

**Desenvolvimento de um método  
baseado em algoritmos de  
inteligência artificial para ajustar  
aparelhos auditivos**

*Relatório submetido à Universidade Federal de Santa Catarina  
como requisito para a aprovação na disciplina  
**DAS 5511: Projeto de Fim de Curso***

***Natan Grando***

*Florianópolis, Agosto de 2015*

# **Desenvolvimento de um método baseado em algoritmos de inteligência artificial para ajustar aparelhos auditivos**

***Natan Grando***

Esta monografia foi julgada no contexto da disciplina  
**DAS5511: Projeto de Fim de Curso**  
e aprovada na sua forma final pelo  
**Curso de Engenharia de Controle e Automação**

***Prof. Jomi Fred Hübner***

---

Assinatura do Orientador

Banca Examinadora:

João Carlos Silva dos Santos  
*Orientador na Empresa*

Prof. Jomi Fred Hübner  
*Orientador no Curso*

---

*Avaliador*

---

---

*Debatedores*

## Agradecimentos

Dedico este espaço para agradecer todos aqueles que, de alguma forma, contribuíram para fosse possível a conclusão deste curso.

Em primeiro lugar gostaria de agradecer a minha mãe, Marlene Grandó, a minha irmã, Muriel Grandó, e a minha namorada, Fernanda Gasparin, por todo o apoio, carinho e companheirismo demonstrado até hoje, sem os quais nada disto seria possível.

Em seguida gostaria de agradecer aos colegas e amigos do curso de Engenharia de Controle e Automação com quem passei muitos momentos inesquecíveis. Em especial, gostaria de agradecer à Lucas Ventura, Gustavo Martins, Eduardo Schmidt e Pedro Peralta.

Também gostaria de agradecer à todos os professores do curso de Engenharia de Controle e Automação da UFSC pelas lições transmitidas. Em especial, gostaria de agradecer ao professor Jomi Fred Hübner pela orientação ao longo deste projeto.

Por fim, mas de forma igualmente importante, gostaria de agradecer à WaveTech Soluções Tecnológicas, onde este projeto foi desenvolvido, e a toda sua equipe. De maneira especial, gostaria de agradecer a Alexandre Ferreira e a Guillaume Barrault pela oportunidade de trabalhar nesta empresa. Também gostaria de agradecer a João Carlos da Silva Santos pela orientação e amizade e a André Vicente Milack pelo coleguismo no desenvolvimento do WaveFit.

## Resumo

O ajuste de aparelhos auditivos é um processo complexo, demorado e requer um conhecimento profundo sobre o funcionamento de cada aparelho auditivo para que se obtenham bons resultados. Durante este processo são modificados os parâmetros do chip do aparelho auditivo para gerar uma resposta em frequência que atenda às necessidades do paciente com deficiência auditiva. Caso o ajuste não seja satisfatório, o paciente é prejudicado e pode até mesmo desistir do tratamento.

Buscando facilitar este processo e obter melhores resultados de tratamentos através do uso de aparelhos auditivos, a WaveTech Soluções Tecnológicas propôs automatizar parcialmente o ajuste de um aparelho auditivo através do software que está desenvolvendo em colaboração com profissionais da saúde auditiva da região. Tal software é denominado WaveFit e está integrado com a plataforma Noah, que permite o gerenciamento de pacientes, cadastro de audiogramas e ajuste de aparelhos de diversos fabricantes de forma unificada.

Este projeto está relacionado ao desenvolvimento do WaveFit e seu foco é ajustar os parâmetros do chip de forma automatizada através de algoritmos de inteligência artificial. Na literatura acadêmica existem diversos algoritmos de otimização que podem ser usados para resolver o problema de ajuste dos parâmetros de um aparelho auditivo. Neste trabalho, os resultados obtidos pelos algoritmos *Hill Climbing*, *Simulated Annealing* e algoritmo genético são comparados. Também é proposto um algoritmo para realizar o ajuste de parâmetros de forma mais rápida e seus resultados são comparados aos resultados obtidos pelos demais algoritmos.

## **Abstract**

Hearing aid adjustment is a complex and slow processes that demands profound knowledge of how each hearing aid work in order to obtain good results. Throughout this process, the hearing aid chip parameters are modified to generate a frequency response that meets the hearing impaired patient needs. If the adjustment is not satisfactory, the patient is hindered and may even quit his treatment.

Seeking ease this process and obtain better results of treatments that uses hearing aids, WaveTech Soluções Tecnológicas proposed to partially automatize the hearing aid adjustment through the software that it is developing in collaboration with local hearing health professionals. This software is called WaveFit and is integrated with Noah platform, which allows patient management, audiograms register and unified multiple brands hearing aids adjustment.

This project is related to WaveFit development and its goal is adjust hearing aid chip parameters in an automated way through artificial intelligence algorithms. In academic literature there are many optimization algorithms that could be used to solve the hearing aid parameters adjustment problem. In this work, the results obtained by Hill Climbing, Simulated Annealing and genetic algorithm are compared. Another algorithm is also proposed to perform the hearing aid parameters adjustment faster and its results are compared to the results of the other algorithms.

# Sumário

Capítulo 1: Introdução .....	1
1.1 Sobre a Empresa .....	2
Capítulo 2: Fundamentação Teórica .....	4
2.1 Audição .....	4
2.2 Sinal Sonoro .....	5
2.3 Perda Auditiva .....	7
2.4 Aparelhos Auditivos .....	9
2.5 Regras de Ganho .....	12
2.5.1 FIG6 .....	13
2.6 Algoritmos de Otimização .....	14
2.6.1 <i>Hill Climbing</i> .....	15
2.6.2 <i>Simulated Annealing</i> .....	16
2.6.3 Algoritmo Genético.....	18
Capítulo 3: WaveFit e Análise do Problema .....	20
3.1 WaveFit.....	20
3.1.1 Cadastro de Pacientes .....	21
3.1.2 Cadastro de Audiograma .....	22
3.1.3 Seleção de Aparelho Auditivo .....	23
3.1.4 Ajuste dos Parâmetros .....	24
3.2 Análise do Problema.....	25
Capítulo 4: Desenvolvimento dos Algoritmos de Otimização .....	27
4.1 Correção das Curvas de Resposta em Frequência .....	29
4.2 <i>Hill Climbing</i> .....	31

4.2.1 Resultados .....	32
4.3 <i>Simulated Annealing</i> .....	34
4.3.1 Resultados .....	36
4.4 Algoritmo Genético.....	38
4.4.1 Resultados .....	40
Capítulo 5: Algoritmo de Otimização Proposto .....	43
5.1 Identificação dos Parâmetros Ajustados Automaticamente .....	43
5.2 Separação dos Parâmetros em Canais.....	45
5.3 Algoritmo Proposto.....	46
Capítulo 6: Resultados .....	49
Capítulo 7: Conclusões e Perspectivas .....	51



## **Simbologia**

AASI – Aparelho de Amplificação Sonora Individual

BTE – Behind the Ear

CIC – Completely in Canal

DLL – Dynamic-link library

DSL – Desired Sound Level

HL – Hearing Level

NAL – National Acoustic Laboratories

OMS – Organização Mundial da Saúde

SPL – Sound Pressure Level

RITE – Receiver in The Ear

UCL – Uncomfortable Listening Level

VA – Via Aérea

VO – Via Óssea

## Capítulo 1: Introdução

Uma parcela significativa da população mundial possui algum tipo de deficiência auditiva. Segundo o censo realizado pelo IBGE em 2010, mais de 9,7 milhões de brasileiros são acometidos por esse mal, o que corresponde a aproximadamente 5% da população nacional [1].

Indivíduos com deficiência auditiva podem sofrer de outros problemas decorrentes da sua dificuldade de comunicação, como irritabilidade, depressão, isolamento social, redução da saúde psicológica e física, entre outros. Mas existem tratamentos para alguns casos de deficiência auditiva que podem prevenir tais problemas.

Alguns tipos de deficiência auditiva podem ser amenizados através do uso de Aparelhos de Amplificação Sonora Individual – AASI, ou popularmente, aparelhos auditivos. Tais dispositivos tiveram grande avanço tecnológico ao longo das últimas décadas, entrando no mundo digital, o que trouxe uma maior flexibilidade para moldar sinais acústicos para atender as necessidades de cada paciente. Em contrapartida, tornaram-se muito mais complexos, pois possuem muitos parâmetros que devem ser corretamente ajustados por um fonoaudiólogo [2].

O ajuste dos parâmetros não é uma tarefa simples, pois exige que o fonoaudiólogo possua um conhecimento profundo sobre o funcionamento de cada AASI. Mesmo para um fonoaudiólogo bem capacitado, o processo de ajuste é uma tarefa cansativa e demorada.

O objetivo deste trabalho é apresentar uma solução que permita ajustar os parâmetros de um aparelho auditivo, baseado nas necessidades individuais do paciente, de forma automatizada e rápida. A solução apresentará uma configuração de parâmetros para o AASI que não deve ser tomada como final, mas sim como um ponto de partida mais próximo da solução ideal, para facilitar o trabalho do fonoaudiólogo.

É extremamente importante ajustar de forma correta um aparelho auditivo para que o paciente obtenha bons resultados de seu tratamento. Um aparelho mal ajustado pode levar até mesmo a desistência do tratamento.

Este projeto foi desenvolvido na empresa WaveTech e está relacionado ao desenvolvimento do software de audiologia WaveFit. Tal software permite que um fonoaudiólogo cadastre exames de audiometria de pacientes e realize ajuste de aparelhos auditivos. Atualmente o WaveFit encontra-se em avançado estágio de desenvolvimento.

Este documento está organizado em seis capítulos. O Capítulo 2 mostra uma visão geral sobre os conceitos relacionados ao projeto, a fim de contextualizar o leitor. O Capítulo 3 apresenta o problema que se propõe resolver e as técnicas usadas para resolvê-lo. No Capítulo 4, são apresentadas as implementações das soluções vistas no capítulo anterior. Também neste capítulo, é apresentada uma solução proposta pelo autor. Os resultados de cada solução são analisados e comparados no Capítulo 5. No Capítulo 6, são apresentadas as conclusões e perspectivas futuras relacionadas a este projeto.

## **1.1 Sobre a Empresa**

Este projeto foi desenvolvido na WaveTech Soluções Tecnológicas<sup>1</sup>, que é uma empresa startup de alta tecnologia, fundada em 2012 por dois sócios, ambos engenheiros eletricitas. Está instalada no edifício CELTA, pertencente ao parque Tecnológico Alfa, em Florianópolis – SC.

A empresa conta com cerca de 20 funcionários e desenvolve atividades nas áreas de engenharia biomédica, engenharia submarina, controle ativo de ruído e vibração. Além do desenvolvimento do WaveFit, destacam-se os seguintes projetos da empresa:

- Supervisão no desenvolvimento do primeiro chip nacional para aparelhos auditivos;

---

<sup>1</sup> Site da empresa: <http://www.wavetech-st.com/>

- Elaboração de algoritmos de processamento de sinais para aparelhos auditivos;
- Aplicativo simulador de próteses auditivas para iPhone;
- Projeto e design de cápsulas para aparelhos auditivos;
- Desenvolvimento, em conjunto com a UFSC e demais entidades, de um transdutor implantável e algoritmos avançados para implantes cocleares com objetivo de adaptação à realidade brasileira;
- Desenvolvimento de um simulador de implante coclear;
- Desenvolvimento de dispositivo sonar.

## Capítulo 2: Fundamentação Teórica

Neste capítulo, são introduzidos alguns conceitos envolvidos no contexto deste trabalho, a fim de facilitar o entendimento do que é abordado nos capítulos seguintes.

### 2.1 Audição

O sistema auditivo humano é constituído por estruturas sensoriais e conexões responsáveis pela audição. Este sistema pode ser classificado em duas partes distintas: o sistema auditivo periférico e o sistema auditivo central.

O sistema auditivo periférico, ilustrado na Figura 1, é dividido em: orelha externa, orelha média, orelha interna e sistema nervoso periférico. A orelha externa é composta pelas estruturas externas e pelo meato acústico externo. A orelha média compreende a membrana timpânica, os ossos e os músculos intratimpânicos. A orelha interna compreende a cóclea e as estruturas intracocleares.

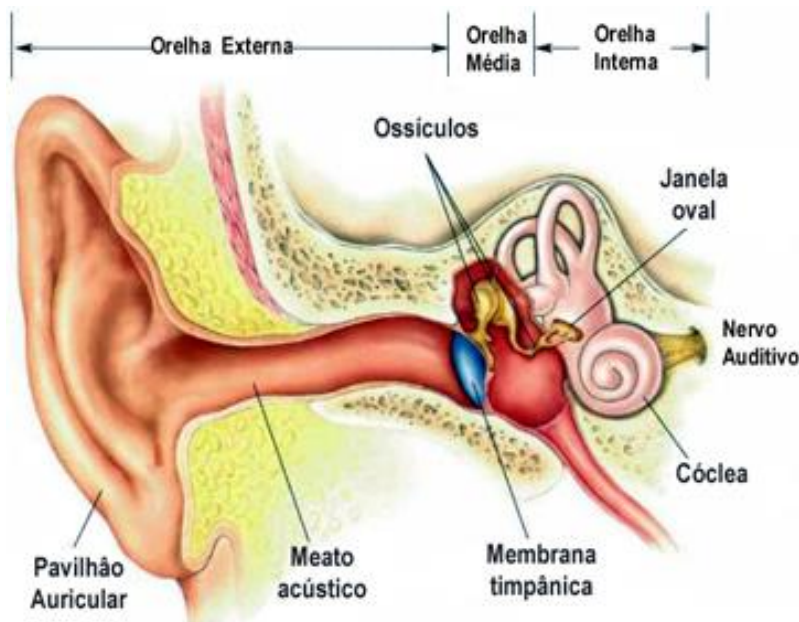


Figura 1 - Estruturas da orelha<sup>2</sup>

<sup>2</sup> Fonte: [http://www2.ibb.unesp.br/Museu\\_Escola/2\\_qualidade\\_vida\\_humana/Museu2\\_qualida\\_de\\_corpo\\_sensorial\\_audicao1.htm](http://www2.ibb.unesp.br/Museu_Escola/2_qualidade_vida_humana/Museu2_qualida_de_corpo_sensorial_audicao1.htm)

A cóclea é preenchida por perilinfa, um líquido praticamente incompressível e é limitada por paredes ósseas. Dentro desta estrutura existem cílios que captam as vibrações mecânicas da perilinfa provocadas pelo som. A cóclea pode receber a transmissão sonora de duas formas distintas: via aérea – VA, ou via óssea – VO.

Na via aérea, a onda sonora é inicialmente captada pela orelha e conduzida à membrana timpânica pelo meato acústico externo. A cóclea, então, recebe a transmissão sonora através da vibração da membrana timpânica, da cadeia ossicular e da janela oval.

Já na via óssea, o sinal é transmitido à cóclea diretamente através da vibração dos ossos do crânio.

O sinal sonoro dentro da cóclea provoca uma estimulação nos cílios que, por sua vez, polarizam e despolarizam as células ciliadas internas, liberando neurotransmissores e formando uma mensagem sonora eletricamente codificada. Este sinal elétrico é enviado, através do nervo coclear, ao sistema nervoso central [3].

## 2.2 Sinal Sonoro

A orelha humana pode perceber uma grande faixa de intensidade sonora, cuja medida física é pressão sonora. Um humano com audição saudável, pode ouvir sons que vão de aproximadamente 20  $\mu\text{Pa}$  a 20 Pa. O som mais intenso que percebemos possui pressão sonora cerca de um milhão de vezes mais alta que o som mais suave. Devido à magnitude desta diferença, a pressão sonora costuma ser tratada através de uma escala logarítmica, dB SPL (Sound Pressure Level, ou nível de pressão sonora). Essa conversão resulta em uma faixa numérica muito menor, que varia de 0 a 120 dB SPL [4].

Para uma pressão sonora de  $p$  Pa, o nível sonoro  $N$  dB SPL é calculado através da Equação 1:

$$N = 20 * \log_{10} \left( \frac{p}{p_0} \right) \quad \text{Equação 1}$$

Onde  $p_0$  é o valor da menor pressão sonora audível, 20  $\mu\text{Pa}$ .

Na Tabela 1 podemos ver o quanto um ganho em dB SPL representa em relação ao aumento da pressão sonora.

Pressão Sonora	Nível de Pressão Sonora
O dobro	+6 dB
Metade	-6 dB
Dez vezes mais	+ 20 dB
Dez vezes menos	-20 dB

*Tabela 1 - Relação entre pressão sonora e nível de pressão sonora<sup>3</sup>*

O valor mínimo de nível de pressão sonora para que se escute um som está próximo de 0 dB SPL, enquanto o valor máximo para que se escute, sem causar danos ao sistema auditivo, está próximo de 120 dB SPL. O nível de pressão sonora de algumas fontes é mostrado na Tabela 2 [5].

Som	Nível de Pressão Sonora (em dB)
Limiar da dor	120
Motosserra	110
Cortador de grama	100
Avenida barulhenta	90
Conversa	60
Música suave	40
Sussurro	20
Limiar da audição	0

*Tabela 2 – Comparação entre diversos sons em Níveis de Pressão Sonora<sup>4</sup>*

Os humanos podem perceber sons com frequências entre 20 Hz e 20 kHz. Sons com frequências entre 500 Hz e 6 kHz requerem um nível de pressão sonora

<sup>3</sup> Dados obtidos de [4]

<sup>4</sup> Adaptado de [5]

menor para serem detectados. A Figura 2 ilustra o limiar de audibilidade conforme a frequência.

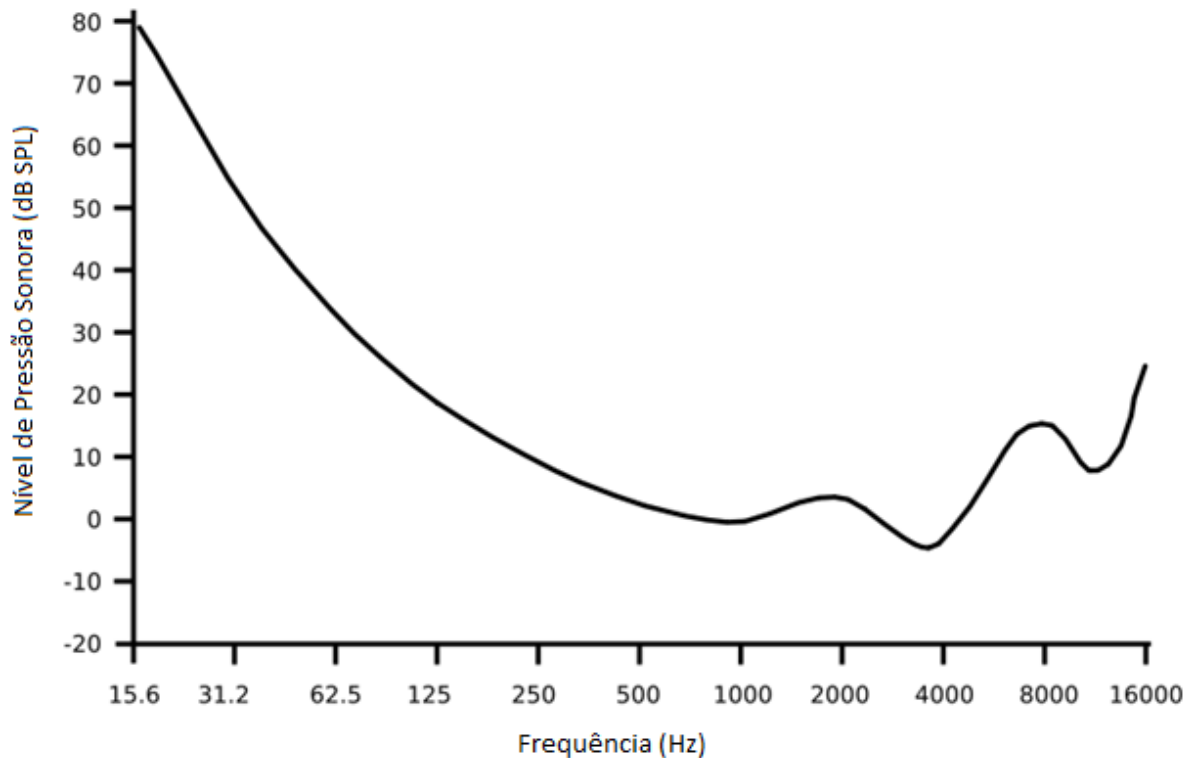


Figura 2 – Limiar de audibilidade<sup>5</sup>

## 2.3 Perda Auditiva

Perda auditiva é a diferença entre o limiar de audição de um indivíduo considerado saudável e de um indivíduo com deficiência auditiva. Quanto maior a perda, maior deve ser o nível de pressão sonora para que se perceba um som. Geralmente é expressa em dB HL, onde HL do significa nível de audição, ou em inglês *Hearing Level* [4].

O exame que avalia a perda auditiva é a audiometria, a mais comum sendo a audiometria de tons puros, que consiste em usar sinais senoidais puros para determinar o limiar da audição em cada frequência. O fonoaudiólogo gradativamente eleva o nível de pressão sonora de cada sinal e quando o paciente passa a escutar

<sup>5</sup> Fonte: <https://en.wikipedia.org/?title=Psychoacoustics>



o som, ele sinaliza ao fonoaudiólogo. As frequências geralmente avaliadas são: 125 Hz, 250 Hz, 500 Hz, 1 kHz, 1.5 kHz, 2 kHz, 3 kHz, 4 kHz, 6 kHz e 8 kHz.

Este exame pode ser realizado através do uso de fones de ouvido, para avaliar a via aérea, ou vibradores ósseos colocados próximos ao ouvido, para avaliar a via óssea. Outros dados podem ser coletados em uma audiometria, como os níveis de pressão sonora que causem desconforto, também conhecido como UCL, ou nível de audição desconfortável, do inglês *Uncomfortable Listening Level*. O resultado deste exame é o audiograma, mostrado na Figura 3.

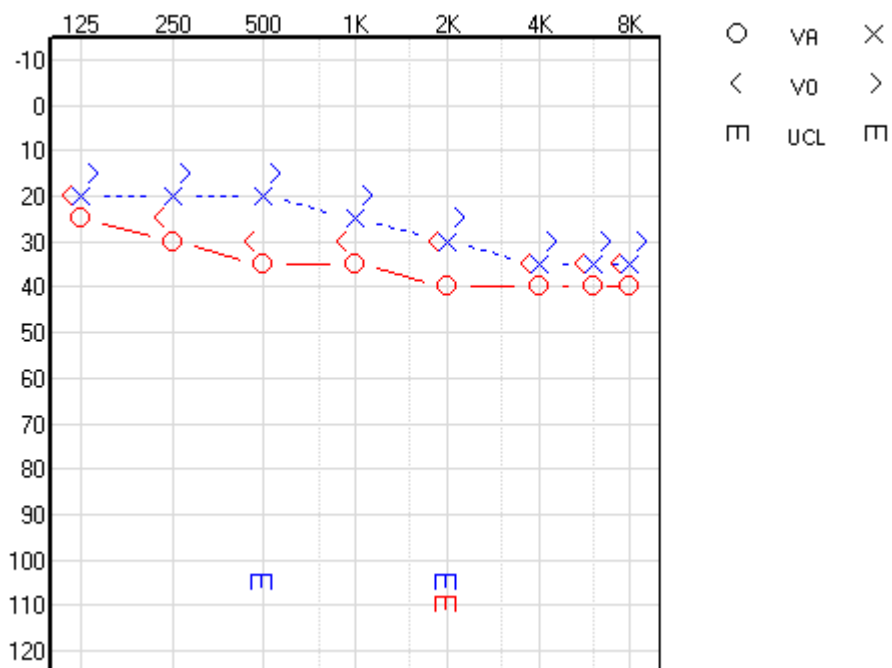


Figura 3– Exemplo de audiograma

O audiograma, por convenção, usa os símbolos de “O” para marcações de perdas na via aérea para o lado direito e “X” para o esquerdo. Para perdas na via óssea, usa-se, o símbolo “<” para o lado direito e “>” para o lado esquerdo. Vale ressaltar que todas as marcações referentes ao lado direito são mostradas em vermelho, enquanto para o lado esquerdo são mostradas em azul.

Porém, nem toda perda auditiva é um sinal de que o indivíduo é um deficiente auditivo. Perdas de até 15 dB HL em crianças e até 25 dB HL são consideradas normais. A Tabela 3 apresenta a classificação das perdas auditivas pela OMS – Organização Mundial da Saúde.

Graus de perda auditiva	Média entre as frequências de 500 Hz, 1 kHz, 2 kHz e 4 kHz (dB HL)		Desempenho
	Criança	Adulto	
Audição normal	0-15	0-25	Nenhuma ou pequena dificuldade. Capaz de ouvir cochichos
Leve	16-30	26-40	Capaz de ouvir e repetir palavras em volume normal a um metro de distância
Moderado	31-60	41-60	Capaz de ouvir e repetir palavras em volume elevado a um metro de distância
Severo	61-80	61-80	Capaz de ouvir palavras em voz gritada próximo à melhor orelha
Profundo	Maior que 81	Maior que 81	Incapaz de ouvir e entender mesmo em voz gritada próximo à melhor orelha

*Tabela 3 – Classificação dos graus de perda auditiva segundo a OMS<sup>6</sup>*

Como tratamento para alguns casos de deficiência auditiva são recomendados aparelhos auditivos, ou até mesmo implantes cocleares.

## **2.4 Aparelhos Auditivos**

A história dos AASI, ou aparelhos auditivos, pode ser dividida em cinco eras: acústica, carbono, tubo de vácuo, transistor e digital [2].

A era acústica teve início quando um indivíduo, pela primeira vez, colocou sua mão atrás da orelha em forma de concha. Este artifício produz um ganho de 5 a 10 dB nas frequências médias e altas, assim como reduz a interferência dos sons

<sup>6</sup> Fonte: [http://www.who.int/pbd/deafness/hearing\\_impairment\\_grades/en/](http://www.who.int/pbd/deafness/hearing_impairment_grades/en/)

provenientes de fontes localizadas atrás do indivíduo. Ao longo do tempo, dispositivos que produzem efeito semelhante foram desenvolvidos, como funis e cornetas.

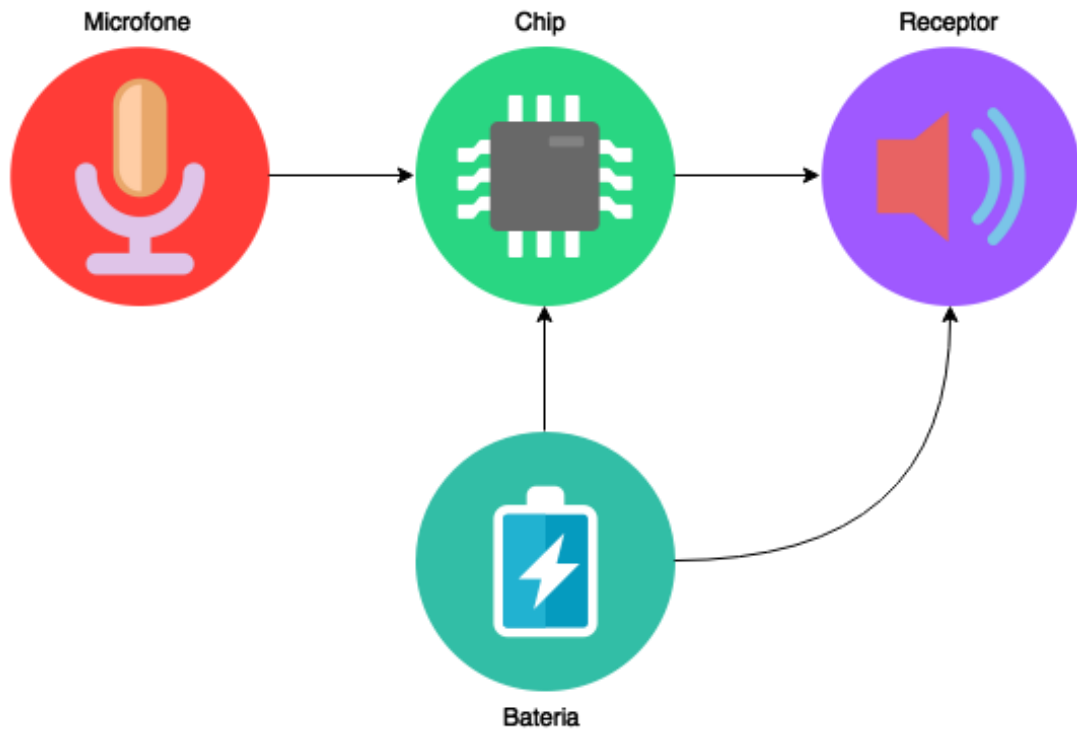
O primeiro aparelho auditivo de carbono foi criado em 1899. Este aparelho consistia em um microfone de carbono, uma bateria e um receptor magnético, que atua como um alto-falante. Neste modelo, a onda sonora é captada pelo diafragma do microfone, que modifica sua resistência. A corrente elétrica que passa por ele produz, então, um campo magnético flutuante que faz com que o diafragma do receptor oscile, produzindo novamente uma onda sonora. Esta onda sonora possui um ganho de 20 a 30 dB em relação à onda original.

O tubo a vácuo eletrônico foi inventado em 1907, mas só foi aplicado em aparelhos auditivos em 1920. Este dispositivo permite que pequenas tensões elétricas provenientes do microfone controlem flutuações em uma grande corrente elétrica. Combinando vários tubos a vácuo é possível criar amplificadores muito poderosos, com ganhos entre 70 dB e saídas de 130 dB SPL.

O transistor, que consome muito menos energia para operar, passou a ser comercializado em 1952. Essa queda no consumo de energia é tamanha que fez com que todos os aparelhos auditivos existentes migrassem para esta tecnologia até 1953. Na década de 1960, o uso de circuitos integrados diminuiu ainda mais o dispositivo. A redução dos componentes e da bateria tornou os aparelhos auditivos menores e permitiu que passassem a ser usados diretamente na cabeça do paciente, o que reduz o ruído provocado pelas vestimentas e pelos cabos. Os dispositivos criados nesse período têm aparência muito semelhante aos aparelhos usados atualmente.

Os aparelhos auditivos digitais convertem as ondas sonoras para representações numéricas, que são manipuladas usando circuitos digitais. Apenas em 1996, estes aparelhos tornaram-se disponíveis comercialmente. A principal vantagem destes dispositivos é que o processamento de sinal passa a não depender mais apenas de fenômenos físicos, mas também do software que está embarcado. O hardware de um aparelho digital simples é composto de bateria, microfone, chip e receptor, como mostrado na Figura 4. Em modelos mais sofisticados podem existir

mais componentes, como um microfone extra, bobina eletromagnética e transmissor *bluetooth*.



*Figura 4 – Componentes básicos de um aparelho auditivo*

Dentre os aparelhos modernos, pode-se classificá-los em relação ao seu tamanho e à posição onde são usados. O aparelho BTE, do inglês *Behind The Ear*, é usado atrás da orelha. A versão miniaturizada do BTE, chamada Mini-BTE, é usada na mesma posição. Existe uma variação dos aparelhos BTE, chamada RITE, ou *Receiver In The Ear*, onde a bateria, o circuito integrado e o microfone localizam-se atrás da orelha e o receptor do aparelho é inserido no canal auditivo. Os aparelhos usados no canal auditivo são chamados ITC, do inglês *In The Canal*. Ainda existem aparelhos que ficam completamente dentro do canal, chamados CIC, do inglês *Completely In Canal*. A Figura 5 ilustra tais modelos.



*Figura 5 – Modelos de aparelhos auditivos<sup>7</sup>*

Sabemos que a audição humana pode detectar sons entre as frequências de 20 a 20.000 Hz, mas informações referentes à fala encontram-se entre as frequências de 100 a 7.000 Hz. Como o maior objetivo do tratamento de um deficiente auditivo é recuperar a inteligibilidade, que é a quantidade de informação transmitida através do som, os aparelhos auditivos geralmente operam entre as frequências de 100 a 8.000 Hz [6].

## **2.5 Regras de Ganho**

Como as perdas auditivas variam em seu grau, configuração e tipo, as características de amplificação do aparelho auditivo devem ser ajustadas para cada paciente. Para definir quais são os ganhos apropriados para cada indivíduo se usam regras de ganho. Tais regras levam em conta as características do indivíduo e de sua perda auditiva para gerar curvas de ganho alvo.

---

<sup>7</sup> Fonte: <http://www.audiologyofnassau.com/types-of-hearing-aids/>

Em 1935, Knudsen e Jones propuseram que os ganhos de um aparelho auditivo deveriam espelhar o audiograma do paciente, ou seja, para cada 1 dB de perda auditiva, o aparelho deveria compensar adicionando 1 dB de ganho na frequência da perda.

Após muitos relatos de que essa regra de ganho produzia ganhos excessivos, Lybarger, em 1944, fez uma observação importante: a maioria dos pacientes preferia ganhos de aproximadamente metade de sua perda auditiva. Tal regra ficou conhecida como a regra do meio ganho.

Desde então muitas regras de ganho foram propostas no intuito de maximizar a inteligibilidade da fala. As mais comuns utilizadas atualmente são a regra NAL, do inglês *National Acoustic Laboratories*, ou Laboratórios Nacionais de Acústica, localizado na Austrália, a regra DSL, do inglês *Desired Sound Level* e a regra FIG6, baseada em um artigo de Killion e Fikret-Pasa [7]. Dessas três regras, NAL e DSL são privadas, enquanto FIG6 é uma regra de padrão aberto e será apresentada a seguir.

### **2.5.1 FIG6**

Proposto em 1993 e nomeado a partir da figura 6 do artigo de Killion e Fikret-Pasa, esse método define algumas fórmulas para normalizar os níveis para entradas leves (40dB), moderadas (65dB) e altas (90dB). Foi desenvolvido observando uma grande amostra de pacientes com níveis de perdas auditivas similares.

Por ser a regra de ganho com fórmulas conhecidas mais popular, a regra FIG6 será usada para gerar as curvas de ganhos alvo usadas neste projeto. Estas fórmulas se encontram na Tabela 4.

Níveis de entrada	Ganho proposto
Alto (90 dB)	0, se $0 < HL < 40$ ; $0,1 \cdot (HL - 40)^{1.4}$ , se $HL \geq 40$
Moderado (65 dB)	0, se $0 < HL < 20$ ; $0,6 \cdot (HL - 20)$ , se $20 \leq HL < 60$ ; $0,8 \cdot HL - 23$ , se $HL \geq 60$
Suave (40 dB)	0, se $0 < HL < 20$ ; $HL - 20$ , se $20 \leq HL < 60$ ; $HL - 20 - 0,5 \cdot (HL - 60)$ , se $HL \geq 60$

*Tabela 4 – Ganhos prescritos pela regra FIG6 [8]*

## 2.6 Algoritmos de Otimização

Algoritmos de otimização são procedimentos bem definidos para resolução de problemas em um número finito de etapas. A solução encontrada pode ser a melhor entre todas as soluções, mas também pode ser uma solução próxima a melhor. Por possuírem procedimentos bem definidos, é possível codificar algoritmos em linguagens de programação e executá-los em um computador.

Os procedimentos destes algoritmos geralmente envolvem a definição das variáveis a serem otimizadas, das restrições impostas pelo problema, da definição da função objetivo e dos limites das variáveis [9].

A formulação de um problema de otimização começa com a identificação das variáveis que são modificadas durante o processo de otimização. Não existe um guia rígido para escolher tais variáveis e estas devem ser identificadas cuidadosamente em cada problema.

As restrições associadas ao problema costumam estar associadas a satisfazer determinadas condições usando uma quantidade limitada de recursos. Nem sempre é preciso existir uma função matemática explícita que defina uma restrição, mas um algoritmo ou mecanismo que calcule tal restrição é obrigatório.

A definição da função objetivo geralmente está associada à maximização ou minimização de um parâmetro, como redução do custo de um processo, maximização do lucro, maximização do ciclo de vida ou outros. Em alguns casos pode-se existir mais de um objetivo a ser otimizado, tornando o problema mais complexo e custoso computacionalmente.

A definição dos limites das variáveis está associada aos valores máximos e mínimos que cada variável pode assumir. Em alguns algoritmos esse limite é usado para confinar o espaço de busca dentro de uma região específica. Nem sempre a determinação desses limites é trivial, pois os valores de algumas variáveis podem interferir no limite de outras.

Existem muitos outros algoritmos de otimização relatados na literatura que poderiam ser usados para resolver o problema de ajuste automático dos parâmetros de um aparelho auditivo. Entretanto o tempo deste projeto é limitado e por este motivo foi possível analisar apenas alguns destes algoritmos.

Este projeto aborda apenas três algoritmos de otimização: *Hill Climbing*, *Simulated Annealing* e Algoritmo Genético. Tais algoritmos pertencem à classe dos algoritmos heurísticos, que não garantem encontrar a melhor solução dentre todas, mas costumam encontrar uma solução próxima a melhor, geralmente de forma rápida e simples.

### **2.6.1 Hill Climbing**

*Hill Climbing*, ou subida da colina, é um algoritmo de busca e otimização simples para solucionar problemas com função objetivo única. Os algoritmos de *Hill Climbing* realizam iterações onde a melhor solução conhecida é usada para gerar uma solução nova. Caso a solução nova seja melhor que sua geradora, ela toma seu lugar e o ciclo recomeça [10].

O maior problema do *Hill Climbing* é sua convergência prematura, o que pode levar a uma solução que é um máximo local, como mostrado na Figura 6. Devido a isto, não é aconselhável o uso deste algoritmo quando existem muitos máximos locais no espaço de soluções. Entretanto, pode-se contornar este problema através



de várias técnicas. A técnica mais comum é reiniciar o algoritmo para uma nova solução aleatória quando um máximo local é atingido.

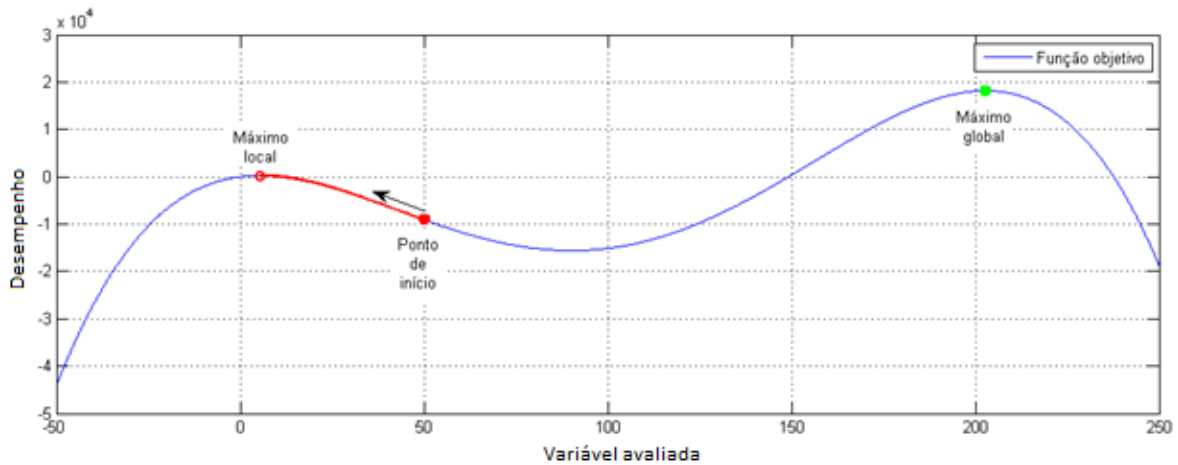


Figura 6 – Máximo local e global

Existem diversas variações deste algoritmo que utilizam heurísticas para melhorar o desempenho do algoritmo. Deve-se avaliar cuidadosamente o problema para determinar qual variação do *Hill Climbing* apresentará melhores resultados.

No caso do ajuste de parâmetros do aparelho auditivo, por existirem muitos parâmetros a serem ajustados e, conseqüentemente, muitos vizinhos para cada solução, optou-se por utilizar a variação *Steepest Ascent*, que em inglês significa “subida mais íngreme”. Nesta variação, todos os vizinhos da solução atual são avaliados a cada iteração. Somente então se escolhe o vizinho com melhor desempenho para continuar o algoritmo. Isto evita que um vizinho com desempenho insignificamente melhor que a solução atual seja escolhido para continuar o algoritmo quando há um vizinho com desempenho muito melhor disponível [11].

### 2.6.2 Simulated Annealing

*Simulated Annealing*, ou recozimento simulado, é uma analogia do processo metalúrgico de recozimento de metais. O recozimento é um tratamento térmico de materiais com o objetivo de alterar suas propriedades físicas. Este processo consiste em elevar a temperatura do metal até um determinado valor e resfriá-lo lentamente. Geralmente o processo de recozimento visa reduzir pequenos defeitos estruturais dos cristais metálicos que compõem o material, elevando tenacidade. Este processo é mostrado na Figura 7.

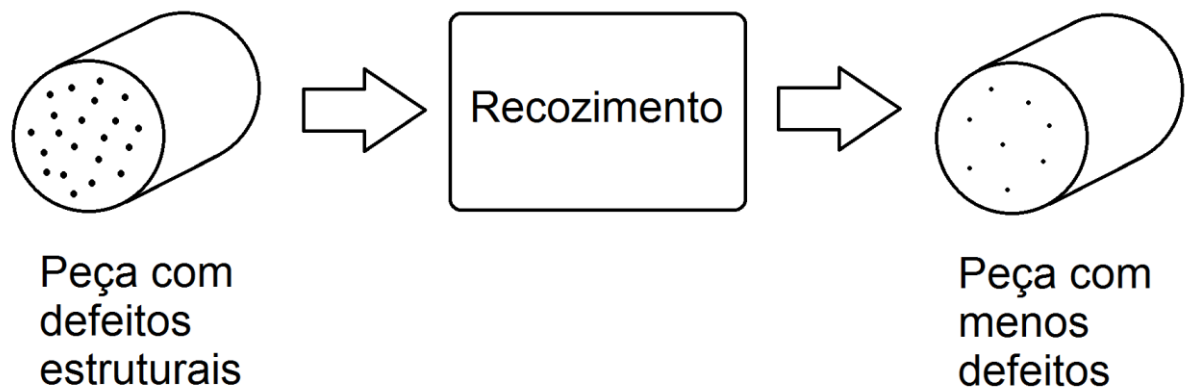


Figura 7 – Redução de defeitos através do recozimento

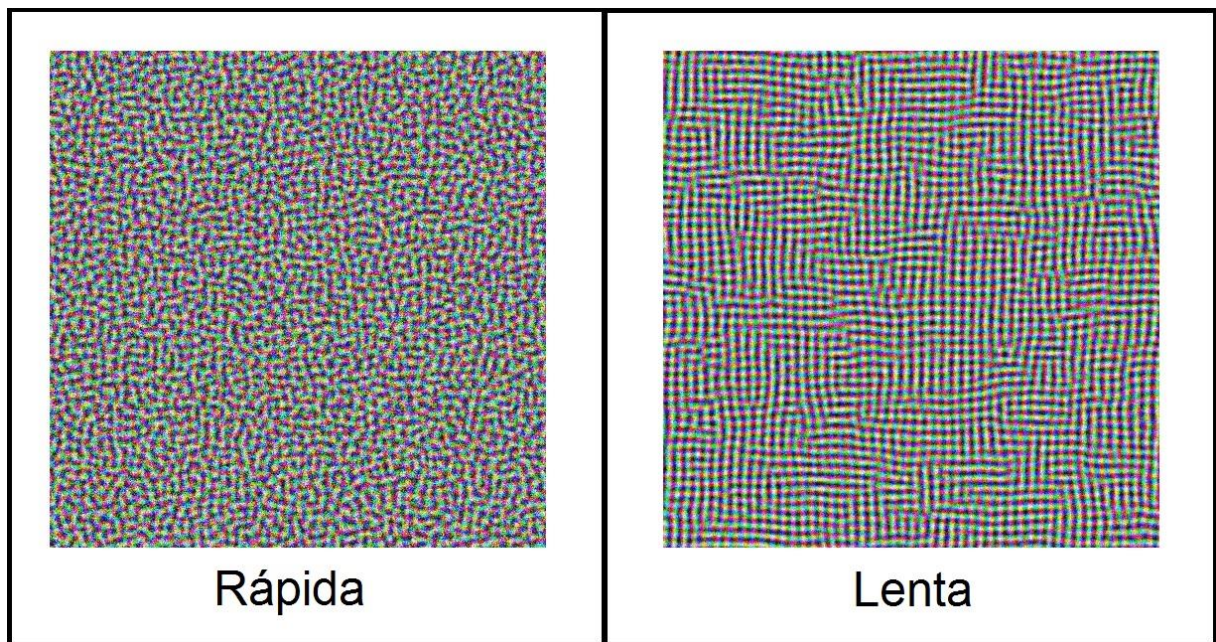
Em 1953, Metropolis [12] desenvolveu um método de Monte Carlo para calcular as propriedades de uma substância composta de moléculas individuais que interagem entre si. Esse método permite simular a maneira como as microestruturas de um metal são reconfiguradas e atingem o equilíbrio ao longo do processo de recozimento.

A pesquisa de Metropolis inspirou Kirkpatrick [13] a desenvolver o algoritmo *Simulated Annealing* para otimização global na década de 1980. A cada iteração deste algoritmo, a próxima solução tem uma chance, de acordo com a temperatura atual, de ir para uma solução pior. À medida que o tempo e as iterações avançam, a temperatura é reduzida e se aproxima de 0 K, o que eleva a probabilidade da nova solução ser melhor que a anterior. Existe uma pequena chance de reiniciar a solução atual para o valor da melhor solução encontrada a cada iteração. Isto evita que o algoritmo fique preso em uma região do espaço de soluções com desempenho pior do que uma região onde ele já encontrou desempenho melhor.

Enquanto a temperatura está elevada, o algoritmo realiza uma busca de forma mais abrangente e aleatória pelo espaço de soluções. Quando a temperatura é reduzida, o algoritmo passa a realizar uma busca em uma região mais restrita de forma similar ao *Hill Climbing*, buscando um ótimo local.

Para qualquer problema, a probabilidade de que o algoritmo *Simulated Annealing* encontre a solução ótima eleva-se quando se reduz a velocidade com que a temperatura é reduzida, prolongando o tempo de processamento. A Figura 8 mostra os resultados deste algoritmo quando a temperatura é reduzida de forma

rápida e quando é reduzida lentamente para a solução do problema de reorganizar pixels de uma imagem no intuito de minimizar certa função, que faz cores similares se atrair quando próximas e se repelir quando afastadas.



*Figura 8 – Resultados para diferentes velocidades de resfriamento no problema de reorganização de pixels*

### **2.6.3 Algoritmo Genético**

Algoritmos genéticos são analogias do processo de seleção natural, onde a evolução ocorre através de mutações, recombinações genéticas e ao fato de que os indivíduos mais bem adaptados ao meio conseguem se reproduzir com maior facilidade, passando suas características para as gerações futuras. Estes algoritmos pertencem à subclasse dos algoritmos evolucionários onde os elementos do espaço de soluções são vetores de binários ou de outros tipos básicos.

As primeiras pesquisas envolvendo algoritmos genéticos iniciaram-se na década de 1950, onde biólogos como Barricelli [14] e o cientista da computação Fraser [15] começaram a simular computacionalmente processos genéticos, evolução e seleção natural.

Tais algoritmos começam com a definição de uma população inicial. A cada iteração, uma nova geração de soluções é criada baseada na geração anterior. Os

indivíduos com melhor avaliação através da função objetivo apresentam melhor adaptação, ou em inglês *fitness*, e por isso, tem maior chance de gerar descendentes.

Cada novo indivíduo gerado ainda está sujeito à recombinação genética e mutação. A recombinação genética, ou em inglês *crossover*, faz com que os novos indivíduos não sejam exatamente iguais a seus pais, mas sim uma mescla das características genéticas de cada um deles. Já a mutação altera parte do indivíduo de forma aleatória, o que pode trazer características genéticas que não estejam presentes na população atual.

A Figura 9 ilustra o funcionamento deste tipo de algoritmo.

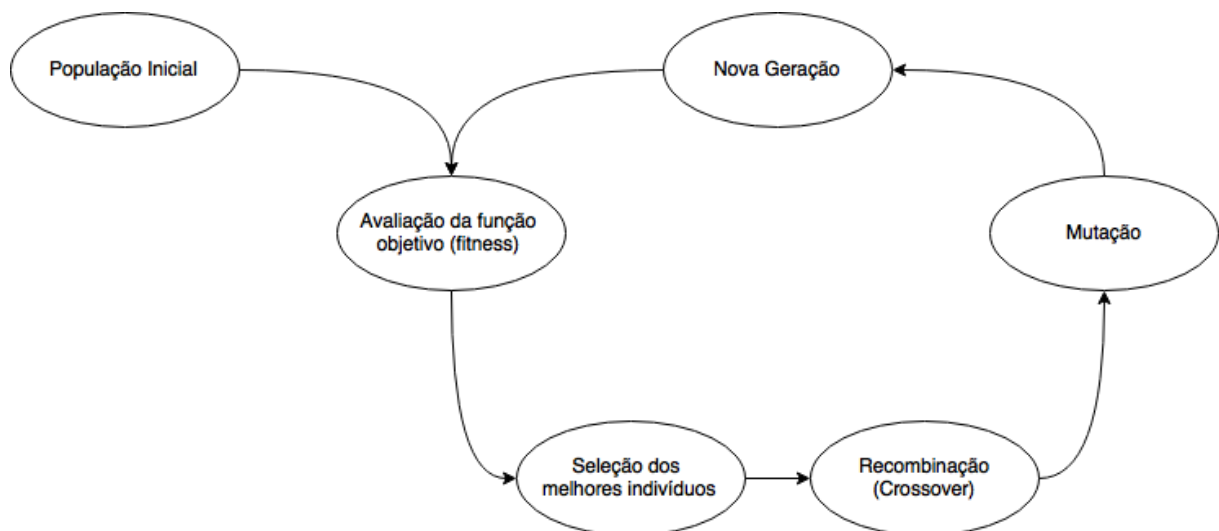


Figura 9 – Ciclo de gerações de um algoritmo genético

## Capítulo 3: WaveFit e Análise do Problema

Neste capítulo, é apresentado o WaveFit, que é o software para ajuste de aparelhos auditivos digitais da WaveTech, e também é definido o problema a ser resolvido neste projeto.

### 3.1 WaveFit

Como o objetivo desta seção é apenas contextualizar o leitor, não são abordados aspectos de implementação. Também se deve ressaltar que parte do que é apresentado nesta seção não foi desenvolvido pelo autor.

O WaveFit é um software de ajuste de aparelhos digitais. Foi concebido para o uso em clínicas de fonoaudiologia, mas pode ser usado também por fonoaudiólogos de forma individual. Dentre as principais funcionalidades do software, destacam-se:

- Cadastro de pacientes;
- Registro das sessões realizadas;
- Cadastro de audiogramas;
- Seleção dos AASI usados;
- Ajuste do AASI;
- Configuração do próprio software, inserindo novos usuários, aparelhos auditivos, chips, microfones e receptores, bem como modificar a interface gráfica do ajuste.

O WaveFit pode ser integrado ao Noah, uma plataforma padronizada internacional que permite o controle centralizado das informações dos pacientes, e o acesso aos softwares de ajuste de cada fabricante de aparelhos auditivos. Quando o WaveFit está integrado ao Noah, o cadastro de pacientes, registro de sessões e cadastro de audiogramas deixam de ser efetuados através do WaveFit e passam a ser realizados através do Noah.

Atualmente o WaveFit dá suporte ao ajuste apenas dos aparelhos fabricados pela WaveTech e pela empresa parceira, Amplivox. Nos dois casos, os aparelhos auditivos possuem chips da empresa OnSemiconductor. Futuramente o software deverá permitir o ajuste de aparelhos com chips de outros fabricantes.

O software possui três níveis de permissão e dois módulos distintos. Os níveis de permissão são: fonoaudiólogo, administrador da clínica e administrador do software. Os módulos são: fonoaudiólogo, onde se realizam ações relacionadas ao atendimento de pacientes, e administração, onde se pode configurar o funcionamento do software.

O usuário com permissão fonoaudiólogo tem acesso apenas ao módulo fonoaudiólogo. O usuário com permissão administrador da clínica também tem acesso à parte do módulo de configuração, onde pode gerenciar usuários e recuperar pacientes removidos. Já o usuário com permissão de administrador do software também tem acesso a todas as funcionalidades do módulo de configuração, como cadastrar novos aparelhos, chips, microfones e receptores.

As interfaces gráficas do software foram projetadas com o objetivo de serem intuitivas, guiando o usuário ao longo do processo de ajuste. As telas do módulo fonoaudiólogo que representam um fluxo normal de uso do software, desde o cadastro de um novo paciente até o fim do ajuste de seus aparelhos auditivos são apresentadas a seguir.

### **3.1.1 Cadastro de Pacientes**

A tela de cadastro de pacientes, mostrada na Figura 10, é por onde se inicia o fluxo de atendimento de pacientes. As etapas deste fluxo são mostradas na barra superior central, onde cada botão representa uma etapa, associada a uma tela diferente.

**Pacientes**

Id	Nome	Sobrenome	Data de Nascimento	Última Consulta	Próxima Consulta
1	Alexandre	Ferreira	01/01/1980		31/08/2015
2	Guillaume	Barrault	01/01/1980		Clique para agendar
3	João Carlos	Silva dos Santos	01/01/1980		Clique para agendar
4	Natan	Grando	15/10/1990		17/09/2015
5	André Vicente	Milack	01/01/1990		17/07/2015

**Formulário**

Nome: Natan  
 Nascimento: 15/10/1990  
 Sobrenome: Grando  
 CPF: 000.000.000-00  
 Fonoaudiólogo: Kátia Almeida  
 RG: 000000000000  
 Médico: Célio Fahl  
 Telefone residencial: (00)0000-0000  
 Plano de saúde: (nenhum)  
 Telefone celular: (00)0000-0000  
 E-mail: natan.grando@wavetech-st.com  
 Telefone comercial: ( ) - -  
 Gênero:  Masculino  Feminino  Outro  
 Fax: ( ) - -  
 Cidade: Florianópolis  
 Observações:  
 Estado: Santa Catarina  
 Possui histórico de perda auditiva na família.  
 País: Brasil  
 Endereço: SC 401, Km 01  
 Complemento:  
 CEP: 88055-480

**Histórico**

02/07/2015

Audiograma 08:46

Orelha Direita

Orelha Esquerda

Ajuste 08:49 AASI Esquerdo: Wave Mini AASI Direito: Wave BTE

Figura 10 – WaveFit: tela de cadastro de pacientes

Nesta etapa é possível inserir um novo paciente, visualizar ou editar as informações de um paciente existente, visualizar uma prévia dos audiogramas cadastrados e dos ajustes realizados em ordem cronológica.

### 3.1.2 Cadastro de Audiograma

O segundo passo do fluxo do atendimento de um paciente é cadastrar um audiograma que represente suas perdas auditivas, como visto na Figura 11. É possível cadastrar uma curva de perda na via aérea – VA, perda na via óssea – VO e nível de audição desconfortável – UCL.

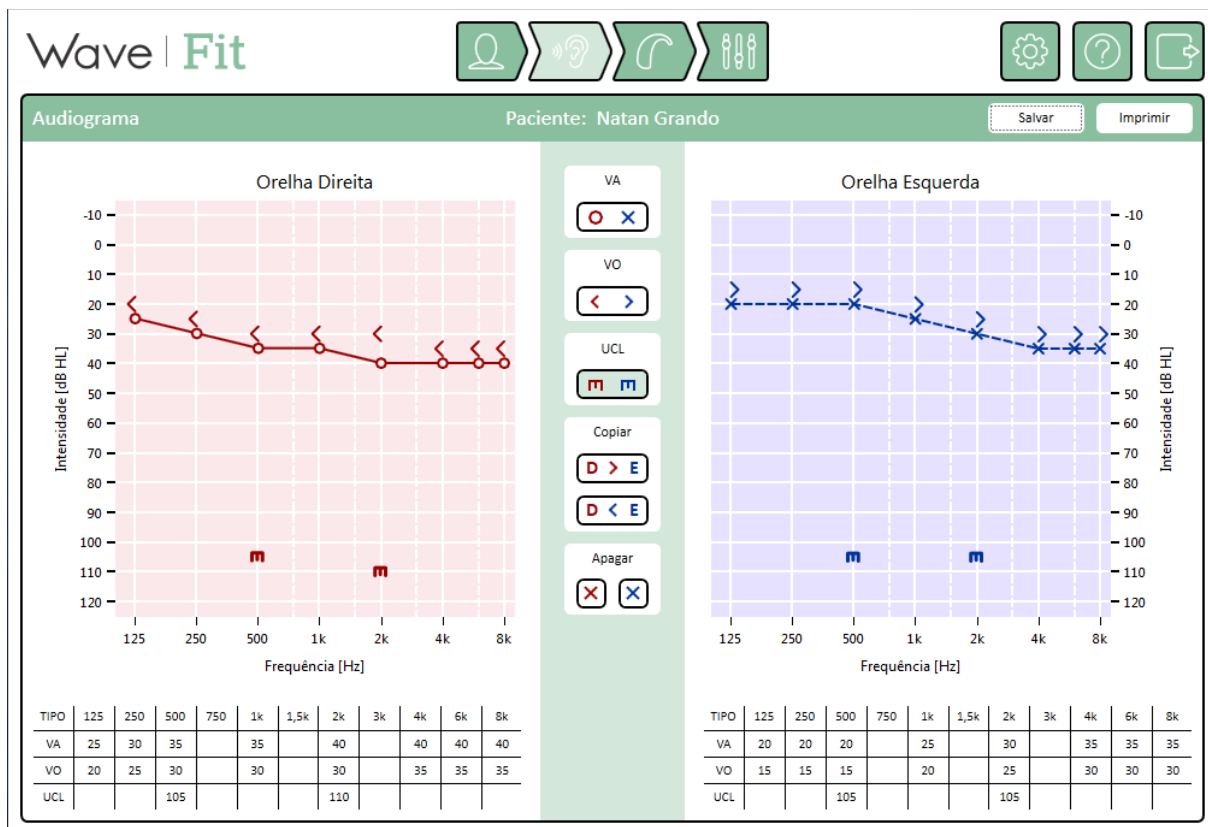


Figura 11 – WaveFit: tela de audiograma

Por convenção, o audiograma correspondente ao lado direito é apresentado na cor vermelha e é mostrado no lado esquerdo da tela, já o audiograma correspondente ao lado esquerdo é apresentado na cor azul e é mostrado no lado direito da tela.

### 3.1.3 Seleção de Aparelho Auditivo

A terceira etapa é a seleção dos aparelhos auditivos usados pelo paciente, como mostra a Figura 12. Em um caso real, se os aparelhos estiverem conectados a uma programadora, esta seleção ocorre de forma automática. Porém, o fonoaudiólogo deve informar qual é o receptor usado, caso este não seja o receptor padrão do aparelho. O fonoaudiólogo também deve informar qual é o diâmetro da ventilação do molde usado.



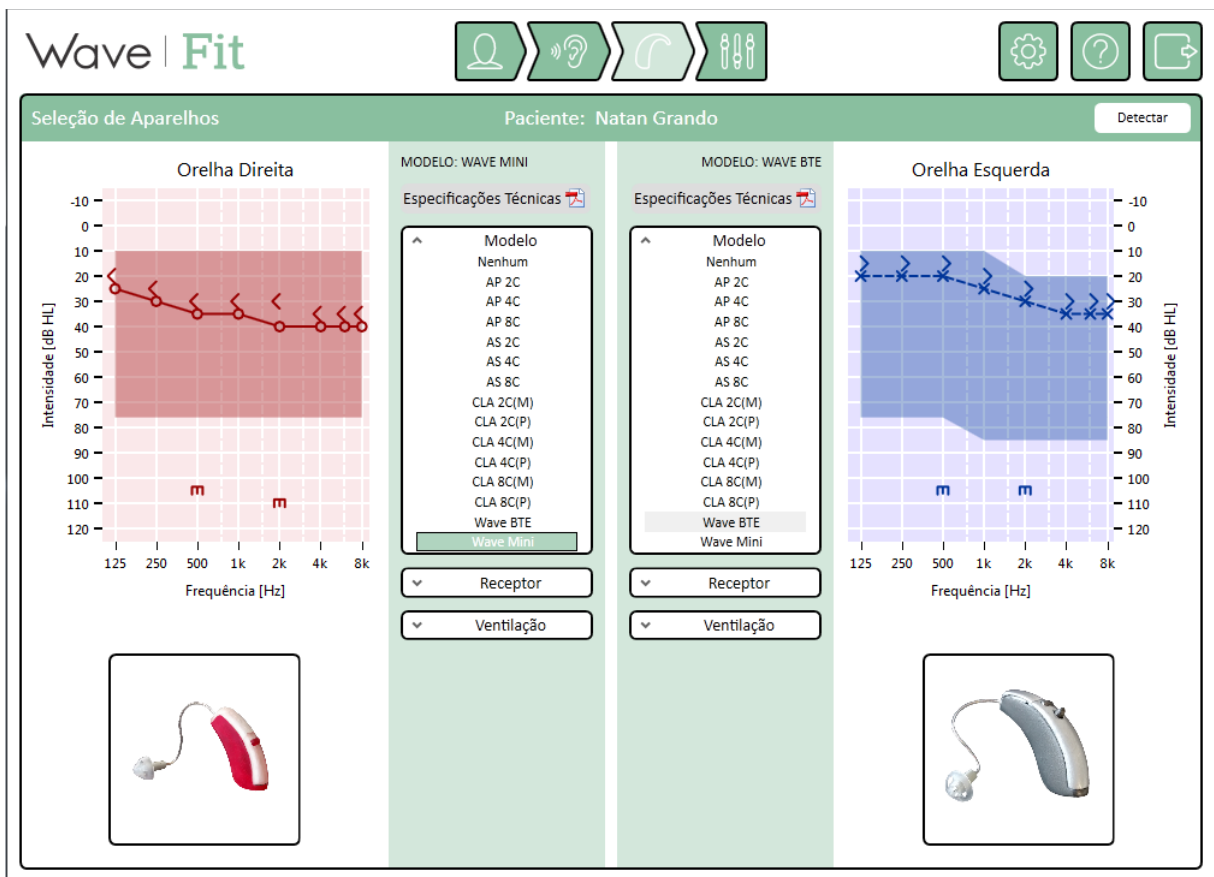


Figura 12 – WaveFit: tela de seleção de aparelhos auditivos

O software também permite executar uma simulação de ajuste onde é possível escolher manualmente qual aparelho se deseja ajustar.

### 3.1.4 Ajuste dos Parâmetros

A última etapa do processo é o ajuste dos parâmetros do aparelho auditivo. Na tela exibida na Figura 13, são mostradas as curvas de resposta dos aparelhos auditivos no domínio da frequência ou na forma entrada/saída. No mesmo gráfico, ainda são mostradas as curvas prescritas pela regra de ganho selecionada. As abas que contém os parâmetros do aparelho auditivo estão localizadas abaixo das curvas de resposta. A disposição e o tipo dos controles que permitem modificar o valor dos parâmetros podem ser configurados através do próprio software.

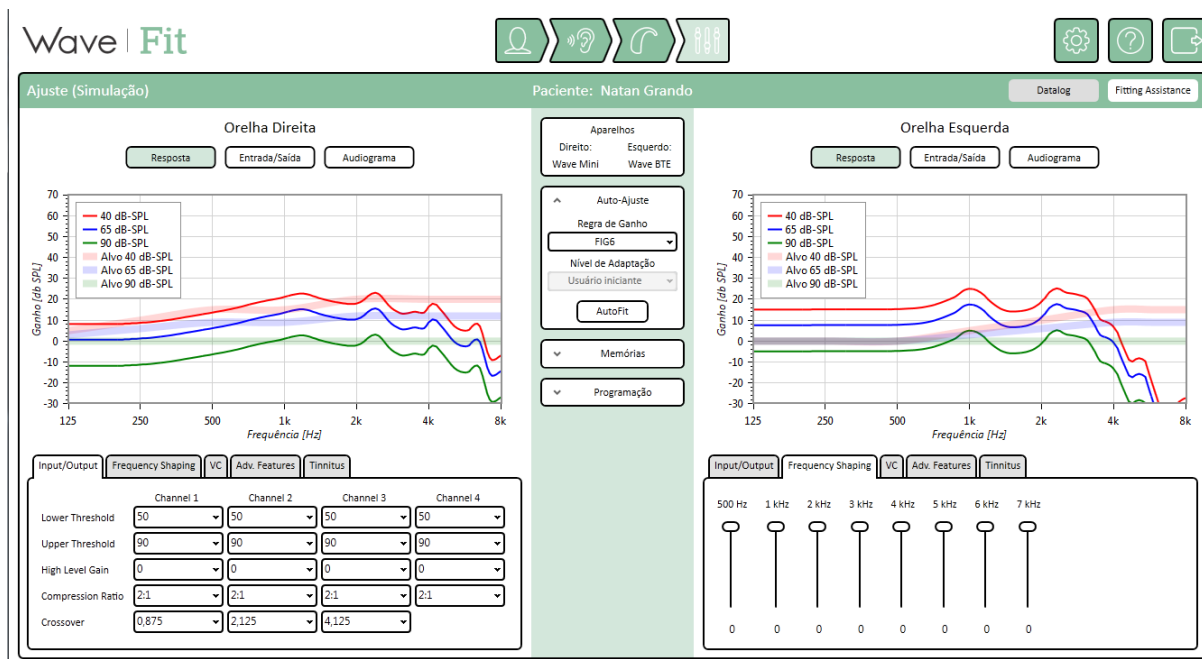


Figura 13 – WaveFit: tela de ajuste dos parâmetros

Nesta tela também é apresentada a função de *Fitting Assistance*, que auxilia o fonoaudiólogo a traduzir queixas dos pacientes em modificações dos parâmetros do aparelho auditivo no intuito de resolver o problema. Caso exista mais de uma solução para resolver um mesmo problema todas elas são apresentadas ao fonoaudiólogo, mesmo que seja uma solução que não envolva o ajuste de nenhum parâmetro, como, por exemplo, remover o cerume presente no molde.

No menu central, encontram-se, além dos botões que gravam a configuração de parâmetros no aparelho auditivo e que retornam os parâmetros para sua configuração original, um submenu que permite o ajuste automático dos parâmetros. Neste submenu é possível selecionar qual regra de ganho se deseja aplicar, o que fará com que as curvas de ganho prescrito sejam mostradas em cada um dos gráficos de resposta em frequência. Ainda neste submenu, encontra-se o botão de auto ajuste, que será o foco deste trabalho.

### 3.2 Análise do Problema

O problema que este projeto visa resolver é a otimização da configuração dos valores dos parâmetros de um aparelho auditivo genérico visando uma curva de ganho real o mais próximo possível à curva de ganho prescrita de acordo com a

regra de ganho selecionada. Esta solução será usada para realizar o ajuste automático no software WaveFit.

O objetivo de realizar um ajuste automatizado é acelerar o processo de ajuste de aparelhos auditivos e também auxiliar o fonoaudiólogo a realizar ajustes mais precisos que resultem em melhores resultados para o tratamento do paciente.

A tarefa de ajustar um aparelho auditivo não é trivial, pois os parâmetros que são ajustados variam de acordo com cada chip, sendo que alguns parâmetros respondem de uma forma para um dado chip, e de forma diferente em outro chip. Devido a este fato, é muito custoso estabelecer uma relação matemática entre o valor de um dado parâmetro e o impacto causado na resposta em frequência. A modificação de um parâmetro com o mesmo nome para um mesmo valor em dois chips distintos pode levar a resultados diferentes.

Existem dois requisitos que devem ser atendidos pelo algoritmo que será implementado no software. O primeiro é minimizar o erro entre as curvas de ganho prescritas e as curvas de ganhos reais no domínio da frequência. O segundo é realizar o ajuste em um tempo que seja pequeno o suficiente para que o fonoaudiólogo e o paciente não se incomodem.

O ajuste automático não precisa encontrar a configuração ótima global para os parâmetros do chip, mas uma configuração que seja suficientemente boa para que o fonoaudiólogo possa realizar um ajuste fino a fim de maximizar a inteligibilidade da fala reconhecida pelo paciente.

Em relação ao critério temporal, o fonoaudiólogo e o paciente não devem ter de esperar muito tempo para obter uma configuração através do ajuste automático. Este tempo não deve ser maior que 5 segundos para que se realize o ajuste de cada aparelho auditivo.

A configuração de parâmetros do aparelho auditivo encontrada pelo algoritmo proposto não deve ser tomada como solução ideal, pois cabe ao fonoaudiólogo avaliar se a configuração está correta e, caso não esteja, realizar ajustes para corrigi-la.

## Capítulo 4: Desenvolvimento dos Algoritmos de Otimização

Para resolver o problema apresentado no Capítulo 3 existem diversos algoritmos propostos na literatura acadêmica. Infelizmente não há tempo suficiente para que todos eles sejam avaliados durante este projeto. Por esta razão este projeto aborda apenas os algoritmos *Hill Climbing*, *Simulated Annealing* e algoritmo genético. Neste capítulo serão apresentados os desenvolvimentos e os resultados de cada um destes.

Para que todos os algoritmos possam ser comparados, a função objetivo é a mesma para os três. Esta função é dada pelo erro quadrático entre pontos em determinadas frequências nas curvas de resposta do aparelho e curvas prescritas pela regra de ganho FIG6. As curvas de resposta são avaliadas para as intensidades sonoras de entrada de 40, 65 e 95 dB SPL.

A escolha de calcular o erro apenas em alguns pontos foi tomada para reduzir o custo computacional da operação de avaliação do valor da função objetivo. O erro será calculado em 28 frequências distribuídas com um espaçamento logarítmico entre 125 Hz e 8 kHz, que é a faixa de operação da maioria dos aparelhos auditivos. O número de pontos em que se calcula o erro foi definido de forma empírica, procurando evitar subamostragem e superamostragem da curva de resposta em frequência.

A função objetivo é apresentada através da Equação 2:

$$FO = \sum_{s=1}^3 \sum_{f=1}^{28} (Gp(s, f) - Gr(s, f))^2 \quad \text{Equação 2}$$

Onde  $FO$  é a função objetivo,  $s$  é o índice da intensidade sonora de entrada,  $f$  é o índice da lista de frequências onde é calculado o erro,  $Gp$  o ganho prescrito e  $Gr$  o ganho real com a configuração atual dos parâmetros.

Nos três algoritmos, as variáveis a serem otimizadas são os valores dos parâmetros do chip de um aparelho auditivo. Um chip simples geralmente conta com 2 canais de áudio possui cerca de 60 parâmetros, enquanto um chip mais complexo pode possuir até 8 canais de áudio e 200 parâmetros ajustáveis.

Um exemplo de um chip complexo é o SA3286, da Onsemiconductor, que possui 8 canais de áudio e 173 parâmetros ajustáveis. O total de combinações possíveis de parâmetros para este chip é de  $2,7185e+188$ .

Os limites que cada variável pode assumir são dados pelos valores máximos e mínimos de cada parâmetro e também pelas restrições internas, que não permitem determinadas configurações de parâmetros. Tais restrições internas são checadas pela DLL – *Dynamic-link library*, ou biblioteca de vínculo dinâmico, fornecida pelo fabricante dos chips.

Todos os algoritmos apresentados foram implementados em C#, por ser a mesma linguagem usada pelo software WaveFit. Mesmo já existindo bibliotecas prontas que implementem estes métodos, o autor decidiu realizar sua própria implementação para poder analisar o comportamento de cada algoritmo, e também para adotar estratégias que melhorem seu desempenho para a resolução do problema em questão.

Para que se tenha uma métrica similar na avaliação dos resultados de cada algoritmo, todos eles serão avaliados em relação ao ajuste do chip SA3286. Ainda serão avaliados usando as mesmas curvas de ganho prescrito, geradas a partir dos audiogramas 1, 2, 3, 4 e 5 através da regra FIG6. Os audiogramas são mostrados na Tabela 5 e representam deficiências auditivas comuns.

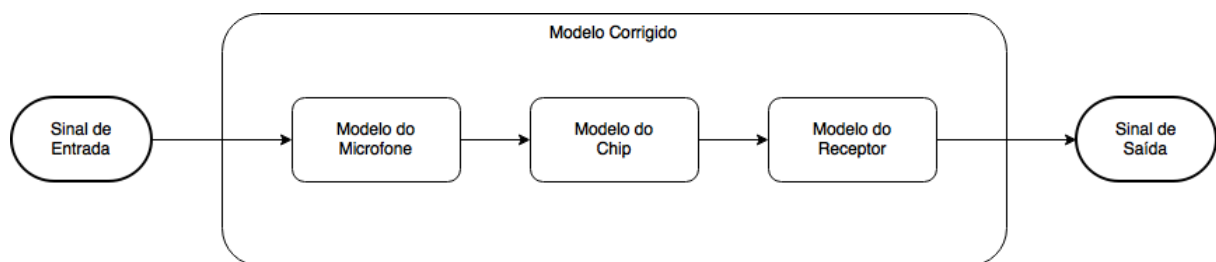
Audiograma	125 Hz	50 Hz	500 Hz	750 Hz	1000 Hz	1500 Hz	2000 Hz	3000 Hz	4000 Hz	6000 Hz	8000 Hz
1	30	30	35	35	35	40	45	50	50	55	55
2	35	35	35	40	50	50	45	40	40	40	40
3	40	40	40	40	40	40	40	40	40	40	40
4	50	50	50	45	45	40	40	35	35	30	30
5	30	30	40	45	45	45	45	45	45	50	50

*Tabela 5 – Perdas auditivas de via aérea em dB SPL de cada audiograma usado para avaliar o desempenho dos algoritmos*

## 4.1 Correção das Curvas de Resposta em Frequência

O ajuste dos parâmetros é baseado na curva de resposta em frequência do aparelho auditivo. Tais curvas são calculadas a partir de um modelo embutido na DLL de cada chip. Porém este modelo apresenta respostas muito diferentes das curvas que são obtidas através do AURICAL Plus<sup>8</sup>, que testa aparelhos auditivos. O erro entre as curvas geradas pelo modelo fornecido pelo fabricante e as curvas obtidas através do AURICAL Plus é de até  $\pm 20$  dB SPL com um intervalo de confiança de 95%..

Para que as curvas usadas sejam fidedignas, além do uso do modelo do chip, também se utiliza um modelo do microfone e outro do receptor. O diagrama de blocos da Figura 14 representa o modelo corrigido para gerar curvas de resposta em frequência.



*Figura 14 – Diagrama do modelo para gerar curvas de resposta em frequência*

O erro entre as curvas geradas pelo novo modelo e as curvas obtidas através do AURICAL Plus é de até  $\pm 5$  dB SPL com um intervalo de confiança de 95%.

A Figura 15 mostra uma resposta dada pelo WaveFit usando o algoritmo corrigido e a Figura 16 mostra a resposta do mesmo aparelho auditivo com a mesma configuração de parâmetros obtida através do Aurical Plus. A curva gerada através do modelo corrigido é muito parecida com a curva obtida pelo Aurical Plus.

---

<sup>8</sup> AURICAL Plus: <http://www.otometrics.com/hearing-aid-fitting/aurical-plus>

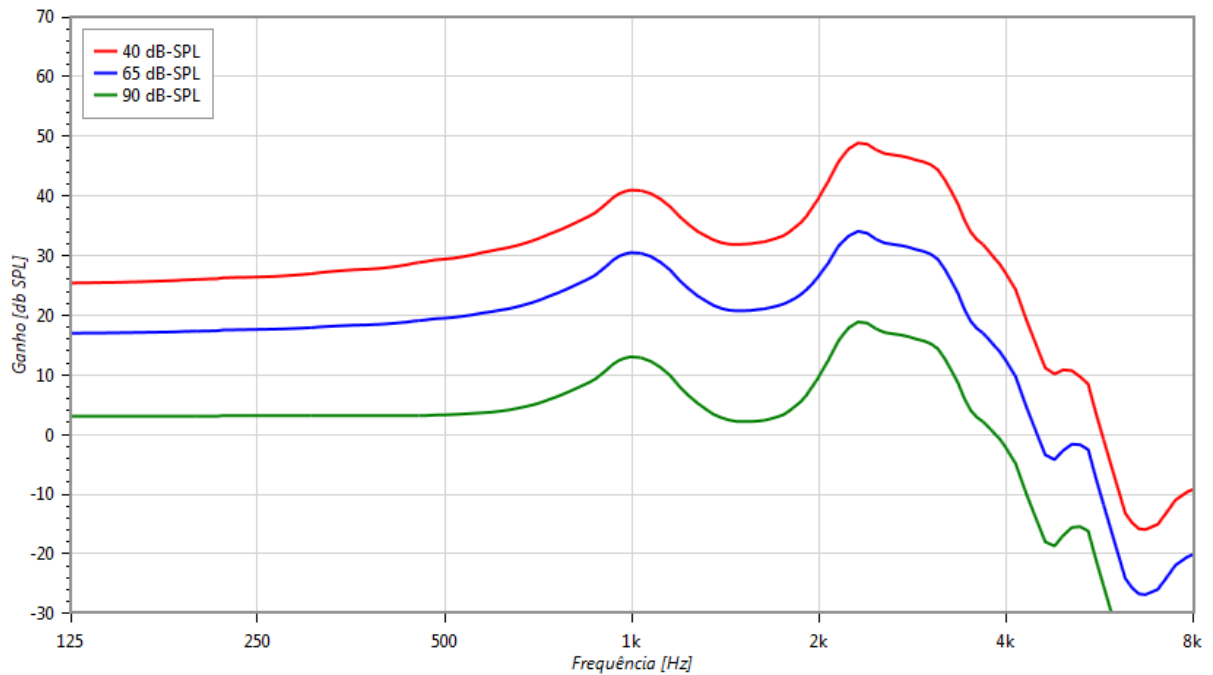


Figura 15 – Curva de ganhos em frequência gerada usando o modelo corrigido no WaveFit.

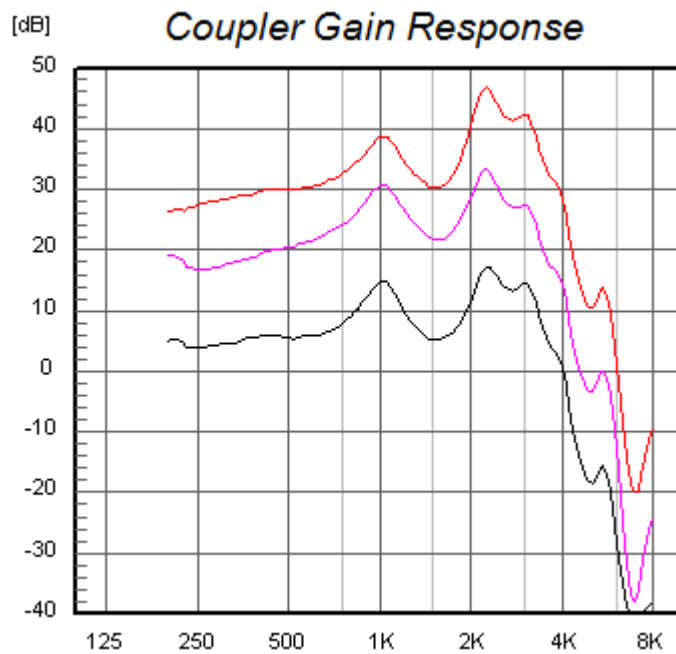


Figura 16 – Curva de ganhos em frequência obtida através do Aurical Plus

Para obter resultados melhores, todos os algoritmos usam este modelo corrigido para gerar as curvas de resposta em frequência do aparelho auditivo durante o cálculo da função objetivo.

## 4.2 Hill Climbing

O algoritmo de *Hill Climbing* foi a primeira opção para resolver o problema de ajuste automático dos parâmetros por ser um algoritmo simples e de convergência rápida. O algoritmo avalia as soluções vizinhas da solução atual procurando uma solução melhor a cada iteração. Os vizinhos são definidos por aquelas soluções com apenas um parâmetro sendo variado para cima ou para baixo em uma unidade em relação à solução atual.

Como podem existir muitos vizinhos para cada solução, a variação *Hill Climbing* com *Steepest Ascent* foi escolhida, pois fará com que o algoritmo ignore vizinhos com melhorias pequenas quando existem vizinhos que apresentem melhorias maiores.

O valor de configuração padrão dos parâmetros foi definido como solução inicial para o algoritmo de *Hill Climbing*, pois esta configuração normalmente não se encontra muito distante da solução ideal para cada paciente.

O algoritmo inicia seu processo calculando o valor da função objetivo para o indivíduo inicial. Em seguida são calculados os vizinhos da solução e qual o valor da função fitness para cada vizinho. Caso o vizinho possua uma configuração inválida de parâmetros, ele é descartado.

Caso o vizinho com a melhor avaliação através da função objetivo for melhor que a solução atual, o algoritmo é reiniciado tomando o melhor vizinho como solução atual. Caso nenhum vizinho possua uma avaliação melhor que a solução atual, uma região plana ou um mínimo local é atingido e o algoritmo é encerrado.

O pseudocódigo simplificado deste algoritmo é apresentado a seguir:



```

encontraSolucao () {
    solucaoAtual = valorPadrao;
    fitnessAtual = calculaFitness (solucaoAtual);
    otimoLocalEncontrado = false;
    while (!otimoLocalEncontrado) {
        solucoesVizinhas[] = calculaSolucoesVizinhas (solucaoAtual);
        if (solucoesVizinhas[].tamanho () == 0) {
            otimoLocalEncontrado = true;
            break;
        }
        melhorFitnessNosVizinhos = +inf;
        melhorVizinho = solucoesVizinhas[0];
        foreach(solucaoVizinha em solucoesVizinhas[]) {
            fitnessDoVizinho = calculaFitness (solucaoVizinha);
            if (fitnessDoVizinho < melhorFitnessNosVizinhos){
                melhorFitnessNosVizinhos = fitnessDoVizinho;
                melhorVizinho = solucaoVizinha;
            }
        }
        if (melhorFitnessNosVizinhos < fitnessAtual) {
            solucaoAtual = melhorVizinho;
            fitnessAtual = melhorFitnessNosVizinhos;
        }
        else {
            otimoLocalEncontrado = true;
        }
    }
}

```

#### 4.2.1 Resultados

O método de auto ajuste utilizando este algoritmo foi repetido para as cinco curvas de ganhos prescrito, geradas a partir dos cinco audiogramas mostrados na Tabela 5 através da regra de ganho FIG6. O aparelho auditivo que foi ajustado possui o chip SA3286. O computador em que se realizaram os testes de ajuste possui um processador i5-3330 e 8 GB de memória RAM. Os resultados do teste são apresentados na Tabela 6.

Audiograma	Erro quadrático (dB SPL <sup>2</sup> )	Tempo de execução (segundos)	Número de iterações	Número de vezes que se avaliou a função objetivo
Audiograma 1	4159,83	122,12	103	33372
Audiograma 2	3839,46	117,86	99	32897
Audiograma 3	4232,76	119,23	101	33052
Audiograma 4	3923,02	125,38	104	33435
Audiograma 5	4377,14	126,41	106	33622
Média	4106,442	122,20	102,6	33275,6

Tabela 6 – Desempenho do algoritmo de Hill Climbing para audiogramas distintos.

A resposta em frequência para o ajuste realizado para o Audiograma 1 utilizando o algoritmo *Hill Climbing*, pode ser vista na Figura 17.

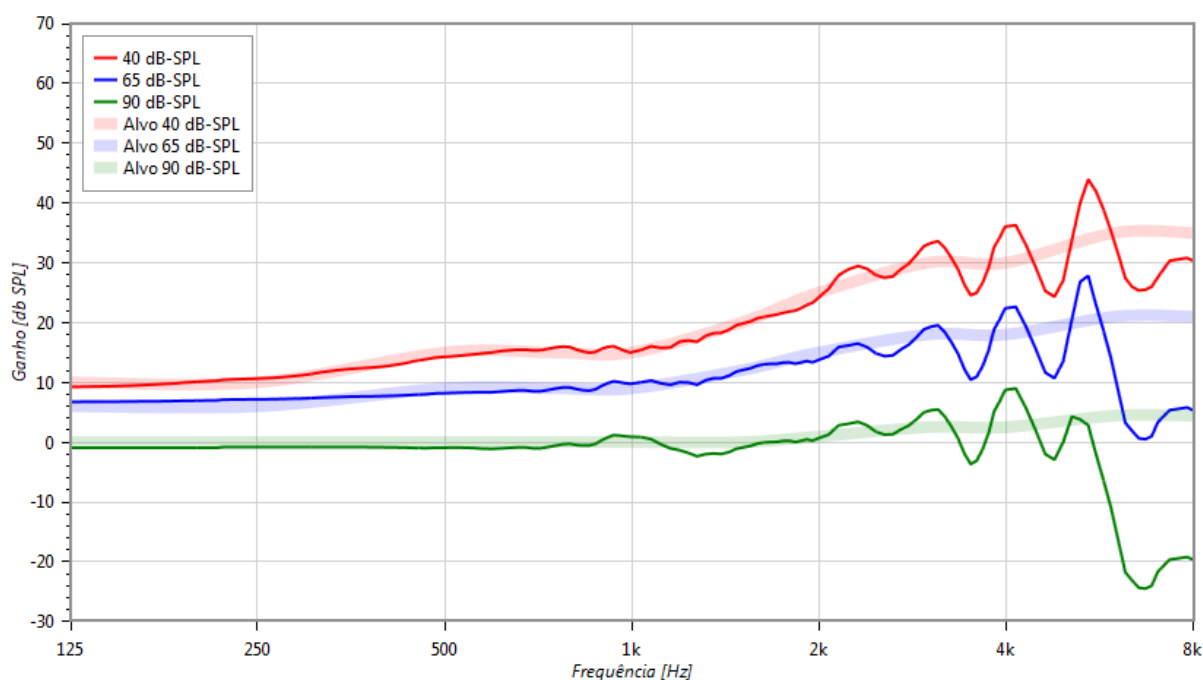


Figura 17 – Resposta em frequência do ajuste para o Audiograma 1 utilizando o algoritmo *Hill Climbing*

O algoritmo conseguiu reduzir o erro entre as curvas de forma satisfatória, mas o processo levou muito tempo e por isso o algoritmo não foi aprovado para a implementação no WaveFit.

Existem algumas técnicas na literatura que poderiam ser aplicadas ao *Hill Climbing* para tentar reduzir o tempo de processamento. Este é o caso do *Hill Climbing* com reinício aleatório. Tal técnica consiste em reiniciar o algoritmo para soluções aleatórias, pois frequentemente é melhor gastar o tempo de processamento explorando o espaço de soluções do que otimizar cuidadosamente uma condição inicial. Infelizmente esta solução não pode ser avaliada pois não houve tempo suficiente para isto.

Outra técnica para reduzir o tempo de processamento do *Hill Climbing* é apresentada no Capítulo 5.

### **4.3 Simulated Annealing**

O *Simulated Annealing* é muito parecido com o *Hill Climbing*. Devido a isto, a escolha da solução inicial e o cálculo para gerar novos vizinhos serão os mesmos do algoritmo anterior.

Este algoritmo também se inicia com a escolha da solução inicial e o cálculo da função objetivo para esta solução. A melhor solução é iniciada com o valor do indivíduo inicial. O valor do melhor fitness é iniciado com o valor do fitness do indivíduo inicial. A temperatura é iniciada no valor máximo de 1 K.

A cada iteração, um vizinho aleatório é escolhido e seu valor objetivo é calculado. Se seu valor objetivo for pior do que o da solução atual, ainda assim existirá uma chance desta solução ser aceita. Esta chance está ligada ao valor da temperatura atual. Quanto maior a temperatura, maior é a chance. Caso a solução não seja aceita, um novo vizinho aleatório é gerado e o processo de aceitação é reiniciado.

No final de cada iteração, a temperatura é reduzida, o que aumenta a chance da próxima solução apresentar um fitness melhor que seu antecessor e, conseqüentemente, aumenta a convergência do algoritmo. Caso o fitness da nova solução seja melhor que a melhor solução encontrada até agora, ela e seu valor fitness são atribuídos aos melhores valores encontrados.

Ainda existe uma pequena chance de reiniciar o valor da solução atual para o valor da melhor solução encontrada. Isto evita que o algoritmo fique preso em um

ótimo local, quando já tenha explorado uma região que apresenta resultados melhores.

O algoritmo é encerrado quando a temperatura chegar a 0,0001 K ou o tempo limite estiver esgotado.

O pseudocódigo simplificado deste algoritmo é apresentado a seguir:

```
encontraSolucao () {
    solucaoAtual = valorPadrao;
    fitnessAtual = calculaFitness (solucaoAtual);
    melhorSolucao = solucaoAtual;
    melhorFitness = fitnessAtual;
    temperatura = temperaturaMaxima;
    while (temperatura > 0.0001 && tempoAtual < tempoMaximo) {
        vizinhoAceito = false;
        while (!vizinhoAceito) {
            vizinho = geraVizinhoAleatorio (solucaoAtual);
            fitnessDoVizinho = calculaFitness (vizinho);
            vizinhoAceito = aceitaVizinho (fitnessAtual, fitnessDoVizinho, temperatura);
        }
        solucaoAtual = vizinho;
        fitnessAtual = fitnessDoVizinho;
        if (fitnessAtual < melhorFitness) {
            melhorSolucao = solucaoAtual;
            melhorFitness = fitnessAtual;
        }
        elseif (gerarAleatorio < chanceDeReiniciar) {
            solucaoAtual = melhorSolucao;
            fitnessAtual = melhorFitness;
        }
        temperatura = temperatura*alpha;
    }
}
```

Neste algoritmo pode-se modificar o valor de alpha, que reduz a temperatura a cada iteração e a chance de reiniciar a solução. Estes parâmetros foram definidos analisando a eficiência do algoritmo para diferentes valores. As análises foram feitas comparando os resultados de dez simulações distintas variando o valor de cada parâmetro.

Os resultados para o valor de alpha são mostrados na Tabela 7. Nesta tabela podemos observar que um alpha maior acarreta um maior tempo de execução e também um erro quadrático menor.

Valor de alpha	Erro quadrático (dB SPL <sup>2</sup> )	Tempo de execução (segundos)	Iterações
0,99	38677,84	6,97	688
0,995	35856,12	13,51	1379
0,999	26096,47	42,36	6905

*Tabela 7 – Valores de alpha para o algoritmo de Simulated Annealing<sup>9</sup>*

Devido ao erro ser menor com o alpha fixado em 0.999, este foi o valor escolhido para o parâmetro.

Com o valor de alpha fixado, avaliou-se qual o melhor valor para a chance de reiniciar. Os resultados são apresentados na Tabela 8. O valor escolhido para o parâmetro foi 0,05, pois foi o valor que mais reduziu o erro quadrático.

Chance de reiniciar	Erro quadrático (dB SPL <sup>2</sup> )	Tempo de execução (segundos)	Iterações
0,01	31448,33	41,25	6905
0,05	26096,47	42,36	6905
0,10	27612,28	43,08	6905

*Tabela 8 – Comparação entre diferentes valores para a chance de reiniciar o algoritmo de Simulated Annealing<sup>10</sup>*

#### **4.3.1 Resultados**

O método de auto ajuste utilizando este algoritmo foi repetido dez vezes para cada uma das cinco curvas de ganhos prescritos distintas utilizadas no *Hill Climbing*, pois existe um elemento randômico neste algoritmo. O aparelho auditivo ajustado foi o mesmo, assim como o computador onde se realizaram os testes. Os resultados do teste são apresentados na Tabela 9.

<sup>9</sup> Chance de reiniciar fixada em 0.05

<sup>10</sup> Valor de alpha fixado em 0.999

Audiograma	Erro quadrático (dB SPL <sup>2</sup> )	Tempo de execução (segundos)	Número de iterações	Número de vezes que se avaliou a função objetivo
Audiograma 1	28103,12	42,12	6905	7321,25
Audiograma 2	25543,94	42,85	6905	7356,75
Audiograma 3	32389,18	42,33	6905	7352,00
Audiograma 4	27443,51	41,78	6905	7286,25
Audiograma 5	17849,77	42,21	6905	7308,00
Média	26265,904	42,258	6905	7324,85

Tabela 9 – Desempenho do algoritmo de Simulated Annealing para audiogramas distintos.

A resposta em frequência para o ajuste realizado para o Audiograma 1 utilizando o algoritmo *Simulated Annealing* pode ser vista na Figura 18.

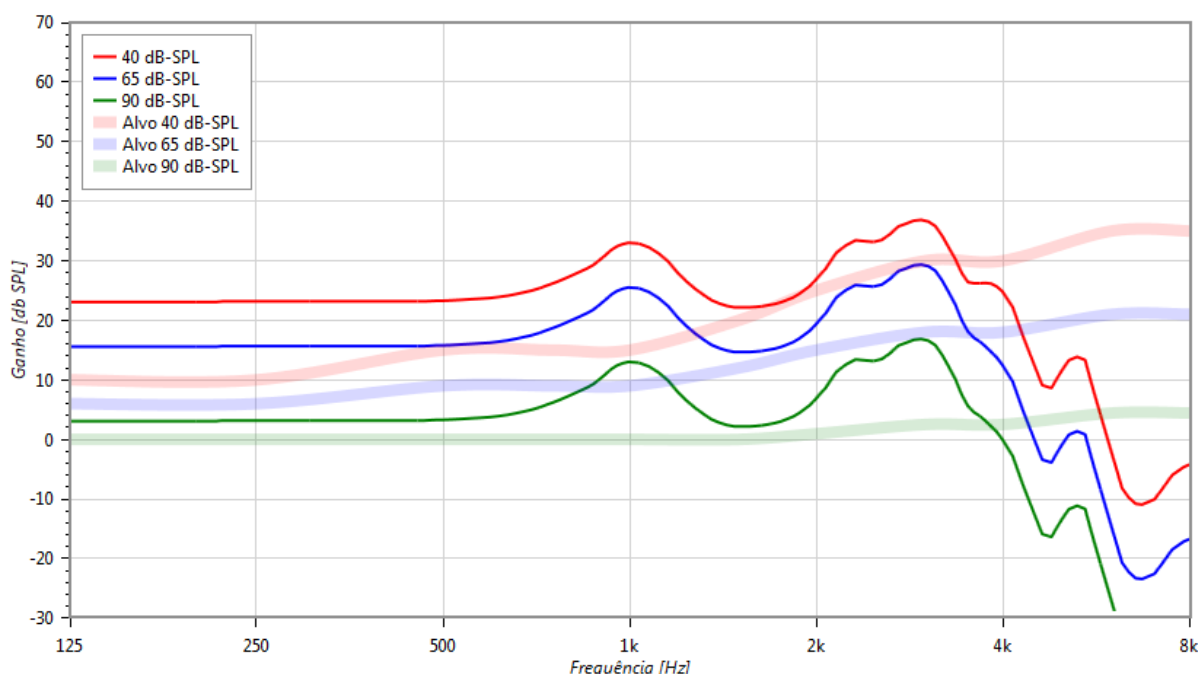


Figura 18 – Resposta em frequência do ajuste para o Audiograma 1 utilizando o algoritmo *Simulated Annealing*

Este algoritmo não conseguiu reduzir o erro entre as curvas da forma como era esperado no tempo que lhe foi concedido. Por esta razão, não foi aprovado para a implementação no WaveFit.

A variante *Adaptive Simulated Annealing*, ou Recozimento Simulado Adaptativo, poderia ser empregada para melhorar o desempenho deste algoritmo. Nesta técnica, os parâmetros são representados como números contínuos e como

dimensões de um hipercubo. Alguns algoritmos podem aplicar movimentos Gaussianos para mudar de estado enquanto outros possuem distribuições que permitem reduzir a temperatura mais rapidamente. A temperatura e o tamanho do passo são ajustados para que todo o espaço de busca seja amostrado de uma forma mais grossa nas primeiras iterações e se realize uma busca mais fina nas regiões mais favoráveis durante as últimas iterações. Porém, não houve tempo de desenvolver tal solução.

#### **4.4 Algoritmo Genético**

Para resolver o problema em questão usando algoritmo genético, a população inicial é escolhida aleatoriamente entre indivíduos que representam combinações válidas de parâmetros. Isto é feito para tentar elevar o espaço de soluções exploradas.

Em seguida, calcula-se qual o fitness de cada indivíduo e qual o valor da soma do fitness de todos os indivíduos. Como o objetivo do algoritmo é minimizar o erro entre as respostas prescritas e as respostas reais, o fitness será dado pelo inverso do erro quadrático, que foi usado como função objetivo nos algoritmos anteriores.

O próximo passo é sortear indivíduos que serão usados para gerar descendentes. Isto é feito através de um sorteio onde os indivíduos com maior fitness tem mais chance de serem escolhidos. Uma vez escolhidos, estes indivíduos passam pelo processo de crossover com outro indivíduo, na tentativa de encontrar uma solução nova melhor que a anterior. Ainda existe uma pequena chance de que cada indivíduo sofra mutação, o que irá alterar o valor de algum parâmetro aleatoriamente.

Neste caso específico, um indivíduo igual à melhor solução encontrada na geração anterior é copiado para a geração de descendentes, para que se evite a perda da melhor solução.

Caso o critério temporal ou de fitness seja atingido, o algoritmo é encerrado neste ponto. Se nenhum critério for atingido, a geração descendente ocupa o lugar de seus antecessores e o processo é reiniciado.

O pseudocódigo simplificado deste algoritmo é apresentado a seguir:

```
encontraSolução () {
    geracaoAtual[] = geraSolucoesAleatorias (tamanhoDaPopulacao);
    while (tempoDeExecucao < tempoMaximo && melhorFitness < fitnessMaximo) {
        foreach (solucao em geracaoAtual){
            solucao.calculaFitness();
        }
        geracaoAtual[] = ordenaPorMaiorFitness (geracaoAtual[]);
        fitnessAcumulado[] = geraFitnessAcumulado (geracaoAtual[])
        fitnessTotal = fitnessAcumulado[ultimo];
        novaGeracao[0] = geracaoAtual[0];
        for ( indicePopulacao = 1; indicePopulacao < tamanhoDaPopulacao;
            indicePopulacao++) {
            valorAleatorio = geraValorAleatório (fitnessTotal);
            indiceSorteado = fitnessAcumulado[].encontraPrimeiroIndice (
                fitnessAcumulado => fitnessAcumulado > valorAleatorio);
            novaGeracao[indicePopulacao] = geracaoAtual[indiceSorteado];
        }
        novaGeracao[] = executaCrossover (novaGeracao[]);
        foreach (solucao em novaGeracao) {
            if (valorAleatorio < chanceDeMutacao) {
                solucao.ocorreMutacao;
            }
        }
        geracaoAtual[] = novaGeracao[];
    }
}
```

Neste algoritmo pode-se modificar o tamanho da população e a chance de mutação. O tamanho da população é um parâmetro delicado neste caso, pois, dependendo do aparelho auditivo, o tamanho do espaço de soluções pode variar muito. Caso o tamanho da população seja muito pequeno, o espaço de soluções ficará subamostrado. Caso o tamanho da população seja grande, o algoritmo levará muito tempo para criar uma nova geração. Estes parâmetros foram definidos analisando a eficiência do algoritmo para diferentes valores. As análises foram feitas comparando os resultados de dez simulações distintas variando o valor de cada parâmetro.

Para se definir o tamanho da população foram comparados três casos onde o tempo limite do algoritmo foi de 30 segundos. Os resultados são mostrados na Tabela 10.



Tamanho populacional	Erro quadrático (dB SPL <sup>2</sup> )	Número de gerações
100	37586,55	67
500	38692,12	14
1000	39052,49	4

*Tabela 10 – Comparação entre diferentes tamanhos populacionais para o algoritmo genético<sup>11</sup>*

O caso em que o tamanho populacional é de 100 indivíduos apresentou menor erro quadrático. Devido a isto, o tamanho populacional foi fixado em 100 indivíduos.

Com o tamanho populacional fixado, avaliou-se qual o melhor valor para a chance de mutação. Foram comparados três casos onde o tempo limite também foi de 30 segundos. Os resultados são apresentados na Tabela 11.

Chance de mutação	Erro quadrático (dB SPL <sup>2</sup> )	Número de gerações
0,01	37688,91	67
0,05	37586,55	67
0,10	37823,41	67

*Tabela 11 – Comparação entre diferentes valores para a chance de reiniciar o algoritmo de Simulated Annealing<sup>12</sup>*

O valor escolhido para a chance de mutação foi 0,05, pois foi o valor que mais reduziu o erro quadrático.

#### **4.4.1 Resultados**

Simulações utilizando este algoritmo foram repetidas para as mesmas cinco curvas de ganhos prescrito utilizadas nas simulações dos algoritmos *Hill Climbing* e *Simulated Annealing*. O aparelho auditivo ajustado foi o mesmo, assim como o computador onde se realizaram os testes. Como este algoritmo apresenta um elemento randômico, cada teste foi repetido dez vezes. Os resultados destes testes são apresentados na Tabela 12.

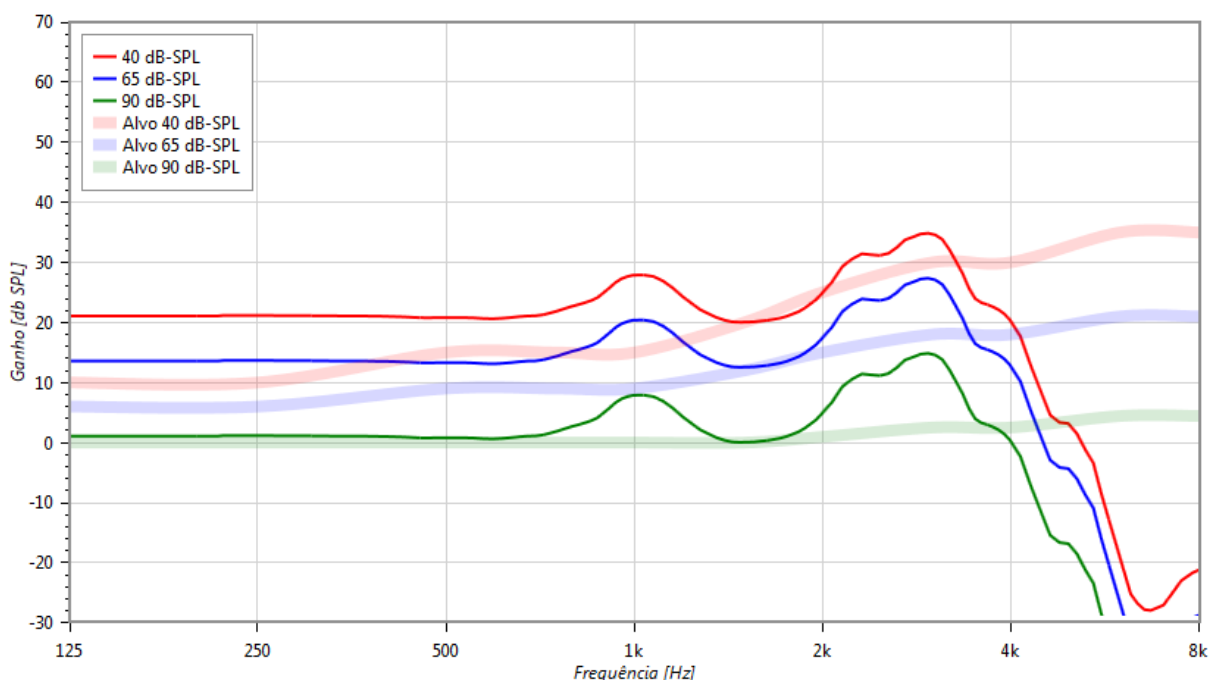
<sup>11</sup> Chance de mutação fixada em 0.05

<sup>12</sup> Tamanho populacional fixado em 100 indivíduos

Audiograma	Erro quadrático (dB SPL <sup>2</sup> )	Tempo de execução (segundos)	Número de gerações	Número de vezes que se avaliou a função objetivo
Audiograma 1	37265,51	30,25	67	6700
Audiograma 2	39874,01	30,12	67	6700
Audiograma 3	43182,19	30,26	67	6700
Audiograma 4	35851,73	30,32	67	6700
Audiograma 5	30865,71	30,18	67	6700
Média	37407,83	30,226	67	6700

*Tabela 12 – Desempenho do algoritmo genético para audiogramas distintos.*

A resposta em frequência para o ajuste realizado para o Audiograma 1 utilizando o algoritmo genético, pode ser vista na Figura 19.



*Figura 19 – Resposta em frequência do ajuste para o Audiograma 1 utilizando o algoritmo genético*

Este algoritmo não conseguiu reduzir o erro de forma aceitável no tempo que lhe foi concedido. Devido a isto não foi considerado como uma boa alternativa para resolver o problema de ajuste dos parâmetros.

Existem muitas variantes deste algoritmo que também poderiam ser aplicadas para tentar melhorar seu desempenho. Uma destas é Implementações Paralelas, onde podem existir diversas populações e a reprodução ocorre apenas em

populações que são consideradas vizinhas ou próximas. A idéia básica por trás destes algoritmos é dividir o problema em partes e resolver cada parte simultaneamente. Não houve tempo para explorar esta solução, mas a idéia de dividir o problema foi usada para desenvolver um algoritmo no Capítulo 5.

## Capítulo 5: Algoritmo de Otimização Proposto

O algoritmo que apresenta melhores resultados em relação à redução do erro é o *Hill Climbing*, porém nenhum dos algoritmos implementados é considerado suficientemente bom. Os três algoritmos necessitam de muito mais tempo do que o tempo limitado pelos requisitos do problema para apresentar soluções com um erro aceitável.

Para reduzir o tempo de processamento seria possível parar o algoritmo quando o critério temporal fosse atingido. Entretanto esta solução não produziria resultados com o erro quadrático entre as curvas de ganho prescrito e as reais tão pequenos.

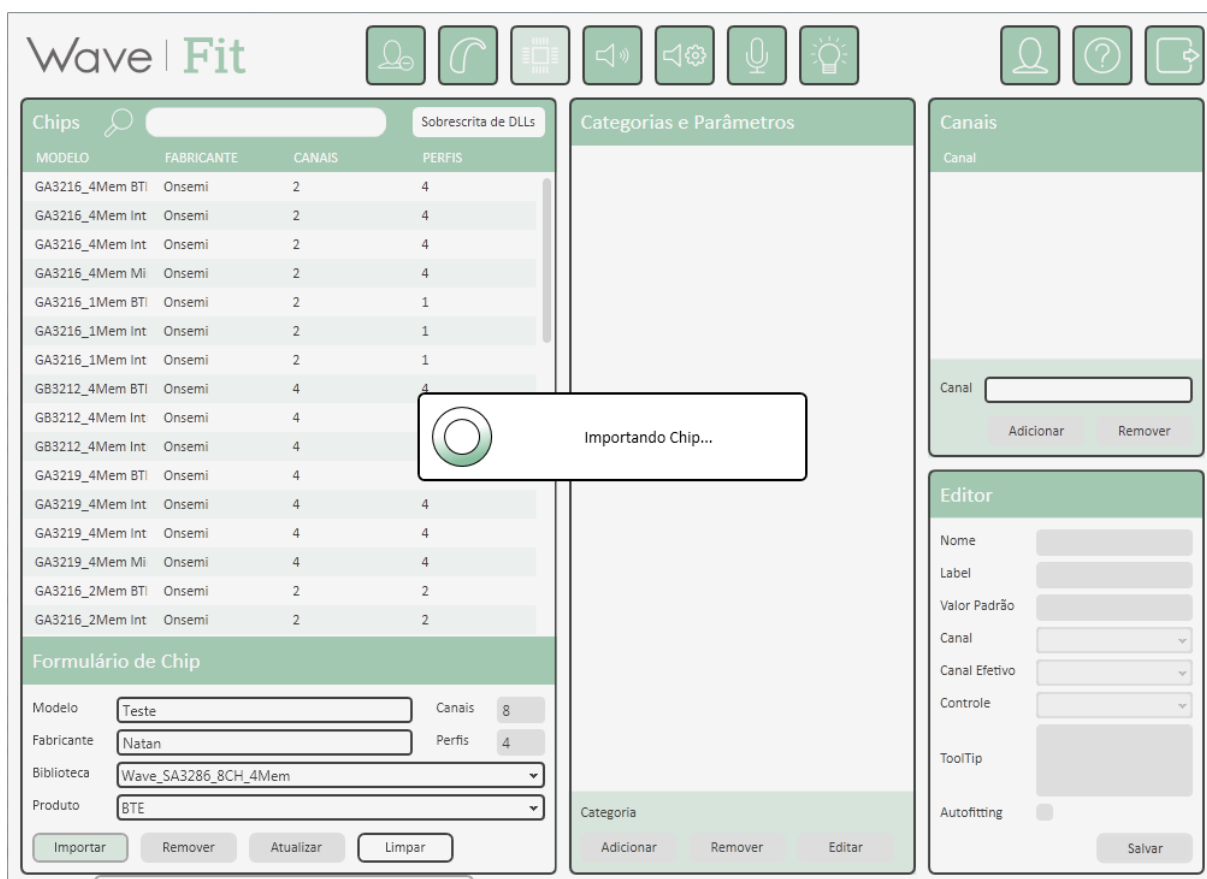
Uma solução alternativa é proposta neste capítulo. Esta solução consiste em modificar o algoritmo de *Hill Climbing* com o objetivo de reduzir o tempo de processamento.

A cada iteração o algoritmo de *Hill Climbing* avalia a função objetivo para cada vizinho da solução atual. Como existem 173 parâmetros no chip mais complexo, podem existir até 346 soluções avaliadas a cada iteração e um total de  $2.7185e+188$  combinações possíveis entre tais parâmetros. O foco do algoritmo proposto neste capítulo é reduzir o número de soluções que são avaliadas a cada iteração, o que reduz o tempo de processamento do algoritmo.

### 5.1 Identificação dos Parâmetros Ajustados Automaticamente

Reduzindo o número de parâmetros a serem ajustados, reduzimos o número de cálculos em cada iteração, bem como o total de combinações possíveis entre eles. Com isto em mente, foi proposto que, durante o cadastro de chips aparelhos auditivos no software, se realize uma pré-seleção de parâmetros que serão usados no ajuste automático.

A etapa de cadastro de chip ocorre somente uma vez e seus resultados são salvos no banco de dados. Esta etapa ocorre apenas na WaveTech durante a configuração do software. A tela de importação de chips é mostrada na Figura 20.



*Figura 20 – Tela de importação de chip no WaveFit*

Durante o cadastro de chips de aparelhos auditivos todos os parâmetros do chip são avaliados. Esta avaliação determina se o parâmetro impacta realmente na curva de resposta em frequência. Muitos parâmetros não modificam a curva, como é o caso dos parâmetros relacionados a sons emitidos durante troca de perfil e redutores de ruído dinâmico. Após esta seleção automatizada, foi possível identificar que apenas 60 parâmetros presentes no SA3286 realmente influenciam a curva de resposta em frequência.

Mesmo reduzindo para 60 parâmetros a serem avaliados, a cada iteração são avaliadas até 120 soluções. Como ocorrem cerca de 100 iterações a cada ajuste, isto implica em avaliar até 12000 soluções durante o ajuste, impactando em um tempo de processamento próximo de 40 segundos no mesmo computador onde foram realizadas as avaliações anteriores.

Após a seleção automatizada de parâmetros, verificou-se que alguns dos parâmetros selecionados não causam alterações significativas na curva de resposta em frequência. Uma revisão manual determinou que apenas 36 parâmetros realmente devem ser ajustados automaticamente para o chip SA3286, o que resulta em até 72 cálculos da função objetivo a cada iteração e avaliar até 7200 soluções durante todo o processo. Isto implica em um tempo de processamento próximo de 25 segundos no mesmo computador.

## 5.2 Separação dos Parâmetros em Canais

Foi verificado que muitos destes parâmetros que foram selecionados impactam uma zona específica da curva de resposta em frequência. Um exemplo é o parâmetro de ganho do canal 1 que impacta apenas a resposta do canal 1 e de forma menor a resposta do canal 2 devido ao filtro de corte que não é perfeito. Devido a isto, é proposto que se realize um ajuste separado para cada canal no lugar de um ajuste com todos os parâmetros juntos. Durante a importação de chip no software também se avalia qual canal de frequência o parâmetro afeta.

Se um parâmetro provocar mudanças de forma similar em todos os canais, não lhe é atribuído nenhum canal específico. Mas se ele impactar de forma mais significativa apenas um canal, ele então é associado a este canal. Na Tabela 13 podemos ver a distribuição dos parâmetros do chip SA3286 por canal.

Canal	Número de Parâmetros	Número de combinações possíveis
Nenhum canal atribuído	2	400
1	5	720192
2	4	45012
3	4	45012
4	4	45012
5	4	45012
6	4	45012
7	4	45012
8	5	720192
Total	36	1,71e+6

*Tabela 13 – Número de parâmetros a serem ajustados por canal no chip SA3286*

Durante o processo de ajuste automático, o algoritmo irá ajustar primeiro aqueles parâmetros que não possuem nenhum canal associado e em seguida irá ajustar os parâmetros relacionados a um canal por vez. Isto reduz o número de iterações em comparação ao caso onde todos os parâmetros são ajustados em conjunto.

Como em um ajuste através do *Hill Climbing* são necessárias em torno de 100 iterações e que 36 parâmetros são modificados, podemos considerar que em média cada parâmetro é modificado 3 vezes. A Tabela 14 mostra o número estimado de cálculos da função objetivo para que se realize um ajuste completo.

Canal	Número máximo de cálculos da função objetivo por iteração	Número estimado de iterações	Total estimado de cálculos da função objetivo
Nenhum canal atribuído	4	6	24
1	10	15	150
2	8	12	96
3	8	12	96
4	8	12	96
5	8	12	96
6	8	12	96
7	8	12	96
8	10	15	150
Total	36	108	900

*Tabela 14 – Número estimado de cálculos da função objetivo durante um ajuste usando o algoritmo proposto*

### 5.3 Algoritmo Proposto

A solução final proposta é realizar a identificação automatizada de parâmetros a serem ajustados e em seguida filtrar os resultados desta identificação de forma manual combinado com a separação dos parâmetros em canais. Este processo pode ser lento, mas ocorre apenas uma vez durante o cadastro de um chip de aparelho auditivo.

Durante o ajuste automático, realiza-se um ajuste automático dos parâmetros que não estão associados a nenhum canal através do algoritmo *Hill Climbing* e em

seguida se ajusta os parâmetros de cada canal através do mesmo algoritmo. O pseudocódigo simplificado deste algoritmo será apresentado a seguir.

```
encontraSolucao () {
    solucaoAtual = valorPadrao;
    fitnessAtual = calculaFitness (solucaoAtual);
    for (canal = 0; canal <= numeroDeCanais; canal++) {
        otimoLocalEncontrado = false;
        while (!otimoLocalEncontrado) {
            solucoesVizinhasDoCanal[] =
                calculaSolucoesVizinhasDoCanal (solucaoAtual, canal);
            if (solucoesVizinhasDoCanal [].tamanho () == 0) {
                otimoLocalEncontrado = true;
                break;
            }
            melhorFitnessNosVizinhos = +inf;
            melhorVizinho = solucoesVizinhas[0];
            foreach(solucaoVizinha em solucoesVizinhasDoCanal []) {
                fitnessDoVizinho = calculaFitness (solucaoVizinha);
                if (fitnessDoVizinho < melhorFitnessNosVizinhos){
                    melhorFitnessNosVizinhos = fitnessDoVizinho;
                    melhorVizinho = solucaoVizinha;
                }
            }
            if (melhorFitnessNosVizinhos < fitnessAtual) {
                solucaoAtual = melhorVizinho;
                fitnessAtual = melhorFitnessNosVizinhos;
            }
            else {
                otimoLocalEncontrado = true;
            }
        }
    }
}
```

Simulações utilizando o algoritmo proposto para realizar o ajuste automático dos parâmetros do chip SA3286 foram repetidas para as mesmas cinco curvas de ganho prescrito utilizadas nas simulações dos algoritmos apresentados no Capítulo 4. O aparelho auditivo ajustado foi o mesmo, assim como o computador onde se realizaram os testes. Os resultados do teste são apresentados na Tabela 15.



Audiograma	Erro quadrático (dB SPL <sup>2</sup> )	Tempo de execução (segundos)	Número de iterações	Número de vezes que se avaliou a função objetivo
Audiograma 1	4482,12	2,32	106	860
Audiograma 2	3972,53	2,28	104	855
Audiograma 3	4139,75	2,45	105	873
Audiograma 4	3845,11	2,38	109	862
Audiograma 5	4542,65	2,46	114	868
Média	4196,432	2,378	107,6	863,6

Tabela 15 – Desempenho do algoritmo proposto para audiogramas distintos

O algoritmo proposto reduz bastante o tempo de processamento e chega a uma solução final muito próxima do *Hill Climbing* inicial. Os resultados e as comparações serão apresentados no capítulo seguinte.

A resposta em frequência para o ajuste realizado para o Audiograma 1 utilizando o algoritmo *Hill Climbing*, pode ser vista na Figura 21.

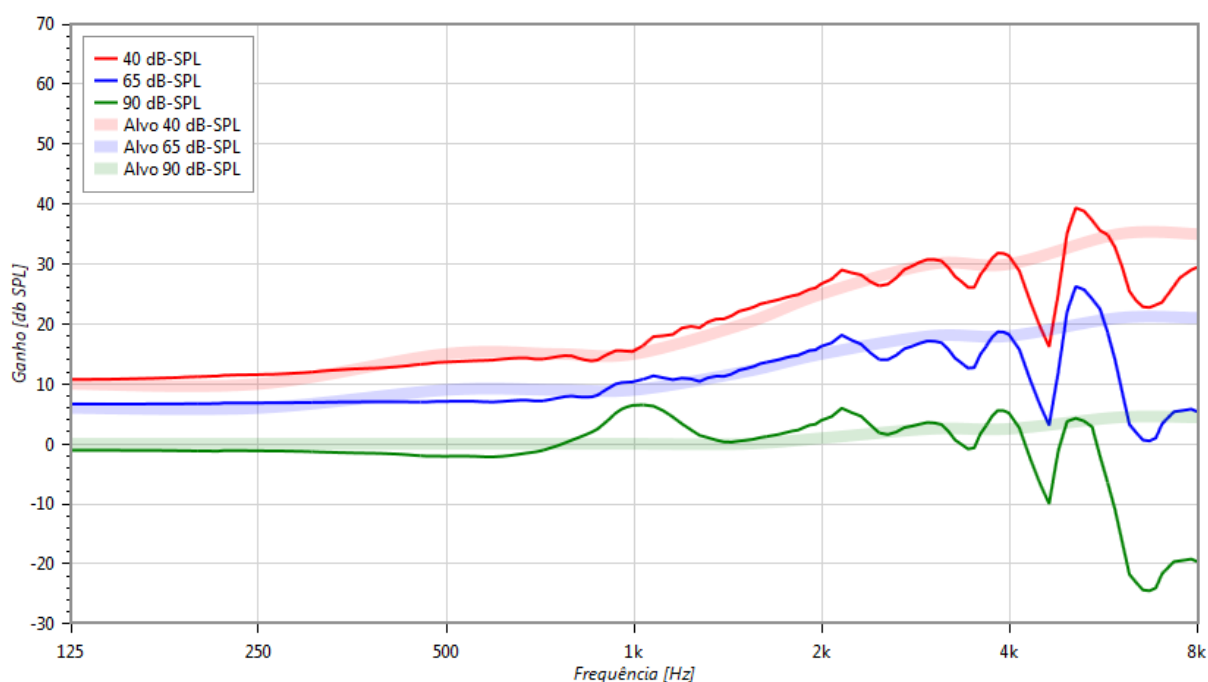


Figura 21 – Resposta em frequência do ajuste para o Audiograma 1 utilizando o algoritmo proposto

## Capítulo 6: Resultados

Neste capítulo são abordadas as comparações entre os algoritmos desenvolvidos durante este projeto em relação ao erro quadrático, tempo de processamento e número de vezes que se avalia a função objetivo.

Na Tabela 16 são apresentados os valores médios das simulações para o ajuste dos parâmetros do chip SA3286 usando como curvas de ganho prescrito as curvas geradas pela regra FIG6 para os audiogramas 1, 2, 3, 4 e 5 apresentados no Capítulo 4. Para os algoritmos *Simulated Annealing* e algoritmo genético, por existir um fator randômico, os testes foram repetidos dez vezes para cada uma das curvas de ganho prescrito.

Algoritmo	Média do erro quadrático (dB SPL)	Média do tempo de processamento (segundos)	Média do número de vezes que se avaliou a função objetivo
<i>Hill Climbing</i>	4106,44	122,20	33276
<i>Simulated Annealing</i>	26265,90	42,26	7325
Algoritmo genético	37407,83	30,23	6700
Algoritmo proposto	4196,43	2,38	864

Tabela 16 – Comparação entre o desempenho dos algoritmos *Hill Climbing*, *Simulated Annealing*, algoritmo genético e o algoritmo proposto.

O algoritmo genético não apresenta bons resultados, pois o tamanho do espaço de soluções é muito maior do que o número de soluções que são avaliadas. Existem  $2,7185e+188$  combinações possíveis de parâmetros para o chip SA3286 e são avaliadas no máximo 6700 delas. Isto implica em uma amostragem muito pequena dos indivíduos do espaço de soluções. Para que se avaliem mais indivíduos este algoritmo precisa de mais tempo de execução.

O algoritmo *Simulated Annealing* também não apresenta bons resultados. Existem muitos vizinhos para cada solução e este algoritmo escolhe um vizinho aleatório para avaliar. Alguns dos vizinhos apresentam um desempenho muito melhor do que outros, mas a chance de serem escolhidos aleatoriamente é muito pequena. Isto resulta no desempenho ruim deste algoritmo.

Já o algoritmo *Hill Climbing* com *Steepest Ascent* avalia todos os seus vizinhos para somente então escolher aquele que será a solução da próxima iteração. Ele leva muito mais tempo em cada iteração, mas garante a escolha do vizinho com melhor desempenho. Devido a isto, ele apresenta um erro quadrático médio bem menor do que os algoritmos *Simulated Annealing* e algoritmo genético, mas apresenta um tempo de execução muito maior do que os 5 segundos estabelecidos como meta.

O algoritmo proposto assimila a característica de *Steepest Ascent* e reduz muito o tempo de processamento do algoritmo. Isto se deve à redução do número de vezes que se avalia a função objetivo em cada iteração através da separação de parâmetros por canais de influência. Ele apresenta um erro quadrático levemente maior que o *Hill Climbing*, mas também apresenta um tempo de simulação muito menor.

O algoritmo proposto é o único algoritmo que satisfaz as metas estabelecidas no Capítulo 3 e por isso foi escolhido para ser usado no WaveFit.

É importante ressaltar que as curvas de ganho esperadas pelos ajustes realizados nos chips de aparelho auditivo foram validadas através do AURICAL Plus, onde apresentaram um erro de  $\pm 5$  dB SPL com um intervalo de confiança de 95%.

## Capítulo 7: Conclusões e Perspectivas

Através da análise de algoritmos de otimização foi possível desenvolver um algoritmo próprio para o ajuste de parâmetros de aparelhos auditivos que atendesse as metas propostas pela empresa. As contribuições do autor permitiram automatizar grande parte do processo de ajuste de parâmetros de um aparelho auditivo.

Para o autor, este trabalho foi de grande importância e aprendizado, pois não só teve a oportunidade de estudar sobre algoritmos de inteligência artificial e sobre desenvolvimento de software, como também mostrou as dificuldades de embutir um projeto acadêmico dentro de um produto.

Porém, o desenvolvimento do algoritmo não se dá como terminado, pois ainda existem melhorias a serem realizadas. Uma das melhorias identificadas está relacionada à realização de um ajuste mais preciso na faixa de frequências com maior relevância para o paciente no intuito de elevar ainda mais a inteligibilidade.

O trabalho desenvolvido durante este projeto é de grande valor, pois auxiliará os fonoaudiólogos a realizarem ajustes mais precisos nos aparelhos auditivos de seus pacientes, proporcionando um tratamento mais eficaz.

Em breve o software será implantado em uma clínica de fonoaudiologia para que se obtenha retorno dos fonoaudiólogos e se adapte o software às suas necessidades.

## Bibliografia

- [1] IBGE, “Censo Demográfico - Características Gerais da População, Religião e Pessoas com Deficiência,” 2010. [Online]. Available: [ftp://ftp.ibge.gov.br/Censos/Censo\\_Demografico\\_2010/Caracteristicas\\_Gerais\\_Religiao\\_Deficiencia/caracteristicas\\_religiao\\_deficiencia.pdf](ftp://ftp.ibge.gov.br/Censos/Censo_Demografico_2010/Caracteristicas_Gerais_Religiao_Deficiencia/caracteristicas_religiao_deficiencia.pdf). [Acesso em 25 Junho 2015].
- [2] H. Dillon, *Hearing Aids*, Thieme, 2001.
- [3] E. M. Boéchat, P. d. L. Menezes, C. M. do Couto, A. C. F. Frizzo, R. C. Scharlach e A. R. T. Anastasio, *Tratado de Audiologia*, Rio de Janeiro: Guanabara Koogan, 2015.
- [4] A. SCHAUB, *Digital Hearing Aids*, New York: Thieme, 2008.
- [5] W. J. Mullin, W. J. Gerace, J. P. Mestre e S. L. Velleman, *Fundamentals of Sound with Applications to Speech and Hearing*, Amherst: Allyn and Bacon, 2003.
- [6] E. Zwicker e H. Fastl, *Psychoacoustics - Facts and Models*, München: Springer, 1999.
- [7] M. Killion e S. Fikret-Pasa, “The 3 types of sensorineural hearing loss: Loudness and inteligibility considerations,” *The Hearing Journal*, vol. 46, pp. 31-34, 1993.
- [8] M. Killion, “FIG6,” em *Jackson Hole Rendezvous*, Jackson, 1994.
- [9] K. Deb, *Optimization for Engineering Design*, New Delhi: PHI Learning Private Limited, 2004.
- [10] T. Weise, *Global Optimization Algorithms - Theory and Application*}, Self-Published, 2009.

- [11] S. Evers, "Hill Climbing Variations," 08 09 2013. [Online]. Available: <https://steveevers.wordpress.com/2013/09/08/hill-climbing-variations-2/>. [Acesso em 06 07 2015].
- [12] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller e E. Teller, "Equation of state calculations by fast computing machines," *The Journal of Chemical Physics*, pp. 1087-1092, Junho 1953.
- [13] S. Kirkpatrick, C. D. Gelatt e M. P. Vecchi, "Optimization by simulated annealing," *Science*, pp. 671-680, 13 Maio 1983.
- [14] N. A. Barricelli, "Esempi numerici di processi di evoluzione," *Methodos*, pp. 45-68, 1954.
- [15] A. S. Fraser, "Simulation of genetic systems by automatic digital computers," *Australian Journal of Biological Science*, pp. 484-491, 1957.