Universidade do Estado do Rio de Janeiro

Centro de Tecnologia e Ciências

Faculdade de Engenharia

Flavio Considera El-Kareh

Algoritmos genéticos aplicados ao projeto de filtros com coeficientes em soma de potências de dois

Rio de Janeiro

2011

Flavio Considera El-Kareh

Algoritmos genéticos aplicados ao projeto de filtros com coeficientes em soma de potências de dois

Dissertação apresentada como requisito parcial para obtenção do título de Mestre, ao Programa de Pós-Graduação em Engenharia Eletrônica, da Universidade do Estado do Rio de Janeiro. Área de concentração: Sistemas Inteligentes.

Orientador: Prof. Dr. Lisandro Lovisolo

Rio de Janeiro
2011

Flavio Considera El-Kareh

Algoritmos genéticos aplicados ao projeto de filtros com coeficientes em soma de potências de dois

Dissertação apresentada, como requisito parcial para obtenção do título de Mestre, ao Programa de Pós-Graduação em Engenharia Eletrônica, da Universidade do Estado do Rio de Janeiro. Área de concentração: Sistemas Inteligentes.

Aprovado em: 29 de Março de 2011
Banca Examinadora:

Prof. Dr. Lisandro Lovisolo. (Orientador)
Faculdade de Engenharia - UERJ

Prof. Dr. Jorge Luis Machado do Amaral
Faculdade de Engenharia - UERJ

Prof.ª Dr.ª Karla Tereza Figueiredo Leite
Pontifícia Universidade Católica do Rio de Janeiro

Prof. Dr. José Antonio Apolinário Jr
Instituto Militar de Engenharia

Rio de Janeiro
2011

Agradecimentos

Agradeço primeiramente aos meus pais, Almir e Eliane, pelo amor, incentivo e apoio ao longo de toda minha vida. À minha querida irmã Anouk por todos esses anos de boa convivência. E à Paula, por me aturar durante durante os anos que nos conhecemos, por seu carinho e compreensão.

Agradeço ao meu orientador Prof. Lisandro Lovisolo, por acreditar em mim, me guiar e ajudar durante esta jornada. Sem esse apoio não seria possível terminar esta etapa. Sou grato principalmente pela amizade e pelas conversas durante os almoços no MBC e no café da tarde.

A todos os meus queridos primos, principalmente ao Eduardo, Júlia e Vitor pelos divertidos momentos que passamos juntos, e também a todos os meus tios pelos jantares e almoços em família sempre muito alegres e engraçados.

Agradeço aos meus amigos por me distraírem dos meus afazeres durante estes últimos anos, pois relaxar é preciso. Em especial aos meus colegas do mestrado Felipe “Barão”, Gabriel “Lion” e Rafael “JFC”, pelos momentos de descontração.

Finalmente agradeço a todas as pessoas que de alguma forma contribuíram para que eu completasse mais uma etapa da minha vida.

MUITO OBRIGADO!

RESUMO

EL-KAREH, Flavio Considera. **Algoritmos genéticos aplicados ao projeto de filtros com coeficientes em soma de potências de dois**. 2011. 160f. Dissertação (Mestrado em Engenharia Eletrônica) - Faculdade de Engenharia, Universidade do Estado do Rio de Janeiro, Rio de Janeiro, 2011.

Esta dissertação tem como objetivo aplicar um algoritmo genético (GA) ao projeto de filtros FIR com coeficientes quantizados representados em somas de potências de dois com sinal (SPT). Os filtros FIR apresentam configurações que permitem a obtenção de fase linear, atributo desejado em diversas aplicações que necessitam de atraso de grupo constante. A representação SPT, de fácil implementação em circuitos, foi discutida e uma comparação das representações SPT mínimas e canônicas foi feita, baseada no potencial de redução de operações e na variedade de valores representáveis. O GA é aplicado na otimização dos coeficientes SPTs do filtro, para que este cumpra as suas especificações de projeto. Foram feitas análises sobre o efeito que diversos parâmetros do GA como a intensidade de seleção, tamanho das populações, cruzamento, mutação, entre outros, têm no processo de otimização. Foi proposto um novo cruzamento que produz a recombinação dos coeficientes e que obteve bons resultados. Aplicou-se o algoritmo obtido na produção de filtros dos tipos passa-baixas, passa-altas, passa-faixas e rejeita-faixas.

Palavras-chave: Algoritmos genéticos (GA). Somas de potências de dois com sinal (SPT). Filtros FIR.

ABSTRACT

This work uses a genetic algorithm (GA) in the design of finite impulse response (FIR) filters with quantized coefficients represented in the signed-power-of-two (SPT) format. FIR filters presents linear phase and for that reason this type of filter is chosen when an application demands constant group delay. The SPT numbers, which are easy to implement into hardware, were discussed and a comparison between its minimal and canonical forms was made aiming at the potential reduction of arithmetical operations and the possibilities each one offered for number representation. The GA searches for the optimum filter coefficients that produce a filter according to its project specifications. Many analysis were made on the effects of changes made to the GA's parameters like selection intensity, population size, mutation, crossover among others. A new crossover operator was proposed, in which the filter's coefficients are repositioned produced a good results in filter optimization. The algorithm was implemented in the design of low-pass, high-pass, band-pass and band-stop filters.

Keywords: Genetic algorithms. Signed-power-of-two. FIR filters.

LISTA DE FIGURAS

LISTA DE TABELAS

SUMÁRIO

INTRODUÇÃO

O projeto de filtros é um tema de grande importância devido às suas diversas aplicações, como a filtragem dos sinais de um eletrocardiograma [16,86,89], a transmissão e recepção de sinais [13, 42], o processamento de imagens [48, 95] e áudio [27, 68], entre muitas outras.

Com o advento dos filtros digitais houve muita pesquisa neste assunto, pois estes permitem fácil implementação em circuitos [63,67]. Dentre os filtros digitais, os filtros FIR têm uma posição especial devido à sua simplicidade de projeto, baixa sensibilidade dos seus coeficientes e BIBO estabilidade, ambas intrínsecas, além de permitir a fácil obtenção de filtros de fase linear [6,24]. A última característica é muito importante em aplicações onde a fase do sinal deve ser mantida. Porém, a ordem de um filtro FIR é consideravelmente maior do que a ordem de um filtro IIR construído para a mesma resposta em magnitude [62, 74]. Assim, filtros FIR requerem uma quantidade maior de operações aritméticas, o que implica em mais componentes de circuitos [99]. Isso torna a sua implementação mais custosa, principalmente quando é necessário cumprir com especificações com estreitas bandas de transição e grande atenuação na banda de rejeição.

A utilização de filtros digitais implica na quantização dos coeficientes dos filtros [4], neste processo introduz-se inevitavelmente um ruído de quantização advindo do truncamento do valor do coeficiente para o valor mais próximo possível na representação escolhida [85]. Este erro provoca desvios na resposta em frequência do filtro quantizado, que podem levá-lo a uma configuração inadmissível para as especificações desejadas [24]. Nos últimos 50 anos muitos estudos foram feitos sobre o efeito da quantização nos filtros, o que produziu diversos métodos de otimização dos coeficientes quantizados para que os requerimentos das especificações fossem mantidos, sem que haja uma mudança muito grande na ordem do filtro ou no tamanho da palavra-código empregada [40, 80].

Para reduzir o custo de implementação de um filtro FIR pode-se reduzir a "complexidade" de seus coeficientes [29]. Esta redução se faz pelo encurtamento da palavra-código [99], ou por uma representação diferente dos números, ou ainda pela limitação da quantidade de operações aritméticas [93], esta última possível pela limitação da quantidade de dígitos diferentes de zero em uma representação, entre outra técnicas [94, 97]. Por isso, a representação por potências de dois com sinal (**SPT**) vem, nas últimas décadas, sendo muito utilizada devido às suas vantagens sobre a representação binária [38].

Os métodos de otimização empregados são em geral algoritmos de busca que procuram a combinação de coeficientes quantizados para um filtro de modo a que este produza uma resposta em frequência mais próxima possível das especificações [40]. Diversos algoritmos de busca rápida foram desenvolvidos uma vez que, para filtros, quanto maior sua ordem e a quantidade de dígitos de representação dos coeficientes, maior sua complexidade, o que inviabiliza a utilização de um método de busca exaustiva [25]. Nos dias de hoje, a obtenção de um bom filtro não é o único fator crítico ao se escolher uma determinada técnica de busca, muitas vezes a velocidade com a qual a resposta é obtida conta muito, como é o caso de aplicações em tempo (quase)real. Infelizmente, a busca por um bom filtro não é uma tarefa fácil e por isso os algoritmos de busca devem contemplar mecanismos de busca local e global, para poderem obter uma boa resposta sem estarem sujeitos a ficar estagnados em um máximo local [25].

Hoje temos diversos métodos, chamados de métodos estocásticos [31], que podem ser utilizados para gerar melhorias na resposta em frequência de um filtro FIR quantizado pela otimização de seus coeficientes, dentre eles podemos destacar o *simulated annealing*, o *Particle Swarm Optimization* [92], a alocação dinâmica de termos SPT [51], a quantização por sensibilidade do coeficiente [3, 81], os algoritmos genéticos (**GA**) [18], entre outras.

Os GAs pertencem ao grupo dos algoritmos evolucionários e utilizam o princípio da sobrevivência do indivíduo mais apto para encontrar suas soluções [88]. Aplicando conceitos como cruzamento e mutação a um grupo de possíveis soluções chamado de população, esses algoritmos percorrem o espaço de busca à procura do indivíduo mais apto [18]. Os GAs provaram ser algoritmos de busca muito robustos, capazes de encontrar boas soluções para problemas com grande espaço de busca, mesmo este sendo povoado por diversos máximos locais [99].

A demanda crescente de filtros digitais de fácil implementação e especificações de projeto cada vez mais exigentes, levou à combinação de técnicas de filtragem digital com métodos de otimização dos coeficientes quantizados dos filtros, com o intuito de melhorar o desempenho destes.

Objetivos

Nesta dissertação procurou-se desenvolver um algoritmo de projeto de filtro FIR com coeficientes quantizados com até 8 dígitos que satisfaça às mesmas especificações de um filtro de mesma ordem projetado com coeficientes densamente quantizados, de resposta próxima aos filtros de precisão infinita. Uma vez que a diferença de precisão entre estes dois filtros é grande, favorecendo os filtros de coeficientes quantizados com muitos dígitos, utilizamos a codificação SPT, que permite a implementação de somadores e subtratores simultaneamente, além de possíveis técnicas de redução de operações. Uma vez que a otimização dos coeficientes do filtro é uma tarefa que exige uma capacidade computacional extremamente grande utilizamos os GAs como método de busca dos coeficientes apropriados para que que os filtros produzidos estejam de acordo com as especificações de projeto.

Além disso, procuramos estabelecer os efeitos das variações dos diversos parâmetros do GA produzem na otimização filtros. Para isso diversos testes foram efetuados variando um determinado parâmetro por vez. Foram testados com isso, a influência do tamanho da população inicial e da aptidão desta, diferentes tipos e taxas de cruzamento e mutação entre outros. Como avaliação foi utilizado a progressão das curvas de aptidão dos indivíduos em cada teste de modo que pudesse ser analisado o comportamento dos indivíduos de uma população.

Aproveitamos o uso da representação SPT para avaliarmos o quão eficaz esta é quando comparada à representação binária. Além disso, foram feitas comparações entre diversas representações SPT, a representação padrão, uma representação mínima e a canônica. Testamos também uma técnica de redução da quantidade de operadores aritméticos, que consiste na limitação do número de dígitos SPTs diferentes de zero. Foi também feita uma análise da distribuição de valores em cada uma das representações SPTs.

Organização da Dissertação

Este trabalho foi dividido nesta introdução seguida de cinco Capítulos e a conclusão. No Capítulo 1 apresentaremos conceitos básicos de filtros FIR. Serão apresentadas características de filtros FIR, assim como seus subtipos. Em seguida, no Capítulo 2, falaremos sobre a quantização dos coeficientes, nos concentrando em seus efeitos sobre a resposta em frequência dos filtros. Serão vistas também algumas representações binárias assim como a representação SPT e suas variantes. Uma apresentação de GAs será feita no Capítulo 3, onde seu funcionamento e seus parâmetros serão explicados. Abordamos também a teoria dos padrões, que apresenta a base matemática para o funcionamento dos GAs. No Capítulo 4 descreveremos o algoritmo criado neste trabalho e faremos a união entre filtragem, representação SPT e GA para testar seu funcionamento. Serão discutidos os efeitos dos parâmetros do GA e da representação numérica na otimização de filtros FIR. Com o conhecimento adquirido nos quatro primeiros Capítulos, produzimos filtros para diversas especificações com o objetivo de compará-los à filtros de coeficientes contínuos.

1 FILTROS DIGITAIS

Este Capítulo aborda filtros de resposta finita ao impulso, chamados de filtros FIR (*finite impulse response*). Por terem seus coeficientes de fácil projeto, terem fase linear e garantirem a BIBO (*Bounded Input Bounded Output*) estabilidade, filtros FIR são utilizados em diversas aplicações [62]. A simplicidade de projeto e implementação dos filtros FIR permite sua fácil utilização as mais diversas técnicas aplicáveis aos filtros sem que haja necessidade de fazer muitas mudanças ao projeto base. Este tipo de filtro digital é propício para o trabalho desenvolvido, uma vez que teremos que realizar a quantização dos coeficientes do filtro e a otimização dos mesmos com algoritmos genéticos para obtermos filtros dentro de especificações pré-determinadas, como será visto nos Capítulos 2, 3 e 4.

1.1 Estruturas dos Filtros Digitais

A relação entrada e saída, no domínio do tempo, do filtro FIR causal de comprimento M é caracterizada pela equação em diferenças [24]

$$y(n) = \sum_{l=0}^{M} h(l)x(n-l), \tag{1}$$

onde $y(n)$ e $x(n)$ são a saída e entrada, respectivamente, e h(l) é a resposta do filtro. A equação 1 é mostrada em diagrama de blocos na Figura 1 e consiste na convolução entre a entrada x e a resposta ao impulso do filtro h.

Aplicando a transformada Z à equação 1 obtemos a relação entre entrada $X(z)$ e a saída $Y(z)$ chamada de função de transferência $H(z)$

$$H(z) = \frac{Y(z)}{X(z)} = \sum_{l=0}^{M} h(l)z^{-l}. \tag{2}$$

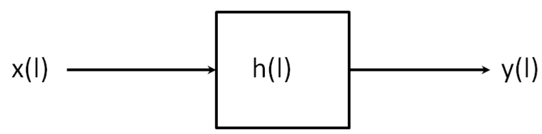


Figura 1 Diagrama de blocos de um filtro no domínio do tempo

Já a resposta em frequência de um filtro FIR é um polinômio obtido quando $H(z)$ é avaliado em $z = e^{-j\omega}$ e é dada por

$$H(e^{j\omega}) = \sum_{l=0}^{M} h(l)e^{-j\omega l}. \tag{3}$$

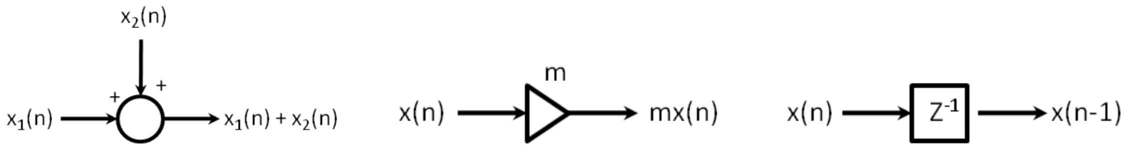
$H(e^{j\omega})$ pode ser representado por seu módulo $|H(e^{j\omega})|$ e fase $\Theta(\omega)$ obtendo

$$H(e^{j\omega}) = |H(e^{j\omega})|e^{j\Theta(\omega)} \tag{4}$$

A magnitude e fase de um filtro são características muito importantes no projeto de filtros, por definirem as faixas de operação deste assim como seu comportamento. O projeto de um filtro deve obedecer a certos pré-requisitos impostos pela aplicação. Esses parâmetros são chamados de especificações do filtro e normalmente contêm as faixas de frequências nas quais um filtro deve permitir a passagem do sinal, chamadas de bandas de passagem e as faixas na qual o sinal deve ser eliminado, chamadas de bandas de rejeição [66]. Um filtro deve obedecer o mais fielmente possível às suas especificações. Em alguns casos as especificações de um filtro são ideais e por isso apresentam descontinuidades em sua magnitude. Um filtro ideal é um filtro não-causal de comprimento infinito e portanto irrealizável [6, 24], o que torna impossível a realização rigorosa das especificações por parte de um filtro real, de comprimento finito.

O que ocorre normalmente é o projeto de um filtro com magnitude suficientemente próxima das especificações para que a aplicação não seja comprometida. Sendo assim, a resposta em magnitude de um filtro é de suma importância na avaliação do desempenho deste.

Mesmo sendo muito importante, a magnitude não é a única característica de interesse, se fosse o caso optaríamos pela utilização de filtros IIR que têm melhor resposta em magnitude do que filtros FIR de mesma ordem [6, 24, 62, 99]. Outro fator importante é a fase do filtro, que afeta todas as frequências do sinal filtrado. No projeto de filtros além das especificações de magnitude é as vezes necessário obedecer à certas especificações de fase, uma vez que filtros de fase não linear afetam diferentemente as frequências do sinal.



(a) Somador (b) Multiplicador (c) Atraso

Figura 2 Elementos básicos do diagrama de blocos

Este efeito é chamado de atraso de grupo $\tau(\omega)$ e é definido por

$$\tau(\omega) = -\frac{d\Theta(\omega)}{d\omega}, \tag{5}$$

onde $\tau(\omega)$ fornece o atraso para a frequência ω em número de amostras. Algumas aplicações são sensíveis ao atraso de grupo e por isso ao projetarmos filtros devemos garantir fase $\Theta(\omega)$ linear para então obtermos atraso de grupo constante em toda faixa de frequências. Em filtros FIR é fácil garantir a linearidade da fase, basta garantir uma certa simetria nos coeficientes do filtro [6, 24], como veremos à frente na Subseção 1.1.2.

1.1.1 Forma Direta

A função de transferência da equação 2 pode ser implementada em diversas formas usando como elementos básicos somadores, multiplicadores e atrasos [24], mostrados na Figura 2.

A implementação derivada diretamente da equação 2 é chamada de forma direta uma vez que os seus multiplicadores são obtidos a partir da função de transferência do filtro. A representação esquemática da forma direta é mostrada na Figura 3.

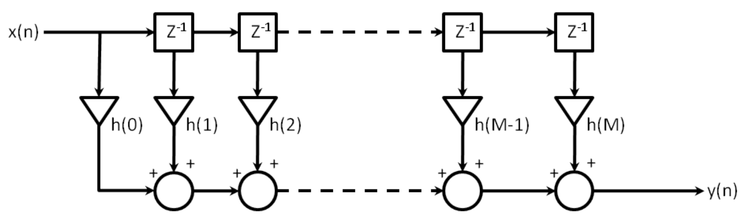


Figura 3 Forma direta para filtros FIR

Além da forma direta existem outras formas muito usadas como a forma de cascata, a forma polifásica e a forma de fase linear [62, 67]. As duas primeiras não fazem parte do escopo deste trabalho e podem ser estudadas em [6,24,62,67], a última será vista à seguir na subseção 1.1.2.

1.1.2 Filtros FIR de Fase Linear

É importante salientarmos a família de filtros FIR composta pelos filtros de fase linear, pois esta é uma característica que faz com que os filtros FIR muitas vezes sejam preferidos a filtros IIR de mesma ordem, que normalmente possuem magnitude mais próxima às especificações ao custo de fase não linear [67]. O atraso de grupo é uma propriedade dos filtros, derivada de sua resposta em fase, conforme a equação 5. Filtros com fase linear implicam em filtros com atraso de grupo τ constante. Em filtros FIR nota-se que há necessidade da existência de simetria ou assimetria entre os coeficientes dos filtros para garantir fase linear e logo atraso de grupo constante [6]. Esta propriedade permite a redução de até metade da quantidade de multiplicadores utilizados na forma direta [62]. Em [24] são apresentados cálculos envolvendo a obtenção das formas de fase linear, dos quais nos ateremos apenas ao que é exibido na Tabela 1.

Tipo	Ordem (M)	$h(n)$	$H(e^{j\omega})$	$\Theta(\omega)$	τ
I	Par	Simétrico	$e^{-j\omega\frac{M}{2}}\sum_{i=0}^{\frac{M}{2}} a(i)cos(\omega i)$ $a(0) = h(\frac{M}{2})$; $a(i) = 2h(\frac{M}{2} - m)$	$-\omega\frac{M}{2}$	$\frac{M}{2}$
II	Ímpar	Simétrico	$e^{-j\omega\frac{M}{2}}\sum_{i=1}^{\frac{M+1}{2}} b(i)cos\left[\omega\left(i-\frac{1}{2}\right)\right]$ $b(i) = 2h\left(\frac{M+1}{2} - i\right)$	$-\omega\frac{M}{2}$	$\frac{M}{2}$
III	Par	Antissimétrico	$e^{-j(\omega\frac{M}{2}-\frac{\pi}{2})}\sum_{i=1}^{\frac{M}{2}} c(i)sen(\omega i)$ $c(i) = 2h\left(\frac{M}{2} - i\right)$	$-\omega\frac{M}{2} + \frac{\pi}{2}$	$\frac{M}{2}$
IV	Ímpar	Antissimétrico	$e^{-j(\omega\frac{M}{2}-\frac{\pi}{2})}\sum_{i=1}^{\frac{M+1}{2}} d(i)sen\left[\omega\left(i-\frac{1}{2}\right)\right]$ $d(i) = 2h\left(\frac{M+1}{2} - i\right)$	$-\omega\frac{M}{2} + \frac{\pi}{2}$	$\frac{M}{2}$

Tabela 1 Principais características de filtros FIR de fase linear

Cada um dos quatro tipos de filtros de fase linear citados na Tabela 1 têm limitações quanto ao tipo de implementação que se deseja fazer [6]. Isto é, nem sempre todos os tipos de filtros de fase linear são adequados para o projeto de filtros das quatro configurações básicas de filtros [24]: passa-baixa, passa-alta, passa-faixa ou rejeita-faixa. A Tabela 2 mostra os projetos possíveis, marcados com um “Sim”. A impossibilidade de realizá-los, marcadas por um “Não”, se deve às limitações impostas a $H(e^{j\omega})$ em cada um dos quatro tipos de filtros de fase linear.

	Tipo de filtro de fase linear			
Tipo de filtro	**Tipo I**	**Tipo II**	**Tipo III**	**Tipo IV**
Passa-baixa	Sim	Sim	Não	Não
Passa-alta	Sim	Não	Não	Sim
Passa-faixa	Sim	Sim	Sim	Sim
Rejeita-Faixa	Sim	Não	Não	Não

Tabela 2 Adequação do tipo de filtro de fase linear às quatro configurações básicas de filtros

Da Tabela 2 tiramos que os filtros de fase linear tipo I são adequados ao projeto dos quatro tipos básicos. Nos Capítulos 4 e 5 utilizaremos a configuração de filtros lineares tipo I para o projeto dos filtros usando algoritmos genéticos para a otimização de seus coeficientes.

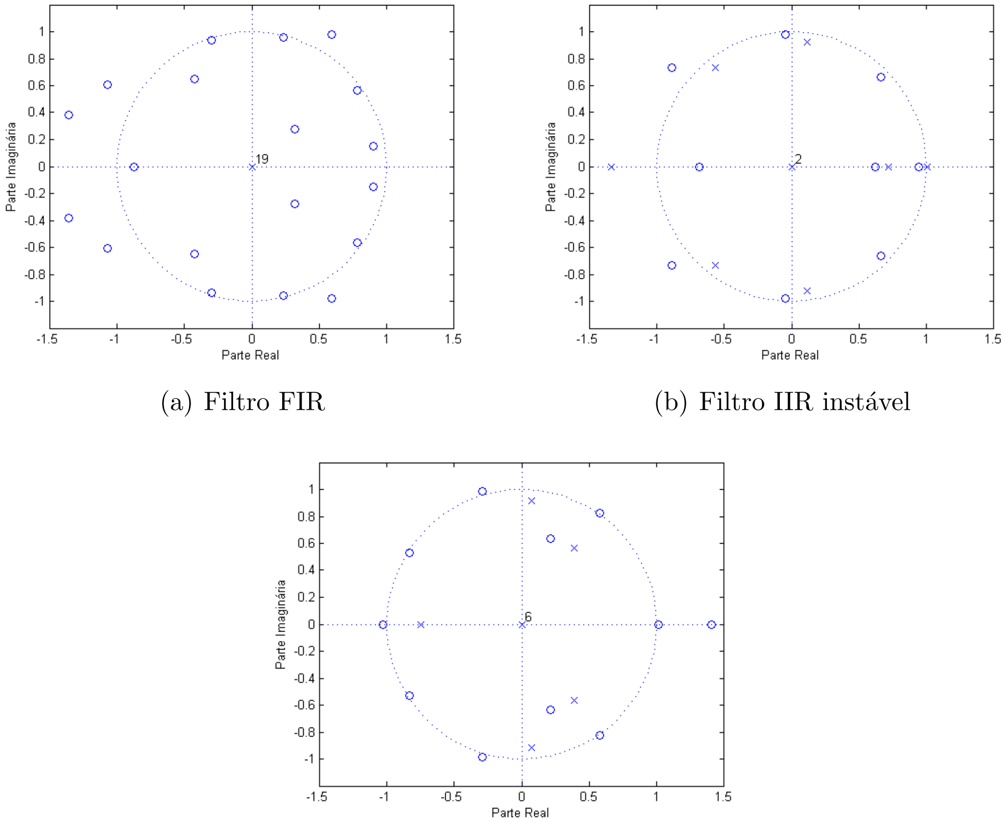
1.2 BIBO Estabilidade

A BIBO (*Bounded Input Bounded Output*) estabilidade de um sistema implica que para entradas limitadas em amplitude as saídas também serão limitadas [6]. Em um sistema de entrada $x(n)$ e saída $y(n)$ a condição de estabilidade é definida por [62]

$$|x(n)| < B_x \Rightarrow |y(n)| < B_y \tag{6}$$

onde B_x e B_y são finitos.

Em um filtro esta estabilidade está associada à posição de seus polos em relação ao círculo unitário do plano complexo [24]. A Figura 4 mostra três distribuições de zeros e polos no círculo unitário. Nota-se que há nesta figura multiplicidade de polos na origem. Para o caso de um filtro causal, se todos os polos de um determinado filtro estiverem dentro da circunferência, diz-se que ele é BIBO estável [6]. Mas caso um ou mais polos estejam localizados na borda do círculo ou no exterior deste o filtro é instável. Como os filtros FIR possuem todos os seus polos localizados em $z = 0$ [6] eles são sempre estáveis e não há necessidade de preocupação com estabilidade de seus coeficientes durante o projeto.



(a) Filtro FIR

(b) Filtro IIR instável

(c) Filtro IIR estável

Figura 4 Distribuição de polos (x) e zeros (o) em filtros estáveis e instáveis

A estabilidade de um filtro é uma característica necessária uma vez que o objetivo do filtro é permitir a passagem do sinal em certas faixas de frequências o que equivale a permitir a passagem de uma parte da energia deste sinal. Um sistema instável tende a produzir um aumento descontrolado na energia do sistema o que acarreta respostas indesejáveis.

O fato de filtros FIR serem sempre estáveis [6, 24, 62] é uma outra vantagem que estes filtros têm sobre filtros IIR, por mais que haja soluções para garantir a estabilidade de filtros IIR [6, 24, 62, 67].

1.3 Projeto de Filtros Digitais

O processo de obtenção de filtros visando atender uma dada especificação é chamado de aproximação de filtros, pois consiste na obtenção de um filtro real com resposta em frequência próxima da resposta do filtro ideal que atenda às especificações dadas. Existem diversos métodos de aproximação de filtros divididos em três principais grupos que são: funções janela, amostragem e otimização; descritos em [6, 12, 24, 76, 78, 82]. Ressaltaremos e explicaremos rapidamente os métodos dos mínimos quadrados ponderados (*Weighted-Least-Square* ou WLS) e o método computacional de Parks-McClellan [77], ambos pertencentes ao grupo dos métodos de otimização, pois o princípio de otimização do primeiro é utilizado na construção do método empregado neste trabalho e o segundo foi empregado como base de comparação para os resultados deste trabalho, uma vez que esse é bem conhecido e muito empregado no projeto de filtros. Ambos os métodos utilizam como objetivo a minimização da função erro $E(\omega)$ dada por

$$E(\omega) = W_q(\omega)[D_q(\omega) - P(\omega)], \tag{7}$$

onde $W_q(\omega)$ são pesos, $D_q(\omega)$ é a resposta desejada do filtro e $P(\omega)$ é a resposta em magnitude de um dos quatro tipos básicos de filtros. Se desconsiderarmos o trecho de $H(e^{j\omega})$ referente à fase visto na Tabela 1, obtemos os resultados para $P(\omega)$ mostrados na Tabela 3.

A maior diferença entre os dois está na abordagem utilizada para a minimização da função erro como veremos a seguir nas Subseções 1.3.1 e 1.3.2.

1.3.1 Método dos Mínimos Quadrados Ponderados

O método dos Mínimos Quadrados Ponderados, do inglês *Weighted-least-squares* (WLS), tem como objetivo a minimização da energia da função de erro $E(\omega)$ ou seja

$$min\left(||E(\omega)||_2^2\right) = min\left(\int_0^\pi |E(\omega)|^2 d\omega\right), \tag{8}$$

Tipo	$P(\omega)$
I	$\sum_{i=0}^{\frac{M}{2}} a(i)cos(\omega i)$ $a(0) = h(\frac{M}{2})$; $a(i) = 2h(\frac{M}{2} - m)$
II	$\sum_{i=1}^{\frac{M+1}{2}} b(i)cos\left[\omega\left(i - \frac{1}{2}\right)\right]$ $b(i) = 2h\left(\frac{M+1}{2} - i\right)$
III	$\sum_{i=1}^{\frac{M}{2}} c(i)sen(\omega i)$ $c(i) = 2h\left(\frac{M}{2} - i\right)$
IV	$\sum_{i=1}^{\frac{M+1}{2}} d(i)sen\left[\omega\left(i - \frac{1}{2}\right)\right]$ $d(i) = 2h\left(\frac{M+1}{2} - i\right)$

Tabela 3 Valores de $P(\omega)$ para os quatro tipos básicos de filtros

que para frequências discretas pode ser aproximado para

$$||E(\omega)|| \approx \frac{1}{\bar{K}\bar{M}} \sum_{k=1}^{\bar{K}\bar{M}} |E(\omega_k)|^2, \tag{9}$$

onde M é a ordem do filtro e K é recomendável estar entre 8 e 16 [24]. Já $\bar{K}$ e $\bar{M}$ equivalem à K e M quando a banda de transição não é considerada. Neste trabalho, o produto $KM = 512$, ou seja, utilizamos 512 amostras de frequências para todo o espectro. Para mais informações e aplicações do método WLS ver [6, 24]. A equação 9 serviu de inspiração para a modelagem da função de erro do algoritmo, apresentada na equação (73).

1.3.2 Método FIR Parks-McClellan

Este método, doravante chamado de filtro FIRPM, é um método geração de filtros por otimização apresentado originalmente em [77]. Assim como no método WLS ocorre a otimização da função erro $E(\omega)$ porém neste caso ela é dada por

$$min\left(||E(\omega)||_\infty\right) = min\left(max(|E(\omega)|)\right). \tag{10}$$

Uma vez definida a função erro, basta utilizá-la em um algoritmo de otimização para gerar filtros contínuos. O método FIRPM tem suas rotinas detalhadas em [24, 77]. A implementação deste no MATLAB é feita utilizando a função *firpmord*, que retorna a ordem, bandas, ganhos e pesos que deverão ser os parâmetros de entrada da função *firpm*, que por sua vez retorna o filtro para as entradas fornecidas. Lembrando que o filtro produzido desta maneira é um filtro de coeficientes contínuos, ou seja, de precisão infinita.

Como o objetivo deste trabalho é a produção de filtros com coeficientes discretos, ouseja, quantizados com um número finito de bits, devemos levar em conta a diferença de precisão entre os coeficientes do filtro, o que será feito no Capítulo 2.

2 EFEITOS DA PRECISÃO FINITA

A eficácia da implementação de um algoritmo aritmético depende amplamente de como seus dados numéricos estão armazenados em memória nos registradores de um computador digital [39]. Diferentes representações numéricas resultam em projetos com desempenhos diferentes. A escolha apropriada de um sistema de numeração impacta no processamento dos dados, nos métodos numéricos aplicados e na precisão dos valores obtidos [79]. Como é implementada uma aritmética numérica de precisão finita, em computadores digitais, toda representação numérica utilizada deve se restringir a números com quantidade finita de dígitos.

Na prática, um sistema de processamento de sinais é implementado por um programa num computador digital, usando um processador, por *hardware* específico para a aplicação. Nestes casos, erros de quantização advindos da precisão finita estão sempre presentes, seja devido à conversão do sinal analógico para digital, onde temos a quantização das amplitudes do sinal analógico por valores pré-definidos, chamados erros de quantização da entrada; seja erros na resposta em frequência dos filtros devido ao tamanho finito da palavra-código na representação de constantes multiplicativas, erros de quantização do produto; e seja quando informações internas do sistema são quantizadas antes ou depois de adições em sequência, erros de quantização dos coeficientes [24]. Esta seção, aborda a representação de números e sua manipulação, conforme utilizados neste trabalho.

2.1 Representação de Coeficientes de Filtros

A representação escolhida para os coeficientes do filtro influi na magnitude do erro de quantização. Outro fator a ser levado em conta durante a representação dos coeficientes do filtro é o tamanho da palavra código que influi muito no consumo de energia por parte do processador digital e na máxima taxa de amostragem possível de ser obtida [72, 98]. Nesta seção consideraremos números pertencentes ao intervalo (-1,1). Sendo assim utilizaremos números fracionários representados por bases com potências negativas.

2.1.1 Sinal e Magnitude

Esta representação de números consiste em um número binário representando uma magnitude precedido de um bit definindo o seu sinal conforme a equação 11. Onde s_x é o bit referente ao sinal, 0 para números positivos e 1 para negativos, e a sequência de bits seguinte representa um número em base 2. Neste modelo, o bit utilizado para a codificação de sinal não influi na precisão do número binário uma vez que ele não altera o módulo do número.

$$x_m = s_x x_1 x_2 ... x_{n-1} x_n \tag{11}$$

onde $x_i = k2^{-i}, \; k = \{0, 1\}$. A conversão entre decimais e binários com sinal e magnitude é dada por

$$[x]_{10} = s_x \sum_{i=1}^{N} k2^{-i} \tag{12}$$

A representação em sinal e magnitude é eficiente para a implementação de multiplicações [39].

2.1.2 Complemento de dois

Outra representação muito utilizada para números binários é a representação em complemento de dois nessa representação o número apresenta o formato

$$x_{2c} = \begin{cases} B(x), & \forall x \geq 0 \\ B(2 - |x|), & \forall x < 0 \end{cases}$$

onde $B(x) = x_1 x_2 ... x_n$ e $x = B^{-1}(x_1 x_2 ... x_n) = x_1 + x_2 2^{-1} + ... + x_n 2^{-n}$. Sendo assim para $x > 0$ a representação em complemento de dois é idêntica à representação sinal e magnitude vista anteriormente excluindo o bit de sinal. De modo geral a representação de um número em complemento de dois é dada por

$$x = -s_x + x_1 2^{-1} + x_2 2^{-2} + ... + x_n 2^{-n} \tag{13}$$

A representação em complemento de dois é a mais utilizada em projetos [24] e por isso é a base para conversões entre representações [43]. A representação em complemento de dois é efetiva para implementações de adições [24].

2.1.3 Potência de dois com sinal

A representação de potência de dois com sinal, ou **SPT** (*Signed Power-of-Two*), é uma representação ternária onde cada dígito de um número pertence ao conjunto $\{-1, 0, 1\}$. O uso dessa representação é tão direto quanto o da binária uma vez que para operar os números negativos basta implementar um subtrator binário ao invés de um somador binário. A flexibilidade advinda do uso de subtratores permite a representação de números com a utilização de menos dígitos diferentes de zero se comparada à representação binária [80]. Um exemplo pode ser dado com o número 7 que em binário com quatro dígitos é representado pela sequência de bits 0111, que apresenta 3 dígitos diferentes de zero, enquanto em SPT uma das representações deste número pode ser dada pela sequência $100\bar{1}$, onde $\bar{1}$ representa o número -1, nesta ultima sequência de dígitos temos 2 deles diferentes de zero. Esta representação é muito usada atualmente por reduzir o número de dígitos não-zero de uma constante, em média essa redução é da ordem de 33% se comparada à representação binária [69]. Essa redução implica na eliminação de somadores e multiplicadores do sistema o que reduz o tempo de processamento e o custo de implementação em circuito [50]. A implementação de um multiplicador genérico é particularmente custosa, por isso é interessante transformar a multiplicação em uma sequência de adições e deslocamentos [15, 40]. Sendo assim multiplicações por potências de dois podem ser obtidas facilmente em um circuito dedicado bastando deslocar os dígitos do valor para a direita em uma certa quantidade, permitindo a representação de diversos coeficientes de filtros com baixo custo computacional [80].

A seguir faremos um comparativo entre números binários e SPT visando suas vantagens. A representação de um número X em SPT com Q dígitos ternários $k(i)$ é dada por

$$X = \sum_{i=0}^{Q} k(i)2^{-i}, \qquad k(i) \in \{-1, 0, 1\}, \qquad Q \leq i \leq L - 1. \tag{14}$$

Enquanto para números binários de mesma precisão 2^{-Q} e mesmo número de dígitos Q

temos a representação

$$Y = \sum_{i=0}^{Q} 2^{-i}. \tag{15}$$

Nota-se que Q dígitos, com três possibilidades cada, podemos obter com números SPTs um número de combinações dado por

$$N_{spt} = \sum_{i=0}^{Q} 3^{i}, \tag{16}$$

ou seja,

$$N_{spt} = 3^{Q}. \tag{17}$$

Em números binários as possibilidades de representação são menores e são dadas por

$$N_{bin} = \sum_{i=0}^{Q} 2^{i}, \tag{18}$$

O que resulta em

$$N_{bin} = 2^{Q}. \tag{19}$$

Porém os números SPTs representam potências de base 2 positivas e negativas, logo a quantidade de valores distintos que um número SPT pode assumir é

$$N_{val} = \sum_{i=0}^{Q} 2^{i} + \sum_{i=1}^{Q} 2^{i}. \tag{20}$$

onde o somatório da esquerda é referente à quantidade de potências positivas e inclui o zero e o somatório da direita refere-se à quantidade de potências negativas sem o zero. Simplificando a equação (20) obtemos

$$N_{val} = 2\sum_{i=0}^{Q} 2^{i} - 1, \tag{21}$$

que por sua vez pode ser reescrita como

$$N_{val} = \sum_{i=0}^{Q} 2^{i+1} - 1. \tag{22}$$

Por fim obtemos

$$N_{val} = 2^{Q+1} - 1. \tag{23}$$

Se $Q \geq 2$, temos das equações (17) e (23) que $N_{spt} > N_{val}$. Ou seja, existe uma quantidade de números SPTs maior do que a quantidade de valores representáveis e portanto existem números com mais de uma representação possível. Se compararmos a quantidade de valores representáveis com SPTs N_{val} com a quantidade de valores representáveis com números binários N_{bin} temos

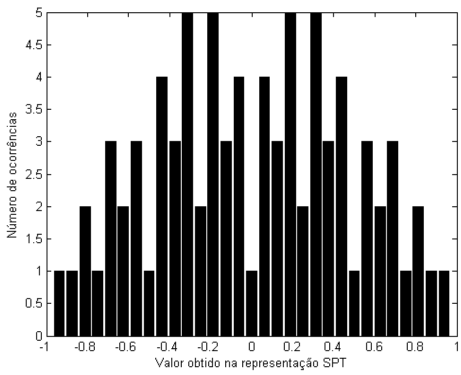
$$N_{val} - N_{bin} = 2^{Q} - 1, \tag{24}$$

onde essa quantidade equivale a todos os números negativos que podem ser representados com um número SPT de mesmo tamanho que um binário. Esta é a primeira vantagem dos números SPTs sobre os números binários. Uma importante avaliação, que será usada futuramente na Subseção 3.2.1, consiste na quantidade de números em excesso, que é obtida subtraindo-se (23) de (17) o que produz

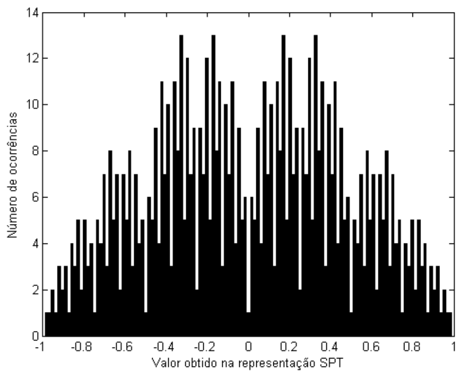
$$N_{rep} = 3^{Q} - 2^{Q+1} + 1, \tag{25}$$

onde $Q \geq 1$.

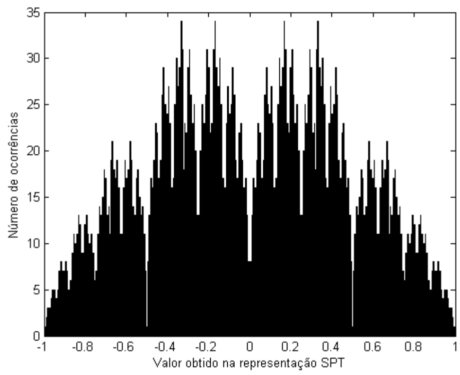
Como os números SPTs têm repetições podemos imaginar uma função de densidade de valores representados. A Figura 5 mostra esta distribuição para números SPTs, no intervalo $I_1 = (-1; 1)$, representados com 4, 6 e 8 dígitos. Podemos notar que como era de se esperar quanto menor o número de dígitos na representação, menor o número de valores representáveis e portanto mais esparsamente distribuídas são as distribuições. Nota-se que nas 3 curvas de densidade da Figura 5 uma concentração dos pontos nos intervalos $(-0,5; 0)$ e $(0; 0,5)$,o que nos faz concluir que a representação em SPT, com potências negativas de 2, têm maior ocorrência de números nestes intervalos. Nessa mesma figura estão marcados, no eixo das abcissas, pontos que representam os valores obtidos na representação SPT. Vemos que o espaçamento entre estes pontos são iguais, e de tamanho igual à precisão 2^{-Q} da representação, o que mostra a capacidade desta representação de exibir diversos valores no intervalo I_1, estando limitada apenas por sua precisão.



(a) 4 dígitos



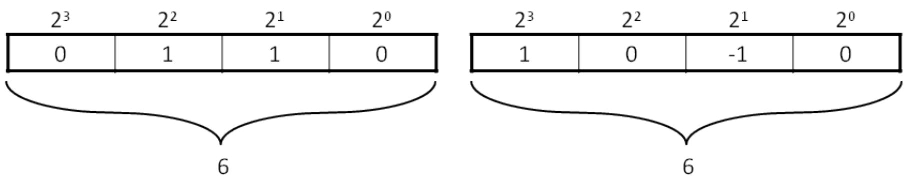
(b) 6 dígitos



(c) 8 dígitos

Figura 5 Densidade de distribuição dos números SPT

A partir dessas observações surgiu a representação mínima em dígitos com sinal, MSD (*Minimal Signed Digit*), ou seja, a representação que requer a quantidade mínima de termos diferentes de zero [70]. Uma delas em especial chamada de representação canônica de dígito com sinal ou CSD (*Canonic Signed Digit*) [36, 37] permite garantirmos a existência de uma única representação para um dado número, porém aumentamos a quantidade de dígitos SPT necessários para a representação em relação a outra representação não canônica. Existem diversos métodos de conversão de números binários para CSD [9, 43, 52, 79], uma vez que a representação única é muito interessante para monitoramento e para garantir o menor número de operações necessárias. Porém [69] aponta que o fato de termos em CSD serem únicos, os tornam bons para implementação, de algoritmos e não na implementação de hardware. Sendo assim uma representação MSD, que seja não canônica, seria mais adequada por dar maior flexibilidade ao projeto de filtros, pois a redundância pode acarretar circuitos menores do que aqueles obtidos através da



(a) Representação MSD (b) Representação CSD

Figura 6 Exemplos de representações mínimas em potência de dois com sinal

representação em CSD se uma boa representação MSD for escolhida. Do ponto de vista da implementação, a representação CSD e as representações MSD acarretam a mesma complexidade [15].

2.1.4 Representações Mínimas e a Base Canônica

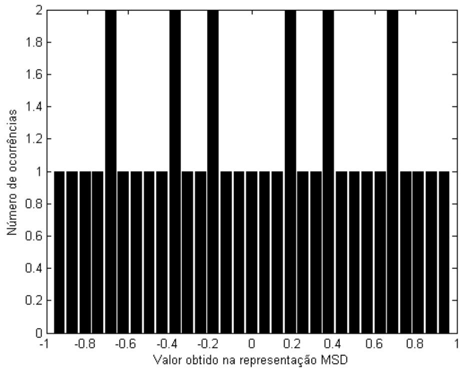
Uma representação mínima, MSD, é um código onde a representação dos números possui o menor número de dígitos diferentes de zero possível [70]. Porém, para representações SPT pode haver mais de uma representação mínima possível para um determinado número. Por isso, convencionou-se que a representação mínima que não possui dígitos diferentes de zero adjacentes é a forma canônica [37]. Ou seja, se uma sequência de Q dígitos, representativa de um número, obedecer à forma [99]

$$x(i) * x(i+1) = 0, \qquad \forall i \in \{1, 2, ..., Q\} \tag{26}$$

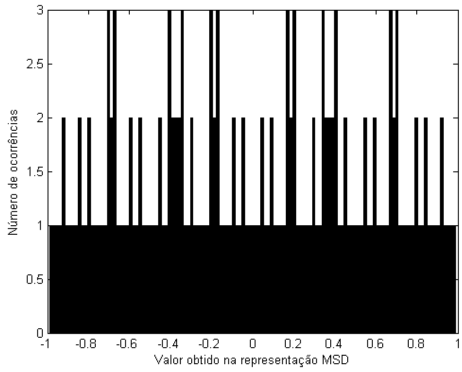
então esse número está na forma canônica.

O número de somadores ou subtratores de um circuito implementado a partir de uma representação CSD é $nD - 1$, onde nD é o número de dígitos não-zero na representação. A Figura 6 mostra duas representações SPT, nota-se que a representação canônica não permite a existência de números diferentes de zeros adjacentes ao contrário das representações mínimas não-canônicas. Devido a esta restrição existente na representação CSD, a quantidade de números que nela pode ser representada é inferior à quantidade de números nas representações MSD quando limitamos as duas representações ao mesmo número de dígitos.

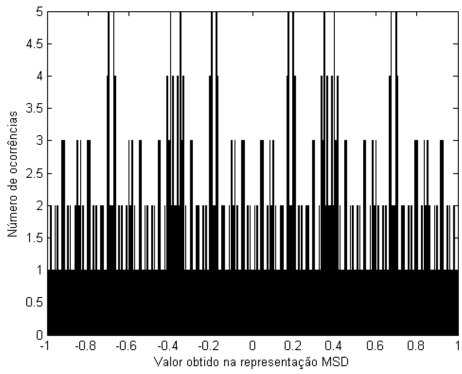
Para representações MSD, temos a função densidade de distribuição dos valores representáveis dada pela Figura 7. Quando comparamos esta densidade de distribuição com a densidade de números SPT sem representação mínima a primeira coisa que reparamos é o quão mais uniforme é a distribuição MSD. Enquanto para um número de 4 dígitos SPT temos até 5 repetições de um mesmo número, para um número de 4 dígitos MSD a quantidade de repetições máxima é de 2. Análises semelhantes podem ser feitas para números de 6, 8 ou mais dígitos. A semelhança entre as duas representações está na quantidade de valores representáveis que são as mesmas.



(a) 4 dígitos



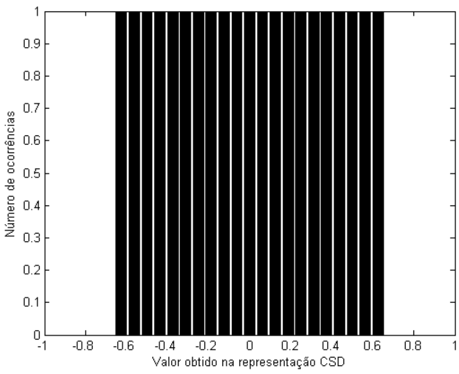
(b) 6 dígitos



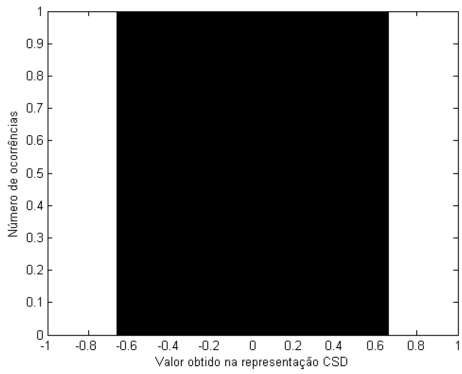
(c) 8 dígitos

Figura 7 Densidade de distribuição dos números MSD

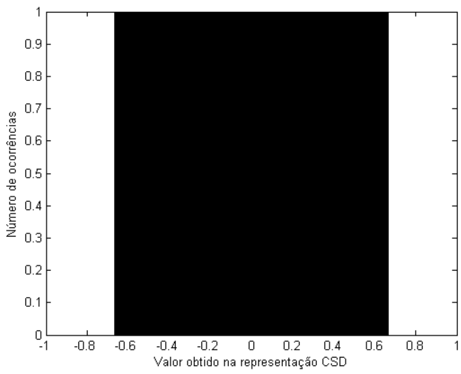
Quando olhamos para a densidade de distribuição dos valores representáveis de um número CSD, apresentada na Figura 8, notamos uma grande diferença em relação às representações SPT e MSD. Primeiramente a função densidade é homogênea de valor 1, ou seja, como era de se esperar não há repetição de valores. Em seguida, a quantidade de valores representáveis é notavelmente inferior às outras duas representações. Por fim, enquanto as representações SPT e MSD podem representar valores no intervalo I_1, tendo como limitação sua precisão 2^{-Q}, a representação CSD está limitada à representação de números dentro de um intervalo I_2 que equivale a aproximadamente 65% do intervalo I_1, contado da origem em direção às extremidades [80]. Ou seja, $I_2 \cong [-0,65; 0,65]$, novamente limitado pela precisão 2^{-Q}, o que implica em uma menor variedade de representação de valores para números CSD.



(a) 4 dígitos



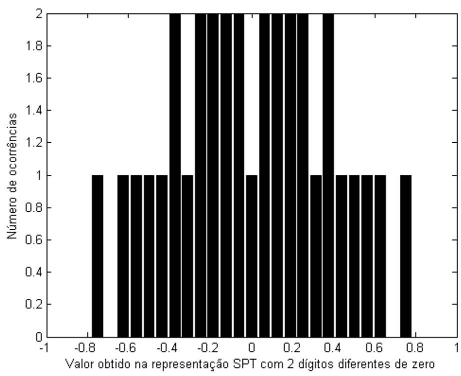
(b) 6 dígitos



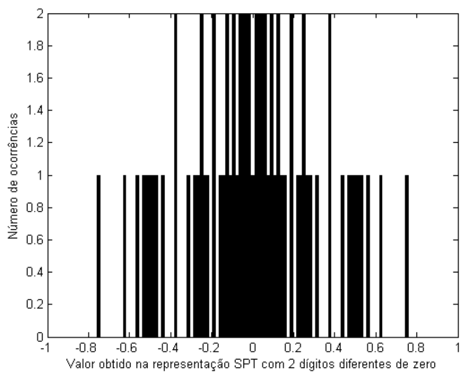
(c) 8 dígitos

Figura 8 Densidade de distribuição dos números CSD

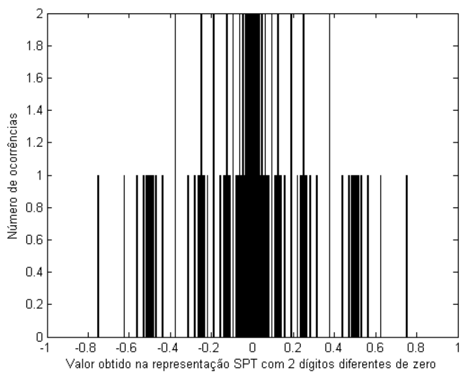
Quando trabalhamos com números SPTs é possível utilizarmos algoritmos e técnicas para reduzir o número de somadores e multiplicadores de um determinado circuito e assim reduzir seu custo de implementação [96]. Uma delas consiste na limitação de dígitos diferentes de zero durante a representação de um número em SPT, visando forçar uma redução do número de operações aritméticas. Seja um número representado por Q dígitos com limitação da quantidade de dígitos diferentes de zero nD. Porém, ao fixarmos nD impomos também uma nova limitação dos números representáveis, em outras palavras teremos um intervalo I_3 que será uma fração de $I_1 = [-1; 1]$. Mas ao contrário do que ocorre com a representação CSD, que também tem um intervalo de representação dos números I_2 reduzido, esta limitação implica também em uma distribuição não uniforme dos valores dentro deste intervalo. Mesmo se fixarmos o número de dígitos diferentes de zero nD nesta representação e aumentarmos o número total de dígitos do coeficiente, a não uniformidade da distribuição permanece. A Figura 9 apresenta a densidade de distribuição assim como a distribuição de valores para a representação SPT para números representados por 4, 6 e 8 dígitos quando o número de dígitos diferentes de zero é $nD = 2$. A Figura 10 apresenta a densidade e distribuição de valores para a representação MSD quando $nD = 2$.



(a) 4 dígitos



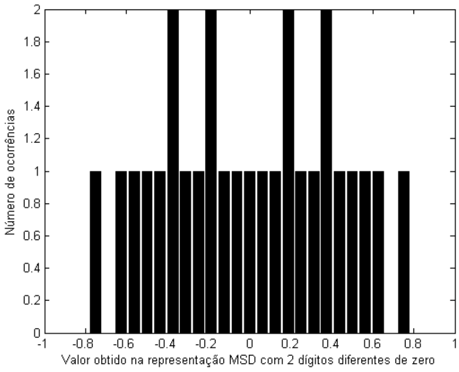
(b) 6 dígitos



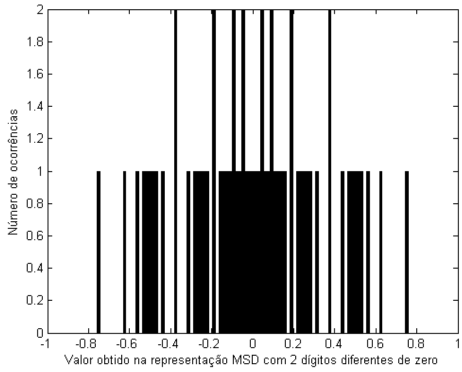
(c) 8 dígitos

Figura 9 Densidade de distribuição dos números SPT quando fixamos $nD = 2$

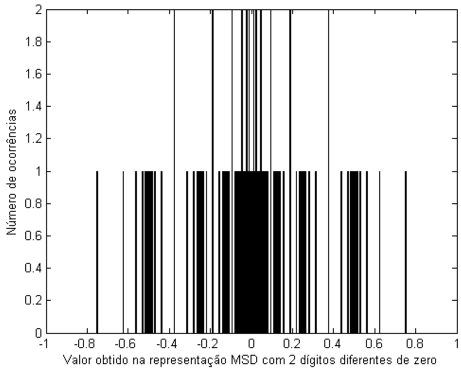
A Figura 11 apresenta a densidade e distribuição de valores de uma representação CSD quando $nD = 2$. Nota-se que na Figura 11(a) a distribuição é a mesma da Figura 8(a) uma vez que para 4 dígitos o número máximo de números diferentes de zero é 2. Uma



(a) 4 dígitos



(b) 6 dígitos



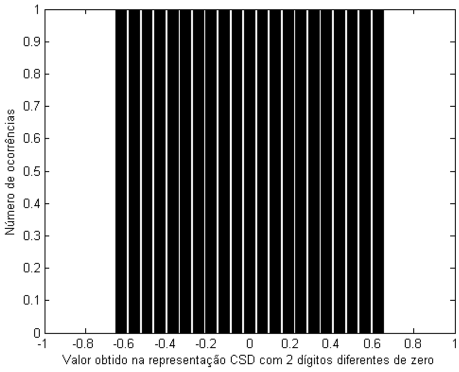
(c) 8 dígitos

Figura 10 Densidade de distribuição dos números MSD quando fixamos $nD = 2$

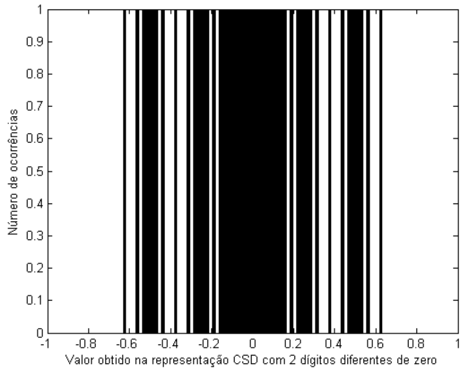
vez que uma representação CSD de 4 dígitos têm no máximo 2 dígitos diferentes devido à restrição vista na equação (26). Analogamente, se fizermos $nD = 3$ para números CSDs de 6 dígitos a distribuição com e sem limitação deverão ser idênticas e o mesmo vale para distribuições CSDs de 8 dígitos e $nD = 4$. Ou seja, dado um número CSD com Q dígitos

$$nD = \frac{V}{2}, \qquad \begin{cases} V = Q, & \forall Q = 2n \\ V = Q+1, & \forall Q = 2n+1 \end{cases} \qquad \forall n \in \{1, 2, ..., \infty^+\}.$$

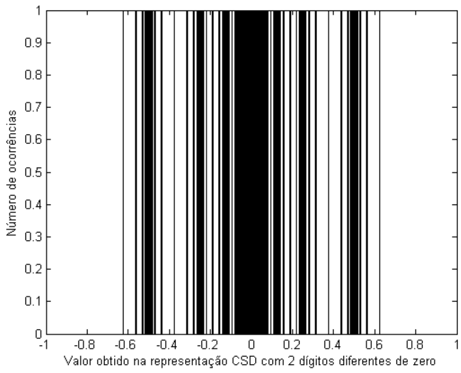
Neste trabalho escolhemos trabalhar com todas as combinações de SPT possíveis, ou seja, estão inclusas as representações não mínimas pois trabalhamos com algoritmos genéticos, e o fator **mutação**, explicitado adiante no Capítulo 3, devido ao seu comportamento aleatório gera números sem garantir o mínimo de somas. Porém, somente durante a conversão dos coeficientes de SPTs para números reais, no momento de da geração do filtro e sua resposta em frequência, faz-se o uso da representação MSD, assim garantimos o mínimo número de somas sem prejudicar as operações do algoritmo genético, como será



(a) 4 dígitos



(b) 6 dígitos



(c) 8 dígitos

Figura 11 Densidade de distribuição dos números CSD quando fixamos $nD = 2$

visto no Capítulo 4. Existem trabalhos [8] que apresentam alternativas para a utilização dos operadores dos algoritmos genéticos com representação SPT, mais especificamente com a representação CSD.

2.2 Quantização

Uma vez determinada a quantidade de dígitos, sejam eles binários ou ternários, para a representação de um número devemos nos atentar para o surgimento de erros de quantização. Se tivermos Q dígitos significativos, excluindo os dígitos que definem o sinal, temos uma precisão máxima, tanto em representações binárias ou em SPT, de 2^{-Q}. Sendo assim qualquer número que tiver valores inferiores a essa precisão será truncado para o valor mais próximo na representação adotada. O fato de números estarem representados usando SPT não implica em maior precisão do que números usando representação binária.

Porém por números SPT não precisarem de bits de codificação de sinal normalmente temos maior precisão neste formato quando comparados aos números binários possuindo a mesma quantidade de dígitos. Ou seja, se um número codificado em binário com sinal tem Q bits no total, serão necessários $Q-1$ elementos SPT para representá-lo. Portanto, quando o número de dígitos Q é fixado a precisão em binário é 2^{-Q}, enquanto em SPT essa precisão é de 2^{-Q+1}.

Durante o truncamento do número para o seu valor mais próximo na representação escolhida perdemos uma parte valiosa da informação do número, por isso, quanto maior a precisão de uma representação maior a fidelidade do número. Infelizmente, quanto maior a precisão, maior é a quantidade de elementos necessários para a representação isso implica em mais informações a serem analizadas e processadas e logo maior é o gasto de energia durante o processamento. Por isso, ao implementarmos um sistema devemos levar em conta tanto a precisão quanto a energia utilizada no processamento para uma dada aplicação e tentar estabelecer uma relação de compromisso.

2.2.1 Quantização dos Coeficientes de Filtros

Os coeficientes de um filtro são representados por números e, quando os quantizamos, introduz-se os erros anteriormente comentados. Quando se trata de coeficientes de um filtro, variações em seus coeficientes geram variações na posição dos polos e zeros que acarretam variações na resposta em frequência dos mesmos. Em certos casos, normalmente quando a precisão é baixa, a resposta do filtro obtido pode nem sequer atender às especificações do projeto. Os casos mais sérios ocorrem em filtros IIR que possuem pólos situados próximos à borda do circulo unitário, pois a quantização pode alterar a posição dos pólos, colocando-os na borda ou além da borda do circulo unitário afetando a sua estabilidade e transformando um filtro estável em um filtro instável de acordo com o critério de estabilidade descrito na Seção 1.2.

Outro problema da quantização ocorre quando a aproximação dos coeficientes é aplicada a um filtro com função de transferência com degraus, como é o caso das especificações de um filtro que normalmente retratam a resposta de um filtro ideal, produzindo uma resposta com *ripple* de alta magnitude nas proximidades das descontinuidades. Esse *ripple* é comumente chamado de oscilações de Gibbs e ocorre devido à lenta convergência das séries de Fourier em funções que apresentam descontinuidades [32, 87]. Este efeito possui a incomoda propriedade de não diminuir mesmo que a ordem do filtro seja drasticamente aumentada [44, 66].

Em um filtro FIR, a quantização dos coeficientes causa uma mudança de sua resposta em frequência uma vez os zeros da função de transferência sofrem deslocamentos devido ao arredondamento para o valor quantizado mais próximo. Essa mudança depende do nível de quantização escolhido, quanto maior a quantidade de dígitos usados na representação, menor é a variação da resposta em frequência. A simples quantização dos coeficientes de um filtro pode fazer com que ele passe a não estar de acordo com as especificações para o qual foi projetado e por isso é necessário cuidado e algumas alterações para que ele continue dentro das especificações [24]. Normalmente, uma vez que um filtro foi quantizado, seus coeficientes devem ser alterados para compensar erros de quantização, muitas vezes é até mesmo necessário o aumento da ordem do filtro para que ele continue atendendo suas especificações [6].

A distribuição dos valores para as três representações com limitação de dígitos diferentes de zero são mais densas para números menores, como visto na Subseção 2.1.4, assim os erros de quantização, nestes casos, afetam mais os coeficientes de maior módulo [80]. O único modo de melhorarmos essa precisão é o aumento do número de dígitos diferentes de zero nD na representação. Porém, mesmo com esse aumento existe na representação CSD a limitação imposta pela equação (26), que é a não existência de números diferentes de zero adjacentes o que impede a representação de alguns números e por isso interfere na distribuição dos valores representáveis. Se tomarmos um coeficiente de um filtro codificado com 6 dígitos, e dois dos quais diferentes de zero o maior valor possível de ser obtido é 0,625 na configuração $c_1 = 101000$. Se desejarmos escrever números menores do que o máximo com a restrição dos dois dígitos diferentes de zero teremos algumas limitações na representação dos números. Um exemplo é a representação do número 0,5625 que não é possível, o número mais próximo representável é o 0,625 o que acarreta um erro de quantização de 0,0625. Isto ocorre para a representação de diversos valores no intervalo $I \in [-1, 1]$, principalmente para os números mais próximos ao máximo.

Um método para limitar o erro de quantização na representação CSD é o escalonamento dos valores do coeficiente para a unidade. A multiplicação de todos os coeficientes do filtro CSD por uma mesma constante não afeta o formato da resposta em frequência deste, este escalonamento adiciona apenas um fator de ganho na resposta em frequência. Este escalonamento afeta profundamente o processo de otimização dos coeficientes quando a representação CSD é utilizada, uma vez que o escalonamento reduz a magnitude do erro de quantização que reflete diretamente na resposta em frequência do filtro [80].

Uma vantagem da representação MSD sobre a CSD é a distribuição uniforme dos valores uma vez que não há limitação dos valores representáveis, além da precisão 2^{-Q}. Se limitarmos o número de dígitos diferentes de zero em uma representação MSD, teremos uma distribuição mais uniforme dos valores do que na representação CSD com a mesma restrição de dígitos diferentes de zero. E, principalmente, a vantagem de termos uma maior possibilidade de representação dos números com um número limitado de dígitos se comparamos uma representação CSD de mesma quantidade de dígitos. O escalonamento para números MSD assim como para números CSD produz uma melhoria ao reduzir o erro de quantização, porém pelo fato da representação MSD ter uma distribuição dos números representáveis mais uniforme do que a CSD, este ganho com escalonamento é menor. Estas vantagens dos números MSD também estão presentes nas representações não mínimas utilizadas neste trabalho, porém, como o nome indica, os números MSD garantem o mínimo de operações aritméticas.

2.2.2 Ruído de Quantização

Toda vez que um número é arrendondado para seu valor mais próximo na representação escolhida surgem erros devido à discrepância entre o valor ideal e o valor representável. Este erro é chamado de ruído de quantização. Para avaliá-lo utilizaremos um modelo estatístico comumente usado [2]. O passo de quantização q é definido para números binários como

$$q_s = 2^{-(b-1)}, \tag{27}$$

onde b é a quantidade de bits usada para representar um coeficiente do filtro. Já o erro de quantização gerado por um multiplicador é considerado um ruído de distribuição uniforme e de variância $\sigma = \frac{q_s^2}{12}$ [2]. Estendendo este conceito a um filtro arbitrário com M multiplicadores obtemos um ruído de quantização total na saída do filtro de média zero e variância [14]

$$\sigma_a^2 = \frac{Mq_s^2}{12}. \tag{28}$$

Sendo assim, vemos da equação (28) que ao reduzirmos o número de multiplicadores reduzimos também o ruído de quantização [2]. Daí a busca por uma estrutura e uma representação mais eficientes. Como estamos trabalhando com filtros de fase linear na forma direta introduziremos o seu erro de quantização que é dado por [14]

$$\sigma_l^2 = \frac{M+1}{2}\frac{{q_s}^2}{12} \tag{29}$$

2.3 Representação Numérica dos Coeficientes dos Filtros

Os coeficientes dos filtros utilizados neste trabalho foram representados em binários e em SPT com Q dígitos, cada um correspondendo a uma potência de dois negativa, conforme a Figura 12 e a Figura 13. Essas figuras mostram como é feita a composição do valor de um coeficiente, quando um determinado dígito é 0, sua potência de dois equivalente é desconsiderada, quando um dígito é 1 ocorre a soma do valor da potência equivalente e quando o dígito é -1 ocorre a subtração da potência. Os dois primeiros casos são válidos tanto para números SPT quanto para binários, já o último caso só é valido para números com representação SPT.

Sendo assim os valores que esses coeficientes c_x podem assumir são representados como

$$c_x = \sum_{i=0}^{L} j_i 2^{-i}, \tag{30}$$

onde $j_i \in \{-1, 0, 1\}$ para números SPT e $j_i \in \{0, 1\}$ para binários. Desse modo o máximo

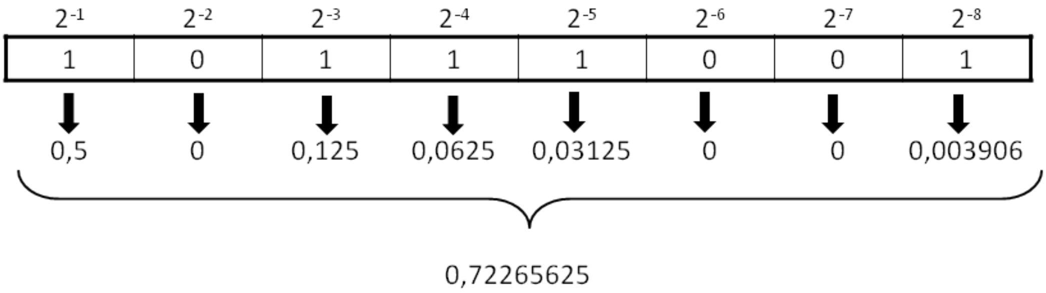


Figura 12 Representação binária

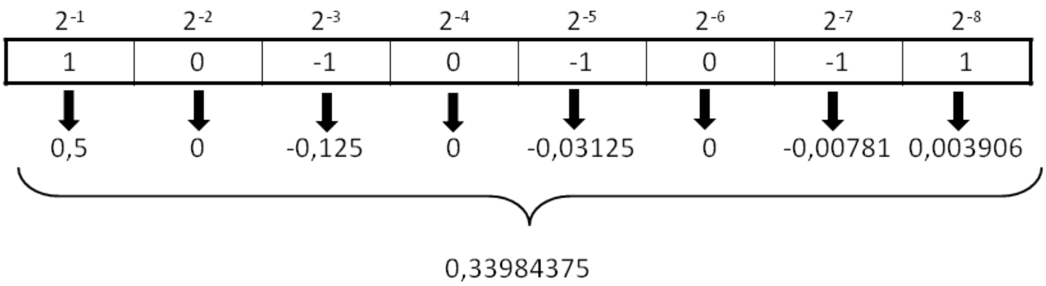


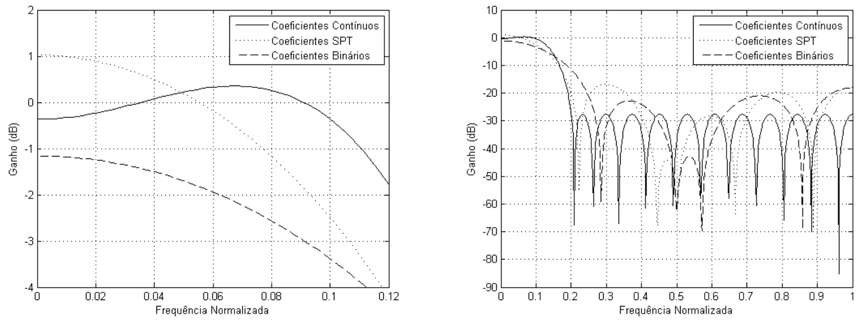
Figura 13 Representação SPT

valor do coeficiente que pode ser representado neste modelo ocorre quando $j_i = 1\ \forall i$, em ambos os casos e é dado por

$$c_{max} = \lim_{L\to\infty} \sum_{i=0}^{L} 2^{-i}, \tag{31}$$

o que nos leva a $c_{max} = 1$. Analogamente para números SPT, $c_{min} = -1$ ocorre para $j_i = -1\ \forall i$. Para números binários a representação de c_{min} depende da representação escolhida com foi visto na Seção 2.1, para a representação sinal e magnitude deve-se adicionar um bit $s_x = 1$ à frente da codificação de c_{max}.

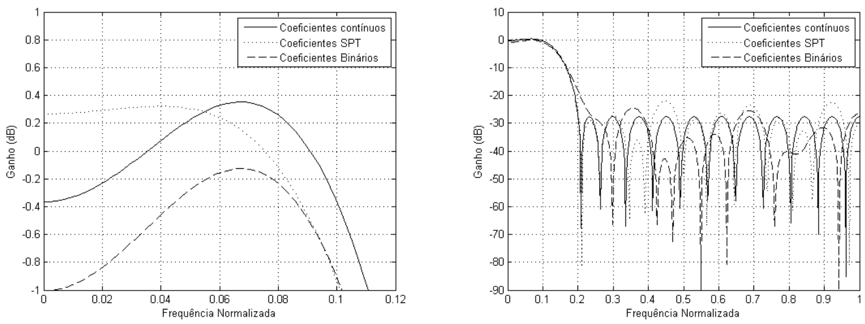
Para ter-se uma ideia do efeito da quantização sobre um filtro de coeficientes contínuos fez-se o arredondamento dos coeficientes deste utilizando-se representações binária e SPT e analisou-se suas respostas em frequência. Os coeficientes do filtro FIRPM foram arredondados com 4, 6 e 8 dígitos para as duas representações. A contagem do número de dígitos utilizados na representação binária dos coeficientes levou em conta o bit de sinalização, o que impactou na precisão desta. A Figura 14, juntamente com a Figura 15 e a Figura 16 apresentam as respostas em frequência do filtro FIRPM de coeficientes contínuos, do mesmo filtro com coeficientes arredondados com a representação SPT e binária, para cada um dos níveis de quantização supracitados. Como era de se esperar quanto maior o número de dígitos escolhidos maior a proximidade entre a curva do filtro contínuo e a curva do filtro arredondado. Para 4 dígitos de arredondamento nota-se que a banda passante dos filtros SPT e binário passam a apresentar variações superiores à 1dB. Para todos os níveis de quantização, os filtros SPT apresentaram superioridade em relação aos filtros binários, no que diz respeito à resposta em frequência de um filtro, o que justifica a sua utilização.



(a) Resposta em frequência da banda passante

(b) Resposta em frequência

Figura 14 Comparação entre o filtro FIRPM e suas aproximações por filtros binários e SPT de 4 dígitos

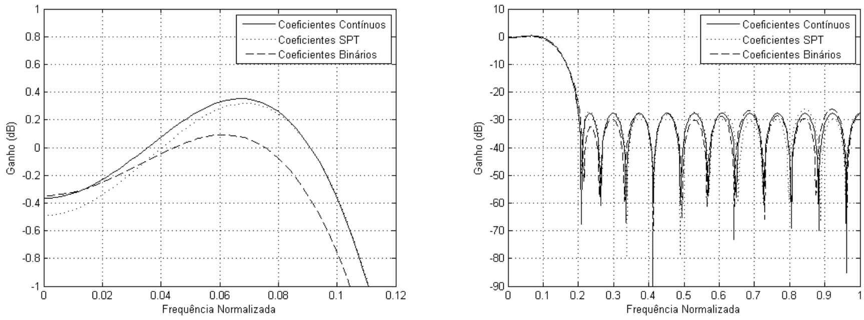


(a) Resposta em frequência da banda passante

(b) Resposta em frequência

Figura 15 Comparação entre o filtro FIRPM e suas aproximações por filtros binários e SPT de 6 dígitos

O passo seguinte, após a definição da representação dos coeficientes dos filtros, é a caracterização de um algoritmo genético e seus parâmetros, o que será feito no Capítulo 3.



(a) Resposta em frequência da banda passante

(b) Resposta em frequência

Figura 16 Comparação entre o filtro FIRPM e suas aproximações por filtros binários e SPT de 8 dígitos

3 ALGORITMOS GENÉTICOS APLICADOS AO PROJETO DE FILTROS

Mesmo com a melhoria constante da capacidade de processamento dos computadores [46], ainda existem diversos problemas quando o computador lida com grande número de variáveis, sejam problemas de quantização, memória ou processamento. Por sua vez, essas variáveis podem ter grandes faixas de variação, onde a aplicação de um algoritmo de busca exaustiva não é viável por requerer demasiado tempo para cumprir a tarefa de encontrar o resultado procurado. Por outro lado, alguns mecanismos rápidos de busca correm o risco de ficarem presos a um máximo/mínimo local e assim, as vezes longe de uma boa resposta [61]. Para suprir esse déficit uma série de algoritmos ditos evolucionários surgiram, dentre eles, temos o algoritmo genético ou GA (*Genetic Algorithm*) [18] que demonstrou ser uma ferramenta robusta e poderosa para projetos de filtros com coeficientes discretos [99]. O GA é um algoritmo de otimização, proposto por J. H. Holland [10], que visa reduzir o tempo de busca e ao mesmo tempo encontrar uma boa solução para um dado problema. A solução encontrada normalmente não é a solução ótima para o problema, porém se bem ajustado pode gerar respostas sub-ótimas [28]. A escolha da utilização de GA para a otimização de filtros tem como motivação encontrar em um espaço de busca S um filtro S_i, onde $i \in \{1, 2, ..., s\}$, onde s é número de filtros contidos em S, que tenha a resposta em frequência mais próxima possível de uma dada especificação [47].

Os algoritmos genéticos têm esse nome por terem sido baseados na teoria de Darwin de evolução das espécies , baseada na sobrevivência dos indivíduos mais aptos [19]. Durante o processo evolutivo diversos indivíduos são criados, os mais aptos, por terem maiores chances de sobrevivência, terão maior probabilidade de passar adiante suas características. Sendo assim, o GA tem como objetivo realizar a busca da solução de um problema de otimização baseado na aptidão dos indivíduos. A estrutura básica de um GA é apresentada na Figura 17 e começa com a representação numérica dos indivíduos que definirá o tipo de operações possíveis de serem aplicadas a estes. Em seguida, é inicializada uma população com diversos membros que terão sua aptidão avaliada em função de suas respostas ao problema. Desta população serão selecionados alguns indivíduos que participarão do processo de alteração genética, caracterizado pelo cruzamento e pela mutação, para a produção de indivíduos filhos com características herdadas dos indivíduos pais. Os indivíduos não selecionados neste estágio não produzem descendentes e são descartados. O resultado deste processo é a criação de uma nova população que por sua vez será avaliada e que terá seus indivíduos selecionados, alterados e descartados. Este processo cíclico continua até que um critério de parada pré-determinado seja atingido, produzindo então um conjunto de possíveis respostas ao problema. Esta não é a única possível estrutura de aplicação do GA como veremos ao longo deste capítulo, mas permite termos uma visão geral do funcionamento do algoritmo.

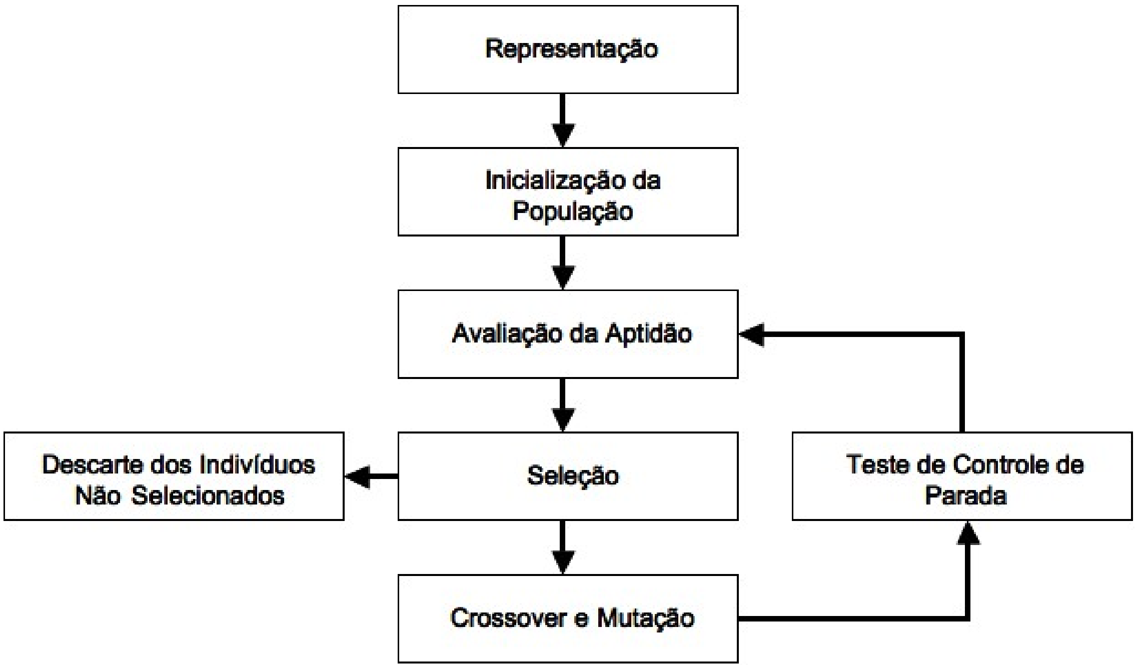


Figura 17 Estrutura de um GA

Ao submetermos filtros à otimização de seus coeficientes através de algoritmos genéticos alguns fatores devem ser levados em consideração. Neste trabalho, cada indivíduo de uma população é um filtro. Estes filtros são representados por cromossomos, que são compostos por uma quantidade c de genes, sendo estes genes os coeficientes do filtro. Por sua vez os genes podem ser subdivididos em d unidades menores que chamaremos de dígitos, pois são de fato dígitos binários ou SPT, dependendo da utilização, como foi visto no capítulo 2. Como os filtros possuem uma ordenação definida de seus coeficientes, nos cromossomos a ordenação dos genes é igual à ordenação dos coeficientes do filtro. A Figura 18 mostra a representação de um filtro de quatro coeficientes, cada um codificado com 4 dígitos.

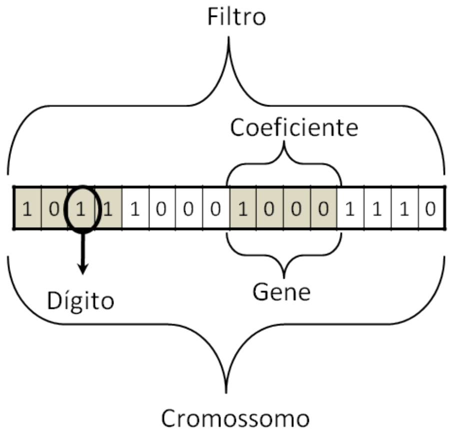


Figura 18 Representação de um indivíduo, um filtro (cromossomo) com quatro coeficientes (genes) cada um codificado com 4 dígitos

Neste capítulo abordaremos os diversos parâmetros dos GAs, seus efeitos e os cuidados que devem ser tomados quando o objetivo é a obtenção de filtros fiéis à uma especificação.

3.1 Função de Avaliação

Durante a evolução diversos indivíduos novos são criados, alguns com características favoráveis e adaptados às condições ambientais enquanto outros surgem com características desfavoráveis. Na natureza o ambiente é responsável pela seleção desses indivíduos, quanto maior a aptidão de um deles maior será a sua chance de sobrevivência.

No GA o ambiente é representado por uma modelagem do problema, que toma a forma de uma função de avaliação que tem como objetivo atribuir uma nota a um determinado indivíduo. Esta avaliação é feita com base na resposta do indivíduo avaliado em relação ao problema para o qual o GA esta sendo aplicado. É essa resposta, que normalmente toma forma de um único número, que é chamada de aptidão [23]. A definição da aptidão dos indivíduos é o método que o algoritmo tem para identificar e, logo, diferenciar os indivíduos, uma vez que antes da avaliação todos os membros da população são potencialmente iguais em termos de aptidão. A função de avaliação é crítica na elaboração de uma otimização usando GA, pois é ela que define a progressão do algoritmo. Uma função de avaliação pode ser estática, quando o espaço de busca não varia ou varia muito pouco ao longo do tempo, ou dinâmica, quando o espaço é mutável [88]. Outra classificação importante das funções de avaliação é quanto à sua quantidade de objetivos. As chamadas funções multi-objetivo, possuem mais de um objetivo e necessitam de uma atenção particular no momento da união das suas diversas aptidões.

No caso de filtros, a modelagem é feita com base nas especificações e por isso podemos considerar que a sua função de avaliação é estática. Esta etapa é, juntamente com a escolha da representação numérica, a mais complexa [18], pois precisamos garantir uma boa modelagem para que o algoritmo possa progredir. Modelos ruins podem levar muito tempo para produzir uma boa resposta, isto quando chegam a convergir [28]. A função de avaliação para filtros reflete a capacidade de atender às especificações de projeto. Alguns parâmetros de projeto são difíceis de serem modelados, estes normalmente consistem em restrições de projeto que invalidam um indivíduo que não as respeite [88]. Para que todas as conformidades do projeto sejam respeitadas são utilizadas penalidades [1] como veremos à seguir.

3.1.1 Penalidades

A maioria das situações de otimização reais possuem limitações advindas da natureza do problema. Porém, no algoritmo genético, existe o potencial de que em uma população inicial sorteada haja membros que possuem uma característica indesejada, ou que um indivíduo apto em uma geração, ao sofrer alterações genéticas passe a apresentar características indesejadas. Essas características, por normalmente conflitarem com as restrições do projeto, devem ser eliminadas. O grande problema dos indivíduos que apresentam conflitos com as especificações é que nem sempre a função de avaliação por si só é capaz de classificá-los como indivíduos inaptos. Muitas vezes isso ocorre quando a função de avaliação não abrange as restrições. Um exemplo disso é uma função que avalia a resposta em frequência de um filtro IIR, esta não tem como avaliar a estabilidade do filtro, que depende das posições dos pólos e por isso pode julgar um filtro instável e portanto indesejável como apto. Isto faria com que o indivíduo inapto passasse suas características para as gerações seguintes do GA. Para contornar problemas desse tipo insere-se funções de penalidade, que são funções auxiliares à função de avaliação e que podem seguir as seguintes estratégias para lidar com indivíduos inválidos [1]:

- Descartar os indivíduos inaptos, chamada de penalidade de morte e gerar um novo até que este seja apto.
- Reduzir a aptidão dos membros da população que infrinjam restrições.
- Alterar os operadores genéticos, mutação e cruzamento, para que estes não alterem suficientemente um determinado indivíduo a ponto deste infringir restrições de projeto.
- Consertar os indivíduos inaptos para que estes tornem-se aptos.

Nota-se que, com a inserção de funções de penalidade aumenta-se a carga computacional de otimização, na medida que aumenta a aparição de membros conflitantes com as restrições na população, além de nem sempre ser simples criar uma função de penalidade adequada ao problema [88]. Por isso, deve-se evitar a utilização destas funções e para isso uma boa modelagem do problema é necessária assim como uma boa escolha da representação dos números. Diferentes modelagens do mesmo problema podem resultar em, por exemplo, diferentes funções de avaliação, como ocorre neste trabalho. As duas funções de avaliação que serão apresentadas no Capítulo 4, apresentam resultados diferentes, sendo a que não usa funções penalidade mais eficiente do que a que usa. A representação dos números pode influir na velocidade da obtenção dos resultados e na qualidade dos resultados obtidos como visto no Capítulo 2.

3.2 População

Quando organismos são submetidos à seleção natural, isso não ocorre com um indivíduo apenas e sim com um grupo. É esta população que estabelece a diversidade genética, dando assim a base para as possibilidades de mudanças e portanto define os organismos gerados nas populações futuras. Assim como na natureza as mutações em um GA não afetam uniformemente toda a população e não necessariamente afetam todos os indivíduos da população. Sendo assim ao fim da otimização do GA é possível encontrarmos um indivíduo da população inicial inalterado, por mais improvável que isto seja. Ao contrário de outros métodos de otimização a saída do GA não consiste em uma única solução para o problema e sim em uma população de indivíduos-solução cada qual com suas características. Desta população se pode retirar o melhor indivíduo como resposta final ao problema.

3.2.1 População Inicial

A população inicial consiste em uma amostra do espaço de busca do algoritmo. Para iniciar o GA uma população inicial deve ser definida, pois é ela que define os sub-espaços de busca que o algoritmo irá percorrer. Normalmente, o tamanho da população inicial determina o número de indivíduos em cada população nas gerações subsequentes. Existem aplicações do GA usando populações de tamanhos diferentes ao longo das gerações [7], usa-se o aumento da população como artifício para introduzir diversidade ao longo do processo evolutivo. O processo de definição da população inicial pode ser aleatório ou baseado em fatores predefinidos [34, 54]. A avaliação consiste em medir a aptidão de cada indivíduo da população. Neste trabalho, o espaço de busca é um hiper-espaço que contém os coeficientes do filtro, que tem dimensão igual ao número de coeficientes c do filtro. Estes coeficientes por sua vez são também hiper-espaços de dimensão Q igual à quantidade de dígitos que representam cada coeficiente. Ou seja, o espaço de busca total possui dimensão total $dim = cQ$.

Quando trabalhamos com números binários o espaço de busca é dado por $Eb_{BIN} = 2^{dim}$ [18]. Analogamente quando trabalhamos com a representação SPT que é ternária temos um espaço de busca de tamanho $Eb_{SPT} = 3^{dim}$. Porém como foi visto no Capítulo 2 alguns números possuem mais de uma representação possível, o que faz o espaço de busca efetivo em SPT ser de tamanho $Ebf_{SPT} = Nf_{val}$, onde Nf_{val} é derivado de N_{val}, visto na Subseção 2.1.3, e leva em conta o número de coeficientes do filtro para o cálculo das representações. O espaço de busca efetivo consiste no conjunto de números que ao serem escolhidos produzirão um resultado único. Quando utilizamos a representação SPT o único jeito de obtermos um espaço de busca total igual ao espaço de busca efetivo é utilizando a representação CSD, vista na Subseção 2.1.4. Portanto, como não restringimos a representação dos números, o espaço de busca do algoritmo proposto neste trabalho é Eb_{SPT} que é significantemente maior do que o espaço para o caso binário.

A população inicial tem como objetivo definir a diversidade genética do algoritmo, pois uma vez iniciados, os GAs não inserem normalmente novos indivíduos na busca. Deve-se evitar populações iniciais com um indivíduo, ou grupo de indivíduos, com aptidão altíssima, pois isto limitará a progressão do algoritmo, uma vez que este será induzido a restringir sua busca aos arredores destes super-indivíduos fazendo com que o GA perca diversidade [49, 55]. Por outro, lado populações com muitos indivíduos inaptos, normalmente obtidas em inicializações aleatórias [11], tendem a demorar muito a convergir e por isso deve-se escolher uma função de seleção com alta intensidade de seleção, como será visto na seção 3.3.1. Uma população inicial com indivíduos com aptidão razoável é ideal para uma boa evolução do algoritmo e uma redução no tempo de processamento. Porém, obter populações iniciais com membros de aptidão razoável implica em executar algum tipo de busca anterior ou restringir o espaço de busca. Seja esta restrição imposta pela simplicidade do problema, onde mesmo com inicialização aleatória, pelo fato do espaço de busca ser por natureza pequeno, boas respostas são encontradas na população inicial ou pelo conhecimento prévio da provável posição de uma boa solução o que permite uma delimitação do espaço a ser vasculhado [99]. Mas como o objetivo deste trabalho é uma busca diversificada a escolha foi a inicialização aleatória, pois a principio, para uma quantidade razoável de indivíduos esta permite uma grande diversidade genética.

Para filtros, a população inicial consiste na inicialização de sequências numéricas representado seus coeficientes. Uma população inicial aleatória em filtros gera respostas bem distantes da desejada o que causa demora na convergência do algoritmo, e conforme a ordem do filtro cresce aumenta drasticamente o tempo de convergência. Para reduzir o espaço de busca existem muitas técnicas para gerar uma entrada adequada. Uma delas por exemplo é a utilização dos coeficientes do filtro com precisão infinita arredondados, outra que é superior a esta ultima é apresentada em [99] e também baseada no filtro de precisão infinita, onde os coeficientes são otimizados sucessivamente de acordo com a sensibilidade de cada coeficiente. Estas técnicas por envolverem um filtro de resposta muito boa para as especificações dadas levam normalmente a bons resultados, o que era de se esperar uma vez que o espaço de busca foi bem reduzido e direcionado. Isso é o equivalente na natureza a levar um pinguim para o polo norte, ao menos no que diz respeito à resistência a baixas temperaturas, qualquer indivíduo normal dessa espécie é capaz de sobreviver nas novas condições e pode evoluir lentamente pois não há risco de vida. Porém, na inicialização direcionada é muito difícil de encontrar a resposta desejada fora do espaço de busca determinado pela população inicial, então, se não se sabe ao certo onde estão localizadas as melhores respostas ou se as respostas estão distantes umas das outras no espaço de busca, uma inicialização aleatória é recomendável.

Uma alternativa para implementação de filtros com coeficientes iniciais aleatórios é o relaxamento da avaliação inicial. Normalmente, a função de avaliação nos algoritmos genéticos para otimização do filtro consiste no cumprimento por parte deste de certas especificações. Porém, exigir de um filtro aleatório uma resposta em frequência que muitas vezes tem limites rígidos, como banda de transição pequena, grande atenuação nas bandas de rejeição e etc, torna-se inviável, resultando em uma população inicial inapta. Porém, se margens de erro forem permitidas na avaliação inicial, pode-se obter uma população com aptidão aceitável para ser empregada como entrada do GA. Essas adaptações são particularmente úteis nos casos de funções de avaliação que possuem muitas funções de penalidade, uma vez que obter de um sorteio um filtro que cumpra todas restrições é um feito raro. Nos casos em que as restrições são poucas, ou que estejam bem amarradas dentro da função de avaliação inicial, o relaxamento inicial das margens de erro é desnecessário.

3.2.2 População Final

Ao fim de cada ciclo do GA, é gerada uma nova população a partir dos indivíduos selecionados e alterados da população anterior. Esta nova população passa a ser usada no próximo ciclo do GA. Porém, existem diversas maneiras de se substituir os indivíduos de uma geração para a outra. A mais óbvia consiste na substituição de todos os membros da população do ciclo anterior pelos novos indivíduos gerados. Neste modelo pode-se ter a exclusão do melhor membro de uma população das populações seguintes e por isso é normalmente utilizado o **elitismo** [88]. O elitismo garante a existência do melhor elemento de uma população nas seguintes e provou-se muito efetivo na rápida convergência dos GAs [88], porém pode levar à dominação da população por um super-indivíduo prematuramente. Outro método de substituição é a troca de apenas alguns membros da população antiga, normalmente os piores, por um número equivalente dos novos indivíduos [88]. Conforme o processo evolutivo avança os membros da população passam a partilhar características, consideradas benéficas pelo algoritmo, e convergem para uma solução aceitável.

A população final é a ultima população obtida antes que um dos critérios de parada do algoritmo seja atingido. Se o GA for bem projetado e tiver progredido ao longo das gerações, esta população será composta por diversos indivíduos onde cada um deles será uma possível resposta ao problema. Não necessariamente todos os indivíduos desta última geração são boas soluções para o problema, mas normalmente se o algoritmo tiver tempo suficiente para convergir a média da aptidão dos indivíduos será alta. Dentre os indivíduos desta população o que apresentar melhor aptidão é normalmente escolhido como solução para o problema. Muitas vezes o acompanhamento da aptidão média das populações ao final de cada ciclo permite uma boa avaliação do comportamento do algoritmo e pode ser base de comparação para a evolução do melhor indivíduo [54].

A Figura 19 mostra a evolução de um algoritmo com população inicial de $N = 100$ indivíduos e com parada na geração $g = 1000$ gerações, o que totaliza $Tp = 100000$ indivíduos avaliados, para populações iniciais com indivíduos muito aptos, pouco aptos e razoavelmente aptos. A população inicial muito apta e a razoavelmente apta foram obtidas a partir do filtro FIRPM de precisão infinita descrito na Subseção 1.3.2. Para a população muito apta foi somado a cada coeficiente deste filtro um fator aleatório $r_1 \in [-0,01; 0,01]$ enquanto para a população razoável este fator foi $r_2 \in [-0,033; 0,033]$. A população inicial inapta foi sorteada aleatoriamente. Nota-se que a progressão da curva de indivíduos muito aptos já parte de um valor muito alto e que em poucas gerações entra em estagnação tendo atingido uma aptidão final muito elevada. A curva de aptidão inicial razoável parte de um patamar médio e atinge uma boa aptidão final mesmo esta estando abaixo da aptidão final da curva de indivíduos de aptidão muito alta. Já para a população inicial pouco apta sua inicialização se dá com indivíduos de aptidão baixíssimas, beirando o zero, e cresce até um valor de aptidão razoável. Mesmo esta última progressão sendo a pior das três no que diz respeito à aptidão final atingida, ela é a que apresenta maior variação entre aptidão final e inicial, seguida da curva de aptidão razoável e por fim da curva obtida para a população inicial muito apta. Outro comportamento interessante que pode ser percebido é que enquanto as curvas para a aptidão inicial alta e razoável crescem rapidamente até atingir um máximo em poucas gerações, a curva de aptidão inicial baixa progride mais lentamente e apresenta melhorias na sua aptidão ao longo das 1000 gerações.

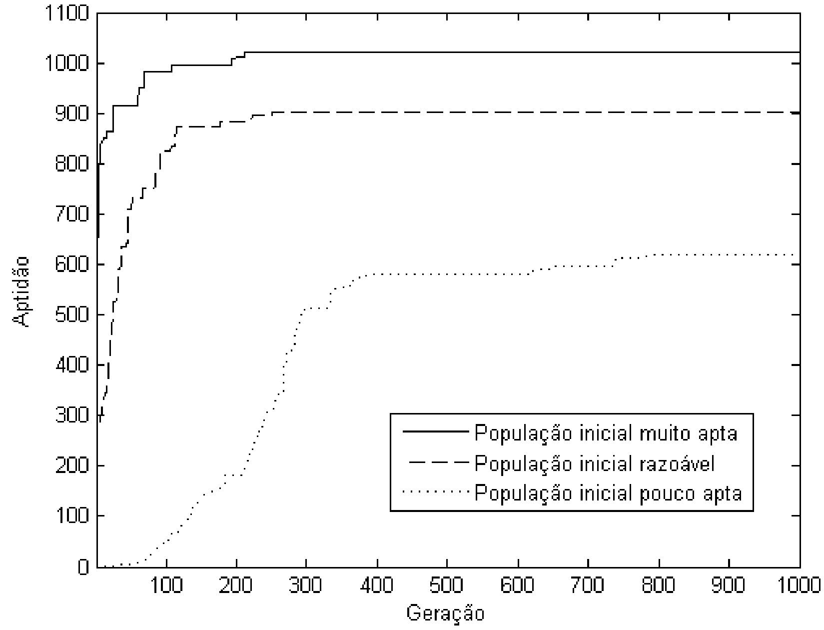


Figura 19 Evolução do melhor indivíduo para diferentes populações iniciais com $N = 100$ e $g = 1000$

Outra questão importante a ser considerada é o tamanho das populações, normalmente definido pelo tamanho da população inicial, pois ele influi diretamente no tempo de execução do algoritmo. Sendo assim, deve-se balancear o tamanho da população inicial e número de gerações utilizadas para minimizar o tempo de execução do GA e ao mesmo tempo maximizar o resultado. Lembrando que em um GA comum o número total de indivíduos é dado pelo produto $Tp = Ng$ onde N é a população por geração e g o número de gerações. Se para um mesmo problema os parâmetros de um GA forem configurados das seguintes maneiras $Tp_1 = N_1 g_1$ e $Tp_2 = N_2 g_2$, e fixarmos o número total de indivíduos de modo que $Tp_1 = Tp_2$, tem-se que se fizermos $N_1 > N_2$ obrigatoriamente teremos $g_1 < g_2$. Ou seja, enquanto na primeira configuração a população é maior, a variação dos indivíduos é menor pois decorreu-se um menor número de gerações.

A Figura 20 mostra uma outra evolução para populações de indivíduos muito aptos, razoáveis e pouco aptos, definidas conforme usado para gerar os resultados apresentados na Figura 19, porém desta vez com população inicial de $N = 200$ indivíduos e $g = 500$ gerações, totalizando também $Tp = 100000$ indivíduos. A Figura 19 e a Figura 20 ilustram o comportamento de uma execução do GA para cada configuração e não uma média de execuções e por isso não se pode extrair informações conclusivas destas. Porém, elas indicam como comportam-se GAs com com populações iniciais de diferentes aptidões no que diz respeito ao tempo de convergência destes quando consideramos um número fixo de indivíduos Tp e variações nos termos do produto Ng. GAs com alta aptidão inicial resultam, como era de se esperar, em respostas melhores e de rápida convergência, para aptidões médias a convergência é mais lenta, mas mesmo assim geram indivíduos com boa aptidão. Para populações pouco aptas a convergência é bem lenta e o resultado, mesmo por melhor que seja, não rivaliza com os outros dois casos. Os resultados de GAs com populações pouco aptas pode ser melhorado com variações nos parâmetros do GA, isso é abordado no Capítulo 4.

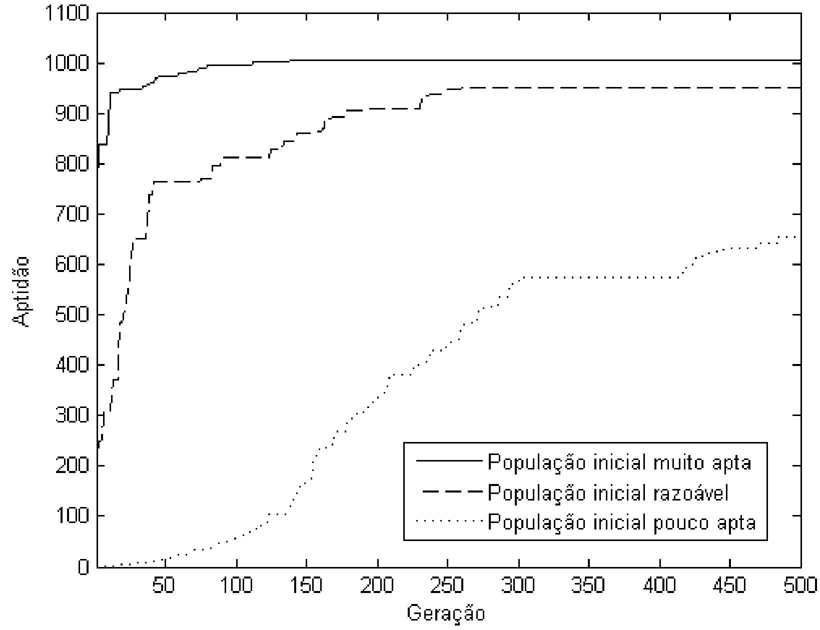


Figura 20 Evolução do melhor indivíduo para diferentes populações iniciais com $N = 200$ e $g = 500$

3.3 Seleção dos Indivíduos

Para efetuarmos as operações de cruzamento e mutação no GA, explicadas adiante nas seções 3.4 e 3.5, devemos a priori selecionar os indivíduos que passarão pelas modificações evolucionárias. Existem diversas formas de selecionar os indivíduos, todas são implementadas por um sorteio onde as chances de seleção de um indivíduo depende do método escolhido, na literatura os cinco principais são: seleção proporcional, por torneios, por truncamento, por normalização linear e por normalização geométrica [54]. Existem outras como a recombinação elitista [90,91] que não possui fase de seleção e recombinação separadas, mas que não serão abordadas aqui. A seleção proporcional ou roleta, que é usada neste trabalho, consiste na escolha dos indivíduos de modo que membros mais aptos de cada geração tenham maior chance de serem selecionados [53], isso é uma analogia com a natureza onde os seres mais aptos têm maior chance de se reproduzir e gerar descendentes. A função de seleção Ω é uma função que transforma uma distribuição de aptidão f em uma nova distribuição de aptidão f' [10], de modo que

$$f' = \Omega(f). \tag{32}$$

Uma forma de se caracterizar o método de seleção escolhido é a intensidade de seleção.

3.3.1 Intensidade de Seleção

Para definirmos intensidade de seleção devemos inicialmente definir a aptidão média de uma população de N indivíduos dada por

$$M = \sum_{i=1}^{N} \frac{f_i}{N}, \tag{33}$$

onde f_i é a aptidão do indivíduo i.

A definição de intensidade de seleção, ou pressão de seleção, é a variação na aptidão média da população induzida pela função de seleção [10]. A intensidade de seleção mede o quão mais favorecidos serão os melhores indivíduos de uma população, quanto maior a intensidade de seleção mais favorecidos são os melhores indivíduos. A convergência de um GA é diretamente dependente da intensidade de seleção, quanto mais alta esta for mais alta é a velocidade de convergência [10, 60]. Porém, uma intensidade de seleção demasiadamente alta podem levar o GA à uma convergência prematura [60]. A função de seleção ideal deve favorecer os melhores indivíduos e ao mesmo tempo preservar a diversidade da população, o que evita a convergência prematura. A intensidade de seleção mede as melhorias na população antes e depois da seleção dos indivíduos [10] e sua expressão é dada por

$$I = \frac{M' - M}{\sigma}, \tag{34}$$

onde M' é a aptidão média da população após a seleção obtida aplicando (32) em (33) ,M é a média da população antes da seleção e σ é o desvio padrão dos valores de aptidão da população antes da seleção.

Na seleção proporcional a probabilidade de seleção de um indivíduo i de aptidão f_i em meio à uma população de tamanho N é dada por

$$p_i = \frac{f_i}{NM}. \tag{35}$$

Nota-se que, conforme a população aumenta, a probabilidade de escolha do melhor individuo e do pior tendem a se aproximar, o que não é um fator favorável à convergência do GA [10]. De 35 e 33 obtemos

$$\sum_{i=1}^{N} p_i = 1. \tag{36}$$

Da equação 36 concluímos que quanto maior o número de indivíduos N da população, menor será a chance de cada indivíduo ser selecionado por mais apto que este possa ser, o que causa a aproximação da aptidão do melhor e do pior indivíduo. Para ilustrar esse fato imaginemos uma população de $N = 10$ indivíduos onde as aptidões dos indivíduos sejam $f_i \in \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10\}$ onde $f_{10} = 10$ é a aptidão do melhor indivíduo e $f_1 = 1$ a do pior. Neste caso a média é $M = 5,5$ e da equação 35 temos as seguintes probabilidades de seleção $p_{10} \cong 18\%$ e $p_1 \cong 1,8\%$, para o melhor e pior indivíduo respectivamente. Caso tenhamos uma população com $N = 100$ com distribuição de aptidão semelhante à primeira, ou seja $f_i \in \{1, 2, ..., 99, 100\}$, sua média passará a ser $M = 50,5$. Já a probabilidade de seleção do melhor indivíduo será $p_{100} \cong 2\%$ e a do pior $p_1 \cong 0,02\%$. Nota-se que no primeiro caso a diferença entre as probabilidades de seleção do melhor e pior indivíduo é $\Delta p \cong 16\%$ enquanto no segundo caso é de $\Delta p \cong 1,98\%$. Este efeito ocorre porque a seleção proporcional não é invariável à tradução [10]. Sendo assim, sua intensidade de seleção deve ser representada conforme a (37) [10].

$$I = \frac{\sigma}{M}, \tag{37}$$

onde σ é o desvio padrão da aptidão da população.

Nota-se que em alguns casos, devido à amostragem do espaço de busca, ocorre uma redução na aptidão média da população quando utiliza-se a seleção proporcional [10]. Além da baixa intensidade de seleção, a variância da seleção e a perda de diversidade são dois fatores difíceis de serem analisados na seleção proporcional, o que de acordo com [57] torna este tipo de seleção pouco adequado para a otimização. Para melhorar este quadro diversos métodos de escalonamento da aptidão dos indivíduos foram propostos [88] que consistem em alterações na representação numérica dos indivíduos para manter alta a intensidade de seleção. Nestes métodos as aptidões dos indivíduos mais aptas são aumentadas para que uma alta intensidade de seleção seja mantida. No Capítulo 4 veremos que, ao contrário do que é dito em [10], para a aplicação deste trabalho não há muita diferença entre os resultados quando se usa seleção proporcional e a seleção por classificação linear vista à seguir.

Uma alternativa para a seleção proporcional é a seleção por classificação linear, onde os indivíduos são classificados pela sua aptidão de modo que para as posições $i \in \{1, 2, ..., N\}$, o melhor assume a posição N, o pior a posição 1 e os outros são posicionados crescentemente nesse intervalo. Denominamos por η^+ e η^- as aptidões do melhor e do pior indivíduo respectivamente. Originalmente este método prevê que as condições $\eta^+ = 2-\eta^-$ e $\eta^- \geq 0$, mas estas podem ser alteradas [54]. Assim, a probabilidade de seleção de um indivíduo é designada linearmente conforme

$$p_i = \frac{1}{N}\left(\eta^- + (\eta^+ - \eta^-)\frac{i-1}{N-1}\right). \tag{38}$$

As probabilidades de seleção do melhor e do pior indivíduo são dadas por $\frac{\eta^-}{N}$ e $\frac{\eta^+}{N}$ respectivamente. Mesmo que dois indivíduos possuam a mesma aptidão eles serão classificados em diferentes posições. Neste modelo de seleção, a intensidade de seleção é dada por [10]

$$I(\eta^-) = (1-\eta^-)\frac{1}{\sqrt{\pi}}. \tag{39}$$

Outras formas de seleção mostradas em [10,53,54] não foram aprofundadas pois não foram testadas ao longo da execução deste trabalho.

3.4 Cruzamento

Na natureza o cruzamento consiste na mistura de parte do código genético de um indivíduo com uma parte equivalente do código genético de outro. Neste processo, as características dos pais são combinadas para gerar filhos, isto é novos indivíduos. A Figura 21 exemplifica o cruzamento de dois cromossomos. Neste cruzamento ocorre a recombinação genética dos cromossomos pais gerando dois cromossomos filhos. A taxa de cruzamento do GA p_c é o parâmetro que indica a chance de dois cromossomos selecio-

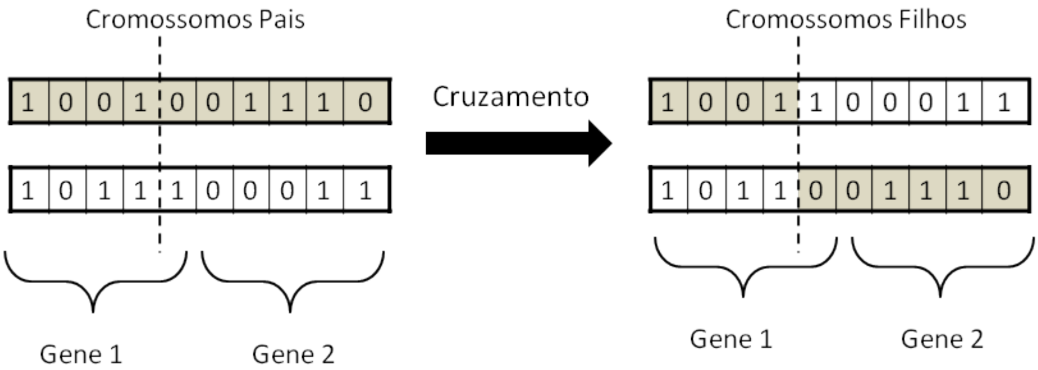


Figura 21 Operador cruzamento

nados participarem do processo de cruzamento, comumente chamado de *crossover*. Para muitos o cruzamento é o processo determinante, que distingue o GA de outros métodos de otimização existentes [88]. Existem diversos tipos de cruzamento, o mais simples consiste na quebra dos cromossomos dos pais em um mesmo ponto e na troca entre eles das partes equivalentes como mostrado na Figura 21. Além de cruzamento com um único ponto de corte pode-se ter cruzamentos com mais de um ponto de corte [88] entre outros [59]. No Capítulo 4 será apresentado o esquema de cruzamento usado neste trabalho, onde os cortes ocorrem logo após o fim de um grupo de genes que define um parâmetro, neste último caso o *crossover* passa a ser uma troca de parâmetros. Porém de [99] temos que o cruzamento quando efetuado desta forma para filtros produz filhos com aptidões muito baixas, pois os danos advindos da quebra dos cromossomos são muito significativos como será visto adiante na seção 3.7. Portanto [99] propõe o uso de técnicas sucessivas para a melhoria dos cromossomos filhos que são aplicadas ao custo de maior carga computacional.

3.4.1 Epistasia

Durante a execução do GA ocorrem as interações entre os cromossomos, nota-se que a contribuição de um gene do cromossomo para a aptidão total do indivíduo é dependente da contribuição dos outros genes, este fenômeno que é chamado de **epistasia** [21]. Este nome foi utilizado por analogia à biologia onde a epistasia é a interação intra-cromossômica dos genes. Um problema com baixa epistasia, ou seja, pouca influência entre os genes, pode ser rapidamente solucionado por um algoritmo de otimização mais simples do que o GA, pois o problema torna-se um ajuste de bits. Já um problema com alta epistasia possui um espaço de busca amplo e qualquer mudança em um dos genes acarreta mudanças nos efeitos dos outros genes, mudanças estas de difícil mapeamento, tornado inviável a utilização de GA [21, 22, 45]. Sendo assim, existem níveis de epistasia onde o uso de GA é indicada e recomendado, existem também níveis onde algoritmos mais simples e rápidos são mais efetivos e ainda existem níveis de epistasia onde outros algoritmos devem ser utilizados devido à alta interação intra-cromossômica. Os filtros, objetos deste trabalho, mesmo com alta influência entre os cromossomos encontram-se na faixa onde a utilização do GA é possível [22], como mostram os resultados apresentados nesta dissertação.

Em filtros, a alteração de um ou mais coeficientes pode, de acordo com a posição destes, levar à criação de um novo filtro com resposta bem diferente da original. O efeito da epistasia em filtros é muito visível quando um dado coeficiente do filtro tem seu sinal invertido, isto normalmente causa grandes mudanças na resposta em frequência do mesmo. Um bom exemplo disso é composto pelos filtros de coeficientes $h_1 = [1, 1, 1, 1]$ e $h_2 = [-1, 1, 1, -1]$, que possuem os mesmos coeficientes centrais. No primeiro caso obtêm-se um filtro passa-baixa e no segundo um filtro passa-alta. Assim nota-se que a epistasia, em filtros é um fator a ser levado em conta uma vez que grandes mudanças podem ocorrer nos filtros com a alteração de um coeficiente apenas.

3.5 Mutação

A mutação ocorre quando um ou mais genes do cromossomo são alterados, dando novas características ao ser resultante. Mas não necessariamente as mutações são benéficas, muitas vezes os indivíduos com mutações não sobrevivem por terem sofrido alterações que os tornam vulneráveis às condições ambientais, chamadas de mutações deletérias [20]. A mutação é um processo evolucionário com taxas de ocorrência diferentes em cada espécie [41] e não é necessáriamente a única forma de ocorrerem saltos evolucionários na natureza [73]. Existem alguns algoritmos genéticos que utilizam a mutação como principal fator de busca e não o cruzamento que é o mais usual [18]. A mutação em GA também consiste em alterações nos cromossomos, que é exemplificada na Figura 22.

Figura 22 Operador Mutação

Cada gene é avaliado para a mutação ou não de acordo com uma taxa de mutação pré-definida p_m, normalmente baixa [91], mas que em alguns casos pode ser alta [18]. No GA uma vez que determinado gene foi escolhido ele terá seu valor alterado aleatoriamente. Para genes representados por números binários ou SPT, sendo este último o objeto desta dissertação, essa alteração ocorre em um dígito aleatoriamente, o que impacta no valor total do cromossomo. Assim, como no cruzamento existem técnicas diferentes para a mutação. A mais comum envolve a alteração de múltiplos genes aleatoriamente, outras formas de mutação muito usadas em representações reais são a mutação de fronteira, que consiste na mudança de determinado gene para seu valor máximo ou mínimo, e a mutação de deslocamento que faz a troca de genes [59]. Em [99] é apresentada uma mutação onde a probabilidade de alteração dos bits que representam um dado coeficiente varia de acordo com a significância do bit, quanto mais significativo for um determinado bit menor é sua probabilidade de alteração. Porém não decidimos incluir esse método nesta dissertação. Neste trabalho, ao contrário do que ocorre na natureza, todos os genes do filtro controlam um mesmo fator, a resposta em frequência do filtro, o que torna possível a compensação de uma mutação em um gene com a mutação de outro [99], por isso é interessante utilizar fatores limitadores para a mutação, como por exemplo uma quantidade máxima de mutações por cromossomos ou então por genes para que não haja uma mudança muito radical na aptidão deste de uma geração para a outra.

A mutação utilizada neste trabalho é a mutação usual, ou seja, a mutação na qual um dígito do cromossomo é selecionado aleatoriamente e tem seu valor alterado, porém decidimos usar uma taxa variável de mutação e tivemos que adaptá-la aos dígitos SPT, como será visto na Seção 4.4.

3.6 Ciclos ou Gerações

O processo de seleção natural que ocorreu no planeta Terra levou bilhões de anos para atingir o ponto onde hoje estamos [5]. Num GA o tempo de execução do algoritmo é medido em ciclos ou gerações, a cada ciclo os operadores de cruzamento e mutação, vistos anteriormente nas Seções 3.4 e 3.5, são aplicados e a população resultante avaliada. A seguir, à população obtida no final de uma geração é reaplicada nesse mesmo processo. O processo se repete até que uma das condições de parada seja atingida. Condições de parada podem envolver um número máximo de gerações atingidas, ou um valor aceitável de aptidão para o melhor indivíduo, ou então um certo número de gerações sem melhoria da aptidão do melhor indivíduo, entre outros. O diagrama de funcionamento de um GA é exibido na Figura 23.

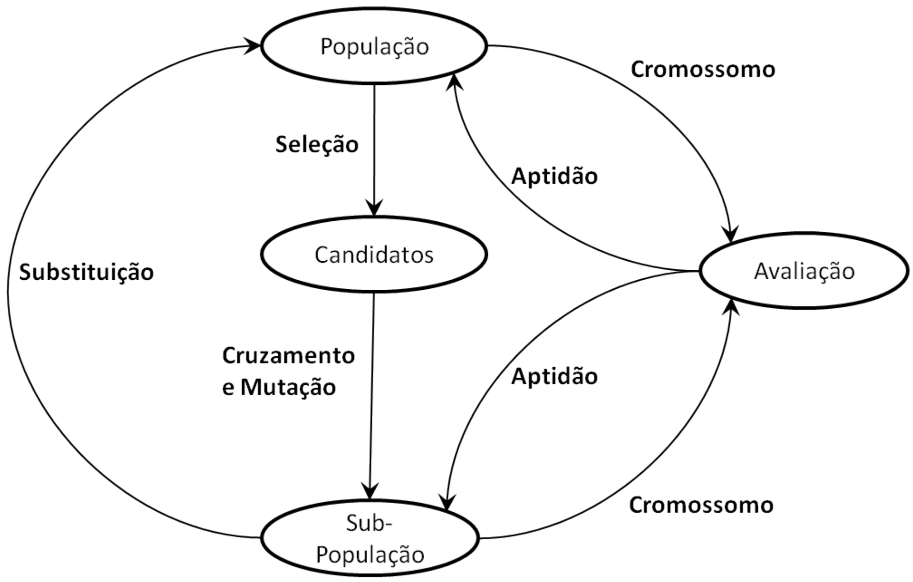


Figura 23 Diagrama de funcionamento de um algoritmo genético

3.7 Teoria dos Padrões

A teoria dos padrões, ou em inglês *Schema Theory* [10], nos permite analisar o efeito das alterações genéticas ocorridas durante a evolução do GA [18]. Ela tem como objetivo explicar como um GA pode resultar em um algoritmo de busca complexo e robusto pela amostragem de partições de hiperplanos do espaço de busca. A teoria dos padrões parte do principio que um cromossomo, chamado de **padrão**, representa uma amostra do hiperplano do qual ele faz parte. Esta teoria tem como objetivo analisar o efeito de mudanças sobre indivíduos de população, que nada mais é do que um conjunto de amostras de diferentes hiperplanos, de uma geração t para a seguinte $t+1$.

Para ilustrar como o GA faz uma amostragem de hiperplanos tomaremos um hiperplano de dimensão três como exemplo. Ou seja, caso seja feita uma analogia entre os hiperplanos e filtros teríamos o equivalente a filtros com 3 coeficientes, cada um codificado com um dígito apenas. Em [18] temos um exemplo com números binários de fácil entendimento, daremos um exemplo usando números SPT, pois foi a representação numérica escolhida para este trabalho. Assim, um plano será representado por todos os filtros que tiverem o primeiro coeficiente igual à 0, analogamente os outros planos são representados por um dígito comum. Sendo assim podemos escrever que a sequência $0xx$ representa o plano pl_1, onde x implica na não necessidade de determinar os bits dessa posição. A ordem de um hiperplano é dada pelo número de dígitos com valor determinado em um dado cromossomo. A aptidão de uma partição do hiperplano é obtida a partir da média de todos os elementos nela contidos tomados em uma sequência pré-definida, uma vez que combinações em ordens diferentes dos coeficientes geram resultados diferentes. Se considerarmos dois pontos do hiperplano $v_1 = 101$ e $v_2 = 11\bar{1}$ nota-se que ambos pertencem ao hiperplano $pl_2 = 1xx$ de ordem 1, e que ambos formam uma amostra de duas das nove possíveis combinações presentes no hiperplano pl_2 e a partir da amostra v_1 e v_2 a aptidão parcial da partição poderá ser obtida. Sendo assim o GA utiliza a população como uma amostra para estimar a aptidão de uma partição, quanto maior o número de amostras em uma determinada partição mais preciso é o valor estimado. O GA é por natureza compelido a restringir sua busca a hiperplanos de maior aptidão. Na primeira geração ocorre o efeito aleatório de maior escala no GA que é o sorteio de diversos indivíduos de diversas partições de hiperplanos, que vão compor a população inicial. Em seguida, o GA avalia a aptidão das partições de hiperplanos representadas pelos indivíduos da população inicial. Nas gerações subsequentes a busca é feita dentro dos hiperplanos que possuem maior aptidão estimada. O único efeito aleatório que ocorre da segunda geração em diante é a mutação, que tem baixa ocorrência como visto na Seção 3.5. Conforme a busca avança, a ordem dos hiperplanos vasculhados também aumenta o que implica na convergência do algoritmo. Esta teoria é bem explicada em [18, 53] e por isso nos ateremos a uma breve explicação matemática para fundamentar o uso do GA neste trabalho.

Seja H um hiperplano, $M(H,t)$ o número de representações (amostras) de H em uma geração t de uma determinada população, $t+inter$ a etapa intermediária logo após a seleção dos indivíduos que sofrerão mudanças, $f(H)$ a média da aptidão dos indivíduos do hiperplano H na geração atual e $\bar{f}$ a média de todos os indivíduos da geração atual. Considerando que apenas a seleção ocorreu, temos

$$M(H,t+inter) = M(H,t)\frac{f(H)}{\bar{f}}. \tag{40}$$

Para calcularmos M(H,t+1), devemos avaliar o efeito do cruzamento sobre os cromossomos, e para isso devemos levar em conta a taxa de cruzamento p_c, que é a probabilidade com a qual os indivíduos selecionados sofrem cruzamento. Sendo assim, a parte da população, que não sofreu cruzamento, permanecerá com a representação anterior, (40), porém, para a parte que sofre *crossover*, deve-se calcular a perda do rompimento da sequência de informações que o cromossomo representa, resultando em

$$M(H,t+1) = (1-p_c)M(H,t)\frac{f(H)}{\bar{f}} + p_c\left[M(H,t)\frac{f(H)}{\bar{f}}(1-l)+g\right], \tag{41}$$

onde l representa as perdas relativas à quebra do cromossomo e consequentemente das suas informações e g os ganhos obtidos no processo.

Neste ponto, assume-se que os efeitos do cruzamento no meio de um padrão são essencialmente maléficos, por quebrar o código que representa a partição do hiperplano no qual a busca está ocorrendo. Porém, nem sempre um corte intra-padrão gera uma perda, uma vez que o cromossomo escolhido para o cruzamento passará para cada um dos seus dois filhos parte das informações da sua partição do hiperplano transmitindo essas informações ao menos para a geração seguinte. Caso as informações que foram cortadas se rejuntarem perfeitamente, ou seja, sem nenhuma alteração, em uma geração futura não haverá perda nem ganhos para a representação do hiperplano dos cromossomos pais ela estaria reconstituída integralmente. Pode também haver um cruzamento entre pais que dividem uma mesma partição de hiperplano o que gera filhos com as informações do hiperplano dos pais e novas características, que representam os ganhos na equação 41. Nos outros casos o cruzamento é um efeito nocivo. Se considerarmos que o cruzamento gera apenas perdas obtemos

$$M(H,t+1) \geq (1-p_c)M(H,t)\frac{f(H)}{\bar{f}} + p_c\left[M(H,t)\frac{f(H)}{\bar{f}}(1-l)\right]. \tag{42}$$

persia

Se considerarmos crossover em um único ponto a probabilidade deste corte separar $\Delta(H)$ dígitos significativos que representam um hiperplano é

$$p = \frac{\Delta(H)}{L-1}, \tag{43}$$

onde L é o tamanho de um cromossomo. Sendo assim as perdas podem ser descritas por

$$l = \frac{\Delta(H)}{L-1}(1-P(H,t)), \tag{44}$$

onde $P(H,t)$ é o tamanho da população. Usando (44) em (42) e dividindo ambos os lados da desigualdade pelo tamanho da população $P(M,t)$, para $P(H,t+1)$ resulta em

$$P(H,t+1) \geq P(H,t)\frac{f(H)}{\bar{f}}\left[1-p_c\frac{\Delta(H)}{L-1}(1-P(H,t)\frac{f(H)}{\bar{f}})\right]. \tag{45}$$

A equação 45 leva em conta o cruzamento quando a seleção de ambos pais é feita por sua aptidão. Existem outras possibilidades de escolha dos cromossomos a serem cruzados, como a escolha aleatória [18], mas estas não são analisadas neste trabalho.

Nenhuma das equações acima descritas contempla a mutação e para introduzi-la assumiremos que $o(H)$ seja uma função que retorna a ordem do hiperplano H e que p_m seja a probabilidade de um dígito aleatório ser alterado. Se a mutação ocorrer em um dígito identificado por x, ou seja, um dígito que não está definido, o padrão desse cromossomo não será alterado. Logo a mutação só altera a informação de um cromossomo se o dígito que sofrer mutação estiver definido. Portanto a probabilidade da mutação ocorrer e não afetar o padrão representado é dada por

$$\bar{p}_m = (1 - p_m)^{o(H)}. \tag{46}$$

Obtemos então uma expansão da equação 45 que inclui a mutação

$$P(H, t+1) \geq P(H, t)\frac{f(H)}{\bar{f}}\left[1 - p_c\frac{\Delta H}{L-1}(1 - P(H, t)\frac{f(H)}{\bar{f}})\right](1 - p_m)^{o(H)}. \tag{47}$$

A equação 47 permite analisar como se comporta a distribuição das amostras de um determinado hiperplano, levando em conta os efeitos prejudiciais do cruzamento e da mutação, de uma geração para a seguinte. Sendo assim os padrões de baixa ordem tendem a proliferar ou desaparecer de acordo com sua aptidão média [54]. Conforme o GA converge tende a aumentar a quantidade de amostras nos hiperplanos de maior aptidão a cada geração e consequentemente maior é a ordem dos hiperplanos percorridos pelo GA. Lembrando que uma maior ordem implica em maior número de dígitos definidos em uma sequência representativa de um hiperplano. De acordo com a equação 46 aumenta a probabilidade de alteração de um padrão. Isso nos leva a considerar uma mutação com taxa variável e decrescente ao longo das gerações como será visto na Subseção 4.4.3.

Para preservar a diversidade genética e assim manter as informações das partições dos hiperplanos, devemos minimizar o efeito de perdas de informações produzidos pelos operadores cruzamento e mutação. Porém são estes operadores que estão por trás da evolução do algoritmo e assim são indispensáveis. Por isso, devemos balancear as taxas dos dois operadores para que haja garantia que boa parte do espaço de busca será varrido e não descartado devido a perdas de informação. Nesta seção vimos que de acordo com a teoria dos padrões o cruzamento é o principal responsável por fazer com que o GA percorra um sub-espaço de busca.

Uma vez decorridas várias gerações, podemos inferir que é possível que um determinado gene do cromossomo assuma um mesmo valor em todos os cromossomos de uma dada população. Caso isto se produza, antes da obtenção de uma solução satisfatória, temos o que chamamos de convergência prematura, uma vez que com o *crossover* apenas não teremos mais como alterar o valor desta posição. Este efeito é mais crítico em pequenas populações onde essa convergência prematura pode ocorrer rapidamente [18]. Neste aspecto, a mutação pode intervir beneficamente, pois ao alterar o valor de um elemento aleatório do cromossomo, ela pode inserir informações previamente perdidas.

Uma vez que nos Capítulos 1, 2 e 3 nós definimos as três estruturas básicas deste trabalho, ou seja, apresentamos as características dos filtros, dos números com representação SPT e dos algoritmos genéticos, as uniremos para atingir o nosso objetivo. No Capítulo 4 veremos como essa união foi feita.

4 PROJETO DE FILTROS UTILIZANDO GA

Neste trabalho projetam-se filtros com coeficientes quantizados em SPT, visando atender uma determinada especificação, utilizando GA. O computador onde todos os testes foram realizados tem as seguintes configurações: processador Intel Core 2 Duo (2 núcleos) 2Ghz (P7350), 4GB de memória RAM, sistema operacional Windows Vista 64-bits SP2. O programa utilizado para todos os testes, desde a modelagem até a execução é o MATLAB versão 7.6.0.324 (R2008a). A implementação básica do GA utilizado foi o “Binary and Real-Valued Simulation Evolution for Matlab” Copyright (C) 1996 C.R. Houck, J.A. Joines, M.G. Kay [17].

Neste capítulo combinaremos as técnicas vistas nos capítulos 1, 2 e 3 para a obtenção de filtros com representação SPT otimizados utilizando GA. Serão discutidos os ajustes feitos e a escolha dos parâmetros baseados no desempenho do algoritmo.

Todos os filtros utilizados neste capítulo são FIR de fase linear do tipo passa-baixa, projetados na forma direta conforme descrito na Subseção 1.1.1, de ordem 27 quando não-especificado. As especificações do filtro de teste foram: ganho na banda de passagem igual à 0dB, atenuação de -38dB na banda de rejeição, variação na banda de passagem máxima de 0,2dB e banda de passagem $pb \in [0, 0.1]$ e banda de rejeição $cb \in [0.2, 1]$, ambas normalizadas. As especificações e o filtro obtido através do algoritmo FIRPM estão representados na Figura 24. Muitos dos filtros obtidos neste capítulo não atendem às especificações, pois aqui o foco é o efeito que cada parâmetro do GA tem na produção de filtros. A obtenção de filtros adequados será discutida no capítulo 5.

Originalmente criado para otimizar funções de múltiplas variáveis independentes em representação real e binária, a implementação do algoritmo genético citado no início deste capítulo foi alterada para ajustar seu funcionamento à geração de filtros com coeficientes representados em SPTs.

Para podermos comparar algoritmos genéticos nos quesitos de qualidade dos resultados e velocidade deve-se estabelecer um conjunto de parâmetros padrão à partir do qual as modificações serão feitas em busca de melhores resultados. A Tabela 4 contém os

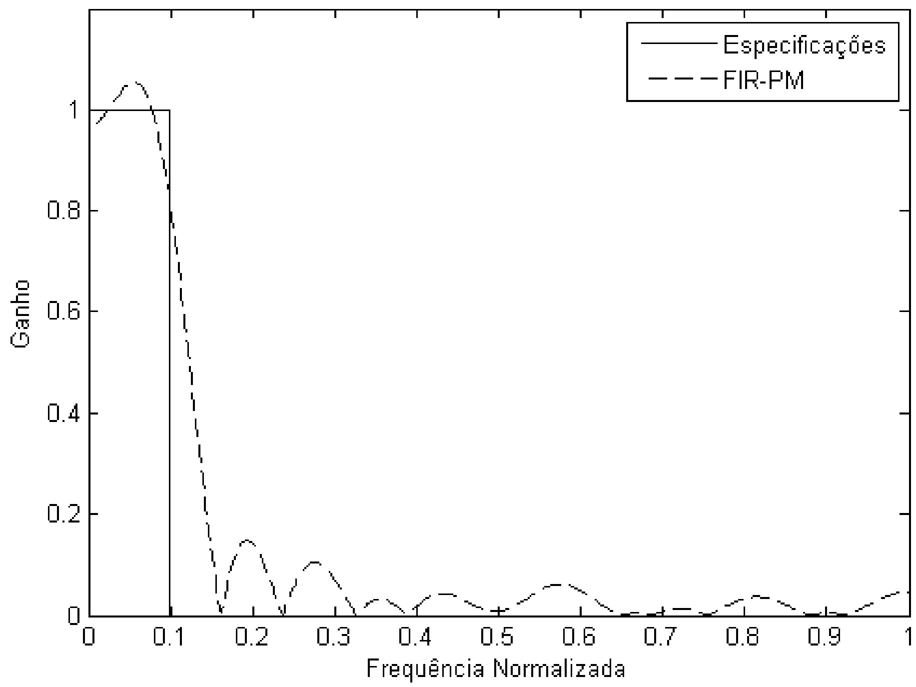


Figura 24 Especificações do filtro e resposta do filtro FIRPM

parâmetros básicos do GA que serão utilizados toda vez que um teste for feito à menos que algo diferente seja dito. Neste capítulo, por analisarmos os efeitos das mudanças das características do GA, são feitas alterações nas taxas de mutação e cruzamento, no tipo de função de seleção, no tamanho da população inicial, entre outros fatores. Assim obteremos os valores ideais dos parâmetros do GA para a otimização de filtros.

Parâmetro	**Padrão**
Representação Numérica	SPT
Dígitos para Representação	8
Função de Seleção	Seleção Proporcional
População Inicial	Aleatória
Tamanho da População	100 indivíduos
Operador de Cruzamento	1 ponto de corte
Taxa de Cruzamento	30%
Operador Mutação	Mutação Aleatória
Taxa de Mutação	1%
Critério de parada	1000 gerações decorridas

Tabela 4 Parâmetros padrões do GA

4.1 Critérios de Avaliação

Os filtros obtidos ao longo deste capítulo são oriundos de uma bateria de dez testes para cada parâmetro variado. Considerando que a aptidão de um indivíduo de uma população é dada pela função $fit(x)$, que será apresentada em 4.2, temos que o melhor indivíduo B de uma geração é

$$B(j) = \max_i(fit(x_{i,j})), \qquad i \in \{1, 2, ..., N\} \quad e \quad j \in \{1, 2, ..., g\}, \tag{48}$$

onde N é á quantidade de indivíduos em uma geração, g é a quantidade de gerações e a função $max(\cdot)$ retorna o maior valor de $\cdot$. O GA utilizado mantém um registro da progressão do melhor indivíduo $B(j)$ e a partir deste valor obtido dos T testes traçamos uma progressão média do melhor indivíduo dada por

$$C_m(j) = \sum_{k=1}^{T} \frac{B_k(j)}{T} = \sum_{k=1}^{T} \frac{C_k(j)}{T}, \tag{49}$$

onde C_k é a curva que contém o melhor indivíduo por geração para um determinado teste k.

Para avaliar a progressão da otimização de filtros com GA, as análises serão feitas baseadas em duas abordagens. A primeira consiste na avaliação da progressão dos indivíduos que resultam na mais alta e na mais baixa aptidão, ou seja, no melhor e no pior filtro. Em outras palavras, procura-se C_b e C_w tal que se

$$b = \max_{k \in \{1,2,...,T\}}(C_k(j = g)), \tag{50}$$

onde $max(x)$ retorna o valor máximo de x, teremos

$$C_b = \{C_k | C_k(j = g) = b\}. \tag{51}$$

E se

$$w = \min_{k \in \{1,2,...,T\}}(C_k(j = g)), \tag{52}$$

onde $min(\cdot)$ retorna o valor mínimo de $\cdot$, obtemos

$$C_w = \{C_k | C_k(j = g) = w\}. \tag{53}$$

A segunda abordagem consiste na avaliação do melhor e do pior indivíduo por geração em todos os testes o que permite uma visualização de como está convergindo o algoritmo. Ou seja,

$$C_{max}(j) = max(C_k(j)), \qquad j \in \{1, 2, ..., g\} \quad k \in \{1, 2, ..., T\}. \tag{54}$$

E

$$C_{min}(j) = min(C_k(j)), \qquad j \in \{1, 2, ..., g\} \quad k \in \{1, 2, ..., T\}. \tag{55}$$

Não existe uma relação direta entre C_b e C_{max} assim como não existe relação direta entre C_w e C_{min}, porém quando

$$max(C_b(j)) = b \Leftrightarrow C_b(j) = C_{max}(j). \tag{56}$$

Analogamente,

$$max(C_w(j)) = w \Leftrightarrow C_w(j) = C_{min}(j). \tag{57}$$

Estas duas abordagens serão apresentadas graficamente sempre que um dado parâmetro do filtro for alterado para permitir a visualização do comportamento do GA.

4.2 Função de Avaliação

Uma parte fundamental do GA é a definição da função de avaliação, esta define a aptidão de cada individuo da população. Para um filtro a função de avaliação deve modelar matematicamente a especificação desejada. Uma função de avaliação ruim leva o GA a resultados inadequados. Para chegar a uma função de avaliação adequada diversas abordagens foram consideradas e múltiplos testes e ajustes foram feitos. A seguir explicaremos as principais funções utilizadas.

4.2.1 Modelo 1

A primeira função desenvolvida analisava a resposta em frequência de cada um dos filtros gerados pelo GA levando em conta os seguintes critérios: variação máxima na magnitude (ΔP) e média da magnitude (M_P) na banda de passagem, assim como variação máxima na magnitude (ΔC) e média da magnitude na banda de rejeição (M_C). Estes quatro parâmetros formavam a base da função de avaliação e cada um deles possuía um peso diferente que dependia da característica que desejava-se exaltar, deste processo resultava a aptidão do indivíduo. Esta é a abordagem clássica para funções multi-objetivo e consiste em transformar o problema em um problema único com uma função objetivo escalar [1]. Neste modelo, a função de avaliação tem como objetivo aproximar a média da banda passante de um e minimizar a variação da função de transferência nesta banda, assim como minimizar a média e a variação na faixa de rejeição, quanto mais eficaz um dado parâmetro maior seu valor. Para a banda de transição nenhum tipo de análise foi feito, assim como nenhuma limitação foi imposta, nela qualquer comportamento da função de transferência foi permitido.

A seguir apresentaremos algumas funções essenciais ao entendimento da modelagem do problema, o cálculo é desenvolvido visando a obtenção de filtros passa-baixa, mas as funções que serão apresentadas podem ser facilmente adaptadas para gerar qualquer outro tipo de filtro.

Seja N o número de pontos equidistantes na metade superior do círculo unitário onde a resposta em frequência do filtro foi calculada. Deste modo cada N_i é equivalente a uma frequência dada por

$$N_i = \frac{i\omega}{N}, \tag{58}$$

onde $i \in \{1, 2, ..., N\}$. Temos então que N_p é quantidade destes pontos correspondentes à banda passante, N_t à banda de transição e N_c à banda de rejeição. Podemos definir a média na banda passante como

$$M_P = \sum_{i=1}^{N_p} \frac{|H(e^{jN_i})|}{N_p}. \tag{59}$$

Do mesmo modo definiremos a média da banda de rejeição por

$$M_C = \sum_{i=N_p+N_t}^{N_p+N_t+N_c} \frac{|H(e^{jN_i})|}{N_c}. \tag{60}$$

Além dessas duas médias definiremos a variação máxima na banda de passagem como

$$\Delta(P) = |\max_i(|H(e^{jN_i})|) - \min_i(|H(e^{jN_i})|)|, \tag{61}$$

onde $i \in [0, N_p]$. A variação máxima na banda de rejeição será

$$\Delta(C) = |\max_i(|H(e^{jN_i})|) - \min_i(|H(e^{jN_i})|)|, \tag{62}$$

onde $i \in [N_p + N_t, N_p]$. Unindo as equações (59), (60), (61) e (62) e adicionando pesos a cada uma destas obtemos a função

$$fit = \frac{W_t}{W_1\Delta(P) + W_2M_P + W_3\Delta(C) + W_4M_C}, \tag{63}$$

onde W_1,W_2,W_3,W_4 e W_t são pesos e $W_t = W_1 + W_2 + W_3 + W_4$.

A utilização desta função de avaliação não resultou em bons filtros, pois consistia praticamente em duas análises independentes, uma para banda passante e outra para a banda de rejeição. Sem uma função que interligasse as duas partes, os filtros resultantes produziam as mais diversas formas de funções de transferência, tudo dependia dos pesos escolhidos na função de avaliação. Outro grande problema desta abordagem é a definição manual dos pesos. Em [35,64,65] são propostos métodos de atribuição de pesos. Notou-se que caso pesos maiores fossem atribuídos aos fatores que regulavam as variações, surgiam filtros rejeita-faixa que tinham como banda de rejeição a banda de transição, caso os pesos favorecessem as médias apareciam grandes variações nas bandas de passagem e rejeição. Para limitar este efeito foi adicionado um parâmetro extra, chamado de At, que representava a atenuação média da banda de rejeição em relação à média na banda de passagem conforme

$$At = \frac{M_C}{M_P}. \tag{64}$$

A equação (65) consiste na equação (63) com a inclusão deste novo parâmetro e de um peso para este.

$$fit = \frac{W_t}{W_1\Delta(P) + W_2M_P + W_3\Delta(C) + W_4M_C + W_5At} \tag{65}$$

onde W_1,W_2,W_3,W_4,W_5, W_t são pesos e $W_t = W_1 + W_2 + W_3 + W_4 + W_5$.

Com esta alteração filtros próximos à especificação começaram a ser produzidos pelo GA. Para melhorar os resultados foram adicionadas ao código algumas funções que objetivavam limitar alguns parâmetros dos filtros produzidos, as chamadas funções de penalidade citadas na Subseção 3.1.1. Isto se fez necessário, uma vez que os pesos apenas ressaltam ou relevam uma característica e um filtro com quatro dos parâmetros bons não necessariamente implica em um filtro próximo das especificações. Sendo assim, estes limitadores entraram para coibir alguns excessos como variações muito grandes na resposta em frequência, tanto na banda passante $\Delta(P)$ como na banda de rejeição $\Delta(C)$, os quais não conseguíamos controlar apenas variando os pesos. Toda vez que estes limites eram ultrapassados a função limitadora reduzia em 90% a aptidão do indivíduo avaliado. Porém mesmo com todas as considerações descritas neste sub-item notou-se que por melhor que fosse um determinado filtro ele ainda possuia uma função de transferência que se afastava das especificações em algumas faixas do espectro, seja pelo surgimento de um "joelho"de amplitude maior do que a máxima permitida na banda de passante, ou baixa atenuação média na banda de rejeição. Estes resultados nunca apareciam juntos, porém sempre um deles estava presente. A Figura 25 mostra a resposta em frequência de um filtro obtido a partir deste modelo. Notou-se que havia uma compensação, quando uma das bandas atingia resultados muito bons a outra apresentava falhas e este efeito não podia ser resolvido alterando os pesos, pareceu ser um problema intrínseco à modelagem escolhida. Além disso, uma função de avaliação com 5 parâmetros e diversas funções limitadoras tornavam o GA muito complexo. O maior problema dessa função multi-objetivo é a tendência a obter-se soluções superiores em um objetivo, porém ruins em outros.

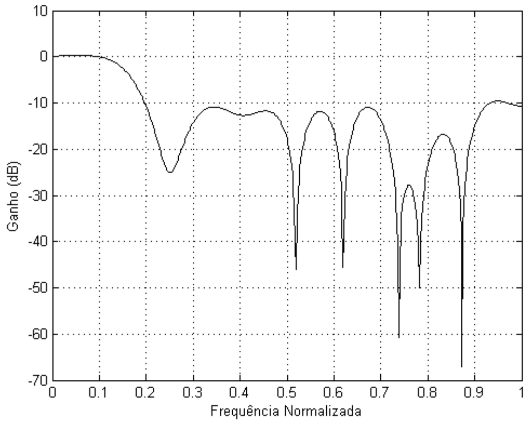


Figura 25 Função de transferência de um filtro obtido no primeiro modelo de avaliação

4.2.2 Modelo 2

A alternativa ao primeiro modelo foi baseada no método dos mínimos quadrados ponderados explicado na Subseção 1.3.1 e tem como objetivo minimizar a energia da função de erro dos filtros obtidos (H_f) em relação à especificação (H_e). Neste modelo H_e é definido por

$$H_e(\omega_i) = \sum_{i=1}^{N_p} |H_{esp}(N_i)|\delta(\omega - \omega_i) + \sum_{i=N_t+1}^{N_p+N_t+N_c} |H_{esp}(N_i)|\delta(\omega - \omega_i), \tag{66}$$

onde H_{esp} é a resposta em frequência que representa as especificações do filtro. A equação (66) pode ser reescrita como

$$H_e(\omega_i) = H_{ep}(\omega_i) + H_{ec}(\omega_i), \tag{67}$$

onde $H_{ep}(\omega_i)$ representa a resposta em frequência da especificação na banda de passagem e H_{ec} na banda de rejeição. Por sua vez H_f é dada por

$$H_f = \sum_{i=1}^{N_p} |H_{fil}(N_i)|\delta(\omega - \omega_i) + \sum_{i=N_t+1}^{N_c} |H_{fil}(N_i)|\delta(\omega - \omega_i), \tag{68}$$

onde H_{fil} é a resposta em frequência do filtro avaliado. Reescrevendo a equação (68) como fizemos com a equação (66) temos

$$H_f(\omega_i) = H_{fp}(\omega_i) + H_{fc}(\omega_i), \tag{69}$$

Novamente, neste modelo, a faixa correspondente à banda de transição é ignorada, não entrando no cálculo da função de erro, e assim permitindo qualquer comportamento neste intervalo. A função de avaliação é dada por

$$fit = \frac{1}{e}. \tag{70}$$

Sendo assim quando inserimos o erro entre as especificação e o filtro obtido com GA que é dado por $e = |H_e - H_f|^2$ chegamos à equação

$$fit = \frac{1}{|H_e(\omega) - H_f(\omega)|^2}. \tag{71}$$

Para ter maior controle do comportamento do filtro foram inseridos pesos na equação (71) de modo que a banda de passagem e a banda de rejeição pudessem ser ponderadas diferentemente. A nova função de erro é dada por

$$e = \frac{P_p|H_{ep}(\omega) - H_{fp}(\omega)|^2 + P_c|H_{ec}(\omega) - H_{fc}(\omega)|^2}{P_p + P_c}. \tag{72}$$

O que resulta em uma função de avaliação com a seguinte forma

$$fit = \frac{P_p + P_c}{P_p|H_{ep}(\omega) - H_{fp}(\omega)|^2 + P_c|H_{ec}(\omega) - H_{fc}(\omega)|^2}. \tag{73}$$

A escolha desses pesos foi feita com base em testes e será vista em profundidade no Capítulo 5. Pela experiência obtida com a função de avaliação discutida na seção 4.2.1 escolheu-se primeiramente utilizar pesos variáveis que relacionassem as bandas de passagem e rejeição. A primeira escolha de peso relacionava as variações dentro de cada banda e era dada por $P_p = \frac{\Delta pass}{\Delta cut}$ na banda de passagem e para a banda de rejeição o peso foi fixado em $P_c = 1$, como é feito no método dos mínimos quadrados ponderados [24]. Sendo assim, caso um determinado filtro apresentasse boa relação entre seus *ripples* o erro na banda de passagem seria crucial para sua boa avaliação uma vez que o peso desta faixa seria maior. Notou-se que a construção de filtros deste modo ainda deixava a desejar e por isso foram testados outros valores para o peso P_p que relacionavam as médias das bandas e combinações entre médias e variações nas bandas. Nenhum desses apresentou melhoria significativa em relação à primeira abordagem e por isso esses resultados não serão apresentados aqui.

Uma vez que a utilização de pesos variáveis apenas em P_p não foi muito frutífera, optou-se por utilizar variações no peso da banda de rejeição P_c também. Os parâmetros de variação utilizados foram os mesmos utilizados para P_p na primeira tentativa, mas dessa vez aplicavam-se aos dois pesos. Os resultados dessa alteração foram catastróficos, uma vez que nenhum filtro aceitável foi obtido. Analisando o porque desse acontecimento, para podermos solucionar esse problema, notamos que a função de avaliação de filtros vista na equação (73) é diretamente proporcional à soma dos pesos P_p e P_c e o algoritmo genético sendo capaz de encontrar diversas formas de evoluir, optava pela maximização do numerador em detrimento do erro. Normalmente os resultados obtidos nessas simulações possuíam um dos pesos muito maior do que o outro, o que tornava os resultados da banda à qual o peso menor estava atribuído desprezível, resultando nas mais diversas respostas que na maioria das vezes eram ruins. Concluiu-se então que o uso de pesos variáveis iria inexoravelmente levar à situação de troca entre as faixas, ou seja, sempre que se obtiver um bom resultado para uma a outra apresentará um resultado moderado ou ruim.

Por fim, optou-se pela utilização de pesos fixos, ajustados manualmente, pois estes não sofreriam do problema da troca e poderiam ser ajustados de acordo com as necessidades. Foi utilizando esse modelo que obtiveram-se bons resultados, que serão discutidos no Capítulo 5. Um exemplo de filtro é apresentado na Figura 26.

Para melhorar ligeiramente a aptidão foi utilizado um artifício visto em [24] que consiste em uma pequena alteração das especificações no fim da banda de passagem de modo que esta apresente uma tendência de comportamento. Para um filtro passa-baixa

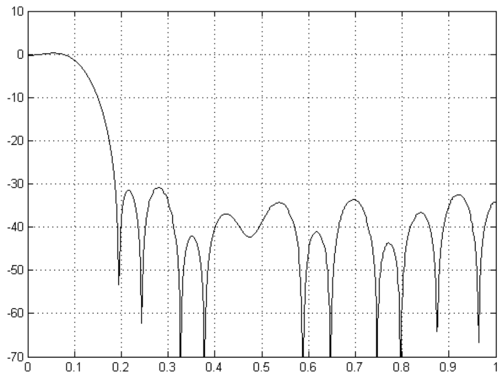


Figura 26 Função de transferência de um filtro obtido no segundo modelo de avaliação

esta tendência corresponde a reduções da ordem de $a_1 = -0,2dB$ e $a_2 - 0,5dB$ na função objetivo para que filtros com queda semelhante tenham maior aptidão do que filtros que não possuam essa característica. Para filtros passa-alta essas mudanças consistem nas atenuações de $a_3 = -0,5dB$ e $a_4 = -0,2dB$ no início da faixa de passagem. Sendo assim a especificação tem sua função de transferência H_e alterada conforme

$$H_{ea} = \sum_{i=1}^{N_{a_1}-1} |H_{esp}(N_i)|\delta(\omega - \omega_i) + \sum_{i=N_{a_1}}^{N_{a_2}-1} [|H_{esp}(N_i)|\delta(\omega - \omega_i) - a_1] + \\ + \sum_{i=N_{a_2}}^{N_p} [|H_{esp}(N_i)|\delta(\omega - \omega_i) - a_2] + \sum_{i=N_t+1}^{N_c} |H_{esp}(N_i)|\delta(\omega - \omega_i), \quad (74)$$

onde N_{a_1} e N_{a_2} são os pontos correspondentes às faixas de atenuação a_1 e a_2 respectivamente. De modo geral, o que é feito são atenuações no fim das faixas de passagem toda vez que estas precedem uma banda de transição e atenuações no início das bandas de passagem toda vez que estas sucedem uma banda de transição. A Figura 27 mostra uma generalização desse artifício para um filtro com três bandas passantes e duas bandas de rejeição.

Ao aplicarmos essas mudanças à especificação estamos indicando ao GA o caminho que este deve seguir, mas qualquer outro comportamento é aceitável uma vez que não são feitas alterações nas bandas de transição. Um decréscimo no fim de uma banda passante indica que a função tende a reduzir à partir do fim da banda e um decréscimo no início indica que a função vem aumentando na faixa anterior.

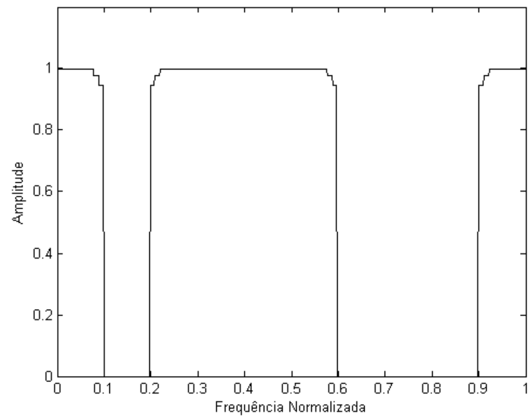


Figura 27 Artifício utilizado na banda de passagem para indicar o comportamento nas bandas de transição num filtro de múltiplas bandas

Assim sendo a função de avaliação utilizada neste trabalho será

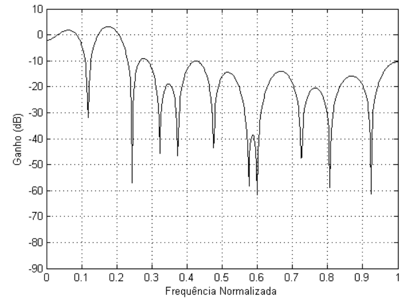
$$fit = \frac{P_p + P_c}{P_p|H_{eap}(e^{j\omega}) - H_{fp}(e^{j\omega})|^2 + P_c|H_{ec}(e^{j\omega}) - H_{fc}(e^{j\omega})|^2}, \quad (75)$$

onde $H_{eap}(e^{j\omega})$ consiste na resposta em frequência da especificação do filtro considerando os artifícios vistos acima.

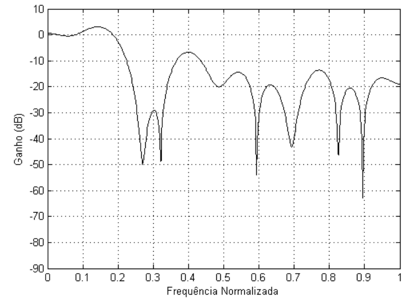
Quando trabalhamos com filtros de outros tipos temos que adaptar a função de avaliação às suas especificações. No caso de filtros passa-altas utilizamos as atenuações a_3 e a_4 no início da faixa de passagem. Nos casos dos filtros passa-faixa e rejeita-faixa, devem ser utilizadas combinações das quatro atenuações, no inicio e fim de cada faixa.

4.2.3 Aptidão

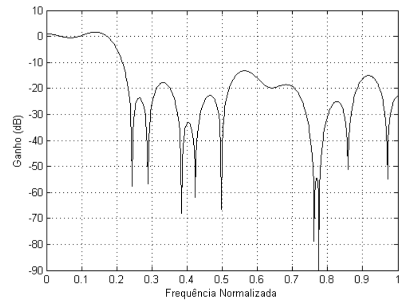
Para a interpretação dos gráficos de evolução dos indivíduos que serão apresentados ao longo deste Capítulo é necessário que o valor numérico da aptidão seja associado à um filtro. Caso contrário a aptidão torna-se um número sem significado. Utilizaremos o segundo modelo de função de avaliação, representado pela equação (75), para calcularmos a aptidão dos filtros. Para fazermos analogia entre valor da aptidão e filtro apresentaremos a resposta em frequência de filtros de diversas aptidões obtidos com o GA. A Figura 28 traz a relação entre função de transferência e uma das possíveis configurações de filtro para um dado valor de aptidão.



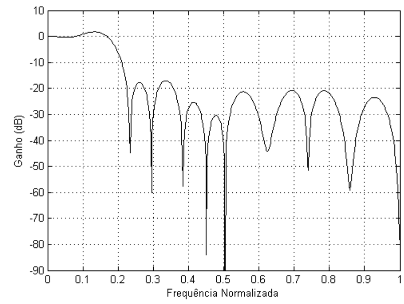
(a) Filtro com aptidão próxima a 25



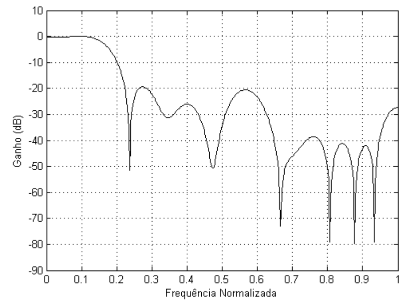
(b) Filtro com aptidão próxima a 50



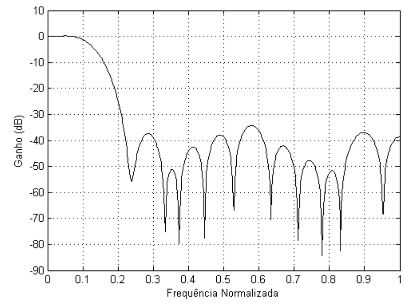
(c) Filtro com aptidão próxima a 100



(d) Filtro com aptidão próxima a 200



(e) Filtro com aptidão próxima a 400



(f) Filtro com aptidão próxima a 800

Figura 28 Funções de transferência para diversas aptidões

A partir da Figura 28 pode-se ter uma boa ideia do que representam as aptidões dos filtros. Lembrando que as especificações requerem um filtro de ganho de $0dB$ na banda passante, que normalizada representa o intervalo $[0; 0,1]$, e $-\infty dB$, idealmente, no restante do espectro de frequências. Nota-se também que filtros com aptidão acima de 50 apresentam uma resposta em frequência na banda de passagem razoável, por terem pequenas variações nesta banda. Outro ponto notável é que a banda de rejeição dos filtros que têm aptidão igual ou superior a 100 apresentam boa parte da sua função de transferência abaixo da marca dos -20dB de atenuação, e esta atenuação aumenta conforme aumenta a aptidão. Nota-se também que todos os filtros com exceção do filtro de aptidão 800 possuem como ponto fraco a baixa atenuação no início da banda de rejeição onde estes apresentam atenuação em torno de -10dB. Nesta região o filtro de aptidão 800 apresenta atenuação de -25dB contra -27dB do filtro obtido pelo método de Parks-McClellan. além de não ser *equiripple*. Nos filtros projetados neste trabalho foi notado que este é o ponto crítico na produção dos filtros, muitas vezes são produzidos bons filtros que pecam apenas na atenuação no início da banda de rejeição.

Uma importante observação consiste na avaliação da aptidão do filtro obtido pelo método de Parks-McClellan pela função de avaliação utilizada. A aptidão retornada para este filtro é de $fit = 478$. Durante os testes encontramos com facilidade filtros com melhor aptidão do que a do filtro de referência, porém isso não implica dizer que todo filtro com aptidão maior do que o filtro do método de Parks-McClellan é melhor do que este. Como o cálculo da aptidão é dado pela equação (75), que consiste num somatório dos erros, pode ocorrer a criação de um filtro com uma determinada função de transferência superior em boa parte do espectro de frequências à do filtro de precisão infinita, com exceção de um certo trecho, sendo assim esse filtro terá no total um erro menor. Mas é possível que, no segmento onde o filtro criado é inferior ao FIRPM, ele também viole as especificações, o que o torna menos efetivo para a aplicação para o qual ele foi projetado. A Figura 29 mostra a resposta em frequência do filtro FIRPM para comparação.

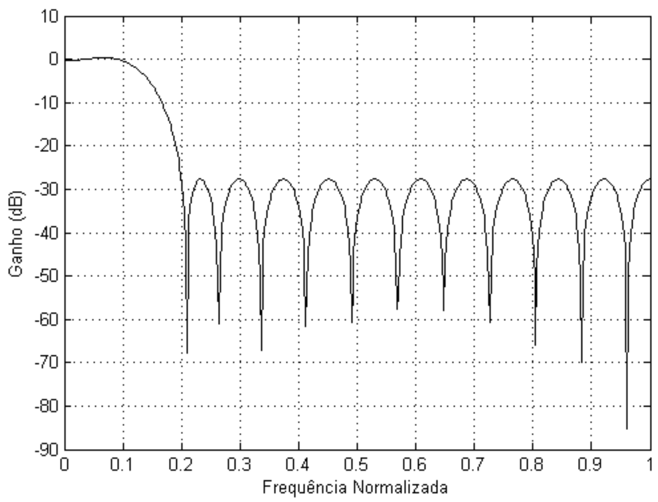


Figura 29 Função de transferência do filtro FIRPM

O valor da aptidão depende do filtro que está sendo projetado e este, por sua vez, depende da resposta em frequência do filtro, logo para especificações distintas haverá diferentes aptidões. Por isso não se pode comparar a aptidão entre filtros que não compartilhem as mesmas especificações.

4.3 População Inicial

Como foi visto no Capítulo 3, a população inicial provê ao GA as informações sobre o espaço de busca a ser vasculhado na procura da solução.

4.3.1 Tamanho da População Inicial

Um importante fator a ser avaliado no momento da inicialização do GA é o tamanho da população inicial. Esta delimita o espaço de busca, logo, quanto maior esta população, a princípio melhores serão os resultados das aptidões dos indivíduos. Além disso, ela define o tamanho das sub-populações do algoritmo. Sendo assim, quanto maior for esta população mais lento será o algoritmo se este tiver como princípio de parada um determinado número de gerações. Então, um balanço entre espaço de busca e velocidade deve ser realizado. A Tabela 5 apresenta o tempo necessário para a execução de uma execução do algoritmo para a criação de filtros de ordem 10, 27 e 39, com populações iniciais de 50, 100, 150 e 200 indivíduos, para o GA usado, cujos parâmetros estão explicitados na Tabela 4, que tem como critério de parada mil gerações.

Tempo (s)	**Ordem 10**	**Ordem 27**	**Ordem 39**
Pop=50	301	696	910
Pop=100	595	1342	1830
Pop=150	981	2012	2824
Pop=200	1180	2773	3591

Tabela 5 Tempo (em segundos) de execução do GA para filtros de diferentes ordem e população inicial

A Tabela 6 relaciona aptidão máxima do melhor indivíduo com a ordem do filtro e a população inicial para os mesmos casos usados na Tabela 5.

Da Tabela 5 e da Tabela 6 podemos concluir que a escolha adequada do tamanho da população leva à uma otimização com compromisso entre o tempo de processamento

Aptidão	**Ordem 10**	**Ordem 27**	**Ordem 39**
Pop=50	230	595	13
Pop=100	231	992	13
Pop=150	231	904	752
Pop=200	231	869	828

Tabela 6 Aptidão obtida para 10 testes para filtros gerados com GA de diferentes ordens e populações iniciais

e aptidão obtida. Para filtros de ordem 10 uma população de 50 indivíduos por geração é suficiente para uma convergência do algoritmo, caso populações maiores sejam utilizadas pode-se reduzir o número total de gerações uma vez que estes convergirão em um menor número de gerações como foi discutido na seção 3.2.2.

No caso de filtros de ordem 27, o uso de 50 indivíduos por geração resultou em resultados inferiores aos obtidos com populações de 100, 150 e 200 indivíduos, porém os resultados com estes três últimos tamanhos de populações demonstrou ser semelhante o que nos faz concluir que 100 indivíduos são suficientes para essa ordem de filtros. É importante notar que não necessariamente uma maior população implica em um melhor resultado, uma vez que o tamanho da população representa o espaço de busca, logo ele representa as tentativas por geração que um determinado algoritmo genético tem para achar melhorias. Com um maior número de tentativas pressupõe-se uma melhor progressão, o que de fato ocorre, porém pela natureza aleatória da inicialização e dos operadores genéticos pode ocorrer em alguns casos o surgimento de um melhor resultado em GAs usando populações menores.

Já no caso de filtros de ordem 39 percebe-se que populações de mais de 150 indivíduos são necessárias. A aptidão mostrada na Tabela 6 é originária de 10 testes e representa o padrão de comportamento. Em um dos testes efetuados com filtros de ordem 39 com população de 50 indivíduos ocorreu uma evolução até uma aptidão de 430, este resultado é por nós atribuído a uma inicialização adequada na qual diversos indivíduos com características favoráveis foram gerados.

4.4 Mutação

A mutação utilizada neste trabalho não limita o número de mutações por coeficiente, ou seja, pode ocorrer de não haver mudanças em um dos coeficientes enquanto para outro tem-se todos seus dígitos alterados. As taxas de mutação tendem a ser pequenas, normalmente menores do que 5%. Altas taxas de mutação são na maioria das vezes prejudiciaich:fcfigs por alterarem muito profundamente os cromossomos [18]. Usualmente o que se vê nos trabalhos [88] é o uso de uma taxa de mutação fixa com um valor adequado a aplicação de modo que o GA ao convergir tenha meios de escapar de máximos locais. Existem também algoritmos projetados de modo que a mutação seja o único operador de busca, descartando totalmente o cruzamento [18]. Para filtros, principalmente os de alta ordem, um efeito aleatório muito intenso como a mutação com altas taxas tem baixa probabilidade de gerar bons filtros [34], e por isso suas taxas devem permanecer baixas. Nesta Subseção abordaremos algumas análises do efeito da mutação nos filtros otimizados com GA.

4.4.1 Mutação SPT

Por usarmos números codificados em SPT e não uma codificação binária dos valores SPT como ocorre em [99], a mutação deve ser alterada pois um dígito escolhido para ser alterado pode assumir dois outros possíveis valores e não apenas um como ocorre para dígitos binários. Assim foi definida uma mutação aleatória, mostrada na Figura 30, equivalente à binária para SPT que consiste na alteração de um ou mais dígitos selecionados aleatoriamente, onde $d \in \{-1, 0, 1\}$, para um dos seus outros dois valores possíveis.

Neste trabalho não optou-se pelas formas canônicas ou mínimas dos números representados em SPT devido à possibilidade de operadores genéticos do GA, ao alterarem os cromossomos, poderem resultar em números sem conformidade com as representações CSD ou MSD vistas na Subseção 2.1.4. Existem diversos trabalhos que abordam alternativas à este problema [8, 30, 71] e usam artifícios para manter os números SPT em conformidade com a representação escolhida. Por não escolhermos utilizar as representações mínimas, o algoritmo perde em rapidez de processamento, porém ganha em diversidade

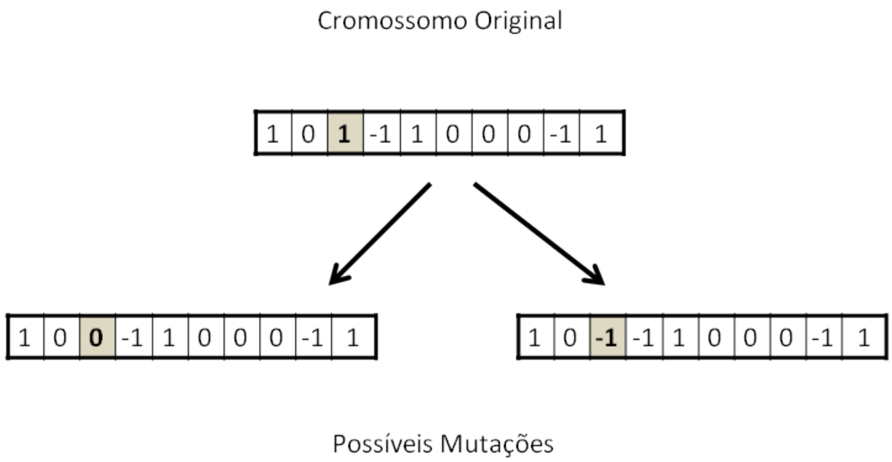


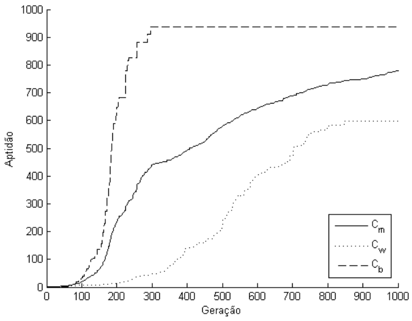
Figura 30 Mutação em coeficientes representados por números SPT

pois o produto de uma operação genética resulta em uma maior variedade de resultados. Para ilustrar este fato, consideraremos os seguintes números SPTs, $c_1 = 100\bar{1}$ e $c_2 = 0111$ ambos representando o valor 7, porém o número de somas em c_1 é $a_1 = 2$ enquanto em c_2 é $a_2 = 3$. O número c_1 está na base canônica, uma vez que não existem dois números não-zero adjacentes conforme visto na Subseção 2.1.4. Uma única alteração em c_1, advinda da mutação, que mantenha a base canônica só pode ocorrer neste caso no primeiro e no último dígito o que limita o número de mutações possíveis. Já o número c_2 não tem restrições e por isso tem um maior número de possibilidades, daí a maior diversidade do emprego de números SPT sem base mínima. No algoritmo proposto, o resultado final é apresentado na representação MSD, assim caso este filtro venha a ser implementado ele contemplará um número reduzido de somas, o que o tornará menos custoso.

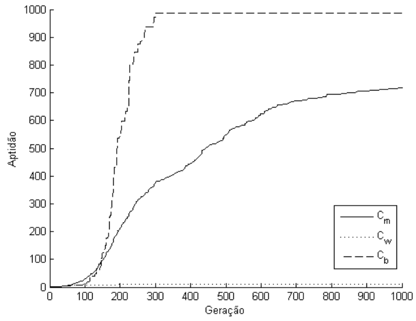
4.4.2 Mutação com Taxa Fixa

Outra característica importante da mutação empregada neste trabalho é o fato dela ser a única responsável pelas mudanças numéricas nos valores dos coeficientes do filtro, pois o modelo de cruzamento escolhido faz apenas o reposicionamento dos coeficientes como será visto na Seção 4.5. Quando um determinado indivíduo é selecionado para gerar descendentes todos os seus dígitos estão sujeitos à sofrerem mutação. Assim, o GA sorteia, para cada dígito, de acordo com as chances impostas pela taxa de mutação, se ele terá seu valor alterado.

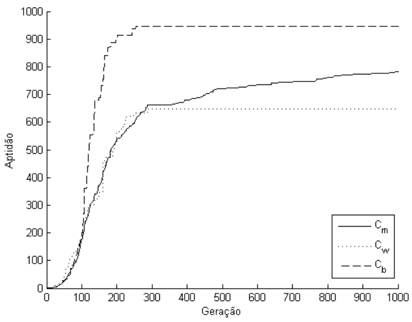
A Figura 31 mostra a evolução do algoritmo genético com taxas de mutação de 0,5%, 1%, 3% e 5% para a geração de filtros quando apenas a mutação é aplicada, isto é sem cruzamento. Nota-se que as progressões das curvas que produzem o melhor indivíduo C_{max} na Figura 31, têm comportamento diferente conforme a taxa mutação. Na figura Figura 31(a) e na Figura 31(b) com taxas de mutação de 0,5% e 1% a evolução ocorre gradualmente nas primeiras gerações e depois de atingir um máximo elevado fica estagnada nele até a parada do algoritmo como pode ser visto pelo comportamento das médias C_m. A característica mais marcante desta evolução é a sutileza com a qual a progressão se dá, uma vez que essa não apresenta muitos saltos. Uma curiosidade marcante vista na Figura 31(b) é o comportamento da curva C_{min} que é praticamente constante com baixíssima aptidão. Este comportamento particular poderia ser atribuído à baixa taxa de mutação que por um acaso não produziu coeficientes adequados ao filtro. Porém este acontecimento se repete, como será visto adiante nas Seções 4.4.3 e 4.4.4, para outras progressões que estão sob efeito de diferentes taxas de mutação. É interessante notar que isso só se dá para algoritmos que não utilizam o operador cruzamento, pois resultados semelhantes não foram obtidos nas Seções 4.5 e 4.6 quando há cruzamento. Atribuimos este efeito ao baixo potencial da população inicial, que representa um espaço de busca muito distante de uma boa resposta. Sendo assim, a mutação, por ter um comportamento aleatório de efeito simultâneo sobre diversos coeficientes, não é capaz de gerar alterações no cromossomo que produzam respostas melhores, fazendo com que o algoritmo fique perdido em uma subdivisão do seu espaço de busca. Na Figura 31(c) observa-se o comportamento do GA com taxa de mutação de 3%, este gráfico apresenta melhorias sutis misturadas com melhorias em saltos, o que reflete uma mutação mais agressiva do que a mutação anterior. A aptidão máxima é inferior à aptidão do GA com taxa de 1%, mas em contra-partida fica-se menos sucetível a paradas em máximos locais. Por fim quando a taxa de mutação é de 5%, Figura 31(d), a evolução do GA se dá por saltos e as chances de estagnação são menores, porém a aptidão média obtida é inferior à outras, uma vez que pequenos ajustes dos coeficientes tornam-se menos prováveis. Desta análise tira-se a conclusão que uma taxa de mutação variável é preferível, pois pode combinar a busca minuciosa das baixas taxas de mutação com a capacidade de evitar a convergência para um máximo local de taxas de mutação mais altas.



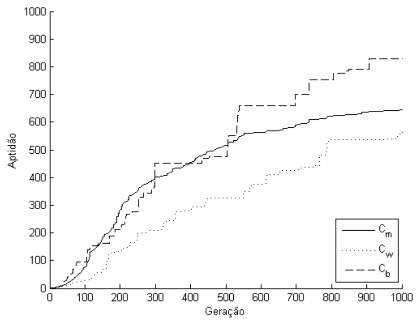
(a) Taxa de Mutação em 0,5%



(b) Taxa de Mutação em 1%



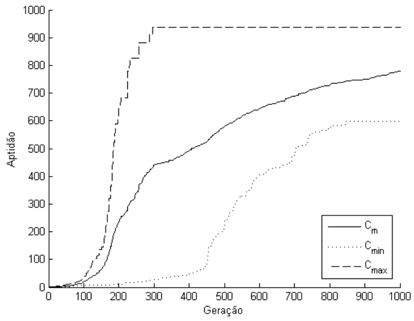
(c) Taxa de Mutação em 3%



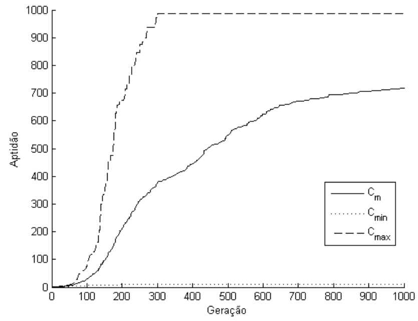
(d) Taxa de Mutação em 5%

Figura 31 Melhor e pior progressão do GA, assim como a média das progressões com diferentes taxas de mutação

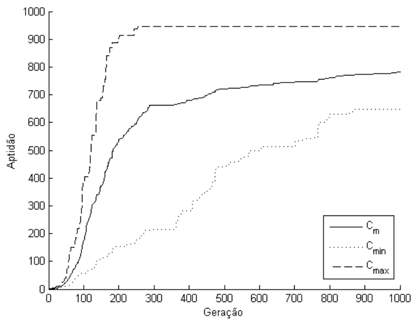
A Figura 32 mostra a progressão por geração do GA levando em consideração o melhor, o pior indivíduo e a média dos indivíduos para diferentes taxas de mutação. Ao analisarmos as curvas das médias percebemos que as progressões têm comportamentos semelhantes, ou seja, apresentam um crescimento pequeno no início que corresponde ao período em que o GA está a procura dos sub-espaços de busca onde boas respostas estão concentradas. Em seguida, as curvas apresentam um período de crescimento rápido, onde o algoritmo, uma vez tendo definido seu sub-espaço de busca, passa a procurar pela melhor resposta. Por fim o GA converge, mas continua vasculhando o espaço de busca, que neste ponto já é bem restrito, visando encontrar a melhor solução possível, o que caracteriza um baixo crescimento nas curvas das médias. O comportamento em saltos das taxas de mutação de 3% e 5% pode ser visto claramente na Figura 32(c) e na Figura 32(d).



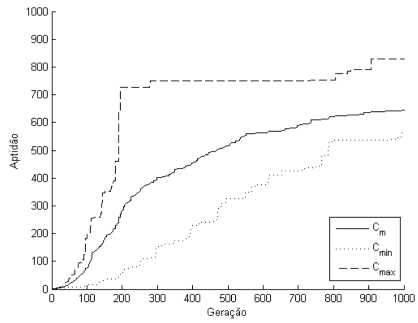
(a) Taxa de Mutação em 0,5%



(b) Taxa de Mutação em 1%



(c) Taxa de Mutação em 3%



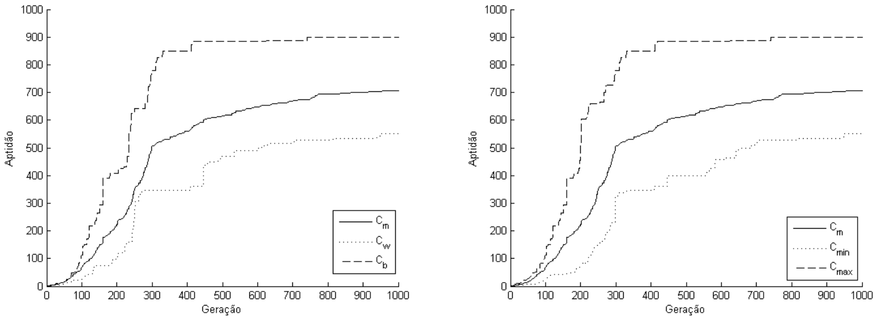
(d) Taxa de Mutação em 5%

Figura 32 Melhor, pior indivíduo e média dos indivíduos por geração com diferentes taxas de mutação

4.4.3 Mutação com Taxa Variável

Decidiu-se inicialmente pela utilização de uma mutação variável com taxa decrescente variando de 5% até 1%, com passo de 1%. Sendo assim, as gerações finais do GA têm baixa taxa de mutação e funcionam como um ajuste fino dos resultados obtidos nas gerações anteriores. A Figura 33 mostra a progressão das aptidões em um algoritmo genético com a mutação variável descrita. Analisando as curvas das médias C_m vemos, como era de se esperar, que as melhorias na aptidão dos indivíduos com este tipo de mutação ocorrem em saltos nas primeiras gerações e nas gerações finais elas passam a ser mais sutís.

Percebe-se que este tipo de mutação propicia um bom aproveitamento das taxas uma vez que no início, quando a população ainda não representa bons filtros, a alta taxa de mutação permite a busca de filtros com características bem diferentes das encontradas na população inicial, por isso sua evolução se dá muitas vezes em saltos. Uma vez determinadas as características boas para os filtros, o GA passa a aperfeiçoá-las com pequenas alterações, porém essa última etapa pode levar o algoritmo a ficar preso a um máximo local. Mas essa convergência, mesmo que prematura, normalmente já produz um bom filtro.



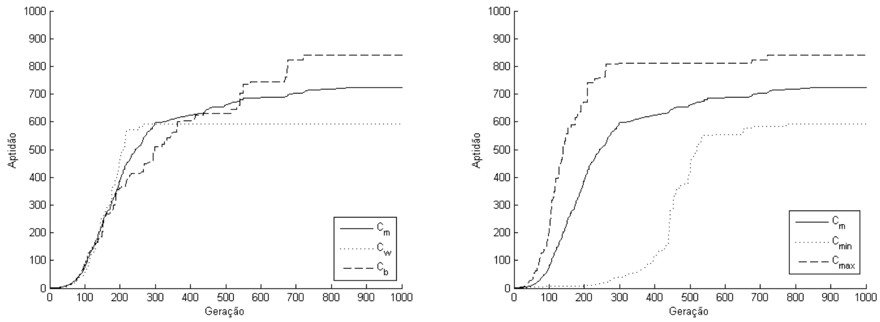
(a) Melhor, pior e média das progressões

(b) Melhor, pior e média dos indivíduos por geração

Figura 33 Progressão de um GA com taxa mutação decrescente

Outra alternativa é a utilização de taxas de mutação crescentes onde taxas mais altas ao final do algoritmo propiciariam a possibilidade de fuga do máximo local encontrado. Assim como com a taxa de mutação decrescente, a mutação aqui utilizada está entre 1% e 5% com passo de 1%. A Figura 34 mostra a progressão do GA com taxa de mutação crescente. Como era de se esperar a melhoria da aptidão se dá sutilmente nas primeiras gerações e mais abruptamente nas ultimas gerações onde os gráficos apresentam saltos. Nota-se a capacidade deste tipo de mutação de evitar a estagnação em máximos locais ao observarmos o comportamento da melhor progressão na Figura 34(a), onde após a metade do total de gerações as melhorias passaram a se dar aos saltos o que fez com que a progressão, que estava abaixo da média, se torna-se a de melhor resultado final. Por outro lado a alta taxa de mutação pode ter seu lado ruim, como ocorre quando essa altera demasiadamente as informações do cromosso levando o algoritmo que vinha progredindo bem à estagnação, como pode ser visto para a curva C_w na Figura 34(a).

Uma vez provadas as vantagens e desvantagens de taxas de mutação crescentes e decrescentes, foi imaginado um modo de combiná-las para potencializar a mutação. Chegamos então à uma mutação variável com taxas oscilantes, onde os máximos e mínimos de variação pertencem ao conjunto $tx = \{1\%, 2\%, 3\%, 4\%, 5\%\}$. As oscilações da taxa de

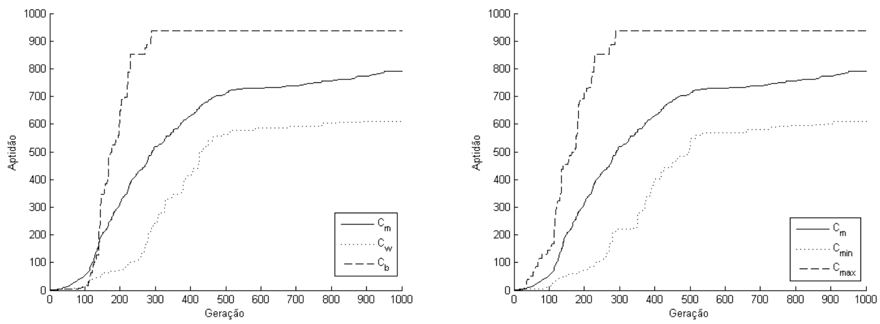


(a) Melhor, pior e média das progressões

(b) Melhor, pior e média dos indivíduos por geração

Figura 34 Progressão de um GA com taxa de mutação crescente

mutação são definidas aleatoriamente, com mesma probabilidade de ocorrência $p = 20\%$, e variam a cada vez que um indivíduo da população é sorteado para a mutação, assim uma mesma geração têm diferentes taxas de mutação. Com esta mutação obtêm-se a evolução apresentada na Figura 35. Nota-se que esta mutação obteve um bom resultado pois combina as vantagens das diversas taxas de mutação ao longo de todas as gerações. Chamaremos este método de mutação de **mutação oscilante tipo I** uma vez que outro modelo, derivado desse, será proposto.

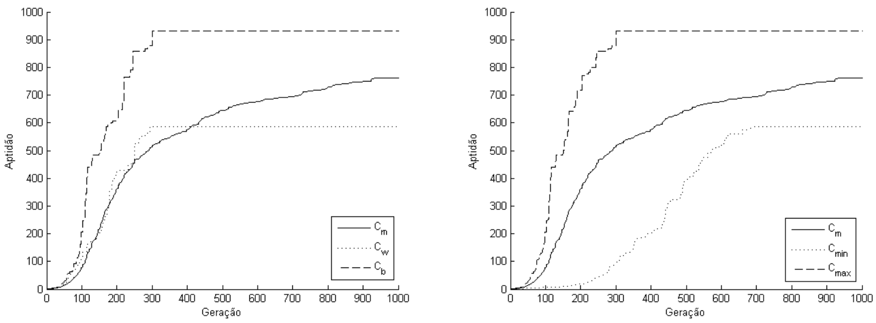


(a) Melhor, pior e média das progressões

(b) Melhor, pior e média dos indivíduos por geração

Figura 35 Progressão de um GA com taxa de mutação oscilante tipo I

A **mutação oscilante tipo II** tem seu comportamento apresentado na Figura 36. A diferença entre as mutações oscilantes tipo I e tipo II está na probabilidade de escolha das taxas de mutação. Na Subseção 4.4.2 percebemos que baixas taxas de mutação levam à médias de aptidão dos indivíduos mais altas do que altas taxas de mutação, sendo assim foi desenvolvido um método de mutação derivado da mutação oscilante tipo I onde as baixas taxas de mutação têm maior chance de serem escolhidas do que as taxas mais altas. Para as taxas de mutação $tx = \{1\%, 2\%, 3\%, 4\%, 5\%\}$ as probabilidades de sorteio são respectivamente $p = \{33\%, 27\%, 20\%, 13\%, 7\%\}$, esta escolha foi definida empiricamente. Porém, ao compararmos o comportamento da mutação oscilante tipo I com o tipo II não notamos grandes diferenças.



(a) Melhor, pior e média das progressões

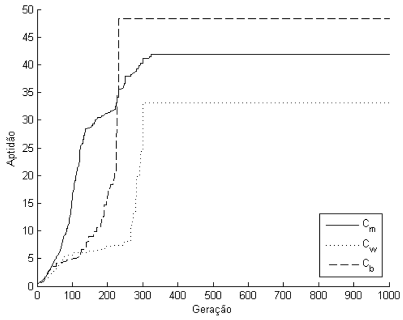
(b) Melhor, pior e média dos indivíduos por geração

Figura 36 Progressão de um GA com taxa de mutação oscilante tipo II

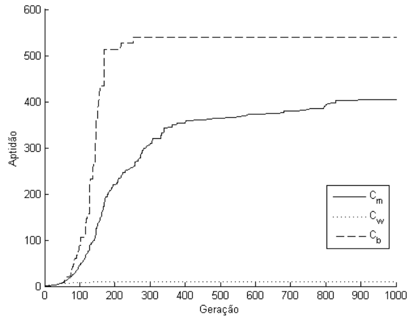
Mesmo não apresentando tantas vantagens em relação à mutação oscilante tipo I, o tipo de mutação escolhido para este trabalho foi a mutação oscilante tipo II, pois por ter maior probabilidade de ocorrências das baixas taxas de mutação, o desempenho médio do algoritmo tende a ser melhor para baixas taxas de mutação como pode ser visto se compararmos as sub-figuras da Figura 32.

4.4.4 Efeito da Mutação no Projeto Filtros de Diferentes Comprimentos

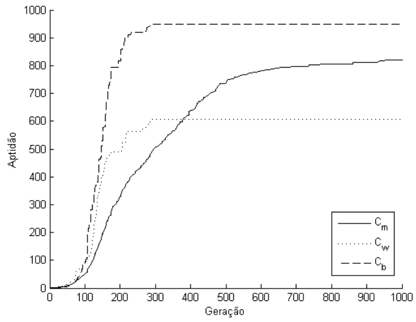
Outro fator que é de interesse avaliar é o efeito da mutação em filtros de diferentes comprimentos. Para isso, escolheu-se a taxa de mutação variável aleatória tipo II acima descrita e dois grupos de filtros, o primeiro composto por três filtros com 27 coeficientes e representações SPT com 4, 6 e 8 dígitos e outro grupo com mais três filtros, desta vez representados com 8 dígitos SPT e com 10, 27 e 39 coeficientes. A Figura 37 apresenta as progressões que resultaram na maior e pior aptidão final quando apenas a mutação é aplicada a filtros de ordem 27 para diferentes quantidades de dígitos representativos de seus coeficientes. A Figura 38 mostra a progressão da dos melhores e piores indivíduos por geração quando diferentes níveis de quantização são utilizados.



(a) Coeficientes quantizados com 4 dígitos SPT



(b) Coeficientes quantizados com 6 dígitos SPT

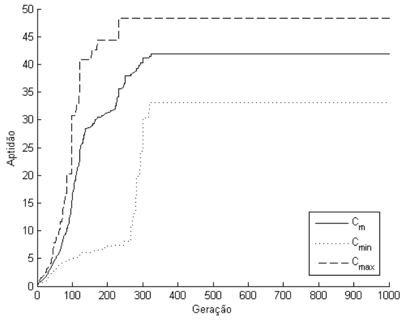


(c) Coeficientes quantizados com 8 dígitos SPT

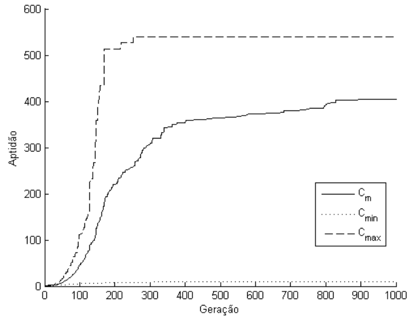
Figura 37 Melhor e pior progressão do GA, assim como a média das progressões obtidas com mutação apenas e com diversos níveis de quantização para filtros de ordem 27

Como era de se esperar para coeficientes quantizados com 4 dígitos o filtro resultante possui baixa aptidão e a mutação produz pequenas melhorias, uma vez que não existe muito 'material genético' a ser trabalhado. Como os coeficientes de 4 dígitos possuem elementos representativos das maiores potências de dois, 2^{-1} até 2^{-4}, pequenos ajustes são impossíveis o que causa a produção de filtros ruins. Para filtros quantizados com um maior número de dígitos SPT vemos que a mutação por ter um maior número de possibilidades de alterações e produz filtros melhores seja com 6 dígitos ou 8 dígitos. Para 8 dígitos temos uma aptidão maior do que para a quantização em 6 dígitos, como era de se esperar.

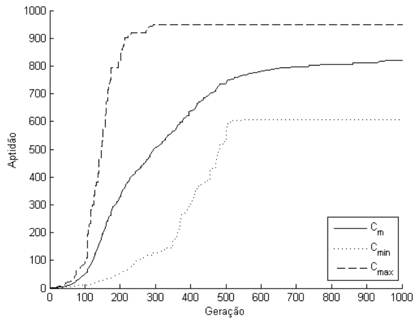
A Figura 39 mostra o efeito que a mutação tem sobre filtros de coeficientes com 8 dígitos de diferentes ordens. Para filtros com 10 coeficientes a mutação produz resultados



(a) Coeficientes quantizados com 4 dígitos SPT



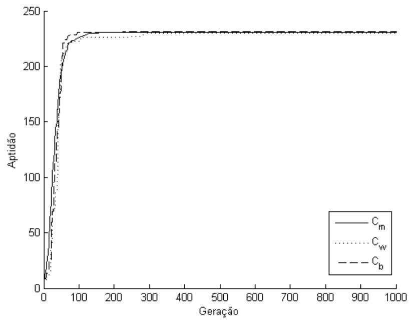
(b) Coeficientes quantizados com 6 dígitos SPT



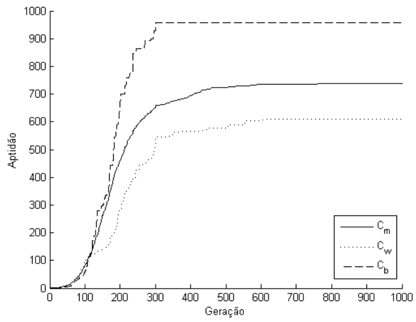
(c) Coeficientes quantizados com 8 dígitos SPT

Figura 38 Melhor, pior indivíduo e média dos indivíduos por geração obtidos com mutação apenas e com diversos níveis de quantização

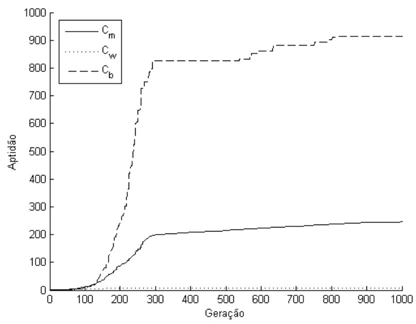
rápidos, o que é demonstrado pela inclinação da curva nas primeiras gerações, seguida da estagnação do algoritmo em um patamar de aptidão baixa, uma vez que um filtro com 10 coeficientes não é capaz de rivalizar em resposta em frequência com um filtro de ordem 27 ou 39 projetados para a mesma especificação. No caso do filtro de 27 coeficientes, que é o filtro base usado neste trabalho, a progressão ocorre durante um maior número de gerações e atinge uma boa aptidão. Para o caso de filtros com 39 coeficientes nestas condições nota-se que este praticamente não evolui e produz na média filtros ruins. Por outro lado, devemos levar em conta que um filtro com 39 coeficientes é um filtro mais complexo do que os outros dois, na Subseção 4.3.1 foi discutida a evolução de filtros dessa ordem e por isso fizemos o teste dobrando a população por geração, mantendo os outros parâmetros e o resultado foi um comportamento evolutivo semelhante aos outros filtros conforme vemos na Figura 41.



(a) Filtro com 10 coeficientes



(b) Filtro com 27 coeficientes



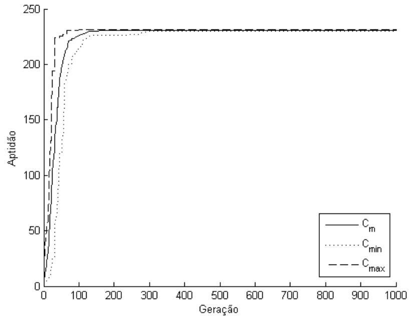
(c) Filtro com 39 coeficientes

Figura 39 Melhor e pior progressão do GA, assim como a média das progressões obtidas com mutação apenas, para filtros com diferentes números de coeficientes

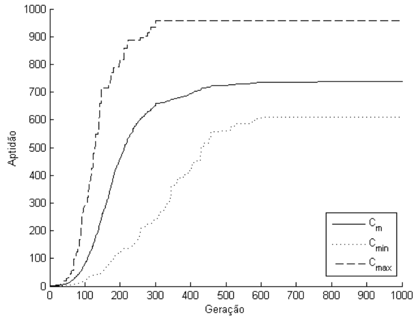
4.5 Cruzamento SPT

O cruzamento ocorre quando separamos dois cromossomos em um certo ponto e recombinamos as partes do código de um com o código do outro. Nesta seção trabalharemos com o operador cruzamento apenas, descartando a mutação. Alguns algoritmos, como o Breeder Genetic Algorithm (**BGA**) baseado nos métodos de criação de animais, utilizam apenas a recombinação genética, ou cruzamento, como gerador de novos indivíduos [56]. Foi visto na Seção 3.7 que o cruzamento é um operador potencialmente nocivo, pois fragmenta as informações dos hiperplanos presentes nos cromossomos. Sendo assim diversas formas de cruzamento foram aplicadas ao sistema para determinar qual melhor se encaixa a filtros FIR.

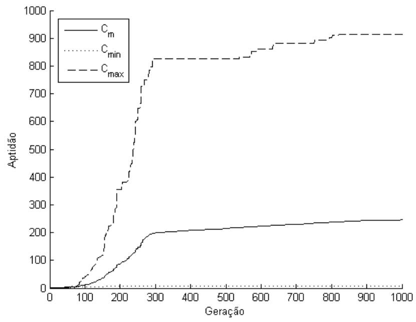
Para filtros, podemos considerar cada coeficiente um hiperplano codificado de acordo com a representação numérica escolhida e o número de elementos para a codificação. Então, a representação de um determinado coeficiente é também um hiperplano



(a) Filtro com 10 coeficientes



(b) Filtro com 27 coeficientes

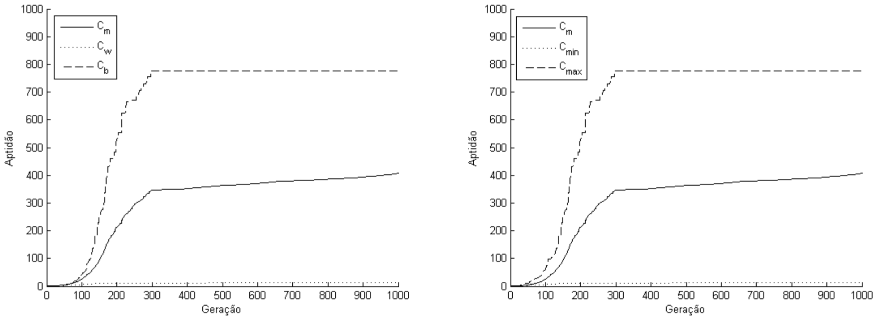


(c) Filtro com 39 coeficientes

Figura 40 Melhor, pior indivíduo e média dos indivíduos por geração com mutação apenas, para filtros com diferentes números de coeficientes

contido dentro de um hiperplano de ordem maior caracterizado pela ordenação dos genes. Sendo assim, devemos considerar o efeito de um corte no cromossomo duas vezes, uma para o hiperplano do coeficiente e outra para o hiperplano do cromossomo.

Os genes representam os coeficientes do filtro e o efeito resultante do cruzamento é basicamente a troca em blocos de dígitos, que podem conter mais de um gene ou até mesmo frações de genes, dos pais. No ponto de corte, que é aleatório, ocorre a perda de informações, uma vez que este corte pode ocorrer no meio dos dígitos que representam um determinado coeficiente, fazendo com que este assuma um novo valor ao ser combinado com o pedaço equivalente do cromossomo com o qual o cruzamento está ocorrendo. Esta alteração numérica só ocorre quando o ponto de corte não coincide com o fim do gene que representa um dado coeficiente. Este comportamento faz do *crossover* em filtros um operador pouco eficiente, pois gera pequenas melhorias de ciclo para ciclo e em poucas gerações o algorítimo cessa sua evolução e não converge para uma boa resposta. O tamanho da população altera significantemente o resultado obtido do cruzamento [84]. A Figura 42



(a) Melhor, pior e média das progressões

(b) Melhor, pior e média dos indivíduos por geração

Figura 41 Mutação em um filtro com 39 coeficientes com população de 200 indivíduos

mostra a evolução média do GA sem mutação, o limite de gerações foi reduzido para 100 uma vez que notou-se que com cruzamento apenas o GA não progride após a centésima geração. A aptidão média do GA sem mutação não atinge sequer o valor 2 para um filtro de 27 coeficientes e 8 dígitos SPT por coeficiente, contra algumas centenas no valor da aptidão com mutação apenas, vistos na Seção 4.4.

Objetivando uma melhoria nos resultados entre gerações por meio do cruzamento algumas modificações devem ser feitas. O cruzamento mais básico envolve o corte do cromossomo em apenas um ponto, outras envolvem pontos de corte múltiplos.

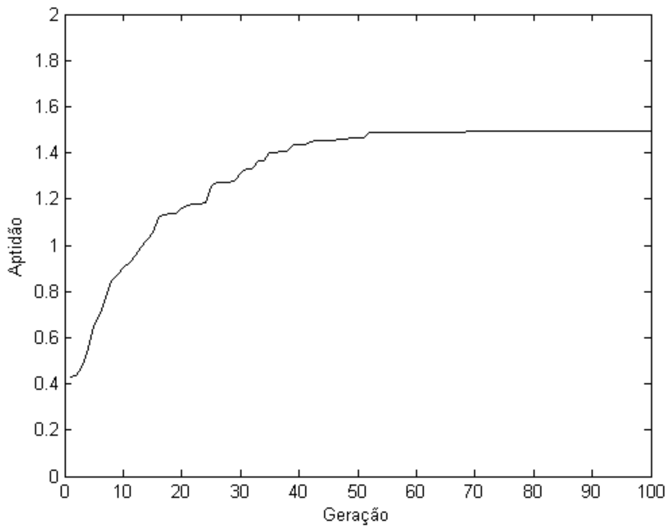


Figura 42 Evolução média do GA sem mutação

4.5.1 Cruzamento em Múltiplos Pontos e seu Efeito no Projeto de Filtros

Durante a execução deste trabalho notou-se que para filtros o tamanho do cromossomo pesa muito na escolha do número de pontos de corte. Filtros de grande comprimento ao serem repartidos poucas vezes, normalmente possuem pelo menos um de seus segmentos que concentra boa parte da informação do filtro que ele representa. Esta informação vai influir diretamente na resposta em frequência do filtro resultante, e a menos que este segmento concentre dados para uma boa resposta, o que normalmente não é o caso nas primeiras gerações do GA, resultará em um filtro ruim.

Para avaliarmos o efeito do cruzamento sobre os hiperplanos representados pelos coeficientes do filtro e seus dígitos devemos levar em consideração a forma como um hiperplano está representado. Tomando como exemplo dois filtros com representações em hiperplanos $f_1 = 1\bar{1}{*}{*}{*}{*}{*}{*}$ e $f_2 = 1{*}{*}{*}0{*}{*}{*}$, nota-se que ambos têm mesma dimensão $dim = 2$ e mesmo número de dígitos $L = 8$. Porém as chances de um corte romper a informação contida em f_1 é de $pf_1 = \frac{1}{L-1}$ uma vez que só haverá perda de informação se o corte ocorrer entre o primeiro e segundo dígito. Por outro lado, para f_2 a probabilidade de perda de informação é de $pf_2 = \frac{4}{L-1}$, pois qualquer corte entre o primeiro e o quinto dígito leva a uma perda. Vemos então, que quanto mais espaçados estiverem os dígitos representativos de um determinado hiperplano, e quanto maior a ordem do hiperplano, maior as probabilidades de ocorrer perdas durante o cruzamento.

Se pensarmos em hiperplanos descartando os efeitos da epistasia, a quantidade de hiperplanos alterados durante um corte em um único ponto é pequena, mas por outro lado as chances do cromossomo par repor essas informações é ainda menor, pois os segmentos a serem reconstituídos são grandes. Analogamente, as perdas geradas por múltiplos pontos de corte são potencialmente maiores, mas como o tamanho dos segmentos são pequenos, é maior a probabilidade de ocorrência de uma reposição de um dado hiperplano. Se definirmos o número de cortes como p, teremos que o número de segmento nos quais o cromossomo é dividido será $p' = p + 1$. Em contra-partida, quanto maior for p' menor o tamanho médio dos segmentos dado por $L_{ms} = \frac{c*d}{p'}$, lembrando que c é o número de coeficientes do filtro e d o número de dígitos que compõe cada coeficiente. Nas primeiras gerações, onde o número de hiperplanos definidos pela otimização do GA ainda é pequeno, o cruzamento é pouco nocivo, pois há pouca informação a ser quebrada. Conforme o GA progride e seus hiperplanos vão sendo definidos aumentam as chances de quebras prejudiciais, como visto na Seção 3.7. Sendo assim, utilizando os filtros f_1 e f_2, acima descritos, para 2 pontos de corte temos que as probabilidades de perda de informação serão respectivamente $pf_1 = \frac{1}{L-1}$ e $pf_2 = \frac{4}{L-1} + \frac{3}{L-1} * \frac{3}{L-2}$. Ou seja, para hiperplanos de baixa ordem e compactos, como é o caso de f_1, e considerando que durante o cruzamento os pontos de corte são necessariamente diferentes, a probabilidade de perda de informação é pequena e tende a ficar inalterada. Se a probabilidade de quebra de um hiperplano é pequena, maiores são as chances de que boas soluções sejam encontradas. Logo, para filtros otimizados via GA, um número de cortes elevado nas primeiras gerações é um atrativo, pois propicia uma melhoria na aptidão dos indivíduos.

Um aumento do número de cortes também aumenta o número de coeficientes que terão seus valores alterados quando o corte se dá no meio de sua sequência de dígitos, reduzindo as limitações do cruzamento quando aplicado à filtros digitais. Em [26,83] vemos que um grande número de cortes ajuda a superar a limitada diversidade de informação de uma pequena população. Porém, conforme [18] se aumentarmos demasiadamente o número de cortes o efeito destrutivo do cruzamento torna-se muito evidente fazendo com que o algoritmo atinja uma convergência prematura, resultando em um filtro ruim.

Agora, consideraremos apenas o efeito da epistasia nos filtros, que foi definida na Subseção 3.4.1. Durante o cruzamento há mistura do código genético dos pais, o que, em filtros, equivale à troca de uma parcela dos coeficientes de cada filtro envolvido no processo. O resultado disso são indivíduos filhos, que por possuírem uma nova configuração de cromossomos, possuirão respostas em frequência muito diferente dos pais, principalmente nas primeiras gerações quando os indivíduos são muito diferentes entre si devido à inicialização aleatória. Estas novas interações entre os coeficientes permitem um rápida varredura do espaço de busca em busca de regiões de máximo. Ao aumentarmos o número de cortes aumentamos as combinações possíveis e portanto aumentamos a área de varredura.

A Figura 43 apresenta um esquema de *crossover* com múltiplos pontos de corte, nesta vemos que a troca de informações entre os cromossomos, quando ocorre o cruzamento com múltiplos cortes, é feita trocando os segmentos delimitados pelos pontos de corte alternadamente entre os pais. Todos os pontos de corte neste tipo de cruzamento são aleatórios. O *crossover* com múltiplos pontos de corte é abordado aqui pois é base de inspiração para o cruzamento utilizado neste trabalho que será apresentado na Subseção 4.5.2.

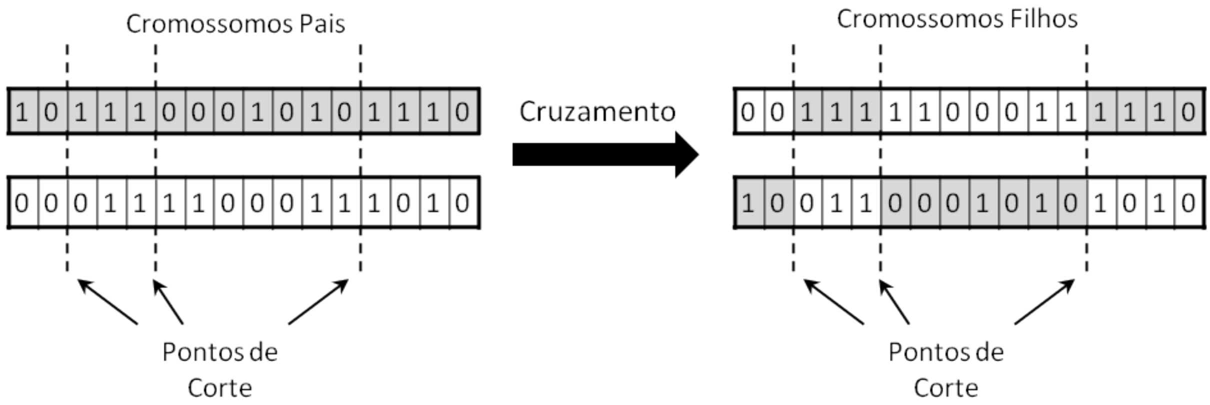
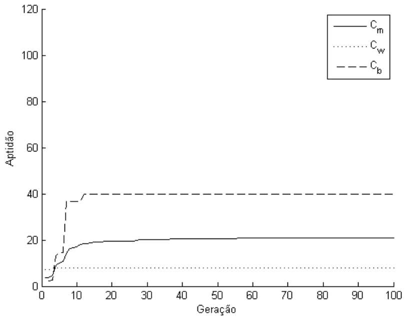
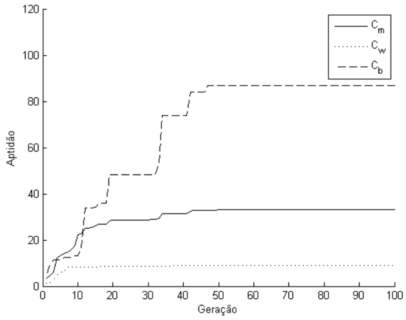


Figura 43 Esquema de cruzamento com múltiplos pontos de corte

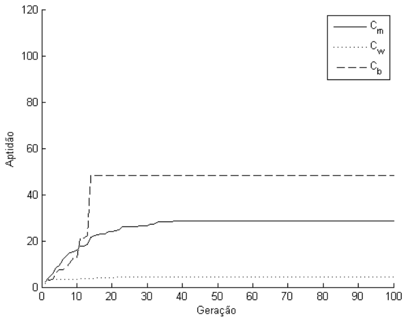
Assim como foi feito com a mutação, faremos um comparativo entre o tamanho do filtro e diferentes tipos de cruzamento. Os filtros submetidos à esses testes têm, novamente, ordens 10, 27 e 39. Os cruzamentos utilizados neste teste são cruzamentos de $p \in \{1, 2, 3, 10\}$ pontos de corte. A Figura 44 mostra a evolução do GA sem mutação para filtros de ordem 10 e número de pontos de corte p. Nota-se que para filtros dessa ordem conforme aumentamos o número de corte aumenta também a aptidão média do melhor indivíduo, obtida pela equação (49). Devemos atentar para o fato da aptidão dos melhores indivíduos da população inicial ser alta se compararmos com as aptidões dos indivíduos das populações iniciais dos filtros de ordem 27 e 39 como pode ser visto na Tabela 7. Isso é devido à razão entre o tamanho da população inicial e a ordem do filtro, o que aumenta a probabilidade de sorteio de filtros de aptidão elevada, considerando que as aptidões nestes estágios iniciais são baixíssimas, para filtros de ordens maiores. Na Figura 44(b) mesmo o melhor filtro encontrado sendo bem superior aos outros encontrados nos cruzamentos com 1, 2 e 3 pontos de corte temos que a média dos filtros nesses testes não é elevada.



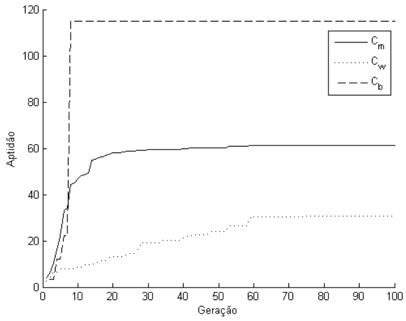
(a) Cruzamento com 1 ponto de corte



(b) Cruzamento com 2 pontos de corte



(c) Cruzamento com 3 pontos de corte



(d) Cruzamento com 10 pontos de corte

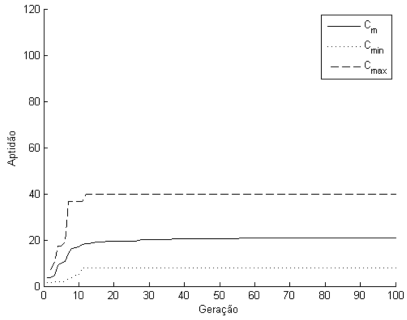
Figura 44 Melhor, pior e média das progressões do GA com cruzamento com diversos pontos de corte em filtros de 10 coeficientes

Número de coeficientes	**10**	**27**	**39**
Média da aptidão total	0,564	0,167	0,107
Aptidão do melhor indivíduo	9,743	0,645	0,303
Média da aptidão dos melhores indivíduos	3,872	0,5	0,264

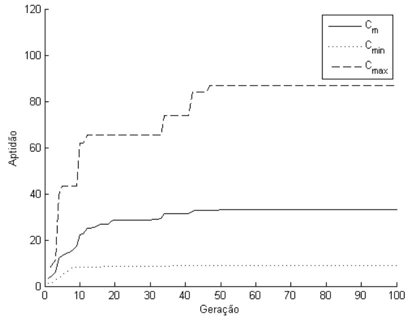
Tabela 7 Resultados obtidos em populações inicias de 100 indivíduos para 10 testes com filtros de 3 comprimentos diferentes

Na Figura 45 podemos ver a progressão do GA com cruzamento com diferentes quantidades de pontos de corte e sem mutação, para um filtro de 10 coeficientes ao considerarmos todos os testes efetuados em conjunto. Uma análise das médias dessa figura mostra um comportamento, que se repete nos filtros de 27 e 39 coeficientes, que consiste em uma média menor para filtros com 1 ponto de corte, seguida das médias com 2 e 3 pontos de corte ligeiramente mais altas e tendo a mais alta média com cruzamento com 10 pontos de corte.

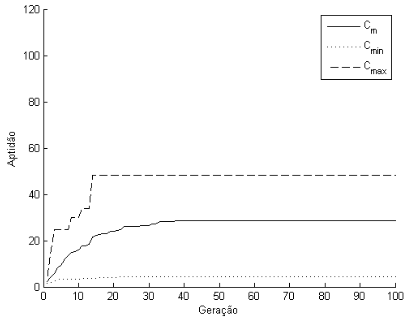
A Figura 46 apresenta a aptidão de filtros gerados apenas com cruzamento para filtros de ordem 27. Neste caso novamente observa-se que um maior número de cortes



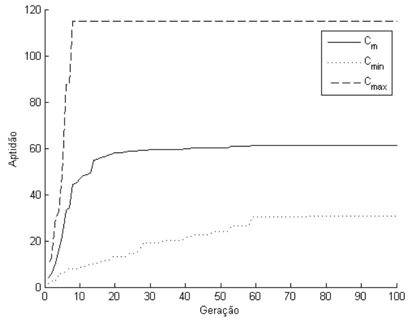
(a) Cruzamento com 1 ponto de corte



(b) Cruzamento com 2 pontos de corte



(c) Cruzamento com 3 pontos de corte



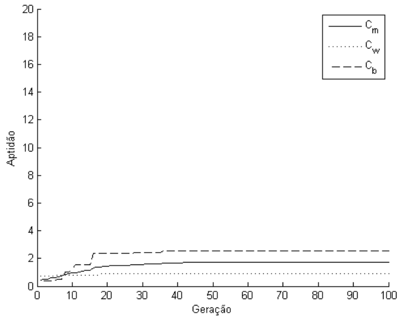
(d) Cruzamento com 10 pontos de corte

Figura 45 Melhor, pior e média dos melhores indivíduos por teste do GA com cruzamento com diversos pontos de corte em filtros de 10 coeficientes

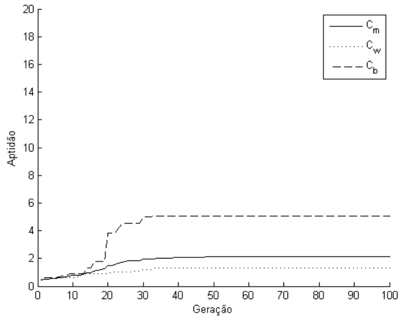
aumenta a aptidão média dos indivíduos gerados. Porém, as aptidões obtidas com filtros de ordem 27 são muito inferiores às aptidões obtidas sob mesmas condições para filtros de ordem 10.

A Figura 47 apresenta as progressões dos indivíduos quando todos os testes são considerados para cruzamento com diferentes quantidades de pontos de corte para filtros de ordem 27.

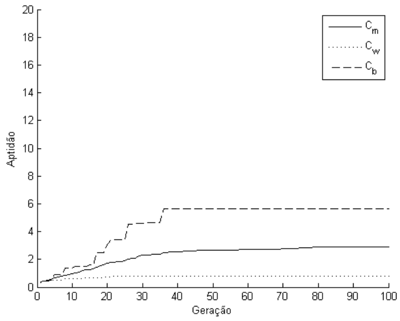
A Figura 48 apresenta a progressão da aptidão dos filtros de ordem 39 gerados com GAs sem mutação. Como era de se esperar a aptidão destes filtros ficou abaixo da aptidão dos filtros de ordem 10 e 27. Lembramos que parte deste resultado é devido ao pequeno número de indivíduos nas populações para filtros de ordem 39. Porém, conforme aumenta-se a ordem do filtro e mantém-se o número de cortes reduzimos a proporção entre número de cortes e tamanho de filtros o que prejudica a busca com o operador cruzamento exatamente nas gerações onde ele é menos nocivo, ou seja quando poucos hiperplanos estão definidos. Assim como nos outros casos, o aumento do número de pontos de corte



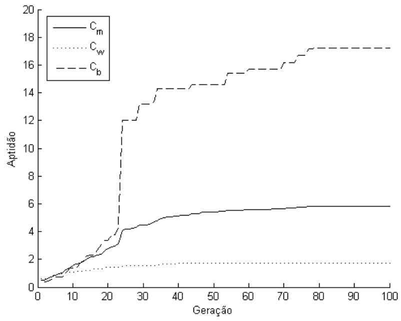
(a) Cruzamento com 1 ponto de corte



(b) Cruzamento com 2 pontos de corte



(c) Cruzamento com 3 pontos de corte



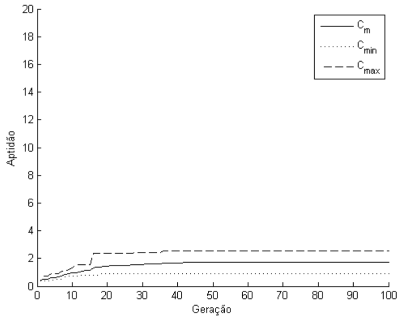
(d) Cruzamento com 10 pontos de corte

Figura 46 Melhor, pior e média das progressões do GA com cruzamento com diversos pontos de corte em filtros de 27 coeficientes

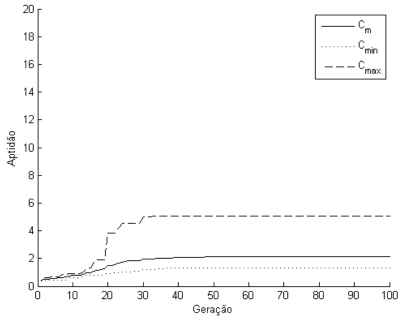
propiciou melhorias na aptidão dos indivíduos. Na Figura 49 temos as progressões quando são considerados todos os testes efetuados com cruzamento com diversos pontos de corte em filtros de 39 coeficientes.

Com os resultados apresentados concluímos que quanto maior é a quantidade de pontos de corte melhor é o resultado do GA. De fato, comprovamos isto ao utilizarmos o cruzamento uniforme, método de cruzamento muito utilizado [18]. Nele cada dígito que compõe os cromossomos torna-se passível de ser um ponto de corte. Durante o processo de cruzamento faz-se um sorteio a cada dígito para saber se estes serão ou não trocados entre os cromossomos pais, produzindo cromossomos filhos onde a mistura das informações é feita dígito a dígito. Este é o tipo de cruzamento com o maior efeito destrutivo sobre o cromossomo [75]. A Figura 50 apresenta a evolução do GA com este tipo de cruzamento sem influência da mutação.

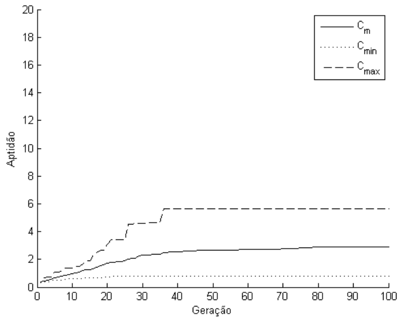
O cruzamento uniforme produz resultados muito superirores aos outros cruzamentos apresentados até o momento. O seu comportamento aleatório durante a escolha dos



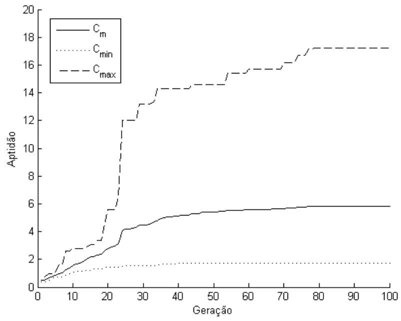
(a) Cruzamento com 1 ponto de corte



(b) Cruzamento com 2 pontos de corte



(c) Cruzamento com 3 pontos de corte



(d) Cruzamento com 10 pontos de corte

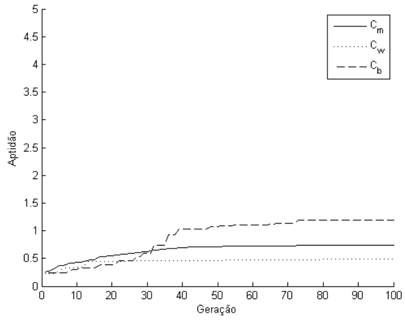
Figura 47 Melhor, pior e média dos melhores indivíduos por teste do GA com cruzamento com diversos pontos de corte em filtros de 27 coeficientes

dígitos a serem herdados pelos cromossomos filhos causa a quebra de grande parte dos padrões existentes nos cromossomos pais, porém estas quebras são benéficas nas primeiras gerações, que possuem indivíduos de baixíssima aptidão, permitindo a geração de indivíduos mais aptos.

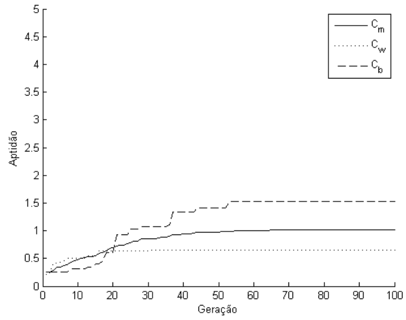
4.5.2 Cruzamento Aprimorado por Blocos

O número máximo de cortes possível em um cromossomo de c coeficientes cada qual contendo d dígitos é dado por $cp = c * d - 1$. Porém, este modelo de cruzamento equivale ao cruzamento uniforme, que leva a uma completa reorganização dos bits do gene.

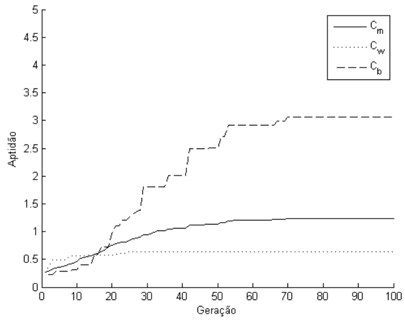
Se formos pensar em uma cadeia genética como o DNA, não faz muito sentido subdividir um determinado cromossomo em mais partes do que seu número de genes, por isso deve-se limitar o número de cortes a $cp = c - 1$. Considerando que nos filtros devemos levar em conta tanto os efeitos do corte e da recombinação do cromossomo quanto a epistasia entre os coeficientes chega-se à conclusão que podemos reduzir as perdas no



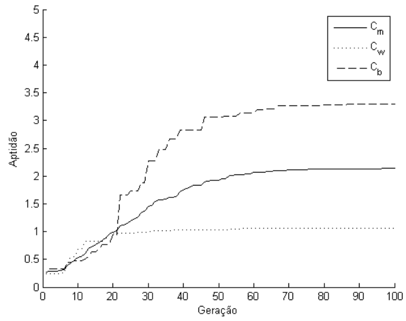
(a) Cruzamento com 1 ponto de corte



(b) Cruzamento com 2 pontos de corte



(c) Cruzamento com 3 pontos de corte

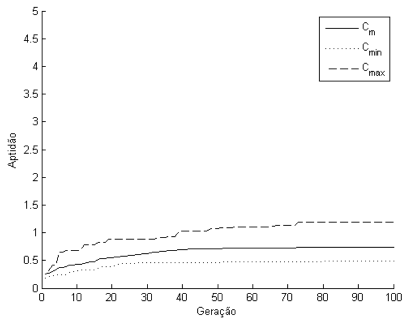


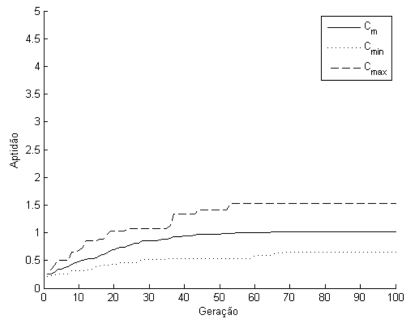
(d) Cruzamento com 10 pontos de corte

Figura 48 Melhor, pior e média das progressões do GA com cruzamento com diversos pontos de corte em filtros de 39 coeficientes

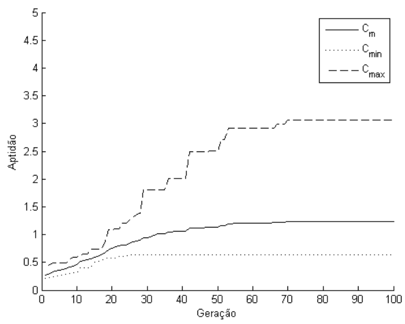
processo de cruzamento se não permitirmos que o corte ocorra no meio de um gene. Uma vez que para a redução dos efeitos da epistasia deveríamos partir da premissa de uma configuração dos coeficientes ideal ou próxima da ideal, na população inicial, o que implicaria no conhecimento prévio do espaço de busca.

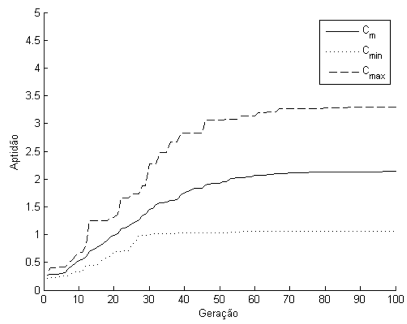
Assim chegamos a outra abordagem para o cruzamento que envolve o reposicionamento dos genes, e logo dos coeficientes do filtro, uma vez que um determinado coeficiente pode não ser adequado em uma determinada posição e sim em uma outra. Este tipo de cruzamento assume que o bloco a ser trocado entre os cromossomos pais equivale exatamente a um gene do cromossomo, ou seja, cada bloco é um coeficiente do filtro. A Figura 52 exemplifica um cruzamento em blocos onde ocorre a quebra da sequência genética ao fim de cada gene dos cromossomos pais seguida de uma recombinação destes genes em ordem aleatória, formando os cromossomos filhos. Esta aleatoriedade na recombinação dá ao algoritmo maior capacidade de busca uma vez que devido à epistasia, a reordenação dos coeficientes influi nos filtros formados e logo há uma inserção de novas informações no





(a) Cruzamento com 1 ponto de corte (b) Cruzamento com 2 pontos de corte



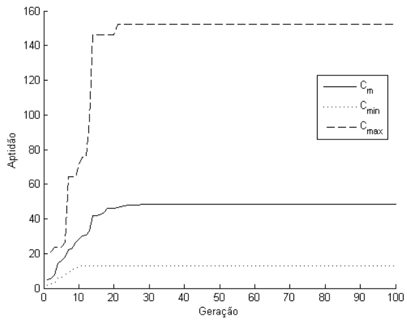


(c) Cruzamento com 3 pontos de corte (d) Cruzamento com 10 pontos de corte

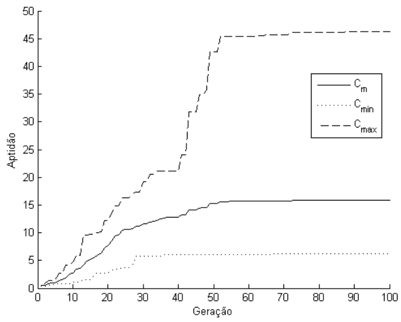
Figura 49 Melhor, pior e média dos melhores indivíduos por teste do GA com cruzamento com diversos pontos de corte em filtros de 39 coeficientes

GA que consistem em informações antigas reordenadas. Por outro lado, esse método pode gerar filtros ruins, principalmente depois que uma boa ordenação dos filtros for encontrada, o que leva as melhorias na aptidão a cessarem uma vez que o algoritmo convergiu parcialmente, ou seja, uma vez que alguns valores de hiperplanos já foram definidos. Por isso deve-se controlar a taxa de mutação neste método, o que permite limitar o efeito ruim deste como será visto na seção 4.5.3.

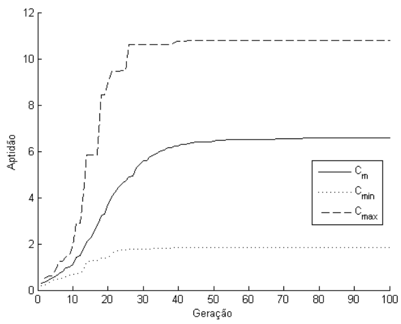
O sistema de cruzamento em blocos proposto neste trabalho tem seu comportamento mostrado na Figura 53. Se comparado com os outros tipos de cruzamento apresentados nota-se a sua superioridade para todas as ordens de filtros testadas. Este resultado se deve em parte à não destruição dos genes durante o processo, o que reduz o efeito nocivo do operador cruzamento apresentado teoricamente na Seção 3.7. Este resultado pode ser também atribuído à maior diversidade de filtros encontrados com a recombinação aleatória dos coeficientes. O cruzamento em blocos ao efetuar a troca entre genes inteiros dos indivíduos selecionados, provoca melhorias utilizando as informações já existentes desde



(a) Filtros com 10 coeficientes



(b) Filtros com 27 coeficientes

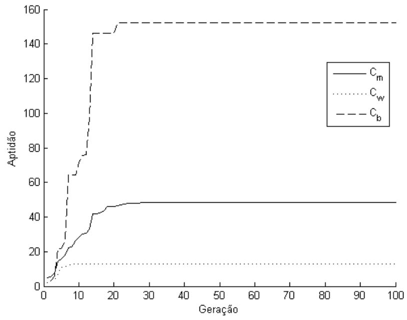


(c) Filtros com 39 coeficientes

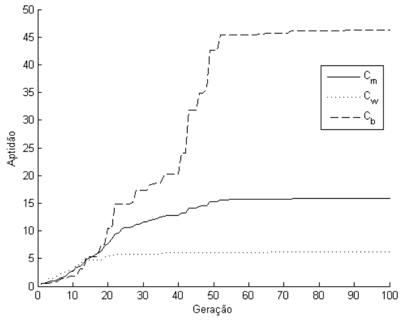
Figura 50 Progressão do melhor e pior indivíduo por geração do GA em filtros de diferentes comprimentos com cruzamento uniforme apenas

o início do GA sem que haja corrompimento desses dados, e por isso está fadada a atingir um patamar máximo caso não exista um meio de inserir novas informações no sistema, função esta atribuída à mutação. A Figura 54 apresenta os resultados dos indivíduos considerando todos os testes efetuados.

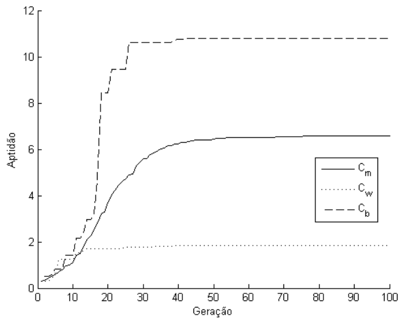
Nota-se que o cruzamento em blocos nos filtros de maiores ordens, que apresentaram progressões ruins nos outros métodos, tem um comportamento médio acima do melhor comportamento dos outros métodos. Esta ocorrência não se repetiu para filtros de baixas ordens. Atribuiu-se este fato à relação entre o tamanho da população inicial e o tamanho dos filtros. Para filtros pequenos a população inicial é uma amostra ampla do espaço de busca e por mais destrutivo que seja o operador cruzamento aplicado, a população consegue manter uma boa diversidade genética, como pode ser visto no caso do cruzamento uniforme. Porém, conforme a ordem dos filtros cresce, a população inicial, se mantida no mesmo tamanho, passa a ser uma amostra cada vez menor do espaço de busca e quando, durante o cruzamento, as informações da população são alteradas estas



(a) Filtros com 10 coeficientes



(b) Filtros com 27 coeficientes



(c) Filtros com 39 coeficientes

Figura 51 Progressões que resultam no melhor e pior indivíduo do GA em filtros de diferentes comprimentos com cruzamento uniforme apenas

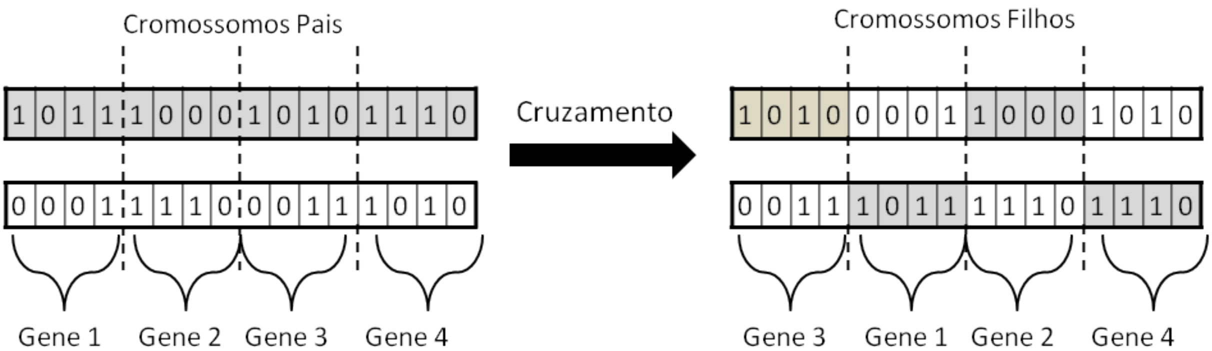
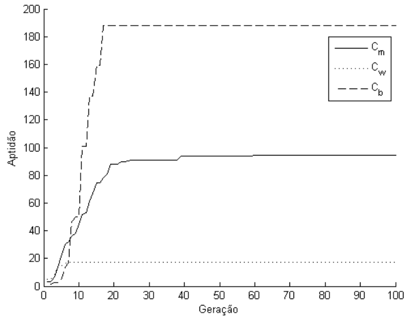
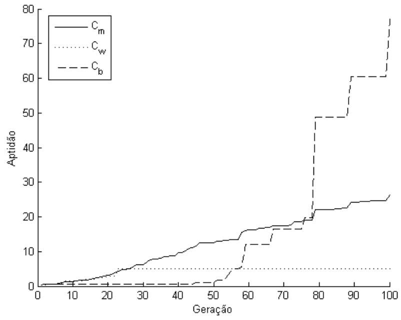


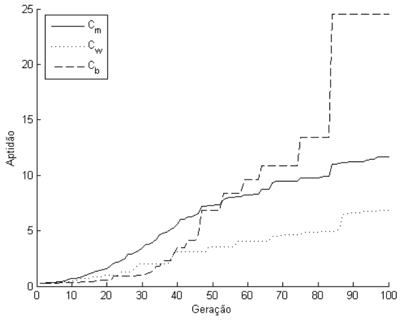
Figura 52 Esquema de cruzamento em blocos



(a) Filtros com 10 coeficientes

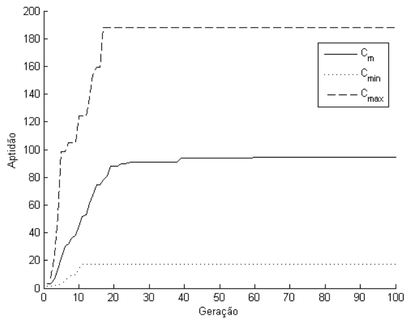


(b) Filtros com 27 coeficientes

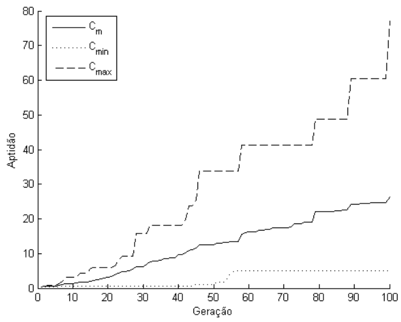


(c) Filtros com 39 coeficientes

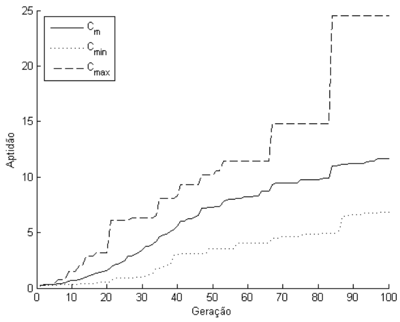
Figura 53 Melhor, pior e média das progressões do GA com cruzamento em blocos



(a) Filtros com 10 coeficientes



(b) Filtros com 27 coeficientes



(c) Filtros com 39 coeficientes

Figura 54 Melhor, pior e média dos melhores indivíduos por teste do GA com cruzamento em blocos podem ser perdidas para sempre, reduzindo a diversidade genética da população, que sem um fator externo não é capaz de reinseri-las. O cruzamento em blocos, por não alterar os valores dos genes, permite a perpetuação durante uma quantidade maior de gerações de características que podem ser benéficas em gerações posteriores de um GA.

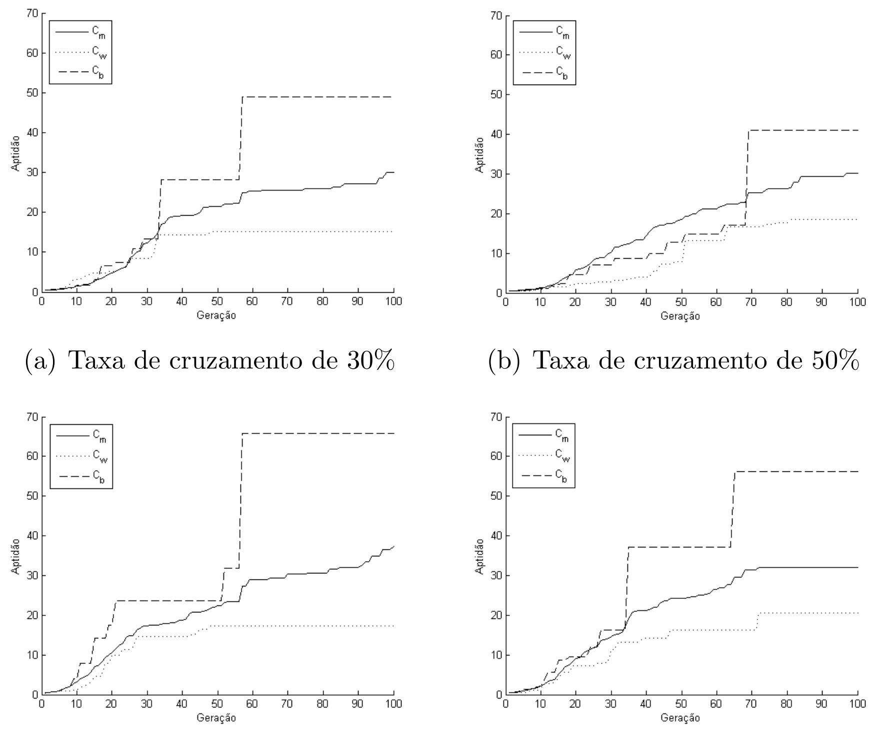
4.5.3 Efeito de diferentes taxas de cruzamento

Uma vez que o tipo de cruzamento foi definido deve-se escolher uma taxa de cruzamento eficaz para o bom funcionamento do algoritmo. Lembrando que a taxa de cruzamento representa a probabilidade de um indivíduo selecionado passar pelo processo de cruzamento. A Figura 55 mostra o efeito que o cruzamento com taxas de 30%, 50%, 70% e 90% tem na evolução do algoritmo com cruzamento em blocos. Percebe-se que conforme a taxa de cruzamento aumenta cresce a aptidão média. Este resultado era de se esperar uma vez que aumentarmos a taxa de cruzamento aumentamos as chances de um indíviduo selecionado passar pelo processo de cruzamento, logo aumentamos as chances de encontrarmos melhores combinações genéticas. Já a Figura 56 apresenta as progressões dos indivíduos ao longo de todos os testes.

Se compararmos os resultados obtidos nesta seção com os resultados da Seção 4.4, onde só trabalhamos com a mutação, veremos que esta última é um operador de busca mais eficiente para projetos de filtros otimizados com GA. Isso ocorre pois de acordo com [58] um GA, utilizando cruzamento apenas, não é um otimizador global. A convergência com recombinação genética não se dá em direção à um pico isolado, mesmo este sendo o máximo global, e sim na direção da região de maior concentração de bons resultados de aptidão e esta depende da população inicial sorteada. Por outro lado, a mutação permite uma busca global, e por isso a seguir faremos uma análise dos dois operadores em conjunto.

4.6 **Cruzamento e Mutação**

Até aqui, trabalhamos com os operadores cruzamento e mutação separadamente para analisarmos os efeitos que ambos têm sobre o projeto de filtros SPT otimizados com GA. Porém, ambos devem ser aplicados simultaneamente para melhorarmos o desempe-



(a) Taxa de cruzamento de 30%

(b) Taxa de cruzamento de 50%

(c) Taxa de cruzamento de 70%

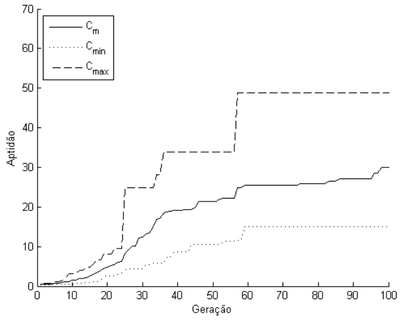
(d) Taxa de cruzamento de 90%

Figura 55 Melhor, pior e média das progressões do GA com cruzamento em blocos com diferentes taxas

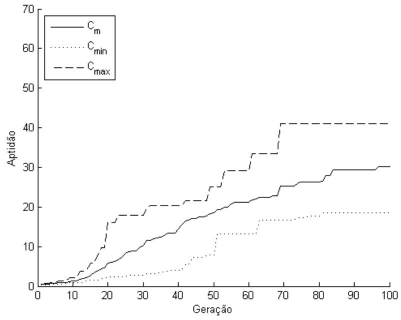
nho do algoritmo e por isso dedicaremos esta seção para a análise da combinação desses operadores.

Existem alguns métodos como os apresentados em [57] para determinar os parâmetros adequados de cada um dos dois operadores assim como formas de combiná-los. Não usaremos esses métodos uma vez que testes exaustivos serão apresentados, a seguir utilizaremos os resultados destes para determinar a melhor combinação de parâmetros.

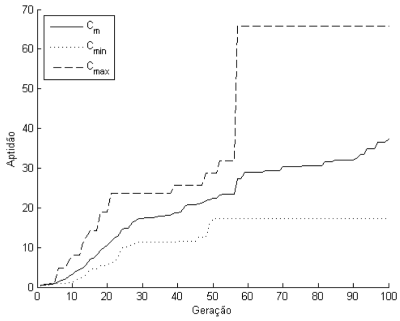
Como vimos o cruzamento e a mutação são operadores complementares durante a varredura do espaço de busca pelo GA. O cruzamento procura por combinações de genes mais eficientes, fazendo com que estes assumam diferentes posições dentro do cromossomo. Por sua vez, a mutação é responsável pela busca de valores mais adequados para os genes, alterando-os aleatoriamente. Quando um indivíduo é selecionado pela função de seleção avalia-se se ele deve ou não sofrer mudanças. Para o cruzamento é feito um sorteio baseado na taxa de cruzamento para determinar se o individuo selecionado sofrerá cruzamento. Em seguida, não importando se o cruzamento ocorreu ou não, sorteia-se quais



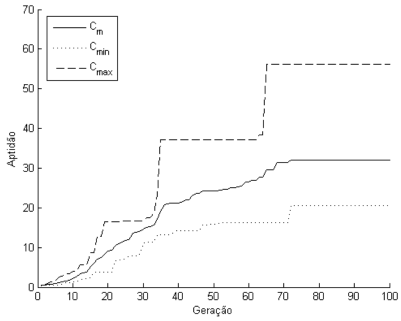
(a) Taxa de cruzamento de 30%



(b) Taxa de cruzamento de 50%



(c) Taxa de cruzamento de 70%



(d) Taxa de cruzamento de 90%

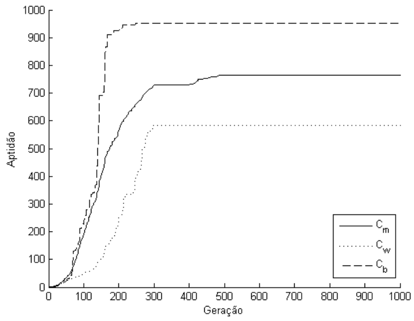
Figura 56 Melhor, pior e média dos melhores indivíduos por teste do GA com cruzamento em blocos com diferentes taxas

dígitos do cromossomo devem ou não sofrer alterações de acordo com a taxa de mutação. Deste modo, podem ocorrer simultaneamente cruzamento e mutação em um determinado indivíduo o que pode acarretar mudanças muito pronunciadas em sua aptidão. Como a probabilidade de um dígito mudar por mutação é calculada para cada dígito do cromossomo, limitá-la torna-se complicado, pois afetaria seu caráter aleatório. Neste ponto, a baixa ocorrência da mutação já é um bom limitador. Por outro lado, cada indivíduo selecionado passa apenas por uma única verificação para determinar se este irá ou não sofrer cruzamento, e por isso caso as alterações conjuntas do cruzamento e mutação sejam muito danosas ao sistema, a primeira taxa de ocorrência a ser alterada deve ser a taxa de cruzamento.

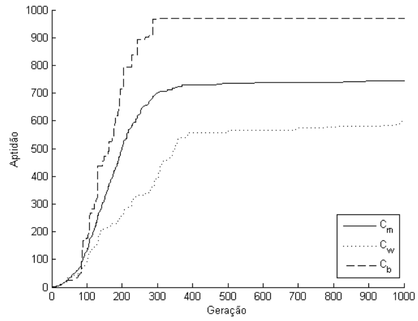
A Figura 57 mostra o comportamento do GA padrão funcionando com mutação em 1% juntamente com cruzamento em blocos com taxas de 30%, 50%, 70% e 90%, as mesmas usadas na Subseção 4.5.3. A Figura 58 mostra a evolução do melhor e do pior indivíduo por geração assim como a média de aptidão por geração. Quando o cruzamento é combinado com a mutação notam-se melhorias consideráveis na aptidão do indivíduo, uma vez que a mutação, ao alterar os indivíduos da população gera indivíduos com novas características, algumas boas, e o cruzamento permite a obtenção de novas combinações de coeficientes ao introduzir novas informações na população. Porém, taxas de cruzamento muito altas combinadas com mutação têm um potencial danoso muito grande e por isso a convergência nesses casos é mais lenta e os valores de aptidão final obtidos não são tão bons quanto os valores obtidos com taxas de cruzamento mais baixas. Quando as taxas de cruzamento são baixas, apenas uma pequena parte da população sofre *crossover* o que permite balancear o efeito do cruzamento com o efeito de mudança aleatória dos valores dos coeficientes proporcionado pela mutação. Como o cromossomo do GA proposto é haplóide, composto por um filete de código genético apenas, não há possibilidades de inibição do efeito de mutações deletérias com o cruzamento. Ao contrário do que ocorre com um cromossomo diplóide, ou outra cadeia genética formada por uma cobinação de dois ou mais filetes genéticos, onde o gene equivalente no filete complementar pode, desde que não tenha sofrido mutações prejudiciais, determinar uma característica, independentemente da informação contida no gene comprometido [20, 33].

Uma vez comparado o comportamento do GA com taxa de mutação fixa, utilizaremos a mutação escolhida para este trabalho, ou seja, a mutação oscilante tipo II vista na Subseção 4.4.3, para avaliar o seu efeito combinado com o cruzamento. A Figura 59 mostra o comportamento do GA quando combinamos a mutação oscilante vista na Subseção 4.4.3, e as diversas taxas de cruzamento utilizadas na Subseção 4.5.3. A Figura 60 apresenta a melhor, pior e a média das aptidões dos indivíduos por geração quando cruzamentos com diferentes taxas são combinados com a mutação oscilante tipo II discutida apresentada na Subseção 4.4.3.

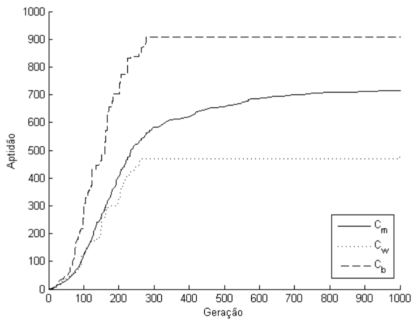
Ao compararmos as progressões com taxas de mutação fixas e progressões com taxas de mutação variáveis, notamos um comportamento semelhante, que consiste em melhores aptidões quando a taxa de cruzamento é baixa. Além disso, percebe-se na Figura 58(a) e na Figura 58(b), que a média converge rapidamente e cessa de apresentar mudanças, enquanto na Figura 60(a) e na Figura 60(b) temos uma pequena variação da



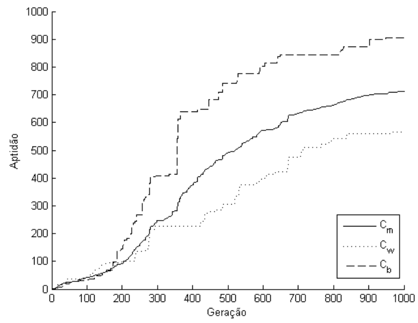
(a) Taxa de cruzamento de 30%



(b) Taxa de cruzamento de 50%



(c) Taxa de cruzamento de 70%



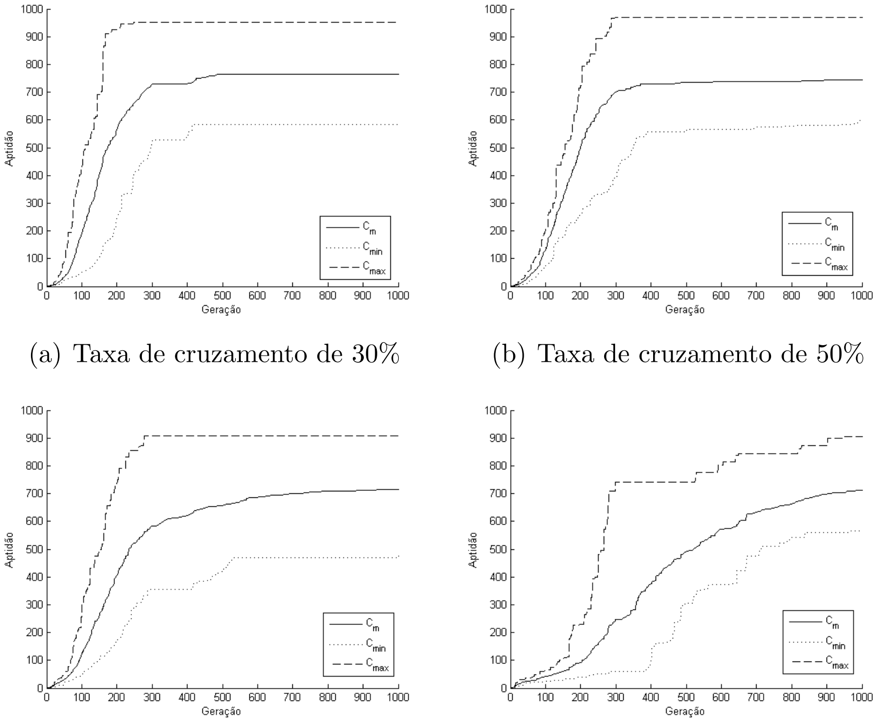
(d) Taxa de cruzamento de 90%

Figura 57 Melhor e pior progressão do GA, assim como a média das progressões com diferentes taxas de cruzamento e com taxa de mutação fixa de 1%

média até o fim do algoritmo. Ambas atingem aptidão média em torno de 760.

4.7 Comparativo entre Tipos de Seleção

O tipo de função de seleção dos indivíduos escolhido influi na intensidade de seleção do GA, e altera significativamente a progressão deste como foi discutido na Seção 3.3. Para avaliar a diferença da escolha da função de avaliação, o GA desenvolvido foi executado com a função de seleção proporcional e com a função de seleção por normalização linear, sob mesmas condições. O método de normalização linear do GA permite que sejam feitos ajustes na probabilidade de escolha do melhor indivíduo e por isso foram feitos testes com $b \in \{5\%, 10\%, 15\%\}$ de chances deste indivíduo ser selecionado. Os resultados obtidos são exibidos na Figura 61 para filtros de 27 coeficientes. Nota-se que, assim como obtido em [47], não há muita diferença entre as progressões do GA com os dois tipos de função de seleção escolhidos para os testes. Em ambos os casos a progressão se dá rapidamente em torno das primeiras trezentas gerações após as quais ocorrem pequenas melhorias



(a) Taxa de cruzamento de 30%   (b) Taxa de cruzamento de 50%

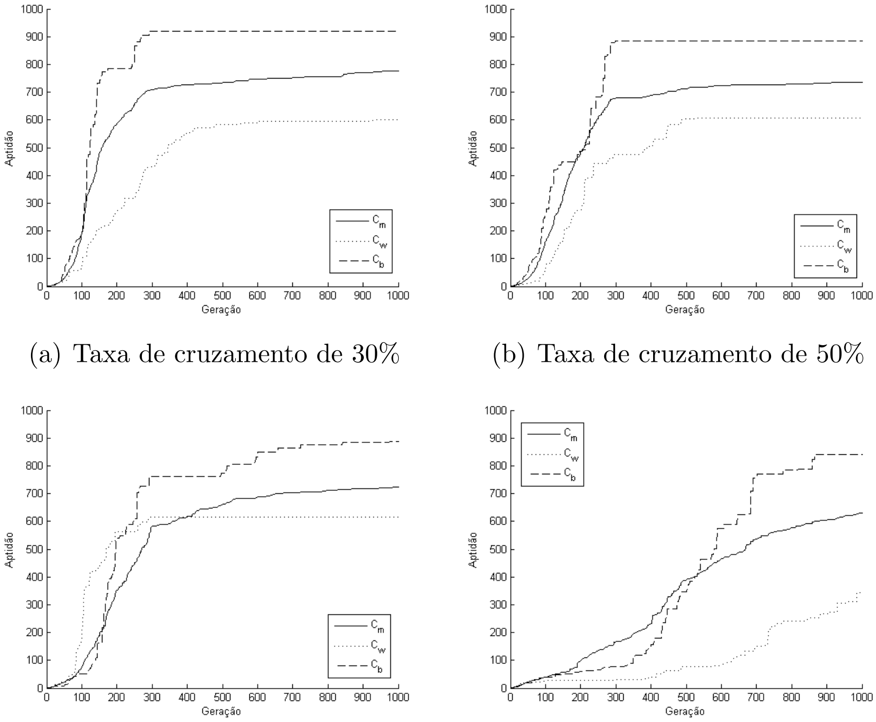
(c) Taxa de cruzamento de 70%   (d) Taxa de cruzamento de 90%

Figura 58 Melhor, pior indivíduo e média dos indivíduos por geração com diferentes taxas de cruzamento e com taxa de mutação fixa de 1% em paralelo

esparsas. A Figura 62 mostra a progressão do melhor indivíduo por geração para as diferentes funções de seleção empregadas.

4.8 Redução do Número de Adições

A grande vantagem da utilização de números representados em SPT é o ganho em processamento, por redução do número de somadores do sistema, que esta representação pode alcançar sobre a representação binária. A cada vez que o GA gera uma população, que neste trabalho vem representada em números reais, ele deve convertê-la para a representação escolhida para poder utilizar os seus operadores de cruzamento e mutação. Assim que ele termina de efetuar as modificações necessárias ele deve reconvertê-la para números reais para poder testar o filtro obtido. É nessas transformações que a escolha da representação agiliza o processo. Cada dígito diferente de zero representa uma multiplicação e uma adição, o que totaliza duas operações aritméticas. A multiplicação é feita entre o dígito não-zero e a potência de dois à ele atribuída, enquanto a adição é feita



(a) Taxa de cruzamento de 30%

(b) Taxa de cruzamento de 50%

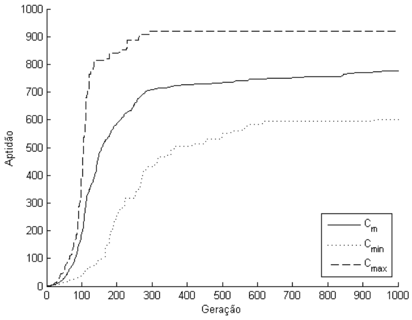
(c) Taxa de cruzamento de 70%

(d) Taxa de cruzamento de 90%

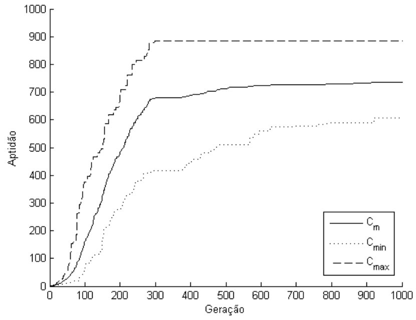
Figura 59 Melhor e pior progressão do GA, assim como a média das progressões com diferentes taxas de cruzamento e mutação oscilante tipo II em paralelo

entre os resultados das multiplicação que compõem um gene. Para garantirmos o bom funcionamento do GA, sem implementações de operadores de cruzamento e mutação para as representações MSD e CSD, vistas na Subseção 2.1.4, devemos garantir a operação destes na representação SPT. Porém, no momento da conversão dos coeficientes de números reais para SPTs, utiliza-se sempre uma representação com o menor número de somas, o que equivale à uma das representações MSD. Deste modo, se formos implementar o filtro obtido ele contemplará o número mínimo de somas. Usando este método não garantimos o número mínimo de somas durante a execução do algoritmo e portanto não há ganho de processamento sobre a representação SPT comum.

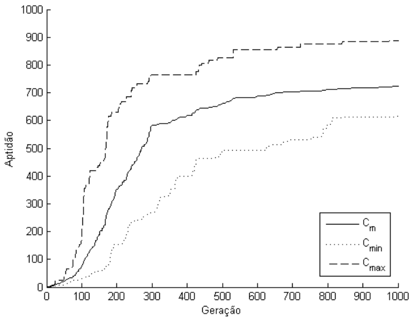
A contagem do número de operações aritméticas nesse trabalho será feita a partir da contagem dos elementos diferentes de zero dos filtros que compõem uma dada população. Esses dígitos não-zero contarão como apenas uma operação. Deste modo, um coeficiente representado em números binários com a configuração $c_b = 1011$ acarreta um total de 3 operações, assim como um coeficiente SPT $c_s = \bar{1}10\bar{1}$. Uma vez somado o total



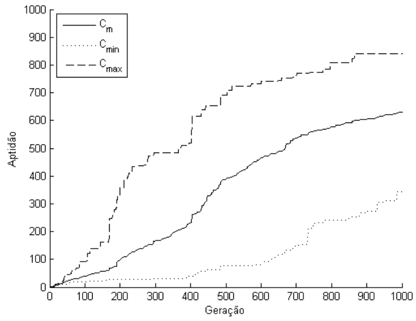
(a) Taxa de cruzamento de 30%



(b) Taxa de cruzamento de 50%



(c) Taxa de cruzamento de 70%



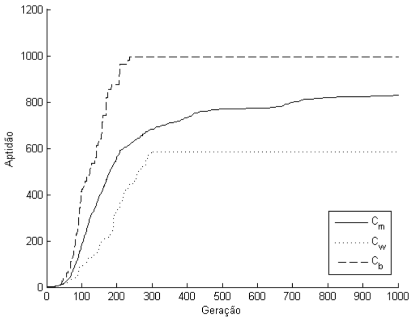
(d) Taxa de cruzamento de 90%

Figura 60 Melhor, pior indivíduo e média dos indivíduos por geração com diferentes taxas de cruzamento e mutação oscilante tipo II em paralelo

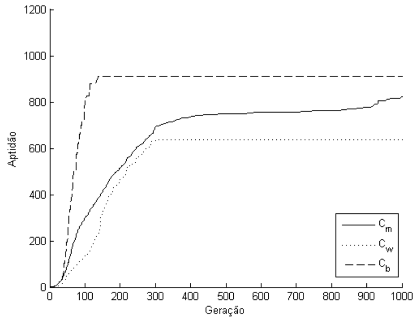
de operações aritméticas durante a execução do GA com representação binária e SPT, estes serão comparados para avaliarmos o ganho médio de processamento da utilização de números SPT neste trabalho. Este ganho em processamento só levará em conta o número de operações realizadas para a representação do coeficiente, ou seja, quanto menor o número de operações menor a carga computacional. Vale lembrar que por não exigirmos formatação CSD ou MSD dos cromossomos SPT, não necessariamente obtemos as representações com mínimo de somas.

O ganho avaliado aqui não implica necessariamente em um menor tempo de execução do GA uma vez que a conversão para binários e a mutação em binários são operações mais simples do que seu equivalente em SPT. A Tabela 8 apresenta a quantidade média de operações, depois de dez execuções do GA para cada representação, de um GA binário e de um GA SPT, ambos com critério de parada em mil gerações, assim como a porcentagem da redução do número de operações.

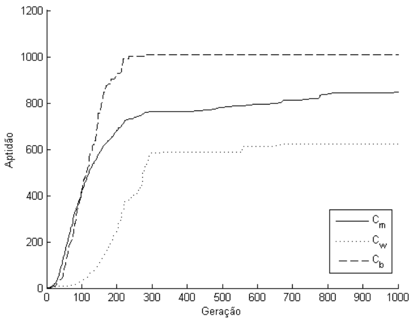
Quando SPTs foram utilizados, não houve preocupação em minimizar o número de



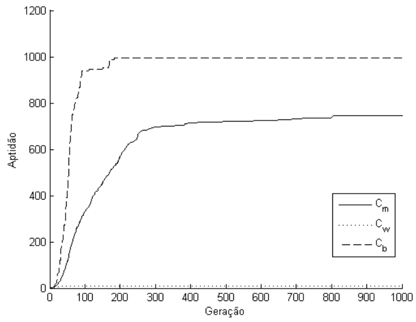
(a) Seleção proporcional



(b) Seleção por normalização linear e $b = 5\%$



(c) Seleção por normalização linear e $b = 10\%$

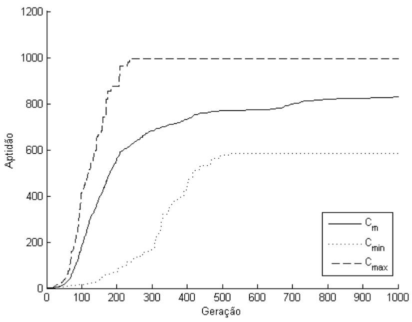


(d) Seleção por normalização linear e $b = 15\%$

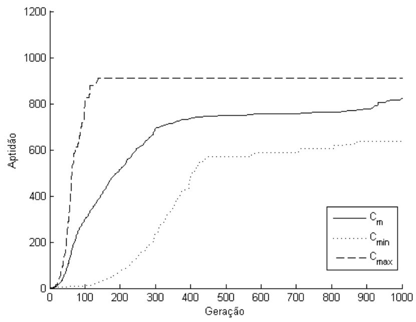
Figura 61 Melhor e pior progressão do GA, assim como a média das progressões obtidas com diferentes funções de seleção

	Binário	**SPT**	**Diferença**	**Redução (%)**
Ordem 10	1694368,2	1259964,7	434403,5	25,63
Ordem 27	4009655,1	2816230,2	1193424,9	29,76
Ordem 39	6020836,4	4534492	1486344,4	24,69

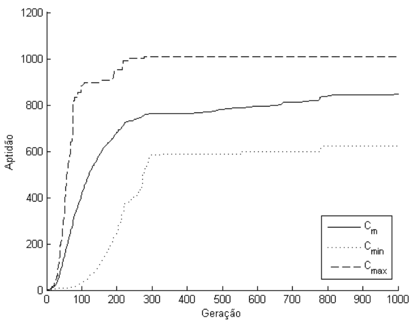
Tabela 8 Relação de número médio de operações aritméticas em um GA com representação binária e SPT



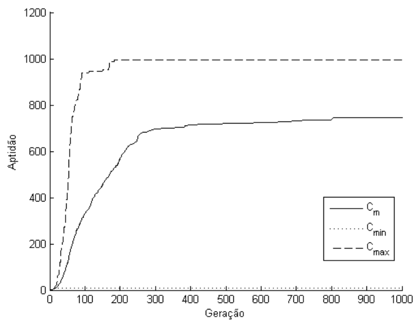
(a) Seleção proporcional



(b) Seleção por normalização linear e $b = 5\%$



(c) Seleção por normalização linear e $b = 10\%$



(d) Seleção por normalização linear e $b = 15\%$

Figura 62 Melhor, pior indivíduo e média dos indivíduos por geração obtidos com diferentes funções de seleção

Parâmetro	Utilizado
Representação Numérica	SPT
Dígitos para Representação	8
Função de Seleção	Seleção Proporcional
População Inicial	Aleatória
Tamanho da População	100 indivíduos
Operador de Cruzamento	Cruzamento em Blocos
Taxa de Cruzamento	30%
Operador Mutação	Mutação Aleatória
Taxa de Mutação	Oscilante com taxas entre 1% e 5%
Critério de parada	1000 gerações decorridas

Tabela 9 Parâmetros selecionados para o GA

somas dos números representados, exceto no momento da conversão de número contínuo para SPT, onde a representação inicial desse valor é realizada de forma a possuir o menor número de operações. A Tabela 8 apresenta a redução de operações inerentes ao uso de uma representação SPT. Nota-se que esta redução não é constante e oscila entre 25% e 30%. Esta oscilação se dá pela natureza aleatória da representação dos números, ao contrário de trabalhos onde representações mínimas ou canônicas são exigidas e nos quais de acordo com [69] a redução do número de operações é da ordem de 33%. Mesmo tendo resultados inferiores a não restrição dos números permite uma redução no número de operações aritméticas próxima às representações onde representações mínimas são impostas. Durante a execução do algoritmo o ganho em velocidade de processamento da otimização via GA depende da representação dos números das populações deste. O motivo pelo qual não optou-se por garantir representação MSD ou CSD, foi para permitir a utilização dos operadores de mutação e cruzamento sem restrições como visto nas Seções 4.4 e 4.5.

4.9 **Parâmetros escolhidos para o GA**

Uma vez discutidos os efeitos dos diversos operadores e parâmetros sobre a progressão do GA, definiremos os que foram escolhidos para as realizações dos projetos de filtros apresentados no Capítulo 5. A Tabela 9 contém a configuração escolhida para o GA.

A representação numérica permaneceu sendo a representação SPT pois esta é um dos objetivos deste trabalho. A função de seleção escolhida foi a seleção proporcional em detrimento da seleção por normalização linear, por elas quase não apresentarem diferenças nas aptidões obtidas. A população inicial foi mantida em cem indivíduos uma vez que este tamanho permite um balanço entre diversidade genética e tempo de processamento aceitável para os critérios deste trabalho. A inicialização desta população foi aleatória, mesmo que inicializações pré-definidas tenham melhor resultado, pois o foco deste trabalho é a obtenção de um método que por si só seja capaz de encontrar bons filtros. A mutação variável escolhida consiste na mutação com taxas oscilantes tipo II, onde para cada indivíduo a ser alterado a taxa de mutação é sorteada, permitindo com isso juntar as boas características das altas e baixas taxas de mutação. A taxa de cruzamento utilizada foi de 30%, pois como visto na Seção 4.6 as baixas taxas de cruzamento potencializam os benefícios dos operadores mutação e cruzamento. O critério de parada foi mantido em mil gerações, mesmo que muitas vezes o algoritmo alcance a convergência muitas gerações antes disso, de forma a tentar garantir que o GA de fato complete sua rotina de busca.

Por fim definimos neste Capítulo os parâmetros que serão usados no GA, logo no Capítulo 5 apresentaremos os filtros gerados com o GA proposto.

5 RESULTADOS

Neste capítulo serão apresentados os resultados obtidos, ao longo do desenvolvimento deste trabalho, para diversas especificações de filtros FIR utilizando o algoritmo proposto no capítulo 4. Estes resultados serão apresentados em profundidade para filtros passa-baixa e de modo superficial para filtros passa-alta, passa-faixa e rejeita-faixa. Deste modo mostraremos a eficácia do método proposto de modo abrangente, uma vez que com pequenos ajustes no algoritmo podemos utilizá-lo para os diversos tipos de filtro. Todos os filtros projetados neste capítulo têm 8 dígitos por coeficientes, não foram testados outras profundidades de dígitos. Para cada filtro produzido neste capítulo fizemos 10 testes com os parâmetros fixados, para avaliar o desempenho médio do algoritmo proposto.

Para comparação foi utilizado o algoritmo de Parks-McClellan explicitado na Subseção 1.3.2 que gera filtros de coeficientes densamente quantizados. Esta quantização com muitos dígitos, que no MATLAB são 64, faz com que ele se aproxime muito do filtro de coeficientes contínuos. Sendo assim, o filtro obtido do método de Parks-McClellan serve de base para saber quão efetivo é nosso filtro. Nos referenciaremos a este filtro por FIRPM. A ordem do filtro escolhido é baseada na ordem do filtro FIRPM para as mesmas especificações, assim temos a possibilidade de fazer um comparativo direto entre as respostas dos dois métodos. Para simplificar, utilizaremos o nome FIRGA para nos referir aos filtros gerados pelo GA apresentado neste trabalho.

Para os quatro tipos de filtro, o que faremos neste capítulo será alterar os pesos da função de avaliação dada pela equação (73), para avaliarmos os melhores resultados. Os pesos utilizados pelo filtro FIRPM serão aplicados ao nosso algoritmo, assim como outras pesagens, escolhidas com intuito de comparação, serão aplicadas. Os pesos do algoritmo FIRPM são representados por $W_{PM} = [W_{pb}, W_{cb}]$, onde W_{pb} é o peso da banda de passagem e W_{cb} é o peso da banda de rejeição, de modo que [24]

$$W_{pb} = \begin{cases} \frac{\delta_{cb}}{\delta_{pb}}, & se \quad \delta_{cb} \geq \delta_{pb} \\ 1, & se \quad \delta_{cb} < \delta_{pb} \end{cases} \text{ e } W_{cb} = \begin{cases} \frac{\delta_{pb}}{\delta_{cb}}, & se \quad \delta_{pb} > \delta_{cb} \\ 1, & se \quad \delta_{pb} \leq \delta_{cb} \end{cases},$$

onde δ_{pb} e δ_{cb} são, respectivamente, as variações máximas nas bandas de passagem e rejeição. O peso do filtro FIRPM não será alterado durante os experimentos para uma mesma especificação, servindo assim de grupo controle para os filtros FIRGA.

A avaliação da qualidade dos filtros será usando: 1) a banda de passagem que será avaliada de acordo com seu ganho médio e a variação máxima do ganho. Como os filtros produzidos neste trabalho utilizam 10% da banda passante para a aplicação do artifício visto na Subseção 4.2.2 este fator prejudicial à avaliação da variação de ganho de pico a pico nesta banda será levado em conta. 2) a banda de rejeição é avalida com base na atenuação mínima obtida. Para a faixa de transição nenhuma avaliação foi feita, pois neste trecho do espectro adimite-se quaquer resultado como foi visto nas seções 4.2.1 e 4.2.2.

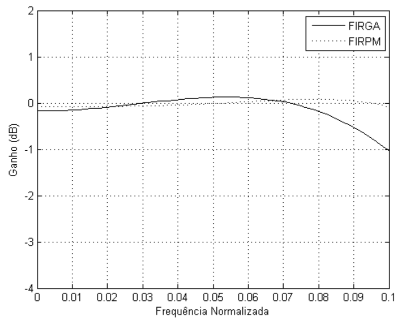
5.1 **Filtro Passa-Baixas**

A escolha das especificações não visam nenhuma aplicação específica. Os filtros passa-baixas escolhidos possuem bandas de passagem e rejeição de tamanhos diferentes assim como diferentes ganhos nestas bandas e, logo, são filtros de ordens distintas.

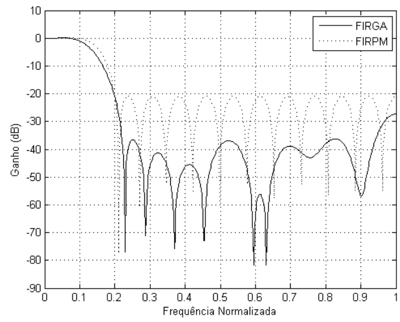
5.1.1 Especificação 1

O primeiro filtro aqui abordado tem ordem 27, ganho na banda de passagem igual a 0dB, variação do ganho na banda de passagem inferior a 0,2dB e atenuação na banda de rejeição de no mínimo -20dB, a banda de passagem é de 0 a 0,1 e banda de rejeição de 0,2 a 1. Para este caso o peso obtido para o filtro FIRPM é $W_{PM} = [10; 1]$.

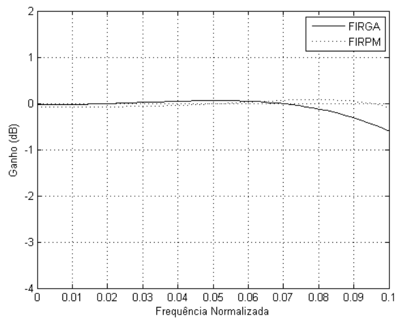
A Figura 63 mostra o comparativo entre a resposta obtida pelo filtro FIRPM e por dois filtros FIRGA quando o mesmo peso $W_{PM} = W_{GA} = [10; 1]$ é utilizado. Nota-se que a bandas passantes dos filtros FIRGA apresentam uma leve ondulação, ou seja, há uma pequena variação no ganho, de aproximadamente 1dB, nesta faixa de frequências que na banda passante do FIRPM é mais suave. No final desta faixa de frequências os filtros FIRGA apresentam uma atenuação mais intensa, entre 0,5dB e 1dB, que no FIRPM só aparece na banda de transição, mas isso ocorre principalmente devido ao artifício explicado na seção 4.2.2 que inclui nas especificações reduções de ganho no final da banda passante. Para o início da banda de rejeição, notamos que os filtros FIRGA possuem uma atenuação próxima à atenuação do filtro FIRPM que é de -20,2dB. Quando o ganho na banda de passagem distancia-se mais das especificações, como pode ser observado na Figura 63(a), temos uma melhor resposta em frequência no início da banda de rejeição, uma vez que neste caso há uma antecipação da queda da curva de ganho, que deveria ocorrer apenas na banda de transição. Porém, quanto mais a curva de ganho mantém-se próxima a 0dB durante a banda de passagem, maior é a distância entre as curvas FIRPM e FIRGA no início da banda de rejeição, a melhor resposta tendendo a ser do filtro FIRPM. Após esse início crítico o filtro FIRGA, que não têm especificações de filtro *equiripple* assume um comportamento com *ripples* variados. Nota-se que a atenuação mínima no filtro FIRGA gira em torno de -30dB, enquanto o filtro FIRPM atinge -22dB.



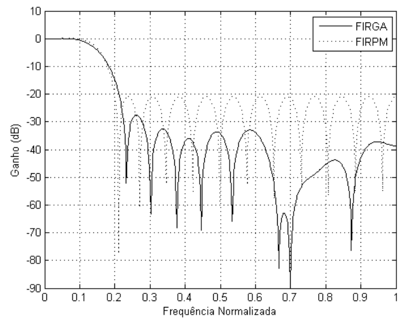
(a) Resposta em frequência na banda passante



(b) Resposta em frequência



(c) Resposta em frequência na banda passante

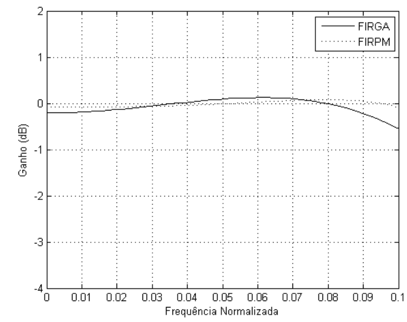


(d) Resposta em frequência

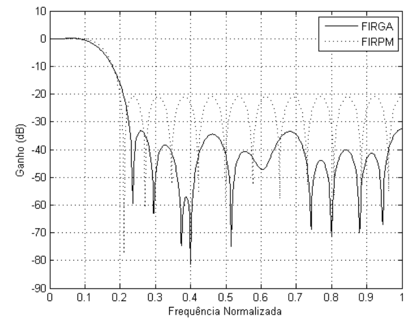
Figura 63 Comparativo entre filtros FIRGA e FIRPM passa-baixas especificação 1 de ordem 27 para pesos $W = [10, 1]$

Aplicamos agora pesos unitários à banda de passagem e à banda de rejeição $W_{GA} = [1; 1]$ para os filtros FIRGA. O resultado obtido está apresentado na Figura 64, onde pode-se notar que as bandas de passagem dos filtro FIRGA tendem a apresentar ondulações de

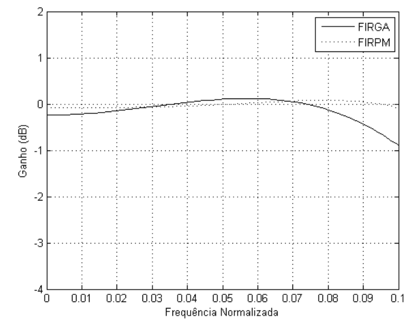
1dB de pico a pico. Este comportamento era de se esperar, uma vez que com a mudança do peso de 10 para 1 na banda de passagem, reduzimos as exigências da reposta em frequência do filtro FIRGA para essa banda. Novamente, no início da banda de rejeição os filtros FIRGA e FIRPM tiveram respostas muito próximas. No restante da banda de rejeição a atenuação mínima do filtro FIRGA, de aproximadamente -35dB, foi mais baixa do que a do filtro FIRPM, que obteve -22dB.



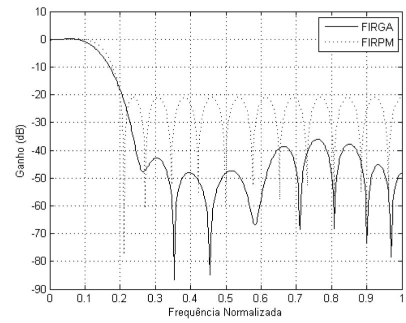
(a) Resposta em frequência na banda passante



(b) Resposta em frequência



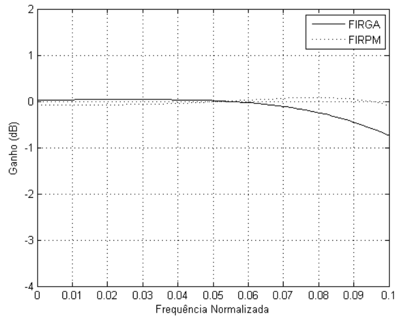
(c) Resposta em frequência na banda passante



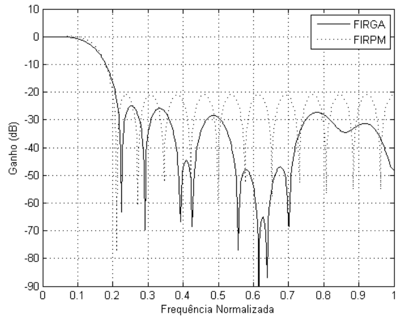
(d) Resposta em frequência

Figura 64 Comparativo entre filtros FIRGA e FIRPM passa-baixas especificação 1 de ordem 27 para pesos $W = [1, 1]$

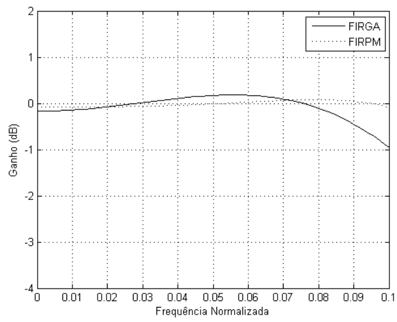
Utilizamos também, com objetivo de testar outros tipos de pesagens, pesos variáveis na banda de rejeição. O peso inicial escolhido para este teste foi $W_{GA} = [10; 1]$ que durante a progressão do algoritmo mudava lentamente até atingir o peso $W_{GA} = [10; 2]$. Obtivemos o resultado mostrado na Figura 65. Esse resultado foi muito semelhante ao do teste com peso $W_{GA} = [10; 1]$, porém usando pesos variáveis conseguimos um melhor decaimento do ganho na banda de transição, o que consequentemente levou a resultados mais competitivos quando comparado com o filtro FIRPM. Em um dos experimentos feitos com filtros FIRGA, conseguimos uma atenuação na banda de rejeição de 21,6dB superior, em toda banda, à atenuação do FIRPM de 20,2dB, em detrimento da banda passante que apresentou ondulações de aproximadamente 1dB de pico à pico.



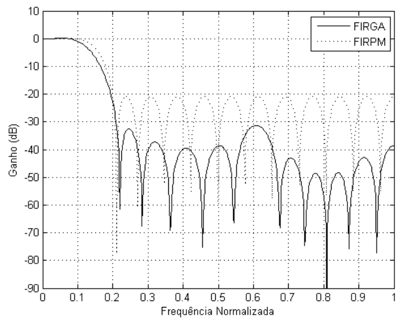
(a) Resposta em frequência na banda passante



(b) Resposta em frequência



(c) Resposta em frequência na banda passante

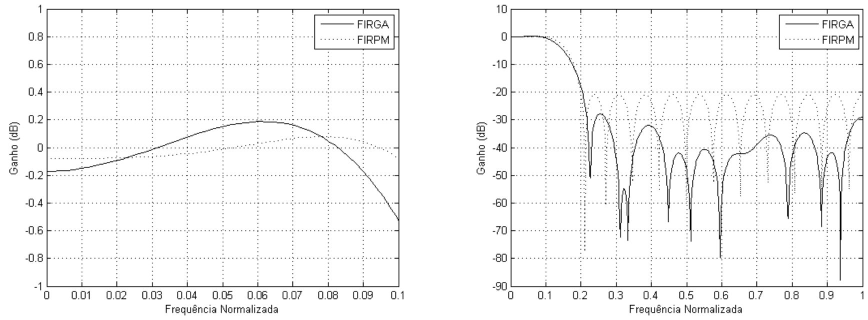


(d) Resposta em frequência

Figura 65 Comparativo entre filtros FIRGA e FIRPM passa-baixas especificação 1 de ordem 27 para pesos $W_{GA} = [10, 1 \rightarrow 2]$

Decidimos inicializar o algoritmo com os coeficientes do filtro FIRPM, em sua população inicial, para testarmos o efeito da inicialização do GA com uma superfície de busca muito reduzida e caracterizada por ser uma região de máximos. Os resultados estão apresentados na Figura 66. Vemos que a variação da resposta em frequência atinge um valor máximo de 0,5dB na banda passante e que a atenuação mínima fica em 18,7dB exatamente no início da banda de rejeição. No restante desta banda, a atenuação ficou acima dos 20dB.

Para testar-se o efeito do do número de dígitos na representação dos coeficientes na resposta em frequência, aumentou-se o número de 8 dígitos para 12. A Figura 67 mostra um comparativo entre um filtro FIRGA com 12 dígitos de representação e o FIRPM,

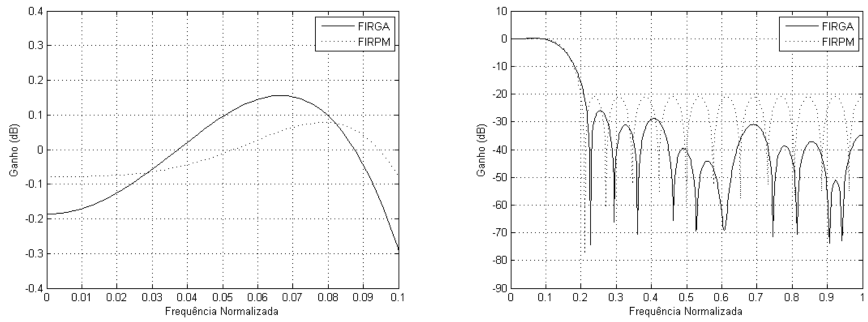


(a) Resposta em frequência na banda passante

(b) Resposta em frequência

Figura 66 Comparativo entre filtros FIRGA, inicializados com coeficientes determinados, e FIRPM passa-baixas especificação 1 de ordem 27 para pesos $W_{GA} = [10, 1]$

ambos com ordem 27. Nota-se que este está bem mais próximo as especificações do que os filtros com 8 dígitos apenas.



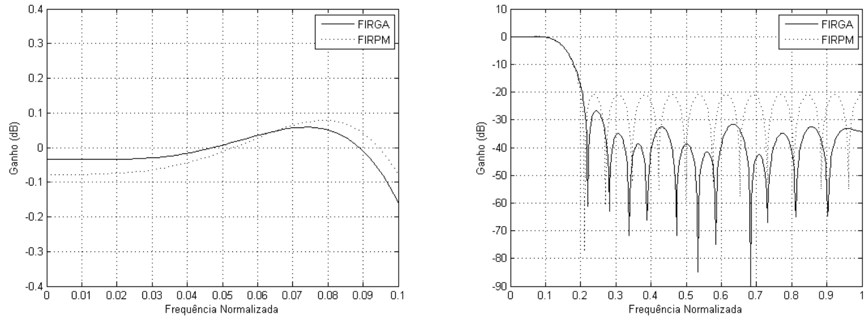
(a) Resposta em frequência na banda passante

(b) Resposta em frequência

Figura 67 Comparativo entre filtros FIRGA, com 12 dígitos por coeficiente, e FIRPM de ordem 27 passa-baixas especificação 1 para pesos $W_{GA} = [1, 1]$

Aumentou-se a ordem do filtro até que este cumprisse integralmente as especificações de projeto, uma vez que um filtro que não atende às suas especificações não é útil. Para filtros passa-baixas com as especificações anteriormente determinadas o menor filtro que atende às especificações é um filtro de ordem 31 que é mostrado na Figura 68.

O algoritmo desenvolvido conseguiu gerar filtros passa-baixas em todas as tentativas, ou seja, o algoritmo está bem otimizado para a geração de filtros com esta especificação. Quando comparados comparados com filtros FIRPM de mesma ordem os resultados dos filtros FIRGA deixaram a desejar um pouco na resposta em frequência da banda



(a) Resposta em frequência na banda passante

(b) Resposta em frequência

Figura 68 Comparativo entre filtros FIRGA de ordem 31 e FIRPM de ordem 27 passa-baixas especificação 1 para pesos $W_{GA} = [1, 1]$

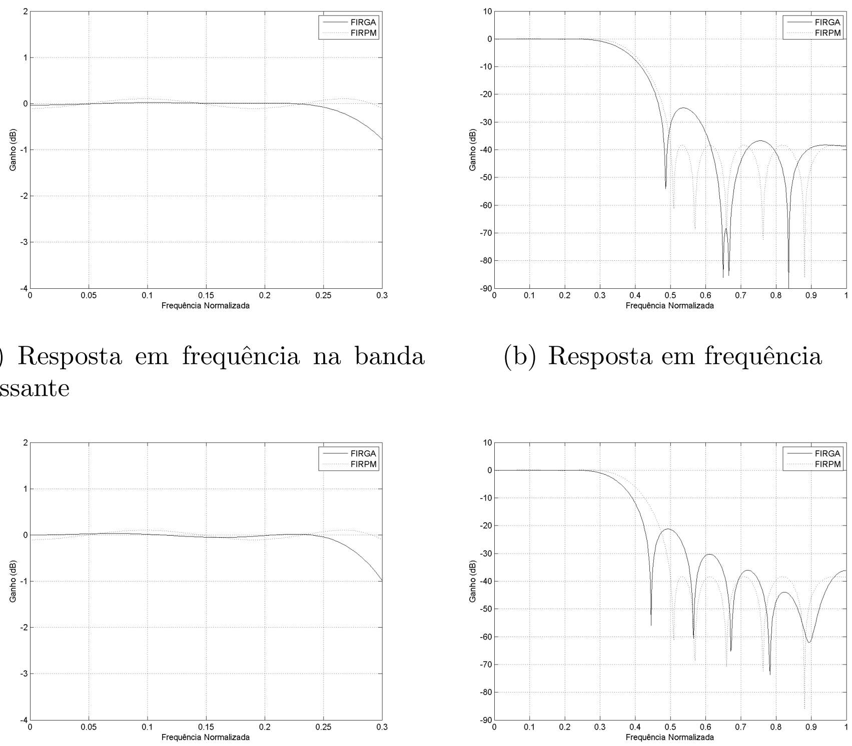
passante, apresentando variações de até 1dB. As variações no final da banda de passagem são devidas à utilização do artifício visto na Subseção 4.2.2. Na banda de rejeição, temos como ponto crítico o início desta, onde normalmente os filtros, apresentaram baixa atenuação. No restante dessa banda a resposta dos filtros FIRGA foi superior à resposta dos filtros FIRPM. Durante o projeto destes filtros não houve ocorrência de convergência prematura, e atribui-se este fato à otimização do GA para filtros com estas especificação. O filtro FIRGA de ordem 31 atende perfeitamente às especificações.

5.1.2 Especificação 2

Para comprovar o funcionamento do algoritmo de otimização do GA, foi executado outro teste com as seguintes especificações: ganhos na banda de passagem e de rejeição respectivamente iguais à 0dB e -38dB e banda de passagem de 0 a 0.3 e banda de rejeição de 0.5 a 1. A variação máxima da banda de passagem imposta foi de 0,2dB. O filtro FIRPM encontrado para estas especificações tem ordem $N = 19$ e pesos $W_{PM} = [1; 1]$.

Aplicamos os mesmos pesos ao nosso GA o que resultou nos filtros com as respostas mostradas na Figura 69. Na banda passante, os filtros FIRGA apresentaram pouca variação em sua resposta em frequência ficando razoavelmente próximas FIRPM, com exceção do final desta faixa que apresenta a queda explicada na Subseção 5.1.1. Na faixa de transição nota-se que os filtros FIRGA possuem uma atenuação, nas frequências iniciais dessa banda, em torno de -25dB enquanto o filtro FIRPM tem aproximadamente -38dB.

A superioridade de atenuação do filtro FIRPM é notória em toda a banda de rejeição. Para estas especificações o filtro FIRGA é adequado somente se uma atenuação de -20dB na banda de rejeição for aceitável para a aplicação, uma vez que na banda de passagem temos variações de pico a pico inferiores à 0,1dB.



(a) Resposta em frequência na banda passante

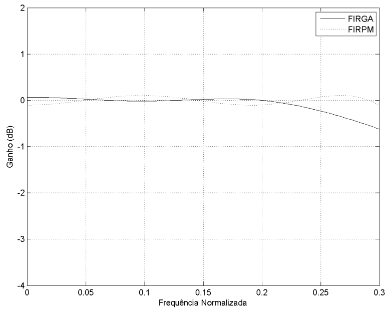
(b) Resposta em frequência

(c) Resposta em frequência na banda passante

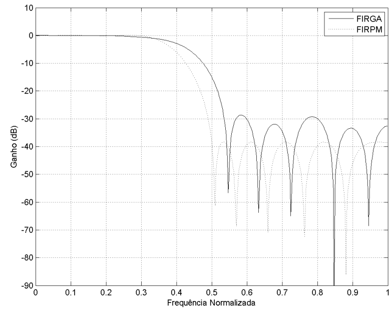
(d) Resposta em frequência

Figura 69 Comparativo entre filtros FIRGA e FIRPM passa-baixas especificação 2 de ordem 19 para pesos $W_{GA} = [1, 1]$

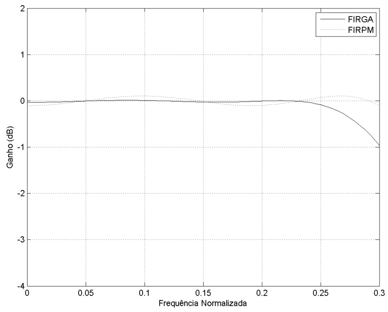
Testamos também o peso $W_{GA} = [10; 1]$ e obtivemos o resultado da Figura 70. Nela vemos resultados melhores e piores do que os obtidos com peso $W_{GA} = [1; 1]$. De um lado, algumas das respostas obtidas apresentaram uma banda de transição com decaimento lento, o que provocou uma atenuação inicial na banda de rejeição em torno de -15dB como pode ser observado na Figura 70(b). Por outro lado, alguns resultados apresentaram um bom comportamento atingindo uma atenuação mínima da ordem de -30dB, conforme Figura 70(d). Os *ripples* nesta configuração foram bem homogêneos. Na banda passante o comportamento foi satisfatório, com variação de 1dB no ganho. Esta variação ocorreu principalmente no final da banda passante, pelos motivos anteriormente comentados.



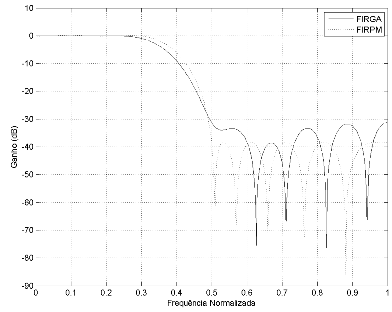
(a) Resposta em frequência na banda passante



(b) Resposta em frequência



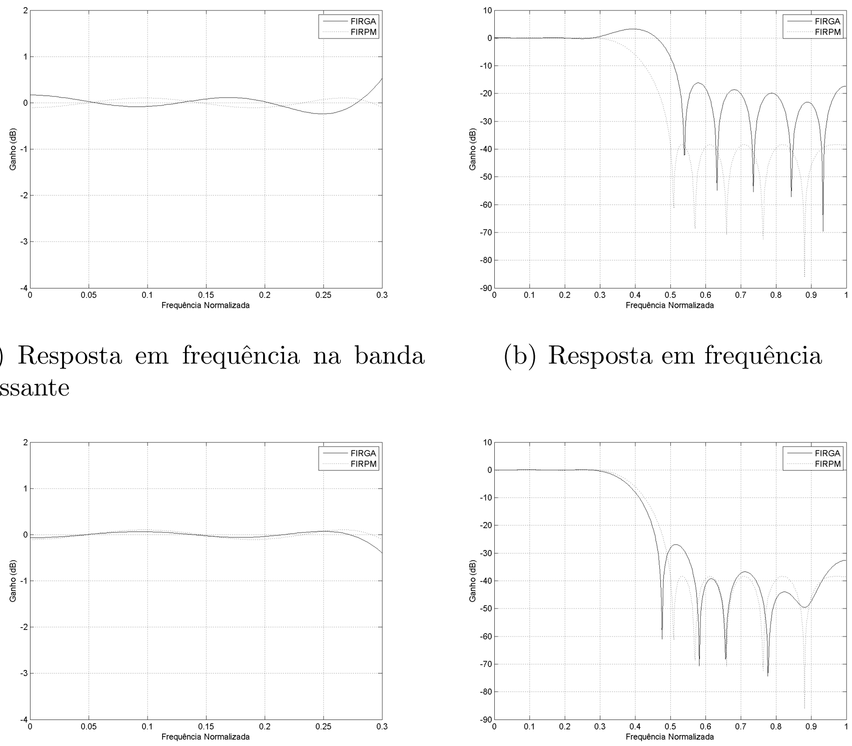
(c) Resposta em frequência na banda passante



(d) Resposta em frequência

Figura 70 Comparativo entre filtros FIRGA e FIRPM passa-baixas especificação 2 de ordem 19 para pesos $W_{GA} = [10, 1]$

Uma vez visto que o artifício, apresentado na subseção 4.2.2, parece prejudicar-nos na obtenção de filtros com comportamento dentro das especificações para a banda de passagem, foi executado um outro teste sem sua utilização, para peso $W_{GA} = [10; 1]$. A Figura 71 apresenta as respostas em frequência de filtro que não utilizam o artifício. Na maioria dos casos os filtros obtidos foram superiores aos filtros com artifício se considerarmos apenas as bandas de passagem. Eles apresentaram variação do ganho de pico-à-pico abaixo de 1dB para a banda passante. Porém, o melhor dos filtros obtidos neste caso apresenta ganho mínimo na banda de rejeição de aproximadamente -27dB contra -32dB da outra configuração. Durante o projeto de filtros foram contados os casos que difeриam muito da progressão do FIRPM, estes casos abrangem as convergências prematuras e comportamentos pouco esperados na banda de transição. Em 30% dos casos ocorreu um crescimento do ganho no final da banda passante que pode ser visto em detalhes na Figura 71(a). Este comportamento foi prejudicial aos filtros gerados desta forma, uma vez que reduz a atenuação total obtida na banda de transição, o que implicou em uma banda de rejeição com baixa atenuação. Sendo assim, concluímos que mesmo não sendo necessário, o artifício ajuda na formação de bons filtros.



(a) Resposta em frequência na banda passante

(b) Resposta em frequência

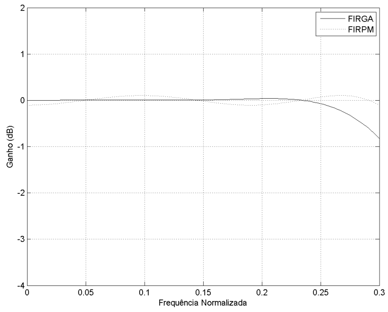
(c) Resposta em frequência na banda passante

(d) Resposta em frequência

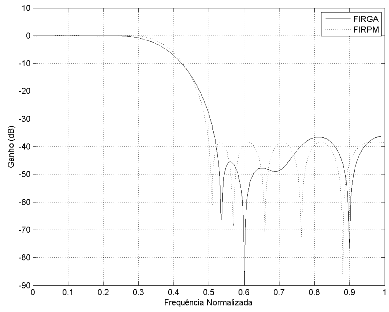
Figura 71 Comparativo entre filtros FIRGA e FIRPM passa-baixas especificação 2 de ordem 19 para pesos $W_{GA} = [10, 1]$ sem artifício

O experimento com peso $W_{GA} = [10; 1]$ com artifício apresentou uma mistura de bons e maus resultados, resolvemos aumentar o peso na banda passante e para conferir se de fato o comportamento se repetiria, ou se este tenderia a produzir uma quantidade maior de bons filtros ou filtros ruins. O resultado deste teste está exibido na Figura 72, e nela vemos que o comportamento apresentado para filtro FIRGA com peso $W_{GA} = [10; 1]$ se repete. Como era de se esperar a banda passante apresentou variações menores em seu ganho.

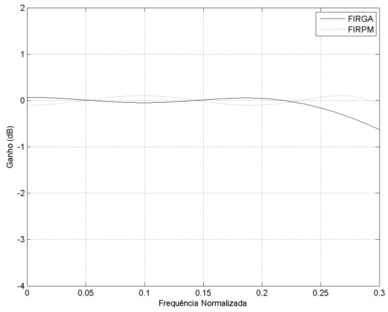
Por fim, aplicamos a pesagem variável que havíamos apresentado na seção 5.1.1 utilizando o peso $W_{GA} = [10, 1 \rightarrow 2]$. Seu resultado é mostrado na Figura 73. Esta configuração apresentou baixa variação na banda de passagem, e atenuação mínima alta



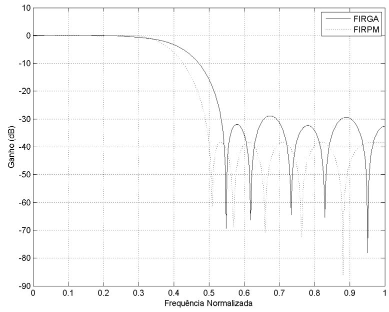
(a) Resposta em frequência na banda passante



(b) Resposta em frequência



(c) Resposta em frequência na banda passante



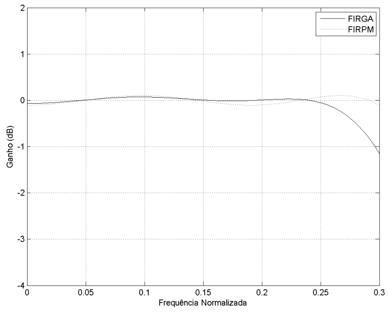
(d) Resposta em frequência

Figura 72 Comparativo entre filtros FIRGA e FIRPM passa-baixas especificação 2 de ordem 19 para pesos $W_{GA} = [15, 1]$

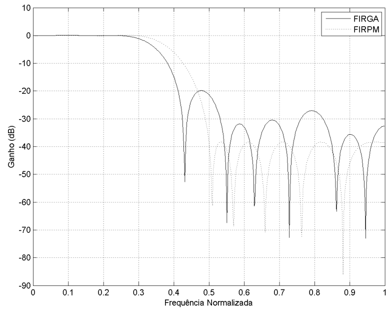
por volta de -20dB.

Inserindo na população inicial do GA os valores dos coeficientes do filtro FIRPM obtemos as respostas em frequência exibidas na Figura 74. Notamos uma banda de passagem dentro das especificações. A banda de rejeição atingiu a atenuação mínima de -38dB, limite das especificações, além disso uma análise do conjunto da faixa de rejeição mostra que o FIRGA tem picos de atenuação mais baixos do que o FIRPM. Portanto o filtro FIRGA, obtido a partir dos filtros FIRPM, é superior ao filtro FIRPM para estas especificações.

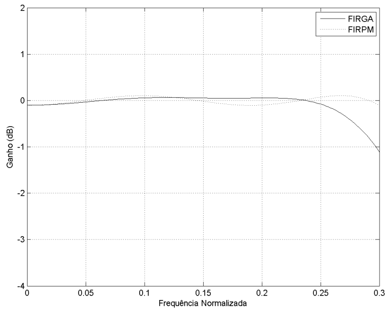
Aumentando-se o tamanho do filtro até que este cumpra as especificações obtemos um filtro FIRGA passa-baixas de ordem 25 cuja resposta em frequência é exibida na Figura 75. Enquanto para a especificação 1 houve um aumento de 4 unidades na ordem, neste caso tivemos um aumento de 6 unidades. Veremos que de todos os casos este foi o que precisou de maior aumento na ordem para que as especificações fossem atendidas.



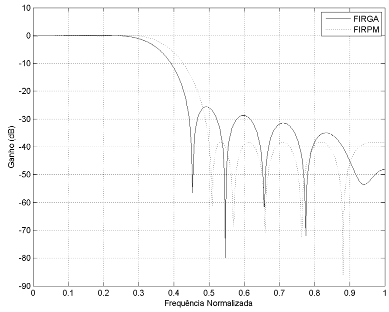
(a) Resposta em frequência na banda passante



(b) Resposta em frequência

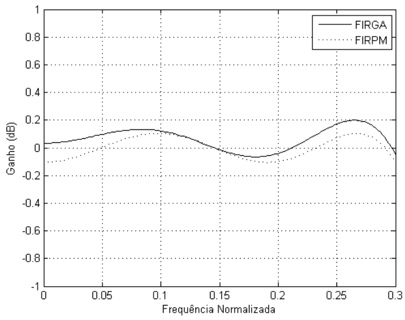


(c) Resposta em frequência na banda passante

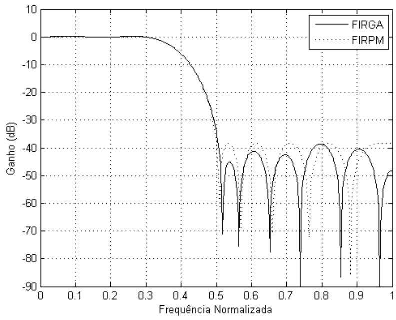


(d) Resposta em frequência

Figura 73 Comparativo entre filtros FIRGA e FIRPM passa-baixas especificação 2 de ordem 19 para pesos $W_{GA} = [10, 1 \rightarrow 2]$



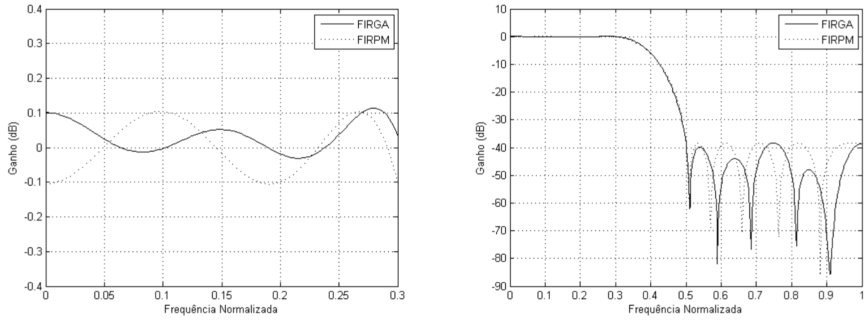
(a) Resposta em frequência na banda passante



(b) Resposta em frequência

Figura 74 Comparativo entre filtros FIRGA, inicializados com coeficientes determinados, e FIRPM passa-baixas especificação 2 de ordem 19 para pesos $W_{GA} = [1, 1]$

De modo geral, esse filtro apresentou comportamento na banda de passagem um pouco acima das especificações. Esta diferença é bem pequena e poderia ser desconsiderada se não fosse pela baixa atenuação na banda de rejeição que se provou um empecilho para a criação de filtros com estas especificações para a profundidade de dígitos e ordem



(a) Resposta em frequência na banda passante

(b) Resposta em frequência

Figura 75 Comparativo entre filtros FIRGA de ordem 25 e FIRPM de ordem 19 passa-baixas especificação 2 para pesos $W_{GA} = [1, 1]$

empregada. Com estas especificações 10% dos experimetos apresentaram convergência prematura.

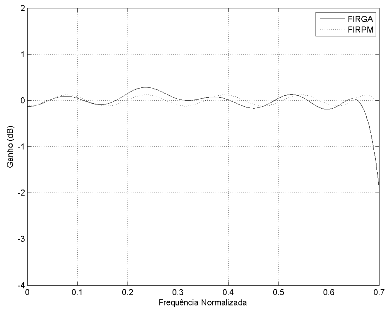
5.1.3 Especificação 3

Esta especificação de filtro passa-baixa consiste em ganho na banda de passagem de 0dB com variação máxima de 0,2dB e atenuação na banda de rejeição de -38dB, banda de passagem de 0 à 0.7 e banda de rejeição de 0.85 à 1.

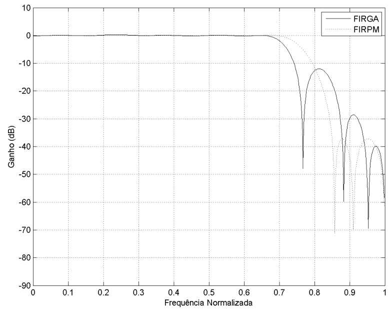
O algoritmo de geração do filtro FIRPM utiliza o peso $W_{PM} = [1; 1]$ e gera um filtro de ordem 27. Com este mesmo peso aplicado nos filtros FIRGA obtemos os resultados da Figura 76. Nota-se que para esta especificação e pesos, o GA produz filtros com variação de 0,3dB na faixa de passagem, desconsiderando os últimos valores desta banda, e atenuação mínima entre -15dB e -20dB. A variação na faixa de passagem do FIRGA é próxima à variação do filtro FIRPM, porém a atenuação do FIRGA é insuficiente. Portanto os filtros gerados com esses pesos são ruins.

Se utilizarmos $W_{GA} = [10; 1]$ conseguimos gerar filtros com as respostas em frequência exibidas na Figura 77. Percebe-se que produzimos filtros com ganho na banda de passagem pouco variável. Em alguns filtros notou-se uma variação acima de 1dB de pico a pico no fim da banda passante. Na banda de rejeição tivemos algumas repostas em frequência com atenuação baixa, da ordem de -10dB.

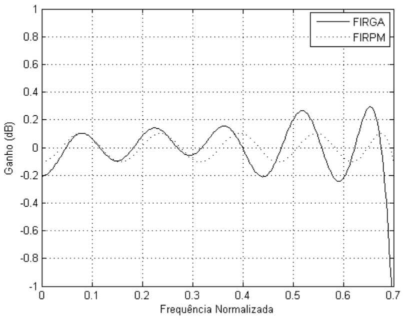
Foram feitos experimentos introduzindo os coeficientes do filtros FIRPM na popu-



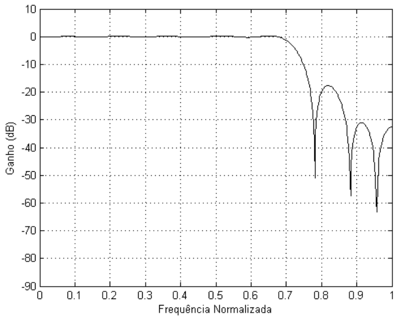
(a) Resposta em frequência na banda passante



(b) Resposta em frequência



(c) Resposta em frequência na banda passante



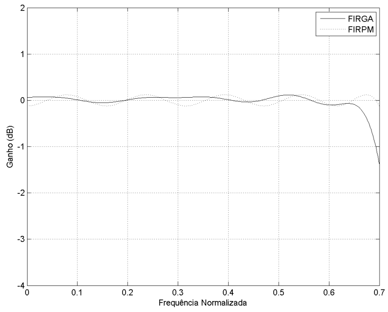
(d) Resposta em frequência

Figura 76 Comparativo entre filtros FIRGA e FIRPM passa-baixas especificação 3 de ordem 27 para pesos $W_{GA} = [1, 1]$

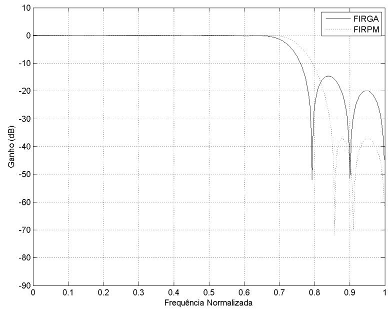
lação inicial dos GA para o projeto de filtros com estas especificações. A Figura 78 nos mostra os resultados para este caso. Vemos novamente que uma vez inicializado com uma boa população o GA produz um muito bom filtro dentro das especificações.

Para estas especificações, 31 é a menor ordem para a qual consegue-se um filtro dentro das especificações de projeto. A figura Figura 79 mostra a resposta em frequência para tal filtro.

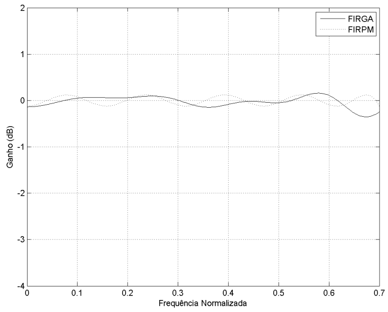
Estes testes com diferentes especificações mostram que a otimização de filtros com GA funciona para filtros passa-baixa. As respostas em frequência dos filtros FIRGA deixam a desejar um pouco quando comparadas a filtros FIRPM, mas isso era de se esperar uma vez que este último tem precisão infinita enquanto os filtros FIRGA têm 8 dígitos de precisão. Mesmo assim, em alguns casos, os filtros FIRGA se aproximaram em muito do filtro FIRPM para as mesmas especificações. Notou-se que, para as especificações 2 e 3, houve criação de uma parcela de filtros ruins, devido à convergência prematura do GA,



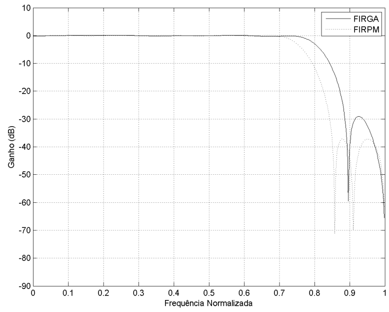
(a) Resposta em frequência na banda passante



(b) Resposta em frequência

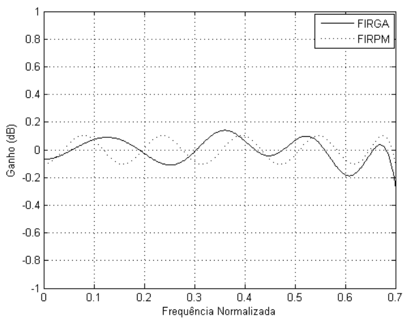


(c) Resposta em frequência na banda passante

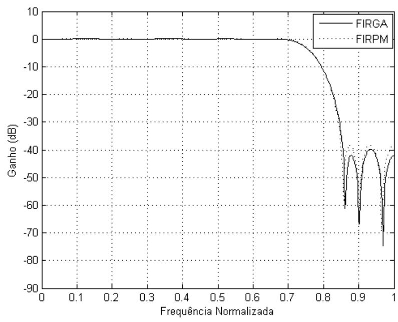


(d) Resposta em frequência

Figura 77 Comparativo entre filtros FIRGA e FIRPM passa-baixas especificação 3 de ordem 27 para pesos $W_{GA} = [10, 1]$



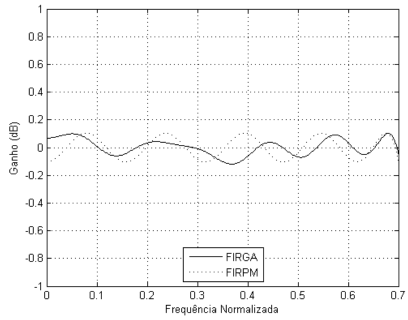
(a) Resposta em frequência na banda passante



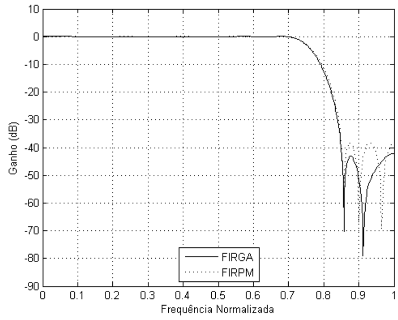
(b) Resposta em frequência

Figura 78 Comparativo entre filtros FIRGA, inicializados com coeficientes determinados, e FIRPM passa-baixas especificação 3 de ordem 27 para pesos $W_{GA} = [1, 1]$

isto foi justificado pelo fato do GA ter sido projetado e otimizado, ao longo do Capítulo 4, para a criação de filtros passa-baixa com a especificação 1.



(a) Resposta em frequência na banda passante



(b) Resposta em frequência

Figura 79 Comparativo entre filtros FIRGA de ordem 31 e FIRPM de ordem 27 passa-baixas especificação 3 para pesos $W_{GA} = [1, 1]$

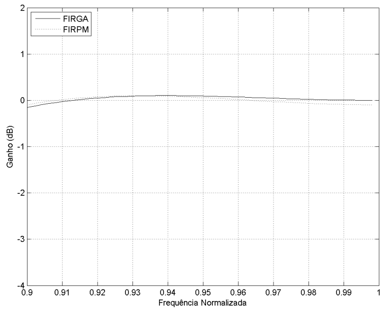
5.2 Filtro Passa-Altas

As especificações escolhidas para o filtro passa-altas foram banda de rejeição de 0 a 0.7 com atenuação mínima de -40db e banda de passagem de 0.9 até 1, esta última com ganho de 0dB e variação máxima deste ganho de 0,2dB. O peso do filtro FIRPM neste caso é de $W_{PM} = [1; 1]$.

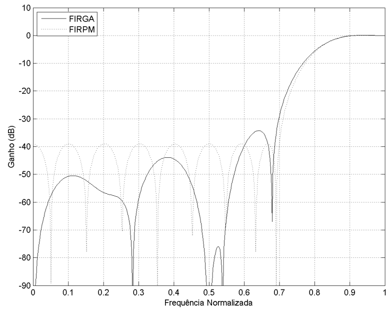
Usando o mesmo peso dos filtros FIRPM para filtros FIRGA ($W_{GA} = [1; 1]$) temos os resultados mostrados na Figura 80. Nestes filtros notamos que o ponto crítico está localizado no final da banda de corte, quando a resposta em frequência começa a aumentar para atingir os níveis de ganho requeridos para a banda de passagem. Percebe-se que os filtros FIRGA criados são razoáveis, tendo atenuação em torno de -30dB na banda de rejeição e baixa variação do ganho na banda de passagem, este ficou abaixo dos 0,2dB da especificação. Infelizmente não cumprem com a especificação de atenuação.

Os filtros passa-altas obtidos para estas especificações e peso $W_{GA} = [10, 1]$ têm respostas em frequência exibidas na Figura 81, que são semelhantes às da Figura 80. Em um dos casos a atenuação do filtro FIRGA foi na maioria da banda de rejeição inferior à atenuação do filtro FIRPM, no trecho em que não foi este apresentou -37dB de atenuação contra -39dB do filtro FIRPM. Por outro lado, este filtro possui uma variação acima de 1dB em seu ganho na banda de passagem.

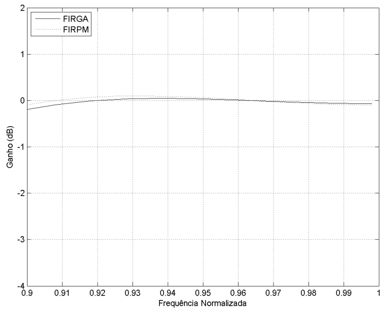
Notamos neste teste, que o filtro FIRPM não atinge as especificações, ficando ligei-



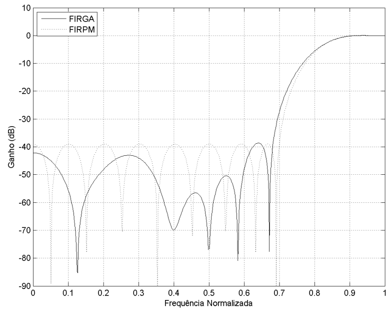
(a) Resposta em frequência na banda passante



(b) Resposta em frequência



(c) Resposta em frequência na banda passante



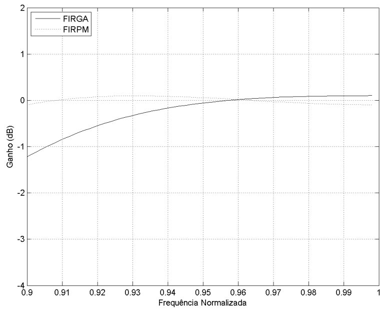
(d) Resposta em frequência

Figura 80 Comparativo entre filtros FIRGA e FIRPM passa-altas de ordem 21 para pesos $W_{GA} = [1, 1]$

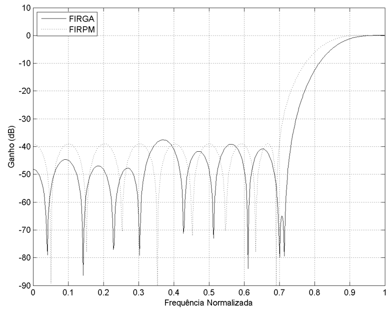
ramente abaixo delas quando levamos em conta a atenuação da faixa de rejeição, atingindo -39dB contra os -40dB das especificações. Este é um fator conhecido no algoritmo FIRPM, e a indicação nestes casos é o aumento da ordem do filtro em 1 ou 2 unidades. De fato ao aumentarmos a ordem em duas unidades obtemos filtros dentro das especificações. Para continuarmos efetuando as comparações em condições equitáveis aumentaremos a ordem do nosso filtro também.

Ao criarmos filtros passa-alta de ordem 23 com pesos $W_{GA} = [1; 1]$ não conseguimos boas respostas dentro da faixa de passagem como vemos na Figura 82. O ganho nesta banda teve variações de até 3dB de pico a pico. Já na banda de rejeição conseguimos atenuações abaixo de -30dB. Em um dos casos a resposta em frequência, exibida na Figura 82(d), apresenta na sua banda de rejeição atenuação abaixo de -43dB, enquanto o filtro FIRPM apresentou -41dB, porém a variação na banda de passagem alcançou 3dB.

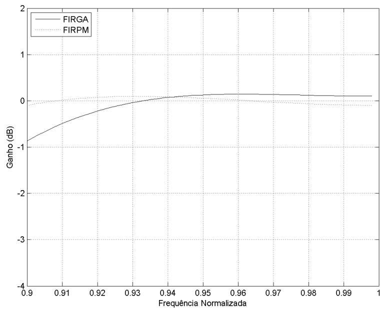
Utilizando os coeficientes do filtro FIRPM no GA obtemos o filtro da Figura 83.



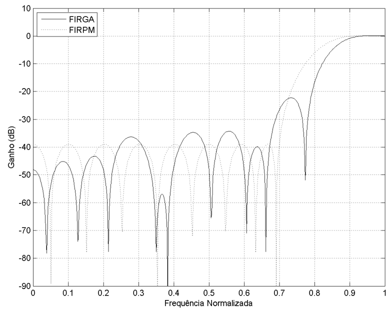
(a) Resposta em frequência na banda passante



(b) Resposta em frequência



(c) Resposta em frequência na banda passante



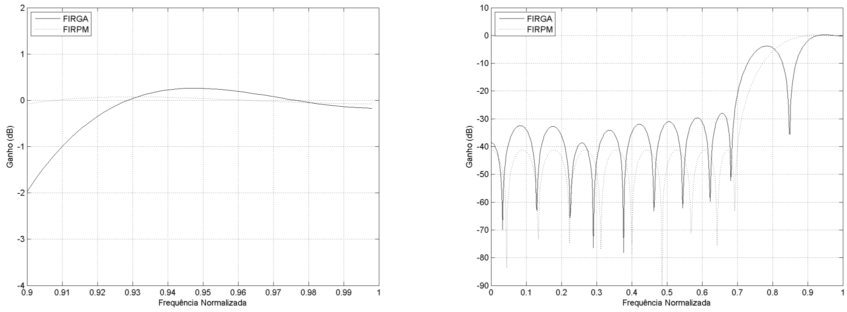
(d) Resposta em frequência

Figura 81 Comparativo entre filtros FIRGA e FIRPM passa-altas de ordem 21 para pesos $W_{GA} = [10, 1]$

Inicializando o GA com os coeficientes do filtro FIRPM obtivemos um filtro que é o filtro FIRPM com coeficientes quantizados. Este filtro apresenta bom resultado, estando de acordo com as especificações tanto da banda de passagem quanto da banda de rejeição.

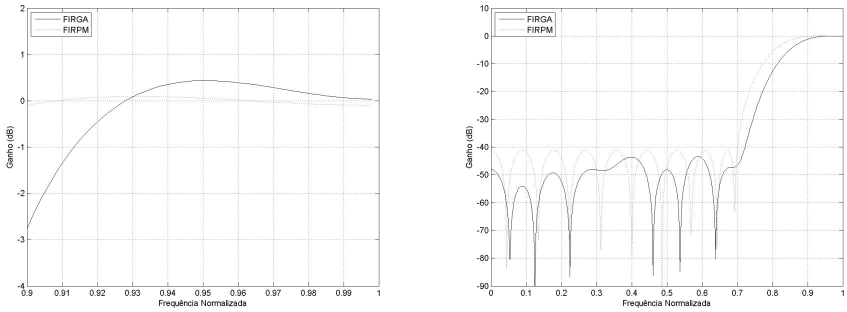
Para os filtros passa alta também procedeu-se ao aumento da ordem dos filtros para ter-se uma ideia do resultado, mesmo que este já tivesse boa resposta por encontrar os coeficientes do filtro FIRPM quantizados. Notou-se que com o aumento da ordem do filtro não há diferença alguma na resposta em frequência, ou seja, o GA zera os coeficientes que ultrapassam a ordem mínima do filtro resultando sempre no mesmo filtro de coeficientes quantizados.

De modo geral, obtivemos bons filtros passa-altas com ocorrências de filtros fora das especificações. O maior problema obtido nestes filtros foi a variação na banda de passagem e portanto a ordem destes deve ser aumentada para que eles possam cumprir as especificações. Ocorreu também a geração de filtros ruins, devido à rápida convergência



(a) Resposta em frequência na banda passante

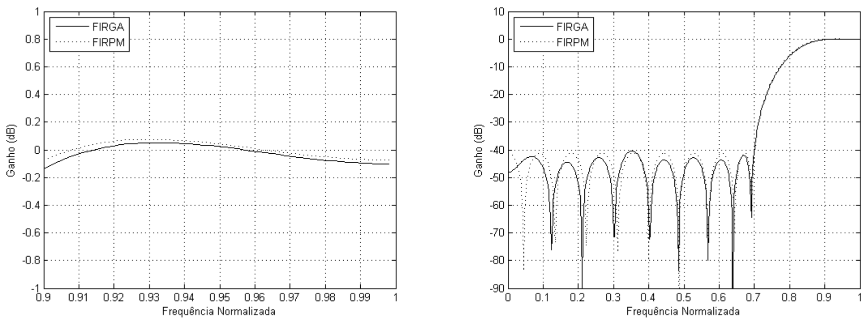
(b) Resposta em frequência



(c) Resposta em frequência na banda passante

(d) Resposta em frequência

Figura 82 Comparativo entre filtros FIRGA e FIRPM passa-altas de ordem 23 para pesos $W_{GA} = [1, 1]$



(a) Resposta em frequência na banda passante

(b) Resposta em frequência

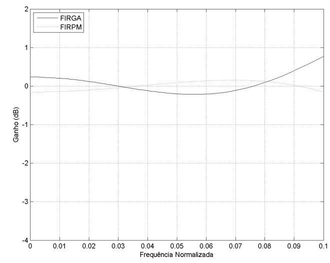
Figura 83 Comparativo entre filtros FIRGA, inicializados com coeficientes determinados, e FIRPM passa-altas de ordem 23 para pesos $W_{GA} = [1, 1]$

destes. Um quinto dos filtros gerados tiveram convergência prematura. Este foi o único tipo de filtro, que ao ter o GA inicializado com os coeficientes do filtro FIRPM, obteve como resposta o filtro FIRPM com os coeficientes quantizados.

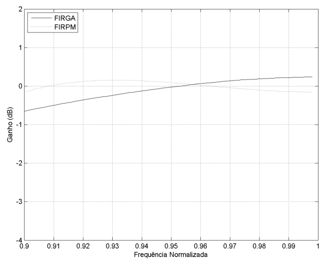
5.3 Filtro Rejeita-Faixas

As especificações do filtro rejeita-faixas foram bandas de passagem de 0 a 0,1 e de 0,9 a 1, ganho nestas faixas de 0dB com variação máxima de 0,2dB. Para a banda de rejeição a atenuação mínima foi de -35dB nas frequências normalizadas entre 0,25 e 0,75. O filtro FIRPM para estas especificações tem ordem 27.

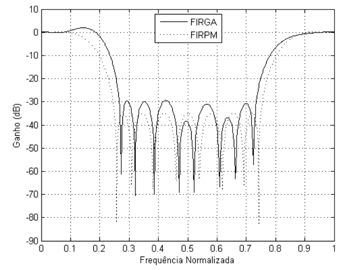
Na Figura 84 vemos o comportamento de filtros FIRGA rejeita-faixas de ordem 27. Os filtros gerados pelo GA são filtros que não condizem com as especificações, estes normalmente apresentam grande variação de ganho na banda passante, chegando a atingir 1dB, e atenuação na banda de rejeição em torno de -30dB. Nota-se que este filtro, como era de se esperar, tem dois pontos críticos que são o início e o fim da banda de rejeição. Porém, conseguimos obter um filtro de atenuação mínima de -40dB em detrimento da variação da banda passante que ficou em 1dB, este filtro é exibido na Figura 84(f).



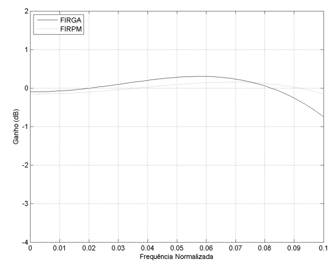
(a) Resposta em frequência na 1^a banda passante



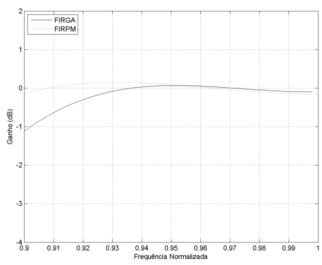
(b) Resposta em frequência na 2^a banda passante



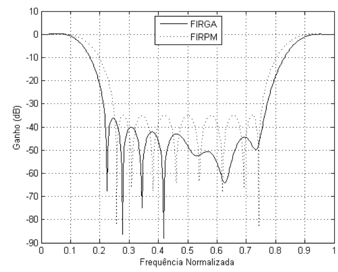
(c) Resposta em frequência



(d) Resposta em frequência na 1^a banda passante



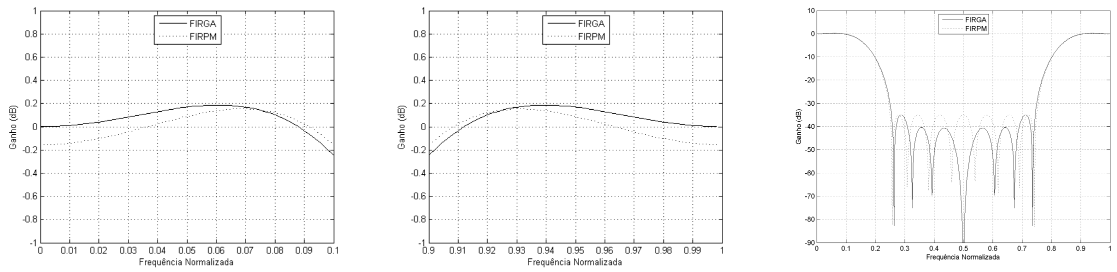
(e) Resposta em frequência na 2^a banda passante



(f) Resposta em frequência

Figura 84 Comparativo entre filtros FIRGA e FIRPM rejeita-faixas de ordem 27 para pesos $W_{GA} = [1, 1]$

Quando utilizamos os coeficientes do filtro FIRPM obtemos um FIRGA de ordem 25, ou seja, o GA zera o primeiro coeficiente, o que devido à simetria acarreta o aparecimento de outro zero no último coeficiente. Logo, o filtro FIRGA é um filtro duas ordens menor do que o filtro FIRPM. A resposta em frequência deste filtro é exibida na Figura 85. Este tem atenuação de -30dB no início e fim da banda de rejeição, porém na frequência 0,256 ele já apresenta -36dB e em 0,744 ele ainda apresenta -36dB de atenuação, ou seja, ele tem uma pequena faixa de 0,006 no inicio e fim da banda de rejeição nas quais estes filtro está fora das especificações. No restante desta banda o filtro está em concordância com as especificações apresentando atenuação inferior ou igual a -35dB. Nas faixas de passagem o filtro apresentou baixas variações do ganho, porém ultrapassaram o limite das especificações de 0,2dB ao atingirem 0,22dB.

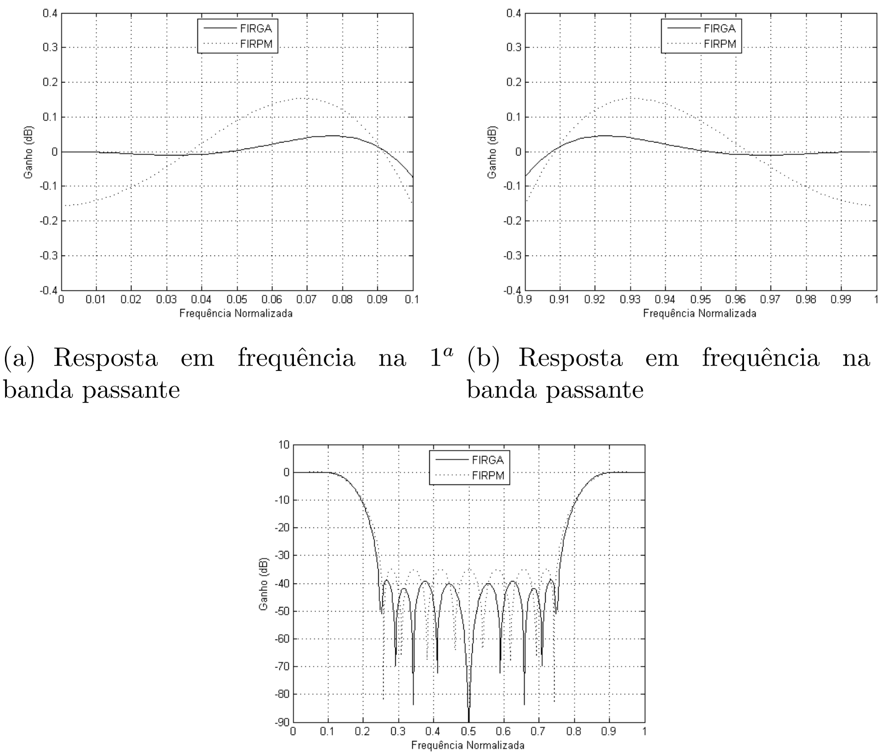


(a) Resposta em frequência na 1^a banda passante (b) Resposta em frequência na 2^a banda passante (c) Resposta em frequência

Figura 85 Comparativo entre filtros FIRGA, inicializados com coeficientes determinados, e FIRPM rejeita-faixas de ordem 25 e 27 respectivamente para pesos $W_{GA} = [1, 1]$

O filtro que atende plenamente a estas especificações é um filtro de ordem 29, que tem sua resposta em frequência exibida na figura Figura 86. Neste caso um aumento de duas ordens no comprimento do filtro permitiu a produção de um filtro FIRGA dentro das especificações, juntamente com os filtros passa-faixa, que serão vistos na seção 5.4, este filtro foi o que necessitou de menor aumento na ordem para o cumprimento das especificações de projeto.

O GA novamente provou ser capaz de criar filtros próximos às especificações, mas ainda assim houve variações de aproximadamente 1dB na banda de passagem ou atenuação mínima de -30dB na banda de rejeição. Para filtro rejeita-faixas notou-se que o número de filtros com convergência prematura foi de um terço da quantidade total de filtros gerados.



(a) Resposta em frequência na 1^a banda passante

(b) Resposta em frequência na 2^a banda passante

(c) Resposta em frequência

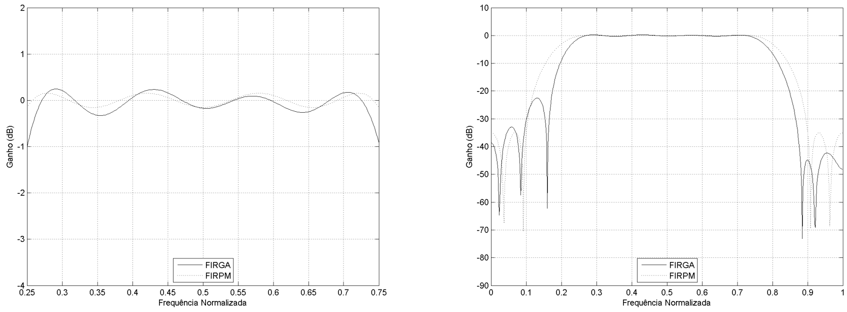
Figura 86 Comparativo entre filtros FIRGA de ordem 29 e FIRPM de ordem 27 rejeita-faixas para pesos $W_{GA} = [1, 1]$

5.4 **Filtro Passa-Faixas**

A especificação do filtro passa-faixas escolhido é uma faixa de passagem extensa de .25 até .75 e bandas de rejeição de 0 a 0,1 e 0,9 a 1, todas normalizadas. Atenuação mínima na banda de rejeição de -35dB, e variação máxima do ganho na faixa de passagem de 0,2dB. Para cumprir estas especificações é preciso de um filtro FIRPM de ordem 27.

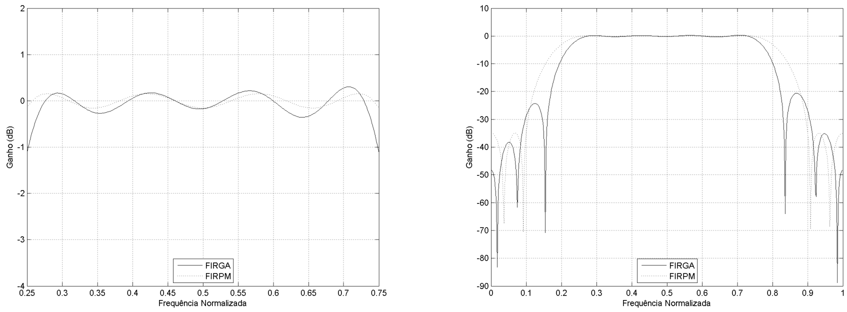
Amostras dos filtros FIRGA, de ordem 27 e peso $W_{GA} = [1; 1]$, criados são exibidas na Figura 87. A variação do ganho na banda de passagem nestes filtros atingiu 1dB nas extremidades desta faixa e oscilou entre -0,4dB e 0,4dB no restante desta banda, o que faz que estes filtros estejam em sua maioria fora das especificações de projeto. Estes filtros apresentaram atenuação mínima em torno de -30dB em uma ou nas duas bandas de rejeição, o que não condiz com os -35dB exigidos.

Ao aplicarmos os coeficientes do filtro FIRPM passa-faixas ao GA encontramos bons resultados como mostra a Figura 88. Ao contrário dos outros 3 tipos de filtros des-



(a) Resposta em frequência na banda passante

(b) Resposta em frequência

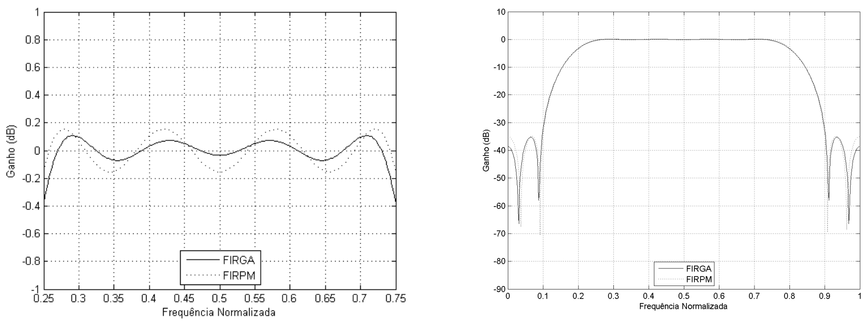


(c) Resposta em frequência na banda passante

(d) Resposta em frequência

Figura 87 Comparativo entre filtros FIRGA e FIRPM rejeita-faixas de ordem 27 para pesos $W_{GA} = [1, 1]$

critos anteriormente, o filtro FIRGA inicializado com os coeficientes do filtro FIRPM fica ligeiramente fora das especificações, atingindo uma variação do ganho nas extremidades da banda de passagem de 0,38dB, acima dos 0,2dB exigidos.

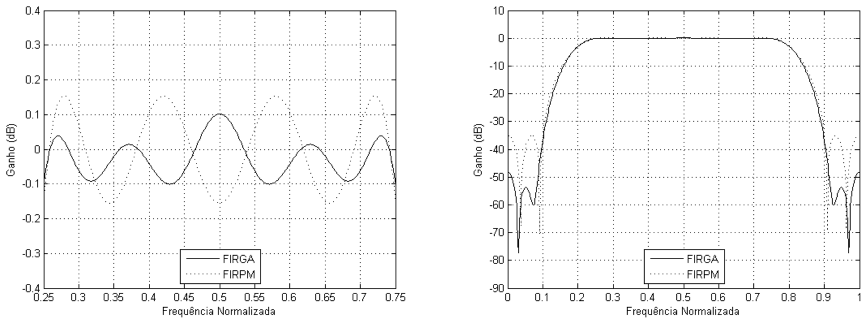


(a) Resposta em frequência na banda passante

(b) Resposta em frequência

Figura 88 Comparativo entre filtros FIRGA, inicializados com coeficientes determinados, e FIRPM rejeita-faixas de ordem 27 para pesos $W_{GA} = [1, 1]$

Para que as especificações fossem atingidas foi necessário o aumento de duas ordens no filtro. Sendo assim, um filtro FIRGA de ordem 29 cumpre todas as especificações de projeto e sua resposta em frequência é mostrada na Figura 89.



(a) Resposta em frequência na 1^a banda passante

(b) Resposta em frequência

Figura 89 Comparativo entre filtros FIRGA de ordem 29 e FIRPM de ordem 27 passa-faixas para pesos $W_{GA} = [1, 1]$

Conseguiu-se produzir bons filtros passa-faixas com o método proposto neste trabalho, mesmo não atingindo as especificações quando projetado para ter mesma ordem dos filtros FIRPM. Durante a produção deste filtro tivemos 40% dos filtros com convergência prematura, sendo o tipo de filtro entre os quatro que apresentou maior formação de filtros ruins.

CONCLUSÃO

Neste trabalho foi proposto um algoritmo genético (GA) para otimização do projeto de filtros com coeficientes representados por números em soma de potências de dois (SPT). O algoritmo tem como objetivo encontrar a melhor combinação de coeficientes para que um filtro quantizado obedeça as suas especificações, uma vez definida a ordem deste. O modelo utilizado tem como objetivo a minimização da função de erro da resposta em frequência do filtro em relação às suas especificações.

O algoritmo proposto une diversos métodos de projeto e aprimoramento de filtros para obter uma boa resposta em frequência. A representação numérica SPT utilizada neste trabalho permite melhor desempenho do que a representação binária, ao empregar somadores e subtratores simultaneamente, ela aumenta a quantidade de números representáveis para uma mesma quantidade de dígitos. Além disso, a representação SPT é de fácil implementação em circuitos, uma vez que seus dígitos correspondem a potências de base 2 e a implementação de um subtrator é tão direta quanto a de um somador.

Na representação usada não garantimos a quantidade mínima de somas quando representamos um número, mesmo que reduções na quantidade total de somas ocorram quando números SPTs são comparados com binários. Por outro lado, garantimos a representação de todos os possíveis valores dos coeficientes do filtro, estando limitado apenas pela precisão, ao contrário da representação canônica de dígito com sinal (CSD). A representação SPT permite a diminuição da quantidade de dígitos diferentes de zero, visando a redução do número de operações aritméticas, sem haver uma perda de valores representáveis tão grande quanto a perda ocorrida na representação CSD.

Combinada à representação SPT está a otimização via GA, que vasculha o espaço de busca a procura da melhor combinação de coeficientes para que o filtro projetado atenda às especificações. O algoritmo foi desenvolvido para o projeto de filtros passa-baixas, passa-altas, passa-faixas e rejeita-faixas.

Exaustivos testes foram feitos para a determinação dos parâmetros do GA utilizado, durante os quais foi criado um operador de cruzamento que produz bons resultados quando aplicado ao projeto de filtros. Estes resultados são superiores aos resultados obtidos com cruzamento em múltiplos pontos, pois o cruzamento proposto trabalha apenas com o reposicionamento dos coeficientes sem alterar seus valores. Aplicamos também uma mutação variável, que permite conciliar os efeitos de baixas e altas taxas de mutação. Foi feita também uma análise qualitativa da variação dos diversos GAs para a otimização de filtros.

Combinando as técnicas de representação e busca, conseguimos criar um algoritmo para o projeto de filtros digitais com coeficientes SPTs quantizados com 8 dígitos, capaz de criar filtros com razoável desempenho quando utilizado para projetar filtros passa-baixas, passa-altas, passa-faixas e rejeita-faixas de mesma ordem que o seu equivalente de precisão infinita. Quando o método desenvolvido foi inicializado aleatoriamente os filtros criados tiveram na maioria das vezes desempenho próximo às especificações, mas poucas vezes respeitando integralmente estas. Por outro lado, quando a inicialização foi feita com os coeficientes do filtro de precisão infinita os resultados foram bons, apresentando um número maior de filtros dentro das especificações. Assim, uma pré avaliação da superfície de busca torna-se necessária uma vez que além de limitar o tempo de execução do algoritmo, os resultados obtidos são superiores aos resultados obtidos com uma inicialização aleatória.

De modo geral, o algoritmo de projeto de filtros otimizado via GA proposto produziu bons resultados, caracterizado por filtros com respostas em frequência próximas às especificações, levando em conta as severas restrições impostas, que foram: mesma ordem do filtro de precisão infinita e profundidade de quantização de 8 dígitos.

Possibilidades Futuras

Uma vez que o GA desenvolvido neste trabalho é baseado em algoritmos de uso geral desenvolvido por terceiros, alterações muito profundas neste tornam-se difíceis e trabalhosas, por isso deveria-se criar um GA que contemplasse melhorias como:

- Aumento dos parâmetros de controle do processo de otimização. No GA original as únicas saídas que podemos avaliar são as aptidões máximas e a aptidão média da população por geração. Adicionamos alguns parâmetros de verificação como as variações nas faixas de passagem e rejeição, assim como as médias nessas bandas que são parâmetros simples de implementar. Porém, em um projeto próprio poderemos inserir outros tipos de parâmetros de saída para o monitoramento do progresso do filtro como o cálculo sensibilidade dos coeficientes por geração, detecção de coeficientes recorrentes e monitoramento da posição de polos e zeros do filtro.

- Implementação um sistema de variação de precisão dos coeficientes. Atualmente a precisão de todos os coeficientes é a mesma, ou seja cada gene tem exatamente a mesma quantidade de dígitos representativos. Mas, uma vez respeitando a quantidade total de dígitos, podemos variar a precisão de um ou outro coeficiente conforme as necessidades de projeto. Com este método podemos também realocar os dígitos de genes que repetidamente aparecem nos cromossomos e que têm dígitos inutilizados.

- Otimização para uso em computadores de múltiplos núcleos. Notou-se que o presente GA não usa a capacidade total de um computador de mais de um núcleo, o trabalho fica todo concentrado em apenas um deles. Podemos então distribuir as tarefas entre os núcleos, que representa um enorme ganho uma vez que GAs são algoritmos que podem ter suas etapas processadas em paralelo.

- Implementação de populações variáveis e de quantidade de genes variáveis. Não é possível, com o GA utilizado, fazer alterações no número de indivíduos a cada geração. A implementação desta funcionalidade permitiria controlar o tamanho da população em diversas gerações, agilizando o processo ou retardando-o de acordo com as necessidades de busca do algoritmo. Com a implementação de uma quantidade variável de genes poderemos fazer efetivamente a busca pelo filtro de ordem mínima, fazendo a busca iniciar com um pequeno número de coeficientes e ir aumentado a ordem dos filtros ao longo das gerações. O GA de hoje limita o tamanho da saída ao tamanho da entrada, o que nos força a escolher a ordem final do filtro no início do algoritmo. Para simular um efeito de busca em filtros de diferentes ordens devemos colocar um GA do tipo utilizado dentro do outro, o que torna o processo muito demorado e de difícil ajuste.

- Transformação do GA em um verdadeiro algoritmo de múltiplos objetivos. Hoje temos a função de avaliação como conciliadora dos objetivos do GA, ela o faz regendo a influência que a banda de passagem e rejeição têm sobre a aptidão do indivíduo. Podemos implementar um GA com diversos objetivos em paralelo com diversas funções de avaliação permitindo a classificação do indivíduo em diversos quesitos. Outra possibilidade é a implementação de nichos, que são subdivisões da população, que evoluam separadamente, tendo ou não os mesmos objetivos, onde para garantir diversidade da população introduz-se uma migração de indivíduos entre os nichos.

Outras alterações no GA que não dizem respeito às capacidades do algoritmo em si podem ser alteradas. A primeira delas é a função de avaliação, que pode ser modificada para levar em consideração outros parâmetros, como a variação e média na banda avaliada, entre outros. O uso de uma nova função de avaliação pode suprimir o uso do artifício utilizado para moldar a resposta em frequência na banda de transição. Além de alterações na formulação da função de avaliação podemos implementar algoritmos para a determinação dos pesos, uma vez que estes influenciam em muito na resposta em frequência do filtro obtido. Novos operadores de cruzamento e mutação podem ser aplicados.

Além disso, normalmente ao quantizarmos os coeficientes de um filtro aumentamos sua ordem para que este permaneça dentro das especificações, o que deveria ser feito para avaliarmos qual a ordem mínima para que o filtro quantizado atinja as especificações. Outro fator que deve ser levado em consideração é a profundidade de dígitos de quantização escolhida, que influi diretamente na quantidade de números representáveis e no erro de quantização, utilizamos neste trabalho apenas 8 dígitos. Testes com um maior número de dígitos permitirão fazer um balanço entre o aumento da ordem e o aumento da precisão.

Quanto à representação, seria interessante utilizar a representação CSD com escalonamento dos coeficientes, para garantir que estes cubram todos os valores que deseja-se representar, e a representação MSD para comparação. Nesta área poderemos avaliar os ganhos entre as representações quando avaliamos a resposta em frequência em função do número de operações aritméticas.

REFERÊNCIAS

[1] Abdullah Konak and David W. Coit and Alice E. Smith. Multi-objective optimization using genetic algorithms: A tutorial, 2006.

[2] Adams, J. W., and Willson, A. N. A New Approach to FIR Digital Filters with Fewer Multipliers and Reduced Sensitivity. *IEEE Transactions on Circuits and Systems 30*, 5 (May 1983), 277–283.

[3] Ait-Boudaoud, D., and Cemes, R. Modified sensitivity criterion for the design of power of two filters. *IEEE Transactions on Signal Processing 29*, 16 (August 1993), 1467–1469.

[4] Anekal B. Sripad and Donald L. Snyder. A Necessary and Sufficient Condition for Quantization Errors to be Uniform and White. *IEEE Transactions on Acoustics, Speech and Signal Processing.*

[5] Antonio Lazcano and Stanley L. Miller. The origin and early evolution of life: prebiotic chemistry, the pre-RNA world, and time. *Cell.*

[6] Antoniou, A. *Digital Filters: Analysis, Design and Applications.* McGraw-Hill, 1993.

[7] Arabas, J., Michalewicz, Z., and Mulawka, J. GAVaPS - a Genetic Algorithm with Varying Population Size. 73–78.

[8] Ashrafzadeh, F., and Nowrouzian, B. Crossover and Mutation In Genetic Algorithms Employing Canonical Signal-Digit Number System. In *Procedings of the 40th Midwest Symposium on Circuits and Systems* (Sacramento, CA , USA, August 1997), vol. 2, pp. 702–705.

[9] Backenius, E., Säll, E., and Gustafsson, O. Bidirectional Conversion to Minimum Signed-Digit Representation.

[10] Blickle, T. *Theory of Evolutionary Algorithms and Application to System Synthesis.* PhD thesis, Swiss Federal Institute of Technology, Zurich, January 1996.

[11] Burke, E. K., and Kendall, G. *Search Metodologies Introductory Tutorials in Optimization and Decision Support Techniques.* Springer US, 2005.

[12] Burrus, C., Soewito, A., and Gopinath, R. Least Squared Error FIR Filter Design with Transition Bands. *IEEE Transactions Signal Proc. 40*, 6 (June 1992), 1327–1340.

[13] Capmany, J., Ortega, B., and Pastor, D. A Tutorial on Microwave Photonic Filters. *Journal of Lightwave Technology 24*, 1 (2006), 201–229.

[14] Chan, D., and ., L. R. Analysis of Quantization Errors in the Direct Form for Finite Impulse Response Digital Filters. *IEEE Transactions on Audio and Eletroacoustics 21*, 4 (August 1973), 354–366.

[15] Chen, C.-L., and Jr, A. N. W. A Trellis Search Algorithm for the Design of FIR Filters with Signed-Powers-of-Two Coefficients. *IEEE Transactions on Circuits and Systems 46*, 1 (January 1999), 29–39.

[16] Chen, H., and Chen, S. W. A moving average based filtering system with its application to real-time QRS detection. *Computers in Cardiology* (September 2003), 585–588.

[17] Christopher R. Houk and Jeffery A. Joines and Michael G. Kay. A Genetic Algorithm for Function Optimization: A Matlab Implementation, 1995.

[18] Darrell Whitley. A Genetic Algorithm Tutorial, 1993.

[19] Darwin, C., and Carroll, J. *On the origins of species by means of natural selection*. Broadview Press, 2003.

[20] David Butcher. Muller's Ratchet, Epistasis and Mutation Effects. *Genetics Society of America*.

[21] Davidor, Y. Epistasis Variance: Suitability of a Representation to Genetic Algorithms. *Complex Systems 4*, 4 (1990), 369–383.

[22] Davidor, Y. Epistasis Variance: A Viewpoint on GA Hardness. In *Foundations of Genetic Algorithms* (Vail, Colorado, USA, July 1992), pp. 23–35.

[23] de Jong, K. Learning with Genetic Algorithms: An Overview. *Machine Learning 3* (1988), 121–138.

[24] Diniz, P., Silva, E., and Netto, S. *Digital Signal Processing: System Analysis and Design*. Cambridge University Press, 2002.

[25] Emad Elbeltagia and Tarek Hegazyb and Donald Grierson. Comparison among five evolutionary-based optimization algorithms. *Advanced Engineering Informatics*.

[26] Eshelman, L. The CHC Adaptive Search Algorithm. *Foundations of Genetic Algorithms 1* (1990), 265–283.

[27] Fong, W., and Godsill, S. J. Monte Carlo smoothing with application to audio signal enhancement. *IEEE Transactions on Signal Processing 50*, 2 (February 2002), 438–449.

[28] Forrest, S. Genetic Algorithms. *ACM Computing Surveys* (January 1996).

[29] Fox, T., and Turner, L. The design of peak-constrained least squares FIR filters with low-complexity finite-precision coefficients. *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing 49*, 2 (February 2002), 151–154.

[30] Fuller, A. T., Ashrafzadeh, F., and Nowrouzian, B. Optimization of FIR Digital Filters over the Canonical Signed-Digit Coefficient Space using Genetic Algorithms. In *Procedings of 1998 Midwest Symposium on Circuits and Systems* (Notre Dame, IN, USA, August 1998), pp. 456–459.

[31] Giotis, A., Emmerich, M., Naujoks, B., Giannakoglou, K., and Bäck, T. Low-cost stochastic Optimization for engineering applications. In *1st Conference on Advances and Applications of GiD* (Barcelona, Spain, February 2002), pp. 1–6.

[32] Gottlieb, D., and Chu, C.-W. On the Gibbs Phenomenon and its Resolution. *Society for Industrial and Applied Mathematics Review 39*, 4 (December 1997), 305–378.

[33] Gouveia, J. F. . Master's thesis, Universidade Federal Rural de Pernambuco, 2010.

[34] Grefenstette, J. J. Optimization of Control Parameters for Genetic Algorithms. *IEEE Transactions on Systems, Man and Cybernetics 16*, 1 (1986), 122–128.

[35] Hajela, P., and Lin, C. Genetic search strategies in multicriterion optimal design. *Structural and Multidiciplinary Optimization 4*, 2 (1992), 99–107.

[36] Hartley, R. I. Optimization of Canonic Signed Digit Multipliers for Filter Design. 1992–1995.

[37] Hartley, R. I. Digital Filter Synthesis Based on an Algorithm to Generate All Minimal Signed Digit Representations. *IEEE Transaction on Circuits and Systems II: Analog and Digital Signal Processing 43*, 10 (October 1996), 677–688.

[38] Hewlitt, R. M., and Swartzlander, E. S. Canonical signed digit representation for FIR digital filters. In *IEEE Workshop on Signal Processing Systems* (Lafayette, LA , USA, October 2000), pp. 416–426.

[39] Hwang, H. *Computer Arithmetics: Principles, Archtecture and Design*. Wiley and Sons, 1979.

[40] Jie, T. F., Ito, R., Suyama, K., and Hirabayashi, R. A New Heuristic Signed-Power of Two Term Allocation Approach for Designing of FIR filters. In *Proceedings of the 2003 International Symposium on Circuits and Systems* (May 2003), pp. 285–288.

[41] John W. Drake and Brian Charleswortha and Deborah Charlesworth and James F. Crow. Rates of Spontaneous Mutation. *Genetics Society of America*.

[42] Kobrinski, H., and Cheung, K. W. Wavelength-tunable optical filters: applications and technologies . *IEEE Communications Magazine 27*, 10 (October 1999), 53–63.

[43] Koren, I. *Computer Arithmetic Algorithms*. Prentice Hall, 1993.

[44] Kreider, D., Ostberg, D., Kuller, R., and Perkins, F. *An Introduction to Linear Analysis*. Addison-Wesley, 1966.

[45] Kubalík, J., and Lazansky, J. Genetic Algorithms and their Testing. *AIP Conference Proceedings 465* (March 1999), 217–229.

[46] Kurzweil, R. *The Singularuty is Near*. Penguin Books, 2005.

[47] Lee, A., Ahmadi, M., Julien, G., Miller, W., and Lashkari, R. Digital Filter Design Using Genetic Algorithm. *IEEE Symp.* (June 1998), 34–38.

[48] Lee, C.-S., Guo, S.-M., and Hsu, C.-Y. Genetic-based fuzzy image filter and its application to image processing. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics 35*, 4 (August 2005), 694–711.

[49] Leung, Y., Gao, Y., and Xu, Z.-B. Degree of Population Diversity: A Perspective on Premature Convergence in Genetic Algorithms and its Markov Chain Analysis. *IEEE Transactions on Neural Networks 8*, 5 (1988), 121–138.

[50] Li, D. Minimum Number of Adders for Implementing a Multiplier and Its Application to the Design of Multiplierless Digital Filters. *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing 42*, 7 (July 1995), 453–460.

[51] Li, D., Lim, Y., and Song, J. A polynomial-time algorithm for design of digital filters with power-of-two coefficients. *IEEE International Symposium on Circuits and Systems 1*, 3 (May 1993), 84–87.

[52] Lim, Y., Evans, J., and Liu, B. Decomposition of Binary Integers into Signed Power-of-Two Terms. *IEEE Transactions on Circuits and Systems 38*, 6 (June 1991), 667–672.

[53] Lothar M. Schimitt. Theory of Genetic Algorithms, 2000.

[54] Marco Aurélio Cavalcanti Pacheco. Algoritmos Genéticos: Princípios e Aplicações, 1999.

[55] Mauldin, M. L. Maintaining Diversity in Genetic Search. *AAAI-84 Proceedings* (1984), 247–250.

[56] Mühlenbein, H. The Equation for the Response to Selesction ant its Use for Prediction.

[57] Mühlenbein, H., and Schilierkamp-Voosen, D. Analysis of Selection, Mutation and Recombination in Genetic Algorithms. *Lecture Notes in Computer Science 899* (1995), 142–168.

[58] Mühlenbein, H., and Voigt, H.-M. Gene Pool Recombination in Genetic Algorithms.

[59] Michalewicz, Z., and Janicow, C. Z. Handling Constraints in Genetic Algorithms. In *Proceedings of the Fourth International Conference on Genetic Algorithms* (San Diego, CA, USA, July 1991), pp. 151–157.

[60] Miller, B. L., and Goldberg, D. E. Genetic Algorithms, Selection Schemes, and the Varying Effects of Noise. *Evolutionary Computation 4*, 2 (August 1996), 113–131.

[61] Miller, J. A., Potter, W. D., Gandham, R. V., and Lapena, C. N. An Evaluation of Local Improvement Operators for Genetic Algorithms. *IEEE Transactions on Systems, Man and Cybernetics 23*, 5 (September 1993), 1340–1351.

[62] Mitra, S. *Digital Signal Processing: A computer Based Approach.* McGraw-Hill, 1998.

[63] Mitra, S., and Kaiser, J. *Handbook for Digital Signal Processing.* John Wiley and Sons, 1993.

[64] Murata, T., and Ishibuchi, H. MOGA: Multi-Objective Genetic Algorithms. In *IEEE International Conference on Evolutionary Computation* (Perth, WA, Australia, November 1995), pp. 289–294.

[65] Murata, T., Ishibuchi, H., and Tanaka, H. Multi-Objective Genetic Algorithm and its Applications to Flowshop Scheduling. *Computers & Industrial Engineering 30*, 4 (September 1996), 957–968.

[66] Oppenheim, A., Willsky, A., and Young, I. *Signals and Systems.* Prentice-Hall, 1983.

[67] Oppenheim, A. V., and Schafer, R. W. *Digital Signal Processing.* Prentice Hall, 1975.

[68] Paatero, T., and Karjalainen, M. Kautz Filters and Generalized Frequency Resolution Theory and Audio Applications. *Journal of the Audio Engineering Society 51*, 1 (February 2003), 27–44.

[69] Park, I. C., and Kang, H. J. Digital Filter Synthesis Based on Minimal Signed Digit Representation. In *Proceedings of the 38th annual Design Automation Conference* (Las Vegas, USA, June 2001), pp. 468–473.

[70] Park, I. C., and Kang, H. J. Digital Filter Synthesis Based on an Algorithm to Generate All Minimal Signed Digit Representations. *IEEE Transaction on Computer-Aided Design of Integrated Circuits and Systems 21*, 12 (December 2002), 1525–1529.

[71] Parker, Y. L. S. Filter Design over a Discrete Power-of-Two Coeficient Space. *IEEE Trans. on Acoustics, Speech and Signal Processing 31*, 3 (June 1983), 583–591.

[72] Peled, A. A New Hardware Realization of Digital Filters. *IEEE Transactions on Acoustics, Speech and Signal Processing 22*, 6 (December 1974), 456–462.

[73] Philip Hunter. The great leap forward. Major evolutionary jumps might be caused by changes in gene regulation rather than the emergence of new genes. *European Molecular Biology Organization.*

[74] Pun, C. K. S., Chan, S. C., Yeung, K. S., and Ho, K. L. On the design and implementation of FIR and IIR digital filters with variable frequency characteristics. *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing 49*, 11 (November 2002), 689–703.

[75] Qi, X., and Palmieri, F. The Diversification Role of Crossover in the Genetic Algorithm. In *Proceedings of the Fifth International Conference on Genetic Algorithms* (Urbana-Champaign, IL, USA, June 1993), pp. 132–137.

[76] Rabiner, L. Techniques for Designing Finite-Duration Impulse-Response Digital Filters. *IEEE Transactions. Comm. Tech. 19*, 2 (April 1971), 188–195.

[77] Rabiner, L., McClellan, J., and Parks, T. A Computer Program for Designing Optimum FIR Linear-Phase Digital Filters. *IEEE Transactions on Audio and Electroacoustics 21*, 6 (December 1973), 506–526.

[78] Rabiner, L., McClellan, J., and Parks, T. FIR Digital Filter Design Techniques Using Weighted Chebyshev Approximation. *The IEEE 63*, 4 (April 1975), 596–609.

[79] Reitweisner, G. W. *"Binary Arithmetics" in Advances in Computers.* Academic Press, 1960.

[80] Samueli, H. An Improved Search Algorithm for the Design of Multiplierless FIR Filters with Powers-of-Two Coefficients. *IEEE Transactions on Circuits and Systems 36*, 7 (July 1989), 1044–1047.

[81] Shaffeu, H., Jones, M., Griffiths, H., and Taylor, J. Improved Design procedure for multiplierless FIR digital filters. *Electronics Letters 27*, 13 (June 1991), 1142–1144.

[82] Sharma, S., Saxena, R., and Saxena, S. Design of FIR filters using variable window families: A comparative study. *J. Indian Inst. Sci. 84* (September-October 2004), 155–161.

[83] Spears, W., and DeJong, K. An Analysis of Multi-Point Crossover. *Foundations of Genetic Algorithms 1* (1990), 301–315.

[84] Spears, W., and DeJong, K. An Analysis of the Interacting Roles of Population Size and Crossover in Genetic Algorithms. *Lecture Notes in Computer Science 496* (1991), 38–47.

[85] S.R. Powell and P.M. Chau. Efficient Narrowband FIR and IFIR Filters Based on Powers-of-Two Sigma-Delta Coefficient Truncation. *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*.

[86] Sun, Y., Chan, K. L., and Krishnan, S. M. ECG signal conditioning by morphological filtering. *Computers in Biology and Medicine 32*, 6 (November 2002), 465–479.

[87] Tadmor, E. Filters Mollifiers and the Computation of Gibbs Phenomenon. *Acta Numerica 16*.

[88] Tang, K., Man, K., Kwong, S., and He, Q. Genetic Algorithms and their Application. *IEEE Signal Processing Magazine 13*, 6 (November 1996), 22–37.

[89] Thakor, N. V., and Y. S, Z. Applications of adaptive filtering to ECG analysis: noise cancellation and arrhythmia detection. *IEEE Transactions on Biomedical Engineering 38*, 8 (August 1991), 785–794.

[90] Thierens, D., and Goldberg, D. Convergence Models of Genetic Algorithm Selection Schemes. *Parallel Problem Solving from Nature 866* (October 1994), 119–129.

[91] Thierens, D., and Goldberg, D. Elitist Recombination:an integrated selection recombination GA. In *Proceedings of the First IEEE Conference on Evolutionary Computation* (Orlando, FL , USA, Jene 1994), pp. 508–512.

[92] Vrahatis, K. P. M. Recent approaches to global optimization problems through Particle Swarm Optimization. *Natural Computing 1*, 2 (May 2002), 235–306.

[93] Wang, Y., and Roy, K. CSDC: a new complexity reduction technique for multiplierless implementation of digital FIR filters. *IEEE Transactions on Circuits and Systems I: Regular Papers 52*, 9 (September 2005), 1845–1853.

[94] Xu, F., Chang, C. H., and Jong, C. C. Design of Low-Complexity FIR Filters Based on Signed-Powers-of-Two Coefficients With Reusable Common Subexpressions. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems 26*, 10 (October 2007), 1898–1907.

[95] Yang, J., Liu, L., Jiang, T., and Fan, Y. A modified Gabor filter design method for fingerprint image enhancement . *Pattern Recognition Letters 24*, 12 (August 2003), 1805–1817.

[96] Yli-Kaakinein, J., and Saramäki, T. A Systematic Algorithm for the Design of Multiplierless FIR Filters. 185–188.

[97] Yli-Kaakinen, J., and Saramäki, T. A systematic algorithm for the design of multiplierless FIR filters. In *The 2001 IEEE International Symposium on Circuits and Systems ISCAS.* (Sydney, NSW , Australia, May 2001), vol. 2, pp. 185–188.

[98] Yli-Kaakinen, J., and Saramäki, T. A Systematic Algorithm for the Design of Lattice Wave Digital Filters With Short-Coefficient Wordlength. *IEEE Transactions on Circuits and Systems 54*, 8 (August 2007), 1838–1851.

[99] Yu, Y. Multiplierless Multirate FIR Filter Design and Implementation. Master's thesis, National University of Singapore, 2003.