

一种集成电路全局布局混合整数规划模型

解 飞

北京邮电大学理学院, 北京

收稿日期: 2023年5月25日; 录用日期: 2023年7月29日; 发布日期: 2023年8月3日

摘 要

本文提出了一种基于混合整数规划的集成电路全局布局模型, 全局布局问题是将一些电子元件摆放在电路板指定的区域, 使得元件之间没有重叠同时紧凑的排列。与已有的解析布局算法对比, 本文用优化模型精确刻画了半周长线长(Half Perimeter Wire Length, HPWL)目标和无重叠(non-overlap)约束可以得到更精确的布局结果, 避免了光滑近似的非精确解。数值实验表明, 在中等规模的数据集上, 本文的模型可以完全满足无重叠约束的同时, 半周长线长也是较小的, 得到的高质量解可以作为非精确算法的参考解。本文提出的模型在中小规模的布局问题上有较好的效果, 可以用在车载芯片或蓝牙芯片设计等实际问题中, 具有良好的应用前景。

关键词

最优化, 混合整数规划, 集成电路布局

A Mixed Integer Programming Model for Global Placement of Integrated Circuit

Fei Xie

School of Science, Beijing University of Posts and Telecommunications, Beijing

Received: May 25th, 2023; accepted: Jul. 29th, 2023; published: Aug. 3rd, 2023

Abstract

This paper proposes a mixed-integer programming model for global placement of integrated circuit. The global placement problem can be defined as placing electronic components in a designated circuit board, with no overlap and compact arrangement between these components. Compared with existing analytical placement solvers, the model we presented can accurately characterize the Half Perimeter Wire Length (HPWL) objective and the non-overlap constraint, which can obtain more accurate placement results and avoid non-precise solutions based on smooth approxi-

mations. Numerical experiments demonstrate that on medium-sized datasets, the proposed model can fully satisfy the non-overlap constraint while achieving a relatively small HPWL, and the obtained high quality solution can serve as a reference for non-precise algorithms. The proposed model performs well on small to medium-sized placement problems and can be applied to practical design problems such as vehicle-mounted chips or Bluetooth chips, with good application prospects.

Keywords

Optimization, Mixed-Integer Programming, Integrated Circuit Placement

Copyright © 2023 by author(s) and Hans Publishers Inc.

This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

1. 引言

数字经济日益成为国民经济发展重要的方向, 对人们的生产生活都有很多有益的影响。计算机作为数字经济的基础设施在其中起到了重要的作用, 支撑计算机高效运行的关键部件是各种电子元件, 例如 CPU, 存储器, 控制器等。这些电子元件都是由各种类型的晶体管构成, 计算机的运行效率取决于这些电子元件的计算能力, 尤其是取决于 CPU 即通常所说的芯片的计算能力。通常我们将这些电子元件看作是一种集成电路(integrated circuit), 由于国外对我国的技术封锁, 如何自主制造这些精密的电子元件是一个亟待解决的问题。

集成电路设计和制造是一个复杂且漫长的过程, 大致包括系统设计, 逻辑设计, 物理设计, 参数检验, 功能验证, 封装等步骤[1]。本文研究的焦点在于设计流程中的物理设计, 物理设计主要作用是在指定的区域中摆放晶体管即物理单元, 使得电子元件的性能高的同时, 功耗和电路面积尽可能的少。物理设计的主要流程有布图规划, 布局, 布线和版图规划。布图规划是确定布图区域的大小, I/O 模块, 规定宏物理单元的位置, 设计电源线。布局是摆放标准物理单元到合法的位置, 插入填充单元, 进行时序优化, 无重叠优化使得设计目标最优。布线主要是设计时钟网络, 实现时钟同步, 功耗低等目标。版图规划是将布线过程的连线进行通道化(track)和实体化, 使得线长短, 时序功耗低, 设计规则满足。对于布局流程, 又可以分为全局布局(global placement), 合法化(legalization), 详细布局(detailed placement)三个部分[1]。全局布局的作用是确定宏单元和其他物理单元的位置, 优化目标有线长, 功耗, 可制造性等, 约束主要是密度约束。合法化主要是将单元放置在电路板中特定的行中, 确定火线和零线的方向, 保证单元的电路功能正确。详细布局是通过行单元的交流, 插入, 单元反转等, 将前道工序的布局结果进一步优化, 比如去除单元之间的重叠。在设计结束后, 需要将设计图提交工厂进行制造, 制造过程主要是光刻机在电路板上进行蚀刻光照等复杂物理过程, 最终生产出可用的芯片。

本文主要关注集成电路设计中的物理设计, 从中选取布局流程的全局布局步骤作为研究对象。建立了一个混合整数规划模型。用数学模型精确的刻画半周长线长(HPWL)和无重叠约束(non-overlap) [2], 这两个指标的优化会对集成电路的性能有深刻的影响, 更小的线长代表着更满足时序要求且高可制造性, 无重叠则表示芯片的功耗低从而提升芯片的计算性能, 通过求解混合整数规划模型对实际的集成电路布局问题进行求解, 以达到设计出更好的芯片的目的。

2. 背景

本文以 CPU 即通常指代的芯片为例, 说明一下集成电路全局布局的问题和背景, 现有的研究成果。芯片又称为中央处理器。是计算机进行科学计算的主要部件, 现代计算机使用的芯片有大约数十亿个晶体管即物理单元组成以实现复杂和高效的计算, 芯片的面积在 100 mm^2 左右, 功耗在数百瓦。通常来讲, 评价芯片性能优劣的指标有性能(performance), 功耗(power), 面积(area) [2], 好的芯片特征是拥有更好的性能, 更低的功耗和尽可能小的面积, 业界将这三个关键指标称为 PPA 设计目标。

为了达到最佳的 PPA 指标, 需要对芯片中的部件在电路板上进行精密的排列与连接。其中物理单元的排列称为布局(placement), 布局通过对单元的排列得到尽可能紧凑且较少重叠的布局结果。用数学语言表述为, 通过极小化半周长线长(Half Perimeter Wire Length, HPWL)表征紧凑的排列, 通过计算物理单元之间的坐标代数关系, 表征晶体管之间无重叠(non-overlap) [2]。

电路板中的布局问题描述为, 给定版图规格后, 设计者需要根据晶体管的尺寸大小和晶体管的之间的连接, 将晶体管在版图规定的区域内恰当放置, 一般而言, 晶体管为矩形形状。满足晶体管之间无重叠, 最优布线长, 时序, 功耗等性能, 芯片制造商会提供晶体管尺寸文件(nodes file), 晶体管连接情况(netsfile 即线网文件)以让设计者根据这些信息进行布局问题的求解。

已有理论证明, 布局问题是 NP-难问题。国内外有许多关于布局问题的求解算法, 例如有二次规划, 模拟退火等智能算法, 超图划分等方法求解布局问题[1]。近年来解决布局问题的主要方法是基于解析模型的布局算法和基于强化学习的布局算法。

2.1. 基于解析模型的布局算法

以 NTUplace3 [3], ePlace [4]和 DREAMPlace [5]为代表布局求解器都是解析模型来求解布局问题, 这类布局求解器考虑一个光滑化的半周长线长函数, 并且建立一个密度函数来近似表示无重叠约束。对于线长的处理, 这三种算法采用了一些数学方法对半周长线长做光滑近似, NTUplace3 引入 bin (一种特殊的矩形区域)近似无重叠约束, ePlace 引入物理学中的静电场概念, 将无重叠约束用偏微分方程近似。DREAMPlace 借助了 ePlace 的静电场思想, 将布局问题转化为一个机器学习中求损失函数极小值的问题, 通过 Pytorch 这类深度学习框架高效求解布局问题并采用光滑化的方法, 可以使用罚函数法和共轭梯度法求解布局问题。同时, 为了求解更大规模的问题, 这些解析方法会使用多级聚类解聚类框架[11]以减少晶体管的数量规模。近两三年, 学者在光滑模型的基础上, 考虑时序裕量[6]和雾化效应[7]等约束, 进一步扩展模型的使用范围。

从实际应用来看, 光滑化的模型广泛的使用在布局求解器中。但光滑化模型的不足是, 对于只含有 2 个或 3 个引脚的布局问题, 光滑化模型会低估实际的半周长线长, 并且求到的解不能完全去除物理单元之间的重叠。虽然在后续流程可以处理重叠约束, 若可以将关口前移, 在全局布局阶段处理好无重叠约束, 对后续的流程会有一定的增益。目前工业界中也广泛使用解析模型的布局算法或布局求解器, 国内厂商以上海立芯布局求解器 LePlace 和华大九天 PowerPCB 等为代表, 可以实现 7 nm 制程的芯片设计布局求解。这些工业界使用的布局求解器以解析模型为主, 在全局布局阶段得到一个重叠度较低的初始布局结果和半周长线长的下界, 之后通过合法化和详细布局调整单元最终得到合适的布局。

2.2. 基于强化学习的布局算法

2021 年, Mirhoseni 等人在 Nature 上提出用强化学习的方法[8]求解芯片设计中的布局问题, 文章收集了大量的已布局好的芯片文件作为训练数据集, 训练一个强化智能学习体, 对布局结果进行编码学习从而得到一个好的布局智能体。随后再向智能体输入测试文件, 智能体首先对确定一部分单元例如 I/O

模块的位置，随后使用其他布局算法例如 Force-Direct 确定其余物理单元的位置，获得最终的布局结果。

强化学习方法的主要优势在于，该方法本质上求解的是一个多目标优化问题，得到的解不仅在无重叠约束上有改进，在其他的指标例如时序裕量也有所改进。相比于解析方法，模型覆盖的情形更加全面，得到的解会更接近工业界的实际需要。文章中指出，该方法在 6 个小时内可以得到芯片布局结果，其求解质量可以和人类专家相当，即将用在 Google 下一代张量处理器(Tensor Processing Unit, TPU)的设计中。但同时文章也提到，这种方法需要大量的芯片布局数据进行训练，需要巨大的算力和训练数据支撑，模型训练代价高昂且智能体对于不同类型芯片的泛化能力仍有不确定因素。

3. 一种布局问题的混合整数规划模型

3.1. 模型提出

本文使用混合整数规划模型刻画芯片设计中的全局布局问题，基本思路是用绝对值函数来表达半周长线长，并通过引入一些 0-1 变量精确表达单元之间无重叠的约束。该模型是对全局布局问题的精确描述，因此相比光滑化的近似模型可以得到更精确的解，在数学上也具有更好的可解释性。同时可以为非精确的算法提供一组参考解。

3.2. 模型建立

3.2.1. 符号说明

芯片或者集成电路全局布局问题可以用超图来建模，记 $H(V, E)$ 表示一个布局问题，记布局区域的物理单元指标集合 $I = \{1, 2, \dots, n\}$ 。其中 $V = \{v_i | i \in I\}$ 表示物理单元的集合(下称此集合为 cells，集合中的每个元素称为单元)， $E = \{e_i | i \in I\}$ 是线网(nets)的集合，这里的线网记录了单元之间的连接，可以看作超图的一条超边。记布局区域的宽和高分别为 W 和 H ，记集合 $W = \{w_i | i \in I\}$ 和 $H = \{h_i | i \in I\}$ 分别是集合 cells 元素中的宽和高。记 $V_x = \{x_i | i \in I\}$ 和 $V_y = \{y_i | i \in I\}$ 是各个单元的中心点坐标。

布局问题的数学模型可以用模型(1)表达，其中的约束就是对无重叠情形的精确描述，即单元之间在横坐标上或纵坐标没有重叠，数学上表示为两个单元的横坐标或纵坐标之差的绝对值大于两个单元宽或高之和的一半。

$$\begin{aligned} \min HPWL(x, y) &:= \sum_{e \in E} \max_{v_i, v_j \in e} \{|x_i - x_j| + |y_i - y_j|\} \\ \text{s.t. } &|x_i - x_j| \geq w_{ij} \text{ or } |y_i - y_j| \geq h_{ij}, i \neq j \in I \\ &(x, y) \in X \times Y \end{aligned} \quad (1)$$

其中， $w_{ij} = \frac{w_i + w_j}{2}$ ， $h_{ij} = \frac{h_i + h_j}{2}$ ， X, Y 是布局边界约束。

$$\begin{aligned} X &:= \left\{ x \in \mathbb{R}^n \mid \frac{w_i}{2} \leq x_i \leq W - \frac{w_i}{2}, i \in I \right\} \\ Y &:= \left\{ y \in \mathbb{R}^n \mid \frac{h_i}{2} \leq y_i \leq H - \frac{h_i}{2}, i \in I \right\} \end{aligned}$$

3.2.2. 优化目标

从模型(1)出发，为了去掉目标函数的绝对值运算，引入两个连续变量 α_e, β_e 将目标函数重写，并且为了和(1)等价，需要在约束中加入两条额外的约束表示极小化目标的含义。

$$\begin{aligned}
& \min \sum_{e \in E} (\alpha_e + \beta_e) \\
& \text{s.t. } -\alpha_e \leq x_i - x_j \leq \alpha_e, \forall (i, j) \in e, \forall e \in E \\
& -\beta_e \leq y_i - y_j \leq \beta_e, \forall (i, j) \in e, \forall e \in E \\
& |x_i - x_j| \geq w_{ij} \text{ or } |y_i - y_j| \geq h_{ij}, i \neq j \in I \\
& (x, y) \in X \times Y
\end{aligned} \tag{2}$$

以第一条约束为例, $x_i - x_j$ 被 α_e 和 $-\alpha_e$ 控制, 因此当 α_e 取最小值时, 也就意味着 $|x_i - x_j|$ 取到了最小值, 第二条约束同理可得。

3.2.3. 问题约束

同样从模型(1)出发, 为了去掉约束中的绝对值符号和两个约束的逻辑 or 运算, 可以使用大 M 法并引入一些 0-1 变量处理。首先对无重叠约束进行如下处理,

$$\frac{|x_i - x_j|}{w_{ij}} := |X_{ij}| \geq 1 \text{ or } \frac{|y_i - y_j|}{h_{ij}} := |Y_{ij}| \geq 1 \tag{3}$$

引入两个 0-1 变量 $p_{ij}, q_{ij} \in \{0, 1\}$, 使得上面带有逻辑算子 or 的约束有一个成立, 具体表达式如下。

$$\begin{aligned}
& -(M_{ij} + 1)(p_{ij} + q_{ij}) + 1 \leq X_{ij} \leq -(M_{ij} + 1)(p_{ij} + q_{ij}) + 2M_{ij} + 1 \\
& -(N_{ij} + 1)(q_{ij} - p_{ij}) - N_{ij} \leq Y_{ij} \leq -(N_{ij} + 1)(q_{ij} - p_{ij}) + N_{ij}
\end{aligned} \tag{4}$$

下面对 p_{ij}, q_{ij} 的值列举来证明(3)和(4)在表达上是等价的。

$$\begin{aligned}
& (p_{ij}, q_{ij}) = (0, 0) \rightarrow 1 \leq X_{ij} \leq 2M_{ij} + 1, -N_{ij} \leq Y_{ij} \leq N_{ij} \\
& (p_{ij}, q_{ij}) = (0, 1) \rightarrow -M_{ij} \leq X_{ij} \leq M_{ij}, -2N_{ij} - 1 \leq Y_{ij} \leq -1 \\
& (p_{ij}, q_{ij}) = (1, 0) \rightarrow -M_{ij} \leq X_{ij} \leq M_{ij}, 1 \leq Y_{ij} \leq 2N_{ij} + 1 \\
& (p_{ij}, q_{ij}) = (1, 1) \rightarrow -2M_{ij} - 1 \leq X_{ij} \leq -1, -N_{ij} \leq Y_{ij} \leq N_{ij}
\end{aligned} \tag{5}$$

以第一个约束为例, 当 $(p_{ij}, q_{ij}) = (0, 0)$ 时, $|X_{ij}| \geq 1$ 成立, 通过观察其他三个表达式, 可以看到 (p_{ij}, q_{ij}) 的取值覆盖了所有可能的情形, 并且与(3)的约束等价。

3.2.4. 数学模型

综上所述可以建立如下的混合整数规划模型。

$$\begin{aligned}
& \min W(x, y) = \sum_{e \in E} (\alpha_e + \beta_e) \\
& \text{s.t. } -\alpha_e \leq x_i - x_j \leq \alpha_e, \forall (i, j) \in e, \forall e \in E \\
& -\beta_e \leq y_i - y_j \leq \beta_e, \forall (i, j) \in e, \forall e \in E \\
& -(M_{ij} + 1)(p_{ij} + q_{ij}) + 1 \leq \frac{x_i - x_j}{w_{ij}} \leq -(M_{ij} + 1)(p_{ij} + q_{ij}) + 2M_{ij} + 1, \forall i \in V, \forall j \in V \\
& -(N_{ij} + 1)(q_{ij} - p_{ij}) - N_{ij} \leq \frac{y_i - y_j}{h_{ij}} \leq -(N_{ij} + 1)(q_{ij} - p_{ij}) + N_{ij}, \forall i \in V, \forall j \in V \\
& p_{ij}, q_{ij} \in \{0, 1\}
\end{aligned} \tag{6}$$

考虑到实际问题中 M_{ij}, N_{ij} 的有界性, 可以取恰当的充分大的值, 例如令 $M_{ij} = W, N_{ij} = H$ 。

4. 数值实验

在数值实验中，我们在 Linux 操作系统中使用 C++ 编写数值计算程序，并使用 CPLEX 作为混合整数规划求解器。实验选取的数据集来自密歇根大学维护的 GSRC [9] 和 MCNC [10] 数据集。我们对比了本文提出的混合整数规划模型和光滑化模型得到效果。实验结果表明，光滑化模型得到的线长虽然较小，但是没有完全去除单元之间的重叠，然而本文的模型在和光滑模型线长相当的情形下，可以做到完全没有重叠，从而得到了一个质量更好的解。

4.1. 数据集说明

GSRC 和 MCNC 数据集是由密歇根大学维护的一个电路板布局数据集。我们以 GSRC 中的 HARD 文件夹下的 n10 算例做说明。首先这个算例包含两个文件 n10.blocks 和 n10.nets。blocks 文件存储了单元的信息，其中一些称为宏单元的模块存储了该模块 4 个位置的坐标，其余的单元我们令其宽为 1 高为 1 的正方形，这个文件就是前述的尺寸文件(nodes file)。我们通过编写 Python 程序进行预处理可以计算出这些宏单元的宽高，生成新的 blocks 文件。第二个是 nets 文件，这个文件存储了单元之间的连接。一个 net 即线网包含了若干个单元，同属于一个 net 的单元互相之间都有连接，从数学的角度看，每一个 net 都是超图中的一条超边。这两个数据集分为 HRAD 和 SOFT 两部分，HARD 部分意味着宏单元大小是不可变的，实验中我们使用 HARD 部分的数据。

4.2. 数值结果

图 1 是使用光滑化模型的实验结果，我们选取了 n10 算例，这个算例包含 79 个单元(cells)，其中有 10 个宏单元(blocks)，118 个线网(nets)。光滑模型我们参考文献[3]的方法，使用罚函数方法和共轭梯度法求解。图 1 的横轴表示布局区域的横坐标，纵轴表示布局区域的纵坐标。其中数字代表布局区域的大小，并表示了单元的位置坐标。单元的摆放从(0, 0)开始，但为了作图方便，布局结果图左下端的坐标从(-100, -100)开始。从图 1 中看到，光滑模型得到的半周长线长为 19,233，但是图中的单元之间仍有重叠，尤其是宏单元之间重叠度较高。虽然可以在详细布局等后续流程中得到缓解，但是从光滑模型的近似来看，这个布局结果不是一个高质量的解。

图 2 是使用本文提出的混合整数规划模型，其中横纵坐标的含义与图 1 相同，数字含义也与图 1 相同。该模型得到的半周长线长为 19703，略高于光滑模型的线长，但是从图中看到各个单元之间完全没有重叠，是一个满足约束条件的同时，单元之间尽可能紧凑排列的高质量解。不过使用 CPLEX 求解混合整数规划模型的时间要比光滑化模型的时间长，虽然得到了高质量的布局结果，但是求解时间还有待进一步改进。

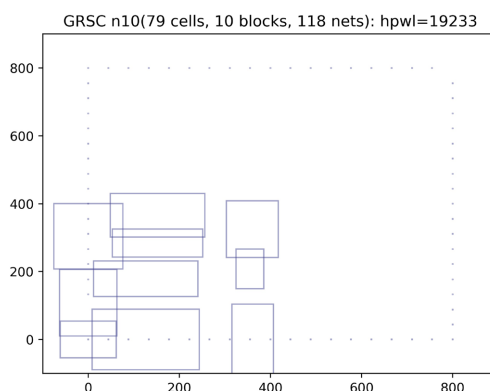


Figure 1. Smooth model result on n10 benchmark
图 1. 光滑化模型求解 n10 算例

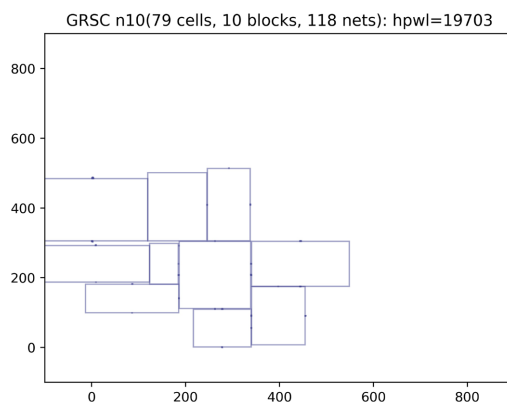


Figure 2. Mixed-integer programming model result on n10 benchmark

图 2. 混合整数规划模型求解 n10 算例

5. 结论

本文提出了一种集成电路全局布局的混合整数规划模型，用优化模型的方法精确刻画了布局问题的关键要素，即半周长线长和无重叠约束。在实际应用中，可以获得高质量的布局结果，相比于已有的解析型布局算法。本文提出的模型可以在全局布局阶段完全满足无重叠约束，对芯片设计的后续流程有增益作用。同时，模型还可以为非精确的算法提供参考解，帮助已有的解析布局器提高求解质量，本文提出的模型还可以与多级聚类解聚类[11]等框架结合求解大规模的集成电路布局问题。数值实验表明，本文提出的模型可以得到高质量的解，可以有效求解中小规模的集成电路布局问题。

参考文献

- [1] 黄志鹏, 李兴权, 朱文兴. 超大规模集成电路布局的优化模型与算法[J]. 运筹学学报, 2021, 25(3): 15-36.
- [2] Kahng, A.B., Lienig, J., Markov, I.L. and Hu, J. (2011) VLSI Physical Design: From Graph Partitioning to Timing Closure. Springer, Dordrecht. <https://doi.org/10.1007/978-90-481-9591-6>
- [3] Chen, T.-C., Jiang, Z.-W., Hsu, T.-C., Chen, H.-C. and Chang, Y.-W. (2008) NTUplace3: An Analytical Placer for Large-Scale Mixed-Size Designs with Preplaced Blocks and Density Constraints. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, **27**, 1228-1240. <https://doi.org/10.1109/TCAD.2008.923063>
- [4] Lu, J., Chen, P., Chang, C.-C., Sha, L., Huang, D.J.-H., Teng, C.-C. and Cheng, C.-K. (2005) EPlace: Electrostatics-Based Placement Using Fast Fourier Transform and Nesterov's Method. *ACM Transactions on Design Automation of Electronic Systems (TODAES)*, **20**, 1-34. <https://doi.org/10.1145/2699873>
- [5] Lin, Y., Dhar, S., Li, W., Ren, H., Khailany, B. and Pan, D.Z. (2019) DREAMPlace: Deep Learning Toolkit-Enabled GPU Acceleration for Modern VLSI Placement. *Proceedings of the 56th Annual Design Automation Conference 2019*, Las Vegas, 2-6 June 2019, 1-6. <https://doi.org/10.1145/3316781.3317803>
- [6] Guo, Z. and Lin, Y. (2022) Differentiable-Timing-Driven Global Placement. *Proceedings of the 59th ACM/IEEE Design Automation Conference*, San Francisco, 10-14 July 2022, 1315-1320. <https://doi.org/10.1145/3489517.3530486>
- [7] Chen, J., Chang, Y.-W. and Huang, Y.-C. (2020) Analytical Placement Considering the Electron-Beam Fogging Effect. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, **40**, 560-573. <https://doi.org/10.1109/TCAD.2020.3002570>
- [8] Mirhoseini, A., Goldie, A., Yazgan, M., et al. (2021) A Graph Placement Methodology for Fast Chip Design. *Nature*, **594**, 207-212. <https://doi.org/10.1038/s41586-021-03544-w>
- [9] GSRC Floorplan Benchmarks. <http://vlsicad.eecs.umich.edu/BK/GSRCbench/>
- [10] MCNC Floorplan Benchmarks. <http://vlsicad.eecs.umich.edu/BK/MCNCbench/>
- [11] Chan, T., Cong, J. and Sze, K. (2005) Multilevel Generalized Force-Directed Method for Circuit Placement. *Proceedings of the 2005 International Symposium on Physical Design*, San Francisco, 3-6 April 2005, 185-192. <https://doi.org/10.1145/1055137.1055177>