

UNIVERSIDADE FEDERAL DE SANTA CATARINA
CENTRO TECNOLÓGICO
CURSO DE GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

Lucas Yuki Imamura

Implementação, Validação e Análise do ePlace, Técnica Clássica de Posicionamento Global de Circuitos VLSI Baseada em Sistemas Eletrostáticos

Florianópolis
2026

Lucas Yuki Imamura

Implementação, Validação e Análise do ePlace, Técnica Clássica de Posicionamento Global de Circuitos VLSI Baseada em Sistemas Eletrostáticos

Trabalho de Conclusão de Curso do Curso de Graduação em Ciência da Computação do Centro Tecnológico da Universidade Federal de Santa Catarina para a obtenção do título de Bacharel em Ciência da Computação.
Orientador: Prof. José Luís Almada Güntzel, Dr.

Florianópolis
2026

Ficha catalográfica gerada por meio de sistema automatizado gerenciado pela BU/UFSC.
Dados inseridos pelo próprio autor.

Imamura, Lucas Yuki
Implementação, Validação e Análise do ePlace, Técnica Clássica de Posicionamento Global de Circuitos VLSI Baseada em Sistemas Eletrostáticos / Lucas Yuki Imamura ; orientador, José Luís Almada Güntzel, 2026.
73 p.

Trabalho de Conclusão de Curso (graduação) - Universidade Federal de Santa Catarina, Centro Tecnológico, Graduação em Ciências da Computação, Florianópolis, 2026.

Inclui referências.

1. Ciências da Computação. 2. EDA. 3. Posicionamento Global. 4. Posicionamento Baseado em Eletrostática. 5. Microeletrônica. I. Güntzel, José Luís Almada. II. Universidade Federal de Santa Catarina. Graduação em Ciências da Computação. III. Título.

Lucas Yuki Imamura

Implementação, Validação e Análise do ePlace, Técnica Clássica de Posicionamento Global de Circuitos VLSI Baseada em Sistemas Eletrostáticos

Este Trabalho de Conclusão de Curso foi julgado adequado para obtenção do Título de "Bacharel em Ciência da Computação" e aprovado em sua forma final pelo Curso de Graduação em Ciência da Computação.

Florianópolis, 2026.

Banca Examinadora:

Prof. José Luís Almada Güntzel, Dr.
Universidade Federal de Santa Catarina

Renan Oliveira Netto, Dr.
Cadence Design Systems, San Jose, CA, USA

Tiago Augusto Fontana, Dr.
Cadence Design Systems, Austin, TX, USA

RESUMO

O aumento exponencial do número de transistores integrados em uma mesma pastilha de silício, originando os chamados Circuitos Integrados (CIs) ou chips, dificulta a escalabilidade e a confiabilidade do fluxo de projetos *Very-Large-Scale Integration* (VLSI). Neste cenário, ferramentas de *Electronic Design Automation* (EDA) tornaram-se indispensáveis para viabilizar o desenvolvimento desses dispositivos. Com o aumento da complexidade dos projetos, é necessário que tais ferramentas sejam cada vez mais efetivas e eficientes para lidar com grandes espaços de busca em um curto período de tempo. O fluxo de EDA compreende diversas etapas, dentre as quais se destaca o posicionamento global, responsável por definir a posição física inicial de cada célula do circuito. Embora etapas posteriores do fluxo possam mover células novamente, o posicionamento global é um passo crucial no processo de projeto de um chip. Enquanto um posicionamento otimizado leva a uma melhor utilização de área, roteabilidade e menor atraso (e consequentemente maior frequência de clock), uma solução ineficiente pode comprometer o desempenho do chip e até inviabilizar sua fabricação. O objetivo deste trabalho é implementar, validar e analisar experimentalmente o algoritmo clássico de posicionamento global baseado em eletrostática, o ePlace, integrando-o ao fluxo de projeto do OpenROAD.

Palavras-chave: EDA. Projeto Físico. VLSI. Posicionamento Global. Posicionamento Baseado em Eletrostática. Microeletrônica.

ABSTRACT

The exponential increase in the number of transistors integrated into a single silicon die, giving rise to so-called CIs or chips, hinders the scalability and reliability of the VLSI design flow. In this scenario, EDA tools have become indispensable to enable the development of these devices. With the increasing complexity of designs, it is necessary for such tools to be ever more effective and efficient to handle large search spaces within a short period of time. The EDA flow comprises several stages, among which global placement stands out, being responsible for defining the initial physical position of each cell in the circuit. Although subsequent stages of the flow may move cells again, global placement is a crucial step in the chip design process. While an optimized placement leads to better area utilization, routability, and lower delay (and consequently higher clock frequency), an inefficient solution can compromise chip performance and even render its manufacturing unfeasible. The objective of this work is to implement, validate, and experimentally analyze the classic electrostatics-based global placement algorithm, ePlace, integrating it into the OpenROAD design flow.

Keywords: EDA. Physical Design. VLSI. Global Placement. Electrostatics-Based Placement. Microelectronics.

LISTA DE FIGURAS

Figura 1 – Fluxo VLSI 14
Figura 2 – Exemplos de posicionamentos 17
Figura 3 – Passos do posicionamento baseado em particionamento 18
Figura 4 – *Half-Perimeter WireLength* (HPWL) 21
Figura 5 – Modelagem do *eDensity* 23
Figura 6 – Remoção da componente DC da densidade de cargas 26
Figura 7 – Comparação entre gradiente descendente e Gradiente Acelerado de Nesterov (NAG) 28
Figura 8 – Ilustração de um passo do NAG 29
Figura 9 – Efeito da normalização de macros e instâncias fixas 35
Figura 10 – Efeito da inserção de *fillers* 35
Figura 11 – Diagrama de classes simplificado 40
Figura 12 – Fluxo de síntese física do OpenROAD 41
Figura 13 – Posicionamento global, posicionamento final e congestionamento final (*sky130hs* - ibex) 48

LISTA DE ALGORITMOS

1 Otimizador Nesterov durante o passo k 28
2 Otimizador Nesterov com previsão da constante Lipschitz durante o passo k com *backtracking* 31
3 ePlace 34
4 Gradiente 34

LISTA DE ABREVIATURAS E SIGLAS

EDA	*Electronic Design Automation*	4, 5
CI	Circuito Integrado	4, 5, 14, 15, 35
VLSI	*Very-Large-Scale Integration*	4–6, 13, 14
HPWL	*Half-Perimeter WireLength*	9, 21, 22, 36, 37, 43, 44, 47, 50
HDL	Linguagem de Descrição de Hardware	14
DC	Corrente Contínua	25
EDP	Equações Diferenciais Parciais	26
NAG	Gradiente Acelerado de Nesterov	6, 20, 27–30, 33, 40, 44, 51

LISTA DE SÍMBOLOS

G	Grafo que representa um posicionamento. $G = (V, E, R)$ 9, 16
V	Conjunto de nodos (células e macros) do circuito. $V = V_f \cup V_m$ 9, 16
V'	Conjunto de nodos do posicionamento. $V' = V_f \cup V_m \cup V_{fc} \cup V_d$ 9, 24, 25
V_f	Conjunto de nodos fixos do circuito (células fixas e macros) 9, 16
V_m	Conjunto de nodos móveis do circuito (células móveis) 9, 16
V_{fc}	Conjunto dos nodos *fillers*. 9
V_d	Conjunto dos nodos *dark*(*dummy*). 9
m	Número de nodos móveis. $\|V_m\| = m$ 9, 16
i	Uma célula do circuito. $i \in V'$ 9, 10, 23–25
v	Um posicionamento. Posições dos nodos $v = (x,y)$. 9, 10, 16, 20, 22, 24, 25, 27
x	Posições horizontais dos nodos. $x = (x_1, x_2, ..., x_m)$ 9, 16
y	Posições verticais dos nodos. $y = (y_1, y_2, ..., y_m)$ 9, 16
E	Conjunto de arestas (redes do circuito) 9, 16, 22
e	Uma rede do circuito 21, 22
R	Região de posicionamento 9, 10, 16, 25–27
$f(v)$	Custo da solução v. $f(v) = W(v) + \lambda N(v)$ 9, 20
$W(v)$	Custo *Half-Perimeter WireLength* (HPWL) da solução v 9, 20, 22
λ	Fator de penalidade da densidade 9, 20, 33, 36
$N(v)$	Penalidade da densidade (Energia potencial elétrica) da solução v. 9, 20, 24, 27

N_i	Energia potencial elétrica do nodo i	24
q_i	Carga elétrica do nodo i	23, 24
$\psi_i(v)$	Potencial elétrico na posição do nodo i	24, 25
$\psi(x,y)$	Potencial elétrico na posição (x, y)	25–27
$\xi_i(v)$	Campo elétrico na posição do nodo i	24
ρ_i	Densidade elétrica na posição do nodo i	25
$\rho(x,y)$	Densidade elétrica na posição (x, y)	24–27
$F_i(v)$	Força eletrostática do nodo i	24
$\hat{n}$	Vetor normal externo da região R	26, 27
∂R	Borda da região R	26, 27

SUMÁRIO

1 INTRODUÇÃO 13
1.1 OBJETIVOS 13
1.1.1 Objetivos Específicos 13

2 CONCEITOS FUNDAMENTAIS 14
2.1 FLUXO DE PROJETO VLSI 14
2.1.1 Projeto Físico 15
2.1.1.1 Etapas do projeto físico 15
2.2 POSICIONAMENTO GLOBAL 16
2.2.1 Classificação de algoritmos de posicionamento global 16
2.2.1.1 Particionamento 17
2.2.1.2 Estocástico 18
2.2.1.3 Analítico 19

3 EPLACE 20
3.1 FUNÇÃO CUSTO 20
3.1.1 Custo do comprimento das redes 21
3.1.2 Custo da densidade (eDensity) 23
3.1.2.1 Equação de Poisson 25
3.2 OTIMIZAÇÃO NÃO LINEAR 27
3.2.1 Gradiente Acelerado de Nesterov (NAG) 27
3.2.2 Tamanho de passo: Predição da constante Lipschitz 29
3.2.2.1 Backtracking 30
3.2.3 Precondicionamento não linear 31
3.3 POSICIONAMENTO GLOBAL 33
3.3.1 Adaptações e Flexibilização do Modelo de Densidade 33
3.3.2 Ajuste automático de parâmetros 36
3.4 TRABALHOS DERIVADOS DO EPLACE 37

4 METODOLOGIA 39
4.1 OPENROAD 39

4.2 IMPLEMENTAÇÃO DO ALGORITMO EPLACE 39

5 RESULTADOS 41
5.1 METODOLOGIA EXPERIMENTAL 41
5.2 RESULTADOS EXPERIMENTAIS 44

6 CONCLUSÃO 50

REFERÊNCIAS 52

APÊNDICES 55

APÊNDICE A – CÓDIGO FONTE 56

APÊNDICE B – ARTIGO 57

ANEXOS 71

ANEXO A – DECLARAÇÃO PADRÃO PARA EMPRESA OU LABORATÓRIO 72

1 INTRODUÇÃO

Atualmente, a indústria de semicondutores tem enfrentado desafios devido ao aumento na dimensão dos projetos e complexidade dos objetivos e restrições. Estes fatores tornam a automação da síntese física dos fluxos de projeto de circuitos de Integração em Escala Muito Grande ou, em inglês, *Very-Large-Scale Integration* (VLSI), um processo cada vez mais difícil e demorado (QIU et al., 2023).

Dentre as etapas do fluxo de projeto, o posicionamento global assume um papel central pois impacta diretamente a roteabilidade e a performance do circuito. Isso ocorre porque, nas etapas seguintes, as células sofrem apenas pequenas mudanças na posição (CHU, 2009).

Considerando que o posicionamento global é um problema *NP-completo* (SAHNI; BHATT, 1980), as técnicas atuais baseiam-se em heurísticas para convergir a uma solução eficiente. De acordo com Zhang e Sapatnekar (2026), a maioria das abordagens contemporâneas deriva do ePlace, que utiliza uma analogia com sistemas eletrostáticos para diminuir a sobreposição das células durante o posicionamento. Dentre os algoritmos que seguem esse paradigma, destacam-se o RePlAce (CHENG et al., 2019), o DREAMPlace (LIN et al., 2019), Xplace (LIU et al., 2022) e Pplace-ms (PENG; ZHU, 2024).

1.1 OBJETIVOS

Este trabalho visa investigar experimentalmente o funcionamento do algoritmo clássico de posicionamento global baseado em eletrostática, o ePlace, através de uma implementação própria no contexto do projeto OpenROAD.

1.1.1 Objetivos Específicos

- Realizar a validação do algoritmo implementado através da geração de resultados de síntese física utilizando circuitos de *benchmarks* de acesso público.
- Comparar a técnica implementada com a técnica RePlAce, por meio da geração de resultados de síntese utilizando a plataforma OpenROAD, com circuitos de *benchmark* de acesso público.

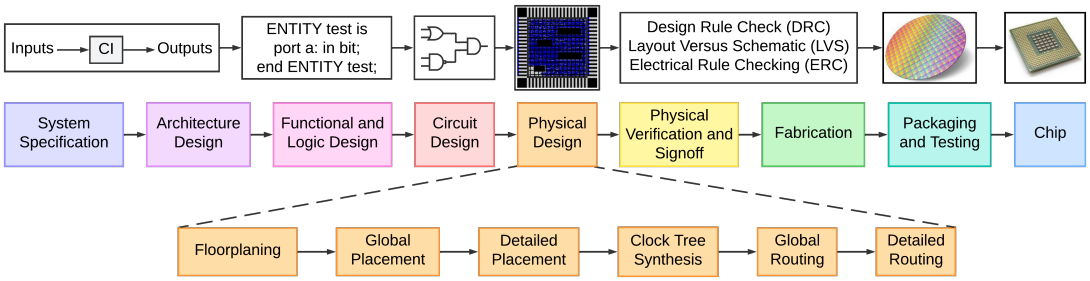
2 CONCEITOS FUNDAMENTAIS

Este capítulo apresenta os conceitos fundamentais para a compreensão deste trabalho, tais como o fluxo de projeto VLSI e o posicionamento global.

2.1 FLUXO DE PROJETO VLSI

O fluxo de projeto VLSI é composto de múltiplas etapas (Figura 1), onde as primeiras são de alto nível e as últimas de baixo nível de abstração.

Figura 1 – Fluxo VLSI



Fonte: Adaptado de Kahng et al. (2011).

Primeiro são definidos os objetivos e requerimentos do Circuito Integrado (CI) em alto nível e, após isso, é escolhida a arquitetura para alcançar as especificações. Durante o projeto funcional e lógico, são determinadas a funcionalidade e as conexões de cada módulo que compõe o sistema, geralmente utilizando uma Linguagem de Descrição de Hardware (HDL) como *Verilog* ou *VHDL*. O projeto da parte digital do circuito é geralmente feito automaticamente por uma ferramenta de síntese lógica, que converte o projeto descrito em uma HDL para um conjunto de *standard cells*, macros e lista de redes. Então, na etapa de projeto físico, é definida a posição física de todos os componentes do CI e todas suas conexões, respeitando as regras de fabricação para que o CI possa ser fabricado, encapsulado e testado (WESTE; HARRIS, 2011).

2.1.1 Projeto Físico

A etapa de projeto físico automatizado, neste trabalho referida por síntese física, é realizada logo após o projeto de circuito. Antes de começar esta etapa, o circuito já foi discretizado em *standard cells*, macros e lista de redes, onde:

- ***Standard cells*** — As *standard cells* são pequenos circuitos previamente projetados fisicamente e validados em uma tecnologia específica para fazer operações simples (como portas lógicas e flip-flops). Através delas é possível decompor circuitos mais complexos em *standard cells* durante o projeto de circuito, possibilitando semi-automatizar o projeto físico, mesmo em circuitos com bilhões de transistores.
- **Macros** — Macro blocos são circuitos maiores previamente projetados e testados, que podem ser reutilizados, como, por exemplo, módulos de memória, somadores, multiplicadores e até processadores.
- **Lista de redes** — A lista de redes é um conjunto de redes de sinais e dos pinos aos quais elas estão conectadas. Nela está contida a informação de como as *standard cells* e macros são conectadas entre si e aos pinos de entrada e saída do CI.

2.1.1.1 Etapas do projeto físico

O projeto físico também pode ser dividido em:

- **Floorplan (Planejamento topológico)** — A etapa de *floorplan* determina a dimensão do chip e faz o posicionamento dos macro blocos.
- **Posicionamento global** — Define a posição aproximada das *standard cells* tentando minimizar o comprimento das redes, visando manter uma densidade de células menor que um valor pré-definido, para ser possível legalizar e rotear o circuito em etapas posteriores. No final desta etapa ainda é esperado que existam problemas de alinhamento e sobreposição de *standard cells*.
- **Posicionamento detalhado** — Realiza a legalização da posição das *standard cells*, garantindo que elas estejam devidamente alinhadas e sem sobreposição.

- **Síntese de relógio (CTS)** — Constrói a árvore de relógio para distribuir o sinal do relógio a todos os elementos sequenciais do circuito, adicionando buffers para diminuir o atraso e diminuir a diferença entre os atrasos do relógio entre diferentes células.
- **Roteamento global** — Aloca os recursos de cada rede do circuito, determinando aproximadamente por onde cada rede será roteada.
- **Roteamento detalhado** — Refina a solução do roteamento global, definindo exatamente a geometria e as camadas de metal utilizadas para conectar as redes do circuito.

2.2 POSICIONAMENTO GLOBAL

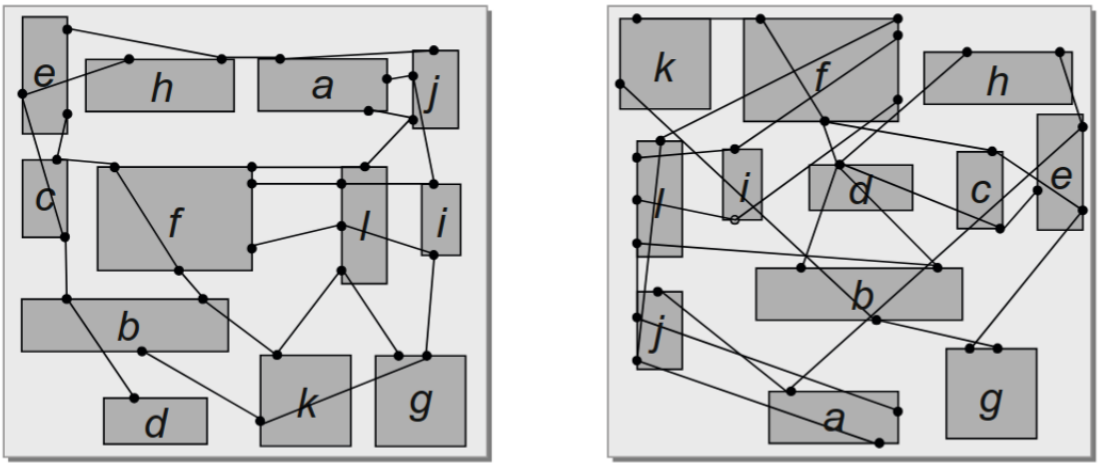
O posicionamento é uma das etapas da síntese física, na qual é definida a localização física de cada célula do circuito de forma que seja possível sua fabricação. Antes do posicionamento, são feitas as etapas de particionamento e *floorplanning*, nas quais o circuito é dividido em módulos menores, são definidas as regiões de cada bloco e a localização dos pinos. A partir disso, o posicionamento das *standard cells* e dos elementos lógicos é feito com o objetivo de minimizar o comprimento das conexões do circuito. A figura 2 ilustra dois arranjos para o mesmo circuito: o da esquerda demonstra um posicionamento com boa otimização do comprimento das redes, enquanto o da direita ilustra uma solução de qualidade inferior (KAHNG et al., 2011).

Matematicamente, o posicionamento pode ser definido como um hipergrafo $G = (V, E, R)$ onde V é um conjunto de vértices (instâncias), E é um conjunto de arestas (redes) e R é a região de posicionamento. V_f e V_m representam os conjuntos de nodos fixos (células fixas e macros) e móveis (células móveis) respectivamente, sendo $|V_m| = m$ o número de nodos móveis. Um posicionador define os locais das células do circuito $v = (x, y)$, onde $x = (x_1, x_2, ..., x_m)$ e $y = (y_1, y_2, ..., y_m)$ (LU, 2014).

2.2.1 Classificação de algoritmos de posicionamento global

Os algoritmos de posicionamento global podem ser categorizados em 3 tipos: baseados em particionamento; estocásticos; e analíticos (MONTEIRO, 2022).

Figura 2 – Exemplos de posicionamentos

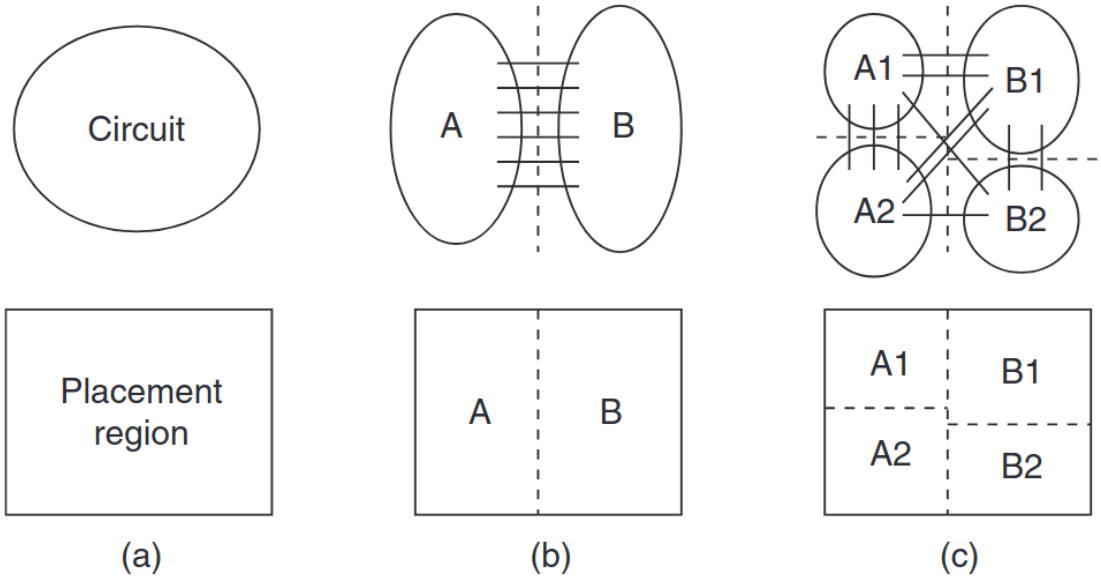


Fonte: Kahng et al. (2011).

2.2.1.1 Particionamento

O problema de particionamento consiste em dividir um grafo, fazendo um ou múltiplos cortes, em subgrafos de forma que minimize o valor das arestas que ligam um subgrafo ao outro. No posicionamento utilizando particionamento, o circuito é representado como um hipergrafo, onde os nodos são instâncias e as arestas são as conexões (KAHNG; REDA, 2006).

Figura 3 – Passos do posicionamento baseado em particionamento



Fonte: Chu (2009).

O circuito inicial (Figura 3.a) é dividido em subcircuitos de tamanhos similares (Figura 3.b), buscando minimizar as conexões entre eles. A partir disso, é atribuída uma sub-região da região original a cada subcircuito. Este processo é repetido recursivamente (Figura 3.c) até que existam poucas células em cada sub-região, viabilizando a definição do posicionamento final. Por meio desse processo, o *wirelength* é indiretamente otimizado através da minimização das arestas cortadas (CHU, 2009).

2.2.1.2 Estocástico

Os algoritmos de posicionamento global estocástico utilizam técnicas de otimização não determinísticas para definir as posições das células. A estratégia utilizada, recozimento simulado (ou *Simulated Annealing*), simula a física do processo de recozimento utilizado na metalurgia. A função custo é otimizada mexendo aleatoriamente no posicionamento das células do circuito e verificando as mudanças que causam na função (RUTENBAR, 1989).

2.2.1.3 Analítico

Segundo Kahng et al. (2011), algoritmos de posicionamento global analítico modelam o problema do posicionamento utilizando uma função de custo que pode ser otimizada por análise numérica ou programação linear. Para estas técnicas funcionarem, geralmente são necessárias algumas suposições, como a diferenciabilidade da função de custo ou considerar as células como pontos adimensionais.

Os métodos de posicionamento analíticos podem ser divididos em quadráticos e não lineares, dependendo da função de custo. Os quadráticos aproximam o tamanho das redes utilizando funções quadráticas, que são otimizadas resolvendo um sistema de equações lineares, enquanto os não lineares utilizam funções suaves com gradientes que possam ser calculados analiticamente, resolvendo o problema de otimização da função custo utilizando técnicas de programação não linear (QIU et al., 2023).

3 EPLACE

Atualmente, grande parte das técnicas de posicionamento global avançadas são fundamentadas na analogia eletrostática (ZHANG; SAPATNEKAR, 2026). O primeiro posicionador global baseado em eletrostática foi o FFTPL (Lu, Jingwei et al., 2013), que em seguida foi aprimorado no ePlace (LU et al., 2014) e no ePlace-ms (LU et al., 2015).

Estes algoritmos fazem parte da classe de posicionadores analíticos não lineares que utilizam um modelo eletrostático para manter a densidade de células menor que o valor desejado. Nesta abordagem, as células e macros do circuito são modeladas como partículas carregadas eletricamente com cargas positivas, gerando forças de repulsão eletrostática mútua, fazendo com que as células sejam uniformemente distribuídas pela região. A partir dessa analogia, é definida uma função de custo diferenciável que permite a aplicação de algoritmos de otimização não linear. Desta forma, o posicionador consegue otimizar simultaneamente o comprimento das redes e diminuir a sobreposição entre instâncias.

3.1 FUNÇÃO CUSTO

A equação (3.1) define o objetivo da otimização realizada durante o posicionamento. A função custo $f(v)$ é composta de dois termos: comprimento das redes $W(v)$ e a penalidade da densidade $N(v)$. O parâmetro λ é utilizado para balancear ambos os custos durante o posicionamento, sendo atualizado iterativamente durante o processo de otimização, comportamento descrito na seção 3.3.2 (Lu, Jingwei et al., 2013).

$$\min_{v} f(v) = W(v) + \lambda N(v) \tag{3.1}$$

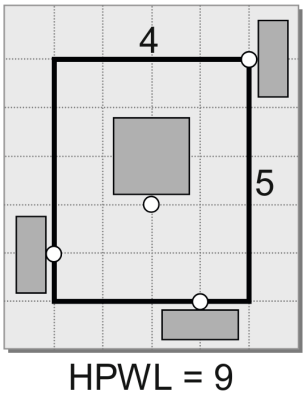
Para otimizar o custo, o ePlace utiliza o gradiente da função $\nabla f(v)$, equação (3.2), para convergir ao mínimo através do algoritmo Gradiente Acelerado de Nesterov (NAG) (LU et al., 2014).

$$\nabla f(v) = \nabla W(v) + \lambda \nabla N(v) \tag{3.2}$$

Nesta seção são abordados o equacionamento da função de custo e o cálculo de seus gradientes.

3.1.1 Custo do comprimento das redes

Figura 4 – *Half-Perimeter WireLength* (HPWL)



Fonte: Kahng et al. (2011).

Posicionadores analíticos dirigidos por comprimento de fio buscam otimizar o tamanho das conexões com uma aproximação do *Half-Perimeter WireLength* (HPWL). O HPWL é definido como o semiperímetro do retângulo que engloba todos os pinos da rede, conforme ilustrado na Figura 4. Para redes de dois e três pinos (que compõem cerca de 70–80% das redes de um circuito) o HPWL é igual ao custo mínimo para conectar todos os pinos. Para redes com mais de 3 pinos, o HPWL subestima o custo por um valor que cresce assintoticamente com a raiz quadrada da quantidade de pinos da rede. Matematicamente o HPWL é calculado pela equação (3.3), onde e é uma rede do circuito, e_x e e_y são respectivamente as coordenadas x e y dos pinos da rede e (KAHNG et al., 2011).

$$
\begin{aligned}
HPWL(v) &= \sum_{e \in E} (HPWL_x(e) + HPWL_y(e)) \\
HPWL_x(e) &= \max(e_x) - \min(e_x) \\
HPWL_y(e) &= \max(e_y) - \min(e_y)
\end{aligned}
\tag{3.3}
$$

Apesar do HPWL ser uma métrica comumente utilizada para estimar comprimento de fio, ele não é uma função diferenciável, o que torna sua utilização em um posicionador analítico não linear inviável. Por este motivo, é necessário utilizar aproximações suaves e deriváveis, como o HPWL de média ponderada (RAY et al., 2015).

O modelo de média ponderada, proposto por Hsu, Balabanov e Chang (2013), aproxima a função máximo (max), e consequentemente mínimo (min), através de uma média ponderada, conforme equação (3.4). Nesta aproximação, os pesos ($e^{\frac{val}{\gamma}}$) crescem exponencialmente com a magnitude dos valores, logo os valores mais altos influenciam mais o resultado, aproximando o comportamento da função max. Analogamente, a aproximação da função min é obtida invertendo o sinal dos pesos (multiplicando por -1).

$$
max(vals) \approx WAW_{max}(vals) = \frac{\sum_{val \in vals} val \cdot e^{\frac{val}{\gamma}}}{\sum_{val \in vals} e^{\frac{val}{\gamma}}} \tag{3.4}
$$

$$
\begin{aligned}
HPWL_x(e) &\approx WAW_{max}(e_x) - WAW_{min}(e_x) \\
HPWL_x(e) \approx W_x(e) &= \left(\frac{\sum_{i \in e} x_i \cdot e^{\frac{x_i}{\gamma}}}{\sum_{i \in e} e^{\frac{x_i}{\gamma}}} - \frac{\sum_{i \in e} x_i \cdot e^{\frac{-x_i}{\gamma}}}{\sum_{i \in e} e^{\frac{-x_i}{\gamma}}} \right)
\end{aligned}
\tag{3.5}
$$

$$
HPWL(v) \approx W(v) = \sum_{e \in E} (W_x(e) + W_y(e)) \tag{3.6}
$$

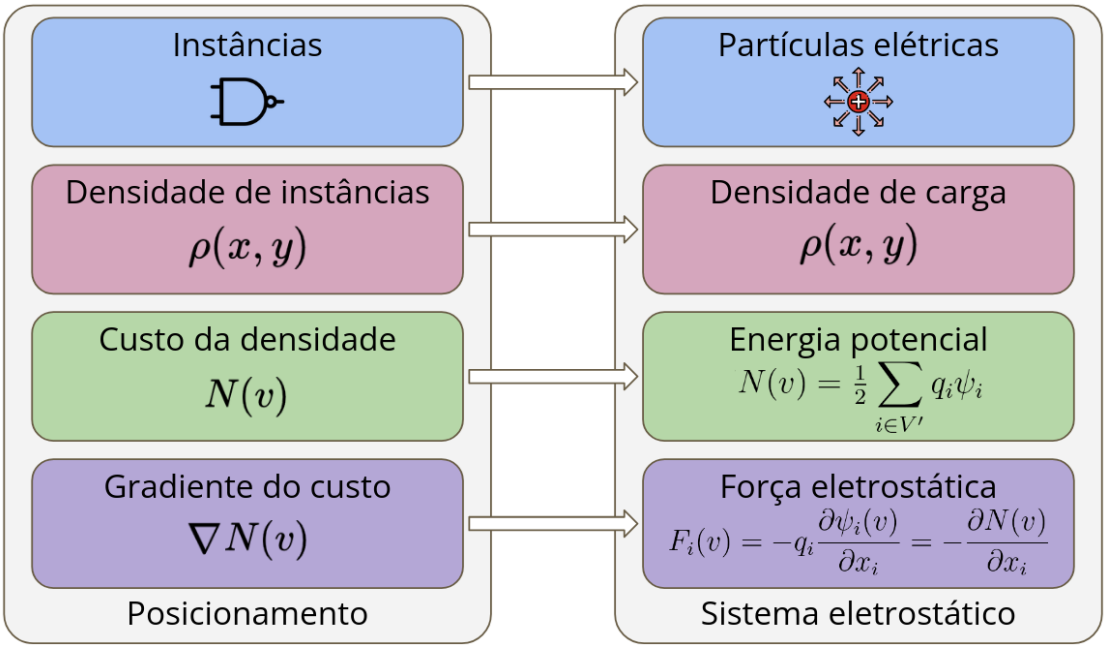
As equações (3.5) e (3.6) definem o modelo de média ponderada para a coordenada x, sendo realizado o mesmo processo para a coordenada y. O parâmetro γ é um coeficiente utilizado para ajustar o quão precisa é a aproximação, com $\gamma \to 0$ a função converge para o valor do HPWL (RAY et al., 2015). Durante a execução do ePlace, o valor de γ é atualizado entre iterações para alcançar melhores resultados, detalhes sobre a atualização deste parâmetro se encontram na seção 3.3.2.

Com a aproximação das funções min e max através da média ponderada, é possível derivar analiticamente o custo do comprimento do fio em relação à posição de cada pino através da equação:

$$\frac{\partial W_x(e)}{\partial x_i} = \frac{\left(1+\frac{x_i}{\gamma}\right)\sum_{j\in e} e^{\frac{x_j}{\gamma}} - \frac{1}{\gamma}\sum_{j\in e} x_j e^{\frac{x_j}{\gamma}}}{\left(\sum_{j\in e} e^{\frac{x_j}{\gamma}}\right)^2} e^{\frac{x_i}{\gamma}} - \frac{\left(1-\frac{x_i}{\gamma}\right)\sum_{j\in e} e^{\frac{-x_j}{\gamma}} + \frac{1}{\gamma}\sum_{j\in e} x_j e^{\frac{-x_j}{\gamma}}}{\left(\sum_{j\in e} e^{\frac{-x_j}{\gamma}}\right)^2} e^{\frac{-x_i}{\gamma}} \tag{3.7}$$

3.1.2 Custo da densidade (eDensity)

Figura 5 – Modelagem do *eDensity*



Fonte: Adaptado de Lu et al. (2014).

Lu et al. (2014) propuseram o eDensity, um modelo de custo de densidade baseado em um sistema eletrostático, conforme ilustrado na Figura 5. Nesse modelo, as instâncias i são tratadas como partículas elétricas positivamente carregadas, cuja carga q_i

é igual à sua área. Desse modo, células maiores exercem uma força de repulsão maior sobre as demais, abrindo maior espaço para seu posicionamento. Consequentemente, a densidade das células passa a ser igual à densidade das cargas $\rho(x,y)$. O custo da densidade é definido como a energia potencial do sistema $N(v)$, de forma que as forças eletrostáticas correspondam ao gradiente negativo do custo. Isto ocorre porque as forças guiam o sistema para o equilíbrio, que é o estado de menor energia potencial. Assim, ao minimizar a energia do sistema, as forças eletrostáticas também reduzem o custo da densidade do circuito.

$$N(v) = \frac{1}{2}\sum_{i \in V'} N_i = \frac{1}{2}\sum_{i \in V'} q_i \psi_i(v) \tag{3.8}$$

A energia potencial elétrica do sistema, custo da densidade, pode ser calculada através da equação (3.8), onde N_i é a energia potencial, q_i é a carga e $\psi_i(v)$ é o potencial elétrico da partícula *i*. Para otimizar a função de custo, é necessário calcular o gradiente de $N(v)$, equação (3.9). Como as cargas nas partículas são constantes, só é preciso calcular o gradiente do potencial elétrico.

$$\frac{\partial N(v)}{\partial x_i} = q_i \frac{\partial \psi_i(v)}{\partial x_i} \tag{3.9}$$

Considerando que o campo elétrico $\xi_i(v)$ é definido como o gradiente negativo do potencial elétrico, $\xi_i(v) = -\frac{\partial \psi_i(v)}{\partial x_i}$, a força eletrostática descrita pela lei de Coulomb (equação 3.10) tem a direção oposta ao gradiente do potencial elétrico. Consequentemente, a força eletrostática move as cargas em direção à redução da energia potencial do sistema.

$$F_i(v) = q_i \xi_i(v) = -q_i \frac{\partial \psi_i(v)}{\partial x_i} = -\frac{\partial N(v)}{\partial x_i} \tag{3.10}$$

Para calcular o gradiente, é utilizada a equação de Poisson, relacionando a densidade de cargas ao potencial ao campo elétrico. Esta equação é resolvida numericamente via métodos espectrais, que possuem grande precisão e baixo custo computacional. Através desta técnica é possível calcular o gradiente do custo de densidade a partir da distribuição das instâncias (LU et al., 2014).

3.1.2.1 Equação de Poisson

A equação de Poisson para eletrostática (3.11) relaciona o potencial elétrico $\psi_i(v)$ à densidade de carga ρ_i, onde ε é a permissividade do meio. Como se trata de uma modelagem inspirada no modelo eletrostático e não um sistema físico real, as unidades são irrelevantes e por isso a constante de permissividade do meio é omitida (equação (3.12)).

$$\nabla^2 \psi_i(v) = -\frac{\rho_i}{\varepsilon} \tag{3.11}$$

$$\nabla^2 \psi_i(v) = -\rho_i \tag{3.12}$$

O cálculo da densidade ρ_i e do potencial elétrico $\psi_i(v)$ é feito discretizando o espaço através de uma grade sobre a região de posicionamento R. Desta forma, a grade de densidade pode ser calculada através da equação (3.13), onde $\rho(x,y)$ é a densidade no retângulo (x,y) e $A^i_{(x,y)}$ é a área de sobreposição do retângulo (x,y) com o nodo i.

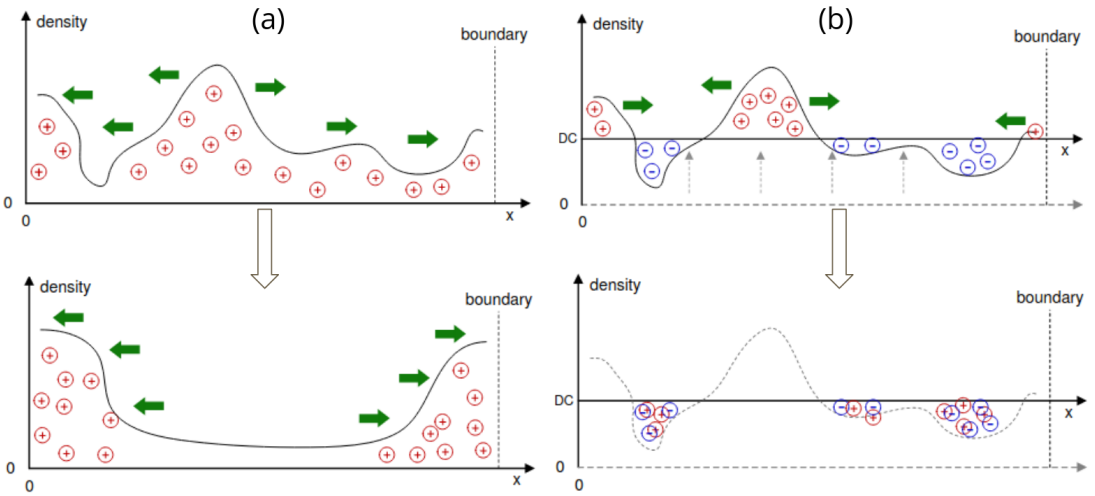
$$\rho(x,y) = \sum_{i \in V'} A^i_{(x,y)} \tag{3.13}$$

Utilizando a equação de Poisson para eletrostática (equação 3.11) podemos calcular o potencial elétrico em cada retângulo da grade ($\psi(x,y)$) segundo a equação (3.14).

$$\nabla^2 \psi(x,y) = -\rho(x,y), (x,y) \in R \tag{3.14}$$

Para garantir a distribuição uniforme das instâncias, é necessário remover a componente de Corrente Contínua (DC) da densidade de cargas (equação 3.15). Em um sistema eletrostático composto apenas por partículas com cargas de mesmo sinal, Figura 6.a, as forças de repulsão entre as partículas forçam a dispersão das mesmas em direção às bordas da região. Com a remoção da componente DC, Figura 6.b, a carga média do sistema se torna 0, assim as áreas com maior densidade que a média permanecem com carga positiva, enquanto as áreas com densidade menor que a média se tornam cargas negativas. Desta forma, as partículas em regiões densas se dispersam e são atraídas para

Figura 6 – Remoção da componente DC da densidade de cargas



Fonte: Adaptado de Lu et al. (2014).

as regiões de baixa densidade, direcionando o sistema para a distribuição uniforme de instâncias.

$$\iint_R \rho(x,y) = \iint_R \psi(x,y) = 0 \tag{3.15}$$

Para restringir as instâncias à região de posicionamento, é utilizada a condição de contorno de Neumann (equação 3.16), onde $\hat{n}$ é o vetor normal externo da região R e ∂R é seu contorno. Por meio desta restrição, o campo elétrico no contorno da região é reduzido para zero, evitando que as instâncias sejam movidas para fora da região de posicionamento.

$$\hat{n} \cdot \nabla\psi(x,y) = 0, (x,y) \in \partial R \tag{3.16}$$

A combinação das equações (3.14), (3.15), (3.16) resulta no sistema de Equações Diferenciais Parciais (EDP) (3.17). A solução do sistema, calculada numericamente através de métodos espectrais, permite obter o gradiente do potencial elétrico $\nabla\psi(x,y)$ a partir da densidade de cargas $\rho(x,y)$. Então é possível calcular o gradiente

da função de custo de densidade $\nabla N(v)$ necessária para otimizar o posicionamento (LU et al., 2014).

$$\begin{cases} \nabla^2 \psi(x,y) = -\rho(x,y), (x,y) \in R \\ \hat{n} \cdot \nabla \psi(x,y) = 0, (x,y) \in \partial R \\ \iint\limits_R \rho(x,y) = \iint\limits_R \psi(x,y) = 0 \end{cases} \tag{3.17}$$

3.2 OTIMIZAÇÃO NÃO LINEAR

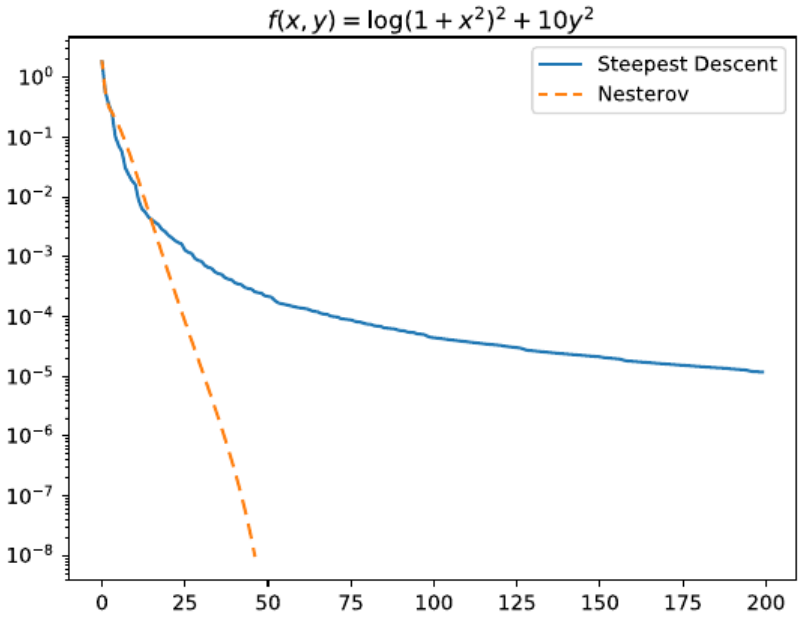
Para otimizar a função de custo do posicionamento, Lu (2014) propôs a utilização do algoritmo NAG, empregando uma heurística baseada na previsão da constante de Lipschitz para estimar o tamanho do passo de cada iteração.

3.2.1 Gradiente Acelerado de Nesterov (NAG)

O NAG foi escolhido por precisar apenas do gradiente de primeira ordem e memória linear em relação ao tamanho do problema. Enquanto métodos clássicos, como gradiente descendente, possuem taxa de convergência de $O(1/k)$ (onde k representa o número de iterações), o NAG foi o primeiro a alcançar $O(1/k^2)$, limite teórico provado para métodos de primeira ordem em funções convexas com gradiente Lipschitz contínuo. A Figura 7 ilustra a otimização de uma função $f(x,y)$ por ambos os algoritmos. Neste exemplo é possível notar que o NAG atinge 10^{-8} em menos de 50 iterações, enquanto o gradiente descendente não ultrapassa 10^{-5} mesmo após 200 iterações (WALKINGTON, 2023).

Dado um espaço de Hilbert E, o NAG soluciona o problema de programação convexa construindo uma sequência minimizante de pontos $\{x_k\}_0^\infty$ que não é relaxacional. Esta propriedade permite reduzir o esforço computacional por iteração ao mínimo (NESTEROV, 1983). O algoritmo 1 descreve uma iteração do método para um problema $min\{f(x)|x \in E\}$ com um conjunto não vazio de minimos X^*. Aqui x é a solução do problema de otimização convexa, y é uma solução de referência utilizada para calcular o gradiente, a é um parâmetro de aceleração e $step$ é o tamanho do passo. No início ($k = 0$) o método começa com uma solução $y_0 \in E$, $x_0 = v_0$ e $a_0 = 1$, sendo estes os

Figura 7 – Comparação entre gradiente descendente e NAG



Fonte: Walkington (2023).

parâmetros atualizados em cada iteração.

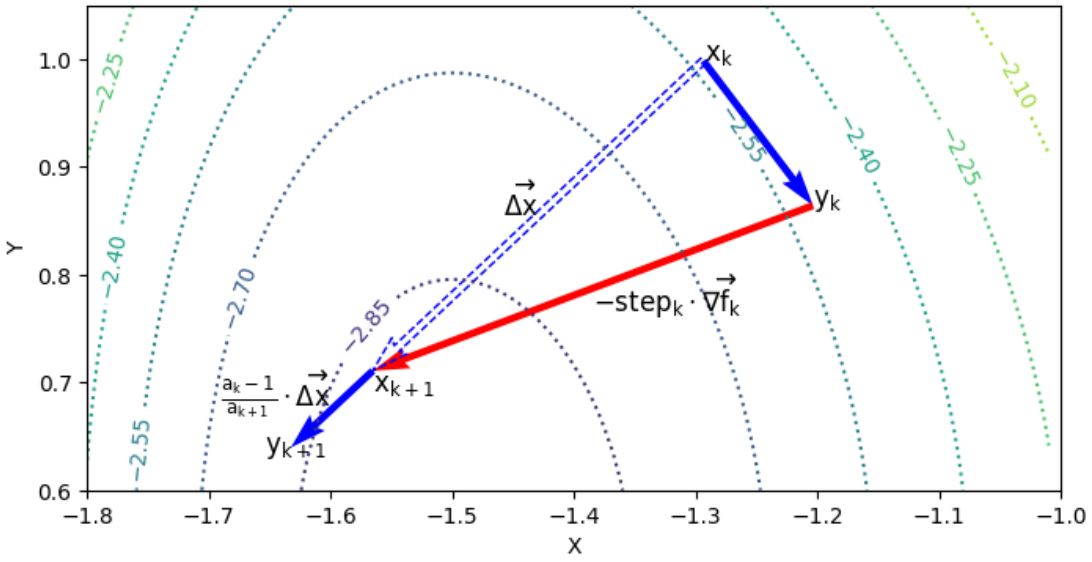
Algoritmo 1 Otimizador Nesterov durante o passo k

Entradas: Tamanho do passo $step_k$, parâmetro a_k, solução x_k, referência y_k
Saídas: Parâmetro a_{k+1}, solução x_{k+1}, referência y_{k+1}

1: $a_{k+1} \leftarrow \left(1 + \sqrt{4a_k^2 + 1}\right)/2$
2: $\nabla f_k \leftarrow \nabla f(y_k)$
3: $x_{k+1} \leftarrow y_k - step_k \cdot \nabla f_k$
4: $y_{k+1} \leftarrow x_{k+1} + (a_k - 1)(x_{k+1} - x_k)/a_{k+1}$

A Figura 8 ilustra o comportamento de um passo do algoritmo 1. Em cada iteração, o algoritmo recebe como entrada o tamanho do passo $step_k$, o parâmetro a_k, a solução x_k e a referência y_k. Na linha 3, o NAG atualiza a posição x_{k+1} utilizando a referência y_k e o gradiente ∇f_k. Então, a nova referência y_{k+1} é calculada por meio da extrapolação da direção $\overrightarrow{\Delta x}$ percorrida pela solução x. Diferentemente do algoritmo de gradiente descendente convencional, o NAG projeta a solução x em uma referência y para antecipar a tendência da função, permitindo que avance em direção ao mínimo

Figura 8 – Ilustração de um passo do NAG



Fonte: Elaboração própria (2026).

global com mais eficiência.

3.2.2 Tamanho de passo: Predição da constante Lipschitz

O NAG soluciona problemas de otimização convexa para funções $f(x)$ que pertençam à classe $C^{1,1}$, ou seja, cujos gradientes satisfazem a condição de continuidade de Lipschitz. O gradiente de uma função é dito Lipschitz-contínuo se existe uma constante $L > 0$ tal que, para todo $x, y \in E$, a seguinte inequação seja satisfeita:

$$\|\nabla f(x) - \nabla f(y)\| \leq L \|x - y\| \tag{3.18}$$

Para que o NAG atinja a taxa de convergência $O(1/k^2)$, é necessário que o tamanho do passo $step_k$ satisfaça a regra de Armijo (inequação (3.19)) em toda iteração (WALKINGTON, 2023).

$$f(y_k) - f(x_{k+1}) \geq \frac{step_k}{2} \|\nabla f(y_k)\|^2 \tag{3.19}$$

Originalmente, Nesterov (1983) sugere utilizar uma busca por bisseção para encontrar o maior valor de $step_k$ que satisfaça a inequação (3.19). Porém, esta busca aumenta significativamente o custo computacional. Por conta disto, é utilizada uma previsão para o tamanho do passo. Como descrito por Nesterov (1983), caso a constante de Lipschitz seja conhecida, é possível utilizar o valor inverso dela como tamanho de passo para satisfazer a inequação (3.19) sem degradação na velocidade de convergência. Porém, calcular o valor exato da constante de Lipschitz para a função de custo (3.1) é difícil pelos seguintes motivos:

- Apesar da função de custo (3.1) possuir um gradiente contínuo, o termo de densidade (*eDensity*) não é convexo.
- A função de custo do comprimento das redes muda durante as iterações devido ao ajuste do coeficiente γ (detalhes em 3.3.2)
- O coeficiente da penalidade de densidade λ é atualizado em cada iteração para balancear as forças entre o comprimento de fio e densidade (detalhes em 3.3.2)

Por conta destas limitações, Lu (2014) propõe um método para aproximar a constante de Lipschitz dinamicamente. Utilizando a equação (3.20), a aproximação da constante de Lipschitz ($\widetilde{L}_k$) é calculada a partir das soluções de referência (y_k e y_{k-1}) e seus gradientes ($f(y_k)$ e $f(y_{k-1})$). Como estes valores já são computados durante a execução do NAG, esta técnica não aumenta significativamente o custo computacional do algoritmo.

$$\widetilde{L}_k = \frac{\|\nabla f(y_k) - \nabla f(y_{k-1})\|}{\|y_k - y_{k-1}\|}, step_k = \widetilde{L}_k^{-1} \tag{3.20}$$

3.2.2.1 Backtracking

Apesar da aproximação (3.20) acelerar o cálculo do tamanho do passo, ela utiliza apenas o ponto de referência anterior (y_{k-1}) e o atual (y_k), enquanto a inequação 3.18 depende da nova solução (x_{k+1}). Esta diferença pode causar uma superestimação do tamanho de passo, violando a condição de convergência do algoritmo.

Algoritmo 2 Otimizador Nesterov com previsão da constante Lipschitz durante o passo k com *backtracking*

Entradas: Parâmetro a_k, solução x_k, referências y_{k-1} e y_k, gradiente ∇f_{k-1}
Saídas: Parâmetro a_{k+1}, solução x_{k+1}, referência y_{k+1}

1: $a_{k+1} \leftarrow \left(1+\sqrt{4a_k^2+1}\right)/2$
2: $\nabla f_k \leftarrow \nabla f(y_k)$
3: $step \leftarrow \frac{\|y_k - y_{k-1}\|}{\|\nabla f_k - \nabla f_{k-1}\|}$
4: $x_{k+1} \leftarrow y_k - step \cdot \nabla f_k$
5: $y_{k+1} \leftarrow x_{k+1} + (a_k - 1)(x_{k+1} - x_k)/a_{k+1}$
6: $\nabla f_{k+1} \leftarrow \nabla f(y_{k+1})$
7: $ref\_step \leftarrow \frac{\|y_{k+1} - y_k\|}{\|\nabla f_{k+1} - \nabla f_k\|}$
8: **enquanto** $step > \varepsilon \cdot ref\_step$ **faça** ▷ Realiza o *backtracking*
9: $\quad step \leftarrow ref\_step$
10: $\quad x_{k+1} \leftarrow y_k - step \cdot \nabla f_k$
11: $\quad y_{k+1} \leftarrow x_{k+1} + (a_k - 1)(x_{k+1} - x_k)/a_{k+1}$
12: $\quad \nabla f_{k+1} \leftarrow \nabla f(y_{k+1})$
13: $\quad ref\_step \leftarrow \frac{\|y_{k+1} - y_k\|}{\|\nabla f_{k+1} - \nabla f_k\|}$
14: **fim enquanto**

Para mitigar este problema, é utilizado um laço de *backtracking* para ajustar o tamanho do passo com base em y_k e y_{k+1} (algoritmo 2). Inicialmente, o tamanho do passo é estimado utilizando a aproximação (3.20) (linha 3). Após o cálculo dos valores de x_{k+1} e y_{k+1} com essa estimativa, é calculado um tamanho de passo de referência (ref_step) na linha 7. Caso o tamanho de passo seja maior que a referência, é realizado o *backtracking* (linha 8 a 13). Neste processo, o tamanho do passo é definido como o valor de referência anterior, então os valores de x_{k+1} e y_{k+1} são atualizados e uma nova estimativa é realizada. O *backtracking* é repetido até que o tamanho do passo seja maior que a referência ajustada pelo fator $\varepsilon = 0.95$. Este fator é utilizado para interromper o laço precocemente, evitando repetições excessivas sem perda significativa de precisão.

3.2.3 Precondicionamento não linear

O precondicionamento é uma técnica utilizada para reduzir o número de condicionamento de um problema, tornando-o mais adequado para tratamento numérico. Em problemas de otimização, precondicionadores tradicionais aproximam o inverso da

matriz Hessiana H_f da função de objetivo f (NOCEDAL; WRIGHT, 1999). Embora esta técnica seja comum em posicionadores globais quadráticos, ela é rara em posicionadores não lineares devido à função custo não ser convexa. (LU et al., 2014) propõe uma aproximação da matriz Hessiana original H_f por uma matriz diagonal positiva definida $\widetilde{H}_f$ (3.21 e 3.22), atuando como um precondicionador Jacobiano. O vetor de gradiente é multiplicado pelo precondicionador ($\nabla f_{pre} = \widetilde{H}_f^{-1} \nabla f$) antes de ser utilizado pelo otimizador.

$$H_{f_{x,x}} \approx \widetilde{H}_{f_{x,x}} = \begin{bmatrix} \frac{\partial^2 f}{\partial x_1^2} & 0 & \cdots & 0 \\ 0 & \frac{\partial^2 f}{\partial x_2^2} & \cdots & 0 \\ \vdots & \vdots & \ddots & 0 \\ 0 & 0 & 0 & \frac{\partial^2 f}{\partial x_n^2} \end{bmatrix} \tag{3.21}$$

$$\widetilde{H}_f = \begin{bmatrix} \widetilde{H}_{f_{x,x}} & 0 \\ 0 & \widetilde{H}_{f_{y,y}} \end{bmatrix} = \widetilde{H}_W + \lambda \widetilde{H}_N \tag{3.22}$$

Como $\frac{\partial^2 f(v)}{\partial x_i^2} = \frac{\partial^2 W(v)}{\partial x_i^2} + \lambda \frac{\partial^2 N(v)}{\partial x_i^2}$, é necessário calcular o valor de $\frac{\partial^2 W(v)}{\partial x_i^2}$ e $\frac{\partial^2 N(v)}{\partial x_i^2}$ em cada iteração.

O cálculo da derivada de segunda ordem do custo do comprimento das redes ($W(v)$) é computacionalmente custoso, portanto, o precondicionador $\frac{\partial^2 W(v)}{\partial x_i^2}$ é aproximado como sendo o número de redes conectadas à respectiva instância ($|E_i|$).

A derivada de segunda ordem do custo da densidade pode ser calculada pela seguinte equação $\frac{\partial^2 N(v)}{\partial x_i^2} = -q_i \rho_{i_x}(v)$, obtida a partir das equações (3.8) e (3.12). Contudo, como o custo da densidade ($N(v)$) não é uma função convexa, é possível que $\frac{\partial^2 N(v)}{\partial x_i^2} < 0$. Neste caso, o precondicionador negativo inverteria o sentido do gradiente, aumentando o custo e levando à divergência da solução. Para evitar esse comportamento, o precondicionador é simplificado para $\frac{\partial^2 N(v)}{\partial x_i^2} \approx q_i$, ou seja, igual à área da célula. Esta simplificação visa balancear o efeito de células grandes, que possuem gradiente maior por terem uma carga elétrica elevada. De forma análoga à física clássica, onde a aceleração de um objeto depende da força e massa, utilizar a carga como precondicionador mitiga grandes diferenças de gradiente devido à disparidade de área entre as células, conduzindo a resultados melhores.

Como resultado, a equação utilizada para calcular o gradiente precondicionado para a célula *i* é:

$$\frac{\partial f_{pre}(v)}{\partial x_i} = \frac{\frac{\partial W(v)}{\partial x_i} + \lambda \frac{\partial N(v)}{\partial x_i}}{|E_i| + \lambda * q_i} \tag{3.23}$$

3.3 POSICIONAMENTO GLOBAL

O posicionamento global realizado pelo ePlace é detalhado nos Algoritmos 3 e 4. O Algoritmo 4 é responsável pelo cálculo dos gradientes, iniciando pela atualização da grade de densidade ($\rho_{m\times m}$) na linha 1, conforme a Equação (3.13). Este valor é utilizado na linha 2 para computar o gradiente do potencial elétrico $\nabla\psi_{m\times m}$ por meio de métodos espectrais. Na linha 3, calcula-se o gradiente do comprimento das redes ∇W pela equação (3.7). Finalmente, nas linhas 4 e 5, são determinados o gradiente da função custo e o gradiente precondicionado, respectivamente.

O Algoritmo 3 detalha o fluxo principal do ePlace, incluindo o método de otimização NAG, nas linhas 5 a 17. A inicialização, descrita nas linhas 1 a 3, a atualização de parâmetros, na linha 18, e a condição de parada, na linha 19, são detalhadas nas seções subsequentes.

3.3.1 Adaptações e Flexibilização do Modelo de Densidade

A aplicação direta do *eDensity* para o posicionamento global exige algumas adaptações no fluxo do algoritmo para lidar com os seguintes problemas: o posicionamento em regiões não retangulares, discrepância no tamanho das instâncias e possibilidade de áreas do circuito possuírem densidades diferentes.

Tratamento de regiões não retangulares. Para realizar o posicionamento em regiões não retangulares, utiliza-se uma região retangular que envolve toda a região de posicionamento. Instâncias retangulares imóveis e sem pinos, denominadas *dark nodes*, são inseridas de modo que cubram todas as áreas não posicionáveis. A força eletrostática gerada pelos *dark nodes* repele as instâncias móveis, mantendo-as dentro da área de posicionamento. Ao término do posicionamento global, tais instâncias são removidas.

Algoritmo 3 ePlace

Entradas: Instâncias V, redes E, região de posicionamento R, posição inicial v_0, densidade alvo ρ_t, número máximo de iterações k_{max}

Saídas: Solução v_{gp}

Inserção de *dark nodes* e *fillers* (seção 3.3.1).
Dividir R em uma grade $(m \times m)$ (3.24).
Inicializar λ_0 (3.25).
para $k \leftarrow 1$ **até** k_{max} **faça**
    $a_{k+1} \leftarrow \left(1 + \sqrt{4a_k^2 + 1}\right)/2$
    $step \leftarrow \frac{\|y_k - y_{k-1}\|}{\|\nabla f_{pre_k} - \nabla f_{pre_{k-1}}\|}$
    $x_{k+1} \leftarrow y_k - step \cdot \nabla f_k$
    $y_{k+1} \leftarrow x_{k+1} + (a_k - 1)(x_{k+1} - x_k)/a_{k+1}$
    $\nabla f_{pre_{k+1}} \leftarrow Gradiente(V, E, R, v_{k+1}, \lambda_k, \gamma_k)$
    $ref\_step \leftarrow \frac{\|y_{k+1} - y_k\|}{\|\nabla f_{pre_{k+1}} - \nabla f_{pre_k}\|}$
    enquanto $step > \varepsilon \cdot ref\_step$ **faça** ▷ Realiza o *backtracking*
        $step \leftarrow ref\_step$
        $x_{k+1} \leftarrow y_k - step \cdot \nabla f_{pre_k}$
        $y_{k+1} \leftarrow x_{k+1} + (a_k - 1)(x_{k+1} - x_k)/a_{k+1}$
        $\nabla f_{pre_{k+1}} \leftarrow Gradiente(V, E, R, v_{k+1}, \lambda_k, \gamma_k)$
        $ref\_step \leftarrow \frac{\|y_{k+1} - y_k\|}{\|\nabla f_{k+1} - \nabla f_k\|}$
    fim enquanto
    Atualiza λ_{k+1}, τ_{k+1} e γ_{k+1} (3.26) (3.27) (3.28)
    se $\tau_{k+1} \leq 10\%$ **então**
        $v_{gp} \leftarrow y_{k+1}$
        Interrompa
    fim se
fim para

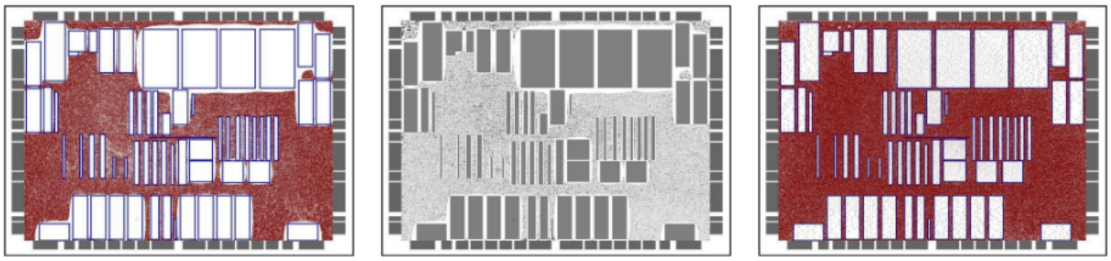
Algoritmo 4 Gradiente

Entradas: Instâncias V, redes E, região de posicionamento R, posição v_k, fator de penalidade da densidade λ_k, coeficiente do comprimento da rede γ_k

Saídas: Gradiente precondicionado ∇f_{pre}

Atualiza grade de densidade $\rho_{m \times m}$ ▷ (3.13)
Resolve o sistema (3.17) utilizando métodos espectrais para calcular $\nabla \psi_{m \times m}$.
Calcula ∇W (3.7).
$\nabla f \leftarrow \nabla W(v) + \lambda \nabla N(v)$ ▷ (3.2)
$\nabla f_{pre} \leftarrow \widetilde{H}_f^{-1} \nabla f$ ▷ (3.22)

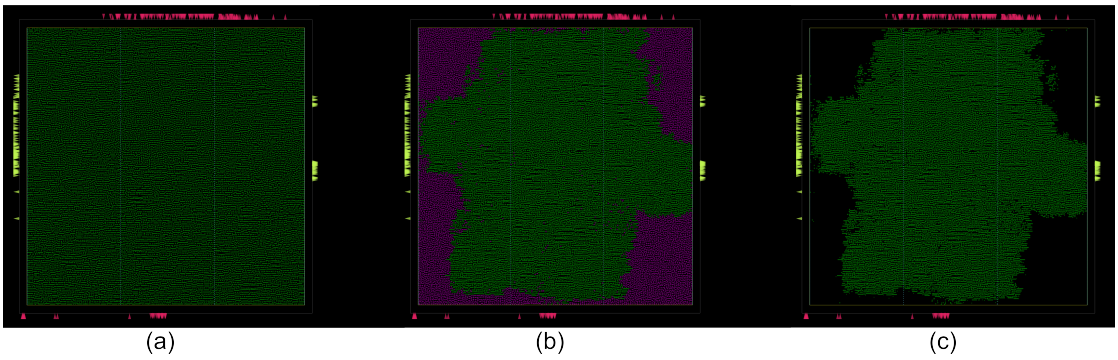
Figura 9 – Efeito da normalização de macros e instâncias fixas



Fonte: Lu et al. (2014).

Normalização de macros e instâncias fixas. Macros e instâncias fixas podem ocupar um espaço muito grande do CI, causando regiões com densidade maior que a densidade alvo. Para evitar isto, a área dessas instâncias, considerada no cálculo da densidade, é multiplicada pela densidade alvo. Desta forma, é garantido que estas instâncias contribuam com, no máximo, a densidade alvo. O efeito da normalização é ilustrado na Figura 9. Enquanto em 9.a e 9.b, onde foi feito o posicionamento sem a normalização das macros, existe uma distância entre as instâncias e as macros, em 9.c, a normalização permitiu que as instâncias fossem posicionadas próximas às macros.

Figura 10 – Efeito da inserção de *fillers*



Fonte: Elaboração própria, (2026).

Inserção de *fillers*. Para permitir que o posicionamento tenha densidade não uniforme, instâncias móveis sem pinos, denominadas *fillers*, são inseridas aleatoriamente na região de posicionamento. A área total de *fillers* inserida é calculada para

que a distribuição uniforme de todas as instâncias resulte na densidade alvo. No início do posicionamento, essas células acabam sendo empurradas para as bordas da região, enquanto as instâncias do circuito permanecem próximas entre si pela otimização das redes. Quando o fator de penalidade da densidade aumenta, as *fillers* limitam o espalhamento das instâncias do circuito, resultando em um posicionamento mais compactado. Ao final do posicionamento global, estas instâncias são removidas, resultando em áreas com a densidade inferior à máxima. Na Figura 10 é possível observar o impacto da inserção de *fillers* no posicionamento. Em 10.a, o posicionamento foi realizado sem *fillers*, resultando em um posicionamento mais esparso, enquanto em 10.b e 10.c as *standard cells* estão mais compactadas, permitindo alcançar menores comprimentos das redes.

3.3.2 Ajuste automático de parâmetros

Dimensão da grade de densidade. O ePlace calcula a densidade de carga discretizando o espaço através de uma grade (3.13). O tamanho da grade, denotado por $m \times m$, é calculado a partir do número total n de instâncias do circuito e de *fillers*, sendo definido por:

$$m = \lceil log_2(n) \rceil \tag{3.24}$$

Fator de penalidade da densidade. O valor inicial da penalidade de densidade é definido pela equação (3.25), onde v_0 é a posição inicial e V'_m é o conjunto de todas as instâncias móveis.

$$\lambda_0 = \frac{\sum_{i \in V'_m} \left(\left| \frac{\partial W(v_0)}{\partial x_i} \right| + \left| \frac{\partial W(v_0)}{\partial y_i} \right| \right)}{\sum_{i \in V'_m} \left(\left| \frac{\partial N(v_0)}{\partial x_i} \right| + \left| \frac{\partial N(v_0)}{\partial y_i} \right| \right)} \tag{3.25}$$

Tradicionalmente, é utilizado um valor constante para λ durante a convergência local. No ePlace, este fator é atualizado em cada iteração por meio da relação $\lambda_k = \mu_k \cdot \lambda_{k-1}$. O multiplicador μ_k é determinado pela variação de HPWL, em que $\Delta HPWL_k = HPWL(v_k) - HPWL(v_{k-1})$, conforme a equação (3.26). O parâmetro μ_0 é uma constante predefinida e $HPWL_{ref}$ é o aumento esperado de HPWL por iteração,

sendo $\mu_0 = 1.1$ e $HPWL_{ref} = 3,5 \cdot 10^5$ os valores utilizados pelo ePlace. O multiplicador μ_k é limitado entre 0.75 e 1.1 para evitar instabilidade na penalidade.

$$\mu_k = \mu_0^{1-\frac{\Delta HPWL_k}{\Delta HPWL_{ref}}} \tag{3.26}$$

Critério de parada. O critério de parada do otimizador baseia-se na quantidade de sobreposição das instâncias, medido pelo valor de *overflow* τ da grade de densidade. O *overflow* é calculado pela equação (3.27), onde A_b é a área da grade b, A_i é a área da instância i e ρ'_b é a densidade das instâncias móveis na grade b. O posicionamento global é terminado quando o valor de *overflow* se torna inferior a τ_{min}, tipicamente 10%.

$$\tau = \frac{\sum_{b \in B} max(\rho'_b - \rho_t, 0) A_b}{\sum_{i \in V'_m} A_i} \tag{3.27}$$

Coeficiente do comprimento da rede. O HPWL de média ponderada (3.6) utiliza um coeficiente γ para controlar a precisão da aproximação do HPWL. No início do posicionamento este coeficiente é relaxado, incentivando as instâncias a se dispersarem devido à força eletrostática e com o passar das iterações este parâmetro é reduzido fazendo ao valor se aproximar mais ao HPWL. Como o gradiente da densidade é mais sensível em grades menores, a atualização do coeficiente γ foi modelado com base no *overflow* e na dimensão da grade utilizando a equação:

$$\gamma(\tau) = 8.0 w_b \times 10^{\frac{20}{9}\tau - \frac{11}{9}} \tag{3.28}$$

3.4 TRABALHOS DERIVADOS DO EPLACE

O ePlace é a base de várias técnicas recentes de posicionamento global, dentre as quais destacam-se:

- O **RePlAce** expande o ePlace para tratar a roteabilidade do circuito (CHENG et al., 2019);

- O **DREAMPlace** acelera o ePlace paralelizando os cálculos de gradiente utilizando GPU (LIN et al., 2019);
- O **Xplace** utiliza GPU para aceleração, mas também faz a previsão do campo elétrico utilizando redes neurais para alcançar melhores resultados (LIU et al., 2022);
- O **Pplace-ms** acelera o ePlace modificando a definição da energia do potencial do sistema de forma que seja rápido calcular uma aproximação de suas derivadas parciais (PENG; ZHU, 2024).

4 METODOLOGIA

A fim de se realizar a análise experimental prevista no objetivo principal deste Trabalho de Conclusão de Curso, o algoritmo de posicionamento global ePlace foi implementado dentro do projeto OpenROAD, utilizando a linguagem C++ sob o paradigma de orientação a objetos. A interface e integração com o fluxo de controle foi desenvolvida utilizando *SWIG* e *Tcl*, permitindo a comunicação entre o algoritmo desenvolvido e a infraestrutura do OpenROAD Flow Scripts.

4.1 OPENROAD

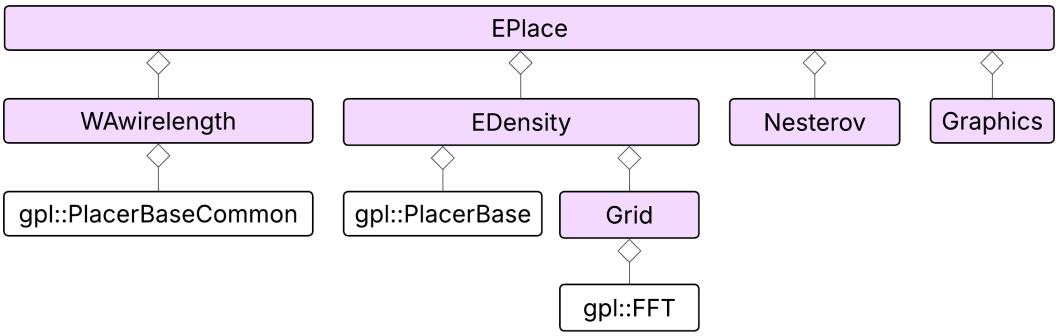
OpenROAD é um projeto *open-source* de ferramentas para automatização do projeto físico de circuitos digitais, visando a conclusão do processo em 24 horas sem intervenção humana. O projeto busca a democratização do desenvolvimento de *hardware* diminuindo as barreiras de custo, experiência e tempo que atrapalham projetistas de sistemas. O OpenROAD é composto de ferramentas modulares integradas através de um banco de dados compartilhado, chamado de OpenDB, permitindo que as diversas etapas do fluxo interajam entre si durante a síntese (AJAYI et al., 2019).

4.2 IMPLEMENTAÇÃO DO ALGORITMO EPLACE

O algoritmo ePlace foi implementado como uma ferramenta integrada ao projeto OpenROAD. Sua implementação foi realizada através de cinco novas classes (*EPlace*, *WAwirelength*, *EDensity*, *Nesterov* e *Grid*) e da reutilização de três classes do módulo de posicionamento global *GPL* (*gpl::PlacerBaseCommon*, *gpl::PlacerBase* e *gpl::FFT*). A Figura 11 ilustra o diagrama de classes simplificado com as dependências entre os componentes desenvolvidos e as classes do módulo *GPL*.

A classe principal, *EPlace*, é responsável pelo laço principal do posicionamento (Algoritmo 3), executando o cálculo do gradiente da função de custo (Equação (3.23)) e a atualização dos parâmetros (Seção 3.3.2). As classes *WAwirelength* e *EDensity* são responsáveis por calcular, respectivamente, os gradientes do comprimento das redes (Equação 3.7) e da penalidade de densidade (Equação 3.9).

Figura 11 – Diagrama de classes simplificado



Fonte: Elaboração própria, (2026).

As classes *gpl::PlacerBaseCommon* e *gpl::PlacerBase* são utilizadas para carregar e pré-processar as informações do banco de dados *OpenDB*. Como o *GPL* possui suporte para múltiplas regiões de posicionamento, estas classes são estruturadas de forma que o *gpl::PlacerBaseCommon* armazena a informação das instâncias em todas as regiões, enquanto o *gpl::PlacerBase* armazena apenas de uma região específica. Embora utilize estas classes do OpenROAD, o algoritmo implementado não possui suporte para múltiplas regiões. A classe *Grid* realiza a discretização da região de posicionamento, calculando as áreas de sobreposição entre instâncias e a grade, enquanto a classe *gpl::FFT* é utilizada para resolver o sistema de equações (3.17) e determinar o campo elétrico. Por fim, o algoritmo de otimização NAG é implementado pela classe *Nesterov*.

Além da implementação do algoritmo, foi desenvolvida a classe *Graphics* para integrar o EPlace à interface gráfica do OpenROAD. Esta classe é utilizada tanto para a depuração do código quanto para a geração de representações visuais, como exemplificado na Figura 10, que ilustra o processo de inserção de *fillers*.

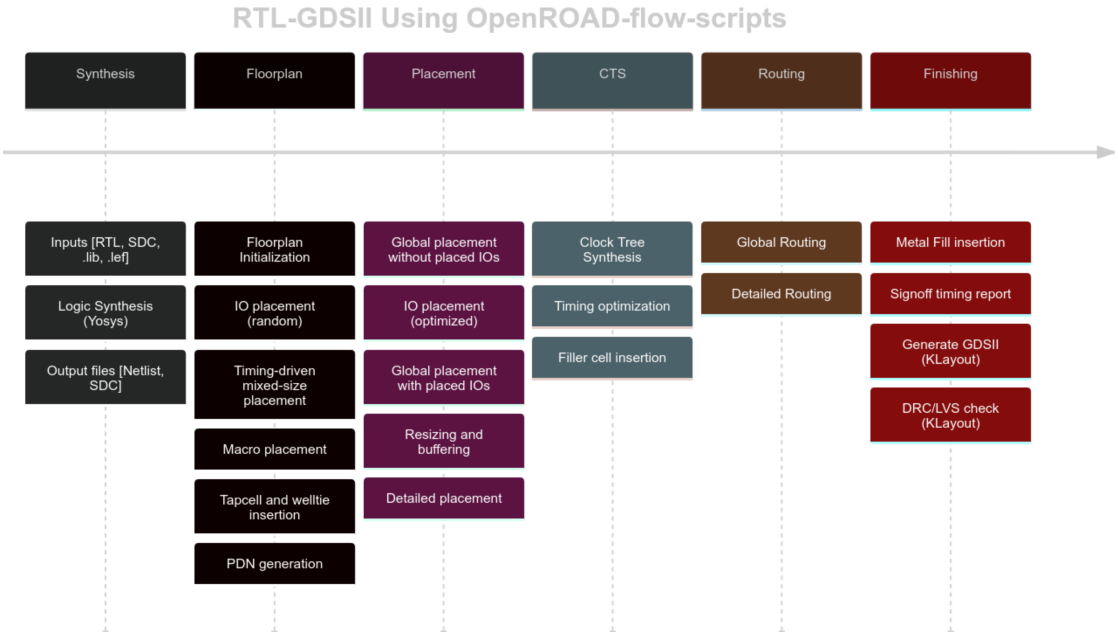
5 RESULTADOS

Este capítulo discute os experimentos realizados para avaliar a técnica implementada. A Seção 5.1 detalha como os experimentos que foram realizados, enquanto a seção 5.2 apresenta os resultados obtidos.

5.1 METODOLOGIA EXPERIMENTAL

Os experimentos foram realizados em uma estação de trabalho com o sistema Linux 24.04.4 LTS, equipada com um processador AMD® Ryzen® 9 9950X de 4.3 GHz (16 núcleos e 32 threads) e 32 GB de memória RAM DDR5 (4800 MHz).

Figura 12 – Fluxo de síntese física do OpenROAD



Fonte: Ajayi et al. (2019).

Para viabilizar a validação do algoritmo, a implementação foi integrada ao fluxo de síntese do OpenROAD Flow Scripts (Figura 12), substituindo a ferramenta de posicionamento global. Essa integração permite realizar o fluxo completo de síntese física, garantindo que a técnica de posicionamento implementada seja submetida às

mesmas condições operacionais e métricas de desempenho que as demais ferramentas do conjunto, assegurando assim a confiabilidade e a reprodutibilidade dos resultados experimentais obtidos.

Os testes foram realizados utilizando os circuitos disponíveis no repositório OpenROAD Flow Scripts. Para a escolha dos *benchmarks*, foram selecionados os circuitos presentes na maioria das seis bibliotecas de *standard cells* disponíveis no repositório, sendo eles:

- *aes*, que implementa o algoritmo de criptografia AES;
- *ibex*, uma CPU de arquitetura RISC-V de 32 bits;
- *jpeg*, um codificador de imagens no formato JPEG;
- *riscv32i*, uma implementação simplificada de um processador RISC-V de 32 bits.

Para a validação do algoritmo, foram utilizadas as seis bibliotecas de *standard cells* disponíveis, abrangendo distintos nós tecnológicos. As principais características dessas tecnologias, incluindo o tamanho do nó tecnológico e a viabilidade de fabricação comercial, estão presentes na tabela 1.

Tabela 1 – Características das bibliotecas de *standard cells*

Biblioteca	**Fabricável**	**Nó tecnológico**
ASAP7	-	7 nm
GF180MCU	✓	180 nm
IHP SG13G2	✓	130 nm
NANGATE45	-	45 nm
SKY130HD	✓	130 nm
SKY130HS	✓	130 nm

A quantidade de instâncias para cada caso de teste é apresentada na Tabela 2. Observa-se que o circuito *riscv32i* não possui uma configuração nativa para a biblioteca *Nangate45* no repositório OpenROAD Flow Scripts, e por este motivo, tal combinação não foi incluída nos experimentos.

Tabela 2 – Quantidade de instâncias por circuito

	asap7	**gf180**	**ihp-sg13g2**	**nangate45**	**sky130hd**	**sky130hs**
aes	15389	16489	13296	12697	11036	10962
ibex	19180	12091	16229	16448	14292	12589
jpeg	55618	39078	69595	61423	36990	34333
riscv32i	9112	6158	8084		4621	4599

O posicionamento realizado pelo algoritmo implementado (ePlace) foi comparado ao posicionador global padrão do projeto OpenROAD, baseado no RePlAce de Cheng et al. (2019). Embora o RePlAce também utilize o modelo eletrostático para o posicionamento global, ele também utiliza métricas de congestionamento para otimizar a roteabilidade do circuito. Adicionalmente, a implementação do RePlAce no OpenROAD incorpora técnicas para reduzir violações de *timing* durante o posicionamento, fazendo a inserção de *buffers* e utilizando pesos para os caminhos críticos. Visto que o algoritmo implementado não incorpora técnicas para medir diretamente congestionamento ou *timing*, a comparação foi realizada utilizando duas configurações: uma desabilitando tais recursos (RePlAce-ref), para permitir uma análise direta do desempenho do modelo eletrostático, e outra com os recursos habilitados (RePlAce), para avaliar o impacto das otimizações adicionais presentes no fluxo padrão. Além disso, as duas configurações do RePlAce foram executadas com a opção de paralelismo (multithreading), usada apenas para a inicialização das estruturas.

Tabela 3 – Posicionadores globais utilizados na avaliação experimental

		Guiado por	
Nome	**Implementação**	**congestionamento**	***timing***
RePlAce-ref	OpenROAD	-	-
ePlace	Este trabalho	-	-
RePlAce	OpenROAD	✓	✓

Para aproximar a comparação entre as implementações, alguns parâmetros do ePlace foram ajustados de acordo com os valores utilizados no OpenROAD. O fator de penalidade da densidade inicial, definido pela equação (3.25), é multiplicado por $8 \cdot 10^{-5}$, enquanto o aumento esperado de HPWL por iteração, $HPWL_{ref}$ na equação (3.26), foi configurado como $4,46 \cdot 10^{8}$ em unidades do banco de dados *OpenDB*. Ressalta-se que, mesmo com estas modificações, o algoritmo de posicionamento glo-

bal do OpenROAD apresenta outras diferenças de implementação e configuração que não foram exploradas, visto que a investigação exaustiva destes detalhes foge ao escopo deste trabalho.

5.2 RESULTADOS EXPERIMENTAIS

As Tabelas 4, 5 e 6 apresentam os resultados obtidos nos experimentos, medidos após diferentes etapas do fluxo de síntese. Cada linha detalha as métricas para uma combinação específica de biblioteca de *standard cells* e circuito, incluindo as linhas com a média por biblioteca e a média global do *benchmark*. A organização das colunas para cada métrica é estruturada da seguinte forma: a coluna mais à esquerda apresenta os valores obtidos com o fluxo RePlAce-ref (posicionador global do OpenROAD com as técnicas de otimização de congestionamento e *timing* desabilitadas), os quais foram usados como referência (*baseline*); a coluna do meio exibe a variação percentual dos resultados do ePlace em relação aos resultados do RePlAce-ref; e a coluna mais à direita, RePlAce, apresenta a variação percentual dos resultados do fluxo padrão do OpenROAD em relação aos resultados do RePlAce-ref. Os valores em negrito destacam o melhor resultado obtido entre os três posicionadores avaliados.

A tabela 4 detalha as métricas obtidas na etapa de posicionamento global, abrangendo o comprimento aproximado das redes (HPWL), o número de iterações do NAG e o tempo de execução. É notável que, embora a técnica implementada apresente um HPWL médio $20,2\%$ pior que a referência, o tempo de execução foi reduzido em $34,0\%$. Já o RePlAce (fluxo padrão do OpenROAD) exibiu resultados piores comparados com a referência, tanto em HPWL quanto em tempo de execução. Os tempos de execução mais longos do RePlAce devem-se à aplicação das técnicas de correção de *timing* e de mitigação do congestionamento das redes. Esta última abre espaço em regiões de alto congestionamento para facilitar o roteamento, consequentemente, aumenta a distância entre as instâncias e comprimento das redes que as conectam.

Na tabela 5 são apresentados os resultados obtidos após a etapa de posicionamento detalhado, utilizando como métricas de avaliação o HPWL, a frequência máxima estimada e o deslocamento total das instâncias entre o resultado do posicionamento global e a posição legalizada. Após a legalização, a diferença média de HPWL da técnica implementada decresce para $11,0\%$. Além disso, apesar do maior comprimento

Tabela 4 – Resultados após o posicionamento global

		HPWL (μm)			Iterações			Tempo de execução (s)		
Biblioteca	Circuito	RePlAce-ref	ePlace	RePlAce	RePlAce-ref	ePlace	RePlAce	RePlAce-ref	ePlace	RePlAce
asap7	aes	61.614	47,0%	**-7,7%**	373	-0,5%	**-3,8%**	51	**-8,8%**	46,2%
	ibex	76.556	11,4%	12,2%	422	**-23,2%**	0,0%	103	11,2%	46,0%
	jpeg	265.335	61,5%	**-31,1%**	433	**-8,1%**	-1,6%	390	**-42,6%**	2,8%
	riscv32i	51.803	6,8%	16,5%	407	**-57,0%**	0,2%	110	**-80,6%**	46,2%
	Média	-	31,7%	**-2,5%**	-	**-22,2%**	-1,3%	-	**-30,2%**	35,3%
gf180	aes	1.304.981	1,3%	**-5,4%**	469	**-34,1%**	102,1%	121	**-32,8%**	157,6%
	ibex	996.322	11,3%	11,0%	428	**-53,7%**	0,7%	107	**-72,3%**	27,9%
	jpeg	2.594.169	42,4%	8,1%	477	**-53,0%**	0,8%	497	**-75,7%**	-3,8%
	riscv32i	451.187	4,8%	26,6%	444	**-54,5%**	84,5%	53	**-60,3%**	121,0%
	Média	-	15,0%	10,1%	-	**-48,8%**	47,0%	-	**-60,3%**	75,7%
ihp-sg13g2	aes	578.235	13,4%	23,1%	399	**-21,3%**	1,5%	155	**-28,9%**	22,6%
	ibex	623.917	6,0%	24,6%	491	**-59,7%**	-0,2%	125	**-50,8%**	13,6%
	jpeg	2.179.257	118,3%	37,4%	523	7,1%	0,4%	363	85,8%	79,3%
	riscv32i	331.185	3,6%	25,6%	432	**-55,3%**	2,1%	44	**-56,9%**	197,4%
	Média	-	35,3%	27,7%	-	**-32,3%**	0,9%	-	**-12,7%**	78,3%
nangate45	aes	201.538	0,9%	22,0%	444	**-31,5%**	75,0%	79	2,6%	248,8%
	ibex	316.901	7,0%	**-14,8%**	453	**-29,6%**	20,3%	139	**-30,6%**	3,2%
	jpeg	954.448	10,1%	**-31,8%**	455	**-56,0%**	123,3%	267	**-8,9%**	326,8%
	Média	-	6,0%	**-8,2%**	-	**-39,1%**	72,9%	-	**-12,3%**	192,9%
sky130hd	aes	382.452	2,3%	41,8%	395	**-21,3%**	42,5%	170	**-56,1%**	23,0%
	ibex	460.179	37,1%	26,9%	461	3,9%	336,9%	137	9,2%	236,5%
	jpeg	1.064.484	28,2%	25,6%	468	**-36,5%**	376,1%	527	**-77,6%**	271,0%
	riscv32i	182.436	7,0%	22,4%	431	**-29,7%**	262,2%	57	**-57,6%**	136,9%
	Média	-	18,7%	29,2%	-	**-20,9%**	254,4%	-	**-45,5%**	166,9%
sky130hs	aes	414.164	2,4%	38,2%	399	**-22,3%**	100,5%	142	**-45,3%**	-2,0%
	ibex	547.584	8,8%	31,7%	436	**-27,1%**	464,7%	90	**-4,4%**	355,8%
	jpeg	1.340.247	28,6%	17,8%	466	**-39,3%**	677,9%	342	**-66,8%**	814,5%
	riscv32i	205.033	5,2%	25,2%	429	**-28,2%**	366,7%	38	**-33,7%**	300,4%
	Média	-	11,2%	28,2%	-	**-29,2%**	402,4%	-	**-37,6%**	367,2%
Média		-	20,2%	15,0%	-	**-31,8%**	131,9%	-	**-34,0%**	150,9%

de fio, a estimativa de frequência máxima é $1,2\%$ superior à da referência. A redução dessa diferença pode ser explicada pelo deslocamento total durante o posicionamento detalhado: no fluxo utilizando o ePlace, o deslocamento foi, em média $23,9\%$ menor, indicando que o resultado do posicionamento global apresentava menor sobreposição de instâncias. Essa discrepância provavelmente é devida ao cálculo da dimensão da grade de densidade (3.24). O posicionador global do OpenROAD utiliza uma abordagem distinta da proposta original do ePlace, resultando em grades menores que causa um erro de discretização maior durante o cálculo do *overflow*, o que gera um posicionamento com maior sobreposição de instâncias. Adicionalmente, é possível notar o efeito da técnica de correção de *timing* utilizada pelo RePlAce do OpenROAD durante o posicionamento global: a estimativa de frequência máxima dos circuitos é, em média $46,4\%$ maior que o mesmo algoritmo sem tal otimização.

Tabela 5 – Resultados do posicionamento detalhado

		HPWL (μm)			Estimativa de frequência máxima (MHz)			Deslocamento total (μm)		
Biblioteca	Circuito	RePlAce-ref	ePlace	RePlAce	RePlAce-ref	ePlace	RePlAce	RePlAce-ref	ePlace	RePlAce
asap7	aes	51.042	25,4%	**-9,9%**	1.810	-0,7%	**35,2%**	4.374	**-33,2%**	10,3%
	ibex	65.281	8,1%	6,4%	624	2,5%	**55,7%**	4.085	**-39,5%**	-0,9%
	jpeg	221.520	42,2%	**-37,7%**	876	12,1%	**58,4%**	14.810	**-31,3%**	2,2%
	riscv32i	40.067	5,0%	11,3%	634	-8,2%	**44,4%**	3.674	0,2%	**-14,9%**
	Média	-	20,2%	**-7,5%**	-	1,4%	**48,4%**	-	**-25,9%**	-0,8%
gf180	aes	1.068.095	-2,7%	**-7,8%**	166	1,4%	**47,0%**	50.757	**-45,7%**	37,8%
	ibex	865.385	2,7%	5,4%	64	0,6%	**57,0%**	67.050	**-35,2%**	0,4%
	jpeg	1.967.155	27,3%	3,8%	120	4,8%	**31,2%**	192.928	**-39,3%**	-3,6%
	riscv32i	383.847	0,1%	19,9%	72	-0,2%	**39,3%**	30.221	**-15,9%**	16,3%
	Média	-	6,8%	5,3%	-	1,7%	**43,6%**	-	**-34,0%**	12,7%
ihp-sg13g2	aes	592.482	0,6%	**-2,8%**	145	5,3%	**78,1%**	44.083	**-17,5%**	-17,1%
	ibex	693.824	7,8%	**-7,5%**	92	-8,8%	**37,7%**	59.874	81,5%	**-8,0%**
	jpeg	2.552.097	53,3%	**-5,8%**	62	2,7%	**113,0%**	172.029	**-39,3%**	-9,0%
	riscv32i	438.745	0,1%	**-24,8%**	137	4,4%	**40,5%**	23.615	**-0,5%**	2,2%
	Média	-	15,5%	**-10,2%**	-	0,9%	**67,3%**	-	6,1%	**-8,0%**
nangate45	aes	168.569	**-5,1%**	15,0%	730	0,2%	**67,7%**	9.043	**-34,3%**	12,6%
	ibex	264.154	1,8%	**-18,9%**	373	-0,8%	**22,3%**	14.167	**-43,5%**	-3,4%
	jpeg	796.138	7,7%	**-41,1%**	567	-0,8%	**53,8%**	87.702	**-38,6%**	8,7%
	Média	-	1,5%	**-15,0%**	-	-0,5%	**47,9%**	-	**-38,8%**	6,0%
sky130hd	aes	333.742	**-3,4%**	30,4%	146	3,4%	**45,3%**	25.667	**-39,9%**	14,7%
	ibex	416.057	15,5%	16,5%	67	12,5%	**47,4%**	32.474	**-10,6%**	-4,6%
	jpeg	793.635	30,6%	21,8%	116	-8,7%	**43,6%**	88.321	**-22,7%**	6,6%
	riscv32i	171.150	3,4%	4,1%	118	1,6%	**21,6%**	12.124	**-19,7%**	-3,8%
	Média	-	11,5%	18,2%	-	2,2%	**39,5%**	-	**-23,2%**	3,2%
sky130hs	aes	361.310	**-2,3%**	27,1%	220	-9,4%	**38,6%**	33.405	**-39,9%**	8,3%
	ibex	516.207	**-0,1%**	17,5%	108	0,3%	**29,0%**	47.389	**-45,2%**	-5,3%
	jpeg	1.006.409	32,9%	14,1%	185	8,3%	**39,2%**	113.317	**-24,7%**	-3,8%
	riscv32i	193.388	1,6%	2,7%	180	4,4%	**20,5%**	17.781	**-14,7%**	-4,6%
	Média	-	8,0%	15,3%	-	0,9%	**31,8%**	-	**-31,1%**	-1,3%
Média		-	11,0%	1,7%	-	1,2%	**46,4%**	-	**-23,9%**	1,8%

A tabela 6 detalha os resultados ao final do fluxo de síntese, apresentando as três principais métricas de qualidade (*Quality of Results* - QoR) de um circuito: a potência, a performance (frequência máxima) e a área (área de instâncias). Os resultados marcados como NF (Não finalizou) se referem aos casos que não finalizaram o fluxo devido a problemas de congestionamento durante as etapas de roteamento global ou detalhado. Excepcionalmente, o teste do circuito *jpeg* com a biblioteca *Nangate45* utilizando o fluxo ePlace (marcado como NF^*) falhou devido a um erro interno na ferramenta de roteamento global do OpenROAD. O fluxo de referência (RePlAce-ref) apresentou apenas 2 falhas durante o roteamento, ambas utilizando a biblioteca *sky130hd* (versão de alta densidade da tecnologia *SKY130*), enquanto o fluxo com o algoritmo implementado falhou em 5 casos devido à congestionamento (3 na biblioteca *sky130hd* e 2 na *sky130hs*).

Apesar dos casos de falha, os resultados finais do fluxo utilizando o ePlace

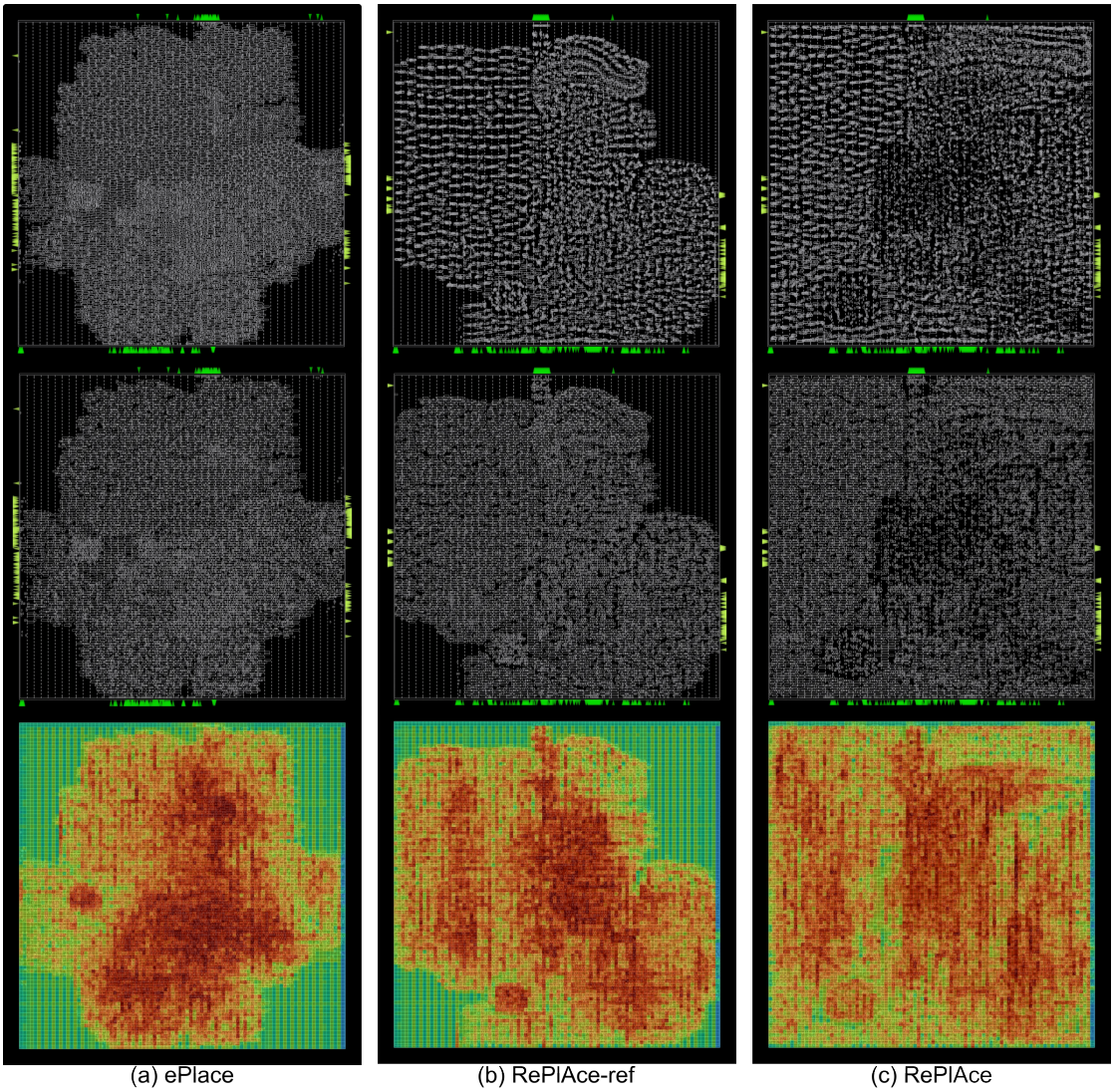
Tabela 6 – Resultados ao final do fluxo de síntese física

		Potência (mW)			Frequência máxima (MHz)			Área de instâncias (μm^2)		
Biblioteca	Circuito	RePlAce-ref	ePlace	RePlAce	RePlAce-ref	ePlace	RePlAce	RePlAce-ref	ePlace	RePlAce
asap7	aes	170	3,9%	**-2,5%**	2.473	-1,1%	-0,7%	1.865	-0,2%	**-1,7%**
	ibex	24	7,8%	0,4%	948	-3,0%	**3,2%**	2.572	0,7%	**-1,5%**
	jpeg	134	9,8%	**-10,8%**	1.348	3,1%	**11,3%**	6.577	0,3%	**-2,1%**
	riscv32i	31	0,5%	**-3,5%**	955	-11,6%	**6,1%**	2.727	-0,9%	**-1,0%**
	Média	-	5,5%	**-4,1%**	-	-3,1%	**5,0%**	-	-0,1%	**-1,6%**
gf180	aes	13.046	0,2%	5,4%	252	-2,9%	**2,0%**	655.940	0,3%	13,4%
	ibex	662	6,7%	**-1,5%**	99	-3,0%	-0,4%	697.090	1,2%	**-1,3%**
	jpeg	18.874	1,5%	0,3%	123	4,3%	**25,9%**	2.102.790	0,0%	**-0,6%**
	riscv32i	374	0,8%	4,7%	109	0,3%	**1,4%**	363.367	0,1%	3,5%
	Média	-	2,3%	2,2%	-	-0,3%	**7,2%**	-	0,4%	3,7%
ihp-sg13g2	aes	166	-0,1%	**-2,8%**	228	-4,3%	**20,0%**	177.708	-0,4%	**-1,5%**
	ibex	58	9,2%	**-3,3%**	124	-7,9%	**2,6%**	280.060	0,7%	**-1,4%**
	jpeg	160	19,8%	1,9%	134	-8,5%	**2,0%**	914.938	0,2%	**-1,0%**
	riscv32i	19	0,1%	**-4,4%**	175	2,4%	**3,6%**	144.466	0,2%	**-2,2%**
	Média	-	7,2%	**-2,2%**	-	-4,6%	**7,1%**	-	0,2%	**-1,5%**
nangate45	aes	309	**-0,8%**	1,9%	1.142	-3,5%	**6,0%**	19.193	0,7%	3,8%
	ibex	85	5,6%	**-4,6%**	435	-0,4%	**3,8%**	30.963	0,8%	**-3,0%**
	jpeg	590	*NF**	**-3,2%**	725	*NF**	**22,2%**	92.823	*NF**	2,0%
	Média	-	2,4%	**-2,0%**	-	-1,9%	**10,7%**	-	0,8%	0,9%
sky130hd	aes	NF	NF	**572**	NF	NF	**238**	NF	NF	**127.082**
	ibex	NF	NF	**36**	NF	NF	**99**	NF	NF	**156.982**
	jpeg	442	NF	13,7%	162	NF	**10,6%**	451.380	NF	5,2%
	riscv32i	23	3,5%	3,8%	143	-1,5%	**3,3%**	75.336	1,0%	2,1%
	Média	-	3,5%	8,7%	-	-1,5%	**7,0%**	-	1,0%	3,7%
sky130hs	aes	984	NF	4,0%	349	NF	**2,1%**	142.684	NF	7,6%
	ibex	83	2,9%	0,5%	129	3,9%	**6,2%**	219.002	1,3%	**-0,1%**
	jpeg	1.642	5,5%	**-1,6%**	249	-11,6%	**4,5%**	690.594	1,7%	**-1,0%**
	riscv32i	48	NF	4,3%	203	NF	**5,4%**	101.904	NF	4,5%
	Média	-	4,2%	1,8%	-	-3,9%	**4,5%**	-	1,5%	2,7%
Média		-	4,5%	0,1%	-	-2,7%	**6,7%**	-	0,4%	1,1%

foram próximos à referência, em média $4,5\%$ maior potência, frequência $2,7\%$ menor e $0,4\%$ aumento da área de instâncias. Estas diferenças provavelmente são impacto direto da maior área de instâncias e maior HPWL durante o posicionamento, apesar dos resultados de estimativa de frequência serem melhores logo após o posicionamento detalhado, o roteamento foi bastante impactado pelo congestionamento do circuito. O fluxo padrão do OpenROAD apesentou melhoria significativa na frequência máxima, em média $6,7\%$ maior, sem aumentar significativamente a potência consumida, em média apenas $0,1\%$ maior.

A Figura 13 apresenta o posicionamento global, o posicionamento final e o mapa de calor do congestionamento para o circuito *ibex*, utilizando a biblioteca *sky130hs* para os três fluxos testados. Na primeira linha dessa imagem é possível notar a diferença no espalhamento das instâncias entre as soluções: o ePlace reduz significativamente a

Figura 13 – Posicionamento global, posicionamento final e congestionamento final (*sky130hs* - ibex)



(a) ePlace (b) RePlAce-ref (c) RePlAce

Fonte: Elaboração própria (2026).

sobreposição das instâncias, enquanto as soluções do RePlAce-ref e RePlAce possuem um grande número de instâncias sobrepostas em pequenos agrupamentos densos. Essa característica também pode ser observada ao comparar o posicionamento global (primeira linha) com o final (segunda linha): o posicionamento global gerado pelo ePlace sofre menos modificações do que os posicionamentos gerados pelo RePlAce-ref e RePlAce. Por fim, na terceira linha é possível notar o efeito da técnica de mitigação de congestionamento aplicada pelo RePlAce: as regiões que teriam alto congestionamento (indicadas em vermelho) têm suas instâncias artificialmente infladas, gerando maior distanciamento entre as células dessas regiões e, desta forma, reduzindo a densidade de pinos e o congestionamento. Essa técnica também é responsável pelo melhor aproveitamento da região de posicionamento do RePlAce. Quando ocorre a inflação das células, também é realizada a retirada de células *fillers* para compensar o aumento artificial das áreas das instâncias. Com isso, as instâncias do circuito continuam sob a mesma restrição de densidade máxima, mas as áreas de baixa densidade são utilizadas para diminuir o congestionamento das redes.

6 CONCLUSÃO

Este trabalho avaliou o algoritmo clássico de posicionamento global baseado em eletrostática, o ePlace, integrado a um fluxo de síntese física completo. A implementação foi realizada utilizando a infraestrutura do projeto *open-source* OpenROAD, permitindo uma comparação direta entre a implementação do ePlace e o algoritmo de posicionamento global baseado no RePlAce, nativo da ferramenta.

Os experimentos utilizaram como referência o RePlAce do OpenROAD sem as técnicas de correção de *timing* e de mitigação de congestionamento das redes (batizado de RePlAce-ref), de modo a aproximar o comportamento do ePlace original. Essa referência foi comparada tanto com a implementação realizada neste trabalho quanto com o posicionador padrão do OpenROAD com todas as técnicas de otimização habilitadas.

Os resultados demonstram que a técnica implementada apresentou um HPWL após o posicionamento global $20,2\%$ pior que a referência, porém com tempo de execução $34,0\%$ menor, enquanto o fluxo padrão do OpenROAD teve um resultado $15,0\%$ pior que a referência e foi em média $150,9\%$ mais lento devido às técnicas adicionais. Após o posicionamento detalhado, a diferença de HPWL reduziu para $11,0\%$ para o ePlace e $1,7\%$ para o RePlace. A técnica implementada exigiu em média $23,9\%$ menos deslocamento das instâncias durante a legalização do posicionamento. Esse dado indica que a solução gerada pelo posicionador global implementado promove uma dispersão de células mais eficiente que a solução da ferramenta de posicionamento do OpenROAD. Adicionalmente, após o posicionamento detalhado, a estimativa de frequência máxima superou a referência em $1,2\%$, apesar do HPWL pior. Em comparação, o RePlAce alcançou uma frequência $46,4\%$ maior, o que demonstra a eficácia da técnica de correção de *timing*.

Ao final do fluxo de síntese, 6 dos 23 casos de teste falharam com a implementação realizada. Dentre essas falhas, 5 ocorreram devido ao alto congestionamento durante as etapas de roteamento e uma por erro interno na ferramenta de roteamento global. Em contrapartida, a referência falhou em apenas 2 casos (devido ao congestionamento), e o RePlAce concluiu todos os testes com sucesso. Para os circuitos que finalizaram o fluxo, a potência consumida foi, em média, $4,5\%$ maior e a frequência máxima foi $2,7\%$ inferior à referência, enquanto o RePlAce alcançou potência $4,5\%$ pior e frequência $6,7\%$ melhor.

As configurações dos circuitos de teste adotadas foram as presentes no OpenROAD Flow Scripts, sendo estes os mesmos parâmetros utilizados no desenvolvimento das ferramentas originais. Tais configurações são normalmente ajustadas para elevar o nível de exigência dos experimentos, permitindo observar variações nos resultados mesmo com mudanças pequenas no software. Visto que o ePlace e o RePlAce-ref não possuem mecanismos de otimização de *timing* e de congestionamento, era esperado que alguns casos de teste falhassem.

Em futuras investigações, seria possível relaxar as restrições dos casos de teste falhos, como a frequência alvo, a densidade alvo, a área do chip e o uso máximo das camadas de roteamento, permitindo chegar ao final do fluxo de síntese em todos os testes realizados. Além disso, este trabalho não explorou diferentes parâmetros de configuração do algoritmo implementado. Por exemplo, foi observado que o número de iterações do NAG no posicionador implementado foi cerca de $31,8\%$ menor que a referência. Ao alterar os parâmetros de $HWPL_{ref}$ (3.26) ou o multiplicador μ_0 (3.26), seria possível desacelerar a convergência, potencialmente alcançando resultados superiores.

REFERÊNCIAS

AJAYI, T. et al. Toward an Open-Source Digital Flow: First Learnings from the OpenROAD Project. In: **Proceedings of the 56th Annual Design Automation Conference 2019**. Las Vegas NV USA: ACM, 2019. p. 1–4. ISBN 978-1-4503-6725-7. Disponível em: https://dl.acm.org/doi/10.1145/3316781.3326334.

CHENG, C.-K. et al. RePlAce: Advancing Solution Quality and Routability Validation in Global Placement. **IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems**, v. 38, n. 9, p. 1717–1730, set. 2019. ISSN 0278-0070, 1937-4151. Disponível em: https://ieeexplore.ieee.org/document/8418790/.

CHU, C. CHAPTER 11 - Placement. In: WANG, L.-T.; CHANG, Y.-W.; CHENG, K.-T. T. (Ed.). **Electronic Design Automation**. Boston: Morgan Kaufmann, 2009. p. 635–685. ISBN 978-0-12-374364-0. Disponível em: https://www.sciencedirect.com/science/article/pii/B9780123743640500187.

HSU, M.-K.; BALABANOV, V.; CHANG, Y.-W. TSV-Aware Analytical Placement for 3-D IC Designs Based on a Novel Weighted-Average Wirelength Model. **IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems**, v. 32, n. 4, p. 497–509, abr. 2013. ISSN 0278-0070, 1937-4151. Disponível em: http://ieeexplore.ieee.org/document/6480847/.

KAHNG, A.; REDA, S. Wirelength minimization for min-cut placements via placement feedback. **IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems**, v. 25, n. 7, p. 1301–1312, jul. 2006. ISSN 0278-0070. Disponível em: http://ieeexplore.ieee.org/document/1634626/.

KAHNG, A. B. et al. **VLSI Physical Design: From Graph Partitioning to Timing Closure**. Dordrecht: Springer Netherlands, 2011. ISBN 978-90-481-9590-9 978-90-481-9591-6. Disponível em: https://link.springer.com/10.1007/978-90-481-9591-6.

LIN, Y. et al. DREAMPlace: Deep Learning Toolkit-Enabled GPU Acceleration for Modern VLSI Placement. In: **Proceedings of the 56th Annual Design Automation Conference 2019**. Las Vegas NV USA: ACM, 2019. p. 1–6. ISBN 978-1-4503-6725-7. Disponível em: https://dl.acm.org/doi/10.1145/3316781.3317803.

LIU, L. et al. Xplace: an extremely fast and extensible global placement framework. In: **Proceedings of the 59th ACM/IEEE Design Automation Conference**. New York, NY, USA: Association for Computing Machinery, 2022. (DAC '22), p. 1309–1314. ISBN 978-1-4503-9142-9. Disponível em: https://dl.acm.org/doi/10.1145/3489517.3530485.

LU, J. **Analytic VLSI Placement using Electrostatic Analogy**. Tese (Doutorado) — UC San Diego, 2014. Disponível em: https://escholarship.org/uc/item/8255z18z.

LU, J. et al. ePlace: Electrostatics based placement using Nesterov's method. In: **2014 51st ACM/EDAC/IEEE Design Automation Conference (DAC)**. [s.n.], 2014. p. 1–6. ISSN: 0738-100X. Disponível em: https://ieeexplore.ieee.org/document/6881448.

LU, J. et al. ePlace-MS: Electrostatics-Based Placement for Mixed-Size Circuits. **IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems**, v. 34, n. 5, p. 685–698, maio 2015. ISSN 0278-0070, 1937-4151. Disponível em: http://ieeexplore.ieee.org/document/7008518/.

Lu, Jingwei et al. FFTPL: An analytic placement algorithm using fast fourier transform for density equalization. In: **2013 IEEE 10th International Conference on ASIC**. Shenzhen, China: IEEE, 2013. p. 1–4. ISBN 978-1-4673-6417-1 978-1-4673-6415-7 978-1-4673-6416-4. Disponível em: http://ieeexplore.ieee.org/document/6811936/.

MONTEIRO, J. VLSI Placement Optimization Algorithms. **Journal of Integrated Circuits and Systems**, v. 17, n. 3, p. 1–15, dez. 2022. ISSN 1872-0234, 1807-1953. Publisher: Journal of Integrated Circuits and Systems. Disponível em: https://jics.org.br/ojs/index.php/JICS/article/view/645.

NESTEROV, Y. **A method for solving the convex programming problem with convergence rate o(1/k2)**. 1983. Pages: 543 Publication Title: Dokl Akad Nauk SSSR Volume: 269. Disponível em: https://cir.nii.ac.jp/crid/1370862715914709505.

NOCEDAL, J.; WRIGHT, S. J. **Numerical optimization**. New York: Springer, 1999. (Springer series in operations research). ISBN 978-0-387-98793-4.

PENG, K.; ZHU, W. Pplace-MS: Methodologically Faster Poisson's Equation-Based Mixed-Size Global Placement. **IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems**, v. 43, n. 2, p. 613–626, fev. 2024. ISSN 1937-4151. Disponível em: https://ieeexplore.ieee.org/document/10266769.

QIU, Y. et al. Progress of Placement Optimization for Accelerating VLSI Physical Design. **Electronics**, v. 12, n. 2, p. 337, jan. 2023. ISSN 2079-9292. Number: 2 Publisher: Multidisciplinary Digital Publishing Institute. Disponível em: https://www.mdpi.com/2079-9292/12/2/337.

RAY, B. B. et al. An Optimized HPWL Model for VLSI Analytical Placement. In: **2015 International Conference on Information Technology (ICIT)**. [s.n.], 2015. p. 7–12. Disponível em: https://ieeexplore.ieee.org/document/7437582.

RUTENBAR, R. Simulated annealing algorithms: an overview. **IEEE Circuits and Devices Magazine**, v. 5, n. 1, p. 19–26, jan. 1989. ISSN 1558-1888. Disponível em: https://ieeexplore.ieee.org/abstract/document/17235.

SAHNI, S.; BHATT, A. The Complexity of Design Automation Problems. In: **17th Design Automation Conference**. [s.n.], 1980. p. 402–411. Disponível em: https://ieeexplore.ieee.org/document/1585278.

WALKINGTON, N. J. Nesterov's Method for Convex Optimization. **SIAM Review**, v. 65, n. 2, p. 539–562, maio 2023. ISSN 0036-1445, 1095-7200. Disponível em: https://epubs.siam.org/doi/10.1137/21M1390037.

WESTE, N. H. E.; HARRIS, D. M. **CMOS VLSI design: a circuits and systems perspective**. 4. ed. ed. Boston, Mass.: Addison-Wesley, 2011. ISBN 978-0-321-54774-3.

ZHANG, H.; SAPATNEKAR, S. S. Accelerating Electrostatics-based Global Placement with Enhanced FFT Computation. In: **2026 31st Asia and South Pacific Design Automation Conference (ASP-DAC)**. [s.n.], 2026. p. 779–785. ISSN 2153-697X. ISSN: 2153-697X. Disponível em: https://ieeexplore.ieee.org/abstract/document/11420709.

Apêndices

APÊNDICE A – CÓDIGO FONTE

O código desenvolvido está disponível nos seguintes repositórios:

- Implementação do ePlace (localizada principalmente dentro da pasta src/epl).
  - https://github.com/eclufsc/OpenROAD_ecl/tree/ePlace
- Fluxo de síntese utilizando a ferramenta desenvolvida.
  - https://github.com/eclufsc/OpenROAD-flow-scripts/tree/ePlace

APÊNDICE B – ARTIGO

Implementação, Validação e Análise do ePlace, Técnica Clássica de Posicionamento Global de Circuitos VLSI Baseada em Sistemas Eletrostáticos

Lucas Yuki Imamura [1], **José Luís Almada Güntzel** [1]

[1]Departamento de Informática e Estatística – Universidade Federal de Santa Catarina (UFSC)
Florianópolis – SC – Brasil

Abstract. *The EDA flow comprises several stages, among which global placement stands out, being responsible for defining the initial physical position of each cell in the circuit. This work presents an experimental investigation of the classic global placement algorithm, ePlace, through a custom implementation developed within the OpenROAD environment. The experiments, conducted with circuits from the OpenROAD Flow Scripts, demonstrate that the implemented technique was, on average, 34% faster than the reference flow, although it generated a 20.2% larger HPWL. At the end of the synthesis flow, 5 out of 23 test cases failed due to routing congestion, and the maximum achieved frequency was 2.7% lower than the reference. In contrast, the default OpenROAD placer obtained an HPWL 15.0% larger than the reference, requiring a 150.9% higher execution time during the global placement stage. At the end of the flow, however, the standard tool achieved a 6.7% higher frequency and zero failures, highlighting the impact of the integrated congestion mitigation and timing correction techniques.*

Resumo. *O fluxo de EDA compreende diversas etapas, dentre as quais se destaca o posicionamento global, responsável por definir a posição física inicial de cada célula do circuito. Este trabalho apresenta uma investigação experimental do algoritmo clássico de posicionamento global ePlace, por meio de uma implementação própria desenvolvida no ambiente OpenROAD. Os experimentos, realizados com circuitos do OpenROAD Flow Scripts, demonstram que a técnica implementada foi, em média, 34% mais rápida que o fluxo de referência, embora tenha gerado um HPWL 20,2% maior. Ao final do fluxo de síntese, 5 dos 23 casos de teste falharam devido ao congestionamento no roteamento, e a frequência máxima alcançada foi 2,7% menor que a referência. Em contrapartida, o posicionador padrão do OpenROAD obteve um HPWL 15,0% maior que a referência, exigindo um tempo de execução 150,9% superior na etapa de posicionamento global. Ao término do fluxo, contudo, a ferramenta padrão alcançou uma frequência 6,7% maior e nenhuma falha, evidenciando o impacto das técnicas integradas de mitigação de congestionamento e correção de timing.*

1. Introdução

Atualmente, a indústria de semicondutores tem enfrentado desafios devido ao aumento na dimensão dos projetos e na complexidade dos objetivos e restrições. Estes fatores

tornam a automação da síntese física, em fluxos VLSI, um processo cada vez mais difícil e demorado [Qiu et al. 2023].

Dentre as etapas do fluxo de design, o posicionamento global assume um papel central pois impacta diretamente a roteabilidade e a performance do circuito. Isso ocorre porque, nas etapas seguintes, as células sofrem apenas pequenas mudanças na posição [Chu 2009].

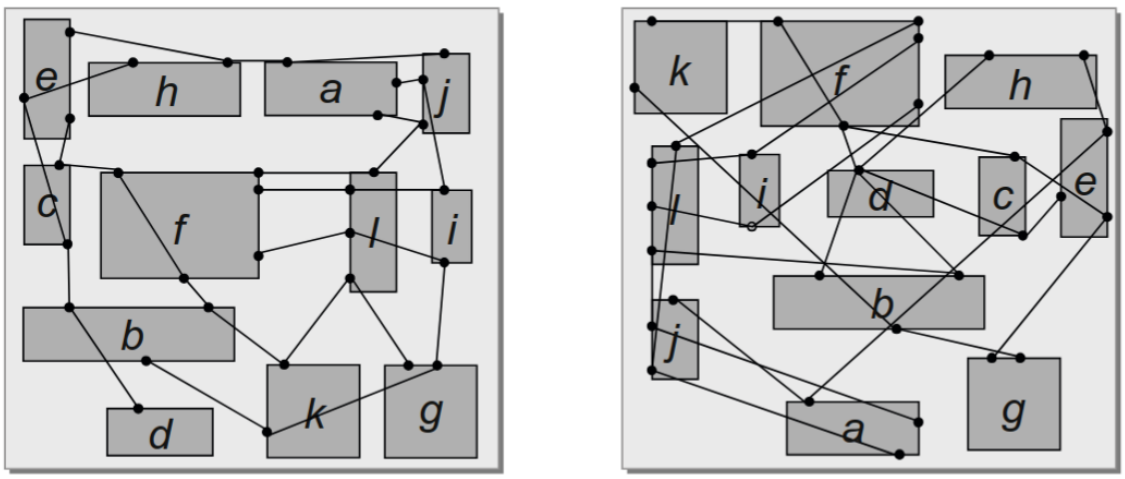
Considerando que o posicionamento global é um problema *NP-completo* [Sahni and Bhatt 1980], as técnicas atuais baseiam-se em heurísticas para convergir a uma solução eficiente. De acordo com [Zhang and Sapatnekar 2026], a maioria das abordagens contemporâneas deriva do ePlace, que utiliza uma analogia com sistemas eletrostáticos para diminuir a sobreposição das células durante o posicionamento. Dentre os algoritmos que seguem esse paradigma, se destacam o RePlAce [Cheng et al. 2019], o DREAMPlace [Lin et al. 2019], Xplace [Liu et al. 2022] e Pplacems [Peng and Zhu 2024].

Este trabalho visa investigar experimentalmente o funcionamento do algoritmo clássico de posicionamento global baseado em eletrostática, o ePlace, através de uma implementação própria no contexto do projeto OpenROAD.

2. Posicionamento Global

O posicionamento é uma das etapas da síntese de design físico no fluxo de design de circuitos Integração em Escala Muito Grande (VLSI), na qual é definida a localização física de cada célula do circuito de forma que seja possível sua fabricação. Antes do posicionamento, são feitas as etapas de particionamento e *floorplanning*, nas quais o circuito é dividido em módulos menores, são definidas as regiões de cada bloco e a localização dos pinos. A partir disso, o posicionamento das *standard cells* e dos elementos lógicos é feito com o objetivo de minimizar o comprimento das conexões do circuito. A figura 1 ilustra dois arranjos para o mesmo circuito: o da esquerda demonstra um posicionamento com boa otimização do comprimento das redes, enquanto o da direita ilustra uma solução de qualidade inferior [Kahng et al. 2011].

Figura 1. Exemplos de posicionamentos



Matematicamente, o posicionamento pode ser definido como um hipergrafo $G = (V, E, R)$ onde V é um conjunto de vértices (instâncias), E é um conjunto de arestas (redes) e R é a região de posicionamento. V_f e V_m representam os conjuntos de nodos fixos (células fixas e macros) e móveis (células móveis) respectivamente, sendo $|V_m| = m$ o número de nodos móveis. Um posicionador define os locais das células do circuito $v = (x, y)$, onde $x = (x_1, x_2, ..., x_m)$ e $y = (y_1, y_2, ..., y_m)$.

3. ePlace

Atualmente, grande parte das técnicas de posicionamento global avançadas são fundamentadas na analogia eletrostática [Zhang and Sapatnekar 2026]. O primeiro posicionador global baseado em eletrostática foi o FFTPL [Lu, Jingwei et al. 2013], que em seguida foi aprimorado no ePlace [Lu et al. 2014] e no ePlace-ms [Lu et al. 2015].

Estes algoritmos fazem parte da classe de posicionadores analíticos não lineares que utilizam um modelo eletrostático para manter a densidade de células menor que o valor desejado. Nesta abordagem, as células e macros do circuito são modeladas como partículas carregadas eletricamente com cargas positivas, gerando forças de repulsão eletrostática mútua, fazendo com que as células sejam uniformemente distribuídas pela região. A partir dessa analogia, é definida uma função de custo diferenciável que permite a aplicação de algoritmos de otimização não linear. Desta forma, o posicionador consegue otimizar simultaneamente o comprimento das redes e diminuir a sobreposição entre instâncias.

3.1. Função custo

A equação (1) define o objetivo da otimização realizada durante o posicionamento. A função custo $f(v)$ é composta de dois termos: comprimento das redes $W(v)$ e a penalidade da densidade $N(v)$. O parâmetro λ é utilizado para balancear ambos os custos durante o posicionamento, sendo atualizado iterativamente durante o processo de otimização [Lu, Jingwei et al. 2013].

$$\min_v f(v) = W(v) + \lambda N(v) \tag{1}$$

Para otimizar o custo, o ePlace utiliza o gradiente da função $\nabla f(v)$, equação (2), para convergir ao mínimo através do algoritmo Gradiente Acelerado de Nesterov (NAG) [Lu et al. 2014].

$$\nabla f(v) = \nabla W(v) + \lambda \nabla N(v) \tag{2}$$

3.1.1. Custo do comprimento das redes

Posicionadores analíticos dirigidos por comprimento de fio buscam otimizar o tamanho das conexões com uma aproximação do *Half-Perimeter WireLength* (HPWL). O HPWL é definido como o semiperímetro do retângulo que engloba todos os pinos da rede [Kahng et al. 2011]. Apesar do HPWL ser uma métrica comumente utilizada para estimar comprimento de fio, ele não é uma função diferenciável, o que torna sua utilização

em um posicionador analítico não linear inviável. Por este motivo, é necessário utilizar aproximações suaves e deriváveis, como o HPWL de média ponderada [Ray et al. 2015].

O modelo de média ponderada, proposto por [Hsu et al. 2013], aproxima as funções máximo ($\max$) e, consequentemente, mínimo ($\min$), utilizadas para computar o HPWL, através de uma média ponderada, conforme equação (3). Nesta aproximação, os pesos ($e^{\frac{val}{\gamma}}$) crescem exponencialmente com a magnitude dos valores, logo os valores mais altos influenciam mais o resultado, aproximando o comportamento da função $\max$. Analogamente, a aproximação da função $\min$ é obtida invertendo o sinal dos pesos (multiplicando por -1).

$$max(vals) \approx WAW_{max}(vals) = \frac{\sum_{val \in vals} val \cdot e^{\frac{val}{\gamma}}}{\sum_{val \in vals} e^{\frac{val}{\gamma}}} \tag{3}$$

$$HPWL_x(e) \approx W_x(e) = \left(\frac{\sum_{i \in e} x_i \cdot e^{\frac{x_i}{\gamma}}}{\sum_{i \in e} e^{\frac{x_i}{\gamma}}} - \frac{\sum_{i \in e} x_i \cdot e^{\frac{-x_i}{\gamma}}}{\sum_{i \in e} e^{\frac{-x_i}{\gamma}}} \right) \tag{4}$$

$$HPWL(v) \approx W(v) = \sum_{e \in E} (W_x(e) + W_y(e)) \tag{5}$$

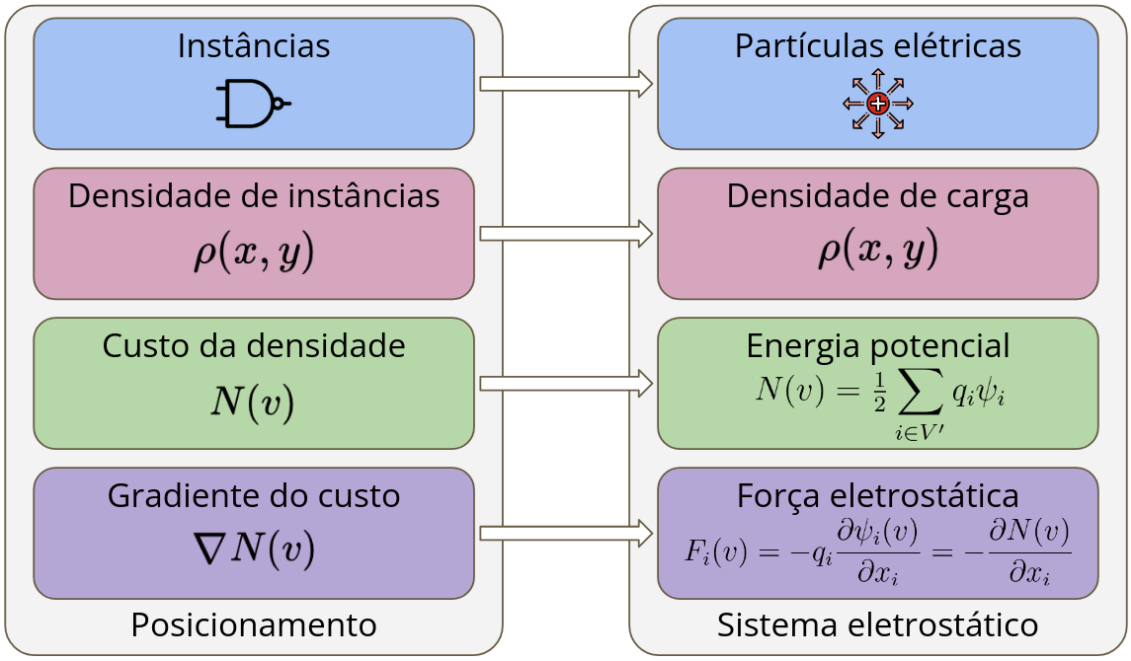
As equações (4) e (5) definem o modelo de média ponderada para a coordenada x, sendo realizado o mesmo processo para a coordenada y. O parâmetro γ é um coeficiente utilizado para ajustar o quão precisa é a aproximação, com $\gamma \to 0$, a função converge para o valor do HPWL [Ray et al. 2015]. Durante a execução do ePlace, o valor de γ é atualizado entre iterações para alcançar melhores resultados. Com a aproximação das funções $\min$ e $\max$ através da média ponderada, é possível derivar analiticamente o custo do comprimento do fio em relação à posição de cada pino, possibilitando a sua otimização pelo algoritmo NAG.

3.1.2. Custo da densidade (eDensity)

[Lu et al. 2014] propuseram o eDensity, um modelo de custo de densidade baseado em um sistema eletrostático, conforme ilustrado na figura 2. Nesse modelo, as instâncias i são tratadas como partículas elétricas positivamente carregadas, cuja carga q_i é igual à sua área. Desse modo, células maiores exercem uma força de repulsão maior sobre as demais, abrindo maior espaço para seu posicionamento. Consequentemente, a densidade das células passa a ser igual à densidade das cargas $\rho(x, y)$. O custo da densidade é definido como a energia potencial do sistema $N(v)$, de forma que as forças eletrostáticas correspondam ao gradiente negativo do custo. Isto ocorre porque as forças guiam o sistema para o equilíbrio, que é o estado de menor energia potencial. Assim, ao minimizar a energia do sistema, as forças eletrostáticas também reduzem o custo da densidade do circuito.

$$N(v) = \frac{1}{2} \sum_{i \in V'} N_i = \frac{1}{2} \sum_{i \in V'} q_i \psi_i(v) \tag{6}$$

Figura 2. Modelagem do *eDensity*



A energia potencial elétrica do sistema, custo da densidade, pode ser calculada através da equação (6), onde N_i é a energia potencial, q_i é a carga e $\psi_i(v)$ é o potencial elétrico da partícula i. Para otimizar a função de custo, é necessário calcular o gradiente de $N(v)$, equação (7). Como as cargas nas partículas são constantes, só é preciso calcular o gradiente do potencial elétrico.

$$\frac{\partial N(v)}{\partial x_i} = q_i \frac{\partial \psi_i(v)}{\partial x_i} \tag{7}$$

Considerando que o campo elétrico $\xi_i(v)$ é definido como o gradiente negativo do potencial elétrico, $\xi_i(v) = -\frac{\partial \psi_i(v)}{\partial x_i}$, a força eletrostática descrita pela lei de Coulomb (equação 8) tem a direção oposta ao gradiente do potencial elétrico. Consequentemente, a força eletrostática move as cargas em direção à redução da energia potencial do sistema.

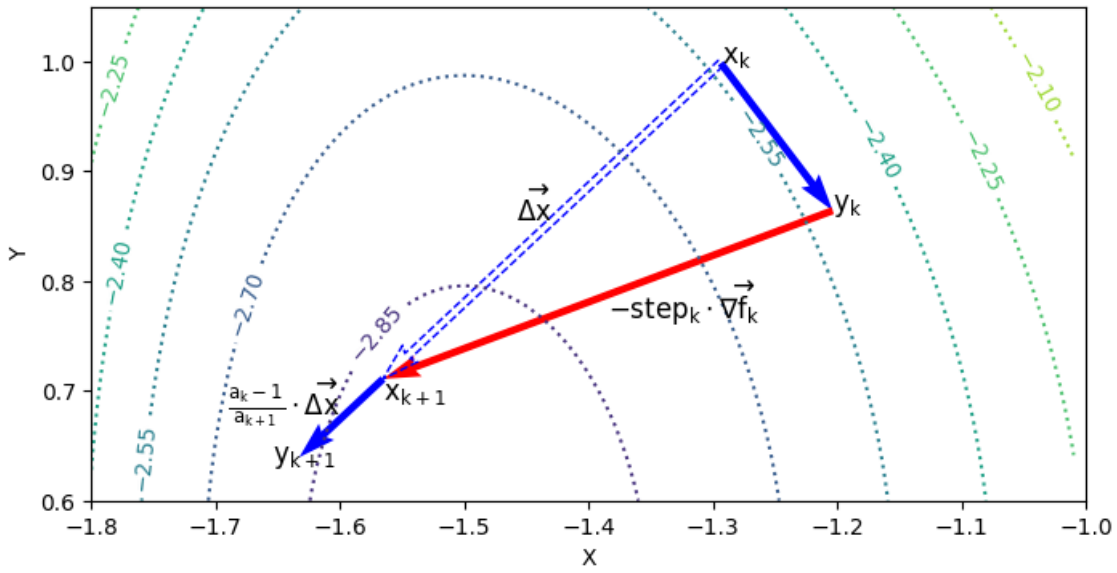
$$F_i(v) = q_i \xi_i(v) = -q_i \frac{\partial \psi_i(v)}{\partial x_i} = -\frac{\partial N(v)}{\partial x_i} \tag{8}$$

Para calcular o gradiente, é utilizada a equação de Poisson, relacionando a densidade de cargas ao potencial ao campo elétrico. Esta equação é resolvida numericamente via métodos espectrais, que possuem grande precisão e baixo custo computacional. Através desta técnica é possível calcular o gradiente do custo de densidade a partir da distribuição das instâncias [Lu et al. 2014].

3.2. Otimização não linear

Para otimizar a função de custo do posicionamento, [Lu 2014] propôs a utilização do algoritmo NAG, empregando uma heurística baseada na previsão da constante de Lipschitz

Figura 3. Ilustração de um passo do NAG



para estimar o tamanho do passo de cada iteração.

O NAG foi escolhido por precisar apenas do gradiente de primeira ordem e memória linear em relação ao tamanho do problema. Enquanto métodos clássicos, como gradiente descendente, possuem taxa de convergência de $O(1/k)$ (onde k representa o número de iterações), o NAG foi o primeiro a alcançar $O(1/k^2)$, limite teórico provado para métodos de primeira ordem em funções convexas com gradiente Lipschitz contínuo [Walkington 2023].

Dado um espaço de Hilbert E, o NAG soluciona o problema de programação convexa construindo uma sequência minimizante de pontos $\{x_k\}_0^\infty$ que não é relaxacional. Esta propriedade permite reduzir o esforço computacional por iteração ao mínimo [Nesterov 1983]. O algoritmo 1 descreve uma iteração do método para um problema $min\{f(x)|x \in E\}$ com um conjunto não vazio de minimos X^*. Aqui x é a solução do problema de otimização convexa, y é uma solução de referência utilizada para calcular o gradiente, a é um parâmetro de aceleração e $step$ é o tamanho do passo. No início ($k = 0$) o método começa com uma solução $y_0 \in E$, $x_0 = v_0$ e $a_0 = 1$, sendo estes os parâmetros atualizados em cada iteração.

Algorithm 1 Otimizador Nesterov durante o passo k

Entradas: Tamanho do passo $step_k$, parâmetro a_k, solução x_k, referência y_k
Saídas: Parâmetro a_{k+1}, solução x_{k+1}, referência y_{k+1}

1: $a_{k+1} \leftarrow \left(1 + \sqrt{4a_k^2 + 1}\right)/2$
2: $\nabla f_k \leftarrow \nabla f(y_k)$
3: $x_{k+1} \leftarrow y_k - step_k \cdot \nabla f_k$
4: $y_{k+1} \leftarrow x_{k+1} + (a_k - 1)(x_{k+1} - x_k)/a_{k+1}$

A figura 3 ilustra o comportamento de um passo do algoritmo 1. Em cada iteração,

o algoritmo recebe como entrada o tamanho do passo $step_k$, o parâmetro a_k, a solução x_k e a referência y_k. Na linha 3, o NAG atualiza a posição x_{k+1} utilizando a referência y_k e o gradiente ∇f_k. Então, a nova referência y_{k+1} é calculada por meio da extrapolação da direção $\overrightarrow{\Delta x}$ percorrida pela solução x. Diferentemente do algoritmo de gradiente descendente convencional, o NAG projeta a solução x em uma referência y para antecipar a tendência da função, permitindo que avance em direção ao mínimo global com mais eficiência.

4. Metodologia

A fim de se realizar a análise experimental prevista no objetivo principal deste Trabalho de Conclusão de Curso, o algoritmo de posicionamento global ePlace foi implementado dentro do projeto OpenROAD, utilizando a linguagem C++ sob o paradigma de orientação a objetos. A interface e integração com o fluxo de controle foram desenvolvidas utilizando *SWIG* e *Tcl*, permitindo a comunicação entre o algoritmo desenvolvido e a infraestrutura do OpenROAD Flow Scripts.

Os experimentos foram realizados em uma estação de trabalho com o sistema Linux 24.04.4 LTS, equipada com um processador AMD® Ryzen® 9 9950X de 4.3 GHz (16 núcleos e 32 threads) e 32 GB de memória RAM DDR5 (4800 MHz).

Os testes foram realizados utilizando os circuitos disponíveis no repositório OpenROAD Flow Scripts. Para a escolha dos *benchmarks*, foram selecionados os circuitos presentes na maioria das seis bibliotecas de *standard cells* disponíveis no repositório. A quantidade de instâncias para cada caso de teste é apresentada na Tabela 1. Observa-se que o circuito *riscv32i* não possui uma configuração nativa para a biblioteca *Nangate45* no repositório OpenROAD Flow Scripts, e por este motivo, tal combinação não foi incluída nos experimentos.

Tabela 1. Quantidade de instâncias por circuito

	asap7	**gf180**	**ihp-sg13g2**	**nangate45**	**sky130hd**	**sky130hs**
aes	15389	16489	13296	12697	11036	10962
ibex	19180	12091	16229	16448	14292	12589
jpeg	55618	39078	69595	61423	36990	34333
riscv32i	9112	6158	8084		4621	4599

O posicionamento realizado pelo algoritmo implementado (ePlace) foi comparado ao posicionador global padrão do projeto OpenROAD, baseado no RePlAce de [Cheng et al. 2019]. Embora o RePlAce também utilize o modelo eletrostático para o posicionamento global, ele também utiliza métricas de congestionamento para otimizar a roteabilidade do circuito. Adicionalmente, a implementação do RePlAce no OpenROAD incorpora técnicas para reduzir violações de *timing* durante o posicionamento, fazendo a inserção de *buffers* e utilizando pesos para os caminhos críticos. Visto que o algoritmo implementado não incorpora técnicas para medir diretamente congestionamento ou *timing*, a comparação foi realizada utilizando duas configurações: uma desabilitando tais recursos (RePlAce-ref), para permitir uma análise direta do desempenho do modelo eletrostático, e outra com os recursos habilitados (RePlAce), para avaliar o impacto das otimizações adicionais presentes no fluxo padrão.

Tabela 2. Posicionadores globais utilizados na avaliação experimental

Nome	Implementação	Guiado por congestionamento	Guiado por *timing*
RePlAce-ref	OpenROAD	-	-
ePlace	Este trabalho	-	-
RePlAce	OpenROAD	✓	✓

5. Resultados

As Tabelas 3, 4 e 5 apresentam os resultados obtidos nos experimentos, medidos após diferentes etapas do fluxo de síntese. Cada linha detalha a média das métricas para uma biblioteca de *standard cells*, incluindo ao final a média global do *benchmark*. Os valores apresentados são as diferenças percentuais comparadas aos resultados de referência (*baseline*), obtidos com o fluxo RePlAce-ref (posicionador global do OpenROAD com as técnicas de otimização de congestionamento e *timing* desabilitadas). Os valores em negrito destacam o melhor resultado obtido entre os três posicionadores avaliados.

Tabela 3. Resultados após o posicionamento global

	HPWL		Iterações do NAG		Tempo de execução	
Biblioteca	ePlace	RePlAce	ePlace	RePlAce	ePlace	RePlAce
asap7	31,7%	**-2,5%**	**-22,2%**	-1,3%	**-30,2%**	35,3%
gf180	15,0%	10,1%	**-48,8%**	47,0%	**-60,3%**	75,7%
ihp-sg13g2	35,3%	27,7%	**-32,3%**	0,9%	**-12,7%**	78,3%
nangate45	6,0%	**-8,2%**	**-39,1%**	72,9%	**-12,3%**	192,9%
sky130hd	18,7%	29,2%	**-20,9%**	254,4%	**-45,5%**	166,9%
sky130hs	11,2%	28,2%	**-29,2%**	402,4%	**-37,6%**	367,2%
Total	20,2%	15,0%	**-31,8%**	131,9%	**-34,0%**	150,9%

A tabela 3 detalha as métricas obtidas na etapa de posicionamento global, abrangendo o comprimento aproximado das redes (HPWL), o número de iterações do NAG e o tempo de execução. É notável que, embora a técnica implementada apresente um HPWL médio $20,2\%$ pior que a referência, o tempo de execução foi reduzido em $34,0\%$. Já o RePlAce (fluxo padrão do OpenROAD) exibiu resultados piores comparados com a referência, tanto em HPWL quanto em tempo de execução. Os tempos de execução mais longos do RePlAce devem-se à aplicação das técnicas de correção de *timing* e de mitigação do congestionamento das redes. Esta última, abre espaço em regiões de alto congestionamento para facilitar o roteamento, consequentemente, aumenta a distância entre as instâncias e comprimento das redes que as conectam.

Tabela 4. Resultados do posicionamento detalhado

Biblioteca	HPWL ePlace	HPWL RePlAce	Estimativa de frequência máxima ePlace	Estimativa de frequência máxima RePlAce	Deslocamento total ePlace	Deslocamento total RePlAce
asap7	20,2%	**-7,5%**	1,4%	**48,4%**	**-25,9%**	-0,8%
gf180	6,8%	5,3%	1,7%	**43,6%**	**-34,0%**	12,7%
ihp-sg13g2	15,5%	**-10,2%**	0,9%	**67,3%**	6,1%	**-8,0%**
nangate45	1,5%	**-15,0%**	-0,5%	**47,9%**	**-38,8%**	6,0%
sky130hd	11,5%	18,2%	2,2%	**39,5%**	**-23,2%**	3,2%
sky130hs	8,0%	15,3%	0,9%	**31,8%**	**-31,1%**	-1,3%
Total	11,0%	1,7%	1,2%	**46,4%**	**-23,9%**	1,8%

Na tabela 4 são apresentados os resultados obtidos após a etapa de posicionamento detalhado, utilizando como métricas de avaliação o HPWL, a frequência máxima estimada e o deslocamento total das instâncias entre o resultado do posicionamento global e a posição legalizada. Após a legalização, a diferença média de HPWL da técnica implementada decresce para $11,0\%$. Além disso, apesar do maior comprimento de fio, a estimativa de frequência máxima é $1,2\%$ superior à da referência. A redução dessa diferença pode ser explicada pelo deslocamento total durante o posicionamento detalhado: no fluxo utilizando o ePlace, o deslocamento foi, em média $23,9\%$ menor, indicando que o resultado do posicionamento global apresentava menor sobreposição de instâncias. Essa discrepância provavelmente é devida ao cálculo da dimensão da grade de densidade. O posicionador global do OpenROAD utiliza uma abordagem distinta da proposta original do ePlace, resultando em grades menores que causa um erro de discretização maior durante o cálculo do *overflow*, o que gera um posicionamento com maior sobreposição de instâncias. Adicionalmente, é possível notar o efeito da técnica de correção de *timing* utilizada pelo RePlAce do OpenROAD durante o posicionamento global: a estimativa de frequência máxima dos circuitos é, em média $46,4\%$ maior que o mesmo algoritmo sem tal otimização.

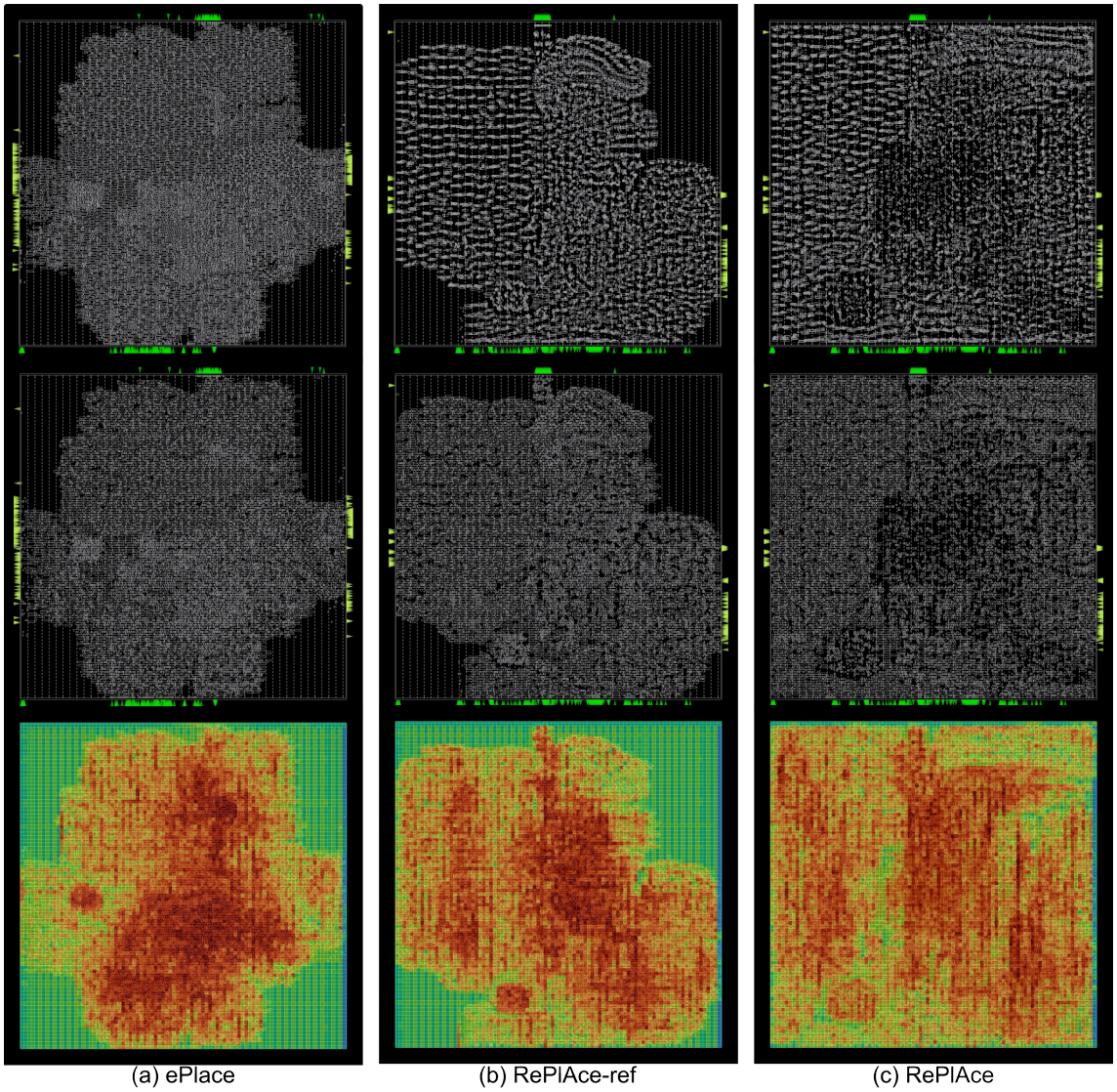
Tabela 5. Resultados ao final do fluxo de síntese física

Biblioteca	Falhas no fluxo RePlAce-ref	Falhas no fluxo ePlace	Falhas no fluxo RePlAce	Potência ePlace	Potência RePlAce	Frequência máxima ePlace	Frequência máxima RePlAce	Área de instâncias ePlace	Área de instâncias RePlAce
asap7	-	-	-	5,5%	**-4,1%**	-3,1%	**5,0%**	-0,1%	**-1,6%**
gf180	-	-	-	2,3%	2,2%	-0,3%	**7,2%**	0,4%	3,7%
ihp-sg13g2	-	-	-	7,2%	**-2,2%**	-4,6%	**7,1%**	0,2%	**-1,5%**
nangate45	-	1*	-	2,4%	**-2,0%**	-1,9%	**10,7%**	0,8%	0,9%
sky130hd	2	3	-	3,5%	8,7%	-1,5%	**7,0%**	1,0%	3,7%
sky130hs	-	2	-	4,2%	1,8%	-3,9%	**4,5%**	1,5%	2,7%
Total	2	6	-	4,5%	0,1%	-2,7%	**6,7%**	0,4%	1,1%

A tabela 5 detalha os resultados ao final do fluxo de síntese, apresentando as três principais métricas de qualidade (*Quality of Results* - QoR) de um circuito: a potência,

a performance (frequência máxima) e a área (área de instâncias). Os resultados marcados como NF (Não finalizou) se referem aos casos que não finalizaram o fluxo devido a problemas de congestionamento durante as etapas de roteamento global ou detalhado. Excepcionalmente, o teste do circuito *jpeg* com a biblioteca *Nangate45* utilizando o fluxo ePlace (marcado como NF^*) falhou devido a um erro interno na ferramenta de roteamento global do OpenROAD. O fluxo de referência (RePlAce-ref) apresentou apenas 2 falhas durante o roteamento, ambas utilizando a biblioteca *sky130hd* (versão de alta densidade da tecnologia *SKY130*), enquanto o fluxo com o algoritmo implementado falhou em 5 casos devido à congestionamento (3 na biblioteca *sky130hd* e 2 na *sky130hs*).

Figura 4. Posicionamento global, posicionamento final e congestionamento final (*sky130hs* - ibex)



(a) ePlace (b) RePlAce-ref (c) RePlAce

Apesar dos casos de falha, os resultados finais do fluxo utilizando o ePlace foram próximos à referência, em média $4,5\%$ maior potência, frequência $2,7\%$ menor e $0,4\%$ aumento da área de instâncias. Estas diferenças provavelmente são impacto direto da maior área de instâncias e maior HPWL durante o posicionamento, apesar dos resultados

de estimativa de frequência serem melhores logo após o posicionamento detalhado, o roteamento foi bastante impactado pelo congestionamento do circuito. O fluxo padrão do OpenROAD apesentou melhoria significativa na frequência máxima, em média $6,7\%$ maior, sem aumentar significativamente a potência consumida, em média apenas $0,1\%$ maior.

A figura 4 apresenta o posicionamento global, o posicionamento final e o mapa de calor do congestionamento para o circuito *ibex*, utilizando a biblioteca *sky130hs* para os três fluxos testados. Na primeira linha dessa imagem é possível notar a diferença no espalhamento das instâncias entre as soluções: o ePlace reduz significativamente a sobreposição das instâncias, enquanto as soluções do RePlAce-ref e RePlAce possuem um grande número de instâncias sobrepostas em pequenos agrupamentos densos. Essa característica também pode ser observada ao comparar o posicionamento global (primeira linha) com o final (segunda linha): o posicionamento global gerado pelo ePlace sofre menos modificações do que os posicionamentos gerados pelo RePlAce-ref e RePlAce. Por fim, na terceira linha é possível notar o efeito da técnica de mitigação de congestionamento aplicada pelo RePlAce: as regiões que teriam alto congestionamento (indicadas em vermelho) têm suas instâncias artificialmente infladas, gerando maior distanciamento entre as células dessas regiões e, desta forma, reduzindo a densidade de pinos e o congestionamento. Essa técnica também é responsável pelo melhor aproveitamento da região de posicionamento do RePlAce. Quando ocorre a inflação das células, também é realizada a retirada de células *fillers* para compensar o aumento artificial das áreas das instâncias. Com isso, as instâncias do circuito continuam sob a mesma restrição de densidade máxima, mas as áreas de baixa densidade são utilizadas para diminuir o congestionamento das redes.

6. Conclusão

Este trabalho avaliou o algoritmo clássico de posicionamento global baseado em eletrostática, o ePlace, integrado a um fluxo de síntese física completo. A implementação foi realizada utilizando a infraestrutura do projeto *open-source* OpenROAD, permitindo uma comparação direta entre a implementação do ePlace e o algoritmo de posicionamento global baseado no RePlAce, nativo da ferramenta.

Os experimentos utilizaram como referência o RePlAce do OpenROAD sem as técnicas de correção de *timing* e de mitigação de congestionamento das redes (batizado de RePlAce-ref), de modo a aproximar o comportamento do ePlace original. Essa referência foi comparada tanto com a implementação realizada neste trabalho quanto com o posicionador padrão do OpenROAD com todas as técnicas de otimização habilitadas.

Os resultados demonstram que a técnica implementada apresentou um HPWL após o posicionamento global $20,2\%$ pior que a referência, porém com tempo de execução $34,0\%$ menor, enquanto o fluxo padrão do OpenROAD teve um resultado $15,0\%$ pior que a referência e foi em média $150,9\%$ mais lento devido às técnicas adicionais. Após o posicionamento detalhado, a diferença de HPWL reduziu para $11,0\%$ para o ePlace e $1,7\%$ para o RePlace. A técnica implementada exigiu em média $23,9\%$ menos deslocamento das instâncias durante a legalização do posicionamento. Esse dado indica que a solução gerada pelo posicionador global implementado promove uma dispersão de células mais eficiente que a solução da ferramenta de posicionamento do OpenROAD. Adicio-

nalmente, após o posicionamento detalhado, a estimativa de frequência máxima superou a referência em $1,2\%$, apesar do HPWL pior. Em comparação, o RePlAce alcançou uma frequência $46,4\%$ maior, o que demonstra a eficácia da técnica de correção de *timing*.

Ao final do fluxo de síntese, 6 dos 23 casos de teste falharam com a implementação realizada. Dentre essas falhas, 5 ocorreram devido ao alto congestionamento durante as etapas de roteamento e uma por erro interno na ferramenta de roteamento global. Em contrapartida, a referência falhou em apenas 2 casos (devido ao congestionamento), e o RePlAce concluiu todos os testes com sucesso. Para os circuitos que finalizaram o fluxo, a potência consumida foi, em média, $4,5\%$ maior e a frequência máxima foi $2,7\%$ inferior à referência, enquanto o RePlAce alcançou potência $4,5\%$ pior e frequência $6,7\%$ melhor.

As configurações dos circuitos de teste adotadas foram as presentes no OpenROAD Flow Scripts, sendo estes os mesmos parâmetros utilizados no desenvolvimento das ferramentas originais. Tais configurações são normalmente ajustadas para elevar o nível de exigência dos experimentos, permitindo observar variações nos resultados mesmo com mudanças pequenas no software. Visto que o ePlace e o RePlAce-ref não possuem mecanismos de otimização de *timing* e de congestionamento, era esperado que alguns casos de teste falhassem.

Em futuras investigações, seria possível relaxar as restrições dos casos de teste falhos, como a frequência alvo, a densidade alvo, a área do chip e o uso máximo das camadas de roteamento, permitindo chegar ao final do fluxo de síntese em todos os testes realizados. Além disso, este trabalho não explorou diferentes parâmetros de configuração do algoritmo implementado. Por exemplo, foi observado que o número de iterações do NAG no posicionador implementado foi cerca de $31,8\%$ menor que a referência. Ao alterar os parâmetros de $HWPL_{ref}$ ou o multiplicador μ_0, seria possível desacelerar a convergência, potencialmente alcançando resultados superiores.

Referências

Cheng, C.-K., Kahng, A. B., Kang, I., and Wang, L. (2019). RePlAce: Advancing Solution Quality and Routability Validation in Global Placement. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 38(9):1717–1730.

Chu, C. (2009). CHAPTER 11 - Placement. In Wang, L.-T., Chang, Y.-W., and Cheng, K.-T. T., editors, *Electronic Design Automation*, pages 635–685. Morgan Kaufmann, Boston.

Hsu, M.-K., Balabanov, V., and Chang, Y.-W. (2013). TSV-Aware Analytical Placement for 3-D IC Designs Based on a Novel Weighted-Average Wirelength Model. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 32(4):497–509.

Kahng, A. B., Lienig, J., Markov, I. L., and Hu, J. (2011). *VLSI Physical Design: From Graph Partitioning to Timing Closure*. Springer Netherlands, Dordrecht.

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. In *Proceedings of the 56th Annual Design Automation Conference 2019*, pages 1–6, Las Vegas NV USA. ACM.

Liu, L., Fu, B., Wong, M. D. F., and Young, E. F. Y. (2022). Xplace: an extremely fast and extensible global placement framework. In *Proceedings of the 59th ACM/IEEE Design Automation Conference*, DAC '22, pages 1309–1314, New York, NY, USA. Association for Computing Machinery.

Lu, J. (2014). *Analytic VLSI Placement using Electrostatic Analogy*. PhD thesis, UC San Diego.

Lu, J., Chen, P., Chang, C.-C., Sha, L., Huang, D. J.-H., Teng, C.-C., and Cheng, C.-K. (2014). ePlace: Electrostatics based placement using Nesterov's method. In *2014 51st ACM/EDAC/IEEE Design Automation Conference (DAC)*, pages 1–6. ISSN: 0738-100X.

Lu, J., Zhuang, H., Chen, P., Chang, H., Chang, C.-C., Wong, Y.-C., Sha, L., Huang, D., Luo, Y., Teng, C.-C., and Cheng, C.-K. (2015). ePlace-MS: Electrostatics-Based Placement for Mixed-Size Circuits. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 34(5):685–698.

Lu, Jingwei, Pengwen Chen, Chin-Chih Chang, Sha, L., Dennis Jen-Hsin Huang, Chin-Chi Teng, and Chung-Kuan Cheng (2013). FFTPL: An analytic placement algorithm using fast fourier transform for density equalization. In *2013 IEEE 10th International Conference on ASIC*, pages 1–4, Shenzhen, China. IEEE.

Nesterov, Y. (1983). A method for solving the convex programming problem with convergence rate o(1/k2). Pages: 543 Publication Title: Dokl Akad Nauk SSSR Volume: 269.

Peng, K. and Zhu, W. (2024). Pplace-MS: Methodologically Faster Poisson's Equation-Based Mixed-Size Global Placement. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 43(2):613–626.

Qiu, Y., Xing, Y., Zheng, X., Gao, P., Cai, S., and Xiong, X. (2023). Progress of Placement Optimization for Accelerating VLSI Physical Design. *Electronics*, 12(2):337. Number: 2 Publisher: Multidisciplinary Digital Publishing Institute.

Ray, B. B., Das, S., Hazra, K., Patra, N., and Mohanty, S. K. (2015). An Optimized HPWL Model for VLSI Analytical Placement. In *2015 International Conference on Information Technology (ICIT)*, pages 7–12.

Sahni, S. and Bhatt, A. (1980). The Complexity of Design Automation Problems. In *17th Design Automation Conference*, pages 402–411.

Walkington, N. J. (2023). Nesterov's Method for Convex Optimization. *SIAM Review*, 65(2):539–562.

Zhang, H. and Sapatnekar, S. S. (2026). Accelerating Electrostatics-based Global Placement with Enhanced FFT Computation. In *2026 31st Asia and South Pacific Design Automation Conference (ASP-DAC)*, pages 779–785. ISSN: 2153-697X.

Anexos

ANEXO A – DECLARAÇÃO PADRÃO PARA EMPRESA OU LABORATÓRIO

DECLARAÇÃO DE CONCORDÂNCIA COM AS CONDIÇÕES PARA O DESENVOLVIMENTO DO TCC NA INSTITUIÇÃO

Declaro estar ciente das premissas para a realização de Trabalhos de Conclusão de Curso (TCC) de Ciência da Computação e Sistema de Informações da UFSC, particularmente da necessidade de que se o TCC envolver o desenvolvimento de um software ou produto específico (ex: um protocolo, um método computacional, etc.) o código fonte e/ou documentação completa correspondente deverá ser entregue integralmente, como parte integrante do relatório final do TCC.
Ciente dessa condição básica, declaro estar de acordo com a realização do TCC identificado pelos dados apresentados a seguir.

Instituição	ECL/INE/CTC
Nome do Responsável	José Luís Almada Güntzel
Cargo/Função	Prof. INE/CTC
Fone de Contato	(48) 99971-1982
Acadêmico	Lucas Yuki Imamura
Título do trabalho	Implementação, Validação e Análise do ePlace, Técnica Clássica de Posicionamento Global de Circuitos VLSI Baseada em Sistemas Eletrostáticos
Curso	Ciência da Computação/INE/UFSC

Florianópolis, 15 de Junho de 2026.

Responsável
Prof. José Luís Almada Güntzel, Dr.