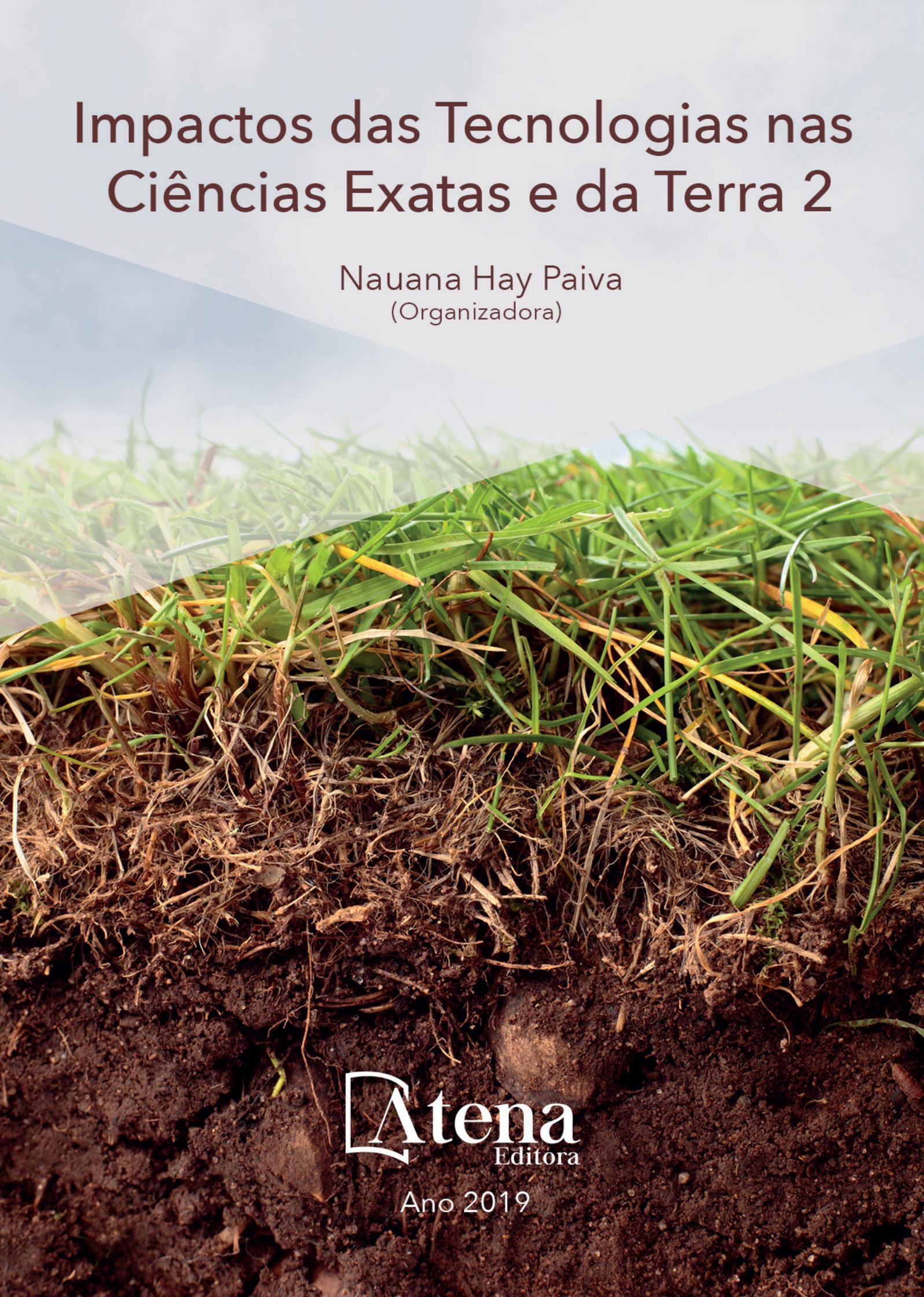


# Impactos das Tecnologias nas Ciências Exatas e da Terra 2

Nauana Hay Paiva  
(Organizadora)



**Atena**  
Editora

Ano 2019

Nauana Hay Paiva  
(Organizadora)

# Impactos das Tecnologias nas Ciências Exatas e da Terra 2

Atena Editora  
2019

2019 by Atena Editora

Copyright © da Atena Editora

Editora Chefe: Profª Drª Antonella Carvalho de Oliveira

Diagramação e Edição de Arte: Geraldo Alves e Karine de Lima

Revisão: Os autores

#### Conselho Editorial

- Prof. Dr. Alan Mario Zuffo – Universidade Federal de Mato Grosso do Sul  
Prof. Dr. Álvaro Augusto de Borba Barreto – Universidade Federal de Pelotas  
Prof. Dr. Antonio Carlos Frasson – Universidade Tecnológica Federal do Paraná  
Prof. Dr. Antonio Isidro-Filho – Universidade de Brasília  
Profª Drª Cristina Gaio – Universidade de Lisboa  
Prof. Dr. Constantino Ribeiro de Oliveira Junior – Universidade Estadual de Ponta Grossa  
Profª Drª Daiane Garabeli Trojan – Universidade Norte do Paraná  
Prof. Dr. Darllan Collins da Cunha e Silva – Universidade Estadual Paulista  
Profª Drª Deusilene Souza Vieira Dall’Acqua – Universidade Federal de Rondônia  
Prof. Dr. Eloi Rufato Junior – Universidade Tecnológica Federal do Paraná  
Prof. Dr. Fábio Steiner – Universidade Estadual de Mato Grosso do Sul  
Prof. Dr. Gianfábio Pimentel Franco – Universidade Federal de Santa Maria  
Prof. Dr. Gilmei Fleck – Universidade Estadual do Oeste do Paraná  
Profª Drª Girlene Santos de Souza – Universidade Federal do Recôncavo da Bahia  
Profª Drª Ivone Goulart Lopes – Istituto Internazionele delle Figlie de Maria Ausiliatrice  
Profª Drª Juliane Sant’Ana Bento – Universidade Federal do Rio Grande do Sul  
Prof. Dr. Julio Candido de Meirelles Junior – Universidade Federal Fluminense  
Prof. Dr. Jorge González Aguilera – Universidade Federal de Mato Grosso do Sul  
Profª Drª Lina Maria Gonçalves – Universidade Federal do Tocantins  
Profª Drª Natiéli Piovesan – Instituto Federal do Rio Grande do Norte  
Profª Drª Paola Andressa Scortegagna – Universidade Estadual de Ponta Grossa  
Profª Drª Raissa Rachel Salustriano da Silva Matos – Universidade Federal do Maranhão  
Prof. Dr. Ronilson Freitas de Souza – Universidade do Estado do Pará  
Prof. Dr. Takeshy Tachizawa – Faculdade de Campo Limpo Paulista  
Prof. Dr. Urandi João Rodrigues Junior – Universidade Federal do Oeste do Pará  
Prof. Dr. Valdemar Antonio Paffaro Junior – Universidade Federal de Alfenas  
Profª Drª Vanessa Bordin Viera – Universidade Federal de Campina Grande  
Profª Drª Vanessa Lima Gonçalves – Universidade Estadual de Ponta Grossa  
Prof. Dr. Willian Douglas Guilherme – Universidade Federal do Tocantins

#### Dados Internacionais de Catalogação na Publicação (CIP) (eDOC BRASIL, Belo Horizonte/MG)

134	Impactos das tecnologias nas ciências exatas e da terra 2 [recurso eletrônico] / Organizadora Nauana Hay Paiva. – Ponta Grossa (PR): Atena Editora, 2019. – (Impactos das Tecnologias nas Ciências Exatas e da Terra; v. 2)  Formato: PDF Requisitos de sistema: Adobe Acrobat Reader Modo de acesso: World Wide Web Inclui bibliografia ISBN 978-85-7247-053-7 DOI 10.22533/at.ed.537192201  1. Ciências exatas. 2. Tecnologia. I. Paiva, Nauana Hay. II. Série.  CDD 016.5
-----	---

Elaborado por Maurício Amormino Júnior – CRB6/2422

DOI O conteúdo dos artigos e seus dados em sua forma, correção e confiabilidade são de responsabilidade exclusiva dos autores.

2019

Permitido o download da obra e o compartilhamento desde que sejam atribuídos créditos aos autores, mas sem a possibilidade de alterá-la de nenhuma forma ou utilizá-la para fins comerciais.

[www.atenaeditora.com.br](http://www.atenaeditora.com.br)

## SUMÁRIO

<b>CAPÍTULO 1</b> .....	<b>1</b>
O ALUNO COMO SUJEITO ATIVO NO PROCESSO DE ENSINO-APRENDIZAGEM: OS IMPACTOS DAS METODOLOGIAS ATIVAS EM DIFERENTES MODALIDADES DA EDUCAÇÃO BÁSICA	
Sidney Silva Simplicio Alexsandra da Costa Andrade Maria do Socorro Tavares Cavalcante	
<b>DOI 10.22533/at.ed.5371922011</b>	
<b>CAPÍTULO 2</b> .....	<b>15</b>
COMPOSIÇÃO QUÍMICA DOS ÓLEOS ESSENCIAIS DE FOLHAS DE GOIABEIRAS: UMA REVISÃO DE LITERATURA	
Luiza Alves Mendes Amélia Carlos Tuler Carolina de Oliveira Bernardes Drielli Canal Marianna Junger de Oliveira Garozi José Henrique Soler Guilhen Lidiane Gomes dos Santos	
<b>DOI 10.22533/at.ed.5371922013</b>	
<b>CAPÍTULO 3</b> .....	<b>24</b>
INFLUÊNCIA DO TEMPO DE CONTATO NA ADSORÇÃO DE NI(II) EM BIOCÁRVÕES ORIUNDOS DAS CASCAS DE EUCALIPTO E PALHA DE CAFÉ	
Ruan de Oliveira Alves D'ávila Leal Polastreli Ueslei Giori Favero Yago Ricardo de Oliveira Tiago Guimarães Lucas Destefani Paquini Bruno Regis Lyrio Ferraz Renato Ribeiro Passos Demetrius Profeti Luciene Paula Roberto Profeti	
<b>DOI 10.22533/at.ed.5371922014</b>	
<b>CAPÍTULO 4</b> .....	<b>30</b>
AVALIAÇÃO DA ADSORÇÃO DE CO(II) UTILIZANDO BIOCÁRVÕES DE PALHA DE CAFÉ COMO MATERIAL ADSORVENTE	
Ueslei Giori Favero Yago Ricardo de Oliveira D'ávila Leal Polastreli Ruan de Oliveira Alves Tiago Guimarães Lucas Destefani Paquini Bruno Regis Lyrio Ferraz Renato Ribeiro Passos Demetrius Profeti Luciene Paula Roberto Profeti	
<b>DOI 10.22533/at.ed.5371922015</b>	

**CAPÍTULO 5 ..... 36**

DEGRADAÇÃO DO FUNGICIDA FLUTRIAFOL UTILIZANDO NANOPARTÍCULAS BIMETÁLICAS DE FE/NI, FE/CU E CU COM ANÁLISE POR GC/MS

Maxwell Daniel de Freitas  
Karla Moreira Vieira  
Vanessa Moreira Osorio  
Isabela Cristina de Matos Cunha  
Renata Pereira Lopes Moreira

**DOI 10.22533/at.ed.5371922016**

**CAPÍTULO 6 ..... 50**

ANÁLISE TEMPORAL DA PRODUÇÃO AGROPECUÁRIA DO MUNICÍPIO DE ARROIO DO PADRE/RS, ENTRE OS ANOS DE 2001 E 2016

Alison André Domingues Teixeira  
Clismam Soares Porto  
Alexandre Felipe Bruch  
Angélica Cirolini  
Marciano Carneiro  
Jéssica Stern Behling

**DOI 10.22533/at.ed.5371922017**

**CAPÍTULO 7 ..... 63**

MAPEAMENTO DO USO DA TERRA E SEUS CONFLITOS EM ÁREAS DE PRESERVAÇÃO PERMANENTE NO MUNICÍPIO DE ARROIO DO PADRE, RS

Alison André Domingues Teixeira  
Clismam Soares Porto  
Angélica Cirolini  
Alexandre Felipe Bruch  
Marciano Carneiro  
Marinêz da Silva

**DOI 10.22533/at.ed.5371922018**

**CAPÍTULO 8 ..... 76**

AValiação DA CONdição CORPORAL DOS CÃES DOMICILIADOS DO MUNICÍPIO DE REALEZA/PR

Jhenifer Cintia Beneti  
Anne Caroline de Aguiar Pesenti  
Andressa Silveira dos Santos  
Glauco Eleutherio da Luz  
Everton Artuso  
Luciana Pereira Machado

**DOI 10.22533/at.ed.5371922019**

**CAPÍTULO 9 ..... 81**

IMPACTO DO TURISMO SOBRE A HIDROGRAFIA DO PARQUE ESTADUAL MARINHO DE AREIA VERMELHA, CABEDELO/PB: CONTRIBUIÇÕES PARA GESTÃO AMBIENTAL

Daniel Silva Lula Leite  
George Emmanuel Cavalcanti de Miranda

**DOI 10.22533/at.ed.53719220110**

**CAPÍTULO 10 ..... 98**

ESTUDO GEOLÓGICO E DO COMPORTAMENTO ESTRUTURAL EM ÁREA PARA PRODUÇÃO DE BRITA EM VERA CRUZ (RS)

Cândida Regina Müller  
Thays França Afonso  
Leandro Fagundes  
Luis Eduardo Silveira da Mota Novaes'

**DOI 10.22533/at.ed.53719220111**

**CAPÍTULO 11 ..... 106**

FLUXOS DE CALOR E RADIAÇÃO DE ONDA LONGA EM SUPERFÍCIE DURANTE TEMPESTADE TORNÁDICA EM TAQUARITUBA/SP

Kelli Silva de Lara  
Allef Patrick Caetano de Matos  
André Becker Nunes

**DOI 10.22533/at.ed.53719220112**

**CAPÍTULO 12 ..... 115**

SOBRE A INTERAÇÃO DE PÓRTICOS PLANOS COM O MEIO CONTÍNUO MODELADOS PELO MEC

Welky Klefson Ferreira de Brito  
José Marcílio Filgueiras Cruz  
Ângelo Vieira Mendonça

**DOI 10.22533/at.ed.53719220113**

**CAPÍTULO 13 ..... 137**

FÍSICA DO MEIO AMBIENTE: ESTADO DA ARTE

Thiago Moura Zetti  
Milton Souza Ribeiro Miltão

**DOI 10.22533/at.ed.53719220114**

**CAPÍTULO 14 ..... 146**

ESTUDO DO GRUPO DE POINCARÉ E DE SUAS REPRESENTAÇÕES IRREDUTÍVEIS

Ana Camila Costa Esteves  
Milton Souza Ribeiro Miltão

**DOI 10.22533/at.ed.53719220115**

**CAPÍTULO 15 ..... 165**

UMA REVISÃO SOBRE O PROBLEMA DE POSICIONAMENTO NO PROJETO DE CIRCUITOS INTEGRADOS MODERNOS

Mateus Paiva Fogaça  
Jacques de Jesus Figueiredo Schmitz Junior  
Paulo Francisco Butzen  
Cristina Meinhardt

**DOI 10.22533/at.ed.53719220116**

**CAPÍTULO 16 ..... 188**

UMA IMPLEMENTAÇÃO DE CONTROLADOR DE ACESSOS DE BAIXO CUSTO UTILIZANDO CARTÕES RFID

Wagner Loch  
Rafael Iankowski Soares

**DOI 10.22533/at.ed.53719220117**

**CAPÍTULO 17 ..... 193**

AGROQUÍMICOS: LEVANTAMENTO DO USO NA CIDADE DE FORMOSA DA SERRA NEGRA/MA E  
UMA PROPOSTA PARA TRABALHOS EM SALA DE AULA

Janyeid Karla Castro Sousa  
Jemmla Meira Trindade Moreira  
Andréa Soares de Souza Barros

**DOI 10.22533/at.ed.53719220118**

**SOBRE A ORGANIZADORA..... 209**

## UMA REVISÃO SOBRE O PROBLEMA DE POSICIONAMENTO NO PROJETO DE CIRCUITOS INTEGRADOS MODERNOS

### Mateus Paiva Fogaça

Universidade Federal do Rio Grande do Sul,  
Departamento de Informática  
Porto alegre – RS

### Jacques de Jesus Figueiredo Schmitz Junior

Universidade Federal do Rio Grande, Centro de  
Ciências Computacionais  
Rio Grande – RS

### Paulo Francisco Butzen

Universidade Federal do Rio Grande, Centro de  
Ciências Computacionais  
Rio Grande – RS

### Cristina Meinhardt

Universidade Federal de Santa Catarina,  
Departamento de Informática e Estatística  
Florianópolis – SC

**RESUMO:** As principais etapas da síntese física de circuitos integrados são o posicionamento e o roteamento. O posicionamento é responsável por encontrar uma posição para cada um dos componentes do circuito enquanto o roteamento é responsável por realizar as interconexões entre os componentes após o posicionamento. Ambas etapas otimizam funções de custo que avaliam a solução gerada. Embora sejam alvo de pesquisas por mais de 50 anos, o constante crescimento da complexidade dos circuitos cria novos desafios. As atuais tecnologias

de fabricação de circuitos integrados, com elevado número de componentes e transistores nanométricos, aumentou a complexidade destas etapas e destacou a importância de lidar com outras métricas além do tamanho das interconexões, como análise de tempos de propagação e roteabilidade do circuito. Dada a vasta literatura na área, este trabalho tem como objetivo apresentar uma revisão bibliográfica sobre as etapas de posicionamento e roteamento.

**PALAVRAS-CHAVE:** Microeletrônica, automação do projeto eletrônico, posicionamento, roteamento

**ABSTRACT:** Placement and routing are the main steps of the physical synthesis of integrated circuits. Placement is responsible for finding a position for each of the circuit components and routing is responsible for tracing the interconnections among the components after placement. Both steps optimize objective functions that evaluate the existing solution. Although placement and routing have been studied for more than 50 years, the steady growth of circuits size raises new challenges. In current technologies, the large number of components and nanometric transistors rise the complexity of these steps and highlight the need for other metrics than wirelength, such as timing and routeability. Given the extensive literature

on this field, this work presents a survey on placement and routing.

**KEYWORDS:** Microelectronics, electronic design automation, placement, routing

## 1 | INTRODUÇÃO

Circuitos integrados (CIs) fazem parte da quase totalidade de equipamentos eletrônicos dos dias atuais. O constante crescimento da complexidade funcional destes circuitos se reflete no aumento do número de componentes integrados dentro de um único chip. Projetos atuais podem contar com mais de 1 bilhão de transistores, com uma demanda do mercado por um curto tempo de desenvolvimento. Estes fatores levaram a necessidade de automação dos processos envolvidos no projeto de CIs, implicando no desenvolvimento de ferramentas de EDA (*Electronic Design Automation*) a fim de tornar o projeto mais rápido, confiável e barato. Tanto a indústria, quanto a academia realizam pesquisas por melhores algoritmos e estruturas de dados capazes de lidar com os desafios destes projetos.

Atualmente, o projeto de circuitos integrados para aplicações específicas, conhecidos como ASICs (*Application-specific integrated circuits*) pode ser realizado através de duas principais metodologias: o projeto *full custom* ou o projeto *standard cell* (Reis, 2000). Projetos *standard cell* permitem maior rapidez de projeto e confiabilidade, por adotar células padrão já devidamente projetadas e caracterizadas. Este trabalho focará no desenvolvimento de ferramentas de EDA voltadas para o fluxo de síntese da metodologia *standard cell*.

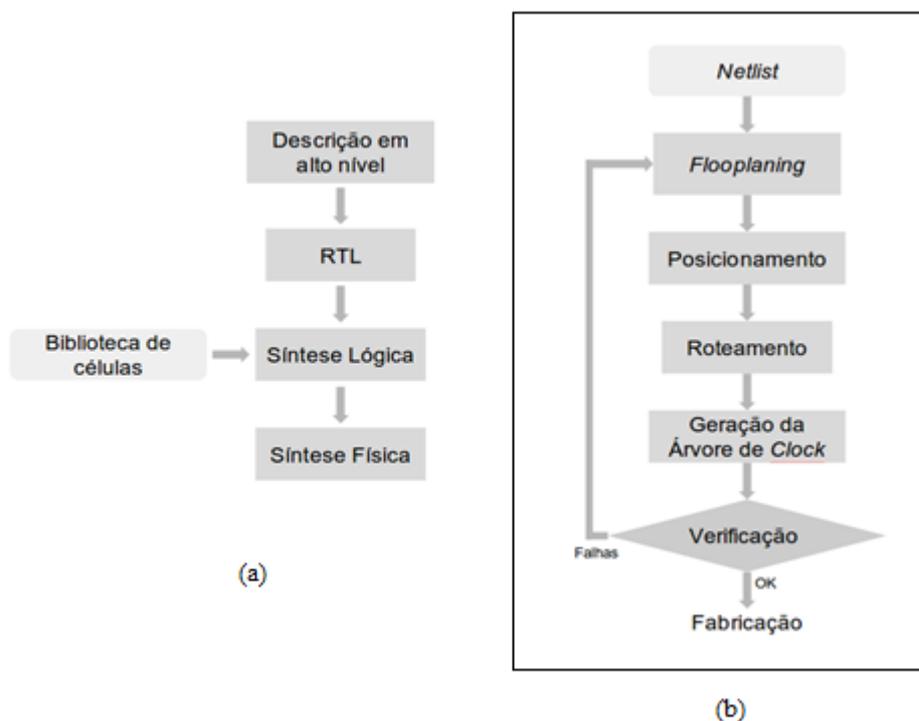
O fluxo de projeto é um conjunto de procedimentos que permite aos projetistas percorrerem um caminho livre de erros desde a especificação do sistema até a sua implementação no silício (WESTE; HARRIS, 2010). A Figura 1 (a) apresenta um fluxo de projeto de circuitos integrados em linhas gerais. O fluxo de síntese da metodologia *standard cell* pode ser dividido em 4 grandes fases: modelagem em alto-nível, síntese RTL, síntese lógica e síntese física.

A primeira fase do projeto é a modelagem em alto nível. Nesta fase, uma descrição comportamental do sistema é construída através de uma linguagem HDL (*Hardware description language*) como SystemC, VHDL, Verilog ou até mesmo C/C++. O objetivo é descrever o comportamento do sistema de forma que ele possa ser simulado e validado.

Em seguida, a descrição em alto nível é transformada em um conjunto de registradores e o fluxo de sinais entre eles através de operações lógicas, originando a descrição RTL (*register transfer level*). Na síntese lógica, a descrição RTL é inicialmente transformada em um circuito com funções lógicas genéricas. Assim é possível realizar otimizações independentes de tecnologia através da reestruturação das funções lógicas no circuito. O circuito passa então pelo mapeamento tecnológico. Esta etapa é responsável por substituir as funções lógicas genéricas por portas lógicas presentes

na biblioteca de células adotada no projeto. Depois de mapeado, o circuito passa pela otimização dependente da tecnologia. Ela reestrutura o circuito baseada em informações elétricas das portas lógicas, como atrasos, consumo de potência e área.

Após a conclusão da síntese lógica, o circuito passa para a etapa de síntese física. A síntese física é responsável por converter um circuito no domínio estrutural para o domínio físico. A Figura 1 (b) mostra um típico fluxo de síntese física para projetos com metodologia *standard-cell*, adotado para a maioria dos projetos de ASICs.



**Figura 1.** Fluxo de projeto (a) e fluxo de síntese física (b).

A primeira etapa da síntese física é chamada de planejamento topológico (*floorplaning*). Com o crescimento da complexidade dos circuitos integrados, o uso de abordagens hierárquicas passou a ser amplamente adotado (WANG; CHANG; CHENG, 2009). Estas abordagens dividem os componentes do circuito em módulos de acordo com a tarefa que exercem. O objetivo do *floorplaning* é realizar uma divisão inicial no *chip*, reservando uma região para cada um desses blocos bem como determinar a posição dos pinos de entrada e saída do circuito, e fazer a distribuição das redes de alimentação. As duas etapas seguintes da síntese física, posicionamento e roteamento, acontecem individualmente em cada um desses módulos.

O posicionamento é a etapa responsável por encontrar uma posição válida para cada uma das portas lógicas, também chamadas de células, na área disponível para o projeto em um circuito integrado, obedecendo ainda outras restrições de projeto. Esta etapa tem uma grande importância para o projeto, pois afeta diretamente o desempenho, consumo de potência e atrasos do circuito. Posicionadores geralmente recebem como entrada a relação de todas as células e suas conexões, conhecida como

*netlist*, e o *floorplaning*. Além disso, ferramentas de posicionamento também recebem a descrição física das portas lógicas da biblioteca de células usadas no projeto.

A etapa de roteamento é responsável por determinar as rotas exatas pelas quais os fios que realizam a troca de sinais entre as células (interconexões) serão realizados. Circuitos modernos podem conter centenas de milhões de conexões e as ferramentas de roteamento devem ser capazes de planejar todas elas. Para realizar esta tarefa, o roteamento é dividido em dois passos: o roteamento global e o roteamento detalhado. O roteamento global divide a área do *chip* em regiões de tamanho regular e cria as rotas que percorrem essas regiões. As rotas criadas pelo roteamento global também são conhecidas como “guias para o roteamento” (*routing guides*). Já o roteamento detalhado é responsável por determinar o caminho exato que os fios percorrerão, obedecendo ao máximo as guias geradas pelo roteamento global.

Em circuitos síncronos, os dados são processados obedecendo um sinal de clock. A última etapa da metodologia standard cell é a geração da árvore de clock, responsável por distribuir este sinal a todos os registradores do circuito. Esta etapa tenta garantir que o atraso relativo entre cada registrador seja o mínimo possível, fazendo com que todos os dados sejam amostrados ao mesmo tempo.

Depois de posicionado, roteado e a árvore de clock gerada, o circuito passa por ferramentas de análise e verificação. Uma delas é responsável pela extração de efeitos parasitas associados as conexões. Esses efeitos ocorrem devido as indutâncias, capacitâncias e resistências do circuito. Neste ponto é possível verificar com maior precisão se o circuito corresponde as especificações de frequência, potência e área. Esta etapa é conhecida por ser um gargalo no projeto pois muitas vezes é necessário realizar novamente o posicionamento e o roteamento repetidamente até que se atinja as especificações de projeto desejadas.

A maioria dos projetos de CIs tem como principal objetivo atingir uma frequência de operação. Por isso, os fluxos de projeto realizam a análise dos tempos de propagação (atrasos) do circuito, conhecida como análise de timing. Por fim, são realizadas simulações para verificar ruídos, queda na tensão de alimentação e limites de eletromigração. Se o projeto atender as especificações, está pronto para fabricação.

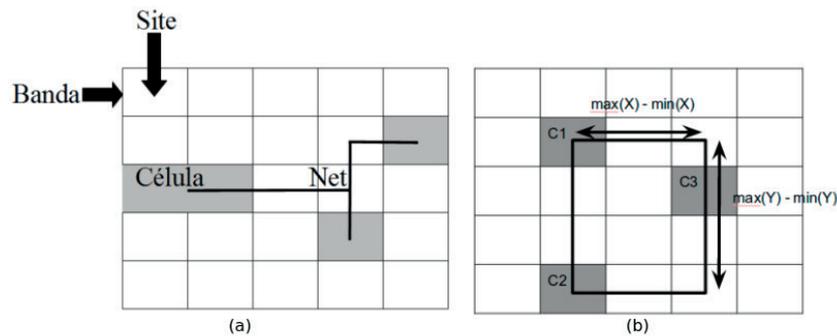
Este trabalho se baseia em um conjunto de conceitos básicos. A Figura 2 (a) ilustra estes principais conceitos, definidos como:

- **Célula:** é uma porta lógica presente na biblioteca de células. Estas bibliotecas contêm todas as portas lógicas de um circuito, assim como suas características físicas (altura, largura, pinos de entrada e saída, tecnologia, dentre outros).
- **Banda:** um chip é dividido em linhas de altura fixa. Estas linhas recebem o nome de bandas.
- **Site:** cada banda por sua vez é dividida em fatias de largura fixa que recebem o nome de sites. O site é o menor espaço que uma célula pode ocupar. En-

quanto portas lógicas mais simples ocupam 1 ou 2 sites, as mais complexas podem ocupar 30 ou mais.

- **Net:** é uma interconexão que liga duas ou mais células. Representa o fio que conectará os pinos de entrada ou saída de duas ou mais células, constituindo um circuito.
- **Half-perimeter wirelength (HPWL):** é uma estimativa do comprimento das interconexões utilizada para avaliar a qualidade do posicionamento. A Figura 2 (b) mostra o cálculo do HPWL para uma net com 3 células. Primeiramente, encontra-se a mínima área retangular que englobe o centro de todas as redes. O HPWL será o meio perímetro dessa área.

O principal objetivo deste texto é fornecer uma completa revisão dos problemas de posicionamento e roteamento no projeto de CIs, apresentando conceitos, definições e os principais algoritmos e metodologias empregados atualmente.



**Figura 2.** Representação gráfica de (a) banda, site, célula e net e (b) estimativa do comprimento das interconexões por HPWL.

## 2 | POSICIONAMENTO

O problema de encontrar uma posição válida para cada um dos componentes do circuito na área do chip pode ser formalmente definido representando o circuito por um hipergrafo  $G = (E, V)$ , onde o conjunto de arestas  $E$  representa os fios que conectam componentes do circuito e o conjunto de vértices  $V$  representa os componentes.

Objetivo: encontrar uma posição  $(X, Y)$  para cada componente, respeitando a restrição apresentada na Eq. 1:

$$(\vec{X}, \vec{Y}) \in [X_{min}, X_{max}] \times [Y_{min}, Y_{max}] \text{ Eq. (1)}$$

O posicionamento é um problema de otimização e NP-completo (HENTSCHKE, 2002). Nesta etapa, otimizar significa minimizar o valor de alguma variável do circuito. São exemplos de variáveis que se deseja minimizar: *wirelength*, atrasos, dissipação de potência e área. Algumas ferramentas modernas visam maximizar a roteabilidade do

circuito, deixando espaços vazios no chip para facilitar o acesso aos pinos de entrada e saída das células. A etapa de posicionamento tradicionalmente é composta por três etapas: posicionamento global, legalização e posicionamento detalhado.

Os algoritmos de posicionamento podem ser classificados em dois grupos de acordo com seus parâmetros de entrada:

- **Construtivos:** produzem um posicionamento apenas a partir da *netlist* do circuito. Posicionadores quadráticos podem ser considerados algoritmos construtivos.
- **Iterativos:** necessitam de um posicionamento inicial e realizam alterações iterativas em busca de uma solução melhor. Posicionadores que utilizam a meta-heurística *simulated annealing* são exemplos de algoritmos iterativos.

Outra maneira de classificar os algoritmos é de acordo com a saída produzida. Os algoritmos que para uma determinada configuração de entrada sempre produzem o mesmo resultado são chamados determinísticos. Os algoritmos que se baseiam em funções aleatórias e probabilidades e por isso podem produzir resultados diferentes para a mesma configuração de entrada são chamados de probabilísticos. Contudo, a maneira mais utilizada para classificar os algoritmos de posicionamento é de acordo com a sua natureza:

- **Algoritmos de particionamento:** utilizam heurísticas de particionamento para dividir o problema em instâncias menores e resolvê-las individualmente. Exemplo são técnicas de divisão e conquistas onde circuito é tratado como um hipergrafo e particionado recursivamente para facilitar a solução do problema.
- **Algoritmos estocásticos:** adotam algoritmos probabilísticos para encontrar uma solução viável mesmo reduzindo o domínio de busca.
- **Algoritmos analíticos:** representam o posicionamento através de uma função custo que se deseja otimizar.

## 2.1 Posicionamento Global

O objetivo do posicionamento global é realizar a distribuição das células pela área do chip enquanto otimiza a função custo. A etapa do posicionamento global tem grande impacto no resultado final em termos de qualidade e tempo de execução. Por este motivo, é alvo da maioria das pesquisas em posicionamento (WANG; CHANG; CHENG, 2009). As primeiras ferramentas de posicionamento se caracterizavam por utilizar técnicas de divisão-e-conquista. Nelas, o circuito era tratado como um hipergrafo e particionado recursivamente para facilitar a solução do problema. A meta-heurística *simulated annealing* (SA) (BROOKS; MORGAN, 1995; MARKOV; HU; KIM, 2012).

foi aplicada com sucesso ao problema do posicionamento, dominando as primeiras ferramentas comerciais e acadêmicas de posicionamento. Entretanto, este algoritmo é conhecido por ser lento quando o número de componentes é elevado. Isto se tornou um problema com o constante crescimento dos circuitos. Para lidar com isso, surgiram os posicionadores analíticos. As abordagens analíticas tratam o problema do posicionamento como um sistema de equações lineares originado de uma função custo que se deseja otimizar (MARKOV; HU; KIM, 2012).

### 2.1.1 Simulated annealing

O *simulated annealing* é uma técnica de otimização não determinística. Ele é inspirado no processo metalúrgico de fabricação dos metais. Neste processo a temperatura do metal é elevada até um valor no qual as moléculas estão tão agitadas que não obedecem a nenhuma organização. A próxima etapa consiste em resfriar o sistema lentamente. Isto permite que os átomos se arranjam de forma uniforme, diminuindo os defeitos do metal. Quando a temperatura assume valores baixos os átomos formam uma rede cristalina.

O Algoritmo 1 (HENTSCHKE, 2002) apresenta uma implementação do *simulated annealing* aplicado ao problema do posicionamento. É necessário conhecer três parâmetros: posicionamento inicial, temperatura inicial e número de perturbações. As principais funções são as de perturbação (linha 6), custo (linha 7) e *schedule* de temperatura (linha 12):

**Função de perturbação:** tem por objetivo alterar um posicionamento prévio. Estas alterações podem ser realizadas através de trocas aleatórias de posição, sejam trocas simples ou duplas. Na troca simples, sorteia-se uma célula e uma posição ainda não ocupada. Já na troca dupla, sorteiam-se duas células para trocarem de posição.

**Função custo:** calcula um valor numérico a fim de avaliar um posicionamento. As abordagens tradicionais estão relacionadas com a estimativa do comprimento total dos fios nas interconexões entre as células, atrasos, potência e congestionamento. A métrica mais utilizada é a estimativa do comprimento dos fios dada pelo HPWL.

**Schedule de temperatura.** O *schedule* de temperatura pode ser definido como uma função matemática decrescente que determina o comportamento da temperatura conforme a execução do algoritmo. Escolher um bom *schedule* de temperatura é de extrema importância para o SA obter bons resultados (LAM; DELOSME, 1988).

O algoritmo apresenta dois laços de repetição: o laço externo controlado pela variável temperatura (linhas 4-13) e o laço interno controlado pela variável *número\_de\_perturbações* (linhas 5-11). Em cada iteração do laço interno a solução é perturbada (linha 6). O próximo passo é avaliar essa troca através de uma função custo. As trocas que diminuem o custo sempre são aceitas enquanto as que aumentam estão sujeitas a expressão  $e^{\frac{\Delta\text{custo}}{\text{temperatura}}}$  (linhas 8-14). Logo, a variável temperatura é responsável pela aceitação das trocas. Inicia-se temperatura com um valor alto visando a aceitação

de muitas trocas “ruins”. Este valor é reduzido ao longo da execução do algoritmo de acordo a função *schedule* (linha 12). Conforme a *temperatura* é reduzida, o algoritmo adquire um comportamento guloso. Desta maneira, o *simulated annealing* consegue obter bons resultados, evitando mínimos locais.

---

**Algoritmo 1:** Pseudocódigo do *Simulated Annealing*.

---

**Entrada:** posicionamento\_inicial, temperatura\_inicial,  
número\_de\_perturbações  
**Saída:** pos

```

1 begin
2   temperatura ← temperatura_inicial;
3   pos ← posicionamento_inicial;
4   repeat
5     for  $i \leftarrow 0$  to número_de_perturbações do
6       novo_pos ← perturbação(pos);
7        $\Delta\text{custo} = \text{custo}(\text{novo\_pos}) - \text{custo}(\text{pos});$ 
8       if  $\Delta\text{custo} < 0$  then
9         pos = novo_pos;
10      else if  $\text{random}(0,1) > e^{\frac{\Delta\text{custo}}{\text{temperatura}}}$  then
11        pos = novo_pos;
12   temperatura ← schedule(temperatura);
13 until temperatura = 0;

```

---

### 2.1.2 Posicionamento analítico

O posicionamento analítico formula o custo do posicionamento como uma função matemática das posições das células do circuito e tenta otimizá-la através de técnicas analíticas. Dependendo da função modelada as técnicas são classificadas como quadráticas ou não-quadráticas. Técnicas quadráticas utilizam funções convexas que podem ser otimizadas resolvendo um conjunto de equações lineares. Em contrapartida, as não-quadráticas utilizam funções não lineares, por exemplo o modelo *log-sum-exp* de *wirelength* e a função de densidade *bell-shaped* (NAYLOR; DONELLY; SHA, 2001). Elas obtêm resultados tão bons quanto as quadráticas, porém são mais difíceis de implementar e pouco eficientes computacionalmente.

O HPWL é uma função convexa que não pode ser derivada, impedindo seu uso no posicionamento analítico. A solução é utilizar outros modelos que capturem o objetivo de redução dos fios. A técnica mais consolidada na literatura é o modelo de *wirelength* quadrático. O *wirelength* quadrático de um fio que liga dois componentes é dado pela Eq. II. Logo, o *wirelength* quadrático total do circuito é dado pela Eq. III, onde  $c_{i,j}$  é um peso associado a cada conexão. Tradicionalmente o valor de  $c_{i,j}$  para um fio que liga dois componentes é 1. Caso não haja conexão entre eles  $c_{i,j}$  assume o valor 0. Este modelo é implementado em sua forma matricial, pela Eq. IV. Nesta

notação,  $x$  e  $y$  são os vetores que armazenam as posições das células, ou seja, são variáveis.  $Q$  é uma matriz obtida a partir da fórmula  $Q = D - C$ , onde  $D$  é uma matriz diagonal tal que  $D_{i,i} = \sum_{j \in V} C_{i,j}$  e  $C$  é a matriz de conectividade do hipergrafo do circuito. Por fim,  $d_x$  e  $d_y$  são vetores que armazenam as conexões com componentes fixos na forma  $d_{x_i} = -\sum_{j \in V_{fixos}} c_{i,j} x_j$ . Para encontrar os valores de  $x$  e  $y$  que gerem o wirelength mínimo, conforme a Eq.V,  $L$  é derivado parcialmente com relação a  $x$  e  $y$  e o resultado é igualado a 0, gerando um conjunto de equações lineares para o eixo  $x$  e outro para o eixo  $y$ .

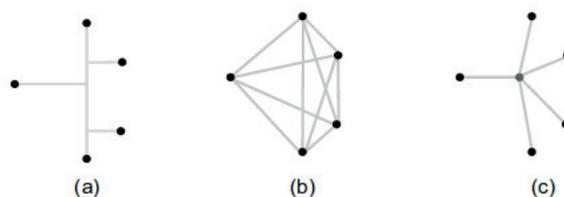
$$L_{i,j} = (x_i - x_j)^2 + (y_i - y_j)^2 \quad \text{Eq. II}$$

$$L = \frac{1}{2} \sum_{1 \leq i < j \leq n} c_{i,j} [(x_i - x_j)^2 + (y_i - y_j)^2] \quad \text{Eq. III}$$

$$L = \frac{1}{2} x^T Q x + d_x^T x + \frac{1}{2} y^T Q y + d_y^T y \quad \text{Eq. IV}$$

$$\frac{\partial L}{\partial x} = Q_x + d_x = 0 \quad (VI) \quad \frac{\partial L}{\partial y} = Q_y + d_y = 0 \quad \text{Eq. V}$$

A abordagem apresentada só trata conexões entre 2 elementos. Na prática, os fios que realizam as conexões do circuito podem conectar múltiplos pinos, como ilustrado na Figura 3 (a). Para se adequar ao modelo, estas conexões precisam ser decompostas. Existem diversas metodologias na literatura para realizar esta tarefa. A Figura 3 (b) mostra o modelo clique. Neste modelo a conexão original é substituída por um grafo completo, ou seja, todos são conectados com todos. Para manter o sistema balanceado, o peso das conexões no modelo clique é  $c/(k-1)$  (VYGEN, 1997), onde  $c$  representa o peso da conexão original e  $k$  é o número de pinos. Uma alternativa é a modelo estrela, apresentado na Figura 3 (c). Neste modelo todos os pinos são conectados a um nodo adicional, chamado de estrela (VYGEN, 1997). O nodo estrela é adicionado ao sistema como se fosse uma célula do circuito.



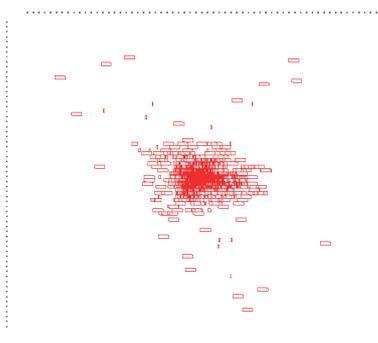
**Figura 3.** Representação para uma conexão com múltiplos pinos (a) através do modelo clique (b) e modelo estrela (c).

O trabalho de Viswanathan (VISWANATHAN; PAN; CHU, 2007) demonstra que os dois modelos são equivalentes desde que os pesos das conexões tenham sido atribuídos corretamente, e propõe um modelo híbrido (VISWANATHAN; PAN; CHU, 2007) que adota cliques para conexões com 3 pinos ou menos, e estrela para as demais. Devido ao uso de nodos estrela, o número de conexões para *benchmarks*

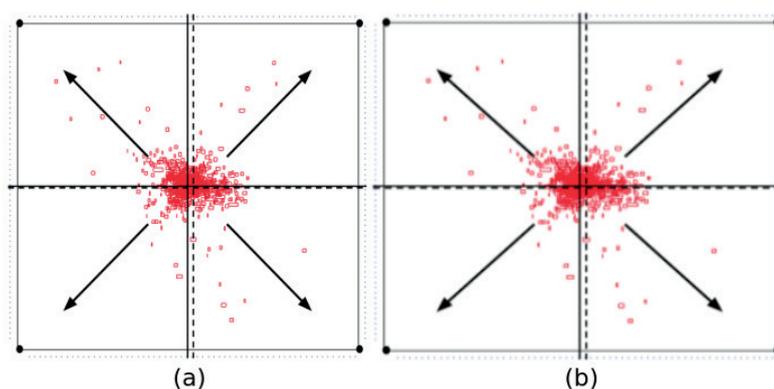
industriais diminuí em torno de 10 vezes em comparação ao modelo clique puro. Além disso, o uso do modelo clique para conexões com 2 ou 3 pinos evita que muitas variáveis adicionais sejam criadas pela inserção de nodos estrela. Estes dois fatores contribuem para otimizar a execução do posicionamento quadrático.

O modelo *Bound2Bound* (SPINDLER; JOHANNES, 2007) se destaca dentre os demais por representar precisamente o HPWL através do *wirelength* quadrático, permitindo que as ferramentas utilizem o HPWL como métrica de otimização; além de gerar um número de conexões menor que os demais, exceto para conexões com 4 pinos ou mais; e não introduz variáveis adicionais. Neste modelo, as conexões resultantes são realizadas apenas com os nodos extremos. Todas as conexões possuem peso  $w = 2/(k-1)l$ , onde  $l$  é a distância entre os pinos.

O posicionamento quadrático pode ser fisicamente interpretado como um sistema de molas onde os componentes do circuito são pontos adimensionais, as conexões entre eles são molas e os *pads* são pontos fixos. A Figura 4 mostra o resultado do posicionamento quadrático para um *benchmark* industrial com 12506 células e 246 *pads*. Devido ao fato desta formulação não considerar as dimensões dos componentes, o resultado geralmente possui muitas sobreposições principalmente na região central do *chip*. Por consequência o resultado do posicionamento quadrático não é uma solução viável. Uma alternativa para lidar com este problema é a adição de forças extras ao sistema chamadas de *spreading* ou *diffusion forces*. Estas forças têm como objetivo iterativamente deslocar as células de regiões mais densas para regiões menos densas. Neste caso, não se resolve apenas um sistema de equações lineares, mas quantos forem necessários até as células ficarem uniformemente distribuídas.



**Figura 4.** Resultado do posicionamento quadrático.



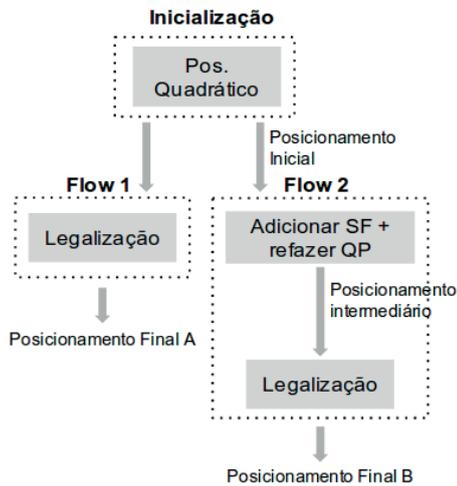
**Figura 5.** (a) Modelo 1 (linhas contínuas) e modelo 2 (linhas tracejadas) e (b) modelo 3 (linhas contínuas) e modelo 4 (linhas tracejadas).

### 2.1.3 Avaliação de metodologias de aplicação de *spreading forces*

Para demonstrar a influência do modelo de força de espalhamento adotado na

solução quadrática, vamos considerar quatro modelos de *spreading forces*. Em todos, as forças foram implementadas como pinos virtuais ligados as células. A Figura 5 (a) apresenta os dois primeiros modelos avaliados. Primeiramente, quatro pinos virtuais são colocados nos cantos do *chip*. A seguir, as células são divididas em quatro grupos, cada um deles conectado a um pino. A diferença entre os modelos está na forma de dividir os grupos. O modelo 1, representado por linhas contínuas, divide o circuito no centro físico do *chip*. O modelo 2, representado por linhas tracejadas, divide o circuito através do centro de massa do posicionamento. Nos modelos 3 e 4, ilustrados na Figura 5 (b), a posição dos pinos virtuais varia para cada célula. A posição do pino virtual é calculada com base em um ponto de origem e a posição da célula. No modelo 3 (linhas contínuas) o ponto de origem é o centro físico do *chip* enquanto que no modelo 4 (linhas tracejadas) é o centro de massa do posicionamento. A posição do pino virtual é determinada através de uma reta que parte do ponto de origem e se prolonga até a borda do *chip*. O pino é colocado no ponto de intersecção da reta com a borda.

Dois fluxos de execução foram aplicados nos quatro primeiros *benchmarks* da Tabela I. A Figura 6 mostra os passos executados neste trabalho. A etapa de inicialização gera uma solução inicial através do posicionamento quadrático. Este posicionamento serve como entrada para os dois fluxos de execução. No fluxo 1 a solução é legalizada gerando o *Posicionamento Final A*. No fluxo 2, as *spreading forces* são adicionadas e o posicionamento quadrático é refeito. Por fim, o posicionamento intermediário gerado nesta etapa é legalizado gerando o *Posicionamento Final B*. Em ambos os fluxos, a etapa de legalização foi realizada com a ferramenta PlaceUtil (PLACEUTIL, 2014).



**Figura 6.** Etapa de inicialização e fluxos de execução

Circuito	Fluxo 1 - HPWL (E+06)	Fluxo 2 - HPWL (E+06)			
		Modelo 1	Modelo 2	Modelo 3	Modelo 4
IBM01	12,0	5,1	5,3	6,0	5,9
IBM02	23,0	12,6	12,5	16,8	16,0
IBM03	31,0	14,5	14,0	14,4	12,8
IBM04	41,3	26,2	17,8	24,2	18,3

**Tabela III.** HPWL final para cada fluxo.

Circuito	Pos. Quadrático Inicialização (s)	Adicionar SF e refazer PQ (s)	Legalização Fluxo 1 (s)	Legalização Fluxo 2 (s)
IBM01	427,19	393,72	20,07	2,94
IBM02	1263,94	2716,02	43,33	5,79
IBM03	2069,17	2205,46	53,28	2,60
IBM04	3172,80	4735,24	64,71	8,24

**Tabela IV.** Tempo de execução para cada passo.

Circuito	Fluxo 1 (s)	Fluxo 2 (s)
IBM01	447,26	823,85
IBM02	1307,27	3985,75
IBM03	2122,45	4277,23
IBM04	3237,51	7916,28

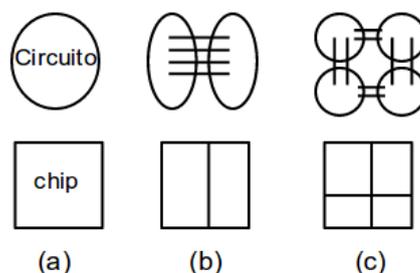
**Tabela V.** Tempo de execução total para cada fluxo.

Todos os experimentos foram implementados utilizando a linguagem C++ e executados em um Intel Core i7-2670QM 2.20GHz com 6GB de memória. A Tabela III mostra uma comparação entre os dois fluxos de execução em termos de *wirelength*. O fluxo 2 foi executado com todos os modelos de *spreading forces* apresentados. Os resultados mostram que o fluxo 2 gera resultados em média 50% melhores, sendo que nenhum modelo se destacou dentre os demais. A outra análise compara os tempos de execução. Os resultados mostram que o tempo de execução do posicionamento quadrático dobra com o uso de *spreading forces*. A única exceção é o IBM02, no qual o tempo triplica. No entanto, o tempo de legalização reduz em 80%. A Tabela V compara o tempo de execução total para cada fluxo. É possível notar que o tempo de execução total é determinado pelo posicionamento quadrático, sendo desprezível o ganho no tempo de legalização.

### 2.1.4 Particionamento

O posicionamento por particionamento pode ser definido da seguinte maneira: dado um circuito e uma área de *chip*, cada célula é atribuída a uma região específica no espaço disponível para posicionamento. A Figura 7 ilustra o funcionamento de uma ferramenta de posicionamento por particionamento. Primeiramente, o circuito é dividido em dois subcircuitos. Da mesma maneira, a área física do *chip* é separada em 2 partes, chamadas de regiões, uma para cada subcircuito (Figura 7 (b)). A divisão é realizada pensando que cada região deve possuir um espaço disponível maior que a área total

das células do seu subcircuito. Esta operação é repetida recursivamente até que a quantidade de células em cada região seja suficientemente pequena (Figura 7(c)). A divisão do circuito em subcircuitos é tratada como um problema de particionamento de hipergrafos. Neste caso, as células são tratadas como vértices e as conexões como arestas. O problema do particionamento de hipergrafos consiste em agrupar os vértices em  $n$  grupos minimizando o número de conexões entre eles.



**Figura 7.** Ilustração do funcionamento de uma ferramenta de particionamento.

O Fiduccia-Mattheyses (FIDUCCIA; MATTHEYSES, 1982) é uma heurística clássica para particionamento. O Algoritmo II apresenta o pseudocódigo do Fiduccia-Mattheyses para o particionamento em 2 grupos. A entrada é um particionamento inicial, geralmente aleatório. Os passos 3-10 são chamados de “passe”. Dentro de cada passe, o algoritmo computa o vértice de maior ganho (linha 5). O ganho de um vértice é dado pelo número de arestas que cruzam os grupos menos o número de arestas que cruzarão os grupos caso o vértice troque de grupo.

A Figura 8 (a) ilustra um grafo particionado em 2 grupos. O número de conexões que cruzam os grupos é igual a 3. Caso o vértice e mude de grupo, o número de conexões que cruzam os grupos será 2 (Figura 8 (b)). Neste caso, diz-se que o ganho de  $g(v_i) = 3 - 2 = 1$ . Uma vez que o vértice de maior ganho tenha sido encontrado, ele troca de grupo e é travado, ou seja, não é considerado até o próximo “passe”. O ganho desta troca é armazenado (linhas 6-8). Esta operação é repetida  $n$  vezes, onde  $n$  é o número de vértices. No fim de cada passe, é encontrada a sequencia de  $k$  trocas que maximizem o custo total (linha 9). Os  $k$  movimentos são mantidos e os demais desfeitos (linha 10). O Fiduccia-Mattheyses se torna ineficiente para hipergrafos muito grandes. Uma solução é utilizar abordagens hierárquicas. Estas abordagens são compostas por três etapas: etapa de engrossamento, particionamento inicial e etapa de refinamento. Na etapa de brutalização, hipergrafos menores são gerados através do agrupamento de vértices com muitas ligações entre si. Na etapa de particionamento inicial, uma heurística é utilizada para criar grupos iniciais com base no hipergrafo agrupado. Por fim, na etapa de refinamento os agrupamentos são desfeitos e a heurísticas de particionamento são utilizadas para melhorar a solução. O *hMetis* (KARYPIS et al., 1999) é um algoritmo rápido e que gera bons resultados utilizando a estratégia descrita.

---

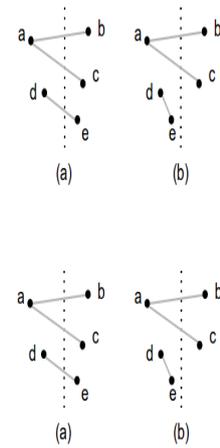
**Algoritmo 2:** Pseudocódigo do *Fiduccia-Mattheyses*.

---

**Entrada:** Particionamento inicial

```
1 begin
2   repeat
3     Destruir todos os vértices;
4     for  $i \leftarrow 0$  to  $|V|$  do
5       Selecionar o vértice  $v_i$  ainda não travado com
        o ganho máximo;
6        $g_i \leftarrow g(v_i)$ ;
7       Mover o vértice  $v_i$  para o outro grupo;
8       Travar o vértice  $v_i$ ;
9     Encontrar  $k$  tal que  $G = \sum_{i=1}^k g_i$ 
10    Tornar os movimentos  $v_1, \dots, v_k$  permanentes e
        desfazer os movimentos  $v_{k+1}, \dots, v_n$ ;
11  until  $G = 0$ ;
```

---



**Figura 8.** Grafo particionado em 2 grupos.

## 2.2 LEGALIZAÇÃO E POSICIONAMENTO DETALHADO

A etapa da legalização recebe como entrada o posicionamento global e tem como objetivo retirar todas as sobreposições alterando solução inicial o mínimo possível. O posicionamento detalhado é a última etapa do fluxo. Esta etapa divide o *chip* em regiões pequenas visando melhorar a solução através de trocas locais. Estas trocas podem ter vários objetivos, por exemplo 1) diminuir o impacto da legalização, em termos de *wirelength*; 2) compensar por imprecisões da função custo adotada no posicionamento global; 3) adicionar novas métricas a função custo, melhorando a solução em termos de roteabilidade, *timing* e/ou dissipação de potência.

O Tetris (HILL, 2002) é um algoritmo clássico de legalização. Primeiramente ele ordena as células de acordo com sua coordenada x em ordem crescente. A seguir, as células são posicionadas individualmente respeitando essa ordenação. Para cada célula, o algoritmo percorre todos os sites de todas as bandas procurando a posição livre no qual o deslocamento da célula é mínimo. É um algoritmo heurístico guloso e, por isso, seu resultado impacta consideravelmente o posicionamento global. O Abacus (SPINDLER; SCHLICHTMANN; JOHANNES, 2008) pode ser visto como uma evolução do Tetris. Ele busca a posição de deslocamento mínimo da célula em cada banda utilizando programação dinâmica, reduzindo em média 30% o movimento das células, com um impacto de apenas 7% no tempo de execução.

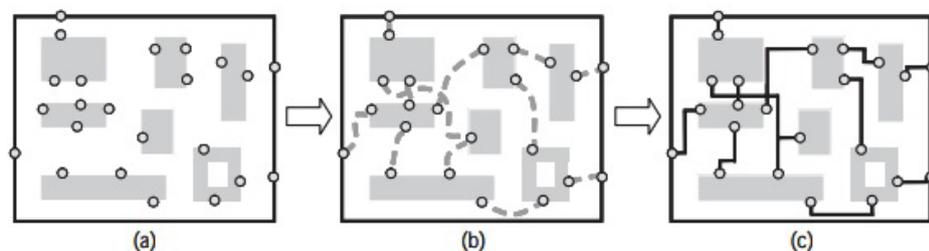
Após legalizada, a solução passa para o posicionamento detalhado. O *simulated annealing* é um algoritmo que pode ser adotado para o posicionamento detalhado desde que a função de perturbação não viole a legalidade do circuito. Embora produza bons resultados, ele não costuma ser adotado devido ao tempo de execução elevado. Uma alternativa é a aplicação de técnicas de *branch-and-bound* (BREUER, 1977;

CALDWELL; KAHNG; MARKOV, 1999) que definem janelas de células vizinhas e as reordenam para achar a melhor combinação.

### 3 | ROTEAMENTO

A etapa de roteamento recebe como entrada o posicionamento, a *netlist* do circuito e tem como objetivo traçar as rotas que os sinais percorrerão. Algoritmos de roteamento visam reduzir métricas como o comprimento dos fios, atrasos de propagação e eletromigração.

Circuitos integrados modernos podem conter centenas de milhões de *redes*(redes) a serem roteadas. Para lidar com a complexidade destes circuitos, a etapa de roteamento é dividida em roteamento global e detalhado. O roteamento global divide o *chip* em uma grade composta por regiões de tamanho regular e determina por quais regiões os fios de cada *net* irão passar. O resultado do roteamento global serve como guia para o roteamento detalhado, que tem por objetivo determinar traçar as interconexões através de fios de metal e vias, seguindo ao máximo os guias do roteamento global. A Figura 9 (a) ilustra um posicionamento, onde os retângulos são células posicionadas e os círculos são os pinos das células. As linhas tracejadas na Figura 9 (b) ilustram os guias do roteamento global. Na Figura 9 (c), o roteamento detalhado é desenhado em linhas contínuas.



**Figura 9.** O problema de roteamento, (a) Resultado recebido de um posicionamento, (b) roteamento global e (c) roteamento detalhado. Retirado de (WANG; CHANG; CHENG, 2009).

De acordo com Johann (JOHANN, 2001), podemos classificar os algoritmos de roteamento em três grupos:

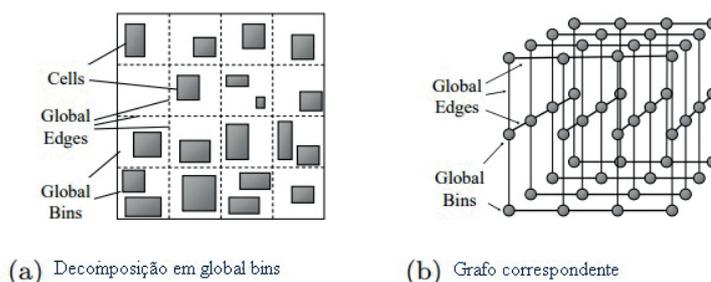
- **Sequenciais:** ordenam as conexões levando em consideração métricas e realizam as conexões uma a uma;
- **Paralelos ou concorrentes:** consideram todas as conexões ao mesmo tempo;
- **Iterativos:** solucionam e depois refazem a solução levando em consideração informações obtidas a cada interação até obter a melhor ou mais próximo da melhor solução.

O maior problema da abordagem sequencial é o critério de ordenação das *redes*. Quando uma ordem é pré-estabelecida, caso bloqueios sejam estabelecidos nas redes iniciais, torna-se mais difícil efetuar o roteamento das *redes* posteriores. Outro problema da abordagem sequencial é que caso uma solução não tenha sido encontrada, não é possível saber ao certo se o motivo foi o de não existir uma solução ou se a ordenação das *redes* foi ruim, além disto, caso uma solução seja encontrada, não sabemos ao certo o quão longe da solução ótima podemos estar (WANG; CHANG; CHENG, 2009). Para estabelecer a ordem com a qual as *redes* ficarão ordenadas, normalmente leva-se em consideração métricas como número de terminais e sua criticidade.

Ao contrário da abordagem sequencial, a abordagem concorrente consegue responder se há uma solução, e o quão perto da solução ótima estamos. Uma possível formulação para o problema do roteamento global utilizando abordagem concorrente é com a formulação do problema como programação linear inteira. O problema de buscar uma solução utilizando esta técnica é que a programação linear inteira é um problema NP-completo, e o tempo gasto ao procurar uma solução limita o tamanho do problema. Dito isto, outras técnicas são utilizadas para encontrar uma solução, tais como relaxar o problema para torná-lo em um problema de programação linear, pois estes podem ser resolvidos em tempo polinomial.

### 3.1 Roteamento Global

O roteamento global é normalmente formulado como um problema de grafos, e resolvido com algoritmos aplicáveis em grafos. Formula-se o problema como um grafo  $G$ , com um conjunto de vértices  $V$  e um conjunto de arestas  $E$ , onde cada vértice  $vi \in V$  corresponde à uma região retangular ou célula do *chip*, e cada aresta  $aij \in E$  corresponde à uma fronteira entre os vértices adjacentes. Além disto, existe um conjunto de *redes*  $N$ , no qual cada *net*  $ni \in E$  é composta por um conjunto de pinos  $pi$ , com cada um destes pinos correspondendo à um vértice  $vi$  (MOFFITT; ROY; MARKOV, 2008). A Figura 10 apresenta um circuito dividido em *Global bins* (retângulos no *chip*), os *edges* (arestas) que ficam entre os *global bins* e as células dentro dos *global bins* (a), e o grafo correspondente (b).



**Figura 10.** Decomposição do *chip* em *global bins*, e o grafo correspondente que formula o problema do roteamento global, adaptado de (MOFFITT; ROY; MARKOV, 2008).

Um roteador global pode normalmente ser dividido em três etapas:

- 1. Definição das regiões:** Todo o espaço de roteamento é dividido em regiões de roteamento. Isto inclui espaço entre os blocos e em cima dos blocos. Entre os blocos existem dois tipos de regiões de roteamento: de canal e *2D-switchboxes*, e acima dos blocos, onde todo o espaço de roteamento é livre, particionando este espaço também em regiões menores chamadas *3D-switchboxes*. Para cada região definida é atribuída uma capacidade, que é a quantidade de *redes* que poderão passar por esta região;
- 2. Assinalamento das regiões:** nesta etapa, são identificadas as sequencias das regiões por onde uma rede irá passar (ser roteada). O congestionamento de cada região deve ser levado em consideração;
- 3. Assinalamento dos pinos:** Depois de concluído o assinalamento das regiões, é assinalado um pino nas fronteiras de cada região, o que permite que as regiões sejam independentes durante o processo de roteamento.

### 3.1.1 MAZE ROUTING

O primeiro algoritmo de busca de caminho entre dois pontos apresentado e publicado foi o *Lee's algorithm*, apresentado em (LEE, 1961). É, possivelmente, o algoritmo mais utilizado para encontrar um caminho entre dois pontos. A técnica utilizada neste algoritmo é pesquisa em largura (BFS). Algoritmos baseados em busca em labirinto utilizam força bruta. Nesta abordagem, são procurados os menores caminhos entre os dois pontos origem e destino, onde são aplicadas duas fases para a busca deste caminho, a dissipação seguida da retração, onde “ondas” são propagadas partindo do nodo origem até encontrarem o nodo destino. Quando o nodo destino é encontrado, acontece então a fase de retração, onde o menor caminho é retornado do nodo destino até o nodo origem. A Figura 11 (a) ilustra as duas fases de propagação e retração, os nodos foram expandidos até atingir a distância 8, e então foi traçado o caminho entre eles (WANG; CHANG; CHENG, 2009).

O algoritmo de Lee também é chamado de **busca cega**, pois ele procura pela rota sem prioridades. Ele garante encontrar um caminho se ele existir, porém possui limitações, tais como a grande demanda de tempo para encontrar uma solução final e a quantidade de memória utilizada no processo. A complexidade de tempo e espaço deste algoritmo é  $O(mn)$ , onde  $m$  e  $n$  são os números das células horizontais e verticais da grade.

### 3.1.2 A\*

O algoritmo A\* foi proposto por (HART; NILSSON; RAPHAEL, 1968). Ele é um

algoritmo de pesquisa em grafos que utiliza uma pesquisa heurística, significando que faz o uso de informações adicionais para tentar descobrir quais os nodos podem levar à uma melhor solução final, e em um tempo otimizado. Também é chamado de **primeiro melhor caminho**, pois a cada interação ele escolhe as rotas que indicam ser o melhor caminho do nodo origem até o nodo destino. A complexidade de tempo e espaço deste algoritmo depende da heurística aplicada.

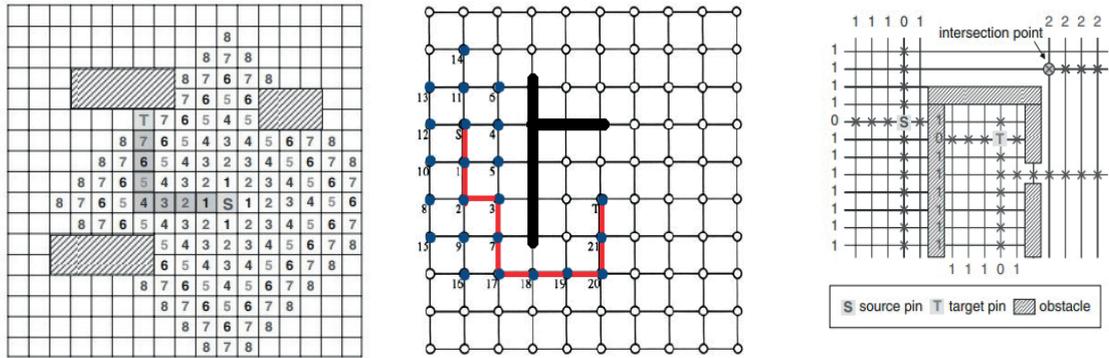
Para isto, o algoritmo utiliza a função  $f(x)=g(x)+b(x)$ , onde  $g(x)$  é o custo do nodo origem até o nodo atual  $x$  enquanto percorrendo o caminho, e  $b(x)$  é a o custo estimado do nodo atual  $x$  até o nodo destino sendo a informação de  $b(x)$  cedida juntamente com o grafo onde a solução será trabalhada, é definida uma função  $f(x)$  para cada nodo  $x$  gerado no caminho. Desta forma, o algoritmo seleciona sempre o nodo com o melhor custo estimado para se propagar, quanto menor o  $f(x)$ , maior as chances de propagação. O algoritmo A\* também pode ser classificado como um algoritmo *maze routing*, porém ele utiliza técnicas diferentes para atingir o objetivo do roteamento. A Figura 11 (b) ilustra o algoritmo A\*, onde os pontos azuis são a propagação a partir do nodo origem, a linha preta grossa indica um obstáculo, e a linha vermelha é o caminho traçado após o encontro dos nodos (SHERWANI, 2012).

### 3.1.3 LINE-SEARCH

Os algoritmos baseados em geometria traçam rotas de uma fonte até um destino, basicamente estes algoritmos percorrem linhas e caso encontrem obstáculos procuram uma saída alternativa e traçam novas linhas até que a linha do nodo origem encontre a linha do nodo destino, traçando assim uma rota a partir do ponto de encontro entre os dois nodos.

A partir dos nodos origem e destino são criados quatro segmentos, dois verticais e dois horizontais, passando pelos nodos que são tidos como nodos base inicialmente. Os segmentos então são expandidos até que encontrem obstáculos ou os limites dos caminhos no espaço utilizado para a solução. Então, cada ponto criado no caminho dos segmentos é interativamente apontado como ponto base, e é gerado um novo segmento através de cada um dos novos pontos base. Isto acontece até que um segmento gerado pelo nodo origem se encontre com um segmento gerado pelo nodo destino. Neste momento, uma conexão é encontrada a partir da intersecção dos dois segmentos. A Figura 11 (c) ilustra o algoritmo *line-search* de (MIKAMI, 1968), onde os nodos se expandem em linhas retas até encontrar obstáculos (WANG; CHANG; CHENG, 2009).

Assim como o algoritmo de Lee, o *line-search* garante encontrar um caminho entre os pontos se ele existir e é melhor se comparados os tempos de execução e memória utilizada, possuindo complexidade de tempo e espaço  $O(L)$ , onde  $L$  é o número de segmento de linhas gerado pelo algoritmo. Como limitações para este algoritmo, o caminho encontrado pode não ser o mais curto ou melhor possível na solução.



A. Maze Router                      B. A\*-search                      C. Line Search:  
**Figura 11.** Algoritmos clássicos de roteamento global. (a) Maze Router (b) A\*-search, (c) *line-search*

### 3.1.4 TÉCNICAS DE ROTEAMENTO GLOBAL

**Roteamento de formas padrão:** Realiza o roteamento de dois pinos em padrões pré-definidos, como em formato de “Z” e “L”. Este método é mais rápido e eficiente se comparado com *maze routing*. Porém, a solução pode não ser ideal se todas as rotas para os dois pinos não forem consideradas (UDGIRKAR; INDUMATHI, 2016). A solução ótima do caminho escolhido não é atingida somente com o uso deste algoritmo. Por este motivo, normalmente são utilizados outros algoritmos em conjunto.

**Roteamento Monotônico:** O roteamento monotônico é empregado em redes de dois pinos, e consiste em expandir do nodo origem sempre em direção ao nodo destino, esta técnica pode ser vista também como uma forma melhorada de *maze routing*, onde a partir da expansão do campo de busca na grade somente dentro do campo entre os dois pontos é possível reduzir o espaço de busca (PAN; CHU, 2007).

**Rip-up and Re-route:** É uma técnica que faz melhorias na solução inicial gerada de forma iterativa. Em resumo, os roteadores globais modernos normalmente consistem em dois maiores passos: um roteamento inicial de todas as redes, com pouco ou nenhum esforço para minimizar o congestionamento, onde é gerado um mapa de congestionamento das redes. O segundo passo é um processo de desfazer e re-rotear, onde o congestionamento é minimizado iterativamente, desfazendo e re-roteando cada rede que passa pelas regiões mais congestionadas, neste processo as redes que passam por estas regiões são desfeitas e re-roteadas até que haja pouco ou nenhum congestionamento no resultado (SIDDIQI; SAIT, 2017). A qualidade da solução gerada com a utilização deste método depende de como as redes foram ordenadas para o processamento do roteamento.

**Roteamento Multi-sink Multi-source:** Um algoritmo de *maze routing* tradicional começa a procurar os caminhos levando em consideração a origem inicial dos pinos origem e destino de uma rede. O algoritmo *multi-sink multi-source* pode ser visto como uma melhoria realizada no *maze routing* original, neste algoritmo as redes multi-pinos são divididas em duas sub-árvores, então, todos os pontos da primeira árvore são tratados como nodos de origem, e todos os pontos da segunda árvore são tratados como pontos de destino, permitindo desta forma que ocorra uma conexão entre os pinos da rede diminuindo o comprimento de fio. Este algoritmo possui o problema

de quando as redes multi-pinos possuírem muitos nodos, aumentar a quantidade de tempo para identificar todos os pontos da grade nas sub-árvores (CHANG; LEE; WANG, 2008).

### 3.2 ROTEAMENTO DETALHADO

Depois de uma rota genérica ser determinada para cada rede durante o roteamento global, o próximo passo é a execução do roteamento detalhado para encontrar as rotas exatas para todas as redes (OZDAL, 2009). Dada uma solução de roteamento global, o roteamento detalhado decide as conexões físicas exatas das redes, incluindo as camadas e conexões entre as mesmas utilizando as vias. O roteador detalhado é encarregado de especificar as rotas de cada conexão, incluindo os seus materiais, furos, posições e dimensões (JOHANN, 2001).

## 4 | CONSIDERAÇÕES FINAIS

Atualmente as técnicas analíticas dominam as ferramentas estado da arte de posicionamento por terem a capacidade de tratar grandes circuitos de forma eficiente. Inicialmente as ferramentas de posicionamento minimizavam somente o comprimento total dos fios (*wirelength*) utilizados para realizar as interconexões entre as células. São os chamados *wirelength-driven placement*. De fato, ainda hoje a maioria dos algoritmos prioriza essa métrica de avaliação. Recentemente, outras métricas ganharam destaque, com o objetivo de melhorar a qualidade das soluções. Minimizar o comprimento dos fios pode gerar áreas muito congestionadas, dificultando a etapa de roteamento e afetando sua qualidade. Pensando nisso, os *routability-driven placement* (LI et al., 2007; SPINDLER; JOHANNES, 2007b; YANG; CHOI; SARRAFZADEH, 2003) reservam espaços vazios em suas soluções para a passagem de fios. Por muito tempo, considerou-se que minimizar o comprimento dos fios indiretamente melhorava o circuito com relação a tempos de propagação. No entanto, isto nem sempre é verdade. Por isto, existem as ferramentas de *timing-driven placement* (HAMADA; CHENG; CHAU, 1993; MARQUARDT; BETZ; ROSE, 2000), para reduzir conexões muito grandes.

Soluções atuais de ferramentas de roteamento adotam as técnicas apresentadas de forma integrada, como o roteador global NTHU-router (CHANG; LEE; WANG, 2008) (LEE; CHANG; WANG, 2010) que aplica *rip-up and reroute* iterativamente, em conjunto com outras técnicas. Outras soluções evitam o uso de *maze routing* em excesso, como o *FastRoute* (PAN; CHU, 2007). Diferente da maioria dos roteadores que fazem o uso da arquitetura Manhattan, o que significa realizar conexão de fios somente na vertical e horizontal, algumas soluções como o *XGRouter* faz o uso da arquitetura X. A arquitetura X é uma arquitetura de fiação baseada no uso difusivo de fios diagonais. De acordo com (LIU, G.; GUO, W.; LI, R.; NIU, Y.; CHEN, G., 2015), a

arquitetura X é totalmente suportada por quase todas tecnologias de produção atuais.

## REFERÊNCIAS

- BROOKS, S.; MORGAN, B. **Optimization using simulated annealing**. The Statistician, JSTOR, p. 241–257, 1995.
- BREUER, M. A. **A class of min-cut placement algorithms**. In: Proceedings of DAC. Piscataway, EUA: IEEE Press, 1977. (DAC '77), p.284–290.
- CALDWELL, A. E.; KAHNG, A. B.; MARKOV, I. L. **Optimal partitioners and end-case placers for standard-cell layout**. In: Proceedings of ISPD. Nova Iorque, EUA: ACM, 1999. (ISPD '99), p. 90–96.
- CHANG, Y.-J.; LEE, Y.-T.; WANG, T.-C. **NTHU-Route 2.0: a fast and stable global router**. In: Proceedings of ICCAD, 2008. [S.l.: s.n.], 2008. p.338–343.
- HART, P. E.; NILSSON, N. J.; RAPHAEL, B. **A formal basis for the heuristic determination of minimum cost paths**. IEEE transactions on Systems Science and Cybernetics, [S.l.], v.4, n.2, p.100–107, 1968.
- FIDUCCIA, C. M.; MATTHEYSES, R. M. A linear-time heuristic for improving network partitions. In: Proceedings of DAC. [S.l.], 1982. p. 175–181.
- HAMADA, T.; CHENG, C.-K.; CHAU, P. M. Prime: a timing-driven placement tool using a piecewise linear resistive network approach. In: Proceedings of DAC. [S.l.], 1993. p. 531–536.
- HENTSCHKE, R. F. Algoritmos para o Posicionamento de Células em Circuitos VLSI. Dissertação (Mestrado em Ciência da Computação) — Instituto de Informática, Universidade Federal do Rio Grande do Sul, Porto Alegre - RS, Brasil, 2002.
- HILL, D. **Method and system for high speed detailed placement of cells within an integrated circuit design**. 2002. US Patent 6,370,673.
- JOHANN, M. **Novos Algoritmos para Roteamento de Circuitos VLSI. Tese (Doutorado em Ciência da Computação)** - Instituto de Informática, Universidade Federal do Rio Grande do Sul, Porto Alegre - RS, Brasil, 2001.
- KARYPIS, G. et al. **Multilevel hypergraph partitioning: applications in VLSI domain**. Transactions on VLSI, v. 7, n. 1, p. 69–79, 1999.
- KIM, M.-C. et al. **Maple: Multilevel adaptive placement for mixed-size designs**. In: Proceedings of ISPD. New York, NY, USA: ACM, 2012. (ISPD '12), p. 193–200.
- LAM, J.; DELOSME, J.-M. Performance of a new annealing schedule. In: IEEE COMPUTER SOCIETY PRESS. Proceedings of DAC. [S.l.], 1988. p. 306–311.
- LEE, C. Y. **An algorithm for path connections and its applications**. IRE transactions on electronic computers, [S.l.], n.3, p.346–365, 1961.
- LEE, Y.-T.; CHANG, Y.-J.; WANG, T.-C. **A temperature-aware global router**. In: Proceedings of VLSI-DAT. [S.l.: s.n.], 2010. p.279–282.
- LI, C. et al. **Routability-driven placement and white space allocation**. Transactions on CAD, IEEE, v. 26, n. 5, p. 858–871, 2007.

- LIU, G.; GUO, W.; LI, R.; NIU, Y.; CHEN, G. **XGRouter: high-quality global router in X-architecture with particle swarm optimization**. *Frontiers of Computer Science*, [S.l.], v.9, n.4, p.576–594, 2015.
- MARQUARDT, A.; BETZ, V.; ROSE, J. **Timing-driven placement for FPGAs**. In: ACM. Proceedings of FPGA. [S.l.], 2000. p. 203–213.
- MIKAMI, K. **A computer program for optimal routing of printed circuit conductors**. IFIP cong.'68, 1968, [S.l.], 1968.
- MOFFITT, M. D.; ROY, J. A.; MARKOV, I. L. **The coming of age of (academic) global routing**. In: Proceedings of ISPD. [S.l.: s.n.], 2008. p.148– 155
- NAYLOR, W. C.; DONELLY, R.; SHA, L. **Non-linear optimization system and method for wire length and delay optimization for an automatic electric circuit placer**. [S.l.]: Google Patents, 2001. US Patent 6,301,693.
- OZDAL, M. M. **Detailed-routing algorithms for dense pin clusters in integrated circuits**. *Transactions on CAD*, [S.l.], v.28, n.3, p.340–349, 2009
- PAN, M.; CHU, C. **FastRoute 2.0: A high-quality and efficient global router**. In: Proceedings of ASP-DAC. [S.l.: s.n.], 2007. p.250–255.
- PLACEUTIL. **Executable Placement Utilities**. 2014. Disponível em: <<http://vlsicad-.eecs.umich.edu/BK/PlaceUtils/>>. Acesso em: 27.12.2014.
- REIS, R. A. d. L. **Concepção de circuitos integrados**. 1a. ed. Porto Alegre, Brasil: Instituto de Informática da UFRGS. Editora Sagra Luzzatto, 2000.
- SHERWANI, N. A. **Algorithms for VLSI physical design automation**. [S.l.]: Springer Science & Business Media, 2012.
- ROY, J. A. et al. **Capo: Robust and scalable open-source min-cut floorplacer**. In: Proceedings of ISPD. Nova Iorque, EUA: ACM, 2005. (ISPD '05), p. 224–226.
- SECHEN, C.; SANGIOVANNI-VINCENTELLI, A. **The timberwolf placement and routing package**. *IEEE Journal of Solid-State Circuits*, v. 20, n. 2, p. 510–522, 1985.
- SIDDIQI, U. F.; SAIT, S. M. **A game theory based post-processing method to enhance VLSI global routers**. *IEEE Access*, [S.l.], v.5, p.1328–1339, 2017.
- SPINDLER, P.; JOHANNES, F. **Kraftwerk: A fast and robust quadratic placer using an exact linear net model**. In: NAM, G.-J.; CONG, J. (Ed.). *Modern Circuit Placement*. [S.l.]: Springer US, 2007, (Series on Integrated Circuits and Systems). p. 59–93.
- SPINDLER, P.; JOHANNES, F. M. **Fast and accurate routing demand estimation for efficient routability-driven placement**. In: Proceedings of DATE. [S.l.], 2007. p. 1226–1231.
- SPINDLER, P.; SCHLICHTMANN, U.; JOHANNES, F. M. **Abacus: Fast legalization of standard cell circuits with minimal movement**. In: Proceedings of ISPD. Nova Iorque, EUA: ACM, 2008. (ISPD '08), p. 47–53.
- UDGIRKAR, G.; INDUMATHI, G. **VLSI global routing algorithms: A survey**. In: Proceedings of INDIACOM. [S.l.: s.n.], 2016. p.2528–2533.
- VYGEN, J. **Algorithms for large-scale flat placement**. In: ACM. Proceedings of DAC. [S.l.], 1997. p. 746–751.

VISWANATHAN, N.; PAN, M.; CHU, C. **Fastplace 3.0: A fast multilevel quadratic placement algorithm with placement congestion control**. In: Proceedings of ASP-DAC. Washington, EUA: IEEE Computer Society, 2007. (ASP-DAC '07), p. 135–140.

WANG, L.-T.; CHANG, Y.-W.; CHENG, K.-T. T. **Electronic design automation: synthesis, verification, and test**. [S.l.]: Morgan Kaufmann, 2009.

YANG, X.; CHOI, B.-K.; SARRAFZADEH, M. Routability-driven white space allocation for fixed-die standard-cell placement. Transactions on CAD, v. 22, n. 4, p. 410–419, 2003.