

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL  
INSTITUTO DE INFORMÁTICA  
PROGRAMA DE PÓS-GRADUAÇÃO EM MICROELETRÔNICA

LEANDRO DE MORAIS NUNES

**Redução de Congestionamento em  
Roteamento Global de Circuitos VLSI**

Dissertação apresentada como requisito parcial  
para a obtenção do grau de  
Mestre em Microeletrônica

Prof. Dr. Ricardo Augusto da Luz Reis  
Orientador

Porto Alegre, dezembro de 2013

## CIP – CATALOGAÇÃO NA PUBLICAÇÃO

Nunes, Leandro de Moraes

Redução de Congestionamento em Roteamento Global de Circuitos VLSI / Leandro de Moraes Nunes. – Porto Alegre: PPGC da UFRGS, 2013.

81 f.: il.

Dissertação (mestrado) – Universidade Federal do Rio Grande do Sul. Programa de Pós-Graduação em Microeletrônica, Porto Alegre, BR-RS, 2013. Orientador: Ricardo Augusto da Luz Reis.

1. Roteamento. 2. Síntese física. 3. Ferramentas CAD. I. Reis, Ricardo Augusto da Luz. II. Título.

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL

Reitor: Prof. Carlos Alexandre Netto

Pró-Reitor de Coordenação Acadêmica: Prof. Rui Vicente Oppermann

Pró-Reitor Pós-Graduação: Prof. Aldo Bolten Lucion

Diretor do Instituto de Informática: Prof. Luís da Cunha Lamb

Coordenador do PGMICRO: Prof. Gilson Inácio Wirth

Bibliotecária-chefe do Instituto de Informática: Beatriz Regina Bastos Haro

## **AGRADECIMENTOS**

A minha esposa Rejane, pelo apoio permanente, incondicional e incentivo durante todo o decorrer dessa jornada.

Ao professor Ricardo Reis, agradeço pelas incontáveis conversas e orientações que ajudaram a definir e nortear esse trabalho, mas acima de tudo pelo voto de confiança para que eu pudesse iniciar e concluir este curso.

Aos amigos e colegas da UFRGS que foram indispensáveis para que este trabalho se tornasse realidade. Um agradecimento especial ao Tiago Reimann e ao Guilherme Flach pela ajuda e pelas conversas esclarecedoras.

*“A vida é como andar de bicicleta. Para manter seu equilíbrio você deve continuar em movimento.”*

— ALBERT EINSTEIN

## Techniques to Reduce Overflow in VLSI Global Routing Phase

### ABSTRACT

Global routing phase is responsible for the interconnect planning and distribution across the circuit area. During the integrated circuit project flow, the global routing is contained in the Physical Synthesis, after the placement, that is when the position of all circuit cells are defined, and before the detailed routing, when the position of all interconnection wires is realized.

A simplified and abstract version of the circuit routing area is used by the global router, that will aggregate in a single vertex, a specific region of the circuit, that represents a bunch of interconnection with their total capacity. This work presents a set of techniques to delimit and threat areas that have high interconnection demand in VLSI circuits.

These techniques are applied in two steps of the global routing flow: the first is executed during the initial routing, where the high interconnection demanding regions are identified. the second step is executed during the iterative routing, where the top offender regions are identified and heva their costs pre-allocated.

In order to evaluate the impact of the proposed techniques, they are implemented in an existing global routing flow, and four metrics are collected: total wirelength, execution time, total overflow and maximum overflow. Tha last two metrics will be different from zero just for the circuits that not have a valid solution.

After the execution of the experiments it was possible to verify a reduction up to 11% in wirelength, in some benchmarks that the literature do no have a valid solution. Furthermore, it was possible to verify a reduction up to 35% in the execution time, when compared to the reference implementation.

Once we are including constraints in form of cost pre-allocation, it is possible to verify an wirelength increase in some cases. In this work, it was possible to observe a small presence of these side-effects, up to 1.39%, according to the executed benchmarks.

**Keywords:** Global Routing, Overflow Reduction, Physical Systesis, Computer Aided Design tools.

## RESUMO

O Roteamento Global é responsável pelo planejamento da distribuição dos meios de interconexão dentro da área do circuito. Dentro da fase do projeto de circuitos conhecida como Síntese Física, essa fase situa-se após a etapa de posicionamento, que define uma posição exata para cada célula do circuito, e antes da etapa de roteamento detalhado que irá definir uma posição para cada meio de interconexão.

Os roteadores globais utilizam uma versão abstrata e simplificada do circuito, que agrega uma região e toda a capacidade de fios que esta região comporta, trabalhando com o planejamento dessas capacidades em relação a demanda de interconexão entre as células do circuito. Este trabalho, apresenta um conjunto de técnicas para delimitação e tratamento de áreas que possuem alta demanda por meios de interconexão em circuitos VLSI.

As técnicas são aplicadas em duas fases do fluxo de roteamento global: a primeira é executada na fase de pré-roteamento, onde são identificadas as regiões que possuem alta demanda por interconexão, isto é, são destino ou origem de um número elevado de fios em relação a sua capacidade de alocar meios de interconexão; a segunda etapa ocorre dentro da fase de roteamento iterativo, identificando e protegendo aquelas regiões que possuem os níveis mais elevados de congestionamento.

Para avaliar os impactos da aplicação das técnicas propostas, foi feita a implementação em um fluxo de roteamento global existente. A avaliação foi feita a partir da extração de quatro métricas de roteamento global comumente utilizadas na literatura de síntese física, para análise de roteamento global: comprimento dos fios, valor total de congestionamento, máximo congestionamento de aresta e tempo de execução.

A partir da execução de experimentos utilizando as técnicas, foi possível verificar ganhos de até 11% em redução do congestionamento total no circuito, em *benchmarks* para os quais ainda não se tem soluções válidas na literatura. Os tempos de execução obtiveram uma redução de até 35%, quando comparados com a implementação usada como referência para aplicação das técnicas, o roteador GR-WL.

Um dos efeitos colaterais da aplicação de técnicas de calibração de custos é o aumento do comprimento médio dos fios. Os resultados dos experimentos mostram que as técnicas propostas conseguem reduzir este efeito colateral para, no máximo, 1.39% de acordo com os *benchmarks* executados.

**Palavras-chave:** Roteamento, síntese física, ferramentas CAD.

## LISTA DE ABREVIATURAS E SIGLAS

CAD	Projeto Auxiliado por Computador ( <i>Computer Aided Design</i> )
CI	Circuito Integrado
EDA	Automação de projeto para eletrônica ( <i>Electronic Design Automation</i> )
GME	Grupo de Microeletrônica
ILP	Programação linear inteira ( <i>Integer Linear Programming</i> )
ISPD	Simpósio Internacional em Síntese Física ( <i>International Symposium on Physical Design</i> )
MCF	Fluxo múltiplo de mercadorias ( <i>Multicommodity Flow</i> )
MST	Árvore de expansão mínima ( <i>Minimum Spanning Tree</i> )
OTC	Roteamento em camadas superiores de metal ( <i>Over-the-cell routing</i> )
FPGA	Field Programmable Gate Array
RRR	Rip-up-and-Reroute
RSMT	Rectilinear Steiner Minimal Tree
UFRGS	Universidade Federal do Rio Grande do Sul
VLSI	Integração em Muito Larga Escala ( <i>Very Large Scale of Integration</i> )

## LISTA DE FIGURAS

Figura 1.1:	Instância básica do problema do posicionamento. Adaptado de (ALPERT et al., 2008). . . . .	14
Figura 1.2:	Roteamento global e roteamento detalhado. Adaptado de (SHERWANI, 1995). . . . .	14
Figura 2.1:	Grade de roteamento global, adaptado de (KASTNER et al., 2002). . . . .	17
Figura 2.2:	Grade de roteamento com <i>preferred direction</i> . . . . .	18
Figura 2.3:	Representação de congestionamento através de mapa de cores. . . . .	19
Figura 2.4:	Conflito entre objetivos de roteamento . . . . .	20
Figura 2.5:	Capacitância de acoplamento . . . . .	20
Figura 3.1:	Impacto do ordenamento em roteamento sequencial. Adaptado de (WANG; CHANG; CHENG, 2008) . . . . .	24
Figura 3.2:	Fases do algoritmo roteamento de labirinto de Lee. . . . .	26
Figura 3.3:	Alternativas de caminho usando algoritmo Lee. . . . .	27
Figura 3.4:	Algoritmo Mikami-Tabuchi, adaptado de (SHERWANI, 1995) . . . . .	28
Figura 3.5:	Algoritmo Hightower, adaptado de (SHERWANI, 1995) . . . . .	29
Figura 3.6:	Exemplos de roteamento através de padrões com 0, 1, e 2 vias. Adaptado de (ALPERT et al., 2008). . . . .	29
Figura 3.7:	Roteamento monotônico, adaptado de (PAN; CHU, 2008). . . . .	30
Figura 3.8:	<i>Rip-up and reroute</i> , adaptado de (HU; SAPATNEKAR, 2002) . . . . .	30
Figura 3.9:	ILP roteamento global . . . . .	31
Figura 3.10:	Roteamento Hierárquico top down. Adaptado de (CONG; XIE; ZHANG, 2005). . . . .	33
Figura 3.11:	Roteamento Hierárquico multi nível. . . . .	33
Figura 4.1:	Anatomia das ferramentas de roteamento . . . . .	36
Figura 5.1:	Cenários envolvendo roteamento de labirinto. Adaptado de (PAN; CHU, 2008). . . . .	41
Figura 5.2:	Roteamento com múltiplas origens e múltiplos destinos. Adaptado de (PAN; CHU, 2008). . . . .	42
Figura 5.3:	Processo de movimentação de arestas do algoritmo MaizeRouter. Adaptado de (MOFFITT, 2008) . . . . .	44
Figura 5.4:	Processo de retração de arestas do algoritmo MaizeRouter. Adaptado de (MOFFITT, 2008) . . . . .	45
Figura 5.5:	Expansão de regiões proibidas do algoritmo NTUgr (HSU; CHEN; CHANG, 2008). . . . .	49



Figura 5.6:	Fluxo de execução do GR. . . . .	53
Figura 6.1:	Máximo congestionamento obtido durante as iterações de roteamento do <i>benchmark newblue5</i> , usando o GR-WL. . . . .	57
Figura 6.2:	Fluxo de execução do GR modificado. . . . .	58
Figura 6.3:	Delimitação inicial de áreas com alta demanda por interconexão. . . . .	59
Figura 6.4:	Atribuição de custo estático em pontos de alta demanda por interconexão . . . . .	61
Figura 6.5:	Atribuição de custo com taxa de variação linear, em pontos de alta demanda por interconexão . . . . .	62
Figura 6.6:	Atribuição de custo com taxa de variação exponencial negativa, em pontos de alta demanda por interconexão . . . . .	63
Figura 6.7:	Expansão de área com alto nível de congestionamento durante o roteamento iterativo. . . . .	65
Figura 6.8:	Mapa de cores representando o congestionamento detalhado no circuito <i>bigblue1</i> e identificação de áreas de alto congestionamento. . . . .	65

## LISTA DE TABELAS

2.1	Circuitos do <i>benchmark</i> ISPD 2007. . . . .	21
2.2	Circuitos do <i>benchmark</i> ISPD 2008. . . . .	22
5.1	Resultados do BoxRouter para o <i>benchmark</i> ISPD 2007 - 3D. . . . .	40
5.2	Resultados do BoxRouter 2.0 para o <i>benchmark</i> ISPD 2007 - 3D. . . . .	40
5.3	Resultados do Archer para o <i>benchmark</i> ISPD 2007 - 3D. . . . .	44
5.4	Resultados do MaizeRouter para o <i>benchmark</i> ISPD 2007 - 3D. . . . .	45
5.5	Resultados do FGR para o <i>benchmark</i> ISPD 2007 - 3D. . . . .	46
5.6	Resultados do NTHU-Route para o <i>benchmark</i> ISPD 2007 - 3D. . . . .	47
5.7	Resultados do GRIP para o <i>benchmark</i> ISPD 2008 . . . . .	50
5.8	Resultados do NCTU-GR para o <i>benchmark</i> ISPD 2008 . . . . .	51
5.9	Resultados do GR-WL para o <i>benchmark</i> ISPD 2008 . . . . .	54
6.1	Resultados da implementação <b>técnicas de delimitação</b> e tratamento de áreas com alta demanda de conexão, no GR-C, com custo de via nulo. A decomposição das redes foi feita usando MST. . . . .	68
6.2	Resultados da implementação <b>técnicas de delimitação</b> e tratamento de áreas com alta demanda de conexão, no GR-C, com custo de via nulo. A decomposição das redes foi feita usando FLUTE. . . . .	69
6.3	Resultados da implementação de referência do GR-C (modelo exponencial), com técnicas de delimitação e tratamento de áreas com alta demanda de conexão, com <b>custo de via nulo</b> . Decomposição das redes feita com MST. . . . .	71
6.4	Resultados da implementação de referência do GR-C (modelo exponencial), com técnicas de delimitação e tratamento de áreas com alta demanda de conexão, com <b>custo de via nulo</b> . Decomposição das redes feita com FLUTE. . . . .	72
6.5	Resultados da implementação de referência do GR-C (modelo exponencial), com técnicas de delimitação e tratamento de áreas com alta demanda de conexão, com <b>custo de via unitário</b> . Decomposição das redes feita com MST. . . . .	73
6.6	Resultados da implementação de referência do GR-C (modelo exponencial), com técnicas de delimitação e tratamento de áreas com alta demanda de conexão, com <b>custo de via unitário</b> . Decomposição das redes feita com FLUTE. . . . .	74

# SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO</b>	13
1.1	Objetivo	15
1.2	Organização Deste Documento	15
<b>2</b>	<b>ROTEAMENTO GLOBAL</b>	16
2.1	Formulação do Problema	16
2.2	Superfície de Roteamento	16
2.2.1	Grade de Roteamento Global	16
2.3	Métricas para Avaliação de Roteamento Global	18
2.3.1	Congestionamento	18
2.3.2	Vias e curvas	19
2.3.3	Comprimento médio dos fios	19
2.3.4	Capacitância de Acoplamento	20
2.3.5	Outras métricas	20
2.4	<i>Benchmarks</i>	21
2.4.1	<i>ISPD 2007 Global Routing Contest</i>	21
2.4.2	<i>ISPD 2008 Global Routing Contest</i>	22
2.5	Conclusão	23
<b>3</b>	<b>ALGORITMOS DE ROTEAMENTO GLOBAL</b>	24
3.1	Classificação dos Algoritmos Roteamento Global	24
3.1.1	Abordagem Sequencial	24
3.1.2	Abordagem Concorrente	25
3.2	Técnicas de Roteamento Global	25
3.2.1	Roteamento de Labirinto ( <i>Maze routing</i> )	25
3.2.2	Algoritmos de Busca em Linha ( <i>Line Searching</i> )	28
3.2.3	Roteamento através de padrões ( <i>Pattern routing</i> )	29
3.2.4	Roteamento Monotônico	29
3.2.5	<i>Rip-up and Reroute</i>	30
3.2.6	Roteamento Baseado em Negociação	31
3.2.7	Programação Linear Inteira 0-1	31
3.2.8	Fluxo Múltiplo de Mercadorias	32
3.2.9	Roteamento Hierárquico	32
3.3	Conclusão	34

<b>4</b>	<b>FLUXO DE EXECUÇÃO DE ROTEAMENTO GLOBAL</b>	<b>35</b>
<b>4.1</b>	<b>Introdução</b>	<b>35</b>
<b>4.2</b>	<b>Anatomia das Ferramentas de Roteamento</b>	<b>35</b>
4.2.1	Preâmbulo	35
4.2.2	Definição das topologias	37
4.2.3	Geração da solução inicial	37
4.2.4	Roteamento Iterativo	38
4.2.5	Finalização	38
<b>4.3</b>	<b>Conclusão</b>	<b>38</b>
<b>5</b>	<b>FERRAMENTAS DE ROTEAMENTO GLOBAL</b>	<b>39</b>
<b>5.1</b>	<b>BoxRouter</b>	<b>39</b>
<b>5.2</b>	<b>FastRoute</b>	<b>40</b>
<b>5.3</b>	<b>Archer</b>	<b>43</b>
<b>5.4</b>	<b>MaizeRouter</b>	<b>44</b>
<b>5.5</b>	<b>FGR e BFG-R</b>	<b>46</b>
<b>5.6</b>	<b>NTHU-Route</b>	<b>46</b>
<b>5.7</b>	<b>SideWinder</b>	<b>48</b>
<b>5.8</b>	<b>NTUgr</b>	<b>48</b>
<b>5.9</b>	<b>GRIP e PGRIP</b>	<b>49</b>
<b>5.10</b>	<b>NCTU-GR</b>	<b>50</b>
<b>5.11</b>	<b>MGR</b>	<b>51</b>
<b>5.12</b>	<b>GR-WL e GR-RT</b>	<b>52</b>
<b>5.13</b>	<b>Conclusão</b>	<b>54</b>
<b>6</b>	<b>IDENTIFICAÇÃO E ATENUAÇÃO DE CONGESTIONAMENTO EM ROTEAMENTO GLOBAL</b>	<b>56</b>
<b>6.1</b>	<b>Introdução</b>	<b>56</b>
<b>6.2</b>	<b>Identificação de Alta Demanda por Interconexão no Roteamento Inicial</b>	<b>57</b>
6.2.1	Modelo Estático	61
6.2.2	Modelo Linear	61
6.2.3	Modelo Exponencial Negativo	62
<b>6.3</b>	<b>Delimitação e Tratamento de Alto Nível de Congestionamento no Roteamento Iterativo</b>	<b>63</b>
<b>6.4</b>	<b>Resultados Experimentais</b>	<b>66</b>
<b>7</b>	<b>CONCLUSÃO</b>	<b>75</b>
	<b>REFERÊNCIAS</b>	<b>77</b>

# 1 INTRODUÇÃO

No contexto social e econômico dos dias atuais, dispositivos eletrônicos exercem um papel fundamental no cotidiano das pessoas. A demanda por novos dispositivos e, conseqüentemente, circuitos integrados cada vez mais complexos é crescente.

Uma simplificação usada para medir a complexidade de um circuito é verificar o número de transistores que ele contém.

A Lei de Moore, definida no início da década de setenta, postula que a cada dois anos, a quantidade de transistores dos circuitos integrados mais modernos do mercado é duplicada. Esse comportamento vem sendo observado e tem se mantido nos últimos quarenta anos.

Projetos modernos de circuitos integrados entregam dispositivos com mais de dois bilhões de transistores. Caso estes projetos dependessem de um processo totalmente manual para criação dos leiautes desses circuitos, o esforço necessário possivelmente inviabilizaria a realização desses projetos.

A necessidade contínua de entregar circuitos complexos em um curto espaço de tempo, representa um desafio para algoritmos e ferramentas de concepção destes componentes.

Na área de projeto de circuitos integrados, a **síntese física** compreende uma série de processos automatizados para criação de leiautes, de forma que um circuito possa ser fisicamente construído a partir de suas descrições funcionais e especificações tecnológicas.

Considerando a complexidade envolvida na síntese física de circuitos, esse processo é dividido em fases que ajudam a tornar o problema tratável. Na literatura, a síntese física é tipicamente apresentada em três fases: particionamento, posicionamento e roteamento.

Na fase de **particionamento**, os componentes do circuito são agrupados em conjuntos, para que possam ser tratados de forma independente, reduzindo assim a complexidade para as outras fases.

A fase de **posicionamento**, conforme ilustrado na figura 1.1, é responsável por atribuir a cada bloco do circuito uma localização fixa, ou seja, cada bloco é colocado em um local que favoreça sua ligação com outros blocos nos quais necessita estar conectado.

Em seguida, a fase de **roteamento** tem o objetivo de definir por onde passam os fios que ligam os componentes, de modo a distribuir os sinais que necessitam ser propagados através das redes de sinais do circuito.

Estima-se que entre 50% a 70% do tempo necessário para propagação dos sinais dentro de um circuito integrado, em tecnologias submicrônicas, seja causado pelos meios de interconexão (CONG, 1996), onde esse percentual aumenta conforme as dimensões de novas tecnologias de fabricação ficam menores (MCPHERSON, 2006).

Além disso, a organização dos meios de interconexão impacta na dissipação térmica

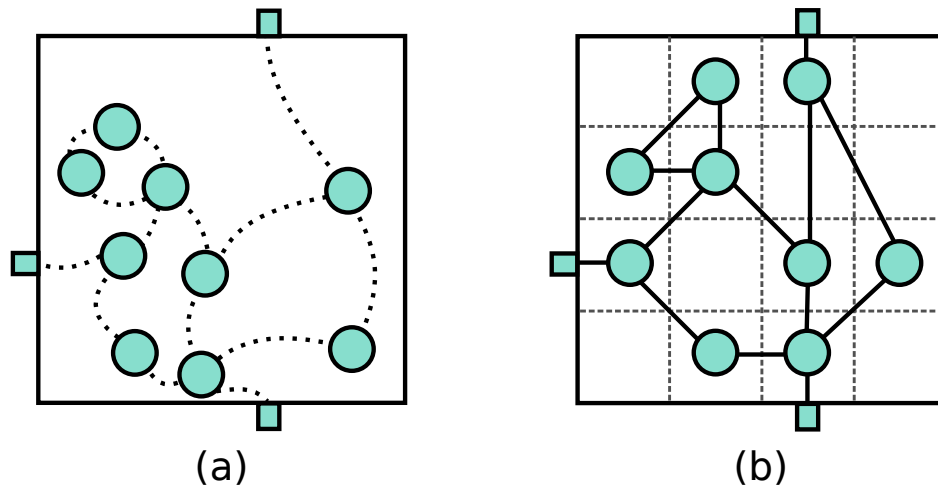


Figura 1.1: Instância básica do problema do posicionamento: (a) nodos sem posição definida e (b) nodos após processo de posicionamento. Adaptado de (ALPERT et al., 2008).

obtida durante o funcionamento do circuito, na confiabilidade do circuito após o processo de fabricação, tendo impacto direto no custo final do dispositivo (MOFFITT et al., 2008).

Devido a criticidade e complexidade do processo de definição dos meios de interconexão, aliada a quantidade elevada de componentes demandados pelos circuitos desenvolvidos atualmente, o roteamento é tipicamente dividido em duas fases: roteamento global e roteamento detalhado.

No **roteamento global**, é feito um planejamento de onde deverão passar os meios de interconexão (figura 1.2(a)), de modo a racionalizar o uso do espaço de roteamento, evitar congestionamentos e reduzir o comprimento médio dos fios.

O **roteamento detalhado** é o responsável por utilizar a saída do roteamento global para definir exatamente por onde cada fio irá passar (figura 1.2(b)), além da localização de cada via, para troca de camada, e espessura dos fios em cada camada do circuito.

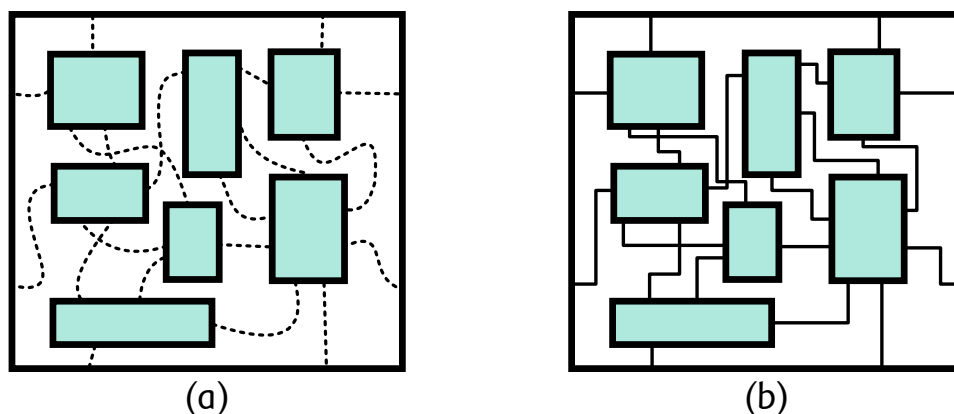


Figura 1.2: Roteamento global (a) e detalhado (b). Adaptado de (SHERWANI, 1995).

A principal contribuição do roteamento global para o roteamento detalhado está em reduzir a área de busca do algoritmo de roteamento detalhado durante a fase de definição do caminho exato de cada fio, o qual se tornaria proibitivo caso fosse executado tendo como área de busca toda a área do circuito.

Além disso, em fluxos de síntese física modernos, o Roteamento global provê estatísticas para as soluções de posicionamento, afim de gerar uma estimativa de roteamento em alto nível para ser usado como retroalimentação para o posicionador. Dessa forma, o tempo de execução do algoritmo de roteamento global é fator determinante para viabilizar a integração entre posicionamento e roteamento global.

Nos últimos anos, um relevante fato que motivou o recente aumento de visibilidade para a pesquisa na área de roteamento global foi a realização de dois desafios para ferramentas de roteamento dentro do ISPD (*International Symposium on Physical Design*) 2007 e 2008, além da disponibilização de arquivos de entrada com problemas atualizados (*benchmarks*) que foram adotados pela comunidade científica para comparação efetiva de resultados obtidos pelas ferramentas (MOFFITT et al., 2008).

## 1.1 Objetivo

O objetivo desta pesquisa foi estudar ferramentas modernas de roteamento global de circuitos integrados, abordando o problema de delimitação e tratamento de áreas onde existe alta demanda por meios de interconexão. Nesse sentido, foram propostas e avaliadas três técnicas de tratamento de áreas com alta demanda por interconexão através de pré-alocação de custos na grade de roteamento.

Para demonstrar a viabilidade de implementação e identificar os impactos do empregos dessas técnicas, elas foram integradas a um fluxo de roteamento global existente, desenvolvida dentro do grupo de pesquisas em microeletrônica da UFRGS.

## 1.2 Organização Deste Documento

Este trabalho está organizado da seguinte forma:

Um levantamento bibliográfico sobre o tema de Roteamento Global é descrito no capítulo 2, onde é apresentada a formulação do problema do roteamento global e dos critérios usados na avaliação de soluções de roteamento. Além disso, são apresentados os *benchmarks* usados para avaliação de roteamento global.

No capítulo 3 são apresentadas os algoritmos clássicos e as técnicas utilizadas pelas ferramentas atuais de roteamento global e detalhado.

As anatomia dos fluxos de execução de roteamento global é apresentado no capítulo 4 onde são descritas as fases de execução das ferramentas que compreendem o estado-da-arte em roteamento global.

No capítulo 5 são apresentados uma série de trabalhos recentes na área de roteamento global, publicados nos últimos cinco anos.

O capítulo 6 detalha as técnicas utilizadas e desenvolvidas no âmbito deste trabalho, bem como sua aplicação em um fluxo de roteamento global existente. No final do capítulo são apresentados os resultados obtidos com a aplicação das técnicas, ante sua aplicação em *benchmarks* adotados pela comunidade científica.

No capítulo 7, por fim, são apresentadas as conclusões e oportunidades de trabalhos futuros geradas a partir deste trabalho.

## 2 ROTEAMENTO GLOBAL

Neste capítulo, será apresentado o problema do roteamento global, algumas métricas usadas para avaliação de roteamento e *benchmarks* utilizados pela comunidade científica para comparação de resultados.

### 2.1 Formulação do Problema

No contexto de síntese física, a fase de roteamento global é responsável por planejar os caminhos por onde irão passar os fios que conectam terminais internos ou externos ao circuito.

A modelagem do problema do roteamento global é feita através de um grafo  $G=(V,E)$ , que possui um conjunto de vértices  $V$  e arestas  $E$ . Cada vértice  $v_i \in V$  corresponde a uma região do circuito. Cada aresta  $e_{ij} \in E$  corresponde a um conjunto de ligações  $m_{ij}$  existente entre dois vértices.

Considerando um conjunto de redes  $N$  que compõem um circuito ou parte dele, tem-se uma lista de redes, onde  $n_i \in N$ . A representação da rede compreende um conjunto de pinos  $P_i$ , onde cada pino está localizado em um vértice  $v_i$ .

O objetivo do roteamento global é fazer com que todos os pinos de cada rede sejam conectados usando as arestas de  $G$ , sem que as capacidades das arestas sejam extrapoladas.

O problema foi inicialmente definido por Burstein e Pelavin (BURSTEIN; PELAVIN, 1983), com o objetivo de simplificar o problema do roteamento detalhado de circuitos.

### 2.2 Superfície de Roteamento

A modelagem da superfície (ou área) de roteamento baseada em roteamento de canais foi utilizada por muitos anos, quando meios de interconexão e regiões ativas do circuito disputavam por um mesmo espaço (SARRAFZADEH; WONG, 1996), tipicamente em tecnologias mais antigas.

Hoje em dia, com o aumento do número de camadas de metal, utiliza-se uma abordagem de roteamento nas camadas superiores àquelas destinadas aos componentes básicos do circuito. Essa abordagem é chamada de roteamento sobre as células, ou OTC (do inglês *over-the-cell routing*).

#### 2.2.1 Grade de Roteamento Global

Em tecnologias modernas de fabricação, conforme o número de camadas de metal aumenta, tornou-se possível destinar algumas dessas camadas exclusivamente para os



meios de interconexão. Nesta abordagem, as camadas inferiores contém células lógicas e componentes ativos do circuito (figura 2.1(a)), enquanto que as camadas superiores são destinadas aos fios que interligam esses componentes.

A área de roteamento não compete por recursos com os demais componentes do circuito. Do ponto de vista do roteamento, global, a área de roteamento é tipicamente modelada por um grafo em formato de grade (figura 2.1(b)), onde cada vértice representa uma áreas retangular do circuito (ALPERT et al., 2008), as quais são chamadas de células globais, ou *g-cells*.

Cada aresta contém um atributo de capacidade, que representa a quantidade de fios que são suportados entre duas áreas de roteamento vizinhas (Figura 2.1(c)).

Considerando as múltiplas camadas que podem existir num circuito, a literatura, muitas vezes, se refere ao grafo que representa as grades de roteamento como sendo uma representação tridimensional de um circuito (MOFFITT et al., 2008), o que pode ser uma ambiguidade com o termo usado para representar o empacotamento de circuitos empilhados, conhecidos como Circuitos 3D (XIE; CONG; SAPATNEKAR, 2010).

Neste trabalho, todas as referências a circuitos tridimensionais se referem à modelagem da grade de roteamento com três dimensões.

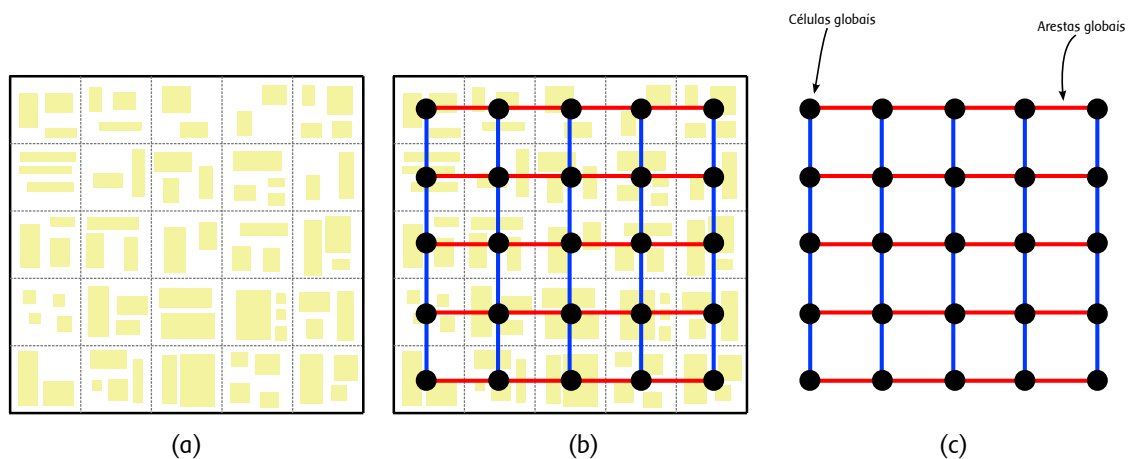


Figura 2.1: Grade de roteamento global, adaptado de (KASTNER et al., 2002).

Na área de roteamento, devido a regras de fabricação, as camadas de roteamento superiores possuem gradualmente, maior largura de fio, o que representa uma redução nas capacidades de roteamento conforme vão se adicionando novas camadas de metal.

É comum que cada camada tenha uma orientação definida: horizontal ou vertical, para racionalizar o uso dos meios de interconexão e facilitar o desenvolvimento de algoritmos que explorem a regularidade da grade.

A primeira camada de metal apresenta, tipicamente, orientação horizontal. A figura 2.2 ilustra esse tipo de organização da grade de roteamento, chamada *direção preferida*.

Apesar a orientação com *direção preferida* ser predominante, alguns trabalhos recentes exploram o uso de orientações variadas na grade de roteamento e de múltiplas orientações em uma mesma camada, tais como arquiteturas-X (TEIG, 2002) e roteamento líquido (PALUSZEWSKI et al., 2004).

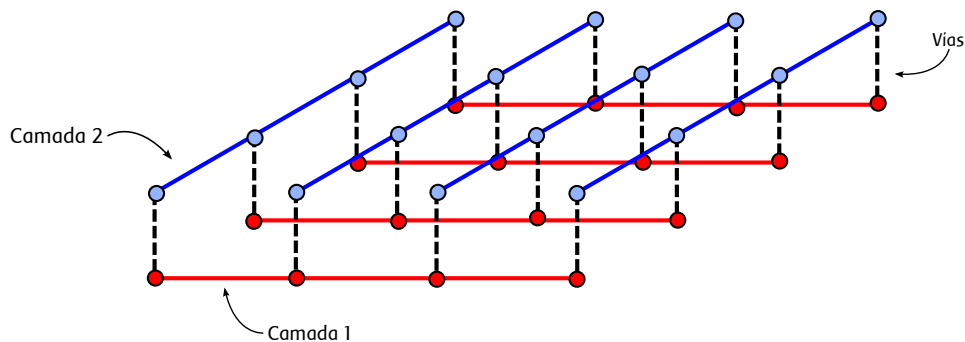


Figura 2.2: Grade de roteamento com *preferred direction*

## 2.3 Métricas para Avaliação de Roteamento Global

O objetivo de uma ferramenta de roteamento global é possibilitar que a localização dos fios de um circuito sejam definidos de forma a viabilizar a interconexão de todos os *terminais* das redes nele contidas. Essa definição é feita com base em fluxos de roteamento, ou seja, elas não representam efetivamente a posição dos fios, mas sim uma abstração que garante que a área de roteamento do circuito é capaz de comportar a quantidade de fios necessária para conectar suas redes.

Algumas métricas são utilizadas para comparar e avaliar as soluções de roteamento. A redução do comprimento médio dos fios, número de vias e congestionamento são consideradas as principais métricas para avaliação de uma ferramenta ou algoritmo de roteamento global (WESTRA et al., 2005).

Entretanto, com o avanço das tecnologias de fabricação, novas métricas vem sendo incorporadas as soluções de roteamento para aumentar a contribuição da solução de roteamento no resultado final do circuito.

As principais métricas adotadas pela literatura são apresentadas nas subseções a seguir.

### 2.3.1 Congestionamento

Na representação abstrata da grade de roteamento global, cada canal de roteamento (aresta do grafo) possui uma capacidade, ou seja, um número máximo de meios de interconexão que são suportados no canal. Quando a quantidade de redes que passam por um canal de roteamento é maior do que sua capacidade, diz-se que ali existe **congestionamento**.

Tipicamente, o roteador global incrementa o custo nas áreas onde há , com o objetivo de "encorajar" os fios a passarem por outros locais onde haja disponibilidade de meios de interconexão.

Embora a redução do congestionamento seja, na maioria dos casos, a prioridade número um de uma ferramenta de roteamento global, é sabido que as ferramentas de roteamento detalhado, tipicamente, conseguem lidar com um nível pequeno de congestionamento (WESTRA et al., 2005) para geração da solução final de roteamento.

Uma forma comumente usada na literatura para demonstrar visualmente o congestionamento em um circuito é através de um gráfico de mapa de cores, onde as regiões com cores quentes representam áreas congestionadas, enquanto que as cores em tons frios são usadas para áreas não congestionadas. Esse formato é utilizado para demonstração de congestionamento no âmbito deste trabalho.

A figura 2.3 apresenta um exemplo desses mapas de congestionamento, gerados no início do fluxo de execução e outro ao final do roteamento.

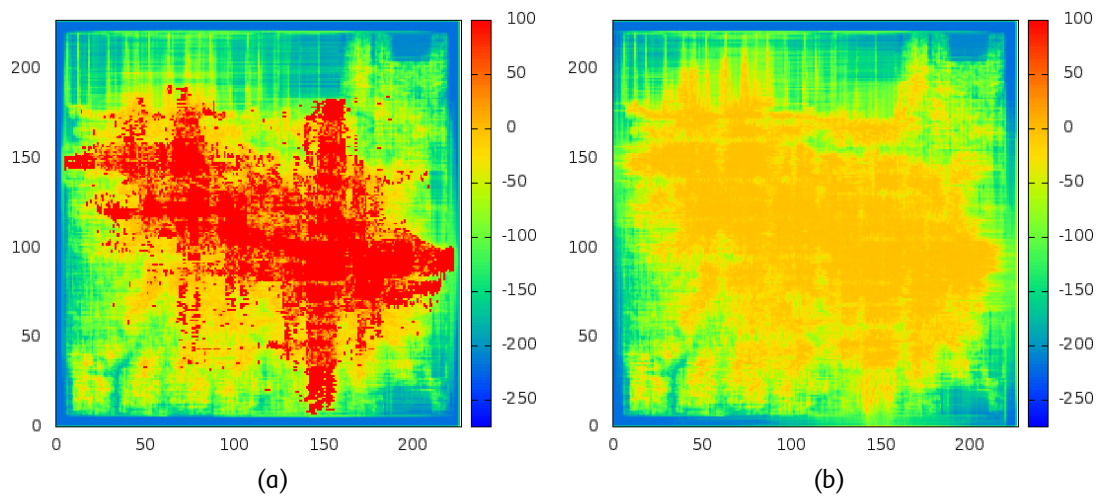


Figura 2.3: Mapa de cores representando o congestionamento em um circuito na área destacada (a) e o circuito sem congestionamento (b).

### 2.3.2 Vias e curvas

Vias são meios de interconexão usados para conectar diferentes camadas de metal que compõem os fios de um circuito.

Considerando a abordagem clássica de roteamento de circuitos, onde cada camada de roteamento permite a realização de fios em um determinado sentido, será necessário empregar uma via para cada troca de sentido do meio de interconexão, dentro da grade de roteamento, conforme ilustra a Figura 2.2.

Um número elevado de vias penaliza o circuito em termos de atraso na propagação de sinais e dissipação térmica, visto que a via possui alta resistência quando comparada a um fio plano. Além disso, vias trazem problemas para o processo de fabricação (ALPERT et al., 2008), em decorrência de falhas na transcrição desses componentes, a partir do leiaute, para a realização no circuito integrado.

Dessa forma, é um objetivo comum das ferramentas de roteamento, favorecer a redução do número de vias na solução de roteamento.

### 2.3.3 Comprimento médio dos fios

O comprimento dos fios tem impacto direto em atraso e consumo de potência do circuito. Embora o objetivo clássico das ferramentas de roteamento seja sempre tentar rotear as redes no caminho ótimo, muitas vezes isso não é possível. Desvios são necessários para evitar zonas congestionadas e bloqueios do circuito.

A métrica de comprimento médio dos fios pode ser conflitante com o objetivo de redução do congestionamento e de número de vias. Entretanto, é necessário fazer a devida calibração do algoritmo para ter-se um meio termo entre comprimento de fios e zonas congestionadas.

A figura 2.4 mostra um exemplo do impacto em optar exclusivamente pelo comprimento mínimo do fio ou pela redução do número de vias. Na primeira parte, o algoritmo opta por reduzir a área de congestionamento do circuito, penalizando o comprimento do fio e inserindo vias a cada troca de sentido. Por outro lado, em (b),

optou-se por reduzir o comprimento do fio e eliminar as vias, contribuindo para aumentar o congestionamento já existente.

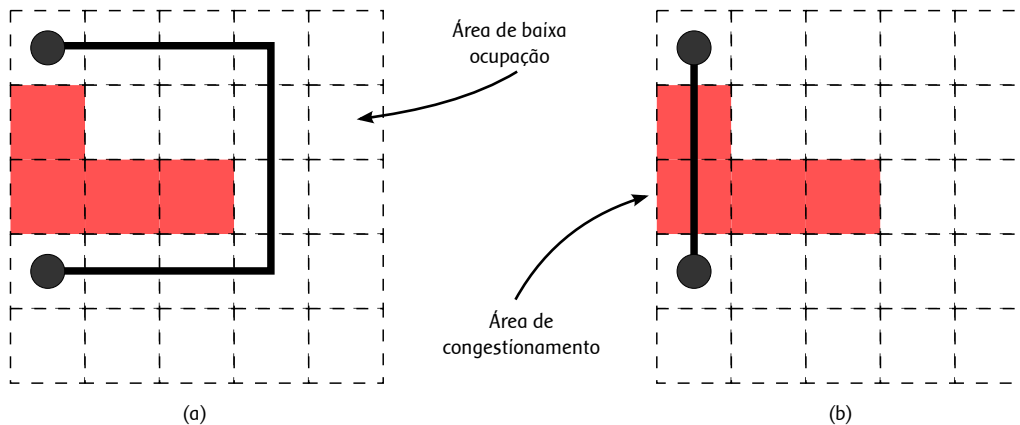


Figura 2.4: Exemplo de conflito entre métricas de roteamento: (a) ênfase em redução de congestionamento e (b) ênfase em redução do comprimento do fio. Adaptado de (ALPERT et al., 2008).

### 2.3.4 Capacitância de Acoplamento

O acoplamento capacitivo (capacitância), decorrente da influência capacitiva parasita entre dois fios muito finos e próximos, os quais podem ocasionar o que comumente chamamos de "ruído".

O trabalho de Zhou e Wong (ZHOU; WONG, 1998) considera técnicas para minimizar o aparecimento de capacitância de acoplamento na fase de roteamento global.

Em (KASTNER et al., 2002), destaca-se que a ordem e o sentido preferencial em que um conjunto de nets é roteado pode favorecer o aparecimento de capacitância de acoplamento.

A figura 2.5 mostra exemplos de possíveis capacitores formados em uma rede de roteamento.

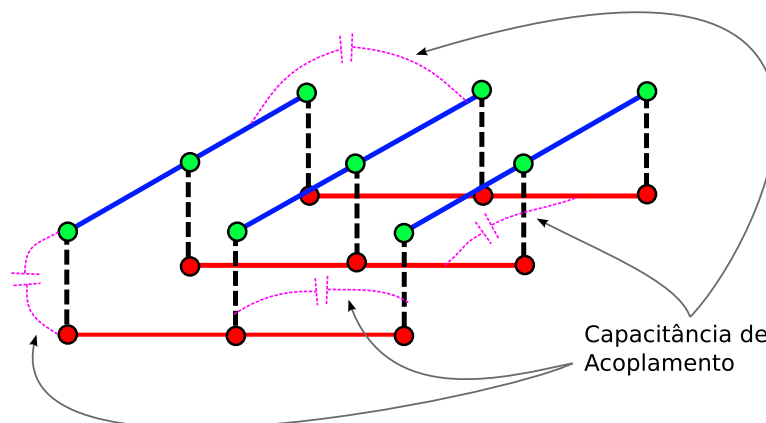


Figura 2.5: Capacitores parasitas que originam a capacitância de acoplamento.

### 2.3.5 Outras métricas

Com o avanço das tecnologias de fabricação e a observância de outros fenômenos decorrentes do processo de redução na dimensão dos componentes dos circuitos (também

conhecido como *scaling*) e aumento da frequência de operação de relógio, novos parâmetros estão sendo incorporados nas fases de síntese física.

O **tempo de execução** da solução de roteamento global aparece como outra métrica importante a ser avaliada (NAM et al., 2008), possibilitando que o roteador seja executado diversas vezes durante o fluxo de concepção do circuito (WESTRA et al., 2005).

## 2.4 Benchmarks

A avaliação de ferramentas e algoritmos de roteamento é uma tarefa complexa, que exige uma análise detalhada sob todas as métricas que forem fornecidas. Um exemplo disso é considerar uma ferramenta que apresente a solução de roteamento em tempo extremamente reduzido, mas que vista sob outra perspectiva, penaliza o circuito com número elevado de vias e mudanças de camada, o que prejudica muito questões de dissipação térmica e atraso do circuito.

A adoção de formatos próprios para entrada e saída de dados é um fator que dificulta ainda mais a comparação de ferramentas de roteamento de maneira precisa.

Para solucionar este problema, a comunidade científica se utiliza de problemas de teste, disponibilizados para serem usados por ferramentas que tem mesma finalidade, a fim de efetuar a comparação de maneira efetiva. A estes problemas de teste, disponibilizados no formato de arquivos e especificações de formato de entrada, dá-se o nome de *benchmarks*.

Em roteamento global, existem conjuntos de *benchmarks* fornecidos em parceria com empresas, que trazem para as ferramentas acadêmicas, os problemas encontrados no dia-a-dia do projeto de circuitos integrados.

Os *benchmarks* mais usados para publicações que compreendem o estado da arte na área de roteamento global são os que foram disponibilizados para os desafios de roteamento do ISPD 2007 e 2008.

### 2.4.1 ISPD 2007 Global Routing Contest

Os *benchmarks* do ISPD 2007 representaram um avanço e atualização em termos tecnológicos desde os circuitos disponibilizados pelo ISPD98.

A grade de roteamento dos circuitos do ISPD 2007 oferece seis camadas de roteamento com capacidades de roteamento sendo reduzidas nas camadas superiores de metal. Além disso, são delimitadas áreas específicas onde a capacidade de roteamento pode ser reduzida para formação obstáculos. A tabela apresenta os circuitos do *benchmarks* ISPD 2007.

Tabela 2.1: Circuitos do *benchmark* ISPD 2007.

Circuito	Posicionador	Dimensões	Camadas (3D)	Nets
adaptec1	Capo	324 x 324	6	176715
adaptec2	mPL6	424 x 424	6	207972
adaptec3	Dragon	774 x 779	6	368494
adaptec4	APlace	774 x 779	6	401060
adaptec5	mFAR	465 x 468	6	548073
newblue1	NTUplace	399 x 399	6	270713
newblue2	FastPlace	557 x 463	6	373790
newblue3	KraftWerk	973 x 1256	6	442005

A obtenção do comprimento do fio, considerando o desafio ISPD 2007 é feito pela equação abaixo (REIMANN, 2011):

$$WL = \sum_{n \in N}^n wl(n) + peso\_via \times via(n)$$

A penalização das vias é bastante severa na avaliação das soluções desse *benchmark*, sendo equivalente a três unidades de fio. A função  $wl(n)$  representa o comprimento de fio e  $via(n)$  o número de vias existentes no fio.

O formato de arquivo de entrada do *benchmark* ISPD 2007 é baseado no padrão utilizado pela ferramenta Labyrinth (KASTNER et al., 2002), com algumas modificações para tornar os circuitos semelhantes aos do ambiente real de projeto de circuitos. Pelas regras do desafio, as redes que possuem mais de mil pinos e aquelas nas quais todos os pinos em uma mesma célula global não necessitam ser roteadas.

O desafio foi proposto em duas categorias: 2D e 3D. Nos circuitos 2D são fornecidas apenas duas camadas com as capacidades somadas, enquanto que na modalidade 3D, são fornecidas as seis camadas disponíveis, com suas orientações e capacidades individuais de roteamento (horizontal e vertical).

Os três primeiros colocados na categoria 2D foram o FGR (ROY; MARKOV, 2008), MaizeRouter (MOFFITT, 2008) e BoxRouter (CHO; PAN, 2006), respectivamente, enquanto que na categoria 3D os mesmo três algoritmos ocuparam as três primeiras posições, alterando-se apenas a ordem, que foi MaizeRouter, BoxRouter e FGR.

#### 2.4.2 ISPD 2008 Global Routing Contest

Os *benchmarks* ISPD 2008 (2008 International symposium on Physical Design) (NAM et al., 2008) foram inicialmente disponibilizados para o concurso de ferramentas de roteamento global, realizado pela conferência de mesmo nome.

As principais mudanças desse concurso de roteamento em relação ao ISPD 2007 foram a retirada dos *benchmarks* bidimensionais, com a utilização exclusiva dos circuitos 3D e a mudança no custo da via alterado, de três unidades de fio, para apenas uma unidade. Um tempo máximo para entrega da solução também foi estipulado 24 horas, sendo que na versão anterior não havia um tempo limite.

Atualmente este *benchmark* é adotado como padrão para avaliação de ferramentas de roteamento que constituem o estado-da-arte da literatura sobre o tema.

Os circuitos propostos pelo desafio são descritos na tabela abaixo.

Tabela 2.2: Circuitos do *benchmark* ISPD 2008.

Circuito	Posicionador	Dimensões	Camadas (3D)	Nets
bigblue1	Capo	227 x 227	6	196885
bigblue2	mPL6	468 x 471	6	428968
bigblue3	APlace	555 x 557	8	665629
bigblue4	FastPlace	403 x 405	8	1133535
newblue4	mPL6	455 x 458	6	531292
newblue5	NUTplace	637 x 640	6	891920
newblue6	mFAR	463 x 464	6	835267
newblue7	KraftWerk	488 x 490	8	1647410

As informações extraídas a partir do resultado da execução de uma ferramenta de roteamento global em um *benchmark* deve conter os seguintes dados:

- **Congestionamento total** (TOF): soma de congestionamento em cada célula global;
- **Congestionamento máximo** (MOF): quantidade de redes que extrapolam a capacidade da célula global mais congestionada;
- **Comprimento total dos fios** (*Total Wirelength*): Soma do comprimento de todos os fios, acrescidos de uma unidade para cada via (troca de camada) utilizada. Esse valor é normalizado ainda pelo tempo de execução da ferramenta em relação a média obtida no concurso;
- **Tempo de execução** (*Runtime*): tempo total em segundos para realização do roteamento.

Todos os participantes do concurso de roteamento global ISPD 2008 são provenientes de projetos acadêmicos (NAM et al., 2008). Os três primeiros colocados foram o NTHU-route (CHANG; LEE; WANG, 2008), NTUgr (HSU; CHEN; CHANG, 2008) e FastRoute 3.0 (ZHANG; XU; CHU, 2008).

## 2.5 Conclusão

Neste capítulo foram apresentados os conceitos básicos de roteamento global, através da formulação do problema e estruturas de dados onde é aplicado, os quais permitiram compreender a área em que este pesquisa se enquadra.

Foram apresentadas ainda, as métricas utilizadas para avaliação das ferramentas de roteamento, através de suas definições. Observa-se que é necessário haver um balanceamento dos algoritmos utilizados pelas ferramentas de roteamento, durante avaliação das métricas em seu fluxo de execução, para que um resultado de roteamento seja aceitável sob as diversas métricas (as vezes conflitantes) de roteamento global apresentadas.

Ao final do capítulo foram apresentados os *benchmarks* adotados atualmente pela comunidade científica, sendo eles, em grande parte, responsáveis pelo aumento recente do interesse de pesquisa na área de roteamento global.

### 3 ALGORITMOS DE ROTEAMENTO GLOBAL

Este capítulo apresenta os algoritmos e estratégias utilizados pelas ferramentas de roteamento global de circuitos integrados.

#### 3.1 Classificação dos Algoritmos Roteamento Global

Os algoritmos de roteamento global podem ser classificadas em sequenciais e concorrentes (SHERWANI, 1995).

##### 3.1.1 Abordagem Sequencial

Na abordagem sequencial as redes são roteadas uma a uma, em seqüência. Desta forma, a estratégia utilizada para ordenação das redes tem impacto direto na solução que será obtida.

Considerando uma área de roteamento bidimensional como mostrado na figura 3.1(a), no caso da rede A ser roteada primeiro (figura 3.1(b)), a rede B necessitará realizar um desvio. Ao contrário da situação mostrada na figura 3.1(c), onde a rede B é roteada primeiro, permitindo que a rede A seja roteada na distância ótima.

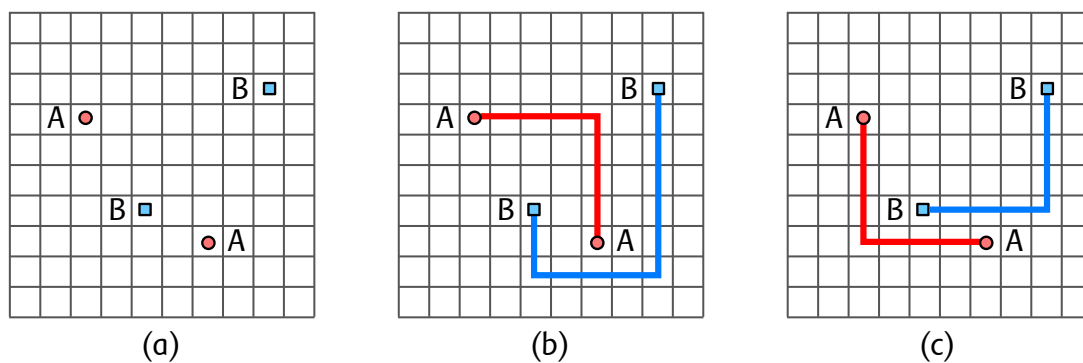


Figura 3.1: Impacto do ordenamento em roteamento sequencial. Adaptado de (WANG; CHANG; CHENG, 2008).

As soluções atuais baseadas em abordagem sequencial utilizam algoritmos heurísticos de ordenação de redes e utilizam *rip-up and reroute* para refinar as soluções. Alguns exemplos de métodos comuns de ordenação de redes, presentes na literatura são:

- Ordenar as redes em **ordem crescente de tamanho** (distância entre os pinos mais extremos da rede) (WANG; CHANG; CHENG, 2008). As redes menores oferecem menor flexibilidade para serem roteadas próximas do ótimo e devem ser roteadas sem desvios excessivos, que congestionam os canais de roteamento; e



- Ordenar as redes com base em informações de **criticidade** ou restrições de tempo (ALPERT et al., 2008).

Em complemento aos métodos de ordenação tradicionais, é possível agregar uma informação de previsão de congestionamento do circuito e priorizar redes que estejam em zonas congestionadas do circuito.

### 3.1.2 Abordagem Concorrente

Os algoritmos de roteamento que utilizam a abordagem sequencial são tipicamente baseados em heurísticas que não apresentam uma garantia teórica sobre a qualidade da solução gerada (ALPERT et al., 2008).

Com o objetivo de reduzir a dependência da solução na ordem em que as redes são roteadas, outra classe de algoritmos tenta encontrar a solução de roteamento de forma concorrente. Nessa modalidade, o algoritmo posterga a avaliação das soluções para o final de cada rodada de roteamento.

Atualmente, a maior parte das ferramentas de roteamento utiliza a abordagem sequencial, baseada em *rip-up and reroute* e dependente de métodos de ordenação de redes. A abordagem concorrente, por outro lado, é dependente da estratégia utilizada para determinar a granularidade e critérios de separação das unidades que serão roteadas de forma concorrente.

Embora alguns resultados de alta qualidade tenham sido obtidos com roteamento concorrente (WU, 2011), o tempo de execução proibitivamente alto, tem sido o fator que reduz o uso de roteamento concorrente quando comparado a abordagem sequencial (ROY; MARKOV, 2008).

Alguns trabalhos recentes que utilizam concorrência em seus fluxos são o SideWinder (HU; ROY; MARKOV, 2008), NCTU-GR (LIU et al., 2010) e o PGRIP (WU; DAVOODI; LINDEROTH, 2010).

## 3.2 Técnicas de Roteamento Global

Dentro da abordagem utilizada por uma ferramenta de roteamento global, diversos algoritmos podem ser usados, em diferentes momentos do fluxo das ferramentas.

Alguns algoritmos se destacam pela rapidez de execução, para um cenário onde não existam muitas limitações e restrições no espaço de roteamento. Outras soluções objetivam um resultado de alta qualidade, em termos de comprimento de fio, número de desvios, embora apresentem tempo de execução mais elevado, ou necessitem de maior quantidade de memória para armazenar seus dados durante a execução.

Esta seção visa apresentar algumas categorias de algoritmos de roteamento que vem sendo utilizadas nas ferramentas acadêmicas ao longo dos últimos anos, ou que tem importância do ponto de vista histórico da evolução da tecnologia de roteamento.

### 3.2.1 Roteamento de Labirinto (*Maze routing*)

Os algoritmos para roteamento de labirintos são estudados desde a década de sessenta, em conexão com o problema de encontrar o menor caminho em um domínio geométrico.

A ideia básica por trás dos roteadores de labirinto, utiliza o algoritmo de Dijkstra (DIJKSTRA, 1959) para encontrar um caminho numa grade.

Considerando uma grade de roteamento, na qual deseja-se encontrar o caminho entre dois pontos, a ideia é definir um dos pontos como sendo a origem e o outro como sendo

o destino. De maneira geral, os roteadores de labirinto atuam em duas fases: **expansão e retro propagação**.

Na fase de **expansão**, alguns nodos da grade são mapeados e marcados, até que se atinja o nodo destino (SHERWANI, 1995).

Uma vez que o nodo destino seja encontrado, a fase de **retro propagação** utiliza os nodos descobertos para definir qual o caminho mais conveniente, seguindo as políticas de cada implementação, a ser utilizado para ligar os dois nodos. Alguns exemplos dessas políticas podem ser: redução do número de desvios; preferência por caminhos menos ocupados por outras redes, ou até mesmo a priorização dos caminhos mais próximos de outros fios da mesma rede.

O algoritmo de Lee (LEE, 1961), Hadlock (HADLOCK, 1977) e o A\* (HART; NILSSON; RAPHAEL, 1968) são exemplos de algoritmos para busca em labirinto que utilizam a ideia base de Dijkstra baseada em expansão e retro propagação.

### 3.2.1.1 Algoritmo de Lee

O algoritmo Lee, parte do nodo de origem e rotula os nodos adjacentes com números, de forma incremental, até que o nodo destino seja alcançado. O número sequencial representa a distância de um determinado nodo a partir da origem (LEE, 1961).

A figura 3.2 ilustra os passos do algoritmo de Lee, que garante encontrar um caminho entre origem e destino, caso ele exista.

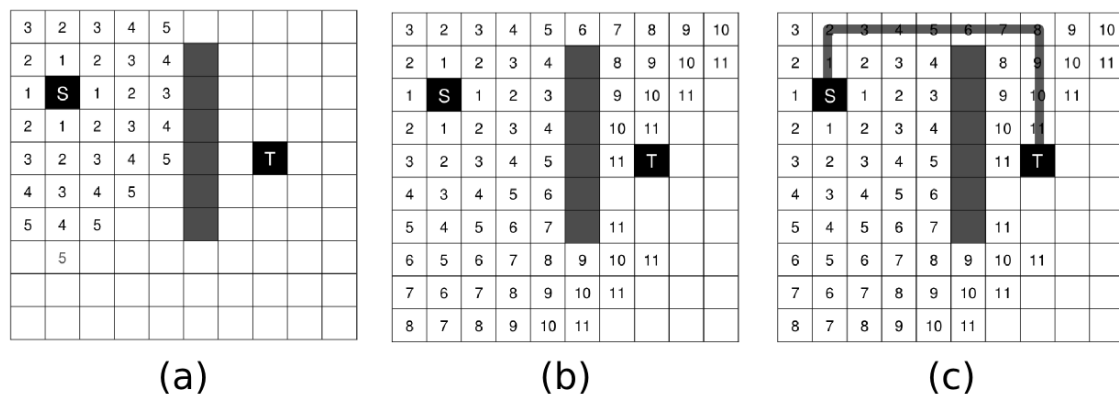


Figura 3.2: Fases do algoritmo de roteamento de labirinto de Lee: (a) expansão dos nodos, (b) localização do destino e (c) marcação do caminho.

É importante destacar que na fase de retro propagação, conforme mostrado na Figura 3.2(c), existem algumas opções que podem ser exploradas para atenuar o problema do congestionamento ou reduzir o número de vias, por exemplo. Por outro lado, algumas soluções podem contribuir com o aumento de vias, sem aumentar o tamanho do fio. A Figura 3.3 apresenta algumas dessas possibilidades.

Observando a solução do algoritmo Lee, no exemplo apresentado na figura 3.2, é possível verificar que grande parte dos nodos da grade foram visitados, em uma situação onde destino e origem estão relativamente próximos.

A expansão de nodos é feita de maneira simétrica, o que faz com que nodos mais distantes do objetivo sejam expandidos na mesma proporção que os nodos que estão no sentido nodo de destino (SHERWANI, 1995). Isso representa um problema para grades de tamanho elevado, em termos de requisitos de memória e tempo de execução.

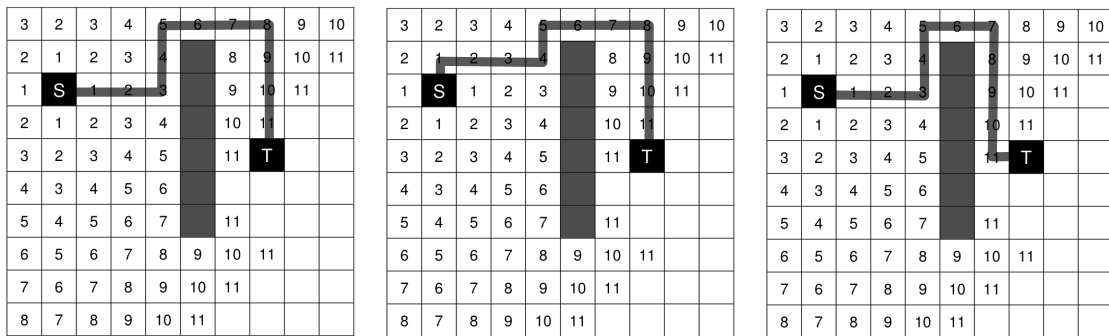


Figura 3.3: Alternativas de caminho usando algoritmo Lee.

Com base nas deficiências encontradas no algoritmo Lee, a ideia base apresentada neste algoritmo foi estendida por diversos novos algoritmos como o de Soukup, Hadlock (HADLOCK, 1977), A\* (HART; NILSSON; RAPHAEL, 1968), LCS\* (JOHANN, 2000), entre outros.

Essas heurísticas que visam reduzir o número de nodos expandidos para obtenção da resposta, tendo impacto também na redução do tempo de execução e memória consumida.

### 3.2.1.2 Algoritmo de Hadlock

O algoritmo de Hadlock (HADLOCK, 1977) é uma variação do algoritmo de Lee que reduz o tempo de execução durante a fase de expansão de nodos, através da seleção dos nodos com menor desvio estimado do caminho ótimo.

A marcação dos nodos é feita com base no desvio do caminho delimitado pelo retângulo que compreende o nodo de origem e o nodo de destino dentro da grade.

Quando um obstáculo é encontrado é necessário expandir nodos que possuam número de desvio maior, mas considerando que nesse caso, os nodos expandidos estão em ordem crescente de desvio do caminho ótimo, este algoritmo garante que, caso exista, o caminho entre os nodos será encontrado.

### 3.2.1.3 A\*

O algoritmo A\* (HART; NILSSON; RAPHAEL, 1968) é uma modificação simples do algoritmo de Dijkstra que reduz de forma significativa seu tempo de execução. Essa modificação consiste em priorizar o próximo nodo a ser expandido através de uma função de estimativa de alcance do destino.

A função de custo clássica do A\* é representada por  $f(x) = g(x) + h(x)$ , onde  $g(x)$  representa a distância da origem até o nodo atual e  $h(x)$  a distância estimada deste nodo até o destino, tipicamente, num caminho ortogonal, podendo variar de acordo com a função alvo da aplicação.

Através do ordenamento dos nodos colocados na fila de expansão, os nodos que retornarem menores valores em  $f(x)$  serão expandidos antes. Por essa lógica, os nodos mais distantes do destino serão colocados no fim da fila, e os nodos mais próximos, ou de menor custo, serão expandidos mais rapidamente.

A fase de retro propagação ocorre de maneira idêntica ao algoritmo de Lee.

### 3.2.2 Algoritmos de Busca em Linha (*Line Searching*)

A principal ideia por trás dos algoritmos de busca em linha é a de representar o espaço de roteamento através de um conjunto de linhas, ao invés da abordagem de matriz de pontos, usada pelos roteadores de labirinto.

Essa mudança de abordagem de mapeamento do espaço faz com que os algoritmos de busca em linha execute, em determinadas situações, em menos tempo e ocupando menos memória.

Os algoritmos clássicos de busca em linha são os propostos por **Mikami-Tabuchi** (MIKAMI; TABUCHI, 1968) e o algoritmo **Hightower** (HIGHTOWER, 1969).

É importante destacar que os algoritmos de busca em linha são indicados para situações onde exista pouco congestionamento e baixo número de zonas de bloqueio. Caso contrário, esses algoritmos podem requerer mais memória e rodar em tempo de execução igual ou pior do que os roteadores de labirinto.

#### 3.2.2.1 Algoritmo Mikami-Tabuchi

O Algoritmo Mikami-Tabuchi (MIKAMI; TABUCHI, 1968) começa traçando uma linha horizontal e vertical a partir dos pontos de origem e destino.

As linhas são interrompidas sempre que chegam na borda da área de roteamento ou em alguma zona de bloqueio, conforme ilustra a figura 3.4.

O processo de expansão de linhas continua até que alguma linha expandida pelo nodo de origem, toque alguma linha expandida pelo nodo de destino.

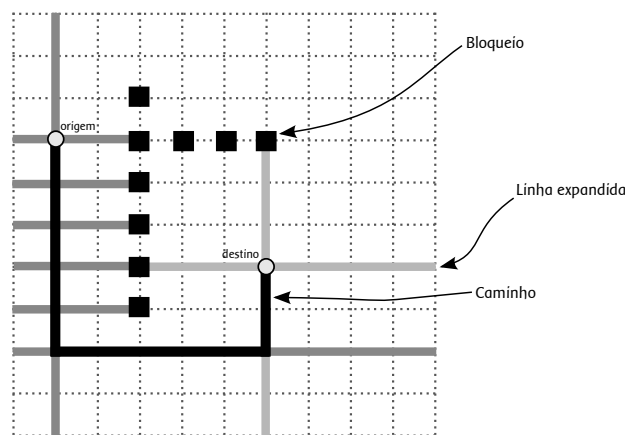


Figura 3.4: Algoritmo Mikami-Tabuchi, adaptado de (SHERWANI, 1995)

As intersecções criadas durante o processo de expansão recebem rótulos sequenciais para orientar o processo de retro propagação (ou *backtracking*), no qual podem ser realizadas otimizações para reduzir o número de vias do caminho.

#### 3.2.2.2 Algoritmo Hightower

Este algoritmo é bastante semelhante ao Mikami-Tabuchi (MIKAMI; TABUCHI, 1968). Seu ponto de melhoria acontece na fase de expansão das linhas, onde o Hightower (HIGHTOWER, 1969) é mais seletivo na escolha dos candidatos a expansão, conforme ilustra a figura 3.5.

A heurística se baseia em escolher para expansão, apenas aqueles nodos que estão livres de qualquer tipo de bloqueio.

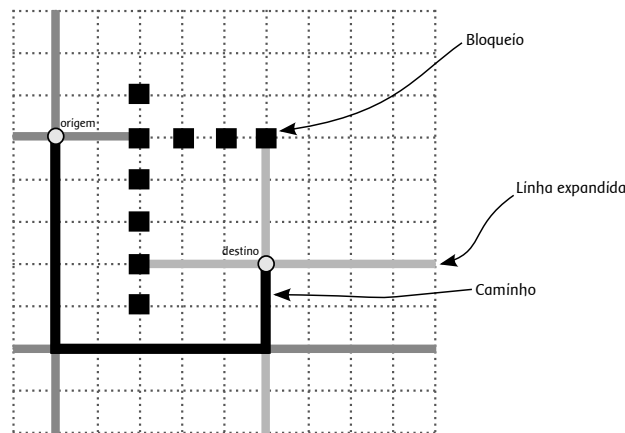


Figura 3.5: Algoritmo Hightower, adaptado de (SHERWANI, 1995)

Uma vez que o espaço de busca pode não ser totalmente explorado, o algoritmo Hightower não dá garantias de encontrar um caminho, mesmo que ele exista.

### 3.2.3 Roteamento através de padrões (*Pattern routing*)

A representação da solução de roteamento através de linhas, ao invés de caminhos completamente mapeados, pode trazer uma economia de memória e redução do tempo de execução.

Quando se aplica roteamento baseado em padrões (*pattern routing*) restringe ainda mais o espaço através de rotas pré-definidas, sendo elas representadas pelas letras I, Z, U e L.

A liberdade de rotas pré-determinadas varia em função do número máximo de vias permitidas (KASTNER et al., 2002). A figura 3.6 apresenta alguns exemplos de padrões.

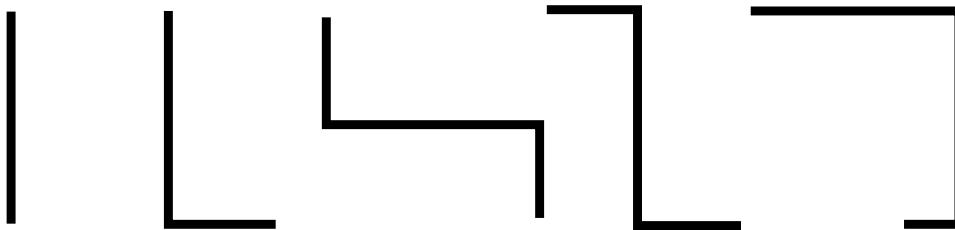


Figura 3.6: Exemplos de padrões com 0, 1, e 2 vias. Adaptado de (ALPERT et al., 2008).

Estima-se que cerca de metade do tempo total para execução de uma ferramenta de roteamento seja em roteamento de labirinto.

O uso de roteamento baseado em padrões é citado como uma abordagem efetiva para redes que necessitam poucas vias, cabendo ao algoritmo de roteamento de labirinto apenas a solução das redes que necessitem um número elevado de vias (ALPERT et al., 2008).

### 3.2.4 Roteamento Monotônico

O roteamento monotônico é uma simplificação implementada no roteador de labirinto que objetiva reduzir o número de nodos expandidos durante o roteamento.

Nesta abordagem, a expansão ocorre em menos dimensões do que todas as permitidas, semelhante ao ilustrado pela figura 3.7.

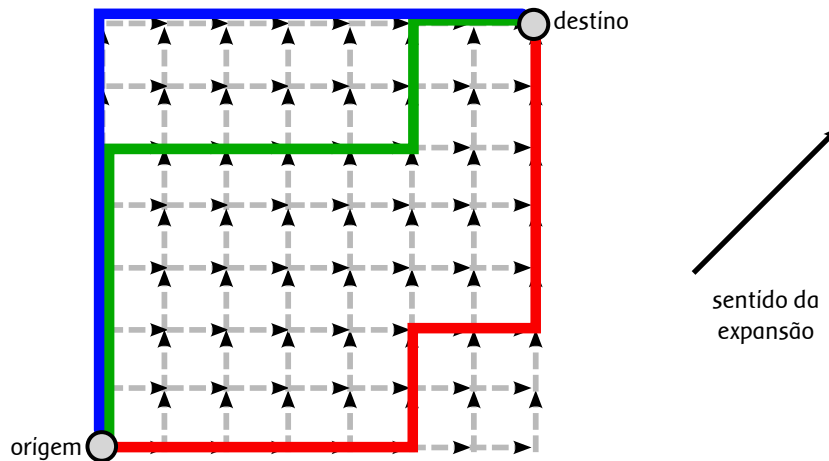


Figura 3.7: Roteamento monotônico, adaptado de (PAN; CHU, 2008).

Esse procedimento é aplicado por diversas soluções de roteamento global, tipicamente nas fases iniciais, quando ainda há pouco congestionamento.

### 3.2.5 *Rip-up and Reroute*

*Rip-up and reroute* é uma técnica iterativa que consiste em remover algumas conexões existentes e trocar por novas opções de roteamento.

A técnica, largamente utilizada em ferramentas de roteamento comerciais, é de implementação simples e se adapta a muitas situações do contexto de síntese física, desde a geração até o refinamento de soluções de roteamento (ALPERT et al., 2008).

O roteamento iterativo é feito através do roteamento das redes de forma independente, ou seja, sem considerar o impacto de outras redes e com os algoritmos de roteamento que for mais conveniente para a solução planejada.

A figura 3.8 ilustra a execução dessa técnica em uma grade de exemplo. Após a primeira execução do algoritmo de roteamento foi identificada uma área em que ocorreu congestionamento. Na segunda execução, as redes que passavam pela área congestionada, optaram por caminhos alternativos.

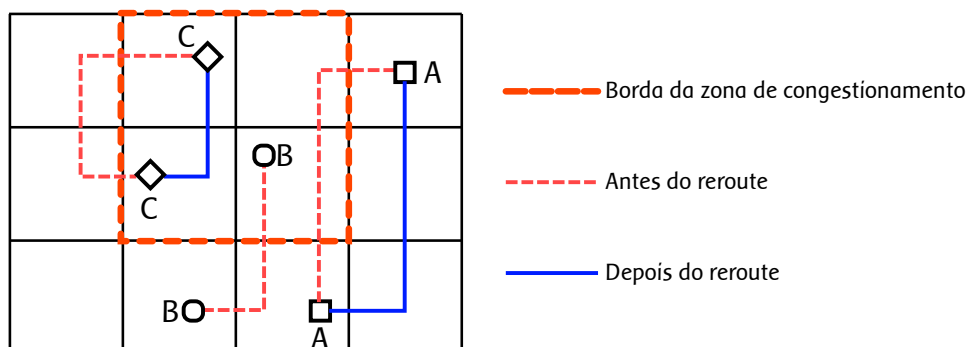


Figura 3.8: *Rip-up and reroute*, adaptado de (HU; SAPATNEKAR, 2002)

A partir da etapa de avaliação da solução, as redes que estão em zonas congestionadas são removidas e roteadas novamente. Essa remoção pode ser total ou parcial, com a criação de pontos auxiliares de roteamento, se necessário.

Essa independência existente entre as unidades de roteamento faz com que *rip-up*

and reroute seja aplicado também em soluções de utilizam concorrência para realizar o roteamento (DICKSON, 2007).

### 3.2.6 Roteamento Baseado em Negociação

Um problema das soluções baseadas em *rip-up and reroute* é a oscilação entre duas ou mais regiões do circuito (ALPERT et al., 2008). O problema ocorre quando os custos de uma célula  $g_1$  são ajustados de forma que um conjunto de redes, na próxima iteração será roteado em outra célula global  $g_2$ , gerando um congestionamento nesta última. Na próxima iteração, o custo de  $g_2$  é ajustado, fazendo com que as redes passem novamente por  $g_1$ .

Para evitar o aparecimento deste problema, alguns algoritmos consideram um histórico de congestionamento como variável da função de cálculo de custo no processo de roteamento (ALPERT et al., 2008).

O algoritmo PathFinder é um representante dessa classe de algoritmos (MCMURCHIE; EBELING, 1995). Originalmente desenvolvido para roteamento de FPGA e estendido para diversas outras categorias de problemas como placas de circuito impresso.

A técnica de negociação ocorre a partir do roteamento das redes individualmente, sem considerar o impacto de outras redes.

Ao término de cada rodada de roteamento, a solução global é avaliada com o objetivo de calibrar os custos da grade para uma nova rodada de roteamento. Áreas que apresentam congestionamento tem seu custo aumentado, estimulando as redes a encontrarem caminhos alternativos.

### 3.2.7 Programação Linear Inteira 0-1

A programação linear inteira (ILP) 0-1 modela o problema do roteamento global num grafo  $G = (V, E)$  onde cada nodo representa uma célula global. Cada aresta  $e \in E$  representa uma ligação entre dois nodos e possui uma capacidade  $c_e$ , que representa o número máximo de canais de roteamento disponíveis naquela aresta.

Considerando uma rede  $n_i \in N$ , todos os seus possíveis roteamentos podem ser enumerados e mapeados para uma estrutura, onde cada possibilidade  $r_i \in R$  representa uma possível solução para essa rede.

A variável  $x_{i,j} \in (0, 1)$  indica quando um padrão de roteamento  $r_{i,j}$  foi selecionado para ser a solução de roteamento para a rede  $n_i$ . A figura 3.9 demonstra o problema do roteamento global em um grafo de roteamento  $G = (V, E)$  com uma lista de redes  $N$  modelado em programação linear inteira 0-1.

$$\begin{array}{ll}
 \text{Minimizar} & \lambda \\
 \text{Sujeito a} & \sum_{r_{i,j} \in R_i} x_{i,j} = 1, \quad \forall n_i \in N \\
 & x_{i,j} \in \{0, 1\}, \quad \forall n_i \in N, \forall r_{i,j} \in R_i \\
 & \sum_{i,j:e \in r_{i,j}} x_{i,j} \leq \lambda c_e, \quad \forall e \in E
 \end{array}$$

Figura 3.9: ILP roteamento global

As duas primeiras restrições garantem que apenas uma solução será encontrada para cada rede e a terceira faz com que o congestionamento seja minimizado.

Se uma solução com  $\lambda \leq 1$  existir, significa que uma solução para o roteamento global sem congestionamento pode ser atingida.

Considerando que a ILP 0-1 é um problema NP-completo, a aplicação dessa técnica no mundo real, para um circuito completo, pode deixar o tempo necessário para o retorno de uma solução se torne impraticável (WANG; CHANG; CHENG, 2008), pois o tempo necessário para formular a saída cresce de maneira exponencial em função das entradas (SHERWANI, 1995).

Nas soluções vistas na literatura, a aplicação de ILP 0-1 está como sendo uma parte de um fluxo maior, onde o que é enviado para solução com ILP são problemas menores e tratáveis, para que o tempo de resposta seja aceitável (CHO; PAN, 2006).

### 3.2.8 Fluxo Múltiplo de Mercadorias

A modelagem do problema do roteamento global (*Multicommodity Flow*, ou MCF) considera que cada meio de interconexão de uma rede (neste caso, a *commodity*) deve ser transportado de sua origem até seu destino, respeitando a capacidade de cada nodo da rede (representados pelas arestas existentes entre as células globais).

Existem duas modelagens do problema: a que permite valores fracionários de fluxo e a que apenas permite valores inteiros. Enquanto a abordagem que permite valores fracionários pode ser solucionada em tempo polinomial, a que compreende apenas valores inteiros é um problema NP-completo (ALPERT et al., 2008).

A abordagem em fluxo múltiplo de mercadorias é feita tanto para redes de dois (SHRAGOWITZ et al., 1987) ou três (RAGHAVAN et al., 1991) pinos. O trabalho de Albrecht (ALBRECHT, 2000) aplica fluxo múltiplo de mercadorias em conjunto com ILP 0-1 para selecionar, dentre um conjunto de redes  $T$ , a rede específica  $t_i \in T$  que minimize o congestionamento do circuito roteado.

### 3.2.9 Roteamento Hierárquico

A utilização de abordagens hierárquica é uma alternativa para dividir o problema do roteamento global é o uso de roteamento hierárquico.

A abordagem do roteamento hierárquico utiliza uma combinação de duas estratégias básicas, conforme o nível de detalhe da grade de roteamento: quando a grade varia de uma versão menos detalhada para uma mais detalhada chamamos de variação (*top-down*); quando a variação é de uma versão mais detalhada para uma menos detalhada (*bottom-up*). A quantidade de versões da grade é variável, são os chamados *níveis*.

No modelo *top-down*, a cada mudança de nível é possível ver mais detalhes sobre a área de roteamento. O traçado do roteamento pode então ser refinado e considerar mais detalhes do seu caminho, conforme ilustrado na figura 3.10.

No modelo *bottom-up*, a construção da solução de roteamento começa a partir das unidades de células globais originais (figura 3.10(c)), a partir das quais serão feitas versões menos detalhadas da grade.

Com essas unidades básicas é possível simplificar a abordagem de roteamento, traçando soluções em mais alto nível com um tempo de execução reduzido. A partir dessas soluções é possível e realizar sucessivas rodadas de roteamento para terminar a solução de roteamento com nível de detalhamento original da grade.

Vários trabalhos fazem combinações das duas abordagens básicas, como o modelo hierárquico híbrido e o roteamento multi nível.

O **modelo hierárquico híbrido** objetiva atenuar o problema da propagação de erros no decorrer das mudanças de nível no roteamento hierárquico. O fluxo de execução



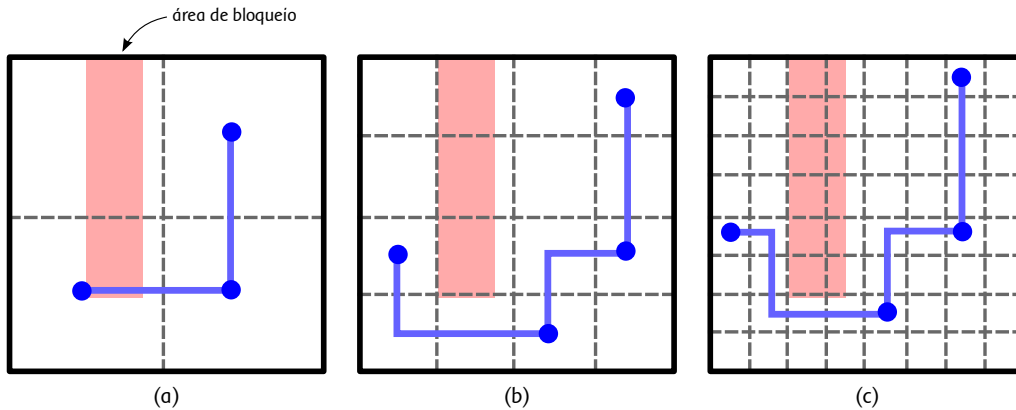


Figura 3.10: Roteamento Hierárquico *top-down*: (a) nível 3, (b) nível 2 e (c) nível 1. Adaptado de (CONG; XIE; ZHANG, 2005).

consiste em executar o modelo *bottom-up* dentro de cada rodada *top-down* (REIMANN, 2011).

Na abordagem de **roteamento hierárquico multi nível**, ocorre um processo *bottom-up*, seguido de um processo *top-down*. A figura 3.11 ilustra um exemplo do processo de roteamento hierárquico multi nível.

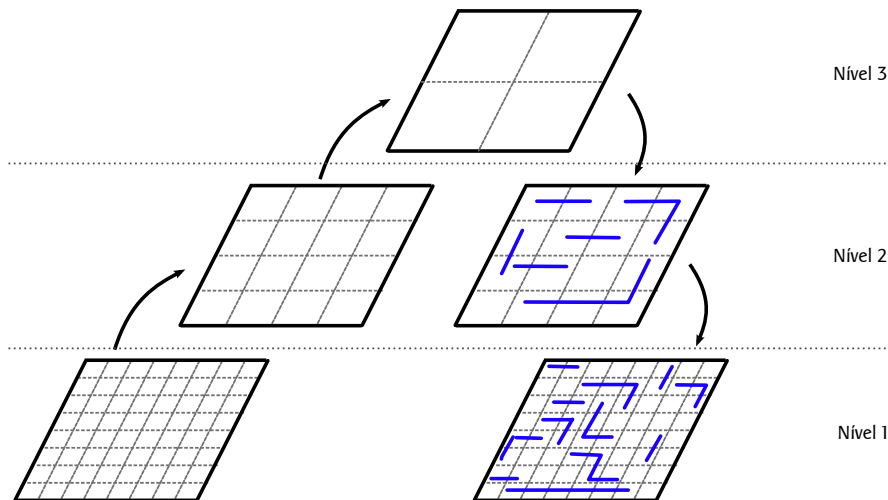


Figura 3.11: Roteamento Hierárquico multi nível.

A rodada de roteamento *bottom-up* realiza a estimativa de congestionamento para calibrar os custos da grade, gerando como saída uma grade agrupada  $2 \times 2$ . O processo *top-down* executa o roteamento, de fato, nessa grade que já possui o congestionamento estimado.

Os experimentos reportados em (CONG; XIE; ZHANG, 2005; XU; CHU, 2011) indicam que o roteamento multi nível pode fornecer resultados semelhantes aos algoritmos que não utilizam o modelo hierárquico, porém, obtendo tempos de execução substancialmente menores.

Vários trabalhos aplicam roteamento multi nível no roteamento global, como pode ser visto no roteadores MGR (XU; CHU, 2011), MARS (CONG; XIE; ZHANG, 2005) e em (CHEN; CHANG; LIN, 2006).

### 3.3 Conclusão

Ao longo deste capítulo foram apresentados os algoritmos clássicos de roteamento de labirinto, busca em linha, roteamento através de padrões e roteamento monotônico. Algumas estratégias utilizadas em fluxos modernos de roteamento como programação linear inteira 0-1, fluxo múltiplo de mercadorias *rip-up and reroute* foram apresentadas e discutidas na parte final do capítulo.

O estudo dessas unidades básicas de roteamento é fundamental para entendimento dos fluxos de roteamento das ferramentas que compõem o estado-da-arte de roteamento global, pois estão distribuídos pelos fluxos de execução dessas ferramentas.

## 4 FLUXO DE EXECUÇÃO DE ROTEAMENTO GLOBAL

Este capítulo apresenta um fluxo de execução de ferramentas de roteamento global, usado como base para análise dos trabalhos relacionados.

### 4.1 Introdução

Nos últimos anos, a pesquisa na área de roteamento global viu surgir uma nova geração de ferramentas, que se desenvolveram no entorno de dois desafios focados em roteamento global e redução de congestionamento: o ISPD 2007 e 2008 (MOFFITT et al., 2008).

Esses desafios contribuíram, ainda, com dados de entrada padronizados para avaliação, num formato já adotado pela comunidade científica para comparação dos resultados, o que possibilita quantificar o impacto de cada técnica aplicada na estratégia de roteamento global.

### 4.2 Anatomia das Ferramentas de Roteamento

Embora sejam muitos os avanços obtidos pelos trabalhos publicados na área de roteamento global, as ferramentas adotam, tipicamente, um fluxo comum de execução na abordagem do problema de planejamento de conexões.

Em cada fase, as ferramentas usam uma combinação de técnicas que guia sua estratégia de roteamento. A Figura 4.1 apresenta esse fluxo comum de execução.

Esse fluxo comum foi identificado durante a pesquisa bibliográfica realizada por este trabalho, considerando artigos publicados nos últimos cinco anos, e será usado como pano de fundo para a análise das ferramentas, descrita no próximo capítulo.

#### 4.2.1 Preâmbulo

Na fase de preâmbulo (ou pré-roteamento), os arquivos de entrada são interpretados e importados para a ferramenta.

Considerando os *benchmarks* ISPD, os arquivos de entrada trazem as posições dos pinos em formato de coordenada na grade de roteamento completa, com o objetivo de que seja possível usar os mesmos *benchmarks* para comparar soluções de roteamento detalhado. Sendo assim, a fase de preâmbulo compreende também a tradução do modelo de coordenada para o modelo de células globais, conforme mostrado na figura 2.1(b).

Durante o processo de importação as ferramentas, de forma geral, optam por um modelo de grade bidimensional ou por uma grade tridimensional.

No caso de uma grade **bidimensional** é feita uma adaptação do modelo multi-camada

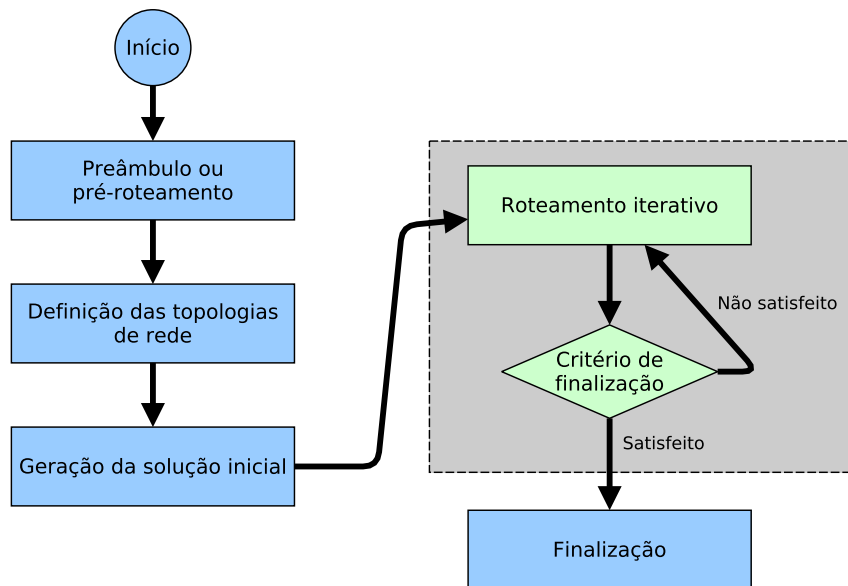


Figura 4.1: Anatomia das ferramentas de roteamento.

fornecido no arquivo de entrada, para um modelo de uma camada, através da soma das capacidades que estão nas diversas camadas de uma célula global para que seja representada por apenas uma.

A modelagem bidimensional acarretará na necessidade de uma fase de assinalamento de camadas, antes da entrega do resultado final. O assinalamento de camadas faz o mapeamento do modelo 2D novamente para o modelo 3D, que representa o circuito de múltiplas camadas.

A simplificação do roteamento feita no roteamento bidimensional pode ter um impacto negativo no comprimento médio dos fios, considerando que as vias são mapeadas para unidades de fio (como é feito nos desafios ISPD). Embora as ferramentas, tipicamente, penalizem a troca de sentido do fio durante o roteamento, muitas vezes isso não reflete o comportamento real do circuito, onde irá ocorrer o empilhamento de vias entre as diversas camadas, e não apenas entre as duas consideradas no roteamento bidimensional.

Apesar dos problemas apontados, o roteamento mapeado em bidimensional é usado pela maioria das ferramentas de roteamento global como o BoxRouter (CHO; PAN, 2006), NTHU-Route (CHANG; LEE; WANG, 2008), NTUgr (HSU; CHEN; CHANG, 2008), Archer (OZDAL; WONG, 2007), BFG-R (HU; ROY; MARKOV, 2010), NCTU-GR (LIU et al., 2010) e GLADE (CHANG; LEE; WANG, 2010).

Outra abordagem é fazer o roteamento numa grade **tridimensional**, onde é possível ter uma representação da grade de roteamento com todas as camadas de metal, sendo representadas por arestas específicas.

O uso de roteamento tridimensional completo é reportado no algoritmo GRIP (WU, 2011). No trabalho de Roy e Markov (algoritmo FGR), é feita uma comparação de roteamento usando mapeamento bidimensional em relação ao mapeamento tridimensional (ROY; MARKOV, 2008). Os resultados mostram que é possível obter ganho em comprimento médio dos fios usando roteamento tridimensional, porém, com um incremento significativo em tempo de execução.

Entretanto, no trabalho desenvolvido em (XU; CHU, 2011) (algoritmo MGR), o uso de roteamento hierárquico multi nível mostrou-se efetivo para realização de roteamento

3D com tempo de execução reduzido, sem perda na qualidade final do roteamento, com foco na redução do número de vias.

#### 4.2.2 Definição das topologias

Considerando que em projetos modernos de circuitos VLSI, mais de 50% das redes possuem mais de dois pinos (WU, 2011), grande parte dos algoritmos modernos de roteamento executam uma fase de geração de redes de dois pinos.

A fase de definição de topologias consiste em traduzir a localização dos pinos de cada rede para os pares de nodos que serão roteados nas próximas fases.

Dependendo do fluxo de roteamento global adotado, ele pode utilizar um algoritmo próprio para definição das redes de dois pinos equivalentes ou reusar uma solução pronta para criação dessas topologias.

No caso da reutilização de uma solução pronta, uma opção frequente das ferramentas de roteamento atuais é pelo algoritmo FLUTE (CHU, 2004).

O FLUTE foi originalmente projetado para realizar estimativas de comprimento médio de fio (CHU, 2004). Posteriormente, o FLUTE foi estendido para criação de RSMT (árvore mínima retilínea de Steiner), em (CHU; WONG, 2004).

A técnica se baseia em uma pesquisa em tabela, pelos chamados vetores com média de comprimento de fio potencialmente ótimo (POWV, do inglês, *potentially optimal wirelength vectors*). Para encontrar as RSMT de uma rede é necessário encontrar os pontos de partição e criar as topologias mínimas a partir deles.

Nas redes de até 9 pinos é possível obter o resultado ótimo na construção da RSMT usando FLUTE. Em redes que possuem mais de 30 pinos é possível obter resultados próximos ao ótimo através da decomposição do problema global em instâncias menores de execução do algoritmo básico do FLUTE.

O uso do FLUTE ferramenta para definição das RSMT, é feito em diversos trabalhos relacionados e fluxos de ferramentas atuais de roteamento global (MOFFITT et al., 2008), tais como: BoxRouter (CHO; PAN, 2006), MaizeRouter (MOFFITT, 2008) e NTHU-Route (CHANG; LEE; WANG, 2008).

Entre os algoritmos que não utilizam o FLUTE, estão o NTUgr (HSU; CHEN; CHANG, 2008), FGR (ROY; MARKOV, 2008) e BFG-R (HU; ROY; MARKOV, 2010), que possuem sua própria implementação de geração de MSTs (árvores de expansão mínima).

#### 4.2.3 Geração da solução inicial

A geração da solução inicial é um passo importante no processo de roteamento, visto que ela pode reduzir a carga deixada para o roteamento utilizando algoritmos simples e rápidos, deixando a execução dos algoritmos mais complexos apenas para as regiões do circuito que necessitarem desse tipo de tratamento.

Os algoritmos de roteamento através de padrões e roteamento monotônico são muito usados nessa fase, onde a grade possui pouco grau de congestionamento e a baixa ocupação de meios de interconexão já fixados em alguma célula global.

O uso **roteamento baseado em padrões** é destacado em diversas ferramentas. Como exemplo podem ser citados o Labyrinth (KASTNER et al., 2002), BoxRouter (CHO; PAN, 2006), BoxRouter 2.0 (CHO et al, 2007), FastRoute 2.0 (PAN; CHU, 2008), Archer (OZDAL; WONG, 2007), MaizeRouter (MOFFITT, 2008), NTHU-Router (CHANG; LEE; WANG, 2008).

Nos experimentos reportados em (PAN; CHU, 2008), em um roteador que realiza uma

abordagem de rotear todo o possível com roteamento através de padrões, partindo para a solução de problemas com um roteador de labirinto, apenas 2% das redes não puderam ser roteadas com roteamento através de padrões.

O **roteamento monotônico** também é empregado em diversas soluções modernas de roteamento com baixo tempo de execução como FastRoute (ZHANG; XU; CHU, 2008), NTHU-Route (CHANG; LEE; WANG, 2008) e NTUgr (HSU; CHEN; CHANG, 2008).

#### 4.2.4 Roteamento Iterativo

A etapa de roteamento iterativo é a que, tipicamente, consome fração dominante do tempo de execução das ferramentas de roteamento global.

Nessa fase são executadas várias rodadas de roteamento iterativo, tipicamente baseado em *ripup e reroute*, programação linear, roteamento baseado em negociação, entre outras técnicas abordadas no capítulo anterior. Para obter um meio termo entre a qualidade da solução e tempo de execução, as ferramentas tipicamente utilizam diferentes técnicas de roteamento, dependendo da localização da rede. Se estiver em uma área com pouco congestionamento, é empregado um algoritmo com tempo de execução reduzido (roteamento baseado em padrões ou monotônico, por exemplo), enquanto que se estiver em uma área congestionada, são empregadas técnicas que priorizem a qualidade do caminho gerado (por exemplo ILP ou roteamento de labirinto baseado em A\*).

Entre duas rodadas de roteamento, tipicamente, é feita a calibração de custos da grade para orientar as próximas rodadas. As áreas que apresentam congestionamento tem seu custo aumentado, para incentivar as redes a procurarem caminhos alternativos.

Um critério de saída é definido para orientar o final desta etapa, quando a solução estiver pronta para ser entregue. São utilizadas as métricas para roteamento global (apresentadas na seção 2.3) e, tipicamente, constituem critérios para uma saída com sucesso (congestionamento igual a zero) ou sem sucesso, no caso de um número significativo de iterações sem progresso na solução, ou até mesmo um limite de tempo previamente especificado.

#### 4.2.5 Finalização

A fase de finalização ou pós-roteamento, é usada por alguns algoritmos para realizar o ajuste fino da solução obtida.

Nessa fase são empregadas técnicas de roteamento de labirinto para melhorar o roteamento de algumas redes que tenham ficado com desvios em demasia, como no BoxRouter (CHO; PAN, 2006).

Além disso, é efetuada a criação dos arquivos de saída com os caminhos globais de cada rede e as métricas do roteamento.

### 4.3 Conclusão

Nesse capítulo foi apresentado um fluxo genérico de execução de roteamento global, baseado na observação das ferramentas acadêmicas que constituem o estado da arte das publicações da área, a serem apresentadas no próximo capítulo.

O entendimento desse fluxo permite analisar e classificar as ferramentas de roteamento de acordo com os algoritmos utilizados em cada fase, bem como identificar possíveis oportunidades de aplicação de técnicas alternativas em pontos específicos do fluxo de roteamento.

## 5 FERRAMENTAS DE ROTEAMENTO GLOBAL

Neste capítulo serão apresentados os ciclos de execução das ferramentas que compreendem o estado da arte em roteamento global. Diferentes versões da mesma ferramenta foram agrupadas para facilitar a compreensão da evolução e combinações de técnicas utilizadas.

### 5.1 BoxRouter

O BoxRouter 1.0 (CHO; PAN, 2006) é uma ferramenta de roteamento global baseada em programação linear inteira e roteamento de labirinto. Este trabalho obteve a terceira colocação no desafio ISPD 2007, modalidade 2D, e segunda colocação na modalidade 3D.

O processo de roteamento começa com a geração das árvores mínimas de Steiner através do FLUTE, sendo que o BoxRouter trata cada unidade de fio de maneira individual, sem considerar a relação do fio com a rede ao qual ele pertence.

Em seguida, é executada uma fase de pré-roteamento, onde são priorizados os fios que não possuem curvas. De acordo com os resultados do trabalho, cerca de 60% dos fios do circuito são roteados nesta fase, em tempo ínfimo.

A partir da fase de pré-roteamento, são descobertas as primeiras regiões do circuitos que possuem congestionamento. Nessa etapa começa o processo de BoxRouting.

O processo de BoxRouting começa com a definição de uma área retangular onde estão concentradas as principais áreas de congestionamento. Os fios dessa área são então classificados em: *dentro do box* e *fora do box*. O processo de seleção de caminhos de roteamento usando ILP é executado, considerando-se, para os pontos que estiverem em linha reta, o menor caminho reto. Para os caminhos onde é possível aplicar o padrão "L" são considerados os dois caminhos ótimos possíveis.

Para aqueles caminhos que não puderem ser traçados usando ILP, é executado então um algoritmo chamado AMR (*adaptive maze routing*, ou roteamento de labirinto adaptativo), que irá utilizar a classificação dos nodos realizada no primeiro passo. Será considerado um custo unitário para aquelas arestas que foram classificadas como *dentro do box*. Para as demais, é utilizado um valor que representa a diferença entre os recursos de roteamento disponíveis e os recursos já utilizados.

Após a execução dos dois passos do BoxRouting, o tamanho da área considerada é aumentado, e o processo recomeça até que toda a área do circuito seja coberta. Uma etapa de pós-processamento é executada após o BoxRouting iterativo.

Na etapa de pós-processamento, o BoxRouter aplica técnicas para rotar novamente algumas redes que possuam muitos desvios e que possam ser roteadas novamente sem prejudicar a solução global já obtida.

Os resultados do BoxRouter são apresentados na tabela 5.1.

Tabela 5.1: Resultados do BoxRouter para o *benchmark* ISPD 2007 - 3D.

<b>Benchmark</b>	<b>WL (<math>e^5</math>)</b>	<b>MOF</b>	<b>TOF</b>
adaptec1	104,05	0	0
adaptec2	102,97	0	0
adaptec3	235,87	0	0
adaptec4	211,95	0	0
adaptec5	298,08	0	0
newblue1	101,83	2	400
newblue2	155,07	0	0
newblue3	195,51	1088	38976

O BoxRouter 2.0 é a continuação do trabalho no BoxRoute e apresenta diversas mudanças no fluxo original da ferramenta. As principais mudanças realizadas foram a implementação de uma nova técnica baseada em negociação usando pesquisa A\* e um novo processo de *ripup e reroute* para manipulação das topologias de rede. Essa técnica objetiva traçar, oportunamente, alguns caminhos adicionais de uma determinada rede que apresente congestionamento, para proporcionar maior liberdade ao processo de negociação de uma nova rota.

Devido ao fato de que o BoxRouter realiza mapeamento do espaço tridimensional em uma grade bidimensional, é necessário uma fase de assinalamento de camadas, a qual é feita usando ILP, onde o objetivo é minimizar o número de vias.

Os resultados do BoxRouter 2.0 são mostrados na tabela 5.2, para a modalidade 3D do concurso ISPD 2007.

Tabela 5.2: Resultados do BoxRouter 2.0 para o *benchmark* ISPD 2007 - 3D.

<b>Benchmark</b>	<b>WL (<math>e^5</math>)</b>	<b>MOF</b>	<b>TOF</b>
adaptec1	92,04	0	0
adaptec2	94,28	0	0
adaptec3	207,41	0	0
adaptec4	186,42	0	0
adaptec5	270,41	0	0
newblue1	92,94	2	394
newblue2	134,64	0	0
newblue3	172,44	364	38958

## 5.2 FastRoute

O algoritmo FastRoute foi pensado para ser aplicado tanto na fase de posicionamento quanto na fase de roteamento. Seu foco é reduzir ao máximo o tempo de execução através do emprego de algoritmos de roteamento de alto desempenho, para gerar estimativas de roteamento de forma rápida.

No algoritmo FastRoute original, o processo começa com a definição das topologias de rede usando o algoritmo FLUTE e aplicação de roteamento baseado em padrões,



usando apenas o padrão "I" e "L". Em seguida as topologias geradas pelo FLUTE são modificadas de acordo com o mapa de congestionamento gerado no passo anterior.

Para as redes que possuem mais de 4 pinos, é aplicada uma técnica de movimentação de arestas, visando o compartilhamento de duas arestas L que possam compartilhar um mesmo canal de roteamento. O mapa de congestionamento é atualizado após cada movimentação.

Após a geração das árvores de Steiner modificadas e melhoradas, a próxima etapa visa realizar o roteamento, que é feito através de roteamento baseado em padrões, utilizando o padrão "Z". Para as redes que permanecerem em regiões de congestionamento, é aplicado então um algoritmo de roteamento de labirinto. Apenas 2.15% das redes são roteadas utilizando o roteador de labirinto.

Na versão 2.0, o roteamento baseado em padrões foi substituído por roteamento monotônico, e um novo algoritmo de roteamento de labirinto com múltiplas origens e múltiplos destinos foi proposto.

A motivação para criação deste algoritmo de roteamento com múltiplas origens e múltiplos destinos, está no fato de que a simplificação utilizada pelo roteador de labirinto, que consiste em quebrar as conexões de uma rede complexa em caminhos de dois pinos pode trazer problemas para a solução final de roteamento da rede, conforme ilustra a 5.1.

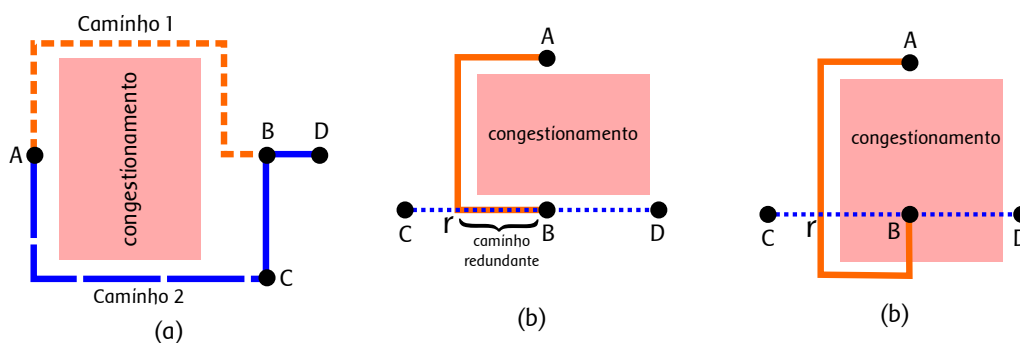


Figura 5.1: Cenários envolvendo labirinto. Adaptado de (PAN; CHU, 2008).

No cenário (a) da figura 5.1, um roteador de labirinto tradicional, considerando uma aresta (A,B), optaria pelo caminho 1, enquanto que, considerando que esta aresta não é obrigatória, e que é possível completar o roteamento da rede por um caminho (A,C), o caminho 2 apresenta menor custo de roteamento e duas vias a menos.

Na figura 5.1(b), é apresentado um caminho com redundância. Considerando a necessidade da topologia conectar [(A,B), (C,B) e (B,D)] como pontos da rede de dois pinos, o caminho (r,B) será traçado de forma redundante por um roteador de labirinto tradicional.

O cenário da figura 5.1(c) apresenta a mesma rede mostrada em (b), porém com uma alteração na área de congestionamento. Nesse caso, um roteador de labirinto tradicional poderia criar um caminho de roteamento desnecessário, devido a necessidade de conectar, individualmente, os pontos das redes de dois pinos.

Para atenuar o problema, foi proposto um algoritmo de roteamento com múltiplas origens e múltiplos destinos. A ideia central é que a integridade da rede seja mantida, embora os pontos de contato entre os pinos possam ser modificados para favorecer os objetivos do roteamento.

A figura 5.2 apresenta um exemplo do algoritmo numa rede. Considerando que a topologia da rede tenha sido definida, a conexão necessária seja entre a ligação

(A,B), e que exista uma área de congestionamento entre os dois pontos, o algoritmo irá primeiramente remover esta conexão da topologia da rede, e buscar um novo ponto de contato que favoreça a passagem por áreas não congestionadas. Neste caso, os pontos escolhidos foram X e Y.

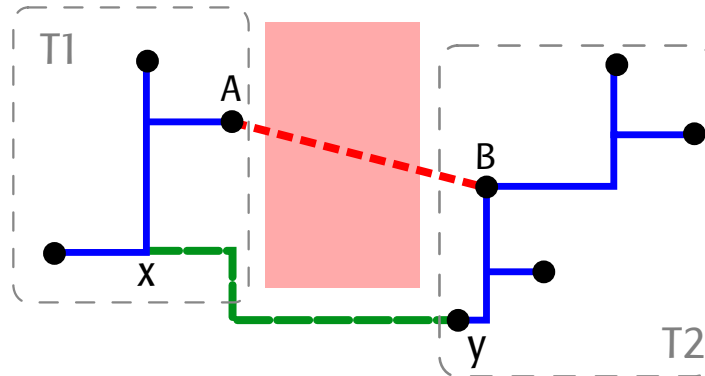


Figura 5.2: Roteamento com múltiplas origens e múltiplos destinos. Adaptado de (PAN; CHU, 2008).

Durante o processo de roteamento iterativo, o FastRoute 2.0 seleciona apenas as redes que tem superior a um limite mínimo definido (não especificado no trabalho) que serão encaminhadas ao roteamento com múltiplas origens e múltiplos destinos. As redes de tamanho inferior ao mínimo definido e aquelas que não passam por regiões de congestionamento são roteadas usando roteamento através de padrões.

Os resultados do FastRoute 2.0 apresentam uma redução de 73% em relação ao FastRoute original, mas ainda assim 78x mais rápido que seu concorrente *Labyrinth* (KASTNER et al., 2002) e 37x mais rápido que o *Chi Dispersion*. Além disso, o congestionamento total foi reduzido em cerca de 10x em relação ao *Labyrinth* e 20x em relação ao *Chi Dispersion*.

A versão 3.0 do FastRoute (ZHANG; XU; CHU, 2008), apresenta o conceito de **capacidade virtual** (*virtual capacity*), que visa reduzir dinamicamente a capacidade das arestas onde há congestionamento para aumentar o custo de roteamento nessa área.

A aplicação da técnica de capacidade virtual foi feita através da inclusão de dois passos dentro do fluxo do FastRoute 2.0, sendo eles: inicialização da Capacidade Virtual e atualização da capacidade virtual. A inicialização é feita através de uma técnica denominada ACE (*adaptive congestion estimation*), que considera o custo de roteamento interno ao perímetro da rede.

Para uma determinada aresta da grade, a capacidade virtual é calculada através da seguinte equação:

$$vc_e = rc_e - (\max(0, p_e - rc_e)) \forall e$$

O termo  $rc_e$  se refere a capacidade real da aresta e  $p_e$  é o valor retornado pelo algoritmo de estimativa de congestionamento.

Para atenuar os problemas decorrentes dos desvios que o roteador de labirinto pode vir a criar após muitas rodadas de roteamento, tornou-se necessário fazer correções no valor da capacidade virtual, sendo ele ajustado para  $vc_e - o_e$  quando  $o_e \geq 0$  e  $vc_e - F \times o_e$  se  $o_e < 0$ . Onde  $F = 0,85$ , sendo utilizado para corrigir a capacidade virtual após a aresta não apresentar mais congestionamento.

O termo  $o_e$  é definido como sendo a demanda na aresta, representado por  $u_e$  menos o custo da aresta.

O custo das vias é levado em consideração no algoritmo FastRoute 3.0. Cada via inserida acrescenta o equivalente a uma unidade de fio a cada troca de sentido de roteamento.

O foco em redução de vias continua no FastRoute 4.0. Três modificações no algoritmo foram propostas para racionalizar o uso de vias:

- Algoritmo para geração de Árvores de Steiner com foco em redução no número de vias;
- Roteamento com até três desvios complementando o roteamento com múltiplas origens e múltiplos destinos e
- Modificações no assinalamento de camadas de redes e arestas.

### 5.3 Archer

O algoritmo Archer (OZDAL; WONG, 2007) faz uso de roteamento de labirinto iterativo com *rip-up e reroute*. Ele faz um tratamento diferenciado para as redes que estão em locais congestionados e redes que estão em áreas de pouca ocupação. A geração das árvores de Steiner é feita através do algoritmo FLUTE.

Na primeira rodada de roteamento é feita uma estimativa de todas as redes roteadas com seu caminho ótimo. Para as redes em locais de baixa ocupação, um algoritmo de roteamento baseado em padrões é aplicado, com o objetivo de reduzir o tempo gasto com essas regiões. Por outro lado, nas redes que estão em locais congestionados, é executado um algoritmo guloso baseado em roteamento de labirinto que objetiva melhorar a qualidade do roteamento nessas regiões.

Durante o processo de *rip-up and reroute*, são empregadas diversas técnicas, progressivamente: A primeira tentativa é feita com roteamento baseado em padrões sem desvios do caminho de tamanho ótimo ("I", "L" e "Z"). Caso esta técnica falhe, é empregado o roteamento baseado em padrões no formato "U", seguido de roteamento monotônico e finalmente roteamento de labirinto.

A função de cálculo de custo do Archer é baseada em negociação, considerando a informação histórica de roteamento no cálculo do custo entre as rodadas de roteamento. Desta forma, se uma aresta estiver congestionada repetidamente nas últimas iterações, o custo para passar pelas mesmas arestas vai crescer vertiginosamente, desencorajando a rede a passar por essas arestas.

Embora utilize as árvores de Steiner geradas pelo FLUTE, o algoritmo Archer realiza modificações com o objetivo de evitar as áreas de congestionamento, através de uma formulação baseada em multiplicadores de Lagrange.

O assinalamento de camadas é a etapa final do algoritmo Archer, realizado em três passos, sendo eles: ordenamento das redes em ordem crescente de comprimento de fio; assinalamento da rede inteira a uma camada, desde que esse assinalamento não cause congestionamento na aresta. Finalmente, ele processa as redes que não conseguiram ser atribuídas, atribuindo seus segmentos a camadas, desde que não causem congestionamento.

Os resultados do algoritmo Archer para o *benchmark* ISPD 2007 demonstram uma redução de 32% em comprimento médio dos fios, quando comparado ao FastRoute. Os

resultados detalhados de execução do Archer no *benchmark* ISPD 2007 – modalidade 3D – são mostrados na tabela 5.3.

Tabela 5.3: Resultados do Archer para o *benchmark* ISPD 2007 - 3D.

<i>Benchmark</i>	WL ( $e^5$ )	TOF	CPU(min)
adaptec1	113,8	0	87
adaptec2	112,56	0	23
adaptec3	244,08	0	50
adaptec4	221,57	0	12
adaptec5	334,09	0	247
newblue1	116,08	682	50
newblue2	166,5	0	7
newblue3	198,77	33394	163

## 5.4 MaizeRouter

O MaizeRouter (MOFFITT, 2008) foi o algoritmo vencedor da categoria 3D do desafio ISPD 2007.

Este algoritmo utiliza mapeamento da grade 3D fornecida pelo arquivo de entrada do *benchmark*, para uma grade 2D interna, onde é feito o roteamento.

O processo de roteamento usando MaizeRouter baseia-se em duas técnicas para acomodação das redes na área do circuito: movimentação de arestas (*edge shifting*), ilustrado pela figura 5.3 e retração de arestas (*edge retraction*), ilustrado na figura 5.4, sendo efetuadas algumas modificações na topologia das redes geradas pelo FLUTE, quando necessário.

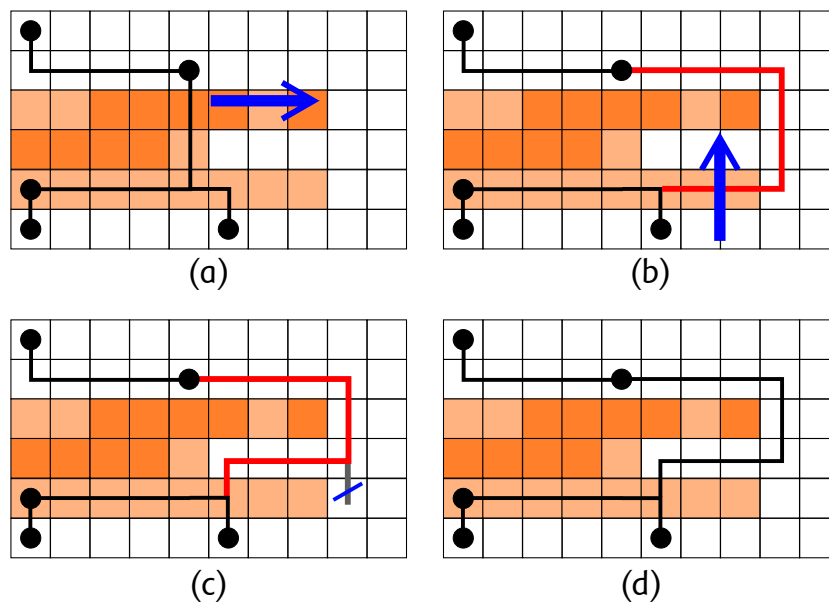


Figura 5.3: Processo de movimentação de arestas do algoritmo MaizeRouter. Adaptado de (MOFFITT, 2008)

Durante o processo de movimentação de arestas, são criados desvios nas redes situadas nas áreas mais congestionadas do circuito. Em seguida, na fase de retração de

arestas, são identificados pontos em que a movimentação das arestas pode ser melhorada ou que curvas possam ser removidas, para reduzir o comprimento total da rede.

Tanto a movimentação quando a retração de arestas é seguida de um procedimento de limpeza e remoção de lixo, que pode ser originado pela manipulação dos meios de roteamento. Nesse processo, o excedente de interconexão é removido para liberar as arestas que já não fazem mais sentido para uma determinada rede. As figuras 5.3(c) e 5.4(b) ilustram esse processo.

Ao final, é realizado um processo de assinalamento de camadas da grade 2D utilizada no roteamento para a grade 3D que deverá ser utilizada no resultado.

Os resultados do trabalho, executando os benchmarks ISPD 2007 (apresentados na tabela 5.4), demonstram que o MaizeRouter obtém comprimento médio de fio em média 2.02% menor, com tempo de execução 7.5x mais rápido, quando comparado com a primeira versão do BoxRouter.

Tabela 5.4: Resultados do MaizeRouter para o *benchmark* ISPD 2007 - 3D.

<i>Benchmark</i>	WL ( $e^5$ )	TOF
adaptec1	99,7	0
adaptec2	99,53	0
adaptec3	210,19	0
adaptec4	198,81	0
adaptec5	303,34	0
newblue1	100,38	1372
newblue2	139,22	0
newblue3	131,68	32234

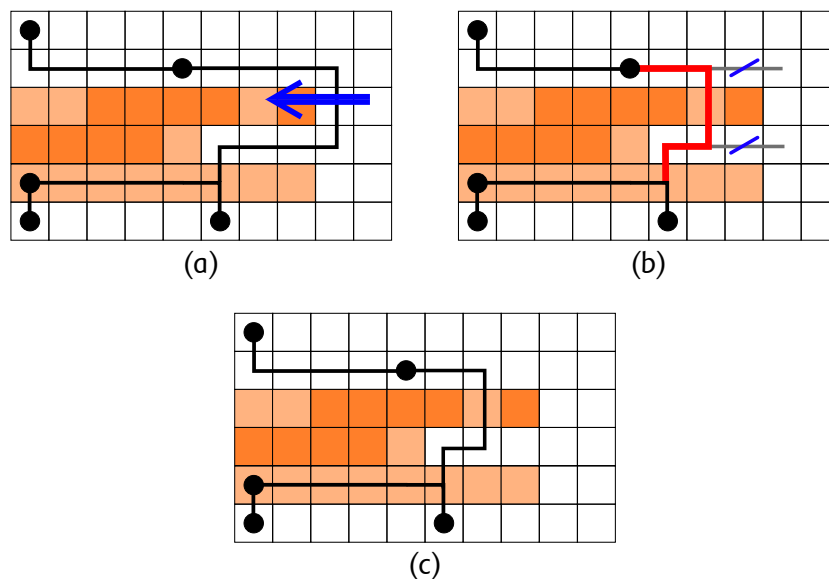


Figura 5.4: Processo de retração de arestas do algoritmo MaizeRouter. Adaptado de (MOFFITT, 2008)

## 5.5 FGR e BFG-R

O "Faily Good Router", ou FGR, foi o vencedor da categoria 2D do desafio ISPD 2007 e terceiro lugar na categoria 3D.

O algoritmo FGR (ROY; MARKOV, 2008) é baseado no PathFinder (MCMURCHIE; EBELING, 1995), que foi modificado para se adaptar ao número de componentes dos circuitos atuais e à quantidade de redes e camadas de roteamento envolvidas. Este algoritmo foi o primeiro a conseguir completar com sucesso, todos os *benchmarks* do desafio ISPD98.

Para geração das árvores mínimas de Steiner, o FGR utiliza uma combinação dos resultados do algoritmo FLUTE e do FastSteiner. Diferente de outros algoritmos, o FGR faz adaptações nas redes em tempo de roteamento.

Quanto a modelagem da área de roteamento, o FGR pode utilizar tanto o modelo de mapeamento da área tridimensional numa grade bidimensional ou roteamento diretamente na grade tridimensional. Ambos resultados são reportados no trabalho.

O processo de roteamento do FGR utiliza uma combinação de *rip-up e reroute* associada com Multiplicadores da Lagrange, para encontrar o mínimo comprimento de fio, quando sujeito a um conjunto de restrições.

Os resultados do FGR são apresentados na tabela 5.5.

Tabela 5.5: Resultados do FGR para o *benchmark* ISPD 2007 - 3D.

<b>Benchmark</b>	<b>WL (<math>e^5</math>)</b>	<b>TOF</b>
adaptec1	88,2	0
adaptec2	89,96	0
adaptec3	200,14	0
adaptec4	178,9	0
adaptec5	260,53	0
newblue1	90,68	238
newblue2	129,3	0
newblue3	163,41	38398

O BFG-R (HU; ROY; MARKOV, 2010) é a sequência do trabalho com o FGR, com melhorias em consumo de memória através do agrupamento de arestas vizinhas durante o processo de *rip-up and reroute*. Além disso, uma nova função para cálculo do custo em função do custo histórico de uma rota foi proposta com o objetivo de melhorar o processo de negociação de caminhos.

## 5.6 NTHU-Route

O algoritmo NTHU-Route realiza o roteamento numa grade bidimensional, através do emprego de roteamento baseado em padrões e roteamento monotônico. A geração das redes de steiner é feita através do algoritmo FLUTE.

A fase de geração da solução inicial é feita através de roteamento baseado em padrões, onde são utilizados os padrões "L" e "I". Quando utilizado o padrão "L", é atribuído um custo de 0.5 para cada um dos possíveis caminhos ótimos. Já para o padrão "I", é atribuído um custo 1 para a reta gerada.

Nas redes onde foram atribuídos custos probabilísticos 0.5, o algoritmo realiza uma

etapa de refinamento, através da modificação das árvores geradas, quando necessário, e posterior processo de roteamento, de acordo com as técnicas apresentadas no algoritmo FastRoute (PAN; CHU, 2008).

Na etapa de roteamento, é feito um processo de *ripup e reroute* em todas as redes que cruzem por nodos congestionados. Todas as zonas de congestionamento são identificadas, agrupadas e ordenadas, de modo que são priorizadas aquelas zonas que apresentam maior congestionamento. Nesta fase são aplicados algoritmos de roteamento monotônico para resolver os congestionamento. Uma vez que o grupo de áreas com maior congestionamento foi reroteado, parte-se para uma nova etapa de cálculo de congestionamento e reinício do processo.

Caso o algoritmo de roteamento monotônico não encontre um caminho válido, é feito um processo de modificação na topologia da rede para encontrar um novo caminho. Ao final do processo é feito o assinalamento de camadas para criação da grade tridimensional resultante.

Os resultados do NTHU-Route para o *benchmark* ISPD 2007, modalidade 2D, são apresentados na tabela 5.6.

Tabela 5.6: Resultados do NTHU-Route para o *benchmark* ISPD 2007 - 3D.

<b>Benchmark</b>	<b>WL (<math>e^5</math>)</b>	<b>TOF</b>
adaptec1	90,56	0
adaptec2	92,17	0
adaptec3	205,04	0
adaptec4	188,43	0
adaptec5	265,03	0
newblue1	90,91	352
newblue2	136,01	0
newblue3	168,4	31800

A partir do trabalho com o algoritmo NTHU-Route, foi desenvolvido o NTHU-Route 2.0, que através das melhorias realizadas, foi o vencedor do desafio de roteamento global do ISPD 2008.

As modificações realizadas pelas versão 2.0 do NTHU-Route foram:

- Implementação de uma nova função de cálculo de custo, baseado no histórico de congestionamento;
- Novos métodos de ordenação para regiões de congestionamento para o processo de *ripup e reroute* e
- Duas novas técnicas de implementação com o objetivo de reduzir o tempo de execução.

Através dessas melhorias, foi possível reduzir em cerca de 8 vezes o tempo de execução, quando comparado ao NTHU-Route original, obtendo um ganho de cerca de 1% em comprimento médio dos fios.

## 5.7 SideWinder

O processo de roteamento do algoritmo SideWinder (HU; ROY; MARKOV, 2008) ocorre de maneira concorrente, aplicando programação linear inteira (semelhante ao BoxRouter), roteamento baseado em padrões e roteamento de labirinto.

A aplicação de roteamento baseado em padrões vai sendo aplicada de forma gradual, partindo do padrão "L", alternando para "Z" e "U" nas redes que necessitem um pouco de desvio.

Uma diferença significativa em relação ao BoxRouter é que o espaço de busca considerado em todas as técnicas de roteamento não restringe o espaço de expansão de nodos. Entretanto, os resultados atuais dessa ferramenta não são competitivos em termos de congestionamento e comprimento médio dos fios (WU, 2011).

## 5.8 NTUgr

O NTUgr (HSU; CHEN; CHANG, 2008) foi segundo colocado no concurso de roteamento global ISPD 2008.

O processo de roteamento com NTUgr começa com o mapeamento do espaço de roteamento tridimensional, numa grade bidimensional. A decomposição do circuito em redes é feita através de um algoritmo de MST.

Após a decomposição das redes e estimativa de congestionamento, são executadas duas técnicas com o objetivo de tratar as zonas de congestionamento do circuito, numa fase chamada "pré-roteamento":

1. Pré-incremento de custo histórico de congestionamento (*Congestion-hotspot Historical cost Pre-increment*), e
2. Roteamento de pequenas áreas congestionadas (*Small bounding-box Area Routing*).

No pré-incremento de custo histórico de congestionamento, é feito um aumento do custo nas áreas adjacentes àquelas que estão congestionadas. O número de vizinhos que terão seu custo aumentado é configurável e foi definido em 10, para os experimentos realizados no NTUgr.

O roteamento de pequenas áreas congestionadas visa resolver primeiro os problemas de pequenas áreas onde há congestionamento. Segundo os autores, essa prática é necessária, pois as redes pequenas tem menos flexibilidade para serem roteadas sem congestionamento. A priorização dessas redes irá ajudar na hora de rotar as redes maiores, uma vez que não terão o impacto de muitas redes menores que desviem de seu caminho ótimo.

Na próxima fase, é feito o roteamento iterativo através de roteamento monotônico. Entre duas rodadas de roteamento, é feita uma avaliação da redução do congestionamento obtida na última rodada. O algoritmo passa para a próxima fase quando o valor de redução de congestionamento for menor do que 5%.

A próxima fase consiste na aplicação da técnica chamada Ripup e Reroute iterativo de regiões proibidas (*Iterative Forbidden-Region Ripup and Reroute*, ou IFR). A cada iteração do IFR são identificadas as regiões proibidas, baseadas em seu congestionamento, conforme ilustrado na figura 5.5. O algoritmo de roteamento aplicado para busca dos caminhos é baseado em A\*.

Durante a fase de expansão de vizinhos, a função utilizada para calcular o custo de passar por uma aresta varia caso essa aresta esteja contida em uma região proibida.



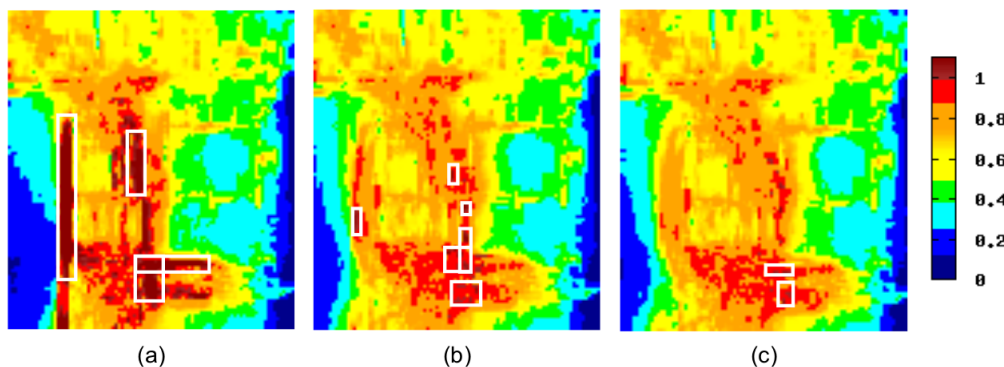


Figura 5.5: Expansão de regiões proibidas do algoritmo NTUgr (HSU; CHEN; CHANG, 2008), usando o *benchmark* adaptec5.

Conforme vão sendo executadas as rodadas de tratamento das áreas congestionadas, o ganho obtido pode acabar oscilando entre duas soluções, que não levam a um ganho na solução global.

Para evitar que a solução fique travada numa solução de ótimo local, o NTUgr detecta quando há pouca variação no congestionamento obtido entre as soluções e passa para uma fase chamada de nivelamento de regiões de propagação (*Region propagation levelling*, ou RPL), onde, ao invés de recriar as regiões proibidas a cada nova iteração, essas regiões são tratadas simultaneamente. A intenção desta técnica é evitar que o congestionamento seja propagado entre as regiões proibidas, conforme elas são tratadas individualmente.

Quando o número de congestionamentos representar 0.5% do obtido na fase de roteamento monotônico, o algoritmo passa para a terceira fase de solução dos congestionamentos, chamada expansão final (*final expansion*). Nesta fase, o roteamento das redes congestionadas restantes poderá usar toda a área do circuito.

A etapa de assinalamento de camadas no NTUgr é realizada por um algoritmo baseado em programação dinâmica, onde o custo de primeira ordem é a quantidade de congestionamento introduzida e a de segunda ordem é o número de vias usadas.

Os resultados do trabalho apresentam comparações segundo os *benchmarks* ISPD 2007 e 2008. Para os circuitos do ISPD 2007, são reportados ganhos em tempo de execução entre 10.8x e 74.8x, sendo, atualmente, os melhores tempos de execução reportados na literatura, para esses circuitos.

Na avaliação dos resultados frente ao resultado final do desafio *benchmark* ISPD 2008, o NTUgr obteve tempos cerca de 3% piores, quando comparado de com o FastRoute 3.0. Entretanto o FastRoute 3.0 conseguiu completar a execução de apenas dez circuitos do *benchmark*, enquanto o NTUgr conseguiu onze. É destacado também que as principais diferenças em tempo de execução e comprimento médio dos fios, a favor do NTUgr, ocorre nos circuitos que são considerados difíceis do conjunto de circuito ISPD 2008 .

## 5.9 GRIP e PGRIP

O algoritmo (WU; DAVOODI; LINDEROTH, 2009) faz o roteamento numa grade 3D, considerando o impacto das vias e comprimento de fio, durante o processo de roteamento. A definição das topologias é feita com o uso do FLUTE.

A principal técnica utilizada no algoritmo GRIP é programação linear inteira, tanto para o processo de roteamento, quanto para a geração das alternativas de solução. Devido

à complexidade envolvida no processamento baseado em ILP, este algoritmo subdivide a área de roteamento em algumas regiões usando critérios de congestionamento, colocando terminais virtuais caso a rede extrapole sua fronteira. Desta forma obtém-se as partes que serão processadas em paralelo.

Após realizar o roteamento em paralelo das sub-regiões, é necessário reconectá-las, o que é feito através de outra formulação ILP. Caso ocorra congestionamento durante este processo, é executado um algoritmo de solução de congestionamento que desconsidera o impacto de comprimento dos fios envolvidos.

Os resultados da execução do GRIP nos benchmarks ISPD 2008 são apresentados na tabela 5.7.

Tabela 5.7: Resultados do GRIP para o *benchmark* ISPD 2008

<b>Benchmark</b>	<b>WL (<math>e^5</math>)</b>	<b>TOF (MOF)</b>	<b>CPU(min)</b>
adaptec1	51,73	0	388
adaptec2	50,32	0	455
adaptec3	127,17	0	478
adaptec4	118,87	0	509
adaptec5	150,35	0	584
bigblue1	54,00	0	339
bigblue2	86,51	0	690
bigblue3	126,47	0	731
bigblue4	222,14	176 (22)	726
newblue1	44,37	0	483
newblue2	72,27	0	467
newblue3	102,79	47812 (636)	1430
newblue4	124,38	132 (16)	529
newblue5	223,76	0	821
newblue6	171,97	0	448
newblue7	338,59	54 (18)	985

O trabalho PGRIP (WU; DAVOODI; LINDEROTH, 2010) foi desenvolvido como sendo a evolução do trabalho GRIP, com a melhoria de algumas técnicas aplicadas na paralelização dos procedimentos de roteamento global, sendo elas:

- Procedimento para estimativa de distribuição das redes entre as sub-regiões de roteamento;
- Isolamento de alguns blocos de código, para possibilitar a execução completamente em paralelo, e
- Inclusão de um critério de finalização baseado em tempo de execução e parametrização explícita da quantidade de iterações a serem realizadas em cada técnica aplicada.

## 5.10 NCTU-GR

O algoritmo NCTU-GR (LIU et al., 2010) utiliza uma abordagem de evolução simulada, combinado com *ripup* e *reroute*. A decomposição das redes em sequências

de dois pinos é feita com a utilização do FLUTE.

Uma organização diferente em relação ao ordenamento das redes é proposta na técnica chamada de roteamento monotônico com ordenação fixa circular, que visa aplicar nas regiões próximas das áreas onde existe congestionamento, um algoritmo de roteamento monotônico.

Após a fase de redução do congestionamento, começa a fase de roteamento baseado em *ripup e reroute* onde é definido um escore entre 0,1 e 0,9 para cada rede. Em seguida um valor entre 0 e 1 é sorteado. Caso o valor sorteado seja maior que o valor atribuído pelo escore, a rede é roteada novamente.

As redes que possuem mais violações são priorizadas na hora de realizar uma nova rodada de roteamento, e ao final, é realizada uma etapa de assinalamento de camadas antes da finalização do algoritmo.

O NTCU-GR foi segundo colocado no desafio ISPD 2008. Os resultados obtidos neste desafio são descritos na tabela 5.8.

Tabela 5.8: Resultados do NCTU-GR para o *benchmark* ISPD 2008

<b>Benchmark</b>	<b>WL (<math>e^5</math>)</b>	<b>TOF</b>
adaptec1	53,50	0
adaptec2	51,97	0
adaptec3	130,08	0
adaptec4	120,62	0
adaptec5	155,01	0
bigblue1	58,39	0
bigblue2	90,01	0
bigblue3	130,20	0
bigblue4	228,03	180
newblue1	46,01	0
newblue2	74,80	0
newblue3	106,22	31872
newblue4	127,98	152
newblue5	231,08	0
newblue6	180,77	0
newblue7	338,68	144

## 5.11 MGR

O algoritmo MGR (XU; CHU, 2011) realiza o roteamento multi nível em uma grade tridimensional.

Na fase de inicialização do algoritmo, a geração das árvores de Steiner é feita através do algoritmo FLUTE. Em seguida, é aplicado roteamento baseado em padrões para geração da solução inicial de congestionamento. Ainda na etapa de inicialização, o assinalamento de camadas é realizado com uma técnica baseada em programação dinâmica.

O processo de roteamento hierárquico é feito com versões diferentes da grade de roteamento global, que apresentam menos detalhes, simplificando o processo de expansão dos roteadores de labirinto aplicados. As soluções de roteamento geradas nessas grades

com menos detalhes, o algoritmo acrescenta gradativamente os detalhes à grade, usando como entrada o mapa de congestionamento das versões simplificadas.

O trabalho apresenta um novo algoritmo de roteamento para redes de três pinos baseado no FastRoute 2.0. O processo começa com a expansão dos vizinhos dos três pinos, até que todos eles consigam encontrar o caminho para, pelos menos um dos outros pinos da rede. A partir das informações da expansão, o algoritmo identifica o ponto de Steiner de menor custo, para servir de ponto de conexão para a rede. Uma vez definido o ponto de conexão, é feita a retro propagação para os três pinos da rede serem conectados.

Os resultados do trabalho, executados nos circuitos do ISPD 2008 apresentam ganhos em comprimento médio dos fios de 1.5% em relação ao NTHU-Route 2.0, 5.3% em relação ao NTUgr e 2.9% em relação ao FastRoute 4.0. Além disso, em relação aos mesmos algoritmos, são reportados ganhos em tempo de execução de 1.7%, 19.8% e 1.4%, respectivamente. Esses resultados consideram apenas os circuitos onde foi possível obter uma solução sem congestionamento.

Para os demais circuitos, são apresentados avanços de 1% em termos de redução do congestionamento, com até 20% de redução no tempo de execução.

## 5.12 GR-WL e GR-RT

O GR é uma ferramenta de roteamento global iterativo baseado em negociação, que realiza mapeamento do espaço de roteamento tridimensional em uma matriz bidimensional.

A ferramenta foi distribuída em dois formatos, de acordo com a configuração de parâmetros que foca em um objetivo diferente de roteamento. O GR-WL tem como objetivo obter o menor comprimento de fio para as redes, enquanto que o GR-RT tem como foco a redução do tempo de execução para conclusão do roteamento. Embora as configurações mudem, o fluxo de roteamento (ilustrado na figura 5.6) é compartilhado pelas duas ferramentas.

Na fase de geração das topologias, foram feitos experimentos usando o FLUTE e a decomposição por MST, sendo que os melhores resultados foram obtidos com MST, pois este dá maior liberdade para escolha dos caminhos, onde a ferramenta explora o compartilhamento de arestas para ligação de todos os fios de uma rede.

A rodada de roteamento inicial é feita através de roteamento baseado em padrões (I ou L). Na versão WL são roteados os fios de comprimento menor que três unidades, enquanto que na versão RT são roteados todos os fios.

A fase de roteamento iterativo utiliza uma estratégia baseada em negociação, utilizando roteamento monotônico e roteamento de labirinto (baseado em A\*). A solução gerada pelo roteamento monotônico utilizada se, e somente se, não gerar nenhum congestionamento (e não ocupar arestas com mais de 95% da sua utilização já preenchida, no caso do GR-RT), caso contrário é utilizada a solução gerada pelo roteamento de labirinto.

Na configuração GR-WL, o roteamento monotônico é desabilitado quando o congestionamento total for menos que 2,5% do congestionamento inicial do circuito, enquanto que na versão RT, parte-se exclusivamente para roteamento de labirinto quando o congestionamento atual for inferior a 15% do inicial.

Na execução do algoritmo de roteamento de labirinto, é feita uma delimitação da área de busca dentro da área de roteamento para reduzir o número de nodos expandidos no decorrer da execução do fluxo de roteamento. O tamanho da área de busca  $\theta$  (Eq. 5.1)

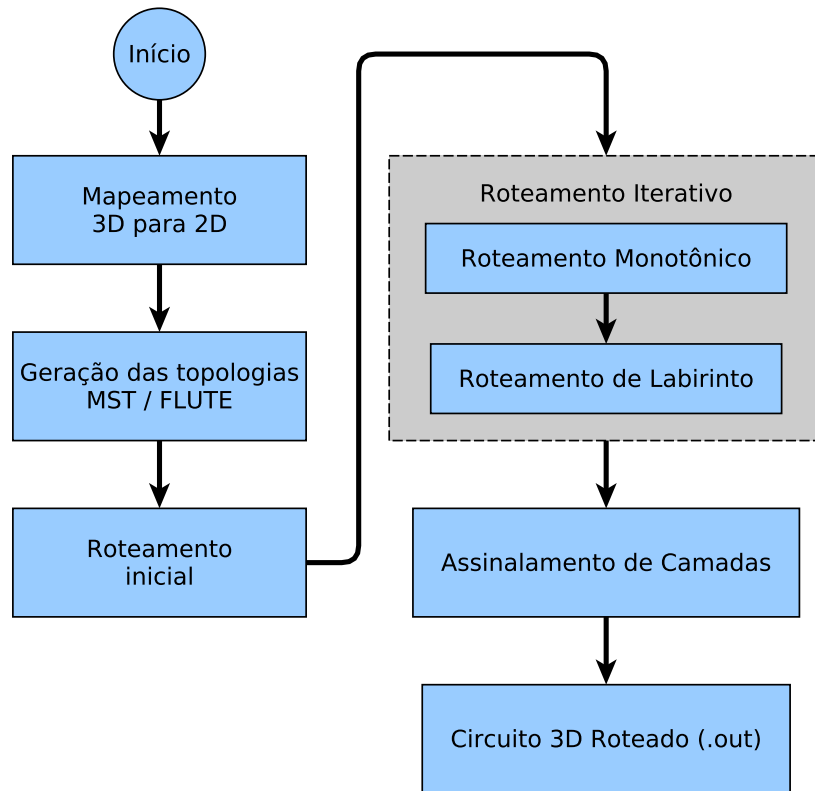


Figura 5.6: Fluxo de execução do GR.

varia de acordo com o andamento da redução de congestionamento. Isso é feito para evitar grandes desvios nas primeiras iterações do roteador.

$$\theta = \min(\zeta, 1.01 + \frac{\eta}{\min_{ov} + 1} + \min(2, \frac{TOF_i}{TOF_{i-1}} \times \frac{e^{\epsilon \times i}}{10})) \quad (5.1)$$

Na equação 5.1,  $i$  é um número sequencial que é incrementado ao final de cada iteração, sempre representa a iteração atual.  $i - 1$  representa iteração anterior a  $i$ . TOF representa o congestionamento total das arestas obtido numa determinada iteração. Na implementação de referência os valores dos outros parâmetros são os seguintes:  $\zeta = 10$ ,  $\eta = 500$  e  $\epsilon = 0.20$ [5].

Durante o roteamento iterativo é verificado antes de realizar o roteamento, se o fio passar por alguma zona de congestionamento. Isso reduz o re-roteamento entre 10% e 60%, principalmente nas primeiras iterações.

A função de custo do algoritmo A\* é apresentada na equação 5.2, onde  $b_e$  representa o custo base, tem valor estático igual a 1.  $p_e$  representa a penalização da iteração corrente e  $h_e$  representa o custo histórico, decorrente de congestionamento existentes em iterações anteriores.

$$cost_e = b_e + h_e \times p_e \quad (5.2)$$

O custo histórico é calculado com base na diferença entre demanda e capacidade, seguindo o que é descrito na equação 5.3.  $h_{inc}$  é 1,25 se a diferença entre demanda e capacidade é maior que 95%, e 1 caso contrário. O valor de  $k$  foi definido em 1, através de experimentos.

$$h_e^i = \begin{cases} h^{(i-1)}, & \text{se } d_e < c_e \\ h^{(i-1)} + k \times h_{inc} & \text{caso contrário} \end{cases} \quad (5.3)$$

O valor da penalização, apresentado na equação 5.2 é calculado de acordo com o apresentado na equação 5.4.

$$p_e = c_0 \times e^{((i \times \alpha + \frac{\beta}{(min_{ov} + 10) + \gamma}) \times (d_e - c_e + 1))} \quad (5.4)$$

Na equação 4,  $i$  representa a iteração atual,  $min_{ov}$  contém o valor do menor TOF encontrado em toda o processo de roteamento atual.  $c_0$ ,  $\alpha$ ,  $\beta$  e  $\gamma$  são constantes definidas através de experimentos. Na implementação atual contém os seguintes valores:  $c_0 = 1.0$ ,  $\alpha = 0.015$ ,  $\beta = 50$  e  $\gamma = 0.30$ [5].

Os resultados do GR-WL com custo de via unitário, são apresentados na tabela 5.9.

Tabela 5.9: Resultados do GR-WL para o *benchmark* ISPD 2008

<b>Benchmark</b>	<b>WL (<math>e^5</math>)</b>	<b>TOF (MOF)</b>	<b>CPU(min)</b>
adaptec1	63,80	0	46,18
adaptec2	60,84	0	15,33
adaptec3	151,55	0	45,42
adaptec4	139,81	0	9,38
adaptec5	181,02	0	120,88
bigblue1	66,85	0	165,90
bigblue2	112,04	122 (2)	281,30
bigblue3	155,94	0	56,25
bigblue4	274,10	780 (6)	197,05
newblue1	54,37	588 (2)	89,75
newblue2	92,45	0	5,37
newblue3	131,34	36292 (1194)	1416,80
newblue4	153,29	462 (2)	323,45
newblue5	269,83	0	283,83
newblue6	207,08	0	164,60
newblue7	421,39	5172 (4)	974,20

### 5.13 Conclusão

Neste capítulo foram apresentadas as ferramentas de roteamento global que representam o estado da arte da área, publicadas nos últimos seis anos. Foram destacadas as técnicas empregadas em cada fase do fluxo de roteamento, tendo como base o fluxo genérico apresentado no capítulo 4.

A partir da análise das ferramentas apresentadas, foi possível observar que a parte dominante dessas ferramentas, fazem uso de mapeamento da área de roteamento numa matriz bidimensional, o que reduz o tempo necessário para executar os algoritmos numa versão simplificada do espaço tridimensional.

Para realização do roteamento inicial, são empregados alguns algoritmos baseados em roteamento monotônico ou roteamento baseado em padrões, para reduzir o tempo necessário para resolver as redes menos complexas, em áreas onde não irá haver

congestionamento. Para as demais áreas, o uso de programação linear inteira e técnicas baseadas em negociação foi observado nas ferramentas que apresentam os melhores resultados em comprimento de fio.

Outra técnica que está diretamente relacionada as ferramentas que apresentam os melhores resultados, tanto em tempo de execução quanto em comprimento dos fios, consiste em inserir no fluxo de roteamento, uma fase de identificação e tratamento de as áreas onde há alta demanda por conexão. Essa fase está integrada ao fluxo de ferramentas como o NTUgr, BoxRouter e ARCHER.

Partindo dessa análise, emerge a oportunidade para investigar e avaliar técnicas de delimitação e tratamento de áreas de alta demanda por interconexão, através de integração dessas técnicas no fluxo de uma ferramenta de roteamento global.

## 6 IDENTIFICAÇÃO E ATENUAÇÃO DE CONGESTIONAMENTO EM ROTEAMENTO GLOBAL

Este capítulo apresenta as técnicas de identificação e tratamento de áreas com alta demanda por meios de interconexão durante o roteamento inicial e alto nível de congestionamento durante o roteamento iterativo. Para cada técnica, são apresentados os parâmetros envolvidos em sua execução e exemplos de uso, através de *benchmarks*.

As técnicas foram inseridas em um fluxo de roteamento global existente, utilizando o roteador GR-WL (REIMANN, 2011) como código-fonte de referência. São descritas as alterações e etapas adicionadas ao fluxo do GR-WL, com o objetivo de medir o impacto da aplicação das técnicas propostas.

No final do capítulo são apresentados os dados resultantes das rodadas de execução e análise dos experimentos, utilizando como dados de entrada os *benchmarks* do desafio ISPD 2008, os melhores resultados da implementação original do GR-WL e uma ferramenta de roteamento que aplica técnicas semelhantes as que são propostas neste trabalho.

### 6.1 Introdução

A partir da análise da execução do fluxo do GR-WL, foi possível observar que os pontos de congestionamento que demandam maior esforço (número de re-roteamentos) na fase de roteamento iterativo, estão localizados em nodos que são origem ou destino de um número elevado de fios, em relação a sua capacidade de roteamento.

O fluxo típico de roteamento global consegue resolver mais de 90% do congestionamento em 10% do tempo total gasto no roteamento, sendo o restante do tempo, utilizado para resolução dos 10% (ou menos) de congestionamento restante. Um exemplo dessa situação pode ser observado na figura 6.1, que ilustra o máximo congestionamento obtido no decorrer das iterações realizadas pelo GR-WL, durante a fase de roteamento iterativo do circuito newblue5 (ISPD 2008-3D), com decomposição de árvores de roteamento usando o algoritmo MST.

No exemplo da figura 6.1, foram necessárias 384 rodadas de roteamento iterativo para gerar a solução final. Deste total, considerando a métrica de congestionamento máximo, foram necessárias dezesseis rodadas de roteamento para reduzir em 90% o máximo obtido na primeira iteração, partindo de 125, para 10, após essas dezesseis rodadas de roteamento.

Com o objetivo de identificar e tratar as áreas com alta demanda de interconexão e alto nível de congestionamento, foram incluídas duas novas etapas no fluxo do GR-WL (REIMANN, 2011), com foco em identificação e atenuação de congestionamento. Para



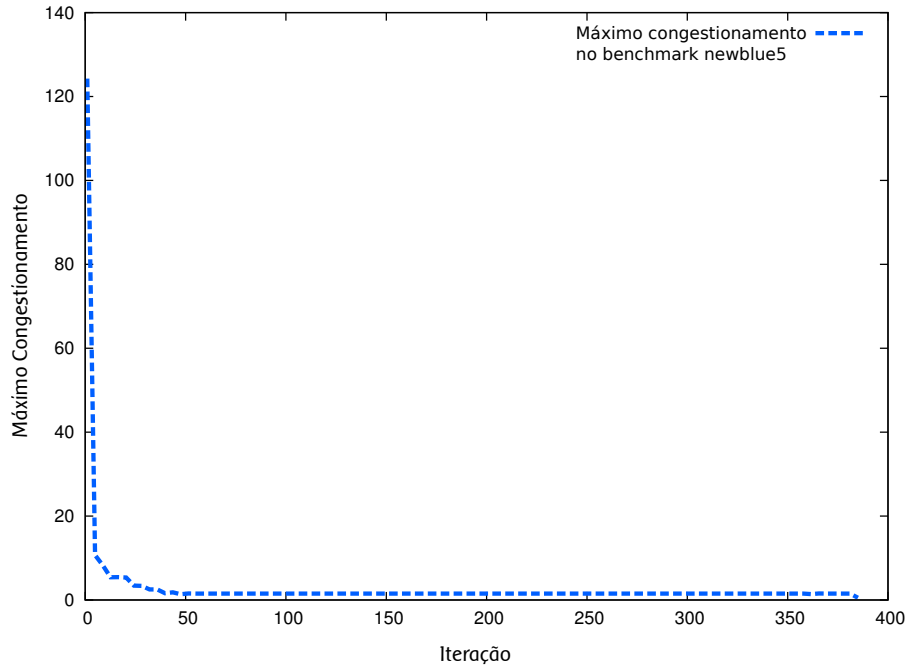


Figura 6.1: Máximo congestionamento obtido durante as iterações de roteamento do *benchmark newblue5*, usando o GR-WL.

simplificar o entendimento, os resultados desse fluxo modificado do roteador GR-WL para abordar problemas de congestionamento, são tratados pelo acrônimo **GR-C**.

A primeira técnica do GR-C é aplicada na fase de preâmbulo, com o objetivo de identificar e tratar as áreas com alta demanda de interconexão, conforme ilustra a figura 6.2(a).

A segunda técnica é executada durante as rodadas de roteamento iterativo, delimitando as áreas com alto nível de congestionamento, conforme mostrado na figura 6.2(b).

## 6.2 Identificação de Alta Demanda por Interconexão no Roteamento Inicial

O processo de identificação dos nodos com alta demanda por interconexão é feito com base na contagem do número de fios que possuem um determinado nodo  $n$  como sua origem ou destino, ou seja, sua demanda  $D_n$ , em relação à capacidade do canal  $C_n$  do nodo de comportar fios.

O valor de  $C_n$  corresponde a soma das capacidades das arestas  $\{C_{norte} + C_{sul} + C_{leste} + C_{oeste}\}$ . A partir desses dados, é possível calcular o percentual de ocupação inicial de um determinado nodo.

Para classificar os nodos, foi definido um limite mínimo de ocupação inicial  $\beta$  para que o nodo seja incluído na lista de alta ocupação inicial. Um nodo  $n$  será incluído na lista de alta ocupação se a seguinte condição for satisfeita:

$$C_n / (D_n \times 100) > \beta \quad (6.1)$$

A Figura 6.3 apresenta um exemplo de nodos com alto percentual de ocupação inicial no *benchmark adaptec1*, onde  $\beta = 50$ , ou seja, a ocupação mínima para que o nodo seja marcado, é que a demanda seja maior do que 50% a capacidade do nodo.

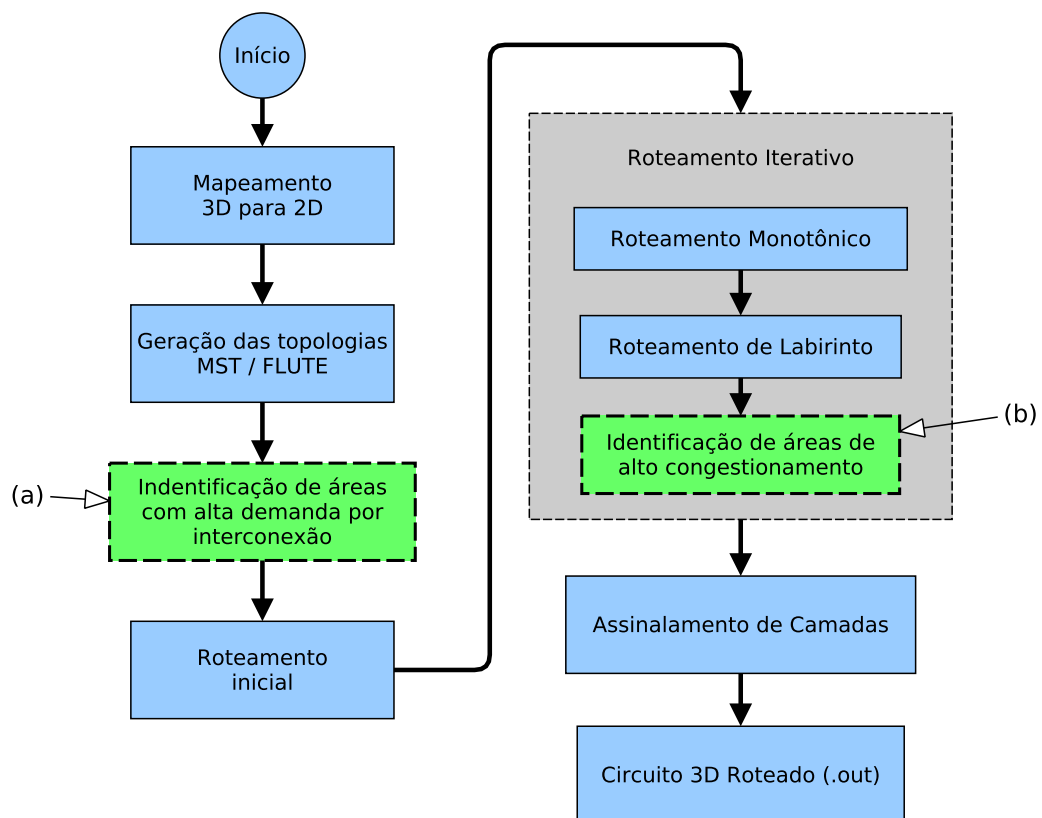


Figura 6.2: Fluxo de execução do GR, com as fases inseridas neste trabalho: identificação de áreas com alta demanda por interconexão (a) e identificação de áreas com alto congestionamento, no roteamento iterativo (b).

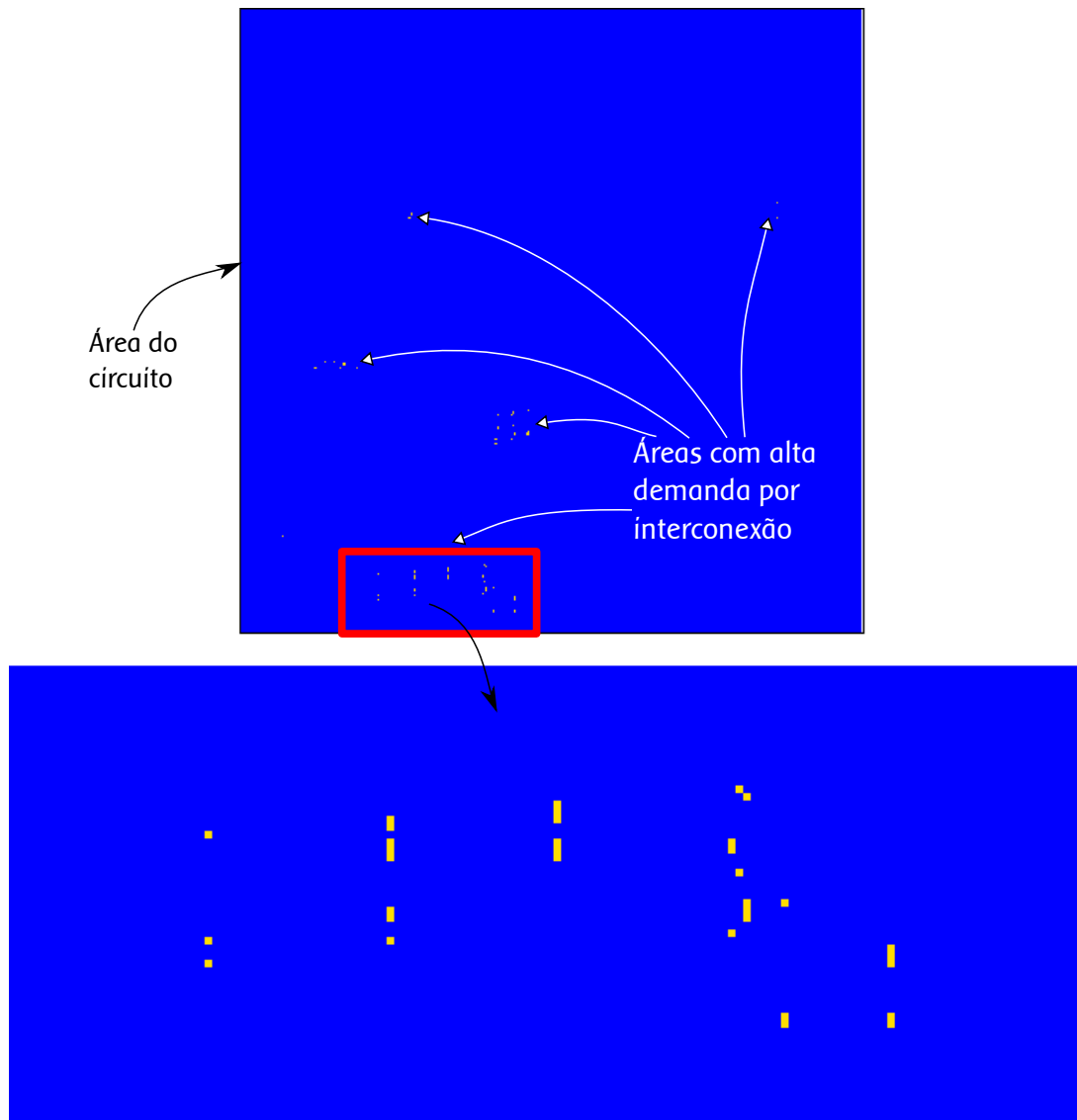


Figura 6.3: Delimitação inicial de áreas com alta demanda por interconexão.

Para tratar as áreas de alta demanda por interconexão, foi necessário ajustar de forma permanente os custos de determinados nodos da grade de roteamento, durante a execução do algoritmo de roteamento iterativo. Isso foi feito através do pré-incremento de custo histórico nas áreas identificadas pelo método descrito acima.

O custo histórico  $h_e$ , conceitualmente, nunca tem seu valor decrementado durante o processo de roteamento, sendo este parâmetro responsável por aumentar o custo das arestas que ficam congestionadas muitas vezes no decorrer das rodadas de roteamento. Esse valor permite que haja uma penalização maior para estas regiões frequentemente congestionadas, evitando que o algoritmo fique alternando entre duas ou mais soluções locais.

Para guiar o processo de pré-incremento do custo histórico, dentro da fase de tratamento de alta demanda por interconexão, foram identificados três parâmetros, sendo eles:

1. Quantidade de custo a ser incrementado;
2. Amplitude de propagação do pré-incremento; e
3. Formato (função) de distribuição.

O primeiro parâmetro se refere a **quantidade de custo**, representada por  $\psi$ , define a quantidade de custo histórico que será incrementado nas arestas que possuem demanda de conexão superior ao valor de  $\beta$ .

A determinação do valor de quantidade de custo tem influência direta na qualidade da solução e na efetividade da aplicação desta técnica. Valores muito baixos ( $\psi = 2$ , por exemplo) não irão elevar o custo o suficiente para que as redes evitem cruzar pelos pontos de alta demanda por interconexão, tornando essa fase pouco efetiva em seu objetivo. Por outro lado, caso o valor escolhido seja muito elevado ( $\psi = 15$ , por exemplo), irá ocorrer um processo de criação de áreas de baixa ocupação por sobrecarga de custo. São as chamadas ilhas de baixo congestionamento, que causam um aumento desnecessário do comprimento dos fios.

A **amplitude** do pré-incremento, representada por  $\alpha$ , define a distância máxima, em nodos, para os quais o custo do pré-incremento de custo será propagado.

A definição de um valor para amplitude tem impacto direto na qualidade da solução final obtida com esta técnica. Caso o valor da amplitude seja muito elevado, é possível observar um aumento excessivo do comprimento total dos fios, decorrente do espalhamento exagerado dos meios de interconexão. Por outro lado, um valor muito baixo (como por exemplo 1 ou 2) pode aumentar o número de rodadas de roteamento, o que tem como efeito colateral o aumento do tempo de execução, sem refletir em ganhos na redução de congestionamento.

Por fim, a **função de distribuição de custo** define como serão calculados os valores de custo histórico a partir do ponto de alta demanda, na amplitude de pré-incremento definida. Foram investigados três modelos de distribuição de custo:

1. Estático,
2. Linear, e
3. Exponencial negativo.

### 6.2.1 Modelo Estático

O modelo **estático** faz a distribuição de valores iguais em toda a amplitude de pré-incremento de custo histórico. Ela é utilizada por ferramentas de roteamento que utilizam técnicas de identificação e atenuação de congestionamento, como o NTUgr (HSU; CHEN; CHANG, 2008).

A figura 6.4 apresenta a aplicação de pré-incremento de custo histórico seguindo o modelo estático, sendo  $\alpha = 5$  e  $\psi = 5$ . Áreas que se sobrepõem tem seus custos incrementados em 50%, para evitar uma sobrecarga de custo. O mapa de pontos de alta demanda por congestionamento é apresentado na figura 6.3, para os pontos de alta demanda, ilustrados na região destacada da figura 6.3.

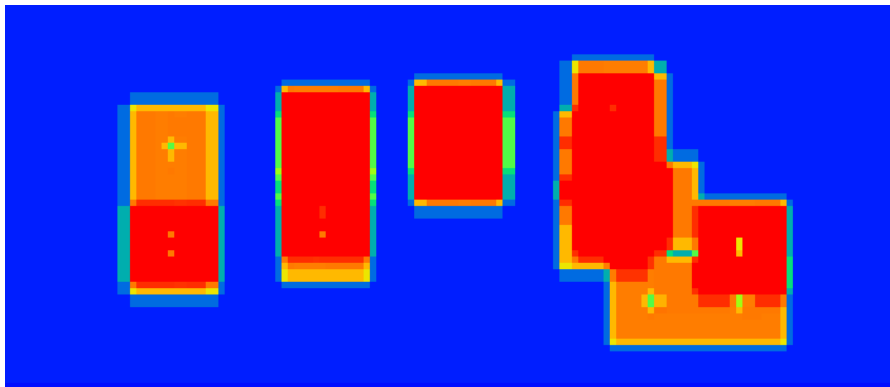


Figura 6.4: Atribuição de custo estático em pontos de alta demanda por interconexão.

A partir dos experimentos realizados, verificou-se que a utilização do modelo estático com baixa amplitude de espalhamento ( $\alpha = 2$  e  $\psi = 5$ , por exemplo), auxilia na redução do comprimento total dos fios e, em alguns casos, no congestionamento total em relação à implementação de referência do GR-WL.

Entretanto, amplitudes de espalhamento maiores ( $\alpha = 10$  e  $\psi = 5$ , por exemplo) favorecem a criação de ilhas de baixa utilização. Através da observação deste efeito colateral, foi possível concluir que conforme uma determinada aresta se afasta do foco de congestionamento, é necessário calibrar seu custo de acordo com essa distância.

Para atenuar esse efeito colateral da aplicação de custo estático, foi proposta a aplicação e avaliação dos outros dois modelos de espalhamento de custo que tem essa característica de calibração de custo em função da distância ao foco de congestionamento.

### 6.2.2 Modelo Linear

Este modelo faz a distribuição de custos usando uma função **linear**, com o objetivo de incentivar gradativamente as redes a encontrarem seus caminhos desviando das áreas de alta demanda por interconexão.

As áreas que estão mais longe do ponto de alta demanda por interconexão receberão um incremento de custo menor do que áreas mais próximas.

A função de custo linear realiza o pré-incremento de custo histórico a partir de um nodo de origem, que é um ponto de alta demanda por interconexão, em todo o perímetro de amplitude  $\alpha$ , seguindo a seguinte função:

$$h_e = h_e + (\psi / (\text{dist}(\text{nodo\_atual}, \text{nodo\_origem}) + 1)) \quad (6.2)$$

Onde  $h_e$  representa o custo histórico do nodo que está sendo visitado. A função  $dist(nodo\_atual, nodo\_origem)$  calcula a distância ortogonal entre o nodo visitado (representado por  $nodo\_atual$ ) e o nodo de origem (ponto de alta demanda por interconexão, representado por  $nodo\_origem$ ). Através dessa função de cálculo da distância que ocorre a variação linear para o custo aplicado.

A figura 6.5 apresenta o resultado do pré-incremento de custo histórico usando essa função linear para  $\alpha = 5$ ,  $\psi = 5$ , aplicada à área do circuito apresentado na figura 6.3.

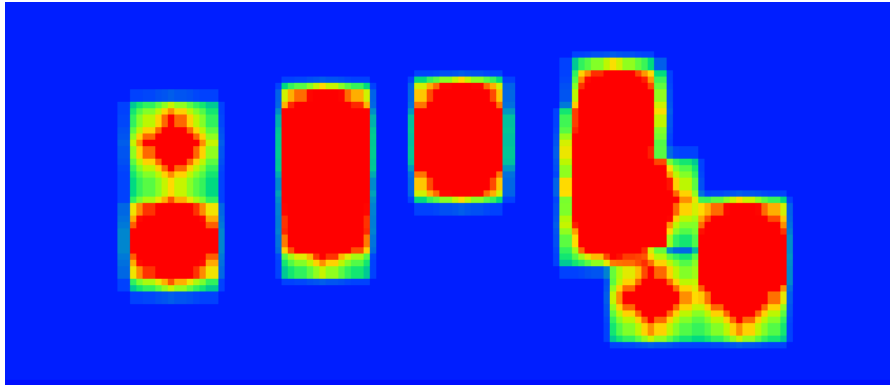


Figura 6.5: Atribuição de custo com taxa de variação linear, em pontos de alta demanda por interconexão.

A aplicação da função linear apresentou resultados equivalentes à distribuição estática, em termos de redução de congestionamento, entretanto, com o efeito de gradiente, que faz com que algumas redes ainda passem próximas aos pontos de congestionamento, possibilitaram uma melhora na métrica de comprimento de fio.

### 6.2.3 Modelo Exponencial Negativo

Conforme dito anteriormente, o pré-incremento de custo histórico ocorre antes mesmo do algoritmo de roteamento iniciar sua execução. Por esta razão, qualquer incremento de custo histórico, por menor que seja, irá interferir no resultado final, que irá incidir na comprimento dos fios e também, potencialmente, na formação de ilhas de baixo congestionamento.

Observando esse comportamento, e visando reduzir o impacto da aplicação de custos históricos em nodos vizinhos aos pontos de alta demanda, foi implementada a função de atribuição de custo **exponencial negativa**, cuja principal característica é a atribuição de um custo igual a  $\psi$  na borda do ponto de alta demanda, e uma queda exponencial dos valores de custos aplicados na vizinhança, mantendo valores sempre maiores do que zero.

A distribuição de custo usando exponencial negativa utiliza a seguinte fórmula para o cálculo do custo a ser atribuído:

$$h_e = h_e + (\psi \times (\exp(-1 \times dist(nodo\_atual, nodo\_origem) + 1))) \quad (6.3)$$

A figura 6.6 apresenta uma distribuição de custo histórico seguindo a função exponencial negativa, que incide sobre a área do circuito indicada na figura 6.3.

Nos experimentos realizados, foi possível verificar que a função exponencial negativa obteve, em média, melhores resultados para comprimento de fio, quando comparados com duas outras funções de distribuição apresentadas. A razão para esses resultados consiste em que as áreas de alta demanda estão protegidas por uma camada de custo igual a  $\psi$ ,

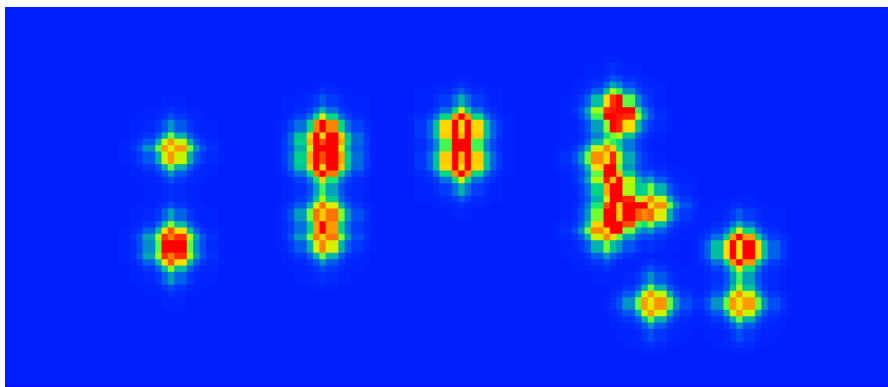


Figura 6.6: Atribuição de custo com taxa de variação exponencial negativa, em pontos de alta demanda por interconexão.

enquanto que a vizinha não é penalizada com altos custos. Ao invés disso, um custo histórico é incrementado com valores baixos, mas que nessa fase do fluxo de roteamento garantem que algumas redes irão evitar passar por essas áreas.

Os dados resultantes da execução de *benchmarks* com as três funções de espalhamento de custo histórico estão descritos nas tabelas 6.1 (usando MST para decomposição das redes) e 6.2 (usando FLUTE para decomposição das redes).

### 6.3 Delimitação e Tratamento de Alto Nível de Congestionamento no Roteamento Iterativo

O procedimento de delimitação e tratamento de alto nível de congestionamento é executado entre o final de uma iteração, e o início de outra, visando marcar os vértices críticos que estão localizados nas áreas onde se deseja aumentar o custo a fim de reduzir a passagem de fios na próxima iteração, como forma de reduzir o congestionamento.

Por vértices críticos, entende-se aqueles vértices que possuem os maiores quantitativos de congestionamento em relação ao máximo congestionamento obtido na iteração.

Após a identificação dos vértices críticos, é executado um procedimento de expansão, para criar uma borda de custo que irá proteger a área que apresenta alto congestionamento e juntar áreas com alto nível congestionamento que estejam próximas, sendo este valor configurável e definido em 3 arestas, na implementação de referência. A figura 6.7(b) apresenta um exemplo dessa expansão.

A técnica utilizada para identificar os vértices críticos, se baseia na abordagem do NTUgr (HSU; CHEN; CHANG, 2008), com algumas adaptações. A aplicação desta técnica consiste em três fases:

- Identificação de nodos com alto congestionamento;
- expansão de nodos vizinhos e
- atribuição de custo de penalização.

A identificação ocorre apenas a partir dos nodos com congestionamento igual ao máximo congestionamento obtido na iteração atual. A tolerância para os nodos que serão incluídos na expansão, varia gradualmente para os nodos vizinhos que possuem congestionamento de 10% a 40% menores que o máximo congestionamento da iteração.

Para os circuitos cuja diferença o máximo congestionamento é menor que o dobro da média, a técnica empregada pelo NTUgr é efetiva. Entretanto, nos circuitos onde a diferença entre o congestionamento máximo e a média é superior a duas vezes o valor da média de congestionamento, o número dos nós identificados pelos critérios do NTUgr é tão reduzido que não reduz o potencial de identificação de regiões que a técnica possibilita, reduzindo assim seu impacto no resultado final do roteamento.

Dessa forma, optou-se por adaptar o algoritmo de identificação e tratamento de áreas de alto congestionamento para usar os valores de congestionamento num intervalo definido entre a média e o máximo congestionamento, de modo a garantir que um número maior de nós seja identificado nessa fase.

Através das experiências realizadas, o valor de base para delimitação de áreas congestionadas  $C_{min}$  foi definido como o mínimo valor entre 0,5 e 90% da média de congestionamento, ou seja, referente a 50% do valor do máximo congestionamento obtido na iteração anterior ou, quando esta for inferior a 0,5, reduz-se o requisito para 90% do valor da média, garantindo assim, que mais nós serão identificados com base na situação do circuito.

Dessa forma, o valor de  $C_{min}$  foi definido como:

$$C_{min} = \min(0.5, \text{media\_congestionamento} * 0.9) \quad (6.4)$$

A fase de expansão dos nós identificados na primeira fase é executada de maneira semelhante a expansão de um algoritmo de busca em labirinto. Partindo de um nó de origem, uma lista de nós expandidos e uma fila de nós a serem expandidos, são visitados todos os nós vizinhos do nó de origem.

Após essa fase de expansão, é realizada uma etapa de marcação de bordas das áreas identificadas, com o objetivo de juntar áreas de congestionamento que estejam muito próximas, com o objetivo de evitar a criação de canais estreitos que irão gerar congestionamento nas próximas iterações, visto que serão a primeira opção de desvio. Para essas áreas, é feita um incremento na penalização com metade do valor atribuído às áreas identificadas nos passos anteriores.

O resultado de uma expansão, na oitava iteração do fluxo de roteamento no circuito *bigblue1* é apresentado na figura 6.7.

Durante a visita de um nó, são incluídos na lista de nós a serem visitados, todos os vizinhos que possuam pelo menos uma aresta com congestionamento superior ao limite mínimo para expansão.

Na fase de expansão de nós vizinhos, o limite de congestionamento mínimo para ser incluído na área de alto congestionamento, definido de maneira experimental, foi definido como uma redução de 10% no valor de  $C_{min}$ .

A figura 6.8 apresenta um exemplo de mapa de congestionamento detalhado, após a execução das duas primeiras fases (identificação e expansão), no circuito *adaptec5* e seu respectivo mapa de áreas de alto nível de congestionamento (área clara da figura "b").

Nas últimas iterações, quando o congestionamento é igual ou inferior a duas unidades, a técnica é desabilitada para não causar um efeito colateral de aumento dos fios, pois todos os nós serão selecionados como sendo em áreas de alto congestionamento, causando um incremento de custo desnecessário que irá incorrer em aumento do comprimento dos fios envolvidos.

Diferentemente do tratamento realizado na fase do pré-roteamento, o tratamento de arestas identificadas no roteamento iterativo, irá incrementar o valor de penalização  $p_e$ . A utilização deste parâmetro da função custo do roteador de labirinto deve-se ao fato de



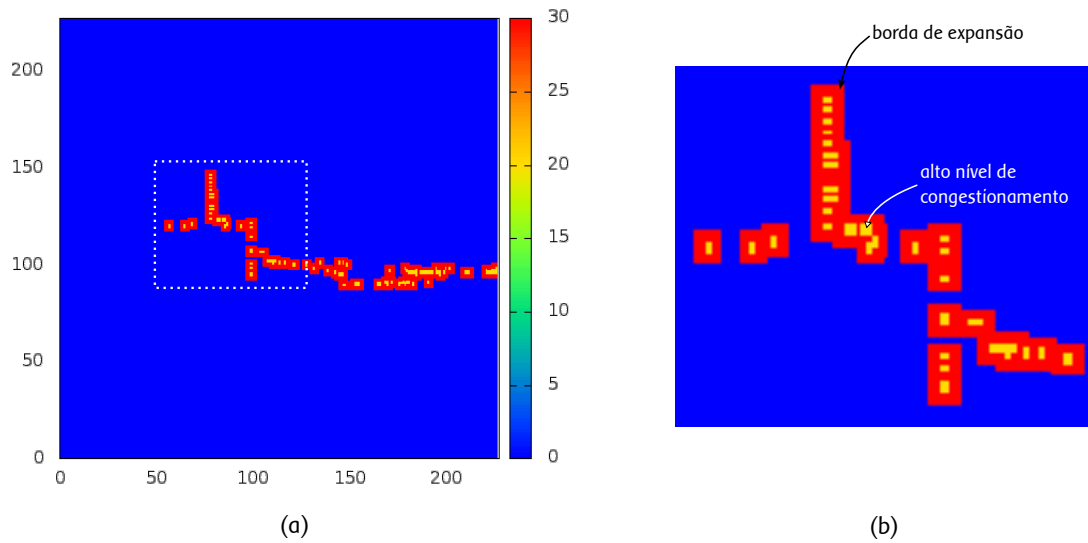


Figura 6.7: Expansão de área com alto nível de congestionamento durante o roteamento iterativo.

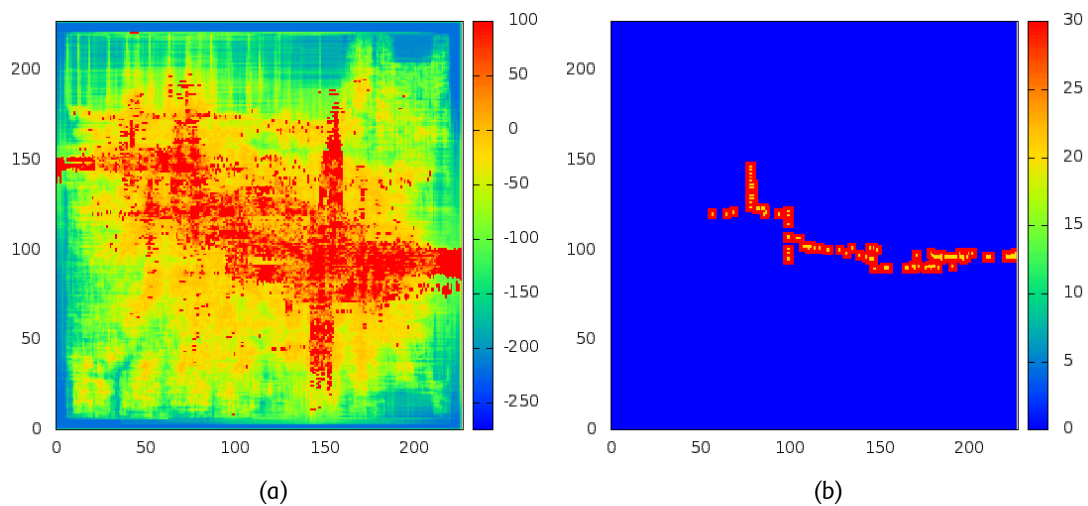


Figura 6.8: Mapa de cores representando o congestionamento detalhado no circuito bigblue1 (a). Áreas de alto congestionamento, identificadas após as fases de análise e expansão (b).

que a cada rodada de roteamento, esse número é novamente calculado, diferentemente do propósito do parâmetro de custo histórico, cujo valor nunca é decrementado.

Os nodos do circuito que estiverem em uma área de alto congestionamento, tem seu custo de penalização multiplicado por 1,8 (incremento de 80%), de modo incentivar com que os fios busquem outros caminhos menos custosos para fazer seu traçado.

Os vizinhos expandidos tem seus custos incrementados usando uma função exponencial em relação à distância do nodo com alto nível de congestionamento.

## 6.4 Resultados Experimentais

Nesta seção é feita a apresentação e discussão dos resultados obtidos com a aplicação das técnicas de delimitação e tratamento de áreas com alta demanda por interconexão, aplicadas ao fluxo do GR-RT e GR-WL.

As simulações foram executadas em uma máquina rodando sistema operacional Linux 2.6, equipada com oito núcleos Intel Xeon(R) E5520 (2.27GHz), 6GB de memória RAM, usando o compilador GCC, versão 4.4.3.

Os cenários utilizados para validação do impacto das técnicas na solução final de roteamento global foram feitos através da execução da ferramenta GR-C, utilizando as três abordagens propostas para espalhamento de custo histórico, em relação aos dois métodos de decomposição de redes suportados pelo GR: MST e FLUTE.

Foram utilizados os 17 circuitos do *benchmark* ISPD 2008, descritos nas tabelas 2.1 e 2.2.

Em cada execução dos *benchmarks*, foram coletadas as seguintes métricas:

1. Comprimento total dos fios (WL);
2. Máximo valor de congestionamento, ocorrido na solução (MOF);
3. Soma de todos os valores de congestionamento obtidos (TOF); e
4. Tempo de CPU utilizado pelo processo de roteamento.

Os resultados da execução do GR-C nos *benchmarks* ISPD 2008, para os três modelos de espalhamento de congestionamento são apresentados na tabela 6.1 (estático, linear e exponencial negativo), utilizando MST para decomposição das redes. A tabela 6.2 apresenta a execução dos mesmos circuitos, porém utilizando o FLUTE para geração das topologias.

Na execução usando MST, é possível verificar que a implementação usando modelo estático obteve os piores resultados para a métrica de comprimento de fio, enquanto que a função exponencial negativa obteve, em média, os melhores resultados. Cabe salientar, no entanto, que a diferença máxima entre o pior e o melhor comprimento de fio, não ultrapassa 2% do valor total desta métrica.

A análise das duas métricas relacionadas ao congestionamento (TOF, que se refere ao congestionamento total e MOF, que se refere ao congestionamento máximo), evidencia o potencial das três técnicas para redução do congestionamento máximo, principalmente nos circuitos para os quais ainda não se tem uma solução válida na literatura, quando comparados com a implementação original do GR, pelo fato de que todas as técnicas empregadas aplicam um incremento em custo histórico, antes mesmo do início do processo de roteamento iterativo.

É possível observar que um incremento em custo histórico antes começo da fase de roteamento iterativo, mesmo que de valor pequeno, já causa um impacto positivo na geração da solução inicial, visto que as redes localizadas em locais potencialmente congestionados já começam essa etapa de roteamento com seu custo calibrado levemente acima das outras regiões do circuito.

O impacto em nodos vizinhos quando há muitos pontos de alta demanda por interconexão próximos, evidencia a habilidade do modelo exponencial negativo em proteger os pontos de congestionamento ao mesmo tempo em que atenua o congestionamento em nodos vizinhos, reduzindo o efeito colateral de criação de ilhas baixo congestionamento, observado nos resultados dos métodos de distribuição estático e o linear.

Desta forma, o modelo exponencial negativo consegue reduzir o congestionamento máximo obtido, obtendo até 10% de vantagem sobre os outros dois métodos, com destaque para os circuitos que possuem maiores quantitativos de congestionamento (newblue3 e newblue7), onde foram observados os maiores ganhos nesta métrica.

Nos experimentos realizados usando o FLUTE para decomposição das redes em redes de dois pinos, foi possível verificar o impacto das arestas de conexão criadas pelo FLUTE para decompor a rede.

Por um lado, as arestas aumentam a granularidade do processo de roteamento, em alguns casos reduzem tempo total gasto para geração da solução, como pode ser observado na solução dos circuitos adaptec4, newblue1, newblue2 e newblue7. Por outro lado, as arestas de conexão podem representar barreiras artificiais, que nos circuitos que são mais compactos e naturalmente congestionados, podem prejudicar o processo de eliminação de congestionamento, como pode ser observado no circuito newblue3, newblue4, newblue5 e newblue7.

Na análise dos modelos de distribuição de custo usando FLUTE, foram verificados comportamentos semelhantes aos resultados usando MST. Em termos de comprimento dos fios, o modelo estático, que faz o incremento mais agressivo nos nodos vizinhos, entre os três modelos testados, acaba incrementando desnecessariamente o comprimento dos fios, em relação as outras duas técnicas. Apesar disso, a diferença entre os melhores e os piores resultados de comprimento de fio, não ultrapassa 3%.

O modelo exponencial negativo obteve os melhores resultados de comprimento de fio (exceto nos circuitos bigblue1 e newblue3). Considerando os resultados obtidos com as três distribuições, optou-se pela utilização deste modelo como sendo a implementação de referência do GR-C, para obtenção dos resultados de roteamento e comparação com outras ferramentas.

Tabela 6.1: Resultados da implementação **técnicas de delimitação** e tratamento de áreas com alta demanda de conexão, no GR-C, com custo de via nulo. A decomposição das redes foi feita usando MST.

Circuito	Modelo Estático				Modelo Linear				Modelo Exponencial			
	WL	TOF	MOF	Tempo	WL	TOF	MOF	Tempo	WL	TOF	MOF	Tempo
adaptec1	3684730	0	0	35,8	3688964	0	0	37,6	3687170	0	0	41,7
adaptec2	3367414	0	0	25,6	3373886	0	0	16,5	3353751	0	0	13,1
adaptec3	9678027	0	0	66,2	9674767	0	0	41,5	9672203	0	0	47,5
adaptec4	9018618	0	0	17,2	9018698	0	0	11,9	9017759	0	0	10,5
adaptec5	10471716	0	0	216,1	10466296	0	0	117,2	10449972	0	0	103,1
bigblue1	3859189	0	0	201,1	3854539	0	0	167,5	3853711	0	0	182,1
bigblue2	5392124	90	2	476,1	5823638	124	2	281,2	5179736	102	2	229,6
bigblue3	7970535	0	0	126,7	7990625	0	0	110,1	7976437	0	0	61,5
bigblue4	12083140	908	8	478,5	12216332	950	8	260,2	12200150	866	8	161,7
newblue1	2512976	582	2	56,5	2536596	482	2	109,1	2510780	504	2	108,3
newblue2	4674998	0	0	3,1	4675327	0	0	3,2	4674438	0	0	3,5
newblue3	7710930	36964	998	1413,5	7741509	36280	1132	1231,5	7719207	37382	608	1406,1
newblue4	8773794	454	2	284,5	8539176	486	2	165,6	8584383	440	2	213,2
newblue5	15126175	0	0	304,6	15349297	0	0	352,1	15244712	0	0	182,5
newblue6	10524904	0	0	203,1	10524448	0	0	188,1	10442837	0	0	176,4
newblue7	18686500	5390	4	729,7	18718713	4888	4	1408,3	18622241	4620	4	1138,2

Tabela 6.2: Resultados da implementação **técnicas de delimitação** e tratamento de áreas com alta demanda de conexão, no GR-C, com custo de via nulo. A decomposição das redes foi feita usando FLUTE.

Circuito	Modelo Estático				Modelo Linear				Modelo Exponencial			
	WL	TOF	MOF	Tempo	WL	TOF	MOF	Tempo	WL	TOF	MOF	Tempo
adaptec1	3735231	0	0	129,1	3729919	0	0	80,9	3687708	0	0	70,5
adaptec2	3386751	0	0	55,2	3371359	0	0	24,9	3366929	0	0	29,5
adaptec3	9636071	0	0	70,5	9630675	0	0	44,7	9609489	0	0	38,5
adaptec4	8927844	0	0	17,5	8927534	0	0	10,5	8925680	0	0	6,6
adaptec5	10547200	0	0	262,8	10571935	0	0	187,7	10461568	0	0	175,6
bigblue1	3887140	0	0	491,5	3875698	0	0	407,4	3873552	0	0	390,2
bigblue2	5354123	694	4	381,4	5363201	638	4	339,9	5200232	662	4	233,9
bigblue3	7949645	52	2	180,3	7943978	66	2	230,5	7926125	50	2	236,6
bigblue4	12222004	1472	36	417,5	12244851	879	32	409,0	12064981	2108	12	198,9
newblue1	2450453	1876	4	104,6	2443547	1878	4	71,5	2432417	1754	4	86,2
newblue2	4623772	0	0	3,3	4623643	0	0	3,4	4623067	0	0	3,2
newblue3	7754279	35210	1192	1410,1	7719811	36238	1168	1074,2	7819473	36398	1126	1404,1
newblue4	8449052	1526	4	300,8	8350658	1686	4	250,4	8333030	1566	4	256,3
newblue5	14880546	1856	8	473,6	14921388	1778	6	603,9	14834297	1788	6	508,1
newblue6	10678085	0	0	326,3	10666044	0	0	303,5	10479528	0	0	196,9
newblue7	18632023	7694	6	683,3	18640651	7868	6	785,2	18505708	6724	6	672,1

A tabela 6.3, apresenta a comparação da implementação de referência do GR-C (usando MST), com a implementação original do GR-WL e NTUgr, que aplica técnicas de identificação de áreas de alto congestionamento, considerando custo de via nulo. Isso faz com que as trocas de camadas sejam ignoradas, facilitando a análise de incidência de vias, nos resultados que serão apresentados posteriormente.

Na tabela 6.4 são apresentados os resultados usando o FLUTE para decomposição das redes. Esses resultados consideram o custo de via igual a zero.

Nas tabelas 6.5 e 6.6 são apresentados os resultados do GR-C, em comparação com a implementação de referência do GR-WL e do NTUgr, porém, com custo de via unitário, onde pode ser observado a capacidade da ferramenta em gerar congestionamento menor, tanto em quantidade de congestionamento quanto em máximo congestionamento obtido, com uma diferença, em média de 10% de acréscimo em tempo de execução.

Pode ser verificado, também, um aumento do comprimento dos fios, entre a solução obtida pelo GR-C e a implementação original do GR-WL. Neste caso, o aumento se justifica pela capacidade de redução de congestionamento, tendo impacto no comprimento dos fios, que necessitam realizar desvios maiores para serem roteados sem congestionamento.

Nos circuitos onde não há uma solução final para o problema, é possível verificar uma redução de até 16% no congestionamento total (nos circuitos bigblue2, newblue1, newblue4 e newblue7) e até 49% no valor do máximo congestionamento obtido (especificamente no circuito newblue3). Neste caso específico do newblue3, houve um aumento no valor total de congestionamento do circuito. Esse fato evidencia que o pré-incremento de custo histórico tem potencial para redução do congestionamento.

Analisando a métricas de número de vias, realizada a partir da variação entre os dados das tabelas 6.3 e 6.4 com 6.5 e 6.6, respectivamente, foi possível observar um aumento máximo de 1% em relação a implementação original do GR-WL. Essa análise foi feita através da observação da variação proporcional entre as soluções obtidas com a mesma ferramenta, a partir dos dados presentes na tabela que considera custo de via nulo e a tabela que considera custo unitário de via.

Tabela 6.3: Resultados da implementação de referência do GR-C (modelo exponencial), com técnicas de delimitação e tratamento de áreas com alta demanda de conexão, com **custo de via nulo**. Decomposição das redes feita com MST.

Circuito	NTUgr				GR-WL (MST)				GR-C (MST)			
	WL	TOF	MOF	Tempo	WL	TOF	MOF	Tempo	WL	TOF	MOF	Tempo
adaptec1	3715732	0	0	46,8	3672608	0	0	46,2	3687170	0	0	41,7
adaptec2	3411210	0	0	7,6	3354386	0	0	15,3	3353751	0	0	13,1
adaptec3	9879292	0	0	37,8	9677595	0	0	45,4	9672203	0	0	47,5
adaptec4	9119619	0	0	14,3	9026091	0	0	9,4	9017759	0	0	10,5
adaptec5	10699866	0	0	110,9	10452584	0	0	120,9	10449972	0	0	103,1
bigblue1	3903183	0	0	136,2	3826593	0	0	165,9	3853711	0	0	182,1
bigblue2	5013784	118	4	215,9	5237687	122	2	281,3	5179736	102	2	229,6
bigblue3	8037384	0	0	29,9	7940693	0	0	56,3	7976437	0	0	61,5
bigblue4	12478456	410	10	313,1	12075198	780	6	197,1	12200150	908	8	161,7
newblue1	2491986	212	4	157,5	2486311	588	2	89,8	2510780	504	2	108,3
newblue2	4753628	0	0	4,3	4681756	0	0	5,4	4674438	0	0	3,5
newblue3	11005863	33636	374	183,1	7683682	36280	1194	1416,8	7719207	37382	608	1406,1
newblue4	8517969	284	4	254,5	8567326	462	2	323,5	8584383	440	2	213,2
newblue5	15296919	0	0	117,9	15045883	0	0	283,8	15244712	0	0	182,5
newblue6	10561501	0	0	72,6	10420794	0	0	164,6	10442837	0	0	176,4
newblue7	37485508	906	6	1421,0	18617528	5172	4	974,2	18622241	4620	4	1138,2

Tabela 6.4: Resultados da implementação de referência do GR-C (modelo exponencial), com técnicas de delimitação e tratamento de áreas com alta demanda de conexão, com **custo de via nulo**. Decomposição das redes feita com FLUTE.

Circuito	NTUgr				GR-WL (FLUTE)				GR-C (FLUTE)			
	WL	TOF	MOF	Tempo	WL	TOF	MOF	Tempo	WL	TOF	MOF	Tempo
adaptec1	3715732	0	0	46,8	3683810	0	0	65,9	3687708	0	0	70,5
adaptec2	3411210	0	0	7,6	3346596	0	0	30,7	3366929	0	0	29,5
adaptec3	9879292	0	0	37,8	9605438	0	0	44,7	9609489	0	0	38,5
adaptec4	9119619	0	0	14,3	8932222	0	0	9,5	8925680	0	0	6,6
adaptec5	10699866	0	0	110,9	10430261	0	0	161,2	10461568	0	0	175,6
bigblue1	3903183	0	0	136,2	3836507	0	0	368,7	3873552	0	0	390,2
bigblue2	5013784	118	4	215,9	5089263	776	4	392,9	5200232	662	4	233,9
bigblue3	8037384	0	0	29,9	7951793	0	0	312,5	7926125	50	2	236,6
bigblue4	12478456	410	10	313,1	12053813	1546	30	306,5	12064981	2108	12	198,9
newblue1	2491986	212	4	157,5	2422752	1766	4	120,5	2432417	1754	4	86,2
newblue2	4753628	0	0	4,3	4628547	0	0	5,0	4623067	0	0	3,2
newblue3	7785823	33636	374	183,1	7666336	35242	1230	876,9	7819473	36398	1126	1404,1
newblue4	8517969	284	4	254,5	8321997	1456	4	247,0	8333030	1566	4	256,3
newblue5	15296919	0	0	117,9	14839333	1576	8	612,8	14834297	1788	6	508,1
newblue6	10561501	0	0	72,6	10443840	0	0	216,7	10479528	0	0	196,9
newblue7	19290419	906	6	1421,0	18511502	6888	6	803,4	18505708	6724	6	672,1



Tabela 6.5: Resultados da implementação de referência do GR-C (modelo exponencial), com técnicas de delimitação e tratamento de áreas com alta demanda de conexão, com **custo de via unitário**. Decomposição das redes feita com MST.

Circ.	NTUgr				GR-WL (MST)				GR-C (MST)			
	WL	TOF	MOF	Tempo	WL	TOF	MOF	Tempo	WL (var. GR-WL)	TOF	MOF	Tempo
ad1	5609943	0	0	46,8	6380538	0	0	46,2	6418268 (+0,59%)	0	0	41,7
ad2	5421602	0	0	7,6	6084400	0	0	15,3	6101664 (+0,28%)	0	0	13,1
ad3	13649616	0	0	37,8	15155118	0	0	45,4	15182090 (+0,17%)	0	0	47,5
ad4	12408382	0	0	14,3	13981761	0	0	9,4	13992148 (-0,07%)	0	0	10,5
ad5	16358988	0	0	110,9	18102395	0	0	120,9	18149818 (+0,26%)	0	0	103,1
bb1	5949946	0	0	136,2	6685407	0	0	165,9	6741135 (+0,83%)	0	0	182,1
bb2	94667229	118	4	215,9	11204002	122	2	281,3	11181906 (-0,2%)	138	2	229,6
bb3	13491167	0	0	29,9	15594684	0	0	56,3	15688085 (+0,59%)	0	0	61,5
bb4	24049722	410	10	313,1	27602383	780	6	197,1	27481693 (+0,56%)	1090	8	161,7
nb1	4824320	212	4	157,5	5437199	588	2	89,8	5495484 (+1,07%)	504	2	108,3
nb2	7769446	0	0	4,3	9245356	0	0	5,4	9266090 (+0,22%)	0	0	3,5
nb3	11005863	33636	374	183,1	13316366	36280	1194	1416,8	13218553 (+0,73%)	37382	608	1406,1
nb4	13442722	284	4	254,5	15329779	462	2	323,5	15410331 (+0,52%)	460	2	213,2
nb5	24021512	0	0	117,9	26983599	0	0	283,8	27360317 (+1,39%)	0	0	182,5
nb6	18735225	0	0	72,6	20708074	0	0	164,6	20769146 (+0,29%)	0	0	176,4
nb7	37485508	906	6	1421,0	42224489	5172	4	974,2	42503107 (+0,65%)	4620	4	1138,2

Tabela 6.6: Resultados da implementação de referência do GR-C (modelo exponencial), com técnicas de delimitação e tratamento de áreas com alta demanda de conexão, com **custo de via unitário**. Decomposição das redes feita com FLUTE.

Circ.	NTUgr				GR-WL (FLUTE)				GR-C (FLUTE)			
	WL	TOF	MOF	Tempo	WL	TOF	MOF	Tempo	WL (var. GR-WL)	TOF	MOF	Tempo
ad1	5609943	0	0	46,8	6371496	0	0	65,9	6394310 (+0,35%)	0	0	70,5
ad2	5421602	0	0	7,6	6054036	0	0	30,7	6098328 (+0,73%)	0	0	29,5
ad3	13649616	0	0	37,8	15023535	0	0	44,7	15052401 (+0,19%)	0	0	38,5
ad4	12408382	0	0	14,3	13791352	0	0	9,5	13803184 (+0,08%)	0	0	6,6
ad5	16358988	0	0	110,9	18007166	0	0	161,2	18109075 (+0,56%)	0	0	175,6
bb1	5949946	0	0	136,2	6674218	0	0	368,5	6748879 (+1,11%)	0	0	390,2
bb2	94667229	118	4	215,9	10946954	776	4	392,9	11191192 (+2,23%)	700	2	233,9
bb3	13491167	0	0	29,9	15538857	0	0	312,5	15540453 (+0,01%)	50	2	236,6
bb4	24049722	410	10	313,1	27315327	1546	30	306,5	27390471 (+0,27%)	2108	12	198,9
nb1	4824320	212	4	157,5	5309688	1766	4	120,5	5338103 (+0,53%)	1754	4	86,2
nb2	7769446	0	0	4,3	9123742	0	0	5,0	9143611 (+0,21%)	0	0	3,2
nb3	11005863	33636	374	183,1	13059064	35242	1230	876,9	13320170 (+1,98%)	36398	1126	1404,1
nb4	13442722	284	4	254,5	14905590	1456	4	247,0	14963422 (+0,38%)	1566	4	256,3
nb5	24021512	0	0	117,9	26563051	1576	8	612,8	26628291 (+0,24%)	1788	6	508,1
nb6	18735225	0	0	72,6	20702529	0	0	216,7	20783201 (+0,38%)	0	0	196,9
nb7	37485508	906	6	1421,0	41821161	6888	6	803,35	42153902 (+0,79%)	6724	6	672,1

## 7 CONCLUSÃO

A partir da avaliação do fluxo das ferramentas de roteamento global desenvolvidas nos últimos anos, pode-se traçar um perfil de execução comum do roteamento global, através do qual as ferramentas foram analisadas no âmbito deste trabalho.

Este trabalho apresentou a implementação e avaliação de duas técnicas para identificação e tratamento de áreas com alta demanda por interconexão e alto nível de congestionamento: pré incremento de custo histórico durante o roteamento inicial e tratamento de alto nível de congestionamento durante o roteamento iterativo.

Para as áreas identificadas com alta demanda por interconexão, foram descritos os parâmetros para aplicação de pré-incremento de custo e realizados experimentos com três diferentes funções para distribuição de custos entre os nodos vizinhos: estático, linear e exponencial negativo.

A importância da etapa de estimativa de congestionamento sendo realizada antes do começo do roteamento propriamente dito e durante as rodadas do roteamento iterativo é reforçada em diversos trabalhos como NTUgr, GR-WL e Archer.

Através da implementação e realização de experimentos com as técnicas propostas, foi possível verificar que elas proporcionam um ganho em redução de congestionamento total em até 16%, em relação a implementação original do GR-WL, sendo os maiores ganhos percebidos nos circuitos de *benchmark* que ainda não possuem soluções válidas na literatura (*newblue7*, por exemplo).

Além disso, as técnicas de delimitação de áreas congestionadas mostraram-se efetivas para reduzir em até 35.7% (em média 6.30%) o tempo de execução necessário para realizar o roteamento global. Isso representa um avanço no sentido de viabilizar uma melhoria na integração deste fluxo, tanto com fluxos de roteamento detalhado quanto com fluxos de posicionamento, gerando estimativas da distribuição de focos de congestionamento na solução de circuito posicionada.

Em alguns experimentos, foi possível observar um aumento no comprimento dos fios de até 1,39% na versão que realiza a construção de topologias usando MST e 2,23% na versão FLUTE, em relação a implementação original do GR-WL, que se justifica pelo espalhamento das conexões que viabilizaram a redução do congestionamento nessas situações.

A partir da implementação dessas técnicas, emerge a oportunidade para desenvolvimento de diversos trabalhos futuros, tanto para as técnicas de identificação e tratamento de áreas com alta demanda por interconexão, quanto para o fluxo de roteamento global baseado em GR-WL.

Uma oportunidade de trabalho futuro é o desenvolvimento de um algoritmo para calibração dos parâmetros de distribuição de custo histórico, o qual poderia, dinamicamente, definir a amplitude e o valor do custo histórico a ser distribuído, dependendo da região

onde o congestionamento está concentrado. Essa implementação poderia tornar mais expressivos os ganhos obtidos com a identificação de congestionamento nas fases iniciais do roteamento global.

No sentido de melhorar os resultados de comprimento de fio para os experimentos onde é feita a decomposição das redes através do FLUTE, existe a oportunidade para implementação de um algoritmo que faça a movimentação ou exclusão das arestas adicionais, que não fazem parte da rede original, quando estas se encontrarem em áreas congestionadas. Essa implementação tem potencial para aumentar o reaproveitamento de fios de uma mesma rede e liberando meios de interconexão para que outras redes possam ter seu tamanho reduzido.

Explorar o uso de paralelismo durante o roteamento iterativo consiste em outra possibilidade de melhoria. Essa abordagem poderia reduzir o tempo de execução da ferramenta, enquanto mantém a qualidade dos resultados.

O desenvolvimento de uma ferramenta para comparação detalhada de suas soluções de roteamento que tenham saída de dados compatível com os *benchmarks* ISPD 2008, pode contribuir para verificação dos ganhos decorrente da aplicação de ganhos obtidos com a aplicação de diferentes técnicas nos fluxos de roteamento global.

## REFERÊNCIAS

ALBRECT, C. A provably good global routing by a new approximation algorithm for multicommodity flow. In: IEEE/ACM INTERNATIONAL SYMPOSIUM ON PHYSICAL DESIGN, 2000, San Diego, CA, EUA. **Proceedings...** New York: ACM, 2000. p. 19–25.

ALPERT, C. J.; MEHTA, D. P.; SAPATNEKAR, S. S. **Handbook of Algorithms for physical design automation**. Boston, MA, EUA: Auerbach Publications, 2008. 1044p.

BURSTEIN, M.; PELAVIN, R. N. Hierarchical Wire Routing. In **IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems**, [S.I.], v.2, n.4, p. 223–234, Outubro, 1983.

CAO, Z.; JING, T.; XIONG, J.; HU, Y.; HE, L.; HONG, X. DPRouter: A fast and accurate dynamic-pattern-based global routing algorithm. In: IEEE/ACM ASIA AND SOUTH PACIFIC DESIGN AUTOMATION CONFERENCE, 2007, Pacifico Yokohama, Yokohama, Japão. **Proceedings...**[S.I.]: IEEE, 2007. p. 256–261.

CHANG, S.-C. **Parallel Algorithms for Several VLSI Routing Problems**. 1988. 148p. Tese (Doutorado em Engenharia Elétrica) – University of Maryland, EUA. 1988.

CHANG, Y.-J.; LEE, Y.T.; WANG, T.-C. NTHU-Route 2.0: a fast and stable global. In: IEEE/ACM INTERNATIONAL CONFERENCE ON COMPUTER-AIDED DESIGN, 2008, San Jose, CA, EUA. **Proceedings...**[S.I.]: IEEE, 2008, p. 338–343.

CHANG, Y.-J.; LEE, Y.T.; WANG, T.-C. GLADE: a modern global router considering layer directives. In: IEEE/ACM INTERNATIONAL CONFERENCE ON COMPUTER-AIDED DESIGN, 2010, San Jose, CA, EUA. **Proceedings...**[S.I.]: IEEE, 2010. p. 319–323.

CHEN, T.-C.; CHANG, Y.-W.; LIN, S.-C. A novel Framework for Multilevel Full-Chip Gridless Routing. In: IEEE/ACM ASIA AND SOUTH PACIFIC DESIGN AUTOMATION CONFERENCE, 2006, Yokohama, Japão. **Proceedings...**[S.I.]: IEEE, 2006. p. 636–641.

CHU, C. FLUTE: Fast Lookup Table Based Wirelength Estimation Technique. In: IEEE/ACM INTERNATIONAL CONFERENCE ON COMPUTER AIDED DESIGN, 2004, San Jose, CA, EUA. **Proceedings...** New York: ACM, 2004. p.696–701.

CHU, C; WONG, Y.-C. Fast and Accurate Rectilinear Steiner Minimal Tree Algorithm for VLSI Design. In: INTERNATIONAL SYMPOSIUM ON PHYSICAL DESIGN, 2005, San Francisco, CA, EUA. **Proceedings...** New York: ACM, 2005. p.28–35.

CHO, M; PAN, D. Z. BoxRouter: A new global router based on box expansion and progressive ILP. In: IEEE/ACM DESIGN AUTOMATION CONFERENCE, 2006 San Francisco, CA, EUA. **Proceedings...** New York: ACM, 2006. p.373–378.

CHO, M; LU, K.; YUAN, K.; PAN, D. Z.. BoxRouter 2.0: architecture and implementation of a hybrid and robust global router. In: IEEE/ACM INTERNATIONAL CONFERENCE ON COMPUTER-AIDED DESIGN, 2007, San Jose, CA, EUA. **Proceedings...** New York: ACM, 2007. p. 503–508.

CONG, J.; LEI, H.; KOH, C.-K.; MADDEN, P. Performance optimization of VLSI interconnect layout. **Integration, the VLSI Journal**, [S.I.], v.21, n.1–2, p.1–94, Novembro, 1996.

CONG, J.; XIE, M.; ZHANG, Y. MARS: A multilevel full chip gridless routing system. **IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems**, [S.I.], v.24, n.3, p. 382–394, Março, 2005.

DICKSON, C. **Global Routing in VLSI: Algorithms, Theory and Computation**. 2007. 76p. Dissertação (Mestrado em Ciência da Computação) – McMaster University, Hamilton, Ontario, Canadá. 2007.

DIJKSTRA, E. W. A note on two problems in connexion with graphs. **Numerische Mathematik**, [S.I.], v.1, n.1, p. 269–271, Dezembro, 1959.

HADLOCK, F.O. A shortest path algorithm for grid graphs. **Networks**, [S.I.], v.7, n.4, p. 323–334, Junho, 1977.

HASHIMOTO, A; STEVENS, J. Wire routing by optimizing channel assignment within large apertures. In: 8TH DESIGN AUTOMATION WORKSHOP, 1971. **Proceedings...** [S.I.]: ACM, 1971. p. 155–169.

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.I.], v.4, n.2, p.100–107, Julho, 1968.

HER, T.W.; WONG, D.F. On over-the-cell channel routing with cell orientations consideration. **IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems**. [S.I.], v.14, n.6, p. 766–772, Junho, 1995.

HIGHTOWER, D. W. A solution to line-routing problem on the continuous plane. In: 6TH ANNUAL CONFERENCE ON DESIGN AUTOMATION, 1969. **Proceedings...** [S.I.], 1969, p. 1–24.

HU, J.; ROY, J. A; MARKOV, I. L. Sidewinder: a scalable ILP-based router. In: ACM/IEEE SYSTEM LEVEL INTERCONNECT PREDICTION, 2008, New Castle, Reino Unido. **Proceedings...** [S.I.:s.n.], 2008. p. 73–80.

HU, J.; SAPATNKEAR, S. S. A survey on multi-net global routing for integrated circuits. **Integration: the VLSI Journal**, [S.I.], v.31, n.1, p.1–49, 2001.

HU, J.; ROY, J. A.; MARKOV, I. L. Completing high-quality global routes. In: IEEE/ACM INTERNATIONAL SYMPOSIUM ON PHYSICAL DESIGN, 2010, San Francisco, CA, EUA. **Proceedings...** New York: ACM, 2010. p. 35–41.

HSU, C.-H.; CHEN, H.-Y.; CHANG, Y.-W. Multi-layer Global Routing Considering Via and Wire Capacities. In: IEEE/ACM INTERNATIONAL CONFERENCE ON COMPUTER-AIDED DESIGN, 2008, San Jose, CA, EUA. **Proceedings...** New York: ACM, 2008. p. 350–355.

HSU, C.-H.; CHEN, H.-Y.; CHANG, Y.-W. Multilayer Global Routing With Via and Wire Capacity Considerations. **IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems**, [S.I.], v.29, n.5, p. 685–696, Maio, 2010.

JOHANN, M. d. O. New Bidirectional Heuristic Shortest Path Search Algorithm. In: INTERNATIONAL ICSC CONGRESS ON ARTIFICIAL INTELLIGENCE AND APPLICATIONS, 2000. **Proceedings...**, [S.I.:s.n.], 2000.

JOHANN, M. d. O. **Novos algoritmos para roteamento de circuitos VLSI**. 2001. 158p. Tese (Doutorado em Ciência da Computação) – Instituto de Informática, Universidade Federal do Rio Grande do Sul, Porto Alegre. 2001.

KASTNER, R.; BOZORGZADEH, E.; SARRAFZADEH, M. Pattern Routing: use and theory for increasing predictability and avoid coupling. **IEEE Transactions on Computer Aided Design of Integrated Circuits and Systems**, [S.I.], v.21, n.7, p. 771–791, Julho, 2002.

KHANNA, S.; GAO, S.; THULASIRAMAN, K. Parallel hierarchical global routing for general cell layout. In: IEEE/ACM GREAT LAKES SYMPOSIUM ON VLSI, 1995, Washington, DC, EUA. **Proceedings...** [S.I.]: IEEE, 1995. p. 212–215.

LEE, C. Y. An algorithm for path connection and its applications. **IEEE Transactions on Electronic Computers**, [S.I.], v.EC–10, n.3, p. 374–365, Setembro, 2010.

LEISERSON, C. E; PINTER, R. Y. Optimal Placement for River Routing. **SIAM Journal on Computing**, [S.I.], v.12, n.3, p. 447–462, Agosto, 1983.

LIU, W.-H.; KEO, W.-C.; LI, Y.-L.; CHAO, K.-Y. Multi-threaded collision-aware global routing with bounded-length maze routing. In: IEEE/ACM DESIGN AUTOMATION CONFERENCE, 2010, Anaheim, CA, EUA. **Proceedings...** [S.I.]: IEEE, 2010. p. 200–205.

MCMURCHIE, L.; EBELING, C. PathFinder: a negotiation-based performance-driven router for FPGAs. In: ACM INTERNATIONAL SYMPOSIUM ON FIELD PROGRAMMABLE GATE ARRAYS, 1995. **Proceedings...** New York: ACM, 1995. p. 111–117.

MCPHERSON, J. W. Reliability challenges for 45nm and beyond. In: IEEE/ACM DESIGN AUTOMATION CONFERENCE, 2006, San Francisco, CA, EUA. **Proceedings...** New York: ACM, 2010. p. 176–181.

MIKAMI, K.; TABUCHI, K.. A computer program for optimal routing of printed circuit connectors. In: INFORMATION PROCESSING 68, 1968, Edinburgh, Reino Unido. **Proceedings...** [S.I.:s.n.], 1968. p.1475–1478.

MOFFITT, M. D.; ROY, J. A. and MARKOV, I. L. The coming of age of (academic) global routing. In: INTERNATIONAL SYMPOSIUM ON PHYSICAL DESIGN, 2008, Portland, OR, EUA. **Proceedings...** New York: ACM, 2008. p. 148–155.

MOFFITT, M. D. MaizeRouter: Engineering an effective global router. **IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems**, [S.I.], v.27, n.11, p. 2017–2026, Novembro, 2008.

NAM, G.-J.; SZE, C.; YILDIZ, M. The ISPD global routing benchmark suite. In: IEEE/ACM INTERNATIONAL SYMPOSIUM ON PHYSICAL DESIGN, 2008, Portland, OR, EUA. **Proceedings...** New York: ACM, 2008. p. 156–159.

OZDAL, M.; WONG, M. Archer: a history-driven global routing algorithm. In: IEEE/ACM INTERNATIONAL CONFERENCE ON COMPUTER AIDED DESIGN, 2007, San Jose, CA, EUA. **Proceedings...** [S.I.]: IEEE, 2007. p. 488–495.

PAN, M.; CHU, C.. FastRoute 2.0: a high-quality and efficient global router. In: IEEE/ACM ASIA AND SOUTH PACIFIC DESIGN AUTOMATION CONFERENCE, 2007, Pacifico Yokohama, Yokohama, Japão. **Proceedings...** [S.I.]: IEEE, 2007. p.250–255.

PALUSZEWSKI, M.; WINTER, P.; ZACHARIASEN, M. A new paradigm for general architecture routing. In: ACM GREAT LAKES SYMPOSIUM ON VLSI, 2004, Boston, MA, EUA. **Proceedings...** New York: ACM, 2004. p. 202–207.

RAGHAVAN, P.; THOMPSON, C. D. Multiterminal global routing: A deterministic approximation scheme. **Algorithmica**, [S.I.], v.6, n.1, p. 73–82, Junho, 1991.

REIMANN, T. **Roteamento Global de Circuitos VLSI**. 2011. 109p. Dissertação (Mestrado em Microeletrônica) – Instituto de Informática, Universidade Federal do Rio Grande do Sul, Porto Alegre. 2011.

ROY, J. A.; MARKOV, I. High-performance routing at the nanometer scale. **Transactions on Computer-Aided Design of Integrated Circuits and Systems**, [S.I.], v.27, n.6, p. 1066–1077, June, 2008.

SARRAFZADEH, M.; WONG, C. K. **An Introduction to VLSI Physical Design**. [S.I.]: McGraw-Hill Higher Education, 1996. 416p.

SHERWANI, N. A. **Algorithms for VLSI Physical Design Automation**. Norwell, MA, EUA: Kluwer Academic Publishers, 1995. 538p.

SHRAGOWITZ, E.; KEEL, S. A global router based on a multicommodity flow model. **Integration: the VLSI Journal**, [S.I.], v.5, n.1, p. 3–16, Março, 1987.

TEIG, S. L. The X architecture: not your father's diagonal wiring. In: INTERNATIONAL WORKSHOP ON SYSTEM-LEVEL INTERCONNECT PREDICTION, 2002, San Diego, CA, EUA. **Proceedings...** New York: ACM, 2002. p. 33–37.

WESTRA, J.; GROENVELD, P.; YAN, T.; MADDEN, P. H.. Global routing: Metrics, benchmarks, and tools. In: IEEE DATC ON ELECTRONIC DESIGN PROCESS, 2005, Monterey, CA, EUA. **Proceedings...** [S.I.:s.n.], 2005.

WANG L.-T.; CHANG, Y.-W.; CHENG, K.-T. **Electronic Design Automation: Synthesis, Verification, and Testing**. [S.I.]: Elsevier/Morgan Kaufmann, 2008. 972p.



WU, T.-H.; DAVOODI, A.; LINDEROTH, J. GRIP: scalable 3D global routing using integer programming. In: IEEE/ACM DESIGN AUTOMATION CONFERENCE, 2009, San Francisco, CA, EUA. **Proceedings...** New York: ACM, 2009. p. 320–325.

WU, T.-H.; DAVOODI, A.; LINDEROTH, J. A Parallel Integer Programming Approach to global routing. In: IEEE/ACM DESIGN AUTOMATION CONFERENCE, 2010, Anaheim, CA, EUA. **Proceedings...** New York: ACM, 2010. p. 194–199.

WU, T.-H. **A parallel Integer Programming Approach to Global Routing**. 2011. 119p. Tese (Doutorado em Engenharia da Computação) – Universidade do Winsconsin, Madison, EUA. 2011.

XU, Y.; CHU, C. MGR: Multi-Level Global Router. In: IEEE/ACM INTERNATIONAL CONFERENCE ON COMPUTER-AIDED DESIGN, 2011, San Jose, CA, EUA. **Proceedings...** [S.I.]: IEEE, 2011. p. 250–255.

XU, Y.; ZHANG, Y.; CHU, C. Fastroute 4.0: global router with efficient via minimization. In: IEEE/ACM ASIA AND SOUTH PACIFIC DESIGN AUTOMATION CONFERENCE, 2009, Pacifico Yokohama, Yokohama, Japão. **Proceedings...** [S.I.]: IEEE, 2009. p. 576–581.

XIE, Y.; CONG, J.; SAPATNEKAR, S. **Three-Dimensional Integrated Circuit Design**. **Three-Dimensional Integrated Circuit Design: EDA, Design and Microarchitectures**. [S.I.]: Springer, 2010. 284p.

ZHANG, Y.; XU, Y.; CHU, C. Fastroute 3.0: A fast and High Quality Global Router Based on Virtual Capacity. In: IEEE/ACM INTERNATIONAL CONFERENCE ON COMPUTER-AIDED DESIGN, 2008, San Jose, CA, EUA. **Proceedings...** [S.I.]: IEEE, 2008. p. 344–349.

ZHOU, H.; WONG, M. D. F. Global routing with crosstalk constraints. In: IEEE/ACM DESIGN AUTOMATION CONFERENCE, 1998, San Francisco, CA, EUA. **Proceedings...** New York: ACM, 1998. p. 374–377.