

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

Elias de Almeida Ramos

Posicionamento de Células em FPGAs  
Chaotic Place

Dissertação apresentada como requisito parcial para a  
obtenção do grau de Mestre em Computação.  
Orientador: Prof. Dr. Ricardo Augusto da Luz Reis

Porto Alegre 2018

## CIP – CATALOGAÇÃO NA PUBLICAÇÃO

Ramos, Elias de Almeida  
Posicionamento de células em FPGAs Chaotic Place / Elias de Almeida Ramos. –  
Porto Alegre: PPGC da UFRGS, 2018.  
83 f.: il.  
Thesis (Master) – Universidade Federal do Rio Grande do Sul.  
Programa de Pós-Graduação em Ciência da Computação, Porto Alegre, BR-RS, 2017. Orientador: Ricardo Augusto da Luz Reis.  
1. Microeletrônica. 2. EDA. 3. FPGA. 4. Posicionamento. I. Reis, Ricardo Augusto da Luz. II. Título.

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL

Reitor: Prof. Rui Vicente Oppermann

Vice-Reitor: Profa. Jane Fraga Tutikian

Pró-Reitor de Pós-Graduação: Prof. Celso Giannetti Loureiro Chaves

Diretor do Instituto de Informática: Profa. Carla Maria Dal Sasso Freitas

Coordenador do PPGC: Prof. Joao Luiz Dihl Comba

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

## Agradecimentos

Agradeço aos meus pais e ao meu irmão: Edson, Elze e Everton, por estarem comigo e me apoiarem quando decidi deixar minha terra natal e me aventurar por terras distantes. Em especial ao meu pai, o Capitão Edson Ramos que está agora com as Estrelas ao lado do Poder Superior, e como ele dizia: “Pelos seiscentas mil P...”.

Ao meu Triângulo Equilátero: Minha esposa Fernanda e meus meninos Fernando e Mateus (Sem vocês nada disso seria possível!).

À minha família gaucha: minha sogra Marilaine, meus cunhados Rodrigo e Mauricio.

Aos meus amigos de infância e de Rock: Dilton, Fabio, Hermelino e Renato (“*East of the Sun, and West of the moon...*”).

Ao meu Padrinho Lucio, pois através dele encontrei o Poder Superior.

Aos meus colegas de docência Rubens e Antônio Reis, pois seus “incentivos” foram essenciais (“Sai dessa cadeira! Isso vai dar em nada! Vai dar aula!”). E ao colega Eduardo Medeiros pela ajuda com o Netbeans.

Aos meus professores de matemática Antenor, Danilo e Genaro (*in memorian*), por apresentarem para mim a realeza deste mundo abstrato, lindo e único.

Aos meus professores do IMPA Marcelo Viana, Jacob Palis, Elon Lima (*in memorian*), Carlos Gustavo Moreira, por me lançarem dentro do Caos.

Aos professores Dante Barone, Roberto Cabral, Daltro Nunes e Luis Lamb por me apresentarem ao PPGC da UFRGS, este centro de excelência impar.

Ao meu orientador Ricardo Reis, meu Mestre, meu Amigo. Palavras não descrevem a sua importância para minha formação. Muito Obrigado.

Ao Guilherme Bontorin e a Vitor Bandeira pela ajuda sem precedentes e por me apresentarem aos FPGAs e as demais maravilhas da Microeletrônica.

Aos meus colegas de laboratório e de EDA Jucemar, Mateus, Guilherme, Julia, André, Calebe e Luciana, pela ajuda e paciência com alguém de outra área (aprendi e aprendo a cada dia com vocês).

A Roger Waters, David Gilmour, Rick Wright, Nick Mason, Syd Barret (“*Shine on, you crazy diamond*”), John Lennon, Paul McCartney, George Harrison e Ringo Star (“*Strawberry fields forever...*”).

## **Lista de Acrônimos**

**ASIC** *Application Specific Integrated*

**BRAM** *Block RAM*

**CI** *Circuito Integrado*

**CMOS** *complementary metal-oxide-semiconductor*

**CLB** *Logic block*

**CP** *Chaotic Place*

**CPLD** *Complex Programmable Logic Device*

**DSP** *Digital Signal Processor*

**EDA** *Electronic Design Automat*

**FF** *Flip-flop*

**FPGA** *Field Programmable Gate Array*

**HCPLD** *High Complex Programmable Logic Devices*

**HDL** *Linguagem de Descrição de Hardware*

**HPWL** *Half-Perimeter-Wire-Length*

**LLE** *Largest Lyapunov Exponent*

**LSI** *Large Scale Integration*

**LUT** *Look Up Table*

**MLP** *Multiple layer perceptron*

**MSI** *Medium Scale Integration*

**PLD** *Programmable logic device*

**SPLD** *Small Programmable logic device*

**SSI** *Small Scale Integration*

**TSP** *Traveling Salesman Problem*

**TEG** *Teoria Espectral dos Grafos*

**ULSI** *Ultra Large Scale Integration*

**VLSI** *Very Large Scale Integration*

**VHDL/VHSIC** *Hardware Description Language*

**VPR** *Versatile Packing, Placement and Routing*

## Lista de Figuras

Figura 2.1 Estrutura geral de um FPGA .....	21
Figura 2.2 Estrutura de um CLB.....	22
Figura 2.3 Estrutura de um FF .....	22
Figura 2.4 Fluxo de projeto para FPGA .....	24
Figura 3.1 A importância do posicionamento para o roteamento .....	26
Figura 3.2 Representação das fases posicionamento global, legalização e posicionamento detalhado .....	27
Figura 3.3 Comparação entre a distância Quadrática e Manhattan .....	28
Figura 3.4 Representação do HPWL .....	30
Figura 3.5 Representação do grafo completo .....	30
Figura 3.6 Representação do método saída para entradas .....	31
Figura 3.7 Representação do método menor caminho .....	31
Figura 3.8 Modelo de <i>Spanning Tree</i> .....	32
Figura 3.9 Modelo de <i>Stainer Tree</i> .....	32
Figura 3.10 Modelo clique.....	33
Figura 3.11 Modelo estrela .....	34
Figura 3.12 Modelo <i>bound to bound</i> .....	34
Figura 4.1 Algoritmos de posicionamento e o estado da arte .....	37
Figura 4.2 Etapas do particionamento .....	40
Figura 5.1 Conjunto de Cantor .....	45
Figura 5.2 Diagrama de Bifurcação do Mapa Logístico .....	46
Figura 5.3 Atrator de Lorenz .....	47
Figura 5.4 Reconstrução do atrator de Lorenz .....	49
Figura 6.1 O neurônio humano .....	53
Figura 6.2 O Perceptron.....	54
Figura 6.3 Conjuntos linearmente independentes .....	55
Figura 6.4 Rede de múltiplas camadas .....	56
Figura 6.5 A função sigmóide.....	56
Figura 6.6 Rede de Hopfield.....	57
Figura 6.7 Atrator Asteriscus.....	58
Figura 7.1 Fluxograma do Posicionamento .....	60
Figura 7.2 Descrição do Método do Gradiente Descendente .....	63

Figura 7.3 Atrator resultante (FPGA apex2) .....	65
Figura 7.4 Representação de um <i>bounding box</i> .....	66
Figura 7.5 Representação de uma célula fora das dimensões do FPGA sendo inserida	67
Figura 7.6 Representação do posicionamento dos IO/s .....	67
Figura 7.7 Cálculo do tamanho de cada <i>net</i> .....	67
Figura 7.8 Representação da diminuição do tamanho de cada <i>net</i> .....	68
Figura 7.9 Tamanhos médios das <i>nets</i> em cada FPGA .....	69
Figura 7.10 Posicionamento final (FPGA “seq”) .....	70
Figura 7.11 Chaotic Place vs Diffusion .....	71
Figura 7.12 Chaotic Place vs VPR .....	72
Figura 7.13 Chaotic Place vs Star+ .....	72
Figura 7.14 Comparação entre o Chaotic Place e os demais .....	73
Figura 7.15 Representação do Chaotic Place dentro da faixa de tolerância em relação aos outros posicionadores .....	74

## Lista de Tabelas

Tabela 1.1 Representação dos circuitos integrados pela quantidade de Portas Lógicas .....	16
Tabela 7.1 Número de épocas em cada FPGA .....	64
Tabela 7.2 Diferenças entre o posicionamento inicial e final .....	70
Tabela 7.3 Comparação com o estado da arte .....	71
Tabela 7.4 Desvio padrão de cada FPGA dentre os posicionadores .....	73

*Non, rien de rien  
Non, je ne regrette rien  
Ni le bien qu'on m'a fait, ni le mal  
Tout ça m'est bien égal  
Non, rien de rien  
Non, je ne regrette rien  
C'est payé, balayé, oublié  
Je me fous du passé  
Avec mes souvenirs  
J'ai allumé le feu  
Mes chagrins, mes plaisirs  
Je n'ai plus besoin d'eux  
Balayés mes amours  
Avec leurs trémolos  
Balayés pour toujours  
Je repars à zéro*

*Non, rien de rien  
Non, je ne regrette rien  
Ni le bien qu'on m'a fait, ni le mal  
Tout ça m'est bien égal  
Non, rien de rien  
Non, je ne regrette rien  
Car ma vie  
Car mes joies  
Aujourd'hui  
Ça commence avec toi...*

Michel Vaucaire

Dédié à ma femme Fernanda.

# Resumo

O posicionamento de componentes em um circuito integrado é de vital importância para um projeto físico de qualidade. O posicionamento deste trabalho parte de um posicionador global de células para *Field Programmable Gate array* (FPGAs) baseado em algoritmos provenientes da Teoria dos Sistemas Dinâmicos não Lineares, também conhecidos como Sistemas Caóticos. Nossa metodologia consiste em utilizar uma versão simplificada do Posicionamento Analítico não linear, que utiliza a função LOG-SUM-EXP como função custo (para minimizar o comprimento dos fios entre as células). Enquanto o posicionamento usual trabalha em duas dimensões, nossa metodologia realiza a otimização das distâncias entre as células em apenas uma dimensão, para depois obter o levantamento dos pontos em duas dimensões gerando o posicionamento global. Para isso utilizamos uma Rede Neural de Hopfield para realizar a otimização das distâncias entre células. A estrutura bidimensional final do posicionamento será obtida através do uso do Teorema de Takens. Os experimentos mostraram resultados satisfatórios em relação ao estado da arte, onde foi diminuído o *Half-perimeter-wire-length* (HPWL) além de ser obtida uma redução da área utilizada em um FPGA. Os resultados mostram entre 5% a 6% de diminuição do HPWL em comparação com o estado da arte para FPGAs homogêneos.

**Palavras-Chave**—Microeletronica, EDA, Posicionamento, FPGA, Atratores, Sistemas Dinâmicos

# Placement Cells in FPGA

## The Chaotic Place

### Abstract

The placement of components in an integrated circuit is of vital importance for a quality physical design. The placement in this work starts by a global cell placement for Field Programmable Gate Array (FPGAs) based on algorithms from the Dynamic Nonlinear Systems Theory, also known as Chaotic Systems. Our methodology is to use a simplified version of Non-linear Analytical Placement, which uses the LOG-SUM-EXP function as a cost function (to minimize the length of the wires between cells). While the usual placement works in two dimensions, our methodology accomplishes the optimization of the distances between the cells in only one dimension, to then obtain the survey of the points in two dimensions generating the global placement. For this we use a Hopfield Neural Network to perform the optimization of distances between cells. The final two-dimensional structure of the placement is obtained through the use of Takens's Theorem. The experiments showed satisfactory results in relation to the state of the art, where the Half-perimeter-wire-length HPWL was reduced in addition to obtaining a reduction of the area used in an FPGA. The results show between 5% to 6% decrease of HPWL compared to the state of the art for homogeneous FPGAs

**Keywords**—Microelectronics, EDA, Placement, FPGA, Attractors, Dynamic Systems

# Sumário

<b>I Introdução .....</b>	<b>16</b>
1.1 Circuitos Integrados .....	16
1.2 Projetos de Circuitos Integrados .....	17
1.3 Posicionamento de Componentes em Circuitos Integrados .....	17
1.4 Motivação .....	18
1.5 Contribuições deste Trabalho .....	19
1.6 Organização do Trabalho .....	19
<b>II FPGAs .....</b>	<b>21</b>
2.1 Estrutura de um FPGA 2d-array .....	21
2.1.1 Blocos Lógicos .....	22
2.1.2 IOBs .....	23
2.1.3 Switch Matrix .....	23
2.2 Estrutura de um FPGA 3d-array .....	23
2.3 Programação .....	23
2.4 Fluxo de um projeto em FPGA .....	24
2.5 Resumo do capítulo .....	25
<b>III Posicionamento de Células em Circuitos Integrados .....</b>	<b>26</b>
3.1 Posicionamento .....	26
3.1.1 Posicionamento Global .....	26
3.1.2 Legalização .....	27
3.1.3 Posicionamento Detalhado .....	27
3.2 Métricas para o Comprimento dos fios .....	28
3.2.1 <i>Half-Perimeter-Wire-Length</i> .....	29
3.2.2 Grafo Completo .....	30
3.2.3 Saída para Entradas ou Origem para Destino .....	30
3.2.4 Menor Caminho .....	31
3.2.5 Menor “ <i>Spanning Tree</i> ” .....	31
3.2.6 Menor “ <i>Stainer Tree</i> ” .....	32
3.3 Modelos das nets .....	33
3.3.1 Modelo Clique .....	33
3.3.2 Modelo Estrela .....	33
3.3.3 Modelo <i>Bound to Bound</i> .....	34
3.4 Caminhos Críticos .....	35

3.5 Congestionamentos .....	35
3.6 Atrasos .....	35
3.7 Consumo de Energia.....	35
3.8 Distribuição do Calor .....	35
3.9 Resumo do Capítulo.....	36
<b>IV O Estado da Arte .....</b>	<b>37</b>
4.1 Estocástico .....	37
4.1.1 Simulated Annealing.....	38
4.1.2 Algoritmos Genéticos .....	38
4.2 Particionamento .....	39
4.3 Analítico.....	40
4.3.1 Quadráticos .....	40
4.3.2 Autovalores .....	40
4.3.3 Particionamento .....	41
4.3.4 Forças.....	41
4.5.5 Não Quadráticos .....	41
4.4 Novas Técnicas de Posicionamento .....	42
4.5 Resumo do Capítulo.....	42
<b>V Sistemas Dinâmicos.....</b>	<b>43</b>
5.1 Efeito Borboleta .....	44
5.2 Fractais .....	44
5.3 Estabilidade de Lyapunov.....	46
5.4 Pontos de Equilíbrio .....	46
5.5 Atratores .....	46
5.6 Atratores Estranhos.....	47
5.7 Teorema de Takens.....	48
5.7.1 Dimensão de Correlação .....	48
5.8 Posicionamento Analítico como um Sistema Caótico.....	49
5.9 Teoria Espectral dos Grafos .....	50
5.10 Sistemas Dinâmicos na Computação.....	51
5.11 Resumo do Capítulo.....	51
<b>VI Redes Neurais .....</b>	<b>53</b>
6.1 Algumas Redes Neurais.....	54
6.1.1 O Perceptron .....	54

6.1.2 Redes de Múltiplas Camadas .....	55
6.1.3 Redes de Hopfield.....	57
<b>6.2 Resumo do Capítulo.....</b>	<b>59</b>
<b>VII Chaotic Place.....</b>	<b>60</b>
<b>7.1 Mapeamento das <i>nets</i>.....</b>	<b>61</b>
7.1.1. Estrutura das <i>nets</i> .....	61
<b>7.2 Formulação Matemática .....</b>	<b>62</b>
<b>7.3 Aplicação do Teorema de Takens.....</b>	<b>65</b>
<b>7.4 Espalhamento .....</b>	<b>66</b>
<b>7.5 Legalização .....</b>	<b>66</b>
<b>7.6 Posicionamento Detalhado .....</b>	<b>67</b>
<b>7.7 Resultados.....</b>	<b>70</b>
<b>VIII Conclusões e Trabalhos futuros.....</b>	<b>75</b>
8.1 Conclusões .....	75
8.2 Trabalhos Futuros .....	75
<b>Referências.....</b>	<b>78</b>

## I. INTRODUÇÃO

Circuitos Integrados (CIs) compõem a base dos dispositivos eletrônicos onipresentes em nosso cotidiano, como computadores, leitores de MP3 e MP4, videogames, *smartphones*, automóveis, equipamentos médicos e eletrodomésticos em geral. O investimento no avanço tecnológico é crescente e contínuo demandando a pesquisa em novas estratégias para a solução dos novos problemas que se apresentam. Dentre os circuitos integrados, uma classe que teve sua origem em meados dos anos 80 merece destaque em nosso trabalho. Os *Fields Programmable Gate Array* (FPGAs) são circuitos reconfiguráveis pelo usuário. Uma propriedade importante desses circuitos é a possibilidade de existirem vários blocos de hardware operando em paralelo aumentando substancialmente a capacidade computacional (KUON, IAN; ROSE, JONATHAN, 2006). Circuitos reconfiguráveis deram um grande poder de aceleração de um projeto além da redução do custo, quando o volume de exemplares inviabiliza o projeto de um ASIC.

O tempo entre a idealização de um circuito e sua implementação é definido na maioria das vezes em função da necessidade da entrega ao mercado (*time-to-market*), pois há muito investimento envolvido e resultados precisam ser cada vez mais rápidos e com menor custo financeiro (MANSMANN, J. et al, 1992).

A automação do projeto de circuitos integrados demanda o uso de ferramentas de *Electronic Design Automation* (EDA). Tendo em vista que hoje um simples circuito pode conter bilhões de componentes tais ferramentas são mais que necessárias e fundamentais.

### 1.1 Circuitos Integrados

São pequenas pastilhas (chips) contendo em seu interior componentes eletrônicos, especialmente os transistores. Os transistores compõem portas lógicas gerando circuitos combinatórios e sequenciais (HOROWITZ, PAUL; HILL, WINFIELD, 1989).

Em relação ao número de portas lógicas os circuitos são classificados seguindo a Tabela 1.1 (TANENBAUM, A. S, 1987).

Tabela 1.1 Representação dos circuitos integrados pela quantidade de portas lógicas

<b>Tipo</b>	<b>Número de Portas Lógicas</b>
<b>SSI (<i>Small Scale Integration</i>)</b>	1 a 10
<b>MSI (<i>Medium Scale Integration</i>)</b>	11 a 100
<b>LSI (<i>Large Scale Integration</i>)</b>	101 a 100000
<b>VLSI (<i>Very Large Scale Integration</i>)</b>	100000 a 100000000
<b>ULSI (<i>Ultra Large Scale Integration</i>)</b>	Acima de 100000000

Outra classe de circuitos são os *Programmable Logic Device* (PLDs) onde são configurados pelo próprio usuário. Assim é eliminado o processo de fabricação do circuito, pois o projeto pode ser modificado. Comparados aos anteriores possuem custo reduzido e menor tempo de projeto. Circuitos reconfiguráveis deram um grande poder de aceleração de um projeto além da redução do custo, quando o volume de exemplares inviabiliza o projeto de um ASIC. (PELLERIN, DAVID; HOLLEY M, 1991).

Dentre os PLDs serão destacados os FPGAs que serão o foco neste trabalho e serão definidos melhor no próximo capítulo (FREEMAN, ROSS H, 1989).

No projeto de circuitos integrados se deve atentar para as tendências:

- Aumento da complexidade.
- Diminuição do tamanho dos componentes.
- Diminuição do consumo de energia.
- Ferramentas de EDA mais sofisticadas.
- Aumento da velocidade.

As observações acima justificam o estudo mais aprofundado no projeto de um circuito.

## 1.2 Projeto de Circuitos Integrados

Um conjunto de ferramentas de EDA é necessário para implementar um fluxo do projeto. O desenvolvimento de uma ferramenta em EDA não é uma tarefa trivial. A maioria dos problemas envolve algum tipo de otimização, onde muitos são de natureza NP - completo. Para isso é necessário além de conhecer os processos tradicionais de a computação investigar novas técnicas para a resolução dos algoritmos que modelam os problemas em EDA. O progresso em computação teórica implica profundamente nos avanços em EDA (KAHNG, A. B. et al, 2011).

## 1.3 Posicionamento de Componentes em Circuitos Integrados

Nas últimas décadas a complexidade lógica de um circuito aumenta seguindo a Lei de Moore (MOORE, 1965) tendo como consequência o aumento do número de seus componentes. Inevitavelmente é necessário o desenvolvimento de ferramentas de EDA mais eficientes para suprir as necessidades do projetista. A etapa do posicionamento é de fundamental importância na síntese física de um circuito ASIC ou FPGA (MONTEIRO, 2014). O posicionamento é definido como o processo de alocar as células lógicas de um circuito em uma região da planta baixa do mesmo, sem sobreposições de células, nem células alocadas em espaços indevidos. O posicionamento influencia consideravelmente o desempenho de um circuito integrado. O posicionamento deve garantir com que o circuito seja roteável, em outras palavras: deve permitir que sejam realizadas todas as interconexões entre as células, caso contrário o circuito não será funcional.

Devido à complexidade do problema, diversos fatores precisam ser considerados, o posicionamento é direcionado a algum fator o que o torna um problema de otimização restrita. Dentre os fatores destacamos os mais importantes (WANG, L.-T.; CHANG, Y.-W.; CHENG, K.-T. T, 2009), (KAHNG, A. B. et al, 2011):

- Comprimento total dos fios  
A minimização do comprimento total dos fios minimiza o tamanho do circuito gerando um menor custo. Como também minimiza problemas de energia e atraso. É um dos principais parâmetros a ser considerado pelos posicionadores.
- *Timing* (Atraso)  
O ciclo de relógio de um circuito é definido como o atraso do maior caminho, este também conhecido como caminho crítico. A ideia é garantir que o atraso do caminho crítico não seja maior que o atraso pré-definido na especificação.
- Congestionamentos  
Além da diminuição do comprimento dos fios é necessário o cuidado com a alta densidade de componentes em regiões do circuito. Uma alta densidade pode gerar congestionamento das rotas fazendo com conexões tenham de usar caminhos longos, e podendo aumentar os caminhos críticos e talvez impossibilitando o roteamento.
- Potência  
O objetivo é diminuir o consumo de energia, como também eliminar regiões que possam ter regiões de alta temperatura (*hot spots*).

## 1.4 Motivação

Sistemas Dinâmicos são deterministas, apresentando um fenômeno chamado usualmente de “Sensibilidade às Condições Iniciais”. Isto significa que após um tempo razoavelmente grande o sistema terá um comportamento imprevisível. O estudo teve início nos trabalhos de Henry Poincaré no final do século XIX. Em seus trabalhos sobre Mecânica Celeste, na tentativa de resolver algumas classes de Equações Diferenciais não Lineares, percebeu a impossibilidade de resolvê-las quantitativamente. Poincaré introduziu novos conceitos e trouxe para a análise do problema técnicas provenientes de outras áreas da Matemática como Álgebra, Topologia, etc. Estava então criada a Teoria dos Sistemas Dinâmicos, também conhecida usualmente por Teoria do Caos. A ideia inicial era não apenas resolver a equação, mas estudar o comportamento topológico das soluções (BONATTI, DIAZ, VIANA, 2005).

Nos últimos cinquenta anos a Teoria dos Sistemas Dinâmicos passou de coadjuvante a uma das principais linhas de pesquisa em Matemática Pura e Aplicada sendo utilizada para diversos fins e diversas áreas do conhecimento, como Física, Química, Economia, Computação, etc. Em meados dos anos 80, Jacob Palis (BONATTI, DIAZ, VIANA, 2005) conjecturou que a “maioria” dos Sistemas Dinâmicos converge para uma quantidade finita de estruturas de comportamentos bem peculiares e denominadas de atratores onde a bacia de atração (os pontos que se aproximam do atrator com o passar do tempo) tem medida total (significa ter a maioria dos pontos no sentido estatístico), se o sistema está definido em uma superfície compacta. A conjectura está longe de ser demonstrada, por outro lado tem-se resultados em casos particulares, suficientes para justificar aplicações, como no estudo comportamental das Redes Neurais (BONATTI, DIAZ, VIANA, 2005).

Uma Rede Neural de Hopfield é um Sistema Dinâmico que converge a um estado estacionário (um atrator), através de um número suficiente de interações e através de uma função denominada Energia atingindo seu mínimo caracterizando o estado final da Rede (HAYKIN, 2001). Pelo Teorema de Takens (TAKENS, 1981), (MAÑE, 1981) é possível tratar problemas modelados por funções do tipo  $f: R^n \rightarrow R^n$  utilizando um único eixo e posteriormente reconstruindo o espaço de fase. Logo a Rede Neural poderá trabalhar obtendo o estado estacionário através da minimização da energia utilizando apenas uma coordenada e

a seguir as demais coordenadas serão geradas pelo teorema, formando uma solução  $n$ -dimensional topologicamente similar à original.

O posicionamento de células pode ser modelado como um problema de otimização convexa além de que, como será provado no decorrer do texto, um posicionamento analítico será identificado como um Sistema Dinâmico Caótico. Para encontrar a solução do sistema gerado pela diferenciação da função custo (neste caso para diminuir o comprimento dos fios) será utilizada uma Rede Neural de Hopfield.

A escolha do modelo de rede se dá porque é uma excelente ferramenta para resolução de problemas envolvendo otimização de um modo geral (HOPFIELD, J. J., 1982). Logo, se analisarmos a solução do posicionamento obtida pela Rede de Hopfield gerada em seu estado estacionário como um atrator, o Teorema de Takens pode ser aplicado de imediato.

## 1.5 Contribuições deste Trabalho

Neste trabalho foi desenvolvido um posicionamento analítico para FPGAs onde foi simplificado o posicionamento não linear utilizando a LOG-SUM-EXP como função custo na etapa do posicionamento global utilizando como ferramenta uma Rede Neural de Hopfield. É introduzida uma versão unidimensional do posicionamento com o intuito de se aplicar o Teorema de Takens que consiste na reconstrução de atratores estranhos proveniente da Teoria dos Sistemas Dinâmicos para a obtenção da versão bidimensional definitiva.

São apresentadas a seguir as contribuições deste trabalho:

- Caracterização de um Posicionamento Analítico como um Sistema Dinâmico Caótico.
- Revisão da Teoria dos Sistemas Dinâmicos, Atratores, o Teorema de Takens, Teoria Espectral dos Grafos e Aplicações.
- Revisão da Teoria das Redes Neurais com Ênfase em Redes recorrentes destacando as Redes de Hopfield e Aplicações.
- Implementação de um Posicionamento Global baseado em algoritmos inspirados na Teoria dos Sistemas Dinâmicos utilizando como ferramenta uma Rede Neural de Hopfield.

## 1.6 Organização do Trabalho

Este trabalho está organizado da seguinte maneira. No capítulo II introduzimos os FPGAs explicando suas finalidades e descrevendo seus componentes. Para o leitor familiarizado com os conceitos é possível iniciar a leitura a partir do capítulo III onde é apresentado o problema do posicionamento de componentes em um circuito descrevendo as métricas utilizadas e as consequências do posicionamento nas outras fases que constituem a fabricação de um circuito. No capítulo IV são descritos os mais importantes posicionadores realizando uma revisão bibliográfica apresentando o estado da arte. No capítulo V são apresentados os Sistemas Dinâmicos, exemplos e definições adjuntas com os atratores estranhos, Expoentes de Lyapunov, Teoria Espectral dos Grafos e o Teorema de Takens. No capítulo VI são apresentadas as Redes Neurais dando ênfase às redes recorrentes, em particular às Redes Neurais de Hopfield e suas aplicações. No capítulo VII é apresentado o Chaotic Place, o tema principal deste trabalho definindo-o formalmente. São mostrados os

resultados e a comparação com o estado da arte. No capítulo final são apresentadas as conclusões além de possíveis trabalhos futuros.

## II. FPGAs

Os circuitos *Programmable Logic Devices* (PLDs) são dispositivos lógicos reconfiguráveis, cuja programação é feita pelo próprio usuário (PELLERIN, DAVID; HOLLEY M., 1991). Dentre os PLDs destacamos aqueles que possuem maior número de aplicações na indústria e na academia, os FPGAs.

FPGAs são circuitos integrados reconfiguráveis, ou seja, podem ser personalizados após a fabricação, diferentes dos ASICs onde todas as funcionalidades são definidas no ato de fabricação. Foram criados pela Xilinx Inc tendo como cofundadores Ross Feeman e Bernard Vonderschmitt, e a primeira fabricação aconteceu no ano de 1983, tendo seu lançamento oficial no ano de 1985. Hoje as duas maiores empresas fabricantes de FPGAs são a própria Xilinx e a Altera (Intel). Não possuem planos “OR” e “AND” sendo um conjunto de células configuráveis para programar funções lógicas (SIDIROPOULOS, H. et al, 2012).

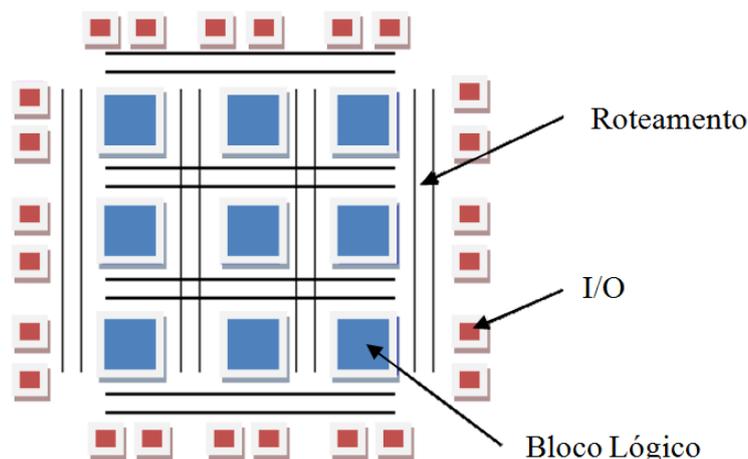
FPGAs podem resolver a maioria dos problemas de natureza computacional, desde simples portas lógicas como funções complexas, pois qualquer equação booleana pode ser neles implementada. Além disso, a estrutura paralela e as otimizações em termos de portas lógicas aceleram consideravelmente qualquer processo. Também são aplicados como aceleradores de hardware. Substituem partes de um algoritmo e dessa forma o processo se torna híbrido sendo composto pelo FPGA e pelo processador usual (SIDIROPOULOS, H. et al, 2012).

Possuem aplicações em diversas áreas da indústria: processamento digital de sinal em tempo real, *switches* e roteadores, processamento de imagens em tempo real, criptografia, setor automotivo, aplicações espaciais como foguetes, satélites e sondas, etc. (SIDIROPOULOS, H. et al, 2012). Os FPGAs podem apresentar dois tipos de Geometria: FPGA 2d-array e FPGA 3d-array.

### 2.1 Estrutura de um FPGA 2d-array

É composto geralmente por três componentes: Blocos Lógicos (CLBs), Entradas e Saídas (I/O) e a Matriz de Interconexões (*Switch Matrix*) descritas na figura 2.1 (PELLERIN, DAVID; HOLLEY M., 1991).

Figura 2.1- Estrutura geral de um FPGA



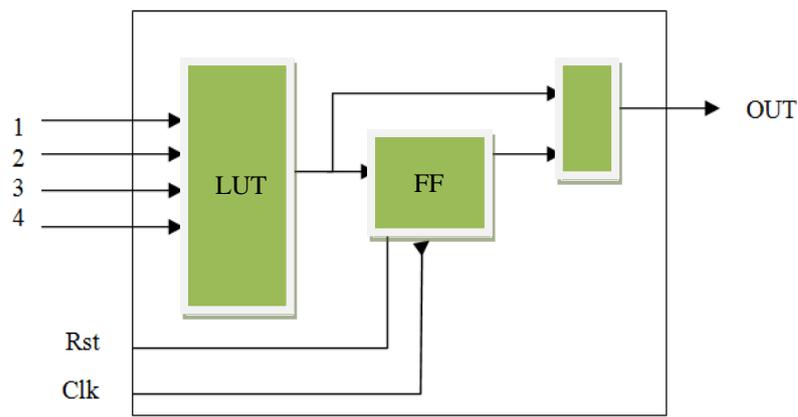
Fonte: (BENCHMARKS, 2017)

### 2.1.1 Blocos lógicos

Os Blocos lógicos (CLB) podem ser constituídos da reunião de flip-flops (FF) e uma lógica combinacional denominada *Look Up Table* (LUT), descrito na figura 2.2.

- LUTs: São os componentes responsáveis pelo processamento da lógica. Cada LUT pode ter de duas a quatro entradas onde a saída é programada em relação a todas as possíveis entradas. Cada saída de uma LUT pode ser conectada a um FF e cada grupo de LUTs e FFs é chamado de Fatia (*Slice*) (SIDIROPOULOS, H. et al, 2012).

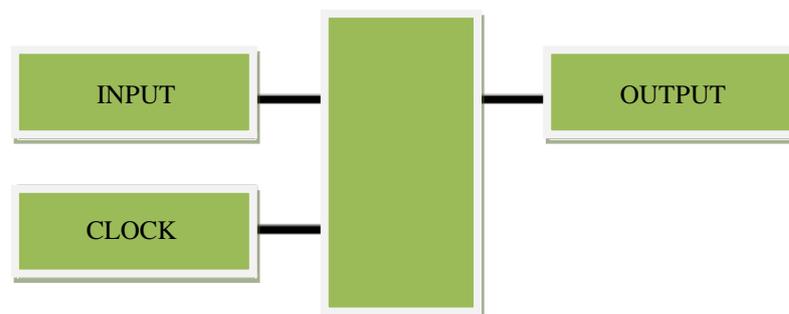
Figura 2.2- Estrutura de um CLB



Fonte: Figura adaptada de (BENCHMARKS, 2017)

- FF: São elementos de memorização que sincronizam as portas lógicas e registram estados lógicos entre ciclos de relógio (*clock*) no interior de um FPGA (SIDIROPOULOS, H. et al, 2012), como é observado na figura 2.3.

Figura 2.3- Estrutura de um FF



Fonte: Figura adaptada de (BENCHMARKS, 2017)

### 2.1.2 I/O

São os componentes responsáveis pela comunicação com o meio externo, em outras palavras representam as saídas provenientes das combinações entre os CLBs.

### 2.1.3 Switch Matrix

Já a *Switch Matrix* corresponde às trilhas de conexões entre os CLBs e os I/Os. O processo de escolha das interconexões é denominado roteamento.

No caso bidimensional existem dois tipos de FPGAs: Homogêneos e Heterogêneos (SIDIROPOULOS, H. et al, 2012).

- Homogêneos: Compostos apenas por CLBs, I/O, e FF.
- Heterogêneos: Compostos pelos elementos descritos acima adicionados a Memória (BRAMs), Processadores de Sinais (DSPs), e outros módulos Circuitos reconfiguráveis deram um grande poder de aceleração de um projeto além da redução do custo, quando o volume de exemplares inviabiliza o projeto de um ASIC. onde:
  - DSPs: Utilizados para poupar o trabalho das LUTs e FFs em aplicações matemáticas, e processamento de sinal.
  - BRAMs: Utilizadas para armazenamento de dados ou na passagem de dados entre tarefas paralelas.

## 2.2 Estrutura de um FPGA 3d-array

Esta arquitetura foi introduzida com o intuito de diminuir o consumo de energia e o tamanho dos circuitos em geral. Os componentes básicos de um FPGA 3d-array são análogos aos de um 2d-array, porem existem camadas empilhadas e ligadas pelas Vias 3D. Devido ao empilhamento, o comprimento médio dos fios diminui. Por outro lado, surgem mais possibilidades de posicionamento e roteamento dificultando a criação de uma ferramenta EDA para desenvolvê-los (ABABEI, C et al. 2016).

Outro problema a ser trabalhado em um FPGA 3d-array é o fato de que um caminho crítico, se possível, não deve atravessar duas camadas, ou seja, para o bom funcionamento do circuito o maior número de caminhos críticos deve corresponder a caminhos totalmente imerso em uma única camada (HENTSCHKE R., JOHANN M., REIS R., 2012). Com isso os algoritmos de posicionamento devem tratar cuidadosamente deste detalhe importante.

Outro detalhe importante é a questão dos algoritmos de posicionamento e roteamento 3d. Até o momento o estado da arte deles consiste em algoritmos de posicionamento de circuitos 2d que foram adaptados para os circuitos 3d. Em resumo, não há algoritmos que ataquem o problema diretamente (ABABEI, C et al. 2016). Os FPGAs 3d-array não serão explorados neste trabalho.

## 2.3 Programação

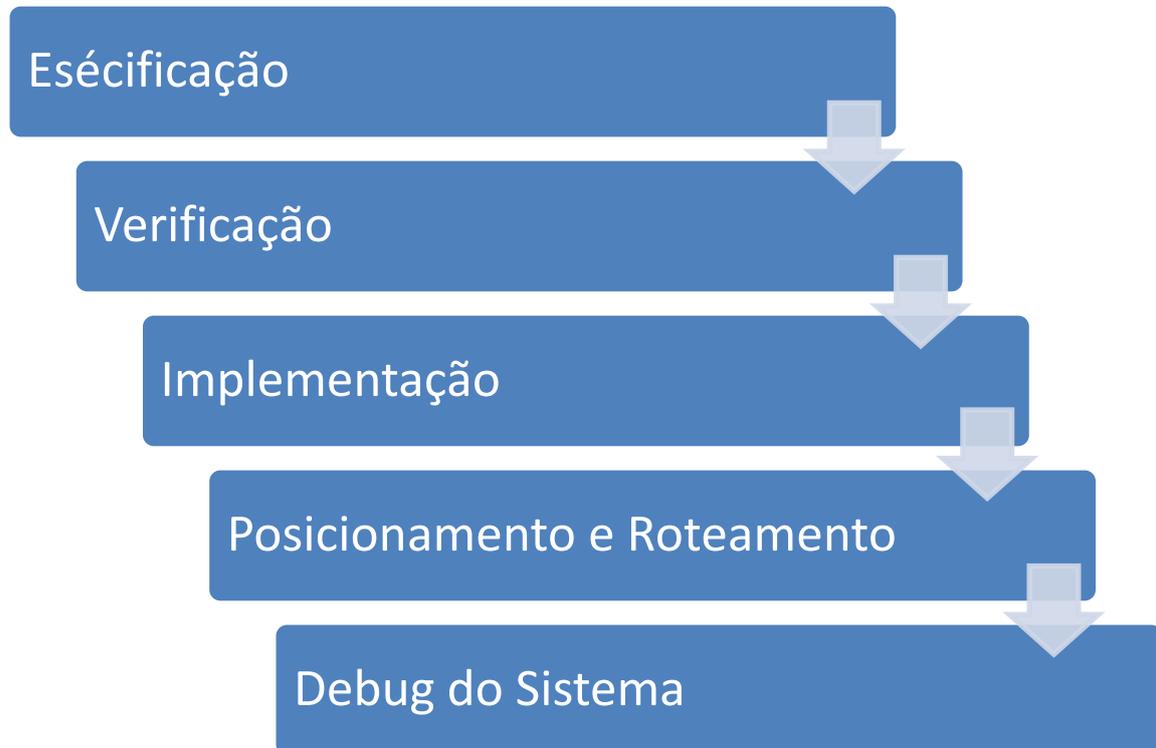
O comportamento do FPGA é definido pelo usuário onde este define o sistema através de uma Linguagem de Descrição de Hardware (HDL). Daí se faz necessário uma ferramenta

de EDA onde serão geradas as *netlists* adaptadas para a estrutura do FPGA (BHASKER J. 1995).

## 2.4 Fluxo de um Projeto em FPGA

O fluxo de um projeto é uma sequência de passos ordenados até a implementação final (BHASKER J. 1995). Suas fases principais estão descritas na figura abaixo:

Figura 2.4: Fluxo de um projeto para FPGA



Fonte: Figura elaborada pelo autor

Especificação é a etapa onde a *netlist* é gerada. Nesta etapa são definidos os componentes do circuito. Seus CLBs, suas interconexões como os I/O para as ferramentas de verificação e implementação do circuito. A geração da *netlist* pode ser feita de duas maneiras. Pelo esquemático ou pelo HDL (os mais utilizados são o VHDL e o Verilog). Um esquemático é definido como a representação gráfica do *netlist*. Há uma desvantagem, pois haverá uma dependência do fabricante. Se criarmos um esquemático visando um fabricante e depois tentarmos fazê-lo para outro é possível que tenhamos que refazer todo o esquemático novamente. Pelo HDL a mudança de fabricante é irrelevante, pois basta uma ferramenta de síntese para que o HDL seja interpretado e seja gerada uma *netlist* otimizada.

Na fase de verificação são feitas simulações a procura de algum possível erro na especificação.

Na fase de implementação temos o mapeamento da *netlist* gerada. São mapeados os Blocos Lógicos (CLBs), Memórias (RAMs), Processadores de Sinal (DSPs), etc.

No posicionamento as locações finais das células em um circuito são definidas atribuindo os componentes físicos aos componentes lógicos gerados na fase anterior.

Após o posicionamento temos o roteamento, onde são definidas as trilhas de conexões entre os componentes onde os recursos disponíveis de interconexão são utilizados realizando a comunicação entre os componentes. A etapa final consiste em simulações e *debug* do dispositivo.

## **2.5 Resumo do Capítulo**

Neste capítulo foi efetuada uma breve descrição de FPGAs, suas estruturas, aplicações e propriedades. Para ser realizado o posicionamento das células é necessário interpretar o circuito como um grafo. Cada célula é representada por um nó e cada ligação por uma aresta. No capítulo seguinte será apresentado o formalismo de um circuito integrado como um grafo, assim como as técnicas utilizadas para o posicionamento como suas fases, distâncias utilizadas e métricas para o comprimento dos fios.

## III Posicionamento de Células em Circuitos Integrados

### 3.1 Posicionamento

O posicionamento é a etapa da síntese cuja meta é encontrar a melhor posição de cada componente em um circuito integrado. Uma ferramenta de posicionamento recebe como entrada a *netlist* do circuito. Cada *netlist* é um grafo onde cada nodo é um componente e as arestas são as conexões entre eles (KAHNG et al., 2011). Diferente dos ASICs os FPGAs já possuem suas trilhas pré-definidas como também todas as locações.

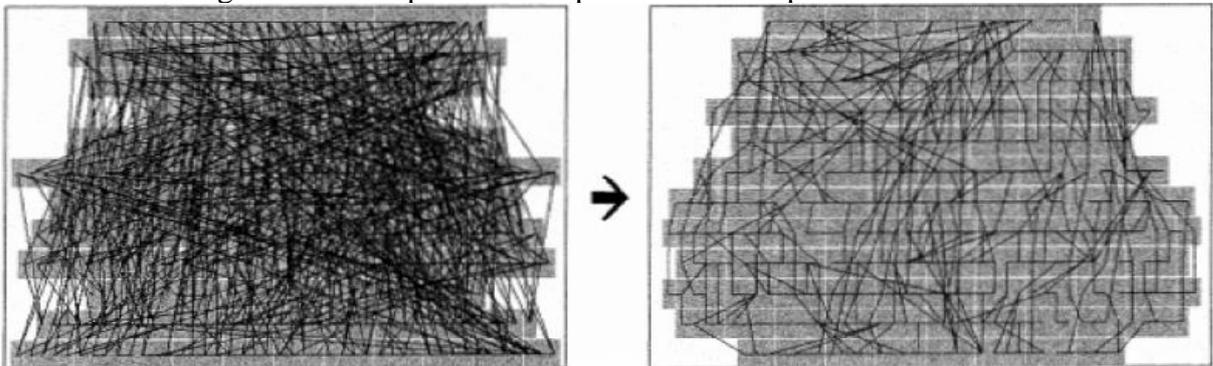
O posicionamento pode ser definido formalmente da seguinte maneira: Consideremos um grafo  $H = (V, E)$  onde  $V = \{v_1, v_2, v_3, v_4, \dots, v_n\}$  é o conjunto de vértices e  $E = \{e_1, e_2, e_3, e_4, \dots, e_k\}$  o conjunto de *nets*. O problema consiste em determinar os pontos  $(x_i, y_i)$  de cada bloco lógico  $v_i$  onde a posição seja válida e que minimize o comprimento total das arestas e cumpra os requisitos solicitados dependendo da topologia de cada circuito (CHAN, T. et al 2007).

Posicionamento é um problema de otimização combinatória, de natureza NP - completa (SHERWANI, N, 1999). Seu objetivo é alocar células em um circuito integrado proporcionando um bom funcionamento do mesmo (SPINDLER, P.; SCHLICHTMANN, U; JOHANNES, F. M. 2008). Um bom funcionamento pode ser entendido se todas as condições abaixo são rigorosamente satisfeitas:

- Menor comprimento de fios ou menor área do circuito.
- Evitar congestionamentos.
- Menor comprimento dos caminhos críticos
- O circuito deve ser roteável, ou seja, quando todas as rotas são traçadas conectando todas as células onde as conexões definidas no *netlist*.

Dos parâmetros citados o primeiro a ser trabalhado é o comprimento dos fios, pois esta análise indiretamente influencia as demais observações, como está exemplificado na Figura 3.1 onde a primeira imagem representa um posicionamento aleatório e a segunda um posicionamento baseado em *Simulated Annealing* (HENTSCHKE, R.; JOHANN, M.; REIS, R. 2005).

Figura 3.1- A importância do posicionamento para o roteamento



Fonte: (HENTSCHKE, R.; JOHANN, M.; REIS, R. 2005)

Devido à complexidade do problema, o posicionamento de células é dividido em três etapas:

#### 3.1.1 Posicionamento Global

O objetivo principal do posicionamento global não é a posição definitiva das células no circuito integrado, mas a distribuição das mesmas tendo como objetivo principal deixar as células conectadas em posições onde estas estarão mais próximas possíveis. E o conjunto de células é distribuído ao longo do espaço disponível para que os locais definitivos sejam encontrados nas fases seguintes. Esta fase é a mais importante de todas, pois influencia diretamente a qualidade da solução final. Deve-se ter o cuidado de nas fases posteriores a estrutura topológica das células não seja modificada radicalmente, invalidando a etapa (WANG; CHANG; CHENG, 2009).

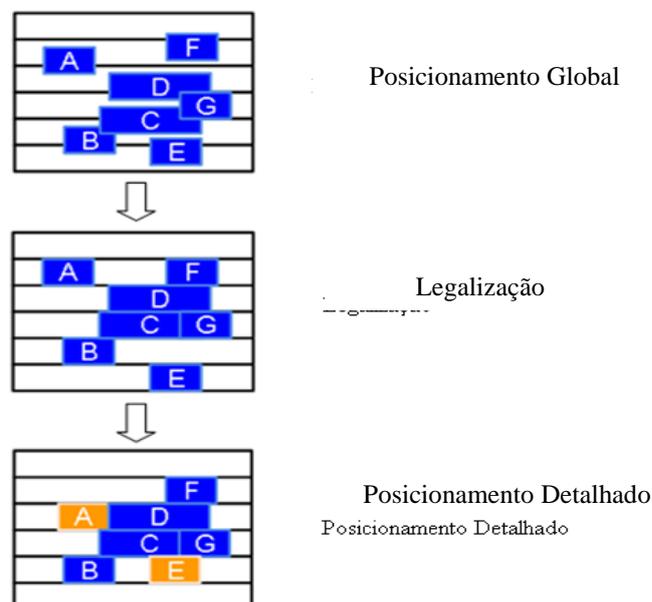
### 3.1.2 Legalização

Se considerarmos somente o posicionamento global a probabilidade de encontrarmos células em locais indevidos é grande. Daí, na fase de legalização as células que não estiverem em seus devidos locais serão realocadas. Um cuidado fundamental é que esta mudança nas células não afete consideravelmente o posicionamento global obtido na fase anterior, ou seja, devem ser deslocadas o mínimo possível (PUGET, J. C. et al, 2015). De modo geral dizemos que um posicionamento está bem definido se não há células sobrepostas e se todas as células estão em locais que não violem os requisitos do circuito (WANG; CHANG; CHENG, 2009).

### 3.1.3 Posicionamento Detalhado

Nesta fase é realizada uma análise local. Pode-se nesta fase melhorar algumas variáveis que possam melhorar o funcionamento do circuito, como diminuir o tamanho dos caminhos críticos, reduzir congestionamentos, análise dos *hotspots*, etc. (WANG; CHANG; CHENG, 2009). As etapas de um posicionamento estão ilustradas na Figura 3.2.

Figura 3.2- Representação das fases posicionamento global, legalização e posicionamento detalhado



Fonte: (FOGAÇA, MATEUS PAIVA, 2016)

### 3.2 Métricas para o Comprimento dos Fios.

Considere  $\mathcal{C} = \{c_1, c_2, c_3, \dots, c_n\}$  um conjunto de células. Para cada célula em  $\mathcal{C}$  serão utilizadas suas coordenadas retangulares:  $(x, y)$  e para  $i, j = 1, 2, 3, \dots, n$ . Definimos o comprimento do fio entre duas células  $c_i$  e  $c_j$  pela Distância Euclideana:

$$L_{ij} = \sum_i \sum_j \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} \quad (3.1)$$

ou pela Distância Manhattan:

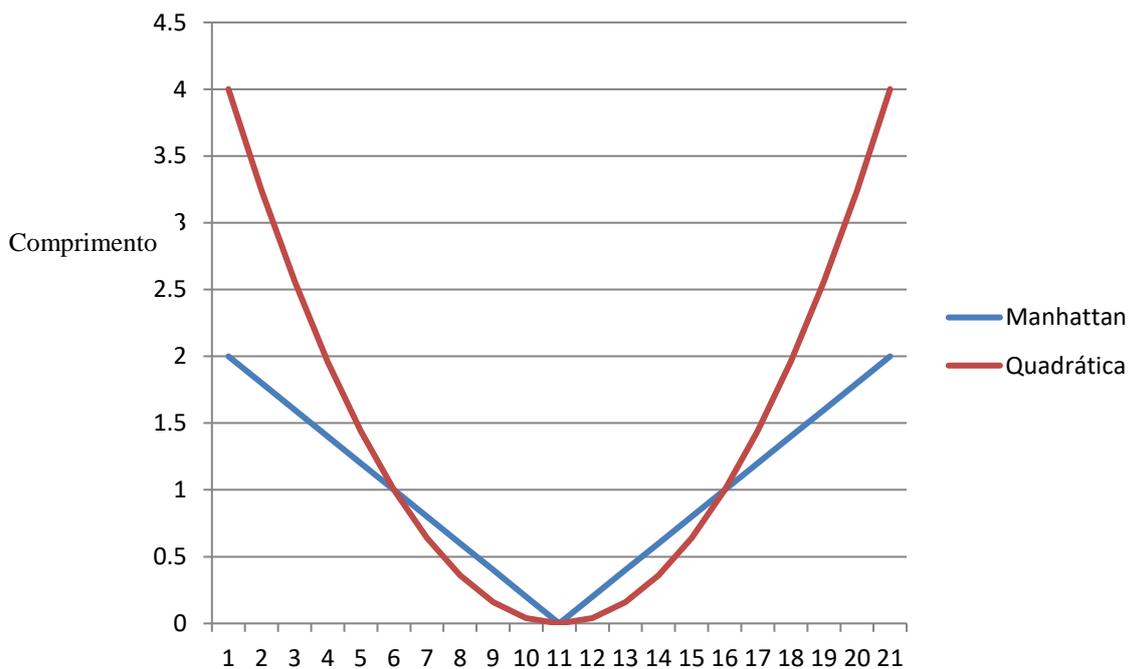
$$L_{ij} = \sum_i \sum_j |x_i - x_j| + |y_i - y_j| \quad (3.2)$$

Porém, as funções acima apresentam problemas na diferenciabilidade em alguns pontos. Logo, na prática elas são regularmente substituídas pela distância Euclideana Quadrática:

$$L_{ij} = \sum_i \sum_j (x_i - x_j)^2 + (y_i - y_j)^2 \quad (3.3)$$

A distância Quadrática possui diferenciabilidade em todos os pontos, como também tem maior inclinação permitindo maior velocidade para o encontro do mínimo, como é visto na figura 3.3.

Figura 3.3 – Comparação entre a distância Quadrática e Manhattan



Fonte: Figura elaborada pelo autor

Porém, a distância quadrática apresenta uma deficiência. Formalmente não pode ser considerada uma “distância”, pois falha na desigualdade triangular. Considere  $x, y, z \in R^n$ . A desigualdade triangular é definida por:

$$d(x, z) \leq d(x, y) + d(y, z) \quad (3.4)$$

A desigualdade 3.4 é um dos requisitos para termos uma função distância bem definida em um espaço métrico (RUDIN, W. 1976). Para eliminar este problema são introduzidas distâncias não lineares como a LOG-SUM-EXP(CHAN, CONG, SZE. 2005):

$$L_i = \alpha \left( \log \left( \sum_{i=1}^n \frac{e^{x_i}}{\alpha} \right) + \log \left( \sum_{i=1}^n \frac{e^{-x_i}}{\alpha} \right) \right) + \alpha \left( \log \left( \sum_{i=1}^n \frac{e^{y_i}}{\alpha} \right) + \log \left( \sum_{i=1}^n \frac{e^{-y_i}}{\alpha} \right) \right) \quad (3.5)$$

Onde  $\alpha \in R$

Ou à distância LP (CHAN, CONG, SZE. 2005):

$$L = \sum (\sum_i (x_k^p)^{1/p} - \sum_i (x_k^p)^{-1/p} + \sum_i (y_k^p)^{1/p} - \sum_i (y_k^p)^{-1/p}) \quad (3.6)$$

Após a definição da distância é preciso elaborar medidas bem definidas para as técnicas de posicionamento serem comparadas. O desempenho de um circuito é medido pela roteabilidade, velocidade de processamento e pelo consumo de energia, dentre outros. Um primeiro fator que se bem trabalhado tem como consequência o bom funcionamento do circuito é o comprimento dos fios (CONG; LUO, 2010). Para isso serão definidos abaixo estimadores para este parâmetro de vital importância:

### 3.2.1 O Halph-Perimeter-Wire-Length (HPWL)

Consiste na menor *bounding box* que contem todas as células de uma *net*. O HPWL nesta *net* é a soma das diferenças entre a maior e a menor coordenadas  $x$  e  $y$  entre seus elementos. O HPWL total do circuito será o somatório do HPWL de cada *net*. Frequentemente utilizada por ter fácil computação e apresentar resultados satisfatórios. Formalmente:

Dados um conjunto de células e uma *netlist* precisamos determinar suas posições minimizando o comprimento médio das ligações (HPWL) respeitando as regras de legalização propostas (AUDIP PANDIT, ALI AKOGLU, 2007) como mostra a equação 3.7 e a Figura 3.4, onde  $\max(x) - \min(x) = W$  e  $\max(y) - \min(y) = H$ .

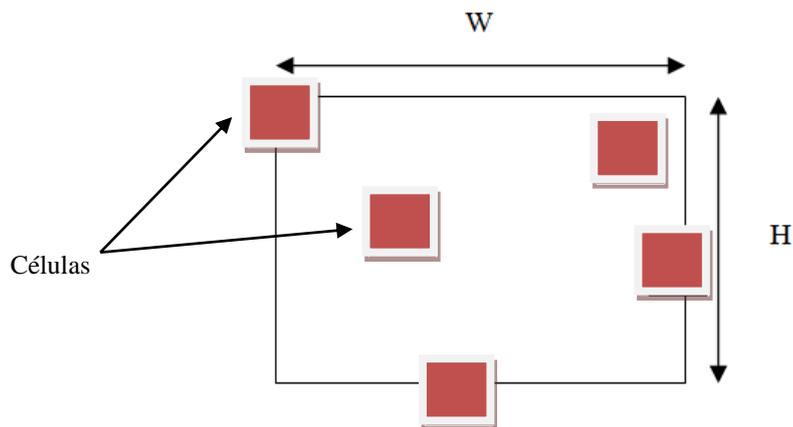
$$\text{HPWL}_{net} = \frac{1}{2} ((\max(x) - \min(x)) + (\max(y) - \min(y))) \quad (3.7)$$

Deste modo o HPWL do circuito será definido pelo somatório dos HPWL de cada *net*.

$$\text{HPWL} = \frac{1}{2} \sum_{j=1}^k (\max(x) - \min(x)) + (\max(y) - \min(y)), \quad (3.8)$$

Onde  $P = \{(x, y) | (x, y) \in net\}$

Figura 3.4- Representação do HPWL

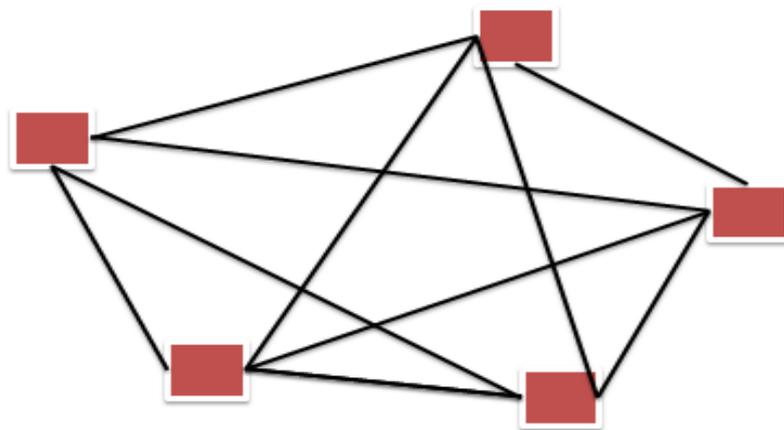


Fonte: Figura elaborada pelo autor

### 3.2.2 Grafo Completo

Nesta medida, ilustrada da Figura 3.5 são calculadas todas as distâncias de todas as células conectadas. Podemos utilizar tanto a distância Manhattan quanto a distância Euclideana. Tem utilização muito menor que a anterior por ter maior número de cálculos, não sendo aconselhável seu uso em circuitos integrados com um número elevado de componentes (KAHNG, A. B. et al, 2011b).

Figura 3.5- Representação do grafo completo

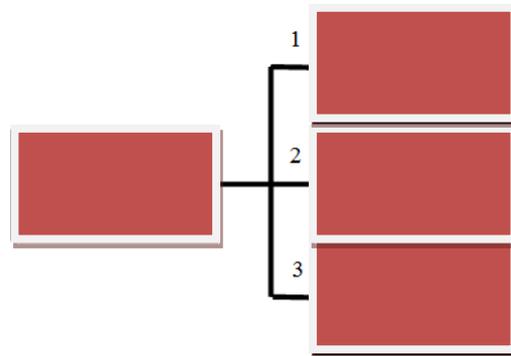


Fonte: figura elaborada pelo autor

### 3.2.3 Saída para Entradas ou Origem para Destino

Uma *net* possui um pino de saída e muitos pinos de entrada. Seu cálculo consiste em medir todos os comprimentos entre a saída e todas as entradas, como é apresentada na Figura 3.6. Muito utilizada para o cálculo do *timing*, por outro lado peca no quesito “Tamanho dos Fios”. Um ponto positivo é sua computação rápida (KAHNG, A. B. et al, 2011b).

Figura 3.6-Representação do método saída para entradas

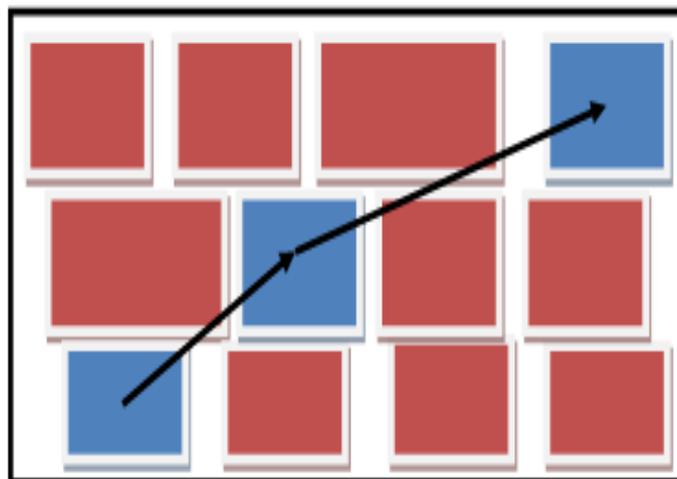


Fonte: figura elaborada pelo autor

### 3.2.4 Menor Caminho

Teoricamente um dos melhores estimadores, sendo descrito na figura 3.7. Na prática não é, pois encontrar o menor caminho é um problema difícil em termos computacional (KAHNG, A. B. et al, 2011b).

Figura 3.7- representação do método menor caminho

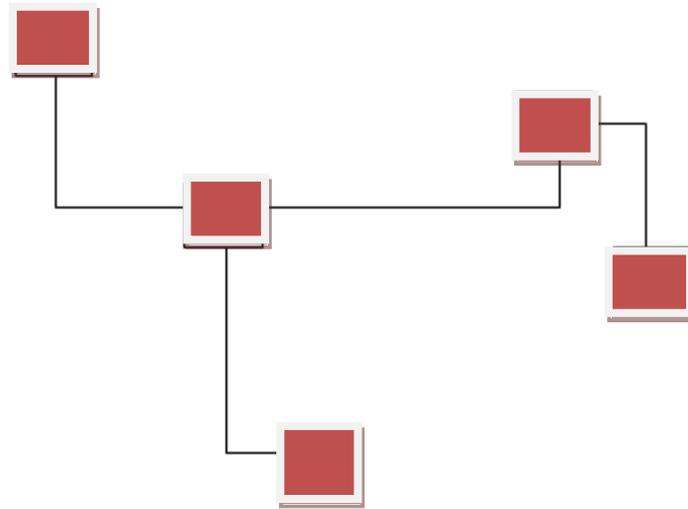


Fonte: Figura elaborada pelo autor

### 3.2.5 Menor *Spanning Tree*

Descrito na figura 3.8, teoricamente é um método eficiente, tanto que é utilizado em alguns algoritmos de roteamento. O lado negativo consiste no lado computacional tal como no item anterior (KAHNG, A. B. et al, 2011b).

Figura 3.8- Modelo de *Spanning Tree*

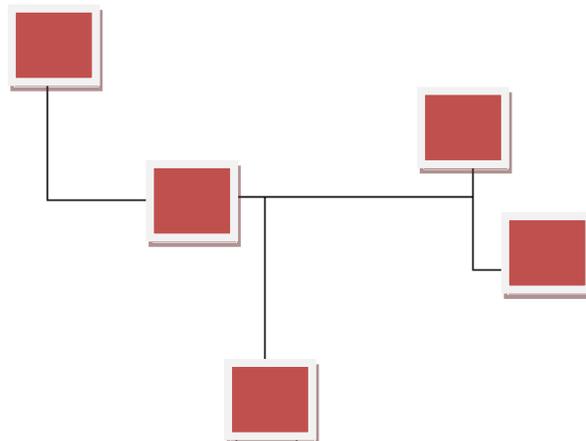


Fonte: Figura elaborada pelo autor

### 3.2.6 Menor *Stainer Tree*

Descrito na figura 3.9 trata-se do objetivo principal do algoritmo de roteamento. É plausível imaginar que seria também a melhor heurística para o algoritmo de posicionamento. O problema, mais uma vez consiste no tempo de execução. Por outro lado, alguns posicionadores utilizam esta heurística para comparação (KAHNG, A. B. et al, 2011b).

Figura 3.9- Modelo de *Stainer Tree*



Fonte: Figura elaborada pelo autor

### 3.3 Modelo das Nets

Dado o conjunto de nets é necessário definir um modelo para suas conexões. Pois o modelo será a base para posteriormente definirmos o sistema linear para a minimização do comprimento de fios. Dentre os vários modelos existentes destacaremos os mais utilizados.

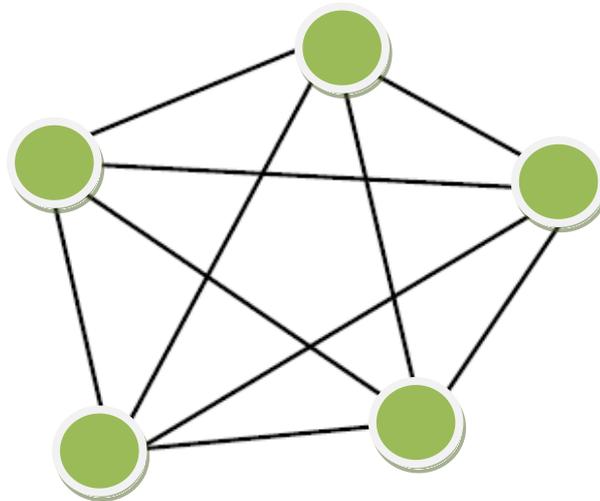
#### 3.3.1 Modelo Clique

Neste modelo, conforme a Figura 3.10, todos os nodos da *net* são conectados entre si. Geralmente é utilizado quando as *nets* possuem um número reduzido de células. Quando o número de células de um circuito é elevado este modelo não é aconselhável devido ao número alto de operações envolvidas aumentando consideravelmente o custo computacional. Por exemplo: para cada componente  $p$  temos  $p-1$  conexões. Pode-se deduzir trivialmente que para uma net com  $P$  componentes teremos:

$$N = \frac{P(P-1)}{2} \quad (3.9)$$

Onde  $N$  é o número total de conexões (VISWANATHAN, N.; CHU, C. C.-N. 2004).

Figura 3.10- Modelo clique

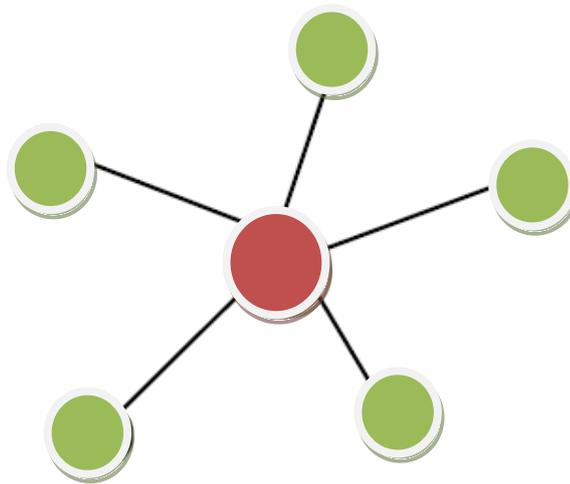


Fonte: (MONTEIRO, 2014)

#### 3.3.2 Modelo Estrela

Neste modelo, conforme a figura 3.11, cada nodo é conectado a um nodo em particular denominado nodo estrela. Utilizado quando as nets possuem um número grande de células. É um método aconselhável quando se analisa questões de *timing*. Se  $p$  é o número de pinos em uma *net* são necessárias somente  $p-1$  arestas (VISWANATHAN, N.; CHU, C. C.-N. 2004).

Figura 3.11- modelo estrela



Fonte: (MONTEIRO, 2014)

Em (VISWANATHAN, N.; CHU, C. C. 2004) foi provado que os modelos em relação ao posicionamento quadrático são equivalentes, além de apresentarem um modelo Híbrido, que em resumo é uma amálgama entre os modelos clique e estrela. A principal vantagem do modo estrela caracteriza o aumento de zeros na matriz que será usada no sistema linear diminuindo o tempo de execução.

Formalização matemática: Seja  $N$  uma *net*. Em relação ao pino estrela, seu ponto de equilíbrio está localizado no centro de gravidade da *net* denominado baricentro. Consideremos uma *net* de  $n$  nodos e  $x_s$  a coordenada do pino estrela e  $W_s$  o peso desta conexão. O cálculo da força total entre todos os nodos será:

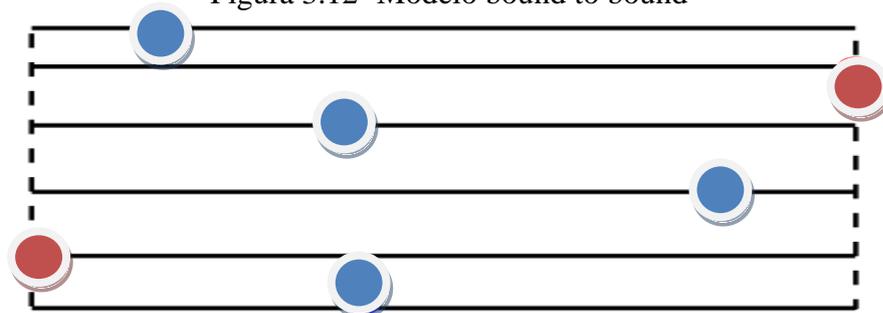
$$F = \sum_{j=1}^n W_s (x_s - x_j) \quad (3.10)$$

Onde  $W_s = \frac{1}{N-1}$  tal que  $N = \#net$ .

### 3.3.3 Modelo *Bound to Bound*

Aqui é construída uma *bound box* na rede, como é apresentado na Figura 3.12. As conexões entre as células internas são removidas restando somente as conexões entre as células internas e a fronteira (KAHNG, A. B. et al, 2011b).

Figura 3.12- Modelo bound to bound



Fonte: (MONTEIRO, 2014)

### 3.4 Caminhos Críticos

Um caminho crítico é aquele formado pelas células que possuem maior atraso de propagação do sinal. Obviamente um comprimento longo prejudica o desempenho de um circuito aumentando o tempo de processamento. Logo para um bom posicionamento precisamos identificar os caminhos críticos, diminuí-los e ter todo o cuidado para não aumentarmos os outros caminhos tornando-os novos caminhos críticos (HOU, W; HONG X. 2003).

As técnicas que atacam problemas de desempenho são chamadas *Timing Driven* (dirigidas a desempenho) e são divididos em:

- *Net-based*: Estes algoritmos focam o pior caso possível ou definem um peso para cada net construindo escalabilidade para controlar o atraso e assim são aplicados em todas as redes do circuito.
- *Path-based*: estes algoritmos trabalham diretamente nos caminhos críticos (gerados ou recebidos). A vantagem consiste em um menor tempo de execução. Por outro lado, não são menos escaláveis em relação aos anteriores. São aplicados em pequenos circuitos ou em parte de um circuito maior.

### 3.5 Congestionamentos

O congestionamento é a relação entre o número de conexões e o espaço definido para elas. Se uma região está congestionada se faz necessário criar outras rotas para os fios. Neste caso o comprimento dos mesmos é aumentado. Por outro lado existe a possibilidade de não haver possibilidade de novas rotas.

Minimizar o comprimento dos fios não é o único problema a ser tratado. É preciso também evitar a alta densidade de células em regiões do circuito. O minimizar dos comprimentos diminui indiretamente os congestionamentos porem, matematicamente não há garantias de que não existam regiões com alta densidade influenciando negativamente o roteamento, às vezes o impossibilitando (TAGHAVI, T. et al, 2007).

### 3.6 Atrasos

A frequência máxima de relógio que o sistema pode atingir está diretamente relacionada pelo comprimento dos fios. Inicialmente era necessário apenas trabalhá-la observando somente as portas lógicas. Hoje, devido ao aumento significativo do atraso nas conexões e do numero de componentes, esse quadro mudou drasticamente (BHASKER, J; CHADHA, R., 2009).

### 3.7 Consumo de Energia

Em relação ao consumo de energia, o comprimento dos fios é um fator importante. A capacitância de uma rede interconectada é proporcional ao seu comprimento. Deste modo para trabalhar o consumo de energia, a principio é necessário trabalhar no comprimento dos fios (WANG, L.-T; CHANG, Y. - W; CHENG, K. - T. T, 2009).

### 3.8 Distribuição do Calor

O posicionamento pode ser direcionado às regiões de calor. A distribuição heterogênea do calor no circuito pode gerar problemas de confiabilidade e também comprometer o tempo de vida do dispositivo (WANG, L.-T; CHANG, Y. - W; CHENG, K. - T. T, 2009).

### **3.9 Resumo do Capítulo**

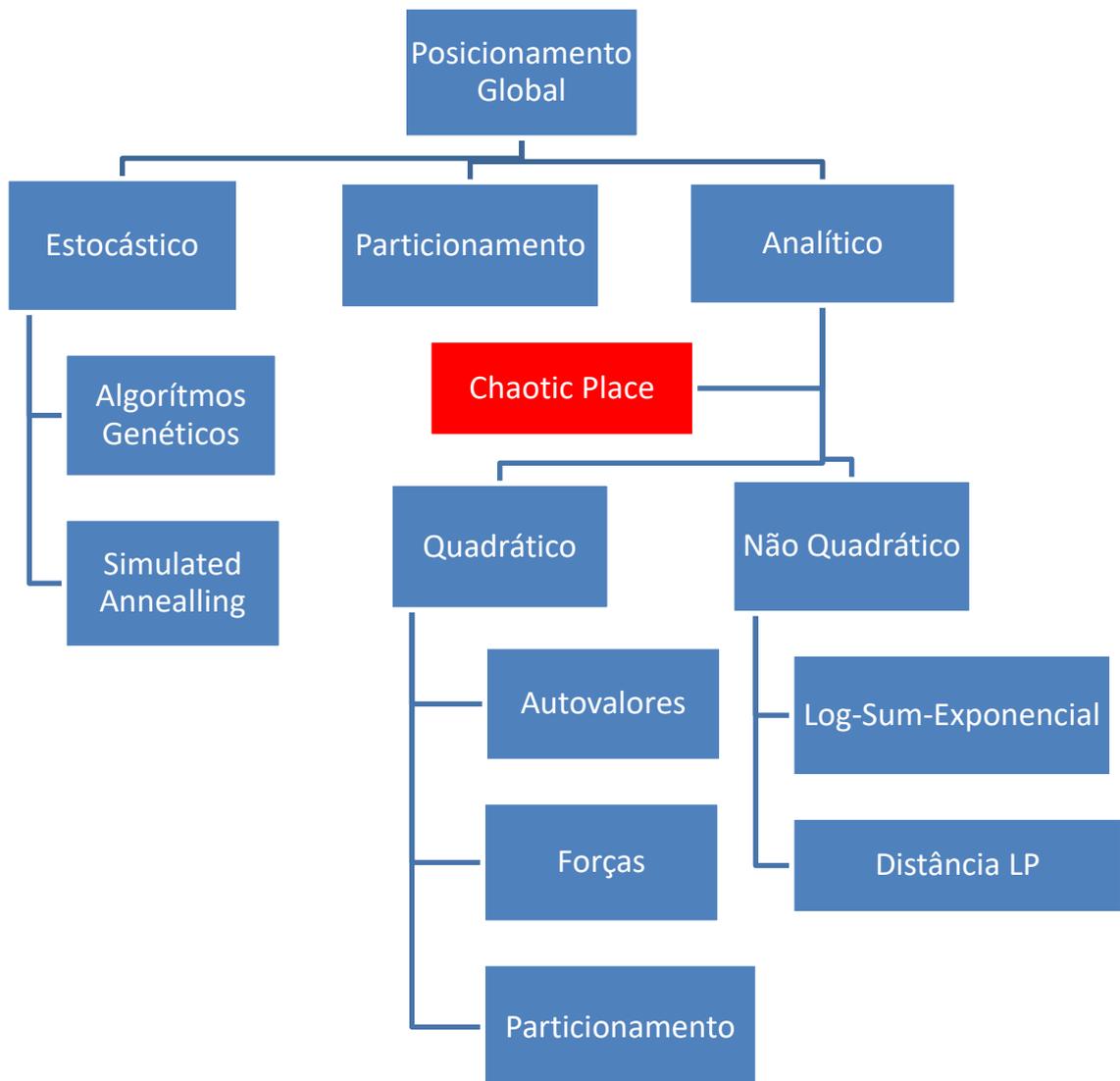
Neste capítulo foi definida a etapa do posicionamento de células em circuitos integrados. Foram apresentadas as métricas para avaliar o posicionamento, e por último foram apresentados os problemas que um bom posicionamento pode evitar: congestionamentos, atrasos e consumo de energia.

No próximo capítulo será exibida uma visão geral no estado da arte, ou seja, os posicionadores que apresentam melhores resultados em cada caso além da inserção de nosso trabalho dentro do contexto atual.

## IV O ESTADO DA ARTE

No decorrer dos anos foram pesquisados e desenvolvidos diversos algoritmos de naturezas distintas para resolver o problema do posicionamento. Estes algoritmos estão divididos em classes apresentadas no fluxograma da Figura 4.1 contextualizando o trabalho realizado neste texto:

Figura 4.1- Algoritmos de posicionamento e o estado da arte



Fonte: Figura elaborada pelo autor

### 4.1 Estocástico.

Em Teoria da Probabilidade dizemos que um processo é estocástico se este é uma família de variáveis aleatórias que representam a evolução de um sistema (MAÑE R. 1983). Dizemos que um algoritmo é estocástico se este é desenvolvido seguindo um processo onde

há várias direções (em alguns casos infinitos) por onde o processo irá evoluir. Diferente das equações diferenciais, por exemplo. Dentre os algoritmos estocásticos, para posicionamento podemos citar:

#### 4.1.1 Simulated Annealing

Tem sua base e inspiração em um processo termodinâmico geralmente utilizado na metalúrgica: A princípio o sólido é aquecido à temperatura de 1000 graus centígrados, em seguida a temperatura é diminuída gradualmente até que o material se solidifique (BETZ V, ROSE J. 1997). Nesta fase seus átomos organizam-se de forma que a energia tenda ao mínimo possível.

Dada  $F$  uma função objetivo, o algoritmo inicia com uma solução aleatória  $S$ . Em um laço de repetição o processo ocorre para gerar nova solução  $S'$ . Então para cada par  $S$  e  $S'$  é testada a Função objetivo através da diferença:

$$\Delta = F(S) - F(S') \quad (4.1)$$

O algoritmo para quando o objetivo final é alcançado.

---

#### Algoritmo: Simulated Annealing

---

```

temperatura = temperatura Inicial ;
posicionamento = posicionamento Inicial;
enquanto (condição para continuar o loop)
    enquanto (condição para continuar o loop)
        novo Posicionamento = Perturbação(posicionamento)
        delta = Custo(novo posicionamento)
    Custo (posicionamento);
    se( Aceita ( delta, temperatura ) ) então
        Posicionamento = novo Posicionamento;
    fim se
fim enquanto
temperatura = Histórico( temperatura )
fim enquanto
return Posicionamento

```

---

O algoritmo parte de uma solução aleatória e a partir da troca de posições das células é minimizado o HPWL, onde somente são aceitas trocas que minimizem o tamanho das ligações.

Mesmo tendo sua força perdida nos dias de hoje devido à introdução de novas técnicas ainda se faz presente com ótimos resultados, em particular para FPGAs. O estado da arte para o posicionamento de células em FPGAs homogêneos é o VPR (ASGHAR, PARVEZ, 2015).

#### 4.1.2 Algoritmos Genéticos

É um caso particular do que denominamos Computação Evolutiva. Esta área tem como inspiração as Ciências Biológicas. São utilizados com frequência para solucionar problemas que envolvem otimização e busca (COLLIER R. et al, 2014). Um exemplo notável

é quando são utilizados para a resolução do *Traveling Salesman Problem* (TSP), ou o Problema da Caixa (HUSSAIN A. et al. 2017).

O algoritmo genético parte de uma solução (denominada população) aleatória e a partir dela cada etapa do algoritmo (geração) gera novas soluções onde sofrem modificações (mutações, crossover) e soluções piores são descartadas, restando as melhores. O algoritmo para quando encontramos uma solução que satisfaça o problema.

Um algoritmo genético é composto dos seguintes elementos:

- **Função Objetivo:** É a função que desejamos minimizar ou maximizar dentro de um espaço possível de soluções.
- **Indivíduo:** Uma possível solução dentro da população (espaço de soluções bem definido)
- **Seleção:** Utilizamos alguma heurística para descartar os indivíduos menos aptos selecionar os mais aptos, e estes irão gerar as novas populações.
- **Reprodução:**
  - **Mutação:** Um gene do indivíduo é sorteado aleatoriamente com uma probabilidade e seu valor é modificado.
  - **Cruzamento:** A lista de indivíduos selecionada é aleatoriamente permutada criando uma segunda lista. Então cada indivíduo da primeira lista é cruzado com o da segunda lista na mesma posição.

---

**Função Algoritmo Genético (população, função-objetivo) saídas: indivíduo**

---

entradas: população → uma lista de indivíduos

função-objetivo → uma função que recebe um indivíduo e retorna um número real.

repetir

lista de pais = seleção(população, função-objetivo)

população = reprodução(lista de pais)

enquanto nenhuma condição de parada for atingida

retorna o melhor indivíduo da população de acordo com a função-objetivo

---

## 4.2 Particionamento

Para estes algoritmos segue a premissa: “Dividir para conquistar”. Um algoritmo em tempo polinomial que encontre uma solução ótima é desconhecido onde este problema é caracterizado na classe NP - completo. Toma-se toda a área de um circuito sendo dividida em partições, conforme a Figura 4.2. Partições são retângulos disjuntos onde armazenarão as células, módulos e as nets que as conectam. Cada partição é novamente subdividida e assim sucessivamente até que a partição contenha apenas uma célula (CRISTINEL ABABEI, 2009).

Figura 4.2- Etapas do particionamento



Fonte: Figura elaborada pelo autor

Este método tem a vantagem de gerar menores congestionamentos facilitando a etapa seguinte do roteamento. Também apresentam melhor tempo de execução apresentando resultados inferiores quando se trata dos comprimentos de fio se comparados ao estado da arte em geral. Neste método destacamos o CAPO (JARROD A. ROY et al, 2007) e o DRAGON (TAGHAVIT, X. YANG, 2007).

### 4.3 Analítico

São algoritmos que minimizam uma função custo utilizando funções matemáticas. O princípio básico é definir algebricamente uma função custo e por algum método de otimização combinatória realizar o processo de minimização da mesma. Os algoritmos são divididos em duas classes:

#### 4.3.1 Quadráticos

Dos algoritmos analíticos é o mais citado na literatura e como consequência, se for desconsiderado o *Simulate Annealing* é o que possui maior número de aplicações tanto em ASICs quanto em FPGAs (XU M., GREWAL G., AEIBI S., 2011). Considere um grafo  $G(v, e)$ , onde “ $v$ ” representam os (nós) e “ $e$ ” representam as (arestas) com  $v_i = (x_i, y_i)$

Para todas as células do circuito é definida a função  $\Phi: M \rightarrow M$  onde o objetivo é minimizá-la seguindo as restrições que variam em cada caso.

$$\Phi(X, Y) = \frac{1}{2} \sum_{i,j=1}^n c_{x,ij} (x_i - x_j)^2 + c_{y,ij} (y_i - y_j)^2 \quad (4.2)$$

Onde  $c_{x,ij}$  e  $c_{y,ij}$  são os pesos entre as conexões.

E sua representação em forma matricial:

$$\Phi(X) = \frac{1}{2} x^T Q_x x + c_x^T d + constante \quad (4.3)$$

Onde  $Q_x$  é a matriz que representa as conexões células moveis e  $d$  representa o vetor que representa as conexões entre as células moveis e as fixas (I/Os, por exemplo) (STRUZYNA M, 2013).

#### 4.3.2 Autovalores

Neste posicionamento se supõe  $d=0$  na equação 4.3 e é assumido que todas as células são moveis, e  $x^T x = constante$ . Podem-se combinar as suposições com a função  $\Phi$  (definida no posicionamento quadrático) e por relaxação Lagrangeana é obtida uma nova função cujo mínimo é encontrado quando sua derivada em relação a  $x$  e a  $y$  for nula resultando:

$$Q_x x - \lambda x = 0 \quad (4.4)$$

A equação supra é utilizada para serem obtidos os autovalores e os autovetores de  $Q_x$ , donde vem o nome do método. Computacionalmente encontrar autovalores e autovetores é uma tarefa árdua. Por isso este método é pouco utilizado em trabalhos atuais para o

posicionamento de células em circuitos demasiadamente grandes, porém em (FLACH, GHILHERME AUGUSTO, 2015) este problema foi sumariamente contornado.

### 4.3.3 Particionamento

Este posicionamento divide o circuito em partições similar ao *mini cut* e em cada etapa minimiza uma função de custo quadrática (KAHNG et al. 2011).

### 4.3.4 Forças

Com inspiração na Física, mais precisamente no estudo dos sistemas de molas, estes algoritmos geram bons resultados em todos os parâmetros. Considerando o conjunto de células a ser posicionado, cada célula exerce uma força de atração em todas as outras sendo que esta força é diretamente proporcional ao peso de cada conexão entre as células. O processo acaba quando o ponto de equilíbrio é alcançado, ou seja, quando isto ocorre o comprimento mínimo das conexões é obtido.

Como é de se esperar, as células em sua maioria, no final do processo estão concentradas em uma região, daí é utilizada uma técnica de espalhamento (introduzindo no processo as chamadas forças de espalhamento) aonde os comprimentos dos fios vão aumentando lentamente, o número de sobreposições é diminuído e com o cuidado de não mudar drasticamente a topologia das células obtidas no processo inicial.

Dentre estes posicionadores destacamos o MAPLE (KIM M.-C., VISWANATHAN N., ALPAER C. J., MARKOV I. L., e RAMIJI S, 2012), ComPLx (KIM M.-C. e MARKOV I.L, 2012). SimPL(LEE D.-J., KIM M.-C. and MARKOV I. L, 2012) e o BonnPlace(BRENNER U. et al. 2008).

### 4.3.5 Não Quadráticos

Apresentam melhor desempenho se comparados aos outros posicionadores analíticos diminuindo o tamanho das *nets* de modo mais eficiente que os quadráticos. Também possuem menor *runtime* se comparados ao demais. Aqui o sistema de equações é composto por funções não lineares como exponenciais, logarítmicas, somatórios, transformadas, etc.

No processo de otimização são utilizados métodos como gradiente, gradiente conjugado, gradiente conjugado não linear, por exemplo. Dentre os não quadráticos podemos destacar o LOG-SUM-EXP(LSE) (LU J., CHEN P., CHANG C.-C., SHA L., HUANG D. J.-S., TENG C.-C. e CHENG C.-K, 2014) definido pela equação 4.5.

$$LSE_{\alpha} = (z_1, z_2, z_3 \dots, z_n) = \alpha(\log(\sum_{i=1}^n e^{z_i} / \alpha)) + \alpha \left( \log \left( \sum_{i=1}^n \frac{e^{-z_i}}{\alpha} \right) \right) \quad (4.5)$$

Logo o HPWL para cada net definido anteriormente será aproximado pela função:

$$LSE_{net} = \alpha(\log \left( \sum_{i=1}^n \frac{e^{x_i}}{\alpha} \right) + \log \left( \sum_{i=1}^n \frac{e^{-x_i}}{\alpha} \right)) + \alpha \left( \log \left( \sum_{i=1}^n \frac{e^{y_i}}{\alpha} \right) + \log \left( \sum_{i=1}^n \frac{e^{-y_i}}{\alpha} \right) \right) \quad (4.6)$$

A função descrita pela equação 4.6 é contínua, estritamente convexa e possui derivada em todos os pontos e converge para o  $HPWL_{net}$  quando  $\alpha \rightarrow 0$ .

Como exemplo de posicionamento não linear destacamos os posicionadores APlace3(KAHNG, A. B. et al, 2005), NTUPlace3 (C. CHEN, Z.-W. JIANG, T.-C. HSU, H.-C. CHEN, 2008) e mPL6 (CHAN, T. et al. 2007).

O Estado da arte do posicionamento global é o algoritmo e-Place (LU J., CHEN P., CHANG C.-C., SHA L., HUANG D. J.-S., TENG C.-C. e CHENG C.-K., 2014). Este posiciona e espalha as células simulando um campo eletrostático reduzindo ao mínimo a sobreposição melhorando os resultados anteriores.

Neste trabalho será utilizada a função LSE simplificada. Realizando a otimização considerando somente um eixo.

#### 4.4 Novas Técnicas de Posicionamento

É natural pensar que se temos um problema importante mais e mais pesquisadores estão trabalhando no mesmo e por sua vez introduzindo novas ferramentas além de melhorar as já existentes. Redes Neurais foram utilizadas em (ANDRZEJ KOS, ZBIGNIEW NAGÓRNY, (2009)) e (AYKANAT C., BULTAN T, HARITAOGLU I. 1998) com resultados satisfatórios. Em paralelo (ASGHAR A. and PARVEZ H. 2015) utilizou conceitos de Termodinâmica como Difusão para o Posicionamento Detalhado com o objetivo de diminuir o congestionamento entre as ligações. Em (CHAN, J. CONG, 2005) foram apresentados Posicionadores utilizando Equações Diferenciais parciais generalizando o posicionamento quadrático. No Trabalho (FLACH, GHILHERME AUGUSTO, 2015) foi proposto Relaxação Lagrangeana para o estudo do dimensionamento dos transistores do circuito.Paralelismo no intuito de acelerar o tempo do posicionamento também foi utilizado em (MATTHEW A., STEFFAN J., BENTZ V, 2014).Observaremos nos parágrafos seguintes será constatado que de forma implícita os Sistemas Dinâmicos, que serão formalmente definidos no capítulo a seguir, se fazem presentes em cada um desses trabalhos.

#### 4.5 Resumo do Capítulo

Neste capítulo foi apresentada a definição e a formalização da etapa posicionamento. Seus tipos, propriedades, os posicionadores mais utilizados, como também uma visão sobre o estado da arte. Parâmetros que devem ser observados foram analisados e por último foram citadas as métricas para serem utilizadas por um posicionador. No próximo capítulo serão apresentadas as ferramentas que são os principais componentes deste trabalho. Os Sistemas Dinâmicos não Lineares.

## V Sistemas Dinâmicos

Diversos fenômenos naturais, mesmo regidos por simples equações apresentam comportamento caótico. A presença de variáveis em um primeiro momento desprezíveis acaba por gerar comportamentos imprevisíveis no sistema. A teoria do caos (sistemas dinâmicos não lineares) é composta por uma gama de ferramentas oriundas de várias, senão todas as áreas da matemática, e sua generalidade permite que possa ser utilizada na solução de problemas de caráter linear ou não (MAY ROBERT M. 1976).

No fim do século XIX Poincaré, em seus trabalhos envolvendo mecânica celeste percebeu que para certas equações diferenciais não era possível encontrar um entendimento pelos métodos quantitativos. Daí Poincaré introduziu outros elementos não convencionais da matemática para atacar tais problemas como Álgebra, Topologia e Geometria Diferencial. Na década de trinta, Andronov e Pontrajim iniciaram o estudo da “estabilidade estrutural” que consiste em identificar quais sistemas dinâmicos submetidos a pequenas perturbações mantêm suas características topológicas preservadas. No mesmo período o matemático russo Alexander Lyapunov demonstrou teoremas que abordam a estabilidade assintótica dos sistemas dinâmicos.

Na década de sessenta, Mauricio Matos Peixoto demonstrou, para superfícies bidimensionais, quais sistemas gozam da estabilidade estrutural (PEIXOTO, M. M. 1959). Ainda na década de sessenta, o físico Lorenz iniciou o estudo dos atratores estranhos ao trabalhar em problemas de previsão do tempo (BONATTI C, DIAZ L.; VIANA M, 2005).

Em 1975 o matemático James Yorke usou pela primeira vez o vocábulo “Caos” no sentido de uma “desordem ordenada”, em outras palavras, uma ordem em padrões aparentemente aleatórios (Li T. Y and Yorke J.A. 1975). No final dos anos sessenta a teoria obteve um avanço considerável através de Stephen Smale e Jacob Palis, onde demonstraram diversos teoremas fundamentais além de enunciarem diversas conjecturas (PALIS, JACOB; SMALE, S., 1970). A mais importante chamada Conjectura da Estabilidade Estrutural foi demonstrada em 1987 por um ex-aluno de doutorado do Matemático Jacob Palis: Ricardo Mañe (MAÑE R, 1987). Sistemas dinâmicos estão divididos em duas classes:

- Conservativos  
Aqueles que conservam a energia, em outras palavras, a energia do sistema é constante. Na prática esta medida dependerá da natureza do sistema. Ex: temperatura, pressão, volume, etc.
- Dissipativos  
Aqueles em que há perda de energia através de fatores dissipativos. Ex: queda de temperatura, diminuição do volume, etc.

Sistemas dinâmicos são definidos formalmente: Seja  $f: M \rightarrow M$  uma função definida em  $M$ . Para facilitar nossa compreensão, nossas necessidades e sem perda de generalidade vamos admitir que  $f$  é diferenciável e  $M$  é uma superfície compacta, isto é: fechada e limitada. Dado  $x \in M$  a órbita de  $x$  pode ser definida pelo conjunto:

$$Ox = \{f^i(x), i \in \mathbb{Z}\} \quad (5.1)$$

Um sistema dinâmico pode ser discreto (uma função) ou contínuo (um campo de vetores). Aqui serão exemplificados e utilizados somente os sistemas discretos.

O objetivo principal da teoria dos sistemas dinâmicos é estudar o comportamento do conjunto  $Ox$ , para todo  $x \in M$  quando  $i$  tende a valores arbitrariamente grandes onde o estado alcançado é denominado estado assintótico. Definimos por espaço de fase o conjunto onde o sistema dinâmico está bem definido.

Não menos importante A Teoria Ergódica, uma subárea dos sistemas dinâmicos tem por finalidade o estudo da dinâmica de uma transformação no sentido probabilístico. Muitos problemas de caráter geométrico obtiveram solução através desta abordagem (BONATTI C, DIAZ L.; VIANA M, 2005). Neste trabalho a Teoria Ergódica não será abordada.

## 5.1 Efeito Borboleta

Característica marcante nos sistemas caóticos. Denominada por sensibilidade às condições iniciais. Consideremos um sistema de  $n$  variáveis e um conjunto de condições iniciais. Após um tempo considerável o sistema irá adquirir seu estado assintótico. Agora tomando o mesmo sistema e realizando uma sutil perturbação numa ou mais das condições iniciais, o comportamento do mesmo sistema após o mesmo tempo será completamente diferente do primeiro caso. Um exemplo clássico são os fenômenos da natureza. Por isso previsões do tempo são feitas com certa precisão em um intervalo de tempo pequeno, sendo de grande dificuldade prever quando uma região do planeta será atingida por um ciclone, por exemplo. (BOEING, G. 2016).

Formalmente: Dizemos que um sistema dinâmico tem sensibilidade às condições iniciais se existe um  $x$  no espaço de fase tal que, para todo  $y$  suficiente próximo de  $x$  no início do processo, para  $n$  suficientemente grande  $f^n(x)$ , e  $f^n(y)$  estão distantes.

## 5.2 Fractais

Informalmente pode-se definir um Fractal como um objeto da geometria não euclidiana dividido em partes onde cada parte se assemelha ao objeto inteiro. Podem ser encontrados na natureza e aparecem com frequência em diversas aplicações das Ciências Físicas (PALIS, JACOB; TAKENS, FLORIS. 1995).

Exemplo 1: O conjunto de Cantor.

Considere o intervalo  $I = [1,0]$ .

Deve-se retirar o terço médio do intervalo acima obtendo dois intervalos fechados:

$$A_1 = \left[0, \frac{1}{3}\right] \cup \left[\frac{2}{3}, 1\right]$$

Segue-se retirando o terço médio de cada um dos dois intervalos:

$$A_2 = \left[0, \frac{1}{9}\right] \cup \left[\frac{2}{9}, \frac{3}{9}\right] \cup \left[\frac{6}{9}, \frac{7}{9}\right] \cup \left[\frac{8}{9}, 1\right]$$

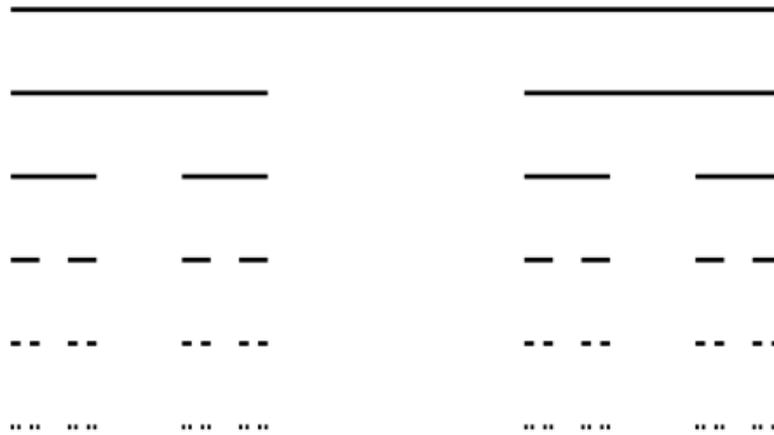
Seguindo infinitamente o processo iterativo é obtido o conjunto resultante:

$$C_{1/3} = \bigcap_{n=1}^{\infty} A_n \quad (5.2)$$

O Conjunto de Cantor, ilustrado na Figura 5.1, será a interseção de todos os conjuntos do processo descrito.

Geometricamente o Conjunto de Cantor possui um comportamento patológico. É totalmente desconexo, porém existe uma relação bijetora entre ele e os números reais.

Figura 5.1- Conjunto de Cantor



Fonte: Figura elaborada pelo autor

O conjunto resultante chamado Maximal-Invariante é um exemplo de fractal (PALIS, JACOB; TAKENS FLORIS, 1995).

Formalmente: O conjunto de Cantor ( $K$ ) no intervalo  $[0,1]$  é o conjunto que pode ser escrito na base 3 utilizando apenas os algarismos 0 e 2.

$$K = \left\{ \sum_{n=1}^{\infty} \frac{\sigma_n}{3^n}, \sigma_n \in \{0,2\} \text{ para todo } n \in \mathbb{N} \right\} \quad (5.3)$$

Exemplo 2:

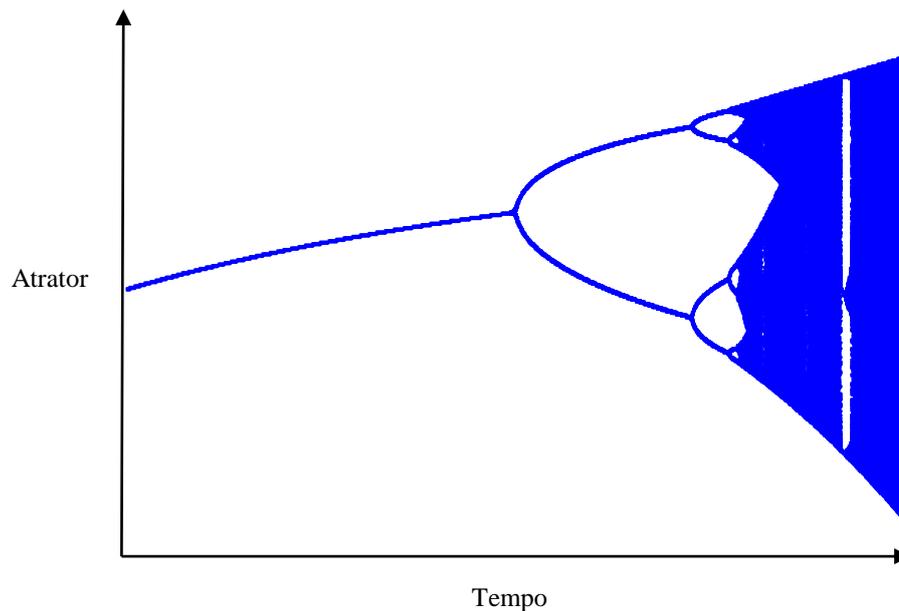
Mapa Logístico: Definido por uma regra onde para cada  $x_n$  há uma associação

$$x_{n+1} = rx_n(1 - x_n) \quad (5.4)$$

Onde  $r$  é um parâmetro que regula a velocidade do processo.

Utilizado pela primeira vez em (MAY ROBERT M, 1976) pelo biólogo Robert May no estudo da evolução populacional de formigas. Tem-se na figura 5.2 o gráfico do Mapa Logístico, onde a imagem do atrator pertence ao intervalo  $[0,1]$  e o tempo varia entre 0 e 4 unidades.

Figura 5.1- Diagrama de Bifurcação do Mapa Logístico



Fonte: Figura elaborada pelo autor

### 5.3 Estabilidade de Lyapunov

Diz-se que um sistema dinâmico é estável, segundo Lyapunov, se para um número suficientemente grande de interações, ele tende a um ponto de equilíbrio. A estabilidade de Lyapunov caracteriza qualquer processo de otimização.

Considere  $f: R^n \rightarrow R^n$  um sistema dinâmico diferenciável (todas as suas derivadas estão bem definidas em todos os pontos do domínio) e  $x_e$  um ponto de estabilidade. Dado  $\varepsilon > 0$  existe um  $n_0$  tal que para todo  $n > n_0 \Rightarrow \|f^n(x) - x_e\| < \varepsilon$ .

Exemplo: Os posicionamentos analíticos definidos no capítulo anterior.

### 5.4 Pontos de Equilíbrio

Seja  $f: M \rightarrow M$  um sistema dinâmico diferenciável. Um ponto de equilíbrio  $x_{eq}$  é um estado no qual  $Grad(f) = 0$ . Onde  $Grad(f)$  é o Vetor Gradiente de  $f$ . Em outras palavras: Seja  $\varepsilon > 0$ . Se  $f(x_0) = x_{eq}$ , então o estado do sistema será:  $f(x) = x_{eq}$ , para todo  $x$  onde  $\|x - x_0\| < \varepsilon$ .

### 5.5 Atratores

São definidos como o conjunto de estados que um sistema dinâmico tende após um número suficientemente grande de interações (PALIS, JACOB, 2005). São definidos quatro tipos de Atratores:

- Pontos Fixos: Tendem a um estado estável. Não tem sensibilidade às condições iniciais e não possuem comportamento caótico.
- Ciclos-limite: São atratores cíclicos, não caóticos e não são sensíveis a pequenas perturbações.

- Toro: É uma superfície homeomorfa ao produto entre dois círculos. Aparecem em sistemas com alto grau de liberdade.
- Atratores Estranhos: Apresentam uma geometria complexa, não euclidiana altamente irregular. São oriundos de sistemas dissipativos e não periódicos. Caracterizam o que denominamos Caos.

## 5.6 Atratores Estranhos

Possuem estrutura fractal e surgem após um número considerável de interações em um sistema dinâmico como os demais atratores. Sobre os efeitos produzidos, tais atratores revelam-se extremamente sensíveis às mais simples variações nas condições iniciais do seu desenvolvimento, a medida que as interações avançam ao longo do tempo, assim será desenvolvido certo padrão de desordem. Possuem dimensão fractal e modelam diversos fenômenos naturais e de caráter computacional (PALIS JACOB, 2005).

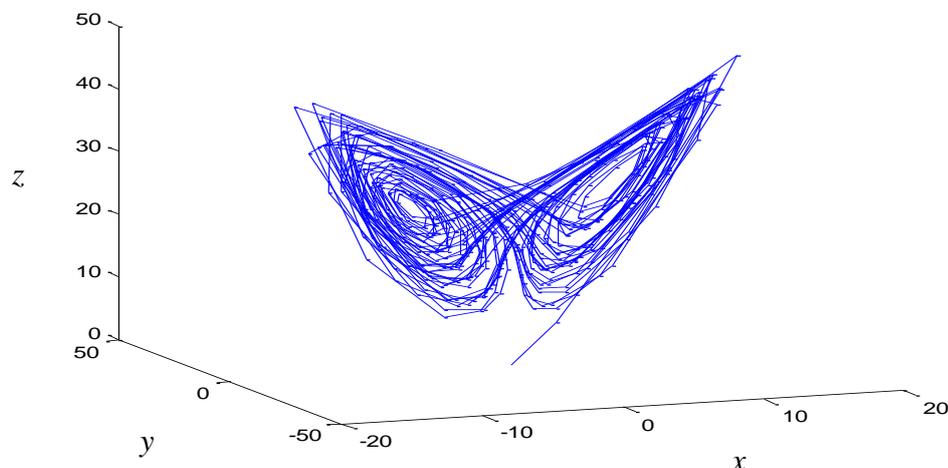
O estudo dos atratores estranhos teve seu início quando o físico Edward Lorenz, através de métodos computacionais ao estudar o comportamento da atmosfera questionando a fundamentação das técnicas de previsão do tempo da época percebeu que mudando suavemente as condições iniciais das equações diferenciais obtinha resultados completamente diferentes dos resultados anteriores.

Ao simplificar as equações de Navier-Stokes obtendo o sistema de equações 5.5 notou que o sistema possuía as seguintes características: Não linearidade, determinístico, e de comportamento caótico. Onde posteriormente foi batizado de atrator estranho, descrito na Figura 5.3 onde o espaço tridimensional é representado pelos eixos  $x$ ,  $y$  e  $z$  cujos valores são o espaço de fase do sistema dinâmico.

$$\frac{dx}{dt} = \sigma(y - x), \quad \frac{dy}{dt} = x(\rho - z) - y, \quad \frac{dz}{dt} = xy - \beta z \quad (5.5)$$

$$\text{Tomando } \sigma = 8, \rho = 28 \text{ e } \beta = -\left(\frac{8}{3}\right)$$

Figura 5.3- Atrator de Lorenz



Fonte: Figura elaborada pelo autor

Desta forma após a minimização das distâncias dos pontos definidos pela equação (3.6), vamos fazer o levantamento no sentido: Dados

$$X = \{x_1, x_2, x_3, x_4, \dots, x_n\} \quad (5.6)$$

A solução para a minimização de (3.6) devemos obter uma  $F: R \rightarrow R^n$ , denominada levantamento de  $X$ . Para isso utilizaremos o célebre teorema proposto pelo matemático Floris Takens em 1981.

## 5.7 Teorema de Takens

O teorema é uma ferramenta valiosa para Análise de Séries Temporais em geral através de um método que reconstrói o espaço de fases obtendo as informações de um sistema dinâmico a partir de uma variável ou funções dessa variável preservando a dinâmica do espaço original.

Considere  $F: R^n \rightarrow R^n$ , um sistema dinâmico  $n$ -dimensional, e sem perda de generalidade é suposto  $F$  diferenciável. Agora considere  $P: R^n \rightarrow R$ , a projeção de  $F$  no eixo real. Pergunta: Seria possível obter o sistema dinâmico original a partir da Imagem de  $P$ ?

Formalmente conhecido como o “Teorema do Mergulho de Takens”. Foi publicado em 1981 pelo matemático Floris Takens (F. TAKENS, 1981) e garante que as propriedades do sistema dinâmico original são recuperadas como expoentes de Lyapunov, dimensão de Hausdorff, entropia, geometria limite, etc.

O teorema possui aplicações em diversas áreas do conhecimento sendo largamente utilizadas nos estudos de Séries Temporais para Economia, Física, Neurociências (PETRY, A.; BARONE, D. A. 2003), por exemplo.

Considere um sistema dinâmico. Se suas equações diferenciais são conhecidas é possível caracterizar seu atrator utilizando seus expoentes de Lyapunov, que serão apresentados a seguir. Por outro lado, em problemas reais nem sempre as equações serão conhecidas.

Seja  $f: M \rightarrow M$  um sistema dinâmico discreto com  $M \subset R$  e  $M$  um intervalo compacto e conexo. O Teorema de Takens garante que o espaço de fase  $n$ -dimensional pode ser reconstruído através dos chamados atrasos temporais reconstruindo o atrator correspondente, onde as propriedades topológicas e estatísticas do atrator original sejam preservadas. Ou seja:

$$\phi: R \rightarrow R^{2n}$$

$$\phi(x) = (f(x), f(x - \tau), f(x - 2\tau), \dots, f(x - 2n\tau)) \quad (5.7)$$

Onde  $\tau$  é definido como o atraso estimado pela equação descrita abaixo denominada equação de relação mútua. O  $\tau$  ideal é encontrado no mínimo da função  $S: R \rightarrow R$

$$S = \sum_{ij} p_{ij}(\tau) \ln \left( \frac{p_{ij}}{p_i p_j} \right) \quad (5.8)$$

Onde  $p_i$  é a probabilidade de que seja encontrado um valor da série no  $i$ -ésimo intervalo e  $p_{ij}(\tau)$  é a probabilidade relacional entre o  $i$ -ésimo e o  $j$ -ésimo intervalos (DEYE, R. e SUGIAHARA, G, 2011).

### 5.7.1 Dimensão de Correlação

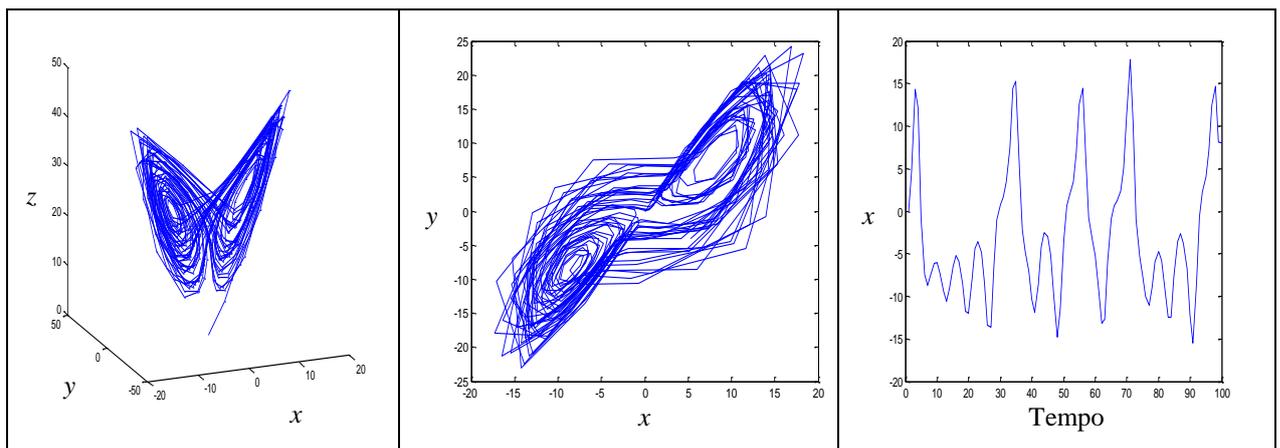
De vital importância, pois caracteriza a complexidade global do sistema medindo a heterogeneidade do atrator (BARONE D., MARON G., RAMOS E, 2012).

$$D = - \lim_{N \rightarrow 0} \left( \lim_{\omega \rightarrow 0} \frac{\log C(\omega)}{\log(\omega)} \right) \quad (5.9)$$

Onde  $C(\omega)$  é a função de correlação com valor calculado sobre todos os pontos do atrator.

Como exemplo, a Figura 5.4 apresenta a reconstrução do atrator de Lorenz a partir de um eixo pelo Teorema de Takens onde os eixos  $x$ ,  $y$  e  $z$  representam o espaço de fase do sistema dinâmico ao longo do tempo.

Figura 5.4- Reconstrução do atrator de Lorenz



Fonte: figura elaborada pelo autor

## 5.8 Posicionamento Analítico como um Sistema Dinâmico Caótico

Considere  $x \in R^n$  e  $P: R^n \rightarrow R^n$  um posicionamento analítico. Por definição existe uma função de custo  $F: R^n \rightarrow R$  não negativa tal que existe  $m > 0$  onde para todo  $n$  natural  $F(P^n(x)) > m$ . Dado  $\epsilon > 0$  existe um  $n_0$  onde para todo  $n > n_0 \Rightarrow F(P^n(x) - m) < \epsilon$ . Neste caso  $F$  será a função de Lyapunov em relação a  $P$ .

Para a demonstração do principal resultado desta sessão é necessário definir uma grandeza responsável por caracterizar o quão caótico é um sistema denominado por expoente de Lyapunov (MAÑE R. 1987).

Seja  $f: R^n \rightarrow R^n$  uma aplicação diferenciável. Definimos por expoente de Lyapunov em relação a um ponto  $x \in R^n$  por:

$$\lambda_{x_0} = \lim_{n \rightarrow \infty} \frac{1}{n} \ln \left\| \frac{df^n(x)}{dx} \right\| \quad (5.10)$$

Define-se também:

$$LEE = \text{Max}\{\lambda_{x_0}, \lambda_{x_0} \in R^n\} \quad (5.11)$$

Como o maior expoente de Lyapunov de  $f$  (BARONE D., MARON G., RAMOS E, 2012).

O Teorema de Oseledec garante que os expoentes de Lyapunov de uma aplicação diferenciável estão bem definidos para quase todo ponto, no sentido estatístico (MAÑÉ R. 1987).

Seguindo (PETRY, A.; BARONE, D. A. 2003), através desta medida é possível quantificar o caos de um sistema dinâmico através das observações:

1. Se  $LEE < 0$  o sistema é uma contração
2. Se  $LEE = 0$  o sistema preserva alguma medida de volume e não é caótico
3. Se  $LEE > 0$  o sistema apresenta caos.

De posse destas informações já existe o embasamento teórico para ser enunciado e demonstrado o resultado principal desta sessão.

**Teorema:** Seja  $P: R^n \rightarrow R^n$  um posicionador analítico.  $P$  é um sistema dinâmico caótico.

**Demonstração:** Seja  $P: R^n \rightarrow R^n$  um posicionador e  $dP(x)$ ,  $x \in R^n$  a sua Jacobiana. Sem perda de generalidade supõe-se que  $P$  é um posicionador quadrático e a imagem de  $P$  seja o conjunto  $I = [0,1]^n$ ,  $n \in N$ . A matriz  $dP(x)$  é positiva, simétrica e definida, logo possui todos os autovalores positivos (LIMA, ELON LAGES, 2014) e pelo Teorema de Perón existe um autovalor  $\lambda$  maior que os demais (SMITH, ROGER. 2006). Daí:

$$d(P)(x)v = \lambda v, v \in R^n \Rightarrow d^n(P)(x)v = \lambda^n v \Rightarrow \ln \|d^n(P)(x)v\| = \ln \|\lambda^n v\|$$

$$\ln \|d^n(P)(x)\| \|v\| = \ln \|\lambda^n\| \|v\| \Rightarrow \frac{1}{n} \ln \|d^n(P)(x)\| = \ln \lambda$$

$$\lim_{n \rightarrow \infty} \frac{1}{n} \ln \|d^n(P)(x)\| = \ln \lambda > 0$$

C.Q. P.

O Teorema revela que há muito a ser estudado sobre o posicionamento ou processos de otimização em geral. Sendo um sistema caótico, sensível às condições iniciais por definição.

É necessário também definir as propriedades da matriz resultante gerada pela diferenciação da função custo em um posicionamento analítico. Logo, a sessão seguinte formalizará alguns conceitos necessários.

## 5.9 Teoria Espectral dos Grafos

Estuda as propriedades de um grafo através de suas matrizes e seus respectivos espectros (o conjunto formado por seus autovalores). O principal objetivo na Teoria Espectral dos Grafos (TEG) é encontrar os parâmetros necessários para identificar se dois grafos são isomorfos. Seja um Grafo  $G(V, E)$ . Define-se por sua matriz de adjacência a estrutura:

$$A(G) = [a_{ij}]$$

onde:

$$\begin{cases} a_{ij} = 1, \text{ se } v_i v_j \in E \\ a_{ij} = 0, \text{ se } v_i v_j \in E^c \end{cases}$$

Define-se por matriz Laplaciana de  $G$  a diferença entre a matriz diagonal com o grau dos vértices e a matriz de adjacência.

$$L = D - A \quad (5.12)$$

Ex1: Considere  $P: R^n \rightarrow R^n$  um posicionamento quadrático. A menos de multiplicação por constantes  $d(P)$  é representada por uma matriz do tipo Laplaciana (MERRIS R. 1994).

Ex2: Considere a função descrita na equação 4.6. A sua Jacobiana será denominada Laplaciana não Linear e esta será utilizada neste trabalho.

Para resolução do sistema definido pela Laplaciana não Linear é utilizada uma rede neural. Ferramenta definida no capítulo seguinte que apresenta bons resultados na resolução de problemas complexos e de natureza caótica.

## 5.10 Sistemas Dinâmicos na Computação

Aparecem de imediato nas definições mais básicas da Ciência da Computação. Quando definimos uma Máquina de Turing estamos modelando um sistema dinâmico onde temos um alfabeto (um conjunto de símbolos), um conjunto de estados, um estado inicial e através de transições chegamos a um estado final, ou seja, um sistema dinâmico por definição, onde existe uma subárea denominada dinâmica simbólica cujo objetivo é compreender a dinâmica destas transições. Por outro lado, é provado que uma Máquina de Turing pode estar relacionada com um autômato celular que nada mais é do que a forma discreta de uma equação diferencial (SIEGELMANN, HAVA T. D. SONTANG, EDUARDO, 1991).

Na década de quarenta o matemático Claude Shannon definiu o conceito de Entropia, medida que quantifica a informação e em termos gerais medindo o quanto o processo modificou o primeiro estado do sistema (SHANNON, CLAUDE E. 1948). Em Computação Teórica são estudados conceitos como Recorrência, Equações Booleanas (KUMAR R.; GARG V.; MARCUS S.I., 1993) que são conceitos dinâmicos. Computação Quântica trata em desenvolver processos baseados nos fenômenos físicos que ocorrem no interior do átomo. Seu estudo é de natureza probabilística. Por outro lado, tais fenômenos são modelados pela Teoria Ergódica, área brevemente descrita anteriormente no texto (LOPES, ARTUR O. 2017).

Uma Rede Neural tem sua formulação matemática tratada como um sistema dinâmico. São máquinas de Turing complexas e tem nas equações diferenciais seus alicerces. E em termos de aplicações temos Processamento de Sinais (HASSIBI A., BOYD S., e J. HOW J. P., 1999), Neurociências, Inteligência Artificial (MARON, G.; BARONE, D. A. C.; RAMOS, E. A. 2013), em resumo, em quase todas as áreas de ponta em se tratando de Ciência da Computação (MAY ROBERT M, 1976).

## 5.11 Resumo do Capítulo

Neste capítulo foram apresentados os sistemas dinâmicos não lineares. São os elementos fundamentais na compreensão da chamada Teoria do Caos. Foram apresentados os atratores, que por sua vez são os conjuntos resultantes no processo de interações de um sistema dinâmico, e os mais importantes: os atratores estranhos. Conjuntos de geometria complexa que são utilizados em aplicações em quase todas as ciências.

Foi enunciado e provado um resultado que relaciona um posicionamento analítico diretamente com um sistema dinâmico, justificando as ferramentas introduzidas neste texto.

Foi apresentada uma rápida visão em Teoria Espectral dos Grafos, onde as matrizes que irão surgir nos processos de otimização durante o texto possuem as características apresentadas.

E por último foram mostradas aplicações da Teoria dos Sistemas Dinâmicos na Ciência da Computação como um todo.

O próximo capítulo tratará das redes neurais. Serão exibidas as mais relevantes e será visto a interpretação de uma rede como um sistema dinâmico. Haverá uma ênfase nas redes recorrentes, pois estas serão utilizadas no processo de otimização envolvidos neste trabalho.

## VI Redes Neurais

São sistemas computacionais inspirados no sistema nervoso central de um animal, conforme a Figura 6.1. Foram apresentadas pela primeira vez no ano de 1943 nos estudos de McCulloch e Pitts até que em 1949 Heeb publicou o artigo: *'The Organization of Behavior'*, onde nesta obra ele propôs uma lei de organização para os neurônios. Redes Neurais são capazes de realizar aprendizagem de máquinas.

Intuitivamente é possível analisar uma rede neural como um conjunto finito de neurônios interconectados que processam dados de entrada. Possuem diversas aplicações em computação como processamento de sinais, reconhecimento (imagens, fala, etc.) processamento de voz, recuperação de informações, previsão de séries temporais, aproximação de funções e nos últimos anos, principalmente são usadas para resolver problemas de otimização convexa (HAYKIN, SIMON, 2001).

Um problema em utilizar redes neurais é encontrar a topologia ideal da rede em cada problema, já que não existe uma regra ou algoritmo que o faça. Atualmente os algoritmos genéticos são opções para essa abordagem.

Em relação ao processamento dos dados de entrada as redes neurais podem ser divididas em duas classes:

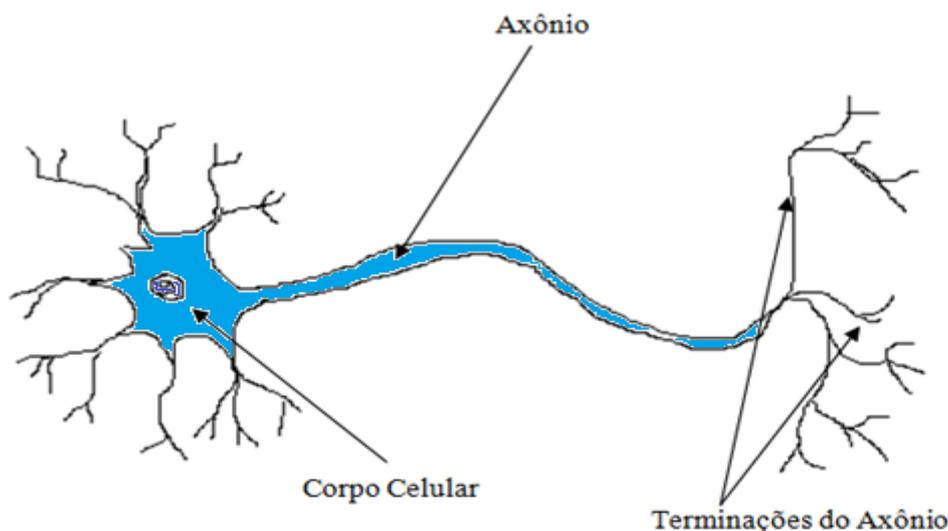
- Analógicas
 

Os dados de entrada são valores reais e na maioria das vezes limitados. São ideais para resolução de problemas matemáticos de um modo geral.
- Discretas
 

Processam dados de entrada discretos, na maioria das vezes são booleanos. Como exemplo tem-se a máquina de Boltzmann.

As redes neurais não possuem uma única topologia. Seus modelos variam em número de componentes, conexões e processamento. Seu modelo mais simples é o perceptron, definido adiante.

Figura 6.1- O neurônio humano



Fonte: Figura elaborada pelo autor

As Redes Neurais apresentam três tipos de aprendizado:

- Supervisionado: é apresentado um conjunto para o treino, onde as saídas irão convergir para este conjunto.
- Não Supervisionado: A rede atualiza seus pesos sem um conjunto treino ou qualquer reforço adicionado.
- Por Reforço: Para cada entrada é adicionado um reforço para adequar as saídas da rede.

## 6.1 Algumas Redes Neurais

### 6.1.1 O Perceptron

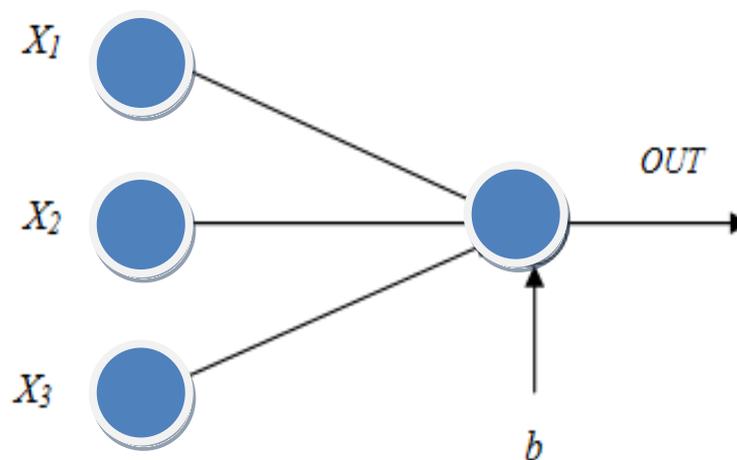
Um tipo simples de rede neural, ilustrado na Figura 6.2. É um algoritmo supervisionado para classificação. Trata-se de um classificador binário mapeando sua entrada cuja saída é calculada utilizando o Produto Escalar (matematicamente denominado de Produto Interno) entre o peso e sua entrada, sendo que ambos são vetores  $n$ -dimensionais e este resultado é avaliado por uma função de saída como segue pela equação abaixo (HAYKIN, SIMON, 2001):

$$v = \sum_{i=1}^m w_i x_i + b \quad (6.1)$$

$$y = \varphi(v + b) \quad (6.2)$$

Onde  $W = \{w_i, i = 1, 2, 3, \dots, m\}$  é o vetor de pesos,  $X = \{x_i, i = 1, 2, 3, \dots, m\}$  o vetor de números reais, e  $b$  uma constante.

Figura 6.2- O Perceptron



Fonte: Figura elaborada pelo autor

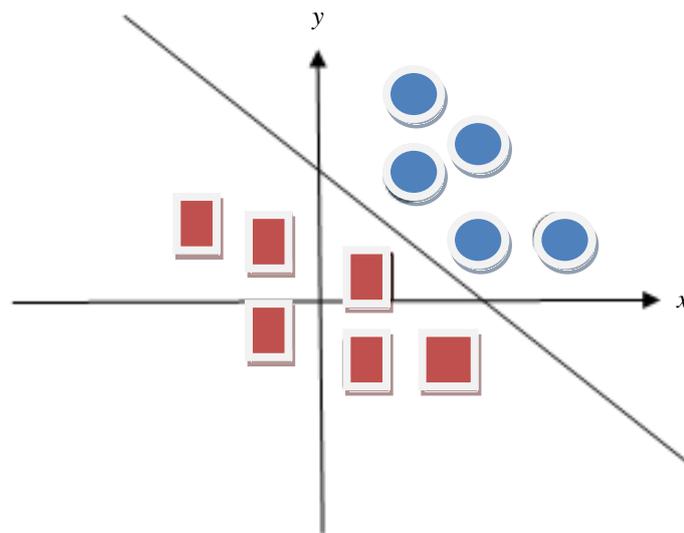
O perceptron possui um item identificado como polarização externa, conhecido por bias e denotado por  $b$  como mostra a figura 6.2. Este item tem o objetivo de aumentar ou

diminuir o argumento da função de ativação. A depender do valor de  $b$  o gráfico resultante pode não mais passar pela origem dos eixos.

Intuitivamente esta variável aumenta o grau de liberdade da função afetando diretamente o grau de aproximação da rede. O bias pode ser ajustado durante as interações como os neurônios. Também podemos ter valores de saída não nulos, mesmo que suas entradas sejam todas nulas. O bias é fundamental em problemas envolvendo otimização, pois na maioria deles os valores iniciais dos neurônios são nulos ou quase nulos (HAYKIN, SIMON, 2001).

Diz-se que dois conjuntos  $A, B \subset R^2$  são linearmente separáveis se existe uma reta  $r$  tal que  $r$  divide o plano  $R^2$  em dois hemisférios onde  $A$  pertence exclusivamente a um e  $B$  exclusivamente ao outro, como é observado na Figura 6.3, neste caso em um plano bidimensional.

Figura 6.3- Conjuntos linearmente separáveis



Fonte: Figura elaborada pelo autor

O perceptron consegue resolver apenas problemas de natureza linearmente separável. Uma função XOR, por exemplo, não pode ser processada. Quando o uso desta ferramenta não é possível é aconselhável utilizar o próximo modelo: A Rede de Múltiplas Camadas, uma rede de natureza não linear apta para resolução de problemas de alta complexidade.

### 6.1.2 Redes de Múltiplas Camadas

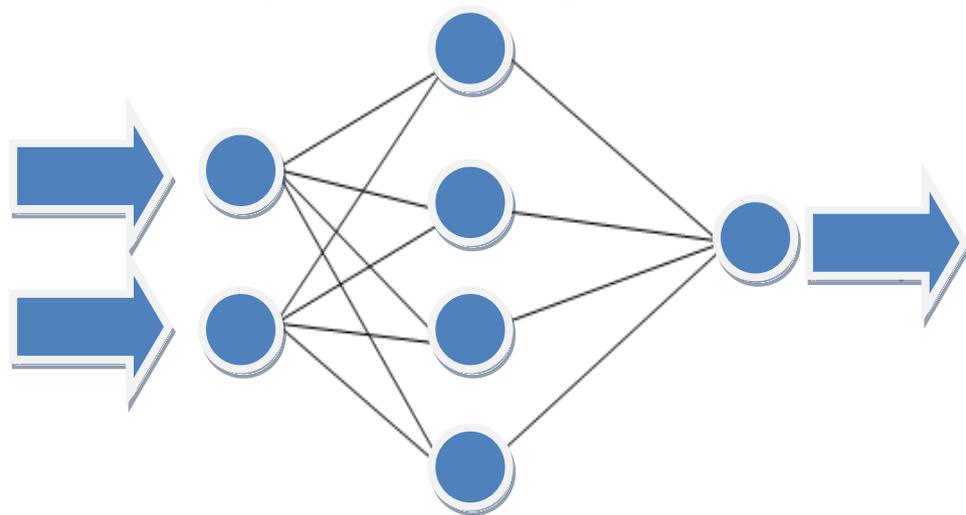
Possui uma arquitetura de perceptron de múltiplas camadas, conforme a Figura 6.4, e são as mais conhecidas e utilizadas. São eficientes na resolução de problemas envolvendo alto grau de não linearidade. Para solucionar problemas que não são linearmente separáveis, as redes de múltiplas camadas apresentam grande êxito em termos de resolução utilizando o algoritmo denominado *Back-propagation*. Intuitivamente é um algoritmo que procura uma função que mapeia um conjunto de entrada para uma saída correta. É um algoritmo de classificação, por exemplo, a entrada é uma bandeira e a saída é o nome do país correspondente. Suas aplicações são das mais variadas como: Classificação e Mineração de

dados, detecção de anomalias, regressão, aprendizagem de recursos, etc. (WERBOS PAUL J. 1994).

Trata-se de um algoritmo supervisionado e está baseado no que entendemos por retro propagação de erro. Em resumo, o algoritmo consiste no seguinte:

- A fase da propagação, onde o sinal se propaga pela rede, camada por camada até chegar a sua saída.
- Na fase de retro propagação fazemos o caminho de volta e os pesos são reajustados de acordo com uma regra pré-definida onde a resposta da rede é subtraída pela resposta real gerando o erro onde o erro é propagado pela rede.

Figura 6.4- Rede de múltiplas camadas



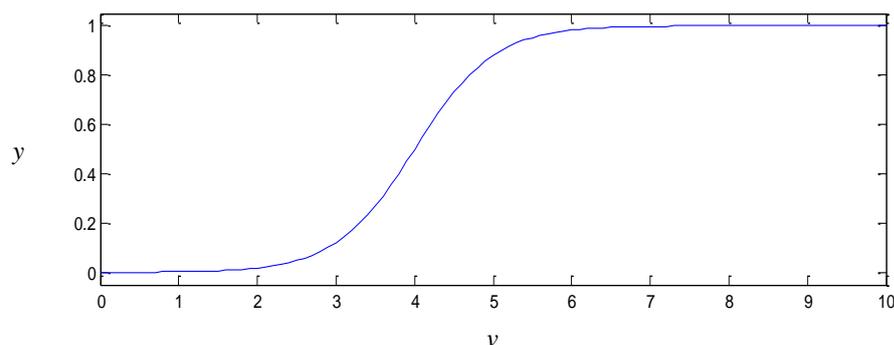
Fonte: figura elaborada pelo autor

Dão características dos perceptrons na MLP:

- O modelo de cada neurônio inclui uma função de ativação não linear, geralmente a sigmóide, descrita pela equação 6.2 e cujo gráfico está representado na Figura 6.5:

$$y_j = \frac{1}{1 + \exp(-v_j)} \quad (6.2)$$

Figura 6.5- A função sigmóide



Fonte: Figura elaborada pelo autor

- A rede possui camadas ocultas, onde cada uma delas consegue fazer com que a rede aprenda tarefas mais complexas facilitando o reconhecimento dos padrões.
- Possui alto grau de conectividade.

Formalmente uma MLP pode ser considerada uma “Super Máquina de Turing”. Uma MLP é um aproximador universal de funções, lineares ou não. Em (CYBENKO, G, 1989) é provado que a arquitetura de uma MLP equivale a uma máquina de Turing através de aproximação por funções contínuas.

### 6.1.3 Redes de Hopfield

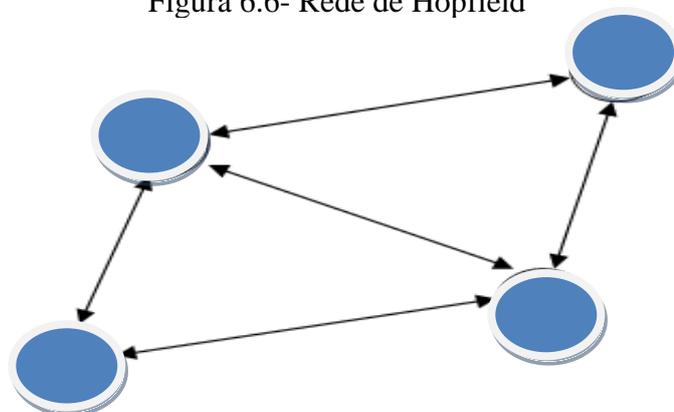
Uma Rede Neural recorrente é uma rede com realimentação envolvendo uma camada com pesos simétricos. Podem ser interpretadas como sistemas dinâmicos e possuem uma função de Lyapunov garantindo sua estabilidade. Correspondem a uma área de estudo denominada Neurodinâmica. A realimentação é uma maneira de incluir a variável “tempo” em uma Rede Neural gerando novo comportamento que será aplicado em suas operações. Formalmente uma rede recorrente é um conjunto de equações diferenciais não lineares de comportamento determinístico onde definem um modelo em função do tempo. Sendo um Sistema Dinâmico não linear o comportamento assintótico de uma rede recorrente tende a um atrator (HOPFIELD, J. J. 1982).

A Rede de Hopfield é uma rede neural de retroalimentação com uma única camada sendo um belo exemplo de um sistema dinâmico aonde após um número considerável de interações tende a chegar a um estado estacionário (estabilidade de Lyapunov), ilustrada na Figura 6.6. Cada neurônio é ao mesmo tempo entrada e saída. Suas aplicações são diversas e em várias áreas da computação como: Recuperação de Padrões, Reconhecimento de Padrões, Mineração de Dados, Otimização, Criptografia, problemas de natureza combinatória, etc.

Intuitivamente a Rede de Hopfield funciona como a memória humana. Um ser humano relembra de um fato completo através de uma pequena lembrança do acontecido. Este processo se chama memória endereçada pelo conteúdo. Na rede, esse processo é denominado memória associativa.

O Modelo foi proposto pelo matemático John Hopfield em 1982 para resolver problemas de otimização. Como primeiro exemplo foi resolvido o TSP, um problema de otimização combinatória NP - completo.

Figura 6.6- Rede de Hopfield



Fonte: Figura elaborada pelo autor

Em primeiro lugar é preciso formular a equação geral da Rede de Hopfield seguindo [referência]. Para isso consideremos seus pesos sinápticos  $w_{i1}, w_{i2}, \dots, w_{in}$ , que representam

as condutâncias onde  $n$  é o número de neurônios. Consideremos também  $u_1(t), u_2(t), \dots, u_n(t)$  as entradas (tensões). Seguindo a lei de correntes de Kirchorff temos:

$$\frac{dv_i(t)}{dt} = -\frac{v_i(t)}{C_i R_i} + \sum_{j=1}^n w_{ij} g_i(v_i(t)) + \frac{I_i}{C_i} \quad (6.3)$$

Onde  $R_i$  é a Resistência da fuga,  $C_i$  a Capacitância da fuga e  $v_i$  é o campo induzido na entrada da Função de ativação do neurônio  $g_i$ .

A equação abaixo define a Função Energia (Função de Lyapunov para a rede de Hopfield) representando a função a ser minimizada enquanto ocorrem as interações da Rede Neural. O Processo acaba quando a Função Energia chega a um estado estacionário caracterizando o final do processo.

$$E = -\frac{1}{2} \sum_i \sum_j w_{ji} + \sum_{j=1}^k \frac{1}{R_j} \int_0^{g_j} \varphi_j^{-1}(x) dx - \sum_{j=1}^k I_j g_j \quad (6.4)$$

Exemplo: Considere a Rede Neural descrita pela equação (6). Este exemplo foi utilizado em um experimento na área de neurociências e está descrito no trabalho (RAMOS, E., BANDEIRA, V., REIS, R., BONTORIN, G, 2017). Alguns cálculos matriciais e mudança de variáveis mostram que as equações 6.3 e 6.5 são equivalentes.

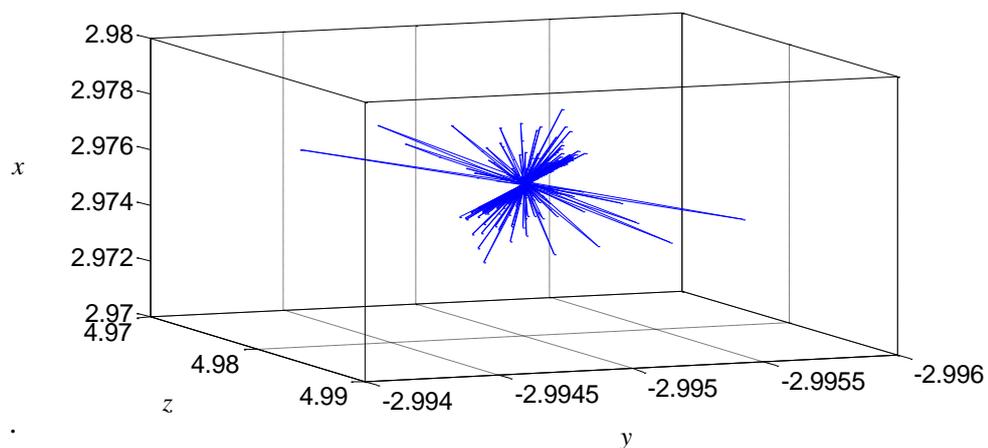
$$\phi'(t) = -C\phi(t) + Af(\phi(t)) + Bf(\phi(t - \tau)) + I \quad (6.5)$$

$$\text{Onde } C = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad A = \begin{bmatrix} -2 & 0 & 6 \\ -4 & 1 & -1 \\ -6 & -4 & -1 \end{bmatrix},$$

$$f(x) = \tanh(x), \quad B = \begin{bmatrix} 1 & 0 & 1 \\ 0 & -3 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad I = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

A solução do Sistema descrito na equação 6.5 possui como solução um atrator estranho denominado “Asteriscus” ilustrado na figura 6.7, onde os eixos  $x$ ,  $y$  e  $z$  representam o espaço de fase do atrator.

Figura 6.7- Atrator Asteriscus



Fonte: (RAMOS, E., BANDEIRA, V., REIS, R., BONTORIN, G, 2017)

Voltando à Conjectura de Palis enunciada no capítulo anterior. Pode-se argumentar: Caso a conjectura esteja correta é possível descrever a maioria dos processos computacionais onde cada processo é a união de Redes de Hopfield. Caso seja constatada a falsidade da conjectura, esta possui resultados verdadeiros em algumas situações (casos particulares). O desafio assim consiste em identificar quais dos processos computacionais podem ser modelados por estes casos particulares (BONATTI C, DIAZ L.; VIANA M, 2005).

## **6.2 Resumo do Capítulo**

Neste capítulo foram definidas as redes neurais, com suas formalizações matemáticas e propriedades dinâmicas. Foi apresentada a relação de equivalência entre redes neurais e sistemas dinâmicos dando uma visão geral da área. Foi dado um foco principal nas redes recorrentes, em particular na rede de Hopfield, pois o estado final de uma rede de Hopfield nada mais é que um atrator estranho.

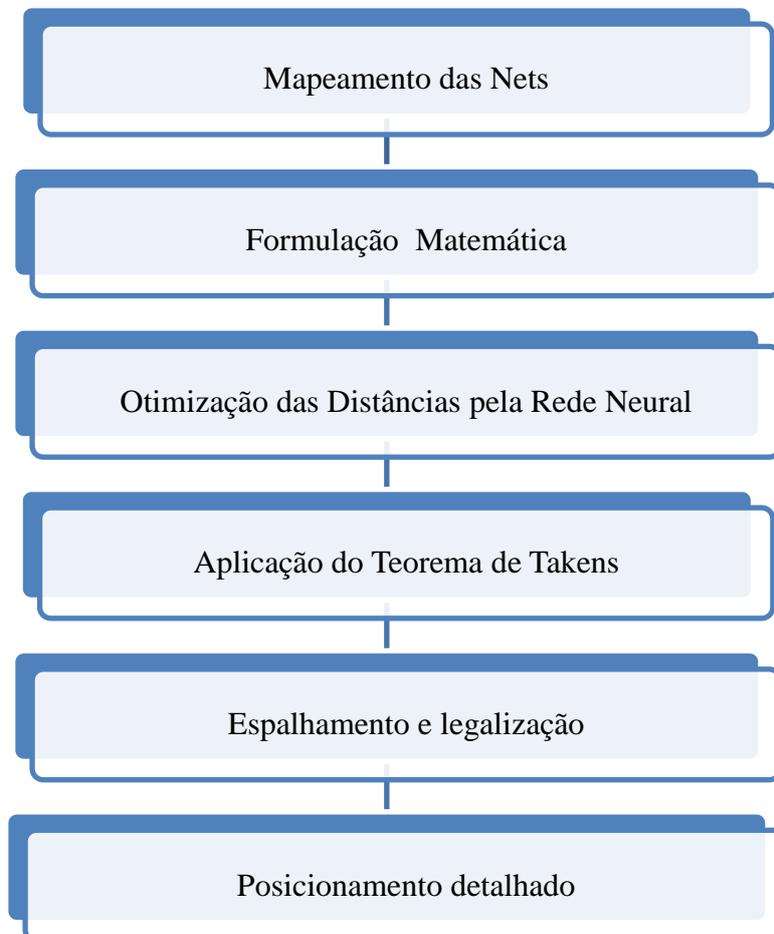
Neste ponto já temos o embasamento teórico necessário para o desenvolvimento do tema principal deste trabalho. No próximo capítulo O Chaotic Place será apresentado. Serão descritos o posicionamento global, espalhamento, legalização e posicionamento detalhado. Serão apresentados os resultados mais relevantes e as comparações com o estado da arte.

## VII. CHAOTIC PLACE

Neste trabalho é proposto um posicionamento global analítico para FPGAs homogêneos direcionado à redução do comprimento dos fios utilizando como função custo a LOG-SUM-EXP simplificada para minimizar as distâncias das células, onde o processo de minimização é realizado utilizando uma Rede Neural de Hopfield. Considerando cada célula do FPGA correspondendo a um ponto no plano cartesiano é tomado somente o eixo das abscissas. Após a minimização das distâncias dos pontos neste eixo é aplicado o Teorema de Takens para obter o levantamento dos pontos no Plano  $R^2$ , obtendo um posicionamento global de todas as células diminuindo o custo computacional e obtendo um resultado compatível como estado da arte para FPGAs homogêneos.

Nossa metodologia, resumida na figura 7.1, consiste a princípio realizar um mapeamento em cada FPGA da lista de *benchmarks*, tomando os CLBs, I/O e nets com todas as interconexões. Logo após é formulada a função custo definida através das interconexões dos CLBs de modo a minimizá-las. Daí a Rede Neural irá encontrar as coordenadas unidimensionais dos CLBs, para aplicarmos o Teorema de Takens de modo a encontrarmos a versão bidimensional final. O processo é finalizado com o espalhamento e legalização dos pontos. Onde cada item será detalhado a seguir.

Figura 7.1- Fluxograma do Posicionamento



Fonte: Figura elaborada pelo autor

Em nosso experimento foram utilizados benchmarks disponibilizados pelo Centro de Microeletrônica da Carolina do Norte (MCNC) (BENCHMARKS. 2017). Em 1997 Vaughn Betz elaborou um desafio chamado “FPGA *Place and Route Challenge*” onde pagaria US\$ 1,00 para quem conseguisse para cada trilha a menos que o VPR posicionando e roteando os circuitos. Essa coleção atualizada de *benchmarks* é frequentemente utilizada na literatura e servindo como testes e parâmetros para o estado da arte. Os FPGAs são de tamanhos variados, compostos de Blocos Lógicos (CLBs) e I/O Blocks (BENCHMARKS. 2017).

O Chaotic Place foi desenvolvido seguindo as etapas:

- Para o Mapeamento das nets, a Rede Neural, espalhamento e legalização com a linguagem de programação JAVA utilizando o Software Netbeans versão 8.0.
- Para a construção do atrator utilizamos o MATLAB versão 2014.

Foi utilizada uma estação de trabalho com as seguintes configurações:

- Sistema Operacional Windows 7 64 bits Enterprise SP1.
- Memória RAM 4 GB
- Processador CORE 2 DUO E 7500 @2,93 GHZ e 2,13 GHZ.

## 7.1 Mapeamento das Nets

Cada Bloco Lógico possui quatro LUTs onde cada uma delas pode estar conectada a uma net ou a um I/O. O objetivo será minimizar o somatório das distâncias entre estes todos os Blocos Lógicos. Como métrica é utilizada o HPWL definido no capítulo 2. Além disso, para medida dos comprimentos dos fios é utilizada a unidade tradicionalmente adotada: o tamanho de cada bloco lógico (BENCHMARKS. 2017).

### 7.1.1 Estrutura das Nets

Os FPGAs dos benchmarks possuem basicamente três elementos: Blocos lógicos, *pads* de entrada e *pads* de saída.

Declarações:

- Pads de entrada: A nomenclatura abaixo avisa que temos um input denominado “my\_pad” ligado a uma rede denominada “my\_net”  

```
.inputmy_pad
.pinlist: my_net
```
- Pads de saída: Pelo mesmo principio do item 1, aqui temos um output denominado “my\_opad” ligado a uma rede denominada “some\_net”  

```
.outputmy_opad
.pinlist: some_net
```
- Blocos Lógicos: Abaixo está declarado um bloco lógico denominado “logic\_block”. A linha “pinlist” mostra quais as redes conectadas ao bloco. Lembrando que cada bloco lógico contém uma LUT de quatro entradas e um flip-flop. A linha que se inicia por “subblock” é irrelevante para o nosso posicionamento.  

```
.clbmy_logic_block
```

```
pinlist: in_ain_bin_cin_dout_netclk
subblock: sb_one 0 1 2 3 4 5
```

## 7.2 Formulação Matemática

Para cada Bloco Lógico  $\{x_i\}$  e suas nets conectadas é preciso minimizar a função custo LSE simplificada unidimensional definida abaixo:

$$\Phi(X) = \alpha \left( \log \left( \sum_{i=1}^n \frac{e^{x_i}}{\alpha} \right) + \log \left( \sum_{i=1}^n \frac{e^{-x_i}}{\alpha} \right) \right) \quad (7.1)$$

A Jacobiana de  $\Phi$  é definida pela matriz abaixo:

$$J\Phi = \begin{pmatrix} \frac{\partial \Phi_1}{\partial x_1} & \dots & \frac{\partial \Phi_1}{\partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial \Phi_n}{\partial x_1} & \dots & \frac{\partial \Phi_n}{\partial x_n} \end{pmatrix} \quad (7.2)$$

Um cálculo trivial mostra que a Jacobiana é nossa Matriz Laplaciana Não Linear definida no capítulo V.

A obtenção do mínimo de  $\Phi$  é encontrada após a resolução do Sistema Homogêneo definido abaixo:

$$J\Phi(x) = 0 \quad (7.3)$$

Para minimizar a função custo na resolução do sistema é utilizado o Método do Gradiente Descendente: consiste em um método iterativo para encontrar mínimos ou máximos locais em um processo de otimização. Se a função em questão for estritamente convexa esse máximo ou mínimo é único como é o caso da LSE.

Em cada interação é tomada a direção negativa do gradiente, em outras palavras, é selecionado o declive máximo gerado naquela medida de tempo, como está descrito na Figura 7.2.

Intuitivamente pode-se imaginar um esquiador deslizando sobre uma montanha. Se o esquiador desejar chegar à base da montanha mais rápido (sem se preocupar com possíveis acidentes) ele irá por onde a montanha for mais íngreme. Este método é bastante utilizado em problemas em regressão linear.

O método é definido pelas equações abaixo:

$$x^{n+1} = x^n - \gamma^n \text{Grad}(F(x^n)) \quad (7.4)$$

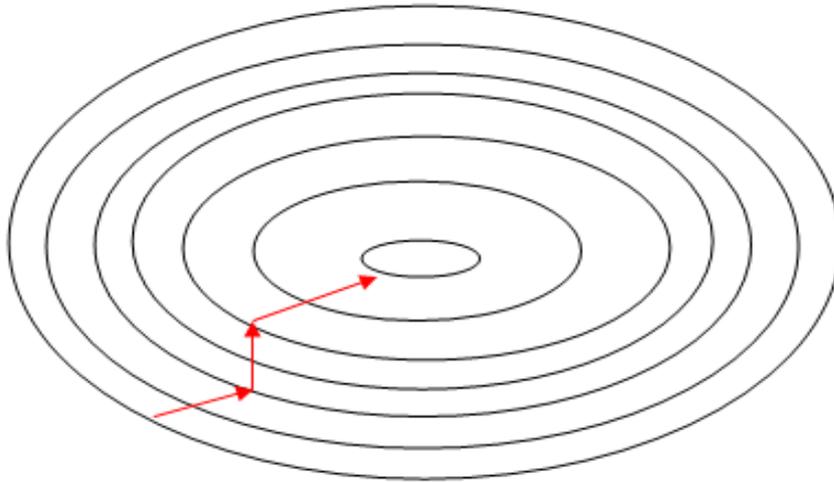
$$\text{Onde: } \text{Grad}(F(x^n)) = J\Phi(x)^T G(x^n)$$

$$E: F(x) = \frac{1}{2} G^T(x)G(x) \quad (7.5)$$

Onde  $x^n$  é a solução no tempo  $n$ ,  $\gamma$  a taxa de convergência também conhecida como taxa de aprendizado,  $J\Phi$  a Jacobiana da função objetivo e  $F$  é uma função monótona não negativa e decrescente.

No experimento foi utilizado  $\gamma = 0,01$ . Valor médio encontrado através dos testes nas nets dos FPGAs.

Figura 7.2. Descrição do Método do Gradiente Descendente



Fonte: Figura elaborada pelo autor

Resta então aplicar o método para a função custo. Para isso utilizaremos uma Rede Neural de Hopfield definida no capítulo anterior.

Neste trabalho, através de algumas manipulações algébricas as equações gerais da Rede de Hopfield são reescritas seguindo a natureza do problema:

$$u_i'(t) = \eta u_i(t) + \sum_{j=1}^N T_{ij}(t) v_j(t) + i_i^b \quad (7.6)$$

$$\text{Onde: } v_i(t) = g(u_i(t))$$

$$E(t) = -\frac{1}{2} v(t)^T T v(t) - v(t)^T i^b \quad (7.7)$$

Nas equações 7.6 e 7.7  $u_i(t)$  representa o estado do neurônio no tempo  $t$ ,  $i$  representa o vetor com as condições iniciais,  $\eta$  a taxa,  $g$  a função sinal da rede,  $T$  a matriz de pesos e  $E$  a função Energia monótona e decrescente a ser minimizada durante as interações da rede.

O experimento utilizou a taxa com valor  $\eta = 0,1$ . Por se tratar de um sistema caótico, e além de termos redes de diferentes tamanhos qualquer variação na taxa poderia resultar na não convergência da rede neural ou um *runtime* muito elevado. O vetor que representa a solução inicial do sistema introduzida na rede deve possuir valores próximos ao zero. Um valor elevado (próximo de 1) resulta na não convergência do método pelo mesmo motivo da taxa  $\eta$ .

A Tabela 7.1 exibe cada FPGA, o número de blocos lógicos e o número de épocas (ciclos) utilizadas para a resolução do sistema encontrando o vetor solução final. Solução que minimiza a função custo.

Tabela 7.1 Número de épocas em cada FPGA

<b>FPGA</b>	<b>CLBs</b>	<b>Épocas</b>
apex4	1262	122
alu4	1522	146
apex2	1878	150
spla	3690	206
pdc	4575	321
seq	1750	218
misex3	1397	124
ex5p	1064	129
ex1010	4598	295
diffeq	1497	106
tseng	1407	111
des	1591	187

### 7.3 Aplicações do Teorema de Takens

Na etapa anterior foi obtido o conjunto de células em seus posicionamentos iniciais em um eixo. Neste passo, seguindo o teorema de Takens pode-se realizar o levantamento dos pontos seguindo a regra dos atrasos temporais. O teorema garante que existe um levantamento  $n$ -dimensional para todos os pontos e este gerou até 12 dimensões bem definidas.

$$F: R \rightarrow R^{12}$$

$$F(x) = (x, x_1, x_2, x_3, \dots, x_{11}) \quad (7.8)$$

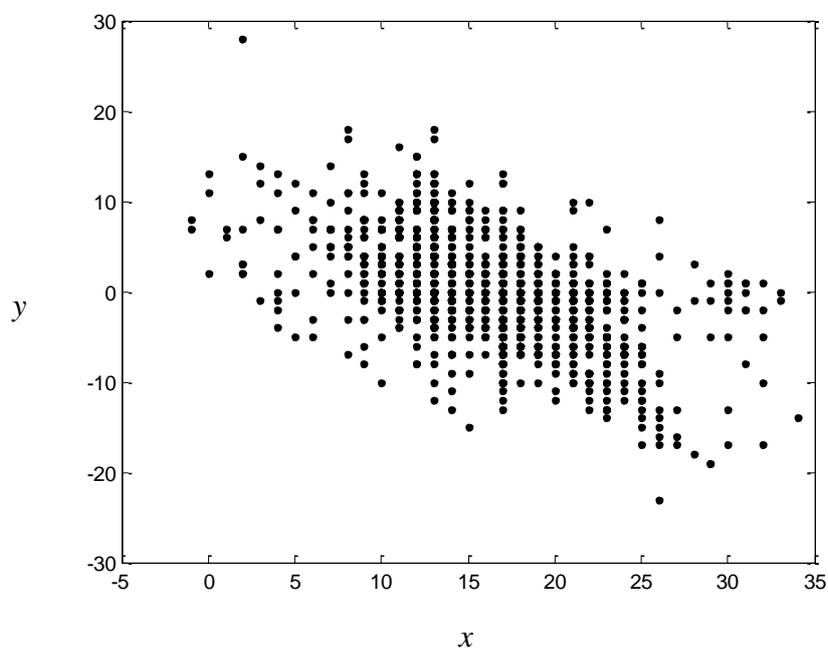
O experimento mostrou que dependendo do FPGA o melhor conjunto de pontos (aquele que gerará melhor resultado final na etapa seguinte) não se dará necessariamente ao utilizar os dois primeiros eixos  $x$  e  $x_1$  variando em alguns FPGAs, mais precisamente nos maiores. Tal fato se dá pelas diferentes dimensões de correlação definidas no capítulo V. Mesmo assim, para um resultado uniforme para vias de comparação foram utilizados somente os dois primeiros eixos em todos os FPGAS.

As coordenadas bidimensionais iniciais do atrator resultante estão limitadas pelo conjunto:

$$I = [0,1] \times [-1,1] \quad (7.9)$$

Para obtenção das coordenadas inteiras, compatíveis com o tamanho do FPGA é aplicada nos dois eixos uma Homotetia, onde cada ponto é multiplicado pela raiz quadrada do número total de células do FPGA. E a seguir no eixo das ordenadas é aplicada uma translação de modo que se tenham somente valores não nulos. O resultado do posicionamento inicial pode ser visto na figura 7.3, onde o plano X x Y representa as dimensões do futuro FPGA.

Figura 7.3- Atrator resultante (FPGA apex2)



Fonte: Figura elaborada pelo autor

Após o posicionamento global ilustrado na Figura 2 podemos aplicar rigorosamente nesta ordem: espalhamento, legalização e o posicionamento detalhado obtendo o posicionamento final.

#### 7.4 Espalhamento

O objetivo desta etapa é distribuir as células na grade do FPGA. As dimensões do FPGA são identificadas como as dimensões do quadrado mínimo necessário para contê-las (Pelas regras estipuladas no *Placement and Route* do MCNC). Tais valores são encontrados calculando a raiz quadrada do total de células. As células pré-definidas são inseridas na grade de modo que resta realizar o tratamento de distintas células com as mesmas coordenadas (sobrepostas).

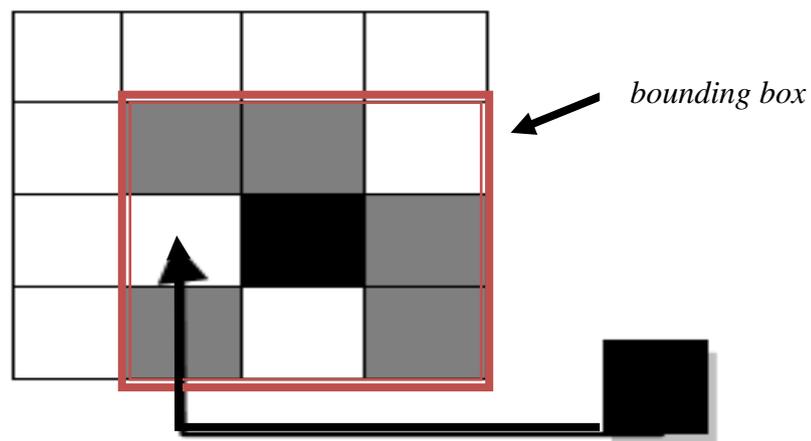
#### 7.5 Legalização

Nesta fase existem alguns problemas que precisam ser trabalhados cuidadosamente: sobreposição, células fora dos limites do FPGA e o posicionamento dos I/Os.

Para a sobreposição são criados *bounding boxes*, em destaque na figura 7.4, ao redor do ponto sobreposto (na parte central do *bounding box*) e por recorrência é procurado dentro da *bounding box* um local vazio e este local é destinado ao ponto sobreposto. O método é aplicado em todos os pontos sobrepostos.

Na figura 7.4 os locais em branco representam os espaços vazios, enquanto os espaços em cinza representam células já inseridas no *bounding box*.

Figura 7.4-Representação de um *bounding box*



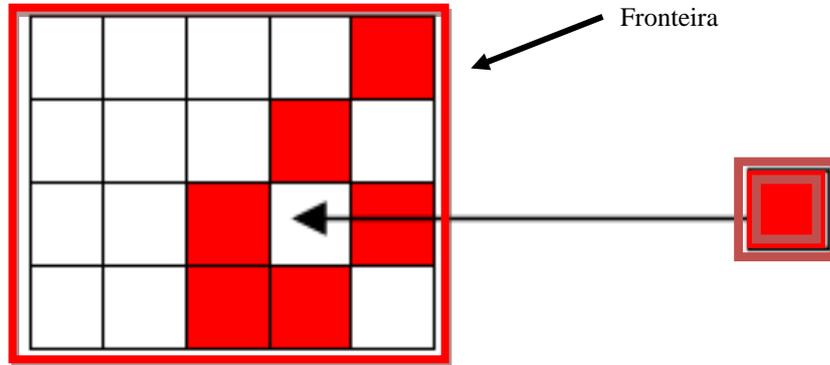
Fonte: figura elaborada pelo autor

Para as células que sobraram não sendo inseridas na grade é feita uma varredura nos locais vazios e estas são alocadas. Como o número destas células é pequeno em relação ao total são colocadas randomicamente. Posteriormente estas células serão realocadas através do posicionamento detalhado.

Se nesta fase ainda são encontradas células fora da região do FPGA, randomicamente as colocamos no interior do FPGA, nas regiões próximas da fronteira não muito distantes de seus locais de origem.

Na figura 7.5 as células em branco representam os locais vazios, as em vermelho os espaços ocupados.

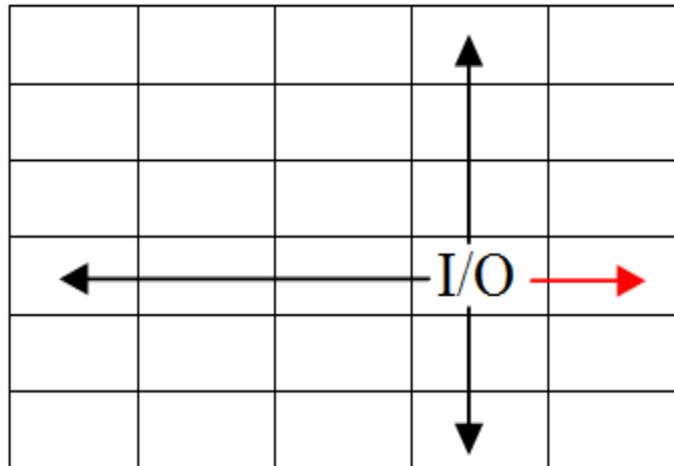
Figura 7.5- Representação de uma célula fora das dimensões do FPGA sendo inserida



Fonte: Figura elaborada pelo autor

Além disso, como é apresentado na figura 7.6, alguns I/Os são encontrados no interior do FPGA. Logo, para todos os I/Os é procurado o ponto mais próximo destes na extremidade e este é alocado. Se este ponto estiver ocupado por um CLB faz-se a permutação. Se o ponto estiver ocupado por outro I/O é procurado o próximo espaço vazio no mesmo eixo da extremidade mais próximo e este é alocado.

Figura 7.6- Representação do posicionamento dos IO/s



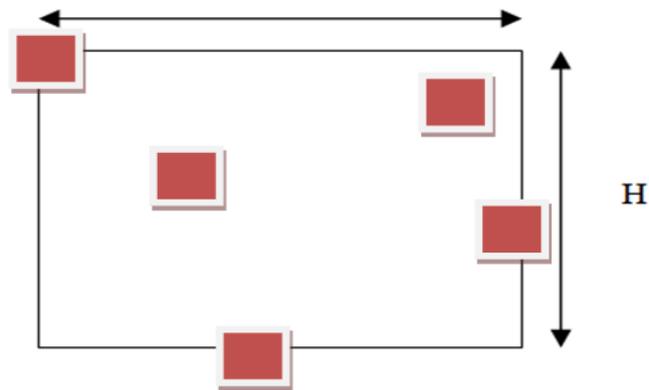
Fonte: figura elaborada pelo autor

## 7.6 Posicionamento Detalhado

O número de portas lógicas (NPL) tem sido tradicionalmente uma forma de comparar o tamanho dos FPGAs. Portanto a unidade de medida dos comprimentos será neste trabalho o

tamanho de cada porta lógica, denominada por PL. Para medir o tamanho de cada *net* é utilizado o semiperímetro descrito na figura 7.7.

Figura 7.7 – Cálculo do tamanho de cada *net*



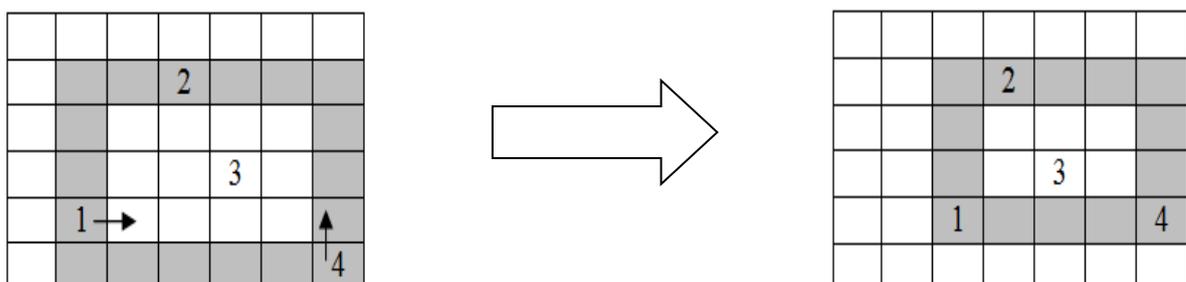
$$\text{COMPRIMENTO} = W + H$$

Fonte: Figura elaborada pelo autor

Após a etapa anterior o posicionamento estaria finalizado, por outro lado o valor do HPWL na maioria dos casos ainda não é satisfatório, pois é ainda elevado se comparado ao valor apresentado pelos demais posicionadores. Para diminuí-lo sem que seja gerado algum tipo de violação o procedimento segue: Na última fase do posicionamento são mapeadas as coordenadas de cada *net* correspondente aos seus extremos. E por recorrência, em cada passo é realizada uma translação à direita da menor coordenada  $x$  e uma translação a esquerda da maior coordenada  $x$  (permutando com a célula original) conforme a figura 7.8.

De modo análogo faz-se o mesmo para as coordenadas do eixo  $y$ . Repetindo este processo para cada *net* cujo HPWL seja maior que a média aritmética dos HPWLs entre todas as *nets*. A escolha da média do comprimento de cada *net* foi obtida através de experimentos nos *benchmarks* dentre outras medidas. Esta apresentou o melhor resultado.

Figura 7.8- Representação da diminuição do tamanho de cada *net*



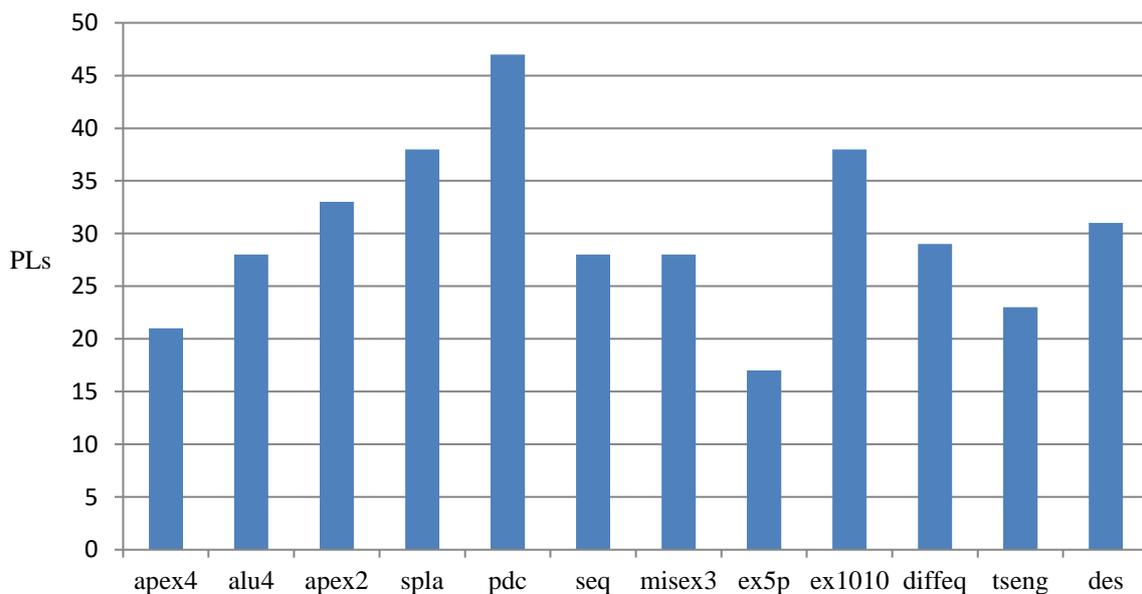
Fonte: Figura elaborada pelo autor

O experimento mostra que o HPWL diminui em cada interação estacionando próximo ao mínimo encontrado pelos outros posicionadores (estado da arte). Por outro lado, deve-se diminuir o HPWL ao seu menor valor possível, pois no estudo dos caminhos críticos algumas células poderão ter suas posições modificadas, aumentando o HPWL. A meta a ser atingida é que tais modificações não gerem discrepâncias em relação ao HPWL encontrado nesta fase.

Posteriormente, em trabalhos futuros será utilizado novamente o último procedimento para diminuir o tamanho dos caminhos críticos, tendo em vista que, a priori, basta que seja diminuindo o tamanho da net que contem um caminho dessa natureza.

O procedimento age diretamente sobre as *nets*. Por isso seus tamanhos são diminuídos consideravelmente. A figura 7.9 representa o tamanho médio final nas nets em cada FPGA, onde o eixo das ordenadas representa o número de PLs (portas lógicas).

Figura 7.9- Tamanhos médios das *nets* em cada FPGA



Fonte: Figura elaborada pelo autor

O experimento apresentou uma diminuição média de 31,75% no HPWL entre os posicionamentos antes de depois do posicionamento detalhado. Com exceção do FPGA “tseng”, as maiores diferenças ocorreram nos FPGAs de maiores tamanhos (“ex1010”, “spla” e “pdc”). Devido ao fato de termos um maior número de células que foram inseridas na grade de forma aleatória na etapa anterior. A exceção “tseng” aconteceu por ter um número elevado de I/Os. Por outro lado, o FPGA “ex5p” obteve apenas 1,2% de diferença. Isto é devido ao pequeno número de CLBs e da natureza topológica das *nets* neste FPGA.

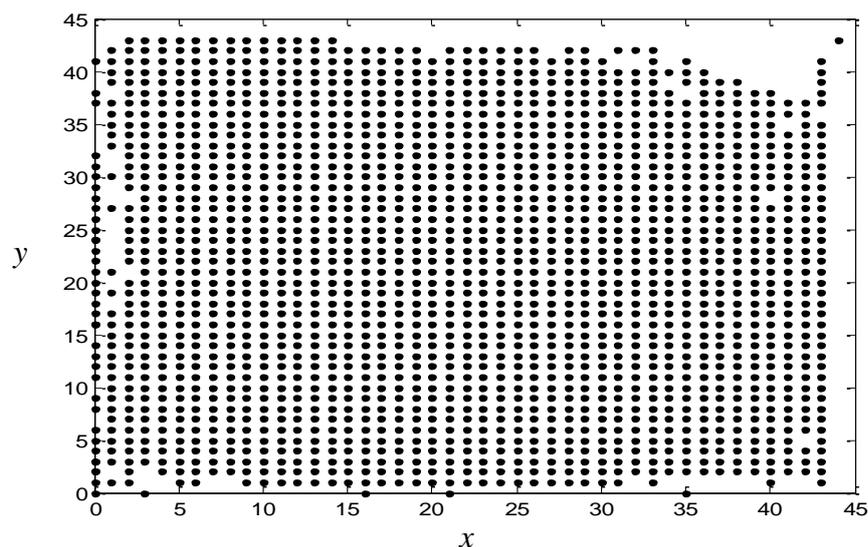
Tabela 7.2 Diferenças entre o posicionamento inicial e final

FPGA	HPWL Inicial	HPWL Final	Diferença (%)
apex4	19647	14115	28,2
alu4	28399	22239	21,7
apex2	44793	30595	31,7
spla	115688	70572	39
pdc	168255	99544	39,6
seq	34602	25076	27,5
misex3	27800	20428	26,5
ex5p	14089	13925	1,2
ex1010	164152	76975	53,1
diffeq	32836	21675	34
tseng	19806	10398	47,5
des	35540	24514	31

## 7.7 Resultados

A figura 7.10 representa o posicionamento final dos CLBs no interior e com todos os I/Os na fronteira do FPGA sem nenhum tipo de violação, onde o plano X x Y representa as coordenadas das células no FPGA.

Figura 7.10- Posicionamento final (FPGA “seq”)



Fonte: Figura elaborada pelo autor

A tabela 7.4 apresenta detalhadamente os resultados dos HPWLs dos FPGAs em cada um dos posicionadores, com seus respectivos números de células e tamanhos finais. Os menores valores dos comprimentos dos fios dentre os quatro posicionadores estão em destaque na tabela.

Tabela 7.3 Comparação com o estado da arte

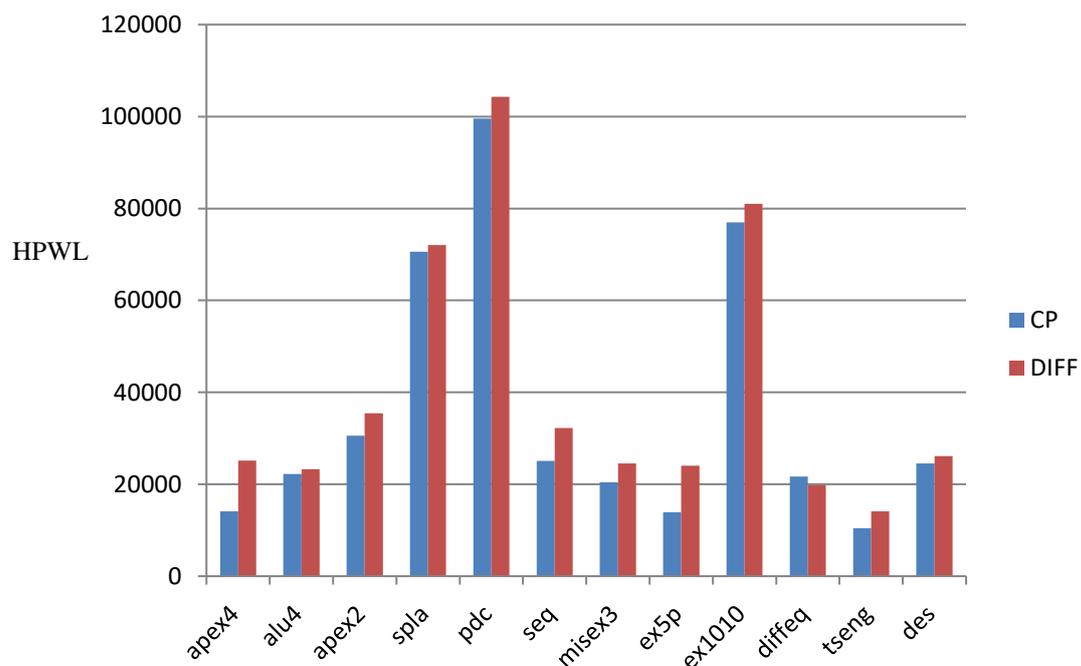
FPGA	#CLBs	Tamanho	VPR	Star+	DIFF*	CP**
apex4	1262	36 × 36	22865	22136	25166	<b>14115</b>
alu4	1522	40 × 40	22038	<b>21090</b>	23269	22239
apex2	1878	44 × 44	32545	31361	35428	<b>30595</b>
spla	3690	61 × 61	71193	70870	72017	<b>70572</b>
pdc	4575	68 × 68	<b>98470</b>	103699	104287	99544
seq	1750	42 × 42	31026	28377	32226	<b>25076</b>
misex3	1397	38 × 38	22699	21751	24534	<b>20428</b>
ex5p	1064	33 × 33	19993	19583	24040	<b>13925</b>
ex1010	4598	68 × 68	72613	<b>71220</b>	81029	76975
diffeq	1497	39 × 39	16363	<b>15555</b>	19830	21675
tseng	1407	33 × 33	10419	<b>9932</b>	14093	10398
des	1591	40 × 40	29165	29488	26137	<b>24514</b>
<b>Média</b>			37449	37088	40171	<b>35838</b>

DIFF = DIFFUSION CP=CHAOTIC PLACE

As figuras 7.11, 7.12 e 7.13 apresentam o desempenho final do Chaotic Place em comparação com os posicionadores Diffusion, VPR e Star+ respectivamente, onde o eixo das ordenadas representa os HPWLs.

Comparando com o desempenho do posicionador Diffusion o Chaotic Place obteve menores comprimentos em todos os FPGAs com exceção do FPGA “diffeq”.

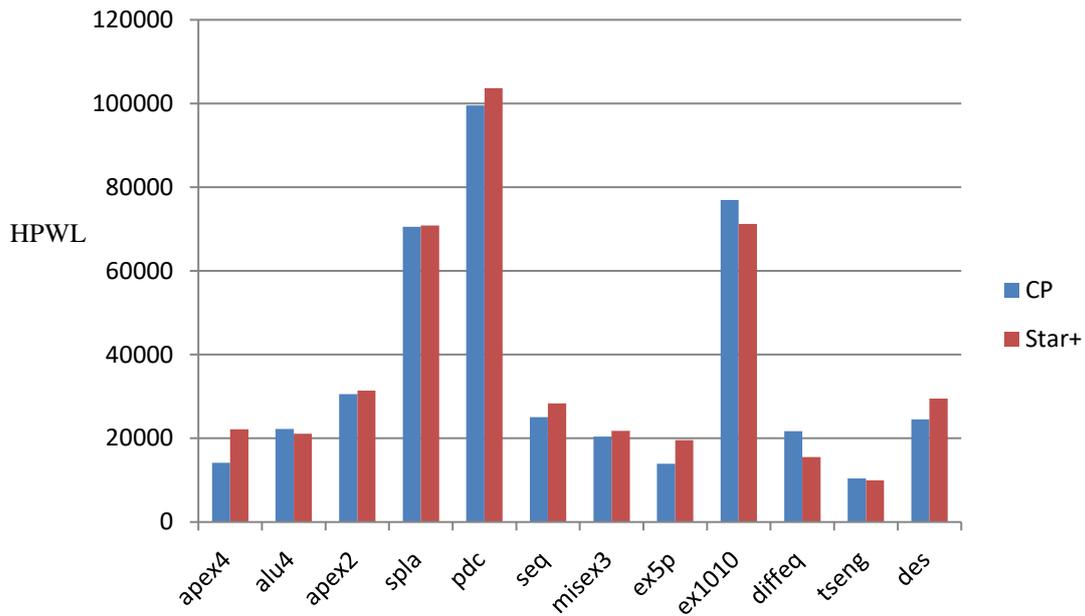
Figura 7.11- Chaotic Place vs Diffusion



Fonte: Figura elaborada pelo autor

Em relação ao posicionador Star+, o Chaotic Place obteve maiores comprimentos nos FPGAs “alu4”, “ex1010”, “diffeq” e “tseng”, e menores comprimentos nos demais.

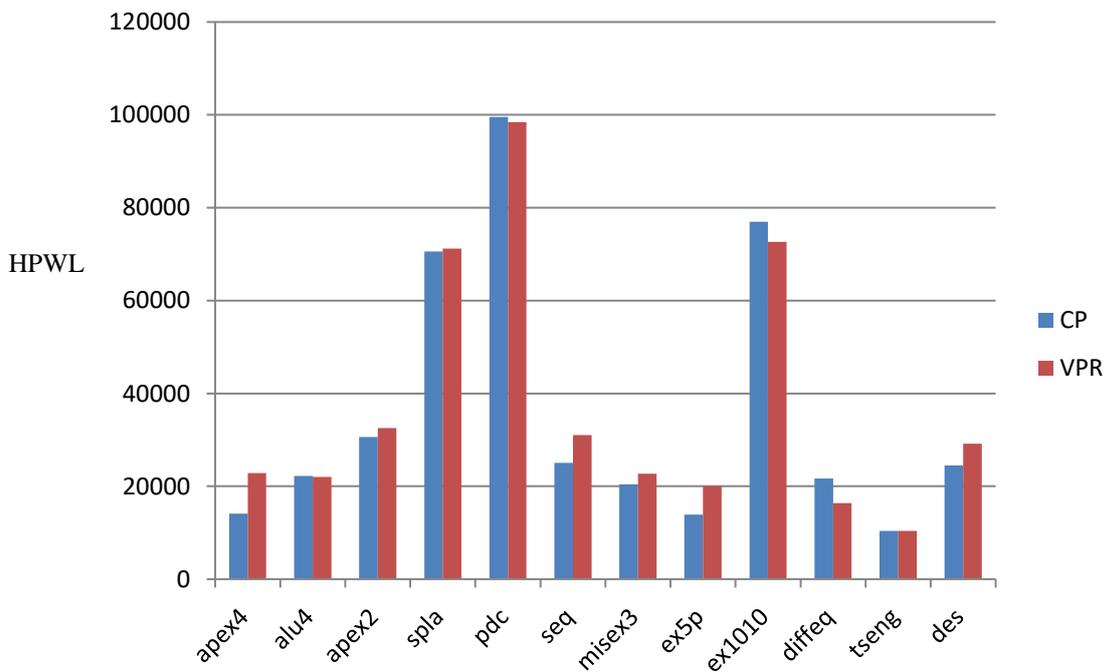
Figura 7.12-Chaotic Place vs Star+



Fonte: Figura elaborada pelo autor

E para finalizar, comparando o Chaotic Place com o VPR o experimento apresentou menores comprimentos com exceção dos FPGAs “alu4”, “pdc”, “ex1010” e “diffeq”.

Figura 7.13-Chaotic Place vs VPR

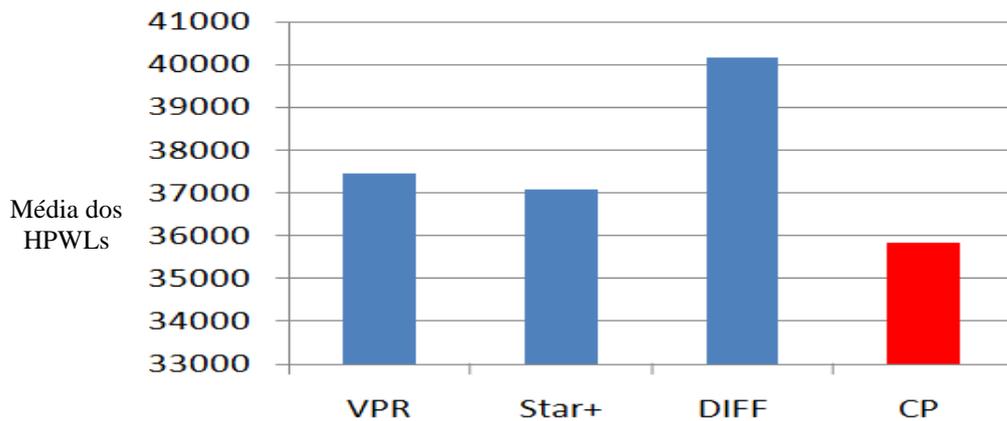


Fonte: Figura elaborada pelo autor

Os resultados foram comparados aos posicionadores: VPR, Star+ e Diffusion. O experimento gerou em média uma redução do HPWL em 4,3% comparado ao VPR, 3,37% comparado ao Star-Place e 10,78% comparado ao Diffusion. Numa média geral entre 5% a 6% de redução no comprimento dos fios. A Figura 7.11 representa o resumo do comparativo

entre o Chaotic Place e os demais posicionadores, onde são comparadas as médias entre os HPWLs finais.

Figura 7.14- Comparativo entre o Chaotic Place e os demais.



Fonte: Figura elaborada pelo autor

Em uma comparação absoluta o Chaotic Place atingiu menores HPWLs em sete FPGAs, num total de doze, comparado com os três posicionadores. Os posicionadores foram escolhidos para comparação em virtude de estarem em publicações recentes correspondendo ao estado da arte para FPGAs homogêneos e também pelo fato de possuírem diferentes algoritmos e abordagens: O VPR baseia-se em simulated annealing, O Star+ é analítico quadrático e o DIFF tem seu posicionamento detalhado baseado nas equações de difusão.

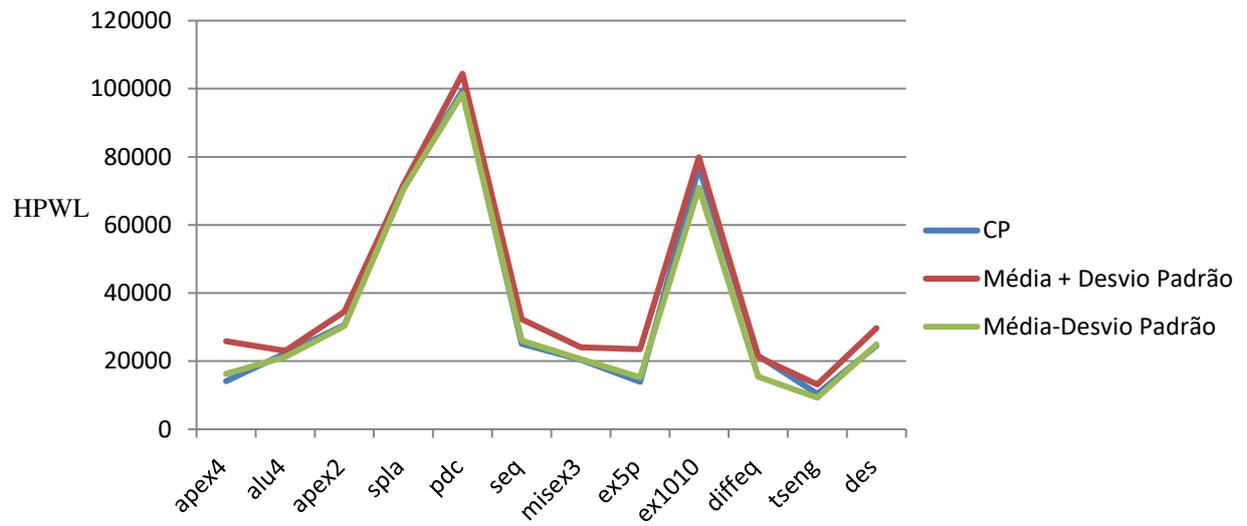
Por outro lado, em uma análise estatística mais profunda foram calculadas as variâncias e consequentemente os desvios padrões dentre os valores obtidos para cada FPGA, e como resultado é observado que o Chaotic Place possui resultados equivalentes ou superiores ao estado da arte dos posicionadores de células em FPGAs Homogêneos.

A tabela 7.3 apresenta cada FPGA, a variância e o desvio padrão gerados enquanto a Figura 7.12 apresenta os valores obtidos pelo Chaotic Place dentro da faixa de tolerância tendo como base em cada desvio padrão constatando que em nenhum caso houve grande dispersão a ponto de invalidar o experimento.

Tabela 7.4 Desvio Padrão de cada FPGA dentre os posicionadores

FPGA	Variância	Média	Desvio Padrão
apex4	23169207	21071	4813
alu4	798634	22159	894
apex2	4500098	32482	2121
spla	388449	71163	623
pdc	8536602	101500	2922
seq	10057800	29176	3171
misex3	2981502	22353	1727
ex5p	17296498	19385	4159
ex1010	19797331	75459	4449
diffeq	8335362	18356	2887
tseng	3743332	11211	1935
des	5792410	27326	2407

Figura 7.15 – Representação do Chaotic Place dentro da faixa de tolerância em relação aos outros posicionadores



Fonte: Figura elaborada pelo autor.

## VIII Conclusões e Trabalhos Futuros

### 8.1 Conclusões

Neste trabalho foi desenvolvido um posicionamento não linear para células em FPGAs (Chaotic Place). Para o experimento foram utilizados os *benchmarks* do *Microelectronics Center of North Carolina* (MCNC) (*Challenge Placement and Route*). De início foi enunciado e provado um resultado relevante constatando que um posicionamento analítico é um Sistema Dinâmico Caótico justificando a metodologia e as ferramentas inseridas. Para isso foram utilizadas ferramentas da Teoria dos Sistemas Dinâmicos Caóticos e da Teoria Espectral dos Grafos.

Dado um FPGA com seu conjunto de *nets* é construído o mapeamento das mesmas, e após esta etapa é definida a função custo sendo uma versão simplificada unidimensional da função LOG-SUM-EXP com o intuito de obter um posicionamento inicial das células no eixo real (na reta). A seguir é definida a Matriz Laplaciana Não Linear gerando o sistema de equações homogêneo a ser resolvido. Para a resolução foi usada uma rede neural de Hopfield e para sua convergência foi utilizado o método do Gradiente Descendente. O experimento mostrou que não houve aumento exponencial do número de épocas utilizado em relação ao aumento de células em cada *benchmark*.

Para o mapeamento das *nets*, resolução do sistema pela rede neural, espalhamento, legalização e posicionamento detalhado foi utilizada a linguagem Java e para o levantamento bidimensional dos pontos foi utilizado o Matlab.

Após a resolução do sistema foi utilizado o Teorema de Takens para construir o levantamento dos pontos obtidos no estado estacionário da rede na forma de um Atrator Estranho consistindo no resultado bidimensional desejado. Para isso foram utilizadas as duas primeiras coordenadas, tendo em vista que o teorema proporciona uma figura espacial de dimensão igual a doze.

O espalhamento das células foi obtido multiplicando cada coordenada pela raiz quadrada do número total de células (Homotetia) e realizando uma translação do eixo das ordenadas de modo a se obter somente pontos com coordenadas inteiras e positivas.

No processo de legalização para as células que ficaram sobrepostas foram criadas *bounding boxes* ao redor da célula sobreposta e em cada um, ao encontrar um espaço vazio a célula foi deslocada e alocada neste espaço vazio até não haver mais sobreposição. Em relação aos I/Os, originalmente posicionados no interior do FPGA, são alocados na extremidade, no ponto mais próximo do original.

E finalmente para o posicionamento detalhado um algoritmo iterativo diminui o tamanho de cada *net* até o HPWL final possuir o menor valor possível, onde a diferença média entre o posicionamento inicial e o final foi de 31,75%.

Tomado como comparativo o *Half-Perimeter-Wire-Length* (HPWL) os resultados finais obtiveram entre 5% a 6% de diminuição do HPWL em comparação com os posicionadores: VPR, Diffusion e Star+ representando o estado da arte para FPGAs homogêneos. Detalhadamente o Chaotic Place obteve melhores resultados em sete benchmarks num total de doze comparados aos três posicionadores. Além disso, uma análise estatística detalhada utilizando as variâncias e os desvios padrões mostraram que não houve discrepâncias entre os resultados do Chaotic Place comparados aos demais.

### 8.2 Trabalhos Futuros

Para Trabalhos futuros são apresentadas algumas observações após a primeira implementação do Chaotic Place. Destacamos:

- **Escolha dos eixos após o levantamento dos pontos**  
O experimento mostrou que nem sempre os dois primeiros eixos apresentaram os melhores posicionamentos. Nos FPGAs pequenos não houve mudanças significativas, porem, nos maiores a escolha de outros eixos gerou posicionamentos melhores. A ideia seria gerar um posicionamento  $n$ -dimensional ( $n > 2$ ) e a partir deste selecionar os eixos através de projeções onde o resultado seja satisfatório. Para isso o autor pretende utilizar ferramentas oriundas da Geometria Diferencial.
- **Posicionamento inicial e Posicionamento final**  
A ideia aqui é diminuir a diferença entre os posicionamentos iniciais e finais gerando menor *runtime* no posicionamento como um todo. Para nossa proposta devem-se utilizar ferramentas da Teoria da Probabilidade, mais especificamente Redes Bayesianas sendo que o problema se resume a um problema de decisão. Assim espera-se obter um posicionamento inicial bem próximo do final diminuindo o processo do posicionamento detalhado.
- **Legalização**  
O maior problema nesta fase é a colocação dos I/Os na grade, pois em alguns casos (quando seu número é elevado) aumentam consideravelmente o HPWL final além de aumentar o processo computacional. Mais uma vez pretende-se utilizar Redes Bayesianas ou outra ferramenta de tomada de decisão para tal.
- **Posicionamento Detalhado**  
Dois itens precisam de bastante atenção: Quando o algoritmo para e quando se deve diminuir o tamanho da net, de modo que não aumente consideravelmente o tamanho das demais. Para o segundo item a ideia seria utilizar algum método de tomada de decisão como análise estatística para decidir qual *net* será alterada sem comprometer as demais.
- **FPGAs heterogêneos**  
Nossa meta é adaptar as técnicas empregadas neste trabalho para posicionar além dos CLBs e IOs, células específicas como BRAMs e DSPs (MARCEL GORT, JASON H. AANDERSON, 2012). Seguindo a topologia dos *benchmarks* do ISPD-Contest de 2016, as células específicas devem ser alocadas em faixas pré-definidas. Daí o levantamento dos pontos pelo Teorema de Takens deverá ser adaptado.
- **Generalização para FPGA-3D**  
A literatura mostra que a tecnologia dos circuitos 3d apresenta algumas vantagens em relação aos usuais. O maior problema em posicionamento 3d seria a construção de um algoritmo viável e suficiente para tal. Hoje não existem algoritmos para posicionamento 3d. Existem algoritmos 2d que são adaptados para tal (HENTSCHK R., JOHANN M., RICARDO R, 2012), (HENTSCHK R., JOHANN M., RICARDO R.; PINTO F., 2007). Nosso objetivo é gerar um posicionamento que posicione as células diretamente na figura espacial. O primeiro passo já foi dado tendo em vista que o teorema de Takens permite esta distribuição dos pontos no objeto tridimensional. Para definir o número de *ties* o autor pretende utilizar ferramentas da Topologia Diferencial, como o Estudo das Folheações. E não menos importante é saber decidir se dado um conjunto de nets qual a melhor topologia para implementação: 2d ou 3d.

- **Implementação em Hardware**

Velocidade de processamento é um parâmetro fundamental para medir o desempenho de um posicionamento. A implementação em hardware garante um runtime muito menor se comparado à implementação em software. Além de se poder explorar a estrutura do hardware em termos de paralelismo aumentando consideravelmente o desempenho da aplicação.

- **Roteamento**

A ideia para este problema seria construir uma técnica híbrida entre consiste em seguir os passos de John Hopfield na construção de uma Rede Neural Recorrente para encontrar as melhores rotas de um circuito para definir o roteamento, e dentro desta inserirmos na função sinal conceitos probabilísticas para tomadas de decisão, de modo que se possa definir a menor rota proporcionando a ausência de congestionamentos.

- **Aprendizagem de Máquina, Posicionamento e caracterização das Nets**

Este é o item mais desafiador. Consiste em catalogar um número suficiente de resultados de posicionamentos em uma lista de *benchmarks* e a partir deles criar classes de modo que um novo posicionamento seja associado a uma delas. Para a classificação dos posicionamentos será utilizada a Teoria Espectral dos Grafos (além de se fazer necessário um estudo aprofundado em aprendizagem de máquina). Uma área com resultados relevantes e com muitos problemas em aberto, fundamentais para a compreensão dos grafos, que por sua vez implicam diretamente em EDA. A *outline* da pesquisa será: (1) Selecionar o maior número possível de benchmarks. (2) Identificar cada Circuito como um grafo e através das propriedades espectrais separá-los em conjuntos (Monóides, Grupos, Anéis, etc.). (3) Dado um novo circuito identificá-lo entre os catalogados compreendendo sua topologia eliminando etapas no processo de posicionamento.

## Referências

ABABEI, C et al. Three-dimensional place and route for FPGAs, **IEEE Transactions on Computer-aided design of integrated Circuits and Systems**. Page(s): 1132 – 1140. Volume: 25 Issue. DOI: 10.1109/TCAD. 2005.855945. June 2006.

ANDRZEJ KOS, ZBIGNIEW NAGÓRNY, (2009) FPGA placement by using Hopfield neural network, **Microelectronics International**, Vol. 26 Issue: 1, pp.22-32, ISSN: 1356-5362. <https://doi.org/10.1108/13565360910923133>.

ASGHAR A. and PARVEZ H. An Improved Diffusion Based Placement Algorithm for Reducing Interconnect Demand in Congested Regions of FPGAs. **International Journal of Reconfigurable Computing**. Volume 2015 (2015), Article ID 756014, 10 pages. <http://dx.doi.org/10.1155/2015/756014>

AUDIP PANDIT, ALI AKOGLU. Wirelength Prediction for FPGAs. **International Conference on Field Programmable Logic and Applications**, 2007. FPL 2007.27-29 Aug. 2007. Amsterdam, Netherlands. DOI: 10.1109/FPL.2007.4380760.

A. AYKANAT C., BULTAN T, HARITAOGLU I. A fast neural-network algorithm for VLSI cell placement. **Neural Networks** Volume 11, Issue 9, December 1998, Pages 1671-168. [https://doi.org/10.1016/S0893-6080\(98\)00089-6](https://doi.org/10.1016/S0893-6080(98)00089-6)

BARONE, D.; MARON, G.; RAMOS, E. A. Measuring the Differences between Spatial Intelligence in Different Individuals using Lyapunov Exponents. In: **International Conference on Mass Data Analysis of Images and Signals, 2012**, Berlin, Germany. ibai publishing, 2012. v. 01. p. 10-24. Disponível em: [http://www.mda-signals.de/BestPaper/MDA2012\\_35.pdf](http://www.mda-signals.de/BestPaper/MDA2012_35.pdf)

BENCHMARKS. **The FPGA Place-and-Route Challenge**. Disponível em: <http://www.eecg.toronto.edu/~vaughn/challenge/challenge.html>. Consultado em: 14/05/2017.

BETZANDB V. and ROSE J. VPR: a new packing, placement and routing tool for FPGA research. In: Luk W., Cheung P.Y.K., Glesner M. Field-Programmable Logic and Applications. FPL 1997. **Lecture Notes in Computer Science**, volume 1304. Springer, Berlin, Heidelberg. Print ISBN978-3-540-63465-2. DOI [https://doi.org/10.1007/3-540-63465-7\\_226](https://doi.org/10.1007/3-540-63465-7_226).

BHASKER J. A VHDL Primer. **Prentice Hall series in innovative technology**. Prentice Hall PTR, 1995. ISBN 0131814478, 9780131814479.

BHASKER, J.; CHADHA, R. **Static Timing Analysis for Nanometer Designs: a practical approach**. Number of Pages XX, 572. eBook ISBN. 978-0-387-93820-2. DOI 10.1007/978-0-387-93820-2. Springer US, 2009.

BOEING, G. Visual Analysis of Nonlinear Dynamical Systems: Chaos, Fractals, Self-Similarity and the Limits of Prediction. **Systems** 2016, 4(4), 37; DOI:10.3390/systems4040037

BONATTI C, DIAZ L.; VIANA M.; Dynamics beyond uniform Hyperbolicity. A Global Geometric and Probabilistic Perspective. **Springer Science & Business Media**, 2006.ISBN 3540268448, 9783540268444. DOI: 10.1007/b138174.

BRENNER U. et al. BonnPlace: Placement of leading-edge chips by Advanced combinatorial Algorithms. **IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems**(Volume: 27, Issue: 9, Sept. 2008).Page(s):1607 – 1620. DOI: 10.1109/TCAD.2008.927674

C. CHEN, Z.-W. JIANG, T.-C. HSU, H.-C. CHEN and Y.-W. CHANG. NTUPlace3:An Analytical Placer for Large-Scale Mixed-Size Designs with Preplaced Blocks and Density Constraint. **IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems**(Volume: 27, Issue: 7, July 2008).Page(s):1228 – 1240. DOI:10.1109/TCAD.2008.923063

CHAN, J. CONG, and K. SZE. Multilevel Generalized Force-directed Method for Circuit Placement. **ISPD '05. 2005 international symposium on Physical design**. Pages 185-192. San Francisco, California, USA — April 03 - 06, 2005. ISBN: 1-59593-021-3 doi: 10.1145/1055137.1055177.

CHAN, T. et al. mPL6: Enhanced multilevel mixed-size placement with congestion control. In: NAM, G.-J.; CONG, J. **International Symposium on Physical Design** , v. 2006. San Jose, California, USA p. 212-214. ISBN 1595932992.

COLLIER R. et al. Advancing genetic algorithm approaches to field programmable gate array placement with enhanced recombination operators. **Evolutionary Intelligence**, Volume 7, Issue 4, pp 183–200. 2014.Print ISSN1864-5909. DOI<https://doi.org/10.1007/s12065-014-0114-6>.

CONG, J.; LUO, G. Thermal-Aware 3D Placement. In: XIE, Y.; CONG, J.; SAPATNEKAR, S. (Ed.).**Three Dimensional Integrated Circuit Design: eda, design and micro architectures**. Number of Pages XII, 284. Edition Number1. Copyright 2010. EBook ISBN 978-1-4419-0784-4. DOI 10.1007/978-1-4419-0784-4.

CRISTINEL ABABEI. Speeding Up FPGA PLACEMENT via Partitioning and Multithreading. **International Journal of Reconfigurable Computing Volume 2009** (2009), Article ID 514754, 9 pages <http://dx.doi.org/10.1155/2009/514754>.

CYBENKO, G. Approximations by superpositions of sigmoidal functions. **Mathematics of Control, Signals, and Systems**, 2 (4), 303-314. 1989.Print ISSN0932-4194. DOI<https://doi.org/10.1007/BF02551274>

DEYE, R. and SUGIAHARA, G. Generalized Theorems for Nonlinear State Space Reconstruction.**US National Library of Medicine National Institutes of Health**. Published: March 31, 2011. <https://doi.org/10.1371/journal.pone.0018295>

F. TAKENS (1981). Detecting strange attractors in turbulence. **Dynamical Systems and Turbulence**, Warwick 1980, Lecture Notes in Mathematics, Volume 898. ISBN 978-3-540-11171-9. Springer-Verlag, 1981, p. 366. DOI: 10.1007/BFb0091924.

FLACH, GHILHERME AUGUSTO. Discrete gate sizing and timing-driven detailed placement for the design of digital circuits. UFRGS. **Programa de Pós-graduação em Microeletrônica**. Tese de Doutorado. 2015.<http://hdl.handle.net/10183/134330>.

FOGAÇA, MATEUS PAIVA. A new quadratic formulation for incremental timing-driven placement. UFRGS. **Programa de Pós-graduação em Microeletrônica**. Dissertação de Mestrado. 2016.<http://hdl.handle.net/10183/164067>.

FREEMAN, ROSS H., Configurable electrical circuit having configurable logic elements and configurable interconnects, **U. S. Patent 4,870, 302**, September 26. 1989. <http://www.google.com/patents/USRE34363>.

HASSIBI A., BOYD S., and J. HOW J. P. Control of Asynchronous Dynamical Systems with Rate Constraints on Events. **IEEE Conference on Decision and Control**. 7-10 Dec. 1999. Phoenix, AZ, USA, USA. **Print ISSN: 0191-2216**. DOI:10.1109/CDC.1999.830133.

HAYKIN, SIMON. Neural Networks: A Comprehensive Foundation 2nd. **Prentice Hall PTR Upper Saddle River**, NJ, USA ©1998. ISBN:0132733501.

HENTSCHKE, RENATO. Algoritmos para posicionamento de células em circuitos VLSI. **Programa de Pós-graduação em Computação**. Dissertação de Mestrado (2002). <http://hdl.handle.net/10183/2598>.

HENTSCHKER., JOHANN M., RICARDO R. Placement de Circuitos 3D Considerando o Planejamento de 3D-Vias. **Rita. Revista de Informática teórica e aplicada**. Volume 19 Número 1. 2012. DOI: <http://dx.doi.org/10.22456/2175-2745.7386>.

HENTSCHKE, R.; JOHANN, M.; REIS, R. New trends and technologies in computer aided learning for computer-aided design: **Ifip tc10 working conference: Edutech 2005**, october 20–21, Perth, Australia. In: Boston, MA: Springer US, 2005. chp. Blue Macaw: A Didactic Placement Tool Using Simulated Annealing, p. 37–47.

HENTSCHKE R., JOHANN M., RICARDO R.; PINTO F. 3D-Vias Aware Quadratic Placement for 3D VLSI Circuits. **IEEE Computer Society Annual Symposium on VLSI, 2007. ISVLSI '07**. 9-11 March 2007. Porto Alegre, Brazil. DOI:10.1109/ISVLSI.2007.1

HOPFIELD, J. J. Neural networks and physical systems with emergent collective computational abilities. **National Academy of Sciences of the United States of America**. Volume - 79. 1982. DOI - 10.1073/pnas.79.8.25541982.

HOROWITZ, PAUL; HILL, WINFIELD. The Art of Electronics (2nd ed.). **Cambridge University Press**. 1989. Pages 1125. ISBN 978-0-521-37095-0.

HOU, W; HONG X. A path-based timing-driven quadratic placement algorithm. **Design Automation Conference, 2003. ASP-DAC 2003. Asia and South Pacific**. 24-24 Jan. 2003.

Kitakyushu, Japan, Japan. Print ISBN: 0-7803-7659-5. DOI:10.1109/ASPDAC.2003.1195119

HUSSAIN A. et al. Genetic Algorithm for Traveling Salesman Problem with Modified Cycle Crossover Operator. **Computational Intelligence and Neuroscience Volume 2017** (2017), Article ID 7430125, 7 pages <https://doi.org/10.1155/2017/7430125>.

JARROD A. ROY et al. CAPO: Congestion Driven Placement for Standard-cell and RTL Netlists with incremental Capability. **Modern Circuit Placement**. Pp 93-133. 2007.Springer, Boston, MA. Online ISBN978-0-387-68739-1. DOI: [https://doi.org/10.1007/978-0-387-68739-1\\_5](https://doi.org/10.1007/978-0-387-68739-1_5).

KAHNG, A. et al.VLSI Physical Design: From Graph Partitioning to Timing Closure. 1. ed. **Springer Netherlands**, 2011. e-book ISBN 978-90-481-9591-6. DOI: 10.1007/978-90-481-9591-6

KAHNG, A. B. et al. Global and Detailed Placement. (2011) In: **VLSI Physical Design: From Graph Partitioning to Timing Closure**. Springer, Dordrecht. Online ISBN978-90-481-9591-6. DOI[https://doi.org/10.1007/978-90-481-9591-6\\_4](https://doi.org/10.1007/978-90-481-9591-6_4)

KAHNG, A. B. et al. Architecture and Details of a High Quality, Large-Scale Analytical Placer. **IEEE/ACM International Conference on Computer-Aided Design, 2005. ICCAD-2005**. 6-10 Nov. 2005. San Jose, CA, USA, USA. Print ISBN: 0-7803-9254-X. DOI: 10.1109/ICCAD.2005.1560188.

KIM M.-C. and MARKOV I.L. ComPLx: A Competitive Primal-dual Lagrange Optimization for Global Placement. **49th ACM/EDAC/IEEE Design Automation Conference (DAC)**, 2012. 3-7 June 2012. ISBN: 978-1-4503-1199-1. DOI: 10.1145/2228360.2228496.

KIM M.-C., VISWANATHAN N., ALPAER C. J., MARKOV I. L., and RAMIJI S. MAPLE: Multilevel Adaptive Placement for Mixed-Size Designs. **ISPD '12 Proceedings of the 2012 ACM international symposium on International Symposium on Physical Design**. Pages 193-200. Napa, California, USA — March 25 - 28, 2012. ISBN: 978-1-4503-1167-0. DOI: 10.1145/2160916.2160958.

KUMAR R.; GARG V.; MARCUS S.I. Predicates and Predicate Transformer for supervisory control of discrete Dynamical System. **IEEE Transactions on Automatic Control**. Volume: 38, Issue: 2, Feb 1993. Page(s):232 – 247. DOI: 10.1109/9.250512.

KUON, IAN; ROSE, JONATHAN. Measuring the gap between FPGAs and ASICs. **IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems**. Volume: 26, Issue: 2, Feb. 2007. Page(s): 203 – 215.DOI: 10.1109/TCAD.2006.884574.

LEE D.-J. , KIM M.-C. and MARKOV I. L. SimPL: An Effective Placement Algorithm. **IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems**(Volume: 31, Issue: 1, Jan. 2012). 19 December 2011. Page(s):50 – 60. DOI: 10.1109/TCAD.2011.2170567.

Li, T.Y and Yorke J.A. Period Three Implies Chaos, **The American Mathematical Monthly** Vol. 82, No. 10 (Dec., 1975), pp. 985-992. Published by: Mathematical Association of

America. DOI: 10.2307/2318254. Stable URL: <http://www.jstor.org/stable/2318254>.

LIMA, ELON LAGES. Álgebra linear. Rio de Janeiro: **IMPA** 2016. Nona Edição. Número de páginas 357. ISBN 9788524400896.

LIN T.H et al. An efficient and effective analytical placer for FPGAs. **Design Automation Conference (DAC)**, 50th ACM/EDAC/IEEE. 29 May-7 June 2013. Austin, Texas. ISBN: 978-1-4503-2071-9. DOI: 10.1145/2463209.2488746.

LOPES, ARTUR O. Uma breve introdução à Matemática da Mecânica Quântica. **31º Colóquio Brasileiro de Matemática-IMPA** 2017. Primeira Edição. Número de páginas 219. ISBN 9788524404405.

LU J., CHEN P., CHANG C.-C., SHA L., HUANG D. J.-S., TENG C.-C. and CHENG C.-K., ePlace: Electrostatics Based Placement Using Nesterov's Method. **Design Automation Conference (DAC)**, 2014 51st ACM/EDAC/IEEE. 1-5 June 2014. San Francisco, CA, USA. Electronic ISBN: 978-1-4799-3017-3. DOI:10.1145/2593069.2593133.

MANSMANN, J. et al. Maximizing the impact of power ICs via time-to-market CAD driven power ASIC strategy. **Applied Power Electronics Conference and Exposition, 1992. APEC '92. Seventh Annual.** 23-27 Feb. 1992. Boston, MA, USA. DOI:10.1109/APEC.1992.228435.

MAÑE R. On the dimension of the compact invariant sets of certain nonlinear maps. In D. A. Rand and L.-S. Young. **Dynamical Systems and Turbulence**, Warwick 1980, **Lecture Notes in Mathematics, Volume 898**. ISBN 978-3-540-11171-9. Springer-Verlag, 1981. pp. 230–242. 1981. DOI: 10.1007/Bfb0091916.

MAÑE R. A proof of the Stability Conjecture. **Publications Mathématiques de L'IHÉS**. Vol 66, page 161-210.1987. ISSN: 0073-8301. Stable URL: [http://www.numdam.org/item?id=PMIHES\\_1987\\_\\_66\\_\\_161\\_0](http://www.numdam.org/item?id=PMIHES_1987__66__161_0).

MAÑE R. Teoria Ergódica. Projeto Euclides. **IMPA**. Primeira Edição 1983. Páginas:388. Rio de Janeiro. ISBN:978-85-244-0137-4

MARCEL GORT, JASON H. ANDERSON. Analytical PLACEMENT for heterogeneous FPGAs. **22nd International Conference on Field Programmable Logic and Applications (FPL)**, 29-31 Aug. 2012. Oslo, Norway. **Electronic ISBN: 978-1-4673-2256-0. DOI: 10.1109/FPL.2012.6339278.**

MARON, G.; BARONE, D. A. C.; RAMOS, E. A. Spatial cognition degree of development classification Using multi-layer perceptron and largest Lyapunov exponents. **Congress on Computational Intelligence and 11th Brazilian Congress on Computational Intelligence (BRICS-CCI & CBIC)**, 2013 **BRICS**. 8-11 Sept. 2013. Ipojuca, Brazil. Electronic ISBN: 978-1-4799-3194-1. DOI: 10.1109/BRICS-CCI-CBIC.2013.88.

MATTHEW A., STEFFAN J., BENTZ V. Speeding Up FPGA Placement: Parallel Algorithms and Methods. **IEEE 22nd Annual International Symposium on Field-Programmable Custom Computing Machines (FCCM)**. 11-13 May 2014. Boston, MA,

USA. DOI:10.1109/FCCM.2014.60.

MAY ROBERT M. Simple mathematical models with very complicated. **Dynamics.Nature.261**, 459–467 (10 June 1976). DOI: 10.1038/261459a0.

MERRIS R. Laplacian Matrices of Graphs: A Survey.**Linear Algebra and its Applications**. Volumes 197–198, ELSEVIER. January–February 1994, Pages 143-176. [https://doi.org/10.1016/0024-3795\(94\)90486-3](https://doi.org/10.1016/0024-3795(94)90486-3)

MONTEIRO J. Algoritmo de Posicionamento analítico-detalhado guiado a caminhos críticos. **Dissertação de Mestrado em Microeletrônica**. UFRGS. 2014. <http://hdl.handle.net/10183/116139>.

MOORE G. E. Cramming more components onto integrated circuits, Reprinted from Electronics, volume 38, number 8, April 19, 1965, pp.114 ff.**IEEE Solid-State Circuits Society Newsletter**( Volume: 11, Issue: 3, Sept. 2006 ). Page(s):33 – 35. PrintISSN: 1098-4232. DOI:10.1109/N-SSC.2006.4785860.

PALIS, JACOB. A global perspective for non-conservative dynamics. **Annales de l'Institut Henri Poincaré (C) Non Linear Analysis**Volume 22, Issue 4, July–August 2005, Pages 485-507. ELSEVIER. <https://doi.org/10.1016/j.anihpc.2005.01.001>

PALIS, JACOB; SMALE, S. Structural Stability Theorems. **Matematika**. 1969, Volume 13, Issue 2, Pages 145–155. <http://mi.mathnet.ru/eng/mat/v13/i2/p145>.

PALIS, JACOB; TAKENS, FLORIS. Hyperbolicity and Sensitive Chaotic Dynamics at Homoclinic Bifurcations Fractal Dimensions and Infinitely Many Attractors in Dynamics. January 1995. **Cambridge Studies in Advanced Mathematics**. ISBN: 9780521475723.

PEIXOTO, M. M. On Structural Stability. **ANNALS OF MATHEMATICS**. Annals of Mathematics Second Series, Vol. 69, No. 1 (Jan., 1959), pp. 199-222. DOI: 10.2307/1970100. Stable URL: <http://www.jstor.org/stable/1970100>.

PELLERIN, DAVID; HOLLEY M. Practical Design Using Programmable Logic. **Prentice Hall**; 1 edition (February 8, 1991).320 pages. ISBN-10: 0137238347. ISBN-13: 978-0137238347.

PETRY, A.; BARONE, D. A. C. Preliminary experiments in speaker verification using time-dependent largest Lyapunov exponents. **Computer Speech & Language**. Volume 17, Issue 4, October 2003, Pages 403-413. [https://doi.org/10.1016/S0885-2308\(03\)00029-9](https://doi.org/10.1016/S0885-2308(03)00029-9).

PUGET, J. C. et al. Jezz: An effective legalization algorithm for minimum displacement. **28th Symposium on Integrated Circuits and Systems Design**. New York, NY, USA: ACM, 2015. (SBCCI '15), p. 19:1–19:5. ISBN 978-1-4503-3763-2. <http://doi.acm.org/10.1145/2800986.2801013>.

RAMOS E., BANDEIRA, V., REIS, R., BONTORIN, G. Chaotic Synchronization of Neural Networks in FPGA. **Computational Neuroscience: First Latin American Workshop, LAWCN**, Porto ALEGRE. NOV-2017.Print ISBN978-3-319-71010-5.

DOI[https://doi.org/10.1007/978-3-319-71011-2\\_2](https://doi.org/10.1007/978-3-319-71011-2_2).

ROY J. A., ADYA S. N., PAPA D. A., and MARKOV I. L. MinCut Floor placement. **IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems**. Volume 25 Issue 7, July 2006. Page 1313-1326. DOI: 10.1109/TCAD.2005.855969.

RUDIN, W. Principles of Mathematical Analysis. **McGraw-Hill Education**; 3 edition (16 Feb. 1976). 352 pages. ISBN-10: 007054235X. ISBN-13: 978-0070542358.

SIDIROPOULOS, H. et al. On supporting efficient partial reconfiguration with just-in-time compilation. **IEEE 26th International Parallel and Distributed Processing Symposium Workshops & PhD Forum (IPDPSW), 2012**. 21-25 May 2012. Shanghai, China. Print ISBN:978-1-4673-0974-5. DOI: 10.1109/IPDPSW.2012.40

SIEGELMANN, HAVA T. D. SONTANG, EDUARDO. Turing computability with neural nets. **Applied Mathematics Letters**. Volume 4, Issue 6, 1991, Pages 77–80. [https://doi.org/10.1016/0893-9659\(91\)90080-F](https://doi.org/10.1016/0893-9659(91)90080-F).

SHANNON, CLAUDE E. A mathematical theory of communication. **The Bell System Technical Journal**(Volume: 27, Issue: 3, July 1948). Page(s):379 – 423. July 1948. Print ISSN: 0005-8580. DOI:10.1002/j.1538-7305.1948.tb01338.x.

SHERWANI, N. Algorithms for VLSI physical design automation, 572. **Springer US**. 1999.eBook ISBN: 978-0-306-47509-2. DOI: 10.1007/b116436.

SMITH, ROGER.A Spectral Theoretic Proof of Perron–Frobenius, **Mathematical Proceedings of the Royal Irish Academy**, The Royal Irish Academy, 102 (1): 29–35. 2006.Stable URL: <http://www.jstor.org/stable/i20459816>.

SPINDLER, P.; SCHLICHTMANN, U.; JOHANNES, F. M. Abacus: fast legalization of standard cell circuits with minimal movement. **ISPD '08 Proceedings of the 2008 international symposium on Physical design**. Pages 47-53. Portland, Oregon, USA — April 13 - 16, 2008. ISBN: 978-1-60558-048-7 DOI: 10.1145/1353629.1353640.

STRUZYNA M. Sub-Quadratic Objectives in Quadratic Placement. **Design, Automation & Test in Europe Conference & Exhibition (DATE)**, 2013. 18-22 March 2013. Grenoble, France. DOI: 10.7873/DATE.2013.372.

TAGHAVIT, X. YANG, and B.-K. CHOI. Dragon2005: Large-Scale Mixed size Placement Tool. **ISPD '05. 2005 International Symposium on Physical design**. Pages 245-247. San Francisco, California, USA — April 03 - 06, 2005. ISBN: 1-59593-021-3 DOI: 10.1145/1055137.1055191.

TAGHAVI, T. et al. Congestion Minimization in Modern Placement Circuits. In: NAM, G.-J.; CONG, J. (Ed.). **Modern Circuit Placement: best practices and results**. Springer, Boston, MA. 2007. Online ISBN978-0-387-68739-1. DOI[https://doi.org/10.1007/978-0-387-68739-1\\_6](https://doi.org/10.1007/978-0-387-68739-1_6).

TANENBAUM, A. S.: Operating Systems: Design and Implementation ed. **Prentice-Hall International**. 1987. ISBN: 0-13-142938-8.

VISWANATHAN, N.; CHU, C. C.-N. FastPlace: efficient analytical placement using cell shifting, iterative local refinement and a hybrid net model. **IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems**(Volume: 24, Issue: 5, May 2005). Page(s):722 – 733. 25 April 2005. DOI: 10.1109/TCAD.2005.846365.

WANG, L.-T.; CHANG, Y.-W.; CHENG, K.-T. T. (Ed.). Electronic Design Automation: Synthesis, Verification, and Test. San Francisco, CA, USA: **Morgan Kaufmann Publishers Inc.** 1 edition (March 12, 2009). 972 pages. ISBN-10: 0123743648. ISBN-13: 978-0123743640.

WERBOS PAUL J. The Roots of Backpropagation. From Ordered Derivatives to Neural Networks and Political Forecasting. **Wiley-Interscience New York, NY, USA**©1994. ISBN: 0-471-59897-6.

XU M., GREWAL G., AEIBI S. Star Place: A new analytic method for FPGA placement. **Integration, the VLSI Journal** Volume 44, Issue 3, June 2011, Pages 192–204.<https://doi.org/10.1016/j.vlsi.2011.02.001>.

YANG M.; ALMAINI A.E.A.; WANG L. ; PENGJUN WANG. FPGA Placement using Genetic algorithm with Simulated Annealing. **6th International Conference On ASIC, 2005. ASICON 2005.** 24-27 Oct. 2005. Shanghai, China. Print ISBN: 0-7803-9210-8. DOI:10.1109/ICASIC.2005.1611450.

YANG S., GAYASEN A. et al. Routability-Driven FPGA Placement Contest. **ISPD '16 Proceedings of the 2016 on International Symposium on Physical Design.** Pages 139-143. Santa Rosa, California, USA — April 03 - 06, 2016. ISBN: 978-1-4503-4039-7 DOI: 10.1145/2872334.2886419.