Per-RMAP：用于带输入/输出分配的布局规划的可行性寻求与优越化方法

Shan Yu[1], Yair Censor[2],Ming Jiang[1] 和 Guojie Luo[3,4]*北京大学数学科学学院信息与计算科学系，北京，中国[2]以色列海法大学数学系，海法，以色列[3]北京大学计算机科学学院多媒体信息处理国家重点实验室，北京，中国[4]北京大学能效计算与应用中心，北京，中国 *yu-shan@pku.edu.cn，yair@math.haifa.ac.il，{ming-jiang、gluo}@pku.edu.cn*

2023年3月31日

arXiv:2304.06698v1 [math.OC] 5 Apr 2023

摘要—**可行性寻求方法为处理和解决连续问题中的复杂约束提供了一套系统方案；本文将其用于约束日益异构的布局规划问题。经典的合法性约束可以表述为凸集的并集。然而，当约束集合非凸时（凸集的并集即属此类），传统基于投影的算法无法保证收敛。本文提出一种重置策略，大幅消除可行性寻求表述中基于投影的算法所面临的发散问题。此外，我们首次将介于可行性寻求与约束优化之间的优越化方法（SM）应用于布局规划。SM 利用扰动引导可行性寻求算法的迭代过程，找到总线长更短的可行解。所提出的算法流程可扩展至处理各种约束和不同类型的布局规划问题，例如带输入/输出分配的布局规划问题。我们在 MCNC 基准测试上评估了所提出的算法。使用 MATLAB 的当前原型仅比分支定界法慢两倍，即可获得合法布局规划结果，且线长仅比最优结果差 3%。我们通过考虑输入/输出分配约束来评估该算法流程的有效性，结果表明我们的算法在线长方面提升了 8%。**

索引词—**可行性寻求、优越化方法、投影算法、布局规划、输入/输出分配。**

I. 引言

布局规划是 VLSI 物理设计流程中的关键阶段，因为它会影响下游阶段的质量。它可以描述为：将给定的一组矩形模块[1]放置在一个矩形区域内，确保模块之间不存在重叠，同时尽量减少线长、拥塞、温度等。这是一个难题 [1],，为生成更有效的集成电路，可以考虑众多不同的条件，例如边界条件、不重叠条件和输入/输出分配条件。**可行性寻求问题（FSP）**旨在为一族（通常是有限个）约束集合寻找约束相容点。同时，正如下面将进一步解释的那样，在对同时包含约束和目标函数的问题进行建模和求解时，FSP 能够采用优越化方法；该方法奉行“满意化”而非优化的理念。

最先进的布局规划方法大致可分为四类：元启发式方法、精确方法（例如分支定界（B&B）方法）、解析方法和基于学习的方法。启发式方法和分支定界方法首先采用一种表示形式，例如序列对 [2] 和 B^*-树 [3], 然后在表示空间中搜索最优或次优解。像 [4] 这样的启发式方法会进行启发式搜索，并在满足特定条件时停止；而精确方法则会在整个搜索空间中搜索，以找到最优解。分支定界方法是精确方法中最重要的一类，它们构建一棵有根决策树来枚举搜索空间，并通过剪除无用分支缩减搜索空间，参见例如 [5]。解析方法将布局规划建模为一个带有二次[6] 或非线性 [7] 目标函数的优化问题。通常，这类方法会采用全局布局规划步骤，确定所有模块的大致位置，再通过合法化步骤消除重叠并确定所有模块的准确位置。近年来，基于学习的算法，尤其是强化学习，逐渐流行起来。GoodFloor-plan [8]将图卷积网络与强化学习相结合，以探索设计空间。Liu 等人 [9] 利用图注意力学习电路连通性与物理线长之间的优化映射，并通过高效的模型推理生成芯片布局规划。

与上述方法相比，本文探讨的 FSP公式采用了一些强大的工具来处理具有众多且多样化约束的问题。启发式方法和人工智能方法需要精心设计，这可能会对处理多样化约束造成不利影响。精确方法的搜索空间可能非常庞大。解析方法则难以在约束与目标之间实现权衡。相比之下，FSP 着眼于可行性，从而简化了问题。此外，迭代投影方法是 FSP 中常用的工具，运行快速、易于实现，并能处理布局规划中的复杂约束集合。其算法流程便于处理扩展约束，例如，本文在基本布局规划的基础上纳入了输入/输出分配，并以可承受的时间成本实现了总线长降低 8%。

此外，**优越化方法**（SM）可用于寻找更优的解。它利用扰动引导可行性寻求算法，使其输出一个可行的结果，并改善其中某个目标函数。关于 SM 的一篇最新教程 [10] 列举了该主题下各种近期研究成果和资料的链接。Blake Schultze 等人 [11] 将全变分（TV）优越化应用于质子计算机断层成像中的图像重建，以改善结果。受其方法启发，我们提出了一种用于布局规划的线长优越化FSP算法。

简而言之，我们的贡献包括：1) 我们将布局规划表述为可行性寻求问题，重点关注可行性，从而简化了问题。2) 我们提出了一种采用重置策略的广义投影方法，以处理布局规划中的复杂约束集合。

[1]布局规划涉及两类模块：硬模块和软模块。软模块的面积固定，但高度和宽度可变；硬模块的高度和宽度则固定。本文仅考虑硬模块。未来工作中，我们可以扩展该方法以处理软模块。

这种重置策略改善了投影法的初始收敛行为。

3) 我们采用线长优越化方法，以找到总线长更短的优越可行解。

4) 我们的算法流程展现出应对多种约束的潜力。考虑布局规划中的输入/输出分配后，我们的流程在可接受的时间内使总线长比 B&B 方法[2] 减少了 8%。

本文其余部分安排如下：第二节结合布局问题介绍可行性寻求公式化。第三节介绍我们提出的方法，称为扰动可重置交替投影法（Per-RMAP）。第四节C给出实验结果，随后在第五节作总结。

II. 将布局规划视为可行性问题 M

本节将布局问题公式化为一个可行性寻求问题，即在有限个集合的交集中寻找一个点，并介绍一种用于寻找解的投影法。

A. 布局问题的设计准则

带输入/输出分配的布局规划问题，是指在减少模块之间连接的总线长的同时，确定矩形布局区域内模块的位置，以及布局区域边界上输入/输出引脚的位置。

布局问题的解应满足以下条件。

I边界：每个模块都位于给定的布局区域内；

II **不重叠**：任意一对模块都不重叠。

引脚是模块边界和布局规划区域边界上连线的位置，而网络表示需要进行电气连接的引脚集合。

将布局问题表述为可行性寻求问题时，其解通常并不唯一。我们希望找到满足以下附加条件的解。

III **线长**：以半周长线长（HPWL）衡量的网络总线长应尽可能短。

设 $\mathcal{P}$ 为引脚集合。第 p 个引脚 $P_p \in \mathcal{P}$ 是位于 $(x_p^{pin},\ y_p^{pin})$ 的点。令 $\mathcal{E}$ 表示来自 $\mathcal{P}$、连接各引脚的连线所组成的网络。那么，布局规划的 HPWL 为

$$\mathrm{HPWL}(x,y) := \sum_{e\in\mathcal{E}} \Big(\max_{p,q\in e} |x_p^{pin} - x_q^{pin}| + \max_{p,q\in e} |y_p^{pin} - y_q^{pin}|\Big). \tag{1}$$

上述条件 I、II 和 III 是本文针对布局问题提出的设计准则。当然，实际应用中还应考虑其他设计准则，例如电源分配和模块密度等。我们认为，这些额外准则也可以表述为可行性约束，但将留待未来研究。

[2]作为比较对象的分支定界法不考虑输入/输出分配。若在分支定界法中考虑额外变量或约束，则必须重新定义并实现搜索空间和剪枝策略。相比之下，可行性寻求法能够轻松适应新的公式化。

B. 布局问题的可行性寻求公式化

为得到布局问题的可行性寻求公式化，我们需要引入更多符号。布局区域是一个二维矩形，左下角位于原点 (0，0)，右上角位于 $(W,\ H)$，其中 W 和 H 是给定的正实数。设 $\mathcal{M}$ 为要放置到给定布局区域中的模块集合。对于第 i个模块 $m_i \in \mathcal{M}$，其宽度和高度分别为 w_i 和 h_i；其“坐标”是其左下角的位置，记为 $(x_i,\ y_i)$，该位置需要确定。假设引脚 $P_p \in \mathcal{P}$ 属于模块 m_i，其位置 $(x_p^{pin},\ y_p^{pin}) = (x_i + x_p^{offset},\ y_i + y_p^{offset})$由 $(x_i,\ y_i)$ 及硬模块的固定引脚偏移量 x_p^{offset}和 y_p^{offset}确定。设 $\mathcal{P}_{io} \subset \mathcal{P}$ 为位于布局区域边界上的输入/输出引脚集合。考虑输入/输出分配时，需要确定坐标为 $(x_p^{pin},\ y_p^{pin})$的输入/输出引脚 $P_p \in \mathcal{P}_{io}$ 的位置。设 $\mathcal{M}$ 中的模块总数为 N_m，输入/输出引脚总数为 N_{io}。设 $N = N_m + N_{io}$ 为待放置的模块和输入/输出引脚总数。

Let $z = (x, y) \in \mathbb{R}^{2N}$ with

$$x = (x_1, \cdots, x_N), \tag{2}$$
$$y = (y_1, \cdots, y_N), \tag{3}$$

通过依次堆叠来自 $\mathcal{M}$ 的模块的 x 坐标，以及来自 $\mathcal{P}_{io}$ 的输入/输出引脚的 y 坐标。此堆叠操作建立了从模块和输入/输出引脚的坐标到 $\mathbb{R}^{2N}$ 的单射线性映射。因此，模块 $\mathcal{M}$ 和引脚 $\mathcal{P}$ 的坐标有两种表示方式。本文按照堆叠约定使用这两种表示方式。$\mathbb{R}^{2N}$ 中的表示用于建立可行性寻求公式化并介绍可行性寻求算法，而以 $(x_i,\ y_i)$ 形式表示的二维表示则用于实现。

对于边界条件 I，当 $1 \le i \le N_m$ 时，令

$$B_i^x(z) := \{z \in \mathbb{R}^{2N} \,|\, 0 \le x_i \le W - w_i\}, \tag{4}$$
$$B_i^y(z) := \{z \in \mathbb{R}^{2N} \,|\, 0 \le y_i \le H - h_i\}. \tag{5}$$

如果模块 m_i 和模块 m_j 均位于布局区域内，则必须满足以下条件，

$$z \in B_{i,j} := B_i^x \cap B_i^y \cap B_j^x \cap B_j^y. \tag{6}$$

对于 $1 \le i,\ j \le N_m$ 和 $i \ne j$。

模块 m_i 与模块 m_j 不重叠这一条件，对于 $1 \le i,\ j \le N_m$ 和 $i \ne j$，等价于以下四个约束之一

$$z \in O_{i,j}^x \iff m_i \text{ is to the left of } m_j, \tag{7}$$
$$z \in O_{j,i}^x \iff m_i \text{ is to the right of } m_j, \tag{8}$$
$$z \in O_{i,j}^y \iff m_i \text{ is below } m_j, \tag{9}$$
$$z \in O_{j,i}^y \iff m_i \text{ is above } m_j, \tag{10}$$

其中

$$O_{i,j}^x(z) := \{z \in \mathbb{R}^{2N} \,|\, x_i + w_i \le x_j\}, \tag{11}$$
$$O_{i,j}^y(z) := \{z \in \mathbb{R}^{2N} \,|\, y_i + h_i \le y_j\}. \tag{12}$$

那么，不重叠条件等价于以下约束

$$z \in O_{i,j} := O_{i,j}^x \cup O_{j,i}^x \cup O_{i,j}^y \cup O_{j,i}^y, \tag{13}$$

对于 $1 \le i,\ j \le N_m$ 和 $i \ne j$。

Let

$$C_{i,j} := O_{i,j} \cap B_{i,j} \tag{14}$$
$$:= \left(O^x_{i,j} \cap B_{i,j}\right) \cup \left(O^x_{j,i} \cap B_{i,j}\right) \tag{15}$$
$$\cup \left(O^y_{i,j} \cap B_{i,j}\right) \cup \left(O^y_{j,i} \cap B_{i,j}\right) \tag{16}$$
$$:= C_{i,j,L} \cup C_{i,j,R} \cup C_{i,j,B} \cup C_{i,j,A}, \tag{17}$$

其中， L、 R、 B和 A分别表示两个模块的相对关系，即左、右、下和上。结合（6）和（13）中的约束，包括**边界**和**不重叠**条件，**布局规划**便成为一个寻求可行性的问题：寻找一点 z ，使得

$$z \in \bigcap_{1 \le i < j \le N_m} C_{i,j}. \tag{18}$$

这是一个典型的可行性寻求问题，即在若干集合的交集中寻找一个点，参见例如 [12]。将布局问题表述为可行性寻求问题时，并未处理**线长**条件。在下文第三节-B2中，我们采用优越化方法来处理**线长**条件。

此外，如果输入/输出引脚的位置允许沿边界移动，则必须施加额外约束。

IV **输入/输出分配**：将输入/输出引脚分配到布局区域相应的边界上。

如果输入/输出引脚 P_{p_i} 位于布局区域的左边界，则

$$D^L_{p_i}(z) := \left\{ z \in \mathbb{R}^{2N} | x^{pin}_{p_i} = 0 \text{ and } 0 \le y^{pin}_{p_i} \le H \right\}. \tag{19}$$

对于位于布局区域右、上、下边界的输入/输出端，也可以构造类似的约束。 $\mathcal{P}_{io} \subset \mathcal{P}$ 是如上所述的输入/输出引脚集合，大小为 N_{io}。**带输入/输出分配的布局规划 FSP 模型**为:

$$\text{Find } z \in \left(\bigcap_{1 \le i < j \le N_m} C_{i,j} \right) \cap \left(\bigcap_{P_p \in \mathcal{P}_{io}} D_p \right). \tag{20}$$

C. 可行性寻求问题的投影法

求解可行性寻求问题的一种方法，是按照预先确定的顺序，反复将点投影到约束族中的各个集合上。点$z \in \mathbb{R}^{2N}$ 到集合 $C \subset \mathbb{R}^{2N}$ 的投影是集合值映射 P_C

$$P_C(z) = \arg\min_{c \in C} \|z - c\|, \tag{21}$$

其中， $\|\cdot\|$ 为欧几里得范数。当 C 为凸集时， $P_C(z)$为单点集。当 C 为非凸集时， $P_C(z)$可能包含多个点，这正是我们的情况，具体如下所述。这就是**交替投影方法（MAP）**，例如参见 [13]。

尽管（18）中的每个集合$C_{i,j}$ 都不是凸集，但通过选择适当的初始猜测以保证算法收敛，仍然可以将MAP 方法应用于（18）。尽管我们目前还无法验证 [14] 中所施加的条件在此处成立，但我们仍然这样做。然而，在实现过程中，我们需要确定选择$P_{C_{i,j}}$（z）中的哪个点来进入下一次迭代。此外，选择哪个集合的顺序

$C_{i,\ j}$ 用于投影的集合也会影响 MAP 的收敛性和性能。算法细节将在第三节介绍和讨论。

III. 扰动可重置交替投影法（PER-RMAP）的提出

本节将介绍算法流程。如图1所示，该流程包含三个阶段：初始化（第三节 A）、全局布局规划（即第三节 B3 中的 Per-RMAP）以及后处理（第三节 C）。

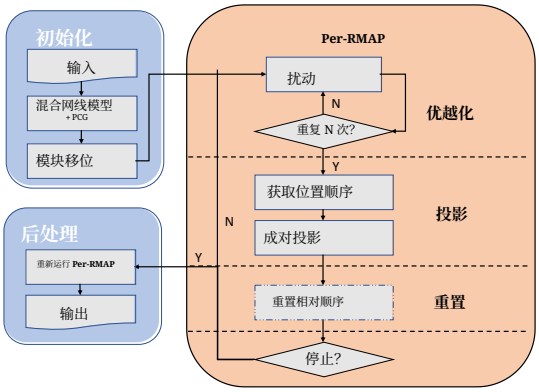

图1. 算法流程

A. 初始化

初始化会影响最终结果。模块位置的初始化过程分为两个步骤。第一步将模块分配到半周长线长最小的位置。第二步调整部分模块的位置，因为这些模块容易受到其他模块的影响。

第一步完全忽略单元重叠，并采用预条件共轭梯度法（PCG）求解二次规划，以实现总线长最小化；这与 SimPL [15] 和 FastPlace [16]等其他力导向布局器类似。网线分解采用混合网线模型 [16] 。该模型结合了经典的团模型和星形模型，既兼顾了速度与精度，也能在网线分解后更好地捕捉相对顺序。然而，如果模块与位于布局区域边界上的输入/输出引脚之间几乎没有连接，模块就容易聚集在一起。

在第二步中，通过移动模块来改进初始化。也就是说，检测可能影响最终结果的关键模块，并将其初始化到布局规划区域的边界。

B. 全局布局规划：*Per-RMAP*

在全局布局规划中，我们设计了一种基于投影的 Per-RMAP 算法，并采用了重置策略和优越化方法。

1) 可重置交替投影方法 *(RMAP)*：MAP（在第二节-C中介绍）用于求解约束集合为凸集的 FSP，而重置策略旨在处理约束集合为凸集并集的情况。为避免陷入不可行解或在多个不可行解之间振荡，我们的思路是将 MAP 推广为一种“启用策略的 MAP”。其核心在于，MAP 总是选择并集中距离当前迭代点最近的子集，而某些选择会阻碍可行解的获得。我们通过并集中各子集之间的“偏好比率”，实现可自定义的选择策略。具体细节见算法 1。迭代过程按特定顺序扫描模块并应用

算法 1 带偏好的广义投影（当其采用重置策略时，$preference\_ratio$ 变为 RMAP）

需要：
初始位置： $z = (x,\ y) \in \mathbb{R}^{2N}$
约束的处理顺序： $order$

确保：
更新后的位置： $z = (x, y)$

对于 $C_{i,j}$ 在 $order$ **do**
  $(\eta_L, \eta_R, \eta_B, \eta_A) = preference\_ratio(i, j)$
  $w_t = \exp(\eta_t/\epsilon)/\sum_k \exp(\eta_k/\epsilon)$，对于 $t \in \{L, R, B, A\}$
  $z = \sum_{t\in\{L,R,B,A\}} w_t \cdot P_{C_{i,j,t}}(z)$
结束 for
return z

成对投影是投影到 4 个凸集上的投影的加权平均。一个关键问题是正确选择要投影到的凸集，即偏好比率的分布。偏好比率表示对四个凸集的选择偏好，并通过指数函数放大。MAP 对偏好比率采用最近点策略，其计算方式为 $\eta_t = -\|z - P_{i,j,t}(z)\|$。设计了一种基于最近点策略的重置策略，该策略在计算偏好比率时考虑了此前的投影行为，如式 (22) 所示，

$$\eta_k = \begin{cases} -\infty, & c_k > T \quad \text{(and reset } c_k = 0\text{)}, \\ -\|z - P_k(z)\|, & otherwise \quad \text{(and set } c_k = c_k + 1\text{)}, \end{cases} \tag{22}$$

其中， T 是预定义的正整数。 c_k 是每个凸集$C_{i,\ j,\ k}$，对于 $k \in \{L, R, B, A\}$，自上次重置以来已被投影到该凸集上的次数。

也就是说，当一对模块反复沿某个方向投影（超过 T 次）却仍未能消除重叠时，就会触发“重置”操作，并在本次迭代中将该方向的优先级比率设为最低，以摆脱振荡或停滞状态。

处理顺序可以是位置顺序，即先处理 x 坐标和 y 坐标较小的模块。这样可以保留大多数模块的相对顺序。

2) 优越化方法（SM）：SM介于可行性寻求与约束优化之间。SM在寻求满足约束条件的同时，会降低目标函数的值，但不一定使其达到最小值。它会主动扰动迭代点，引导其朝可行点移动，并降低目标函数的值。在我们的研究中，我们报告了采用优越化的新型改进方法后总线长有所改善。受现代版优越化方法的启发 [11], ，我们用于改善半周长线长的优越化方法如算法2所示。在迭代 k中，会施加扰动Num次，使模块和输入/输出引脚沿半周长线长函数的负梯度方向移动。扰动的步长参数 $\lambda^{k,\ n}_{pert}$ （第9行）会随迭代次数增加而减小。我们通过考虑以下性质来设计步长参数$\lambda^{k,\ n}_{pert}$：

1) 扰动的步长 $\lambda^{k,\ n}_{pert}$ 必须可求和，即 $\sum_{k=0}^{\infty}\sum_{n=0}^{Num-1}\lambda^{k,\ n}_{pert} < +\infty$。在我们的算法中，步长通过 $\{\Lambda^{\ell}\}_{\ell=0}^{\infty}$ 的一个子序列生成，其中包含用户选择的核 $0 < \Lambda < 1$的 ℓ 次幂。2) 给出一个下界 λ_{min}，以确保扰动的效果。在我们的设定中，$\lambda_{min} = 0.1$。3) 控制优越化中目标函数约简扰动步长的递减。如果步长 $\lambda^{k,\ n}_{pert}$ 递减得过快（即，在上一次迭代扫描的 $\Lambda^{\ell_{k-1}}$ 中， $\ell_{k-1} > k$

算法2优越化方法（SM）

需要：
中间位置： $z = (x,\ y) \in \mathbb{R}^{2N}$
每次迭代的扰动次数：数量
当前迭代次数： k
扰动衰减指数： ℓ_{k-1}
最小扰动长度： λ_{min}
初始扰动长度： λ_{init}
扰动衰减因子： $\Lambda \in (0,\ 1)$

确保：
更新后的位置： z
新的扰动衰减指数： ℓ_k

对于 $n = 1$： 编号 **执行**
  如果 $k < l_{k-1}$，**则**
    $\ell_k = [k, \ell_{k-1}]$ 中的一个随机整数
  else
    $\ell_k = k$
  结束如果
  $v^{k,\ n} = \nabla$半周长线长$(x,\ y)$
  对于 $cnt = 1$： 10 **执行**
    $\lambda^{k,n}_{pert} = max(\lambda_{min},\ \lambda_{init} \cdot \Lambda^{l_k})$
    $(x', y') = z' = z - \lambda^{k,\ n}_{pert} \cdot v^{k,n}/\|v^{k,\ n}\|$
    如果 半周长线长$(x',\ y') <$ 半周长线长$(x,\ y)$ **则**
      $z = z'$
      跳出// 并继续执行第 2 行
    结束 if
    $\ell_k = \ell_k + 1$
  结束 for
结束 for
返回 (z, ℓ_k)

上一次 SM 迭代），那么分配给总线长缩减的杠杆就太小了。因此，扰动衰减指数ℓ_k 是当前迭代索引 k 与上一次迭代扫描中的 ℓ_{k-1} 之间的一个数值，如第 2 行所示。此修改由 [17] [18]提出并加以研究。

如果扰动（第 9–10 行）缩短了总线长，则算法接受该扰动；否则，我们会反复减小步长，最多 10 次，直到获得更短的线长。

3) 扰动可重置交替投影法（Per-RMAP）：Per-RMAP结合了RMAP和SM，其设计如算法3所示。

在Per-RMAP中，首先应用Num 次扰动进行优越化，随后按照位置顺序采用重置策略进行投影。投影的松弛参数（第5行）随着迭代次数增加，并设有上限。因此，随着迭代的进行，投影步骤会逐渐占据主导地位，并且每次迭代都不会使单元移动过远。

C. 后处理

全局布局规划后，我们获得的结果中，相对重叠面积小于 0.1%。优越化可能会使模块彼此靠近，但其对迭代过程中位置变化的影响越来越小。因此，模块之间仍可能存在一些间隙。于是再次运行 RMAP 以改进结果。在此阶段，扰动衰减指数重置为 $k \times \epsilon$，其中 k 为总迭代次数，且 $\epsilon \in (0,1)$。

算法 3 扰动可重置交替投影法（Per-RMAP）

要求：
初始位置： $z = (x,\ y) \in \mathbb{R}^{2N}$
每次迭代中的扰动次数：*Num*
最小扰动长度： λ_{min}
初始扰动长度： λ_{init}
扰动衰减因子： $\Lambda \in (0,1)$
初始投影长度： $\gamma_{init} \in (0,1)$
投影进度因子： $\Gamma > 1$

确保：
更新后的位置 $z = (x, y)$

$\ell_{-1} = 0$
对于 $k = 0:\ \infty$ **执行**
  $(z, \ell_k) = \mathrm{SM}(z,$ 编号 $k, \ell_{k-1}, \lambda_{min}, \lambda_{init}, \Lambda)$
  $order =$ [由位置顺序生成]
  $\gamma_{proj} = min(1,\ \gamma_{init} \cdot \Gamma^k)$
  $z = z + \gamma_{proj} \cdot (\mathrm{RMAP}(z, order) - z)$
  $isStop =$相对重叠面积检查(z)
  如果 $isStop == true$ **那么**
    休息
  结束如果
结束循环
return z

表 II MCNC 基准测试中 MAP 与 RMAP 的比较。结果表明，RMAP 能够缓解振荡。

实例	重置?	运行时间	迭代次数	相对 O. A.
apte	N	0.41	158	11.5%
	Y	0.35	41	$< 0.1\%$
xerox	N	0.27	131	10.5%
	Y	0.13	36	$< 0.1\%$
hp	N	0.47	286	0.6%
	Y	0.06	22	$< 0.1\%$
ami33	N	1.11	279	2.4%
	Y	0.71	80	$< 0.1\%$
ami49	N	8.13	931	3.4%
	Y	3.25	215	$< 0.1\%$

IV. 实验

A. 基准测试

北卡罗来纳州微电子中心（MCNC）基准测试 [19] 是一个常用的基准测试，共包含五个实例。详细信息如表 I所示，包括模块数量、输入/输出引脚、引脚、网络数量以及布局区域的大小（芯片尺寸）。

表 I MCNC 块实例的特征。

实例	数量				芯片尺寸
	模块	输入/输出引脚	Pins	Nets	
apte	9	73	214	97	10,500×10,500
xerox	10	2	696	203	5831×6412
hp	11	45	264	83	4928×4200
ami33	33	42	480	123	2058×1463
ami49	49	22	931	408	7672×7840

B. 重置策略的性能

表 II 展示了我们在 MCNC 基准测试中进行的实验里重置策略的性能。当**相对重叠面积（Relative O. A.）**在若干数值之间振荡、保持不变或小于 0.1% 时，程序便会停止。MCNC 基准测试的结果表明，如果不采用重置策略，MAP 可能会陷入不可行点，导致重叠无法完全消除。然而，采用重置策略后，这种现象便不再出现。

C. 布局规划结果

我们将结果与采用分支定界（B&B）方法获得的一些最先进结果进行比较 [5] 。该方法在前三个实例（apte、xerox 和 hp）上获得了最优布局规划。然而，由于时间限制，它在最后两个较大的实例（ami33 和 ami49）上仅获得了次优布局规划。在初始化步骤中，与同一实例中的其他模块相比尺寸较小的关键模块会被初始化到布局区域的边界[3]。对于不进行输入/输出分配的原始布局规划，结果如表 III 所示。超参数为 $\lambda_{min} = 0.1,\ \Lambda = 0.99,\ \lambda_{init} = 321,\ \gamma_{init} = 0.7804,\ \epsilon = \Gamma = 1.1,\ 0.35$ 3 。我们的方法所得半周长线长仅比 B&B 的最优结果长 %，且 [5] 执行时间为 $1.95\times$ [4]。

表 III 无输入/输出分配的布局规划实验结果。

实例	分支定界 [5]		Per-RMAP 不含输入/输出分配		比率 Per-RMAP/分支定界	
	半周长线长	时间（秒）	半周长线长	时间（秒）	半周长线长	
apte	513061	13	528618	8.84	1.03	0.68
xerox	370993	48	382596	172.09	1.03	3.59
hp	153328	102	159979	5.06	1.04	0.05
ami33	58627	13	61444	23.83	1.05	1.83
ami49	640509	73	637098	261.76	1.00	3.59
平均值	-	-	-	-	1.03	1.95

对于带输入/输出分配的原始布局规划，结果如表 IV 所示。超参数为 $\lambda_{min} = 0.1$、$\Lambda = 0.99$、$\lambda_{init} = 488$ 、$\gamma_{init} = 0.7761$、$\Gamma = 1.0001$ 和 $\epsilon = 0.35$。与分支定界相比，我们的方法使总线长改善了 8%，耗时为分支定界的 $4.99\times$ time [5]。

表 IV 布局规划与输入/输出分配的实验结果。

实例	分支定界 [5]		Per-RMAP 采用输入/输出分配的	比率 Per-RMAP/分支定界	
	半周长线长	时间（秒）	半周长线长	时间（秒）	半周长线长	时间
apte	513061	13	362587	122	0.71	9.39
xerox	370993	48	382587	188	1.03	3.92
hp	153328	102	149545	139	0.98	1.36
ami33	58627	13	54489	79	0.93	6.08
ami49	640509	73	618898	306	0.97	4.19
平均	-	-	-	-	0.92	4.99

[3]对于实例 xerox，它是名为 *BLKP* 的模块。对于实例 hp，它是名为 cc 22 和 $new1$ 的模块。[4]我们的原型使用 MATLAB 编写，若使用 C++，预计运行速度会很快。

V. 结论与未来工作

我们将固定外形布局建模为一个可行性寻求问题（FSP）。然而，用于求解 FSP 的传统交替投影方法（MAP）无法获得合法的布局规划。这是因为布局问题的约束集合并非凸集。我们分析了约束集合的并集凸性，并提出了可重置的交替投影方法（RMAP），以提高其收敛至可行解的能力。此外，我们设计了该方法的优越化版本 Per-RMAP，以减少总线长。实验表明，对于某些布局规划问题，我们的方法取得了近乎最优的结果；在考虑输入/输出分配后，与分支定界（B&B）方法相比，其半周长线长（HPWL）改善了 8%。未来的工作将通过增加更大规模实例的实验来研究该方法的可扩展性，并进一步考察其处理复杂实际约束的能力，例如 2.5D 布局规划中的约束。

致谢

我们感谢北京大学的同事 Tie Zhou 和 Jiansheng Yang，他们在本项目的 Zoom 联合会议期间提出了宝贵意见。

本研究得到中国国家自然科学基金委员会（NSFC）项目（编号：11961141007）以及以色列科学基金会（ISF）项目（编号：2874/19）的资助，资助来源为 NSFC-ISF 联合研究计划。Y.C. 的工作还得到美国国立卫生研究院项目（编号：R01CA266467）的资助。此外，G.L. 的工作得到中国国家重点研发计划项目（编号：2022YFB4500500）的资助。

参考文献

[1] S. N. Adya 和 I. L. Markov，“固定外形布局：实现层次化设计”，*IEEE VLSI* 系统汇刊，第 11 卷，1135，2003 年，1120 页 – 。[2] H. Murata、K. Fujiyoshi、S. Nakatake 和 Y. Kajitani，“基于序列对的矩形填充式 VLSI 模块布局”，*IEEE* 计算机辅助设计与集成电路系统汇刊，第 15 卷，1524，1996年，1518页 – 。[3] Y. Chang、Y. Chang、G. Wu 和 S. Wu，“B*-树：一种非切割布局规划的新表示方法”，载于 *ACM/IEEE* 设计自动化会议论文集 *(DAC)*，美国加利福尼亚州洛杉矶，2000年，第458–463页。[4] J. Cong、J. Wei 和 Y. Zhang，“面向3D IC的热驱动布局规划算法”，载于*IEEE/ACM*计算机辅助设计国际会议论文集 *(ICCAD)*，美国加利福尼亚州圣何塞，2004年，第306–313页。

[5] J. Funke、S. Hougardy 和 J. Schneider，《超大规模集成电路设计中线长最优布局的精确算法》，*Integration*，第52卷，第355–366页，2016年。[6] Y. Zhan、Y. Feng 和 S. S. Sapatnekar，《一种采用解析方法的固定芯片布局规划算法》，载于亚洲及南太平洋设计自动化会议论文集（*ASP-DAC*），日本横滨，2006年，第771–776页。[7] J. Lin 和 J. Wu，《F-FM：考虑电压岛约束的混合尺寸模块固定轮廓平面规划方法》，*IEEE*计算机辅助设计与集成电路系统汇刊，第33卷，1692，2014 1681页。– 。[8] Q. Xu、H. Geng、S. Chen、B. Yuan、C. Zhuo、Y. Kang 和 X. Wen，《GoodFloorplan：基于图卷积网络和强化学习的布局规划》，*IEEE*计算机辅助设计与集成电路系统汇刊，第41卷，第3492–3502页，2022年。[9] Y. Liu、Z. Ju、Z. Li、M. Dong、H. Zhou、J. Wang、F. Yang、X. Zeng 和 L. Shang，《采用图注意力的布局规划》，载于*ACM/IEEE*设计自动化会议论文集（*DAC*），美国加利福尼亚州旧金山，2022年，第1303–1308页。[10] Y. Censor，《优越化：寻求可行性与约简目标函数的不对称作用》，应用集值分析与优化，即将发表，2022年。[11] B. Schultze、Y. Censor、P. Karbasi、K. E. Schubert 和 R. W. Schulte，《应用于质子计算机断层成像重建的改进型全变分优越化方法》，*IEEE*医学成像汇刊，第39卷，第294–307页，2020年。[12] H. H. Bauschke 和 J. M. Borwein，《关于求解凸可行性问题的投影算法》，*SIAM*评论，第38卷，第3期，第367–426页，1996年。[13] R. Escalante 和 M. Raydan，交替投影方法，算法基础。美国宾夕法尼亚州费城：工业与应用数学学会（SIAM），2011年，第8卷。[14] S. Chr´etien 和 P. Bondon，《非凸集合类上的循环投影方法》，数值函数分析与优化，第17卷，第1-2期，第37–56页，1996年。[15] M.-C. Kim、D.-J. Lee 和 I. L. Markov，《SimPL：一种有效的布局算法》，*IEEE*计算机辅助设计与集成电路系统汇刊，第31卷，第50–60页，2012年。[16] N. Viswanathan 和 C. C. . Chu，《FastPlace：采用单元移动、迭代局部细化和混合网线模型的高效解析布局》，*IEEE*计算机辅助设计与集成电路系统汇刊，第24卷，第722–733页，2005年。[17] O. Langthaler，《将优越化方法纳入生物医学成像软件》，马歇尔计划奖学金报告，萨尔茨堡应用科学大学，技术报告，2014年9月。[18] B. Prommegger，《生物医学成像中优越化算法的验证与评估：比较采用和未采用优越化的迭代算法在投影图像重建中的表现》，马歇尔计划奖学金报告，萨尔茨堡应用科学大学，技术报告，2014年10月。[19] K. Kozminski，《布局综合基准测试——演进与现状》，载于*ACM/IEEE*设计自动化会议论文集（*DAC*），美国加利福尼亚州旧金山，1991年，第265–270页。