



**COPPE/UFRJ**

**SÍNTESE DE FILTROS COM BAIXA SENSIBILIDADE UTILIZANDO TÉCNICAS  
EVOLUTIVAS**

Leonardo Bruno de Sá

Tese de Doutorado apresentada ao Programa de Pós-graduação em Engenharia Elétrica, COPPE, da Universidade Federal do Rio de Janeiro, como parte dos requisitos necessários à obtenção do título de Doutor em Engenharia Elétrica.

Orientador: Antonio Carneiro de Mesquita Filho

Rio de Janeiro  
Janeiro de 2009

SÍNTESE DE FILTROS COM BAIXA SENSIBILIDADE UTILIZANDO TÉCNICAS  
EVOLUTIVAS

Leonardo Bruno de Sá

TESE SUBMETIDA AO CORPO DOCENTE DO INSTITUTO ALBERTO LUIZ  
COIMBRA DE PÓS-GRADUAÇÃO E PESQUISA DE ENGENHARIA (COPPE) DA  
UNIVERSIDADE FEDERAL DO RIO DE JANEIRO COMO PARTE DOS  
REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE DOUTOR EM  
CIÊNCIAS EM ENGENHARIA ELÉTRICA.

Aprovada por:

---

Prof. Antonio Carneiro de Mesquita Filho, Dr. d'État

---

Prof. Jorge Lopes de Souza Leão, Dr.Ing.

---

Prof. José Franco Machado do Amaral, D.Sc.

---

Prof. José Soares Coutinho Filho, D.Sc.

---

Prof. Marco Aurélio Cavalcanti Pacheco, PhD

RIO DE JANEIRO, RJ - BRASIL

JANEIRO DE 2009

Sá, Leonardo Bruno de

Síntese de Filtros com Baixa Sensibilidade Utilizando Técnicas Evolutivas / Leonardo Bruno de Sá. – Rio de Janeiro: UFRJ/COPPE, 2009.

X, 121 p.: il.; 29,7 cm.

Orientador: Antonio Carneiro de Mesquita Filho

Tese (doutorado) – UFRJ/ COPPE/ Programa de Engenharia Elétrica, 2009.

Referências Bibliográficas: p. 109-118.

1. Síntese de circuitos. I. Mesquita Filho, Antonio Carneiro de. II. Universidade Federal do Rio de Janeiro, COPPE, Programa de Engenharia Elétrica. III. Título.

## DEDICATÓRIA

*À minha família*

## AGRADECIMENTOS

Ao Prof. Antonio Carneiro de Mesquita Filho orientador e amigo que amalhei nesses três profícuos anos de convívio fraterno.

Resumo da Tese apresentada à COPPE/UFRJ como parte dos requisitos necessários para a obtenção do grau de Doutor em Ciências (D.Sc.)

## SÍNTESE DE FILTROS COM BAIXA SENSIBILIDADE UTILIZANDO TÉCNICAS EVOLUTIVAS

Leonardo Bruno de Sá

Janeiro / 2009

Orientador: Antonio Carneiro de Mesquita Filho

Programa: Engenharia Elétrica

A computação evolutiva é o ramo da ciência da computação que utiliza o processo biológico da evolução natural Darwiniana como metáfora na solução de problemas. Esta tese investiga a aplicação de técnicas evolutivas na síntese de filtros digitais e analógicos com baixa sensibilidade. A inclusão da sensibilidade como parte integrante do processo de síntese permitiu a obtenção de filtros digitais menos sensíveis à quantização dos coeficientes e filtros analógicos menos sensíveis à variação dos valores dos componentes.

No domínio digital é proposta a representação utilizando a matriz adjacência. Esta representação mostrou ser capaz de aumentar o espaço de busca quando comparada com a representação em árvores utilizada pela programação genética.

No domínio analógico, redes de casamento de impedâncias são utilizadas como estudo de caso. As redes sintetizadas pelo processo evolutivo mostraram-se menos sensíveis à variação dos valores dos componentes do que àquelas que empregam técnicas tradicionais.

Abstract of Thesis presented to COPPE/UFRJ as a partial fulfillment of the requirements for the degree of Doctor of Science (D.Sc.)

## SYNTHESIS OF LOW-SENSITIVITY FILTERS USING EVOLUTIONARY TECHNIQUES

Leonardo Bruno de Sá

January / 2009

Advisor: Antonio Carneiro de Mesquita Filho

Program: Electrical Engineering

Evolutionary computation is the branch of computer science that uses the biological process of natural evolution proposed by Darwin as a metaphor to solve problems. This thesis investigates the application of evolutionary techniques in the synthesis of electronic circuits, particularly low sensitivity filters. The inclusion of the sensitivity in the synthesis process allowed the generation of low coefficient sensitivity filters.

In the digital domain, a representation using adjacency matrix is proposed. This representation is able to increase the search space when it is compared with the tree-based representation used by the genetic programming.

In the analog domain, impedance matching networks are used as case study. The networks synthesized by the evolutionary process show to be less sensitive than the networks synthesized by conventional methods.

## ÍNDICE

1	INTRODUÇÃO.....	1
1.1	CONSIDERAÇÕES INICIAIS .....	1
1.2	OBJETIVOS DO TRABALHO.....	3
1.3	ESTRUTURA DO TRABALHO .....	4
2	FUNDAMENTOS TEÓRICOS DAS TÉCNICAS EVOLUTIVAS .....	5
2.1	INTRODUÇÃO .....	5
2.2	COMPONENTES DE UM EA.....	6
2.2.1	Representação.....	6
2.2.1.1	<i>Strings</i> Binários .....	7
2.2.1.2	Números Inteiros ou Reais.....	7
2.2.1.3	Grafos .....	8
2.2.2	Variação.....	10
2.2.2.1	Crossover ou Recombinação.....	10
2.2.2.2	Mutação .....	11
2.2.3	Avaliação.....	12
2.2.4	Seleção .....	12
2.2.4.1	Seleção Proporcional .....	14
2.2.4.2	Seleção por Normalização .....	14
2.2.4.3	Seleção por Torneio.....	16
2.2.4.4	Seleção por Truncamento.....	16
2.2.5	Crítérios de Parada .....	17
2.3	REPRODUÇÃO DA POPULAÇÃO .....	18
2.4	TEOREMA DO ESQUEMA.....	19
2.5	PLATAFORMAS DE EVOLUÇÃO .....	20
2.6	HARDWARE ADAPTATIVO.....	21
2.7	SÍNTESE DE CIRCUITOS.....	22
3	SÍNTESE EVOLUTIVA DE FILTROS DIGITAIS COM BAIXA SENSIBILIDADE..	23
3.1	INTRODUÇÃO .....	23
3.2	COMPONENTES DE UM FILTRO DIGITAL.....	24
3.3	PROJETO DE FILTROS DIGITAIS RECURSIVOS.....	25
3.3.1	Filtros Digitais Recursivos usando Aproximações .....	25
3.3.1.1	Transformação da Resposta Invariante ao Impulso.....	26
3.3.1.2	Transformação Bilinear .....	29
3.3.2	Filtros Digitais Recursivos usando Métodos de Otimização .....	30
3.4	PROJETO E SÍNTESE EVOLUTIVA DE FILTROS DIGITAIS.....	33
3.4.1	Filtros Digitais com Baixa Sensibilidade .....	34
3.4.2	Matriz Adjacência.....	36
3.4.3	Representação de Filtros Digitais Usando Matriz Adjacência .....	37
3.4.4	Indivíduos Inválidos Usando Matriz Adjacência.....	39
3.4.5	Filtros Digitais Computáveis.....	41
3.4.6	Operadores Genéticos usando Matriz Adjacência.....	44
3.4.7	Comparação com GP.....	45



3.4.7.1	Crossover .....	46
3.4.7.2	Mutação .....	48
3.4.7.3	Inversão .....	49
3.5	ESTUDO DE CASOS .....	50
3.5.1	Filtro de Segunda Ordem .....	50
3.5.1.1	Cálculo da Aptidão .....	50
3.5.2	Filtro de Quinta Ordem .....	56
3.5.3	Equalizadores de Fase .....	62
4	SÍNTESE EVOLUTIVA DE CIRCUITOS ANALÓGICOS .....	68
4.1	INTRODUÇÃO .....	68
4.2	CIRCUITOS INTEGRADOS ANALÓGICOS .....	69
4.2.1	Evolução Intrínseca .....	69
4.2.2	Evolução Extrínseca .....	72
4.2.2.1	Estudo de Casos .....	76
4.3	CIRCUITOS DISCRETOS ANALÓGICOS .....	83
4.3.1	Evolução Intrínseca .....	84
4.3.2	Evolução Extrínseca .....	84
4.3.3	Síntese Evolutiva de Filtros Analógicos Passivos .....	85
4.3.4	Estudo de Casos .....	86
4.3.4.1	Redes de Casamento de Impedâncias .....	86
5	CONCLUSÕES E PROPOSTAS DE TRABALHO .....	107

## ABREVIACOES

IC:	Integrated Circuit (Circuito Integrado);
CMOS:	Complementary Metal Oxide Semiconductor (Semicondutor Metal- xido Complementar);
DAC:	Digital-to-Analog Converter (Conversor Analgico-Digital);
EA:	Evolutionary Algorithms (Algoritmos Evolutivos);
EHW:	Evolvable Hardware (Hardware Evolucionrio);
EP:	Evolutionary Programming (Programao Evolutiva);
ES:	Evolution Strategies (Estratgias Evolutivas);
FIR:	Finite Impulse Response (Resposta Finita ao Impulso);
FPGA:	Field Programmable Gate Arrays (Arranjo de Portas Programvel em Campo);
GA:	Genetic Algorithms (Algoritmos Genticos);
GP:	Genetic Programming (Programao Gentica);
IIR:	Infinite Impulse Response (Resposta Infinita ao Impulso);
RFT:	Real Frequency Technique (Tcnica da Freqncia Real);
TPG:	Transducer Power Gain (Ganho de Potncia de Transduo);
VHF:	Very High Frequency (Freqncia Muito Alta)
VSWR:	Voltage Standing Wave Ratio (Razo de Voltagem da Onda Estacionria);

# 1 Introdução

## 1.1 Considerações Iniciais

Nas últimas duas décadas foi constatado um aumento significativo na complexidade dos projetos de sistemas eletrônicos. À medida que a capacidade de processamento aumentou, impulsionada pelos crescentes níveis de integração de componentes, sistemas cada vez complexos puderam ser desenvolvidos. Geralmente, estes sistemas apresentam muitos componentes que, através de interações não-lineares, apresentam uma dinâmica de difícil caracterização. As redes neurais e as equações diferenciais com comportamento caótico são exemplos de sistemas complexos que podem ser utilizados como ferramentas de estudo na modelagem de sistemas complexos reais.

Tendo em vista que os sistemas biológicos são ordens de grandeza mais complexos que os mais complexos sistemas eletrônicos produzidos atualmente, muitos pesquisadores buscaram inspiração na natureza para a solução do problema do aumento da complexidade. Sistemas biológicos possuem características desejáveis a qualquer sistema projetado artificialmente, tais como: capacidade de adaptação ao meio-ambiente, evolução através da troca de material genético e capacidade de “aprendizado”. Dentre as características mais notáveis dos sistemas biológicos, talvez a adaptabilidade seja a mais atraente do ponto de vista prático em sistemas eletrônicos complexos.

Diversas técnicas que simulam o comportamento de sistemas biológicos têm sido empregadas nas mais variadas áreas de aplicação da eletrônica e da computação. Dentre estas técnicas destacam-se: os Algoritmos Genéticos (GA) [1-2], a lógica Fuzzy [3] e as redes neurais [4].

Os GA e a Programação Genética (GP) [5] são subclasses do ramo da Inteligência Computacional definido como Computação Evolutiva, a qual consiste na análise e síntese de algoritmos probabilísticos inspirados nos princípios Darwinianos

da reprodução, mutação e seleção natural das espécies. Mais especificamente, estes algoritmos probabilísticos são conhecidos como Algoritmos Evolutivos (EA).

Nos EA, cada indivíduo é representado por uma única solução de um dado problema de otimização. Portanto, uma população é um conjunto de possíveis soluções e a aptidão de uma solução representa sua medida de desempenho. Novos indivíduos são criados a cada geração variando aleatoriamente os indivíduos da população. Cada indivíduo é avaliado e os melhores indivíduos são probabilisticamente selecionados para a próxima população:

$$P(t+1) = S(E(V(P(t)))) \quad (1-1)$$

Onde  $P(t)$  é a população de soluções na geração  $t$ ,  $V(.)$  é um operador de variação aleatória,  $E(.)$  é um operador de avaliação e  $S(.)$  é um operador de seleção. Este processo se repete a cada geração até que o critério de parada do algoritmo seja satisfeito [6].

A associação de um EA com uma plataforma de hardware reconfigurável deu origem a um termo denominado Hardware Evolucionário (EHW). Resultados superiores àqueles obtidos com o emprego de técnicas de projetos tradicionais ressaltam o potencial do EHW na síntese de sistemas complexos, como é o caso do desenvolvimento de sistemas de controle para robôs autônomos [7-10].

Uma das principais áreas de aplicação da Computação Evolutiva é a síntese de circuitos. Atualmente, o projeto automático de circuitos digitais tem sofrido um significativo progresso [11-13], enquanto o projeto automático de circuitos analógicos mostra-se mais complexo, uma vez que o processo de síntese de circuitos analógicos inclui a topologia e o dimensionamento dos componentes.

Dentre os blocos funcionais de um sistema eletrônico, os filtros ocupam destacada posição. A filtragem é um processo pelo qual o espectro de frequência de um sinal pode ser modificado ou manipulado por alguma especificação a ser atendida. Dentre os diversos usos da filtragem pode-se citar: a eliminação da contaminação do

sinal por ruído, a remoção da distorção causada por um canal de transmissão, a separação de dois sinais que foram misturados para maximizar a utilização do canal, a demodulação de sinais e a limitação em banda de sinais.

Este trabalho investiga a síntese de filtros de baixa sensibilidade utilizando técnicas evolutivas em ambos os domínios: digital e analógico. Na síntese de filtros analógicos, desvios dos valores nominais dos componentes podem ocorrer devido a efeitos de temperatura, umidade e outras alterações das condições ambientais além de erros causados pela tolerância dos componentes. Estas variações de valores alteram os parâmetros do filtro, por exemplo, a frequência central de um filtro passa-faixa. A figura de mérito que mede a mudança em um parâmetro particular de um filtro causada por uma dada mudança em um valor de componente é a sensibilidade [14].

Na síntese de filtros digitais, os coeficientes e os sinais devem ser quantizados antes de serem armazenados e processados por um processador digital de sinais. Caso a sensibilidade do filtro seja considerada, os efeitos da quantização afetam a resposta do filtro e é possível obter estruturas mais ou menos sensíveis aos efeitos causados pela quantização,

## **1.2 Objetivos do Trabalho**

O objetivo principal deste trabalho é investigar o emprego de técnicas evolutivas na síntese de filtros com baixa sensibilidade, apresentando metodologias que incorporem na função de avaliação dos indivíduos a sensibilidade dos filtros.

Em particular, a síntese de filtros utilizando a representação em matriz adjacência é examinada. Esta representação tende, naturalmente, a gerar circuitos com baixa sensibilidade, uma vez que esta representação induz a geração de múltiplos ramos em paralelo e múltiplas realimentações.

### **1.3 Estrutura do Trabalho**

Este trabalho encontra-se estruturado em cinco capítulos. No Capítulo 2 são apresentados os conceitos básicos sobre Algoritmos Evolutivos e suas aplicações em circuitos eletrônicos. No Capítulo 3 são analisados os métodos de síntese de filtros digitais e o emprego da matriz adjacência no projeto de filtros com baixa sensibilidade. No Capítulo 4 diversas abordagens de síntese evolutiva de circuitos analógicos são discutidas. Estudo de casos com circuitos empregando apenas transistores MOS sintetizados por métodos evolutivos visando à construção de circuitos integrados analógicos são estudados neste capítulo. A utilização da matriz adjacência na síntese de uma rede de casamento de impedâncias passiva com baixa sensibilidade também é analisada. As conclusões e sugestões de trabalhos futuros são apresentadas no Capítulo 5.

## 2 Fundamentos Teóricos das Técnicas Evolutivas

### 2.1 Introdução

A Computação Evolutiva é um ramo da ciência da computação que consiste na síntese de algoritmos probabilísticos baseada em mecanismos evolutivos encontrados na natureza, tais como auto-organização e comportamento adaptativo.

Os EA são estratégias gerais de busca que requerem do projetista pouco conhecimento sobre o domínio do problema. O termo EA é um termo geral que descende de três abordagens evolutivas distintas: a Programação Evolutiva (EP) [15-16], os GA [1-2] e as Estratégias Evolutivas (ES) [17]. Embora no passado estas três abordagens mantivessem suas peculiares – a EP só usava seleção por torneio e os GA apenas seleção por roleta – atualmente os pesquisadores, dadas as grandes semelhanças, optaram por não mais fazer distinções, chamando todas as abordagens de EA [6].

Um outro tipo de EA que tem sido largamente utilizado na Computação Evolutiva é a GP [5]. Seu mecanismo de funcionamento é muito semelhante ao do GA, contudo a principal característica da GP é a utilização de estruturas de dados em forma de *árvore* para representar soluções potenciais [18]. Evolução Gramatical [48], GP Cartesiano [49], GA Adaptativo [50], GA Paralelo [51-52] e GA Compacto [53] são exemplos de outros EA encontrados na literatura.

Os EA possuem uma vasta gama de aplicações nas mais variadas áreas, tais como planejamento de rotas de robôs autônomos, determinação da entalpia mínima de reações químicas, otimização da resposta ao degrau de uma turbina, particionamento do espaço de estados em sistemas classificadores, síntese de circuitos eletrônicos entre outras [19]. Entretanto, o desempenho superior dos EA em comparação com outras técnicas de otimização está limitado ao problema analisado.

O Teorema *No Free Lunch* (NFL Theorem) [20] estabelece a inexistência de um algoritmo que seja superior aos demais na solução de todos os problemas de otimização. Desta maneira, os EA possuem um desempenho superior com relação à solução de uma classe específica de problemas e, em contrapartida, um desempenho inferior com relação à solução de outras classes de problemas.

Em geral, os métodos clássicos são mais eficientes para solução de problemas lineares, quadráticos, fortemente convexos e unimodais. Por outro lado, os EA têm sido aplicados a problemas de otimização quando estes apresentam descontinuidades, não-diferenciabilidade, múltiplos objetivos, superfícies de erro multimodais entre outros onde os métodos convencionais não apresentam uma resposta satisfatória [21].

## **2.2 Componentes de um EA**

Todo EA é composto por um conjunto de elementos fundamentais independentemente das possíveis ramificações da Computação Evolutiva. Os mais importantes dentre estes elementos são:

### **2.2.1 Representação**

A representação é o modo pelo qual os parâmetros que descrevem uma solução potencial são codificados em uma estrutura de dados capaz de ser processada computacionalmente.

Em se tratando de representação, são encontradas na literatura diversas analogias com termos biológicos. Um exemplo clássico destas analogias são as definições de *genótipo* e *fenótipo*. Em um EA, cada parâmetro que descreve uma solução potencial é considerado um *gene*. O conjunto de genes é denominado *genótipo* ou *cromossomo*. O *fenótipo* é a realização física do genótipo. No caso particular da síntese de circuitos o fenótipo é o circuito montado na bancada.



Existem três codificações usualmente empregadas em problemas usando EA: *strings binários, números inteiros ou reais e grafos*.

### 2.2.1.1 Strings Binários

Os GA foram inicialmente concebidos como cromossomos binários de tamanho fixo. Isto fez com que os *strings* binários se tornassem as estruturas de dados mais amplamente utilizadas para representar soluções potenciais em um EA.

### 2.2.1.2 Números Inteiros ou Reais

Embora os primeiros GA utilizassem cromossomos com representação binária, um grande número de aplicações utilizando representação inteira e real tem sido reportado na literatura [22]. Em muitos casos é conveniente que os valores dos componentes sejam especificados diretamente. A Fig. 2-1 ilustra as três representações abordadas - binária, inteira e real - em um cromossomo linear.

0	0	1	1	0	0	1	1
34	67	234	345	675	432	678	21
1.9	8.9	5.6	2.7	3.4	0.6	0.1	7.7

Figura 2-1. Exemplos das representações com *strings* binários, números inteiros e reais.

A conversão de *strings* binários em números inteiros é imediata. No entanto, a conversão de *strings* binários em números reais requer uma certa manipulação algébrica.

Considere um *string* binário  $b_0b_1\dots b_L$  de tamanho  $L$  onde se deseja obter um número real  $x \in [x_{min}, x_{max}]$  correspondente. Inicialmente, o *string* binário é transformado em um inteiro sem sinal  $\bar{x}$ .

$$\bar{x} = \sum_{i=0}^{L-1} b_i \cdot 2^i \quad (2-1)$$

Posteriormente, para obter  $x$ , basta substituir  $\bar{x}$  em:

$$x = x_{min} + \bar{x} \left( \frac{x_{max} - x_{min}}{2^L - 1} \right) \quad (2-2)$$

O tamanho do *string* determina a precisão da conversão. Geralmente, 12 bits é um tamanho mais que suficiente para aplicações envolvendo EA [6].

### 2.2.1.3 Grafos

Um grafo  $G$  consiste de dois conjuntos: um conjunto  $V = V(G)$  cujos elementos são chamados de *vértices* ou *nós* de  $G$  e um conjunto  $E = E(G)$  de pares *não* ordenados de vértices distintos, chamados de *arestas* ou *ramos* de  $G$  [23].

A mesma definição vale para um *grafo orientado* ou *dígrafo*, a única diferença é que o conjunto  $E = E(G)$  é composto por pares ordenados de vértices distintos, chamados de *arcos* ou *arestas orientadas*. Em ambos os casos a notação  $G(V, E)$  enfatiza os dois conjuntos que fazem parte de  $G$ .

A Figura 2-2 mostra duas topologias de filtros – uma analógica e uma digital – sendo representadas respectivamente por um grafo orientado e um grafo não orientado. Em particular, a representação descrita em (b) recebe o nome de grafo de fluxo de sinal (*signal flow-graph*) [24].

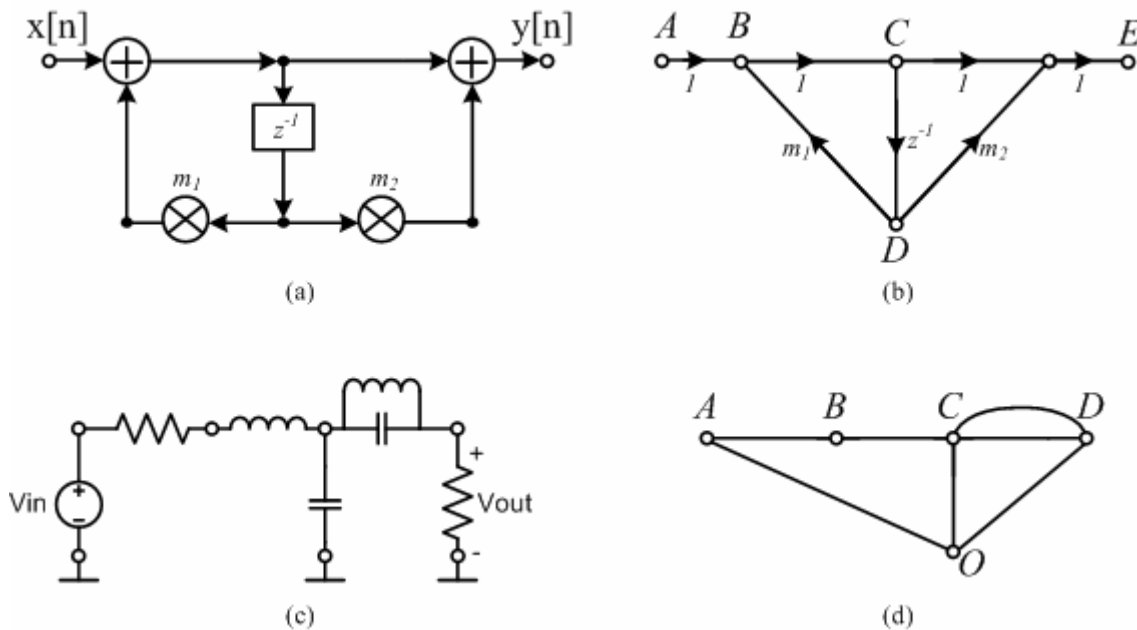


Figura 2-2. (a) Estrutura de um filtro digital de primeira ordem (b) grafo de fluxo de sinal correspondente (c) topologia de um filtro analógico (d) grafo do filtro analógico onde o nó  $O$  representa o nó de terra.

Um *caminho* em um grafo  $G$  consiste em uma seqüência alternada de vértices e arestas  $\{v_0, e_1, v_1, e_2, \dots, e_{n-1}, v_{n-1}, e_n, v_n\}$ . O número  $n$  de arestas é dito o *comprimento* do caminho. Um caminho é dito *fechado* se  $v_0 = v_n$ . Um *ciclo* é um caminho fechado de comprimento maior ou igual a 3 onde todos os vértices são distintos, com exceção de  $v_0 = v_n$ . Um grafo  $G$  é *conexo* se existe um caminho entre quaisquer dois dos seus vértices. Um grafo  $T$  é dito uma *árvore* se  $T$  é conexo e não tem ciclos.

A Fig. 2-3 ilustra o mesmo filtro analógico passivo mostrado na Fig. 2-2(c) representado em forma de árvore. De acordo com a nomenclatura comumente

utilizada em GP, os nós da árvore foram divididos em dois conjuntos: o *conjunto de nós terminais* e o *conjunto de funções primitivas*. Os nós terminais consistem em três tipos de componentes R (resistor), L (indutor) e C (capacitor). As funções primitivas consistem em série e paralelo, denotadas, respectivamente, pelos símbolos + e // [25].

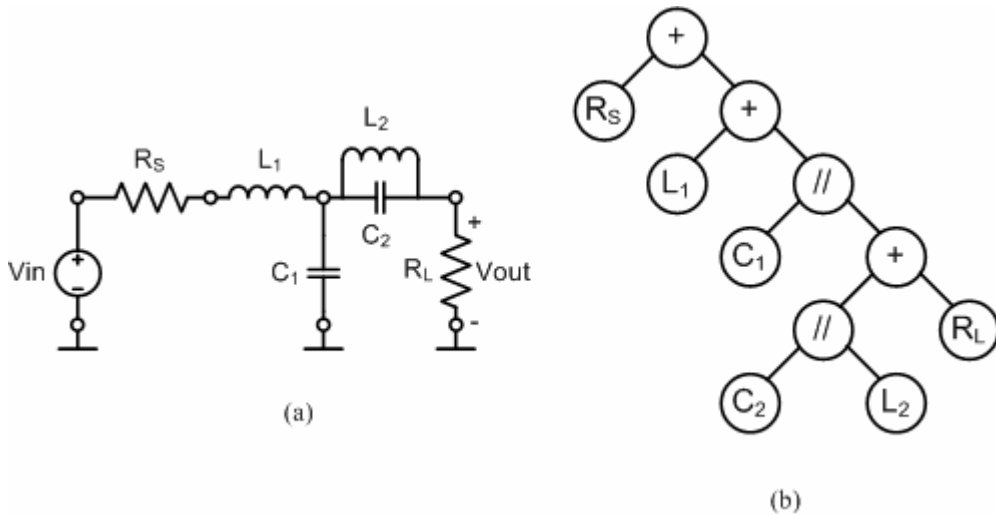


Figura 2-3. (a) Topologia do filtro analógico (b) representação em árvore correspondente.

Uma outra representação que utiliza grafos para descrever seu comportamento é a Máquina de Estados Finita (MEF). As MEFs são uma das abordagens mais utilizadas para modelar comportamentos em jogos [26]. Uma MEF pode ser definida como uma sêxtupla  $\langle \tau, \rho, Q, q_0, s, o \rangle$ , onde:  $\tau$  é o conjunto finito de símbolos de entrada,  $\rho$  é o conjunto finito de símbolos de saída;  $Q$  é o conjunto finito de estados,  $q_0$  é o estado inicial,  $s$  é a função de transição de estados:  $s: Q \times \tau \rightarrow Q$  e  $o$  é a função de saída. Se  $o$  é uma função de  $Q$  e  $\tau$  ( $o: Q \times \tau \rightarrow \rho$ ) a máquina corresponde ao modelo de Mealy e se  $o$  é uma função somente de  $Q$  ( $o: Q \rightarrow \rho$ ) a máquina corresponde ao modelo de Moore [27]. A Fig. 2-4 ilustra uma MEF simplificada do jogo *Pac-Man*.

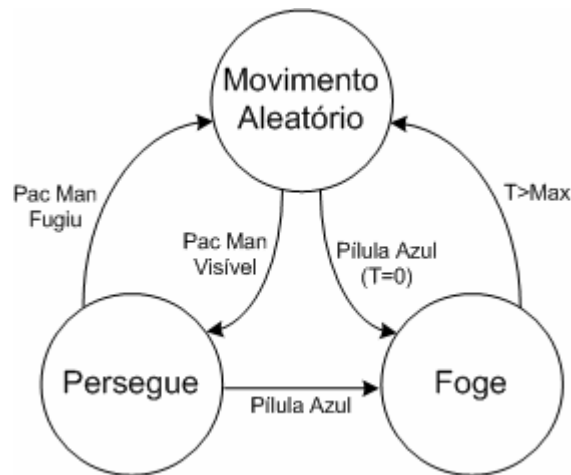


Figura 2-4. MEF simplificada do comportamento dos fantasmas no jogo *Pac-Man*.

Independentemente da estrutura de dados escolhida para a representação de uma solução potencial, a representação pode ser classificada em de *tamanho fixo* ou de *tamanho variável*. Representações de tamanho fixo exibem cromossomos que não alteram seu tamanho ao longo do processo evolutivo, enquanto representações de tamanho variável, embora sejam mais flexíveis, exigem mecanismos de controle de tamanho de genótipos [28].

## 2.2.2 Variação

A diferença genética existente entre indivíduos de uma mesma população gera diferentes valores de aptidão. Os principais mecanismos de variação genética são: a mutação e o crossover. Ambos os mecanismos são dependentes da representação dos indivíduos.

### 2.2.2.1 Crossover ou Recombinação

A idéia básica deste operador, assim como se verifica na natureza, consiste na troca de material genético de dois indivíduos ditos *pais*, resultando em dois novos indivíduos descendentes ditos *filhos*. Em um crossover bem sucedido, os descendentes preservam as informações críticas dos pais mantendo a herança genética de seus genitores.

A operação de crossover é probabilisticamente aplicada a uma dada população. A esta probabilidade dá-se o nome de *taxa de crossover*. Em geral, as taxas de crossover costumam ser superiores às taxas de mutação, levando muitos autores a

considerar o crossover como o principal operador de variação dos algoritmos evolutivos [29].

Existem diversas formas de efetuar o crossover entre dois indivíduos, sendo os *crossover de 1 ponto* e *crossover de 2 pontos* os mais empregados. O *crossover uniforme* também é muito utilizado, mas alguns pesquisadores [30-31] alegam que, por combinarem padrões arbitrários, este tipo de crossover deve ser aplicado em ambientes de reprodução específica, pois tendem a destruir algumas características desejáveis dos indivíduos. A Fig. 2-5 ilustra um exemplo de crossover de 1 ponto e 2 pontos.

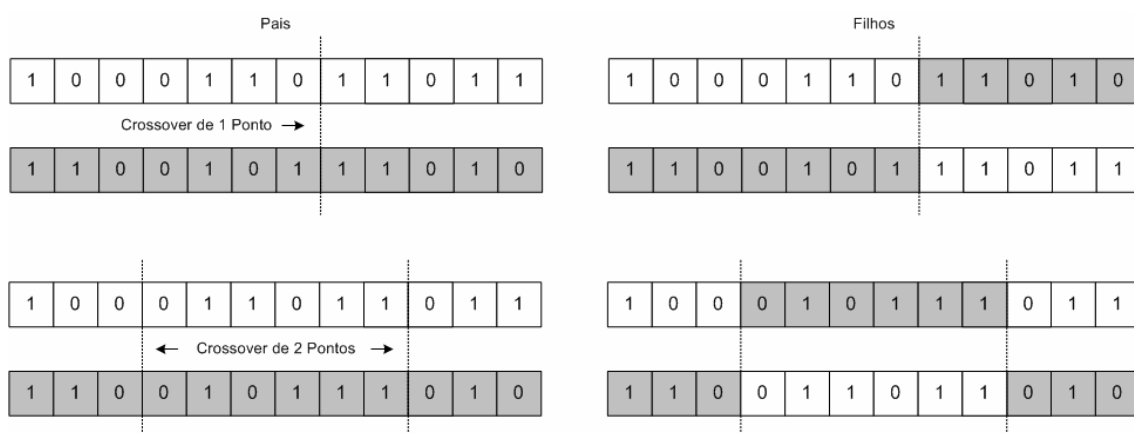


Figura 2-5. Exemplo de crossover de 1 ponto e 2 pontos para um cromossomo binário.

### 2.2.2.2 Mutação

A mutação também é um operador probabilístico aplicado somente a uma parte da população. Ao contrário do crossover que utiliza dois indivíduos, a mutação atua em um único indivíduo alterando um ou mais parâmetros codificados. A mutação se caracteriza por ser um processo de busca aleatória com o objetivo de explorar o espaço de busca nas regiões em que por ventura as soluções potenciais não tenham sido exploradas adequadamente. Desta forma, a mutação garante a exploração de soluções potenciais promissoras que tenham sido eliminadas no processo de seleção ou crossover [2]. A literatura sugere a aplicação deste operador com reduzidas taxas, entretanto diversos autores têm mostrado que o aumento desta taxa para valores superiores a 5% melhora a qualidade da busca por soluções potenciais em determinadas aplicações [32].

Em sua forma mais simples, no caso de um alfabeto binário, a mutação consiste em sortear aleatoriamente um gene e apenas trocar o seu valor por seu

complemento. No caso de representações um pouco mais complexas, como por exemplo, representações que utilizam números reais, uma prática eficaz é adicionar ao gene a ser *mutado* um pequeno número real aleatório [6].

Considere um cromossomo de números reais de tamanho  $L$ ,  $x_1x_2\dots x_{L-1}x_L$ . Sejam respectivamente  $x_i \in \mathfrak{R}$  e  $N(0,1)$ , o gene que irá sofrer mutação também conhecido como *parâmetro objeto* e uma variável aleatória  $N$  com média  $0$  e variância  $1$ . Então o valor de  $x_i$  após a mutação é dado por:

$$\bar{x}_i = x_i + \sigma \cdot N_i(1,0) \quad ; i \in \{1, \dots, L\} \quad (2-3)$$

onde  $\sigma$  é o *tamanho do passo de mutação* também chamado de *parâmetro de estratégia*. Alguns estudos mostram que a evolução de parâmetros objeto conjuntamente com parâmetros de estratégia melhora a qualidade da busca [33].

### 2.2.3 Avaliação

A avaliação consiste em, utilizando uma *função objetivo*, atribuir uma aptidão para cada indivíduo do EA. A aptidão fornece uma medida de quão próximo está um indivíduo de encontrar a solução de um problema de otimização. Normalmente, a função objetivo mapeia a solução em um número real. Se  $f$  é a função objetivo e  $S$  é o espaço de busca, então  $f : S \rightarrow \mathfrak{R}$ .

A definição da função objetivo não é uma tarefa tão simples como pode parecer. A maior parte dos problemas práticos deve atender a múltiplos objetivos e, em geral, conflitantes, uma vez que diversas especificações devem ser levadas em consideração. Um exemplo clássico de especificações conflitantes é a otimização de circuitos digitais que devem operar em altas frequências com baixo consumo.

Outro ponto importante é que a avaliação deve ser relativamente rápida já que em um EA convencional cada indivíduo da população é avaliado. Isto significa que se a avaliação de um indivíduo demorar 1 hora seria necessário 1 ano para executar um EA com 200 indivíduos e 50 gerações.

### 2.2.4 Seleção

A idéia da seleção é privilegiar indivíduos com alta aptidão, aumentando a probabilidade de serem selecionados para a próxima geração. A justificativa para a seleção é baseada no princípio da hereditariedade, estabelece que indivíduos com alta aptidão têm mais chance de produzirem descendentes com elevada aptidão também.

Ao contrário dos operadores de variação que dependem da representação, a seleção depende apenas da aptidão dos indivíduos.

Em uma dada população de  $N$  indivíduos existe um número finito de aptidões diferentes  $f_1 < f_2 < \dots < f_{n-1} < f_n$  ( $n \leq N$ ) associadas a cada um dos  $N$  indivíduos, onde  $f_1$  denota a menor aptidão e  $f_n$  a maior.

O estado da população pode ser descrito pelos valores discretos da *distribuição da aptidão*,  $s(f_i)$ , que representa o número de ocorrências de indivíduos com aptidão  $f_i$  na população. Se  $\bar{s}(f_i)$  descreve a distribuição contínua associada a  $s(f_i)$ , é possível definir a *aptidão média da população antes da seleção* ( $M$ ) e a *aptidão média da população após a seleção* ( $M^*$ ), respectivamente, como [34]:

$$M = \frac{1}{N} \int_{f_1}^{f_n} \bar{s}(f) \cdot f \cdot df \quad (2-4)$$

$$M^* = \frac{1}{N} \int_{f_1}^{f_n} \bar{s}^*(f) \cdot f \cdot df \quad (2-5)$$

De maneira análoga é possível definir a *variância da distribuição da aptidão* antes e após a seleção, respectivamente, como [34]:

$$\sigma^2 = \frac{1}{N} \int_{f_1}^{f_n} \bar{s}(f) \cdot (f - M)^2 \cdot df \quad (2-6)$$

$$(\sigma^*)^2 = \frac{1}{N} \int_{f_1}^{f_n} \bar{s}^*(f) \cdot (f - M^*)^2 \cdot df \quad (2-7)$$

Existem quatro mecanismos principais de seleção: *seleção proporcional*, *seleção por torneio*, *seleção por truncamento* e *seleção por normalização*. Estes mecanismos são caracterizados pela *pressão seletiva* ou *intensidade de seleção*, que avalia quanto peso é dado a indivíduos com elevada aptidão. O emprego de mecanismos de seleção com alta pressão seletiva pode levar o EA a convergir prematuramente, uma vez que

indivíduos com alta aptidão terão probabilidades muito grandes de serem selecionados. A pressão seletiva,  $I$ , é definida por [34]:

$$I = \frac{M^* - M}{\sigma} \quad (2-8)$$

onde  $\sigma$  é o *desvio padrão da distribuição da aptidão* antes do crossover.

#### 2.2.4.1 Seleção Proporcional

A maneira mais intuitiva de aplicar o princípio da hereditariedade é atribuir uma probabilidade de sobrevivência proporcional à aptidão de cada indivíduo. Este mecanismo de seleção é conhecido como seleção proporcional ou *seleção por roleta*. A probabilidade de seleção do  $i$ -ésimo indivíduo da população é:

$$p_i = \frac{f_i}{\sum_{j=1}^N f_j} \quad (2-9)$$

Esta equação exige que as aptidões sejam todas positivas, para isto um simples escalonamento das aptidões deve ser realizado. É possível demonstrar que a pressão seletiva de um mecanismo de seleção proporcional é dada por [35]:

$$I = \frac{\sigma}{M} \quad (2-10)$$

Este mecanismo de seleção é muitas vezes evitado, justamente por apresentar uma elevada pressão seletiva o que causa a rápida perda de diversidade da população e a convergência prematura do algoritmo [6]. Neste caso, mecanismos de controle da pressão seletiva podem atenuar este efeito [2].

#### 2.2.4.2 Seleção por Normalização

Para populações que apresentam uma elevada variância da distribuição da aptidão, alguns estudos indicam que a seleção por normalização mantém a diversidade da população [36].

Inicialmente, todos os indivíduos são ordenados de acordo com suas respectivas aptidões. Em seguida é realizado um escalonamento das aptidões, onde as aptidões



originais são substituídas. Ao indivíduo de menor aptidão é atribuído o índice  $I$  e uma nova aptidão  $\eta^-$  e ao indivíduo de maior aptidão o índice  $N$  e uma nova aptidão  $\eta^+$ . Assim, a probabilidade de seleção usando este mecanismo é uma *função* do índice do indivíduo e não da sua aptidão original como ocorre na seleção por roleta. De acordo com a função utilizada a seleção por normalização pode ser: *linear* ou *exponencial*.

A probabilidade de seleção do  $i$ -ésimo indivíduo da população usando o mecanismo de seleção por normalização linear é dada por [34]:

$$p_i = \frac{I}{N} \cdot \left[ \eta^- + (\eta^+ - \eta^-) \cdot \frac{i-I}{N-I} \right] \quad (2-11)$$

Portanto, as probabilidades de seleção do pior e do melhor indivíduo da população são, respectivamente,  $\frac{\eta^-}{N}$  e  $\frac{\eta^+}{N}$ . Para evitar probabilidades negativas a condição  $\eta^- \geq 0$  deve ser obedecida. Outra condição que deve ser obedecida para que

$$\sum_{i=1}^N p_i = 1 \text{ é } \eta^+ + \eta^- = 2.$$

É possível demonstrar que a pressão seletiva de um mecanismo de seleção por normalização linear é dada por [34]:

$$I = (1 - \eta^-) \frac{1}{\sqrt{\pi}} \quad (2-12)$$

A probabilidade de seleção do  $i$ -ésimo indivíduo da população usando o mecanismo de seleção por normalização exponencial é dada por [34]:

$$p_i = \frac{c-I}{c^N - I} \cdot c^{N-i} \quad (2-13)$$

A pressão seletiva associada a este mecanismo de seleção pode ser calculada com erro inferior a 1,81% para  $\kappa \in [10^{-20}; 0,8]$  como [34]:

$$I \approx \frac{\ln(\kappa)}{-2,548 - 1,086 \cdot \sqrt{\kappa} + 0,403 \cdot \ln(\kappa)} \quad (2-14)$$

onde  $\kappa = c^N$  é introduzido em (2-14) para eliminar a dependência direta da pressão seletiva do tamanho da população. O parâmetro  $c$  obedece à desigualdade  $0 < c < 1$  e determina o grau de “exponencialidade” da seleção, quanto mais próximo de 1 menor a “exponencialidade” [37].

#### 2.2.4.3 Seleção por Torneio

A seleção por torneio consiste em amostrar aleatoriamente  $t > 1$  indivíduos da população e selecionar deterministicamente o de maior aptidão, onde  $t$  é conhecido como o *tamanho do torneio*. Como apenas um indivíduo é selecionado por torneio, é necessário  $N$  torneios para gerar a nova população.

A probabilidade do  $i$ -ésimo indivíduo ser selecionado pela seleção por torneio é dada por [34]:

$$p_i = N^{-t} \left[ (N - i + 1)^t - (N - i)^t \right] \quad (2-15)$$

A pressão seletiva associada a este mecanismo de seleção pode ser calculada como [34]:

$$I = \int_{-\infty}^{\infty} t \cdot x \cdot \frac{1}{\sqrt{2\pi}} \cdot e^{-\frac{x^2}{2}} \cdot \left( \int_{-\infty}^x \frac{1}{\sqrt{2\pi}} \cdot e^{-\frac{y^2}{2}} \cdot dy \right)^{t-1} \cdot dx \quad (2-16)$$

Para uma seleção por torneio binária ( $t = 2$ ), a integral pode ser calculada analiticamente e  $I = \frac{1}{\sqrt{\pi}}$ .

#### 2.2.4.4 Seleção por Truncamento

A seleção por truncamento é realizada com base em um valor de limiar  $T \in [0, 1]$ , onde apenas uma fração de  $T$  indivíduos é selecionada. Todos os indivíduos selecionados possuem a mesma probabilidade de seleção [35].

A pressão seletiva associada a este mecanismo de seleção pode ser calculada como [34]:

$$I = \frac{I}{T} \cdot \frac{I}{\sqrt{2\pi}} \cdot e^{-\frac{f_c}{2}} \quad (2-17)$$

onde  $f_c$  é a menor aptidão dentre os  $T$  indivíduos selecionados.

Dentre os quatro mecanismos de seleção apenas a seleção por truncamento garante a sobrevivência do melhor indivíduo após a seleção. Os demais mecanismos são probabilísticos, o que significa dizer que o melhor indivíduo tem uma probabilidade menor do que 1 de ser selecionado. Uma maneira de evitar que o melhor indivíduo seja descartado pela seleção é usar *elitismo* onde uma parcela dos melhores indivíduos é copiada para a próxima geração.

### 2.2.5 Critérios de Parada

Os critérios de parada mais utilizados na literatura são: *a convergência do algoritmo*, *a solução satisfatória* e *o número máximo de gerações*. O critério de parada baseado na convergência do algoritmo verifica se não há uma variação significativa no processo de busca em um número  $k$  de gerações, enquanto o baseado na solução satisfatória verifica se a solução encontrada satisfaz a todas as especificações.

Tomando como exemplo o caso particular da síntese evolutiva de circuitos, uma população aleatória de circuitos é gerada por um EA dependendo da representação cromossomial escolhida. Ao longo das gerações uma grande variedade de circuitos, compreendendo topologias, valores e tipos de componentes, é criada. A população é avaliada pela função objetivo e uma aptidão é atribuída a cada circuito (indivíduo). Na seqüência, um mecanismo de seleção é empregado para selecionar os indivíduos que sofrerão variação (crossover e mutação). A Fig. 2-6 ilustra o fluxo de funcionamento de um EA.



Figura 2-6. Fluxo de funcionamento de um EA.

### 2.3 Reprodução da População

Outra característica importante em um EA é a maneira como as populações são tratadas, em particular a metodologia de substituição dos indivíduos de uma população para a próxima geração. Os métodos mais comumente empregados são: *generacional* e *steady-state*.

No método generacional a próxima geração é constituída apenas de descendentes da geração corrente, nenhum indivíduo é preservado. Uma pequena variação deste método é utilizar elitismo, preservando apenas o indivíduo mais apto.

No método steady-state a próxima geração é uma combinação de descendentes da geração corrente e indivíduos que foram preservados. Uma parcela da população contendo os piores indivíduos é substituída pelos descendentes da geração corrente, caracterizando um método elitista.

## 2.4 Teorema do Esquema

Um importante teorema [1] que tenta prever o comportamento de um GA quando seus parâmetros são variados é o Teorema do Esquema (*Schema Theorem*). Embora este teorema não consiga explicar muitos aspectos relacionados ao comportamento de um GA, ele fornece um limite inferior para o número de indivíduos que, pertencendo a um dado esquema na geração  $t$ , permanecerá na próxima geração  $t+1$ .

Assumindo um cromossomo binário de tamanho  $L$ , o tamanho do espaço de busca  $S$  associado a esta representação é dado por  $2^L$ . Utilizando símbolos \* (*don't care*) em um *string* binário, é possível definir subconjuntos de  $S$  com atributos em comum, *esquemas*. Por exemplo, (1 1 \* \*) representa o esquema de todas as *strings* pertencentes a  $S$  que possuem os dois primeiros elementos iguais a 1. Portanto, o total de esquemas para o exemplo acima ( $L=4$ ) é dado por  $3^4 = 81$ .

Três conceitos relacionados com um dado esquema devem ser aqui introduzidos: *ordem*, *distância* e *quebra*. A ordem,  $O(H)$ , de um dado esquema  $H$  é o número de *strings* diferentes de \* presentes no esquema. A distância,  $\rho(H)$ , é a distância entre o primeiro e o último *string* diferente de \* presente no esquema. No exemplo,  $O(H)=2$  e  $\rho(H)=2-1=1$ . A quebra de um esquema é caracterizada pela separação de dois bits diferentes de \* após a operação de crossover. A probabilidade dos dois bits 1 do esquema (1 1 \* \*) de serem separados após o crossover de 1 ponto é  $1/3$ .

Seja  $M(H,t)$  o número de indivíduos pertencentes ao esquema  $H$  na geração  $t$ ,  $(t+sel)$  o índice da geração  $t$  após a seleção,  $f(H,t)$  a aptidão média do esquema  $H$  e  $f_m$  a aptidão média da população. A mudança na representação após a seleção do tipo roleta é:

$$M(H,t+sel) = M(H,t) \cdot \frac{f(H,t)}{f_m} \quad (2-4)$$

Assumindo que o crossover é aplicado à geração  $t$  com probabilidade  $p_c$ , uma parte da população não sofre crossover e tem sua representação mantida. Em contrapartida, a parte da população que sofre crossover deve ter calculada as *perdas* e *ganhos* originadas pelas quebras [38]:

$$M(H,t+1) = (1-p_c) \cdot M(H,t) \cdot \frac{f(H,t)}{f_m} + p_c \cdot \left[ M(H,t) \cdot \frac{f(H,t)}{f_m} (1 - \text{perdas}) + \text{ganhos} \right] \quad (2-5)$$

Na derivação do teorema do esquema uma hipótese conservadora é anotada: todas as quebras originam perdas para um dado esquema.

Seja o esquema (1 1 \* \*). O crossover de 1 ponto entre os dois primeiros bits do string 1110 (que pertence ao esquema) com os strings 1000 ou 0100 (que não pertencem ao esquema) resultaria em 1000, 1110, 1100 e 0110. Como em ambos os crossovers pelo menos um descendente pertence ao esquema, nenhuma perda deve ser computada. De maneira semelhante, o mesmo crossover realizado entre os strings 1000 e 0100 resultaria em dois novos strings 1100 e 0100. Como ambos os pais não pertenciam ao esquema e um descendente – 1100 – reside na partição, esta é a fonte dos ganhos na Eq. (2-5). Superestimando as quebras:

$$M(H,t+1) \geq (1-p_c) \cdot M(H,t) \cdot \frac{f(H,t)}{f_m} + p_c \cdot \left[ M(H,t) \cdot \frac{f(H,t)}{f_m} (1-\text{quebras}) \right] \quad (2-6)$$

As quebras podem ser expressas em função da distância e do tamanho do cromossomo:

$$\text{quebras} = \frac{\rho(H,t)}{L-1} \quad (2-7)$$

Substituindo (2-6) em (2-5) e simplificando, resulta:

$$M(H,t+1) \geq M(H,t) \cdot \frac{f(H,t)}{f_m} \left[ 1 - p_c \cdot \frac{\rho(H,t)}{L-1} \right] \quad (2-8)$$

Se uma probabilidade de mutação  $p_m$  é aplicada à geração  $t$ , a chance de um esquema permanecer inalterado após sua aplicação é [18]:

$$M(H,t+1) \geq M(H,t) \cdot \frac{f(H,t)}{f_m} \cdot \left[ 1 - p_c \cdot \frac{\rho(H,t)}{L-1} \right] \cdot [1 - O(H,t) \cdot p_m] \quad (2-9)$$

A conclusão oriunda desta equação é que esquemas com distâncias e ordem reduzidas, bem como indivíduos com aptidão maior que a aptidão média apresentam uma tendência a se proliferarem nas gerações seguintes.

Embora o Teorema do Esquema constitua um resultado matemático importante, ele não permite descrever completamente o comportamento de um GA, fornecendo apenas uma boa aproximação para a geração futura [18]. Algumas hipóteses assumidas até a obtenção da Eq. (2-8) colaboram para esta limitação: a cardinalidade igual a 2, a seleção do tipo roleta, a hipótese de só existirem quebras após o crossover, a utilização de crossover de 1 ponto, a mutação em um único bit e o fato de desconsiderar a influência que um parâmetro codificado em um dado cromossomo exerce sobre outro, fenômeno este conhecido como *epistasia* [39].

## 2.5 Plataformas de Evolução

Uma classificação bastante empregada em Computação Evolutiva refere-se ao método utilizado no cálculo da aptidão de um indivíduo. Se o cálculo da aptidão utiliza

medidas físicas realizadas em circuitos reconfiguráveis, a plataforma de evolução é dita *intrínseca*, também conhecida como Hardware Evolucionário (EHW). Em contrapartida, se o cálculo da aptidão utiliza simuladores para obter a aptidão dos indivíduos, a plataforma de evolução é dita *extrínseca*. De forma sintética, é possível afirmar que: a evolução extrínseca é realizada em *software* e a evolução intrínseca é realizada em *hardware*.

As aplicações em Hardware Evolucionário podem ser divididas em duas grandes categorias: *Hardware Adaptativo* e *Síntese de Circuitos* [6].

## 2.6 Hardware Adaptativo

Um sistema é considerado adaptativo se ele é capaz de modificar automaticamente o seu comportamento a fim de compensar mudanças ambientais ou possíveis falhas [40]. Mudanças na temperatura de operação do sistema [41-42], incidência de radiação ionizante [43] e degradação de desempenho devido ao envelhecimento dos componentes são exemplos de possíveis alterações ambientais que estão sendo alvo de estudo de alguns pesquisadores.

O projeto de sistemas tolerantes a falhas é bastante desafiador, uma vez que as modificações do sistema, para serem eficazes, devem ser executadas em um intervalo de tempo máximo. Um sistema tolerante a falhas é dito eficaz se for capaz de fornecer saídas logicamente e temporalmente corretas.

Desta forma, uma estimativa do tempo de reconfiguração do circuito desempenha um papel fundamental no projeto destes sistemas. O tempo total que o EA leva para reconfigurar  $k$  gerações de  $\lambda$  indivíduos é dado por [6]:

$$T_r(k, \lambda) = k \cdot \lambda \cdot (t_{pgm} + t_{apt}) + k \cdot t_{pro} \quad (2-10)$$

onde  $t_{pgm}$  é tempo gasto para programar a plataforma reconfigurável,  $t_{apt}$  é o tempo para calcular a aptidão de cada indivíduo e  $t_{pro}$  é o tempo de processamento do EA.

O  $t_{pgm}$  é dependente da plataforma reconfigurável utilizada. No caso de uma FPGA (*Field Programmable Gate Arrays*), a latência pode variar de alguns *ms* a centenas de *ms* dependendo do modelo. O  $t_{apt}$  é dependente da aplicação. Se por exemplo são necessárias várias conversões A/D para o cálculo da aptidão, o  $t_{apt}$  tende a ser alto. O  $t_{pro}$  é tempo que o processador gasta para executar a seleção e a variação.

## 2.7 Síntese de Circuitos

Esta categoria de Hardware Evolucionário foi propositalmente deixada para o final do capítulo por se tratar do foco principal deste trabalho. A síntese de circuitos consiste em três etapas distintas: encontrar a topologia do circuito, determinar a natureza dos componentes e definir os valores dos componentes. É conveniente distinguir estas três etapas nos métodos de síntese automática de circuitos de maneira a possibilitar que sejam independentemente processadas pelo processo evolucionário.

Dentre as três tarefas de síntese, a busca pela topologia é o problema mais complexo a ser solucionado pelo projetista. Métodos de síntese tradicionais fazem uso de um reduzido número de métodos analíticos para gerar topologias de uso geral capazes de satisfazer um grande número de problemas [44]. Uma vez que a síntese clássica utiliza um número limitado de topologias, ela se torna ineficiente para explorar as ilimitadas alternativas. Desta forma, técnicas evolutivas têm sido aplicadas para solucionar problemas de alta complexidade que estão além da habilidade humana de lidar com eles [45-47].

Dentro da síntese de circuitos dois grandes ramos podem ser destacados: analógico e digital. Os próximos dois capítulos são dedicados a analisar os principais trabalhos publicados nestes dois ramos.



### 3 Síntese Evolutiva de Filtros Digitais com Baixa Sensibilidade

#### 3.1 Introdução

Os filtros digitais podem ser classificados de acordo com o comprimento da resposta ao impulso como Resposta Finita ao Impulso (FIR) e Resposta Infinita ao Impulso (IIR). Por outro lado, dependendo do procedimento de cálculo da resposta, os filtros digitais também podem ser classificados como *recursivos* e *não-recursivos*. Filtros não-recursivos se caracterizam pelo cálculo da resposta levar em consideração as amostras atuais e anteriores da entrada, enquanto em filtros recursivos a resposta leva em consideração valores atuais e anteriores da entrada e da resposta. A dependência das saídas anteriores (recursividade) faz com que a duração da resposta ao impulso de um filtro recursivo seja infinita, mesmo na ausência da entrada. Donde é possível afirmar que todo filtro IIR é implementado por estruturas recursivas [54].

Um filtro digital é estável se os pólos da função de transferência estão localizados no interior do círculo de raio unitário. A utilização de filtros FIR em projetos de filtros digitais explora duas propriedades muito úteis desta classe de filtros: a estabilidade e a resposta de fase linear. Os pólos dos filtros FIR encontram-se na origem e a simetria da resposta ao impulso destes filtros garante a linearidade de fase. Contudo, para um mesmo número de coeficientes da função de transferência, os filtros FIR apresentam uma magnitude da resposta em frequência muito menos seletiva que os filtros IIR. Desta forma, os filtros IIR são mais comumente utilizados em aplicações que se exija alta seletividade com funções de transferência de baixa ordem.

Uma vez que filtros IIR podem se tornar instáveis, o projeto destes filtros utilizando técnicas evolutivas deve ser acompanhado de um mecanismo que restrinja o espaço de busca, garantindo a estabilidade do filtro. Um exemplo flagrante da necessidade de controle do espaço de busca é o projeto de filtros adaptativos para identificação de sistemas [55].

A síntese de filtros digitais, em particular a de filtros recursivos, não é um problema de simples solução. Em problemas com determinadas características específicas é possível matematicamente derivar a configuração ótima do filtro, mas no caso geral, não se pode afirmar o mesmo. Ao invés disso, um grande número de métodos de aproximação tem sido desenvolvidos, mas que são aplicáveis somente a uma classe particular de problemas [56].

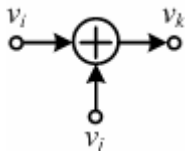

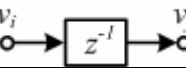
Diversas aplicações na área de projeto e síntese de filtros digitais com EA podem ser encontradas na literatura [12, 56-68]. Ao contrário dos métodos tradicionais de síntese, a síntese evolutiva possui a desejável propriedade de, com significativa facilidade, incluir a topologia do circuito como parte integrante do processo de otimização, possibilitando abordar questões complexas em se tratando de circuitos, tais como *auto-adaptação*, *auto-reparo* e *auto-reconfiguração* [40, 69].

As próximas duas seções descreverão, respectivamente, os componentes básicos de um filtro digital e as técnicas convencionais utilizadas no projeto de filtros digitais. Optou-se pela palavra projeto ao invés de síntese para estabelecer uma distinção. A síntese além de determinar os coeficientes da função de transferência inclui a topologia do filtro como parte da otimização, enquanto o projeto determina a função de transferência e utiliza topologias padrão em sua implementação.

### **3.2 Componentes de um Filtro Digital**

Um filtro digital é um circuito composto pela interconexão de três elementos básicos: somadores, atrasadores e multiplicadores. Cada elemento representa uma operação matemática nos sinais dos filtros, como definido na Tabela 3-1.

Tabela 3-1. Elementos básicos de um filtro digital.

	Símbolo Esquemático	Equação	Acrônimo
Somador		$v_k = v_i + v_j$	<i>a</i>
Multiplicador		$v_k = m_j \cdot v_i$	<i>m</i>
Atrasador		$v_j = z^{-1} \cdot v_i$	<i>d</i>

A função de transferência é um conceito fundamental empregado na construção e análise de filtros digitais lineares, conforme ilustrado na Fig. 3-1.

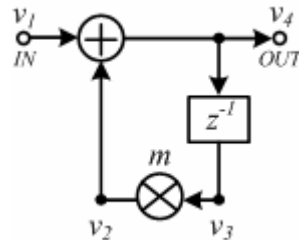


Figura 3-1. Estrutura de um filtro de primeira ordem.

O sistema de equações que descreve este filtro é dado por:

$$\begin{cases} v_2 = m \cdot v_3 \\ v_3 = z^{-1} \cdot v_4 \\ v_4 = v_1 + v_2 \end{cases} \quad (3-1)$$

A função de transferência,  $H(z)$ , é obtida resolvendo o sistema de equações:

$$H(z) = \frac{v_4(z)}{v_1(z)} = \frac{z}{z - m} \quad (3-2)$$

### 3.3 Projeto de Filtros Digitais Recursivos

Os métodos de projetos de filtros digitais recursivos são mais bem compreendidos se divididos em duas categorias distintas.

#### 3.3.1 Filtros Digitais Recursivos usando Aproximações

O emprego de aproximações matemáticas no projeto de filtros analógicos é objeto de estudo dos pesquisadores a mais de cinquenta anos. As aproximações mais

comumente utilizadas são: *Butterworth*, *Chebyshev*, *Chebyshev Inversa*, *Elíptica* e *Bessel* [70].

O projeto de filtros digitais recursivos usando aproximações pode ser dividido em duas etapas. A primeira consiste em encontrar a função de transferência do filtro contínuo no tempo que satisfaz às especificações de projeto. A segunda consiste em utilizar uma transformação que converte a função de transferência do filtro contínuo no tempo na função de transferência do filtro discreto no tempo. Existem duas transformações comumente utilizadas no projeto de filtros digitais recursivos: *transformação da resposta invariante ao impulso* e *transformação bilinear*.

### 3.3.1.1 Transformação da Resposta Invariante ao Impulso

Seja  $h_a(t)$  a resposta ao impulso do filtro analógico com função de transferência  $H_a(s)$  e defina a transformação de tempo contínuo em tempo discreto como:

$$h(n) = h_a(nT) \quad (3-3)$$

Isto significa que as amostras da resposta ao impulso do filtro digital são exatamente iguais às amostras da resposta ao impulso do protótipo analógico. O procedimento de obtenção da resposta em frequência do filtro digital é derivado seguindo os seguintes passos [54]:

1. Cálculo de  $h_a(t)$ ;
2. Substituição de  $t$  por  $nT$  em  $h_a(t)$ ;
3. Cálculo da transformada  $z$  de  $h_a(nT)$ .

Existe um problema com este procedimento, a resposta em frequência do filtro digital não é idêntica à resposta em frequência do filtro analógico conforme discutido a seguir.

A resposta ao impulso no tempo contínuo é amostrada para produzir a resposta ao impulso no tempo discreto conforme ilustrado na Fig. 3-2.

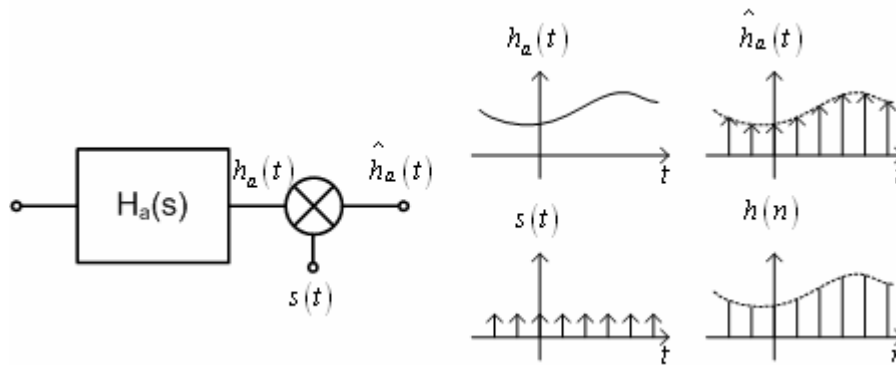


Figura 3-2. Filtro analógico amostrado e sinais envolvidos.

Considerando um amostrador ideal o sinal  $s(t)$  é um trem de impulsos e pode ser escrito na forma:

$$s(t) = \sum_{n=-\infty}^{\infty} \delta(t - nT) \quad (3-4)$$

onde  $T$  é o período de amostragem.

O sinal de saída do amostrador é dado por:

$$\hat{h}_a(t) = h_a(t) \cdot s(t) = h_a(t) \cdot \sum_{n=-\infty}^{\infty} \delta(t - nT) \quad (3-5)$$

Utilizando as propriedades da “função” impulso e considerando o sistema causal, ou seja,  $h_a(t) = 0$  para  $t \leq 0^-$  é possível reescrever (3-5) na forma:

$$\hat{h}_a(t) = \sum_{n=0}^{\infty} h_a(nT) \cdot \delta(t - nT) \quad (3-6)$$

Portanto, a transformada de Fourier de  $\hat{h}_a(t)$  é:

$$\begin{aligned} \hat{H}_a(j\omega) &= \sum_{n=0}^{\infty} h_a(nT) \cdot F\{\delta(t - nT)\} \\ &= \sum_{n=0}^{\infty} h_a(nT) \cdot e^{-j\omega nT} = H(z) \Big|_{z=e^{j\omega T}} \end{aligned} \quad (3-7)$$

onde  $H(z)$  é a transformada  $z$  do sinal discreto no tempo  $h_a(nT)$ .

A fórmula de Poisson estabelece que:

$$\sum_{n=0}^{\infty} x(nT) = \frac{x(0^+)}{2} + \frac{1}{T} \sum_{n=-\infty}^{\infty} X(jn\omega_s) \quad (3-8)$$

onde  $\omega_s = 2\pi/T$  é a frequência de amostragem.

Aplicando a fórmula de Poisson em (3-7) é possível escrever:

$$\hat{H}_a(j\omega) = H(z)\Big|_{z=e^{j\omega T}} = \frac{h(0^+)}{2} + \frac{1}{T} \sum_{n=-\infty}^{\infty} H_a(j\omega + jn\omega_s) \quad (3-9)$$

onde  $h(0^+) = \lim_{s \rightarrow \infty} [s \cdot H_a(s)]$  para  $s = j\omega$ . Esta equação mostra o mapeamento da resposta em frequência contínua no tempo  $H_a(j\omega)$  na resposta em frequência discreta no tempo  $H(z)\Big|_{z=e^{j\omega T}}$ . De (3-9) é possível concluir que  $H(z)\Big|_{z=e^{j\omega T}}$  sofre o fenômeno chamado de *aliasing*. A Fig. 3-3 mostra um exemplo de *aliasing* para um filtro passa-baixas.

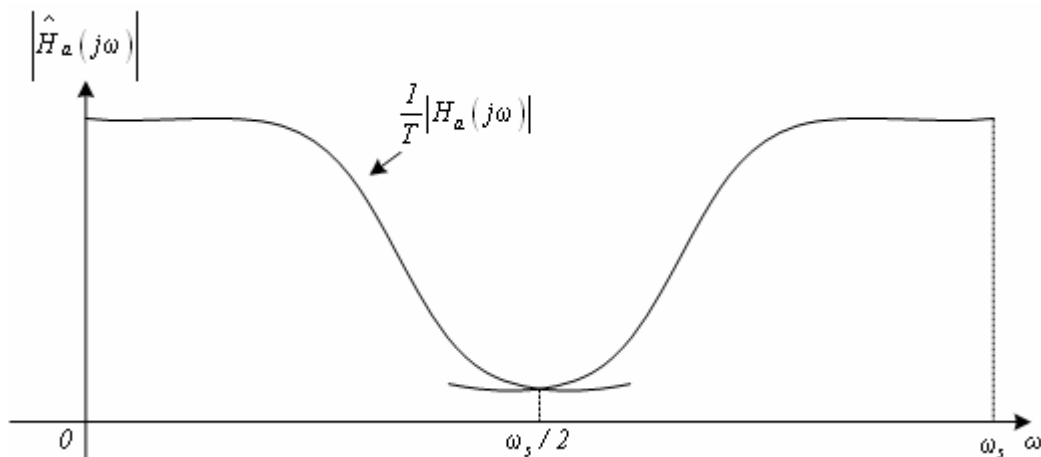


Figura 3-3. Magnitude da resposta resultante da transformação, mostrando *aliasing*.

O *aliasing* seria evitado se e somente se:

$$H(j\omega) = 0 \quad \text{para} \quad |\omega| \geq \frac{\omega_s}{2} \quad (3-10a)$$

e com:

$$h(0^+) = 0 \quad (3-10b)$$

Assumindo (3-10), é possível simplificar (3-9), resultando em uma resposta sem *aliasing*:

$$\hat{H}(j\omega) = H(z) = \frac{1}{T} H(j\omega) \quad \text{para} \quad |\omega| < \frac{\omega_s}{2} \quad (3-11)$$

Contudo, (3-10a) não é fisicamente realizável, ou seja, nenhum filtro analógico é perfeitamente limitado em banda.

O método da resposta invariante ao impulso apresenta resultados satisfatórios para filtros passa-baixas e passa-faixa utilizando aproximações do tipo Butterworth, Chebyshev e Bessel, para os quais (3-10) é aproximadamente atendida.

### 3.3.1.2 Transformação Bilinear

Para evitar *aliasing* da resposta em frequência, é necessário utilizar uma transformação que mapeia um-para-um o plano-s no plano-z. A transformação  $z = e^{sT}$ , empregada na resposta invariante ao impulso, mapeia muitos-para-um conforme ilustra a Fig. 3-4.

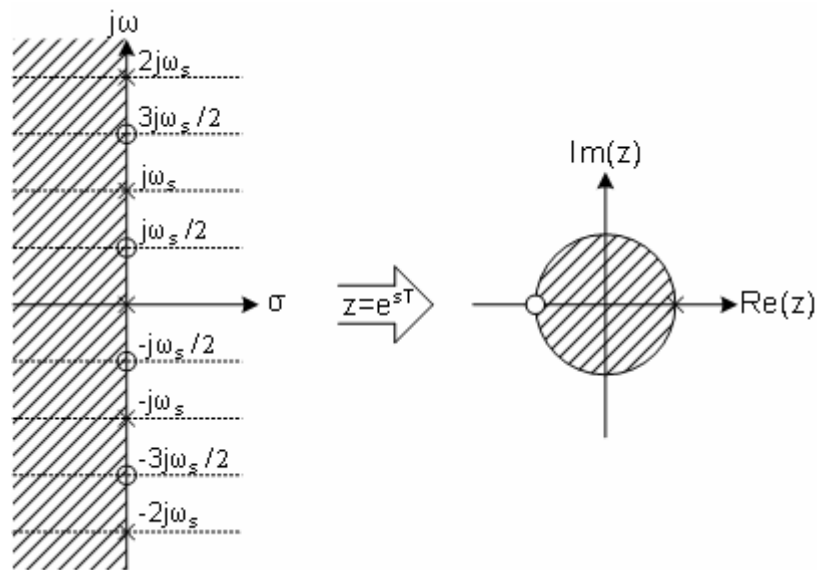


Figura 3-4. Mapeamento do plano-z no plano-s quando  $z = e^{sT}$ .

A transformação bilinear emprega uma transformação de  $s$  para  $s'$ , que comprime o plano  $s$  na faixa de  $-\pi/T \leq \text{Im}(s') \leq \pi/T$ , conforme mostrado na Fig. 3-5.

Desta forma,  $z$  é transformado via  $z = e^{s'T}$  sem que haja aliasing.

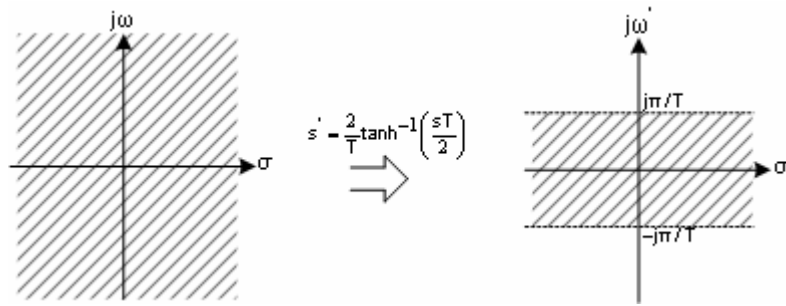


Figura 3-5. Mapeamento de  $s$  em  $s'$  para evitar aliasing.

A transformação de  $s$  em  $z$  é obtida utilizando a função tangente hiperbólica:

$$s = \frac{2}{T} \tanh\left(\frac{s'T}{2}\right) \quad (3-12)$$

Substituindo  $s' = (1/T) \cdot \ln z$  em (3-12) é possível encontrar a equação da transformação bilinear:

$$s = \frac{2}{T} \left( \frac{1 - z^{-1}}{1 + z^{-1}} \right) \quad (3-13)$$

Portanto, o projeto do filtro discreto no tempo é obtido do projeto do filtro contínuo no tempo via:

$$H(z) = H_a(s) \Big|_{s=(2/T)(1-z^{-1})/(1+z^{-1})} \quad (3-14)$$

A relação não-linear de entre  $s$  e  $s'$  cria uma distorção de frequência que deve ser corrigida no projeto, porém as características de  $H_a(s)$  são preservadas em  $H_a(s')$ . Em particular, as características na banda passante e na banda de rejeição são mantidas. Esta propriedade faz da transformação linear a mais empregada no projeto de filtros recursivos usando aproximações.

### 3.3.2 Filtros Digitais Recursivos usando Métodos de Otimização

O projeto de filtros digitais utilizando aproximações é bastante eficiente e leva a uma solução fechada da função de transferência. A principal desvantagem deste método é que ele não é aplicável a filtros com resposta arbitrária, limitando-se a filtros



com resposta cuja banda passante e a banda de rejeição são constantes dentro das tolerâncias das especificações.

O projeto de filtros digitais usando métodos de otimização leva a soluções melhores do que as obtidas usando aproximações, além de serem capazes de atender a especificações arbitrárias de amplitude e fase [54]. Uma importante classe de algoritmos de otimização que tem sido eficientemente aplicada no projeto de filtros digitais é a classe dos algoritmos *quasi-Newton*.

Considere que a magnitude da resposta em frequência  $\left|H(e^{j\omega T})\right|$  de um filtro digital recursivo de ordem  $N$  deve atender às especificações de uma dada resposta arbitrária de magnitude  $M_o(\omega)$ . A função de transferência deste filtro pode ser expandida em seções de segunda ordem na forma:

$$H(z) = H_0 \prod_{j=1}^J \frac{a_{0j} + a_{1j}z + z^2}{b_{0j} + b_{1j}z + z^2} \quad \text{para } N \text{ par} \quad (3-15)$$

onde  $a_{ij}$  e  $b_{ij}$  são coeficientes reais,  $H_0$  é uma constante positiva e  $N$  é a ordem do filtro. A magnitude da resposta do filtro pode ser expressa como:

$$M(\mathbf{x}, \omega) = \left|H(e^{j\omega T})\right| \quad (3-16)$$

onde  $\mathbf{x} = [a_{01} \ a_{11} \ b_{01} \ b_{11} \ \dots \ b_{1J} \ H_0]^t$  é um vetor coluna de  $4J + 1$  posições.

O erro pode ser definido como:

$$e(\mathbf{x}, \omega) = M(\mathbf{x}, \omega) - M_o(\omega) \quad (3-17)$$

Amostrando  $e(\mathbf{x}, \omega)$  em  $K$  frequências  $\omega_1, \omega_2, \dots, \omega_K$ , tem-se:

$$e_i(\mathbf{x}) = e(\mathbf{x}, \omega_i) \quad (3-18)$$

Desta forma, é possível definir uma função objetivo em termos da norma Euclidiana ( $L_2$ ) como:

$$\Psi(\mathbf{x}) = \left[ \sum_{i=1}^K |e_i(\mathbf{x})|^2 \right]^{1/2} \quad (3-19)$$

Se o vetor coluna  $\delta = [\delta_1 \ \delta_2 \ \dots \ \delta_j \ \delta_{4j+1}]^t$  é o acréscimo a ser adicionado ao vetor  $x$  de maneira a fazer  $\partial \Psi(x + \delta) / \partial \delta_p = 0$ , então:

$$\delta = -\mathbf{H}^{-1} \cdot \nabla \Psi(x) \quad (3-20)$$

onde  $\nabla \Psi(x)$  é o vetor *gradiente* e  $\mathbf{H}$  é a *matriz Hessiana*, definidas, respectivamente, como:

$$\nabla \Psi(x) = \left[ \frac{\partial \Psi(x)}{\partial a_{01}} \quad \frac{\partial \Psi(x)}{\partial a_{11}} \quad \dots \quad \frac{\partial \Psi(x)}{\partial H_0} \right]^t \quad (3-21a)$$

$$\mathbf{H} = \begin{pmatrix} \frac{\partial^2 \Psi(x)}{\partial^2 a_{01}} & \dots & \frac{\partial^2 \Psi(x)}{\partial a_{01} \partial H_0} \\ \vdots & \ddots & \vdots \\ \frac{\partial^2 \Psi(x)}{\partial H_0 \partial a_{01}} & \dots & \frac{\partial^2 \Psi(x)}{\partial^2 H_0} \end{pmatrix} \quad (3-21b)$$

O algoritmo que visa a minimizar a função o erro  $\Psi(x)$  é conhecido como *algoritmo de Newton* e está descrito abaixo [54]:

*Algoritmo 1: Algoritmo de Newton*

- S1. Inicialização das iterações  $k=0$ , escolha do ponto inicial  $x_0$  e da tolerância  $\varepsilon$ .
- S2. Cálculo do gradiente  $\nabla \Psi_k(x)$  e da matriz Hessiana  $\mathbf{H}$ .
- S3. Cálculo de  $\mathbf{H}^{-1}$  e  $d_k = -\mathbf{H}_k^{-1} \cdot \nabla \Psi_k(x)$ , onde o vetor  $d_k$  é chamado de *direção de Newton* ou de *direção descendente*.
- S4. Cálculo de  $\alpha_k$ , valor de  $\alpha$  que minimiza  $\Psi(x_k + \alpha \cdot d_k)$ , usando busca linear.
- S5. Cálculo de  $x_{k+1} = x_k + \delta_k$ , onde  $\delta_k = \alpha_k \cdot d_k$  e  $\Psi_{k+1} = \Psi(x_{k+1})$ .
- S6. Verificação se  $|\alpha_k \cdot d_k|^2 < \varepsilon$ , então  $x_{k+1}$  é o mínimo, caso contrário, volta para S2.

Implementações mais eficientes deste algoritmo são sugeridas na literatura [71]. Muito embora estes métodos de otimização sejam, em geral, muito rápidos e eficientes, eles dependem da continuidade e da diferenciabilidade da função objetivo. A qualidade das soluções obtidas depende dos pontos iniciais do algoritmo, levando via de regra a uma solução sub-ótima dada à dificuldade inerente destes métodos de descartar mínimos locais em favor de soluções melhores. Além disso, quando usados em uma nova categoria de problemas é necessária a inclusão de restrições ao espaço de busca tornando o método aplicável apenas àquela classe de problemas [72]. Por estas razões, alguns métodos de otimização global têm sido aplicados ao projeto de filtros digitais em especial empregando técnicas evolutivas.

### **3.4 Projeto e Síntese Evolutiva de Filtros Digitais**

A abordagem evolutiva é capaz de explorar uma vasta gama de alternativas de projeto que os métodos iterativos simples, analíticos e baseados em aproximações não conseguem. Técnicas evolutivas são bastante flexíveis a problemas com múltiplos objetivos podendo lidar com qualquer tipo de função objetivo seja ela descontínua e/ou não-diferenciável [56].

A primeira notícia do uso de GA no projeto de filtros digitais data do início dos anos 80 quando foi aplicada a filtros adaptativos IIR [73]. Abordagens mais recentes para a mesma aplicação utilizando técnicas evolutivas podem ser encontradas na literatura [54, 65, 74-75]. Uma extensa lista de publicações contendo outras aplicações de EA no projeto de filtros digitais lineares também podem ser citadas [47, 57, 59-60, 62-64, 76-77], porém todas comungam da mesma idéia essencial que é a otimização dos coeficientes da função de transferência dos filtros.

Em [78] um EA utilizando uma Função Booleana Positiva representada por somas de produto foi usado no projeto de um filtro não-linear para supressão de ruído em imagens digitalizadas.

Uma abordagem evolutiva que transcende a simples otimização de coeficientes está associada ao projeto de filtros digitais sem o uso de multiplicadores [58, 79-80]. A idéia é encontrar uma função de transferência que atenda às especificações do projeto e cujos coeficientes são representados por potências de 2. Desta forma, é possível substituir os multiplicadores por somadores e registradores de deslocamento, conforme ilustrado no exemplo abaixo:

Multiplicação Real	Multiplicação com Potência de 2:
$\begin{array}{r} 17 \\ \times 5 \\ \hline 85 \end{array}$	$\begin{array}{r} 13 \\ \times 8 \\ \hline 104 \end{array}$
$\begin{array}{r} 00010001 \\ \times 00000101 \\ \hline 00010001 \\ + 00000000 \\ \hline 00010001 \\ \hline 00010011001 = 85_{10} \end{array}$	$\begin{array}{r} 00001101 \\ \Downarrow \text{Desloca 3} \\ \text{para} \\ \text{Esquerda} \\ \hline 1101000 = 104_{10} \end{array}$

Figura 3-6. Exemplo da substituição de multiplicadores por registradores de deslocamento em filtros digitais.

A síntese evolutiva de filtros digitais na qual a topologia faz parte do processo de otimização foi inicialmente explorada por Miller em [66]. Neste trabalho o autor investiga com sucesso a possibilidade de se sintetizar filtros digitais usando uma representação em nível de *portas*. Uma arquitetura reconfigurável para síntese intrínseca de filtros FIR de ordem elevada ( $\geq 100$ ) usando GP pode ser encontrada em [81].

### 3.4.1 Filtros Digitais com Baixa Sensibilidade

Outro aspecto importante que depende fundamentalmente da topologia na síntese de filtros é o estudo da sensibilidade. É comum encontrar especificações de magnitude e fase bastante restritivas. Devido à quantização dos coeficientes desvios intoleráveis da função de transferência podem ocorrer. Embora de considerável

importância, o problema da sensibilidade na síntese evolutiva de filtros digitais é abordado em um reduzido número de trabalhos [67-68].

A síntese de filtros digitais com baixa sensibilidade aos coeficientes é de grande interesse prático desde que qualquer que seja a implementação de um filtro digital, esta envolve operações aritméticas de palavras de tamanho finito e erros de quantização. Contudo, a modelagem desta classe de filtros não é uma tarefa simples desde que não existem métodos analíticos gerais para derivar a configuração ótima do filtro [56].

Existem diversas estruturas equivalentes de filtros digitais capazes de realizar a mesma função de transferência. As estruturas podem ser obtidas por diferentes métodos, variando significativamente com relação à complexidade e ao número de elementos. Uma dada estrutura pode requerer um grande número de multiplicadores e ser relativamente insensível aos erros de quantização, enquanto uma outra estrutura com menos elementos pode gerar oscilações parasitas quando os sinais são quantizados [54]. Portanto, a sensibilidade de um filtro digital está relacionada com a topologia do mesmo e é esperado que métodos de síntese evolutiva que são capazes de explorar grandes espaços de busca possam encontrar melhores soluções do que abordagens determinísticas [82-85] que estão restritas a um conjunto limitado de topologias de filtros.

Embora uma dada função de transferência possa ser realizada por um número ilimitado de estruturas, em geral, somente algumas são suficientemente insensíveis aos erros de quantização. Para ilustrar os efeitos de quantização na função de transferência, considere o filtro da Fig. 3-1.

Uma implementação prática da função de transferência deste filtro requer que os coeficientes sejam quantizados devido ao número finito de bits dos registradores. Após a quantização a função de transferência é expressa como:

$$H_Q(z) = \frac{z}{z - [m]_Q} \quad (3-22)$$

onde  $H_Q(z)$  e  $[m]_Q$  são as versões quantizadas de  $H(z)$  e do coeficiente  $m$ , respectivamente.

Após a quantização, a resposta em frequência do filtro difere da resposta ideal e os pólos e zeros da função de transferência são deslocados das suas posições originais. Se o deslocamento dos pólos for grande o suficiente para mandá-los para fora do círculo unitário no plano  $z$ , o filtro quantizado será instável. Contudo, é possível minimizar estes efeitos sintetizando estruturas inerentemente menos sensíveis a quantização de coeficientes.

As próximas seções deste capítulo destinam-se a discutir um esquema de codificação cromossomial baseado na matriz adjacência [86] para ser utilizado na síntese evolutiva de filtros digitais de baixa sensibilidade. É mostrado que, para uma mesma finalidade, a representação por matriz adjacência é mais flexível que a representação baseada em árvores da GP [67-68]. A representação utilizando matriz adjacência é capaz de mapear um maior número de topologias do que aquelas que podem ser mapeadas por representações baseadas em árvores. Além disso, a detecção de filtros não-computáveis, que afetam a eficiência do método evolutivo, é facilmente implementada com o emprego da representação proposta, requerendo reduzido esforço computacional.

### 3.4.2 Matriz Adjacência

Seja  $G(V, E)$  um grafo orientado sem ramos em paralelo e  $n$  vértices ordenados de  $v_1$  a  $v_n$ , onde  $V$  e  $E$  denotam, respectivamente, o conjunto dos vértices e dos ramos do grafo. A matriz adjacência  $A = [a_{ij}]$  do grafo orientado  $G$  é uma matriz  $n \times n$  com  $a_{ij}$  definido como [88]:

$$a_{ij} = \begin{cases} 1, & \text{se } (v_i, v_j) \in E \\ 0, & \text{outros casos} \end{cases} \quad (3-23)$$

Na Fig. 3-7(a) é mostrada a matriz adjacência  $A$  correspondente ao grafo  $G$  da Fig. 3-7(b). Nesta matriz os elementos não-nulos da diagonal principal representam curto-circuitos como, por exemplo, o elemento  $a_{33}$  da matriz e sua correspondente representação no grafo  $e_3$ .

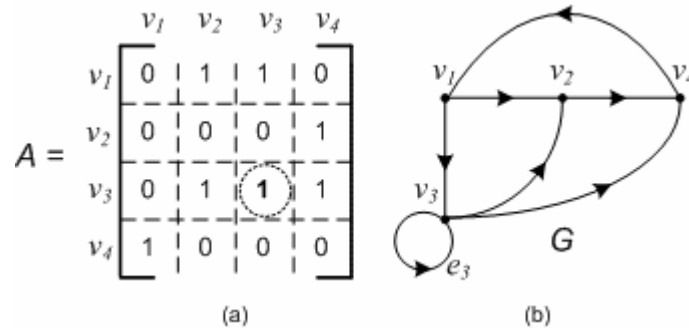


Figure 3-7. (a) Matriz adjacência (b) grafo orientado  $G$ .

Como discutido em [86], existem dois problemas associados com a representação da matriz adjacência: o fato da definição da matriz adjacência assumir que o grafo orientado não possui ramos em paralelo e sua inadequação para representar componentes de três terminais. Desde que não existem ramos em paralelo em estruturas de filtros digitais, somente o segundo problema deve ser analisado na representação de filtros digitais usando matriz adjacência.

### 3.4.3 Representação de Filtros Digitais Usando Matriz Adjacência

As matrizes adjacência correspondentes aos três elementos básicos de um filtro digital são mostradas na Tabela 3-2.

Tabela 3-2. Representação dos Elementos Básicos de um Filtro Digital usando Matriz Adjacência.

	Grafo Orientado	Matriz Adjacência
Somador		$\begin{matrix} & v_1 & v_2 & v_3 \\ v_1 & 0 & 0 & 1 \\ v_2 & 0 & 0 & 1 \\ v_3 & 0 & 0 & 0 \end{matrix}$
Multiplicador		$\begin{matrix} & v_1 & v_2 \\ v_1 & 0 & 2 \\ v_2 & 0 & 0 \end{matrix}$
Atrasador		$\begin{matrix} & v_1 & v_2 \\ v_1 & 0 & 5 \\ v_2 & 0 & 0 \end{matrix}$

Como mostrado na tabela, a solução utilizada para representar os três terminais do somador foi dividi-lo em dois ramos orientados. Desta forma, somente os somadores serão representados com mais de um elemento numa mesma coluna da matriz adjacência. As colunas da matriz que representam o multiplicador e o atrasador terão apenas um elemento. Nas definições acima, outros números além de 1 devem ser usados para distinguir o tipo do elemento correspondente a um ramo.

A Fig. 3-8 ilustra um exemplo de representação de um filtro digital de primeira ordem realizado em forma direta usando a matriz adjacência. Na Fig. 3-8(a) os números indicam os nós do filtro e os acrônimos  $a$ ,  $m$  e  $d$  representam, respectivamente, somadores, multiplicadores e atrasadores. Seguindo a notação na Tabela 3-2, na Fig. 3-8(b) o filtro digital é representado por seu grafo orientado. Na Fig. 3-8(c) a representação da matriz adjacência usando os acrônimos é mostrada. Finalmente, na Fig. 3-8(d) os acrônimos são substituídos por números. O nó mais baixo é sempre o nó de entrada e o nó mais alto é sempre o nó de saída. A primeira coluna e a última linha da matriz adjacência estão vazias, pois, como mostrado na Fig. 3-8(b), não existe nenhum ramo incidente no nó de entrada e nenhum ramo deixando o nó de saída.



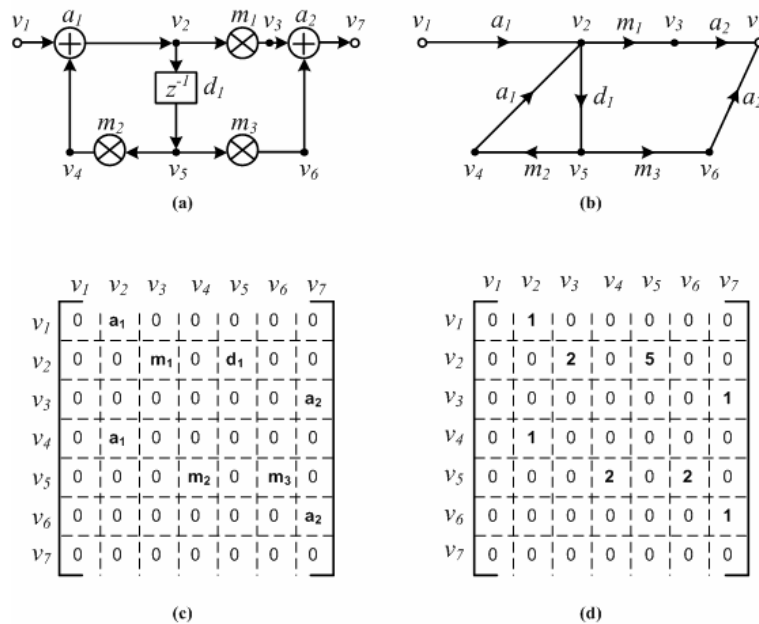


Figura 3-8. Representação usando matriz adjacência de um filtro digital de primeira ordem (a) símbolo esquemático (b) grafo orientado (c) matriz adjacência com acrônimos (d) matriz adjacência com os acrônimos substituídos por números.

### 3.4.4 Indivíduos Inválidos Usando Matriz Adjacência

Uma abordagem evolutiva capaz de sintetizar filtros digitais com baixa sensibilidade é uma alternativa bastante razoável para solucionar o problema do número restrito de estruturas que podem ser sintetizadas por métodos determinísticos. No entanto, qualquer que seja o EA utilizado, a eficiência do algoritmo está associada a esquemas adequados de representação que sejam capazes de evitar a geração de indivíduos anômalos durante o processo evolucionário [86-87].

Em um processo evolutivo, indivíduos inválidos podem ser gerados na população inicial ou após a aplicação dos operadores genéticos. Eles podem ser classificados em duas categorias: *indivíduos estruturalmente inválidos* e *indivíduos não-computáveis*.

Desde que o cálculo da aptidão é a tarefa de maior esforço computacional, a eficiência do algoritmo pode ser consideravelmente aumentada se regras para evitar a avaliação de indivíduos inválidos forem incluídas [87]. Isto é particularmente importante na síntese evolutiva no domínio digital, porque os componentes de um filtro digital são dependentes da direção do fluxo de sinal. Conseqüentemente, as restrições

topológicas em um filtro digital são particularmente mais severas do que as de um filtro analógico.

As possíveis topologias estruturalmente inválidas que podem ocorrer durante a síntese evolutiva de filtros digitais são listadas na Tabela 3-3. A primeira estrutura possui um atrasador e um multiplicador ambos incidindo no mesmo nó  $k$ . A segunda anomalia estrutural é um curto-circuito em algum nó  $j$ , indicado por um elemento diferente de zero na correspondente posição da diagonal principal da matriz adjacência. Finalmente, a terceira anomalia estrutural é representada por um nó isolado, tal como o nó  $j$  na tabela, que aparece como uma linha vazia na matriz adjacência.

Tabela 3-3. Indivíduos estruturalmente inválidos.

Regra	Estrutura Inválida	Matriz Adjacência																
1		<table border="1" style="border-collapse: collapse; text-align: center;"> <tr> <td></td> <td><math>v_i</math></td> <td><math>v_j</math></td> <td><math>v_k</math></td> </tr> <tr> <td><math>v_i</math></td> <td>0</td> <td>0</td> <td>5</td> </tr> <tr> <td><math>v_j</math></td> <td>0</td> <td>0</td> <td>2</td> </tr> <tr> <td><math>v_k</math></td> <td>0</td> <td>0</td> <td>0</td> </tr> </table>		$v_i$	$v_j$	$v_k$	$v_i$	0	0	5	$v_j$	0	0	2	$v_k$	0	0	0
	$v_i$	$v_j$	$v_k$															
$v_i$	0	0	5															
$v_j$	0	0	2															
$v_k$	0	0	0															
2		<table border="1" style="border-collapse: collapse; text-align: center;"> <tr> <td></td> <td><math>v_i</math></td> <td><math>v_j</math></td> </tr> <tr> <td><math>v_i</math></td> <td>0</td> <td>1</td> </tr> <tr> <td><math>v_j</math></td> <td>0</td> <td>1</td> </tr> </table>		$v_i$	$v_j$	$v_i$	0	1	$v_j$	0	1							
	$v_i$	$v_j$																
$v_i$	0	1																
$v_j$	0	1																
3		<table border="1" style="border-collapse: collapse; text-align: center;"> <tr> <td></td> <td><math>v_i</math></td> <td><math>v_j</math></td> <td><math>v_k</math></td> </tr> <tr> <td><math>v_i</math></td> <td>0</td> <td>2</td> <td>5</td> </tr> <tr> <td><math>v_j</math></td> <td>0</td> <td>0</td> <td>0</td> </tr> <tr> <td><math>v_k</math></td> <td>5</td> <td>0</td> <td>0</td> </tr> </table>		$v_i$	$v_j$	$v_k$	$v_i$	0	2	5	$v_j$	0	0	0	$v_k$	5	0	0
	$v_i$	$v_j$	$v_k$															
$v_i$	0	2	5															
$v_j$	0	0	0															
$v_k$	5	0	0															

Para evitar a geração de indivíduos inválidos na população inicial e durante a execução da síntese evolutiva, é necessária a aplicação do seguinte conjunto de regras:

*Regra 1:* Cada coluna da matriz adjacência, exceto a primeira, deve conter somente um tipo de elemento. Com a escolha dos números correspondentes a cada tipo de elemento – 1 para somador, 2 para multiplicador e 5 para atrasador – esta

regra é facilmente verificada, somando os elementos de cada coluna da matriz. Qualquer coluna tendo uma soma diferente de 2 ou 5 resultará em um indivíduo inválido;

*Regra 2:* Desde que os filtros digitais não possuem nó de terra como é o caso dos filtros analógicos, os elementos da diagonal principal da matriz adjacência devem ser zero.

*Regra 3:* Exceto a última linha, a matriz adjacência não pode conter linhas vazias. Portanto, cada nó, a exceção do de saída, deve ter pelo menos um ramo incidente.

### 3.4.5 Filtros Digitais Computáveis

Um filtro digital é dito computável se ele não contém ciclos livre de retardos (*delay-free loops*). Um exemplo simples de uma estrutura com ciclo livre de retardo é mostrada na Fig. 3-9.

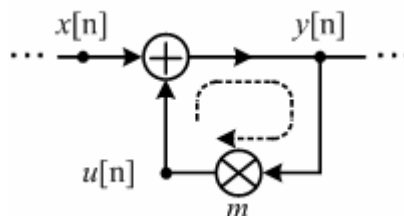


Fig. 3-9. Exemplo de um filtro digital contendo ciclo livre de retardo.

O sistema de equações derivado desta estrutura é:

$$\begin{cases} y[n] = x[n] + u[n] \\ u[n] = m \cdot y[n] \end{cases} \quad (3-24)$$

A combinação destas duas equações resultará em um filtro não-computável como pode ser verificado pela equação abaixo:

$$y[n] = x[n] + m \cdot y[n] \quad (3-25)$$

onde o valor atual da saída  $y[n]$  depende dele próprio para ser calculado. Isto é fisicamente impossível devido ao fato que qualquer processador requer um tempo finito diferente de zero para realizar operações aritméticas.

Para testar a *computabilidade* de um indivíduo usando a representação com matriz adjacência, é necessário introduzir o conceito de *matriz de caminhos* que é derivada da matriz adjacência.

Seja  $G(V, E)$  um grafo orientado com  $n$  vértices  $V = \{v_1, \dots, v_n\}$ . A matriz de caminhos de  $G$  é uma matriz  $n \times n$   $P = [p_{ij}]$  dada por:

$$p_{ij} = \begin{cases} 1, & \text{se existe um caminho orientado de } v_i \text{ para } v_j \\ 0, & \text{outros casos} \end{cases} \quad (3-26)$$

A matriz de caminhos pode ser obtida a partir da matriz adjacência pela aplicação do *algoritmo de Warshall* [89]. Este algoritmo avalia todos os possíveis caminhos através do grafo entre cada par de vértices. A implementação do algoritmo é imediata, mas sua complexidade é  $O(n^3)$ . Contudo, este fato não representa um inconveniente grave, se comparado com o esforço computacional do algoritmo evolutivo e do reduzido número de nós normalmente encontrado na síntese de filtros digitais.

Seja a matriz adjacência  $A$  de um grafo orientado  $G$ , a correspondente matriz de caminhos  $P = [p_{ij}]$  é obtida aplicando o algoritmo abaixo:

*Algoritmo 2 (Warshall): Geração da Matriz de Caminhos*

```

P = A
for i = 1, 2, ..., n
  for j = 1, 2, ..., n
    if pij = 1 then
      for k = 1, 2, ..., n
        pjk = pjk OR pik
      end
    end
  end
end

```

Na Fig.3-10(a) é mostrada a matriz de caminhos  $P$  do grafo  $G$  da Fig. 3-10(b). Como pode ser verificado na figura, se  $p_{ij} = 1$  e  $i \neq j$  existe um caminho entre os vértices  $v_i$  e  $v_j$ , caso contrário, se  $i = j$  existe um ciclo. Portanto, um elemento não-nulo na diagonal principal da matriz de caminhos  $P$  indica a existência de ciclos na estrutura do filtro. Como conseqüência, se os atrasos são eliminados do filtro digital e a correspondente matriz de caminhos para a nova matriz adjacência é obtida, os caminhos livres de retardo na estrutura serão indicados por seus elementos não-nulos na diagonal principal.

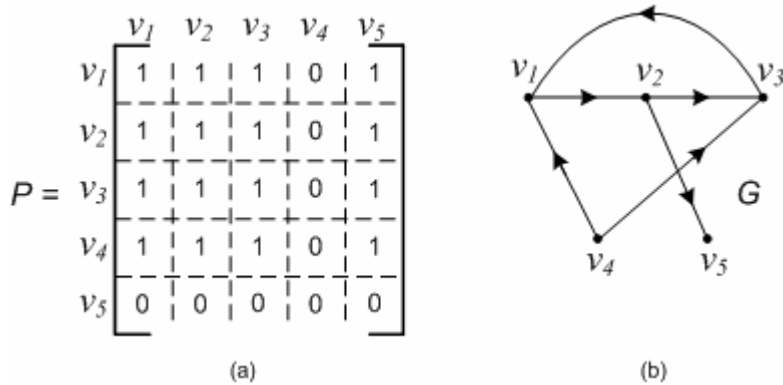


Figura 3-10. (a) Matriz de caminhos (b) grafo orientado correspondente.

Como conseqüência do uso da matriz adjacência, a computabilidade de um filtro digitais pode ser determinada pela aplicação do seguinte algoritmo:

*Algoritmo 3: Verificação de Computabilidade*

- S1. Gerar a matriz adjacência seguindo as regras da seção 3.4.4;
- S2. Retirar os retardos da matriz adjacência, substituindo os elementos da matriz adjacência referentes aos retardos por zero;
- S3. Obter a matriz de caminhos  $P$  através da aplicação do algoritmo de Warshall;
- S4. Se todos os elementos da diagonal principal da matriz de caminhos  $P$  forem zero, o filtro é computável, caso contrário ele é não-computável e considerado um indivíduo inválido.

A Fig. 3-11 ilustra os passos do algoritmo anterior para verificar a computabilidade de dois filtros digitais de primeira ordem. O primeiro exemplo é uma estrutura computável como pode ser verificado nos elementos da diagonal principal da matriz de caminhos  $P$  dada na Fig. 3-11(d). O segundo exemplo é um filtro não-computável com um ciclo livre de retardo indicado pela linha pontilhada na Fig. 3-11(e). O ciclo livre de retardo contém os nós 2, 4 e 5 como indicado pelos elementos não-nulos da diagonal principal da matriz de caminhos em Fig. 3-11(h).

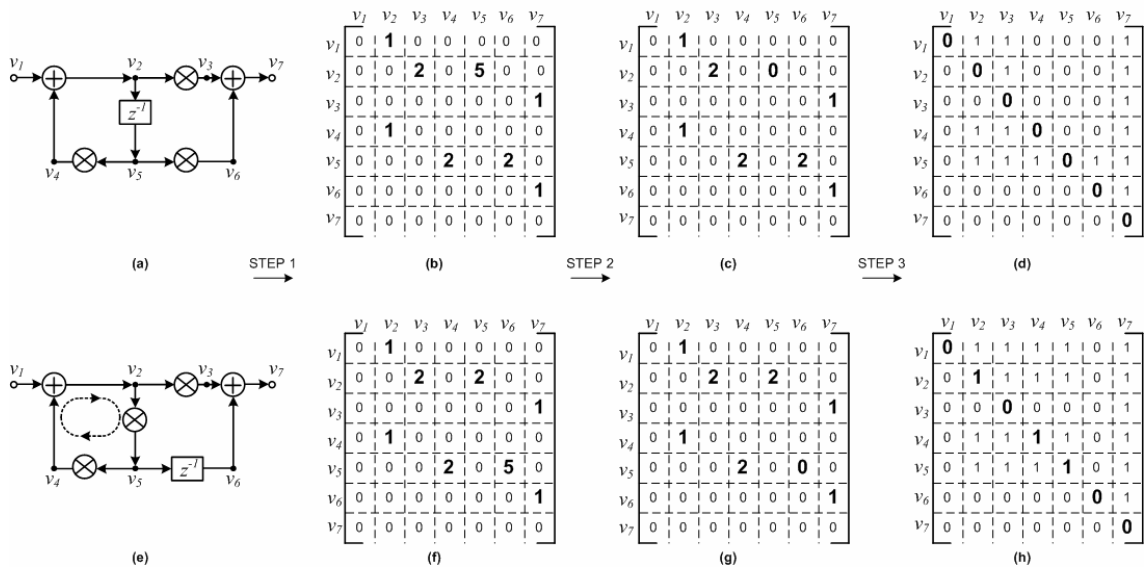


Figura 3-11. Verificação da computabilidade de dois filtros de primeira ordem diferentes (a, e) estrutura de dois filtros digitais de primeira ordem (b, f) matrizes adjacência correspondentes, (c, g) matrizes adjacência com os retardos retirados e (d, h) matriz de caminhos de dois filtros.

### 3.4.6 Operadores Genéticos usando Matriz Adjacência

A estratégia de crossover é ilustrada na Fig. 3-12. Ela consiste em trocar duas submatrizes com dimensões aleatoriamente escolhidas. Considere duas matrizes de dimensões  $m$  e  $n$ . Se  $m < n$ , as coordenadas do ponto de crossover  $(i, j)$  são escolhidas na matriz de menor dimensão, onde  $i$  e  $j$  são inteiros dentro do intervalo:

$$i \in [0, m - 1] \quad (3-27a)$$

$$j \in [0, m - 1] \quad (3-27b)$$

A fim de definir as dimensões das submatrizes a serem trocadas no crossover, dois inteiros  $p$  e  $q$  são aleatoriamente escolhidos nos intervalos:

$$p \in [1, m - i] \quad (3-28a)$$

$$q \in [1, m - j] \quad (3-28b)$$

Finalmente, as coordenadas do ponto de crossover na matriz de dimensão maior são escolhidas aleatoriamente nos intervalos:

$$k \in [0, n - p] \quad (3-29a)$$

$$l \in [0, n - q] \quad (3-29b)$$

Na Fig. 3-12 as dimensões das matrizes, as coordenadas dos pontos de crossover e as dimensões das submatrizes são, respectivamente,  $m = 6$ ,  $n = 8$ ,  $i = 1$ ,  $j = 0$ ,  $k = 2$ ,  $l = 1$ ,  $p = 2$  e  $q = 4$ .

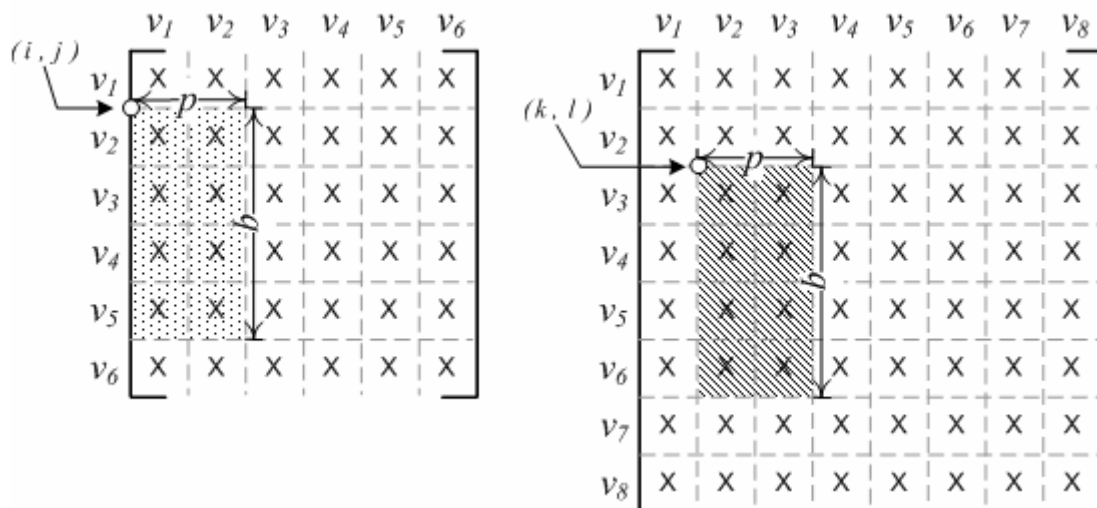


Figura 3-12. Crossover entre duas matrizes adjacência.

O operador de mutação simplesmente substitui uma coluna aleatoriamente da matriz por uma nova coluna contendo um elemento também escolhido aleatoriamente. A Fig. 3-13 mostra um exemplo do esquema de mutação proposto. Nesta figura, um retardo que estava entre os nós 3 e 4 virou um multiplicador entre os nós 5 e 4 após a mutação.

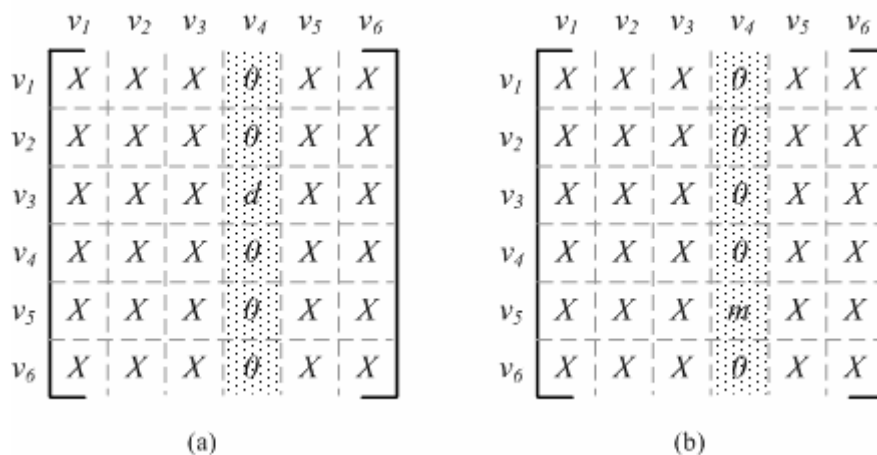


Figura 3-13. Representação da matriz adjacência de um indivíduo antes (a) e depois (b) da mutação.

### 3.4.7 Comparação com GP

Métodos evolutivos para sintetizar filtros digitais com baixa sensibilidade como propostos por Uesaka e Kawamata [67-68] utilizam árvores no esquema de codificação do cromossomo e Funções Definidas Automaticamente (*Automatically Defined Functions – ADFs*) [90].

Um exemplo de representação de um filtro digital de segunda ordem usando ADFs é mostrado na Fig. 3-14. Neste caso, como pode ser visto na Fig. 3-14(b), a

estrutura do filtro é descrita por um programa na linguagem LISP [91] da qual a árvore em Fig. 3-14(c) é construída. Nesta figura, os nós acima da linha pontilhada podem ser divididos em duas categorias: *ramos de funções* e *ramos de resultados*. A primeira categoria é composta pelos nós de  $v_0$  a  $v_7$ , enquanto a segunda categoria é composta pelo nó de saída  $y$ . As subárvores provenientes dos nós acima da linha pontilhada são chamadas *corpos*. Os corpos, por sua vez, são constituídos de um conjunto de *funções* e *argumentos*. Portanto, o ramo correspondente ao nó  $v_3$  consiste do conjunto de funções  $\{a_1, a_3, a_4, m_2\}$  e do conjunto de argumentos  $\{v_1, v_2, v_3\}$ . Os nós acima da linha pontilhada são fixos e não são afetados pelos operadores genéticos.

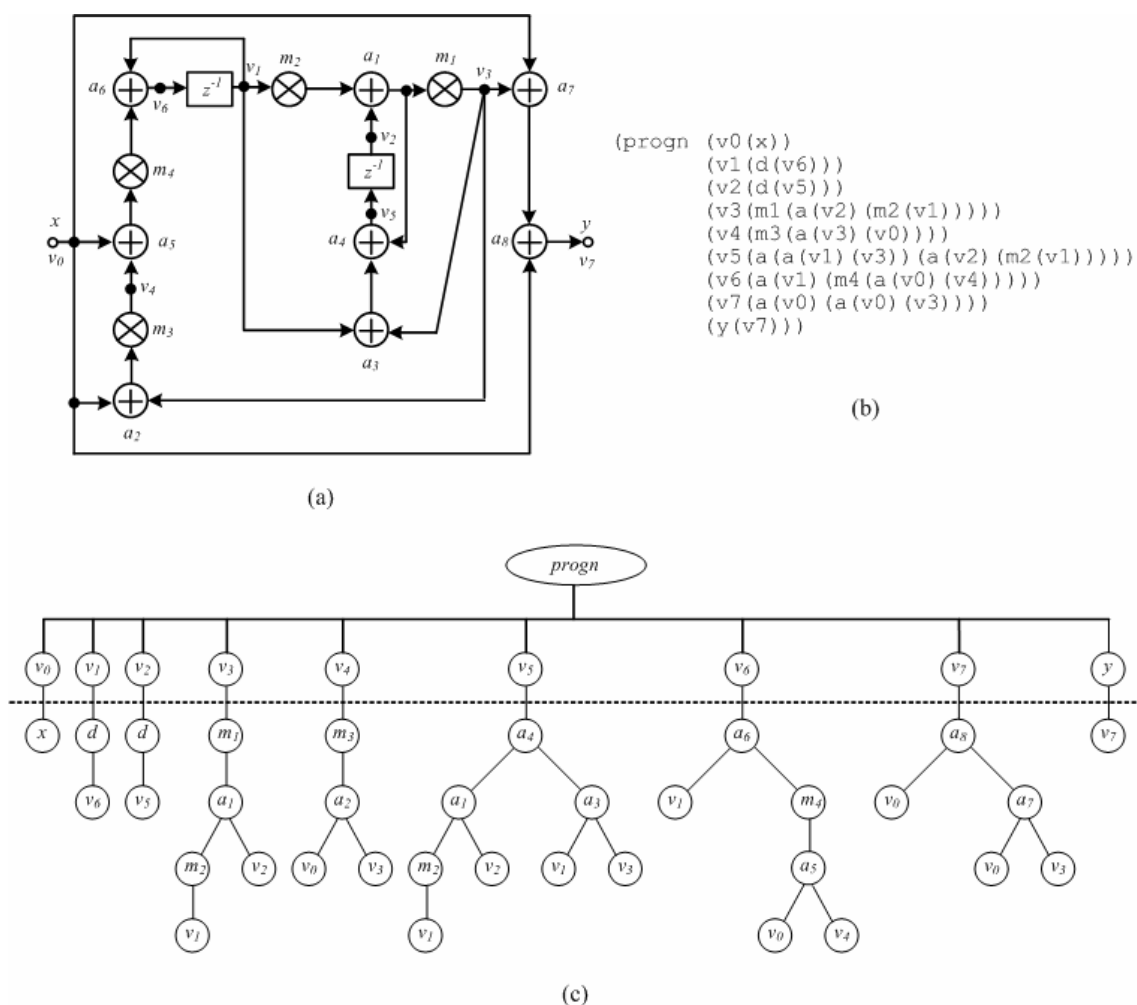


Figura 3-14. (a) Estrutura do filtro de segunda ordem (b) programa em LISP correspondente (c) representação em árvore da GP.

### 3.4.7.1 Crossover

Para comparar os mecanismos de crossover da representação em árvore da GP e a representação utilizando matriz adjacência, considere o ramo da função correspondente ao nó  $v_6$  na figura anterior que é repetido, por conveniência, na Fig. 3-



15(a). Na GP, qualquer nó pertencente a qualquer corpo pode ser selecionado como ponto de crossover. Nesta figura, todos os seis possíveis pontos de crossover estão numerados e indicados por círculos pontilhados.

Para obter a representação usando matriz adjacência que corresponde a este ramo, a estrutura em árvore deve ser transformada em um grafo orientado como mostrado na Fig. 3-15(b). O grafo orientado é obtido pelo mapeamento das funções e dos argumentos da árvore da GP em ramos e vértices do grafo, respectivamente. Note que para realizar o mapeamento é necessário definir dois argumentos adicionais na representação em árvore,  $v_x$  e  $v_y$ , uma vez que não existem argumentos incidindo no nó da função  $m_4$ .

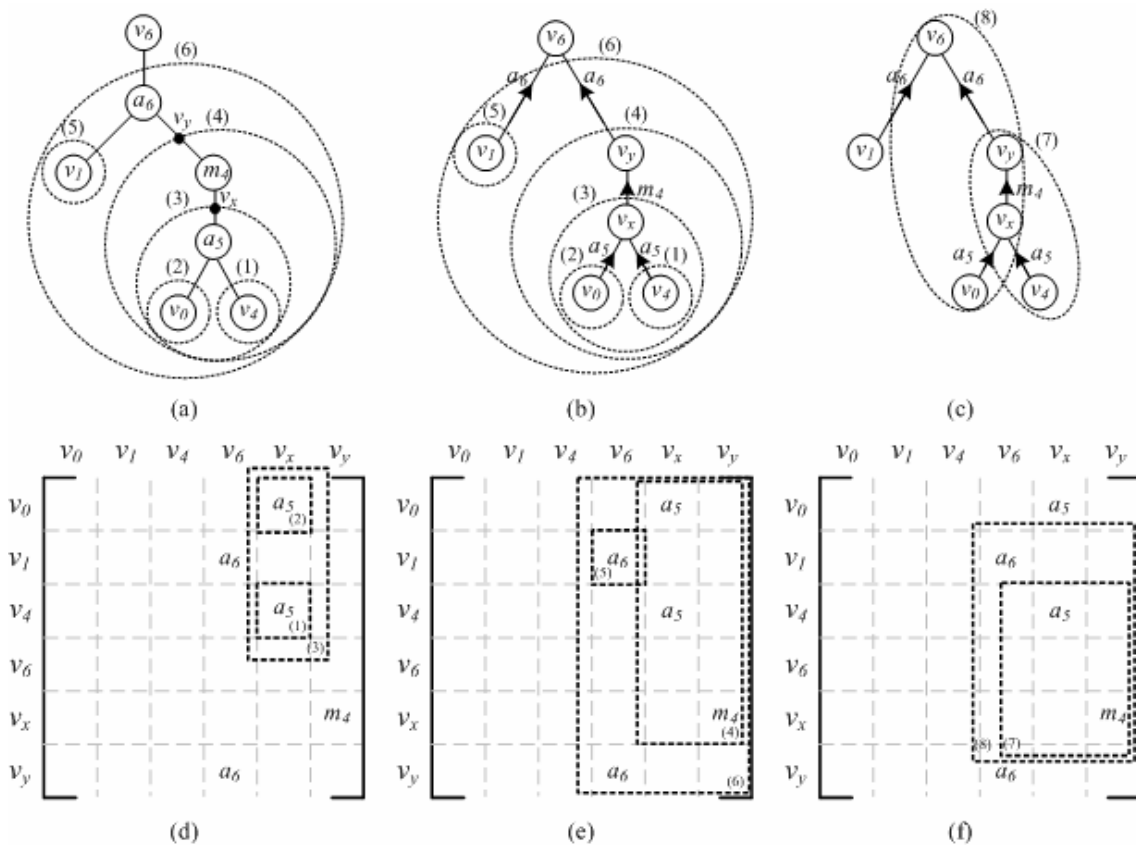


Figura 3-15. (a) Representação em árvore da GP do ramo da função correspondente ao nó  $v_6$  da Fig. 3-14(c) (b) grafo associado à representação usando matriz adjacência indicando os mesmos pontos de crossover da representação com GP (c) grafo associado à representação usando matriz adjacência indicando dois pontos de crossover que não são possíveis utilizando a GP (d) matriz adjacência correspondente à GP indicando os pontos de crossover (1), (2) e (3) (e) matriz adjacência correspondente à GP indicando os pontos de crossover (4), (5) e (6) (f) matriz adjacência correspondente aos dois pontos adicionais de crossover (7) e (8).

Na Fig. 3-15(b) os pontos de crossover que correspondem àqueles da Fig. 3-15(a) são também indicados. De maneira a tornar a figura mais clara, as submatrizes da matriz adjacência correspondentes aos pontos de crossover são representadas em duas matrizes distintas, Figs. 3-15(d-e). Portanto, usando o esquema de crossover

descrito na seção 3.4.6 todos os possíveis pontos de crossover da representação em árvore da GP podem ser mapeados na representação usando matriz adjacência. Por exemplo, o ponto de crossover (3) nas Figs. 3-15(a-b) é definido pelos seguintes parâmetros se a representação usando matriz adjacência é empregada:  $i = 0$ ,  $j = 4$ ,  $p = 1$  e  $q = 3$ .

Na Fig. 3-15(c) são mostrados dois pontos adicionais de crossover que não são possíveis na representação usando a representação em árvore da GP. A representação em matriz adjacência para estes dois pontos é dada na Fig. 3-15(f). Conseqüentemente, a representação em matriz adjacência está associada com espaços de buscas maiores do que aqueles que são possíveis de serem explorados usando GP.

### 3.4.7.2 Mutação

Na GP, a mutação cria um novo ramo de função substituindo um argumento existente por outro possível argumento [67]. Como exemplo, considere o mesmo ramo de função  $v_6$  discutido anteriormente. Os símbolos dos argumentos que podem ser substituídos com a aplicação da mutação são indicados por elipses pontilhadas numeradas de (1) a (3) na Fig. 3-16(a). Na representação usando matriz adjacência, substituir um argumento é equivalente a mover o elemento correspondente da matriz ao longo de uma dada coluna, conforme indicado por setas pontilhadas na Fig. 3-16(b).

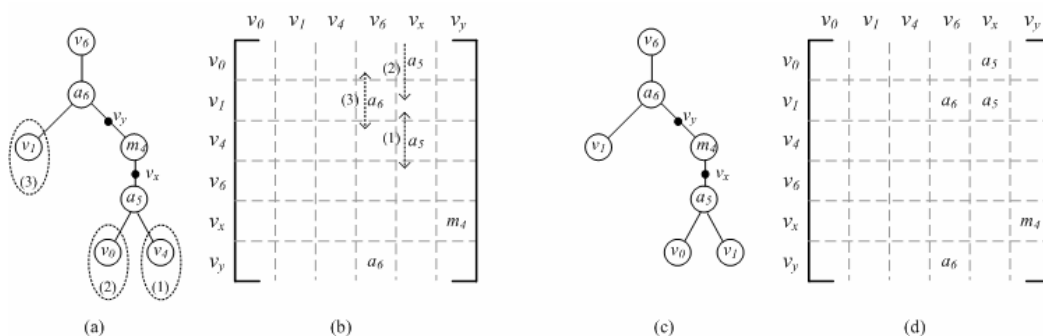


Figura 3-16. (a) Representação usando GP e (b) matriz adjacência antes da mutação (c) representação usando GP (d) matriz adjacência após a mutação.

Por exemplo, se o argumento  $v_4$  indicado pela elipse (1) na Fig. 3-16(a) é substituído durante a mutação pelo argumento  $v_j$  como mostrado na Fig.3-16(c), isto corresponderá a mover a função  $a_5$  da posição (linha  $v_4$ , coluna  $v_x$ ) indicada na Fig. 3-16(b) para a posição (linha  $v_j$ , coluna  $v_x$ ) in Fig. 3-16(d).

Note que o esquema de mutação da GP pode somente mudar um argumento por outro argumento, mantendo os símbolos associados às funções inalterados. Em contrapartida, o esquema de mutação da seção 3.4.6 pode operar em qualquer elemento da árvore movendo os elementos da matriz ao longo das colunas e eventualmente mudando o tipo da função.

### 3.4.7.3 Inversão

Na abordagem utilizando árvores da GP, a operação chamada inversão cria um novo indivíduo permutando duas subárvores em um mesmo indivíduo da população. A mesma operação pode ser implementada na abordagem utilizando matriz adjacência simplesmente permutando duas colunas da matriz. Como um exemplo, considere os dois ramos das funções  $v_5$  e  $v_6$  da Fig. 3-14(c), repetidos na Fig. 3-17(a) por conveniência, com um par de pontos de inversão indicados por elipses pontilhadas. A matriz adjacência que representa estas subárvores é mostrada na Fig. 3-17(c), onde somente os elementos da matriz relacionados a estes dois ramos de funções são representados. A árvore da GP e sua correspondente matriz adjacência antes do operador de inversão são mostradas nas Figs. 3-17(b, d), respectivamente.

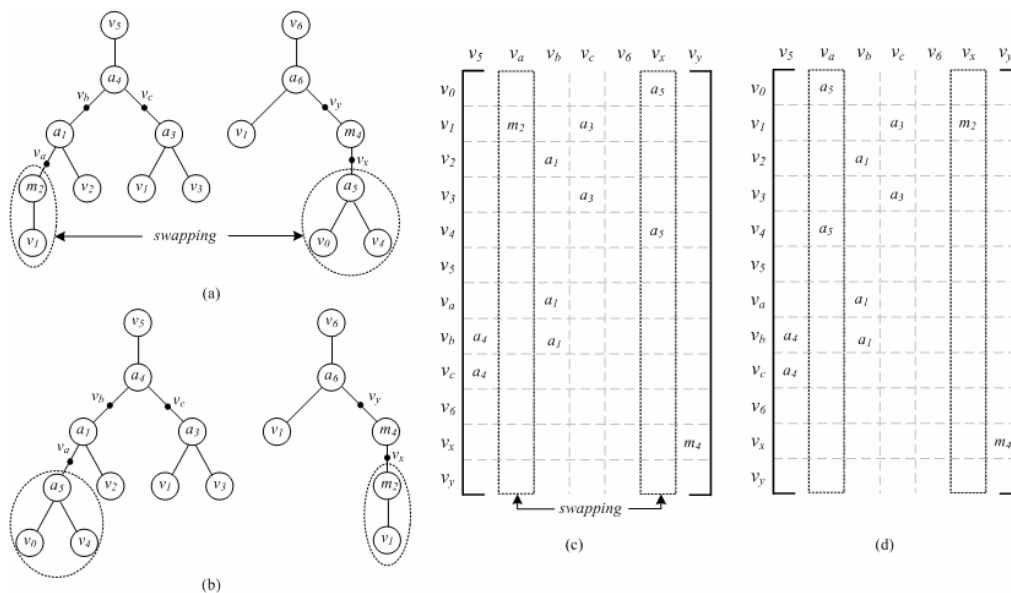


Figura 3-17. (a) Ramos de funções correspondentes aos nós  $v_5$  e  $v_6$  da Fig. 3-14(c) (b) ramos resultantes após o operador de inversão (c) matriz adjacência correspondente aos ramos definidos em (a) (d) matriz adjacência correspondente ao ramo de definição em (b).

Note que o mesmo efeito pode ser atingido por duas mutações sucessivas como descrito na seção 3.4.6. Outro ponto importante é que não é incomum que em uma representação em árvore tenha dois diferentes ramos de funções compartilhando a mesma subárvore, conforme indicado pelas elipses pontilhadas na Fig.3-18(a). Esta

situação é indicada na representação usando matriz adjacência pela presença de mais de um elemento em uma mesma linha conforme mostrado na Fig.3-18(b). Isto não representa uma anomalia e não viola nenhuma regra.

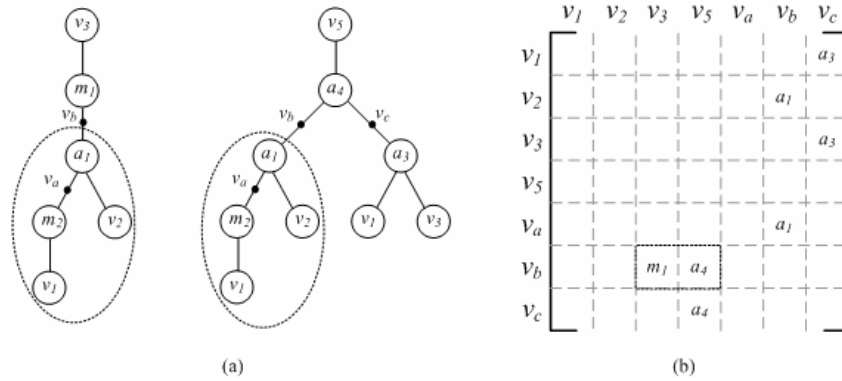


Figura 3-18. (a) Dois ramos de função compartilhando a mesma subárvore (b) matriz adjacência correspondente.

### 3.5 Estudo de Casos

#### 3.5.1 Filtro de Segunda Ordem

Em todas as sínteses evolutivas de filtros digitais a técnica de reprodução utilizada foi a *generacional* com elitismo. O cálculo da aptidão permite verificar se um indivíduo gerado pelo processo evolutivo sintetiza uma dada função de transferência desejada. Além da função de transferência desejada, as entradas do algoritmo são: o número máximo de nós, o intervalo de valores dos multiplicadores, os pesos da função objetivo, duas constantes e os parâmetros de controle usuais de um algoritmo evolutivo.

##### 3.5.1.1 Cálculo da Aptidão

A aptidão é calculada com o inverso da função:

$$\varepsilon = 1 + w_{if} \varepsilon_{if} + w_{sens} \varepsilon_{sens} \quad (3-30)$$

onde  $\varepsilon$  é a função objetivo, os  $w_i$ s são os pesos e  $\varepsilon_{if}$  e  $\varepsilon_{sens}$  são dados por:

$$\varepsilon_{if} = \begin{cases} 0, & \text{se } H_{desejada}(z) \text{ é realizável} \\ E_{if}, & \text{outros casos} \end{cases} \quad (3-31a)$$

$$\varepsilon_{sens} = \begin{cases} S, & \text{se } H_{desejada}(z) \text{ é realizável} \\ E_{sens}, & \text{outros casos} \end{cases} \quad (3-31b)$$

onde  $E_{tf}$  e  $E_{sens}$  são constantes e  $S$  é a sensibilidade do filtro digital dada por [92]:

$$S = \sum_{i=1}^N \max_{\omega} |S_{critério}| \quad (3-32)$$

onde  $\omega$  é a frequência,  $N$  é o número de multiplicadores e  $S_{critério}$  é o critério usado para determinar a sensibilidade. Um dos critérios de sensibilidade é aquele que utiliza a *sensibilidade absoluta* do módulo da função de transferência:

$$S_{mag}^{[H],i} = \frac{\partial |H(e^{j\omega})|}{\partial m_i} \quad (3-33)$$

onde  $|H(e^{j\omega})|$  é o módulo da função de transferência e  $m_i$  é o  $i$ -ésimo multiplicador.

Portanto,  $\varepsilon_{tf}$  e  $\varepsilon_{sens}$  são funções penalidade relativas à função de transferência e à sensibilidade. Estas funções implementam o fato de que muitos indivíduos gerados durante o processo evolutivo não realizam a função de transferência desejada.

A escolha dos pesos e constantes em (3-30) e (3-31) são baseadas em muitas execuções do algoritmo evolutivo até que valores razoáveis para as aptidões média e do melhor indivíduo fossem atingidas [18].

Como um exemplo de cálculo da aptidão, considere a seguinte função de transferência desejada:

$$H_{desejada}(z) = \frac{z + 0.75}{z - 0.50} \quad (3-34)$$

Assuma que a estrutura de um filtro de primeira ordem realizado em forma direta é obtida durante o processo evolutivo, conforme mostrado na Fig.3-19. Em estruturas realizadas em forma direta, os multiplicadores são exatamente os coeficientes da função de transferência. A matriz adjacência correspondente a estrutura do filtro digital é mostrada na Fig. 3-19(b). Cada coluna da matriz adjacência, exceto a primeira, representa uma equação do sistema de equações na Fig. 3-19(c).

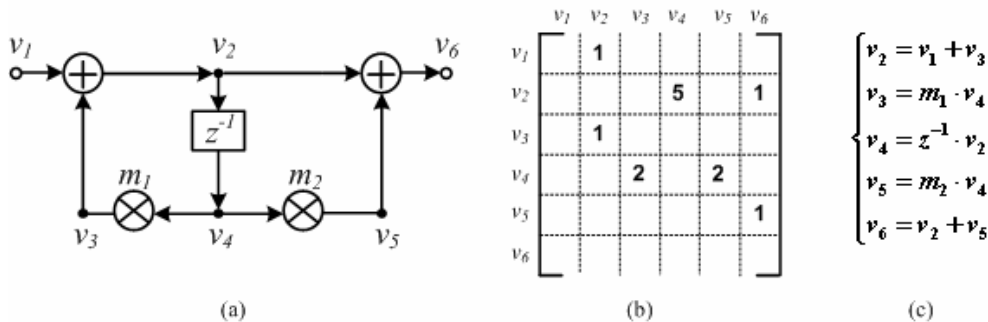


Figura 3-19. (a) Realização em forma direta do exemplo (b) representação da matriz adjacência correspondente (c) sistema de equações derivado da matriz adjacência.

Um pacote de matemática simbólica é usado para obter a função de transferência do *arquivo texto* contendo o sistema de equações da Fig.3-19(c) que descreve a topologia evoluída, conforme indicado na Fig. 3-20.

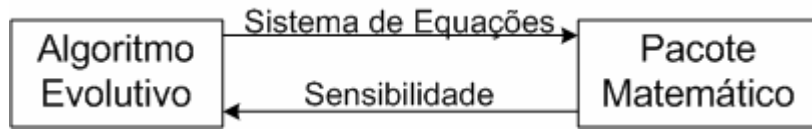


Figura 3-20. Diagrama em blocos do cálculo da sensibilidade.

Os cálculos realizados pelo pacote de matemática simbólica são descritos no algoritmo abaixo:

*Algoritmo 4: Cálculo da Aptidão usando o Critério da Sensibilidade Absoluta*

S1. A função de transferência  $H(z)$  é obtida resolvendo o sistema de equações. Os coeficientes da função de transferência são dependentes dos multiplicadores, conforme ilustra a equação abaixo obtida a partir da Fig.3-19:

$$H(z) = \frac{v_6}{v_1} = \frac{z + m_2}{z - m_1} \quad (3-35)$$

S2. Os valores dos multiplicadores são extraídos resolvendo o sistema de equações obtido igualando os coeficientes em (3-34) e (3-35) ( $m_1 = 0.50 ; m_2 = 0.75$ ).

É importante notar que em uma dada população nem todos os indivíduos realizarão a função de transferência, tornando o sistema de equações impossível de ser resolvido. Neste caso, o pacote de matemática simbólica retorna para o algoritmo evolutivo uma variável booleana, indicando que este indivíduo não realiza a função de transferência. Então, o algoritmo evolutivo calcula a aptidão usando (3-30) e (3-31).

S3. O módulo da função de transferência é calculado substituindo  $z = e^{j\omega}$  em (3-35):

$$|H(e^{j\omega})| = \sqrt{\frac{m_2^2 + 2m_2 \cos \omega + 1}{m_1^2 - 2m_1 \cos \omega + 1}} \quad (3-36)$$

Este procedimento de cálculo da aptidão requer significativo esforço computacional, uma vez que requer a manipulação de funções transcendentais. Será mostrado na seção 3.5.2 que este procedimento pode ser otimizado se outro critério de sensibilidade for utilizado:

S4. A sensibilidade do módulo da função de transferência com relação aos multiplicadores é obtida derivando (3-36):

$$S_{mag}^{|H|,1} = \frac{\partial |H(e^{j\omega})|}{\partial m_1} = \sqrt{\frac{(\cos \omega - m_1)^2 (m_2^2 + 2m_2 \cos \omega + 1)}{(m_1^2 - 2m_1 \cos \omega + 1)^3}} \quad (3-37a)$$

$$S_{mag}^{|H|,2} = \frac{\partial |H(e^{j\omega})|}{\partial m_2} = \sqrt{\frac{(\cos \omega - m_2)^2}{(m_1^2 - 2m_1 \cos \omega + 1)(m_2^2 + 2m_2 \cos \omega + 1)}} \quad (3-37b)$$

S5. Os valores dos multiplicadores extraídos em S2 são substituídos em (3-37), resultando em expressões para as sensibilidades dependentes somente de  $\omega$ .

S6. De maneira a encontrar a frequência,  $\omega_{\max}$ , que maximiza as sensibilidades, as derivadas de (3-37) com relação a  $\omega$  são igualadas a zero. Neste exemplo, a sensibilidade do módulo máxima com relação a  $m_1$  e  $m_2$  foram 7,00 e 2,00, respectivamente. Finalmente, como estabelecido em (3-32), a sensibilidade  $S$  é a soma dos valores de máxima sensibilidade com relação a todos os multiplicadores. A Tabela 3-4 resume os valores obtidos durante o cálculo da sensibilidade.

Tabela 3-4. Cálculo da sensibilidade do exemplo anterior.

Multiplicadores		$\omega_{\max}$	$S_{mag}^{ H ,i}$	S
$m_1$	0.50	0	7.00	9.00
$m_2$	0.75	0	2.00	

O pacote de matemática simbólica retorna para o algoritmo evolutivo o valor da sensibilidade e as aptidões dos indivíduos são calculadas usando (3-30) e (3-31). A Tabela 3-5 resume os valores obtidos no exemplo.

Tabela 3-5. Cálculo da aptidão da função de transferência (3-34).

$w_{sens} = w_{tf}$	1
$\mathcal{E}_{tf}$	0
$\mathcal{E}_{sens}$	9
aptidão	0.10

De forma a comparar o desempenho do algoritmo evolutivo descrito com outras abordagens encontradas na literatura, a seguinte função de transferência de um filtro passa-baixas de segunda ordem foi utilizada [68]:

$$H(z) = \frac{0.098244z^2 - 0.195065z + 0.098244}{z^2 - 1.957184z + 0.9586936} \quad (3-38)$$

Este filtro possui pólos próximos do círculo unitário sendo particularmente sensível ao efeito da quantização de coeficientes que, podendo causar instabilidade.

Os valores permitidos para os multiplicadores pertencem ao intervalo [-1, +1], de forma a aumentar o número de bits necessários para a representação da parte

fracionária após o processo de quantização. A Tabela 3-6 mostra os parâmetros de controle utilizados.

Tabela 3-6. Parâmetros do processo evolutivo.

Tamanho da População	200
Número de Gerações	50
Taxa de Crossover	70
Taxa de Mutação	40
Intervalo dos Multiplicadores	[-1,+1]
Número Máximo de Nós	21
$w_{sens}$	1
$w_{ff}$	0.5
$E_{sens}$	50
$E_{ff}$	49

A Fig. 3-21 mostra a aptidão média e a aptidão do melhor indivíduo ao longo das gerações. Como mostrado na figura, o algoritmo responde bem aos operadores genéticos, aumentando consideravelmente a aptidão ao longo das gerações. Isto sugere que as regras incluídas para prevenir a avaliação de indivíduos inválidos após a aplicação dos operadores genéticos não restringem o espaço de busca e nem reduzem significativamente o desempenho global do algoritmo evolutivo.

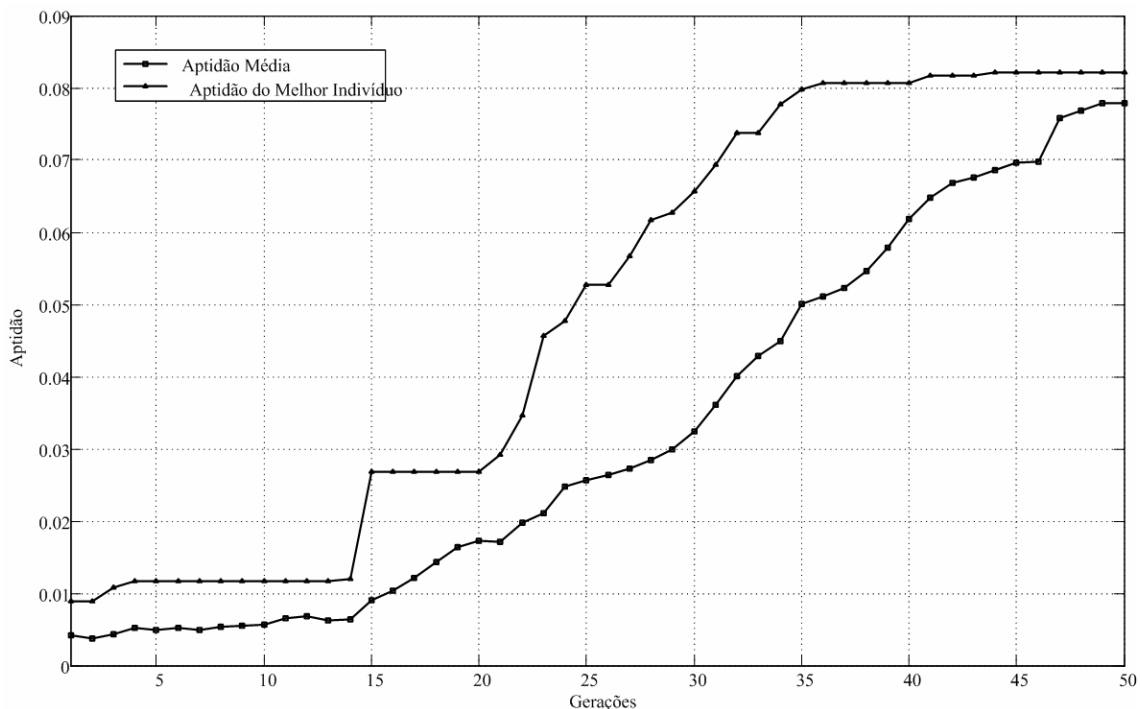


Figura 3-21. Aptidão média e do melhor indivíduo ao longo das gerações para a função de transferência descrita em (3-38).

A topologia sintetizada após 50 gerações é mostrada na Fig. 3-22. Dezessete nós, dois atrasadores, cinco multiplicadores e nove somadores foram necessários para realizar a função de transferência com baixa sensibilidade à quantização de coeficientes. Além disso, esta estrutura utiliza o número mínimo de multiplicadores que



representa um aspecto importante no projeto de filtros digitais, uma vez que os multiplicadores são os componentes mais custosos em qualquer implementação.

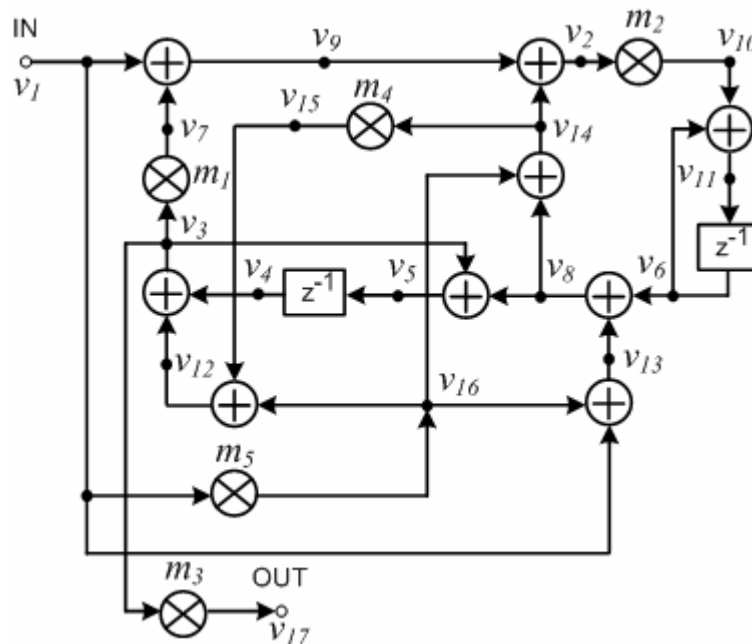


Figura 3-22. Topologia sintetizada pelo processo evolutivo proposto.

Os valores dos multiplicadores, as máximas frequências com relação a cada multiplicador e a sensibilidade absoluta total são dadas na Tabela 3-7.

Tabela 3-7. Resultados do processo evolutivo da função de transferência (3-38).

Multiplicador	Frequência ( $\omega_{max}$ ) [rd/s]	$S_{mag}^{ H ,i}$
$m_1$	0.245993272	0
$m_2$	-0.054782005	0.119944850
$m_3$	-0.231820149	0.022617514
$m_4$	-0.887948984	0.119961408
$m_5$	-0.598216458	0.060374487
Sensibilidade Total = 11.181969612		

Uma comparação dos resultados obtidos pelas diferentes metodologias encontradas na literatura para sintetizar estruturas com baixa sensibilidade aos coeficientes é dada na Tabela 3-8.

Tabela 3-8. Sensibilidade do módulo da função de transferência (3-38)

Forma Direta	2393.883
Diniz & Antoniou [83]	1317.327
Forma Balanceada [93]	75.817
Uesaka & Kawamata [68]	15.370
Topologia Sintetizada	11.182

Todos os resultados referem-se à função de transferência de segunda ordem (3-38). As três primeiras linhas da tabela usam abordagens determinísticas tradicionais,

enquanto as duas últimas linhas usam algoritmos evolutivos. Pode ser observado que a menor sensibilidade foi obtida com a topologia sintetizada pelo algoritmo evolutivo baseado na representação usando matriz adjacência.

### 3.5.2 Filtro de Quinta Ordem

O procedimento de cálculo da aptidão descrito na seção anterior é adequado apenas para filtros com baixa ordem de complexidade. Filtros mais complexos usando o procedimento de cálculo descrito anteriormente iriam requerer grande esforço computacional para serem avaliados pelo pacote de matemática simbólica. De maneira a acelerar o cálculo da aptidão, um novo procedimento usando o *critério da sensibilidade relativa* é empregado [83]:

$$S_{rel}^{|H|,i} = \frac{m_i}{|H(e^{j\omega})|} \frac{\partial |H(e^{j\omega})|}{\partial m_i} \quad (3-39)$$

Se este critério é usado, a sensibilidade relativa do módulo ao quadrado é duas vezes a sensibilidade relativa do módulo:

$$S_{rel}^{|H|^2,i} = \frac{m_i}{|H(e^{j\omega})|^2} \frac{\partial |H(e^{j\omega})|^2}{\partial m_i} \quad (3-40a)$$

$$S_{rel}^{|H|^2,i} = 2S_{rel}^{|H|,i} \quad (3-40b)$$

Portanto, de acordo com (3-32) e (3-40), as frequências que maximizam a sensibilidade relativa do módulo são as mesmas que maximizam a sensibilidade relativa do módulo ao quadrado. Pode ser verificado que esta propriedade não se aplica ao critério da sensibilidade absoluta do módulo.

A vantagem de usar o quadrado do módulo da função de transferência no domínio simbólico advém da possibilidade de eliminar a raiz quadrada que afeta o cálculo do módulo da função de transferência. Como será mostrado a seguir, é possível derivar um algoritmo simples para encontrar as frequências que maximizam a sensibilidade do módulo. Como estabelecido em (3-32), para encontrar estas frequências as seguintes equações devem ser resolvidas para cada multiplicador:

$$\frac{\partial S_{rel}^{|H|^2,i}}{\partial \omega} = 0 \quad (3-41)$$

Usando a regra da cadeia e o fato de que  $z = e^{j\omega}$ , (3-41) pode ser reescrita como:

$$\frac{\partial S_{rel}^{|H|^2,i}}{\partial \omega} = \frac{\partial S_{rel}^{|H|^2,i}}{\partial z} \cdot \frac{\partial z}{\partial \omega} = \frac{\partial S_{rel}^{|H|^2,i}}{\partial z} \cdot jz = 0 \quad (3-42)$$

Esta equação possui duas soluções, uma trivial dada por  $z = 0$  e:

$$\frac{\partial S_{rel}^{|H|^2, i}}{\partial z} = 0 \quad (3-43)$$

Considerando que:

$$\left| H(e^{j\omega}) \right|^2 = H(e^{j\omega})H^*(e^{j\omega}) \quad (3-44)$$

onde  $H^*(e^{j\omega})$  é o complexo conjugado de  $H(e^{j\omega})$ .

Tomando (3-40a) e substituindo (3-44) em (3-42) resulta:

$$\frac{\partial S_{rel}^{|H|^2, i}}{\partial z} = m_i \frac{\partial}{\partial z} \left[ \frac{1}{HH^*} \frac{\partial}{\partial m_i} (HH^*) \right] = 0 \quad (3-45a)$$

$$\frac{\partial S_{rel}^{|H|^2, i}}{\partial z} = m_i \frac{\partial}{\partial z} \left[ \frac{1}{HH^*} \left( H \frac{\partial H^*}{\partial m_i} + H^* \frac{\partial H}{\partial m_i} \right) \right] = 0 \quad (3-45b)$$

Por uma questão de simplificação os argumentos das funções foram omitidos nas equações acima. Excluindo a solução trivial  $m_i = 0$ , o segundo fator no lado direito de (3-45b) pode ser reescrito como:

$$\frac{\partial}{\partial z} \left( \frac{1}{H^*} \right) \frac{\partial H^*}{\partial m_i} + \frac{1}{H^*} \frac{\partial}{\partial z} \left( \frac{\partial H^*}{\partial m_i} \right) + \frac{\partial}{\partial z} \left( \frac{1}{H} \right) \frac{\partial H}{\partial m_i} + \frac{1}{H} \frac{\partial}{\partial z} \left( \frac{\partial H}{\partial m_i} \right) = 0 \quad (3-46)$$

Observando que:

$$\frac{\partial}{\partial z} \left( \frac{1}{H^*} \right) = -\frac{\partial H^*}{\partial z} \frac{1}{H^{*2}} = -\frac{\partial H^*}{\partial z^*} \frac{\partial z^*}{\partial z} \frac{1}{H^{*2}} = -\left( \frac{\partial H}{\partial z} \right)^* - \frac{1}{z^2} \frac{1}{H^{*2}} = \frac{1}{z^2} \left( \frac{\partial H}{\partial z} \right)^* \frac{1}{H^{*2}} \quad (3-47a)$$

$$\frac{\partial H^*}{\partial m_i} = \left( \frac{\partial H}{\partial m_i} \right)^* \quad (3-47b)$$

$$\frac{\partial}{\partial z} \left( \frac{\partial H^*}{\partial m_i} \right) = \frac{\partial}{\partial m_i} \left( \frac{\partial H^*}{\partial z} \right) = \frac{\partial}{\partial m_i} \left[ -\frac{1}{z^2} \left( \frac{\partial H}{\partial z} \right)^* \right] = -\frac{1}{z^2} \left( \frac{\partial^2 H}{\partial z \partial m_i} \right)^* \quad (3-47c)$$

$$\frac{\partial}{\partial z} \left( \frac{1}{H} \right) = -\frac{\partial H}{\partial z} \frac{1}{H^2} \quad (3-47d)$$

Substituindo (3-47) em (3-46) obtém-se:

$$\frac{1}{z^2} \left( \frac{\partial H}{\partial z} \right)^* \frac{1}{H^{*2}} \left( \frac{\partial H}{\partial m_i} \right)^* - \frac{1}{z^2} \frac{1}{H^*} \left( \frac{\partial^2 H}{\partial z \partial m_i} \right)^* - \frac{\partial H}{\partial z} \frac{1}{H^2} \frac{\partial H}{\partial m_i} + \frac{1}{H} \frac{\partial^2 H}{\partial z \partial m_i} = 0 \quad (3-48)$$

A equação (3-48) pode ser expressa na forma:

$$\frac{1}{z^2} \left[ \frac{\partial H}{\partial z} \frac{1}{H^2} \frac{\partial H}{\partial m_i} - \frac{1}{H} \frac{\partial^2 H}{\partial z \partial m_i} \right]^* - \left[ \frac{\partial H}{\partial z} \frac{1}{H^2} \frac{\partial H}{\partial m_i} - \frac{1}{H} \frac{\partial^2 H}{\partial z \partial m_i} \right] = 0 \quad (3-49)$$

Portanto, é possível escrever:

$$z^2 R_i(z) - R_i^*(z) = 0 \quad (3-50)$$

onde  $R_i(z)$  é dado por:

$$R_i(z) = \frac{\partial H}{\partial z} \frac{1}{H^2} \frac{\partial H}{\partial m_i} - \frac{1}{H} \frac{\partial^2 H}{\partial z \partial m_i} \quad (3-51)$$

Uma vez que  $R(z)$  é uma função racional real e  $z = e^{j\omega}$ :

$$R_i^*(z) = R_i(z^*) = R_i(z^{-1}) \quad (3-52)$$

Finalmente, substituindo (3-52) em (3-50):

$$z^2 R_i(z) - R_i(z^{-1}) = 0 \quad (3-53)$$

Portanto, as freqüências que maximizam o quadrado do módulo da função de transferência devem obedecer à equação acima. Isto simplifica o cálculo da aptidão como mostrado no seguinte algoritmo modificado:

*Algorithm 5: Cálculo da Aptidão usando o Critério da Sensibilidade Relativa*

S1. Cálculo de  $H(z)$  a partir do sistema de equações conforme descrito na seção 3.5.1.

S2. Cálculo de  $H^2(z)$ ,  $\frac{\partial H(z)}{\partial z}$ ,  $\frac{\partial H(z)}{\partial m_i}$  e  $\frac{\partial^2 H(z)}{\partial z \partial m_i}$ .

S3. Cálculo de  $R_i(z)$  usando (3-51).

S4. Cálculo de  $R_i(z^{-1})$  substituindo  $z = z^{-1}$  em  $R_i(z)$ .

S5. Extração dos valores dos multiplicadores conforme descrito na seção 3.5.1 e substituição destes valores em (3-53).

S6. Cálculo dos valores de  $z$  que satisfazem (3-53). Uma vez que  $z = e^{j\omega}$ , somente as raízes  $z_{root}$  sobre o círculo de raio unitário devem ser consideradas, as outras raízes devem ser descartadas.

S7. Cálculo das freqüências  $\omega_{max}$  que maximizam a sensibilidade relativa usando  $z_{root} = e^{j\omega_{max}}$ . Note que  $z = \pm 1$  são raízes de (3-53) de forma que  $\omega = 0$  e  $\omega = \pi$  são pontos de máximo do módulo ao quadrado da sensibilidade relativa.

S8. Cálculo da sensibilidade relativa máxima com relação a cada multiplicador, substituindo  $\omega = \omega_{max}$  em (3-40a).

$$S_{rel}^{H(e^{j\omega_{max}})} \Big|_i = \frac{m_i}{\left|H(e^{j\omega_{max}})\right|^2} \frac{\partial \left|H(e^{j\omega_{max}})\right|^2}{\partial m_i} \quad (3-54)$$

S9. Cálculo da sensibilidade relativa do módulo da função de transferência usando (3-32).

$$S = S_{rel}^{|H|} = \frac{1}{2} \sum_{i=1}^N \left| S_{rel}^{|H(e^{j\omega_{max}})|^2, i} \right| \quad (3-55)$$

A manipulação matemática do módulo do quadrado da função de transferência simplifica significativamente o cálculo da aptidão quando se utiliza o pacote de matemática simbólica, uma vez que uma simples equação dada em (3-53) deve ser resolvida. Além disso, todos os cálculos até o passo final do algoritmo são realizados no domínio do plano-z, sendo a substituição  $z = e^{j\omega}$  realizada somente no último passo.

Para verificar o efeito dos procedimentos de cálculo da aptidão no desempenho do algoritmo evolutivo um filtro IIR passa-baixas de quinta ordem é usado:

$$H(z) = \sum_{i=0}^5 \frac{b_i z^i}{a_i z^i} \quad (3-56)$$

onde  $b_5 = b_0 = 0.001515495$ ,  $b_4 = b_1 = -0.004265839$ ,  $b_3 = b_2 = 0.002761523$ ,  $a_5 = 1$ ,  $a_4 = -4.785558557$ ,  $a_3 = 9.195692494$ ,  $a_2 = -8.867703530$ ,  $a_1 = 4.291113683$  e  $a_0 = -0.833521733$ . Esta função implementa um filtro com banda passante estreita, frequência de corte normalizada em  $\omega_p = 0.05$ , atenuação mínima na banda de rejeição de  $A_{st} = 50dB$  e *ripple* na banda passante de  $\delta = 0.5dB$ , com especificações ilustradas na Fig. 3-23.

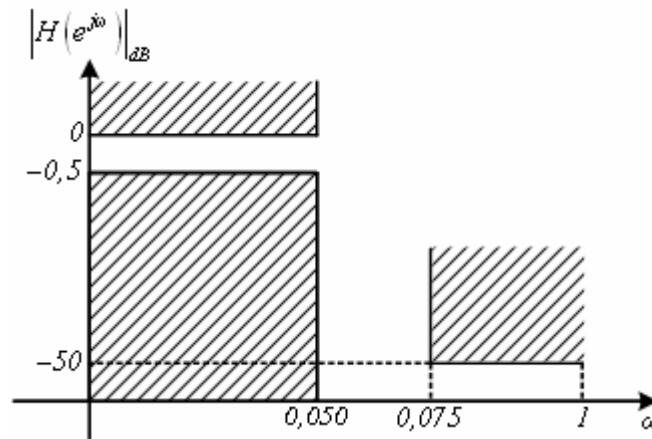


Figura 3-23. Especificações do filtro de quinta ordem.

Duas execuções do algoritmo evolutivo foram realizadas usando ambos os critérios de sensibilidade (absoluta e relativa) com os parâmetros de controle dados na Tabela 3-9. O tempo médio gasto para calcular a aptidão de cada indivíduo usando o procedimento descrito no Algoritmo 3 da seção 3.5.1 foi de 56,34s, enquanto usando o Algoritmo 4 foi de 6,69s. Uma vez que cada execução inclui 200 indivíduos e 40

gerações, esta economia de tempo representa uma considerável economia de tempo no algoritmo como um todo.

Tabela 3-9. Parâmetros de controle do processo evolutivo para o filtro passa-baixas de quinta ordem.

Tamanho da População	200
Número de Gerações	40
Taxa de Crossover	70
Taxa de Mutação	20
Intervalo dos Multiplicadores	[-2.5,+2.5]
Número Máximo de Nós	50
$w_{sens}$	1
$w_{ff}$	1
$E_{sens}$	50
$E_{ff}$	49

A topologia do melhor filtro digital que sintetiza a função de transferência (3-56) é mostrada na Fig. 3-24. A sensibilidade relativa foi 11,712, enquanto a sensibilidade relativa de uma realização em forma direta usando duas seções de segunda ordem e uma seção de primeira ordem é de 680,543.

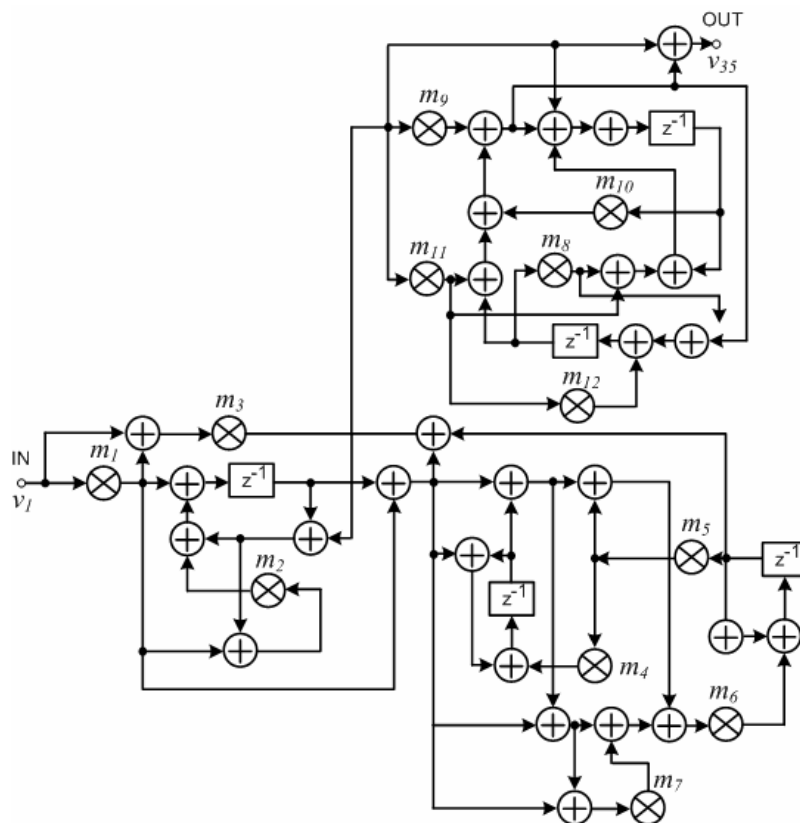


Figura 3-24. Topologia do melhor indivíduo para o filtro de quinta ordem.

A Tabela 3-10 mostra os valores dos multiplicadores, bem como as frequências que maximizam a sensibilidade relativa com relação a cada multiplicador e os valores das respectivas sensibilidades relativas.

Tabela 3-10. Resultados do processo evolutivo para a função de transferência (3-56).

	Multiplicador	Frequência ( $\omega_{\max}$ ) [rd/s]	$S_{mag}^{ H ,i}$
m1	0.001515495	0	0.500756601
m2	-0.065156076	0.034481565	1.034848639
m3	0.001618668	0.132918687	0.500000000
m4	0.531544472	0.052200241	0.929382290
m5	-0.226709521	0.090254672	1.391683310
m6	-0.562486504	0.056481795	0.365785548
m7	-0.423990058	0.045083540	0.037118317
m8	-0.025013851	0.159008896	0.968187419
m9	2.280458105	0	0.000000114
m10	-0.025083040	0.173318687	4.973381753
m11	-2.280458105	0	1.010819695
m12	0.010623156	0	0.000262114
Sensibilidade Total = 11.7122258			

A Fig. 3-25 mostra o módulo da função de transferência (3-56) do filtro ideal usando alta precisão (80 bits), a realização em forma direta usando uma seção de quinta ordem, a realização em forma direta usando duas seções de segunda ordem e uma de primeira ordem e a topologia sintetizada usando 10 bits em formato de ponto fixo. Uma perda não-linear foi adicionada à banda passante nas realizações diretas e um pequeno desvio da localização inicial dos pólos também pode ser observado. Portanto, os efeitos da quantização de coeficientes são menos severos na estrutura sintetizada utilizando a representação em matriz adjacência.

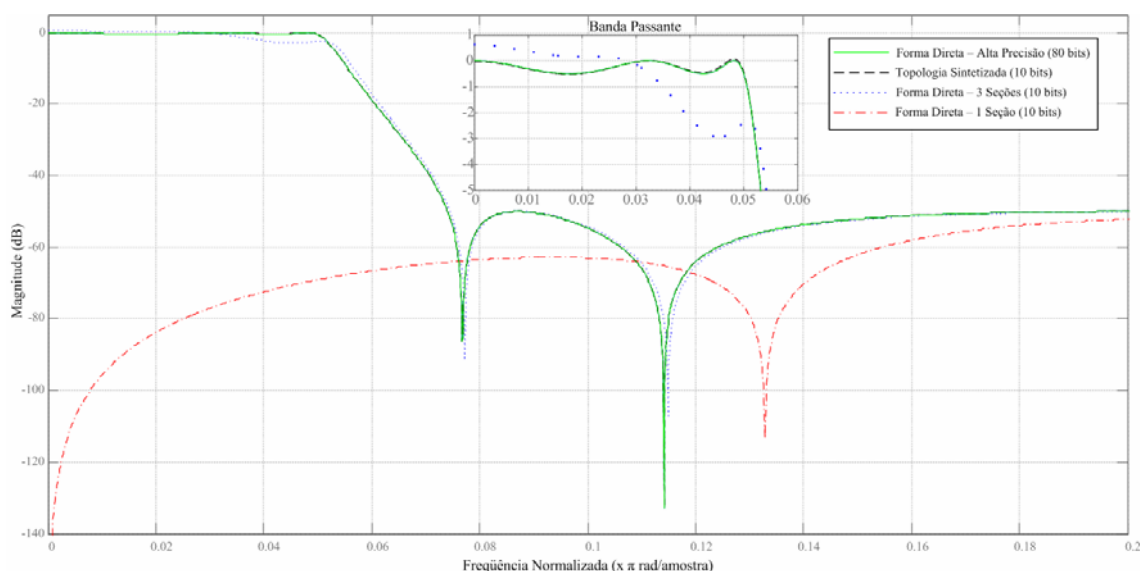


Figura 3-25. Módulo da função de transferência para o caso ideal, em realização direta e filtros sintetizados usando a representação em matriz adjacência.

Os dois exemplos anteriores calcularam a aptidão baseada apenas no módulo da resposta em frequência da função de transferência. No próximo exemplo, além do

módulo, a fase da resposta em frequência também será considerada no projeto de equalizadores de fase com baixa sensibilidade à quantização de coeficientes.

### 3.5.3 Equalizadores de Fase

A transmissão de sinais sem distorção necessita de filtros digitais com fase linear. Uma vez que filtros digitais com fase linear ao longo de todo o espectro de frequências são irrealizáveis, a especificação de fase linear é geralmente limitada à banda passante.

Filtros FIR com fase linear podem ser facilmente implementados. Contudo, filtros FIR satisfazendo as mesmas especificações de módulo que filtros IIR tendem a ter ordens muito mais elevadas. Conseqüentemente, o projeto de filtros IIR com fase linear é mais atrativo em aplicações que se exija alta seletividade do filtro.

Existem duas abordagens no projeto de filtros IIR com fase linear (ou retardo de grupo constante). Projetar um único filtro que realiza as especificações de módulo e fase simultaneamente [94] ou projetar um filtro que atenda as especificações de módulo seguido por um cascadeamento de filtros passa-tudo que corrigem a resposta de fase [95-96]. Neste exemplo, a última abordagem é adotada onde o algoritmo evolutivo é usado para sintetizar uma cascata de filtros passa-tudo com baixa sensibilidade que corrige a resposta de fase de um filtro IIR. A Fig. 3-26 ilustra o diagrama em blocos de um filtro digital com equalização de fase, onde  $H_F(z)$  e  $H_E(z)$  são as funções de transferência do filtro e do equalizador, respectivamente.

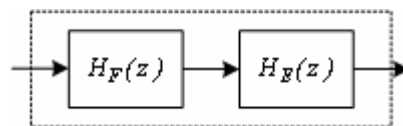


Figura 3-26. Estrutura do equalizador de retardo de grupo.

Os retardos de grupo do filtro e do equalizador são respectivamente dados por:

$$\tau_F(\omega) = -\frac{d\theta_F(\omega)}{d\omega} \quad (3-57a)$$

$$\tau_E(\omega) = -\frac{d\theta_E(\omega)}{d\omega} \quad (3-57b)$$

onde:

$$\theta_F(\omega) = \text{tg}^{-1} \frac{\text{Im}\{H_F(e^{j\omega})\}}{\text{Re}\{H_F(e^{j\omega})\}} = \angle H_F(e^{j\omega}) \quad (3-58a)$$

$$\theta_E(\omega) = \angle H_E(e^{j\omega}) \quad (3-58b)$$

Após a inserção do equalizador, a função de transferência resultante é dada por:



$$H_{FE}(z) = |H_F(e^{j\omega})| e^{j\theta_F(\omega)} \cdot |H_E(e^{j\omega})| e^{j\theta_E(\omega)} \quad (3-59)$$

Então:

$$|H_{FE}(e^{j\omega})| = |H_F(e^{j\omega})| \cdot |H_E(e^{j\omega})| \quad (3-60a)$$

$$\tau_{FE}(\omega) = \tau_F(\omega) + \tau_E(\omega) \quad (3-60b)$$

De acordo com (3-60), um filtro com retardo de fase constante pode ser projetado usando um equalizador passa-tudo que atende às seguintes especificações:

$$\begin{cases} |H_E(e^{j\omega})| = 1 & \text{for } 0 \leq \omega \leq \frac{\omega_s}{2} \\ \tau_E(\omega) = \tau - \tau_F(\omega) & \text{for } \omega_{p1} \leq \omega \leq \omega_{p2} \end{cases} \quad (3-61)$$

onde  $\omega_s$ ,  $\omega_{p1}$ ,  $\omega_{p2}$  e  $\tau$  são a freqüência de amostragem, o limite inferior da banda passante, o limite superior da banda passante e uma dada constante, respectivamente. Para atender à primeira especificação, a função de transferência do equalizador deve ser da forma:

$$H_E(z) = \prod_{i=1}^N \frac{a_{0i}z^2 + a_{1i}z + 1}{z^2 + a_{1i}z + a_{0i}} \quad (3-62)$$

onde  $N$  é o número de seções de segunda ordem.

Para assegurar a estabilidade, os pólos da função de transferência em (3-62) devem estar dentro do círculo de raio unitário. Isto é obtido restringindo os coeficientes do denominador de cada uma das  $N$  seções de segunda ordem pelas seguintes desigualdades:

$$\begin{cases} -1 < a_{0i} < 1 \\ -1 - a_{0i} < a_{1i} < 1 + a_{0i} \end{cases} \quad (3-63)$$

Substituindo  $z = e^{j\omega}$  em (3-62) e considerando (3-57b), o retardo de grupo do equalizador para a  $i$ -ésima seção pode ser escrito como:

$$\tau_{Ei}(\omega) = -\frac{2(a_{0i} - 1)(a_{0i} + a_{1i} \cos \omega + 1)}{2 \cos \omega (2a_{0i} \cos \omega + a_{0i}a_{1i} + a_{1i}) + (a_{0i} - 1)^2 + a_{1i}^2} \quad (3-64)$$

Portanto, os coeficientes  $a_{0i}$  e  $a_{1i}$  de cada seção da função de transferência  $H_E(z)$  devem satisfazer (3-63). Os coeficientes devem ser otimizados de maneira que o retardo de grupo do equalizador (3-64) corrija o retardo de grupo do filtro, tornando o retardo de grupo aproximadamente constante dentro da banda passante.

O procedimento de síntese é realizado em duas etapas. Na primeira etapa, um algoritmo genético com representação de tamanho fixo é usado como otimizador para encontrar os coeficientes  $(a_{0i}; a_{1i})$  da função de transferência do equalizador (3-64).

Nesta etapa, além dos parâmetros de controle usuais do algoritmo genético, as entradas são: a função de transferência a ser equalizada e o *fator de qualidade*  $Q$  da estrutura cascadeada. O fator de qualidade  $Q$  é uma medida do ripple na banda passante do retardo de grupo  $\tau_{FE}$  sendo dado por [54]:

$$Q = \frac{100 \cdot (\max \tau_{FE} - \min \tau_{FE})}{(\max \tau_{FE} + \min \tau_{FE})} \quad (3-65)$$

De maneira a encontrar a função de transferência do equalizador que atende às especificações do fator de qualidade, o otimizador inicia o processo com uma única seção de segunda ordem e vai inserindo seções de segunda ordem até que o valor de  $Q$  especificado seja atingido.

Na etapa seguinte, o algoritmo evolutivo usando a representação em matriz adjacência para filtros digitais é empregado para encontrar uma topologia com baixa sensibilidade usando a função de transferência do equalizador extraída na primeira etapa como entrada. Nesta etapa, que abrange à síntese topológica, além da sensibilidade do módulo da função de transferência um outro fator correspondente à sensibilidade da fase da resposta em frequência é adicionado ao cálculo da aptidão. A equação (3-30) é alterada para incluir a sensibilidade da fase:

$$\varepsilon = 1 + w_{\varepsilon_f} \varepsilon_f + w_{|H_E|} \cdot S^{|H_E|} + w_{\tau_E} \cdot S^{\tau_E} \quad (3-66)$$

onde  $\varepsilon_f$  é dado por (3-31a) e:

$$S^{|H_E|} = \sum_{j=1}^{N_m} \max_{\omega} |S_{m_j}^{|H_E|}(\omega)| \quad (3-67a)$$

$$S^{\tau_E} = \sum_{j=1}^{N_m} \max_{\omega} |S_{m_j}^{\tau_E}(\omega)| \quad (3-67b)$$

onde  $N_m$  é o número de multiplicadores e as sensibilidades do módulo e da fase são calculadas usando o critério da sensibilidade relativa:

$$S_{m_j}^{|H_E|}(\omega) = \frac{m_j}{|H_E(e^{j\omega})|} \frac{\partial |H_E(e^{j\omega})|}{\partial m_j} \quad (3-68a)$$

$$S_{m_j}^{\tau_E}(\omega) = \frac{m_j}{\tau_E(\omega)} \frac{\partial \tau_E(\omega)}{\partial m_j} \quad (3-68b)$$

Como estudo de caso, considere o projeto de um filtro passa-baixas que deve satisfazer às especificações de módulo descritas na Tabela 3-11 e um fator de qualidade menor que 2,50 na banda passante.

Tabela 3-11. Especificações para o filtro elíptico.

Ripple na Banda Passante	1dB
Atenuação na Banda de Rejeição	40dB
Limite da Banda Passante	0,10 rad/s
Limite da Banda de Rejeição	0,15 rad/s
Frequência de Amostragem	2,00 rad/s

Usando métodos convencionais [54], uma função de transferência de quarta ordem elíptica que atende às especificações de módulo é obtida:

$$H(z) = 1.1727 \times 10^{-2} \frac{z^4 - 2.8048z^3 + 3.8418z^2 - 2.8048z + 1}{z^4 - 3.6048z^3 + 4.9787z^2 - 3.1156z + 0.7447} \quad (3-69)$$

O módulo e o retardo de grupo da função de transferência (3-69) são mostrados na Fig. 3-27, onde as áreas sombreadas indicam a banda passante. O fator de qualidade  $Q$  na banda passante sem equalização é 64,61.

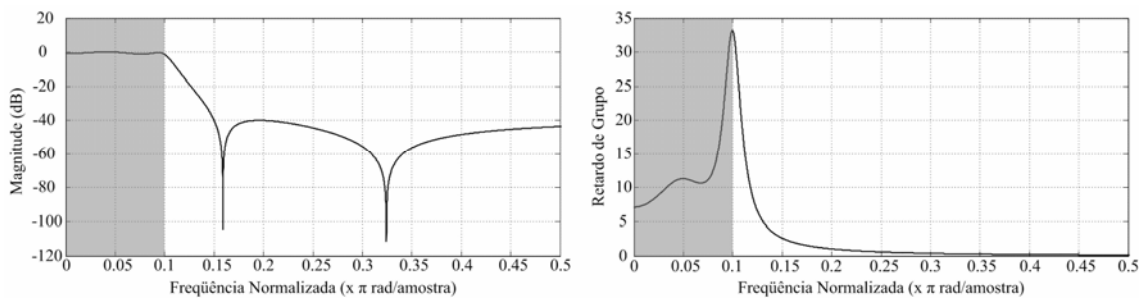


Figura 3-27. Módulo e retardo de grupo do filtro elíptico de quarta ordem.

A primeira etapa do processo de síntese, consistindo do otimizador, é executada. Três seções de funções de transferência passa-tudo de segunda ordem com os coeficientes descritos na Tabela 3-12 são cascadeadas com o filtro elíptico, resultando em um fator de qualidade de 2,10.

Tabela 3-12. Coeficientes das funções de transferência dos filtros passa-tudo de segunda ordem.

	Seção 1	Seção 2	Seção 3
$a_0$	0.856502	0.861843	0.882595
$a_1$	-1.828910	-1.854456	-1.820200

A segunda etapa do processo de síntese, que consiste na busca por estruturas com baixa sensibilidade usando a matriz adjacência, é executada com os parâmetros de entrada do algoritmo evolutivo descritos na Tabela 3-13.

Tabela 3-13. Parâmetros do processo evolutivo.

Tamanho da População	200
Número de Gerações	50
Taxa de Crossover	70
Taxa de Mutação	20
Intervalo dos Multiplicadores	(-4, +4)
Número Máximo de Nós	21
$w_{tf}$	1
$w_{ H_E }$	1
$w_{\tau_E}$	0.5
$E_{tf}$	49

A Fig. 3-28 mostra a aptidão média e a aptidão do melhor indivíduo ao longo das gerações para as três seções do equalizador.

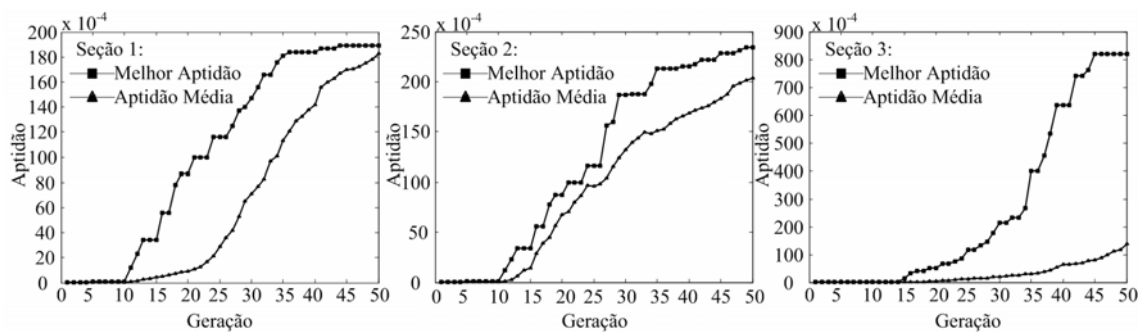


Figura 3-28. Aptidão média e aptidão do melhor indivíduo ao longo das gerações para as três seções de filtros passa-tudo de segunda ordem.

As topologias sintetizadas após 50 gerações são mostradas na Fig.3-29. Estas estruturas empregam o número mínimo de multiplicadores e são ditas canônicas, pois o número de atrasadores é da mesma ordem da função de transferência do filtro. Uma comparação com os resultados obtidos pela realização em forma direta dos equalizadores e as topologias sintetizadas usando matriz adjacência é dada na Tabela 3-14.

Tabela 3-14. Sensibilidades dos equalizadores para a função de transferência (3-69)

	Seção 1		Seção 2		Seção 3	
	$S_{wc}^{ H_E }$	$S_{wc}^{\tau_E}$	$S_{wc}^{ H_E }$	$S_{wc}^{\tau_E}$	$S_{wc}^{ H_E }$	$S_{wc}^{\tau_E}$
Forma Direta	592.21	117.01	1470.85	371.66	182.00	92.21
Topologias Sintetizadas	45.89	5.89	33.65	8.01	7.77	3.42

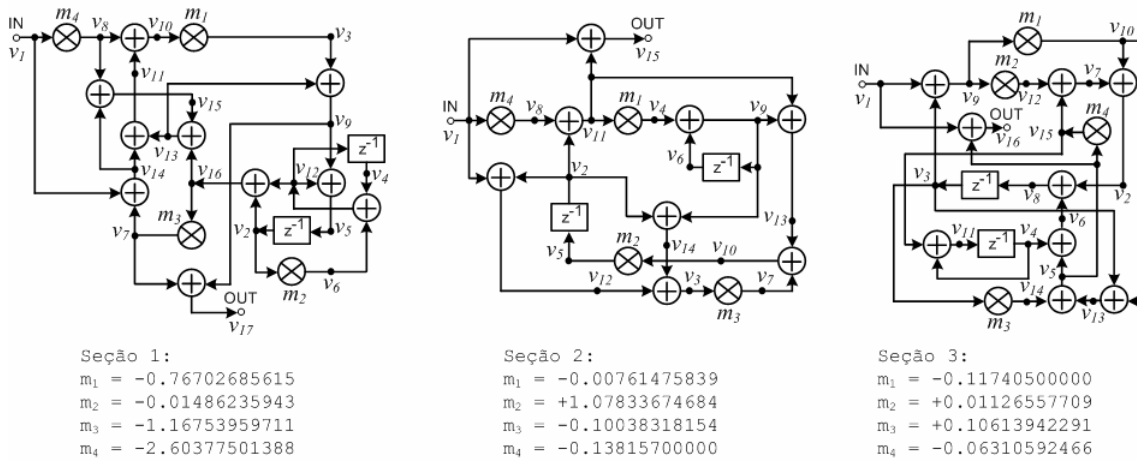


Figura 3-29. Topologias sintetizadas pelo algoritmo evolutivo.

A Fig. 3-30 mostra a função de transferência (3-69) com uma ampliação na banda passante após a equalização usando realização em forma direta e realizações com as topologias sintetizadas para um formato de ponto fixo de 12 bits. Na resposta do retardo de grupo, os fatores de qualidade para as realizações com as topologias sintetizadas e a realização direta foram de 2,47 e 4,10, respectivamente. Portanto, devido à baixa sensibilidade das topologias sintetizadas, o fator de qualidade ainda atende às especificações do fator de qualidade desejado mesmo após a quantização.

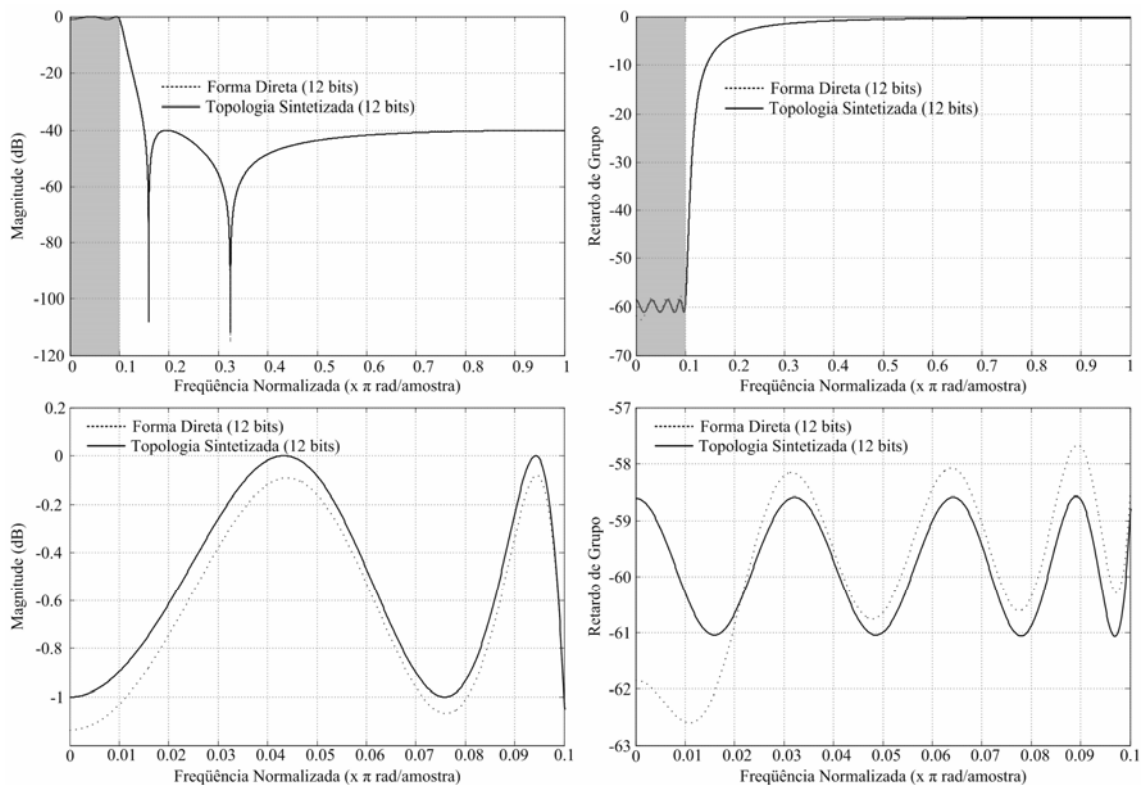


Figura 3-30. Resposta da função de transferência para a realização em forma direta e a sintetizada empregando o algoritmo evolutivo.

## 4 Síntese Evolutiva de Circuitos Analógicos

### 4.1 Introdução

Motivada pelas necessidades impostas pelo mercado, que exigem o projeto de circuitos cada vez mais complexos no menor tempo possível, a indústria se viu obrigada a automatizar ao máximo o projeto de circuitos, desde sua concepção até o produto final. Avanços significativos nas ferramentas de síntese de circuitos digitais com alto nível de abstração foram alcançados nas últimas duas décadas [97], entretanto o mesmo não se verifica igualmente no domínio analógico.

Embora aproximadamente 60% dos ASICs (*Application-Specific Integrated Circuit*) fabricados em tecnologia CMOS incorporem algum circuito analógico [98], geralmente, para interfaceamento com o mundo exterior, a maioria dos projetos de circuitos analógicos ainda depende de engenheiros experientes que possuam conhecimento especializado sobre a classe de problemas a ser projetada. A escassez destes profissionais aliada à crescente demanda de circuitos analógicos contribuem decisivamente para a crescente tentativa da comunidade científica em automatizar as ferramentas de projeto de circuitos analógicos.

Atualmente, o projeto de circuitos analógicos tem se caracterizado por elevados níveis de complexidade, geralmente envolvendo a otimização de funções altamente multimodais. Em muitos casos, a abordagem destes problemas só é possível usando métodos de otimização global [99]. Dentre os métodos de otimização global, os EA têm sido largamente aplicados em problemas complexos de otimização [100-101]. Uma das justificativas do sucesso dos EA na síntese de circuitos é a significativa facilidade de incluir a topologia do circuito como parte integrante do processo de otimização.

Apesar das diversas áreas de aplicação dos EA na eletrônica analógica, é possível separar as aplicações em dois grandes grupos: a síntese evolutiva de circuitos integrados analógicos e a síntese evolutiva de circuitos discretos analógicos.

## 4.2 Circuitos Integrados Analógicos

Uma vez que a tecnologia CMOS é a mais promissora para implementar transistores analógicos programáveis [102], a síntese evolutiva de circuitos integrados analógicos é praticamente dominada por circuitos CMOS.

### 4.2.1 Evolução Intrínseca

A evolução intrínseca de circuitos analógicos pode ser implementada em três níveis: em nível de transistor, em nível de *células* e em nível de *arrays analógicos*.

A implementação em nível de transistor, também conhecida como FPTA (*Field Programmable Transistor Array*) foi proposta em [102]. Ela consiste de 16 x 16 transistores de dimensões programáveis, onde 128 transistores são do tipo-p e 128 do tipo-n, conforme ilustrado na Fig. 4-1.

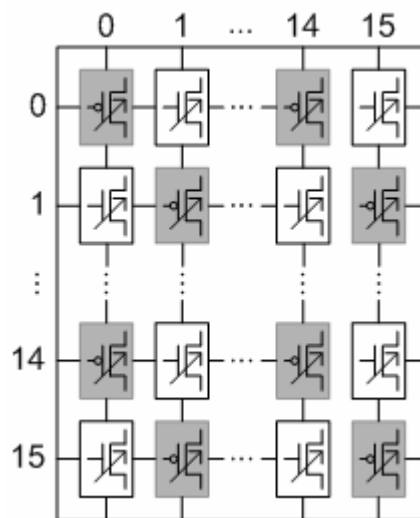


Figura 4-1. Esquemático da FPTA proposta em [102].

Cinco valores de L (0.6, 1, 2, 4, 8 $\mu$ m) e quinze valores de W (1 a 15 $\mu$ m) podem ser programados para cada transistor. Os três terminais dreno, fonte e porta do transistor programável podem ser conectados a seis possíveis pontos: os quatro pontos cardinais (N, S, L, O) e as duas alimentações (Vdd e Gnd). O roteamento dos sinais através do IC (*Integrated Circuit*) é dado pelas seis chaves de roteamento que conectam os quatro pontos cardeais com os demais, conforme ilustrado na Fig. 4-2.

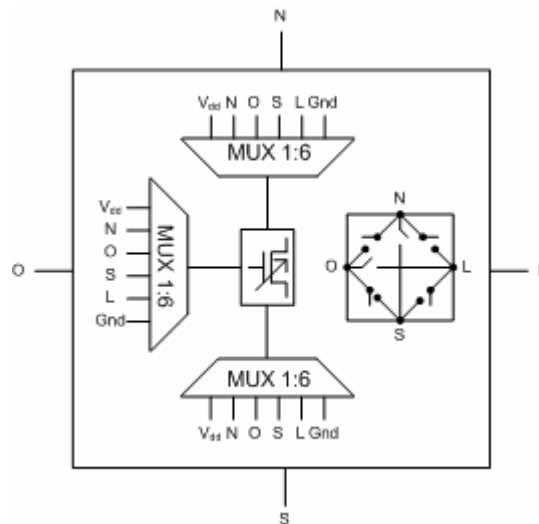


Figura 4-2. Esquema de roteamento dos transistores em [102].

Embora tenha obtido alguns resultados com circuitos em análise DC, a configurabilidade da estrutura proposta insere uma série de resistências e capacitâncias parasitas que limitam a banda passante dos circuitos evoluídos a centenas de KHz.

A implementação em nível de células, foi proposta pelo JPL (*Jet Propulsion Laboratory*) da NASA. Nesta arquitetura não são alterados os valores das dimensões dos transistores. A Fig. 4-3 mostra a arquitetura de uma das 8 x 8 células da versão mais recente do IC fabricado [103]. O circuito reconfigurável consiste de 14 transistores conectados através de 44 chaves e é capaz de implementar uma série de blocos funcionais para processamento de sinais analógicos, desde amplificadores programáveis a fotodetectores logarítmicos. São necessários aproximadamente 5000 bits para programar todo o chip que possui 96 entradas e 64 saídas. O roteamento entre as células utiliza a mesma técnica dos pontos cardeais N, S, L e O.



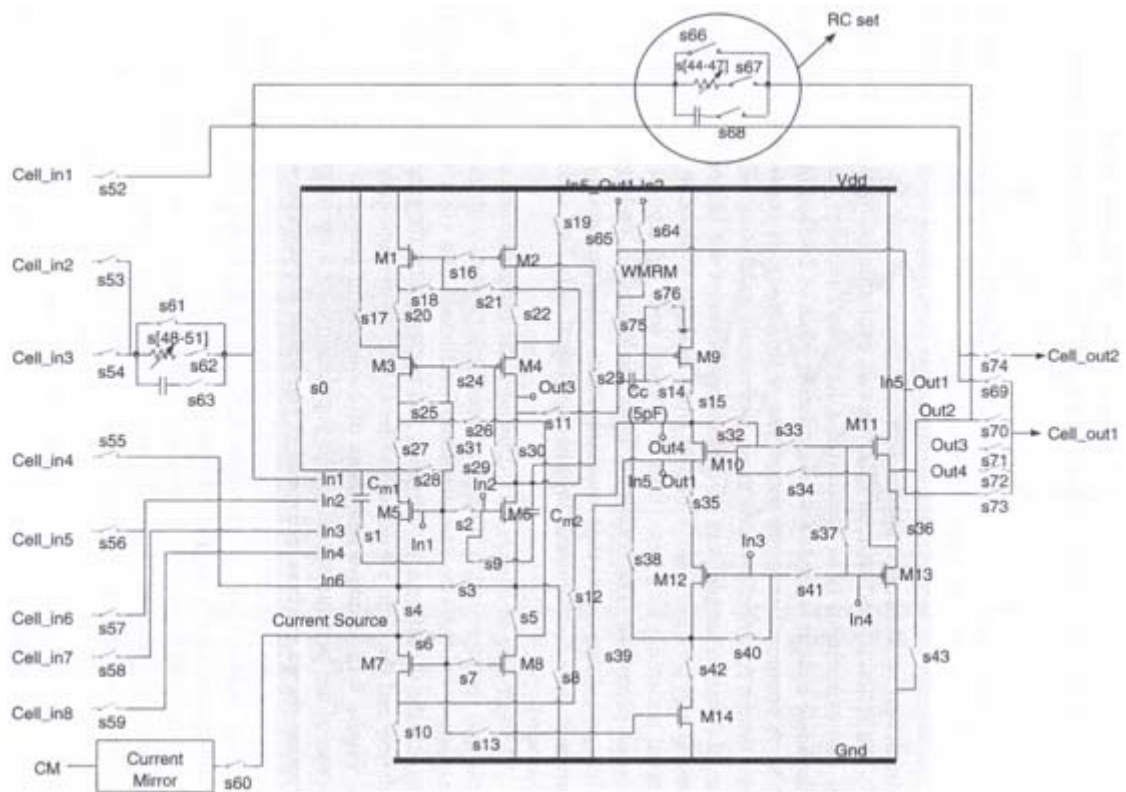


Figura 4-3. Diagrama em blocos da célula proposta em [103].

Nos últimos anos, a implementação utilizando arrays analógicos programáveis têm sido disponibilizada comercialmente, como por exemplo, a série ispPAC da Lattice Semiconductor. Através de uma interface serial padrão é possível configurar circuitos analógicos para condicionamento e filtragem de sinal, conforme mostrado na Fig. 4-4. Em alguns modelos desta família que incorporam DACs e comparadores é possível implementar as funções de conversão e monitoramento de sinais.

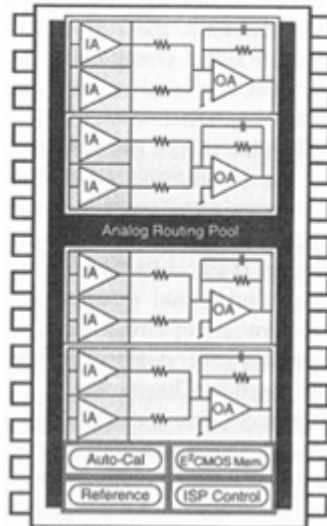


Figura 4-4. Diagrama em blocos interno do ispPAC10.

#### 4.2.2 Evolução Extrínseca

Embora circuitos integrados CMOS possam alcançar altos níveis de integração para componentes ativos, estes circuitos apresentam algumas limitações para implementar componentes passivos, tais como resistores e capacitores. Uma vez que transistores MOS podem implementar, dentro de uma limitada faixa de operação, a funcionalidade de um grande conjunto de componentes, tais como chaves, capacitores e resistores não-lineares, transistores bipolares e diodos, a síntese de circuitos analógicos contendo apenas transistores MOS constitui um relevante ponto que merece investigação.

Embora em [98] os autores tenham inicialmente proposto um algoritmo evolutivo para a síntese de circuitos analógicos discretos, é possível adaptá-lo à síntese de circuitos analógicos integrados CMOS empregando apenas transistores [104-105].

A representação utilizada em [98] é linear e emprega um construtor de circuitos automático, chamado de *cc-bot* (*circuit-constructing robot*), que através de um conjunto de instruções de baixo nível determina como um componente pode ser ligado ao circuito.

A construção da topologia do circuito se inicia com o *nó ativo* sendo o *nó de entrada*. O *nó ativo* é aquele em que o componente será inserido. Todas as instruções,

com exceção da instrução *move-to-new* (MTN), não alteram o *nó ativo*, conforme ilustrado na Tabela 4-1. O *nó de saída* (NS) é o nó de número mais alto e determina o número de nós do circuito.

Tabela 4-1. Conjunto de instruções proposto em [98].

<i>instrução</i>	<i>conexão do terminal de saída</i>	<i>nó ativo</i>
<i>move-to-new</i> (MTN)	à um novo nó é criado	torna-se o nó criado
<i>cast-to-previous</i> (CTP)	ao nó anterior	Inalterado
<i>cast-to-ground</i> (CTG)	à terra	Inalterado
<i>cast-to-input</i> (CTI)	ao nó de entrada	Inalterado
<i>cast-to-output</i> (CTO)	ao nó de saída	Inalterado

A Figura 4-5 ilustra um possível circuito sintetizado pelo *cc-bot*. O conjunto de genes que formam o cromossomo determina a topologia do circuito. O número de genes de um cromossomo é sorteado aleatoriamente, portanto, o tamanho do cromossomo é variável.

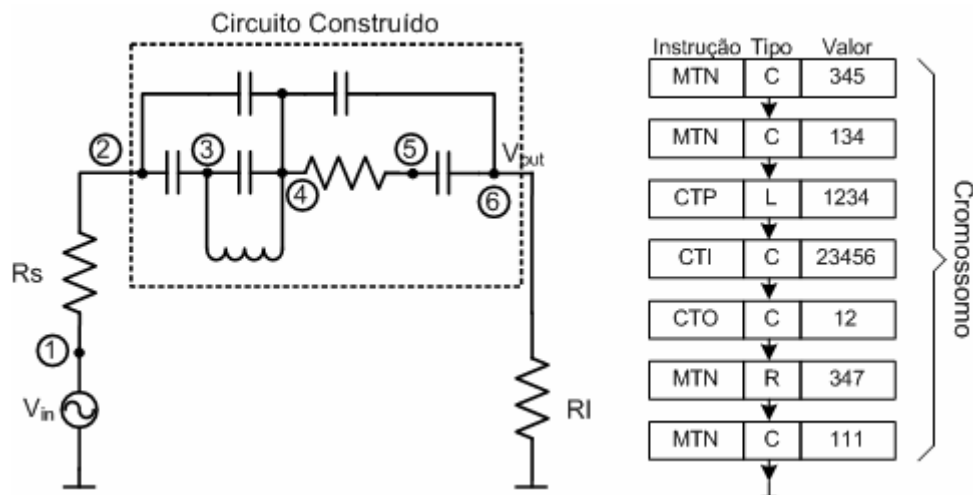


Figura 4-5. Circuito construído pelo *cc-bot* e seu correspondente cromossomo.

O número de campos do gene varia segundo o tipo de circuito a ser sintetizado. No caso da síntese de circuitos com componentes de dois terminais (resistores, capacitores e indutores) cada gene possui três campos, em contrapartida em circuitos contendo transistores bipolares o número de campos aumenta para quatro.

A fim de tratar transistores, que são componentes de três terminais, como componentes de dois terminais os autores em [98] propõem que o terceiro terminal

seja ligado a um nó fixado já existente, tal como: terra, tensão de alimentação (positiva ou negativa), entrada do circuito, saída do circuito, nó anterior ao do terminal de entrada ou nó do terminal de entrada. A Figura 4-6 apresenta as diferenças entre componentes de dois terminais e componentes de três terminais propostos em [98].

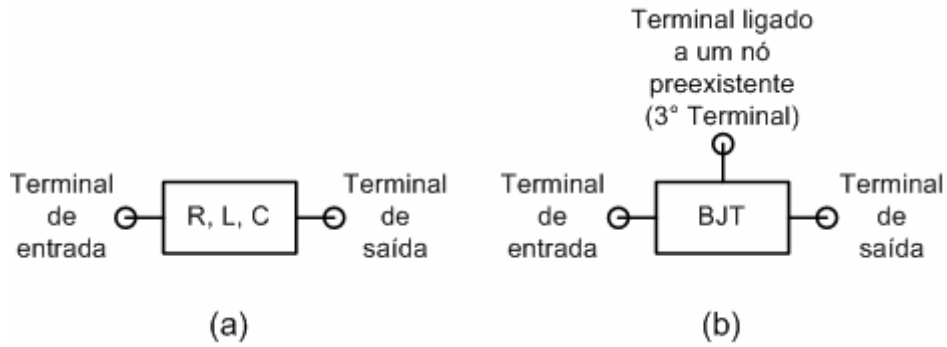


Figura 4-6. (a) Componentes de dois terminais (b) componentes de três terminais convertidos para dois terminais.

A Figura 4-7 exemplifica um possível circuito sintetizado pelo *cc-bot* para um circuito contendo transistores.

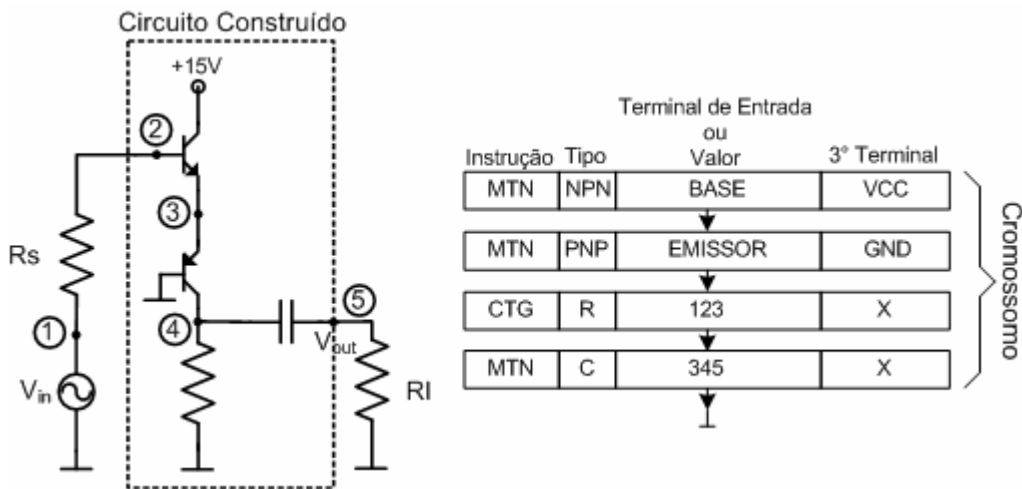


Figura 4-7. Circuito com componentes de três terminais.

A representação proposta, embora flexível, não permite o mapeamento de determinados arranjos de componentes [37]. Para estender o número de possíveis topologias sintetizáveis e permitir a síntese de circuitos MOS foram propostas em [105] três modificações no algoritmo inicialmente concebido.

A primeira limitação consiste no fato de que circuitos que possuem ligações para nós internos não adjacentes não são mapeados com o *cc-bot* original. Como discutido em [104], as topologias de circuitos geradas são limitadas a estruturas do tipo *ladder*. A solução para esta limitação foi a inserção de um parâmetro  $n$  que foi incorporado à instrução CTP. Este parâmetro indica o número de nós a serem decrementados em relação ao *nó ativo* na ligação do terminal do componente a ser inserido. A instrução CTP( $n$ ) permite conexões a todos os nós criados anteriormente, inclusive ao nó de entrada, justificando a remoção da instrução CTI do novo conjunto de instruções original.

A segunda limitação se refere à impossibilidade, usando a representação original, de se criar dois novos nós em um único gene. Desta forma, topologias que possuem *transistores flutuantes*, cujos terminais não estão conectados a nós preexistentes no circuito, não sejam mapeadas pelo *cc-bot* original. Para permitir topologias com transistores flutuantes na síntese de circuitos MOS, uma nova representação para os genes foi adotada, conforme ilustrado na Fig. 4-8.

Tipo	Entrada	Instr_2t	Instr_3t	N_2t	N_3t	W	L
pmos nmos	gate não-gate	MTN, CTO, ...	MTN, CTO, ...	$n_1$	$n_2$	$n_3$	$n_4$

Figura 4-8. Representação do gene para possibilitar topologias com *transistores flutuantes*.

Os campos *Tipo* e *Entrada* especificam, respectivamente, o tipo do transistor (pmos ou nmos) e o terminal de entrada do transistor que será conectado ao *nó ativo*. Uma vez que transistores MOS são dispositivos simétricos, há apenas duas categorias de terminais de entrada a serem consideradas, *gate* ou *não-gate* (dreno ou fonte). Os campos *Instrução\_2t* e *Instrução\_3t* contêm as regras de construção que definem os nós do circuito que serão conectados ao segundo e terceiro terminais do transistor. Esses campos podem assumir qualquer uma das instruções listadas na Tabela 4-2.

Se um ou ambos os campos *Instrução\_2t* e *Instrução\_3t* contiverem uma instrução CTP( $n$ ) o valor de  $n$  será definido pelos campos *N\_2t* e *N\_3t*. As dimensões

dos transistores são atribuídas através dos 16 bits reservados para os valores dos campos  $W$  e  $L$ .

Tabela 4-2. Conjunto de instruções modificado.

<i>instrução</i>	<i>descrição</i>
move-to-new (MTN)	cria um novo nó e conecta um terminal de componente a ele
cast-to-previous (n) (CTP(n))	conecta um terminal de componente ao n-ésimo nó anterior
cast-to-output (CTO)	conecta um terminal de componente ao nó de saída
cast-to-vdd (CVD)	conecta um terminal de componente à alimentação positiva
cast-to-ground (CTG)	conecta um terminal de componente à terra
cast-to-vss (CVS)	conecta um terminal de componente à alimentação negativa
cast-to-bias (CTB)	conecta um terminal de componente à alguma tensão de polarização

A terceira e última modificação consiste na inclusão de uma nova instrução chamada de *cast-to-bias* (CTB) para facilitar a polarização dos circuitos, numa tentativa de aliviar o processo evolutivo. Uma vez sorteada a instrução CTB, um novo sorteio é realizado para determinar qual das seis tensões intermediárias (1,2375V; 0,8250V; 0,4125V; -0,4125V, -0,8250V e -1,2375V) será conectada ao terminal do componente.

#### 4.2.2.1 Estudo de Casos

##### 4.2.2.1.1 Seguidor de Tensão (*Voltage Follower - VF*)

VFs apresentam baixo consumo e respostas em frequência com frequência de corte elevadas, por estas razões estes circuitos têm sido largamente usados como substitutos de blocos funcionais mais complexos para sintetizar filtros [106], osciladores [107] e conversores de impedância [108].

Um VF ideal é definido pelas seguintes condições em seus terminais, conforme ilustrado na Fig. 4-9.

$$i_x = 0 \quad (4-1)$$

$$v_x = v_y \quad (4-2)$$

As equações (4-1) e (4-2) estabelecem, respectivamente, que a impedância de entrada do dispositivo é idealmente infinita e que, para qualquer sinal de entrada, o sinal de saída é igual ao sinal de entrada. Isto significa que a banda passante e a faixa

dinâmica de entrada também sejam infinitas. Outras características desejáveis em um VF prático a ser implementado em um IC são:

- impedância de saída idealmente zero;
- consumo de potência tão baixo quanto possível;
- área reduzida.

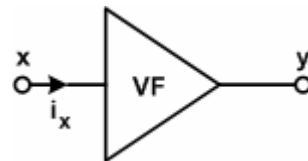


Figura 4-9. Notação simbólica de um VF.

O processo evolutivo proposto em [109] além de levar em consideração todas estas características no cálculo da aptidão, ele propõe uma alteração na representação do gene separando a topologia da sintonia de valores, conforme indicado na Fig. 4-10.

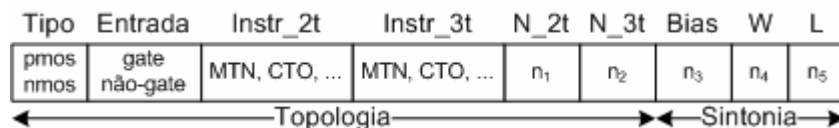


Figura 4-10. Representação do gene ilustrando a separação entre topologia e sintonia.

A natureza epistática do problema de síntese de circuitos analógicos é um fato bem conhecido [22, 104]. Isto significa que o comportamento de qualquer circuito analógico é uma função combinada da topologia e dos valores dos componentes. Portanto, o emprego de algoritmos evolutivos para modificar simultaneamente a topologia e os valores dos componentes de um circuito, possivelmente resultará em um processo evolutivo ineficiente.

Para solucionar este problema, um algoritmo evolutivo realizando a busca topológica através da representação linear descrita anteriormente e a sintonia das dimensões dos transistores e das tensões de gate através de um método de

otimização clássica é utilizado. A Fig. 4-11 mostra o diagrama esquemático do algoritmo utilizado na síntese de um VF usando apenas transistores MOS.

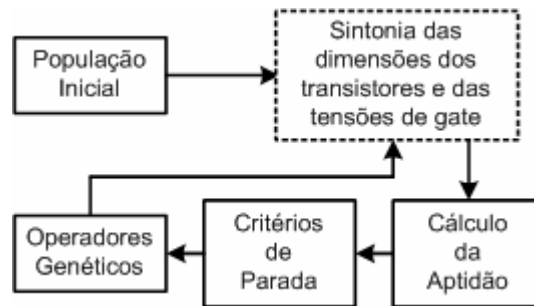


Fig. 4-11. Algoritmo utilizado na síntese de VFs usando apenas transistores MOS.

Uma arquitetura similar foi proposta em [104], contudo a polarização dos gates não foi considerada como parte do passo de sintonia. O problema desta nova abordagem é o aumento do número de redes de polarização devido a introdução da instrução CTB. Para evitar este problema uma função penalidade é inserida no cálculo da aptidão.

O cálculo da aptidão inclui todas as especificações para a implementação prática de um VF em um circuito integrado na tecnologia 0,35 $\mu\text{m}$ : impedância de entrada, impedância de saída, banda passante, característica DC, número de transistores para minimizar a área, consumo e número de tensões de polarização [109].

A Tabela 4-3 ilustra a comparação dos resultados obtidos utilizando o algoritmo e a representação propostos com o desempenho de um VF ideal e com o VF proposto em [110] que utiliza uma rede de compensação com componentes discretos.

Tabela 4-3. Comparação entre um VF ideal, um VF prático proposto em [110] e o VF sintetizado utilizando o algoritmo e a representação propostos.

Especificação	Ideal	[110]	VF Sintetizado
$Z_{in}$ [M $\Omega$ ] @1MHz	$\infty$	80	76
$Z_{out}$ [ $\Omega$ ] @1MHz	0	320	230
BW [MHz]	$\infty$	18	134
Faixa Dinâmica [V]	$\infty$	1.3	1.0
Área [ $\mu\text{m}^2$ ]	0	2500	499
Potência [mW]	0	0.5	0.1



Com relação aos resultados obtidos na tabela, é possível afirmar que as altas impedâncias de entrada não são incomuns e difíceis de serem sintetizadas em circuitos utilizando transistores MOS, uma vez que o terminal de gate fornece um eficiente isolamento elétrico, resultando em correntes de gate extremamente pequenas. Contudo, em projetos envolvendo tecnologia CMOS geralmente os circuitos não precisam oferecer baixa impedância de saída, pois as cargas são predominantemente gates de transistores, que podem ser consideradas aproximadamente como puramente capacitivas.

A potência é cinco vezes menor que a obtida em [110], o mesmo ocorrendo com a área ocupada considerando-se apenas as dimensões dos componentes. Os parâmetros de controle utilizados no processo evolutivo são mostrados na Tabela 4-4.

Tabela 4-4. Parâmetros de controle utilizados na síntese do VF.

População	100
Gerações	100
Taxa de Crossover	60%
Taxa de Mutação	10%
Número de iterações do otimizador	20

A evolução do melhor indivíduo ao longo das gerações para três diferentes configurações do algoritmo é mostrada na Fig. 4-12. Na primeira configuração, sem o passo de sintonia, a topologia e os valores dos componentes são manipulados simultaneamente pelo processo evolutivo. Neste caso, a aptidão do melhor indivíduo praticamente não se alterou ao longo das gerações. Na segunda configuração, a instrução CTB foi eliminada das instruções de construção da topologia. Finalmente, na última configuração, onde a instrução CTB é utilizada e o passo de sintonia é inserido, os resultados são consideravelmente melhores que os obtidos nas duas configurações anteriores.

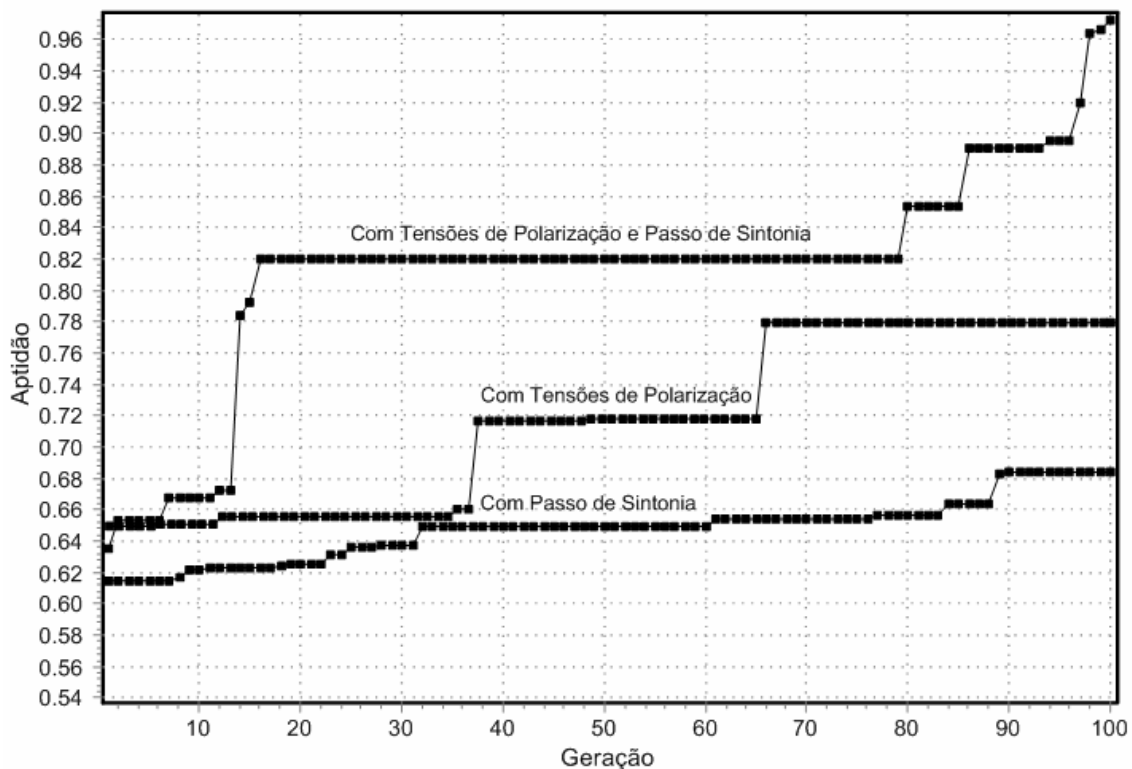


Figura 4-12. Aptidão do melhor indivíduo ao longo das gerações nas três configurações do algoritmo evolutivo proposto.

A Fig. 4-13 ilustra o esquemático do VF sintetizado contendo apenas treze transistores. Para simplificar o esquemático, as conexões dos transistores com o substrato foram removidas. Como esperado, a topologia do circuito sintetizado é inteiramente não convencional.

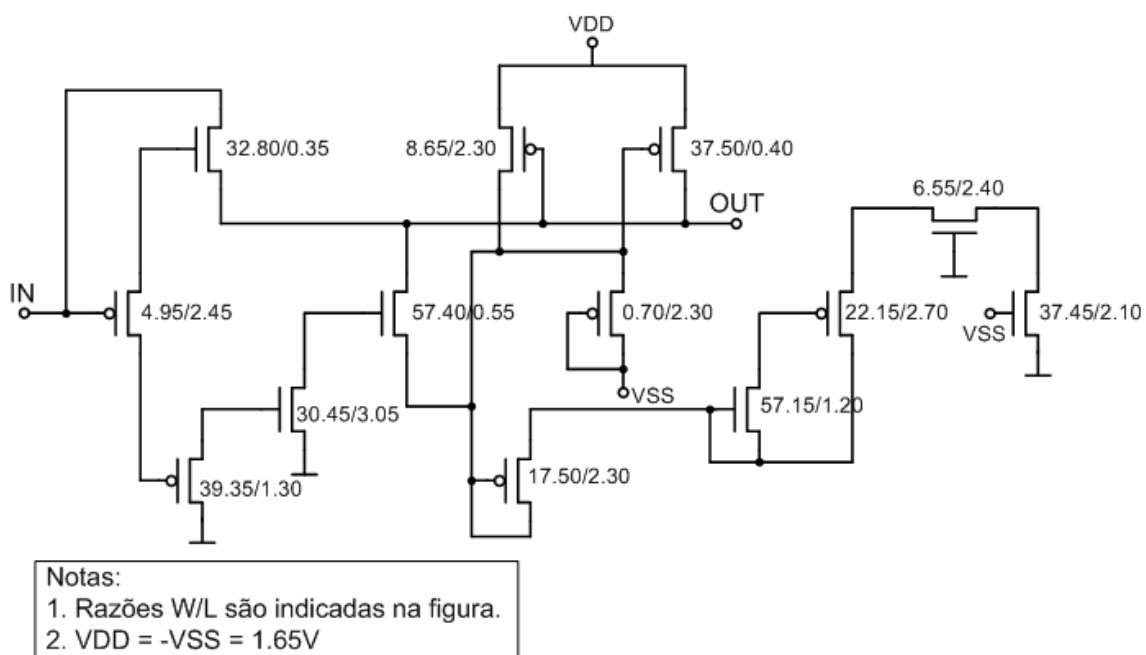


Figura 4-13. VF sintetizado pelo algoritmo evolutivo proposto.

#### 4.2.2.1.2 Amplificador de Transcondutância (OTA) de Miller

O esquemático da Fig. 4-14 mostra o amplificador de transcondutância de Miller com rede de compensação composta por dois componentes passivos – um resistor e um capacitor.

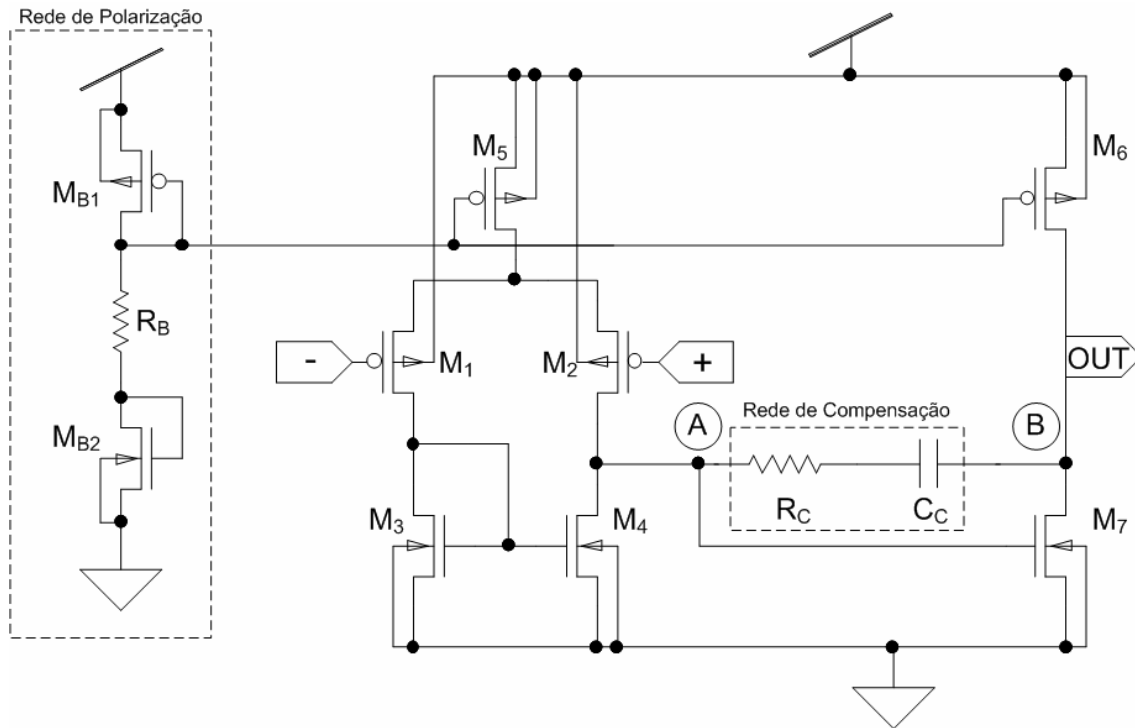


Fig. 4-14. Esquemático do OTA com rede de compensação passiva.

Em tecnologia CMOS, os componentes passivos ocupam grandes áreas e apresentam grandes variações em torno dos valores nominais devido às tolerâncias do processo de fabricação [111].

Com o objetivo de substituir a rede de compensação passiva por transistores, o algoritmo evolutivo discutido nas seções anteriores capaz de construir circuitos contendo apenas transistores MOS foi executado. O circuito evoluído está limitado entre os pontos A e B da Fig. 4-14. Os valores das dimensões dos transistores e de  $R_B$ , localizado exteriormente ao circuito integrado, são ajustados durante o passo de sintonia do algoritmo evolutivo.

A tecnologia utilizada foi a de  $0,35 \mu\text{m}$  com tensão de alimentação simples de 3,3V. O modelo dos transistores utilizado foi o de nível 49 fornecido pela fábrica

responsável pela fabricação de circuitos integrados. Este modelo contempla todos os efeitos de segunda ordem, tais como saturação da velocidade de portadores e efeitos de canal curto [112].

Para avaliar a tolerância do circuito sintetizado às variações do processo de fabricação, o algoritmo avaliou as respostas do circuito utilizando o modelo de transistores para os dois piores casos do processo de fabricação. As três análises DC, AC e TRAN foram consideradas no cálculo da aptidão dos indivíduos. Os circuitos utilizados em cada análise estão mostrados na Fig. 4-15.

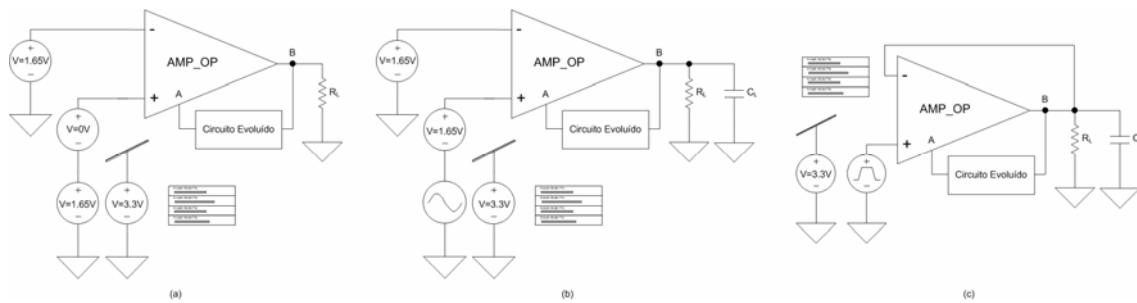


Fig. 4-15. Circuitos para análise (a) DC (b) AC (c) TRAN.

As especificações do circuito a ser sintetizado e a resposta do circuito estão descritas na Tabela 4-5. Estas especificações são particularmente úteis no projeto de *buffers analógicos* para circuitos contendo matrizes de sensores APS (*Active Pixel Sensor*), onde a coexistência de circuitos analógicos e digitais impede a utilização de alimentação simétrica.

Tabela 4-5. Especificações impostas e obtidas pelo circuito sintetizado.

Parâmetro	Análise	Especificado	Obtido
Offset de entrada [V]	DC	<10 $\mu$	5,86
Potência [ $\mu$ W]	DC	<150	139,87
Ganho [dB]	AC	>35	36,43
GBW [MHz]	AC	>60	63,10
Margem de Fase	AC	>45°	46,79°
Slew-rate [V/ $\mu$ s]	TRAN	>20	24,36
$R_L$ [M $\Omega$ ]	--	1	--
$C_L$ [fF]	--	250	--

A topologia do circuito evoluído que substitui a rede de compensação passiva é mostra na Fig. 4-16. A área utilizada por este circuito, desconsiderando a área gasta

pelas interconexões, é de  $103,18 \mu\text{m}^2$ , enquanto nesta mesma tecnologia um capacitor retangular de  $200\text{fF}$  ocupa uma área de  $784,80 \mu\text{m}^2$ . A topologia do circuito evoluído possui vários circuitos em paralelo, indicando a possibilidade de substituição por transistores equivalentes.

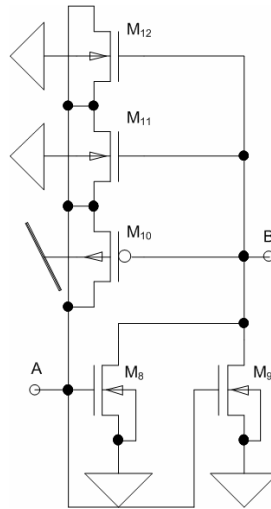


Fig. 4-16. Topologia do circuito evoluído.

A Tabela 4-6 mostra as dimensões de todos os componentes do circuito sintetizado. O valor obtido para o resistor de polarização  $R_B$  foi de  $175\text{k}\Omega$ .

Tabela 4-6. Valores dos componentes.

<i>Componente</i>	<i>W [<math>\mu\text{m}</math>]</i>	<i>L [<math>\mu\text{m}</math>]</i>	<i>Componente</i>	<i>W [<math>\mu\text{m}</math>]</i>	<i>L [<math>\mu\text{m}</math>]</i>
M <sub>1</sub>	8,0	0,4	M <sub>8</sub>	14,0	2,0
M <sub>2</sub>	8,0	0,4	M <sub>9</sub>	3,7	0,8
M <sub>3</sub>	4,0	0,8	M <sub>10</sub>	4,0	2,5
M <sub>4</sub>	4,0	0,8	M <sub>11</sub>	10,0	3,4
M <sub>5</sub>	10,0	0,6	M <sub>12</sub>	16,6	1,7
M <sub>6</sub>	10,0	0,6	M <sub>B1</sub>	10,0	0,6
M <sub>7</sub>	16,0	0,8	M <sub>B2</sub>	8,0	0,8

### 4.3 Circuitos Discretos Analógicos

Circuitos discretos podem ser entendidos por circuitos formados por capacitores, resistores, indutores, transistores bipolares e, no caso da evolução intrínseca, por chaves analógicas que realizam o roteamento dos componentes do circuito.

### 4.3.1 Evolução Intrínseca

Uma plataforma reconfigurável para evolução intrínseca de componentes discretos encontrada na literatura é a PAMA (*Programmable Analog Multiplexer Array*) [113]. A plataforma em sua versão final consiste de 32 multiplexadores 16x1 que possibilitam a conexão de 32 terminais de componentes em 16 nós do circuito. A reconfiguração automática do circuito é conduzida por um GA que especifica através da função objetivo o circuito a ser sintetizado. Circuitos relativamente complexos como controladores fuzzy foram sintetizados usando esta plataforma. A Fig. 4-17 ilustra o diagrama em blocos da arquitetura [114].

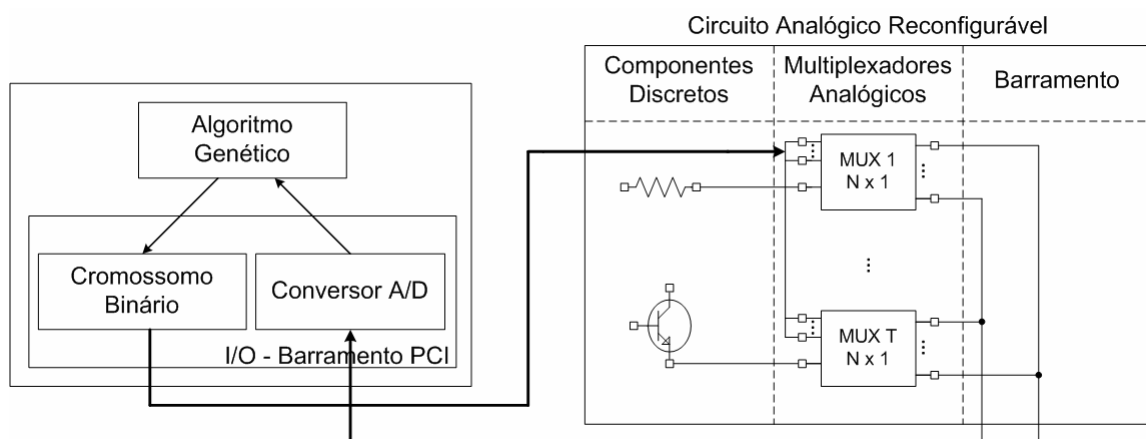


Figura 4-17. Diagrama em blocos da PAMA.

### 4.3.2 Evolução Extrínseca

Na síntese extrínseca de circuitos eletrônicos analógicos, os esquemas de codificação cromossomial podem ser classificados em dois grupos principais: *método de representação direta* e *método de representação indireta* [115].

A forma mais simples e intuitiva de codificação de circuitos analógicos é usar uma representação de tamanho variável onde o cromossomo é composto por uma lista de genes, onde estes últimos representam os componentes e suas respectivas conexões. Cada gene especifica a natureza e o valor dos demais parâmetros associados ao componente. A representação direta [22, 116-117] tem a vantagem da simplicidade na decodificação do cromossomo. No entanto, cada conexão de

componentes é representada no genótipo, com isto o aumento da complexidade dos circuitos provoca um rápido crescimento no tamanho do cromossomo.

Ao contrário dos métodos de representação direta que explicitamente representam os dispositivos e suas conexões, os métodos de representação indireta geram um conjunto de regras que constroem os circuitos eletrônicos. Vários exemplos de métodos de representação indireta na síntese extrínseca de circuitos analógicos são encontrados na literatura [25, 90, 98, 118-119]. A utilização destas representações na síntese de circuitos exige um mapeamento entre o esquema de codificação e o circuito eletrônico. Uma vez que estas abordagens de síntese automática de circuitos ou utilizam métodos que restringem as topologias [25, 98] ou possuem regras complexas de mapeamento [90, 118-119], uma representação direta eficiente de circuitos analógicos pode ser uma importante ferramenta de síntese.

Na próxima seção é proposto um método de representação direta usando a codificação por matriz adjacência para gerar filtros analógicos com baixa sensibilidade.

#### **4.3.3 Síntese Evolutiva de Filtros Analógicos Passivos**

A síntese evolutiva de filtros analógicos foi extensamente abordada na literatura [22, 25, 118, 120]. Contudo, nenhum destes trabalhos investigou a sensibilidade da resposta do circuito em relação aos valores dos componentes. Em [22, 25, 118] os valores dos indutores utilizados no cálculo da resposta em frequência continham quatro casas decimais e como nenhum estudo de sensibilidade foi realizado, é possível que os circuitos sintetizados não atendam às especificações quando implementados na prática. Em [120] foram adotados valores comerciais para os componentes do circuito, mas nenhuma análise da tolerância dos componentes foi realizada, podendo comprometer o desempenho do circuito.

O estudo da sensibilidade é de particular importância uma vez que muitos filtros ativos integrados possuem como base um filtro analógico correspondente. Na maioria

dos projetos de filtros a capacitores chaveados a topologia de uma rede *ladder* passiva serve de base para o projeto do filtro [121-122].

Em particular, uma aplicação de filtros passivos de suma importância em comunicações via radiofrequência é a síntese de equalizadores, também conhecidos como redes de casamento de impedâncias. Estas estruturas são na realidade filtros passivos idealmente sem perdas compostos unicamente por elementos reativos (capacitores e indutores).

Na próxima seção, um método de síntese evolucionária capaz de gerar redes de casamento de impedâncias com baixa sensibilidade é proposto como estudo de caso. Este método utiliza a codificação em matriz adjacência para filtros analógicos. A eficiência do algoritmo proposto é testada na síntese de uma rede de casamento de impedâncias para uma antena real monopolo do tipo “chicote” para a faixa de 64,99-90,00 MHz e os resultados são comparados com o método tradicional de síntese de acopladores de antena encontrado na literatura.

#### **4.3.4 Estudo de Casos**

##### **4.3.4.1 Redes de Casamento de Impedâncias**

Analogamente a reflexão da luz em sistemas ópticos, sinais elétricos são parcialmente refletidos na interface entre dois blocos de circuitos com impedâncias distintas. Portanto, sistemas elétricos eficientes devem incluir redes de casamento de impedâncias a fim de minimizar a perda de potência causada pelo descasamento.

Em termos gerais, o casamento de impedâncias consiste em encontrar uma rede de duas portas linear sem perdas tal que a potência disponível em um gerador resistivo seja entregue a uma carga ao longo de uma dada banda de frequência.

Um problema tradicional e não trivial em comunicações móveis é o casamento de impedâncias entre o estágio de potência do transmissor e a antena de forma a maximizar a potência irradiada.



Na prática, existem duas metodologias de síntese de redes de casamento de impedâncias utilizadas na literatura: a que utiliza a teoria ganho-banda passante [123-124] e a conhecida como Técnica da Freqüência Real (RFT) [125-128]. Contudo, estas teorias apresentam, respectivamente, os seguintes problemas:

- empregam um modelo racional aproximado da carga;
- utilizam topologias fixas de uso geral para implementar a rede de casamento.

Estes dois problemas podem ser simultaneamente resolvidos se uma abordagem evolutiva é empregada. As características da carga podem ser diretamente manipuladas pelo processo evolutivo, evitando erros de aproximação durante a modelagem da carga por funções racionais. Além disso, a exemplo do que foi discutido anteriormente na síntese de filtros digitais, a síntese evolutiva não está restrita à utilização de topologias convencionais, uma vez que a busca da topologia é parte integrante do processo evolutivo de síntese.

#### **4.3.4.1.1 Representação de Redes de Casamento sem perdas Usando Matriz**

##### **Adjacência**

Como discutido no domínio digital para filtros digitais, a representação através da matriz adjacência, além de ser geral e de implementação simples, ela é adequada para redes que contenham elementos com dois terminais. Desta forma, a representação de redes de casamento de impedâncias sem perdas, onde apenas componentes de dois terminais (capacitores e indutores) são utilizados, constitui-se uma possível aplicação da matriz adjacência no domínio analógico.

Para tanto, a matriz adjacência precisa ser redefinida, substituindo os curto-circuitos por ligações à terra.

Seja  $G(V, E)$  com  $V = \{v_0, \dots, v_n\}$  um grafo orientado sem ramos em paralelo representando a topologia de um circuito e o nó  $v_0$  o nó de terra do circuito. A matriz adjacência reduzida  $A = [a_{ij}]$  de  $G$  é uma matriz  $n \times n$  com  $a_{ij}$  definido como [86]:

$$\forall i \neq j \neq 0$$

$$a_{ij} = \begin{cases} 1, & \text{if } (v_i, v_j) \in E \\ 0, & \text{outros casos} \end{cases} \quad (4-3)$$

$$a_{ii} = 1 \quad \text{if } (v_i, v_0) \in E$$

A Fig. 4-18 mostra um exemplo de um grafo representando um circuito analógico e sua respectiva matriz adjacência reduzida. Os curto-circuitos são substituídos por ramos para a terra. Os ramos  $\{e_1, e_2, e_3\}$  são as adjacências para o nó de terra que são representadas por elementos não-nulos na diagonal principal da matriz.

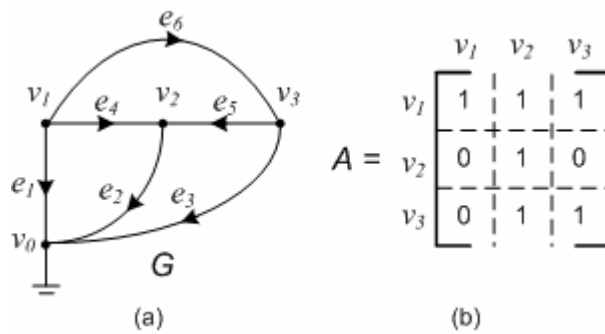


Figura 4-18. Representação da matriz adjacência para circuitos analógicos (a) grafo orientado (b) matriz adjacência reduzida correspondente.

Se a convenção de codificação adotada for 1 para capacitores e 2 para indutores é possível, por exemplo, representar o circuito sintetizado em [25] utilizando a matriz adjacência reduzida conforme indicado na Fig. 4-19.

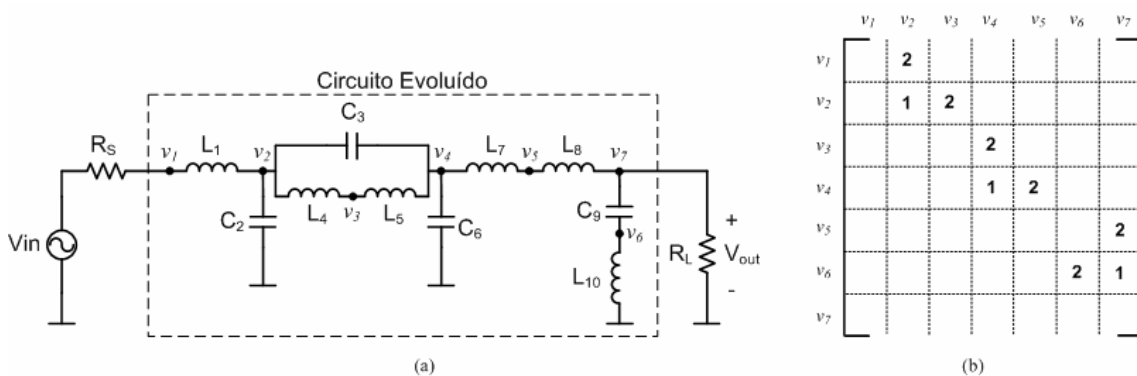


Figura 4-19. (a) Filtro passa-baixas sintetizado em [25] (b) matriz adjacência reduzida correspondente.

Existem duas limitações na definição da matriz adjacência reduzida quando se deseja representar ramos em paralelo em uma rede de casamento de impedâncias sem perdas.

A primeira limitação ocorre em estruturas que possuem mais de dois componentes ligados em paralelo em dois nós quaisquer  $v_i$  e  $v_j$  ( $i \neq j$ ) do circuito. Esta limitação pode ser contornada, tendo em vista que componentes de mesma natureza ligados em paralelo podem ser associados. Desta forma, topologias originalmente com mais de dois componentes em paralelo podem ser representadas por topologias equivalentes com apenas dois componentes em paralelo, conforme mostrado na Fig. 4-20.

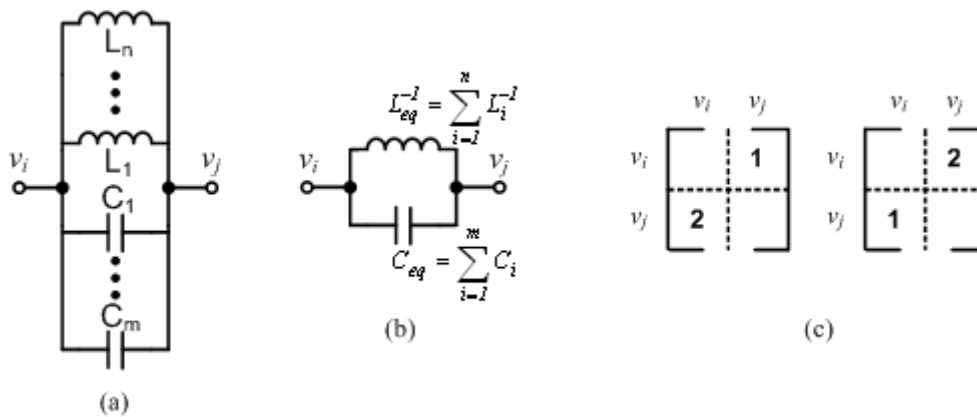


Figura 4-20. (a) Topologia com  $n+m>2$  componentes em paralelo (b) topologia equivalente com apenas dois componentes em paralelo (c) possíveis matrizes adjacência reduzidas representando a topologia equivalente.

Do ponto de vista topológico, os circuitos da Fig. 4-20(a)-(b) são equivalentes, ou seja, basta que a representação proposta consiga mapear dois componentes em paralelo de naturezas diferentes para ser capaz de representar qualquer número de componentes em paralelo. A diferença entre os circuitos da Fig. 4-20(a)-(b) está nos valores dos componentes que, a exemplo do caso de circuitos analógicos CMOS (vide Fig. 4-11), podem ser sintonizados em um passo específico durante o processo evolutivo de síntese.

A segunda limitação imposta pela definição da matriz adjacência reduzida ocorre em estruturas que possuem mais de um componente em paralelo ligado à terra. Neste caso, torna-se necessário codificar o paralelo de dois componentes de natureza diferentes como 3 (C//L=3). Da mesma forma que descrito anteriormente para ligações de mais de dois componentes flutuantes (não ligados à terra) em paralelo, a associação de componentes de mesma natureza em paralelo ligados à terra pode ser utilizada, conforme indicado na Fig. 4-21.

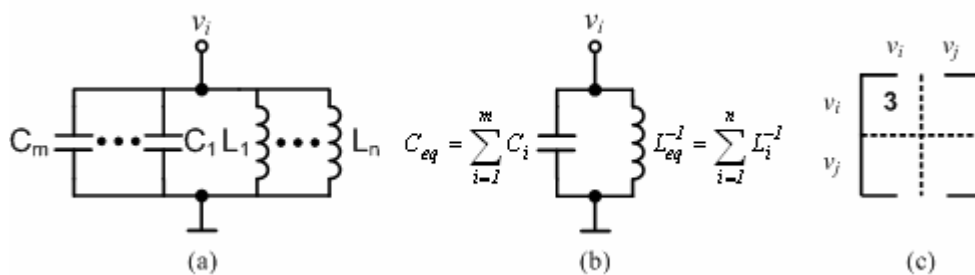


Figura 4-21. (a) Topologia com  $n+m>1$  componentes de natureza diferentes ligados à terra em paralelo (b) topologia equivalente com apenas dois componentes em paralelo (c) matriz adjacência reduzida representando a topologia equivalente.

Uma consequência da inclusão do paralelo de componentes de natureza distinta (C//L=3) na codificação da matriz é o surgimento de representações equivalentes para uma mesma topologia, como indicado na Fig. 4-22. A mesma topologia pode ser representada de quatro maneiras diferentes de acordo com a codificação proposta. Desta forma, mais de um genótipo ou matriz mapeia o mesmo fenótipo ou topologia do circuito, permitindo a existência de uma “neutralidade topológica” [130].

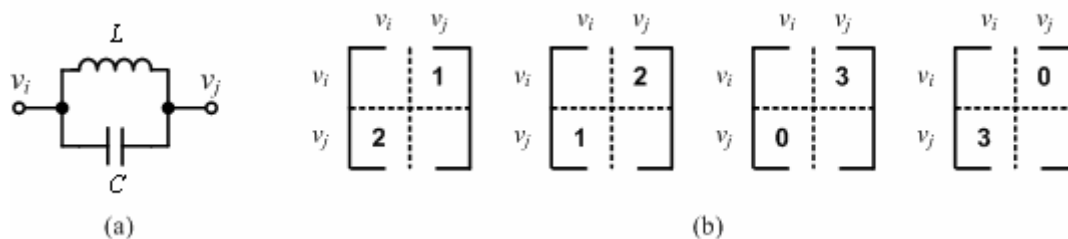


Figura 4-22. (a) Circuito contendo dois componentes flutuantes em paralelo (b) possíveis matrizes adjacência reduzidas correspondentes representando o circuito.

Para representar completamente uma rede de casamento de impedâncias é necessário que a topologia, a natureza e os valores de componentes sejam representados. Para este fim, uma estrutura de dados consistindo de uma matriz tridimensional foi empregada como ilustra a Fig. 4-23. Na matriz, a primeira dimensão define a topologia da rede de acordo com (4-6) e as outras duas dimensões da matriz representam, respectivamente, os valores dos capacitores e dos indutores.

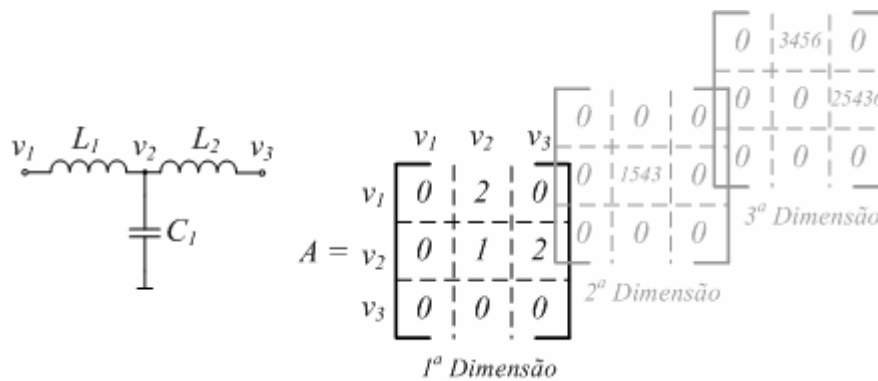


Figura 4-23. Representação completa de uma rede de casamento de impedâncias.

Os valores dos componentes (capacitores e indutores) estão restritos a valores práticos definidos no início do processo evolutivo e são normalizados linearmente de maneira a serem representados por uma matriz de números inteiros positivos. Como um exemplo do procedimento de normalização, suponha que o valor do capacitor  $C_1$  na Fig. 4-23 esteja restrito ao intervalo  $[10\text{ pF}, 100\text{ pF}]$ . Então, o valor 1543 na matriz de capacitores (2ª Dimensão) corresponde a um valor de capacitância de:

$$C_1 = \frac{(100 - 10) \cdot (1543 - 0)}{65535 - 0} + 10 = 12,12\text{ pF} \quad (4-4)$$

Por fim, é necessário notar que o tamanho do genótipo, ou seja, a dimensão da matriz pode variar até um valor máximo definido no início do processo evolutivo, configurando que a representação é uma representação que utiliza cromossomos de tamanho variável.

#### 4.3.4.1.2 Operadores Genéticos

As estratégias de crossover e mutação empregadas na síntese de redes de casamento de impedâncias são análogas às utilizadas na síntese de filtros digitais descritas na Seção 3.4.6. A diferença é que no caso das redes de casamento, a matriz adjacência representa não só a topologia do circuito e a natureza dos componentes como também os valores dos componentes. Conseqüentemente, as operações genéticas no caso das redes de casamento alteram a topologia, a natureza e os valores dos componentes, conforme ilustrado na Fig. 4-24. Neste exemplo, as dimensões das matrizes, as coordenadas dos pontos de crossover e as dimensões das submatrizes são, respectivamente,  $m = 3$ ,  $n = 4$ ,  $j = k = l = 2$ ,  $p = 1$  e  $q = 2$ .

A mutação é um caso particular do crossover, onde  $p = 1$  e  $q = 1$ .

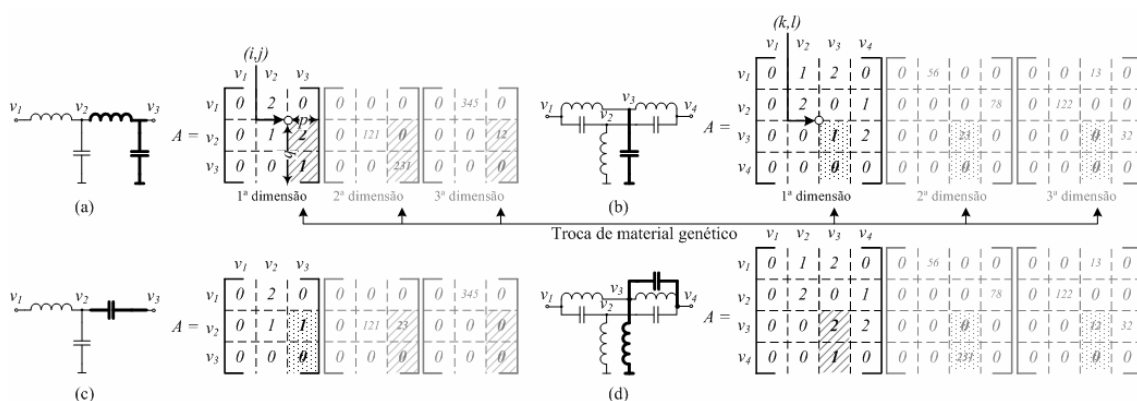


Figura 4-24. Exemplo de crossover entre duas redes de casamento de impedâncias (a) indivíduo 1 (b) indivíduo 2 (c-d) indivíduos resultantes do crossover.

#### 4.3.4.1.3 Circuitos Inválidos

O emprego de componentes reativos ideais sem perdas na síntese de redes de casamento de impedâncias pode gerar duas categorias de circuitos inválidos ou não-simuláveis.

A primeira categoria de circuitos inválidos é constituída de circuitos que contém ciclos indutivos, conforme indicado na Fig. 4-25(a). A solução para este problema é a inclusão de resistores de valores baixos em série com os indutores, conforme ilustrado

na Fig. 4-25. O valor escolhido para a resistência em série foi de  $R_s = 1\mu\Omega$ , de maneira a não influir significativamente no comportamento do circuito.

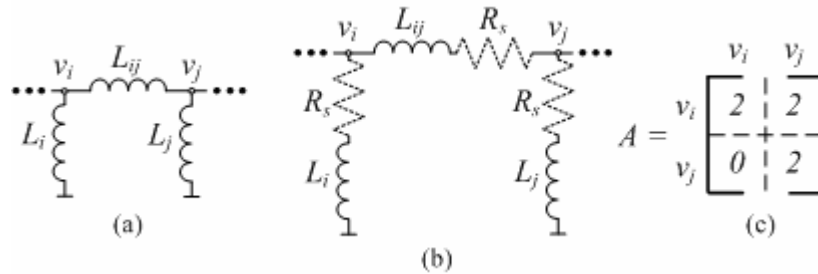


Figura 4-25. (a) Circuito contendo um ciclo indutivo (b) indutores com resistores em série para evitar a geração de indivíduos inválidos (c) matriz adjacência sem representar os resistores.

Os resistores não são representados na matriz adjacência e a inclusão de resistências em série com os indutores já é prevista na descrição de indutores não-ideais em simuladores de circuitos compatíveis com o SPICE (*Simulation Program with Integrated Circuit Emphasis*). Um exemplo de um indutor não ideal de  $10nH$  ligado entre os nós  $n1$  e  $n2$  é:

```
L1 n1 n2 10nH R=1u
```

A segunda categoria de indivíduos inválidos é constituída de circuitos que não “possuem caminho DC para terra”. Este problema decorre do fato de que em uma análise de ponto de operação DC todos os capacitores são abertos. Uma solução análoga àquela utilizada para os indutores é também empregada para os capacitores, com a diferença que para o caso dos capacitores, uma condutância é inserida em paralelo com cada capacitor do circuito, como ilustrado na Fig. 4-26.

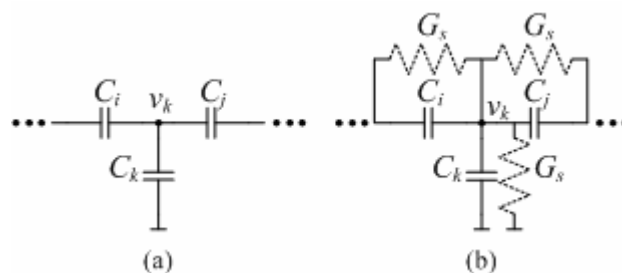


Figura 4-26. (a) Circuito ilustrativo, mostrando que o nó  $v_k$  não possui caminho DC para terra (b) condutâncias inseridas em paralelo com cada capacitor para garantir um caminho DC para terra.

Na realidade, a implementação prática em simuladores compatíveis com o SPICE utiliza o comando DCSTEP, este comando inclui condutâncias em paralelo com cada capacitor durante a análise DC. Um exemplo de utilização deste comando é descrito a seguir, onde condutâncias de  $1p\mathcal{O}$  são incluídas em paralelo com os capacitores do circuito durante a análise do ponto de operação DC:

```
.OPTION DCSTEP=1e-12
```

#### 4.3.4.1.4 Figura de Mérito

O objetivo da rede de casamento de impedâncias é transferir o máximo da potência disponível de uma fonte para uma dada carga ao longo da banda de freqüência de interesse como mostrado na Fig. 4-27. A rede de casamento a ser sintetizada é uma rede de duas portas localizada entre o estágio de potência de um transmissor e a antenna. A resistência  $R_{o1}$  representa a resistência de saída do estágio de potência e a impedância  $Z_{o2}$  representa a impedância da antenna.

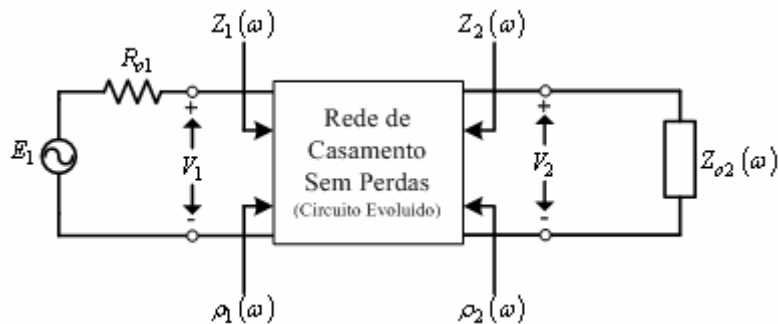


Figura 4-27. Rede de casamento de impedâncias.

O problema do casamento de impedâncias pode ser formulado em termos da razão entre a potência disponível na fonte e a potência entregue a carga. Esta razão é mais conhecida como Ganho de Potência de Transdução (TPG). O TPG particularmente para rede sem perdas pode ser escrito como [129]:

$$TPG = 1 - |\rho_1(\omega)|^2 \quad (4-5)$$



onde  $\rho_1(\omega)$ , o coeficiente de reflexão da porta 1, é uma função da impedância de entrada  $Z_1(\omega)$  da porta 1:

$$\rho_1(\omega) = \frac{Z_1(\omega) - R_{o1}}{Z_1(\omega) + R_{o1}} \quad (4-6)$$

A qualidade de uma rede de casamento de impedâncias pode ser descrita em termos da Razão de Voltagem da Onda Estacionária (VSWR):

$$VSWR = \frac{1 + |\rho_1(\omega)|}{1 - |\rho_1(\omega)|} \quad (4-7)$$

O VSWR representa a interação entre os sinais transmitidos e os refletidos causados por um eventual descasamento de impedâncias. Idealmente, em um casamento perfeito ( $\rho_1 = 0$ ), não há sinal refletido e, neste caso,  $VSWR=1$ .

Substituindo (4-7) em (4-5) é possível exprimir o TPG em função do VSWR:

$$TPG = \frac{4 \cdot VSWR}{(1 + VSWR)^2} \quad (4-8)$$

Derivando (4-8):

$$\frac{dTPG}{dVSWR} = \frac{4 \cdot (1 - VSWR^2)}{(1 + VSWR)^4} \quad (4-9)$$

Como para circuitos práticos  $VSWR \in ]1, +\infty[$ , de (4-9) é possível assegurar que  $\frac{dTPG}{dVSWR} < 0$ . Portanto, maximizar o TPG é equivalente a minimizar o VSWR e este último pode ser utilizado como uma figura de mérito no casamento de impedâncias de redes sem perdas.

#### 4.3.4.1.5 Cálculo da Aptidão

As principais especificações que a rede de casamento de impedâncias sem perdas deve atender são:

- VSWR o mais próximo de 1 ao longo de uma banda de frequências;
- baixa sensibilidade;
- valores práticos de componentes.

A fim de sintetizar circuitos sujeitos às especificações acima, a aptidão é definida como o inverso da função erro  $\varepsilon$  dada por:

$$\varepsilon = I + w_{VSWR} \cdot \varepsilon_{VSWR} + w_{Sens} \cdot \varepsilon_{Sens} + w_{PF} \cdot \varepsilon_{PF} \quad (4-10)$$

Nesta equação  $\varepsilon$  é o erro total,  $\varepsilon_{VSWR}$  é o erro com relação ao descasamento de impedâncias,  $\varepsilon_{Sens}$  é o erro referente à sensibilidade da rede,  $\varepsilon_{PF}$  é o erro dos valores de componentes e  $w_i$  s são os pesos.

#### A. Cálculo de $\varepsilon_{VSWR}$

Antes de se obter o erro,  $\varepsilon_{VSWR}$ , é necessário calcular o valor de VSWR para o circuito da Fig. 4-27. Uma análise AC de pequenos sinais é realizada com o circuito, onde a impedância da antena,  $Z_{o2}$ , é amostrada em um conjunto discreto de freqüências ao longo da banda de interesse. Mais especificamente, uma tabela contendo as partes real e imaginária de  $Z_{o2}$  em cada freqüência amostrada é fornecida ao SPICE.

Contudo, é possível substituir o circuito da Fig. 4-27 pelo circuito equivalente da Fig. 4-28, onde a dependência da freqüência é omitida para simplificar a notação.

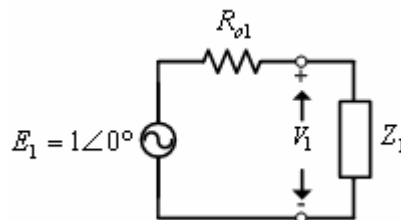


Figura 4-28. Circuito equivalente utilizado para o cálculo de VSWR.

O circuito equivalente da Fig. 4-28 é obtido substituindo o circuito evoluído e a impedância de carga pela impedância de entrada da porta 1. Uma vez que em uma análise AC de pequenos sinais o sinal senoidal de entrada  $E_i$  tem magnitude um e fase zero, a voltagem na porta 1,  $V_1$ , pode ser expressa como:

$$V_1 = \frac{Z_1}{Z_1 + R_{o1}} \quad (4-11)$$

Combinando as equações (4-6), (4-7) e (4-11), o coeficiente de reflexão e o VSWR podem ser, respectivamente, escritos na forma:

$$|\rho_I| = |2V_I - I| \quad (4-12a)$$

$$VSWR = \frac{I + |2V_I - I|}{I - |2V_I - I|} \quad (4-12b)$$

Assim, segundo (4-12b), os valores de VSWR ao longo da banda de interesse podem ser calculados a partir dos valores de  $V_I$  da Fig. 4-27.

De posse dos valores de VSWR, um critério *minimax* é usado para calcular o valor do erro  $\varepsilon_{VSWR}$ :

$$\varepsilon_{VSWR} = \min \left\{ \max_{\omega \in \Omega} VSWR \right\} \quad (4-13)$$

onde  $\Omega = [\omega_{\min}, \omega_{\max}]$  denota a banda de frequência de interesse.

## B. Cálculo de $\varepsilon_{Sens}$

Baixa sensibilidade da rede de casamento com relação aos valores dos componentes é uma condição necessária para a implementação prática dos circuitos. Uma vez que no caso das redes de casamento de impedâncias, a TPG pode ser convenientemente expressa em termos do VSWR, a sensibilidade do VSWR com relação aos componentes pode ser usado como figura de mérito.

Considere uma rede de casamento com  $m$  parâmetros  $p = [p_1, p_2, \dots, p_m]$  onde os  $p_i$ s são os valores dos componentes da rede. O vetor de sensibilidade da rede pode ser definido como:

$$S_p^{VSWR} = \left[ S_{p_i}^{VSWR} \right]_{i=1,m} \quad (4-14)$$

onde a sensibilidade do VSWR com relação a cada componente  $p_i$  é definida como [131]:

$$S_{p_i}^{VSWR} = \frac{p_i}{VSWR} \cdot \frac{\partial VSWR}{\partial p_i} \quad (4-15)$$

Aplicando a regra da cadeia, (4-15) pode ser escrita na forma:

$$S_{p_i}^{VSWR} = \frac{p_i}{VSWR} \cdot \frac{\partial VSWR}{\partial |\rho_1|} \cdot \frac{\partial |\rho_1|}{\partial p_i} \quad (4-16)$$

Após substituir VSWR em termos de  $|\rho_1|$  e  $p_i$  por seu valor dado em (4-12a), a equação (4-16) pode ser escrita em termos de  $V_1$  como:

$$S_{p_i}^{VSWR} = \frac{4p_i}{|2V_1 - 1|(1 - |2V_1 - 1|^2)} \cdot \Re \left[ (2V_1^* - 1) \cdot \frac{\partial V_1}{\partial p_i} \right] \quad (4-17)$$

onde  $V_1^*$  é o complexo conjugado de  $V_1$  e  $\Re[\bullet]$  é a parte real de uma variável complexa.

Em (4-17), o único termo a ser determinado é a derivada parcial de  $V_1$  com relação a  $p_i$ . Para este fim, seja o circuito da Fig. 4-27 descrito por um sistema de equações nodais [132]:

$$\mathbf{Y} \cdot \mathbf{V} = \mathbf{J} \quad (4-18)$$

onde  $\mathbf{Y}$ ,  $\mathbf{V}$  e  $\mathbf{J}$  são, respectivamente, a matriz admitância nodal, o vetor nodal de voltagens e o vetor de fontes de corrente nodais equivalentes. Neste caso, o termo  $\frac{\partial V_1}{\partial p_i}$  em (4-17) pode ser expresso como [132]:

$$\frac{\partial V_1}{\partial p_i} = (\mathbf{V}^a)^t \cdot \frac{\partial \mathbf{Y}}{\partial p_i} \cdot \mathbf{V} \quad (4-19)$$

onde  $\mathbf{V}^a$  é o vetor adjunto do vetor  $\mathbf{V}$  que é dado por:

$$\mathbf{Y}^t \cdot \mathbf{V}^a = -\mathbf{d} \quad (4-20)$$

onde  $\mathbf{d}$  é o vetor:

$$d_i = \begin{cases} 1, & \text{se } i \text{ é o nó de entrada cuja} \\ & \text{sensibilidade é calculada} \\ 0, & \text{outros casos} \end{cases} \quad (4-21)$$

Levando em conta que para uma rede sem perdas  $\mathbf{Y}^t = \mathbf{Y}$  e comparando (4-18) com (4-20), é possível verificar que o vetor coluna  $\mathbf{V}^a$  pode ser obtido através de um análise AC de pequenos sinais do circuito mostrado na Fig. 4-29.



Figura 4-29. Circuito utilizado para a obtenção do vetor adjunto  $V^a$ .

O último termo para calcular (4-19) é  $\frac{\partial Y}{\partial p_i}$ . Em uma rede sem perdas, existem apenas dois possíveis casos como mostrado na Tabela 4-7.

Tabela 4-7. Derivadas parciais de  $Y$  com relação a  $p_i$ .

Natureza do Componente	$\frac{\partial Y_{kl}}{\partial p_i}$
$p_i \equiv \text{capacitor}$	$j\omega$
$p_i \equiv \text{indutor}$	$-\frac{1}{j\omega L_{kl}^2}$

Analogamente ao caso anterior, o critério de *minimax* é utilizado no cálculo do do erro de sensibilidade:

$$\mathcal{E}_{Sens} = \min \left\{ \sum_{i=1}^m \max_{\omega \in \Omega} |S_{p_i}^{VSWR}| \right\} \quad (4-22)$$

onde  $\Omega = [\omega_{\min}, \omega_{\max}]$  denota a banda de frequência de interesse.

#### B. Cálculo de $\mathcal{E}_{PF}$

De maneira a restringir os valores dos componentes do circuito sintetizado a valores práticos, uma função penalidade é utilizada [133]:

$$\mathcal{E}_{PF} = \lambda \cdot \sum_{i=1}^m d_i \quad (4-23)$$

onde  $\lambda$  e  $d_i$  são, respectivamente, uma constante definida pelo usuário e uma distância métrica para a  $i$ -ésima restrição.

#### 4.3.4.1.6 Visão Geral do Algoritmo Evolutivo Proposto

O algoritmo evolutivo proposto realiza a busca pela topologia utilizando a representação em matriz adjacência e a sintonia dos valores dos componentes através de uma combinação de um GA, descrito na Seção 4.3.4.1.2, e um método

convencional de otimização. O papel do GA no processo de sintonia é localizar uma região promissora dentro do espaço de busca, enquanto a otimização local explora mais precisamente a melhor solução dentro da região inicialmente selecionada pelo GA. O diagrama em blocos do algoritmo é mostrado na Fig. 4-30.

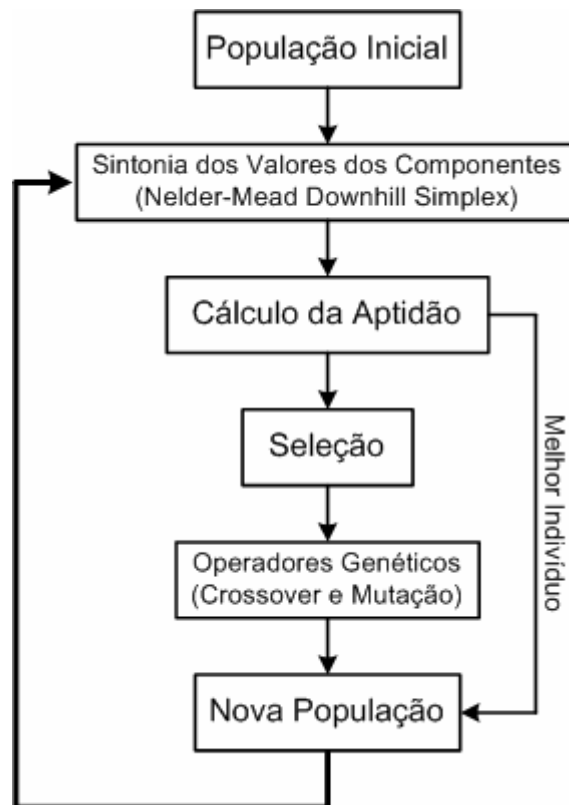


Figura 4-30. Algoritmo evolutivo usado na síntese de redes de casamento de impedâncias incluindo o passo de sintonia dos valores dos componentes.

O algoritmo é iniciado com uma população aleatória de matrizes tridimensionais. Todos os indivíduos da população inicial são otimizados pelo passo de sintonia de valores usando o algoritmo Nelder-Mead Simplex, que possui a vantagem de não usar derivadas. Em seguida, uma nova população é gerada pela aplicação do mecanismo de seleção e dos operadores genéticos à população.

De forma a reduzir o esforço computacional, uma variável Booleana é associada a cada indivíduo indicando se o indivíduo sofreu ou não crossover ou mutação. Assim, somente os indivíduos que foram afetados pelos operadores genéticos serão otimizados no passo de sintonia.

#### 4.3.4.1.7 Resultados Obtidos

A fim de testar a eficácia do algoritmo proposto, uma rede de casamento de impedâncias deve ser sintetizada pelo processo evolutivo entre um transmissor de potência e uma antena monopolo real. O objetivo da rede é casar a impedância de  $50\Omega$  do estágio de potência com a impedância da antena do tipo “chicote” utilizada em rádios portáteis na faixa de VHF. A impedância de radiação da antena foi experimentalmente obtida, usando um Analisador de Rede de  $50\Omega$  em 83 frequências na banda de 64,99-90,00 MHz, em passos de 305 KHz como mostra a Fig. 4-31.

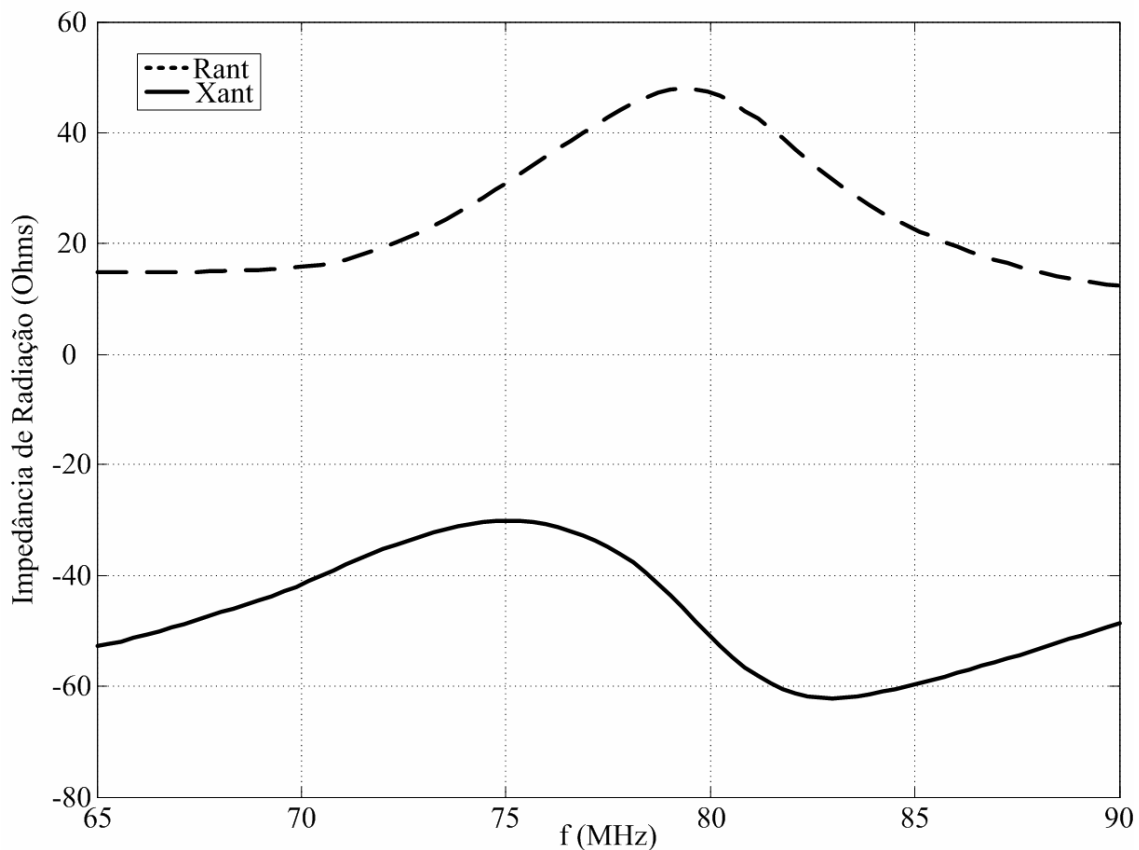


Figura 4-31. Impedância de radiação da antena --- Resistência (Rant) e — Reatância (Xant).

Antes de discutir os resultados obtidos com o processo evolutivo aplicado ao problema proposto, é de interesse, face a viabilizar posteriores comparações, descrever o método de síntese de redes de casamento de impedâncias consagrado pela literatura. Este método foi inicialmente desenvolvido por Carlin em [125] e extensamente revisitado e aplicado a uma série de problemas de casamento de

impedâncias [126-127, 134-136]. A técnica denominada *Real Frequency Technique* (RFT) manipula a parte real da impedância de carga - no caso particular deste estudo de caso, a impedância da antena - sem que haja necessidade de qualquer tipo de descrição analítica da função da impedância de carga, de tal forma que os dados experimentais da carga são diretamente utilizados pelo algoritmo. A Fig. 4-32 mostra a rede de casamento de impedâncias obtida após a aplicação da RFT. A descrição detalhada do método encontra-se no Apêndice I.

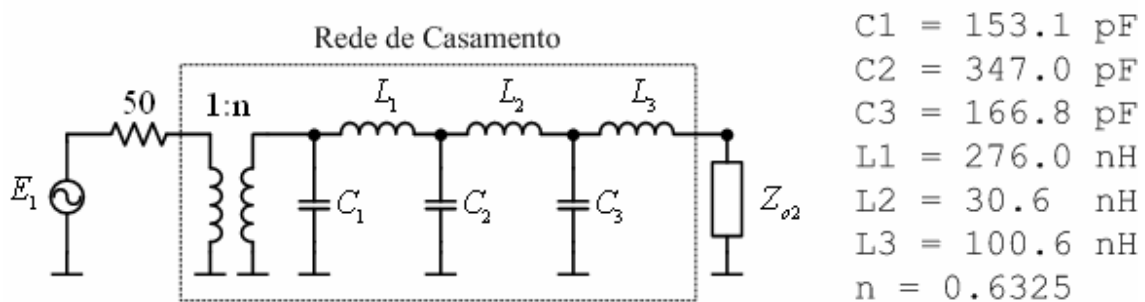


Figura 4-32. Rede de casamento obtida usando a RFT.

Retornando ao processo evolutivo, os parâmetros de controle bem como os pesos e a constante definidos, respectivamente, em (4-10) e (4-23) são resumidos na Tabela 4-8.

Tabela 4-8. Parâmetros de controle usados na síntese da rede de casamento.

Máximo Número de Nós	10
Tamanho da População	200
Número de Gerações	100
Taxa de Crossover	60
Taxa de Mutação	10
Número de Iterações do Simplex	25
$w_{VSWR} = w_{Sens}$	2
$w_{PF}$	1
$\lambda$	0,1
Intervalo dos Capacitores (pF)	[1, 820]
Intervalo dos Indutores (nH)	[25, 900]

Na Fig. 4-33 a evolução do melhor indivíduo ao longo das gerações para duas diferentes configurações do algoritmo é mostrada. Na primeira configuração, sem o passo de sintonia, a topologia e os valores dos componentes foram manipulados pelo GA. Neste caso, como ilustrado na figura, a aptidão permaneceu quase constante ao



longo das gerações. Na segunda configuração, o Nelder-Mead Downhill Simplex foi usado com o GA. É possível notar que as mudanças substanciais na aptidão entre gerações consecutivas foram causadas pela introdução do passo de sintonia.

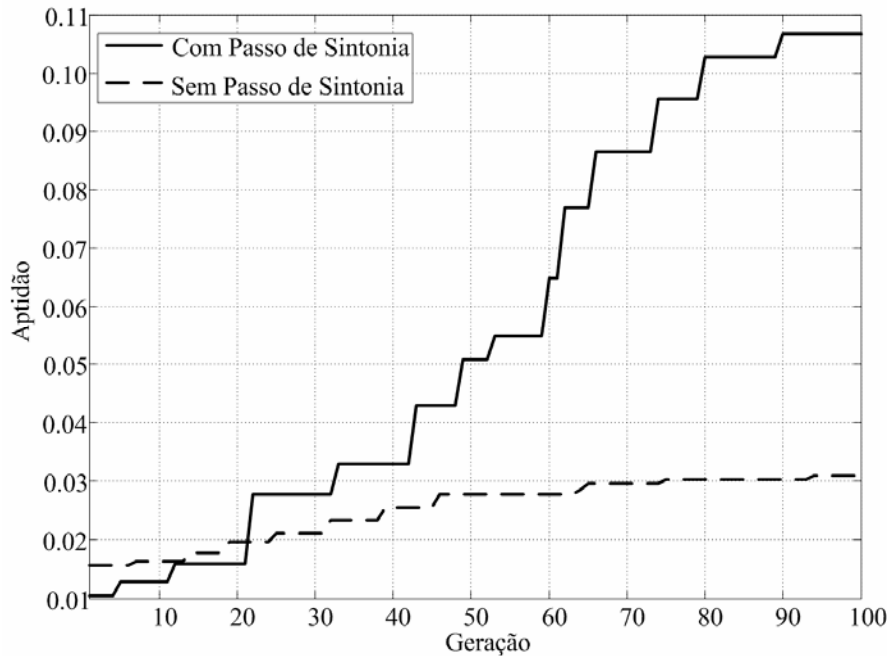


Figura 4-33. Aptidão x Geração para o melhor indivíduo em duas configurações do algoritmo evolutivo.

A rede de casamento de impedâncias contendo somente 6 componentes sintetizada pelo algoritmo evolutivo proposto é mostrada na Fig. 4-34. Como esperado, a topologia do circuito é completamente não-usual quando comparada com redes de casamento resultantes de métodos convencionais. Além disso, a rede sintetizada casa a baixa impedância da antena diretamente com o estágio de potência de  $50\Omega$  sem a necessidade de utilização de qualquer transformador adicional.

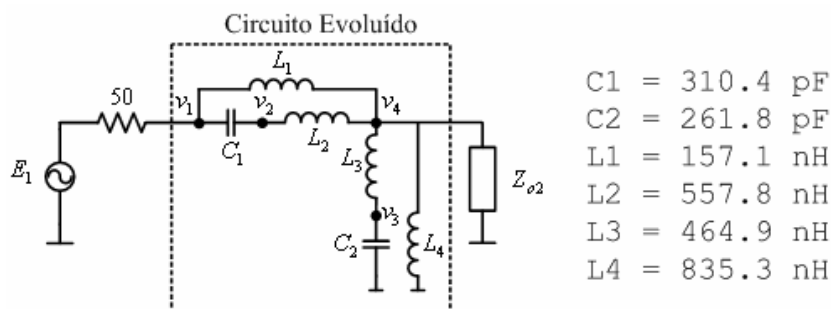


Figura 4-34. Rede de casamento de impedâncias sintetizada pelo algoritmo evolutivo proposto.

Como mostrado na Tabela 4-9, o desempenho do algoritmo proposto quando comparado com o RTF foi ligeiramente melhor com relação ao erro no VSWR e consideravelmente melhor com relação ao erro na sensibilidade. Este resultado já era esperado devido à topologia do tipo *ladder* usada na RTF.

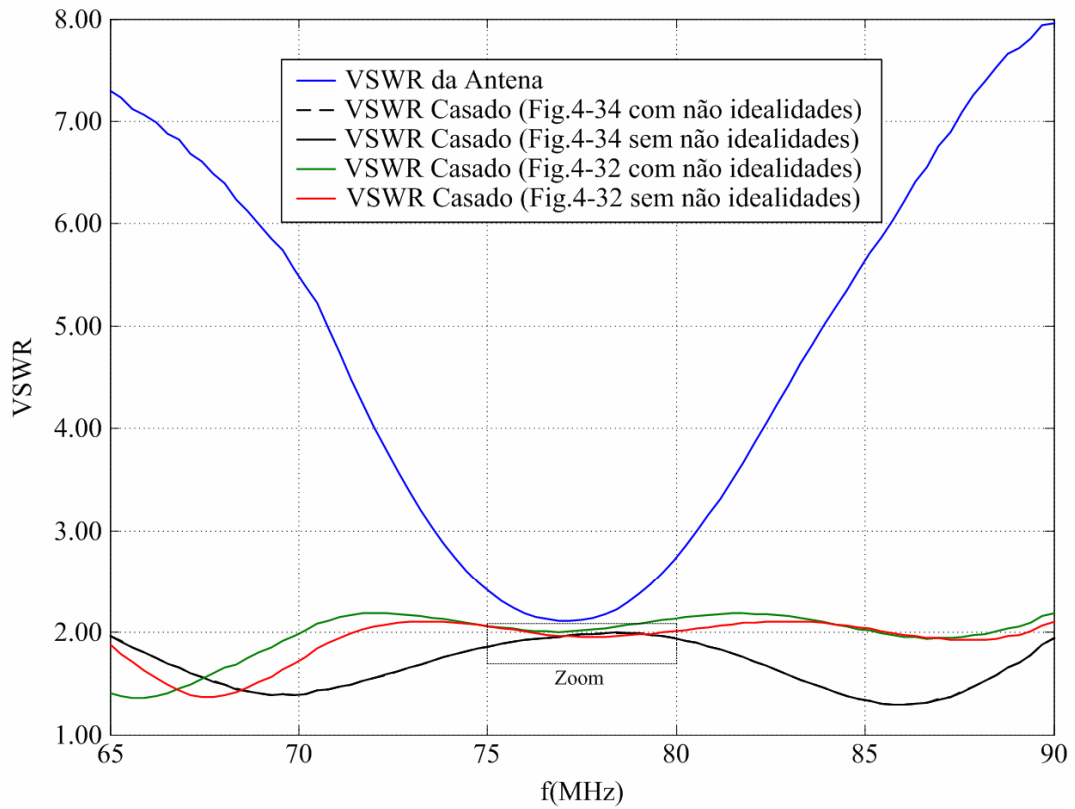
Tabela 4-9. Funções erro para os circuitos sintetizados.

	Método Proposto (Fig. 4-34)	RFT (Fig. 4-32)
$\varepsilon_{VSWR}$	1.99	2.08
$\varepsilon_{Sens}$	2.10	88.60
$\varepsilon_{PF}$	0.00	0.00

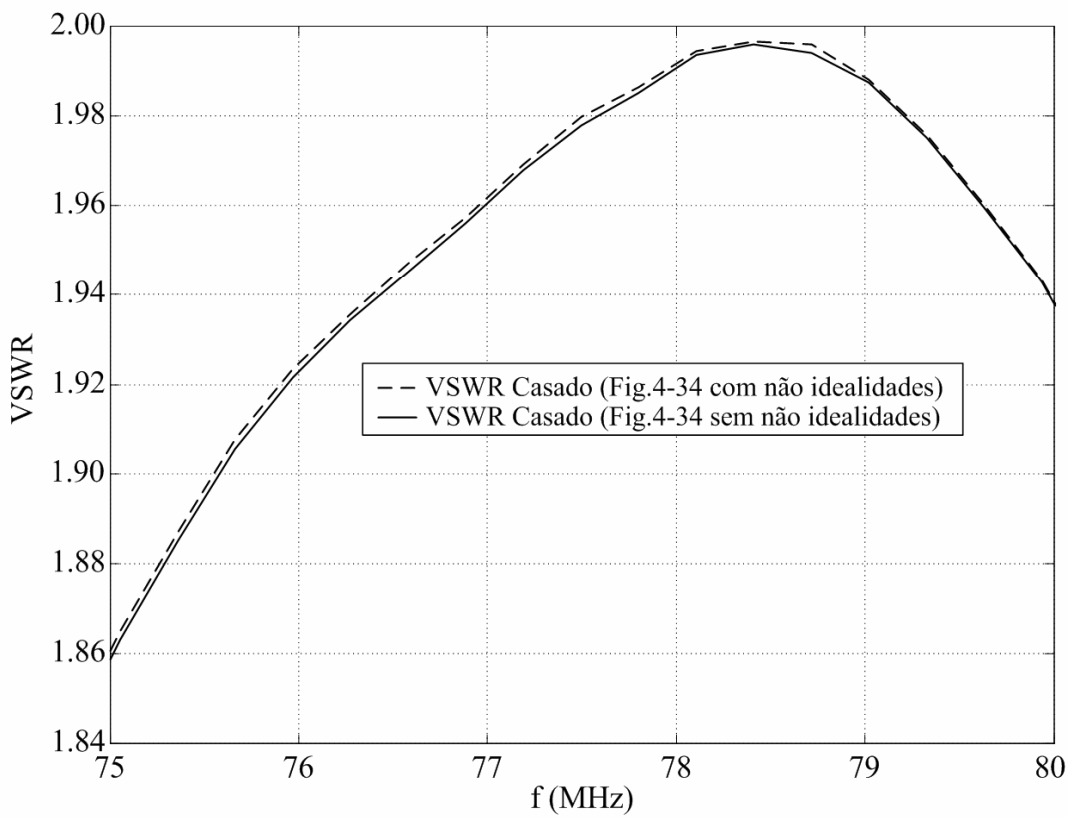
A Fig. 4-35 mostra cinco curvas de VSWR ao longo da banda de frequência de interesse. A primeira curva corresponde ao VSWR original da antena sem nenhuma rede de casamento. Neste caso, o VSWR alcançou aproximadamente 8 o que significa que, no final da banda de frequência, menos de 40% da potência disponível no estágio de potência é transferida para a antena.

A segunda e terceira curvas correspondem ao VSWR da antena depois da introdução das redes de casamento, respectivamente, ilustradas nas Figs. 4-32 e 4-34.

As duas últimas curvas foram obtidas introduzindo não idealidades às redes de casamento de impedâncias mostradas nas Figs. 4-32 e 4-34. A primeira não idealidade corresponde a troca dos valores dos componentes mostrados nas Figs. 4-32 e 4-34 por valores práticos listados na Tabela 4-10. Nesta tabela, alguns valores de componentes foram obtidos após a associação de dois ou mais componentes com valores padrão. Por exemplo, o valor de capacitância  $C_1 = 309 pF$  foi obtido da associação em paralelo de  $270 pF$  com  $39 pF$ .



(a)



(b)

Figura 4-35. (a) VSWR ao longo da banda de frequência de interesse (64,99-90,00 MHz) (b) Zoom na região de maior diferença de VSWR para analisar o desvio após a introdução das não idealidades no circuito da Fig. 4-34.

Tabela 4-10. Valores práticos de componentes.

	Método Proposto (Fig. 4-34)	RFT (Fig. 4-32)
$C_1$	309 pF	150 pF
$C_2$	263 pF	346 pF
$C_3$	---	166 pF
$L_1$	149 nH ( $R_p = 2,11\Omega$ )	280 nH ( $R_p = 4,25\Omega$ )
$L_2$	557 nH ( $R_p = 7,66\Omega$ )	29 nH ( $R_p = 0,54\Omega$ )
$L_3$	470 nH ( $R_p = 7,01\Omega$ )	100 nH ( $R_p = 1,69\Omega$ )
$L_4$	835 nH ( $R_p = 12,19\Omega$ )	---

A segunda não idealidade corresponde à inserção de perdas ôhmicas nos indutores. O principal efeito parasita na banda de frequência de interesse pode ser modelado pela introdução de um resistor em série com o indutor real. Os valores dos resistores foram escolhidos de acordo com [137] e estão indicados na Tabela 4-10 como  $R_p$ .

De maneira a mostrar o quase desprezível impacto da introdução dos efeitos parasitas na curva do VSWR para o circuito da Fig. 4-34, a região da curva do VSWR entre 75 e 80 MHz da Fig. 4-35(a) é ampliada na Fig. 4-35(b).

## 5 Conclusões e Propostas de Trabalho

Um algoritmo evolutivo capaz de sintetizar filtros digitais com baixa sensibilidade foi proposto. Um novo esquema de codificação cromossomial baseado na matriz adjacência que aumenta a dimensão do espaço de busca em relação à GP foi empregado. Foi mostrado que aplicando regras simples que necessitam de reduzido esforço computacional, a representação proposta tem a propriedade de detectar a geração de indivíduos estruturalmente inválidos e não-computáveis durante o processo evolutivo.

De forma a testar o algoritmo, três estudos de casos foram realizados. Nos dois primeiros a síntese considerou apenas o módulo da resposta em frequência e na terceira síntese a fase também foi levada em consideração no cálculo da aptidão. Em todos os testes o algoritmo apresentou resultados superiores àqueles obtidos utilizando outras técnicas encontradas na literatura.

Na síntese de circuitos analógicos empregando somente transistores MOS alguns resultados foram obtidos, embora circuitos que apresentassem ganho dificilmente conseguiram ser sintetizados. Provavelmente, esta dificuldade é originária da escassez de topologias que fornecem ganho e da polarização de circuitos CMOS. Uma alternativa é a utilização de estruturas já conhecidas num esquema chamado de reuso [138]. Neste sentido, é possível associar a estas estruturas o GP Cartesiano para sintetizar estruturas com ganho.

Na síntese de filtros analógicos passivos a representação topológica do circuito empregando matriz adjacência é bastante simples e não restritiva, ou seja, é possível representar qualquer topologia. Como um caso particular de síntese de filtros analógicos passivos, as redes de casamento de impedâncias foram abordadas a título de estudo de caso. Um algoritmo capaz de sintetizar estas redes foi proposto incorporando no cálculo da aptidão a sensibilidade dos circuitos para garantir robustez na implementação prática.

Mais especificamente, o casamento de impedâncias entre um estágio de potência de um radiotransmissor e sua antena foi examinado. Os resultados obtidos com o processo evolutivo proposto foram superiores àqueles obtidos com o consagrado método RFT, principalmente quando não idealidades são inseridas nos circuitos.

Como trabalhos futuros decorrentes desta tese, o circuito da Fig. 4-34 pode ser montado e testado para verificar na bancada os resultados obtidos na simulação.

## Referências Bibliográficas

- [1] HOLLAND, J., *Adaptation in Natural and Artificial Systems*, 1st ed., University of Michigan Press, Ann Arbor, 1975.
- [2] GOLDBERG, D., *Genetic Algorithm in Search, Optimization and Machine Learning*, Addison-Wesley Publishing, Massachusetts, 1989.
- [3] ZADEH, L. A, Fuzzy sets, *Information and Control*, vol. 8, pp. 338-353, 1965.
- [4] HAYKIN, S., *Neural Networks: A Comprehensive Foundation*, 2<sup>nd</sup> Ed., Prentice Hall, 1999.
- [5] KOZA, J. R., *Genetic Programming On the Programming of Computers by Means of Natural Selection*, MIT Press, 1992.
- [6] GREENWOOD, G. W., TYRRELL, A., *Introduction to Evolvable Hardware: A Practical Guide for Designing Self-Adaptive Systems*, IEEE Press, 2007.
- [7] MATIUSSI, C., *Evolutionary Synthesis of Analog Networks*, PhD Thesis, EPFL, 2005.
- [8] FLOREANO, D., MONDADA, F., Evolution of homing navigation in a real mobile robot., *IEEE Transactions on Systems, Man, and Cybernetics, Part B*, pp. 396-407, 1996.
- [9] HARVEY, I., HUSBANDS, P., CLIFF, D., THOMPSON, A., JAKOBI, N., Evolutionary robotics: the Sussex approach, *Robotics and Autonomous Systems*, 20 (2-4), pp. 205-224, 1997.
- [10] NOLFI, S., FLOREANO, D. *Evolutionary Robotics: The Biology, Intelligence, and Technology of Self-Organizing Machines*, MIT Press, Cambridge, MA, 2000.
- [11] MILLER, J. F., JOB, D., VASSILEV, V. K., Principles in the Evolutionary Design of Digital Circuits, Part I, *Journal of Genetic Programming and Evolvable Machines*, vol. 1(1), pp. 8-35, 2000.
- [12] MILLER, J. F., JOB, D., VASSILEV, V. K., Principles in the Evolutionary Design of Digital Circuits, Part II, *Journal of Genetic Programming and Evolvable Machines*, vol. 3(2), pp. 259-288, 2000.
- [13] CHEANG, S. M., LEE, K. H., LEUNG, K.S., Applying Genetic Parallel Programming to Synthesize Combinational Logic Circuits, *IEEE Transactions on Evolutionary Computation*, vol. 11(4), pp. 503-520, 2007.
- [14] WILLIAMS, A. B., *Electronic Filter Design Handbook*, McGraw Hill, 1981.
- [15] FOGEL, L., *Autonomous Automata*, Industrial Research, Vol. 4, pp. 14-19, 1962.
- [16] FOGEL, L., OWENS, A., WALSH, M., *Artificial Intelligence Through Simulated Evolution*, John Wiley & Sons, New York, 1966.

- [17] SCHWEFEL, H., Numerical Optimization of Computer Models, New York, John Wiley & Sons, 1981.
- [18] ZEBULUM, R. S., PACHECO, M. A. C., VELLASCO, M. M. B. R., Evolutionary Electronics: Automatic Design of Electronic Circuits and Systems by Genetic Algorithms, CRC Press, 2000.
- [19] BEASLEY, D., Possible Applications of Evolutionary Computation, in Handbook of Evolutionary Computation, IOP Publishing Ltd. and Oxford University Press, 1997.
- [20] WOLPERT, D., MACREADY, W., No Free Lunch Theorem for Optimazation, IEEE Transactions on Evolutionary Computation, pp. 67-82, 1997.
- [21] SCHWEFEL, H., Advantages and Disadvantages of Evolutionary Computation Over Other Approaches, in Handbook of Evolutionary Computation, IOP Publishing Ltd. and Oxford University Press, 1997.
- [22] GRIMBLEBY, J. B., Automatic Analogue Network Synthesis Using Genetic Algorithms, Proc. of the International Conference on Genetic Algorithms in Engineering Systems, 1995.
- [23] LIPSCHUTZ, S., Theory and Problems of Discrete Mathematics, McGraw-Hill, 1997.
- [24] MASON, S. J., ZIMMERMAN, Electronic Circuits, Signals and Systems, pp. 122-123, Wiley, NY, 1960.
- [25] CHANG, S., HOU, H., SU, Y., Automated Passive Filter Synthesis Using a Novel Tree Representation and Genetic Programming, IEEE Transactions on Evolutionary Computation, vol. 10, pp.93-100, 2006.
- [26] GALLAGHER, M., RYAN, A., Learning to play Pac-Man: an evolutionary, rule-based approach Evolutionary Computation, Congress on Evolutionary Computation, vol. 4, pp. 2462 - 2469, 2003.
- [27] ERCEGOVAC, M., LANG, T., MORENO, J., Introduction to Digital Systems, Publisher: Wiley, 1999.
- [28] BANZHAF, W., FRANCONI, F., NORDIN, P., On some emergent properties of variable size evolutionary algorithms, ICGA Workshop on Evolutionary Computation with Variable Size Representation, 1997.
- [29] EIBEN, A. E.; HINTERDING, R.; MICHALEWICZ, Z., Parameter control in evolutionary algorithms. IEEE Transactions on Evolutionary Computation, vol. 3, issue 2, 1999.
- [30] PACHECO, M. A. C. P., Algoritmos Genéticos: Princípios e Aplicações, available at [www.ica.ele.puc-rio.br/cursos/download/CE-Apostila-Comp-Evol.pdf](http://www.ica.ele.puc-rio.br/cursos/download/CE-Apostila-Comp-Evol.pdf).



- [31] SYWERDA, G., Uniform crossover in genetic algorithms, Proceedings of the third international conference on Genetic algorithms, pp. 2 – 9, 1989.
- [32] MILLER, J. F., THOMSON, P., FOGARTY, T., Designing Electronic Circuits Using Evolutionary Algorithms. Arithmetic Circuits: A Case Study, Genetic Algorithms Recent Advancements and Industrial Applications, Chapter 6, 1997.
- [33] BÄCK, T., SCHUTZ, M., Intelligent mutation rate control in canonical genetic algorithms, Proceedings of the Ninth International Symposium on Methodologies for Intelligent Systems, Lecture Notes in Computer Science, 1996.
- [34] BLICKLE, T., Theory of Evolutionary Algorithms and Application to System Synthesis, PhD Thesis, Swiss Federal Institute of Technology, 1996.
- [35] MÜHLENBEIN, H., VOOSSEN, D., Predictive model for the breeder genetic algorithm, Evolutionary Computation, 1(1), 1993.
- [36] BLICKLE, T., THIELE, L. A Mathematical Analysis of Tournament Selection, Proceedings 6<sup>th</sup> International Conference on Genetic Algorithms, pp. 9-16, 1995.
- [37] ZEBULUM, R. S., Síntese de Circuitos Eletrônicos por Computação Evolutiva, PhD Thesis, PUC-Rio, 1999.
- [38] WHITLEY, D., A Genetic Algorithm Tutorial, available at [www.cs.iastate.edu/~honavar/ga\\_tutorial.pdf](http://www.cs.iastate.edu/~honavar/ga_tutorial.pdf).
- [39] ZHOU, S., HECKENDORN, R. B., ZENGGI, S. Detecting the epistatic structure of generalized embedded landscape, Genetic Programming and Evolvable Machines, 2007.
- [40] GREENWOOD, G., On the practicality of using intrinsic reconfiguration for fault recovery, IEEE Transactions on Evolutionary Computation, pp. 398-405, 2005.
- [41] KEYMEULEN, D., ZEBULUM, R., RAJESHUNI, R., STOICA, A., KATKOORI, S., GRAVES, S., NOVAK, F., ANTIL, C., Extreme Temperature Electronics based on Self-Adaptive System using Field Programmable Gate Array, IEEE Aerospace Conference, vol. 1, pp. 1-6, 2007.
- [42] WISMER, M., Steady-state operation of a high-voltage multiresonant converter in a high-temperature environment, IEEE Transactions on Power Electronics, pp. 740-748, 2003.
- [43] VLADIMIROVA, T., BRIDGES, C. P., PRASSINOS, G., XIAOFENG, W., KAWSU, S., BARNHART, D. J., ABDUL, J., PAUL, J. R., LAPPAS, V., BAKER, A., MAYNARD, K., MAGNESS, R., Characterizing Wireless Sensor Motes for Space Applications, NASA/ESA Conference on Adaptive Hardware and Systems, pp. 43-50, 2007.
- [44] OPPENHEIM, A. V., SCHAFER, R. W., Digital Signal Processing, Prentice-Hall International, London, 1975.

- [45] ROSENMAN, M.A., An evolutionary model for non-routine design, in Proc. of the Eighth Australian Joint Conference on Artificial Intelligence, pp. 363–370, 1995.
- [46] YAO, X., LIU, Y., Towards designing artificial neural networks by evolution, Applied Mathematics and Computation, vol. 91, pp. 83–90, 1998.
- [47] SCHNIER, T., YAO, X., LIU, P., Digital filter design using multiple pareto fronts, Soft Computing vol. 8, pp.332–343, Springer-Verlag, 2004.
- [48] RYAN, C., COLLINS, J. J., NEIL, M. O., Grammatical evolution: Evolving programs for an arbitrary language,” in Proc. 1st European Workshop Genetic Program., vol. 1391, pp. 83–95, 1998.
- [49] MILLER, J., An empirical study of the efficiency of learning Boolean functions using a Cartesian genetic programming approach, in Proc. Genetic. Evol. Comput. Conf., vol. 1, pp. 1135–1142, 1999.
- [50] KO, M. S., KANG, T. W., HWANG, C. S. Hwang, Function optimisation using an adaptive crossover operator based on locality, Eng. Appl. Artif. Intell., vol. 10, no. 6, pp. 519–524, 1997.
- [51] CANTÚ-PAZ, E., A survey of parallel genetic algorithms, Calculateurs Paralleles, Reseaux et Systems Repartis, vol. 10, no. 2, pp. 141–171, 1998.
- [52] ALBA, E., TOMASSINI, M., Parallelism and evolutionary algorithms, IEEE Trans. Evol. Comput., vol. 6, no. 5, pp. 443–462, 2002.
- [53] HARIK, G., LOBO, F. GOLDBERG, D., The compact genetic algorithm, IEEE Trans. Evol. Comput., vol. 3, no. 4, pp. 287–297, 1999.
- [54] A. Antoniou, Digital Signal Processing: Signals, Systems, and Filters, McGraw-Hill, 2005.
- [55] KARABOGA, N., Digital IIR Filter Design Using Differential Evolution Algorithm, EURASIP Journal on Applied Signal Processing, pp. 1269–1276, 2005.
- [56] KOZA, J., BENNETT, F. H., ANDRE, D., KEANE, M. A., Genetic Programming III. Darwinian Invention and Problem Solving, CA: Morgan Kaufmann, 1999.
- [57] MASTORAKIS, N. E., GONOS, I. F., SWAMY, M. S., Design of two-dimensional recursive filters using genetic algorithms, IEEE Trans. Circ. and Syst. I, vol. 50, no. 5, pp. 634-639, 2003.
- [58] THUMVICHAI, R., BOSE, T., HAUPT, R. L., Design of 2-D multiplierless IIR filters using the genetic algorithm, IEEE Trans. Circ. and Syst. I, vol. 49, no. 6, pp. 878-882, 2002.
- [59] ONER, M., ASKAR, M., Incremental design of high complexity FIR filters by genetic algorithms, Proc. Int. Symp. Signal Processing and Its Applications, vol. 3, pp. 1005-1008, 1999.

- [60] LEE, A., AHMADI, M., JULLIAN, G. A., MILLER, W. C., LASHKARI, R. S., Design of 1-D FIR filters using genetic algorithms, Proc. IEEE Int. Symp. Circ. and Syst., vol. 3, pp. 295-298, 1999.
- [61] TANG, K. S., MAN, K. F., KWONG, S., LIU, Z. F., Design and optimization of IIR structure using hierarchical genetic algorithms, IEEE Trans. Industrial Electronics, vol. 45, No. 3, pp 481-487, 1998.
- [62] TANG, K. S., MAN, K. F., KWONG, S., HE, Q., Genetic algorithms and their applications, IEEE Signal Processing Magazine, vol. 13, no. 6, pp.22-37, 1996.
- [63] ARSLAN, T., HORROCKS, D. H., A Genetic algorithm for the design of finite word length arbitrary response cascaded IIR digital filters, Proc. IEE Conf. of Genetic Algo. in Engineering Syst: Innov. and Applications, pub. no. 414, pp. 276-281, 1995.
- [64] ERBA, M., ROSSI, R., LIBERALI, V., TETTAMANZI, A. G. B., Digital Filter Design through Simulated Evolution, European Conference on Circuit Theory and Design, vol. 2, pp. 137-140, 2001.
- [65] TUFTE, G., HADDOW, P. C., Evolving an Adaptive Digital Filter, Proc. of the 2000 NASA/DoD Conference on Evolvable Hardware, IEEE Computer Press, pp. 143 – 150, 2000.
- [66] MILLER, J., Digital filter design at gate-level using evolutionary algorithms, Proc. of the Genetic and Evol. Comp. Conf. (GECCO'99), pp. 1127-1134, 1999.
- [67] UESAKA, K., KAWAMATA, M., Evolutionary Synthesis of Digital Filter Structures Using Genetic Programming, IEEE Trans. Cir. and Syst. II, vol. 50, pp. 977-983, 2003.
- [68] UESAKA, K., KAWAMATA, M., Synthesis of low-sensitivity second-order digital filters using genetic programming with automatically defined functions, IEEE Signal Processing Lett., vol. 7, pp. 83-85, 2000.
- [69] SANTINI, C. C., AMARAL, J. F. M., PACHECO, M. A. C., TANSCHKEIT, R., Evolvability and reconfigurability, in Proc. IEEE Int. Conf. Field-Program. Technol., pp. 105-112, 2004.
- [70] BALABANIAN, N., Network Synthesis, Prentice-Hall, Englewood Cliffs, 1958.
- [71] MITRA, S. K., Digital Signal Processing. New York: McGraw-Hill, 1998.
- [72] LU, W. S., Design of stable IIR digital filters with equiripple passbands and peak-constrained least-squares stopband. IEEE Trans. on Circ. and Syst. II: Analog and Digital Signal Processing, vol. 46, pp. 1421-1426, 1999.
- [73] NAMBIAR, R., TANG, C. K., MARS, P., Genetic and learning automata algorithms for adaptive filter design, Proc. IEEE Int. Conf. Acoust. Speech and Signal Processing, vol. 50, pp.1253-1256, 1981.

- [74] ESPARCIA, A. I., SHARMAN, K. C., Some Applications of Genetic Programming in Digital Signal Processing, in Late Breaking Papers at Genetic Programming, pp. 24-31, 1996.
- [75] SUNDARALINGAM S., SHARMAN, K. C., Genetic Evolution of Adaptive Filters, in Proc. of DSP, pp. 47-53, 1997.
- [76] CHELLAPILLA, K., FOGEL, D. B., RAO, S. S., Gaining Insight into Evolutionary Programming Through Landscape Visualization: An Investigation into IIR Filtering, Evolutionary Programming, pp. 407-417, 1997.
- [77] REDMILL, D. W., BULL, D. R., Design of Low Complexity FIR Filters using Genetic Algorithms and Directed Graphs, in Proc. of the Second IEEE Int. Conf. on Genetic Algorithms in Eng. Syst.: Innovations and Applications, vol. 446, 1997.
- [78] DELIBASIS, K. K., UNDRILL, P. E., CAMERON, G. G., Genetic algorithm implementation of stack filter design for image restoration, IEE Proceedings in Vision, Image and Signal Processing, vol. 143, no. 3, pp. 177-183, 1996.
- [79] SRIRANGANATHAN, S., BULL, D. R., REDMILL, D. W., Design of 2-D Multiplierless FIR Filters using Genetic Algorithms, pp. 282-286, in Proc. of the Second IEEE Int. Conf. on Genetic Algorithms in Eng. Syst.: Innovations and Applications, 1995.
- [80] WADE, G., ROBERTS, A., WILLIAMS, G., Multiplier-less FIR filter design using a genetic algorithm, IEE Proceedings in Vision, Image and Signal Processing, vol. 141, no. 3, pp. 175-180, 1994.
- [81] ZVADA, S., KOKAI, G., VANYI, R., FRUHAUF, H. H., evolFIR: Evolving redundancy-free FIR structures, Proc. of the NASA/ESA Conference on Adaptive Hardware and Systems, IEEE Computer Press, pp. 439-446, 2007.
- [82] AGARWAL, R. C., BURRUS, C. S. Burrus, New recursive digital filter structures having very low sensitivity and roundoff noise, IEEE Trans. Circ. and Syst., vol. 22, pp. 921-927, 1975.
- [83] DINIZ, P. S. R., ANTONIOU, A., Low-sensitivity digital-filter structures which are amenable to error spectrum shaping, IEEE Trans. Circuits Syst., vol. 32, pp. 1000-1007, 1985.
- [84] RAO, Y. V. R., ESWARAN, C., A Pole-Sensitivity Based Method for the Design of Digital Filters for Error-Spectrum Shaping, IEEE Trans. Circuits Syst., vol. 36, pp. 1017-1020, 1989.
- [85] BARNES, C., On the design of optimal state-space realizations of second-order digital filters, IEEE Trans. Circuits Syst., vol. 31, pp. 602-608, 1984.
- [86] MESQUITA, A., SALAZAR, F. A., CANAZIO, P. P., Chromosome Representation through Adjacency Matrix in Evolutionary Circuits Synthesis, Proc. of the 2002

NASA/DoD Conference on Evolvable Hardware, IEEE Computer Press, pp. 102-109, Alexandria, USA, 2002.

[87] SAPARGALIYEV, Y.; KALGANOVA, T., Constrained and Unconstrained evolution of LCR low-pass filters with oscillating length representation, IEEE Congress on Evolutionary Computation, pp. 1529-1536, 2006.

[88] SWAMY, M. N. S., THULASIRAMAN, K., Graphs, Networks and Algorithms, John Wiley & Sons, 1981.

[89] WARSHALL, S., A Theorem on Boolean Matrices, Journal ACM, vol. 9, pp. 11-12, 1962.

[90] KOZA, J. R., ANDRE, D., BENNETT III, F. H., KEANE, M. A., Use of Automatically Defined Functions and Architecture-Altering Operations in Automated Circuit Synthesis with Genetic Programming, in Proc. of the First Annual Conference in Genetic Programming, 1996.

[91] STEELE, G. L., Common Lisp : The Language, 2nd ed. Bedford, MA: Digital, 1990.

[92] CROCHIERE, R. E., OPPENHEIM, A. V., Analysis of linear digital networks, in Proc. IEEE, vol. 63, pp. 581-595, 1975.

[93] KAWAMATA, M., IWATSUKI, M., HIGUCHI, T., Balanced realizations as minimum sensitivity structures in linear systems, Trans. Soc. Instrum. Contr. Eng., vol. 21, pp. 900-906, 1985.

[94] INUKAI, T. Inukai, A unified approach to optimal recursive digital filter design, IEEE Trans. Circuits Syst., vol. CAS-27, pp. 646-649, 1980.

[95] CHARALAMBOUS, C., ANTONIOU, A., Equalization of recursive digital filters, in Proc. Inst. Elec. Eng., vol. 127, part. G, no. 4, pp. 219-225, 1980.

[96] DECZKY, A. G., Synthesis of recursive digital filters using the minimum p-error criterion, IEEE Trans. Audio Electroacoust., vol. AE-32, pp.949-967, 1984.

[97] ARAÚJO, S. G., Síntese de Sistemas Digitais Utilizando Técnicas Evolutivas, PhD Thesis, COPPE-UFRJ, 2004.

[98] LOHN, J. D., COLOMBANO, S. P., A Circuit Representation Technique for Automated Circuit Design, IEEE Trans. on Evol. Comput., vol. 3, no. 3, pp. 205-219, 1999.

[99] DAMAVANDI, N.; SAFAVI-NAEINI, S., A hybrid evolutionary programming method for circuit optimization, IEEE Trans. Circuits Syst., vol. CAS-52, pp. 902-910, 2005.

[100] THOMPSON, M., FIDLER, J. K., Application of the genetic algorithm and simulated annealing to LC filter tuning, in IEE Proc. Circ., Dev. and Syst., vol. 148, pp.177-182, 2001.

- [101] JIAXIN L., MINAN L.; XI, C., HU, J., The use of genetic algorithms for statistical circuit design, IEEE Int. Conf. on Electronics, Circuits and Systems, vol. 2, pp. 975-978, 2000.
- [102] LANGEHEINE, J.; BECKER, J.; FOLLING, S.; MEIER, K.; SCHEMMEL, J., A CMOS FPTA chip for intrinsic hardware evolution of analogelectronic circuits, in Proc. of the 3rd NASA/DoD Workshop on Evolvable Hardware, pp. 172-175, 2001.
- [103] STOICA, A. et ali.; Evolution of analog circuits on field programmable transistor arrays, in Proc. of the 2nd NASA/DoD Workshop on Evolvable Hardware, pp. 99-108, 2000.
- [104] SÁ, L. B., VIEIRA, P.F., BOTELHO, J.P.B., MESQUITA, A., Evolutionary synthesis of analog circuits using only MOS transistors, in Proc. of the NASA/DoD Conference on Evolvable Hardware, pp. 38-45, 2004.
- [105] SÁ, L. B., VIEIRA, P.F., BOTELHO, J.P.B., MESQUITA, A., An experiment on nonlinear synthesis using evolutionary techniques based only on CMOS transistors, in Proc. of the NASA/DoD Conference on Evolvable Hardware, pp. 50-58, 2003.
- [106] KUNTMAN, H., CICEKOGLU, O., OZCAN, S., Realization of current-mode third-order Butterworth filters employing equal valued passive elements and unity gain buffers, Anal. Int. Cir. and Sig. Proc., vol. 30, pp. 253–256, 2002.
- [107] GUPTA, S. S., SENANI, R., New Single-Resistance-Controlled Oscillator Configurations Using Unity-Gain Cells, Anal. Int. Cir. and Sig. Proc., vol. 46, pp. 111-119, 2006.
- [108] KESKIN, A. U., TOKER, A., A NIC with impedance scaling properties using unity gain cells, Anal. Int. Circ. and Sig. Proc., Mix. Sig. Letters, 2004.
- [109] SÁ, L. B., MESQUITA, A., Synthesis of Voltage Follower with Only CMOS Transistors Using Evolutionary Methods, in Proc. of the NASA/ESA Conference on Adaptive Hardware and Systems, pp. 478-485, 2007.
- [110] PALUMBO, G., PENNISI, S., A high performance CMOS voltage follower, Proc. IEEE Int. Conf. ECS, vol. 2, pp. 21–24, 7–10, 1998.
- [111] GRAY, P. R., HURST, P. J., LEWIS, S. H., MEYER, R. G., Analysis and Design of Analog Integrated Circuits, 4th Ed., John Wiley & Sons, 2001.
- [112] TSIVIDIS, Y., Operation and Modeling of the MOS Transistor, 2nd Ed., ISBN: 0-07-065523-5, McGraw-Hill, 1999.
- [113] SANTINI, C. C., ZEBULUM, R. S., PACHECO, M. A. C., VELLASCO, M. M., SZWARCMAN, M., Evolution of Analog Circuits on a Programmable Analog Multiplexer Array, Aerospace Conference, vol. 5, pp. 2301-2308, 2001.
- [114] AMARAL, J. F. M., AMARAL, J. L. M., SANTINI, C. C., TANSCHKEIT, R., VELLASCO, M. M. R., PACHECO, M. A. C., Towards Evolvable Analog Fuzzy Logic

- Controllers, in Proc. of the NASA/DoD Conference on Evolvable Hardware, pp. 123-128, 2002.
- [115] YAO, X., HIGUCHI, T., Promises and Challenges of Evolvable Hardware, IEEE Trans. On Syst., Man & Cyber., Part C, vol. 29(1), pp. 87-97, 1999.
- [116] ZEBULUM, R. S., VELLASCO, M. M., PACHECO, M. A. C., Variable length representation in evolutionary electronics, *Evol. Comput.*, vol. 8, no. 1, pp. 93-120, 2000.
- [117] ANDO, S., ISHIZUKA, M., IBA, H., Evolving analog circuits by variable length chromosomes, in *Advances in Evolutionary Computing*, pp. 643-662, 2003.
- [118] KOZA, J. R., BENNET, F. H., ANDRE, D. KEANE, M. A., DUNLAP, F., Automated Synthesis of Analog Electrical Circuits by Means of Genetic Programming, *IEEE Trans. on Evol. Comp.*, vol. 1, pp. 109-128, 1997.
- [119] MATTIUSSI, C., FLOREANO, D., Analog Genetic Encoding for the Evolution of Circuits and Networks, *IEEE Trans. on Evol. Comp.*, vol. 11, no. 5, pp. 596-607, 2007.
- [120] ZEBULUM, R. S., PACHECO, M. A. C., VELLASCO, M., Comparison of Different Evolutionary Methodologies Applied to Electronic Filter Design, *IEEE International Conference on Computational Intelligence*, pp. 434-439, 1998.
- [121] LEE, M. S., TEMES, G. C., CHANG, C., Bilinear switched-capacitor ladder filters, *IEEE Trans. Circuits and System, CAS-28*, 811-821, 1981.
- [122] DATAR, R. B., SEDRA, A. S., Exact design of stray-insensitive switched – capacitor ladder filters”, *IEEE Trans. Circuit and System, CAS-30*, 888-898, 1983.
- [123] FANO, F. M., Theoretical limitations on the broadband matching of arbitrary impedances, *J. Franklin Inst.*, vol. 249, pp. 57–83, 1950.
- [124] YOULA, D. C., A new theory of broadband matching, *IEEE Trans. Circuit Theory*, vol. CT-11, pp. 30–50, 1954.
- [125] CARLIN, H. J., A New Approach to Gain-Bandwidth Problems, *IEEE Trans. on Circ. and Syst.*, vol. 24, no. 4, pp. 170-175, 1977.
- [126] CARLIN, H. J., YARMAN, B. S., The Double Matching Problem: Analytic and Real Frequency Solutions, *IEEE Trans. on Circ. and Syst.*, vol. 30, no. 1, pp.15-28, 1983.
- [127] RODRÍGUEZ, J. L., GARCÍA-TUÑÓN, I., TABEADA, J. M., BASTEIRO, F. O., Broadband HF Antenna Matching Network Design Using Real-Coded Genetic Algorithm, *IEEE Trans. Antennas Propag.*, vol. 55, no. 3, pp. 611-618, 2007.
- [128] HELTON, J. W., Broadbanding: Gain equalization directly from data, *IEEE Trans. Circuits Syst.*, vol. 28, no. 12, pp. 1125–1137, 1981.
- [129] BALABANIAN, N., BICKART, T. A., SESHU, S., *Electrical Network Theory*, John Wiley & Sons, 1969.

- [130] MILLER, J. F., THOMSON, P., Cartesian Genetic Programming, Lecture Notes in Computer Science, vol. 1802, pp. 121-132, 2000.
- [131] DARYANANI, G., Principles of Active Network Synthesis and Design, John Wiley & Sons, 1980.
- [132] VLACH, J., SINGHAL, K., Computer Methods for Circuit Analysis and Design, Van Nostrand Reinhold, 2nd Ed., 1994.
- [133] SMITH, A. E, COIT, D. W., Handbook of Evolutionary Computation, Eds. K. De Jong, L. Fogel, H. Schwefel, C.5.2, 1997.
- [134] YEGIN, K., MARTIN, A. Q., On the Design of Broad-Band Loaded Wire Antennas Using the Simplified Real Frequency Technique and a Genetic Algorithm, IEEE Trans. Antennas Propag., vol. 51, no. 2, pp. 220-228, 2003.
- [135] YARMAN, B. S., CARLIN, H. J., A Simplified Real Frequency Technique Applicable to Broadband Multistage Microwave Amplifiers, IEEE MTT-S Digest, pp. 529-531, 1982.
- [136] RAMAHI, O. M., MITTRA, R, Design of a Matching Network for an HF antenna Using the Real Frequency Method, IEEE Trans. Antennas Propag., vol. 39, no. 4, pp. 506-509, 1989.
- [137] Sítio da empresa Coilcraft, uma das maiores fabricantes de indutores do mundo <http://www.coilcraft.com/>.
- [138] DASTIDAR, T. R., CHAKRABARTI, P. P., RAY, P., A Synthesis System for Analog Circuit Based on Evolutionary Search and Topological Reuse, IEEE Trans. on Evol. Comp., vol. 09 (2), pp. 211-225, 2005.



## Apêndice I

### Descrição do Método Real Frequency Technique (RFT)

O problema do casamento de impedâncias através de uma rede sem perdas entre um estágio resistivo e uma carga representada por uma antena é novamente ilustrado na Fig. I-1.

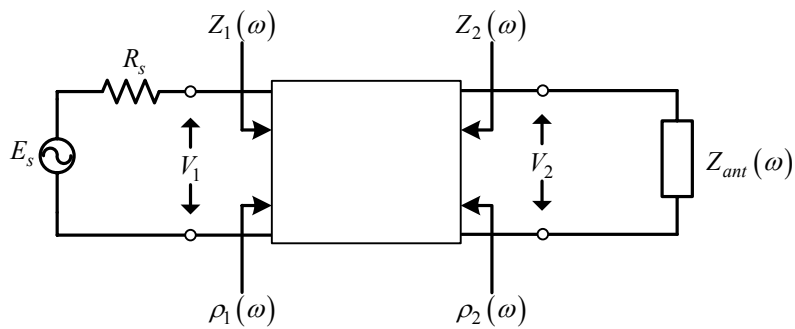


Figura I-1. Circuito de casamento de impedâncias usando RFT.

Para uma rede de duas portas sem perdas, o coeficiente de reflexão da porta 1 é igual ao coeficiente da porta 2 [129]. Portanto, a TPG pode ser expressa na forma:

$$TPG(\omega) = 1 - |\rho_1(\omega)|^2 = 1 - |\rho_2(\omega)|^2 \quad (I-1)$$

O coeficiente de reflexão da porta 2 é dado por:

$$\rho_2(\omega) = \frac{Z_2(\omega) - Z_{ant}^*(\omega)}{Z_2(\omega) + Z_{ant}(\omega)} \quad + \quad (I-2)$$

onde  $Z_{ant}^*(\omega)$  é o complexo conjugado da impedância da antena  $Z_{ant}(\omega)$ .

Se  $Z_{ant}(\omega) = R_{ant}(\omega) + j \cdot X_{ant}(\omega)$  e  $Z_2(\omega) = R_2(\omega) + j \cdot X_2(\omega)$ , o TPG pode ser reescrito sob a forma:

$$TPG(\omega) = \frac{4 \cdot R_{ant}(\omega) \cdot R_2(\omega)}{|Z_{ant}(\omega) + Z_2(\omega)|^2} \quad (I-3)$$

Note que em (I-3) a impedância da antena  $Z_{ant}(\omega)$  é supostamente conhecida, permanecendo como parâmetro a ser determinado a impedância da porta 2  $Z_2(\omega)$ .

A impedância  $Z_2$  com terminação resistiva é uma função positiva real que pode ser escrita como uma função racional:

$$Z_2(s) = \frac{a_0 + a_1s + \dots + a_ms^m}{1 + b_1s + \dots + b_ns^n} = Ev\{Z_2(s)\} + Od\{Z_2(s)\} \quad (I-4)$$

onde  $Ev\{Z_2(s)\}$  e  $Od\{Z_2(s)\}$  são, respectivamente, as partes par e ímpar de  $Z_2(s)$ .

No caso particular do eixo-j $\omega$ :

$$R_2(\omega) = Ev\{Z_2(s)\}|_{s=j\omega} = \frac{N_2(\omega^2)}{D_2(\omega^2)} = \frac{A_0\omega^{2k}}{1 + B_1\omega^2 + \dots + B_n\omega^{2n}} \quad (I-5)$$

Dependendo do valor de  $k$ , a rede de casamento pode ser do tipo passa-baixas ( $k=0$ ) ou passa-faixa ( $k>0$ ). O número de elementos reativos  $n$  usados na realização da rede de casamento é a igual a metade da ordem de  $D_2(\omega^2)$ .

Os passos do método RFT são:

- 1) Os valores inteiros positivos de  $n$  e  $k$  são escolhidos com  $n \geq k$ ;
- 2) Um vetor real  $X = (x_0, x_1, x_2, \dots, x_n)$  de  $n+1$  posições é escolhido. Para

garantir a geração de uma função positiva real dada por (I-5), um polinômio  $P_n(\omega)$  é gerado:

$$P_n(\omega) = 1 + x_1\omega + \dots + x_n\omega^n \quad (I-6)$$

O denominador,  $D_2(\omega^2)$ , em (I-5) é gerado da seguinte forma:

$$D_2(\omega^2) = \frac{P_n^2(\omega) + P_n^2(-\omega)}{2} \quad (I-7)$$

Substituindo (I-7) em (I-5) e fazendo  $x_0^2 = A_0$ , (I-5) pode ser reescrita na

forma:

$$R_2(\omega) = \frac{2x_0^2\omega^{2k}}{P_n^2(\omega) + P_n^2(-\omega)} \quad (I-8)$$

o que garante que  $R_2(\omega)$  é uma função par positiva real;

3) São selecionadas as raízes no semi-plano lateral esquerdo do polinômio  $D_2(\omega^2)$  dado em (I-7) para construir o polinômio de Hurwitz do denominador de  $Z_2(s)$ ;

4) Usando o procedimento de Gewertz [129], a impedância  $Z_2(s)$  é determinada;

5) O TPG é calculado usando (I-3);

6) O TPG é maximizado variando os valores do vetor  $X = (x_0, x_1, x_2, \dots, x_n)$ .