

基于强化图注意力网络的数字芯片布局方法

侯泓秋, 仝明磊, 李易婉

(上海电力大学 电子与信息工程学院, 上海 200090)

摘要: 在数字芯片设计后端流程中, 宏和标准单元的布局是一项耗时的工作, 通过机器学习快速有效地提供解决方案能够加快芯片开发的周期, 降低人工布局带来的风险; 然而布局问题是一个多目标优化问题, 目前大多数方法都注重在满足各项指标下最大化减小线长, 已换取时钟延迟的降低, 忽略了其他指标仍然存在下降的空间, 例如良好的拥塞指标有利于降低芯片散热和功耗; 针对上述问题, 设计一种新的带有密集型奖励函数的深度强化学习框架, 将拥塞信息映射到图像中, 给出新的特征嵌入模型对版图的全局信息进行多尺度提取, 并引入图注意力网络捕获网表的连接关系, 采用 Advantage Actor Critic (A2C) 算法更新策略函数, 实现了数字版图的自动布局, 并在公共的数字芯片网表基准上验证了该方法的有效性。

关键词: 图卷积神经网络; GAT; 数字集成电路; 深度强化学习; EDA

Digital Chip Layout Method Based on Reinforcement Graph Attention Network

HOU Hongqiu, TONG Minglei, LI Yiwan

(College of Electronics and Information Engineering, Shanghai University of Electric Power, Shanghai 200090, China)

Abstract: In the back-end process of digital chip design, it is a time-consuming task for the placement of macros and standard cells, a machine learning can provide a fast and effective solution, accelerating the cycle of chip development and reducing risks caused by manual layout. However, the layout is a multi-objective optimization problem. Currently, most methods focus on maximizing the reduction of line length when meeting various indicators, with an exchange for a decrease in clock delay, while ignoring the potential for further decline in other indicators, such as good congestion indicators that are beneficial for reducing chip heat dissipation and power consumption; To solve the above problems, this paper proposes a new deep reinforcement learning framework with intensive reward function, maps the congestion information to the images, provides a new feature embedding model to extract the global information of the layout at multiple scales, introduces the connection relationship of the graph attention network to capture the netlist, updates the policy function by using the Advantage Actor Critic (A2C) algorithm, realizes the automatic layout of the digital landscape, and verifies the effectiveness of the proposed method on the public digital chip netlist benchmark.

Keywords: Graph CNN; GAT; digital integrated circuits; reinforcement deep learning; EDA

0 引言

在芯片后端设计流程中, 布局和布线阶段可以描述为在一个 2D 的平面上按照复杂的设计规则布局百万个标准单元^[1]和宏^[2], 并根据确定的电路结构和设计约束连接它们, 构成一个高度集成的电路系统。以往的布局布线方法需要工程师根据自身经验将宏和标准单元精确地布局在版图上, 并且要考虑面积、线长、拥塞等性能指标, 这是一项复杂且耗时的任务, 往往需要几个月的时间来完成, 极大地制约了芯片后端的开发周期。

为了加快设计周期, 提高设计效率。近年来, 在解决数字芯片版图布局问题的方法上, 可以分为基于组合优化的方法和基于学习的方法。

在基于组合优化的方法中, 研究者通常将布局问题转

化为组合问题, 通过施加各种约束, 以求解出满足实际要求的布局结果。文献 [3-6] 在布局问题中提出的自适应算法虽然具有一定的灵活性和全局最优解的能力, 但是随着电路规模的不断扩大, 自适应算法的求解耗时长、难度大, 难以满足现代布局设计的需求。文献 [7-8] 引入了基于静电场的全局平滑密度代价函数和 Nesterov 方法来设计非线性优化器, 并通过在多线程 CPU 上使用分区技术来减少运行时间, 已达到能够处理数百万个标准单元的目的。然而, 随着设计复杂性的增加, 仅仅优化传统的布局指标 (如线长指标) 已经难以满足设计需求。文献 [9-10] 提出了一种由可布线性驱动的布局方法, 并且引入了拥塞指标来衡量布局结果的可布线性, 强调了布局结果可能存在无法布线的问题。文献 [11] 提出了时序驱动的布局方法和文献

收稿日期: 2023-10-08; 修回日期: 2023-11-20。

基金项目: 国家自然科学基金项目(62105296)。

作者简介: 侯泓秋(1998-), 男, 硕士。

通讯作者: 仝明磊(1976-), 男, 博士, 副教授。

引用格式: 侯泓秋, 仝明磊, 李易婉. 基于强化图注意力网络的数字芯片布局方法[J]. 计算机测量与控制, 2024, 32(11): 235-242.

[12] 提出了由功率驱动的布局方法体现出衡量布局质量的指标存在多样化的特点。

在基于学习的方法中,文献 [2] 提出的 DREAMPlac 方法,通过将分析布局算法 RePlAcE 与深度学习工具包 Py-Torch 相结合^[13],实现了对数以百万的标准单元的布局。文献 [1] 提出了一种端到端强化学习方法,用于宏布局问题的求解,它将芯片布局建模为一个序列决策问题,并利用图卷积神经网络 (GCN)^[14]提取网表的连接关系。文献 [15] 提出了一种基于深度强化学习 (DRL) 的联合学习方法,以实现宏的布局和布线。最近,基于视觉图像的强化学习方法在布局质量取得进步,文献 [16] 提出的 Mask-Place 方法将网表信息的特征表述为一组掩膜图像,能够最大限度的提取网表中的关键特征,并且解决了宏的重叠问题。

为了全面地获取全局状态信息和量化布局质量的指标,提出一种基于强化图注意力网络的布局方法 (RAG-Place),首先通过卷积神经网络 (CNN)^[17]对网表信息编码,并利用图注意力网络 (GAT)^[18]模型提取网表的连接关系,然后训练一个由 CNN 构造的策略函数,每次训练迭代中,智能体根据宏的大小顺序布局一个宏,并通过重复的事件决策列表中每个宏的全局位置,当完成一系列宏的布局后,利用 Dreamplace 的标准单元布局器实现对标准单元的布局,以最大化累积奖励 (拥塞和密度的加权组合),最后使用 A2C 算法^[19]来优化下一次迭代的网络参数,具体示意图见图 1。

如表 1 所示,DeepPR 和 GraphPlace 都利用超图 $G=(V, E)$ 来表示网表。在超图中,度较大的节点与其他节点相比,与整体的关联性较强,其地位比较重要。一般而言,在实际布局中,宏上引脚数量的越多就决定了该宏在电路系统中的相关性越强 (仅针对连接关系而言,不考虑电路特性),因此引脚数越多的宏,它的位置特征对布局质量具有较大影响,利用 GCN 无法有效地解决此问题,在 RAG-

Place 中引入 GAT 代替 GCN 的原因是 GAT 引入了注意力机制。通过计算节点之间的注意力系数,能够自适应地聚焦于图中最重要的节点 (引脚数较多的宏),从而更有效地捕捉节点之间的关系。

RAGPlace 实现了以下 4 个贡献: (1) 设计了一种密集型奖励函数来进行表征不同宏模块之间相互作用对线长和拥塞的影响。(2) 在引入引脚信息时,设计的多重限制方法减弱不同尺寸宏模块的引脚数差异对布局质量的影响。(3) 设计了一种新的策略网络和多尺度图像表示方法,能够在最小化线长和拥塞的同时解决宏模块之间的重叠问题,并为后续的布线提供可能性。(4) 引入 GAT 模型,通过注意力机制,来提高了布局质量。实验表明,RAGPlace 在 ISPD2005^[20]的 6 个不同电路结构上,布局质量上优于目前先进的方法。

1 RAGPlace 框架

芯片布局问题可以被表述为马尔可夫决策问题^[21],每轮布局结束后的奖励被计算为近似无线长度、拥塞的线性组合的相反数,并作为反馈提供给智能体,以优化下一次迭代的参数。布局结束后的奖励如公式 (1) 所示:

$$R = -\text{Wirelength}(p, g) - \gamma \text{Congtion}(p, g) \quad (1)$$

在强化学习模型中,RAGPlace 为智能体与环境交互提供了可能。它提供了 A2C 算法所需要一系列参数,构成了一个完整的序列决策方案。

A2C (Advantage Actor Critic) 使用优势函数可以作为衡量选取动作值和所有动作平均值好坏的指标。相对于 Actor-Critic 出现的高方差问题,引入优势函数可以使梯度下降平缓,训练过程更加稳定。公式 (2) 和公式 (3) 为值网络函数,用于构建优势函数,如公式 (4) 所示,但在实际的操作中,因为 TD-error 是优势函数的无偏估计,因此一般用 TD-error 代替优势函数进行计算。其中 R_t 表示当前奖励, γ 为折扣因子。

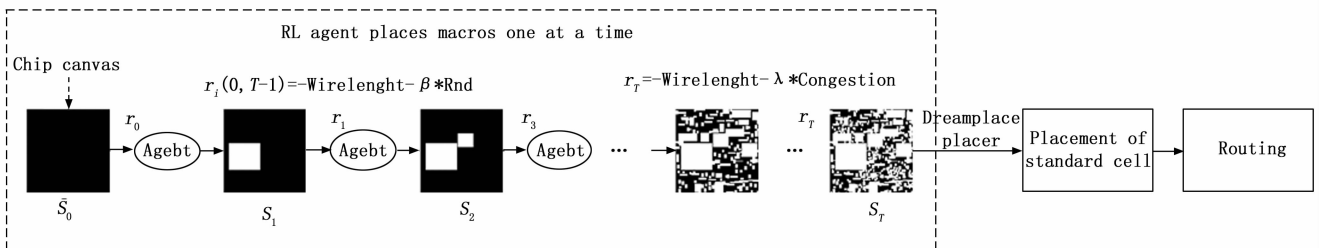


图 1 RAGPlace 方法和训练方案概述。

s_t 表示由芯片版图映射到 128×128 的状态空间, r_t 表示获得的奖励

表 1 宏布局方法的不同方面比较

方法	状态空间	宏重叠	奖励	状态信息	嵌入方式
GraphPlace	128^2	×	预训练奖励函数	位置	CNN+GCN
DeepPR	32^2	×	线长奖励、RND ^[22]	位置、线长	CNN+GCN
MaskPlace	224^2	✓	线长奖励	位置、线长	CNN
RAGPlace	128^2	✓	线长奖励、拥塞奖励、RND	位置、线长、拥塞	CNN+GAT

$$Q_{\pi}(s_t, a_t) = IE_{s_{t+1}, a_{t+1}}[R_t + \gamma Q_{\pi}(s_{t+1}, a_{t+1})] =$$

$$IE_{s_{t+1}}[R_t + \gamma IE_{a_{t+1}}[Q_{\pi}(s_{t+1}, a_{t+1})]] = IE_{s_{t+1}}[R_t + \gamma V_{\pi}(s_{t+1})] \quad (2)$$

$$V_{\pi}(s_t) = IE_{a_t}[Q_{\pi}(s_t, a_t) | s_t] \approx r_t + \gamma V_{\pi}(s_{t+1}) \quad (3)$$

$$A(s_t, a_t) = Q_{\pi}(s_t, a_t) - V_{\pi}(s_t) \quad (4)$$

式中, 利用蒙特卡洛近似, 可以计算出 TD-error 值, 见公式 (5), A2C 的 Actor 网络的更新函数见公式 (6), Critic 网络的更新使用 EMS 损失函数来迭代更新, 见公式 (7):

$$\delta = r_t + \gamma V_{\pi}(s_{t+1}) - V_{\pi}(s_t) \quad (5)$$

$$\theta = \theta + \alpha \Delta_{\theta} \log \pi_{\theta}(s_t, a_t) A(s_t, a_t) = \theta + \alpha \Delta_{\theta} \log \pi_{\theta}(s_t, a_t) \delta \quad (6)$$

$$loss = \frac{1}{n} \sum_{n=1}^{\infty} \delta^2 \quad (7)$$

RAGPlace 的每一次循环都代表着智能体完成一次宏的动作, 并给出了相应的奖励, 如图 1 所示。图 2 阐述了展示了整个 RAGPlace 的网路框架。它由四部分组成, 预处理阶段、特征嵌入、策略函数、价值函数。预处理阶段实现了对网表数据结构的细化, 为后续特征嵌入提供所需要的条件, 同时通过实时地将环境的状态反馈给智能体, 促使能够做出有利的决策。特征嵌入模型由 CNN 和 GAT 构成, 从图像和图的角度实现多维特征提取。策略函数 $\pi_{\theta}(a_t | s_t)$ 由一系列反卷积块和残差块组成, 输入网表的状态特征 s_t , 输出智能体的动作值 a_t (宏的位置信息), 该动作值输入到环境中获得对应的奖励记为 r_t , 策略网络通过预先布局宏模块的位置已获得最大期望的奖励来更新策略。价值函数 $V_{\pi}(s_t)$ 由一组线性层组成, 表征在当前状态下, 采取某个动作可以带来的长期回报 (累计奖励)。

RAGPlace 把实际布局的画布映射成 128×128 的网格像素图, 用二进制矩阵嵌入宏模块的实际位置和可放置位置的信息, 命名为 VM 和 PM, 定义为 f_{VM} 和 f_{PM} , 将表示每一状态下线长的全局信息的热力图命名为 WH, 定义为 f_{WH} , 将表示当前状态下拥塞的全局信息的热力图命名为 CH, 定义为 f_{CH} 。超图作为网表之间的连接关系的表示, 利用 GAT 模型实现连接关系的嵌入。对特征向量 f^{sh} 进行

解码的反卷积块, 在这个过程中, 由于反卷积操作本身会造成信息丢失, 还可能会破坏图像的整体结构。例如, 输出特征图中的一些边缘区域会变得锐利, 或者产生棱角。将 f_{PM} 和 f_{VM} 的特征作为补充信息, 完善丢失的信息。由于芯片布局问题的动作是离散的, 策略网络输出的结果严格地是一个动作的概率矩阵, 将 f_{VM} 作为硬约束, 可以实现零重叠, 此外 f_{WH} 作为权重矩阵, 可以影响到动作概率的分布, 加速目标函数快速收敛。

2 RAGPlace 内容

2.1 度量指标

引脚是信号输入输出接口的物理载体, 位于宏和标准单元上, 它在版图的位置取决于所在模块的实际位置。例如, 图 3 (a) 中, 宏 M_0 上的 3 个引脚 P_{01} 、 P_{02} 、 P_{03} , 若 M_0 的实际位置坐标为 (x, y) , P_{01} 的偏移量为 $(\Delta x, \Delta y)$, 则 P_{01} 在版图上的实际位置为 $(x + \frac{L}{2} + \Delta x, y + \frac{H}{2} + \Delta y)$ 。节点表示宏模块 (DRAM、Caches 等)。网络指包含一系列引脚的连接关系的电路。

2.1.1 Wirelength

在计算线长时, 利用半周长电线长度 (HPWL) 去近似代替线长是一种最常用的方法。在一个电路中, 把该电路包含的所有节点的引脚所围成的最大边界框的半周长近似为该电路的线长^[23]。例如, 给定网络 (电路) net_i 和端点 (引脚) 集合 $\{x_i\}$ 和 $\{y_i\}$, $\{x_i\}$ 表示所有端点的横坐标, $\{y_i\}$ 表示所有端点的纵坐标, 则该网络的线长的计算方法见公式 (8), 网表的总线长的计算公式见公式 (9):

$$HPWL(net_i) = \max\{x_i\} - \min\{x_i\} + \max\{y_i\} - \min\{y_i\} \quad (8)$$

$$HPWL(netlist) = \sum_i^{\log(net)} HPWL(net_i) \quad (9)$$

2.1.2 Congestion

拥塞作为衡量布局质量的指标之一是确保布局结果具有可布线性。布线失败指的是在布局中出现较多的导线穿越某一特定宽度的区域, 导致无法进行后续的布线。为了

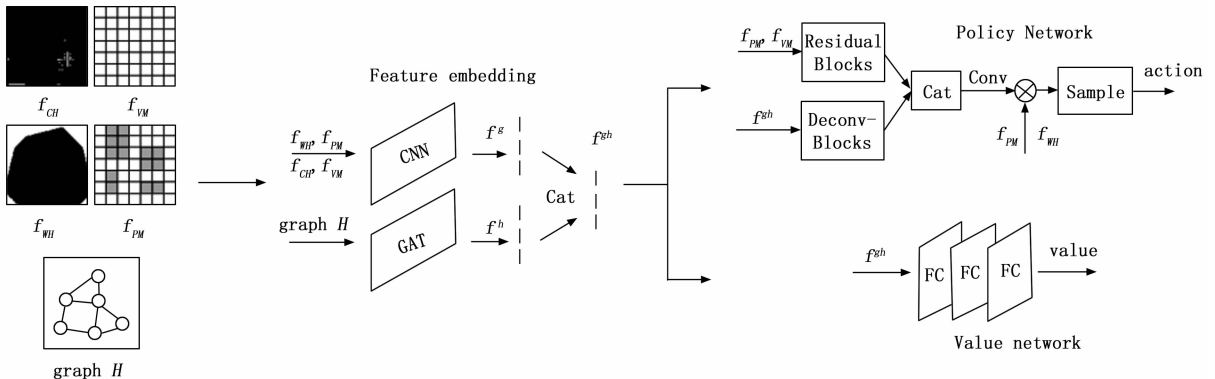


图 2 RAGPlace 的框架图

估计布局结果的拥塞程度，常见的方法之一是通过 RUDY^[24]算法获取的拥塞指标。如图 3 (b) 所示，每个网格单元视为一个布局区域，该区域的拥塞值 s 表示为累积覆盖其自身的所有净边界框（即边界框的高度和宽度之差）的倒数，如公式 (10) 所示。其中， n 为包围了该网格的子网络个数， l 和 h 为子网络的最大物理边框的长宽：

$$s = \sum_i^n \frac{l_i + h_i}{l_i \times h_i} \quad (10)$$

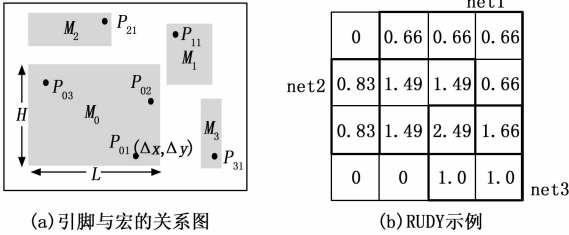


图 3 示意图

2.2 数据预处理

由于网表数据的格式为 BOOKSHELF，很难直观地获取网表的状态和特征信息，一个常见的网表由宏模块，引脚，标准单元组成，如图 4 所示，细化网表结构的目标是提取网表中的各元素的物理关系，以方便后续程序的处理。

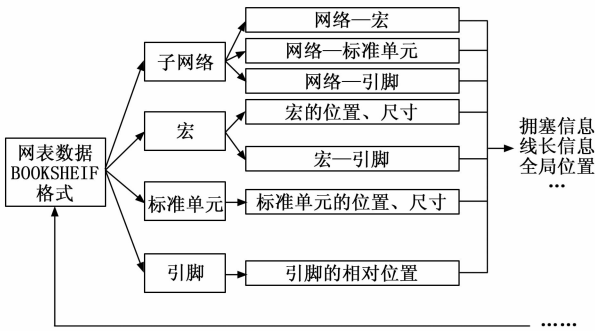


图 4 网表数细化流程图

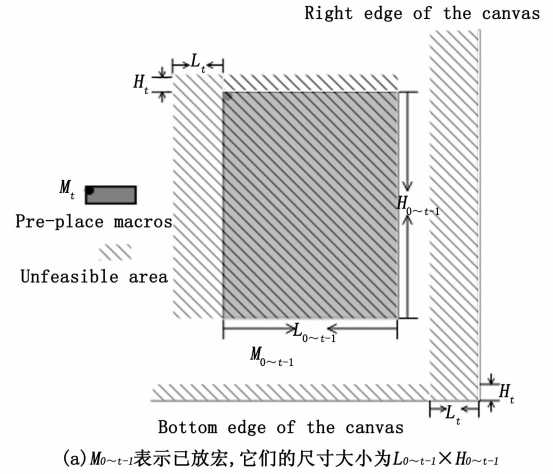
2.3 多维特征融合 (MR)

2.3.1 可布局位置信息图 (PM)

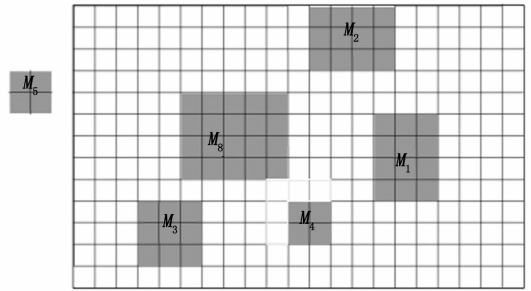
PM 表征的是对于预布局宏在当前版图上能够布局的区域，目的是指导智能学会将宏模块布局在不会造成重叠的区域，同时也作为硬约束修正后续的动作概率矩阵。

常见的方法是累积一个二维数组来检查每个宏的占据位置，当 2D 网格被划分为 $N \times M$ 个网格时，其复杂度为 $O(N^2)M$ ， M 表示网表宏的总个数。而 PM 的复杂度为 $O(M)$ ，初始一个 128×128 的零值矩阵 H ，当布局第 M_i 模块时，需要获得 $M_{i \in (0, i-1)}$ 的布局轨迹信息（尺寸和位置）和 M_i 的尺寸，如图 5 (a) 所示。每布局一个宏 M_i 时，需要扫描一遍已经布局在版图宏 $M_{i \in (0, i-1)}$ 的轨迹信息，根据这些信息，去填充零值矩阵。已经被宏 $M_{i \in (0, i-1)}$ 占据的版图区域所映射的矩阵区域的值设置为 1。由于实现零重叠是

布局的目标之一，因此应该将导致 M_i 与 $M_{i \in (0, i-1)}$ 重叠的区域的值设置为 1。此外，使 M_i 超出边界，违反设计规则的区域的价值应当设置为 1，如图 5 (b) 所示。



(a) $M_{0 \sim t-1}$ 表示已放宏，它们的尺寸大小为 $L_{0 \sim t-1} \times H_{0 \sim t-1}$



(b) 布局 M_5 时，需考虑 $M_0 \sim M_8$ 所有宏的轨迹

图 5 示意图

2.3.2 线长增量信息图 (WH)

WH 表征预布局宏在不同区域对线长的影响。不同区域的数值表示待布局宏在该区域上的总线长增量大小。常见的方法是将待布局宏的引脚位置坐标通过轮询的方式获取在版图区域上每个位置的线长增量，这种方法的复杂度为 $O(N^2MP)$ ， P 为宏上所有引脚的总个数。WH 的设计需要实时地缓存网表中每个子网络的状态信息，它包含由 x_{max} 、 x_{min} 、 y_{max} 和 y_{min} 构成的子集电路最大物理边框和每个子集电路中已经布局上引脚的个数。在布局前，初始化子网络的状态信息，当布局宏 M_i 时，把它包含的引脚看作一个集合 P_i ，其中 $i \in (0, z)$ ， z 为引脚总数。对于 P_i 集合中的每一个引脚，从数据集中找到对应的所属子集电路结构和状态信息，根据这些信息和当前引脚的相对位置构建 WH。

2.3.3 全局位置信息图 (VM)

VM 表征的是宏模块映射在尺寸为 128×128 的网格上的位置和大小信息。VM 作为当前版图的全局观测状态特征，不同于 DeepPR，它将所有的宏都看成具有真是尺寸的矩形，而 DeepPR 是用一个端点来代替宏，端点表示法缺乏对宏的大小信息，布局结果存在重叠问题。在 VM 中，如

果一个宏的真实尺寸为 w 和 h , 画布的实际物理大小为 x 和 y , 则宏映射到画布中的尺寸为 $\frac{Nw}{x}$ 和 $\frac{Nh}{y}$ 。

2.3.4 拥塞信息图 (CH)

CH 表征的是待布局宏在不同位置的拥塞情况。每个区域上的值反应了待布局宏模块在该区域上的拥塞程度。在对当前宏模块布局时, 定义一个 128×128 的矩阵 \mathbf{H} , \mathbf{H} 中每个元素的值反应了该宏模块落在此位置所带来的拥塞情况, 当 M_i 被布局在指定区域时, 对集合 P_i^j 中的每个引脚, 通过 RUDY 算法获取的拥塞值, 拥塞热力图的构建方法见算法 1。

算法 1: 拥塞热力图的计算方法 (下列所有值都是实际值映射在 0 至 128 的区间内)

输入: 待布局宏 M_i 的位置 (M_i^x, M_i^y) , 尺寸 (M_i^w, M_i^h) , 宏的引脚 P_i^j 的偏移量 $(\Delta x_i^j, \Delta y_i^j)$, 网络 Net^j 的最大物理边框 MaxMinBox, 其包含的引脚数量为 k , 其参数为 $x_{\max}, x_{\min}, y_{\max}$ 和 y_{\min} , 128×128 的零值矩阵 \mathbf{H} , 其中每个元素的位置坐标表示都要被 M_i 尝试占据。

输出: 尺寸为 128×128 的矩阵 \mathbf{A} 。

Step1: 将 M_i 的位置坐标在 \mathbf{A} 中滑动

Step2: 计算出 M_i 上所有引脚 P_i^j 的映射在 128×128 空间大小的实际位置坐标 $(P_i^{j,x}), (P_i^{j,y})$ 。

Step3: 初始化 128×128 零值矩阵 \mathbf{G} 。

Step4: 轮询 P_i^j 并获取包含当前 P_i^j 的 Net^j 的 MaxMinBox 和 k 值, 若轮询结束跳转到 Step8。

Step5: 如果 $k=0$, 返回到 Step4。

Step6: 如果 $k=1$, 则在矩阵 \mathbf{G} 中, MaxMinBox 的空间内填充 s ,

其值为 $s = \frac{|P_i^{j,x} - x_{\max}| + |P_i^{j,y} - y_{\max}|}{|P_i^{j,x} - x_{\min}| \times |P_i^{j,y} - y_{\max}|}$, 返回到 Step4。

Step7: 如果 $k \geq 2$, 如果 P_i^j 在 MaxMinBox 内, 返回到 Step4, 如果 P_i^j 在 MaxMinBox 外, 更新 MaxMinBox 的边界参数, 但不保存, 在矩阵 \mathbf{G} 的新的 MaxMinBox 空间中叠加新的 s , 其值为 $s = \frac{|x'_{\max} - x'_{\min}| + |y'_{\max} - y'_{\min}|}{|x'_{\max} - x'_{\min}| \times |y'_{\max} - y'_{\min}|}$, 返回到 Step4。

Step8: 将矩阵 \mathbf{G} 中元素按大小排序, 计算前 128 个元素的均值 μ 设置为 M_i 在 \mathbf{A} 中的元素值, 返回到 Step1。

2.4 密集型奖励函数设计 (RF)

一个合适的奖励函数有利于智能体快速地收敛到最佳的解决方案。RAGPlace 的密集型奖励设计框架如图 6 所示, 它包括了以下三部分。

线长奖励: 线长奖励考虑了位置与线长之间的关系。通过测量新增加的路径长度, 可以将其作为一项奖励信号来引导智能体将宏布局到有利于减小线长增量的位置。这种奖励设计使得智能体倾向于选择更短的路径, 从而提高收敛速度。每一步的线长奖励设计方法见算法 2。

算法 2: 密集型线长奖励计算方法 (x 维度)

输入: 待布局宏 M_i 的位置 (M_i^x, M_i^y) , 尺寸 (M_i^w, M_i^h) , 宏的引脚 P_i^j 的偏移量 $(\Delta x_i^j, \Delta y_i^j)$, 网络 Net^j 的最大物理边框 MaxMinBox, 其包含的引脚数量为 k , 其参数为 $x_{\max}, x_{\min}, y_{\max}$ 和 y_{\min} 。

输出: $r_{\text{totalwire}}$ (线长奖励)。

Step1: 初始化 r_{wire}^x 。

Step2: 遍历 M_i 的引脚 P_i^j , 并计算出每个引脚的实际位置: $x_i = M_i^x + M_i^w / 2$ 。

Step3: 获取包含该引脚 P_i^j 的 Net^j 的 k 值, $k=0$ 跳转到 Step4, $k=1$ 跳转到 Step5, $k \geq 2$ 跳转到 Step6。

Step4: 更新所属网络的物理边框参数: $k=k+1, x_{\max}^i = x_{\min}^i = x_i$, 设置: $r_{\text{wire}}^x = 0$, 跳转到 Step7。

Step5: 更新所属网络的物理边框参数: $k=k+1, x_{\max}^i = \max(\hat{x}_{\max}^i, x_i), x_{\min}^i = \min(\hat{x}_{\min}^i, x_i)$, 设置: $r_{\text{wire}}^x = -|x_i - \hat{x}_{\max}^i|$, 跳转到 Step7。

Step6: 更新所属网络的物理边框参数: $k=k+1, x_{\max}^i = \max(\hat{x}_{\max}^i, x_i), x_{\min}^i = \min(\hat{x}_{\min}^i, x_i)$, 如果 $x_i \in [\hat{x}_{\min}^i, \hat{x}_{\max}^i]$, 设置: $r_{\text{wire}}^x = \hat{x}_{\max}^i - x_i$, 如果 $x_i < \hat{x}_{\min}^i$, 设置: $r_{\text{wire}}^x = -|\hat{x}_{\min}^i - x_i|$, 如果 $x_i > \hat{x}_{\max}^i$, 设置: $r_{\text{wire}}^x = -|x_i - \hat{x}_{\max}^i|$, 跳转到 Step7。

Step7: 累计线长奖励: $r_{\text{totalwire}}^x = r_{\text{totalwire}}^x + r_{\text{wire}}^x$, 返回 Step2。

拥塞奖励: 拥塞奖励函数描述了宏的预放位置与拥塞增量之间的关系, 并采用了多步反馈的设计, 给定步数结束后, 观察环境的拥塞程度, 并将其作为奖励信号反馈给智能体。该设计可以帮助智能体在布局宏模块时, 倾向于选择拥塞值较小的位置。

内在奖励: RND 模型的误差作为内在奖励是为了激励智能体探索更多新颖状态 $f^*(s_t)$ 和 $f(s_t)$ 行为。它正比于预测网络预测的状态特征和目标网络的状态特征之间的误差, 内在奖励的计算方法见式 (11):

$$r_{\text{rnd}} = k \times \|f^*(s_t) - f(s_t)\|^2 \quad (11)$$

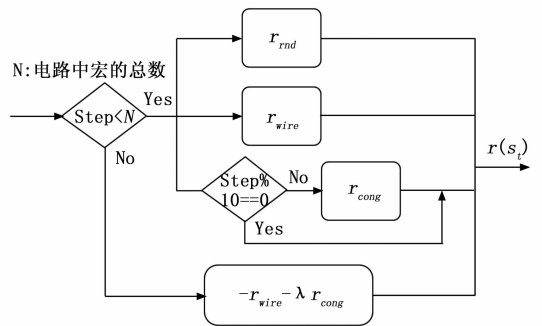


图 6 密集型奖励设计框架图

2.5 多重限制法 (MF)

2.5.1 惩罚限制

按大小布局宏时, 若考虑到宏上面所分布的引脚对布局质量的影响, 密集型奖励函数就会存在较多的负面奖励。产生这一现象的原因是前期每个引脚所属的子集网络都在生成物理边框, 这一过程中线长始终都在增加, 而一连串的负面奖励会导致智能体过于谨慎, 陷入局部最优。通过减小惩罚力度来减小前期一连串负面奖励对布局结果带来的影响有利于智能体敢于探索。

2.5.2 区域限制

在构建 PM 时, 要先验电路结构, 限制前几个宏的动作空间。通过在版图中间布局大型宏模块, 可以将其他小

型宏模块相对均匀地分布在周围，使整个芯片布局在物理上更加均衡。较大的宏模块通常具有更多的引脚，将它们布局在中间可以更方便地连接到其他功能模块和外部引脚，从而减少布线的长度和复杂度。

3 实验结果与分析

为验证 RAGPlace 的有效性，在开源数据集上进行了广泛的对比实验，并与最近的几种高级布局方法进行了比较 (DRAMPlace、GraphPlace 和 DeepPR)。由于 MaskPlace 没有开源，无法复现其实验环境 (如何计算总线长)，因此引用了该论文部分数据作为对比。这些方法都是在不同的公共电路基准上进行评估的，因此先前的工作都是通过以下实验进行评估的。

3.1 实验环境

详细的硬件配置和模型的超参数设置见表 2。

表 2 实验环境和参数配置

Configuration	Value	Configuration	Value
Optimizer	Adma	Lr	2.5×10^{-3}
Epoch	300	Epoch-update	Num Macros
Batch size	1	PPO epoch	4
Clip param	0.1	Num GPU	1
Discount λ	0.99	GPU	RTX 2080Ti
Value loss	0.5	CPU	Intel 4110
Entropy loss	0.01	System	Ubuntu

3.2 基准

ISPD2005 数据集包含了 8 个完整的实例，每个实例都有不同的规模和结构，包括电路的功能描述、约束条件和目标函数，以及所有需要在设计过程中考虑的设计规则，此外还提供了一个完整的参考布局和一个初始布局，以供算法进行优化。由于硬件条件的客观限制，“adaptec4”，“bigblue2”，“bigblue4”在 RAGPlace 无法获取有效的实验结果，因此仅在剩余 6 个测试电路上进行实验。8 个测试电路的详细信息见表 3。

3.3 实验结果与分析

为了评估 RAGPlace 在性能指标上要优于之前的工作，本实验复现了不同的方法，在布局完所有的宏和标准单元后对比了所获取的线长和拥塞。为控制变量，将 DeepPR 的动作空间设置为 128×128 ，由于 DeepPR 没考虑到真实

的宏尺寸大小和引脚信息，将 RAGPlace 中解决宏重叠的方法应用在 DeepPR 中，命名为 DeepPR-NO，在此基础上，进一步考虑引脚信息，命名为 DeepPR-NOP。从表 4 中，能够看出 RAGPlace 在 6 个基准电路中具有较低的线长 (“adaptec4”的线长质量仍然高于基于学习的方法，在 adaptec1”上，DreamPlace 与 RAGPlace 线长质量较为接近)，此外，也能发现传统的方法在一些电路布局中会导致布线失败，这是因为拥塞指标过高，无法在有限得宽度内走线。表 5 呈现的是从拥塞指标的角度来评价布局质量，在宏布局阶段完成后，利用 RUDY 算法获取全局拥塞信息，其结果表明 RAGPlace 因为其采用了多步反馈拥塞指标的形式在 4 个测试电路上表现优异。表 6 中，针对引脚信息的引入是否有利于 RAGPlace 获得更好的布局结果，为了简化实验，仅在 3 个测试电路上做了对比实验，从表中可以看出通过对引脚信息的编码，能够指导智能体给出更高质量的布局方案。

表 3 不同芯片基准的电路数据

benchmark	Macros	Fix Macros	Standard Ceil	Nets	Pins
adaptec1	543	29	210 904	3 709	11 001
adaptec2	566	14	254 457	4 346	13 815
adaptec3	723	10	450 927	6 252	17 953
adaptec4	1 329	18	494 716	5 939	20 371
bigblue1	560	9	227 604	657	3 633
bigblue2	23 084	16	534 782	33 407	105 630
bigblue3	1 293	66	1 095 519	5 569	22 019
bigblue4	8 170	42	2 169 183	28 873	140 529

3.4 消融实验

为了探索 RAGPlace 对布局结果的质量是否具有促进作用，将所设计的模块随机组合添加到 Baseline 上，通过实验记录了每个模块对线长和拥塞的影响。表 7 中的 Baseline 可以描述为对网表的特征输入采用 DeepPR 的改进特征嵌入模型 (状态空间改进为 128×128)，在强化学习方面，密集型奖励函数由随机蒸馏模型给出。从表 7 可以看出，RF 在 RAGPlace 中具有较大的贡献，可见合适的奖励函数设计不仅可以促进智能体快速获得结果，还提高了获得结果的质量。

值得注意的是，多重限制法在部分网表中表现不佳，原

表 4 不同布局方法的 HPWL ($\times 10^7$) 对比结果

Method	adaptec1	adaptec2	adaptec3	adaptec4	bigblue1	bigblue3
DreamPlace	7.73 ± 1.44	Routing faild	23.19 ± 2.82	21.58 ± 2.11	Routing faild	Routing faild
GraphPlace	8.66 ± 2.53	12.41 ± 2.53	25.80 ± 1.21	24.40 ± 1.27	16.85 ± 1.25	46.06 ± 2.53
DeepPR	7.16 ± 2.55	13.10 ± 1.83	22.47 ± 1.42	23.23 ± 1.84	17.20 ± 2.87	35.67 ± 1.53
DeepPR-NP	12.30 ± 2.89	18.15 ± 3.03	30.14 ± 3.70	28.46 ± 2.30	19.54 ± 3.66	47.21 ± 2.53
DeepPR-NOP	11.29 ± 1.07	16.18 ± 2.33	29.29 ± 3.55	27.00 ± 4.58	19.10 ± 4.12	46.06 ± 4.53
RAGPlace	7.15 ± 1.44	10.33 ± 1.18	21.95 ± 2.72	22.56 ± 2.46	11.57 ± 2.01	36.18 ± 3.53

表 5 不同布局方法的 Congestion 对比结果

Method	ada1	ada2	ada3	big1
DeepPR	0.743	0.835	0.742	0.732
DeepPR-NP	0.759	0.791	0.794	0.755
DeepPR-NOP	0.721	0.810	0.742	0.781
MaskPlace	0.691	0.699	0.713	0.695
RAGPlace	0.652	0.591	0.614	0.610

表 6 引入引脚信息下对宏模块的总线长估计结果

Method	ada1	ada2	ada3
RAGPlace+ Pins	2 680	18 367	15 472
RAGPlace+ Macros	2 854	20 568	18 003

表 7 各个模块对总线长 ($\times 10^7$) 的影响

Method	Benchmark			
	adaptec1	adaptec2	adaptec3	adaptec4
Baseline	10.89	17.60	30.25	29.00
Baseline+MR	10.14	15.57	28.51	28.18
Baseline+RF	8.89	12.49	23.45	25.24
Baseline+MF	9.56	13.20	23.69	22.69
Baseline+MR+RF+MF	7.15	10.33	21.95	22.56

因是多重限制法针对的是宏模块引脚数量差异大的情况, 对于 adaptec4 而言, 其宏之间引脚数量差异小, 其带来的效益较小。在表 8 中的 Baseline 仅仅将表示拥塞的特征图和奖励函数从 RAGPlace 模型中移除。由于拥塞奖励的设计通过拥塞视图来获取, 存在耦合关系不利于分解, 从表中可以看出多次不间断的反馈拥塞奖励可以促进拥塞指标的下降。此外, 为了验证引用 GAT 代替 GCN 是否可以促进布局质量, 在 adaptec1、adaptec2、adaptec4 和 bigblue3 网表上进行对比实验, 如表 9 所示, 在 adaptec4 网表中, 引入 GAT 并无促进作用, 这是由于内部宏之间的大小差异比较小, 而对于尺寸差异比较大的网表, GAT 的引入客观地促进了线长的减小。

表 8 多步反馈拥塞方法对拥塞指标的作用

Method	ada1	ada2	ada3
Baseline	0.721	0.688	0.703
Baseline+CH+r _{cong}	0.652	0.591	0.614

表 9 引入 GAT 模型对布局结果的影响

Method	ada1	ada2	ada4	big3
GCN+CNN	8.80	11.01	22.48	38.33
GAT+CNN	7.15	10.33	22.56	36.18

4 结束语

针对布局问题提出的 RAGPlace 是一种基于 DRL 框架的布局方法, 通过像素级图像去表示线长、位置和全局信

息, 在不减少搜索空间的情况下有效地顺序确定宏的位置。在不重叠的条件下, 该方法在所有关键指标上获得了有效的结果。这项工作大大的加速了芯片后端的布局过程, 提高了布局质量。在未来, 将通过设计合适的表示方法, 并结合 DRL 方法来探索多层布局布线技术

参考文献:

- [1] MIRHOSEINI A, GOLDIE A, et al. A graph placement methodology for fast chip design [J]. Nature, 2021, 594: 207-212.
- [2] LIN X, JIANG Z, GU J, et al. Dreamplace: Deep learning toolkit-enabled gpu acceleration for modern vlsi placement [J]. TCAD, 2021, 40: 748-761.
- [3] AMEYA R A, SATOSHI O, PATRICK H MADDEN. Recursive bisection placement; Feng shui 5.0 implementation de-tails [J]. ISPD, 2005: 230-232.
- [4] CALDWELL A E, KAHNG A B, et al. Optimal partitioners and end-case placers for standard-cell layout [J]. TCAD, 2000, 19 (11): 1304-1313.
- [5] YAOWEI C, SHIPING L. Mr: A new framework for multilevel full-chip routing [J]. TCAD, 2004, 23: 793-800.
- [6] 黄彩红. 基于自适应遗传算法的 DPR-FPGA 资源布局研究 [D]. 西安: 西安电子科技大学, 2021.
- [7] JINGWEI L, PENGWEN C, LU S, et al. Eplace: Electrostatics-based placement using fast fourier transform and nesterov's method [J]. DAC, 2014: 1-5.
- [8] CHUNG-KUAN C, ANDREW B K, et al. Replace Advancing solution quality and routability validation in global placement [J]. TCAD, 2018, 38: 1717-1730.
- [9] ALPERT C, ZHOU L, MOFFITT M, et al. What makes a design difficult to route [J]. ISPD, 2010: 7-12.
- [10] 郝睿, 蔡懿慈, 周强, 等. DrPlace: 基于深度学习的可布线性驱动布局算法 [J]. 计算机辅助设计与图形学学报, 2021, 33 (4): 624-631.
- [11] 曾诚. 时序驱动的 FPGA 高效布局算法研究 [D]. 西安: 西安电子科技大学, 2022.
- [12] DONG-JIN L, MARKON I. Obstacle-aware clock-tree shaping during placement [J]. TCAD, 2012, 31 (2): 205-216.
- [13] PASZKE A, GROSS S, FRANCISCO M, et al. Pytorch: an imperative style, high-performance deep learning library [J]. NIPS, 2019, 721: 8026-8037.
- [14] HAIQI Z, GUANGQUAN L, MENGMEI Z. Semi-supervised classification with graph convolutional networks [J]. NPL, 2022, 54: 2645-2656.
- [15] RUOYU C, XIANGLONG L, YANG L, JUNCHI Y, et al. On joint learning for solving placement and routing in chip design [J]. NeurIPS, 2021.
- [16] YAO L, YAO M, PING L. MaskPlace: Fast chip placement via reinforced visual representation learning [J]. NeurIPS, 2022: 24019-24030.

- [17] HONGXIANG F, CE G, WAYNE L. Optimizing quantum circuit placement via machine learning [J]. DAC, 2022: 19-24.
- [18] VELICKOVIC P, CUCURULL G, LIO P, et al. Graph attention networks [R]. Vancouver; ICLR, 2018.
- [19] RICHARD S, MCALLESTER D, SINGH S, MANSOUR Y. Policy gradient methods for Reinforcement learning with function approximation [J]. NIPS, 1999: 1057-1063.
- [20] NAM G J, ALPERT C J, VILLARRUBIA P, WINTER B, et al. The ispd2005 placement contest and benchmark suite [J]. ISPD, 2005: 216-220.
- [21] LITTMAN, MICHAEL L. Markov games as a framework for multi-agent reinforcement learning [J]. ICMLC, 2019: 157-163.
- [22] BURDA Y, EDWARDS H, KLIMOV O, STORKEY A. Exploration by random network distillation [J]. ICLR, 2019.
- [23] KAHNG A B, REDA A. tale of two nets: studies of wire-length progression in physical design [J]. ACM, 2006: 17-24.
- [24] SPINDLER P, JOHANNES F M. Fast and accurate routing demand estimation for efficient routability-driven placement [J]. ACM, 2007: 1226-1231.

(上接第 227 页)

围内, 则进行手动功能抽测, 例如在某个雷达出现了距离上无法稳定捕捉的问题, 此时自动测试系统无法进行详细测试工作, 为此需要将测试系统切换到手动测试模式。

在信号转接箱上完成自动测试模式到手动测试模式的转换, 此时测试设备完全处于手动测试状态。以距离捕捉问题进行说明, 手动接通雷达进行预热, 同时将射频信号源频率、脉冲宽度、延迟时间等设置在靶载雷达可接受的范围内, 目标模拟器调整至中心位置, 雷达预热完成后, 将雷达转换至向等效状态, 接通雷达高压, 通过示波器观察靶载雷达的回波信号, 如果雷达能够正常捕捉信号, 则进行正常延迟时间的调节, 检查距离捕捉的上下限, 完成性能指标的摸底。如果雷达不能捕捉信号, 且出现回波脉冲宽度较大的情况, 可以断定雷达距离波门处于“全通”的状态, 从而完成了故障定位在雷达的“接收组合”。

自动测试和手动测试是靶载雷达多功能测试系统的重要组成部分。在靶载雷达存在故障率偏高的情况中, 手动测试是定位雷达故障的重要手段, 同时能够完成性能下降靶载雷达的性能摸底工作。

5 结束语

为满足靶载雷达功能测试, 针对该型雷达故障率偏高的特点, 便于开展雷达测试、故障定位, 设计了一型靶载雷达多功能测试系统, 系统能够完成自动化测试、手动测试, 能够准确定位靶载雷达故障位置, 同时具备数据分析处理能力, 完成了雷达上电压信号、供电信号的采集, 测试人员能够顺利完成雷达的测试和采集功能。靶载雷达多功能测试系统能够完成的自动化测试和手动测试, 工作稳定可靠, 达到了功能设计要求。

参考文献:

- [1] 王子龙, 路景泽. 基于 CPCI 总线的雷达导引头测试系统设计与实现 [J]. 计算机测量与控制, 2016, 24 (7): 141-143.
- [2] 舒梯翔. 某雷达导引头通用自动化测试设备的研制 [J]. 宇航计测技术, 2009, 29 (1): 25-29.
- [3] 郭磊, 冉黎森. 雷达导引头自动测试系统设计 [J]. 计算机测量与控制 2018, 26 (10): 106-109.
- [4] 马艳. 某型雷达自动测试设备的设计 [J]. 计算机测量与控制, 2012, 20 (12): 3260-3262.
- [5] 李磊, 李玲玲. 某雷达测试设备控制系统的设计 [J]. 火控雷达技术, 2017, 46 (2): 68-74.
- [6] 顾振杰, 刘宇. 注入式雷达测试系统构建方法研究 [J]. 现代防御技术 2016, 44 (5): 155-160.
- [7] 陈靖宇, 刘收. 新一代综合自动测试系统在船用雷达测试保障中的应用研究 [J]. 计算机测量与控制, 2020, 28 (4): 146-151.
- [8] 肖开健, 李王玉, 游俊, 等. 某雷达检测维修设备开发与技术研究 [J]. 火控雷达技术, 2023 (1): 146-150.
- [9] 李磊, 李玲玲. 某雷达测试设备控制系统的设计 [J]. 火控雷达技术 2017, 180 (46): 69-74.
- [10] 段道聚, 张永祯, 张鹏, 等. 雷达装备自动测试系统的通用性设计 [J]. 现代雷达, 2015, 37 (10): 69-72.
- [11] 杨幸, 龚琦, 范文晶, 等. 某发控设备一体化自动化测试系统研究 [J]. 计算机测量与控制, 2023, 31 (6): 53-58.
- [12] 姜鑫伟, 郑元珠, 宋常浩, 等. 基于虚拟测点的雷达测试性建模技术研究 [J]. 兵工装备工程学报, 2023 (9): 254-260.
- [13] 闫淑群, 黎玉刚, 母勇民, 等. 基于 PXI 总线的导弹自动测试系统设计 [J]. 计算机测量与控制, 2011, 19 (8): 1925-1928.
- [14] 娄宇, 欧阳晓辉. 基于 PXI 总线的导弹测试系统设计 [J]. 计算机测量与控制, 2015, 23 (3): 828-830.
- [15] 陈亮, 曹兴冈. 基于 PXI 总线的电子装备测试系统设计 [J]. 科学技术与工程, 2011, 33 (11): 8243-8246.
- [16] 刘建斌, 张淑萍. 基于 LabWindows/CVI 的遥测数据采集系统设计 [J]. 测控技术, 2014, 33 (1): 31-35.
- [17] 胡小江, 白云, 雷虎民. 基于虚拟仪器的导弹自动驾驶仪测试系统设计 [J]. 计算机测量与控制, 2014, 22 (10): 3237-3241.
- [18] 孙玉宝, 赵广宁, 陶震鹏. 基于虚拟仪器的导弹测试自动记录仪设计 [J]. 四川兵工学报, 2010, 31 (5): 43-44.
- [19] 李树盛, 魏清新, 李祺, 等. 基于导弹总体效能的设计试验保障一体化测试技术 [J]. 计算机测量与控制, 2013, 21 (6): 1412-1414.
- [20] 罗华峰, 周旦辉, 吕隽. 舰空导弹测试项目优化方法研究 [J]. 现代防御技术, 2010, 38 (3): 31-38.