非默认规则线技术下基于多策略的时延驱动层分配算法

刘耿耿[1),2)] 鲍晨鹏[1)] 王 鑫[3),4)] 郭文忠[1)] 陈国龙[1)]

[1)](福州大学计算机与大数据学院 福州 350116)

[2)](计算机系统结构国家重点实验室 北京 100190)

[3)](天津大学智能与计算学部 天津 300350)

[4)](天津市认知计算与应用重点实验室 天津 300350)

摘 要 层分配作为超大规模集成电路物理设计中的关键环节，在决定布线方案的时延起到非常重要作用.为了优化集成电路的时延性能，现有的层分配工作通常注重优化互连时延和通孔数量，但要么未考虑到对线网中时序关键段的分配问题，要么对线网段的时序关键性的表示不够合理，最终使得算法的时延优化不够理想.为此，本文提出一种非默认规则线技术下基于多策略的时延驱动层分配算法，主要包含4种关键策略：(1)提出轨道数感知的层选择策略，增强层分配器为线网段选择合适布线层的能力；(2)提出多指标驱动的初始线网排序策略，综合考虑线长、信号接收器数和可布线轨道资源等多个指标为线网确定层分配优先级，从而获得高质量的初始层分配结果；(3)提出线网段调整策略，通过重绕线网，将时序关键段调整至上层布线层，优化线网时延；(4)提出线网段时延优化策略，对存在溢出线网进行拆线重绕，从而可同时优化时延和溢出数.实验结果表明，本文提出的算法相比于现有的层分配算法能够在时延和通孔数两个指标上均取得最佳，并且保证不产生溢出.

关键词 超大规模集成电路；层分配；时延；通孔；时序关键段

中图法分类号 TP18 **DOI号** 10.11897/SP.J.1016.2023.00743

Multi-Strategy Delay-Driven Layer Assignment for Non-Default-Rule Wiring Techniques

LIU Geng-Geng[1),2)] BAO Chen-Peng[1)] WANG Xin[3),4)] GUO Wen-Zhong[1)]
CHEN Guo-Long[1)]

[1)](*College of Computer and Data Science*, *Fuzhou University*, *Fuzhou* 350116)

[2)](*State Key Laboratory of Computer System and Architecture*, *Beijing* 100190)

[3)](*College of Intelligence and Computing*, *Tianjin University*, *Tianjin* 300350)

[4)](*Tianjin Key Laboratory of Cognitive Computing and Application*, *Tianjin* 300350)

Abstract As a key step in the physical design of very large scale integration, layer assignment plays a very important role in determining the delay of routing solution. At the same time, with the continuous progress of technical nodes, the density of the nets is increasing. Interconnect delay is an important factor to evaluate the performance of the routing results. In order to optimize the delay in integrated circuits, the existing layer assignment algorithms usually focus on minimizing interconnect delay and via count. However, the existing work either does not consider

收稿日期：2022-01-27；在线发布日期：2022-12-03.本课题得到国家自然科学基金(61877010)、计算机体系结构国家重点实验室开放课题(CARCHB202014)资助.刘耿耿，博士，副教授，博士生导师，中国计算机学会(CCF)高级会员(75198S)，主要研究方向为EDA算法研究、计算智能及其应用.E-mail：liugenggeng@fzu.edu.cn.**鲍晨鹏**，硕士研究生，主要研究方向为EDA算法研究.**王 鑫**，博士，教授，博士生导师，中国计算机学会(CCF)杰出会员，主要研究领域为大规模知识处理、计算智能及其应用.**郭文忠**(通信作者)，博士，教授，博士生导师，中国计算机学会(CCF)杰出会员，主要研究方向为EDA算法研究、计算智能及其应用.E-mail：guowenzhong@fzu.edu.cn.**陈国龙**，博士，教授，博士生导师，中国计算机学会(CCF)高级会员，主要研究方向为EDA算法研究、计算智能及其应用.

the assignment of the timing-critical segments in nets, or the time critical representation of wire segments is not reasonable, which ultimately makes the delay optimization not ideal. For this reason, this paper proposes a multi-strategy delay-driven layer assignment for non-default-rule wire technique, which mainly includes the following key strategies: (1) In order to avoid the local optimization of the nets, a track number aware layer selection strategy is proposed. This strategy enhances the ability of layer assigner to select suitable routing layers for wire segments so that it allows the layer with the largest number of remaining tracks to be given priority in the layer assignment to avoid overflow in the routing layer. (2) A multi-index driven initial net sorting strategy is proposed. This strategy fully considers the characteristics of different nets to determine the priority of nets, and comprehensively considers multiple indicators such as the wirelength, the number of sink points, and the resource of routing tracks to determine the priority of layer assignment for the network, so that the high-quality initial layer assignment results is obtained. (3) In order to further optimize the delay, a wire segment adjusting strategy is proposed. This strategy is adopted to optimize the delay of nets by re-assigning nets and assigning the timing-critical segments on upper layers. (4) A wire segment delay optimization strategy is proposed to optimize the delay of nets while eliminating overflow by ripping up and re-assigning the nets which have overflow. This strategy not only makes the final routing result not overflow, but also makes it have better delay. And the proposed algorithm is mainly composed of three stages. First, a better initial layer assignment result is obtained through the multi-index driven initial net sorting strategy. Then, the proposed algorithm adjusts the routing layer of the timing-critical segments through the wire segment adjusting strategy, and then eliminates the overflow while optimizing delay through the wire segment delay optimization strategy. Finally, under the condition of satisfying the congestion constraint, the proposed algorithm rips up and re-assigns the nets to select a better layer assignment result. In each stage, the track number aware layer selection strategy is used in the single net layer assignment so that the track resources of upper layers can be fully utilized. This paper uses the DAC12 benchmarks to verify the effectiveness of the related strategies and the proposed algorithm. The experimental results show that the proposed algorithm can achieve the best performance in both delay and via count among the existing algorithms without overflow.

Keywords very large scale integration; layer assignment; delay; via; timing-critical segments

1 引 言

随着集成电路规模的不断扩大,晶体管和互连线特征尺寸进入到纳米级范畴[1]. 同时,由于单位面积所需布线线网数量的持续增加,仅仅依赖于单金属层结构的布线区域已经不能满足高密度布线的需求,因此提出了多层布线结构[2]. 为应对日益复杂的电路设计,各个金属层具有不同的厚度[3]. 不同厚度金属层承载着不同宽度导线,导线线宽越大,导线电阻越小,进而时延越小. 为此,多层布线器可以选择更加灵活的布线空间,从而提高了线网规避各种时序违规的能力. 而层分配作为总体布线与详细布线的中间步骤,需要优化层分配结果以满足各种时序约束并且考虑与详细布线需求之间的匹配.

在多层结构布线的背景下,现有的总体布线器主要分为3维(3-dimensional, 3D)布线器和2.5维(2.5-dimensional, 2.5D)布线器两大类. 3D布线器[4-6]直接在多层布线区域得到连接信号发射器与多个信号接收器的合适线网,从而获得高质量的布线结果. 但以文献[4-5]为代表的3D布线器由于直接在多层布线空间上进行布线,导致其算法复杂度高,所需运行时间过长. 文献[6-8]将机器学习的方法引入到总体布线领域,为集成电路设计开阔了一种新的设计思路,但该类算法所解决的测试电路规模小,且存在着运行时间长的问题. 为缩短运行时

间，基于并行机制的布线器引起了学术界广泛研究[9-12]. 文献[9]提出了一种高效的并行框架，有效地避免了布线结果陷入局部最优. 同时，文献[10-11]使用了并行机制对矩形 Steiner 最小树进行构建. 文献[12]提出了通孔感知的并行层分配算法. 但这些布线器大多不能获得各方面指标较为理想的布线结果. 而 2.5D 布线器[13-16]则先使用 2 维(2-dimensional，2D)总体布线器快速得到 2D 拓扑，再通过层分配方法将 2D 布线线网映射成 3D 总体布线结果，可快速获得高质量的布线方案，从而成为主流的布线方法.

随着制造工艺在纳米领域的持续推进[17]，互连时延已经成为时序收敛问题的瓶颈，取代晶体管时延成为影响芯片性能的决定因素[18]. 为此，一方面为了在详细布线前得到时序收敛的总体布线方案，文献[19]将 X 结构总体布线算法引入到布线问题，得到既能有效绕障又能保证时序收敛的优质布线树. 另一方面，线网时延由导线时延和通孔时延组成，而通孔时延与通孔数密切相关，故降低通孔数是减小时延的一种重要方式. 因此，时延优化和通孔数优化是层分配算法设计中的重要目标[20]. 基于整数线性规划方法[21]以及基于动态规划方法[22-23]的层分配算法相继被提出，其中，前者不可避免地损耗大量的运行时间，而后者的运行时间较快，但得到的层分配结果不够优秀. 例如，文献[22-23]均通过在基于动态规划的单一线网层分配算法中调整溢出惩罚项以快速规避拥塞区域，但尚未考虑时延等重要的性能指标. 而文献[24]虽然有效降低线网时延，但在通孔数量的优化力度不足.

为充分利用不同金属层的物理特性以降低信号传输时延，文献[25-26]通过评估线网时序关键性，将时序关键线网调整至上层金属层，即低时延层，提高了算法对线网时延的优化效果. 但由于低时延层布线资源紧张，并且文献[25-26]未考虑时延关键段在层分配中的重要影响，因此文献[25-26]仅能将部分时序关键线网中导线调整至低时延层. 文献[27-28]在多线程条件下提出基于迭代的整数线性框架以减少潜在的时延违规. 文献[29]将动态规划框架与线性规划方法相结合以降低通孔数对详细布线的影响. 文献[30]结合 2D 图案布线和层分配得到针对两引脚线网的快速 3D 布线方法，但拆分线网仍无法摆脱逐网布线过程中的局部最优问题，不利于最终的布线质量.

此外，随着集成电路制程技术的发展，一系列先进制造技术被广泛应用于芯片设计领域，例如，双重图案光刻技术为传统布线引入了非默认规则线技术. 与默认规则线相比，非默认规则线通过增大导线宽度以降低导线电阻，进而优化导线时延. 然而，使用非默认规则线的代价之一是消耗更多布线资源. 因此，对于进入纳米级的超大规模集成电路，面对并不充裕的布线资源，需要充分考虑非默认规则线的使用范围和条件，否则会严重损害最终方案的可布线性. 因此，在层分配中设计合理使用非默认规则线的方法对于优化时延尤为关键.

文献[31]将先进制程中的非默认规则线技术引入到层分配工作，通过混合使用不同类型导线，充分发挥了非默认规则线技术在时延优化上的潜力. 但由于导线规格尺寸的影响，过密的导线分配所增强的耦合效应将不可避免，从而影响线网的时序性能. 文献[32]在引入耦合电容估计方法的基础上，采用基于协商的方式动态调整了线网中的时序关键段，但只针对单个线网中不同线网段进行关键性的区别分配，不能综合识别与其竞争同一布线层的其他线网中线网段的时序关键性，从而对时延优化不够理想. 同时，文献[32]仅在发生溢出时才进行代价更新，而在对一导线段进行层分配时，两种方案即使均不会导致溢出，却会在影响局部拥塞和后续布线的灵活性方面有所区别. 因此，该方法并不能精确评估当前金属层的可布线性，从而影响层分配器对不同层 3D 网格边的代价计算的准确程度. 为了弥补这一不足，文献[33]提出的拥塞感知策略将各层容量分布信息引入拥塞代价评估，增强了层分配器为线网段选择合适布线层的择优能力.

此外，以往工作通过增加代价函数的目标参数来寻找最优解，主要存在以下不足：(1) 在层分配过程中只考虑当前线网的层分配效果最优，而忽略对后布线线网的影响，容易造成局部最优；(2) 所采用的层分配顺序会导致层分配过程对布线资源的利用不够充分，需要设计一种更为有效的层分配顺序；(3) 没有考虑未溢出线网的时延优化问题，导致层分配最终的时延优化能力受限.

为此，针对上述不足，本文以时延和通孔数为优化目标，提出一种有效的基于多策略的时延驱动层分配算法，其主要包含轨道数感知的层选择策略、多指标驱动初始线网排序策略、线网段调整策略以及线网段时延优化策略. 通过在多组标准的测试电路上的实验结果表明，本文算法相比于现有工作，能够保证不产生溢出的情况下在层分配问题中

时延和通孔数这两个重要的优化目标均取得明显优化.

2 问题表述

2.1 问题模型

实际集成电路布线往往分为总体布线和详细布线两个阶段. 总体布线忽略了一些在详细布线中考虑的布线因素，进行宏观布线，并不关心导线在轨道上具体位置，要求在较短时间内得到一个理想的总体布线方案，为详细布线提供指导. 本文采用大多数总体布线算法的布线模型，将多层结构中的金属层划分成多个大小相等的矩形区域，每个矩形区域定义为一个矩形单元，相邻的矩形单元之间存在一定数量的可布线轨道，用于分配线网段. 每个金属层只有水平或垂直一个走线方向，相邻两个金属层之间的走线方向相互正交.

如图1所示，m_1、m_2、m_3分别代表不同布线层，A和B为相邻的两个矩形单元，相邻金属布线层之间用通孔连接. 布线器将多层布线空间映射成一个3D网格图，并将每个矩形单元抽象成3D网格图中的一个点. 同一走线方向上相邻两个矩形单元之间的边界被映射为一条边，对应着具有一定数量的可布线轨道.

由于矩形单元中可能存在各种障碍，所以同一层不同边的可容纳轨道数量也可能不尽相同. 当层分配器在某一个3D网格边上所分配的导线数量超过该边的可布线轨道数时，该3D网格边产生溢出. 网格边溢出意味着后续布线过程中该网格边没有足够轨道用于承载导线，破坏了可布线性. 关于边的溢出数的计算公式[29]如下：

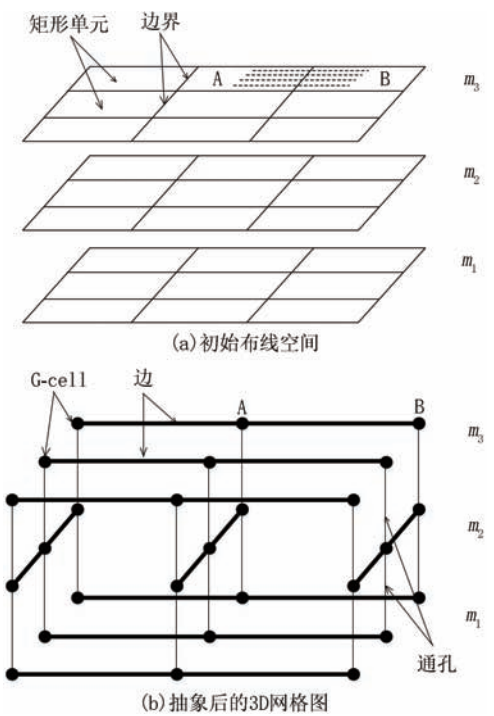

图1 金属层的处理和抽象

$$overflow=\begin{cases} use(e)-cap(e), & use(e)>cap(e) \\ 0, & \text{否则} \end{cases} \tag{1}$$

其中，$use(e)$表示在边e分配的导线数量，$cap(e)$表示e可容纳导线的轨道数.

为简化层分配空间模型，给定一个三元组(G^k, G, S)，其中$G^k(V^k, E^k)$表示k层的布线空间，即k层走线方向相互正交的布线层. 用V^k表示k层布线空间中的点集，E^k表示k层布线层中带有一定轨道容量的边集. $G(V, E)$是k层布线层在二维平面上的水平投影，V和E分别表示2D布线空间中的点集和边集. 对于2D网格点v_i，$v_{iz}(1\leqslant z\leqslant k)$表示该点在3D网格图中的映射，同样的$e_{iz}(1\leqslant z\leqslant k)$表示2D网格边$e_i$在3D网格图中第$z$层的映射. S表示已知的2D布线结果. 在已知2D布线结果以及3D网格容量分布情况的前提下，层分配将2D拓扑中每一个线网段合理地分配到各个金属层，尽可能优化整个线网集合的时延和通孔数等层分配问题中最重要的优化目标.

多层布线结构为层分配过程提供多个对应不同宽度和厚度的金属层. 由于高梯队布线层所使用的默认规则线比低梯队布线层所使用的默认规则线的宽度更大，间距更大，被分配在高梯队布线层的2D线网段对应的电阻值更小，通常具有更小的时延. 因此在上层有限的可布线空间的约束下，如何有效地利用高梯队布线层的轨道资源，对最终层分配的时序效果起到重要影响.

2.2 时延计算

本文采用具有良好保真度的Elmore模型来计算线网时延. 一个线网具有一个信号发射器(源点)和多个信号接收器(漏点)，其中，信号发射器具有驱动电阻(电路信号发射器的电阻)，信号接收器具有负载电容. 每个接收器都需要和发射器相连才可以完成信号的传输. 线网中被分配到同一布线层相邻两个矩形单元之间的部分称为线网段. 从一个信号发射器到一个信号接收器所经过线网段的集合，被定义为一条路径. 一个线网中通常含有一条或多条路径. 依据Elmore模型的时延计算公式，对于线网中线网段s的时延计算公式[32]如下：

$$delay(s)=R(s)\times\left(\frac{C(s)}{2}+C_d(s)\right) \quad (2)$$

其中，$R(s)$和$C(s)$分别表示s段的电阻和电容，$C_d(s)$表示s段的下游电容. 那么一条路径的时延大小等于各部分线网段时延的累加和[32].

$$delay(si)=\sum_{s\in path(si)}delay(s) \quad (3)$$

其中，$path(si)$表示s段所处路径，si表示该路径上的信号接收器. 对于单个线网而言，线网的总时延大小等于各路径时延的加权和[32].

$$delay(N)=\sum_{si\in S(n)}\alpha_{si}\times delay(si) \quad (4)$$

其中，α_{si}是信号接收器si的权重值，表示该路径占所处线网的时延比重. 参考之前相关工作中[32, 33]假定各路径的时延比重相同，本文将α_{si}设置为该线网信号接收器数量的倒数.

2.3　非默认规则线

依据文献[32-33]中的层分配问题模型，本文将3D网格图设定为9层布线层结构，并将布线层分为3个梯队. 其中1-4层处于第一梯队，5-7层处于第二梯队，8-9层处于第三梯队. 具有一般规格(默认线宽)的导线称为默认规则线，而具有特殊预定线宽的导线称为非默认规则线. 非默认规则线主要分为平行线和宽线. 由于亚16纳米设计的低层使用双重图案光刻制造，进而引入许多严格的制造和设计规则，因此非默认规则线在低4层以平行线的形式存在，在高5层以宽线形式存在[32]. 同时，先进制程下集成电路布线中导线不能具有任意线宽，只能使用由代工厂认证的某些默认线宽的导线. 多层结构中不同梯队的金属层有不同默认线宽，即三个梯队的默认线宽分别对应着1W、2W和4W. 金属层上默认的放置导线段的位置称作轨道，每一段轨道可容纳一段具有默认线宽的默认规则线.

同时，由于制造工艺的严格限制，形成了严格的布线规则约束，要求导线线宽越宽，导线之间的间隙越大. 非默认规则线相比于默认规则线需要占据更多的轨道资源. 图2(b)是将图2(a)中的默认规则线替换成非默认规则线后的结果. 其中，m_1层的平行线需要占用2条布线轨道，m_5层的宽线则需要占用3条布线轨道. 在同一布线层，由于非默认规则线相较于默认规则线占据更多轨道，线宽更宽，在物理特性上其电阻更小，因此替换默认规则线后的图2(b)相比于图2(a)可以有效地降低线网时延.

在轨道数量有限的布线环境中，层分配器为线网段合理地选择不同类型的导线，有助于充分发挥非默认规则线在时延优化上的潜力.

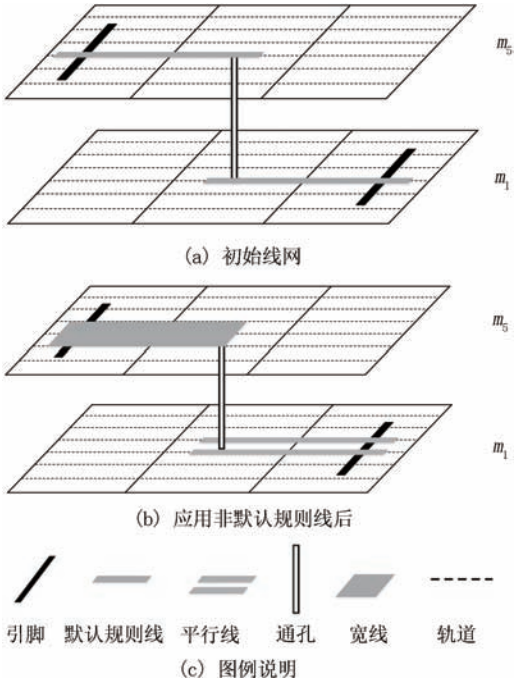

图2　非默认规则线的应用实例

2.4　耦合效应

布线问题中的耦合效应主要指的是导线之间由于电磁场的相互作用而导致对电学特性的影响. 导线之间的距离越近，耦合效应的影响更强. 在超大规模集成电路中，由于同一布线层中所分配的导线密度大，耦合效应所造成的影响不能够忽略.

由于在总体布线阶段不进行轨道分配，而精确的耦合电容取决于具体的轨道分配结果. 因此，本文采用多次分配的概率估计方法[32]得到相对准确的平均单位电容作为耦合电容.

2.5　拥塞约束

为了进一步提高总体布线与详细布线之间的匹配程度，需要保证网格边不发生溢出，进而保证线网在3D网格中的可布线性. 因此，本文算法为线网设置拥塞约束[32-33].

$$TotalO(S_k)\leqslant TotalO(S) \quad (5)$$

$$MaxO(S_k)\leqslant\lceil MaxO(S)\times(2/k)\rceil \quad (6)$$

其中，S和S_k分别表示给定的2D布线结果以及由2D拓扑得到的3D布线结果. k表示金属层数量. $TotalO$和$MaxO$分别表示总溢出数和最大溢出数. 总溢出数等于所有网格边溢出数之和，而最大溢出数等于所有网格边溢出数中的最大值. 同时，文献[23]证明了在满足拥塞约束的条件下，若初始2D拓扑中存在溢出，最终得到的层分配方案将不可避

免产生溢出. 若初始2D拓扑中未产生溢出，则必定存在满足拥塞约束的可行层分配方案，即任意3D网格边上均满足溢出数为0.

随着先进制程中非默认规则线技术的引入，非默认规则线加剧了布线层可布线资源的紧张程度. 而通过引入拥塞约束，可以尽可能规避3D布线结果中溢出情况的发生.

2.6 时序关键段

定义1. 时序关键性. 线网或线网段对整体时序产生的影响. 由于时延是衡量线网时序的重要指标，本文中时序关键性主要指线网或线网段对整体时延的影响.

定义2. 时序关键段. 同一线网中时序关键性强的线网段，即所在线网中时延比重较大的线网段.

定义3. 时序关键线网. 线网集合中时序关键性强的线网. 即线网集合中时延比重较大的线网.

定义4. 违规线网. 不满足拥塞约束公式(5)和(6)中任一公式的线网.

时序关键线网的层分配情况在很大程度上影响了最终总时延. 而2D线网中部分导线段的布线层选择可以在很大程度上影响该线网的时延，因此本文将该部分线网段定义为时序关键段. 一方面，基于Elmore时延计算模型，越靠近信号发射器的线网段通常会有更大的下游电容，而相比于下游段，上游段的时延值更大. 另一方面，多条路径相重叠的线网段由于下游电容的叠加，其布线层的选择也在很大程度上决定了整个线网的时延大小. 因此，这两种情况下的线网段都属于时序关键段. 同时，对于上述两种情况，时序关键段拥有更大的下游电容值，因此本文以下游电容的大小近似表征线网段的时序关键性.

3 基于多策略的时延驱动层分配

3.1 算法的完整流程

基于多策略的时延驱动层分配算法主要包括三个阶段：均衡排序初始化阶段、基于下游电容的时延优化层分配阶段和后优化阶段. 图3是基于多策略的时延驱动层分配算法的完整流程图.

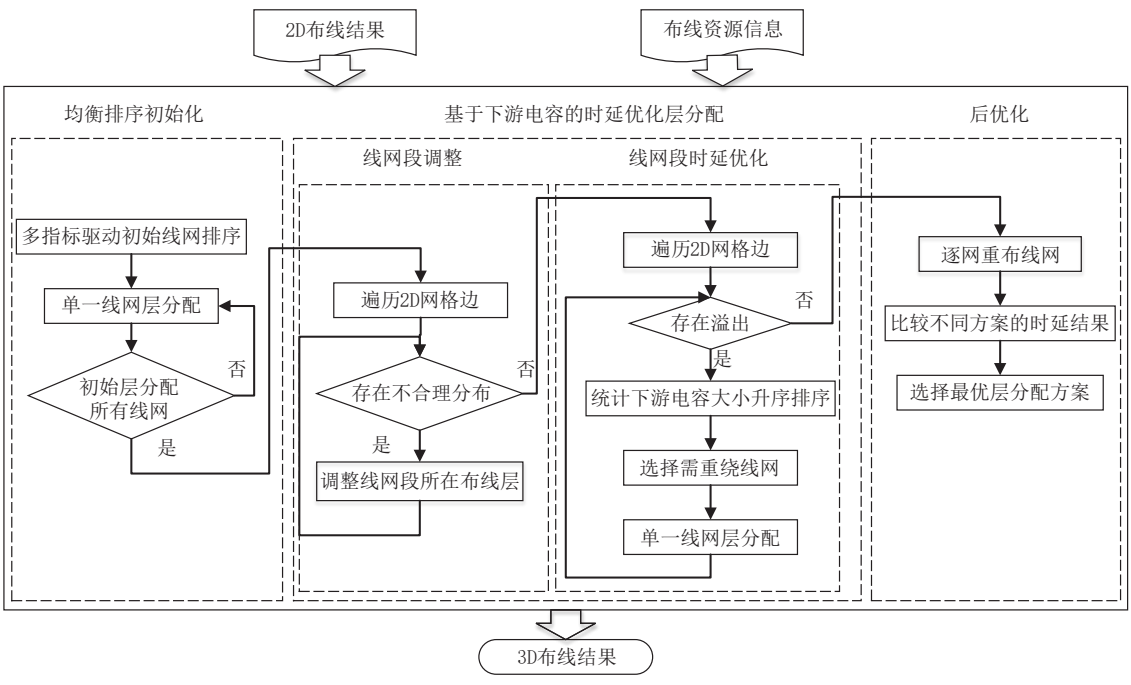

图3 本文算法的流程

在每一阶段，层分配器均通过使用轨道数感知的层选择策略进行目标代价的计算，即每一次线网布线过程都尽可能地将线网段分配在剩余轨道数量最多的金属层，以期保证所有金属层的剩余轨道数量尽量不为负数，进而为下一个预分配的线网预留更多可选布线层. 公式(7)和(8)分别为单一线网动态规划求解的目标函数[32]和拥塞代价计算公式.

$$cost(n)=\alpha\times delay(n)+\beta\times \#v_n+\sum_{s\in n} cong(s_e) \tag{7}$$

$$cong(s_e)=M_0\times trc(s_e)+M_1\times ofc(s_e) \tag{8}$$

其中，$cost(n)$表示线网n的总代价，$delay(n)$和$\#v_n$分别表示线网n的时延和通孔数，$cong(s_e)$表示线网段s在边e处的拥塞代价．trc表示轨道剩余代价，ofc表示溢出代价．α，β，M_0，M_1是自定义权重系数，为了满足层分配各阶段所侧重的主要优化目标，公式(7)和(8)的权重系数在不同阶段需采取不同的取值．同时，目标函数中时延、通孔数以及拥塞代价的数值并不处于一个数量级，因此，权重系数和并不为1．为了避免其失衡，进而更精确地评估各个层分配方案的优劣性，本文采用控制变量的方法，在保持其他参数不变的基础上，针对某一个参数在一个合理的实数范围内，以等梯度增长的方式进行多次实验测试，以此得到各阶段较为合理的权重系数．

均衡排序初始化阶段：在布线问题中电路的线网顺序是随机定义，很多层分配算法按照该顺序进行初始层分配得到的线网很大程度上不能满足约束，大大增加了后续阶段修正线网约束的工作量．为此，本文提出了均衡排序初始化阶段，能够将预分配线网以特定的顺序均匀分配在上方金属层，获得一个时延较好的初始层分配结果．该阶段主要分为两个步骤：(1) 运用多指标驱动初始线网排序策略为各个线网建立分配优先级；(2) 基于线网的优先级大小，依次为每个线网进行初始层分配．为保证优先优化线网时延，算法为时延权重α赋予较大值10．而通孔权重β的值若过大，则基于下游电容的时延优化层分配阶段无法在拆线重绕过程中获得有效的层分配方案．本文通过多次实验最终将其取值范围设为0.5～3.5．公式(8)的权重系数M_0和M_1可以有效地控制轨道剩余代价和溢出代价对层分配过程中的影响．初始层分配过程为获得各层相对均匀的层分配结果，将大幅度增大轨道剩余代价的影响，如初始层分配过程中算法将M_0设为12．同时，为避免溢出代价过大影响线网段的布线层选择，需将M_1设为较小值，即M_1取值范围为0～1，降低层分配过程前期溢出代价的影响．在进行初始层分配过程中，算法将目标函数中的α，β，M_0，M_1的值分别设为10，1，12，0.3．由于α，M_0的取值相对较大，因此层分配器能够在维持各层剩余轨道数量分布均衡的同时尽可能优化时延．

基于下游电容的时延优化层分配阶段：基于下游电容的时延优化层分配阶段主要由线网段调整策略和线网段时延优化策略组成．本文提出的线网段调整策略的主要作用是检查是否存在不合理分布(不合理分布是指在布线资源允许的情况下时序关键段被分配至高时延层，从而阻碍时延优化)．如若存在，本文算法将下游电容较大的上游段尽可能地调整至上层的可布线轨道．在这一阶段α，M_1仍为10和0.3，β，M_0的值则均被设为2，以进一步优化线网时延和控制通孔数．本文提出的线网段时延优化策略是尽量消除布线区域的溢出情况．该策略通过统计各边的可布线情况，对每一个溢出线网段所处线网进行拆线重绕，且每一次遍历2D线网，为溢出边更新溢出代价，即

$$ofc(e)=p_e\times h_e \tag{9}$$

其中，p_e和h_e分别表示该边e的惩罚项和历史代价[32]．

后优化阶段：在后优化阶段，算法通过使用宽线来进一步优化时延数值前5%的时序关键线网时延．后优化阶段定义了更大的拥塞代价权重，因此可以在保证不产生新溢出的情况下对所有线网进行最后一次拆线重绕，通过比较重绕方案与原方案的目标代价，将更优的层分配方案作为最终的层分配结果．为保证在进一步优化时延的同时，通孔数仍保持在理想的水平，适当增加了通孔权重．通过控制变量法，本文经过多次实验观察，将α，M_1设为10和0.3，将β，M_0均设为3.5．

3.2　多种有效策略的设计

3.2.1　轨道数感知的层选择策略

多层布线结构将布线层分成多个梯度，默认导线在不同梯度的布线层采用不同规格的默认线宽．但是在同一梯度上，不同金属层之间由于使用相同线宽的默认规则线，仅仅通过目标函数的代价值对比，层分配器不能有效地判断将线网段分配在哪一层更合理．针对这种情况，现有的拥塞感知策略[33]将各层轨道资源利用率作为衡量拥塞代价的因素，即若某布线层被使用的轨道数占可使用轨道数的比值越小，则优先分配该层轨道．该方法在同等轨道数量的布线层选择中可达到良好效果，但若各层的可使用轨道数量不一致，则轨道资源利用率高的层并不等于存在更少的可分配资源，往往得到不准确的代价评估．

定理1.　任意一个2D网格边e_i，其对应的任意不同层的3D网格边$e_{i,j}$和$e_{i,l}$，若满足条件$use(e_{i,j})/tc(e_{i,j})>use(e_{i,l})/tc(e_{i,l})$，则并不能得到$cap(e_{i,j})<cap(e_{i,l})$．证明过程详见附录1．

基于这一点，本文提出了轨道数感知的层选择

策略，该策略的轨道剩余代价公式如下：

$$trc(e_{i,l})=\frac{1}{1+e^{w(ei,l)}} \tag{10}$$

$$w(e_{i,l})=\frac{cap(e_{i,l})}{\sum_{l\in k}cap(e_{i,l})} \tag{11}$$

其中，$cap(e_{i,l})$表示2D网格边e_i在l层上的剩余轨道数，即$w(e_{i,l})$表示第l层剩余轨道占各层剩余轨道之和的比值.

本文根据公式(10)和(11)计算轨道剩余代价. 同时，轨道剩余代价与溢出代价组成了拥塞代价. 拥塞代价作为目标函数中必不可少的一部分，通过它可以选择出更好的层分配方案，从而有效指导整个层分配过程. 因为轨道剩余代价trc的计算公式(10)是sigmoid函数的变形，所以第l层的剩余轨道数量占总剩余轨道数量的比值越大，e_i边在l层的轨道剩余代价越小，e_i边在l层的拥塞代价越小，该层被选中分配线网段的可能性越大. 将轨道数感知的层选择策略应用到目标函数的拥塞代价中，能够使各个布线层保持相对均衡的轨道资源利用率，因此各层都会在很大程度上留有一定的剩余轨道空间，从而为下一个预分配的线网提供更加丰富的待选布线层.

例如，图4显示两层布线区域，每一条边的轨道总容量为8，黑色、灰色、浅灰色部分对应着已被障碍占用的轨道数分别为8，4，1. VC代表当前层分配结果的通孔数，$dc(e_{i,j})$代表3D网格边$e_{i,j}$被已布线的线网占用的轨道数. 图4(a)中的深色点和浅色点分别代表待连接线网n_1和n_2，灰色水平线代表被已布线的线网占用的轨道. 现需对线网n_1和n_2依次进行层分配. 若依据拥塞感知策略，$e_{2,3}$的拥塞代价小于$e_{2,1}$，因此n_1的层分配结果趋向于图4(b)，而当$e_{2,3}$的轨道空间被占满之后，n_2在可布线性的约束下不可避免地需要绕线到下层空间，因此图4(c)中会产生额外的通孔数和通孔时延.

本文提出的轨道数感知的层选择策略，并不专注于各层轨道资源利用率的大小，而是选择剩余轨道数量最多的金属层进行线网段分配. 图4(a)中$e_{2,3}$的容量为8，被已布线的线网占用的轨道数$dc(e_{2,3})$=6，已被障碍占用的轨道数为1，因此剩余轨道数$cap(e_{2,3})$=8-6-1=1. 同理可知，$cap(e_{2,1})$=8-2-4=2. 由公式(11)可得，$w(e_{2,3})=1/3$，$w(e_{2,1})=2/3$. 结合公式(8)和(10)，则相应的上层的拥塞代价大于下层，线网n_1所选择的布线方式如图4(d)所示. 在对线网n_2进行层分配时，线网可避免额外绕线，因此可以得到更优的层分配方案.

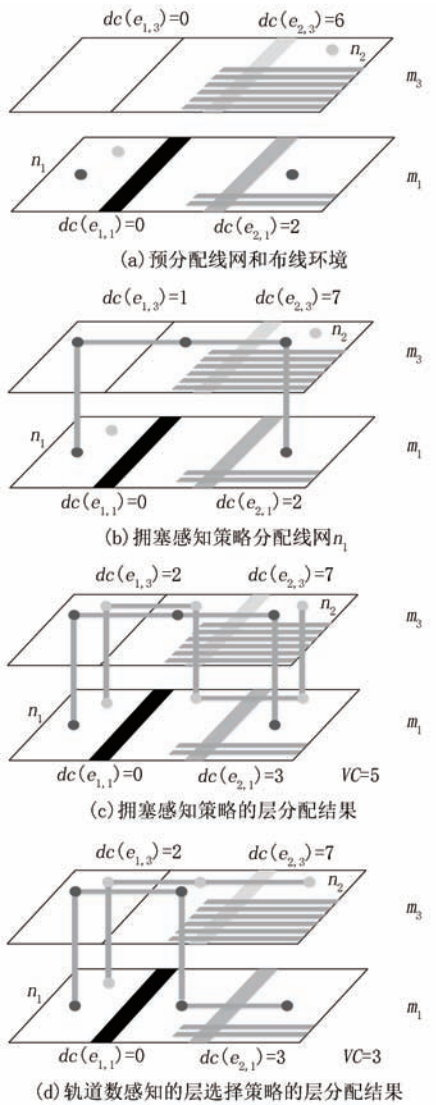

图4 轨道数感知的层选择策略实例

3.2.2 多指标驱动初始线网排序策略

初始线网的层分配结果影响了最终的层分配方案对上层资源的有效利用. 为了得到更为优秀的初始层分配结果，本文提出了多指标驱动初始线网排序策略，综合考虑线网线长、信号接收器数以及布线空间密度，为每一个线网计算分数，分数越高的线网的初始层分配优先级越大. 公式(12)是线网优先级计算公式.

$$Score(n_i)=M_2\times\frac{Pinn(n_i)}{Len(n_i)}+M_3\times avgDen(n_i) \tag{12}$$

其中，n_i表示2D总体布线线网. M_2和M_3是自定义变量，用于平衡前后两项的重要性. 本文将M_2和M_3分别设为1和0.5. $Pinn(n_i)$指的是线网n_i中的信

号接收器数. $Len(n_i)$表示线网n_i在2D网格图中的线长. $avgDen(n_i)$表示线网n_i在2D网格图中的布线密度. 在初始阶段布线前可根据输入的2D布线方案计算线网所经过的网格单元的数量，进而得到线长$Len(n_i)$. 同时，在统计2D布线方案中线网中每一段导线所经过的2D网格边e_i的导线密度$den2(e_i)$后，取其平均值即为线网n_i在2D网格图中的布线密度$avgDen(n_i)$. 在第一项中，$Pinn(n_i)/Len(n_i)$是$Len(n_i)/Pinn(n_i)$的倒数，而$Len(n_i)/Pinn(n_i)$近似表示线网中单个信号接收器的平均路径长度. 若线网中平均路径长度越长，则路径的可布线空间及布线灵活性增大，因此可以为所在线网赋予一个较小的分数. 而对第二项而言，布线密度越大，单个线网的可布线空间越小，因此需要赋予线网一个更高的分数，并对该线网优先进行初始层分配.

层分配器根据上述得到的优先级大小依次将线网分配至各布线层. 通过引入轨道数感知的层选择策略，层分配器可优先分配优先级较高的线网并且充分利用上层的可布线资源.

3.2.3　线网段调整策略

拆线重绕工作是层分配中常见的减少溢出的方法. 基于协商的层分配算法[32]通过不断重绕溢出线网以达到降低时延和规避拥塞的目的. 但常常忽略未溢出线网，限制了算法的时延优化能力. 同时，时序关键段优先占据低时延层可进一步优化时延，而传统层分配算法均未能充分考虑不同线网段的时序关键性.

为此，本文将下游电容作为评估线网段时序关键性指标，通过结合线网段调整策略和线网段时延优化策略，在规避拥塞的同时实现对线网总时延的进一步优化.

线网段调整策略通过调整布线空间中不合理分配的线网段，为线网段时延优化策略提供更多的可重绕线网. 图5显示的是多层布线结构中走线方向一致的3、5两层金属层，由于上梯度层所使用的金属导线更宽，因此线网段被分配在上层可以得到相对更小的时延. 假设上下层的可布线轨道数均为1，实心矩形表示信号发射器，空心矩形表示信号接收器，则图5(a)中线网n_1，n_2虽然无溢出发生，但时延仍有较大优化空间. 因此，本文提出了线网段调整策略，可根据通过2D网格边的各个线网段的下游电容近似表示线网段的时序关键性，有效考虑到与之相关的其他线网的影响，相对文献[32]能从更全局的角度进行时延优化. 线网段调整策略可将时序关键段尽可能调整至低时延层，即将图5(a)方案转换为图5(d)方案，以便于下一阶段的溢出检测以及拆线重绕.

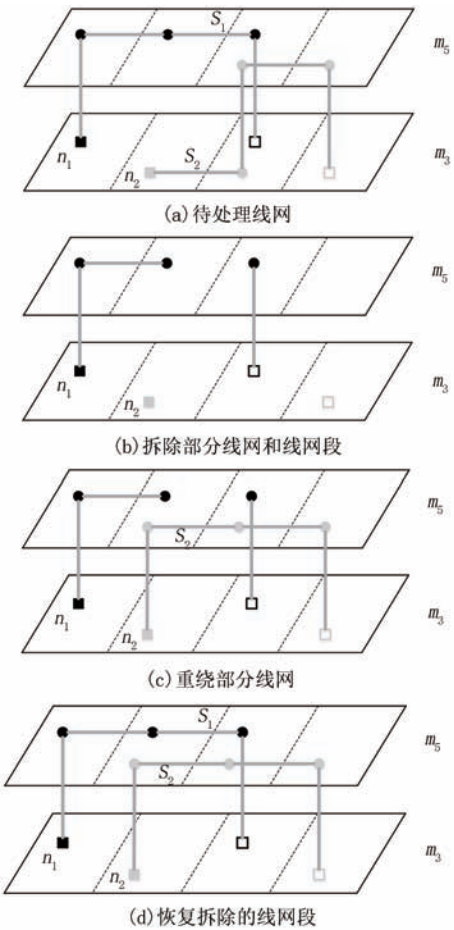

图5　线网段调整策略实例

算法1是线网段调整策略的伪代码. 首先，针对2D网格图中的每一条边，行2至行5为其建立一个存储库，并将通过该边的各层线网的下游电容数值存放在存储库中. 行6将存储库中的下游电容数值按照从大到小的顺序进行排序. 行7至行20对每一条线网进行拆除重绕. 每次重绕前将所有下游电容更小但所在布线层更高的线网段进行拆除，重绕后行15至行19将所拆除的线网段进行恢复. 通着不断调整时序关键段所在布线层，最终遍历各边. 线网段调整策略，一方面通过提升时序关键段所在布线层优化线网时延，另一方面有利于后阶段的拆线重绕对线网溢出的检测和时延优化.

算法1.　线网段调整策略

输入：3D网格图，已有层分配方案

输出：调整后的层分配方案

1. FOR E中的每一个2D网格边e, DO
2.　　建立存储库W

```
3.    IF 在 e 对应的 3D 网格边 e_k 上存在线网段 s_k THEN
4.        计算 s_k 的下游电容 c_d 及所在布线层 k，存储于 W 中
5.    END IF
6.    将 W 中数值按下游电容降序排列
7.    FOR W 中的每一个数值 c_k0 DO
8.        拆除 c_k0 对应线网 n
9.        FOR W 中除 c_k0 外的每一个数值 c_ki DO
10.           IF c_ki < c_k0 且 ki > k0 THEN
11.               拆除 c_ki 所对应线网段 s_ki
12.           END IF
13.       END FOR
14.       在考虑拥塞约束的情况下，对线网 n 进行单线网层分配操作
15.       FOR W 中除 c_k0 外的每一个数值 c_ki DO
16.           IF c_ki < c_k0 且 ki > k0 THEN
17.               恢复 c_ki 所对应线网段 s_ki
18.           END IF
19.       END FOR
20.   END FOR
21. END FOR
```

结合图5说明该策略的基本步骤. 遍历各条存在线网分布的2D网格边，将线网 n_1 和 n_2 中段 S_1，S_2 的下游电容进行统计并存放在库中，并对库中数值按照降序排序. 因为靠近信号发射器的线网段 S_2 的时序关键性强于 S_1，所以线网段 S_2 的下游电容最大，先将 S_2 所在线网 n_2 拆除，且将同一2D网格边上的下游电容较小且位于上层的线网段 S_1 也一并拆除得到图5(b). 然后重新对线网 n_2 进行层分配，则线网 n_2 可充分利用 S_1 拆除后留下的剩余轨道，而在保证可布线性的同时得到图5(c)所示的方案. 最后将拆除段 S_1 进行还原，即可得到图5(d)的方案，其所对应的初始层分配的时延优化效果更好.

3.2.4 线网段时延优化策略

为了确保较好的拆线重绕效果，配合线网段调整策略所产生的解方案，本文提出了线网段时延优化策略. 以线网为单位进行拆线重绕，但在每次查找溢出的过程中，每一条溢出边中线网段所处线网都进行一次拆线重绕，因此对于经过多条溢出边的线网，层分配器将针对性地对其进行多次的重绕过程，以保证该线网中各线网段均满足拥塞约束. 该算法的算法流程图如图6所示.

该算法的具体流程如下：

(1) 根据2D布线结果遍历各边. 对每一条2D网格边均建立一个存储库. 若该边所对应的3D网格图上存在溢出，则统计各层所通过线网的下游电容，存放在库中. 若遍历2D网格边未发现溢出，则结束线网段时延优化过程.

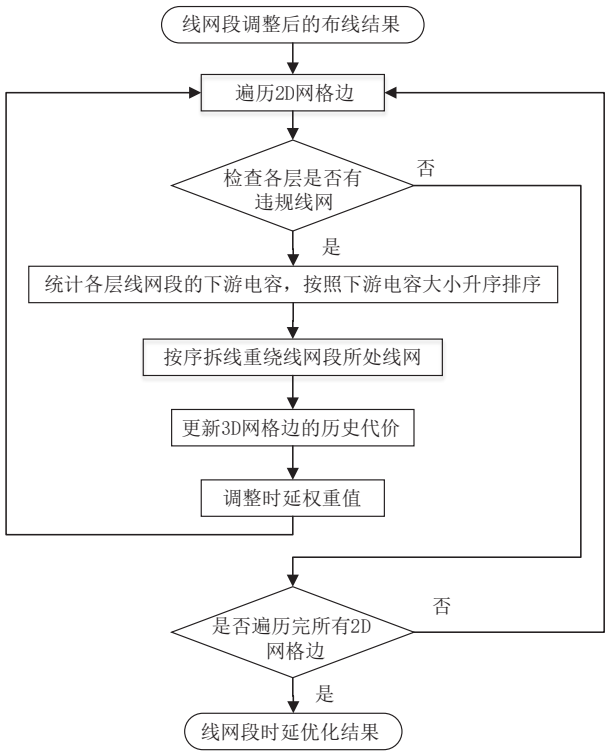

图6 线网段时延优化策略的流程图

(2) 对库中所有下游电容值按升序排序，并按序对线网段所在的线网进行拆线重绕. 因为在线网段调整策略中层分配器已使时序关键段占据上层可分配轨道，所以照此顺序可将与时序关键段竞争的线网段分配至其他布线层.

(3) 若重新绕线后仍存在溢出，则说明现有的拥塞代价不足以令线网规避该3D网格边. 因此，在下一次迭代中增加历史代价和时延权重. 前者增强层分配器规避溢出边的能力，后者引导线网探索时延优化程度更高的布线层.

图7进一步说明线网段时延优化策略的思想. 图7展示了处于不同梯度的两层布线区域. 上层边的可用轨道容量为1，下层边可用轨道容量为2. 图8(a)中线网 n_1，n_2，n_3 在2D网格边 e_1 和 e_2 处均产生溢出，需要对线网进行拆线重绕以消除溢出. 依据线网段时延优化策略，层分配器首先统计图7(a)中边 e_1 所通过线网 n_1，n_2，并且按照线网段 S_1，S_2 的下游电容递增顺序先后对线网 n_2，n_1 进行拆除重绕，得到图7(c). 然后，对图7(c)中通过2D网格边 e_2 的线网 n_1，n_3 再进行相同操作. 通过多次遍历后所有线网段均满足拥塞约束，得到图7(d)方案. 而

若依据现有工作的线网时延递减顺序的拆线重绕方法，则只对线网n_1进行一次拆线重绕过程，得到图7(b). 很明显，相比于图7(b)方案，使用线网段时延优化策略后的图7(d)方案在时延和通孔数量上均取得更优的结果.

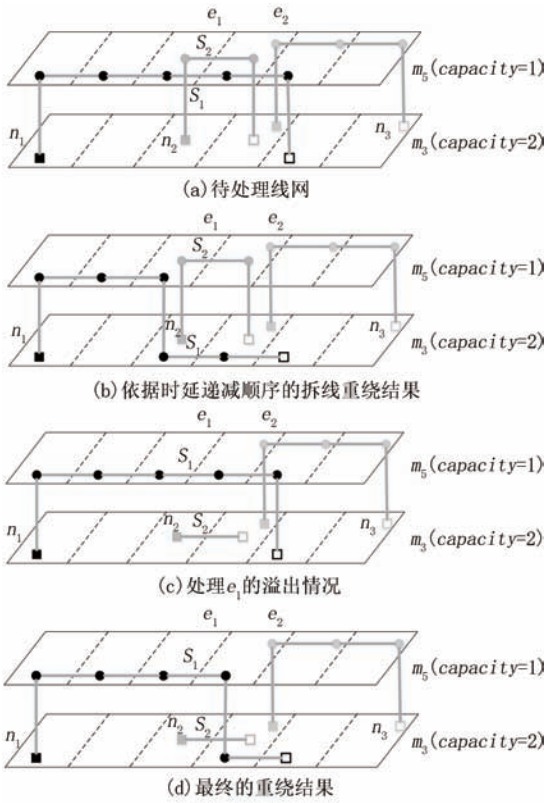

图7　线网段时延优化策略实例

3.3　算法时间复杂度分析

定理2.　假设电路中单一线网对应的布线树，其顶点数量为v，2D网格边数量为q. 布线空间中待分配线网总数为N，2D网格边对应的最大线网数为T，布线金属层数量为k，则本文算法的时间复杂度为$O(N\times\log(N)+N\times q\times T\times(\log(T)+k^5)))$. 证明过程详见附录1.

4　实验结果

本文算法已用C++语言进行实现，实验平台配备的Linux工作站使用3.5 GHz英特尔至强处理器和128 GB内存. 为了体现实验结果的公平性，本文算法与使用非默认规则线技术的文献[32]算法和文献[33]算法均运行在相同的实验环境，采用了相同的DAC12可布线性驱动的标准测试电路[34]. DAC12标准测试电路数据源于国际设计自动化顶级会议比赛数据，其所有设计均源自现代工业专用集成电路. 每个标准测试电路均有多层布线结构，但尚未给定具体的非默认规则线类型. 因此，根据2.3节关于非默认规则线技术这一先进制程技术的说明，在标准测试电路的导线规则库中添加了非默认规则线类型，以形成本文所求解的非默认规则线技术下层分配问题的有效测试电路集合.

此外，本实验中，总时延、最大时延、关键时序线网时延的单位均为皮秒. 本文所有测试电路的布局方案均由NTUplace4[35]产生，2D总体布线方案由NCTU-GR2.0[36]生成. 表1所示的是测试电路的特征数据. 由于NCTU-GR 2.0[36]得到的2D布线结果没有产生溢出，所以实验中由2D拓扑得到的3D布线方案也保证满足拥塞约束.

表1　测试电路特征

测试电路	网格数	线网数	层数
sp2	770Í891	990899	9
sp3	800Í415	898001	9
sp6	649Í495	1006629	9
sp7	499Í713	1340418	9
sp9	426Í570	833808	9
sp11	631Í878	935731	9
sp12	444Í518	1293436	9
sp14	406Í473	619815	9
sp16	465Í404	697458	9
sp19	321Í518	511685	9

4.1　轨道数感知的层选择策略的有效性验证

为了验证本文所提出的布线层选择策略的有效性，将轨道数感知的层选择策略与使用非默认规则线技术的文献[32]算法进行对比. “TD”和“MD”分别表示线网的总时延和最大时延，“#vc”表示通孔数量. “0.5%”、“1.0%”和“5.0%”分别对应按线网时延降序排序下测试电路中前0.5%、前1.0%和前5.0%时序关键线网的平均时延. 由表2可知，采用轨道数感知的层选择策略相比于使用非默认规则线技术的同类工作DLA[32]在总时延、最大时延、前0.5%时延、前1.0%时延、前5.0%时延和通孔数的优化力度分别达到9.0%、3.2%、8.6%、10.6%、13.7%和3.4%. 实验结果表明，采用轨道数感知的层选择策略相比使用非默认规则线的文献[32]算法在时延优化层面具备更强的优势.

4.2　多指标驱动初始线网排序策略的有效性验证

在层分配的过程中，初始线网的层分配情况

将影响后期线网对布线资源的利用.在先前工作[32]中初始时刻将所有线网不计溢出代价分配在最高层,那么将会产生过多的需要拆线重绕的复杂情况,而后期的重绕过程不可能完全将所有线网分配在最合适的金属层,因此不利于最终的时延优化.为验证本文所提出的多指标驱动初始线网排序策略的有效性,本节实验在使用轨道数感知的层选择策略的基础上增加多指标驱动初始线网排序策略,比较初始布线顺序的改变对最终实验结果的影响.

表2 文献[32]算法和轨道数感知的层选择策略实验结果对比

测试	文献[32]算法						使用轨道数量感知策略					
电路	TD	MD	0.5%	1.0%	5.0%	#vc	TD	MD	0.5%	1.0%	5.0%	#vc
sp2	2222970	503.9	145.0	116.2	51.5	7129655	2122190	441.4	140.8	112.9	49.2	6899729
sp3	775357	535.1	89.1	66.2	23.2	6382992	716557	565.5	83.2	61.7	20.7	6160494
sp6	654807	253.7	57.3	44.0	16.9	6202333	592662	225.9	51.7	39.3	14.6	5993626
sp7	624896	233.2	43.1	31.4	10.9	9335272	560782	188.3	38.2	26.9	9.0	9053376
sp9	413723	211.5	46.0	34.2	13.1	5120143	375839	182.7	42.9	30.6	11.2	4948051
sp11	750541	1602.3	109.7	68.2	19.8	5592279	697348	1514.8	104.8	63.3	17.7	5411262
sp12	542782	581.5	40.2	29.9	10.1	8500960	488656	542.6	36.7	26.3	8.7	8265876
sp14	407862	205.0	56.0	41.9	15.2	4038943	364288	292.1	50.8	36.7	12.8	3909247
sp16	544414	186.3	54.4	42.9	18.9	4122687	485414	176.1	49.8	38.8	16.1	3914500
sp19	193844	304.5	24.6	19.0	8.7	3038941	173078	286.6	20.2	15.3	6.9	2937556
AVG	713120	461.7	66.5	49.4	18.8	5946421	657681	441.6	61.9	45.2	16.7	5749372
ratio	1.000	1.000	1.000	1.000	1.000	1.000	**0.910**	**0.968**	**0.914**	**0.894**	**0.863**	**0.966**

如表3所示,本文将轨道数感知的层选择策略应用到时延驱动层分配过程,并分别采用添加多指标驱动初始线网排序策略前后进行实验结果对比.由于多指标驱动初始线网排序策略在初始时刻考虑了拥塞的影响,同时综合考虑了线网线长、信号接收器数以及布线空间密度,为每一个线网计算分数,分数越高的线网越先布线,使得布线顺序更合理.因此添加多指标驱动初始线网排序策略后的层分配方案在各项性能指标上均有不同程度的优化.其中,引入多指标驱动初始线网排序策略后的层分配器对总时延、前0.5%时延、前1.0%时延、前5.0%时延和通孔数的优化率分别达到0.8%、4.1%、3.9%、0.7%和0.2%.且对线网最大时延的优化效果最明显,达到10.5%.

表3 添加多指标驱动初始线网排序策略前后的实验结果对比

测试	未添加多指标驱动初始线网排序策略						添加多指标驱动初始线网排序策略					
电路	TD	MD	0.5%	1.0%	5.0%	#vc	TD	MD	0.5%	1.0%	5.0%	#vc
sp2	2122190	441.4	140.8	112.9	49.2	6899729	2131210	420.8	137.4	111.6	49.0	6815225
sp3	716557	565.5	83.2	61.7	20.7	6160494	710498	444.2	81.2	60.6	20.6	6138531
sp6	592662	225.9	51.7	39.3	14.6	5993626	587249	208.7	48.8	37.2	14.5	5952285
sp7	560782	188.3	38.2	26.9	9.0	9053376	551804	168.0	36.6	25.9	8.9	9083303
sp9	375839	182.7	42.9	30.6	11.2	4948051	374008	139.2	41.9	29.7	11.2	4969299
sp11	697348	1514.8	104.8	63.3	17.7	5411262	696044	1466.2	104.4	62.6	17.7	5460008
sp12	488656	542.6	36.7	26.3	8.7	8265876	483488	573.4	35.4	25.3	8.7	8268086
sp14	364288	292.1	50.8	36.7	12.8	3909247	361087	202.3	47.8	34.4	12.7	3897885
sp16	485414	176.1	49.8	38.8	16.1	3914500	480910	153.9	46.8	36.2	15.8	3855073
sp19	173078	286.6	20.2	15.3	6.9	2937556	171197	298.3	18.5	14.4	6.8	2937368
AVG	657681	441.6	61.9	45.2	16.7	5749372	654750	407.5	59.9	43.8	16.6	5737706
ratio	1.000	1.000	1.000	1.000	1.000	1.000	**0.992**	**0.895**	**0.959**	**0.961**	**0.993**	**0.998**

4.3 线网段调整策略和线网段时延优化策略的有效性验证

线网段调整策略能够将时序关键段所在布线层进行有效的调整,而线网段时延优化策略能够在进一步优化线网时延的同时消除线网溢出.表4展示的是在时延驱动层分配上同时添加线网段调整策略和线网段时延优化策略前后的实验结果对比.从表4中可以看出,添加线网段调整策略和线网段时延优化

策略后的层分配方案在总时延、最大时延、前0.5%时延、前1.0%时延、前5.0%时延以及通孔数上分别优化了2.6%、0.7%、2.4%、2.9%、3.6%和2.6%.

4.4　同类算法的实验结果对比

为了验证基于多策略的时延驱动层分配算法的有效性，本文拿到了同样使用非默认规则线的文献[32]算法和文献[33]算法的源代码后，与本文算法进行了对比. 从表5可以看出，相比于文献[32]算法，虽然付出了少量的运行时间代价，但本文算法可在总时延、最大时延、前0.5%、前1.0%、前5.0%线网平均时延以及通孔数上分别优化12.1%、15.4%、14.3%、16.4%、17.2%和6.2%. 相比于文献[33]算法，基于多策略的时延驱动层分配算法可在总时延、最大时延、前0.5%、前1.0%、前5.0%线网平均时延以及通孔数上分别优化3.3%、0%、5.2%、6.3%、5.7%和2.7%. 基于多策略的时延驱动层分配算法的时延优化效果更优，并且能够同时优化通孔数.

表4　添加线网段调整策略和线网段时延优化策略前后实验结果对比

测试	未添加线网段调整策略和线网段时延优化策略						本文算法					
电路	TD	MD	0.5%	1.0%	5.0%	#vc	TD	MD	0.5%	1.0%	5.0%	#vc
sp2	2131210	420.8	137.4	111.6	49.0	6815225	2107820	439.4	137.8	111.6	48.6	6562343
sp3	710498	444.2	81.2	60.6	20.6	6138531	694919	447.7	79.7	59.2	20.0	5965551
sp6	587249	208.7	48.8	37.2	14.5	5952285	572051	208.1	47.8	36.3	14.0	5799693
sp7	551804	168.0	36.6	25.9	8.9	9083303	535554	189.8	35.4	24.8	8.5	8895986
sp9	374008	139.2	41.9	29.7	11.2	4969299	363185	141.7	40.9	28.8	10.8	4858606
sp11	696044	1466.2	104.4	62.6	17.7	5460008	679591	1402.1	103.3	61.6	17.2	5332042
sp12	483488	573.4	35.4	25.3	8.7	8268086	469263	545.0	34.4	24.3	8.3	8081012
sp14	361087	202.3	47.8	34.4	12.7	3897885	350364	197.0	46.7	33.4	12.2	3794123
sp16	480910	153.9	46.8	36.2	15.8	3855073	468682	140.4	45.3	35.0	15.3	3720892
sp19	171197	298.3	18.5	14.4	6.8	2937368	164841	281.1	17.6	13.6	6.4	2871955
AVG	654750	407.5	59.9	43.8	16.6	5737706	640627	399.2	58.9	42.9	16.1	5588220
ratio	1.000	1.000	1.000	1.000	1.000	1.000	**0.974**	**0.993**	**0.976**	**0.971**	**0.964**	**0.974**

表5　文献[32]算法与本文算法实验结果对比

测试	文献[32]算法							本文算法						
电路	TD	MD	0.5%	1.0%	5.0%	#vc	Runtime	TD	MD	0.5%	1.0%	5.0%	#vc	Runtime
sp2	2222970	503.9	145.0	116.2	51.5	7129655	1025.9	2107820	439.4	137.8	111.6	48.6	6562343	1349.4
sp3	775357	535.1	89.1	66.2	23.2	6382992	671.6	694919	447.7	79.7	59.2	20.0	5965551	836.7
sp6	654807	253.7	57.3	44.0	16.9	6202333	572.2	572051	208.1	47.8	36.3	14.0	5799693	742.0
sp7	624896	233.2	43.1	31.4	10.9	9335272	725.3	535554	189.8	35.4	24.8	8.5	8895986	919.8
sp9	413723	211.5	46.0	34.2	13.1	5120143	426.9	363185	141.7	40.9	28.8	10.8	4858606	578.7
sp11	750541	1602.3	109.7	68.2	19.8	5592279	462.2	679591	1402.1	103.3	61.6	17.2	5332042	649.7
sp12	542782	581.5	40.2	29.9	10.1	8500960	678.6	469263	545.0	34.4	24.3	8.3	8081012	878.8
sp14	407862	205.0	56.0	41.9	15.2	4038943	348.6	350364	197.0	46.7	33.4	12.2	3794123	474.4
sp16	544414	186.3	54.4	42.9	18.9	4122687	448.2	468682	140.4	45.3	35.0	15.3	3720892	560.7
sp19	193844	304.5	24.6	19.0	8.7	3038941	233.4	164841	281.1	17.6	13.6	6.4	2871955	313.1
AVG	713120	461.7	66.5	49.4	18.8	5946421	559.3	640627	399.2	58.9	42.9	16.1	5588220	730.3
ratio	1.000	1.000	1.000	1.000	1.000	1.000	1.000	**0.879**	**0.846**	**0.857**	**0.836**	**0.828**	**0.938**	1.314

为了进一步验证本文算法的有效性，本文复现了基于文献[24,37]的层分配算法. 其中，权重参数或阈值参数的设定均严格按照相关文献中所提及的数值进行设置. 如表7所示，本文将实验结果与引入文献[24]中相关时延优化策略后进行对比. 相比于文献[24]算法，本文算法在总时延、最大时延、前0.5%、前1.0%、前5.0%线网平均时延以及通孔数上的优化率分别达到了10.9%、11.3%、11.5%、13.7%、15.7%和6.7%. 而表8中，本文将所提算法与引入层移动和拆除过程的文献[37]算法实验结果进行对比，其在总时延、最大时延、前0.5%、前1.0%、前5.0%线网平均时延以及通孔数上优化率分别达到了8.4%、14.7%、9.9%、11.4%、12.2%和3.5%.

表6 文献[33]算法和本文算法实验结果对比

测试电路	文献[33]算法							本文算法						
	TD	MD	0.5%	1.0%	5.0%	#vc	Runtime	TD	MD	0.5%	1.0%	5.0%	#vc	Runtime
sp2	2139820	475.2	138.3	111.9	49.7	6857552	926.4	2107820	439.4	137.8	111.6	48.6	6562343	1349.4
sp3	718602	368.0	83.2	62.2	21.1	6186023	653.1	694919	447.7	79.7	59.2	20.0	5965551	836.7
sp6	592284	203.9	50.9	38.9	14.9	5945350	491.2	572051	208.1	47.8	36.3	14.0	5799693	742.0
sp7	558499	187.8	38.6	27.6	9.2	9134247	649.7	535554	189.8	35.4	24.8	8.5	8895986	919.8
sp9	376370	140.7	42.8	30.9	11.5	4990615	395.6	363185	141.7	40.9	28.8	10.8	4858606	578.7
sp11	695731	1596.0	104.1	63.0	17.8	5388497	589.8	679591	1402.1	103.3	61.6	17.2	5332042	649.7
sp12	487128	536.0	36.2	26.0	8.8	8283491	581.2	469263	545.0	34.4	24.3	8.3	8081012	878.8
sp14	363108	192.1	49.5	36.3	13.1	3907452	309.3	350364	197.0	46.7	33.4	12.2	3794123	474.4
sp16	485113	155.4	46.7	36.2	16.0	3796930	647.2	468682	140.4	45.3	35.0	15.3	3720892	560.7
sp19	172812	281.0	20.4	15.8	7.1	2951412	190.9	164841	281.1	17.6	13.6	6.4	2871955	313.1
AVG	658947	413.6	61.1	44.9	16.9	5744157	543.4	640627	399.2	58.9	42.9	16.1	5588220	730.3
ratio	1.000	1.000	1.000	1.000	1.000	1.000	1.000	**0.967**	**1.000**	**0.948**	**0.937**	**0.943**	**0.973**	1.378

表7 文献[24]算法和本文算法实验结果对比

测试电路	文献[24]算法						本文算法					
	TD	MD	0.5%	1.0%	5.0%	#vc	TD	MD	0.5%	1.0%	5.0%	#vc
sp2	2218430	462.2	144.3	115.7	51.3	7133201	2107820	439.4	137.8	111.6	48.6	6562343
sp3	772878	552.3	88.5	65.7	23.1	6385013	694919	447.7	79.7	59.2	20.0	5965551
sp6	618865	219.2	51.3	38.9	15.7	6326157	572051	208.1	47.8	36.3	14.0	5799693
sp7	621769	205.7	42.5	31.1	10.8	9352697	535554	189.8	35.4	24.8	8.5	8895986
sp9	411971	219.7	45.6	33.8	13.0	5130406	363185	141.7	40.9	28.8	10.8	4858606
sp11	747512	1510.8	109.0	67.7	19.7	5600672	679591	1402.1	103.3	61.6	17.2	5332042
sp12	535737	575.7	39.0	29.0	10.0	8568340	469263	545.0	34.4	24.3	8.3	8081012
sp14	397619	207.8	52.7	39.4	14.7	4063943	350364	197.0	46.7	33.4	12.2	3794123
sp16	543261	170.4	53.9	42.6	18.8	4121769	468682	140.4	45.3	35.0	15.3	3720892
sp19	189768	295.9	22.7	17.9	8.4	3092748	164841	281.1	17.6	13.6	6.4	2871955
AVG	705781	442.0	64.9	48.2	18.5	5977495	640627	399.2	58.9	42.9	16.1	5588220
ratio	1.000	1.000	1.000	1.000	1.000	1.000	**0.891**	**0.887**	**0.885**	**0.863**	**0.843**	**0.933**

表8 文献[37]算法和本文算法实验结果对比

测试电路	文献[37]算法						本文算法					
	TD	MD	0.5%	1.0%	5.0%	#vc	TD	MD	0.5%	1.0%	5.0%	#vc
sp2	2176680	422.8	143.0	114.1	50.2	6863534	2107820	439.4	137.8	111.6	48.6	6562343
sp3	748698	563.5	85.3	63.2	22.2	6221563	694919	447.7	79.7	59.2	20.0	5965551
sp6	632612	295.0	54.7	41.7	16.1	6038170	572051	208.1	47.8	36.3	14.0	5799693
sp7	592246	205.5	39.5	28.5	10.0	9118129	535554	189.8	35.4	24.8	8.5	8895986
sp9	400062	189.4	44.4	32.6	12.4	5002066	363185	141.7	40.9	28.8	10.8	4858606
sp11	722414	1571.3	106.2	65.0	18.8	5457598	679591	1402.1	103.3	61.6	17.2	5332042
sp12	514793	584.6	38.0	28.0	9.5	8316707	469263	545.0	34.4	24.3	8.3	8081012
sp14	390125	208.2	52.7	39.0	14.3	3942345	350364	197.0	46.7	33.4	12.2	3794123
sp16	514912	219.4	50.8	39.8	17.5	3918339	468682	140.4	45.3	35.0	15.3	3720892
sp19	185390	308.9	22.9	17.7	8.0	2958072	164841	281.1	17.6	13.6	6.4	2871955
AVG	687793	456.9	63.8	46.9	17.9	5783652	640627	399.2	58.9	42.9	16.1	5588220
ratio	1.000	1.000	1.000	1.000	1.000	1.000	**0.916**	**0.853**	**0.901**	**0.886**	**0.878**	**0.965**

5 总 结

本文针对时延和通孔数两个优化目标提出了基于多策略的时延驱动层分配算法. 该算法主要由三个阶段构成：(1) 通过均衡排序初始化得到较好的初始层分配方案；(2) 在基于下游电容的时延优化层分配阶段，层分配器通过线网段调整策略调整时序关键段所处布线层，继而通过线网段时延优化策略，在优化线网时延的同时消除溢出；(3) 后优化阶段中算法在满足拥塞约束的条件下，对线网进行重新拆线分配，选择更好的层分配方案. 其中每一阶段对单个线网层分配过程中都使用轨道数感知的层选择策略，从而可充分利用上层轨道资源. 最终的实验结果表明本文所提出的多个策略的有效性. 相比于现有工作，本文算法能够在保证不产生溢出的情况下，在层分配问题中最重要的两个优化目标——时延和通孔数量均取得明显优化.

参 考 文 献

[1] Chen H C J, Standaert T E, Alptekin E, Spooner T A, Paruchuri V. Interconnect performance and scaling strategy at 7 nm node//Proceedings of the IEEE International Interconnect Technology Conference. San Hose, USA, 2014: 93-96

[2] Alpert C J, Li Z, Moffitt M D, Nam G J, Roy J A, Tellez G. What makes a design difficult to route//Proceedings of the 19th International Symposium on Physical Design. San Francisco, USA, 2010: 7-12

[3] Tan Hai, He Yue Shun, Jin Wen-Bing, Su Yan. V-Mesh: A low-latency and energy-efficient network-on-chip structure for 3D stacked many-core. Chinese Journal of Computers, 2014, 37(010): 2139-2152 (in Chinese)
(谭海，何月顺，靳文兵，苏岩. V-Mesh:面向三维堆叠芯片的低时延低功耗片上网络结构. 计算机学报，2014，37(10)：2139-2152)

[4] 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

[5] 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

[6] Liao H, Zhang W, Dong X, Poczos B, Shimada K, Kara L B. A deep reinforcement learning approach for global routing. Journal of Mechanical Design, 2019, 142(6): 1-17

[7] Zhou Z, Chahal S, Ho T-Y, Ivanov A. Supervised-learning congestion predictor for routability-driven global routing// Proceedings of the International Symposium on VLSI Design. Hsinchu, Taiwan, 2019: 1-4

[8] Zhou Z, Zhu Z, Chen J, Ma Y, Yu B, Ho T Y, Lemieux G, Ivanov A. Congestion-aware global routing using deep convolutional generative adversarial networks//Proceedings of the 1st Workshop on Machine Learning for CAD. Canmore, Canada, 2019: 1-6

[9] Jiang Y J, Fang S Y. COALA: Concurrently assigning wire segments to layers for 2D global routing//Proceedings of the International Conference On Computer Aided Design. San Diego, USA, 2020: 1-8

[10] Latha N R, Prasad G R. Performance and memory efficient parallel computing framework for RSMT construction// Progress in Computing, Analytics and Networking, Singapore, 2020: 369-381

[11] Shyamala G, Prasad G R. An efficient parallel computing framework for over the obstacle VLSI routing//Progress in Computing, Analytics and Networking. Singapore, 2020: 383-395

[12] 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 (in Chinese)
(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)

[13] 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

[14] Xu Y, Zhang Y, Chu C. FastRoute 4.0: Global router with efficient via minimization//Proceedings of the Asia and South Pacific Design Automation Conference. Yokohama, Japan, 2009: 576-581

[15] Chen H Y, Hsu C H, Chang Y W. High-performance global routing with fast overflow reduction//Proceedings of the Asia and South Pacific Design Automation Conference. Yokohama, Japan, 2009: 582-587

[16] 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

[17] Li Chen, Ma Sheng, Wang Lu, Guo Yang. A survey on architecture for three-dimensional networks-on-chip. Chinese Journal of Computers, 2016, 39(9):1812-1828 (in Chinese)
(李晨，马胜，王璐，郭阳. 三维片上网络体系结构研究综述. 计算机学报，2016，39(9)：1812-1828)

[18] Chu C C N, Wong D F. An efficient and optimal algorithm for simultaneous buffer and wire sizing. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 1999, 18(9): 1297-1304

[19] Liu Geng-Geng, Huang Yi-Fei, Wang Xin, Guo Wen-Zhong, Chen Guo-Long. Hybrid discrete particle swarm optimization algorithm for X-architecture Steiner minimal tree construction with slew constraints. Chinese Journal of Computers, 2021, 44(12): 2542-2559 (in Chinese)

(刘耿耿，黄逸飞，王鑫，郭文忠，陈国龙. 基于混合离散粒子群优化的Slew约束下X结构Steiner最小树算法. 计算机学报，2021，44(12)：2542-2559)

[20] Xu N，Hong X-L. Very large scale integration physical design theory and method. Beijing：Tsinghua University Press，2009 (in Chinese)
(徐宁，洪先龙. 超大规模集成电路物理设计理论与方法. 北京：清华大学出版社，2009)

[21] Cho M，Pan D Z. BoxRouter：A new global router based on box expansion and progressive ILP. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems，2007，26(12)：2130-2143

[22] Liu W H，Li Y L. Negotiation-based layer assignment for via count and via overflow minimization//Proceedings of the Asia and South Pacific Design Automation Conference. Yokohama，Japan，2011：539-544

[23] Lee T H，Wang T C. Congestion-constrained layer assignment for via minimization in global routing. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems，2008，27(9)：1643-1656

[24] Ao J，Dong S，Chen S，Goto S. Delay-driven layer assignment in global routing under multi-tier interconnect structure//Proceedings of the International Symposium on Physical Design. Stateline. Nevada，USA，2013：101-107

[25] Yu B，Liu D，Chowdhury S，Pan D Z. TILA：Timing-driven incremental layer assignment//Proceedings of the International Conference on Computer-Aided Design. Austin，USA，2015：110-117

[26] Liu D，Yu B，Chowdhury S，Pan D Z. TILA-S：timing-driven incremental layer assignment avoiding slew violations. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems，2018，37(1)：231-244

[27] Livramento V，Liu D，Chowdhury S，Yu B，Xu X，Pan D Z，Güntzel J L，Santos L C V. Incremental layer assignment driven by an external signoff timing engine. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems，2017. 36(7)：1126-1139

[28] Liu D，Yu B，Chowdhury S，Pan D Z. Incremental layer assignment for timing optimization. ACM Transactions on Design Automation of Electronic Systems，2017，22(4)：1-25

[29] Shi D，Tashjian E，Davoodi A. Dynamic planning of local congestion from varying-size vias for global routing layer assignment. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems，2017，36(8)：1301-1312

[30] Liu J，Pui C W，Wang F，Young E F Y，CUGR：detailed-routability-driven 3D global routing with probabilistic resource model//Proceedings of the Design Automation Conference. San Francisco，USA，2020：1-6

[31] Ewetz R，Liu W H，Chao K Y，Wang T C，Koh C K. A study on the use of parallel wiring techniques for sub-20nm designs//Proceedings of the Great Lakes Symposium on VLSI. Houston，USA，2014：129-134

[32] Han S Y，Liu W H，Ewetz R，Koh C K，Wang T C. Delay-driven layer assignment for advanced technology nodes//Proceedings of the Asia and South Pacific Design Automation Conference. Chiba，Japan，2017：456-462

[33] Zhang X，Zhuang Z，Liu G，Huang X，Liu W H，Guo W，Wang T C. MiniDelay：Multi-strategy timing-aware layer assignment for advanced technology nodes//Proceedings of the Design，Automation & Test in Europe Conference & Exhibition. Grenoble，France，2020：586-591

[34] Viswanathan N，Alpert C，Sze C，Li Z，Wei Y. The DAC 2012 routability-driven placement contest and benchmark suite//Proceedings of the Design Automation Conference. San Francisco，USA，2012：774-782

[35] Hsu M K，Chen Y F，Huang C C，Chen T C，Chang Y W. Routability-driven placement for hierarchical mixed-size circuit designs//Proceedings of the Design Automation Conference. Austin，USA，2013：1-6

[36] Liu W H，Kao W C，Li Y L，Chao K Y. NCTU-GR 2.0：Multithreaded 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

[37] Chen J W，Di Z X，Chen J J，Feng Q Y，Shi J Y. An enhanced heuristic layer assignment method in global routing//Proceedings of the International Conference on Solid-State & Integrated Circuit Technology. Kunming，China 2020：1-3

附录1.

定理1. 任意一个2D网格边e_i，其对应的任意不同层的3D网格边$e_{i,j}$和$e_{i,l}$，若满足条件$use(e_{i,j})/tc(e_{i,j})>use(e_{i,l})/tc(e_{i,l})$，则并不能得到$cap(e_{i,j})<cap(e_{i,l})$.

证明. 因为在任意一条3D网格边$e_{i,k}$中均有$tc(e_{i,k})=use(e_{i,k})+cap(e_{i,k})$. 将其带入已知条件$use(e_{i,j})/tc(e_{i,j})>use(e_{i,l})/tc(e_{i,l})$，可得到不等式：

$$use(e_{i,j})/(use(e_{i,j})+cap(e_{i,j}))>use(e_{i,l})/(use(e_{i,l})+cap(e_{i,l})).$$

不等式两边同时减去$use(e_{i,l})/(use(e_{i,l})+cap(e_{i,l}))$，可得：

$$(use(e_{i,j})\times cap(e_{i,l})-use(e_{i,l})\times cap(e_{i,j}))/((use(e_{i,j})+cap(e_{i,j}))\times(use(e_{i,l})+cap(e_{i,l})))>0,$$

由该不等式在分母大于0的情况下易知分子必定大于0. 但由于$use(e_{i,j})$与$use(e_{i,l})$的大小关系未定，并不能得到$cap(e_{i,j})<cap(e_{i,l})$. 因此定理1得证. 证毕.

定理2. 假设单一线网对应的布线树顶点数量为v，2D网格边数量为q. 假设布线空间中待分配线网总数为N，2D网格边对应的最大线网数为T，布线金属层数量为k，则基于多策略的时延驱动层分配算法的时间复杂度为$O(N\times\log(N)+N\times q\times T\times(\log(T)+k^5)))$.

证明. 基于多策略的时延驱动层分配算法主要包括均衡排序初始化、基于下游电容的时延优化层分配和后优化三个阶段. 其中，单一线网层分配算法作为各步骤分配单一线网的基础算法. 该算法基于动态规划思想遍历2D线网所对应的所有可能的层分配方案：首先采取自顶向下的顺序通过

广度优先搜索遍历至线网信号接收器，时间复杂度为$O(v+q)$；然后采取自底向上的方式对各顶点遍历可选布线层及导线类型．引入非默认规则线技术后任一3D网格边可选导线类型数量为3，同时一个顶点最多含4个子节点，即任一顶点遍历方案的最大时间复杂度为$O(k^4\times3^4\times k\times3)\approx O(k^5)$；最后选择各顶点最小目标代价的层分配结果需要时间$O(3\times k\times v)$．即单一线网层分配算法的时间复杂度是$O(v+q+k^5\times v+3\times k\times v)$．因为在本文中$q\approx v$，时间复杂度简化为$O(k^5\times q)$．

本文算法各步骤均基于单一线网层分配依次分配线网．均衡排序初始化阶段采用多指标驱动初始线网排序策略前，若线网压缩后2D网格边数量为Q，需统计2D线网于各布线层的资源信息，其复杂度为$O(Q\times k)$．排序策略在此基础上结合线网特征信息确定线网优先级，所需时间$O(N\times\log(N))$．后依次分配各线网．因此算法在该阶段的时间复杂度为$O(Q\times k+N\times\log(N)+N\times k^5\times q)$，由于$Q\leqslant N\times q$，简化后得$O(N\times(\log(N)+k^5\times q))$．

基于下游电容的时延优化层分配阶段由线网段调整和线网段时延优化两部分策略构成．线网段调整策略旨在为不合理分布的线网段重新选择布线层，需预先排序2D网格边所对应线网段，拆除和恢复部分段，并拆线重绕部分线网．其中，假设2D网格边对应的最大线网数为T，则排序过程所需$O(Q\times T\times\log(T))$．由于拆除和恢复线网段数量最大不超过线网段总数$N\times q$，所需最大时间$O(2\times N\times q)$．而拆线重绕的线网数量最大为总线网数$N$，结合单一线网层分配，最大时间复杂度为$O(N\times q+N\times q\times k^5)$．综上可得线网段调整策略的时间复杂度$O(Q\times T\times\log(T)+2\times N\times q+N\times q+N\times q\times k^5)\approx O(N\times q\times(k^5+T\times\log(T)))$．

线网段时延优化策略排序线网段并拆线重绕溢出线网．与上一策略的分析相类似，排序过程时间复杂度为$O(Q\times T\times\log(T))$．拆线重绕的迭代次数不超过最大线网数$T$，即时间复杂度$O(T\times(N\times q+N\times q\times k^5))$．综合两部分过程可得线网段时延优化策略的时间复杂度$O(N\times q\times(T\times\log(T)+T+T\times k^5))\approx O(N\times q\times T\times(\log(T)+k^5))$．综合两项策略，算法第二阶段的时间复杂度为$O(N\times q\times T\times(\log(T)+k^5))$．

后优化阶段只遍历一次全局线网，若重绕后时延未能得到优化则恢复原线网．假设所有线网段均经历拆线、重绕、拆线和恢复四个步骤．其中，拆线和恢复所有线网段的时间复杂度均为$O(N\times q)$，因此后优化阶段的最大时间复杂度为$O(3\times N\times q+N\times q\times k^5)$．

综合本文算法三个阶段，算法时间复杂度为$O(N\times\log(N)+N\times q\times T\times(\log(T)+k^5))$．证毕．

LIU Geng-Geng, Ph. D., associate professor, Ph. D. supervisor. His research interests include EDA algorithm, and computational intelligence and its application.

BAO Chen-Peng, M. S. candidate, His research interests include EDA design algorithm.

WANG Xin, Ph. D., professor, His main research interests include large-scale knowledge processing, and computational intelligence and its application.

GUO Wen-Zhong, Ph. D., professor, Ph. D. supervisor. His research interests include EDA algorithm, and computational intelligence and its application.

CHEN Guo-Long, Ph. D., professor, Ph. D. supervisor. His research interests include EDA algorithm, and computational intelligence and its application.

Background

With the increasing number of integrated circuit components on a chip, the limitation of storage space and packaging technology has posed a new challenge to the design method of VLSI. Some electronic design automation (EDA) tools are difficult to deal with many VLSI design problems with exponential increase in complexity, and lack of consideration for a series of new problems in nano technology. In the current manufacturing process, interconnect delay has surpassed gate delay and become the main factor determining circuit performance. The interconnect delay is mainly optimized in the routing phase. Therefore, in order to optimize the performance of the chip, what we need to consider is not only the optimization of wire length, but also some optimization objectives about performance such as delay.

As a key step in the physical design of very large scale integration, layer assignment plays a very important role in determining the delay of routing solution. In order to optimize the delay in integrated circuits, the existing layer assignment algorithms usually focus on minimizing interconnect delay and via count. However, the existing work either does not consider the allocation of the timing-critical segments in nets, or the time critical representation of wire segments is not reasonable, which ultimately makes the delay optimization not ideal. For

this reason, this paper proposes a multi-strategy delay-driven layer assignment for non-default-rule wiring techniques, which mainly includes the following key strategies: (1) A track number aware layer selection strategy is presented to enhance the ability of layer assigner to select suitable routing layers for wire segments; (2) A multi-index driven initial net sorting strategy is proposed to determine the priority of nets by considering multiple indicators such as the wirelength, the number of sink points, and the resource of tracks; (3) A wire segment adjusting strategy is adopted to optimize the delay by re-assigning nets and assigning timing-critical segments on upper layers; (4) A wire segment delay optimization strategy is proposed to optimize the delay of nets while eliminating overflow by ripping up and re-assigning the nets which have overflow. The experimental results show that the proposed algorithm can achieve the best performance in both delay and via count among the existing algorithms without overflow.

This Research is supported by National Natural Science Foundation of China ("Research on VLSI performance-driven multilayer routing under advanced Via-Pillar technology", 61877010); State Key Laboratory of Computer Architecture (ICT, CAS) ("Research on VLSI performance driver layer assignment algorithm in advanced manufacturing process", CARCHB202014).