

VLSI 中高性能 X 结构多层总体布线器

刘耿耿^{1,2,3} 庄震¹ 郭文忠^{1,2,3} 陈国龙¹

摘要 X 结构带来物理设计诸多性能的提高, 该结构的引入和多层工艺的普及, 使得总体布线算法更复杂. 为此, 在 XGRouter 布线器的基础上, 本文设计了三种有效的加强策略, 包括: 1) 增加新类型的布线方式; 2) 粒子群优化 (Particle swarm optimization, PSO) 算法与基于新布线代价的迷宫布线的结合; 3) 初始阶段中预布线容量的缩减策略, 继而引入了多层布线模型, 简化了 XGRouter 的整数线性规划模型, 最终构建了一种高性能的 X 结构多层总体布线器, 称为 ML-XGRouter. 在标准测试电路的仿真实验结果表明, ML-XGRouter 相对其他各类总体布线器, 在多层总体布线中最重要的优化目标—溢出数和线长总代价两个指标上均取得最佳.

关键词 X 结构, 多层布线, 超大规模集成电路, 总体布线, 粒子群优化

引用格式 刘耿耿, 庄震, 郭文忠, 陈国龙. VLSI 中高性能 X 结构多层总体布线器. 自动化学报, 2020, 46(1): 79–93

DOI 10.16383/j.aas.c170714

A High Performance X-architecture Multilayer Global Router for VLSI

LIU Geng-Geng^{1,2,3} ZHUANG Zhen¹ GUO Wen-Zhong^{1,2,3} CHEN Guo-Long¹

Abstract The introduction of X-architecture can improve many performance standards of the chip in physical design. The proposed X-architecture and pervasive multilayer technology make the global routing problem more complex. For this reason, this paper presents the following enhancements based on XGRouter: 1) the introduction of some new types of routing; 2) the combination of particle swarm optimization (PSO) algorithm and maze routing with new routing cost; 3) a reduction strategy of routing capacity in the initial stage. Then the multilayer routing model is introduced and the integer linear programming model of XGRouter is simplified. Finally, a high performance X-architecture multilayer global router, namely ML-XGRouter, is proposed. The experimental results on benchmark circuits have shown that our proposed ML-XGRouter is effective and superior to state-of-the-art multilayer routing algorithms on overflows and the total cost of wirelength, which are the two most important optimization goals for the multilayer global routing problem.

Key words X-architecture, multilayer routing, very large scale integration (VLSI), global routing, particle swarm optimization (PSO)

Citation Liu Geng-Geng, Zhuang Zhen, Guo Wen-Zhong, Chen Guo-Long. A high performance X-architecture multilayer global router for VLSI. *Acta Automatica Sinica*, 2020, 46(1): 79–93

超大规模集成电路 (Very large scale integration, VLSI) 的布线过程是由总体布线和详细布线

两阶段构成的. 在总体布线中, 每个线网的走线被分配到各个通道布线区域中, 每个通道区域的布线问题得到明确定义. 而详细布线则给出了每个线网在通道区域的具体位置. 因此, 总体布线的质量严重影响详细布线的成功率, 进而对整个芯片的性能起到决定性的作用^[1].

鉴于总体布线是 VLSI 物理设计中极为重要的一部分, 学者们提出很多有效的算法, 主要可分为串行算法和并行算法两种. 以串行算法为代表的总体布线方法能够处理大规模的问题^[2–8]. 但串行算法对线网的布线顺序具有严重的依赖性, 这一潜在特性影响了总体布线的质量. 而以整数线性规划为代表的并行算法能够减少布线结果对线网顺序的依赖性, 取得质量较好的总体布线方案^[9–12], 但这些以整数线性规划为代表的部分方法利用随机取整的方法将线性解转换为非线性解, 这样导致所取得的布线方案可能严重偏离真正的解方案.

收稿日期 2017-12-21 录用日期 2018-04-16
Manuscript received December 21, 2017; accepted April 16, 2018

国家重点基础研究发展计划 (973 计划) (2011CB808000), 国家自然科学基金 (61877010, 11501114), 福建省自然科学基金 (2019J01243), 福建省科技创新平台项目 (2014H2005, 2009J1007) 资助

Supported by National Basic Research Program of China (973 Program) (2011CB808000), National Natural Science Foundation of China (61877010, 11501114), Natural Science Foundation of Fujian Province (2019J01243), and Technology Innovation Platform Project of Fujian Province (2014H2005, 2009J1007)

本文责任编辑 王鼎

Recommended by Associate Editor WANG Ding

1. 福州大学数学与计算机科学学院 福州 350116 2. 福建省网络计算与智能信息处理重点实验室 福州 350116 3. 空间数据挖掘与信息共享教育部重点实验室 福州 350116

1. College of Mathematics and Computer Sciences, Fuzhou University, Fuzhou 350116 2. Key Laboratory of Network Computing and Intelligent Information Processing, Fujian Province, Fuzhou 350116 3. Key Laboratory of Spatial Data Mining & Information Sharing, Ministry of Education, Fuzhou 350116

目前,大部分总体布线算法都是以曼哈顿结构为模型基础开展相关工作,而基于曼哈顿结构的优化策略在进行互连线线长优化时,由于绕线方向只能为水平和垂直,其优化能力受限.因此,有必要从根本入手,改变传统的曼哈顿结构,故研究人员开始尝试以非曼哈顿结构为基础模型进行布线,实现芯片整体性能的优化,特别是出现专门的工业联盟推广 X 结构,为这样的研究提供实现和验证基础^[13].目前关于 X 结构布线的研究主要包括两方面: X 结构布线树的构造^[13-16]和 X 结构总体布线算法设计^[17-20].现有 X 结构布线树的构造^[13-15]大多基于贪心策略,易陷入局部极值,不能保证 Steiner 树的质量.同时关于 X 结构总体布线的研究^[17-20]主要集中在在线长优化,这些算法是基于早期的 Steiner 树构造方法,从而导致了这些 X 结构总体布线算法相对于曼哈顿结构总体布线算法的优化效果不明显.而且以上 X 结构布线算法均属于串行算法,非常依赖于线网的布线顺序或布线代价函数的定义.为此,我们前期构建了布线器 XGRouter,是第一次使用并行算法处理 X 结构总体布线问题,并采用划分策略以减少所构建 ILP 模型的复杂度,从而获得高质量的布线解^[21].

此外,随着集成电路设计进入纳米领域,布线层数增加,使电路的性能和密度得到了很大的提高,因此多层布线应运而生,并且引起了诸多研究机构的广泛关注^[1].而现有关于非曼哈顿结构总体布线问题的研究中,仅局限于单层结构总体布线问题的研究.据我们所知,尚未开展考虑到非曼哈顿结构、多层结构两个条件的总体布线算法研究.因此,本文以 X 结构作为非曼哈顿结构的代表,基于整数规划模型(Integer linear programming, ILP)和粒子群优化算法(Particle swarm optimization, PSO),针对现有工作存在的一系列不足,提出了三种有效的加强策略,并引入了多层布线模型和层调度策略,从而构建了一种高性能 X 结构多层总体布线器,称为 ML-XGRouter.

本文工作是第一次有效解决 X 结构多层总体布线问题,并在基准电路上的仿真结果充分验证本文相关策略和算法的可行性和有效性,且在溢出数和线长总代价两个最重要指标取得显著优化效果.

1 问题描述

给定一个总体布线图 G , 包含一个引脚集合 V , 边集合 E , 一个线网集合 $N = \{N_1, N_2, \dots, N_n\}$, 其中线网 N_k 包含一系列引脚集合, 则以线长为优化目标的总体布线问题可描述如下.

$$\begin{aligned} \min \quad & \sum_{k=1}^n L(T_k) \\ \text{s. t.} \quad & U(e_i) \leq C(e_i), \forall e_i \in E \end{aligned} \quad (1)$$

其中, $L(T_k)$ 表示布线树 T_k 的长度, 布线边的最大容量 $C(e_i)$ 表示该边 e_i 能够允许的最大走线数. $U(e_i)$ 表示总体布线问题中经过边 e_i 的次数, 可表示为

$$U(e_i) = \sum_{k=1}^n x_{ik} \quad (2)$$

其中, n 是布线树的总数, x_{ik} 是一个二进制数. 当布线树 T_k 的走线经过边 e_i , 则 x_{ik} 的值为 1; 否则为 0.

式 (1) 中的容量约束是限制溢出边的个数为 0, 从而满足溢出数这一最重要目标的优化. 同时在多层布线中存在通孔的情况, 所以在线长总代价中需要加入通孔数目的代价.

定义 1 (网表)^[1]. $N = \{N_1, N_2, \dots, N_n\}$, 其中每个线网 N_k 是一系列引脚的集合, 属于同一个线网的引脚集合将由布线工具把它们连接一起. 网表是提供电路的互连信息.

定义 2 (布线容量表)^[1]. 在布线区域内所能容下最大的走线数, 它是由设计规则、布线区域的大小和线网的宽度所决定的.

定义 3 (溢出边). 当一个边的布线资源需求量大于可用的布线资源, 则该边称为溢出边.

图 1 分别给出了基于曼哈顿结构和 X 结构的总体布线图(Global routing graph, GRG). 从图 1 中可看出由于 X 结构布线设计允许相对更多的走线方向, 所以基于 X 结构的 GRG 图相对于基于曼哈顿结构的 GRG 图更复杂.

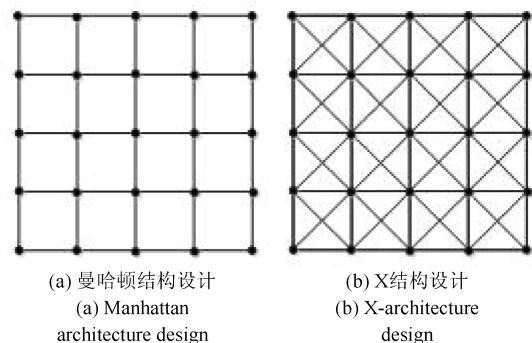


图 1 总体布线图

Fig. 1 The design graph of global routing

X 结构网格是由水平线、垂直线、45° 的线、135° 的线组成的. 引脚 P_i 和 P_j 之间在 X 结构网格下的距离用边 $e(P_i, P_j)$ 的长度表示, 令

$P_i = (X_i, Y_i)$ 和 $P_j = (X_j, Y_j)$ 分别表示两引脚的坐标.

引理 1. 如果 $(X_i == X_j)$ 且 $(Y_i \neq Y_j)$, 那么两引脚 P_i 和 P_j 之间的距离 $e(P_i, P_j) = |Y_i - Y_j|$. 两引脚沿垂直方向走线.

引理 2. 如果 $(X_i = X_j)$ 且 $(Y_i == Y_j)$, 那么两引脚 P_i 和 P_j 之间的距离 $e(P_i, P_j) = |X_i - X_j|$. 两引脚沿水平方向走线.

引理 3. 如果 $(X_i = X_j)$ 和 $(Y_i = Y_j)$ 并且 $(X_i - X_j) == (Y_i - Y_j)$ 与 $(X_i - X_j) == -(Y_i - Y_j)$ 之一成立, 那么两引脚 P_i 和 P_j 之间的距离 $e(P_i, P_j) = \sqrt{2} \times |X_i - X_j|$. 两引脚位于网格的对角线上, 因此两引脚的走线方向为 45° 或 135° .

引理 4. 如果 $(X_i = X_j)$ 和 $(Y_i = Y_j)$ 并且 $|X_i - X_j| < |Y_i - Y_j|$, 那么两引脚 P_i 和 P_j 之间的距离 $e(P_i, P_j) = (\sqrt{2} - 1) \times |X_i - X_j| + |Y_i - Y_j|$. 两引脚的走线方向包括对角线方向和垂直方向.

引理 5. 如果 $(X_i = X_j)$ 和 $(Y_i = Y_j)$ 并且 $|X_i - X_j| > |Y_i - Y_j|$, 那么两引脚 P_i 和 P_j 之间的距离 $e(P_i, P_j) = (\sqrt{2} - 1) \times |Y_i - Y_j| + |X_i - X_j|$. 两引脚的走线方向包括对角线方向和水平方向.

2 算法实现

2.1 XGRouter 的概述

XGRouter 主要包括三个阶段: 初始布线阶段、主阶段、后处理阶段^[21].

2.1.1 初始布线阶段

在初始布线阶段, XGRouter 采用 X 结构 Steiner 最小树算法^[16] 对多端线网进行线网分解. 之后计算每两端点所构成线段的斜率值, 如果其斜率值为 0, -1 , $+1$ 和 ∞ , 则将在不违反通道容量约束的前提下以 X 结构边连接该两端线网. 如果 GRG 边的走线数超过了最大通道容量, 则放弃这类线网的连接. 算法检查每个线网是否满足预布线的条件. 如果满足, 则直接连接该线网并标识为已连接; 否则, 继续检查下一个线网直至所有线网都被检查过.

2.1.2 主阶段

在主阶段, XGRouter 采用划分策略以减少总体布线问题的规模, 从而有助于 PSO 有效求解 ILP 模型. 初始布线子区域的选择对最终总体布线的质量有重大影响, 而选择最拥挤区域作为初始布线子区域是一种不错的选择^[10]. 在初始布线阶段完成后, 可获得芯片的近似拥挤情况, 从中寻找最拥挤区域. 该算法选择网格大小规模为 2×2 的区域作为主阶段的初始布线子区域, 为该子区域的布线问题建

立相应 ILP 模型并采用改进的 PSO 算法进行求解. 在布通每个子区域后, 沿该区域的所有布线方向扩展一个网格大小作为下一个布线子区域, 并采用相同的方法进行布线, 直至目前的布线子区域已包含整个芯片区域.

2.1.3 后处理阶段

在主阶段完成后, 总体布线方案中会存在一些未连接线网. 为此, 后处理阶段设计了基于新布线边代价的迷宫算法用以布通这些线网, 是最终布通所有线网的重要步骤. 在新布线边代价的基础上同时考虑到线长和拥挤均衡, 不断执行迷宫布线直到不存在未连接线网.

2.2 存在的不足及相应的加强策略

2.2.1 增加新型走线方式 (E1 策略)

性质 1. ML-XGRouter 采用 E1 策略, 增加新型走线方式有助于初始阶段中放弃连接的线网集在主阶段的布通率.

在 XGRouter 的初始布线阶段中, 针对分解后两引脚所构成的直线斜率值为 0, -1 , $+1$ 和 ∞ 的线网 (此类线网集称为 NA), 如果采用该直线连接两引脚, 不会超过连接边的容量, 则用该直线连接该两端线网. 但若造成溢出的情况, 如图 2 所示, 则放弃连接该类两端线网 (此类线网集称为 NC), 并放在主阶段进行连接. 而在主阶段其连接仍采用图 2 所示的方式, 将导致这些 NC 线网在主阶段仍不可连接, 从而导致非常多未能连接的线网, 严重影响 PSO 算法的求解性能.

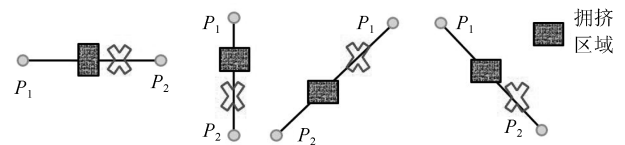


图 2 初始布线阶段由于布线容量约束造成未能直接连线的 4 种情况

Fig. 2 The four cases of the failure to connect directly due to the limitation of routing capacity in the initial stage

因此, 在本文主阶段中, 针对 NC_1 设计了新型走线方式, 针对水平或垂直关系的线网采用图 3 (a) 和 3 (b) 的两种连接方式, 而针对 45° 或 135° 关系的线网 (NC_2) 采用图 3 (c) 的连接方式, 通过新增布线方式, 主阶段可合理避开拥挤区域.

本文将未采用和采用 E1 策略的总体布线算法在 ISPD07 的基准电路上^[22] 进行实验对比, 如表 1 所示. 采用 E1 策略相对未采用 E1 策略 (表中用 E0 表示) 的总体布线结果在总溢出数 (TOF) 方面取得

表 1 未采用和采用 E1 策略 (E2 策略) 的布线结果对比
Table 1 Comparison of routing results without and with E1 strategy (E2 strategy)

| 基准 电路 | E0 | | E1 | | 减少率 (%) | | E2 | | 减少率 (%) | |
|----------|-------|-------|-------|-------|--------------|--------------|-------|-------|--------------|--------------|
| | TOF | TWL | TOF | TWL | TOF | TWL | TOF | TWL | TOF | TWL |
| 11 | 85 | 46.7 | 85 | 46.9 | 0 | -0.43 | 0 | 47.9 | 100 | -2.57 |
| 12 | 75 | 45.9 | 70 | 45.8 | 6.67 | 0.22 | 0 | 46.6 | 100 | -1.53 |
| 13 | 44 | 127.1 | 34 | 127.2 | 22.73 | -0.08 | 0 | 128.9 | 100 | -1.42 |
| 14 | 12 | 110.8 | 7 | 110.9 | 41.67 | -0.09 | 0 | 111 | 100 | -0.18 |
| 15 | 32 | 130.8 | 29 | 131.4 | 9.38 | -0.46 | 0 | 130.3 | 100 | 0.38 |
| 16 | 2 | 33.9 | 1 | 33.9 | 50 | 0 | 0 | 33.9 | 100 | 0 |
| 17 | 0 | 63.4 | 0 | 63.3 | 0 | 0.16 | 0 | 63.5 | 0 | -0.16 |
| 18 | 3 779 | 80 | 3 682 | 80 | 2.57 | 0 | 3 170 | 80.8 | 16.12 | -1 |
| AVG | | | | | 16.63 | -0.09 | | | 77.01 | -0.81 |

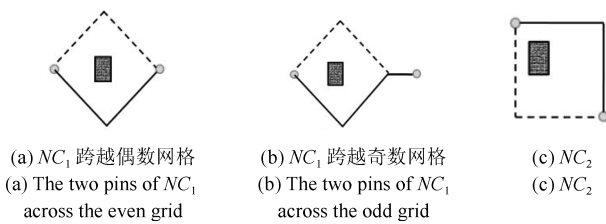


图 3 E1 策略
Fig. 3 E1 Strategy

了 16.63% 的减少率, 表明 E1 策略有助于主阶段尽可能多连接 NC , 提高布通率. 虽然增加了少量的线长总代价 (TWL), 但针对溢出数带来可观的优化, 从而表明 E1 策略的有效性.

2.2.2 PSO 与迷宫算法的结合策略 (E2 策略)

性质 2. ML-XGRouter 采用 E2 策略, 将基于新布线代价的迷宫布线策略移至主阶段中每次 Box 区域内的 PSO 布线后, 可有效提高当前 Box 未布通线网集的布通率.

XGRouter 的迷宫布线是放在主阶段之后, 容易堆积非常多的未布通线网, 严重影响到 PSO 算法的求解质量. 如图 4 所示, P_1 和 P_2 在 Box_0 的阶段中, 由于灰色拥挤区域的存在, P_1 和 P_2 在采用 XGRouter 的边连接方式, 是不能在 Box_0 阶段完成布线. 而如果等到 Box_0 扩张到整个芯片的阶段, P_1 和 P_2 可能要采用图 4 的实线连接, 占用更多的布线资源, 从而导致溢出数的增加. 而 ML-XGRouter 会在 Box_0 布线结束后, 立即引入迷宫布线连接 PSO 未能布线的线网, 即采用图 4 的虚线进行连接, 尽可能占用 Box_0 附近的资源, 可有效减少布线资源的冗余占用, 有效提高布线的完成率.

本文将未采用和采用 E2 策略的总体布线结果进行对比, 如表 1 的 8~11 列所示. 采用 E2 策略相对未采用 E2 策略的总体布线结果在总溢出数取得了 77.01% 的大幅度减少率, 表明 E2 策略有助于在

主阶段合理使用线网附近的布线资源, 从而产生更少的溢出. 虽然付出了少量的线长总代价 (0.81%), 但是对溢出数带来大幅度优化, 从而表明 E2 策略的有效性.

$Box_n =$ 整个芯片

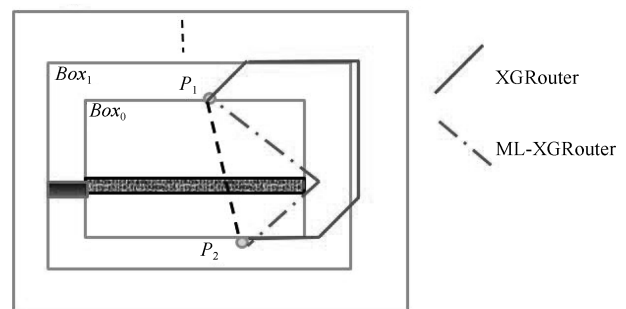


图 4 E2 策略

Fig. 4 E2 Strategy

2.2.3 初始布线阶段的布线容量缩减 (E3 策略)

性质 3. ML-XGRouter 采用 E3 策略, 将初始布线阶段的布线容量缩减到原容量的一半, 使得部分线网留在主阶段用 PSO 算法布线, 从而可有效利用 PSO 算法的全局优化能力, 同时进一步加强 E1 和 E2 策略的优化效果.

在 XGRouter 中, 将基于贪心策略的初始布线算法应用到 NA 中的全部线网, 会使得布线方案陷入较为严重的局部最优. 因此, ML-XGRouter 采用 E3 策略, 预留了部分布线资源和部分两端线网交给 PSO 算法进行布线, 从而可有效利用 PSO 算法的全局优化能力.

如图 5 所示, 假设初始布线中线网的布线顺序为 N_1 (包含 N_{11} 和 N_{12} 引脚), N_2 (包含 N_{21} 和 N_{22} 引脚), N_3 (包含 N_{31} 和 N_{32} 引脚), 布线边的最大容量为 2. 采用 XGRouter 的布线策略, 即将 N_1 和 N_2 优先连接, 而在主阶段中 N_3 的连接需要占用非常多的布线资源, 如图 5 (a) 所示. 但如果采用 E3

策略, 预留一半的布线资源, 初始布线只连接 N_1 , 而 N_2 和 N_3 则留在主阶段处理, PSO 算法可从全局的角度去衡量 N_2 和 N_3 的布线方式, 从而得到图 5 (b) 的方案, 以更少的布线资源完成 N_1 , N_2 和 N_3 的连接.

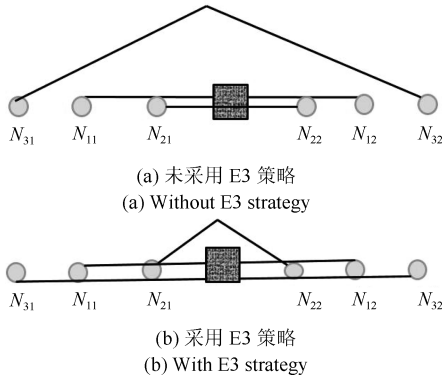


图 5 E3 策略

Fig. 5 E3 Strategy

为了进一步加强 E1 策略的优势, 可引入 E3 策略, 从而提高 ML-XGRouter 在线长总代价的优化能力, 同时增强溢出数的优化能力. 在初始布线阶段, 限制水平和垂直方向的通道容量为所设置容量

的 1/2, 从而通过 E1 策略完成布线的线网数量大大提高, 并在主阶段可有效利用 PSO 算法的优势, 取得更好的优化效果.

如表 2 的第 6 和 7 列所示, 在 E1 策略基础上加入 E3 策略, 可在溢出数和线长总代价分别取得 17.27% 和 1.27% 的优化效果. 如表 2 的第 12 和 13 列所示, 在 E2 策略的基础上加入 E3 策略, 可在溢出数和线长总代价分别取得 0.02% 和 0.39% 的优化效果. 如表 3 的第 6 和 7 列所示, 在 E1 和 E2 策略的基础上加入 E3 策略, 可在溢出数和线长总代价分别取得 1.38% 和 1.45% 的优化效果. 综上, 可表明 E3 策略引入的有效性, 特别是针对 E1 策略, E3 显得尤为有效.

2.3 基于三种加强策略后布线器的新流程

针对 X 结构多层总体布线问题, 在前期工作基础上引入了上述三个有效的加强策略 (E1, E2 和 E3 策略), 从而提出了一种求解 X 结构多层总体布线问题的有效算法, 如图 6 所示, 包括了初始布线阶段、主阶段、层分配阶段: 1) 新布线器的初始布线引入了 E3 策略, 将初始布线阶段的预布线容量缩减到原容量的一半, 使得部分线网和布线资源能够留在主阶段进行布线, 有效利用 PSO 算法的全局优化能力,

表 2 在 E1 策略 (E2 策略) 的基础上未采用和采用 E3 策略的布线结果对比

Table 2 Comparison of routing results without and with E3 strategy, on the basis of E1 strategy (E2 strategy)

| 基准 电路 | E1 | | E1+E3 | | 减少率 (%) | | E2 | | E2+E3 | | 减少率 (%) | |
|----------|------|-------|-------|-------|--------------|-------------|------|-------|-------|-------|-------------|-------------|
| | TOF | TWL | TOF | TWL | TOF | TWL | TOF | TWL | TOF | TWL | TOF | TWL |
| 11 | 85 | 46.9 | 68 | 46.4 | 20 | 1.07 | 0 | 47.9 | 0 | 47.5 | 0 | 0.84 |
| 12 | 70 | 45.8 | 49 | 44.4 | 30 | 3.06 | 0 | 46.6 | 0 | 45.6 | 0 | 2.15 |
| 13 | 34 | 127.2 | 27 | 123.7 | 20.59 | 2.75 | 0 | 128.9 | 0 | 127.4 | 0 | 1.16 |
| 14 | 7 | 110.9 | 7 | 110.2 | 0 | 0.63 | 0 | 111 | 0 | 110.7 | 0 | 0.27 |
| 15 | 29 | 131.4 | 13 | 127.7 | 55.17 | 2.82 | 0 | 130.3 | 0 | 129.8 | 0 | 0.38 |
| 16 | 1 | 33.9 | 1 | 33.9 | 0 | 0 | 0 | 33.9 | 0 | 34 | 0 | -0.3 |
| 17 | 0 | 63.3 | 0 | 63 | 0 | 0.47 | 0 | 63.5 | 0 | 64.2 | 0 | -1.1 |
| 18 | 3682 | 80 | 3226 | 80.5 | 12.38 | -0.63 | 3170 | 80.8 | 3165 | 81 | 0.16 | -0.25 |
| AVG | | | | | 17.27 | 1.27 | | | | | 0.02 | 0.39 |

表 3 在 E1 和 E2 策略共同作用的基础上未采用和采用 E3 策略的布线结果对比

Table 3 Comparison of routing results without and with E3 strategy, on the basis of E1 and E2 strategies

| 基准 电路 | E1+E2 | | E1+E2+ E3 | | 减少率 (%) | |
|----------|-------|-------|-----------|-------|-------------|-------------|
| | TOF | TWL | TOF | TWL | TOF | TWL |
| 11 | 0 | 47.8 | 0 | 47 | 0 | 1.67 |
| 12 | 0 | 46.5 | 0 | 45 | 0 | 3.23 |
| 13 | 0 | 128.8 | 0 | 124.6 | 0 | 3.26 |
| 14 | 0 | 110.9 | 0 | 110.1 | 0 | 0.72 |
| 15 | 0 | 130.2 | 0 | 126.8 | 0 | 2.61 |
| 16 | 0 | 33.9 | 0 | 33.9 | 0 | 0 |
| 17 | 0 | 63.4 | 0 | 63.1 | 0 | 0.47 |
| 18 | 3106 | 80.8 | 2762 | 81.1 | 11.08 | -0.37 |
| AVG | | | | | 1.38 | 1.45 |

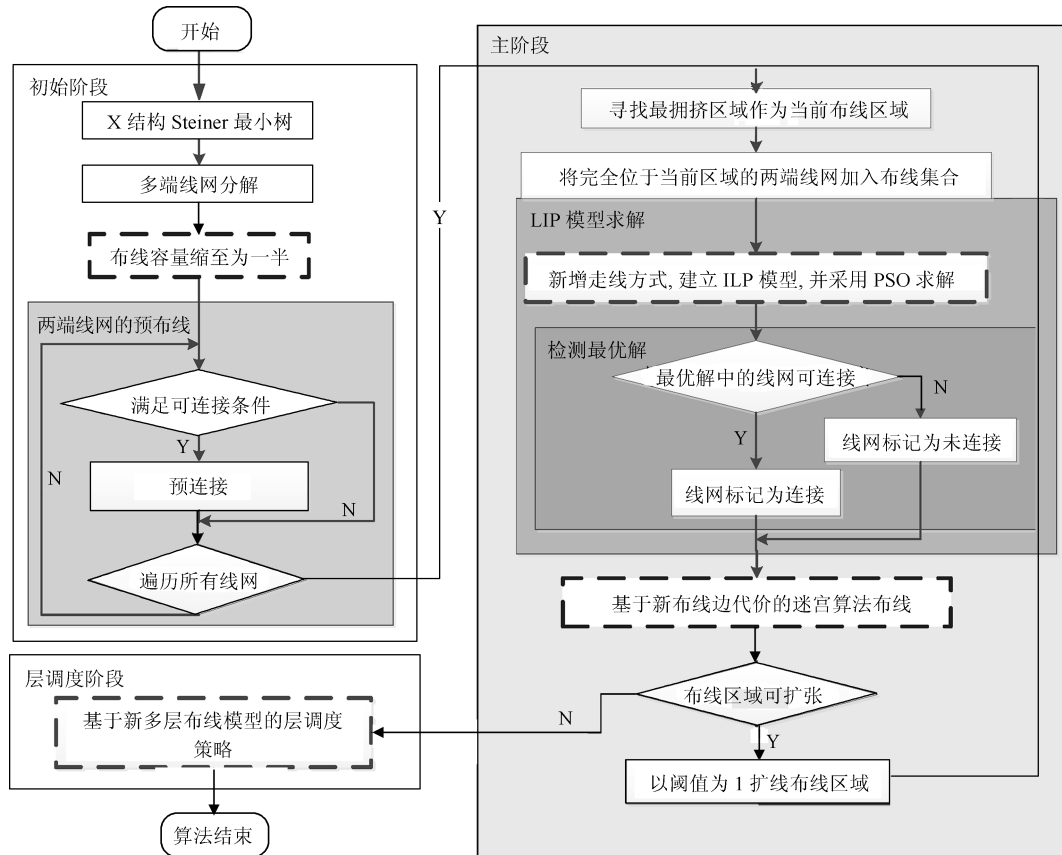


图 6 ML-XGRouter 的算法流程

Fig. 6 The flow chart of ML-XGRouter

同时进一步加强 E1 和 E2 的优化效果. 2) 新布线器的主阶段引入了 E1 和 E2 策略, 即采用 E1 策略, 增加新型走线方式有助于提高 NC 在主阶段的布通率. 而采用 E2 策略, 将迷宫布线策略移至主阶段中每次 PSO 布线之后, 有助于提高当前 Box 区域内未布通线网集合的布通率. 3) 由于 ML-XGRouter 是用于求解多层总体布线问题, 需要在层分配阶段将相关线网的片段分配到合适的布线层上, 本文的层分配策略是基于文献 [23] 验证的最佳多层布线模型.

接下来将介绍新总体布线算法中涉及到的 ILP 模型 (本文的 ILP 模型简称为 MX-ILP) 和 PSO 算法 (本文的 PSO 算法简称为 MX-PSO).

2.4 MX-ILP

本文采用并行算法求解 X 结构多层总体布线问题. 先前的并行算法常用 ILP 模型求解总体布线问题, 并可取得在溢出数和线长总代价两方面优化不错的布线方案. 通常, 基于 ILP 模型的方法, 首先, 寻找所有可能的布线树作为候选解, 并计算每个布线树的代价, 建立相应的带容量约束的线性方程组, 最后, 求解该线性方程组以获得最终的总体布线方

案. 为了更好地求解总体布线问题, 学者提出不同类型的 ILP 模型, 具有各自的优缺点^[21]. 而本文布线器在主阶段采用的 ILP 模型, 围绕多层布线问题中两个最重要的优化目标包括溢出数和线长总代价, 对 XGRouter 的 ILP 模型进行一定程度上的改进使得其在这两个最重要指标的优化能力得到提高.

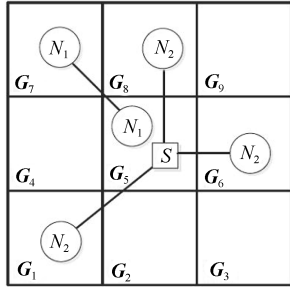
图 7(a) 给出了准备布线的线网 N_1 和 N_2 以及最大通道容量为 2. 本文通过 Steiner 最小树算法将 N_2 分解为 3 个两端线网 (SG_8 , SG_6 和 SG_1), 并为所有两端线网建立相应的布线候选解, 如图 7(b) 所示.

本文的 MX-ILP 模型是为了最小化所有线网的线长总代价和溢出数, 针对图 7(b) 的布线问题, MX-ILP 模型可获得图 7(c) 和图 7(d) 的布线方案. 其具体形式如式 (3) 所示, 式 (4) 则是 MX-ILP 模型的一般形式.

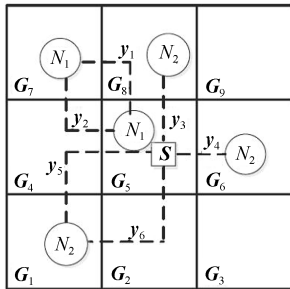
$$\begin{aligned} \min \quad & \beta \times (2y_1 + 2y_2 + y_3 + y_4 + 2y_5 + 2y_6) \\ \text{s. t.} \quad & y_1 + y_2 = 1 \\ & y_3 = 1, y_4 = 1 \\ & y_5 + y_6 = 1 \\ & y_1 + y_3 \leq 2 \end{aligned}$$

$$\begin{aligned}
 &y_2 + y_5 \leq 2 \\
 &y_1 \leq 2, y_2 \leq 2, y_4 \leq 2 \\
 &y_5 \leq 2, y_6 \leq 2 \\
 &y_1, y_2, y_3, y_4, y_5, y_6 \in \{0, 1\}
 \end{aligned}
 \tag{3}$$

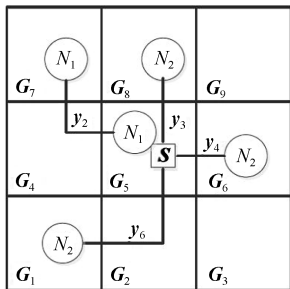
$$\begin{aligned}
 &\min \quad \beta \times \left(\sum_{j=1}^t y_j \times Wl_j \right) \\
 &\text{s. t.} \quad \sum_{y_j \in N_k} y_j = 1, k = 1, 2, \dots, n \\
 &\beta = \begin{cases} 1, & \sum_{j=1}^1 X_{ij} \times y_j \leq C(e), \\ & \forall e, i = 1, 2, \dots, p \\ 1.026^r, & \text{否则} \end{cases} \tag{4} \\
 &y_j \in \{0, 1\}, j = 1, 2, \dots, t
 \end{aligned}$$



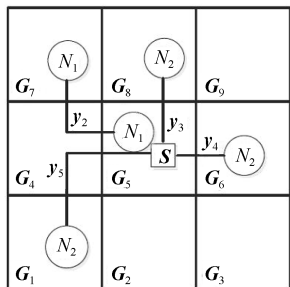
(a) 两个待布线网
(a) Two nets



(b) N_2 的分解
(b) The net decomposition for N_2



(c) 布线方案一
(c) The first routing solution



(d) 布线方案二
(d) The second routing solution

图 7 总体布线问题的实例

Fig. 7 An instance for the global routing problem

其中, Wl_j 表示方案 y_j 的线长, β 是惩罚项. 第一个约束集要求每个线网从其候选解集中只能选取一个作为其布线方案, y_j 代表一个候选解, k 表示线网的总数. 第二个约束集对 GRG 中每条边的布线数超过最大容量限制 $C(e)$ 给予相应的惩罚, r 表示违反约束的次数. X_{ij} 表示一个二进制值, 如果 y_j 的布线经过 GRG 中的边 e_i , 则 X_{ij} 的值为 1; 否则, 为 0. p 表示 GRG 中所有边的总数目. 第三个约束集限制 y_j 是二进制值, t 表示布线候选解的总个数.

性质 4. 本文算法基于 MX-ILP 模型可获得多层布线问题中包括溢出数、线长总代价这两个重要指标的优化.

MX-ILP 模型相对 XGRouter 的 ILP 模型在优化目标上减少了拥挤度均衡策略. 这是由于在 ISPD07 问题中相对 ISPD98 问题^[24] 而言, 线网个数陡增, 布线容量约束更严格, 更容易产生溢出. 而在 ISPD98 问题中, 我们已经可以做到全部基准电路的溢出数都优化至 0, 才进一步考虑拥挤均衡, 而在 ISPD07 部分基准电路的溢出数在所难免, 若进一步考虑拥挤均衡, 将增加优化难度, 产生更多溢出数, 严重影响芯片的可布线性. 另外, 拥挤均衡的计算时间成本非常高, 减少了拥挤均衡有助于算法整体运行时间的减少. 而 MX-ILP 模型的优化目标集中在溢出数和线长总代价的最小化, 从而有助于在 ISPD07 中这两个重要目标的优化, 具体优化结果详见第 3 节的实验结果.

MX-ILP 模型的优势可归纳如下.

1) MX-ILP 模型中每个优化目标都对应一个权重因子, 使得其他优化目标可容易融入至 MX-ILP 模型的目标函数中.

2) MX-ILP 模型的构建是有利于设计更有效的 PSO 算法以求解相应的 ILP 模型. 因为在 MX-ILP 模型的求解中引入的惩罚函数机制, 对优化问题中出现不满足约束条件的解方案施以相应的惩罚, 保留这类解方案不被随意删除. 该类解方案可能携带一些局部最优信息, 在后续进化过程中有机会呈现该类解方案所保留的局部最佳信息, 从而增强种群的多多样性.

3) 本文采用 PSO 算法求解 MX-ILP 模型, 所得到的解方案均为整数解, 可解决由于随机取整方法可能带来的偏差问题。

2.5 MX-PSO

针对 PSO 算法应用于离散问题中, 已经有不同学者采用不同策略将连续型的 PSO 算法离散化, 包括将速度作为位置变化的概率^[25]、直接将连续 PSO 用于离散问题的求解^[26] 和重新定义 PSO 算法操作算子^[27]。其中, 1) 以文献 [25] 为代表的这类离散 PSO 算法, 其参数更新方式是基于线性递减策略, 而本文提出的 MX-PSO 的惯性权值和学习因子采用的是自适应参数调整策略, 相对于基于线性递减策略具有更好的收敛效果; 2) 以文献 [26] 为代表的这类离散 PSO 算法, 由于多个不同连续解可能对应一个整数解, 并使得取整后的数可能违反约束条件或其解不是可行解; 3) 以文献 [27] 为代表的这类离散 PSO 算法, 借鉴进化算法的操作方式, 在所提算法中首先通过遗传变异操作实现自身速度的改变, 继而通过交叉操作实现个体极值和全局极值对粒子的影响, 可有效应用到以旅行商问题为代表的离散问题中。

课题组借鉴以文献 [27] 为代表的这类离散 PSO 算法的粒子位置更新思想, 已在 VLSI 物理设计中电路划分、布线等问题上构建了求解这些离散问题的有效且成熟的 PSO 算法框架^[28-30]。因此, 本文基于该类 PSO 框架, 根据 MX-ILP 问题的特点, 设计有效的编码策略, 重新定义 PSO 的操作算子和适应度函数, 从而提出一种求解 MX-ILP 模型的有效 PSO 算法。

2.5.1 编码策略

由于 PSO 算法的鲁棒性, PSO 算法对编码策略的要求并不严格^[31-35]。尽管如此, 一个好的编码策略不仅能改进算法的搜索效率, 而且能够减少冗余的算法搜索空间, 进而提高算法的性能。编码策略的选择主要考虑三个准则: 完整性、健全性及非冗余性^[28]。

定义 4 (完整性). 问题搜索空间的每个点都能成为粒子编码空间中相应点的表现形式。

定义 5 (健全性). 编码空间中每个粒子均能对应到问题搜索空间中潜在解方案。

定义 6 (非冗余性). 编码空间中每个粒子能与问题搜索空间的潜在解方案一一对应。

对于一个编码策略要完全满足以上三个编码准则是非常困难的。本文算法提出一种 0-1 整数编码策略表示一个布线方案, 即 MX-PSO 算法中的一个粒子。图 8 给出了图 7(b) 所示的布线问题所对应的粒子编码, 包括为 4 个线网所构造的 6 个候选解,

因此该粒子的编码长度是 6。

| | | | | | | |
|---------|---|---|---|---|---|---|
| 布线候选解编号 | 1 | 1 | 3 | 4 | 5 | 6 |
| 粒子的编码 | 0 | 1 | 0 | 0 | 1 | 0 |

图 8 图 7(b) 中布线实例的编码图

Fig. 8 The encoding for the routing instance in Fig. 7(b)

性质 5. 针对总体布线问题所设计的 0-1 整数编码策略, 能够满足完整性和非冗余性准则, 但不能完全满足健全性准则。

如图 7(d) 所示的布线方案, 其粒子的编码为 ‘011110’。假设最大通道容量为 1, 那么如图 7(d) 所示的边 G_{45} 的 (布线单元 G_4 和 G_5 间的边) 实际走线数为 2, 大于最大通道容量, 违反了 MX-ILP 模型的约束, 是一个不可行解, 所以其不满足健全性准则。

为了更好地满足健全性准则, 在本文的交叉算子和变异算子融入检查策略, 并且在适应度函数的设计中引入惩罚机制策略。这样可确保每个线网只有一个候选解会被选中, 且对违反最大通道容量约束的粒子施加严厉的惩罚。

2.5.2 粒子的更新公式

性质 6. 在粒子的更新操作算子中融入检查策略, 使得本文的编码策略可较好地满足健全性准则。

对总体布线问题所建立的 ILP 模型是一个离散问题, 而基本 PSO 构建的粒子更新公式不再适应于该离散问题的求解。为此, 本文算法设计了交叉算子和变异算子以构建离散形式的粒子更新公式^[28-30], 并将检查策略融入这些算子以满足健全性原则, 最终重新定义的粒子更新公式如下所示。

$$X_i^t = N_3(N_2(N_1(X_i^{t-1}, w), c_1), c_2) \quad (5)$$

其中, w 是惯性权重因子, c_1 和 c_2 是加速因子, N_1 表示变异操作, N_2 和 N_3 表示交叉操作。

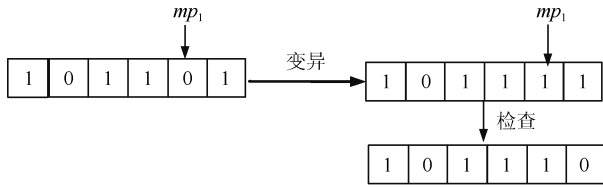
1) 粒子的速度更新可表示如下。

$$W_i^t = N_1(X_i^{t-1}, w) = \begin{cases} M(X_i^{t-1}), & r_1 < w \\ X_i^{t-1}, & \text{否则} \end{cases} \quad (6)$$

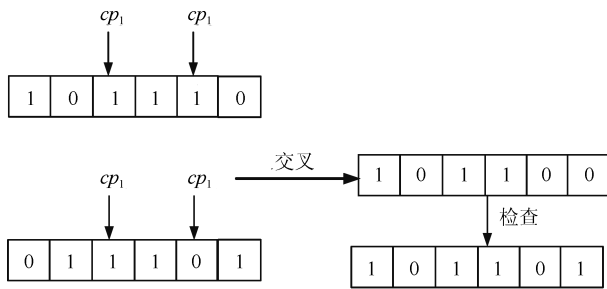
其中, w 表示粒子进行变异操作的概率, 假定 r_1 是在区间 $[0, 1)$ 的随机数。

如图 9(a) 所示的变异操作算子的执行过程中, 粒子编码中一个随机位 (mp_1) 被选中为变异操作位置, 再根据式 (9) 重新计算 mp_1 的值 (val), 其中符号 ‘%’ 表示取模运算符。在执行变异操作后, 粒子的编码更新为 ‘101111’, 其中, 该粒子对应的布线方案违反第一个约束集, 因为如图 7(b) 所示同一个线

网的两个候选解 y_5 和 y_6 同时被选中. 因此, 需要设计一个检查策略用以检查布线方案是否满足第一个约束集, 在执行检查步骤后, 粒子的编码变为如图 9(a) 所示的 ‘101110’.



(a) 本文算法的变异算子
(a) The mutation operator in this paper



(b) 本文算法的交叉算子
(b) The crossover operator in this paper

图 9 本文的两种操作算子

Fig. 9 Two operators in this paper

2) 粒子的个体经验学习可表示如下.

$$S_i^t = N_2(W_i^t, c_1) = \begin{cases} C_p(W_i^t), & r_2 < c_1 \\ W_i^t, & \text{否则} \end{cases} \quad (7)$$

其中, c_1 表示粒子与其历史最优方案进行交叉操作的概率, 假定 r_2 是在区间 $[0, 1)$ 的随机数.

3) 粒子与种群其他粒子进行合作学习可表示如下.

$$X_i^t = N_3(S_i^t, c_2) = \begin{cases} C_p(S_i^t), & r_3 < c_2 \\ S_i^t, & \text{否则} \end{cases} \quad (8)$$

其中, c_2 表示粒子与种群全局最优方案进行交叉操作的概率, 假定 r_3 是在区间 $[0, 1)$ 的随机数.

如图 9(b) 所示的交叉算子, 粒子编码中有两个随机位置 (cp_1 和 cp_2 , 令 $cp_1 < cp_2$) 被选中, 之后粒子与个体历史最优方案 (全局最优方案) 交换从 cp_1 到 cp_2 这一区间的编码值. 类似变异操作算子, 检查策略也被执行, 从而获得交叉后粒子的最终编码为 ‘101101’.

$$val \Leftarrow (val + 1) \% 2 \quad (9)$$

检查策略的步骤如下所示.

步骤 1. 选择发生变异或交叉操作的粒子编码位置.

步骤 2. 对每个位置 y_j , 其表示线网 N_k 的一个候选解, 检查 N_k 的其他候选解的编码值与 y_j 的值是否满足 MX-ILP 模型的第一个约束集. 若不满足, 则改变其他候选解的编码值使其满足约束, 否则保持编码值不变.

2.5.3 适应度函数

多层总体布线问题中最重要的优化指标包括溢出数和线长总代价. 其中, 溢出数是总体布线第一重要的优化目标, 严重影响芯片设计的可制造性和可布性. 因此, 在 MX-ILP 模型中第二个约束集限制 GRG 中边的实际走线数不超过最大通道容量以避免溢出的产生. 线长总代价同样也是非常重要的布线指标, 因此在算法的设计过程中, 不但考虑溢出数的最小化, 还考虑线长总代价的最小化. 算法的最终适应度函数定义如下.

$$fitness = \beta \times \left(\sum_{j=1}^t y_j \times Wl_j \right) \quad (10)$$

其中, β 表示惩罚项, 如果布线方案不违反最大通道容量约束, β 的值为 1, 否则为 1.026^r , r 是布线方案存在违反约束的次数. Wl_j 表示候选解 y_j 的线长, 它是根据 X 结构网格距离进行计算. 每个线网所包含两个端点位置关系可分为如引理 1~5 所示的 5 种情况, 并根据引理 1~5 的边距离式子计算相应的线长.

2.6 算法的收敛分析

定理 1. MX-PSO 算法的马尔科夫链是有限且齐次的.

证明. 在算法的全局搜索过程中, MX-PSO 算法是通过随机变异操作和交叉操作获得全局最优方案和粒子的历史最优方案. 从全局搜索的过程中判断出, 新产生的种群是取决于当前种群的状态, 因此从一个状态到另一个具体状态的算法搜索过程的条件概率是算法满足马尔科夫性质^[36]. 所以, MX-PSO 算法的马尔科夫链是有限且齐次的. 在该算法中, 所有种群 $\{\alpha_1, \alpha_2, \dots, \alpha_m\}$ 组成的集合是有限的. 也就是发生在时刻 $k = 0, 1, \dots$ 的事件都属于一个有限可数的事件集合, 因此, 其马尔科夫链也是有限的^[37]. 综上, 以上关于马尔科夫链的理论分析和方法均可直接应用到本算法的分析中. \square

定理 2. MX-PSO 算法所构成的马尔科夫链的转移概率矩阵是正定的.

证明. 在算法搜索过程中, 种群通过一个粒子自身的变异操作和两个交叉操作 (粒子与历史最优方案及全局最优方案的交叉操作) 从状态 $i_i \in S$ 转移到状态 $i_j \in S$. 这三个操作的转移概率分别为 m_{ij} , g_{ij} , p_{ij} , 它们相应的转移矩阵分别 $M = \{m_{ij}\}$,

$G = \{g_{ij}\}$, $D = \{d_{ij}\}$. 令 $P = M \cdot G \cdot D$, 则 $m_{ij} > 0$, $\sum_{i_j \in E} m_{ij} = 1$; $g_{ij} \geq 0$, $\sum_{i_j \in E} g_{ij} = 1$; $d_{ij} \geq 0$, $\sum_{i_j \in E} d_{ij} = 1$.

因此, M, G, D 都是随机, 且 M 是正定的, 然后可证明 P 是正定的. 令 $B = G \cdot D$, 对于 $\forall i_i \in S, i_j \in S$, 存在 $b_{ij} = \sum_{\lambda_k \in E} g_{ik} \times d_{kj} \geq 0$, $\sum_{\lambda_j \in E} b_{ij} = \sum_{\lambda_j \in E} \sum_{\lambda_k \in E} g_{ik} \times d_{kj} = \sum_{\lambda_k \in E} g_{ik} \times \sum_{\lambda_j \in E} d_{kj} = \sum_{\lambda_k \in E} g_{ik} = 1$.

因此, B 是随机正定矩阵, 同样可知道 $d_{ij} = \sum_{\lambda_k \in E} b_{ik} \times m_{kj} > 0$. \square

定理 3 (马尔科夫链的极限理论). 假定 P 是齐次马尔科夫链对应的正定随机转移矩阵, 则

1) 存在唯一的概率向量 $\bar{P}^T > 0$, 其满足 $\bar{P}^T \cdot P = \bar{P}^T$.

2) 对于任意的一个初始状态 $i(e_i^T$ 作为其相应的初始概率), 可得 $\lim_{k \rightarrow \infty} e_i^T \cdot P^k = \bar{P}^T$.

3) 概率矩阵的极限为 $\lim_{k \rightarrow \infty} P^k = \bar{P}$, 其中 \bar{P} 是一个 $n \times n$ 的随机矩阵且其所有行等同于 \bar{P}^T .

该极限理论解释了马尔科夫链的长期行为不依赖于其初始状态, 并且该理论是算法收敛的基础.

引理 6. 如果变异概率 $m > 0$, 则 MX-PSO 算法是一个遍历且不可约的马尔科夫链, 其中算法只有一个有限分布且与初始分布无关. 另外, 在任意随机时刻和任意随机状态中, 变异概率是大于 0 的.

证明. 在时刻 t , 状态 j , 种群的概率分布 $X(t)$ 是

$$P_j(t) = \sum_{j \in S} P_i(1) \cdot P_{ij}^{(t)}, \quad t = 1, 2, \dots \quad (11)$$

根据定理 2, 可得如下公式.

$$P_j(\infty) = \lim_{t \rightarrow \infty} \left(\sum_{i \in S} P_i(1) \cdot P_{ij}^{(t)} \right) = \sum_{i \in S} P_i(1) \cdot P_{ij}^{(\infty)} > 0, \quad \forall j \in S \quad (12)$$

\square

定义 7. 假设一个随机变量 $Z_t = \max\{f(x_k^{(t)}(i)) | k = 1, 2, \dots, N\}$ 表示种群在状态 i 步骤 t 的粒子历史最优方案, 则算法收敛于全局最优方案, 当且仅当

$$\lim_{t \rightarrow \infty} P\{Z_t = Z^*\} = 1 \quad (13)$$

其中, $Z^* = \max\{f(x) | x \in S\}$ 表示全局最优方案.

定理 4. 对于任意 i 和 j , 一个可遍历的马尔科夫链从状态 i 至状态 j , 其转移概率矩阵是正定的.

证明. 在定理 2 中, 我们已证明在算法搜索过程中, 种群通过一个粒子自身的变异操作和两个交叉操作 (粒子与历史最优方案及全局最优方案的交

叉操作) 从 $i_i \in S$ 转移到状态 $i_{i+1} \in S$, 其转移概率矩阵为正定的, 即 $P_{i,i+1}$ 是正定的. 因此假设 $i < j$, 则 $P_{i,j} = \sum_{i=1}^j P_{i,i+1} > 0$, 即 $P_{i,j}$ 是正定的. 所以, 对于任意 i 和 j , 一个可遍历的马尔科夫链从状态 i 至状态 j , 其转移概率矩阵是正定的. \square

定理 5. MX-PSO 算法可收敛于全局最优方案.

证明. 假设对于 $i \in S, Z_t < Z^*$, 且 $P_i(t)$ 是 MX-PSO 算法在状态 i 步骤 t 的概率矩阵, 显然 $P\{Z_t \neq Z^*\} \geq P_i(t)$, 因此可知 $P\{Z_t = Z^*\} \leq 1 - P_i(t)$.

根据引理 6, 在 MX-PSO 算法中状态 i 的操作算子的概率是 $P_i(\infty) > 0$, 则

$$\lim_{t \rightarrow \infty} P\{Z_t = Z^*\} \leq 1 - P_i(\infty) < 1 \quad (14)$$

给定一个新种群, 例如 $X_t^+ = \{Z_t, X_t\}, t \geq 1$, 其中, $x_{ti} \in S$ 表示搜索空间 (是有限集合或可数集合), 跟定义 7 一样的 Z_t 表示当前种群的个体最优值, X_t 表示在搜索过程中的种群, 可证明群体转移过程 $\{X_t^+, t \geq 1\}$ 仍是一个齐次的且可遍历的马尔科夫链, 从而得到

$$P_j^+(t) = \sum_{i \in S} P_i^+(1) \cdot P_{ij}^+(t) \quad (15)$$

其中, $P_{ij}^+ > 0 (\forall i \in S, \forall j \in S_0), P_{ij}^+ = 0 (\forall i \in S, \forall j \notin S_0)$.

因此

$$\begin{aligned} (P_{ij}^+)^t &\rightarrow 0 (t \rightarrow \infty) \\ P_j^+(\infty) &\rightarrow 0 (j \notin S_0) \end{aligned} \quad (16)$$

综上, $\lim_{t \rightarrow \infty} P\{Z_t = Z^*\} = 1$. 故 MX-PSO 算法可收敛于全局最优方案. \square

3 算法仿真与结果分析

3.1 基准电路的参数说明及其初始阶段的布线结果

表 4 和表 5 分别给出了两类基准电路 ISPD98 和 ISPD07 的相关电路规模信息. 表 4 和表 5 中第 1~3 列分别给出基准电路名称、网格大小、线网个数, 其中 ISPD98 线网的规模大小从 11k 到 64k, 在初始布线阶段进行线网分解后, 两端线网的个数是从 25k 至 169k, ISPD07 线网的规模大小从 219k 到 551k, 在初始布线阶段进行线网分解后, 两端线网的个数是从 560k 至 936k. 可见在 ISPD07 的电路规模远超过 ISPD98, 呈数量级的增长, 使得布线问题变得更复杂. 表 4 和表 5 中 bf.nets 表示在初始布线阶段前需要进行布线的两端线网数目, 而 af.nets 表示在初始布线阶段后已经布通的两端线网数目. 表 4 的第 6 列和表 5 的第 7 列说明在初始布线阶段后已经布通的两端线网数目占所有两端线网

数目的百分比分别为 72.98% 和 76.65%。因此, 两类基准电路在初始布线阶段后均可获得一个近似的布线拥挤分布情况。表 5 中 e3.af.nets 和 e3.Ratio 分别表示采用 E3 策略后初始布线阶段后已经布通的两端线网数目以及占有所有两端线网的比例。

3.2 相关加强策略的优化情况对比

为了验证本文所提三个加强策略的有效性, 在第 2.2.1 节~第 2.2.3 节中, 已经分别给出了在加入和未加入这三个策略的实验对比, 相关实验结果如表 1~表 3 所示, 均表明加入这三个策略的有效性。

表 5 给出了采用 E3 策略, ML-XGRouter 在初始布线阶段的布通率为 76.28%, 相对于未采用 E3 策略的情况下 76.65% 的布通率只是略微减少, 也获得一个近似的布线拥挤分布情况, 从而有助于主阶段对拥挤情况的合理判断。因此, E3 策略的加入, 对 E1 和 E2 两种策略有较大的促进作用, 且对初始布线的布通率影响不大。

为了进一步验证 E3 策略的有效性, 表 6 给出了在 E3 策略的基础上采用 E1 或 (和) E2 策略三

种情况下相对未采用任何策略的算法优化情况。表 6 的 4~5 列给出 E3 和 E1 策略的结合, 相对于未采用加强策略的布线算法在溢出数和线长总代价分别减少 32.37% 和 1.09%。表 6 的 6~7 列给出 E3 和 E2 策略的结合, 相对于未采用加强策略的布线算法在溢出数方面带来 77.03% 的巨大优化率, 而只付出了 0.41% 的线长总代价。表 6 的 8~9 列给出 E3、E1 和 E2 三种策略的结合, 相对于未采用加强策略的布线算法在溢出数方面带来 78.36% 的巨大优化率, 同时也优化 0.76% 的线长总代价。综上, E3 策略可更好地加强 E1 和 E2 策略的优化能力, 从而提高了本文算法在溢出数和线长总代价的优化能力。

3.3 在 ISPD98 基准电路上的对比

为了验证本文算法的有效性, 将本文算法 (Ours) 与 5 种总体布线算法^[2-6] 在 ISPD98 基准电路上进行对比。从表 7 看出, 在线长总代价指标上, 本文算法相对这 5 种串行算法^[2-6] 分别取得 6.66%、7.47%、6.95%、7.43% 及 6.61% 的减

表 4 在 ISPD98 测试电路的属性及初始布线的结果

Table 4 ISPD98 benchmark circuits and the results of the initial stage

| 名称 | 网格大小 | 基准电路初始布线情况 | | | | Ratio (%) |
|-------|----------|------------|---------|---------|--------------|-----------|
| | | 线网数 | bf.nets | af.nets | af.nets | |
| ibm01 | 64 × 64 | 11 507 | 25 698 | 20 127 | 78.32 | |
| ibm02 | 80 × 64 | 18 429 | 55 739 | 39 871 | 71.53 | |
| ibm03 | 80 × 64 | 21 621 | 43 272 | 30 181 | 69.75 | |
| ibm04 | 96 × 64 | 26 163 | 49 821 | 34 193 | 68.63 | |
| ibm06 | 128 × 64 | 33 354 | 75 912 | 59 812 | 78.79 | |
| ibm07 | 192 × 64 | 44 394 | 97 391 | 69 123 | 70.97 | |
| ibm08 | 192 × 64 | 47 944 | 121 912 | 89 134 | 73.11 | |
| ibm09 | 256 × 64 | 50 393 | 115 871 | 91 268 | 78.77 | |
| ibm10 | 256 × 64 | 64 227 | 169 391 | 113 459 | 66.98 | |
| AVG | | | | | 72.98 | |

表 5 在 ISPD07 测试电路的属性及初始布线的结果

Table 5 ISPD07 benchmark circuits and the results of the initial stage

| 名称 | 网格大小 | 线网数 | 基准电路初始布线情况 | | | | Ratio (%) | e3.Ratio (%) |
|-----|-------------|---------|------------|-----------|------------|--------------|--------------|--------------|
| | | | bf.nets | af.nets | e3.af.nets | af.nets | | |
| 11 | 324 × 324 | 219 794 | 560 406 | 430 376 | 428 383 | 76.8 | 76.44 | |
| 12 | 424 × 424 | 260 159 | 600 317 | 452 931 | 450 048 | 75.45 | 74.97 | |
| 13 | 774 × 779 | 466 295 | 1 093 066 | 829 270 | 825 112 | 75.87 | 75.49 | |
| 14 | 774 × 779 | 515 304 | 1 075 147 | 783 478 | 782 257 | 72.87 | 72.76 | |
| 15 | 465 × 468 | 867 441 | 1 664 608 | 1 319 629 | 1 307 906 | 79.28 | 78.57 | |
| 16 | 399 × 399 | 331 663 | 657 467 | 528 243 | 527 043 | 80.35 | 80.16 | |
| 17 | 557 × 463 | 463 213 | 956 459 | 743 183 | 738 101 | 77.7 | 77.17 | |
| 18 | 973 × 1 256 | 551 667 | 936 162 | 700 745 | 699 252 | 74.85 | 74.69 | |
| AVG | | | | | | 76.65 | 76.28 | |

表 6 在 E3 的基础上 E1 策略或 (和) E2 策略结合后的优化情况

Table 6 The optimization results with the combination of E1 strategy or (and) E2 strategy based on E3 strategy

| 基准 电路 | EO | | E1+E3 | | E2+E3 | | E1+E2+E3 | |
|----------|------------|------------|--------------|--------------|--------------|---------------|--------------|--------------|
| | TOF | TWL | TOF | TWL | TOF | TWL | TOF | TWL |
| 11 | 85 | 46.7 | 68 | 46.4 | 0 | 47.5 | 0 | 47 |
| 12 | 75 | 45.9 | 49 | 44.4 | 0 | 45.6 | 0 | 45 |
| 13 | 44 | 127.1 | 27 | 123.7 | 0 | 127.4 | 0 | 124.6 |
| 14 | 12 | 110.8 | 7 | 110.2 | 0 | 110.7 | 0 | 110.1 |
| 15 | 32 | 130.8 | 13 | 127.7 | 0 | 129.8 | 0 | 126.8 |
| 16 | 2 | 33.9 | 1 | 33.9 | 0 | 34 | 0 | 33.9 |
| 17 | 0 | 63.4 | 0 | 63 | 0 | 64.2 | 0 | 63.1 |
| 18 | 3 779 | 80 | 3 226 | 80.5 | 3 165 | 81 | 2 762 | 81.1 |
| Comp. | 100 | 100 | 67.63 | 98.91 | 22.97 | 100.41 | 21.64 | 99.24 |

少率. 因为所有算法均未产生溢出边, 所以关于溢出数的实验数据均为 0, 未在表 7 中列出. 由于本文算法引入了 X 结构布线, 并从初始布线阶段可得到一个近似拥挤区域的预测以方便主阶段拥挤区域的选择, 从全局角度考虑多个线网的布线问题, 不受线网布线顺序的影响, 同时引入了一系列的加强策略, 使得所设计的布线器可以在避免溢出的情况下取得最佳的线长总代价.

3.4 在 ISPD07 基准电路上的对比

为了进一步验证本文算法的有效性, 将本文算法与 2 种总体布线串行算法^[6,8]在 ISPD07 基准电路上进行实验对比. 这些算法基于曼哈顿结构. 从表 8 可看出, 本文算法在溢出数方面相对文献 [6,8] 均取得 11.40% 的优化效果, 特别是在测试实例 18 上分别取得 91.20% 和 91.22% 的减少率, 有力地提高了芯片的可布性和可制造性. 本文算法在线长总代价相对文献 [6,8] 中 2 种算法分别取得 17.17% 和 15.07% 的减少率. 本文算法相对于串行算法能

够有效减少溢出数和线长总代价的原因包括: 本文算法引入 X 结构并从全局的角度进行总体布线, 所以本文算法具有相对更强的线长优化能力, 且能克服这些串行算法对线网布线顺序的依赖性问题, 从而有效减少溢出数.

为了进一步验证本文算法在 ISPD07 基准电路上的有效性, 将本文算法与 2 种总体布线并行算法^[11-12]进行对比. 从表 9 可看出, 本文算法在溢出数相对文献 [11-12] 分别取得 23.89% 和 23.90% 的优化效果, 特别是在测试实例 16 上, 本文算法可以将溢出数优化到 0, 即测试实例 16 没有溢出. 本文算法在线长总代价相对文献 [11-12] 分别取得 14.70% 和 16.85% 的线长总代价减少率. 本文算法相对于其他并行算法能够有效减少溢出数和线长总代价的原因包括: 1) 本文算法引入 X 结构, 可相对曼哈顿结构具有更强的互连线优化能力; 2) 本文算法采用改进的 PSO 算法求解 MX-ILP 模型, 可解决采用随机

表 7 在 ISPD98 上与 5 种总体布线算法的对比

Table 7 Comparison between our algorithm and five global routing algorithms on ISPD98

| 基准 电路 | 算法 | | | | | 减少率 (%) | | | | | |
|----------|---------|---------|---------|---------|---------|---------|-------------|-------------|-------------|-------------|-------------|
| | [3] | [5] | [2] | [4] | [6] | Ours | [3] | [5] | [2] | [4] | [6] |
| ibm01 | 63 332 | 64 389 | 62 659 | 63 720 | 62 498 | 58 564 | 7.53 | 9.05 | 6.54 | 8.09 | 6.29 |
| ibm02 | 168 918 | 171 805 | 171 110 | 170 342 | 169 881 | 158 383 | 6.24 | 7.81 | 7.44 | 7.02 | 6.77 |
| ibm03 | 146 412 | 146 770 | 146 634 | 147 078 | 146 458 | 136 313 | 6.9 | 7.12 | 7.04 | 7.32 | 6.93 |
| ibm04 | 167 101 | 169 977 | 167 275 | 170 095 | 166 452 | 153 190 | 8.32 | 9.88 | 8.42 | 9.94 | 7.97 |
| ibm06 | 277 608 | 278 841 | 277 913 | 279 566 | 277 696 | 261 606 | 5.76 | 6.18 | 5.87 | 6.42 | 5.79 |
| ibm07 | 366 180 | 370 143 | 365 790 | 369 340 | 366 133 | 341 170 | 6.83 | 7.83 | 6.73 | 7.63 | 6.82 |
| ibm08 | 404 714 | 404 530 | 405 634 | 406 349 | 404 976 | 379 684 | 6.18 | 6.14 | 6.4 | 6.56 | 6.25 |
| ibm09 | 413 053 | 414 223 | 413 862 | 415 852 | 414 738 | 389 281 | 5.76 | 6.02 | 5.94 | 6.39 | 6.14 |
| ibm10 | 578 795 | 583 805 | 590 141 | 585 921 | 579 870 | 541 839 | 6.38 | 7.19 | 8.18 | 7.52 | 6.56 |
| AVE | | | | | | | 6.66 | 7.47 | 6.95 | 7.43 | 6.61 |

表 8 在 ISPD07 上与 2 种串行算法的总体布线算法的对比

Table 8 Comparison between our algorithm and two serial global routing algorithms on ISPD07

| 基准 电路 | [6] | | 算法 [8] | | Ours | | 减少率 (%) | | | | |
|----------|--------|-------|-----------|--------|-------|-------|---------|-------------|--------------|-------------|--------------|
| | TOF | TWL | TOF | TWL | TOF | TWL | TOF | TWL | TOF | TWL | |
| 11 | 0 | 55.9 | 0 | 53.5 | 0 | 47 | 0 | 15.92 | 0 | 12.15 | |
| 12 | 0 | 54 | 0 | 51.69 | 0 | 45 | 0 | 16.67 | 0 | 12.94 | |
| 13 | 0 | 134.4 | 0 | 130.35 | 0 | 124.6 | 0 | 7.29 | 0 | 4.41 | |
| 14 | 0 | 127.8 | 0 | 120.67 | 0 | 110.1 | 0 | 13.85 | 0 | 8.76 | |
| 15 | 0 | 157 | 0 | 154.7 | 0 | 126.8 | 0 | 19.24 | 0 | 18.03 | |
| 16 | 0 | 48.6 | 0 | 45.99 | 0 | 33.9 | 0 | 30.25 | 0 | 26.29 | |
| 17 | 0 | 78.5 | 0 | 74.88 | 0 | 63.1 | 0 | 19.62 | 0 | 15.73 | |
| 18 | 31 390 | 94.9 | 31 454 | 104.28 | 2 762 | 81.1 | 91.2 | 14.54 | 91.22 | 22.23 | |
| AVE | | | | | | | | 11.4 | 17.17 | 11.4 | 15.07 |

表 9 在 ISPD07 上与 2 种并行算法的总体布线算法的对比

Table 9 Comparison between our algorithm and two concurrent global routing algorithms on ISPD07

| 基准 电路 | [11] | | 算法 [12] | | Ours | | 减少率 (%) | | | | |
|----------|--------|--------|------------|-------|-------|-------|---------|--------------|-------------|-------------|--------------|
| | TOF | TWL | TOF | TWL | TOF | TWL | TOF | TWL | TOF | TWL | |
| 11 | 0 | 52.82 | 0 | 54.3 | 0 | 47 | 0 | 11.02 | 0 | 13.44 | |
| 12 | 0 | 51.46 | 0 | 52.9 | 0 | 45 | 0 | 12.55 | 0 | 14.93 | |
| 13 | 0 | 128.92 | 0 | 131 | 0 | 124.6 | 0 | 3.35 | 0 | 4.89 | |
| 14 | 0 | 119.96 | 0 | 124 | 0 | 110.1 | 0 | 8.22 | 0 | 11.21 | |
| 15 | 0 | 153.23 | 0 | 155 | 0 | 126.8 | 0 | 17.25 | 0 | 18.19 | |
| 16 | 228 | 45.58 | 228 | 47 | 0 | 33.9 | 100 | 25.63 | 100 | 27.87 | |
| 17 | 0 | 74.46 | 0 | 77.9 | 0 | 63.1 | 0 | 15.26 | 0 | 19 | |
| 18 | 31 026 | 107.22 | 31 484 | 108.5 | 2 762 | 81.1 | 91.1 | 24.36 | 91.23 | 25.25 | |
| AVE | | | | | | | | 23.89 | 14.7 | 23.9 | 16.85 |

取整方法易产生偏差的情况, 从而最终带来线长总代价的优化; 3) 本文算法引入了一系列的加强策略, 进一步提高在溢出数和线长总代价的优化能力. 同时, ML-XGRouter 首先是为了优化溢出数, 其次优化线长总代价, 跟多层总体布线对性能的需求是一致的, 可以更有效地求解多层总体布线问题.

4 结论

ML-XGRouter 是第一次求解非曼哈顿结构下多层总体布线问题. 本文相对 XGRouter 引入多种加强策略和相关模型, 从而设计出求解该问题的有效算法, 克服了 XGRouter 不能用于多层总体布线问题的求解以及在溢出数和线长总代价这两个重要指标上优化能力的不足. 实验结果表明 E1、E2 和 E3 三种策略对溢出数和线长总代价优化的有效性. 最终的实验结果表明 ML-XGRouter 在两类基准电路上均可取得相对当前总体布线器而言最佳的溢出数和线长总代价. 此外, 时延目前也是 VLSI 芯片设

计中一个非常重要的指标, 未来的工作中我们将基于 Elmore 时延模型进一步研究时延驱动的总体布线算法.

References

- 1 Xu Ning, Hong Xian-Long. *Very large scale integration physical design theory and method*. Tsinghua University Press, 2009.
(徐宁, 洪先龙. 超大规模集成电路物理设计理论与方法. 清华大学出版社, 2009.)
- 2 Zhang Y, Xu Y, Chu C. Fastroute 3.0: a fast and high quality global router based on virtual capacity. In: *Proceedings of IEEE/ACM International Conference on Computer-Aided Design*. New York, USA: IEEE, 2008. 344–349
- 3 Roy A J, 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
- 4 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

- 5 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
- 6 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
- 7 Held S, Muller D, Rotter D, Scheifele R, Traub V, Vygen J. Global routing with timing constraints. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 2018, **37**(2): 406–419
- 8 Dai K R, Liu W H, Li Y L. NCTU-GR: efficient simulated evolution-based rerouting and congestion-relaxed layer assignment on 3-D global routing. *IEEE Transactions on very large scale integration (VLSI) systems*, 2012, **20**(3): 459–472
- 9 Wu T H, Davoodi A, Linderoth J T. GRIP: scalable 3D global routing using integer programming. In: Proceedings of ACM/IEEE Design Automation Conference. New York, USA: IEEE, 2009. 320–325
- 10 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
- 11 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
- 12 Han Y, Ancajas D M, Chakraborty K, Roy S. Exploring high-throughput computing paradigm for global routing. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 2014, **22**(1): 155–167
- 13 Teig S L. The X architecture: not your father's diagonal wiring. In: Proceedings of International workshop on System-level interconnect prediction. New York, USA: ACM, 2002. 33–37
- 14 Dong J, Zhu H L, Xie M, Zeng X. Graph Steiner tree construction and its routing applications. In: Proceedings of IEEE 10th International Conference on ASIC. New York, USA: IEEE, 2013. 1–4
- 15 Hung J H, Yeh Y K, Lin Y C, Huang H H, Hsieh T M. ECO-aware obstacle-avoiding routing tree algorithm. *WSEAS Transactions on Circuits and Systems*, 2010, **9**(9): 567–576
- 16 Liu G G, Chen G L, Guo W Z. DPSO based octagonal steiner tree algorithm for VLSI routing. In: Proceedings of IEEE 5th International Conference on Advanced Computational Intelligence. New York, USA: IEEE, 2012. 383–387
- 17 Ho T Y. A performance-driven X-architecture router based on a novel multilevel framework. *Integration, the VLSI Journal*, 2009, **42**(3): 400–408
- 18 Ho T Y. A performance-driven multilevel framework for the X-based full-chip router. In: Proceedings of International Workshop on Power and Timing Modeling, Optimization and Simulation. Berlin, Heidelberg: Springer, 2008. 209–218
- 19 Hu Y, Jing T, Hong X, Hu X, Yan G. A routing paradigm with novel resources estimation and routability models for X-architecture based physical design. In: Proceedings of International Workshop on Embedded Computer Systems. Berlin, Heidelberg: Springer, 2005. 344–353
- 20 Cao Z, Jing T, Hu Y, Shi Y, Hong X, Hu X, Yan G. DraXRouter: Global routing in X-architecture with dynamic resource assignment. In: Proceedings of the 2006 Asia and South Pacific Design Automation Conference. New York, USA: ACM, 2006. 618–623
- 21 Liu G G, Guo W Z, Li R R, Niu Y Z, Chen G L. XGRouter: high-quality global router in X-architecture with particle swarm optimization. *Frontiers of Computer Science*, 2015, **9**(4): 576–594
- 22 ISPD 2007 Global Routing Contest [Online], available: <http://www.sigda.org/ispd2007/contest.html>, 2007
- 23 Liu G G, Huang X, Guo W Z, Niu Y Z, Chen G L. Multi-layer obstacle-avoiding X-architecture Steiner minimal tree construction based on particle swarm optimization. *IEEE Transactions on Cybernetics*, 2015, **45**(5): 989–1002
- 24 ISPD 1998 Global Routing Benchmark Suite [Online], available: <http://cseweb.ucsd.edu/kastner/research/labyrinth/>, 1998
- 25 Kennedy J, Eberhart R C. A discrete binary version of the particle swarm optimization algorithm. In: Proceedings of IEEE International Conference on Systems, Man, and Cybernetics. New York, USA: IEEE, 1997. 4104–4109
- 26 Parsopoulos K E, Vrahatis M N. Recent approaches to global optimization problems through particle swarm optimization. *Natural Computing*, 2002, **1**(2–3): 235–306
- 27 Pan Q K, Tasgetiren M F, Liang Y C. A discrete particle swarm optimization algorithm for the no-wait flowshop scheduling problem. *Computers & Operations Research*, 2008, **35**(9): 2807–2839
- 28 Guo W Z, Liu G G, Chen G L, Peng S J. A hybrid multi-objective PSO algorithm with local search strategy for VLSI partitioning. *Frontiers of Computer Science*, 2014, **8**(2): 203–216
- 29 Huang X, Guo W Z, Liu G G, Chen G L. FH-OAOS: a fast 4-step heuristic for obstacle-avoiding octilinear Steiner tree construction. *ACM Transactions on Design Automation of Electronic Systems*, 2016, **21**(3): 48
- 30 Huang X, Guo W Z, Liu G G, Chen G L. MLXR: multi-layer obstacle-avoiding X-architecture Steiner tree construction for VLSI routing. *Science China Information Sciences*, 2017, **60**(3): 19102
- 31 Wang Dong-Feng, Meng Li. Performance analysis and parameter selection of PSO algorithms. *Acta Automatica Sinica*, 2016, **42**(10): 1552–1561
(王东风, 孟丽. 粒子群优化算法的性能分析和参数选择. *自动化学报*, 2016, **42**(10): 1552–1561)
- 32 Lv Bai-Quan, Zhang Jing-Jing, Li Zhan-Pei, Liu Ting-Zhang. Fuzzy partial swarm optimization based on filled function and transformation function. *Acta Automatica Sinica*, 2018, **44**(1): 74–86
(吕柏权, 张静静, 李占培, 刘廷章. 基于变换函数与填充函数的模糊粒子群优化算法. *自动化学报*, 2018, **44**(1): 74–86)

- 33 Zhang W, Zhang H, Liu J, Li K, Yang D, Tian H. Weather prediction with multiclass support vector machines in the fault detection of photovoltaic system. *IEEE/CAA Journal of Automatica Sinica*, 2017, 4(3): 520–525
- 34 Han Z, Zhao J, Wang W. An optimized oxygen system scheduling with electricity cost consideration in steel industry. *IEEE/CAA Journal of Automatica Sinica*, 2017, 4(2): 216–222
- 35 Tang Y, Luo C, Yang J, He H. A chance constrained optimal reserve scheduling approach for economic dispatch considering wind penetration. *IEEE/CAA Journal of Automatica Sinica*, 2017, 4(2): 186–194
- 36 Rudolph G. Convergence analysis of canonical genetic algorithms. *IEEE Transactions on Neural Networks*, 1994, 5(1): 96–101
- 37 Lv H, Zheng J, Wu J, Zhou C, Li K. The convergence analysis of genetic algorithm based on space mating. In: *Proceedings of IEEE 5th International Conference on Natural Computation*. New York, USA: IEEE, 2009. 557–562



刘耿耿 福州大学数学与计算机科学学院副教授. 2015 年获得福州大学应用数学博士学位. 主要研究方向为计算智能及超大规模集成电路物理设计算法研究. E-mail: liugenggeng@fzu.edu.cn

(**LIU Geng-Geng** Associate professor at the College of Mathematics and Computer Science, Fuzhou University.

He received his Ph.D. degree from Fuzhou University in 2015. His research interest covers computational intelligence and very large scale integration physical design.)



庄震 福州大学数学与计算机科学学院硕士研究生. 主要研究方向为超大规模集成电路物理设计算法研究.

E-mail: zhuang_zhen@126.com

(**ZHUANG Zhen** Master student at the College of Mathematics and Computer Science from Fuzhou University. His research interest covers very large scale integration physical design.)



郭文忠 福州大学数学与计算机科学学院教授. 2010 年获得福州大学通信与信息系统博士学位. 主要研究方向为计算智能及其应用. 本文通信作者.

E-mail: fzugwz@163.com

(**GUO Wen-Zhong** Professor at the College of Mathematics and Computer Science, Fuzhou University. He received his Ph.D degree in Communication and Information System from Fuzhou University in 2010. His research interest covers computational intelligence and its application.

Corresponding author of this paper.)



陈国龙 福州大学数学与计算机科学学院教授. 2002 年获得西安交通大学计算机科学博士学位. 主要研究方向为计算智能及其应用. E-mail: fzucgl@163.com

(**CHEN Guo-Long** Professor at the College of Mathematics and Computer Science, Fuzhou University. He received his Ph.D degree in Computer Science from Xi'an Jiaotong University in 2002. His research interest covers computation intelligence.)

Science from Xi'an Jiaotong University in 2002. His research interest covers computation intelligence.)