Wong 等 <sup>[79]</sup> (2016)	基于协商的轨道分配算法	DAC12 <sup>[80]</sup>
	Liu 等 <sup>[81]</sup> (2019)	可布线性驱动的轨道分配算法	DAC12 <sup>[80]</sup>
	郭文忠等 <sup>[82]</sup> (2019)	基于离散粒子群优化、遗传操作和基于协商的精炼策略的轨道分配算法	DAC12 <sup>[80]</sup>
基于网格的并行布线	Jia 等 <sup>[83]</sup> (2014)	基于多商品流方法的并行布线算法	Foundry & ISPD05 <sup>[77]</sup> & ISPD14 <sup>[85]</sup>
	Jia 等 <sup>[84]</sup> (2018)		
	Kahng 等 <sup>[86]</sup> (2018)	通过混合整数线性规划逐层进行并行布线	ISPD18 <sup>[19]</sup>
基于网格的顺序布线	Sun 等 <sup>[87]</sup> (2018)	考虑全局指导的基于协商的并行布线方法	ISPD18 <sup>[19]</sup>
	Kahng 等 <sup>[80]</sup> (2021)	基于队列的拆线重布算法以减少违例, 用全新的设计规则分析引擎、准确计算路径成本	ISPD18 <sup>[19]</sup> & ISPD19 <sup>[20]</sup>
	Chen 等 <sup>[90]</sup> (2019)	分层并行迷宫布线方法实现总线布线	ICCAD 2018**
	Li 等 <sup>[92]</sup> (2019)	边构造边修正的路径优化并行搜索方法降低设计规则违例的数量	ISPD18 <sup>[19]</sup> & ISPD19 <sup>[20]</sup>
	Zhuang 等 <sup>[93]</sup> (2022)	设计规则驱动的轨道分配和违例优化方法	ISPD19 <sup>[20]</sup>
无网格布线	Cong 等 <sup>[94]</sup> (2005)	基于多级布线框架的无网格布线器	Foundry
	Li 等 <sup>[95]</sup> (2007)	结合瓦片展开和隐式连接图的多层无网格布线器	Foundry
光刻工艺友好的布线	Du 等 <sup>[96]</sup> (2013)	基于符合 SID (Spacer-is-dielectric) 型自对准双重成像设计约束的图模型的拥塞协商详细布线	Foundry
	Ding 等 <sup>[97]</sup> (2015)	基于符合 SIM (Spacer-is-metal) 型自对准双重成像和自对准四重成像设计约束的图模型的拥塞协商详细布线	Generated
	Ou 等 <sup>[98]</sup> (2017)	基于 DSA (Directed self assembly) 光刻工艺友好的图模型的详细布线	OpenSparc T1*
	Yu 等 <sup>[99]</sup> (2018)	考虑 DSA 光刻工艺的基于冲突和兼容性图模型的详细布线	Foundry

注: \* OpenSPARC T1, <http://www.oracle.com/technetwork/systems/opensparc/index.html>

\*\* ICCAD 2018 Contest Problem B: Obstacle-Aware On-Track Bus Routing, <http://iccad-contest.org/2018/problems.html>

利用基于协商的算法在 DAC12<sup>[80]</sup> 基准测试集上分配轨道, 并根据分配结果判断哪些区域可能存在严重的可布线性问题. Liu 等<sup>[81]</sup> 展示了一个可布线性驱动的轨道分配算法 RDTA, 能同时考虑局部线网、通孔位置和引脚访问, 并根据分配结果评估可布线性. 郭文忠等<sup>[82]</sup> 基于离散粒子群优化、遗传操作和基于协商的精炼策略, 综合考虑局部线网、重叠冲突、线长和障碍物, 提出轨道分配算法.

3) 基于 ILP 的并行布线. 详细布线方法包括并行布线和顺序布线、基于网格布线和无网格布线等类型. 其中, MCFRoute<sup>[83-84]</sup> 是一种并行网格布线算法, 在详细布线前通过轨道分配降低问题求解的难度, 然后基于多商品流 (Multi-commodity flow, MCF) 方法, 将具有复杂设计规则约束的详细布线问题化为整数线性规划问题, 同时利用启发式方法消除冗余约束, 通过多线程加快模型求解速度, 在工业基准测试集和学术基准测试集<sup>[77, 85]</sup> 中都验证了其算法的有效性. TritonRoute<sup>[86]</sup> 将每层芯片划分成并行的 GCell 宽度的面板并通过混合整数线性

规划 (Mixed integer linear programming, MILP) 方法逐层对每个面板进行并行布线, TritonRoute 在 ISPD-2018<sup>[19]</sup> 详细布线竞赛中赢得了冠军. ILP 方法虽然克服了线网排序的难题, 但当芯片中的线网数量很多时, 转化为 ILP 后问题的规模很大, 致使求解速度很慢. 虽然可以通过将芯片分区的整个布线问题划分为多个子问题后分别求解, 但相较于基于 Dijkstra 算法和 A\* 搜索的布线算法, 基于 ILP 的布线算法的结果不够稳定.

4) 基于资源协商的拆线重布. 大量文献<sup>[87-93]</sup> 在路径选择时采用基于资源协商的拆线重布方案, 通过任务划分和多线程并行布线实现加速, 布线时考虑设计规则, 以便能够找到一条没有 DRV 的路径, 但这也带来大量规则判断的计算代价. 对于 VLSI 详细布线, 不同于 FPGA 有固定的标准单元位置、端口连接和单元间布线区域大小, VLSI 布局后的单元引脚可能不在布线轨道上, VLSI 布线需要考虑 Off-track 布线、Wrong-way 布线, 以及更多更细的设计规则检查等问题. 针对 ISPD-2018<sup>[19]</sup> 和 IS-



PD-2019<sup>[20]</sup> 详细布线竞赛中引入的现代规模芯片及设计规则, Sun 等<sup>[87]</sup> 采用了基于协商的并行布线方法, 这种方法广泛应用于各种布线器. 其特点是允许线网之间的走线重叠, 但是会在重叠区域的走线代价上增加额外的历史违例惩罚项, 并且惩罚项的大小随着迭代的进行不断增加, 迫使线网在下次迭代中绕开重叠区域, 选择其他代价更低的线路. Kahng 等<sup>[88]</sup> 在 2020 年更新了 TritonRoute 核心算法, 包含一个支持 LEF (Library exchange format) 和 DEF (Design exchange format)<sup>1</sup> 数据模型的内存数据库, 以及一个能够理解连接性和设计规则约束, 基于 A\* 搜索和拆线重布的端到端并行详细布线方案. TritonRoute-WXL<sup>[89]</sup> 在 TritonRoute<sup>[88]</sup> 基础上设计了全新的设计规则分析引擎, 提出了基于队列的重布线算法, 不仅在 ISPD-2018<sup>[19]</sup> 和 ISPD-2019<sup>[20]</sup> 的部分基准测试中获得了无违例 (Violation-free) 的布线结果, 而且成功应用在工业级芯片的布线中. 图 10 演示了 TritonRoute-WXL 中十分关键的基于队列的拆线重布操作过程<sup>[89]</sup>, 该方法大大减少了 DRV 数量和布线时长, 例子中包含二维平面中的 3 个需要布线的双引脚 (Two-pin) 线网. 每个图显示了线网被布线之前的布局和队列中的元素, 每个队列元素是一个三元组: 第 1 项是线网标识; 第 2 项是  $T$  表示需要拆线重布,  $F$  表示不需要拆线重布, 只需要执行设计规则检查; 第 3 项用于跟踪线网被布线的次数. 计数器可以防止线网的布线次数超过允许的拆线重布迭代次数, 或是对已经解决的违例进行不必要的布线. 在图 10(a) 中, 3 个线网按照  $n_0, n_1, n_2$  的顺序进行初始布线. 图 10(b) 说明了  $n_0$  被布线后的布局和队列情况. 从图 10(c) 可以看出, 在  $n_1$  被布线后,  $n_0$  与  $n_1$  之间存在短路违例. 考虑到  $n_0$  在  $n_1$  之前布线, 因此  $n_0$  作为攻击者被推到队列的后面进行重布线,  $n_1$  作为受害者被推到队列的后面进行设计规则检查. 类似地, 图 10(d) 显示,  $n_2$  被布线后,  $n_2$  与  $n_0$  有短路违例. 因此,  $n_0$  被推到队列的后面进行重布线,  $n_2$  被推到队列的后面进行设计规则检查. 图 10(e) 表明, 在  $n_0$  被重布线后, 两个短路违例都已被解决, 且  $n_1$  的设计规则检查没有发现新的违例. 从图 10(f) 可以看出, 当  $n_0$  从队列弹出时, 布线计数器表明  $n_0$  已经布线两次, 而  $n_0$  被推入队列时只布线一次, 因此将跳过这次  $n_0$  的重布线, 随后  $n_2$  的设计规则检查也没有发现新的违例. 队列的最后两个元素被推出后没有新元素推入, 3 个双引脚线网的布线随即完成. 基于队列的拆线重布方案通过对关键资源的重分配, 较好

<sup>1</sup> LEF/DEF Language Reference Version 5.7, <https://www.ispd.cc/contests/18/lefdefref.pdf>

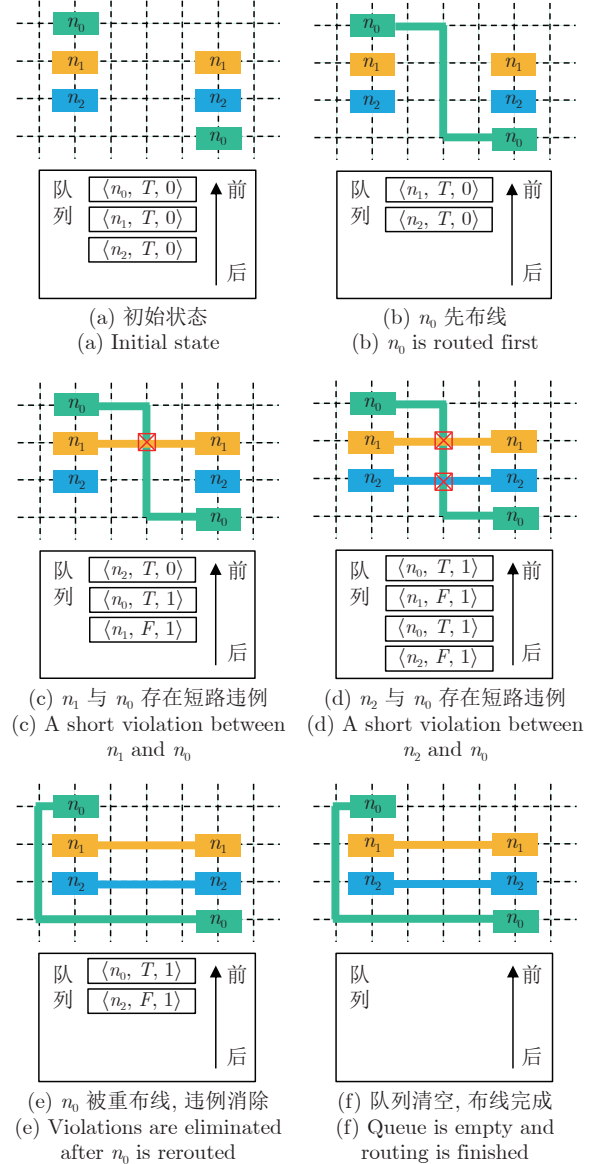


图 10 基于队列的拆线重布过程

Fig. 10 Queue-based rip-up and reroute process

地解决了线网密集区域的布线资源冲突和抢夺的问题. 在国内, 香港中文大学 Chen 等<sup>[90]</sup> 针对总线布线的特点, 提出一种分层并行迷宫布线方法 MARCH, 通过粗粒度的拓扑感知路径规划结合细粒度的轨道分配, 使得一组引脚按照同样或相似的路径并行布线. Chen 等<sup>[91]</sup> 在 2019 年建立了 Dr.CU 布线器, 改进了 Dijkstra 算法, 通过边构造边修正的路径优化并行搜索方法, 降低了设计规则违例的数量. 该布线器为庞大的 3D 详细布线网格图设计了一套两级稀疏数据结构, 为捕捉最小面积约束开发了一个最优的路径搜索算法, 并提出了一种有效的批量同步并行方案, 以进一步减少详细布线过程的周转时间. Dr.CU 2.0<sup>[92]</sup> 在 Dr.CU<sup>[91]</sup> 基础上考虑了引脚可达性



及更多的约束, 包括: 平行运行长度间距、线末间距、角到角间距, 但相对于 TritonRoute<sup>[88]</sup>, Dr.CU 2.0 在 ISPD-2019<sup>[20]</sup> 详细布线基准测试中仍存在大量的设计规则违例. 为降低违例数量, 福州大学的 Zhuang 等<sup>[93]</sup> 在 2022 年提出了一个详细布线器 TRADER, 首先利用设计规则驱动的轨道分配算法获得高质量的初步布线方案, 然后通过拆线重布方法减少局部间距违例, 通过线段迁移减少短路违例, 利用通孔补丁方法减少最小面积违例. TRADER 在 ISPD-2019<sup>[20]</sup> 基准测试上比 Dr.CU 2.0<sup>[92]</sup> 减少了 10% 以上的违例.

5) 无网格布线. 基于网格图搜索的布线算法对内存的要求高, 需要将整个稀疏的网格图存储在内存中, 一般采用间接的方式, 如 R-树等存储网格图中的不可走线段信息. 另外布线的质量很大程度上取决于线网布线的顺序, 并且找到一个最优的线网顺序本身也是一个困难问题, 目前主要通过资源协商的拆线重布方法来解决. 因此, 也有部分文献研究无网格布线, 走线时没有网格限制, 可以根据实际需求使用不同的导线宽度和间距. 这种模型的灵活性更大, 但是由于计算复杂, 它的运算时间大于有网格模型, 适用范围不如网格模型广泛. 如 Cong 等<sup>[94]</sup> 展示了一个多级全芯片无网格布线系统. Li 等<sup>[95]</sup> 提出了一种结合两种常见的无网格布线技术 (瓦片展开和连接图) 的无网格布线器.

6) 光刻工艺友好的布线系统. SADP 光刻技术是高端芯片金属层制备的领先技术. 为了实现成功的分解, 必须进行符合 SADP 的设计. Du 等<sup>[96]</sup> 提出一种符合 SID (Spacer-is-dielectric) 型 SADP 设计约束的图模型, 在详细布线阶段为导线预先着色, 便于后续 SID 分解. Ding 等<sup>[97]</sup> 的图模型考虑到 SIM (Spacer-is-metal) 型的 SADP 和自对准四重成像技术 (Self-aligned quadruple patterning, SAQP) 的约束. DSAR<sup>[98]</sup> 提出一种对定向自组装 (Directed self assembly, DSA) 光刻工艺友好的图模型, 并针对性地对边界处的线网预布线. 台湾

大学的 Yu 等<sup>[99]</sup> 提出一种基于冲突和兼容性的图模型, 使用定向自组装技术的详细布线算法, 以避免吸引更多的通孔插入周围的网格, 减少通孔间距冲突.

### 2.2.3 VLSI 全局详细布线器

VLSI 全局详细布线器 (Global-detailed routing, GDR) 旨在建立一种紧密关联全局布线和详细布线的端到端布线框架. 全局布线输出的布线指引指导详细布线, 并根据详细布线的结果调整全局布线的走线策略, 如图 11 所示. GDRouter 布线工具<sup>[100]</sup> 将全局布线器 FastRoute 4.0<sup>[51]</sup> 与 RegularRoute<sup>[75]</sup> 结合在一个交错的全局布线和详细布线过程中. BonnRoute<sup>[101]</sup> 的全局布线基于 Min-max 资源共享的组合逼近方案, 详细布线使用基于有效数据结构处理引脚访问和长距离连接的最短路径算法. CUGR<sup>[60]</sup> + Dr.CU<sup>[92]</sup> 与 TritonRoute-WXL<sup>[89]</sup> 整合了全局布线和详细布线的很多重要研究成果, 并针对大规模芯片布线进行了大量改进, 是目前开源的两个支持从布局后的 LEF/DEF 输入到最终布线输出的全局详细布线器, 其中, TritonRoute-WXL 已整合到 OpenROAD<sup>[102]</sup> 项目中并成功应用于工业级芯片的布线. CPR2.0<sup>[103]</sup> 整合 CUGR<sup>[60]</sup> 和 TritonRoute<sup>[88]</sup>, 并通过拥塞感知的单元移动和成本缓存等多种技术弥合它们之间的间隙, 提高最终布线质量.

### 2.3 FPGA 智能布线

FPGA 因较低的成本和灵活可编程的特性在过去 10 年获得了广泛且快速增长的应用, 它是一种限定了布局和布线的半定制 VLSI 芯片, 其逻辑块位置和布线区域大小是固定的. 可编程布线由预制接线段和可编程开关组成, 占应用的结构面积和关键路径延迟的 50% 以上, 因此其效率至关重要. 当电路中所有逻辑块的位置选定后, FPGA 布线器计算决定打通哪些可编程开关, 以连接电路需要的所有逻辑块输入和输出管脚. 这种走线是在有限的

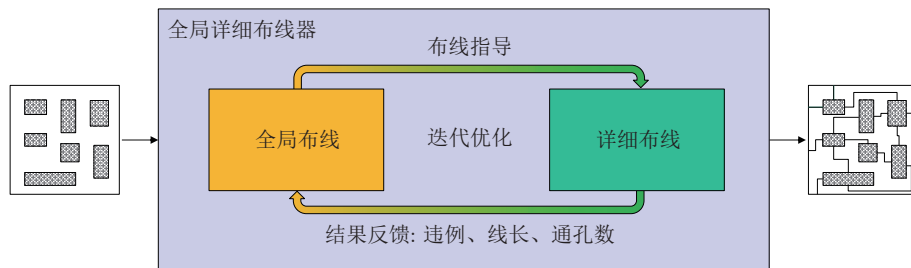


图 11 全局详细布线器框架

Fig. 11 Global-detailed routing framework

资源下进行的,布通率和时延是布线开始时就必须考虑的问题,所以FPGA的布线算法强烈地依赖可编程逻辑器件的结构尤其是布线资源结构.在FPGA布线时,通常用一个有向布线资源图(Routing resource graph, RRG)描述FPGA的布线结构.其中,顶点表示芯片内的逻辑块的引脚或者互连线资源,边表示逻辑块与互连线资源或互连线资源间的可编程开关,再结合图论的路径搜索和资源协商算法求解FPGA布线问题.由于FPGA具有重要的应用价值和特殊的结构特点,有大量文献专门研究FPGA布线方法.

McMurchie等<sup>[30]</sup>在1995年首次提出基于拥塞协商的FPGA布线方法PathFinder,其布线性能和质量都优于早期的两阶段全局详细布线算法.其核心思想是对每条边 $e$ ,基于其需求和容量计算得到的拥塞程度,计算 $e$ 在当前第 $i$ 次迭代的布线成本 $cost(e) = (b(e) + h^i(e)) \times \eta(e)$ ,其中拥塞程度 $\phi(e) = \frac{\eta(e)}{\sigma(e)}$ 定义为经过边 $e$ 的所有线网数量 $\eta(e)$ 除以 $e$ 的容量 $\sigma(e)$ , $b(e)$ 表示使用 $e$ 的基本成本, $h^i(e)$ 表示 $e$ 的历史成本,如果 $\phi(e) > 1$ , $h^i(e) = h^{i-1}(e) + 1$ ,否则不变. $cost(e)$ 随着拥塞度 $\phi(e)$ 的增加而增加,而一个线网的布线成本为 $cost(net) = \sum_{e \in net} cost(e)$ ,从而促使线网选择低使用率低成本的边进行布线,PathFinder在具体走线时使用迷宫算法、Dijkstra算法或A\*搜索算法,目标是在满足边容量情况下的最小布线成本.经过多次迭代后,基于拥塞协商的布线能快速地分散拥塞,避免冲突和溢出.PathFinder的拥塞协商布线算法是目前大部分FPGA布线算法的基础,包括学术上常用的VPR (Versatile place and route)<sup>[104-105]</sup>工具以及nextpnr<sup>[106]</sup>工具在布线阶段使用的都是基于PathFinder的改进算法.其中VPR是目前最成功的FPGA布线算法,它使用同样的资源协商方法解决全局和详细布线问题,并在普通迷宫算法中加入了用于指导布线的资源拥塞度预测,以提高布通率.然而,随着电路规模的不断增大,PathFinder布线方法耗时越来越长,在布线资源受限情况下具有不可忽视的布线质量变化,需要新的解决方案加速布线过程,提高布线质量.

### 2.3.1 FPGA 布线时长优化

对于FPGA布线时间过长问题,通常有两种加速手段.一种是对基于拥塞协商的拆线重布算法进行优化.Gort等<sup>[107]</sup>改进了PathFinder算法,强制线网使用单个输出引脚,且在每次迭代中只重布拥塞的线网,从而获得了3倍的加速.Wang等<sup>[108]</sup>和CRout<sup>[109]</sup>改进了PathFinder中的重布线策略,只

有因拥塞超过时序要求的非法路径被重新布线,而非整个线网,减少了很多无用的重布线工作.2020年,Murray等<sup>[110]</sup>设计了一种自适应的增量布线器AIR,在每次拆线重布时只拆除非法的布线资源节点,或者时延增大的关键路径.同时,针对一些具有高扇出属性的线网制定了特殊的前向搜索机制,与VPR 7布线器相比,AIR布线速度提高了7倍以上,相比于基于连接的FPGA布线方案CRout,减少17%的布线时间.FCRout<sup>[111]</sup>则设计了一种新的软布线空间剪枝方法以优化底层A\*算法的扩展过程,在快速迷宫搜索过程中只探索高优先级节点,当无法找到目标时,允许回溯过程探索可能处于最佳路径上的低优先级节点,然后重新调用新的快速迷宫搜索.通过减少无用的布线空间探索,FCRout在确保全局搜索能力的同时提高了运行效率.与VPR 8<sup>[105]</sup>中最先进的布线器比较表明,FCRout平均探测不到一半的布线资源节点,在不牺牲结果质量的情况下,平均减少了45%的运行时间.FPGA布线包括可配置逻辑块(Configurable logic blocks, CLB)内部连线和CLB之间的连线,基于协商的拆线重布算法常常因CLB内部越来越复杂的连线结构而产生大量迭代,Wang等<sup>[112]</sup>设计了一个基于ILP的CLB内部并行重布线算法,大大减少了CLB内部的布线拥塞,通过了所有ISPD-2016<sup>[113]</sup>布线测试.

另一种加速手段是并行FPGA布线.得益于多核处理器与图形处理器的普及,并行化成为加速PathFinder布线过程的一种越来越具有吸引力的方法.根据并行化方式的不同,又可以将并行布线技术分为粗粒度FPGA并行布线技术、细粒度FPGA并行布线技术以及两种并行布线技术的混合<sup>[8]</sup>.粗粒度FPGA并行布线技术通过将线网划分为不同的分区,在每个分区内执行独立的布线操作,以实现加速.最终布线的性能取决于线网的划分方式、负载均衡度、并行运算过程中数据的同步模式以及冲突的回避方式等多种因素.细粒度FPGA并行布线技术则对单个线网的布线过程实现并行加速,最终的性能取决于并行布线过程中的并行度以及共享数据的同步方式等诸多因素.

一个基本的多核分布式内存并行布线框架如图12所示,所有的线网根据特定策略被分割成互不相交的子集合.分割完成后,布线问题被建模为分布式内存系统的任务调度问题,每个子集的线网被分配到不同的内核进程.每个进程运行一个单独VPR实例,负责对分配给它的线网执行布线操作,并维护自己的数据结构,包括布线路径和相关的拥塞成本信息,不同VPR实例间的信息同步通过消

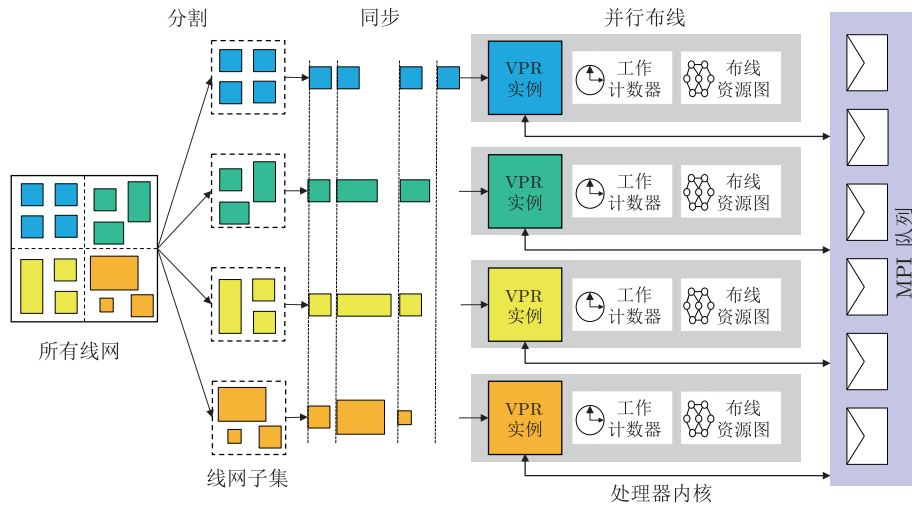


图 12 多核分布式内存并行布线框架

Fig. 12 Multi-core distributed memory parallel routing framework

息传递接口 (Message passing interface, MPI) 来实现。当第一批线网并行布线时, 每个进程需要使用 MPI 消息将中间结果与其他进程通信, 并同步各自的总体布线状态结果。然后对第二批线网进行并行布线, 直至完成当前迭代的布线。

理想的并行布线器不仅速度快, 而且是可扩展和确定的。并行 FPGA 布线研究中, Shen 等<sup>[114]</sup>探索了基于迭代划分的 FPGA 并行布线方案, 首先使用二分法将 FPGA 划分为两个子区域, 根据划分的结果得到三个不同的线网集合: 横跨两个子区域的线网集合以及完全在两个子区域内部的线网集合。由于在两个子区域内部的线网互不相交, 因此可以执行并行布线的操作。可以通过迭代的方式重复上述步骤以增加布线方法的并行度, 但随着递归迭代次数的增加, 跨区域线网的数量也同时增多, 导致 FPGA 并行布线方法的并行度逐渐降低。针对上述问题, Wang 等<sup>[115]</sup>提出了一种混合线网划分机制 ParRA, 在线网递归划分的过程中将线网划分为无冲突线网集合与重叠线网集合, 显著增加了有可能并行布线的线网的数量。随后针对两组不同的集合采取不同的并行布线策略, 并且在布线的过程中很好地解决资源冲突以及负载不均衡的问题。相对于 VPR 7, 在 16 线程下获得大于 24 倍加速。由于线网之间的终端引脚数量不同, 线网布线过程中访问的节点数量不同, 静态分配可能导致内核负载不平衡, 也面临着可伸缩性的挑战。Shen 等<sup>[116-117]</sup>提出了一个同时考虑静态和动态负载均衡的并行布线器, 先通过分级区域划分将布线任务分配到不同的处理器核心以实现静态负载平衡, 然后在运行时协调消息传播和任务迁移, 以动态维护处理器核心之间的负

载平衡。与顺序 FPGA 布线相比, 在不牺牲布线质量的情况下, 在 32 核环境下实现了大于 17 倍的加速。虽然并行布线技术可以提供良好的加速, 但对串行等效的支持仍然非常有限或被忽略, 而串行等效将使得生产级并行计算机辅助设计工具中的回归测试和客户支持更加容易。于是, Shen 等<sup>[118-119]</sup>提出了一种依赖关系感知的调度算法来促进串行等效的并行布线器, 它不仅提供了显著的加速, 而且无论使用多少处理器核, 该调度算法也能使并行布线器获得与串行布线器相同的结果。

不同于粗粒度 FPGA 并行布线技术, 细粒度 FPGA 并行布线技术在布线时不会改变线网的布线顺序。据 Gort 等<sup>[120]</sup>对其实验基准电路布线后统计发现, 约 68% 的布线时间花费在线网走线时搜索邻居节点以及优先队列的处理上。因此, 如何对 PathFinder 算法的迷宫布线器进行细粒度加速是亟需解决的问题。Shen 等<sup>[121]</sup>提出了一种基于 GPU 的细粒度并行布线方法, 可以为大规模 FPGA 实现高并发单线网并行布线。该方法只在一个线网的边界框内, 而不是在整个布线资源图上布线, 然后有选择地扩展它的边界框, 以确保单线网布线最终有一个可行的解决方案, 同时, 在每个边界框内的节点上设置一个 GPU 线程, 对所有节点进行动态规划算法的路径搜索, 实现单 GPU 上的单线网并行布线。

### 2.3.2 FPGA 布线质量优化

对于 FPGA 布线质量问题, Zha 等<sup>[122]</sup>重新审视了 PathFinder 算法本身, 发现其对布线拥塞问题定义不当, 导致其拥塞系数更新策略存在问题, 最终导致不同布线顺序下布线质量的变化。这个问



题在 VTR 8<sup>[105]</sup> 中更加严重, 并且不可避免地降低了布线质量, 因为在大多数情况下预定义的布线顺序不是最优的. 基于这些发现, 他们提出了一种改进型 PathFinder 算法, 重新定义布线拥塞问题, 以解决拥塞系数更新策略中的问题, 并从算法上解决布线质量变化问题. 评估结果表明, 该算法在布线资源约束下, 最大关键路径时延和变化量分别降低了 49.4% 和 96.2%.

## 2.4 小结

总的来说, 基于规划搜索的 VLSI 和 FPGA 智能布线算法在大规模芯片布线中取得了很好的效果, 布线快且每次运行的输出结果一致, 当前主流的商业和开源布线工具的核心布线算法大多都是基于规划和搜索的智能布线算法. 但在芯片规模不断扩大、复杂程度不断提高、设计规则不断增加和调整的情况下, 布线算法的解空间变得极其巨大, 重布线的次数也在不断增加, 加大了用规划搜索类方法求解布线问题的难度. 另外, 这些算法的布线经验依赖于人工设定启发式策略, 无法动态调整, 也没有得到严格的数学验证. 同时, 随着生产工艺的发展, 这些硬编码算法也需要在专家指导下根据工艺要求进行策略调整、优化和完善, 增加了大量的工艺适配和人工优化结果的工作量. 再次, 基于规划搜索的智能布线算法主要使用 CPU 计算资源, 没有充分利用设备中的 GPU 资源. 因此, 寻找更简单、通用且有效的布线算法仍然是芯片布线领域专家的主要任务.

## 3 融合机器学习的智能布线方法

机器学习方法具备学习过往经验并随时间推移

不断改进的能力. 随着近年深度学习和强化学习算法的突破, 人工智能辅助芯片设计 (AI for EDA) 的技术路线得到了广泛关注. 学术界和产业界在芯片设计流程中的拥塞预测、布局优化、设计空间探索等方面应用机器学习技术都取得了重要进展. 为了减少对人工启发式策略的依赖, 研究者们开始探索利用机器学习获取先验知识以辅助芯片布线的方法. 与传统基于搜索的算法不同, 基于学习的方法不会依赖于人工分析获得所有可能的情况来设计布线器, 而是借助其出色的学习能力对训练数据进行深入的探索, 找出有用的信息来分析布局的可布线性, 预测拥塞和违例的数量和位置, 优化布线策略以指导布线动作. 一个融合机器学习的布线过程的布线过程如图 13 所示. 根据设定的策略、规则或特征抽取方式对训练集中的芯片布线问题进行加工处理, 借助规划推理方法完成芯片最终布线, 收集布线轨迹和布线特征, 输入布线经验学习网络进行学习并产生新经验网络, 然后利用新经验网络对验证集中的布线问题进行规划推理, 根据布线结果和代价评估学习的效果, 如果学习效果没有达到预期, 则调整策略、网络参数或特征抽取方式继续学习.

本节综述了近年来提出的基于机器学习的布线算法, 并根据其在布局布线各阶段中所解决的问题分为三类: 基于元件布局优化的布线方法、基于全局布线优化的布线方法和基于详细布线优化的布线方法.

### 3.1 基于元件布局优化的布线方法

当前, 基于元件布局优化的机器学习布线方法主要研究可布线性预测、布局规划优化和布局布线联合学习, 如表 3 所示.

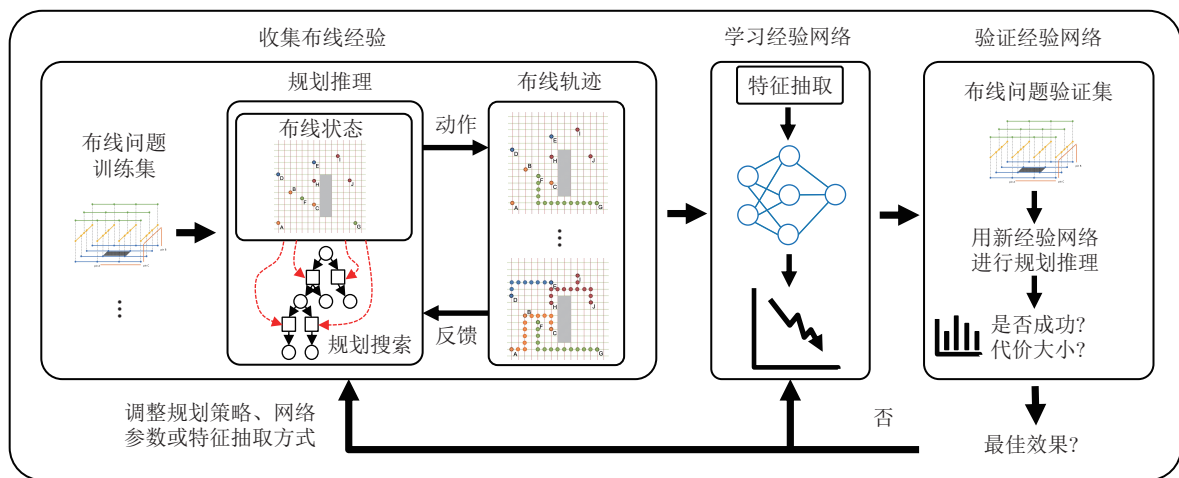


图 13 基于机器学习的布线过程

Fig. 13 Machine learning-based routing process



表 3 基于元件布局优化的布线方法  
Table 3 Routing approaches for cell placement optimization

任务类型	算法框架	文献 (年份)	芯片类型	数据集
可布线性预测	SVN	Chan 等 <sup>[124]</sup> (2017)	VLSI	Foundry
	NN	Tabrizi 等 <sup>[125-126]</sup> (2019, 2018)	VLSI	ISPD15 <sup>[127]</sup>
	CNN	Yu 等 <sup>[128]</sup> (2019)	VLSI	Generated
	CNN	Liang 等 <sup>[129]</sup> (2020)	VLSI	Foundry
	CNN	Xie 等 <sup>[130]</sup> (2018)、Huang 等 <sup>[131]</sup> (2019)	VLSI	ISPD15 <sup>[127]</sup>
	CNN	Chen 等 <sup>[132]</sup> (2022)	VLSI	Foundry & DAC12 <sup>[80]</sup>
	CNN	Alhyari 等 <sup>[133-134]</sup> (2021, 2019)	FPGA	ISPD16 <sup>[113]</sup> & Xilinx
	cGAN	Alawieh 等 <sup>[135]</sup> (2020)、Yu 等 <sup>[136]</sup> (2019)	FPGA	ISPD16 <sup>[113]</sup>
	LHNN	Wang 等 <sup>[137]</sup> (2022)	VLSI	ISPD11 <sup>[138]</sup> & DAC12 <sup>[80]</sup>
布局规划优化	RL + GCN	Mirhoseini 等 <sup>[139]</sup> (2021)	VLSI	Foundry & Ariane <sup>[140]</sup>
布局布线融合	RL + GNN + CNN	Cheng 等 <sup>[141-142]</sup> (2021, 2022)	VLSI	ISPD05 <sup>[77]</sup>

芯片设计过程中的布线阶段特别复杂并耗时, 布线器不仅要完成所有线网的连接, 还要满足设计规则的要求. 当布线结果不能满足功能和性能设计要求时, 就不得不在布线阶段或从布局甚至更早的阶段开始进行迭代优化, 这种反复迭代耗费了工程师相当多的时间和努力. 为了更好地解决这个问题, 改善布线的效率和效果, 一种常用的策略是及早预测芯片布局的可布线性, 并根据需要快速调整布局. 然而, 布局阶段用于评估可布线性的信息只有宏单元放置、引脚分布和密度等, 在大多数情况下, 它们之间的实际关系要复杂得多, 公式不易表达, 对最优参数的设置亦没有给出理论证明. 而机器学习作为一种强大的计算机科学技术, 能够分析大数据并提供预测和找出关联, 因此, 研究人员尝试使用机器学习来克服这一困难.

传统的可布线性驱动的布线算法<sup>[123]</sup>大多着眼于提高全局布线效果, 在目标函数中对拥塞信息进行建模, 但在 14 nm 以下制程的芯片中, 设计规则违例更多的来源于多重图像光刻和引脚访问限制, 基于全局布线的拥塞地图可能与最终设计规则违例地图没有密切相关. 为了实现更好的预测, Chan 等<sup>[124]</sup>从布局和全局布线中提取了引脚和单元密度、拥塞信息、网表连接、引脚接近度等特征信息, 引入机器学习技术来预测全局布线后的设计规则违例热点. 实验结果表明, 该方法预测了 74% 的详细布线 DRC 违例的位置, 并通过优化布局设计减少了 20.6% 的违例. Tabrizi 等<sup>[125-126]</sup>发现在全局布线中常常被忽略的局部线网极大地影响最终布线的质量, 使用全局布线信息来预测违例耗时且不够有效, 决定在预测时完全删除全局布线信息, 只保留布局信息, 并使用有监督神经网络模型来检测短路违例. 在 ISPD15<sup>[127]</sup> 基准测试集上的实验显示, 该预测器可以

检测到 90% 的短路违例.

由于半导体的制造工艺不断缩小, 标准单元变得越来越小, 数量急剧增加, 较低金属层上的标准单元由于高引脚密度、低引脚可达性和有限的布线资源而经常出现设计规则违例的情况, 所以准确分析引脚图案, 预测引脚可达性十分必要. 随着卷积神经网络 (Convolutional neural network, CNN) 的广泛应用以及令人印象深刻的分类和目标检测能力, Yu 等<sup>[128]</sup>使用引脚图案为主要特征, 提出一种 CNN 模型来进行深度学习, 通过直接识别由于给定引脚模式的不良引脚可达性而导致设计规则违例的发生, 优化引脚的可达性, 指导芯片详细布局. 用于设计规则违例热点预测的机器学习模型需要同时考虑高分辨率的引脚形状图案和低分辨率的轨道和障碍物布局信息作为特征输入, 针对具有混合分辨率特征的预测问题, Liang 等<sup>[129]</sup>引入了一种新的自定义卷积神经网络技术 J-Net, 可以灵活地处理各种输入和输出分辨率要求.

然而, 上述工作都是对给定的单元布局进行局部可布线性预测, 不适用于宏布局阶段, 而设计的可布线性主要由宏布局决定, 以设计规则违例的数量来量化. 因此, 建立宏布局的可布线性预测模型以及在该模型指导下的宏布局器是十分必要的. RouteNet<sup>[130]</sup>首先将 CNN 的图像模式识别能力转移到电路中, 以捕获关于可布线性的模式, 对于 DRV 热点预测, RouteNet 的准确率比全局布线提高了 50%. Huang 等<sup>[131]</sup>在 2019 年提出了一种基于 CNN 的可布线性预测模型, 并将其嵌入到宏布局中, 通过深度学习和模拟退火优化过程获得设计规则违例最小化的良好宏布局. Chen 等<sup>[132]</sup>建立了基于全卷积网络 (Fully convolutional network, FCN) 的拥堵预测器 PROS, 可以灵活地预测元器

件布局阶段的布线拥堵情况,并帮助优化全局布线工具的全局布线成本参数.实验表明,在工业基准测试套件上,PROS可以实现较高的全局布线拥塞预测精度,平均可降低11.65%的设计规则违例.在DAC-2012<sup>[80]</sup>基准测试套件上,PROS实现了1.82%的线长估计错误率,大大优于FLUTE<sup>[34]</sup>(21.52%).在FPGA领域,Alhyari等<sup>[133-134]</sup>提出了一种基于CNN模型的新型深度学习框架DLRoute,用于预测布局的可布线性,在包含ISPD-2016<sup>[113]</sup>布线感知的布局竞赛和Xilinx<sup>1</sup>的总共372个测试用例中,在运行时间仅为几毫秒的情况下,获得了97%的可布线性预测精度.

为提高FPGA可布线性预测的效率,美国德克萨斯大学Alawieh等<sup>[135]</sup>、美国康奈尔大学Yu等<sup>[136]</sup>将问题转化为Image-to-image问题,建立了基于条件生成对抗网络(Conditional generative adversarial nets, cGAN)的机器学习模型,以预测FPGA芯片上元件布局阶段的布线拥堵情况,减少布局阶段的运行时间.然而,他们为FPGA设计的模型不能直接应用于专用集成电路芯片布线上.

Wang等<sup>[137]</sup>创新性地组合超图和格图构建格型超图,用以保留网表信息和在几何空间和拓扑空间中传播的消息,同时提出基于格型超图的异构图神经网络,通过多头监督联合学习布线需求回归和拥塞分类两个任务.在ISPD-2011<sup>[138]</sup>和DAC-2012<sup>[80]</sup>基准测试上,相对于CNN类模型U-net, F1值提升35%以上.

基于布局信息的设计规则违例预测框架<sup>[126]</sup>流程及其在芯片物理设计流程中的集成如图14所示,图14中方块表示流程,椭圆表示输入和输出;实线箭头表示用于训练和预测的路径,虚线箭头则专属于训练流程.在模型训练时,首先由布局器布局一个芯片设计,然后将布局好的设计划分为不重叠的矩形区域并提取区域的特征.同时将布局好的网表

提供给布线器进行布线,提取布线违例及其位置,标记违例出现的区域.区域特征和违例标签一同被输入到学习模型来训练预测系统,训练后的模型便能从不同的布局设计中预测出现布线违例的位置.

谷歌大脑Mirhoseini等<sup>[139]</sup>于2021年建立基于深度强化学习(Deep reinforcement learning, DRL)的芯片布局规划方法,通过对已布局好的部分宏单元和网表的特征抽取,预测布线的总线长和拥塞情况,然后选择最优的放置位置.经过在Google TPU(Tensor processing unit)和开源Ariane RISC-V CPU<sup>[140]</sup>组成的数据集上预训练和微调,在不到6h的时间内,自动生成的芯片规划图在所有关键指标(包括功耗、性能和芯片面积)上都优于或与人类绘制的设计图相近.其基于深度强化学习的线长和短路违例预测与布局规划模型<sup>[139]</sup>(如图15所示)的输入包括芯片宏单元特征、网表图邻接信息、当前宏单元标识,以及一些网表元数据,如导线、宏和标准单元的总数等.宏单元特征和网表图输入到基于边缘的图卷积网络,生成边嵌入和宏单元嵌入,并与网表元数据嵌入串联在一起,以形成单个状态嵌入,继而传递到全连接网络,全连接网络捕获有用的特征并将其用作策略网络和价值网络的输入.随后,策略网络对可以在其上放置当前宏单元的所有可能网格单元生成概率分布.在每次训练迭代中,基于DRL的智能体按顺序放置完宏单元,然后通过力引导方法放置标准单元,此过程将电路建模为弹簧系统,以便尽量缩短线长. DRL训练由一个快速但近似的布线奖励信号指导,该奖励信号采用半周长(Half-perimeter wirelength, HPWL),即近似线长,和近似拥塞(已放置网表消耗的布线资源比例)的加权平均值求得.

前述方法更多是通过可布线性预测优化布局,期望良好的布局能产生高质量的布线.上海交通大学Cheng等<sup>[141]</sup>于2021年建立元件布局信息和芯

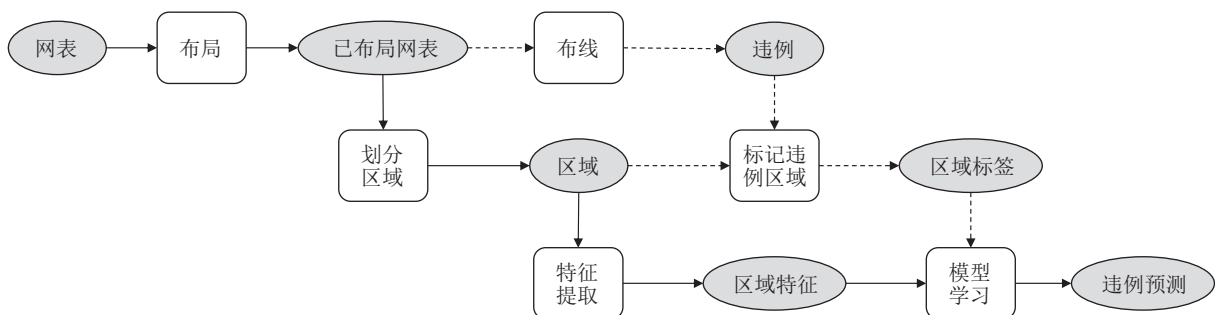


图14 基于布局信息的设计规划违例预测框架

Fig.14 Design rule violation prediction framework based on placement information

<sup>1</sup> Guelph FPGA CAD Group, <http://fpga.socs.uoguelph.ca/>

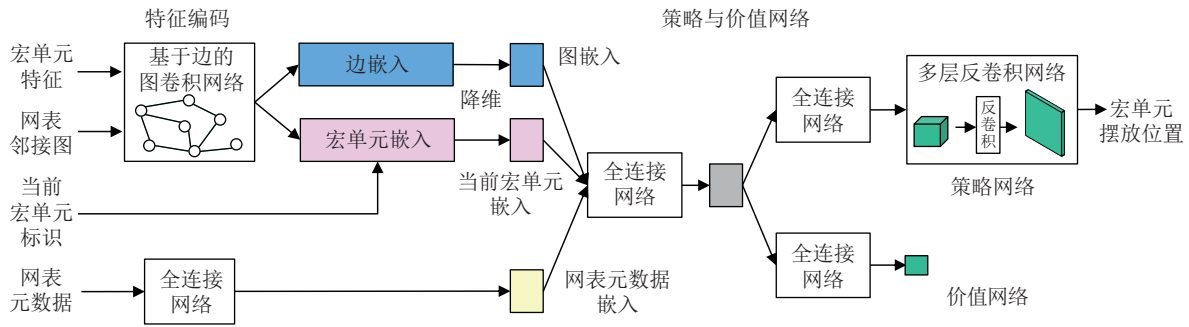


图 15 基于深度强化学习的线长和短路违例预测与布局规划模型

Fig.15 Wirelength and short violation prediction and floor planning model based on deep reinforcement learning

片布线信息融合学习的元件布局方法. 首先将强化学习与基于梯度的优化方案相结合, 提出了一种端到端学习的布局求解方法 DeepPlace, 然后通过强化学习提出了一种联合求解布局与布线任务的学习算法 DeepPR, 此外, 还设计了随机网络蒸馏来激励探索. Cheng 等<sup>[142]</sup> 在 DeepPR 后续版本中设计了一种单样本条件生成对抗网络模型, 该模型由一个特殊设计的输入尺寸自适应生成器和一个双判别器组成, 可对每个线网中的引脚进行单样本布线, 并自适应地学习线网布线的顺序. 这种联合布局布线学习的方案, 为元件布局优化提供了更为准确的反馈和奖励信息, 而优化后的布局又输出质量更高的布线, 这种相互促进、螺旋上升的联合学习方案为基于机器学习的后端设计带来了新的思路, 但 Cheng 等<sup>[142]</sup> 的布线学习仅限于全局布线, 最终指导的仍然是元件布局优化.

总的来说, 在布线的早期阶段, 如全局布线、单元布局甚至宏布局阶段, 预测可布线性和拥堵位置可以加快芯片设计. 但在基于机器学习的可布线性预测任务中, 涉及的阶段越多, 则需要建立与更多阶段相关的模型, 任务难度就越大.

### 3.2 基于全局布线优化的布线方法

当前, 基于全局布线优化的机器学习布线算法

主要研究拥塞预测、最小斯坦纳树构建和全局布线走线选择, 如表 4 所示.

拥塞预测根据不同设计阶段分为拥塞次数预测和拥塞位置预测. 拥塞计数如拥塞总数和设计规则违例的数量, 用于评估布局 and 全局布线的好坏. 而拥塞位置需要详细给出拥塞或设计规则违例的位置, 通常表示为 2D 地图. 准确预测拥塞位置能为拥堵地区预留足够的空间, 有效指导布线路径优化. Hung 等<sup>[143]</sup> 提出了一种基于 CNN 的机器学习框架, 仅基于全局布线产生的拥塞报告, 预测给定设计的详细布线所产生的 DRV 位置. 训练后的模型不仅精度明显高于以往的相关工作, 设计规则违例图在视觉上与实际违例图接近, 且违例图生成时间仅占全局布线时间的 3%. 与现有的基于传统压缩到 2D 特征的模型训练和预测方法不同, Su 等<sup>[144]</sup> 从已经放置的线网列表中提取适当的三维特征, 提出了一种基于深度学习的拥塞估计算法, 利用 U-net 模型建立拥塞估计器, 生成拥塞热图, 作为全局布线器 CUGR<sup>[60]</sup> 中初始模式布线的指南.

虽然机器学习技术在帮助预测布线拥塞和优化布线方面取得了进展, 但完成一个线网的具体连接方法大多仍然基于传统的基于搜索的算法, 如 A\* 搜索算法. 于是, Utyamishv 等<sup>[145-146]</sup> 提出一种全局布线器, 可以从布线电路中学习并自动布线未见

表 4 基于全局布线优化的布线方法  
Table 4 Routing approaches for global routing optimization

任务类型	算法框架	文献 (年份)	芯片类型	数据集
拥塞预测	CNN	Hung 等 <sup>[143]</sup> (2020)	VLSI	Foundry
	CNN	Su 等 <sup>[144]</sup> (2022)	VLSI	ICCAD19 <sup>[61]</sup>
斯坦纳树构建	VAE	Utyamishv 等 <sup>[145-146]</sup> (2022, 2020)	VLSI	ISPD98 <sup>[76]</sup>
	RL	Liu 等 <sup>[83]</sup> (2021)	VLSI	ICCAD19 <sup>[61]</sup>
走线选择	RL	Gandhi 等 <sup>[147]</sup> (2023)	VLSI	ISPD18 <sup>[10]</sup>
	RL	Liao 等 <sup>[148]</sup> (2020)	VLSI	Generated



的布局. 与传统的布线流程不同, 该方法将全局布线重新定义为二维 Image-to-image 处理问题, 并采用由变分自编码器 (Variational autoencoder, VAE) 和自定义损失组成的深度学习系统, 以统一的、单步非迭代的方式处理成像问题. ISPD-1998<sup>[76]</sup> 基准测试结果显示, 与 FastRoute 布线器相比, 可布线性只降低了 3.2%, 运行时速度提高了 5 倍以上. Liu 等<sup>[33]</sup> 则在 2021 年首次利用强化学习方法解决布线中的直线斯坦纳最小树生成问题, 该方法称为 REST, 利用直线边序列来表示 RSMT, 并设计 Actor-critic 神经网络模型生成直线边序列, REST 在中小规模线网中获得较好的实验结果, 对于 50 引脚以内的线网构造出平均长度误差小于等于 0.36% 的 RSMT, 所需的平均时间小于 1.9 ms, 比类似质量的传统启发式方法快得多.

由于芯片中庞大的布线空间和线网数量, 线网排序和走线选择是其中最重要且执行最多的两个步骤. 对于线网选择, Gandhi 等<sup>[147]</sup> 训练了一个强化学习模型 RL-Ripper 用于全局布线拆线重建过程中选择最合适的线网进行拆除以减少最终的短路违例, 虽然在 ISPD-2018<sup>[19]</sup> 布线测试评估中取得了些许的提升, 但实验结果只运行了一轮拆线重建迭代, 最终仍然存在大量的违例. 对于引脚间的走线选择问题, 美国卡内基梅隆大学的 Liao 等<sup>[148]</sup> 首次提出利用深度强化学习方法进行全局布线. 全局布线被建模成马尔科夫决策过程, 使用联合优化的深度 Q 网络 (Deep Q-network, DQN) 模型作为强化学习算法的基础, 每个线网分解成多个双引脚的线网, 智能体被训练以找出尽可能短的布线路径连接这两个引脚. 此外, 为实现快速收敛, 提出了一种参考 A\* 搜索算法的内存刻录 (Burn-in memory) 技术, 在大多数测试用例中取得比 A\* 算法更好的结果. 图 16 展示了 A\* 搜索指导的强化学习全局布线流程<sup>[134]</sup>, 首先, 问题集生成器生成具有指定大小、复杂性和约束的问题. 所有生成的问题都存储在单独的文本文件中, 其中包含描述问题所需的所有信息,

包括三维网格的大小、网格中边的容量、需布线的线网, 以及线网引脚的空间坐标. 读入并解析输入文件后, 利用最小生成树算法将每个线网进一步分解为一组双引脚连接问题, 然后使用 A\* 布线器和 DQN 布线器来解决这些问题, 首先执行 A\*, 以便为 DQN 求解器提供内存刻录. 当所有双引脚连接问题都被解决后, 属于单个线网的解会被合并. 在获得整个线网集的最终解决方案后, 根据总拥塞和总线长来评估该方案的质量.

虽然现有的基于机器学习的拥塞预测方法取得了一定效果, 但应用机器学习特别是强化学习方法解决网络的连线问题仍然处于初期阶段, 面临诸多困难, 现有方法对问题做出了很多简化, 仅考虑单个网络, 且测试集和训练集的规模都很小. Liao 等<sup>[148]</sup> 的设计仅有 50 个线网和 2 个引脚, 网格大小为  $2 \times 8 \times 8$ , 如果将其应用在规模较大的设计中, Liao 等<sup>[148]</sup> 方法的缺点是奖励过于稀疏, 仅在到达终点时才会得到有效的奖励, 这将导致训练十分困难. 因此, 仍需要考虑更完善的单步和最终奖励方案.

### 3.3 基于详细布线优化的布线方法

当前, 基于详细布线优化的机器学习布线算法主要研究违例预测与修复、线网排序、走线选择和异步并行布线, 如表 5 所示.

除了早期预防, 我们观察到一些研究引入机器学习技术来配合传统方法解决详细布线违例检测和问题修复, 帮助优化布线结果. Alpha-PD-Router<sup>[14]</sup> 是一种可以检测和纠正短路违例的布线器. 在强化学习的基础上, Alpha-PD-Router 将布线问题转化为一个双方合作的博弈问题, 其中一个参与者被训练来执行布线, 另一个参与者被训练来检测违例行为, 并进行拆线. 这一过程中两个参与者会相互合作, 一直重复直至所有线网布线完成并且不存在违例. 布线器的奖励与线网的总线长负相关, 布线器的目的是寻找一条连接源点和汇点的最短路径. 拆线器的奖励与布线器布线的难易程度有关, 其目标

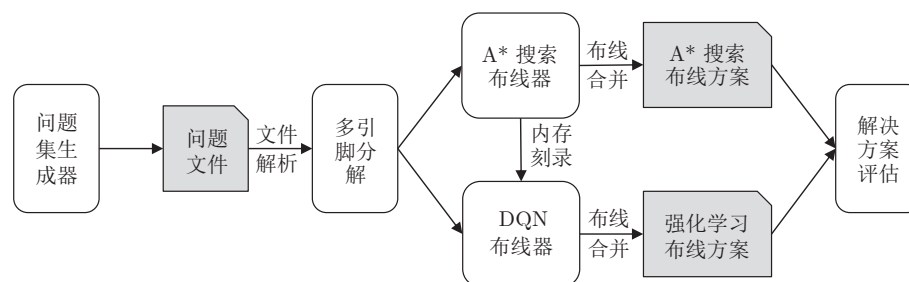


图 16 A\* 搜索指导的强化学习全局布线流程

Fig. 16 Global routing based on A\* search-guided reinforcement learning



表 5 基于详细布线优化的布线方法  
Table 5 Routing approaches for detailed routing optimization

任务类型	算法框架	文献 (年份)	芯片类型	数据集
违例预测与修复	RL	Gandhi 等 <sup>[14]</sup> (2019)	VLSI	Generated
	RL	Ren 等 <sup>[149]</sup> (2021)	VLSI	Foundry
	Random-Forest	Siddiqi 等 <sup>[150]</sup> (2022)	FPGA	VPR 7 <sup>[151]</sup>
线网排序	RL	Lin 等 <sup>[152]</sup> (2022), Qu 等 <sup>[153]</sup> (2021)	VLSI	ISPD18 <sup>[19]</sup> & ISPD19 <sup>[20]</sup>
走线选择	MCTS + RL	He 等 <sup>[15]</sup> (2022)	VLSI	Generated
	RL	Liao 等 <sup>[154]</sup> (2020)	VLSI	Foundry
	RL	Chen 等 <sup>[155]</sup> (2023)	VLSI	Foundry
异步并行布线	MARL	Ju 等 <sup>[156]</sup> (2021)	VLSI	Generated

是寻找一个最优的拆线策略以使布线更加容易. 但为了减小设计的复杂度, 加快整个训练流程, AlphaPD-Router 使用的设计仅有  $1 \times 5 \times 5$  方格的大小, 而且拆线器仅考虑了短路违例, 实际芯片设计的规模和训练所需要的资源显然要庞大得多. 在标准单元内部布线中, 由于布线空间狭小, 以及大量设计规则的限制, 详细布线自动化亦十分困难. Ren 等<sup>[149]</sup> 使用遗传算法创建初始布线候选方案, 然后使用强化学习逐步修复违反设计规则的区域. 设计规则检查器将违例行为反馈给强化学习智能体, 智能体根据数据学习如何修复违例. 在一些标准单元上的实验表明, 该方法可以完成复杂且被认为无法通过手工完成的标准单元的布线, 将单元大小减少 11%. Siddiqi 等<sup>[150]</sup> 在 2022 年首次提出通过先预测布线资源的拥塞代价, 再预加载拥塞代价来加速 FPGA 布线的思想. 该方法通过有监督的机器学习算法, 根据电路网表、电路布局和 FPGA 布线网络的特点, 来预测 FPGA 拥塞和热点区域的布线代价. 这些预测成本用于指导布线器在线网布线时避开高度拥挤的区域, 从而减少迭代和拆线重布操作的总数. 在 VPR 7<sup>[151]</sup> 基准测试集上的实验表明, 通

过上述方法增强的 PathFinder<sup>[30]</sup> 能够在保证方案质量下, 减少 43% 的布线迭代和 28.6% 的运行时长.

详细布线因需要处理复杂的设计规则和庞大的问题规模, 其性能和结果对线网的布线顺序非常敏感. 以往文献中布线顺序主要由为特定基准优化的简单启发式规则确定. 2022 年, Lin 等<sup>[152]</sup> 利用异步强化学习框架 A3C, 基于 Dr.CU<sup>[91]</sup> 给出的详细布线结果, 学习并优化线网布线先后顺序的策略, 并提出了一个迁移学习算法来提高性能. 与 Dr.CU 详细布线器相比, 减少了 26% 的 DRC 违例和 1.2% 的总成本. 线网排序强化学习框架中的策略网络模型和价值网络模型<sup>[152-153]</sup> 如图 17 所示. 为将网络架构与线网数量解耦, 模型引入了一个线网特征编码网络, 对每个线网的上一次布线的 22 维特征向量: 线网尺寸 (边界框的半周长)、布线区域内竞争资源的线网数量、被重布线的次数、违例数量、通孔数、线长和层分配情况, 独立进行编码, 并将编码特征连接到策略和价值网络作为输入. 例如, 给定一个具有  $E$  个线网的设计, 编码器将  $\mathbf{R}^{E \times 22}$  的输入特征向量编码为  $\mathbf{R}^{E \times 64}$  向量. 策略网络获取此向量并为所有线网生成一个排序分数分布数组, 即  $\mathbf{R}^{E \times 2}$

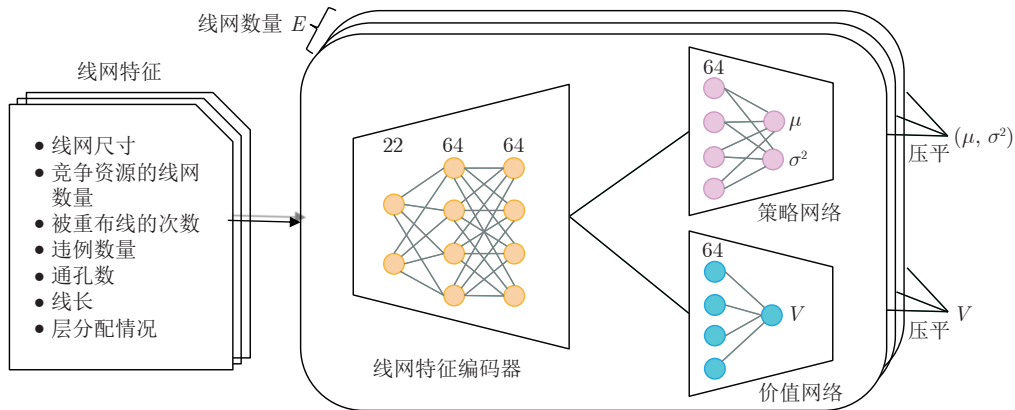


图 17 线网排序强化学习框架中的策略网络模型和价值网络模型

Fig.17 Policy network and value network in the reinforcement learning framework for net ordering

(每个线网的概率分布的均值为  $\mu$ , 方差为  $\sigma^2$ ), 然后从每个线网的正态分布中采样获得其排序分数, Dr.CU 根据线网分数顺序布线. 值网络则将编码器输出的向量输入到全连接层, 并输出一个标量.

详细布线的核心走线算法大多是根据领域知识手工设计或根据特定设计规则设计的, 实现很费力, 且当出现新的设计约束和目标时, 需要人工更新. He 等<sup>[15]</sup>提出一种新的电路布线方法, 将其建模为一个顺序决策问题, 并通过蒙特卡洛树搜索 (Monte Carlo tree search, MCTS) 和深度强化学习的策略引导的扩展 (Rollout) 来解决它. 所提方法可以支持不同的布线约束和目标, 而不需要改变算法本身, 但所提方法的实验规模都比较小, 只有  $1 \times 32 \times 32$ , 最多 10 个线网, 每个线网只有 2 个引脚, 这些限制使得算法还无法应用在实际芯片布线中. 美国卡内基梅隆大学的 Liao 等<sup>[154]</sup>提出了基于注意力机制的深度强化学习模型以解决详细布线的轨道分配问题, 该模型的主要创新是使用监督作为筛选, 使用传统遗传算法产生解决方案的方法, 最小化强化学习策略模型的输出和从遗传求解器获得的解分布之间的 Kullback-Leibler 散度损失. 这样布线器可以在有监督的离线环境中学习策略, 产生的解的质量接近于遗传算法产生的解的质量的同时, 比遗传求解器的运行时间性能提高近 100 倍. 该方法适用于具有数千个线网的模拟芯片的小型基准测试. Chen 等<sup>[155]</sup>设计了一种基于图神经网络的近端策略优化的深度强化学习模型, 并通过该模型引导详细布线, 以处理定制电路的复杂设计约束和严格设计规则. 该强化学习模型在三个混合信号定制电路布线测试中取得了较好的布线效果, 其中最大的电路具有 124 个单元、122 个线网和 620 个

引脚.

芯片详细布线通常采用基于拆线重布的顺序布线来避免布线区域重叠的线网之间的冲突, 但这一方法对线网的排序比较敏感, 为了解决这一挑战, Ju 等<sup>[156]</sup>将传统的详细布线转化为多智能体路径规划任务, 提出了一种基于多智能体深度强化学习的异步详细布线器, 利用智能体之间的异步通信来避免布线冲突. 评估结果表明, 其所提出的多智能体强化学习方案比基线学习方法性能提高了 11.6%. 但该方法的测试用例规模比较小, 只使用  $3 \times 7 \times 7$  的网格, 其中只有两个线网, 每个线网只有两个引脚, 与真实芯片布线的规模相差较大. 图 18 展示了基于多智能体强化学习的详细布线框架<sup>[156]</sup>, 布线器将每个线网视为独立的智能体, 将引脚连接任务视为路径规划. 每个智能体有一个局部视野, 以减少特征大小和训练难度, 智能体利用自身和附近智能体的局部视野信息在网格图中异步地做出布线决策. 为了消除布线拥塞, 框架为每个智能体的信息通信设置一个信息存储单元. 布线时, 多个智能体同时从起始点开始, 当到达一个节点时, 每个智能体通过自己的观察和从信息存储单元中提取的其他智能体的信息来构建自己的状态空间. 每个智能体将状态特征输入动作决策模块, 以选择下一个节点. 最后, 智能体采取行动并向信息存储单元提交新信息.

从以上分析可以看出, 当前基于机器学习的详细布线方法研究还比较少, 虽然这些方法在设计规则违例预测和线网排序上都取得了一定的突破, 但要完成一款大规模芯片的详细布线, 还要解决很多现实困难.

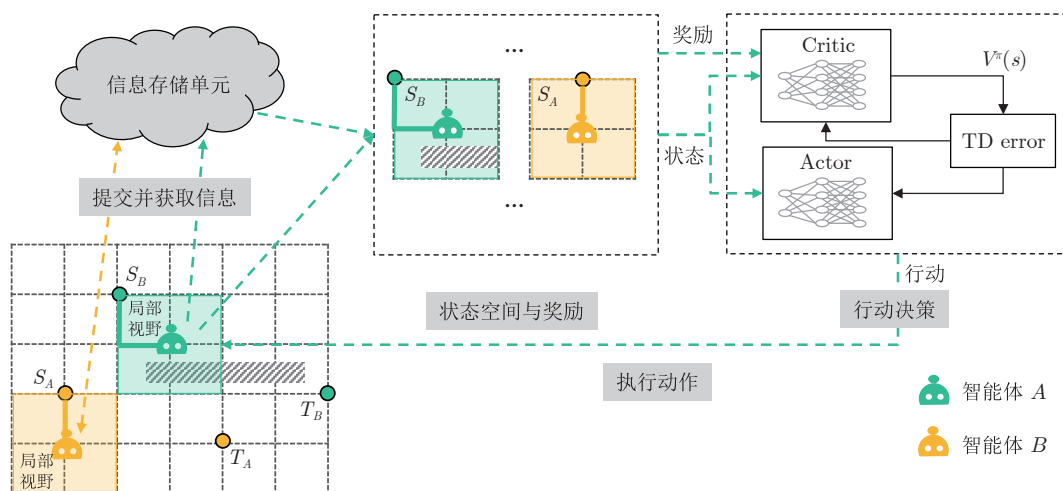


图 18 基于多智能体强化学习的详细布线框架

Fig. 18 Detailed routing framework based on multi-agent reinforcement learning

### 3.4 小结

总的来说, 虽然现有基于机器学习的芯片设计方法在一定程度上缓解了依赖人工设计启发策略的问题, 但在芯片布线领域, 特别是针对具有庞大布线空间和线网数量, 以及大量复杂的空间和电气约束的工业级芯片的布线问题, 基于学习的自动布线技术应用还处于非常初级的阶段. 目前面向芯片布线的机器学习算法主要集中在可布线性预测、设计规则违例预测与修复等个别环节上, 缺少可以直接预测芯片“布线动作”的智能学习方法, 现有方法中使用的测试用例的规模都比较小, 扩展性较弱, 尚无法直接处理工业级芯片, 也很难纳入商业设计工具进行持续优化. 另外, 基于机器学习的智能布线方法还存在以下几个问题:

1) 机器学习擅长解决计算机视觉中的特征提取、分割、分类和预测等问题, 大多数基于机器学习的布线算法都优先将预测问题转化为图像处理问题, 但这样做需要考虑哪些信息应该体现在图像上、分辨率需要多高等问题, 特别是芯片中存在密集的引脚层和稀疏的布线层.

2) 为了获得训练良好的模型, 基于机器学习的布线算法必须在训练前准备大量的标记数据. 在设计数据稀缺的物理设计中, 许多算法会根据一个布局生成多个设计结果, 例如布局结果和违例结果, 来构建训练数据集. 但由于违例区域可能只占整个布局的一小部分, 构造的数据集可能导致数据不平衡问题.

3) 目前尚无一个相对完善的芯片布线机器学习环境, 用以训练、测试和验证机器学习算法在芯片布线上的应用. 缺少电路基础的计算机算法人员需要花费大量的时间了解芯片设计过程, 才能比较准确地提取不同训练任务的状态特征和奖励指标.

4) 强化学习算法的探索空间大、学习效率低. 对于芯片布线来说, 不同芯片、芯片的不同区域或是已部分布线的中间状态对于基于强化学习的布线算法来说可能都是一个新的问题. 另外, 一个线网的成功布线不代表所有线网都能布通, 中间步骤奖励与最终奖励的设置十分关键, 这给强化学习算法带来很大的挑战.

5) 基于学习的算法相对于基于搜索的算法通常需要更多的计算资源和运行时间.

虽然在芯片自动布线工具中应用机器学习方法存在着上述诸多困难, 但基于机器学习的智能布线方法也有其优势, 仍是当前研究的热点, 也是未来的趋势. 关键是如何融合基于机器学习的布线方法和基于规划搜索的布线方法, 以完成大规模芯片的

高质量高效率布线.

## 4 公开数据集及开源工具

### 4.1 公开数据集

ISPD、ICCAD 和国内的集成电路 EDA 设计精英挑战赛每年都会举办芯片设计领域的多项赛事, 吸引全球的科研机构的研究人员共同参与, 其中不少比赛与芯片布线相关. 在这些比赛中, 基准测试用例和数据集大多来自于工业级规模芯片, 为学术界进行真实芯片自动布线研究提供了重要支撑. 表 6 展示了全局布线、详细布线和 FPGA 布线的常用公开数据集, 每个数据集中都包含多个大规模测试用例和相关评测指标和工具.

### 4.2 开源工具

芯片布线领域常用开源工具包括以下内容.

1) TritonRoute<sup>1</sup>: 美国加利福尼亚大学圣迭戈分校开源的一个包含全局布线和详细布线的端到端芯片自动布线器, 该布线器以 LEF 和布局后的 DEF 作为输入, 包含全局布线、层分配、拥塞分析、引脚连接分析、轨道分配、详细布线、设计规则违例检查与修复等功能, 对 ISPD-2018<sup>[19]</sup> 和 ISPD-2019<sup>[20]</sup> 详细布线比赛中的大多数测试用例都能输出无设计规则违例的布线方案. TritonRoute 的目标是实现面向工业的可扩展和灵活的流程, 目前已整合到开源 EDA 工具 OpenROAD<sup>[102]</sup> 中.

2) CUGR<sup>2</sup>: 香港中文大学开源的一个全局布线器. 与以往使用预估线长和拥塞情况来评估布线质量的全局布线器不同, CUGR 是一个详细布线可布线性驱动的全局布线器, 使用详细布线器, 如 Dr.CU<sup>[91]</sup> 的最终布线结果作为评估标准. CUGR 通过多种方式生成布线指导, 包括设计一个复杂的基于概率的成本方案、结合模式布线和层分配的三维模式布线、具有不同代价函数和目标的两级三维迷宫布线, 以及多种“补丁”技术以完善布线指引. CUGR 支持对 ICCAD-2019<sup>[61]</sup> 基于 LEF/DEF 的全局布线数据集中的芯片用例的布线.

3) Dr.CU<sup>3</sup>: 香港中文大学开源的一个详细布线器, 获得过 ISPD-2019<sup>[20]</sup> 详细布线比赛的第一名. Dr.CU 详细布线器以芯片的 LEF、DEF 以及布线指导作为输入, 设计了一套两级稀疏数据结构, 改进了 Dijkstra 算法, 提出边构造边修正的路径搜索并行布线方案, 最终输出已完成布线的 DEF 格式设计文件. Dr.CU 支持对 ISPD-2018<sup>[19]</sup> 和 ISPD-2019<sup>[20]</sup> 详细布线数据集中的芯片用例的布线.



表 6 芯片布线常用公开数据集  
Table 6 Public datasets commonly used for chip routing

类型	数据集	测试用例	数据集特点	评测指标
全局布线	ISPD-2007 <sup>[17]</sup>	16 个	8 个单层和 8 个 6 ~ 8 层测试用例, 多层用例线网数量从 176 715 个到 548 073 个不等	溢出数、总线长
	ISPD-2008 <sup>[18]</sup>	16 个	在 ISPD-2007 多层用例基础上再增加 8 个 6 ~ 8 层测试用例, 线网数量从 176 715 个到 1 647 410 个不等	溢出数、总线长、运行时长
	ICCAD-2019 <sup>[61]</sup>	12 个	从 ISPD-2018 和 ISPD-2019 数据集中选取了 6 个 32 nm 的测试用例, 通过限制用例层数为 5 层, 形成额外的 6 个用例	将全局布线输出的布线指导导入 Dr.CU <sup>[91]</sup> , 并评估详细布线后的设计规则违例数、通孔数、总线长、运行时长
	ICCAD-2020 <sup>[67]</sup>	12 个	3 ~ 16 层, GCell 数量最大为 277 × 277, 线网数量从 6 个到 332 080 个不等	零溢出、零开路、最大单元移动限制、1 小时运行时长限制
详细布线	ICCAD-2021 <sup>[68]</sup>	10 个	3 ~ 16 层, GCell 数量最大为 277 × 277, 线网数量从 6 个到 332 063 个不等	零溢出、零开路、电压区域限制、最大单元移动限制、1 小时运行时长限制
	ISPD-2018 <sup>[19]</sup>	10 个	45 nm 到 32 nm, 9 层, 线网数量从 3 153 个到 182 000 个	设计规则违例数、通孔数、总线长、运行时长
FPGA 布线	ISPD-2019 <sup>[20]</sup>	10 个	65 nm 到 32 nm, 5~9 层, 线网数量从 3 153 个到 895 253 个	设计规则违例数、通孔数、总线长、运行时长
	VPR <sup>[151]</sup>	19 个	来源于多款应用芯片, 六输入查找表数量从 174 到 107 784	总线长、关键路径延时
	Titan <sup>[157]</sup>	23 个	来源于多款应用芯片, 六输入查找表数量从 24 759 到 805 063, 原语数量从 90 778 到 1 859 485	总线长、关键路径延时

4) EDA-AI<sup>4</sup>: 上海交通大学开源的一个基于布局布线联合强化学习的元件布局优化方法的实验环境. 其中深度强化学习智能体先按顺序放置宏单元, 然后选择线网, 基于单样本条件生成对抗网络模型生成线网的全局布线方案, 并根据预估线长和拥塞结果指导宏布局和线网选择. EDA-AI 中的 DeepPlace 和 DeepPR 方法支持对 ISPD-1998<sup>[76]</sup> 和 ISPD-2005<sup>[77]</sup> 的布局和全局布线.

5) VTR (包含 VPR)<sup>5</sup>: 是一个支撑 FPGA 架构从 Verilog 到布线的开源框架. 其中布局布线工具 VPR 已广泛应用于学术界研究不同架构 FPGA 的布局布线过程, VPR 以经典的模拟退火算法为基础, 支持时序驱动的布局布线, 并具有图形化操作界面.

## 5 研究挑战与未来方向

前文介绍了多种基于规划搜索的智能布线方法和基于机器学习的智能布线方法, 这些芯片智能布线方法的对比分析如表 7 所示. 表 7 中根据算法通常处理的线网数量, 分为单线网布线方法和多线网布线方法, 再按照方法类型分为规划搜索类和机器

学习类布线方法, 并给出每种算法主要的应用阶段. 然后从布线效率、布线质量、CPU/GPU 资源要求、内存资源要求、支持布线规模、芯片数据要求六个角度分析给出每种算法的大致表现. “●”表示效率高、质量好、要求低或者规模大; “○”表示效率低、质量差、要求高或者规模小; “◐”表示介于“●”和“○”的水平之间. 从表 7 中可以看出, 每种布线方法都有各自的特点, 在不同的芯片发展时期, 适应不同的布线需求, 组合形成了不同的布线器.

布线问题属于典型的多目标优化问题, 变量空间大, 难以寻找全局最优解, 需要反复进行启发式探索. 无论是在全局布线还是详细布线, 无论是基于模式布线的初次布线连接, 还是基于拥塞协商的拆线重布, 或是基于 Dijkstra 算法和 A\* 搜索算法的路径选择, 基于规划搜索的布线方法在布线效率、布线质量和处理规模上仍然有很大的优势, 仍然是当前商业布线器采用的主流方案. 这些方案在真实芯片持续的布线高要求下不断得到完善和提升, 但其中仍有不少挑战需要突破. 因此, 基于规划搜索的智能布线方法仍是未来布线研究的重要方向, 主要体现在如下四个方面:

1) 并行化提升布线效率. 基于规划搜索的布线算法主要使用 CPU 计算资源, 没有充分利用 GPU 资源. 随着多核 CPU 与 GPU 越来越普及, 处理性能越发强大, 如何充分利用处理器资源, 通过并行化或 CPU/GPU 组合调度加速芯片布线过程成为芯片布线研究的热点, 特别是在 FPGA 布线领域. 但并行加速也会带来布线资源占用上的冲突, 因此

<sup>1</sup> TritonRoute, UCSD Detailed Router, <https://github.com/The-Open-ROAD-Project/TritonRoute>

<sup>2</sup> CUGR, VLSI Global Routing Tool Developed by CUHK, <https://github.com/cuhk-eda/cu-gr>

<sup>3</sup> Dr.CU, VLSI Detailed Routing Tool Developed by CUHK, <https://github.com/cuhk-eda/dr-cu>

<sup>4</sup> EDA-AI, <https://github.com/Thinklab-SJTU/EDA-AI>

<sup>5</sup> Verilog to Routing, Open Source CAD Flow for FPGA Research, <https://github.com/verilog-to-routing/vtr-verilog-to-routing>



表 7 智能布线方法的对比分析  
Table 7 Comparison and analysis of intelligent routing approaches

线网数量	方法类型	布线方法	布线阶段	布线效率	布线质量	CPU/GPU 要求	内存要求	布线规模	数据要求
单线网	规划搜索类	迷宫布线	GR/DR	○	●	○	○	●	●
		线搜索布线	GR/DR	●	●	●	●	●	●
		A* 搜索布线算法	GR/DR	●	●	●	●	●	●
		模式布线	GR/DR	●	●	●	●	●	●
		查表布线	GR	●	●	●	●	●	●
机器学习类	布线路径生成	GR	●	○	○	●	○	○	
	强化学习布线	GR/DR	○	●	○	●	○	●	
多线网	规划搜索类	贪心通道布线	DR	●	●	●	●	○	●
		整数线性规划布线	GR	○	●	○	○	●	●
		拥塞协商	GR/DR	●	●	●	●	●	●
		多级布线	GR/DR	●	●	○	○	●	●
		并行布线	GR/DR	●	●	○	○	●	●
机器学习类	拥塞与违例预测	GR/DR	●	●	○	●	●	○	
	多智能体布线	GR/DR	●	●	○	○	○	●	

需要研究布线区域和冲突线网的划分策略, 布线进程间的高效通信以通过协商减少冲突. 并行化后布线任务需要分配到不同的进程或设备上, 等待任务结束后汇总结果, 因此需要研究并行布线的任务分配与负载均衡策略. 另外, 由于布线顺序和轨道资源占用上的不同, 并行布线可能会造成布线结果的不确定性, 满足串行等效约束的并行布线也是一个研究方向.

2) 适配最新工艺和设计规则要求. 摩尔定律和电子产品更新换代的需求促使芯片晶体管密度不断提高, 集成模块越来越多, 芯片面积也越来越大. 为了保证良品率, 芯片制造工艺也在不断提升, 工艺的改变带来了新的设计规范和约束要求, 其中对布线的影响尤为明显. 基于规划搜索的布线算法在处理设计规则和约束时常常依赖于人工设定启发式策略, 无法动态调整, 这些算法需要在专家指导下根据工艺要求进行策略调整、优化和完善. 因此, 研究最新工艺和设计规则的布线适配方法, 集成芯片布线方法, 高密度引脚的连接分析, 最优通孔类型选择, 设计规则自动检查引擎开发, 以及工艺友好的布线系统都是提高布线系统可扩展性的研究主题.

3) 优化布线算法, 提高布线质量. 随着芯片规模不断扩大、复杂程度不断提高, 布线算法的解空

间变得极其巨大, 重布线的次数也在不断增加, 原来高效的布线算法可能新的芯片规模、线网数量、复杂度下变得低效、不稳定, 甚至在允许时间范围内无法成功布线. 因此, 对现有布线算法的优化和新算法的设计同样重要. 对布线算法的优化包括轨道并行分配的协商方法、布线路径搜索的启发式策略、拥塞或违例成本的快速准确评估方法、线网排序策略和拆线重布选择方法、3D 绕障斯坦纳最小树的快速构建以及时间驱动的布线等. 高效的方法和策略能够大大提高布线效率和质量.

4) 整合布局布线方法, 实现端到端布线. 芯片布线通常在布局结束后才开始, 以布局结果、网表和设计规则作为输入, 输出满足设计规则要求的所有线网的物理连接方案. 布线又分为全局布线和详细布线, 而详细布线是整个芯片设计流程中最复杂和最耗时的部分. 当布线结果不能满足功能和性能设计要求时, 就不得不在布线阶段或从布局甚至更早的阶段开始进行迭代优化, 这种反复迭代耗费了芯片设计工程师相当多的时间和精力. 为了更好地解决这个问题, 改善布线的效率和效果, 需要充分整合布局布线, 即在布局时通过预测最终的布线结果调整布局, 在布线效果不好时主动微调单元的布局位置以修复设计规则违例. 因此, 研究如何融合

布局规划、详细布局、全局布线和详细布线是提高芯片最终输出质量的重要保证。研究方向包括全局详细布线器、支持单元移动的全局布线、模块化的开源布线框架等。

对于复杂的多目标优化问题,机器学习技术非常适合于此类高维数据空间问题的求解,近年来已经涌现出大量的基于机器学习的布线理论和方法,并已逐步在布线工具中的个别环节中有所应用,如布线性预测、拥塞和违例预测等。同时,强化学习算法也在斯坦纳树构建、线网排序、轨道分配等困难问题中有了一些尝试,体现出一定的潜力。但在公平的比较下,现有基于学习的技术在效率和解的质量上都难以系统地超越基于规划搜索的布线算法。大多数基于学习的技术只在具有数百个线网的小电路上工作得很好,但一个实用的布线引擎需要在一个超大型 3D 网格图(大于  $10 \times 10\,000 \times 10\,000$ )上有效地处理数十万甚至上百万个线网,并产生高质量的布线解决方案。这对于现有机器学习类算法来说,存在不少困难。因此,基于机器学习的智能布线方法也是未来布线研究的重要方向。因为这一类方法的研究时间不长,目前仍存在很多开放性研究课题,主要体现在如下六个方面:

1) 利用机器学习技术优化布线的特定环节。现代布线器由于需要处理大规模芯片电路的布线,布线环节较多,模块结构复杂,基于机器学习的布线方法很难覆盖原有布线方法的方方面面,况且现有基于机器学习的布线算法普遍性能不高,能处理的芯片规模较少,研究者需要评估和分析哪些布线环节或模块利用机器学习方法处理会更具有优势,然后设计算法进行替换。成功替换后将使得基于机器学习的布线方法整合到工业 EDA 芯片布线器中进行持续学习、测试、验证和优化。目前比较合适的替换环节包括可布线性预测、拥塞和违例预测、线网排序等。另外,目前开源的布线器较少,而且大多为了性能做出了特定优化,不同模块之间的耦合度较高,不便于用机器学习方法替换其中某一布线步骤或模块。因此,除了探索更加高效的求解学习类算法以外,设计模块化的开源布线框架也是一个具有重要意义的研究方向,这不仅有利于将机器学习算法应用到布线中,也方便针对单一步骤或模块做出优化或者考虑更多工艺设计的要求,同时算法结果也便于相互比较。

2) 构建开源的芯片布线机器学习环境。芯片设计文件是企业的核心知识产权,目前开源芯片的设计较少,而机器学习算法需要大量数据进行模型训

练,生成和规范化学习数据十分重要。另外,目前公开文献中没有一个相对完善的开源芯片布线机器学习环境,用以训练、测试和验证机器学习算法在芯片布线上的应用。而缺少电路基础的计算机算法人员需要花费大量的时间了解芯片设计过程,才能准确地提取不同训练任务的状态特征,设计合理的动作和奖励指标。建议对布线器的布线环节和模块逐步解耦,将特定环节或模块剥离出来,研究构建一个开源的芯片布线机器学习环境,同时设计并构建标准且开放的芯片布线公开数据集,用以训练、测试和验证机器学习算法在芯片布线上的应用,并逐步扩大学习范围,覆盖更多的布线环节和模块,研究在布线中融合智能规划和机器学习算法。

3) 控制学习环境的规模,实现从易到难的布线课程学习(Curriculum learning)。工业级规模的芯片在详细布线时每层可布线的轨道可能上万,需要连接的线网可能超过百万,状态和策略空间规模都特别大,学习算法常常难于收敛,布线结果不稳定。现有基于规划搜索的智能布线算法大多采用分而治之的方式将芯片分区,从而降低芯片的规模,但分区后的布线空间仍然很大,而且需要考虑布线结果的整合问题。对于不同类型的机器学习类布线算法的训练和测试,芯片布线机器学习环境应提供不同规模大小的学习环境和数据集,研究从易到难的布线课程学习设计。

4) 探索多目标驱动的布线学习算法,提高布线质量。芯片布线追求的是全局最优的布线方案,在真实芯片布线的过程中,需要将线长、通孔数、违例数、时延、拥挤度、功耗等诸多因素联合起来进行考虑,以优化芯片功率、性能和面积,满足未来芯片全方位发展的需求。因为布线环境的复杂性和千变万化,目前还没有特别有效的基于学习的布线方法能够在有限的布线时间和资源内实现多个目标之间的平衡。因此,探索高效的多目标驱动的布线学习算法是芯片布线研究的一个重要方向。另外,从纵向上看芯片布线过程可以发现,单个线网的最优布线方案在一个多线网区域不一定是最优的,单个区域的最优布线方案在整个芯片看来也不一定是最优的,所以布线时需要反复迭代。在基于强化学习的布线方法取得一定成果后,研究分层强化学习布线方法有助于平衡在网络层、区域层和芯片层的不同层次的优化目标。从横向上看芯片布线过程,布局、全局布线、详细布线都会影响最终的布线结果。研究在布线学习中整合全局布线和详细布线、整合布局布线,甚至横跨多个设计流程,由上一环节提供

指导,下一环节提供反馈,形成闭环,可以在获得高质量反馈的同时,驱动各环节联合学习。

5) 探索基于学习的并行布线方法,提高布线效率。基于机器学习的智能布线算法相对于基于规划搜索的智能布线算法通常需要更多的计算资源和运行时间。研究简单有效的基于机器学习的并行训练和并行布线方法能充分利用设备上的多核 CPU 和 GPU 资源,提高算法训练速度和布线效率。多智能体强化学习方法在芯片布线中的应用也是一个并行布线的研究方向。

6) 探索基于智能规划的模型学习与推理的布线方法,提高布线质量。芯片布线涉及的布线动作相关的模型可以从历史芯片布线数据中进行学习<sup>[158-161]</sup>,同时结合不同芯片之间的布线相似性实现布线动作模型的迁移学习<sup>[162-163]</sup>,为芯片布线启发式搜索提供模型基础<sup>[164]</sup>。同时,利用布线规划识别技术<sup>[165-167]</sup>有望为芯片布线搜索提供更有潜力的启发式信息,以进一步提高芯片布线质量。

## 6 结束语

布线是芯片设计自动化流程中至关重要也是特别耗时的一环,直接影响最终产品的面积、成本、功耗、速度和可靠性。芯片自动布线问题是一个多约束、多目标和多任务的 NP 困难问题。布线问题规模庞大、任务复杂,布线环境千差万别,设计规则约束随着技术发展不断提升,因此芯片布线一直是 EDA 领域炙手可热的研究问题之一。本文对该领域的任务定义、研究难点、布线方法、常用公开数据集、评价指标等进行了综合梳理,重点综述了规划搜索类算法在 VLSI 和 FPGA 布线的最新研究成果,剖析了机器学习类算法在元件布局优化、全局布线和详细布线各环节的框架设计、应用效果,以及与传统算法的融合思路。

在未来相当长的一段时间内,基于规划搜索的智能布线方法仍将是重要的研究方向,特别是多核并行布线技术。虽然现有的学习类布线器仍无法超越或是接近传统布线器,但学习类布线算法的研究时间远比传统算法短,仍可期待后续的突破。随着技术的进步,集成机器学习来解决布线问题是一个明显的趋势。在此发展过程中,将不再为了应用机器学习而将先进的技术引入到布线问题中,而是期待开发适合布线问题特点的机器学习方法。

机器学习方法在半导体行业中正实践着一个正反馈的循环,它的出现加速了芯片的迭代过程;反之,芯片的迭代也促进了机器学习方法的性能提

升,正逐步实现“用摩尔定律来加速摩尔定律”的这个愿景,更好地支撑万物互联、万物智能时代的应用需求。

## References

- 1 Karp R M. Reducibility among combinatorial problems. In: Proceedings of the Symposium on the Complexity of Computer Computations. New York, USA: Springer, 1972. 85-103
- 2 Garey M R, Johnson D S. The rectilinear Steiner tree problem is NP-complete. *SIAM Journal on Applied Mathematics*, 1977, **32**(4): 826-834
- 3 Chen H Y, Chang Y W. Global and detailed routing. *Electronic Design Automation*. Burlington: Morgan Kaufmann, 2009. 687-749
- 4 Sherwani N A. *Algorithms for VLSI Physical Design Automation* (3rd edition). New York: Springer, 1999.
- 5 Huang G Y, Hu J B, He Y F, Liu J L, Ma M Y, Shen Z Y, et al. Machine learning for electronic design automation: A survey. *ACM Transactions on Design Automation of Electronic Systems*, 2021, **26**(5): Article No. 40
- 6 Lopera D S, Servadei L, Kiprit G N, Hazra S, Wille R, Ecker W. A survey of graph neural networks for electronic design automation. In: Proceedings of the 3rd Workshop on Machine Learning for CAD (MLCAD). Raleigh, USA: IEEE, 2021. 1-6
- 7 Gubbi K I, Beheshti-Shirazi S A, Sheaves T, Salehi S, Manoj P D S, Rafatirad S, et al. Survey of machine learning for electronic design automation. In: Proceedings of the Great Lakes Symposium on VLSI. Irvine, USA: ACM, 2022. 513-518
- 8 Tian Chun-Sheng, Chen Lei, Wang Yuan, Wang Shuo, Zhou Jing, Zhang Yao-Wei, et al. Review on technology of placement and routing for the FPGA. *Acta Electronica Sinica*, 2022, **50**(5): 1243-1254  
(田春生, 陈雷, 王源, 王硕, 周婧, 张瑶伟, 等. 面向 FPGA 的布局与布线技术研究综述. 电子学报, 2022, **50**(5): 1243-1254)
- 9 Yan J C, Lyu X L, Cheng R Y, Lin Y B. Towards machine learning for placement and routing in chip design: A methodological overview. arXiv preprint arXiv: 2202.13564, 2022.
- 10 Tang H, Liu G G, Chen X H, Xiong N X. A survey on Steiner tree construction and global routing for VLSI design. *IEEE Access*, 2020, **8**: 68593-68622
- 11 Qu Tong, Gai Tian-Yang, Wang Shu-Han, Su Xiao-Jing, Su Ya-Juan, Wei Ya-Yi. Research progress of VLSI detailed routing algorithm. *Microelectronics and Computer*, 2021, **38**(11): 1-6  
(屈通, 盖天洋, 王书涵, 苏晓菁, 粟雅娟, 韦亚一. VLSI 详细布线算法研究进展. 微电子学与计算机, 2021, **38**(11): 1-6)
- 12 Li L, Cai Y C, Zhou Q. A survey on machine learning-based routing for VLSI physical design. *Integration*, 2022, **86**: 51-56
- 13 Posser G, Young E F Y, Held S, Li Y L, Pan D Z. Challenges and approaches in VLSI routing. In: Proceedings of the International Symposium on Physical Design. Virtual Event: ACM, 2022. 185-192
- 14 Gandhi U, Bustany I, Swartz W, Behjat L. A reinforcement learning-based framework for solving physical design routing problem in the absence of large test sets. In: Proceedings of the 1st Workshop on Machine Learning for CAD (MLCAD). Cambridge, Canada: IEEE, 2019. 1-6
- 15 He Y B, Li H B, Jin T, Bao F S. Circuit routing using Monte Carlo tree search and deep reinforcement learning. In: Proceedings of the International Symposium on VLSI Design, Automa-



- tion and Test (VLSI-DAT). Hsinchu, China: IEEE, 2022. 1–5
- 16 Chen Chun-Zhang, Ai Xia, Wang Guo-Xiong. *Physical Design of Digital Integrated Circuits*. Beijing: Science Press, 2008. (陈春章, 艾霞, 王国雄. 数字集成电路物理设计. 北京: 科学出版社, 2008.)
- 17 Nam G J, Yildiz M, Pan D Z, Madden P H. ISPD placement contest updates and ISPD 2007 global routing contest. In: Proceedings of the International Symposium on Physical Design. Austin, USA: ACM, 2007. Article No. 167
- 18 Nam G J, Sze C, Yildiz M. The ISPD global routing benchmark suite. In: Proceedings of the International Symposium on Physical Design. Portland, USA: ACM, 2008. 156–159
- 19 Mantik S, Posser G, Chow W K, Ding Y X, Liu W H. ISPD 2018 initial detailed routing contest and benchmarks. In: Proceedings of the International Symposium on Physical Design. Monterey, USA: ACM, 2018. 140–143
- 20 Liu W H, Mantik S, Chow W K, Ding Y X, Farshidi A, Posser G. ISPD 2019 initial detailed routing contest and benchmark with advanced routing rules. In: Proceedings of the International Symposium on Physical Design. San Francisco, USA: ACM, 2019. 147–151
- 21 Lee C Y. An algorithm for path connections and its applications. *IRE Transactions on Electronic Computers*, 1961, **EC-10**(3): 346–365
- 22 Hadlock F O. A shortest path algorithm for grid graphs. *Networks*, 1977, **7**(4): 323–334
- 23 Soukup J. Fast maze router. In: Proceedings of the 15th Design Automation Conference. Las Vegas, USA: IEEE, 1978. 100–102
- 24 Mikami K, Tabuchi K. A computer program for optimal routing of printed circuit conductors. In: Proceedings of the International Federation for Information Processing (IFIP). Edinburgh, UK: North-Holland, 1968. 1475–1478
- 25 Hightower D W. A solution to line-routing problems on the continuous plane. In: Proceedings of the 6th Annual Design Automation Conference. San Francisco, USA: ACM, 1969. 1–24
- 26 Hetzel A. A sequential detailed router for huge grid graphs. In: Proceedings of the Design, Automation and Test in Europe. Paris, France: IEEE, 1998. 332–338
- 27 Dijkstra E W. A note on two problems in connexion with graphs. *Edsger Wybe Dijkstra: His Life, Work, and Legacy*. New York: ACM, 2022. 287–290
- 28 Hart P E, Nilsson N J, Raphael B. A formal basis for the heuristic determination of minimum cost paths. *IEEE Transactions on Systems Science and Cybernetics*, 1968, **4**(2): 100–107
- 29 Clow G W. A global routing algorithm for general cells. In: Proceedings of the 21st Design Automation Conference. Albuquerque, USA: IEEE, 1984. 45–51
- 30 McMurchie L, Ebeling C. PathFinder: A negotiation-based performance-driven router for FPGAs. In: Proceedings of the 3rd International Symposium on Field-Programmable Gate Arrays. Napa Valley, USA: IEEE, 1995. 111–117
- 31 Yang Y, Wing O. Suboptimal algorithm for a wire routing problem. *IEEE Transactions on Circuit Theory*, 1972, **19**(5): 508–510
- 32 Lee J, Bose N, Hwang F. Use of Steiner's problem in suboptimal routing in rectilinear metric. *IEEE Transactions on Circuits and Systems*, 1976, **23**(7): 470–476
- 33 Liu J W, Chen G J, Young E F Y. REST: Constructing rectilinear Steiner minimum tree via reinforcement learning. In: Proceedings of the 58th ACM/IEEE Design Automation Conference (DAC). San Francisco, USA: IEEE, 2021. 1135–1140
- 34 Chu C, Wong Y C. FLUTE: Fast lookup table based rectilinear Steiner minimal tree algorithm for VLSI design. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 2008, **27**(1): 70–83
- 35 Ho J M, Vijayan G, Wong C K. New algorithms for the rectilinear Steiner tree problem. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 1990, **9**(2): 185–193
- 36 Chen H M, Wong M D F, Zhou H, Young F Y, Yang H H, Sherwani N. Integrated floorplanning and interconnect planning. *Layout Optimization in VLSI Design*. New York: Springer, 2001. 1–18
- 37 Kastner R, Bozorgzadeh E, Sarrafzadeh M. Pattern routing: Use and theory for increasing predictability and avoiding coupling. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 2002, **21**(7): 777–790
- 38 Deutsch D N. A “Dogleg” channel router. In: Proceedings of the 13th Design Automation Conference. San Francisco, USA: ACM, 1976. 425–433
- 39 Burstein M, Pelavin R. Hierarchical wire routing. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 1983, **2**(4): 223–234
- 40 Li J T, Marek-Sadowska M. Global routing for gate array. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 1984, **3**(4): 298–307
- 41 Lin S P, Chang Y W. A novel framework for multilevel routing considering routability and performance. In: Proceedings of the IEEE/ACM International Conference on Computer-Aided Design. San Jose, USA: ACM, 2002. 44–50
- 42 Chang Y W, Lin S P. MR: A new framework for multilevel full-chip routing. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 2004, **23**(5): 793–800
- 43 Vannelli A. An adaptation of the interior point method for solving the global routing problem. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 1991, **10**(2): 193–203
- 44 Cho M, Pan D Z. BoxRouter: A new global router based on box expansion and progressive ILP. In: Proceedings of the 43rd Annual Design Automation Conference. San Francisco, USA: IEEE, 2006. 373–378
- 45 Wu T H, Davoodi A, Linderoth J T. GRIP: Global routing via integer programming. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 2011, **30**(1): 72–84
- 46 Vecchi M P, Kirkpatrick S. Global wiring by simulated annealing. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 1983, **2**(4): 215–222
- 47 Shin H, Sangiovanni-Vincentelli A. Mighty: A ‘rip-up and reroute’ detailed router. In: Proceedings of the International Conference on Computer Aided Design. Santa Clara, USA: IEEE, 1986. 2–5
- 48 Cho M, Lu K, Yuan K, Pan D Z. BoxRouter 2.0: A hybrid and robust global router with layer assignment for routability. *ACM Transactions on Design Automation of Electronic Systems*, 2009, **14**(2): Article No. 32
- 49 Moffitt M D. MaizeRouter: Engineering an effective global router. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 2008, **27**(11): 2017–2026

- 50 Chen H Y, Hsu C H, Chang Y W. High-performance global routing with fast overflow reduction. In: Proceedings of the Asia and South Pacific Design Automation Conference. Yokohama, Japan: IEEE, 2009. 582–587
- 51 Xu Y, Zhang Y H, Chu C. FastRoute 4.0: Global router with efficient via minimization. In: Proceedings of the Asia and South Pacific Design Automation Conference. Yokohama, Japan: IEEE, 2009. 576–581
- 52 Chang Y J, Lee Y T, Gao J R, Wu P C, Wang T C. NTHU-route 2.0: A robust global router for modern designs. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 2010, **29**(12): 1931–1944
- 53 Liu W H, Kao W C, Li Y L, Chao K Y. NCTU-GR 2.0: Multi-threaded collision-aware global routing with bounded-length maze routing. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 2013, **32**(5): 709–722
- 54 He J Y, Burtscher M, Manohar R, Pingali K. SPRoute: A scalable parallel negotiation-based global router. In: Proceedings of the IEEE/ACM International Conference on Computer-Aided Design (ICCAD). Westminster, USA: IEEE, 2019. 1–8
- 55 Jiang Y J, Fang S Y. COALA: Concurrently assigning wire segments to layers for 2-D global routing. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 2023, **42**(2): 569–582
- 56 Liu Geng-Geng, Li Ze-Peng, Guo Wen-Zhong, Chen Guo-Long, Xu Ning. Via-aware parallel layer assignment algorithm for VLSI physical design. *Acta Electronica Sinica*, 2022, **50**(11): 2575–2583  
(刘耿耿, 李泽鹏, 郭文忠, 陈国龙, 徐宁. 面向超大规模集成电路物理设计的通孔感知的并行层分配算法. *电子学报*, 2022, **50**(11): 2575–2583)
- 57 Roy J A, Markov I L. High-performance routing at the nanometer scale. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 2008, **27**(6): 1066–1077
- 58 Ozdal M M, Wong M D F. Archer: A history-based global routing algorithm. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 2009, **28**(4): 528–540
- 59 Xu Y, Chu C. MGR: Multi-level global router. In: Proceedings of the IEEE/ACM International Conference on Computer-Aided Design (ICCAD). San Jose, USA: IEEE, 2011. 250–255
- 60 Liu J W, Pui C W, Wang F Z, Young E F Y. CUGR: Detailed-routability-driven 3D global routing with probabilistic resource model. In: Proceedings of the 57th ACM/IEEE Design Automation Conference (DAC). San Francisco, USA: IEEE, 2020. 1–6
- 61 Dolgov S, Volkov A, Wang L T, Xu B Q. 2019 CAD contest: LEF/DEF based global routing. In: Proceedings of the IEEE/ACM International Conference on Computer-Aided Design (ICCAD). Westminster, USA: IEEE, 2019. 1–4
- 62 Liu S T, Pu Y, Liao P Y, Wu H Z, Zhang R, Chen Z T, et al. FastGR: Global routing on CPU-GPU with heterogeneous task graph scheduler. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 2023, **42**(7): 2317–2330
- 63 Fontana T A, Aghaeekiasaraee E, Netto R, Almeida S F, Gandh U, Tabrizi A F, et al. ILP-based global routing optimization with cell movements. In: Proceedings of the IEEE Computer Society Annual Symposium on VLSI (ISVLSI). Tampa, USA: IEEE, 2021. 25–30
- 64 Fontana T A, Aghaeekiasaraee E, Netto R, Almeida S F, Gandh U, Behjat L, et al. ILPGRC: ILP-based global routing optimization with cell movements. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 2024, **43**(1): 352–365
- 65 Zou P, Cai Z J, Lin Z F, Ma C Y, Yu J, Chen J L. Incremental 3-D global routing considering cell movement and complex routing constraints. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 2023, **42**(6): 2016–2029
- 66 Zhu Z R, Shen F H, Mei Y J, Huang Z P, Chen J L, Yang J. A robust global routing engine with high-accuracy cell movement under advanced constraints. In: Proceedings of the 41st IEEE/ACM International Conference on Computer-Aided Design. San Diego, USA: ACM, 2022. Article No. 129
- 67 Hu K S, Yang M J, Yu T C, Chen G C. ICCAD-2020 CAD contest in routing with cell movement. In: Proceedings of the 39th International Conference on Computer-Aided Design. San Diego, USA: ACM, 2020. Article No. 69
- 68 Hu K S, Yu T C, Yang M J, Shen C F C. 2021 ICCAD CAD contest problem B: Routing with cell movement advanced: Invited paper. In: Proceedings of the IEEE/ACM International Conference on Computer Aided Design (ICCAD). Munich, Germany: IEEE, 2021. 1–5
- 69 Nieberg T. Gridless pin access in detailed routing. In: Proceedings of the 48th ACM/EDAC/IEEE Design Automation Conference (DAC). San Diego, USA: IEEE, 2011. 170–175
- 70 Xu X Q, Yu B, Gao J R, Hsu C L, Pan D Z. PARR: Pin-access planning and regular routing for self-aligned double patterning. *ACM Transactions on Design Automation of Electronic Systems*, 2016, **21**(3): Article No. 42
- 71 Xu X Q, Lin Y B, Livramento V, Pan D Z. Concurrent pin access optimization for unidirectional routing. In: Proceedings of the 54th Annual Design Automation Conference. Austin, USA: ACM, 2017. Article No. 20
- 72 Kahng A B, Wang L T, Xu B Q. The Tao of PAO: Anatomy of a pin access oracle for detailed routing. In: Proceedings of the 57th ACM/IEEE Design Automation Conference (DAC). San Francisco, USA: IEEE, 2020. 1–6
- 73 Kahng A B, Kuang J, Liu W H, Xu B Q. In-route pin access-driven placement refinement for improved detailed routing convergence. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 2022, **41**(3): 784–788
- 74 Batterywala S, Shenoy N, Nicholls W, Zhou H. Track assignment: A desirable intermediate step between global routing and detailed routing. In: Proceedings of the IEEE/ACM International Conference on Computer-Aided Design. San Jose, USA: ACM, 2002. 59–66
- 75 Zhang Y H, Chu C. RegularRoute: An efficient detailed router applying regular routing patterns. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 2013, **21**(9): 1655–1668
- 76 Alpert C J. The ISPD98 circuit benchmark suite. In: Proceedings of the International Symposium on Physical Design. Monterey, USA: ACM, 1998. 80–85
- 77 Nam G J, Alpert C J, Villarrubia P, Winter B, Yildiz M. The ISPD2005 placement contest and benchmark suite. In: Proceedings of the International Symposium on Physical Design. San Francisco, USA: ACM, 2005. 216–220
- 78 Nam G J. ISPD 2006 placement contest: Benchmark suite and results. In: Proceedings of the International Symposium on Physical Design. San Jose, USA: ACM, 2006. Article No. 167

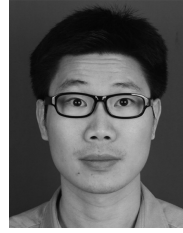
- 79 Wong M P, Liu W H, Wang T C. Negotiation-based track assignment considering local nets. In: Proceedings of the 21st Asia and South Pacific Design Automation Conference (ASP-DAC). Macao, China: IEEE, 2016. 378–383
- 80 Viswanathan N, Alpert C, Sze C, Li Z, Wei Y G. The DAC 2012 routability-driven placement contest and benchmark suite. In: Proceedings of the Design Automation Conference (DAC). San Francisco, USA: IEEE, 2012. 774–782
- 81 Liu G G, Zhuang Z, Guo W Z, Wang T C. RDTA: An efficient routability-driven track assignment algorithm. In: Proceedings of the Great Lakes Symposium on VLSI. Tysons Corner, USA: ACM, 2019. 315–318
- 82 Guo Wen-Zhong, Chen Xiao-Hua, Liu Geng-Geng, Chen Guo-Long. Track assignment algorithm based on hybrid discrete particle swarm optimization. *Pattern Recognition and Artificial Intelligence*, 2019, **32**(8): 758–770 (郭文忠, 陈晓华, 刘耿耿, 陈国龙. 基于混合离散粒子群优化的轨道分配算法. 模式识别与人工智能, 2019, **32**(8): 758–770)
- 83 Jia X T, Cai Y C, Zhou Q, Chen G, Li Z Y, Li Z W. MC-FRoute: A detailed router based on multi-commodity flow method. In: Proceedings of the IEEE/ACM International Conference on Computer-Aided Design (ICCAD). San Jose, USA: IEEE, 2014. 397–404
- 84 Jia X T, Cai Y C, Zhou Q, Yu B. A multicommodity flow-based detailed router with efficient acceleration techniques. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 2018, **37**(1): 217–230
- 85 Yutsis V, Bustany I S, Chinnery D, Shinnerl J R, Liu W H. ISPD 2014 benchmarks with sub-45nm technology rules for detailed-routing-driven placement. In: Proceedings of the International Symposium on Physical Design. Petaluma, USA: ACM, 2014. 161–168
- 86 Kahng A B, Wang L T, Xu B Q. TritonRoute: An initial detailed router for advanced VLSI technologies. In: Proceedings of the IEEE/ACM International Conference on Computer-Aided Design (ICCAD). San Diego, USA: IEEE, 2018. 1–8
- 87 Sun F K, Chen H, Chen C Y, Hsu C H, Chang Y W. A multi-threaded initial detailed routing algorithm considering global routing guides. In: Proceedings of the IEEE/ACM International Conference on Computer-Aided Design (ICCAD). San Diego, USA: IEEE, 2018. 1–7
- 88 Kahng A B, Wang L T, Xu B Q. TritonRoute: The open-source detailed router. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 2021, **40**(3): 547–559
- 89 Kahng A B, Wang L T, Xu B Q. TritonRoute-WXL: The open-source router with integrated DRC engine. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 2022, **41**(4): 1076–1089
- 90 Chen J S, Liu J W, Chen G J, Zheng D, Young E F Y. MARCH: MAze routing under a concurrent and hierarchical scheme for buses. In: Proceedings of the 56th Annual Design Automation Conference. Las Vegas, USA: ACM, 2019. Article No. 216
- 91 Chen G J, Pui C W, Li H C, Young E F Y. Dr. CU: Detailed routing by sparse grid graph and minimum-area-captured path search. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 2020, **39**(9): 1902–1915
- 92 Li H C, Chen G J, Jiang B T, Chen J S, Young E F Y. Dr. CU 2.0: A scalable detailed routing framework with correct-by-construction design rule satisfaction. In: Proceedings of the IEEE/ACM International Conference on Computer-Aided Design (ICCAD). Westminster, USA: IEEE, 2019. 1–7
- 93 Zhuang Z, Liu G G, Ho T Y, Yu B, Guo W Z. TRADER: A practical track-assignment-based detailed router. In: Proceedings of the Design, Automation and Test in Europe Conference and Exhibition (DATE). Antwerp, Belgium: IEEE, 2022. 766–771
- 94 Cong J, Fang J, Xie M, Zhang Y. MARS-a multilevel full-chip gridless routing system. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 2005, **24**(3): 382–394
- 95 Li Y L, Chen H Y, Lin C T. NEMO: A new implicit-connection-graph-based gridless router with multilayer planes and pseudo tile propagation. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 2007, **26**(4): 705–718
- 96 Du Y L, Ma Q, Song H, Shiely J, Luk-Pat G, Miloslavsky A, et al. Spacer-is-dielectric-compliant detailed routing for self-aligned double patterning lithography. In: Proceedings of the 50th Annual Design Automation Conference. Austin, USA: ACM, 2013. Article No. 93
- 97 Ding Y X, Chu C, Mak W K. Detailed routing for spacer-is-metal type self-aligned double/quadruple patterning lithography. In: Proceedings of the 52nd Annual Design Automation Conference. San Francisco, USA: ACM, 2015. Article No. 69
- 98 Ou J J, Yu B, Xu X Q, Mitra J, Lin Y B, Pan D Z. DSAR: DSA aware routing with simultaneous DSA guiding pattern and double patterning assignment. In: Proceedings of the ACM on International Symposium on Physical Design. Portland, USA: ACM, 2017. 91–98
- 99 Yu H J, Chang Y W. DSA-friendly detailed routing considering double patterning and DSA template assignments. In: Proceedings of the 55th Annual Design Automation Conference. San Francisco, USA: ACM, 2018. Article No. 49
- 100 Zhang Y H, Chu C. GDRouter: Interleaved global routing and detailed routing for ultimate routability. In: Proceedings of the 49th Annual Design Automation Conference. San Francisco, USA: ACM, 2012. 597–602
- 101 Gester M, Müller D, Nieberg T, Panten C, Schulte C, Vygen J. BonnRoute: Algorithms and data structures for fast and good VLSI routing. *ACM Transactions on Design Automation of Electronic Systems*, 2013, **18**(2): Article No. 32
- 102 Ajayi T, Blaauw D, Chan T B, Cheng C K, Chhabria V A, Choo D K, et al. OpenROAD: Toward a self-driving, open-source digital layout implementation tool chain. In: Proceedings of the Government Microcircuit Applications and Critical Technology Conference. Albuquerque, USA: Defense Technical Information Center, 2019. 1105–1110
- 103 Aghaeekiasaraee E, Tabrizi A F, Fontana T A, Netto R, Almeida S F, Gandhi U, et al. CRP2.0: A fast and robust cooperation between routing and placement in advanced technology nodes. *ACM Transactions on Design Automation of Electronic Systems*, 2023, **28**(5): Article No. 79
- 104 Betz V, Rose J. VPR: A new packing, placement and routing tool for FPGA research. In: Proceedings of the 7th International Conference on Field-Programmable Logic and Applications (FPL). London, UK: Springer, 1997. 213–222
- 105 Murray K E, Petelin O, Zhong S, Wang J M, Eldafrawy M, Legault J P, et al. VTR 8: High-performance CAD and customizable FPGA architecture modelling. *ACM Transactions on Re-*



- configurable Technology and Systems*, 2020, **13**(2): Article No. 9
- 106 Shah D, Hung E, Wolf C, Bazanski S, Gisselquist D, Milanovic M. Yosys+nextpnr: An open source framework from Verilog to bitstream for commercial FPGAs. In: Proceedings of the 27th Annual International Symposium on Field-Programmable Custom Computing Machines (FCCM). San Diego, USA: IEEE, 2019. 1–4
- 107 Gort M, Anderson J H. Accelerating FPGA routing through parallelization and engineering enhancements special section on PAR-CAD 2010. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 2012, **31**(1): 61–74
- 108 Wang D K, Duan Z H, Tian C, Huang B H, Zhang N. A runtime optimization approach for FPGA routing. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 2018, **37**(8): 1706–1710
- 109 Veracruz D, Vansteenkiste E, Stroobandt D. CRoute: A fast high-quality timing-driven connection-based FPGA router. In: Proceedings of the 27th Annual International Symposium on Field-Programmable Custom Computing Machines (FCCM). San Diego, USA: IEEE, 2019. 53–60
- 110 Murray K E, Zhong S, Betz V. AIR: A fast but lazy timing-driven FPGA router. In: Proceedings of the 25th Asia and South Pacific Design Automation Conference (ASP-DAC). Beijing, China: IEEE, 2020. 338–344
- 111 Wang D K, Feng J, Zhou W, Hao X X, Zhang X D. FCRoute: A fast FPGA connection router using soft routing-space pruning algorithm. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 2023, **42**(3): 887–899
- 112 Wang J R, Mai J, Di Z X, Lin Y B. A robust FPGA router with concurrent intra-CLB rerouting. In: Proceedings of the 28th Asia and South Pacific Design Automation Conference (ASP-DAC). Tokyo, Japan: IEEE, 2023. 529–534
- 113 Yang S, Gayasen A, Mulpuri C, Reddy S, Aggarwal R. Routability-driven FPGA placement contest. In: Proceedings of the International Symposium on Physical Design. Santa Rosa, USA: ACM, 2016. 139–143
- 114 Shen M H, Luo G J, Xiao N. Coarse-grained parallel routing with recursive partitioning for FPGAs. *IEEE Transactions on Parallel and Distributed Systems*, 2021, **32**(4): 884–899
- 115 Wang D K, Duan Z H, Tian C, Huang B H, Zhang N. ParRA: A shared memory parallel FPGA router using hybrid partitioning approach. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 2020, **39**(4): 830–842
- 116 Shen M H, Luo G J, Xiao N. Combining static and dynamic load balance in parallel routing for FPGAs. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 2021, **40**(9): 1850–1863
- 117 Shen M H, Xiao N. Load balance-centric distributed parallel routing for large-scale FPGAs. In: Proceedings of the 31st International Conference on Field-Programmable Logic and Applications (FPL). Dresden, Germany: IEEE, 2021. 242–248
- 118 Shen M H, Zhang W T, Luo G J, Xiao N. Serial-equivalent static and dynamic parallel routing for FPGAs. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 2020, **39**(2): 411–423
- 119 Shen M H, Xiao N. Towards serial-equivalent multi-core parallel routing for FPGAs. In: Proceedings of the Design, Automation and Test in Europe Conference and Exhibition (DATE). Grenoble, France: IEEE, 2020. 1139–1144
- 120 Gort M, Anderson J H. Deterministic multi-core parallel routing for FPGAs. In: Proceedings of the International Conference on Field-Programmable Technology. Beijing, China: IEEE, 2010. 78–86
- 121 Shen M H, Xiao N. Fine-grained parallel routing for FPGAs with selective expansion. In: Proceedings of the 36th International Conference on Computer Design (ICCD). Orlando, USA: IEEE, 2018. 577–586
- 122 Zha Y, Li J. Revisiting pathfinder routing algorithm. In: Proceedings of the ACM/SIGDA International Symposium on Field-Programmable Gate Arrays. New York, USA: ACM, 2022. 24–34
- 123 He X, Huang T, Chow W K, Kuang J, Lam K C, Cai W Z, et al. Ripple 2.0: High quality routability-driven placement via global router integration. In: Proceedings of the 50th Annual Design Automation Conference. Austin, USA: ACM, 2013. Article No. 152
- 124 Chan W T J, Ho P H, Kahng A B, Saxena P. Routability optimization for industrial designs at sub-14nm process nodes using machine learning. In: Proceedings of the ACM on International Symposium on Physical Design. Portland, USA: ACM, 2017. 15–21
- 125 Tabrizi A F, Darav N K, Rakai L, Bustany I, Kennings A, Behjat L. Eh?Predictor: A deep learning framework to identify detailed routing short violations from a placed netlist. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 2020, **39**(6): 1177–1190
- 126 Tabrizi A F, Darav N K, Xu S C, Rakai L, Bustany I, Kennings A, et al. A machine learning framework to identify detailed routing short violations from a placed netlist. In: Proceedings of the 55th Annual Design Automation Conference. San Francisco, USA: ACM, 2018. Article No. 48
- 127 Bustany I S, Chinnery D, Shimmerl J R, Yutsis V. ISPD 2015 benchmarks with fence regions and routing blockages for detailed-routing-driven placement. In: Proceedings of the Symposium on International Symposium on Physical Design. Monterey, USA: ACM, 2015. 157–164
- 128 Yu T C, Fang S Y, Chiu H S, Hu K S, Tai P H Y, Shen C C F, et al. Pin accessibility prediction and optimization with deep learning-based pin pattern recognition. In: Proceedings of the 56th Annual Design Automation Conference. Las Vegas, USA: ACM, 2019. Article No. 220
- 129 Liang R J, Xiang H, Pandey D, Reddy L, Ramji S, Nam G J, et al. DRC hotspot prediction at sub-10nm process nodes using customized convolutional network. In: Proceedings of the International Symposium on Physical Design. Taipei, China: ACM, 2020. 135–142
- 130 Xie Z Y, Huang Y H, Fang G Q, Ren H X, Fang S Y, Chen Y R, et al. RouteNet: Routability prediction for mixed-size designs using convolutional neural network. In: Proceedings of the IEEE/ACM International Conference on Computer-Aided Design (ICCAD). San Diego, USA: IEEE, 2018. 1–8
- 131 Huang Y H, Xie Z Y, Fang G Q, Yu T C, Ren H X, Fang S Y, et al. Routability-driven macro placement with embedded CNN-based prediction model. In: Proceedings of the Design, Automation and Test in Europe Conference and Exhibition (DATE). Florence, Italy: IEEE, 2019. 180–185
- 132 Chen J S, Kuang J, Zhao G W, Huang D J H, Young E F Y. PROS 2.0: A plug-in for routability optimization and routed wirelength estimation using deep learning. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 2023, **42**(1): 164–177

- 133 Alhyari A, Szentimrey H, Shamli A, Martin T, Gréwal G, Areibi S. A deep learning framework to predict routability for FPGA circuit placement. *ACM Transactions on Reconfigurable Technology and Systems*, 2021, **14**(3): Article No. 16
- 134 Alhyari A, Shamli A, Abuwaimer Z, Areibi S, Grewal G. A deep learning framework to predict routability for FPGA circuit placement. In: Proceedings of the 29th International Conference on Field Programmable Logic and Applications (FPL). Barcelona, Spain: IEEE, 2019. 334–341
- 135 Alawieh M B, Li W X, Lin Y B, Singhal L, Iyer M A, Pan D Z. High-definition routing congestion prediction for large-scale FPGAs. In: Proceedings of the 25th Asia and South Pacific Design Automation Conference (ASP-DAC). Beijing, China: IEEE, 2020. 26–31
- 136 Yu C X, Zhang Z R. Painting on placement: Forecasting routing congestion using conditional generative adversarial nets. In: Proceedings of the 56th Annual Design Automation Conference. Las Vegas, USA: ACM, 2019. Article No. 219
- 137 Wang B W, Shen G B, Li D, Hao J Y, Liu W L, Huang Y, et al. LHNN: Lattice hypergraph neural network for VLSI congestion prediction. In: Proceedings of the 59th ACM/IEEE Design Automation Conference. San Francisco, USA: ACM, 2022. 1297–1302
- 138 Viswanathan N, Alpert C J, Sze C, Li Z, Nam G J, Roy J A. The ISPD-2011 routability-driven placement contest and benchmark suite. In: Proceedings of the International Symposium on Physical Design. Santa Barbara, USA: ACM, 2011. 141–146
- 139 Mirhoseini A, Goldie A, Yazgan M, Jiang J W, Songhori E, Wang S, et al. A graph placement methodology for fast chip design. *Nature*, 2021, **594**(7862): 207–212
- 140 Zaruba F, Benini L. The cost of application-class processing: Energy and performance analysis of a Linux-ready 1.7-GHz 64-bit RISC-V core in 22-nm FDSOI technology. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 2019, **27**(11): 2629–2640
- 141 Cheng R Y, Yan J C. On joint learning for solving placement and routing in chip design. In: Proceedings of the 35th International Conference on Neural Information Processing Systems. Virtual Event: Curran Associates Inc., 2021. Article No. 1262
- 142 Cheng R Y, Lyu X L, Li Y, Ye J J, Hao J Y, Yan J C. The policy-gradient placement and generative routing neural networks for chip design. In: Proceedings of the 36th International Conference on Neural Information Processing Systems. New Orleans, USA: Curran Associates Inc., 2022. Article No. 1911
- 143 Hung W T, Huang J Y, Chou Y C, Tsai C H, Chao M. Transforming global routing report into dre violation map with convolutional neural network. In: Proceedings of the International Symposium on Physical Design. Taipei, China: ACM, 2020. 57–64
- 144 Su M D, Ding H Z, Weng S H, Zou C Z, Zhou Z H, Chen Y L, et al. High-correlation 3D routability estimation for congestion-guided global routing. In: Proceedings of the 27th Asia and South Pacific Design Automation Conference (ASP-DAC). Taipei, China: IEEE, 2022. 580–585
- 145 Utyamishev D, Partin-Vaisband I. Multiterminal pathfinding in practical VLSI systems with deep neural networks. *ACM Transactions on Design Automation of Electronic Systems*, 2023, **28**(4): Article No. 51
- 146 Utyamishev D, Partin-Vaisband I. Late breaking results: A neural network that routes ICs. In: Proceedings of the 57th ACM/IEEE Design Automation Conference (DAC). San Francisco, USA: IEEE, 2020. 1–2
- 147 Gandhi U, Aghaeekiasarace E, Bustany I S K, Mousavi P, Taylor M E, Behjat L. RL-ripper: A framework for global routing using reinforcement learning and smart net ripping techniques. In: Proceedings of the Great Lakes Symposium on VLSI. Knoxville, USA: ACM, 2023. 197–201
- 148 Liao H G, Zhang W T, Dong X L, Poczos B, Shimada K, Kara L B. A deep reinforcement learning approach for global routing. *Journal of Mechanical Design*, 2020, **142**(6): Article No. 061701
- 149 Ren H X, Fojtik M. Standard cell routing with reinforcement learning and genetic algorithm in advanced technology nodes. In: Proceedings of the 26th Asia and South Pacific Design Automation Conference (ASP-DAC). Tokyo, Japan: IEEE, 2021. 684–689
- 150 Siddiqi U, Martin T, Van Den Eijnden S, Shamli A, Grewal G, Sait S, et al. Faster FPGA routing by forecasting and pre-loading congestion information. In: Proceedings of the ACM/IEEE Workshop on Machine Learning for CAD. Utah, USA: ACM, 2022. 15–20
- 151 Luu J, Goeders J, Wainberg M, Somerville A, Yu T, Nasartschuk K, et al. VTR 7.0: Next generation architecture and CAD system for FPGAs. *ACM Transactions on Reconfigurable Technology and Systems*, 2014, **7**(2): Article No. 6
- 152 Lin Y B, Qu T, Lu Z Q, Su Y J, Wei Y Y. Asynchronous reinforcement learning framework and knowledge transfer for net-order exploration in detailed routing. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 2022, **41**(9): 3132–3142
- 153 Qu T, Lin Y B, Lu Z Q, Su Y J, Wei Y Y. Asynchronous reinforcement learning framework for net order exploration in detailed routing. In: Proceedings of the Design, Automation and Test in Europe Conference and Exhibition (DATE). Grenoble, France: IEEE, 2021. 1815–1820
- 154 Liao H G, Dong Q Y, Qi W Y, Fallon E, Kara L B. Track-assignment detailed routing using attention-based policy model with supervision. In: Proceedings of the 2nd ACM/IEEE Workshop on Machine Learning for CAD (MLCAD). Reykjavik, Iceland: IEEE, 2020. 105–110
- 155 Chen H, Hsu K C, Turner W J, Wei P H, Zhu K R, Pan D Z, et al. Reinforcement learning guided detailed routing for custom circuits. In: Proceedings of the International Symposium on Physical Design. Virtual Event: ACM, 2023. 26–34
- 156 Ju X H, Zhu K L, Lin Y B, Zhang L. Asynchronous multi-nets detailed routing in VLSI using multi-agent reinforcement learning. In: Proceedings of the 7th IEEE International Conference on Network Intelligence and Digital Content (IC-NIDC). Beijing, China: IEEE, 2021. 250–254
- 157 Murray K E, Whitty S, Liu S Y, Luu J, Betz V. Timing-driven titan: Enabling large benchmarks and exploring the gap between academic and commercial CAD. *ACM Transactions on Reconfigurable Technology and Systems*, 2015, **8**(2): Article No. 10
- 158 Zhuo H H, Kambhampati S. Model-lite planning: Case-based vs. model-based approaches. *Artificial Intelligence*, 2017, **246**: 1–21
- 159 Zhuo H H, Muñoz-Avila H, Yang Q. Learning hierarchical task network domains from partially observed plan traces. *Artificial Intelligence*, 2014, **212**: 134–157
- 160 Zhuo H H, Yang Q, Hu D H, Li L. Learning complex action

- models with quantifiers and logical implications. *Artificial Intelligence*, 2010, **174**(18): 1540–1569
- 161 Zhuo H H. Crowdsourced action-model acquisition for planning. In: Proceedings of the 29th AAAI Conference on Artificial Intelligence. Austin, USA: AAAI, 2015. 3439–3446
- 162 Zhuo H H, Yang Q. Action-model acquisition for planning via transfer learning. *Artificial Intelligence*, 2014, **212**: 80–103
- 163 Zhuo H H, Yang Q, Pan R, Li L. Cross-domain action-model acquisition for planning via web search. In: Proceedings of the 21st International Conference on International Conference on Automated Planning and Scheduling. Freiburg, Germany: AAAI, 2011. 298–305
- 164 Jin K B, Zhuo H H, Xiao Z H, Wan H, Kambhampati S. Gradient-based mixed planning with symbolic and numeric action parameters. *Artificial Intelligence*, 2022, **313**: Article No. 103789
- 165 Zhuo H H, Zha Y T, Kambhampati S, Tian X. Discovering underlying plans based on shallow models. *ACM Transactions on Intelligent Systems and Technology*, 2020, **11**(2): Article No. 18
- 166 Zhuo H H. Recognizing multi-agent plans when action models and team plans are both incomplete. *ACM Transactions on Intelligent Systems and Technology*, 2019, **10**(3): Article No. 30
- 167 Zhuo H H. Human-aware plan recognition. In: Proceedings of the 31st AAAI Conference on Artificial Intelligence. San Francisco, USA: AAAI, 2017. 3686–3693



**周展文** 中山大学计算机学院博士研究生. 主要研究方向为芯片布线, 智能规划, 机器学习和强化学习.

E-mail: [zhouzhw26@mail2.sysu.edu.cn](mailto:zhouzhw26@mail2.sysu.edu.cn)

(**ZHOU Zhan-Wen** Ph.D. candidate at the School of Computer Science and Engineering, Sun Yat-sen

University. His research interest covers chip routing, automated planning, machine learning, and reinforcement learning.)



**卓汉逵** 中山大学计算机学院副教授. 主要研究方向为智能规划, 机器学习, 人工智能. 本文通信作者.

E-mail: [zhuohank@mail.sysu.edu.cn](mailto:zhuohank@mail.sysu.edu.cn)

(**ZHUO Han-Kui** Associate professor at the School of Computer Science and Engineering, Sun Yat-sen

University. His research interest covers automated planning, machine learning, and artificial intelligence. Corresponding author of this paper.)