



# RELATÓRIO DE ESTÁGIO INTEGRADO

**Busca por padrões inovadores para utilização em filtros espaciais à base de Superfície Seletiva em Frequência**

**Realizado por:**

**Vinícius Magno Uchôa Lima Oliveira**


**Matrícula: 20611473**

**Engenharia Elétrica - UFCG**

<b>Período do estágio</b>	01/03/2010 a 19/07/2010
<b>Tutor da empresa</b>	Guy EYMIN
<b>Orientador da PHELMA</b>	Tan Phu VUONG
<b>Orientador da UFCG</b>	Francisco TEJO

**PROJETO DE ESTÁGIO**  
**GRENOBLE INP – PHELMA 2009/2010**

Recherche de motifs innovants pour l'utilisation de filtre spatial à base de Surface Sélective en Fréquence

 <p>centre technique du papier</p>	<p>Domaine Universitaire - BP 251 38044 GRENOBLE - CEDEX 9 FRANCE Tél. : +33 (0)4 76 15 40 15 Fax : +33 (0)4 76 15 40 16 <a href="http://www.webctp.com">www.webctp.com</a></p>
---	---

**Nom et Prénom de l'étudiant :** UCHOA LIMA OLIVEIRA Vinicius Magno

<b>Date du stage</b>	du 01/03/2010 au 19/07/2010
<b>Tuteur Entreprise</b>	Guy EYMIN
<b>Tuteur PHELMA-INPG</b>	Tan Phu VUONG
<b>Module d'Approfondissement</b>	SIRO (Systèmes Intégrés, Radiofréquences et Optoélectroniques)

# RELATÓRIO DE ESTÁGIO INTEGRADO

GRENOBLE INP – PHELMA 2009/2010

**Palavras-chave:** Superfície Seletiva em Frequência, FSS, Algoritmos Genéticos, Macros, Coeficiente de Transmissão, CST Microwave Studio, Software de Simulação Eletromagnética

## Resumo:

Este trabalho tem como objetivo encontrar um bom método para se pesquisar novos padrões de FSS (*frequency selective surface*) para serem utilizados em filtros espaciais.. Um método bem difundido são os Algoritmos Genéticos (AG), que estão fortemente ligados à Biologia, à teoria da evolução de Darwin e à seleção natural. Estudamos esta poderosa ferramenta para resolver problemas de FSS lançando várias simulações no software CST Microwave Studio. Para isto, estudamos as macros, que são seqüências de comandos para que o CST trabalhe de forma automática com ajuda de uma linguagem de programação. Neste trabalho, escolhemos o *software* Matlab para realizar a interface com o CST. Ao final do projeto, encontramos padrões para filtrar os sinais às frequências do GSM (900MHz e 1,8GHz).

**Keywords :** Surface Selective en Frequency, FSS, Genetics Algorithms, Macros, Transmission Coefficient, CST Microwave Studio, Electromagnetic Simulation Software

## Abstract :

The goal of this project is to find a good method to search new figures de FSS (*frequency selective surface*) to be utilized in spatial filters. A very common method is the Genetic Algorithms (GAs), which are strongly related to biology, the theory of evolutionism of Darwin and natural selection. We studied this powerful tool to solve FSS's problems, running various simulations in CST, and because of this, we studied the macros, which are commands that allows that CST to works automatically using a programming language. In this project, we have chosen the Matlab software to interface with the CST. At the end of the project, we found structures FSS capable to filter signals at frequencies of GSM (900 MHz and 1.8GHz).

## Agradecimentos

Dedico este trabalho a todos que contribuíram para a realização destes quatro meses e meio de estágio e durante toda minha estadia na França.

Agradeço a minha maravilhosa família, em especial meus pais Antonio Uchoa e Rosane pela vida, ensinamentos, por me ter permitido escolher o bom caminho e pelo amor imensurável. Meus irmãos Rostonio e Gabriela pelo amor, ambiente familiar, o apoio e amizade. Minha madrinha Solange pelos conselhos, o carinho e apoio. Todos vocês são meus exemplos a seguir.

Agradeço minha namorada Fabi pelo amor, apoio, compreensão, confiança, paciência, seu exemplo que me faz ter orgulho de viver a seu lado. Agradeço também a Francisco e Cleide pelo apoio e por me acolher tão bem como família longe da minha família.

Agradeço com imenso prazer:

- Senhores Tan Phu VUONG, Guy EYMIN, Glauco FONTGALLAND e Raimundo FREIRE; grupo CTP, PHELMA-INPG e UFCG pela oportunidade única na minha vida profissional e pessoal;
- Especialmente aos professores Francisco TEJO, Waslon TERLLIZZIE, Benedito LUCIANO e Edmar GURJÃO.
- Fabien DE BARROS, Giovanni BULLA e Victor FREITAS pelos ensinamentos, consultorias, e por suas paciências;
- Ao governo brasileiro que através da CAPES me permitiu viver esta experiência magnífica no exterior.
- Aos colegas: Alysson VASCONCELOS, Ayslan MAIA, Bruno FRANCISCATTO, Christophe BERTRAND, Letícia NOBREGA, Lucas MUNIZ, Ricardo MALHERBI, Roberto BONFIM, Victor MELO, Vitor FREITAS e Vlad ANICULAESEI, e especialmente meu companheiro Gregory CARLOS.

## Tabela de Figuras

Figura 2.1 - Radar Cross Section (RCS) .....	11
Figura 2.2 - F-117 Nighthawk (US Airforce), o primeiro avião do mundo a utilizara tecnologia Stealth, possuindo estruturas FSS na sua carcaça. ....	12
Figura 2.3 - Estrutura periódica excitada por uma onda plana incidente $E^i$ (Extraída de (1)).	13
Figura 2.4 - Princípio de funcionamento de um filtro espacial; Extraído de (6) .....	15
Figura 2.5 - Elementos típicos de FSS e suas classificações .....	15
Figura 2.6 - Exemplo de estruturas complementares para exposição do Princípio de Babinet. Extraída de (1). ....	16
Figura 3.1 - Terminologia do Algoritmo Genético esquematizada. Extraída de (8).....	18
Figura 3.2 - Esquema do funcionamento de um algoritmo genético simples .....	19
Figura 3.3 - Codificação Binária .....	20
Figura 3.4 - Crossover (a) Um único ponto (b) A dois pontos .....	21
Figura 3.5 - Função hipotética para ilustrar a idéia da mutação .....	21
Figura 3.6 - Mutação de 2 genes ao acaso .....	22
Figura 3.7 - Exemplo de uma seleção por roleta.....	22
Figura 3.8 - Seleção por Torneio .....	23
Figura 4.1 - Ângulo de incidência $\theta$ da onda plana atravessando uma estrutura FSS. ....	26
Figura 4.2 - Criação do novo projeto.....	27
Figura 4.3 - Condições de Fronteira .....	28
Figura 4.4 - Escolha do número de modos de Floquet .....	28
Figura 4.5 - (a) Operações realizadas para a construção de um anel simples no History List; (b) código responsável para estas operações geradas em linguagem VBA. ....	29
Figura 5.1 - Dimensão aproximada para obter a ressonância para um tripolo simples .....	30
Figura 5.2 - Célula elementar e estrutura FSS para os tripolos imbricados.....	30
Figura 5.3 - Coeficiente de Transmissão ( $\tau$ ) para os valores de comprimento do braço do tripolo seguintes: 60 mm ; 80 mm ; 100 mm .....	31
Figura 5.4 - Coeficiente de transmissão ( $\tau$ ) para os valores do ângulo de incidência $\theta$ seguintes: $0^\circ$ ; $30^\circ$ ; $60^\circ$ .....	31
Figura 5.5 - Influência de um dielétrico na freqüência de ressonância. Se uma estrutura FSS é imersa em um meio infinito de $\epsilon_r$ constante, a freqüência de ressonância será deslocada de aproximadamente $f_{oer}$ . Extraído de (1).....	32
Figura 5.6 - À esquerda : Ambiente de simulação de uma estrutura FSS em anel (Grupo 2) imersa em uma camada de dielétrico FR-4 (sem perdas). À direita: variáveis relacionadas ao anel. ....	32
Figura 5.7 - Coeficiente de transmissão ( $\tau$ ) para os valores de raio seguintes: 30 mm ; 45 mm ; 50 mm, 60 mm. ....	33
Figura 5.8 - Coeficiente de transmissão ( $\tau$ ) para os valores de gap seguintes: 1 mm; 15.5 mm, 30 mm.....	33

Figura 5.9 - Coeficiente de transmissão ( $\tau$ ) para os valores do ângulo de incidência $\theta$ seguintes: $0^\circ$ ; $45^\circ$ ; $67.5^\circ$ .....	34
Figura 5.10 - Coeficiente de transmissão ( $\tau$ ) para os valores da espessura do dielétrico seguintes: 10 mm; 20 mm; 30 mm .....	34
Figura 5.11 - Célula Básica de um padrão gerado por um algoritmo genético comum com matriz de ordem $N = 16$ .....	34
Figura 5.12 - Algumas estruturas FSS geradas pelo algoritmo genético construído. ....	35
Figura 5.13 - Estruturas FSS: (a) Tipo N-Pólo (Grupo 1); (b) Tipo caminho fechado (Grupo 2); (c) Tipo Placa (Grupo 3);.....	35
Figura 5.14 - (a) Representação de uma FSS composta por tripolos utilizando malha triangular; (b) Dimensões de um tripolo simples gerado por este tipo de malha.....	36
Figura 5.15 - (a) Célula básica para o tripolo de malha quadrada simulado em CST; (b) Idéia da superfície FSS obtida para este tipo de tripolo. ....	36
Figura 5.16 - Coeficiente de transmissão ( $\tau$ ) para os valores de comprimento de do parâmetro $l$ para 20 mm ; 22,5mm ; 25 mm.....	37
Figura 5.17 - Coeficiente de transmissão ( $\tau$ ) para os valores do ângulo de incidência ( $\theta$ ) para $0^\circ$ ; $30^\circ$ ; $60^\circ$ .....	38
Figura 5.18 - Construção de uma célula elementar e codificação utilização simetria: (a) 2-fold ; (b) 4-fold ; (c) 8-fold ; Os quadrados em azul correspondem à presença de metal (bit 1)....	39
Figura 5.19 - Diferença entre uma estrutura $N$ par (à esquerda) e ímpar (à direita). A linha tracejada representa os eixos de simetria .....	40
Figura 5.20 - Verificação da presença de bons picos na curva do coeficiente de transmissão. O pico é bom se está localizado na área em azul (em torno da frequência de interesse e atenuação inferior a -30dB). ....	41
Figura 5.21 - Esquema demonstrativo para atribuição de <i>fitness</i> no algoritmo construído ...	42
Figura 5.22 - Padrão de testes e Estrutura FSS para obtenção do comprimento de pixel. ....	42
Figura 6.1 - Célula elementar e estrutura FSS correspondente para um filtro rejeita-faixa GSM (900MHz). ....	44
Figura 6.2 - Coeficiente de Transmissão para um filtro rejeita-faixa GSM (900MHz). ....	44
Figura 6.3 - Célula elementar e estrutura FSS correspondente para um filtro rejeita-faixas GSM (900MHz) e (1.8GHz). ....	45
Figura 6.4 - Coeficiente de Transmissão para um filtro rejeita-faixas GSM (900MHz) e (1.8GHz).....	45
Figura 6.5 - Célula elementar e estrutura FSS correspondente para um filtro rejeita-faixa GSM (900MHz) pouco sensível ao ângulo de incidência $\theta$ . ....	46
Figura 6.6 - Coeficiente de Transmissão ( $\tau$ ) para um filtro rejeita-faixa GSM (900MHz) para ângulo de incidencia $\theta = 0^\circ$ .....	46
Figura 6.7 - Coeficiente de Transmissão ( $\tau$ ) para um filtro rejeita-faixa GSM (900MHz) para ângulo de incidência $\theta = 30^\circ$ .....	47
Figura 6.8 - Célula elementar e estrutura FSS correspondente para um filtro rejeita-faixas GSM (900MHz e 1.8GHz) com pouca sensibilidade à variação do ângulo de incidência $\theta$ . ....	48

Figura 6.9 - Coeficiente de Transmissão ( $\tau$ ) para um filtro rejeita-faixas GSM (900MHz e 1.8GHZ) para $\theta = 0^\circ$ .....	48
Figura 6.10 - Coeficiente de Transmissão ( $\tau$ ) para um filtro rejeita-faixas GSM (900MHz e 1.8GHZ) para $\theta = 30^\circ$ .....	48
Figura 6.11 - Coeficiente de Transmissão ( $\tau$ ) para uma faixa de frequência de 0 a 5 GHz .....	49
Figura 6.12 - Vista das magnitudes de campo elétrico à 0.9 GHz (acima) e 5 GHz (abaixo)....	49
Figura 6.13 - Curva de evolução do valor de fitness para cada geração.....	50

## Sumário

1	Introdução.....	6
1.1	Apresentação da empresa CTP .....	6
1.1.1	Atividade.....	6
1.1.2	Situação Geográfica e socioeconômica.....	7
1.1.3	Organograma Funcional do CTP.....	8
1.2	Contexto e Especificações .....	9
1.3	Propósito do estágio .....	9
2	Superfícies Seletivas em Frequência (FSS).....	11
2.1	Definição e Histórico .....	11
2.2	Estruturas Periódicas e Teorema de Floquet .....	12
2.3	Como funcionam os filtros espaciais.....	14
2.4	Tipos de Elementos de FSS.....	15
2.5	Princípio de Babinet .....	16
2.6	Conclusão .....	16
3	Algoritmos Genéticos.....	17
3.1	Histórico .....	17
3.2	Definição e Princípio.....	17
3.3	Componentes de um algoritmo genético .....	19
3.3.1	Operadores genéticos e codificação de um algoritmo genético .....	19
3.3.1.1	Codificação dos cromossomos.....	20
3.3.1.2	Crossover.....	20
3.3.1.3	Mutação .....	21
3.3.2	Estratégias de seleção .....	22
3.3.2.1	Seleção por Roleta .....	22
3.3.2.2	Seleção por Torneio .....	22
3.3.2.3	Elitismo.....	23
3.3.3	Função de avaliação .....	23
3.4	Quando se utilizar os Algoritmos Genéticos? .....	24
3.5	Limitações dos Algoritmos Genéticos .....	24
3.6	Conclusão .....	25
4	CST <sup>®</sup> <i>Microwave Studio</i> .....	26



4.1	Ambiente FSS em CST Microwave Studio .....	26
4.2	Trabalhando com macros.....	28
4.3	Conclusão .....	29
5	Simulações e Construção do Algoritmo .....	30
5.1	Simulação dos padrões FSS de primeiro e segundo grupo .....	30
5.1.1	Tripolos simples imbricados .....	30
5.1.2	Anéis .....	31
5.1.3	Padrões gerados pelos algoritmos genéticos.....	34
5.2	Algoritmo construído e seu método .....	35
5.2.1	Estudos sobre uma malha triangular .....	35
5.2.2	Malha quadrada e simetria <i>n-fold</i> .....	38
5.3	Estrutura geral do Algoritmo Genético construído.....	39
5.3.1	Função de Avaliação.....	40
5.4	Conclusão .....	43
6	Resultados.....	44
6.1.1	Filtro Rejeita-Faixa GSM (900 MHz) .....	44
6.1.2	Filtro rejeita-faixas GSM (900 MHz) e (1,8 GHz) .....	45
6.1.3	Filtro Rejeita-Faixa GSM (900 MHz) insensível ao ângulo de incidência $\theta$ .....	46
6.1.4	Filtro rejeita-faixas GSM (900 MHz) e (1,8 GHz) pouco sensível à variação do ângulo de incidência $\theta$ . .....	47
7	Conclusões e Perspectivas .....	51
8	Bibliografia .....	52
9	Anexos.....	54

# 1 Introdução

## 1.1 Apresentação da empresa CTP

### 1.1.1 Atividade

O Centro Técnico do Papel (CTP) é um organismo convencionado de pesquisa e desenvolvimento industrial, de perícia e de informação que possui um suporte técnico e científico para a indústria de polpa, papéis, papéis cartão e indústrias associadas, clientes do CTP, com o objetivo de incrementar sua produtividade e competitividade, respeitando o meio ambiente e as leis nacionais ou européias em vigor.

Através do alto nível de competência dos seus pesquisadores e engenheiros em todas as disciplinas da cadeia produtiva do papel e papelão, o CTP se destaca:

- Por sua situação geográfica próxima aos pólos científicos grenobleses (EFGP, CNRS, INPG, etc.).
- Por sua capacidade de intervenção sobre os locais de produção: **Prestações, Consultorias e Perícias.**
- Pelo desempenho dos seus pilotos que validam melhorias nos processos estudados em laboratórios.
- Por seu banco de dados de documentos, em colaboração com o KCL, STFI e a PIRA, e seu estudos socioeconômicos e sua tecnologia.

O CTP coloca igualmente à disposição das empresas sua experiência no domínio de polpas, papéis, e papéis cartão para fornecer treinamento *à la carte*, com programas sobre medida. Estes treinamentos de alto nível técnico se endereçam mais particularmente aos engenheiros, técnicos, comerciais e supervisores. O CTP assume um compromisso com a pesquisa do futuro. O lançamento em 2005 dos pólos de competitividade, iniciativa alcançada pelo Estado de desenvolvimento de novas sinergias entre os centros de pesquisa e de treinamento e as empresas, confirmou sua oportunidade. O CTP escolheu se envolver plenamente nesta iniciativa, desempenhando um papel fundamental na montagem de projetos inovadores em cinco áreas:

- Minalogic (Rhône-Alpes)
- Química – Meio Ambiente (Rhône-Alpes)
- Fibras naturais Grande Leste (Alsace-Lorraine)
- Materiais de uso doméstico (Nord Pas-de-Calais)
- Pinheiro marítimo do futuro (Aquitaine)

O Programa Geral de Pesquisa é articulado, há vários anos, segundo quatro eixos ou programas principais: **Polpas Virgens, Fibras Recicláveis, Meio Ambiente e Qualidade dos Produtos.**

### 1.1.2 Situação Geográfica e socioeconômica

#### Sede

**Endereço :**  
Centre Technique du Papier  
Domaine Universitaire  
BP 251  
38044 GRENOBLE - Cedex 9  
FRANCE

**Números de telefone:**  
Tél: (33) (0)4 76 15 40 15  
Fax: (33) (0)4 76 15 40 16

Apesar de um ambiente internacional difícil (crise econômica), o volume de negócios do CTP se estabilizou a 11,5 M€. A porcentagem de financiamentos privados provenientes da indústria é responsável por 86% do volume de negócios sem contar doações.

#### Volume de negócios 11 M€ e Zoom sobre o PGR



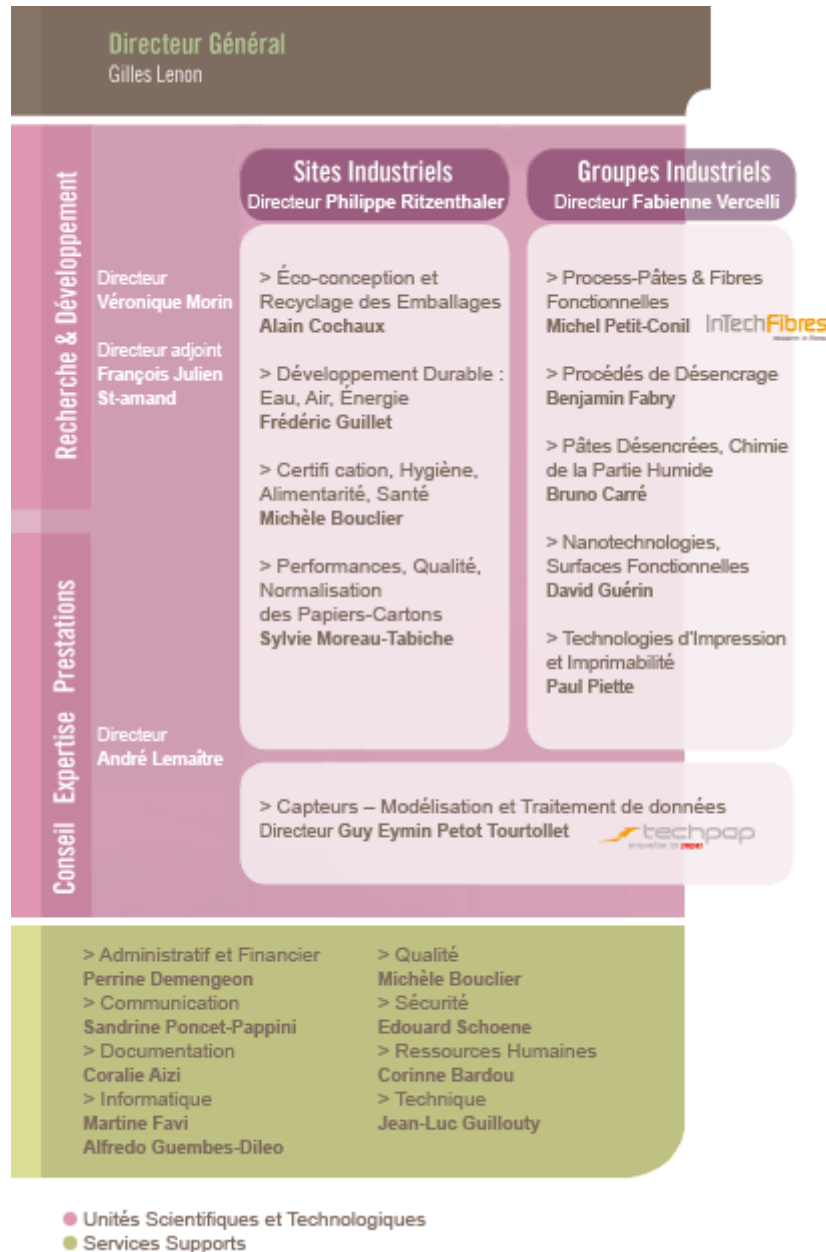
#### Pessoal

- 142 assalariados\*:
  - 58 Engenheiros (sendo 11 doutorandos remunerados)
  - 8 Engenheiros de Apoio

- 54 Técnicos de Pesquisa
- 22 Técnicos e Empregados de Apoio

\* Números de 01.01.2010

### 1.1.3 Organograma Funcional do CTP



## 1.2 Contexto e Especificações

Hoje em dia as ondas eletromagnéticas são onipresentes no nosso meio. Não se sabe ao certo o impacto delas sobre a saúde. Por precaução, novos meios de se proteger estão surgindo. Um deles é a utilização de filtros espaciais à base de superfície seletiva em frequência (FSS - *Frequency Selective Surface*). As características destes últimos, entretanto, dependem muito de seus padrões de base e convém então designar novos padrões com desempenhos mais adequados (como para as antenas).

## 1.3 Propósito do estágio

Este Estágio de Conclusão de Curso faz parte de um projeto chamado "**METAPAPIER**" que reúne o CTP, e os laboratórios IMEP (Instituto de Microeletrônica, Eletromagnetismo e Fotônica) e LCIS (Laboratório de Concepção e Integração de Sistemas) e visa desenvolver filtros espaciais para filtrar sinais GSM e Wi-fi utilizando FSS em papel de parede, ou seja, o papel será utilizado como substrato e os padrões, impressos por jato de tinta condutora. Há também uma tese em desenvolvimento desde janeiro de 2009 neste contexto pelo doutorando Fabien de Barros. O projeto foi dividido em três partes:

- Estudar as possibilidades de impressão dos nossos padrões em papel com tinta condutora.
- **Estudar as possibilidades de utilizar algoritmos genéticos para aperfeiçoar, ou criar, novos padrões de FSS no software CST® Microwave Studio.**
- Estudar as possibilidades de criar um demonstrador ou um meio de caracterização de nossos padrões de FSS com as menores bancadas de testes.

Este relatório corresponde à segunda parte (em negrito). Tem como objetivo simular as estruturas FSS, refletir sobre um bom método para se encontrar novos padrões para serem utilizados em filtros espaciais para se bloquear sinais de frequências bem conhecidas, como por exemplo, GSM (900 MHz, 1.8 GHz) e Wi-fi (2.4 GHz). Realizamos este estágio em três etapas.

Em um primeiro momento, foi realizado um estudo bibliográfico sobre a teoria de FSS e filtros espaciais, para se compreender as limitações desta tecnologia. Simultaneamente, foi feito um estudo da utilização do *software CST® Microwave Studio*.

Na segunda parte, o objetivo definido foi dominar o conhecimento e a utilização dos Algoritmos Genéticos, uma poderosa ferramenta para se encontrar soluções aproximadas para problemas de otimização. Estudamos também uma forma de realizar a comunicação (interface) entre CST e Matlab (o software escolhido para lançar o algoritmo genético) e fazer rodar as simulações de forma automática.

Para concluir, inúmeras execuções do algoritmo genético foram lançadas para se encontrar padrões eficientes. Procurou-se melhorar cada vez mais o algoritmo para aumentar sua velocidade de convergência e reduzir os custos computacionais.

## 2 Superfícies Seletivas em Frequência (FSS)

Nesta parte será iniciada a fundamentação teórica deste trabalho. Primeiramente, introduziremos a definição sobre as FSS, estruturas periódicas e filtros espaciais. Em seguida, será apresentada a teoria que rege esta tecnologia, como o Teorema de Floquet e técnicas de análises para FSS.

### 2.1 Definição e Histórico

As superfícies seletivas em frequência são filtros, essencialmente eletromagnéticos, que refletem algumas bandas de frequência e transmitem outras. Elas são estruturas periódicas bi ou tridimensionais, compostas por um conjunto de elementos idênticos chamados de célula básica (1)(2). Existem dois tipos fundamentais de FSS: um primeiro é composto por *pachtes* condutores sobre um substrato. Esta configuração normalmente se comporta como um filtro passa-baixas e é referenciada como uma FSS capacitiva. O segundo tipo consiste em aberturas (*slots*) em uma placa metálica, e comportamento de um filtro passa-altas e é referenciada como uma FSS indutiva (1).

A história das FSS começa em um contexto pós-guerra, no final dos anos 50, com os esforços da indústria bélica para reduzir a RCS (*Radar Cross Section*) dos veículos militares (1). A RCS é uma medida de detecção de um objeto por um radar. Quanto maior a RCS, mais facilmente um objeto pode ser detectado (Figura 2.1).

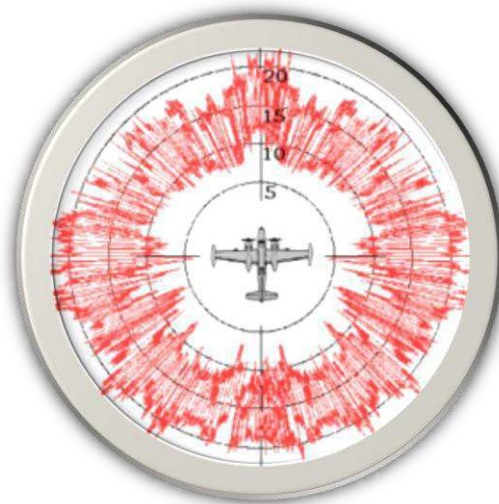


Figura 2.1 - Radar Cross Section (RCS)

Na verdade, o princípio geral vem do início do século XX. A referência mais antiga sobre a FSS pertence à Marconi e Franklin em uma patente datada de 1919. Eles produziram um refletor parabólico composto de fios metálicos no lugar de longas hastes.

As aplicações de FSS aumentaram bastante durante as últimas décadas. No domínio militar, podemos citar as redomas, os mísseis, a tecnologia *Stealth* (Figura 2.2) e satélites.

Como aplicações para fins pacíficos, as FSS podem assegurar a confidencialidade dos dados de uma rede sem fio em um prédio bloqueando a saída dos sinais para o exterior. Sabendo que várias redes sem fio uma ao lado das outras criam perturbações (3), as FSS podem solucionar este problema também. Nas casas, as FSS estão presentes nos fornos a microondas para bloquear as ondas nocivas à nossa saúde. Outra aplicação doméstica interessante é uma janela inteligente, capaz de bloquear as radiações infravermelhas, deixando-se passar luz e sinais de comunicações (Wi-fi, GSM, UTMS, etc.) (4).



Figura 2.2 - F-117 Nighthawk (US Airforce), o primeiro avião do mundo a utilizar tecnologia Stealth, possuindo estruturas FSS na sua carcaça.

## 2.2 Estruturas Periódicas e Teorema de Floquet

Uma estrutura periódica é basicamente um conjunto de elementos idênticos dispostos em uma matriz infinita em uma ou duas dimensões. Tais estruturas podem ser excitadas de duas formas: de forma passiva, por uma onda plana incidente  $E^i$  (caso abordado no escopo do nosso projeto) ou por geradores individuais conectados a cada elemento, forma ativa (1).

No caso passivo (Figura 2.3), a onda plana incidente será parcialmente transmitida na direção  $E^t$  e parcialmente refletida na direção  $E^r$ . Em condições ideais de ressonância, a amplitude do sinal refletido deve ser igual a  $E^i$  enquanto  $E^t = 0$ .



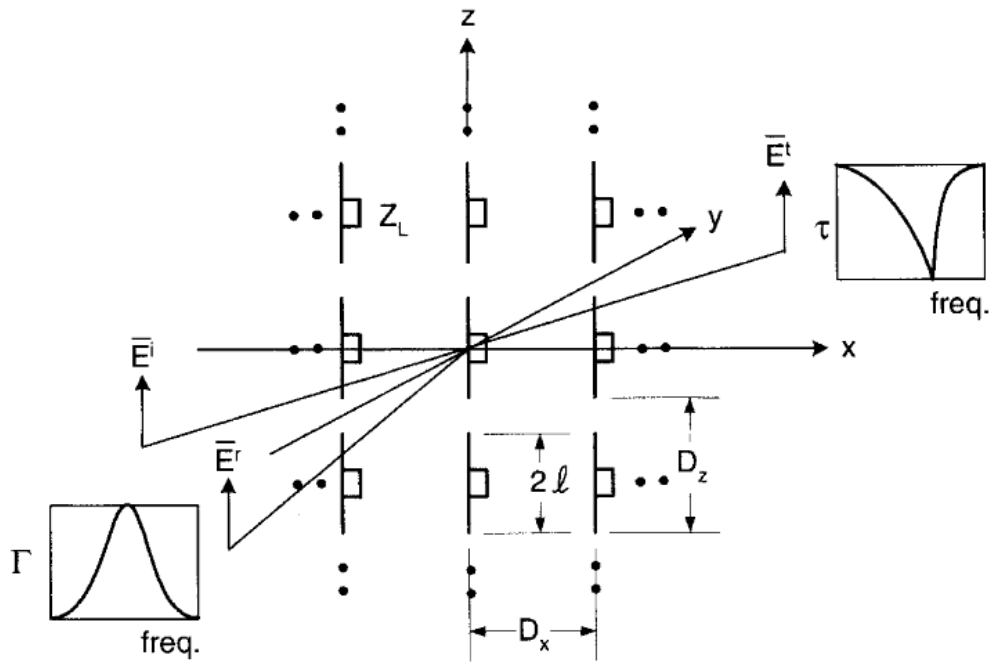


Figura 2.3 - Estrutura periódica excitada por uma onda plana incidente  $E^i$  (Extraída de (1))

A partir destas tensões podemos calcular dois parâmetros fundamentais em nossos estudos: coeficiente de reflexão (E2.1) e coeficiente de transmissão (E2.2).

$$\Gamma = \frac{E^r}{E^i} \quad (E2.1)$$

$$\tau = \frac{E^t}{E^i} \quad (E2.2)$$

Uma vez que as FSS estão arranjadas de forma periódica, é possível descrever o campo no filtro em termos de uma célula unitária. Segundo (5) O Teorema de Floquet determina que se uma equação diferencial linear tem coeficientes periódicos e condições de fronteira periódicas, então soluções estáveis serão geralmente uma função periódica vezes uma função exponencial decrescente. Aplicando-se este teorema para as estruturas periódicas, verificamos que os valores dos campos elétrico e magnético em um ponto arbitrário da estrutura conduzirá aos mesmos valores em um ponto similar de outra célula unitária exceto por um fator de propagação  $e^{-\gamma d}$  de uma célula à célula vizinha, onde  $d$  é o comprimento de uma célula.

Se o campo em uma célula unitária entre  $0 \leq z \leq d$  é  $\mathbf{E}(x, y, z)$  e  $\mathbf{H}(x, y, z)$ , o campo em uma célula unitária localizada na região  $d \leq z \leq 2d$  deve ser

$$e^{-\gamma d} \mathbf{E}(x, y, z - d), e^{-\gamma d} \mathbf{H}(x, y, z - d) \quad (\text{E2.3})$$

Desta forma, o campo em uma estrutura periódica pode ser descrito da seguinte forma:

$$\mathbf{E}(x, y, z) = e^{-\gamma z} \mathbf{E}_p(x, y, z) \quad (\text{E2.4a})$$

$$\mathbf{H}(x, y, z) = e^{-\gamma z} \mathbf{H}_p(x, y, z) \quad (\text{E2.4b})$$

Onde  $\mathbf{E}_p$  e  $\mathbf{H}_p$  são funções periódicas de  $z$  com período  $d$ . Em um estudo mais detalhado (5), que foge ao escopo deste trabalho, foi demonstrado que o campo em uma estrutura periódica pode ser representado como:

$$\mathbf{E}(x, y, z) = \sum_{n=-\infty}^{\infty} E_{pn}(x, y) e^{-\beta_n z} \quad (\text{E2.5})$$

Onde  $\gamma = j\beta$  e  $\beta_n = \beta + 2n\pi/d$ .

Os métodos para analisar as FSS podem ser divididos em várias categorias: técnicas de análise de circuitos, técnicas de expansão nodal e técnicas iterativas.

Nas técnicas de expansão nodal (a mais empregada atualmente), modos de Floquet no espaço são combinados com partes das estruturas de FSS (aberturas ou correntes) para formar uma equação integral, que pode ser resolvida por técnicas computacionais tais como Método dos Momentos ou técnicas de gradiente conjugado. Estas técnicas não serão detalhadas neste projeto, uma vez que foi utilizado um software de simulação para a análise dos padrões de FSS estudados, que emprega tais técnicas para obtenção dos resultados.

### 2.3 Como funcionam os filtros espaciais

Segundo (6), uma fonte de onda plana incidente (à esquerda) atingindo um fio metálico com incidência normal (Figura 2.4a). Se imaginarmos que há um só elétron repousando neste fio, como o fio é ortogonal ao vetor de Poyting, o vetor  $\mathbf{E}$  da fonte se situa no mesmo eixo. Este vetor exerce uma força no elétron e causa uma oscilação. Parte da energia da fonte deve assim ser convertida em energia cinética mantendo o elétron em movimento oscilatório. Se toda a energia da fonte é absorvida pelos elétrons, a transmissão através deste filtro será igual a zero.

Se imaginarmos a situação da Figura 2.4b, a força exercida pelo campo  $\mathbf{E}$  não poderá por em oscilação o elétron uma vez que o fio não está situado no mesmo eixo que o campo elétrico.

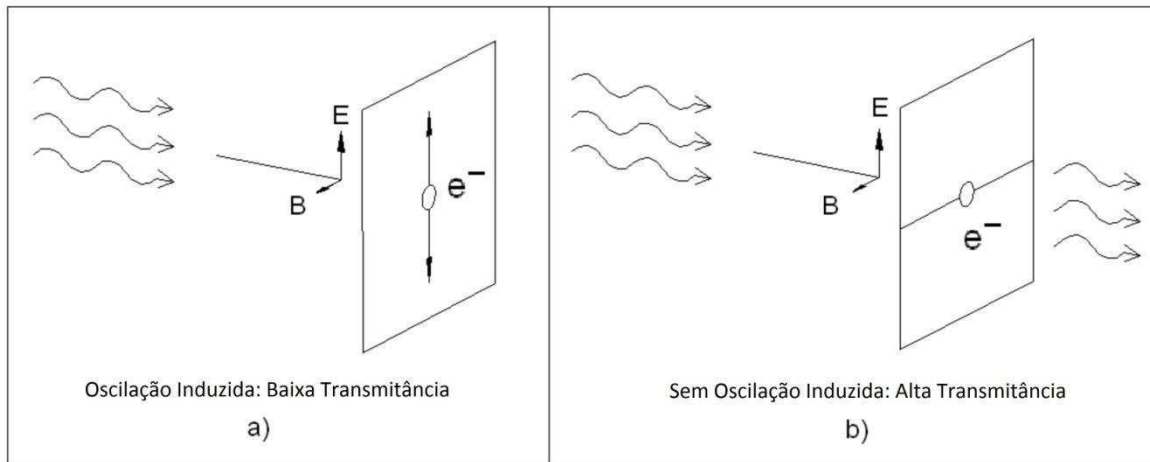


Figura 2.4 - Princípio de funcionamento de um filtro espacial; Extraído de (6)

## 2.4 Tipos de Elementos de FSS

Os elementos de FSS são classificados em 4 grupos a saber (1) (Figura 2.5):

- **Grupo 1** : Elementos de centro conectado ou N-polos, como os dipolos, tripolos, cruz simples e de Jerusalém, tipo âncora, quadrado espiralado, etc..
- **Grupo 2** : Elementos tipo laço, como os quadrados, anéis, pentágonos, hexágonos "ocos", ou figuras que sugiram um "caminho fechado".
- **Grupo 3** : Elementos de interior sólido ou placas de vários formatos.
- **Grupo 4** : Elementos gerados pela combinação dos elementos dos grupos anteriores.



Figura 2.5 - Elementos típicos de FSS e suas classificações

## 2.5 Princípio de Babinet

Como já apresentado, as FSS podem ser definidas como um *array* de aberturas (*slots*) ou de *patches* condutores. Se tomarmos duas estruturas de tipos diferentes, e estas forem complementares, ou seja, se colocadas uma em cima da outra, e obtivermos um plano perfeitamente condutor, então em condições ideais, a curva para o coeficiente de transmissão ( $\tau$ ) de uma estrutura, será igual à curva do coeficiente de reflexão ( $\Gamma$ ) da sua estrutura complementar (Figura 2.6).

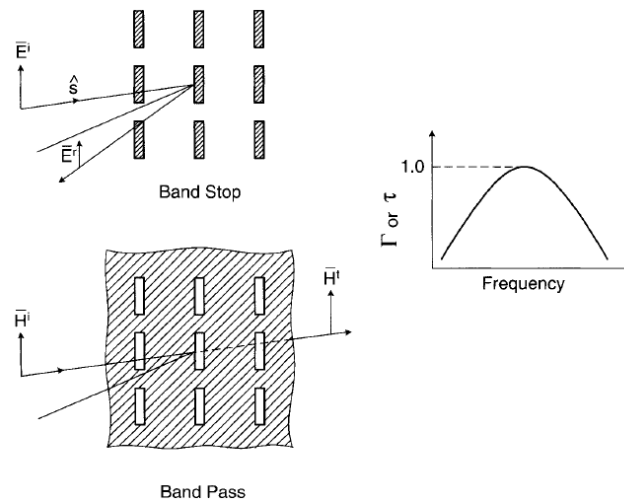


Figura 2.6 - Exemplo de estruturas complementares para exposição do Princípio de Babinet. Extraída de (1).

As condições para que este princípio seja válido são: estrutura infinitamente fina e condutor perfeito; estrutura "imersa" em vácuo; e estrutura confeccionada em uma única folha metálica.

Este princípio é bastante explorado pelos projetistas de FSS. Uma vez que uma estrutura carrega implicitamente as características da sua estrutura complementar, só precisamos analisar um dos dois casos, reduzindo pela metade o espaço de busca do problema.

## 2.6 Conclusão

Nesta seção foi apresentada uma abordagem sucinta das FSS. Descreveu-se também uma ligeira abordagem matemática para análise destas estruturas. Também se apresentou nesta seção conceitos sobre transmissão de ondas planas incidentes, e os parâmetros coeficientes de transmissão e reflexão, bem como o princípio de Babinet, que foram bastante explorados no escopo deste trabalho.

### 3 Algoritmos Genéticos

A computação evolutiva é uma área de pesquisa que vem ganhando espaço rapidamente. A principal razão deste crescimento é o desenvolvimento de algoritmos capazes de encontrar soluções adequadas para problemas de alta complexidade (que inclui problemas no domínio do eletromagnetismo) ainda não resolvidos de forma analítica ou por outras técnicas computacionais.

Os algoritmos evolutivos se baseiam em mecanismos da evolução biológica. O carro chefe destes algoritmos são os Algoritmos Genéticos (AG), que será abordado nesta seção.

#### 3.1 Histórico

Os algoritmos genéticos foram inspirados pela teoria evolucionista de Darwin. Sua invenção é creditada a John Holland. Em 1975 ele publicou "*Adaptation in Natural and Artificial*" (7), a primeira grande obra sobre este assunto. Em 1992, John Koza (8) utilizou os Algoritmos Genéticos para certas aplicações e introduziu o termo "Programação Genética". Devido à simplicidade de implementação e eficácia, os algoritmos genéticos são amplamente empregados para resolver problemas de otimização, como por exemplo, os problemas NP (problema do caixeiro viajante, mochila, etc.)

#### 3.2 Definição e Princípio

Em sua obra, John Holland definiu os algoritmos genéticos como modelos computacionais que imitam os mecanismos da evolução natural para resolver problemas de otimização (7). Eles são fundamentados nos mecanismos da seleção natural e genética, e os problemas são resolvidos por um processo evolutivo que retorna a melhor (mais adequada) solução (a sobrevivente). É um algoritmo essencialmente estocástico, e a solução encontrada é aproximada (normalmente não se sabe qual é a melhor solução). Sua ligação com a genética e terminologia é apresentada por uma infografia (Figura 3.1).

O funcionamento de um AG se assemelha bastante ao princípio da teoria evolutiva de Darwin e as leis da genética de Mendel. De início, geramos aleatoriamente uma população inicial com vários indivíduos, onde cada um representa uma solução. Os **pais** correspondem aos membros da população corrente, e podem através de reprodução (**crossover**), gerar **descendentes** que farão parte de uma nova **geração**. Cada geração é uma nova população gerada e corresponde a uma iteração do algoritmo.

Cada indivíduo porta um **cromossomo** que é a representação codificada (**codificação**) de uma solução (normalmente um vetor binário). Eles são compostos de **genes**, que portam as informações genéticas do indivíduo.

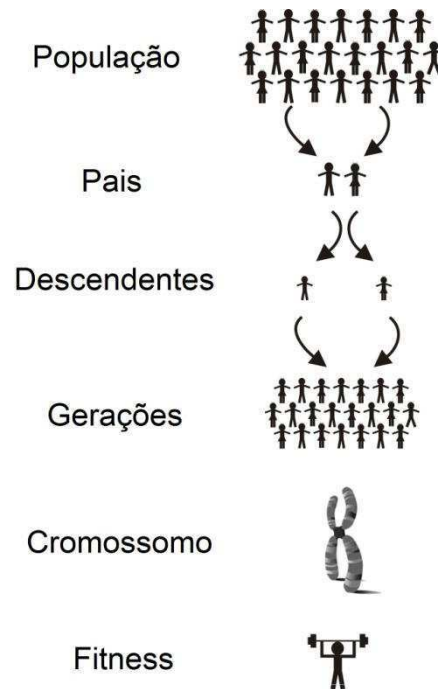


Figura 3.1 - Terminologia do Algoritmo Genético esquematizada. Extraída de (8).

De uma forma simples de se compreender o mecanismo dos AG, uma população gerará novas gerações, de onde se espera obter indivíduos cada vez melhores (mais aptos adaptados ao ambiente). Esta aptidão ou desempenho do indivíduo é representada por seu *fitness*, um número positivo que o qualifica como uma espécie de nota, quanto maior, mais forte é o indivíduo. Podemos resumir o funcionamento de um algoritmo simples pelo esquema apresentado na Figura 3.2.



Figura 3.2 - Esquema do funcionamento de um algoritmo genético simples

### 3.3 Componentes de um algoritmo genético

Nesta seção apresentaremos os componentes necessários ao funcionamento do algoritmo genético.

Um algoritmo genético é composto por:

- Um problema de otimização.
- Um método de codificação para os cromossomos.
- Uma função de avaliação para se calcular o valor de *fitness* de cada solução.
- Um método para se gerar a população inicial.
- Um grupo de parâmetros (taxas de crossover, mutação, elitismo, etc.).
- Um conjunto de operadores para os mecanismos de reprodução.

#### 3.3.1 Operadores genéticos e codificação de um algoritmo genético

Com o processo de reprodução é possível gerar novos indivíduos a partir de indivíduos ascendentes (pais). Uma população irá gerar uma nova geração com o intuito de encontrar indivíduos (soluções) ainda melhores. Durante este processo, os pais misturam seus genes para gerar filhos, que podem ainda sofrer **mutação**. Um conjunto de genes forma os **cromossomos**, que podem ser codificados de diferentes maneiras.

### 3.3.1.1 Codificação dos cromossomos

A codificação aparece no início do problema e possui grande importância. Como codificar os cromossomos para melhor se adaptar às nossas exigências? Uma boa codificação é determinante para a eficácia do algoritmo genético.

A codificação mais simples e mais utilizada é a **codificação binária**, onde cada cromossomo é representado por uma cadeia de bits 0 ou 1, como mostrada na (Figura 3.3).

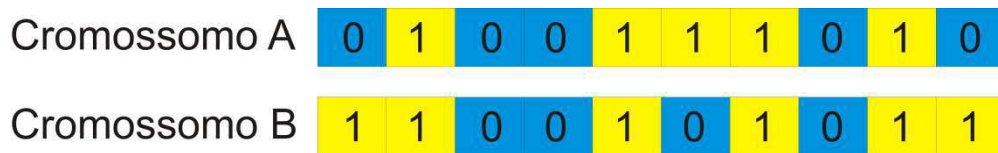


Figura 3.3 - Codificação Binária

Esta codificação foi a primeira a ser utilizada, devido à sua simplicidade. Um exemplo de problema resolvido por esta codificação é o problema da mochila. Uma lista de objetos de diferentes tamanhos e preços nos é dada para se preencher uma mochila de capacidade limitada. O objetivo é encher a mochila de modo a maximizar o valor dos objetos no seu interior. Cada posição (lócus) corresponde a um objeto, e o valor 1 ou 0 à presença ou ausência do objeto, respectivamente.

Outro tipo de codificação utilizada, um pouco mais complexa é a **codificação por permutação**. Ela é utilizada para problemas de ordenação, como por exemplo o problema do caixeiro viajante. Cada lócus representa a ordem de visita e porta um número correspondente a uma cidade.

Por fim, a **codificação de valores**, utilizada quando a codificação binária não é suficiente para representar valores mais complicados. Cada cromossomo é uma seqüência de valores ligados com o problema. Por exemplo, números reais, caracteres, palavras. Para se aplicar operações de crossover e mutação, deve-se desenvolver métodos especiais.

### 3.3.1.2 Crossover

Também conhecido como recombinação ou cruzamento, a idéia do crossover é se conseguir combinar as boas características de dois indivíduos diferentes e transmitir aos seus descendentes. Esta operação funciona da seguinte forma (Figura 3.4). Em um caso mais simples, (Figura 3.4a) (crossover de um único ponto) dois pais selecionados de uma população trocam uma porção de seus cromossomos a partir de um ponto de suas cadeias escolhido ao acaso e geram dois descendentes para a nova população.



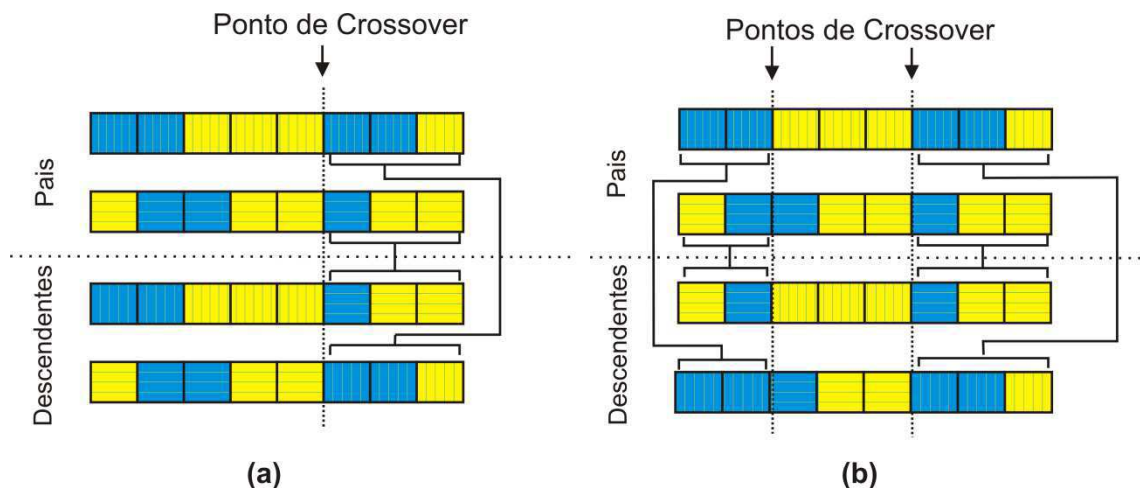


Figura 3.4 - Crossover (a) Um único ponto (b) A dois pontos

A ocorrência ou não de crossover depende de um parâmetro chamado **probabilidade de crossover** ( $p_{cross}$ ). Por exemplo, um número real  $p$  entre 0 e 1 é gerado ao acaso. Se  $p < p_{cross}$ , então o crossover será realizado.

Para o crossover a múltiplos pontos (Figura 3.4b) o processo é semelhante, diferindo apenas em se gerar dois pontos de crossover ao acaso. O valor típico de  $p_{cross}$  varia entre 60% e 80%(9).

### 3.3.1.3 Mutação

A mutação é um operador importante para se aumentar a variabilidade genética do ambiente. Imaginemos certa função  $y = f(x)$  (Figura 3.5) repleta de valores de máximo local, onde  $x$  corresponde às soluções e  $y$  os valores de **fitness** correspondentes e se deseja encontrar a solução  $x$  que apresente o maior **fitness**. Cada ponto em azul representa um indivíduo de uma população. Neste caso, há uma aglomeração de indivíduos próximos ao máximo local circulado em vermelho que gerarão novos indivíduos com características semelhantes, o que poderá levar o algoritmo a convergir prematuramente para este valor. Com a mutação, é possível se conseguir levar as soluções para outros valores de máximo.

A mutação consiste em varrer toda a cadeia de um cromossomo, e substituir um gene por outro aleatoriamente segundo uma **taxa de mutação** ( $p_{mut}$ ) (Figura 3.6).

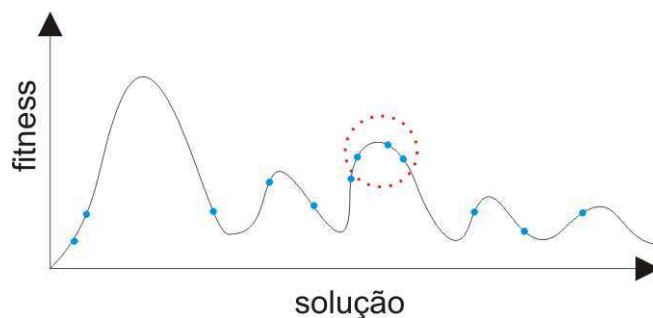


Figura 3.5 - Função hipotética para ilustrar a ideia da mutação

O valor de  $p_{mut}$  deve ser pequeno (0,1% a 1%) (9) a fim de evitar que a busca se torne um processo aleatório, conservando-se também os princípios da seleção e da evolução de Darwin.



Figura 3.6 - Mutaçãõ de 2 genes ao acaso

### 3.3.2 Estrat\u00e9gias de seleçãõ

Para gerar os descendentes naturalmente \u00e9 necess\u00e1rio se escolher dois pais dentre todos os indiv\u00edduos de uma populaçãõ a cada cruzamento. Existem v\u00e1rias maneiras de realizar esta seleçãõ. Algumas s\u00e3o mais r\u00e1pidas, outras s\u00e3o fortemente baseadas no *fitness* dos indiv\u00edduos. A seguir apresentaremos as estrat\u00e9gias de seleçãõ mais conhecidas.

#### 3.3.2.1 Seleçãõ por Roleta

Tomamos uma populaçãõ com  $n$  indiv\u00edduos, os pais ser\u00e3o selecionados segundo um "giro" de uma roleta ponderada por valores de *fitness* de cada cromossomo, como representada pela Figura 3.7. A \u00e1rea de cada fatia da roleta \u00e9 proporcional ao valor de *fitness* de seu cromossomo correspondente. Quanto maior o valor do *fitness* do indiv\u00edduo, maiores ser\u00e3o suas chances de ser selecionado.

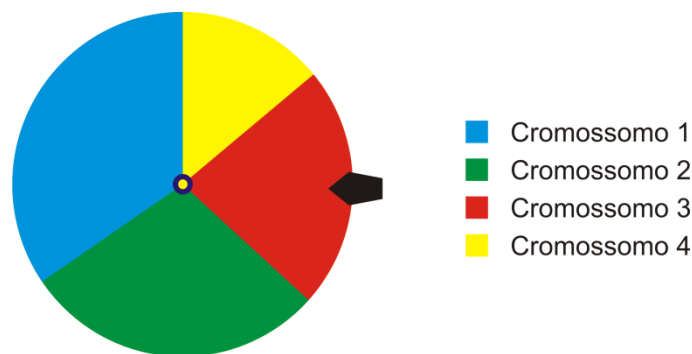


Figura 3.7 - Exemplo de uma seleçãõ por roleta

Este m\u00e9todo apresenta um inconveniente quando existem disparidades entre os valores de *fitness* dos indiv\u00edduos. Por exemplo, se um cromossomo ocupa 80% da roleta. Neste caso, haver\u00e1 cromossomos com pouqu\u00edssimas chances de serem selecionados, o que poder\u00e1 causar uma estagnaçãõ da evoluçãõ e converg\u00eancia prematura para certa soluçãõ.

#### 3.3.2.2 Seleçãõ por Torneio

Este m\u00e9todo consiste em escolher  $n$  (normalmente 2) indiv\u00edduos aleatoriamente na populaçãõ e selecionar um pai dentre eles (com grande probabilidade de vit\u00f3ria do mais

forte)(Figura 3.8). Este método é mais rápido (Complexidade  $O(n)$  em relação a  $O(n^2)$  da Roleta)(9), combate a perda da diversidade na população e proporciona uma ligeira melhora evitando uma convergência prematura nos estágios iniciais do processo de otimização.

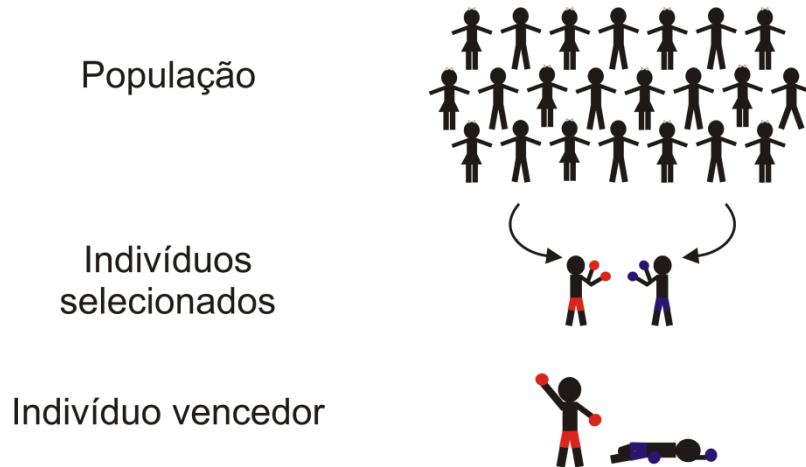


Figura 3.8 - Seleção por Torneio

Para gerar  $n_{pop}$  indivíduos, é necessário se fazer  $n_{pop}$  "disputas", de onde vem uma probabilidade de vitória do melhor indivíduo, que deve ser superior a 70%. Um valor de  $n$  maior que dois pode ser escolhido, mas isto pode aumentar o risco de convergência prematura.

### 3.3.2.3 Elitismo

O método do elitismo é fundamental. A cada geração, replicamos uma cópia do melhor (ou melhores) cromossomo para se assegurar que as boas soluções não irão se perder com o avanço das iterações, e que seus genes possam ser transmitidos às futuras gerações, o que representa uma melhora realmente considerável ao processo.

### 3.3.3 Função de avaliação

De fato, os algoritmos genéticos são estocásticos, fáceis de implementar, uma vez que não é necessário se dominar o assunto do problema. Podemos dizer que ele trabalhar de maneira escrava.

A função de avaliação é única ligação entre o problema real e o algoritmo genético (9), por isso sua enorme importância. Ela é responsável por calcular, atribuir o valor de *fitness* de cada indivíduo, classificar uma solução e permitir criar um *ranking* das melhores soluções. Escolher uma boa função de avaliação aparece com o maior desafio em um algoritmo genético.

Uma boa função de avaliação possui as seguintes características:

- Tempo de cálculo pequeno, visto que será executada para cada indivíduo separadamente.

- Deve ser capaz de classificar os indivíduos a uma boa proporção (boa distinção entre boas e más soluções).

### 3.4 Quando se utilizar os Algoritmos Genéticos?

Os algoritmos genéticos são amplamente utilizados em diversas áreas, como por exemplo: acústica, engenharia aeroespacial, geofísica, robótica, química, astronomia, aplicações militares, biologia molecular, etc.

Não existe um consenso sobre quando se utilizar os algoritmos genéticos, mas vários pesquisadores concordam nos seguintes aspectos (10):

- Grande espaço de busca: Quando testar todas as soluções é impraticável para se encontrar o ponto ótimo global.
- Se não é necessário se encontrar uma solução ótima: Se o usuário deseja encontrar uma boa solução em um curto espaço de tempo, visto que os Algoritmos genéticos podem encontrar ou não um ótimo global.
- Espaço de busca complexo: Funções de avaliação descontínuas, que apresenta muitas variações ao longo do tempo.
- Problema com vários parâmetros: Vários problemas do mundo real não podem ser descrita com apenas uma só variável. Os algoritmos genéticos são realmente indicados para resolver problemas de múltiplos objetivos, dado o seu paralelismo.

### 3.5 Limitações dos Algoritmos Genéticos

Os algoritmos genéticos provaram que podem resolver muitos problemas, mas apresentam certas limitações (10):

- Definir uma representação de problema: a linguagem utilizada deve ser robusta, apta a tolerar mudanças aleatórias como erros fatais e resultados sem sentido.
- Escrever uma função de avaliação: Uma função pobre ou definida de modo impreciso pode levar o algoritmo a resolver o problema incorretamente.
- Parâmetros a escolher: Os parâmetros, como tamanho da população, taxa de crossover e mutação, tipo de seleção deve ser escolhida com cuidado. Uma pequena população pode não explorar uma parte do espaço de busca suficiente. Sabemos que a primeira população deve ser numerosa para permitir se encontrar pelo menos uma boa população para transmitir bons genes às futuras gerações. Normalmente deve-se lançar o algoritmo algumas vezes para verificar o comportamento e em seguida lançar o algoritmo de forma definitiva.
- Possibilidade de convergência imatura: Se um indivíduo de uma população é bem melhor que seus competidores, uma escolha de estratégia de seleção pode diminuir

a diversidade da população, levando o algoritmo a convergir a um ótimo local. Este problema acontece mais freqüentemente com pequenas populações.

- Resolver problemas analíticos: Vários pesquisadores (7)(11) advertem contra a utilização dos algoritmos genéticos para resolver os problemas analíticos. Apesar de eles poderem encontrar boas soluções, os métodos analíticos tomam menos tempo e esforços computacionais.

### 3.6 Conclusão

Neste capítulo foi apresentado o Algoritmo Genético, uma ferramenta muito poderosa para se encontrar soluções em problemas de alta complexidade. Foi apresentada sua terminologia, os operadores genéticos, suas limitações, aplicações.

Este algoritmo é de fácil implementação. Um algoritmo genético não muda acentuadamente de uma aplicação para outra, exceto a sua função de avaliação. Encontrar uma boa função de avaliação, e uma boa sinergia entre os parâmetros são as maiores dificuldades presentes nos Algoritmos Genéticos.

## 4 CST ® *Microwave Studio*

Neste projeto, todas as simulações foram lançadas no CST ® *Microwave Studio* 2009, um *software* de simulações eletromagnéticas baseado na "Técnica de Integração Finita", proposta por Thomas Weiland em 1977 (12). Este método consiste em um esquema de discretização espacial (malha) para resolver problemas de campo eletromagnético nos domínios do tempo e da frequência. Dada uma estrutura, ele a discretiza em pequenas células tetraédricas ou hexaédricas e depois calcula os campos por cada célula utilizando as Equações de Maxwell sob a forma integral. Quanto melhor a resolução da malha, melhor será a precisão do cálculo, mas em contrapartida, o tempo de cálculo aumenta proporcionalmente.

O CST apresenta quatro diferentes métodos de simulação para estruturas de FSS: *Transient Solver*, *Frequency Domain Solver* e *Eigenmode Solver*. Neste projeto, analisamos as possibilidades de utilização do primeiro e segundo métodos.

Utilizando-se o *Transient Solver*, as simulações demandam um tempo menor, além de o CST permitir a utilização de todos os núcleos de processador de computador em paralelo. As desvantagens que nos fizeram abandonar a utilização deste método são: a instabilidade, que requer bastante conhecimento para se conseguir ajustar de forma correta os parâmetros de simulação e a impossibilidade de simulação de estruturas, levando-se em conta a variação do ângulo de incidência da onda ( $\theta$ ) (Figura 4.1), fundamental na obtenção dos objetivos deste projeto. Assim o método utilizado foi *Frequency Domain Solver*.

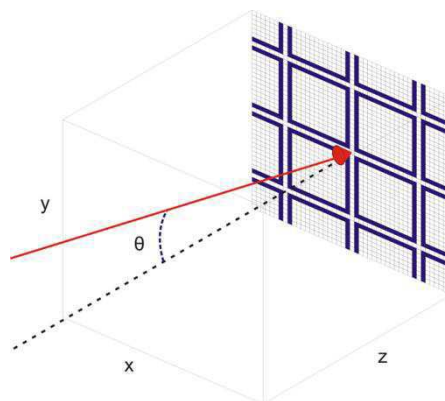


Figura 4.1 - Ângulo de incidência  $\theta$  da onda plana atravessando uma estrutura FSS.

Para melhorar o tempo de simulação, visto que devemos simular várias estruturas, a resolução da malha foi reduzida.

### 4.1 Ambiente FSS em CST *Microwave Studio*

O caso específico da simulação de estruturas FSS será mostrado, passo a passo. Inicialmente escolhe-se um novo projeto do tipo *FSS – Unir Cel. (FD)*, (Figura 4.2).

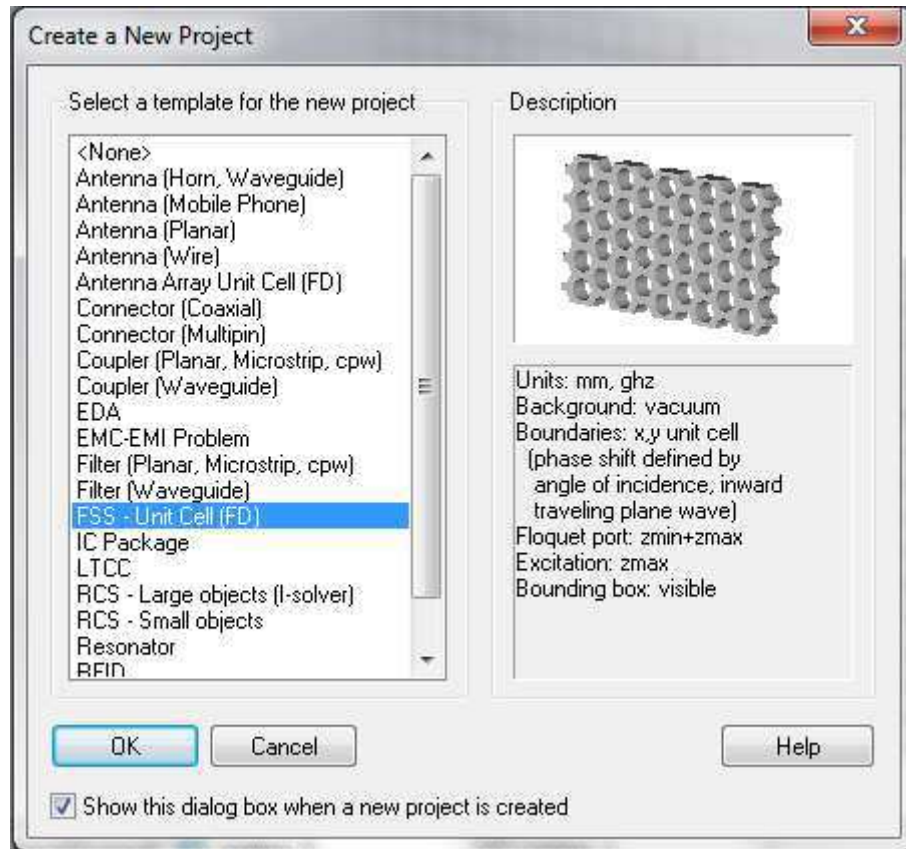


Figura 4.2 - Criação do novo projeto

As simulações se desenrolam no "solver" freqüencial (FDSolver). As condições de fronteira são do tipo *unit cell*, e o cálculo das Equações de Maxwell são realizados sobre uma só célula, em seguida repetida para todas as outras células (Figura 4.3), deste modo as FSS trabalham sobre as estruturas periódicas.

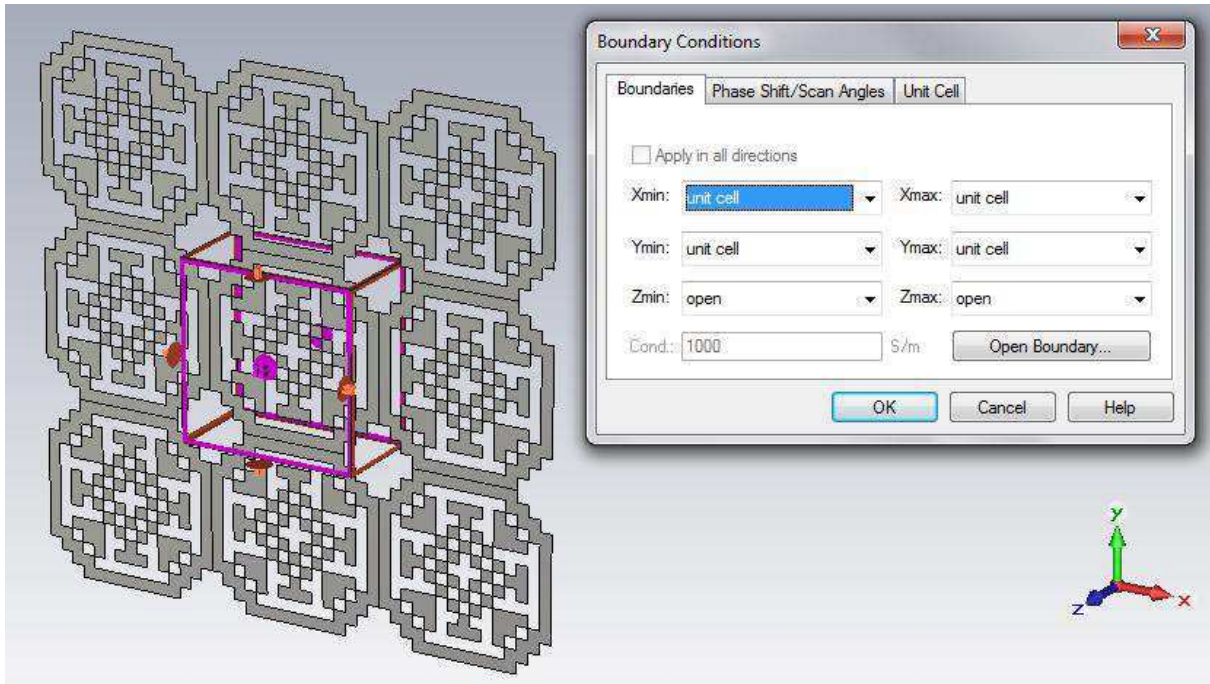


Figura 4.3 - Condições de Fronteira

Para assegurar a veracidade de uma simulação, deve-se escolher um bom número de modos de Floquet sobre as portas, aqui chamadas ( $Z_{min}$ ,  $Z_{max}$ ) (Figura 4.4).

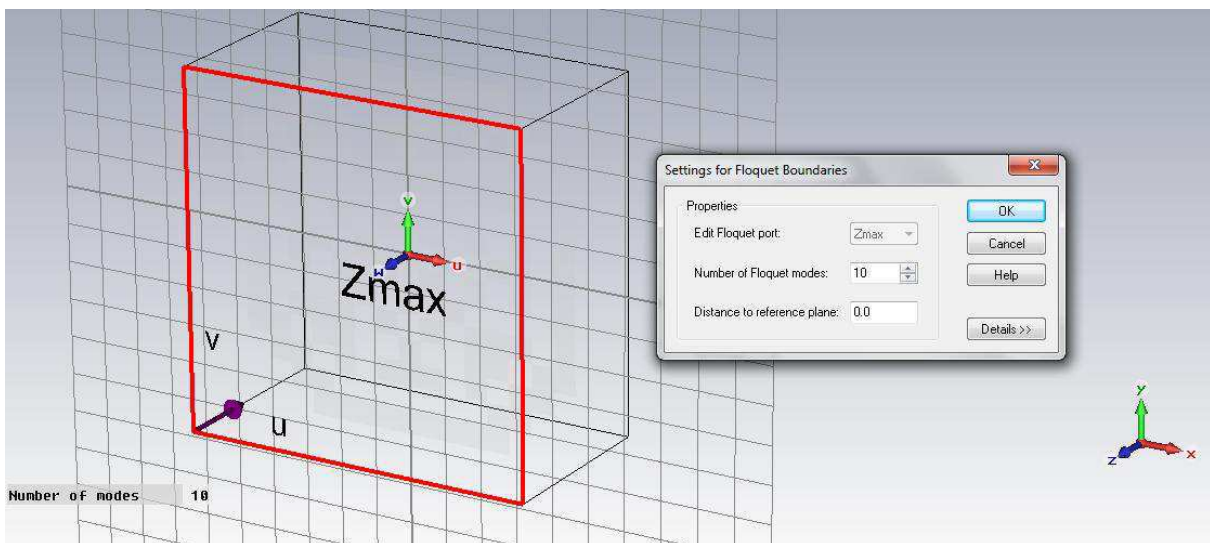


Figura 4.4 - Escolha do número de modos de Floquet

## 4.2 Trabalhando com macros

Após estudar as ferramentas de construção das estruturas para o desenho dos filtros espaciais, o desafio foi realizá-los de forma automática, uma vez que desenhá-los um a um "à mão" é impraticável.



O estudo das macros aparece como uma poderosa solução para este problema. De fato, todas as operações e tarefas feitas em CST são memorizadas em um histórico chamado *History List*. A partir deste histórico, podemos recuperar os códigos referentes às ações realizadas em linguagem de programação *Visual Basic* (VBA) (Figura 4.5).

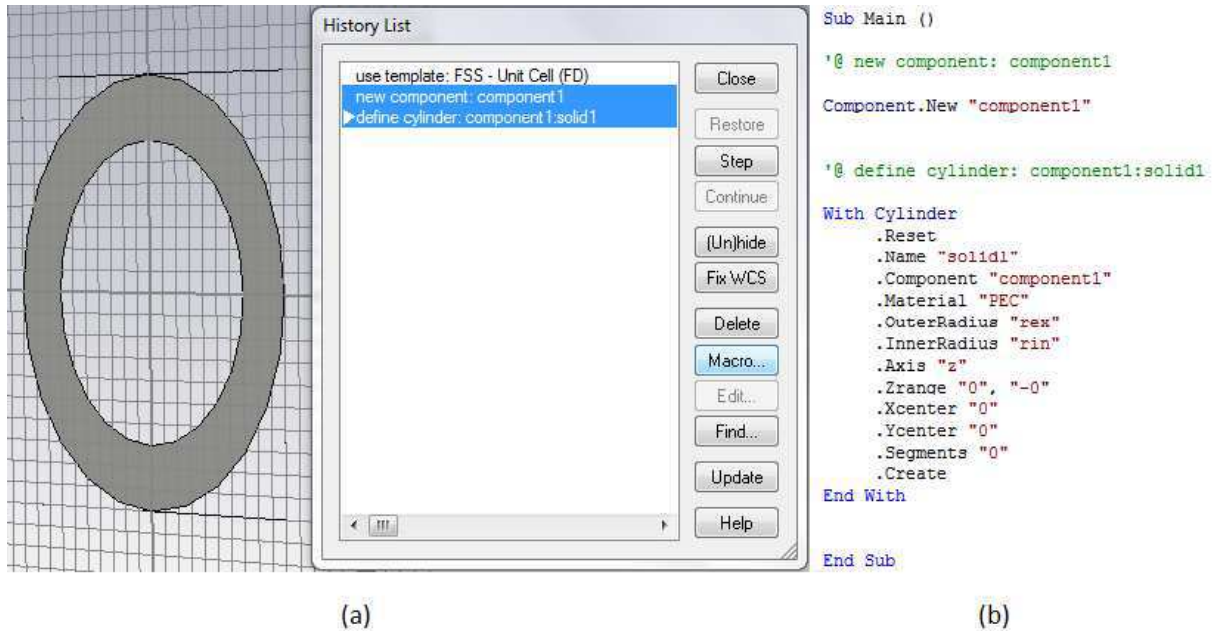


Figura 4.5 - (a) Operações realizadas para a construção de um anel simples no History List; (b) código responsável para estas operações geradas em linguagem VBA.

De posse destas informações, é possível utilizar outro *software*, como o Matlab para controlar o CST e fazê-lo realizar um conjunto de tarefas desejadas pelo programador, e assim, o CST trabalhará de forma automática.

### 4.3 Conclusão

Nesta breve seção, apresentamos de forma bem rápida o software de simulação eletromagnética CST *Microwave Studio*, mostrando mais especificamente o ambiente de simulação para Estruturas FSS.

## 5 Simulações e Construção do Algoritmo

Após a apresentação da fundamentação teórica, mostraremos nesta seção o desenvolvimento do projeto, desde as primeiras simulações de padrões simples, como anéis, quadrados, etc. até estruturas mais complexas como as geradas pelo algoritmo genético construído em Matlab.

### 5.1 Simulação dos padrões FSS de primeiro e segundo grupo

#### 5.1.1 Tripolos simples imbricados

O padrão escolhido para analisar foi um tripolo simples pertencente ao Grupo 1, segundo a classificação de Munk(1). Para que haja ressonância da onda plana incidente, é necessário que a dimensão das duas pernas do tripolo somadas totalizem aproximadamente a  $\lambda_r/2$ , onde  $\lambda_r$  é o comprimento de onda da onda plana incidente.

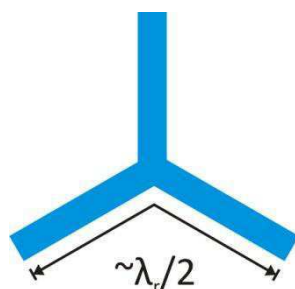


Figura 5.1 - Dimensão aproximada para obter a ressonância para um tripolo simples

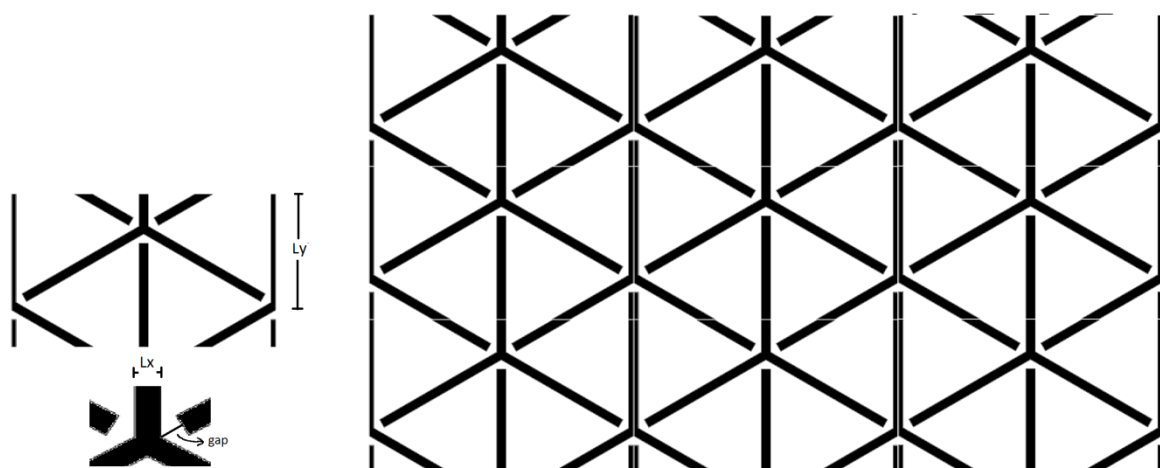


Figura 5.2 - Célula elementar e estrutura FSS para os tripolos imbricados

Os parâmetros (dimensões) escolhidos nas simulações estão mostrados na Tabela 1.

Lx (mm)	Ly (mm)	Gap (mm)
4	60	4

Tabela 1 - Dimensões escolhidas para as simulações dos tripolos simples imbricados.

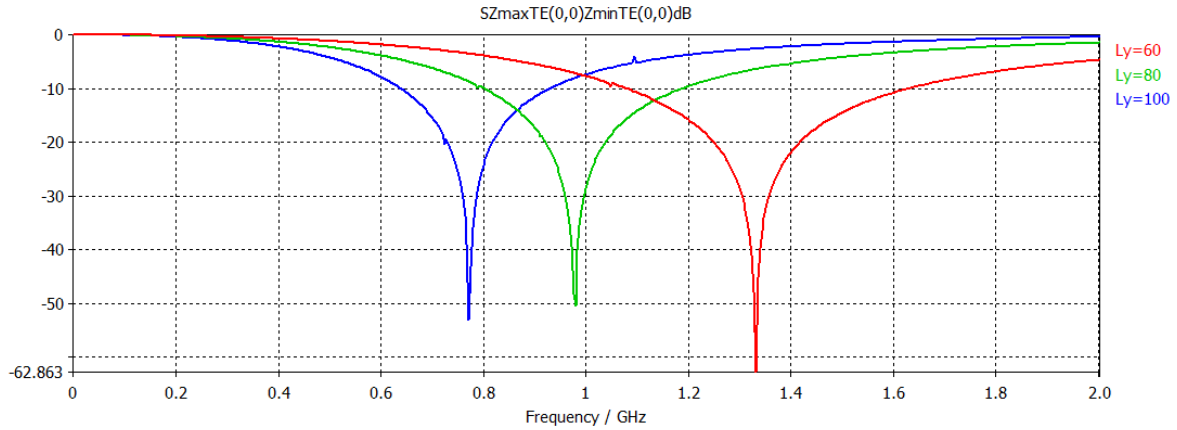


Figura 5.3 - Coeficiente de Transmissão ( $\tau$ ) para os valores de comprimento do braço do tripolo seguintes: 60 mm ; 80 mm ; 100 mm

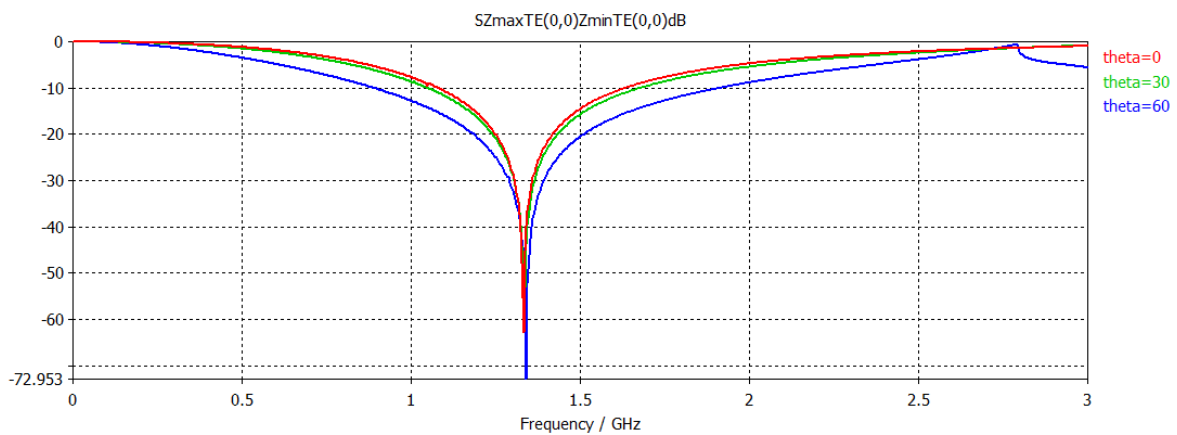


Figura 5.4 - Coeficiente de transmissão ( $\tau$ ) para os valores do ângulo de incidência  $\theta$  seguintes:  $0^\circ$ ;  $30^\circ$ ;  $60^\circ$

Observando-se a resposta freqüencial do coeficiente de transmissão ( $\tau$ ) em relação à variação do ângulo de incidência  $\theta$ , verificamos que a freqüência de ressonância apresenta pouca variação. Esta característica dos tripolos é realmente desejada nas FSS. De fato, como ele preenche bem os espaços devido à imbricação, ele sofre menos influência da variação de  $\theta$ .

### 5.1.2 Anéis

Em seguida, apresentamos um breve estudo sobre os anéis, uma estrutura do tipo caminho fechado (Grupo 2). Para este tipo de estrutura, o comprimento de onda de ressonância deve ser igual ao perímetro do laço (sob condições ideais). Para se saber a influência de um dielétrico (Figura 5.5) nas FSS, introduziu-se uma camada de FR-4 (sem

perdas) a um anel composto de Condutor Elétrico Perfeito (PEC - *Perfect Electric Conductor*) (Figura 5.6).

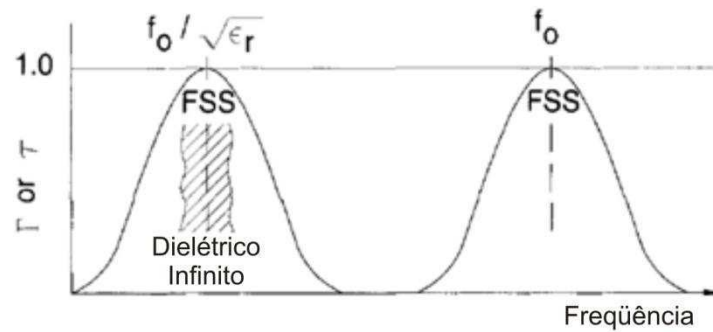


Figura 5.5 - Influência de um dielétrico na frequência de ressonância. Se uma estrutura FSS é imersa em um meio infinito de  $\epsilon_r$  constante, a frequência de ressonância será deslocada de aproximadamente  $f_0/\sqrt{\epsilon_r}$ . Extraído de (1).

Sob condições ideais e sem adição de um dielétrico, a frequência de ressonância esperada é:

$$f_r = \frac{c}{2\pi r} \tag{E 5.1}$$

onde  $c$  é a velocidade de uma onda eletromagnética no vácuo.

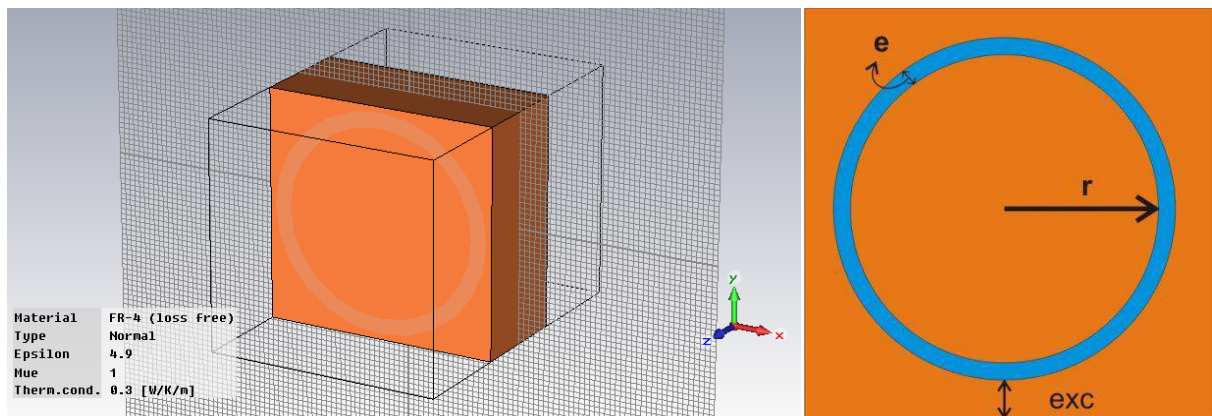


Figura 5.6 - À esquerda : Ambiente de simulação de uma estrutura FSS em anel (Grupo 2) imersa em uma camada de dielétrico FR-4 (sem perdas). À direita: variáveis relacionadas ao anel.

Para se saber o efeito de cada variável em uma estrutura do tipo anel, fez-se variar o valor da variável de interesse, fixando-se o valor das outras variáveis. As dimensões escolhidas como padrão estão apresentadas na Tabela 2. Obtemos as respostas frequenciais (Coeficiente de transmissão) para a variação de três parâmetros apresentados (exceto a espessura do anel  $e$ ) além do ângulo de incidência  $\theta$ .

r (mm)	e (mm)	exc (mm)	z (mm)
50	3	1	20

Tabela 2 - Dimensões padronizadas para a análise da influência de cada variável. A dimensão z corresponde à espessura da camada de dielétrico.

- Análise da variação do raio do anel:

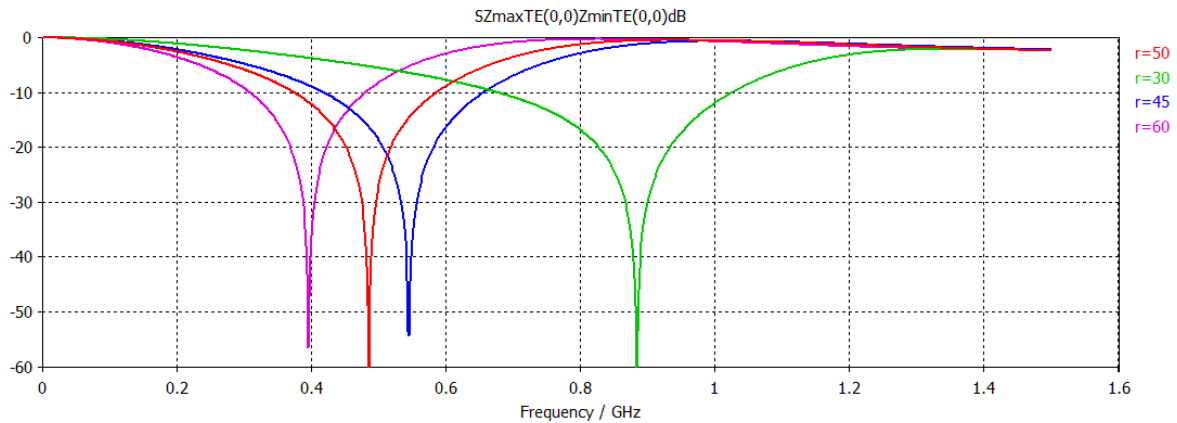


Figura 5.7 - Coeficiente de transmissão ( $\tau$ ) para os valores de raio seguintes: 30 mm ; 45 mm ; 50 mm, 60 mm.

Como esperado, aumentando-se o raio r, aumenta-se o perímetro do laço, e faz-se ressoar a uma frequência menor segundo (E 5.1).

- Análise da variação da distância entre o anel e o tamanho da distância entre o anel e o extremo da célula elementar (exc) :

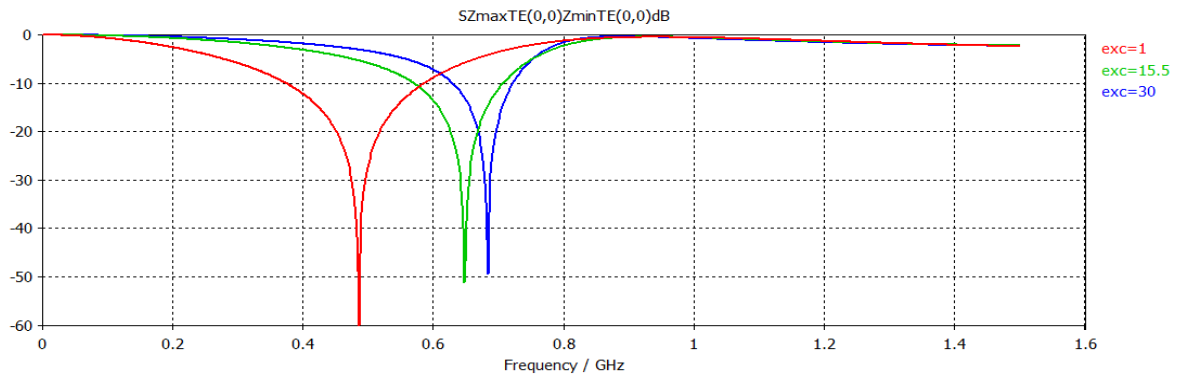


Figura 5.8 - Coeficiente de transmissão ( $\tau$ ) para os valores de gap seguintes: 1 mm; 15.5 mm, 30 mm

Se aumentarmos esta distância, a frequência de ressonância é deslocada à direita e a atenuação se torna menor.

- Análise da variação do ângulo de incidência:

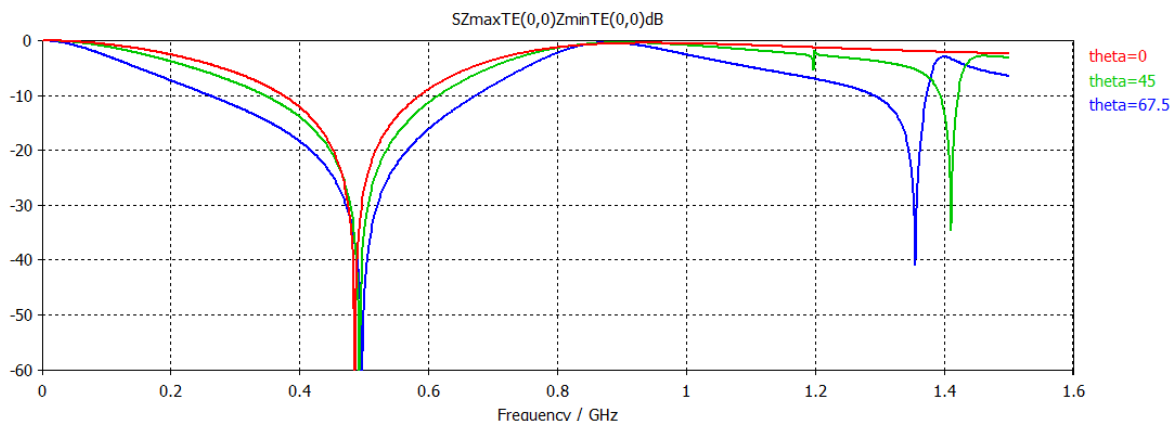


Figura 5.9 - Coeficiente de transmissão ( $\tau$ ) para os valores do ângulo de incidência  $\theta$  seguintes:  $0^\circ$ ;  $45^\circ$ ;  $67.5^\circ$

Verificamos que a frequência de ressonância não apresenta grande variação.

- Análise da variação da espessura do dielétrico:

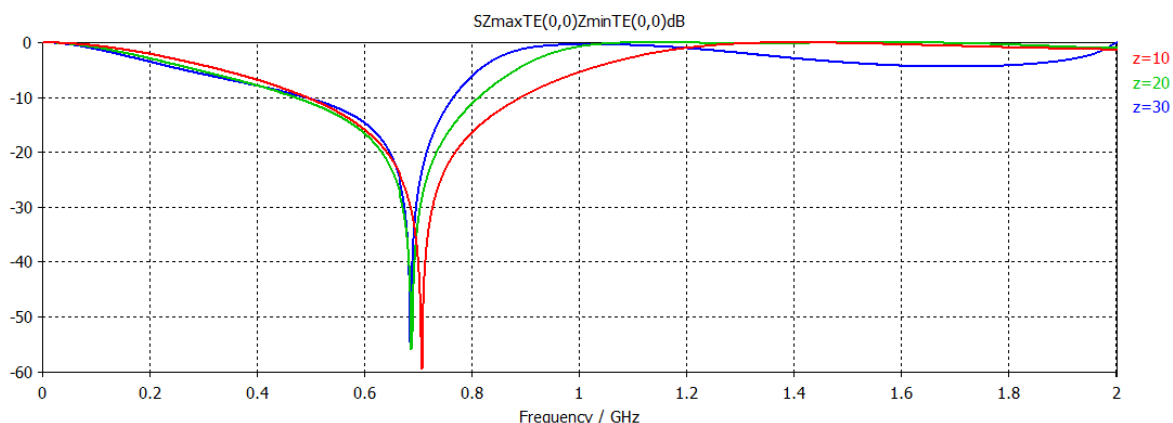


Figura 5.10 - Coeficiente de transmissão ( $\tau$ ) para os valores da espessura do dielétrico seguintes: 10 mm; 20 mm; 30 mm

### 5.1.3 Padrões gerados pelos algoritmos genéticos

Neste momento, apresentamos os padrões gerados pelos algoritmos genéticos. Eles são gerados ao acaso em uma malha, geralmente quadrada de ordem  $N$ , onde os cromossomos são normalmente codificados como um vetor de bits 0 ou 1, que representam a ausência ou presença de metal (13)(14)(15)(16). Para se desenhar os padrões estudados anteriormente, um processo natural é desenhar uma boa parte da estrutura completa, extrair a célula elementar, de forma que esta, ao ser replicada, reproduza fielmente a estrutura completa.

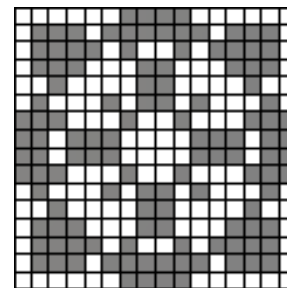


Figura 5.11 - Célula Básica de um padrão gerado por um algoritmo genético comum com matriz de ordem  $N = 16$ .

Agora o processo será realizado de forma inversa. A célula básica é criada por um *software*, e depois é gerada a estrutura inteira. De fato, se observarmos uma célula básica como a apresentada na Figura 5.11, não temos

imediatamente uma idéia sobre o desenho da estrutura completa. Na Figura 5.12, apresentamos três diferentes estruturas FSS. As figuras são compostas pela justaposição de nove células básicas e são desenhadas em CST (podemos verificar que a primeira estrutura corresponde à da célula básica apresentada na Figura 5.11).

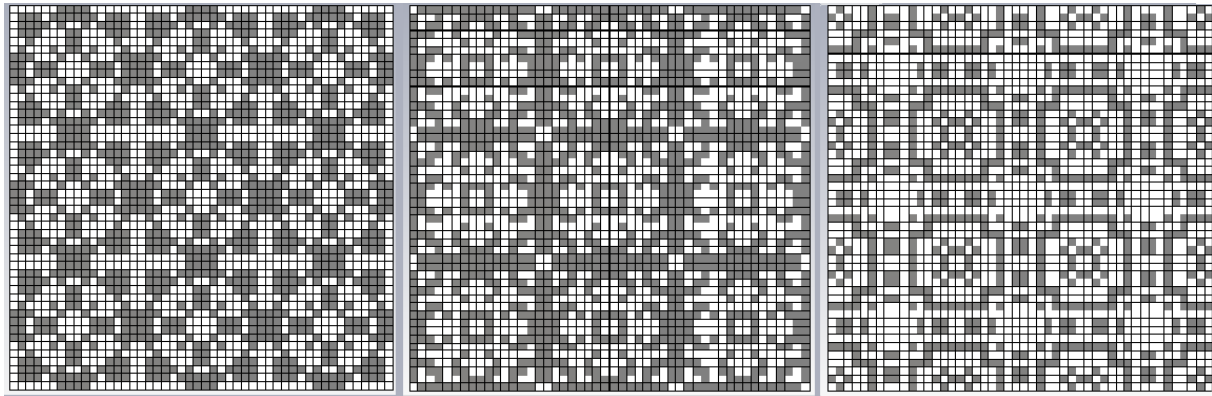


Figura 5.12 - Algumas estruturas FSS geradas pelo algoritmo genético construído.

Através da ferramenta de construção de padrões do algoritmo genético implementado, é possível desenhar padrões dos quatro grupos de elementos (Figura 5.12 e Figura 5.13).

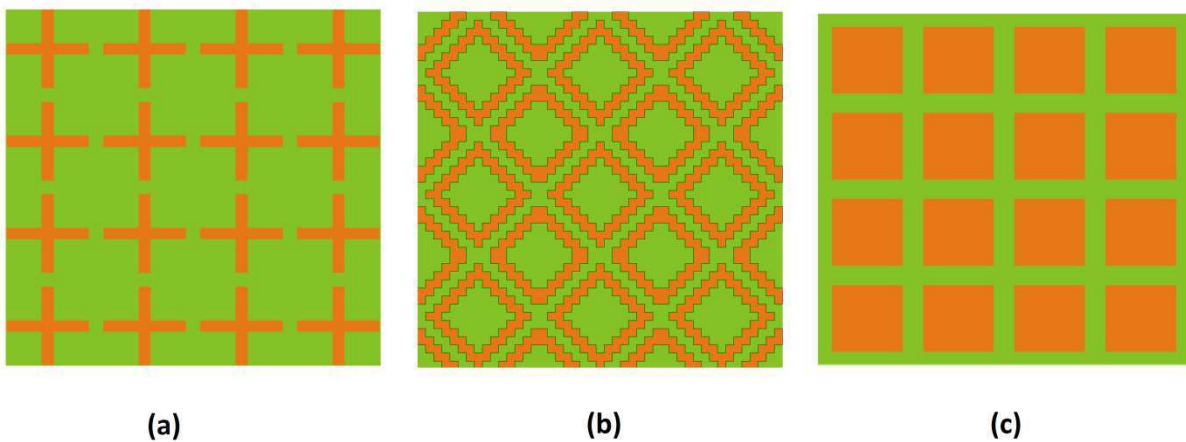


Figura 5.13 - Estruturas FSS: (a) Tipo N-Pólo (Grupo 1); (b) Tipo caminho fechado (Grupo 2); (c) Tipo Placa (Grupo 3);

## 5.2 Algoritmo construído e seu método

Durante os primeiros estudos sobre os algoritmos genéticos para resolver problemas de FSS, constatamos que quase todos os autores trabalharam utilizando uma malha quadrada para se desenhar os padrões (6) (7) (8) (9) (10). Uma malha é uma discretização espacial de um meio contínuo. Este tipo de malha foi utilizado para se desenhar a estrutura mostrada na Seção 5.1.3.

### 5.2.1 Estudos sobre uma malha triangular

Em um primeiro momento, refletimos sobre a utilização de uma malha triangular (contrariamente a diversas publicações que utilizam a malha quadrada), visto a possibilidade que a mesma oferece de se poder construir qualquer padrão gerado por uma malha quadrada, além de estruturas com a presença de ângulos, como os tripolos, triângulos, hexágonos, etc. (Figura 5.14). Podemos assim dizer que este tipo de malha apresenta uma melhor resolução.

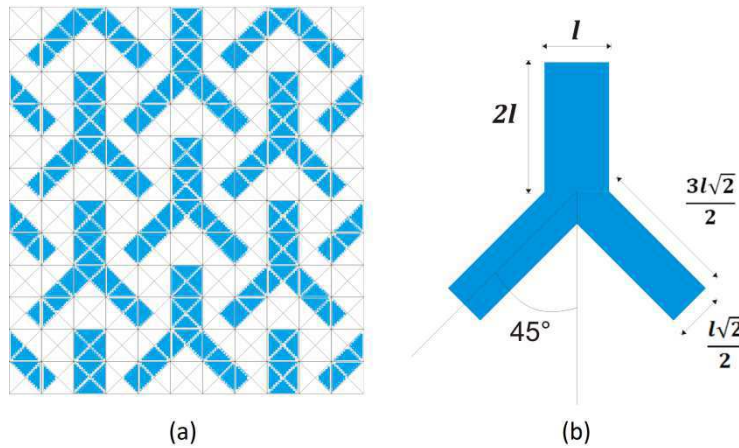


Figura 5.14 - (a) Representação de uma FSS composta por tripolos utilizando malha triangular; (b) Dimensões de um tripolo simples gerado por este tipo de malha.

Em seguida, desenhamos em CST um modelo correspondente ao tripolo apresentado na Figura 5.14 (onde o comprimento  $l$  corresponde ao tamanho da base de um pequeno triângulo que forma a malha) a fim de caracterizar esta estrutura (frequência de ressonância, estabilidade com o ângulo de incidência) em relação ao tripolo regular apresentado na Seção 5.1.1.

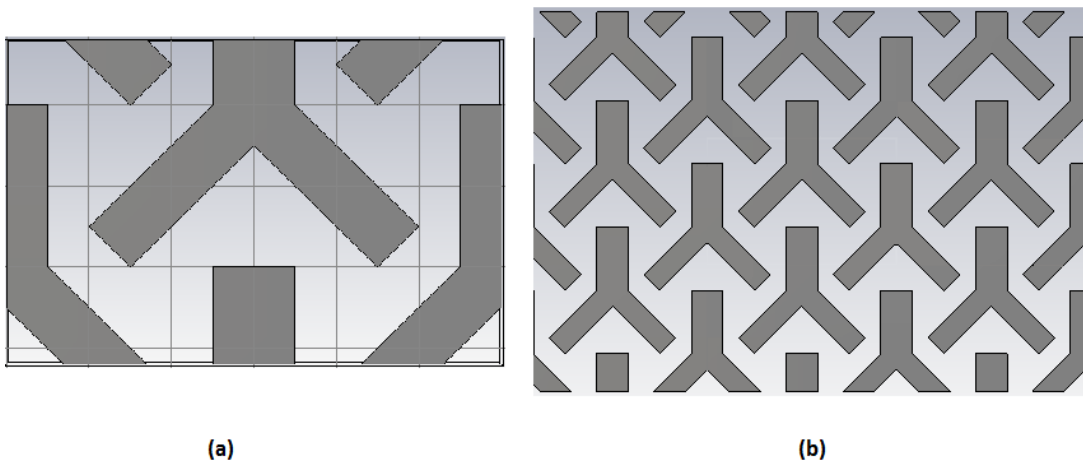


Figura 5.15 - (a) Célula básica para o tripolo de malha quadrada simulado em CST; (b) Idéia da superfície FSS obtida para este tipo de tripolo.



Como parâmetros (dimensões explicitadas na Figura 5.14b), escolhemos como valor padrão  $l = 20$  mm, e o fizemos variar para os seguintes valores [20 mm ; 22,5 mm ; 25 mm], Lançamos simulações para saber a influência das dimensões da malha no nosso padrão. Em seguida, realizamos o mesmo processo, fixando o parâmetro  $l = 20$  mm e fazendo variar os ângulos de incidência  $\theta = [0^\circ ; 30^\circ, 60^\circ]$ . Para a composição do material, escolhemos PEC, e para o substrato, vácuo. A banda de frequência a ser estudada foi entre 1.6GHz e 2.6GHz.

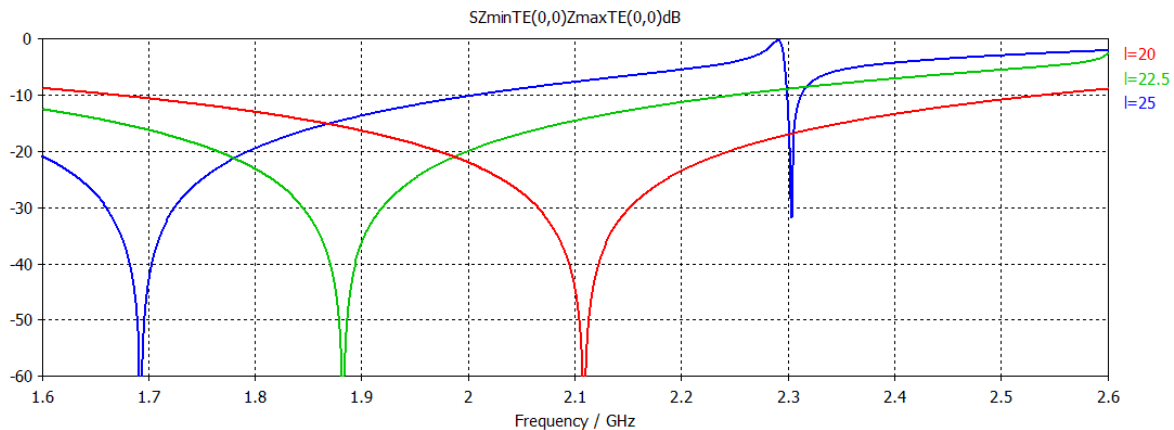


Figura 5.16 - Coeficiente de transmissão ( $\tau$ ) para os valores de comprimento de do parâmetro  $l$  para 20 mm ; 22,5mm ; 25 mm.

Neste caso, desejamos apenas observar o comportamento deste tipo de estrutura. Observando os resultados da simulação em CST (Figura 5.16), verificamos que para os três comprimentos diferentes de  $l$ , têm-se picos bem definidos nas suas respectivas frequências de ressonância (exceto uma em torno de 2.3GHz para  $l = 25$  mm). Após observar este primeiro resultado, criamos a Tabela 3. para se calcular os comprimentos de onda de ressonância correspondentes ( $\lambda_r$ ), utilizou-se a expressão:

$$\lambda_r = \frac{c}{f_r} \tag{E 5.2}$$

Comprimento $l$ (mm)	Frequência de ressonância (GHz)	Comprimento de onda de ressonância ( $\lambda_r$ )	Relação $\frac{\lambda_r}{l}$
20.0	2.108	0.1423	7.115
22.5	1.882	0.1594	7.108
25.0	1.6925	0.1772	7.088

Tabela 3- Estudo da frequência de ressonância para o tripolo gerado em malha triangular em função da variação do parâmetro  $l$ .

De posse destes dados, e sabendo que todas as dimensões deste tripolo estão parametrizadas em função de  $l$ , foi possível encontrar uma relação entre este parâmetro e o comprimento de onda  $\lambda_r$ . Assim, podemos afirmar que para este tipo de tripolo, a frequência de ressonância é aproximadamente igual a sete vezes o comprimento de da base do triângulo que forma a malha, ou seja:

$$\lambda_r \approx 7.l \tag{E 5.3}$$

Em seguida, verificamos a influência do ângulo de incidência sobre esta estrutura fazendo variar o parâmetro  $\theta$  (Figura 5.17).

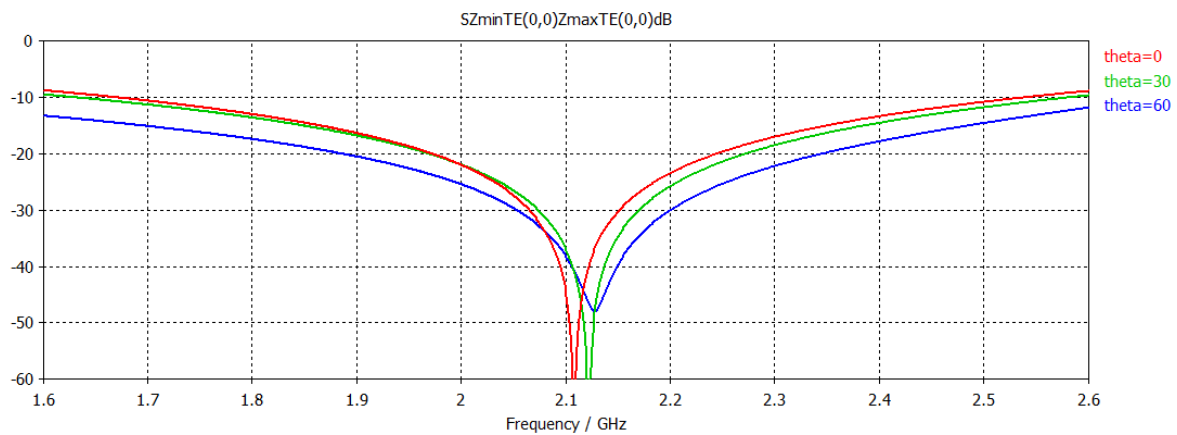


Figura 5.17 - Coeficiente de transmissão ( $\tau$ ) para os valores do ângulo de incidência ( $\theta$ ) para  $0^\circ$  ;  $30^\circ$  ;  $60^\circ$ .

Ângulo de incidência $\theta$ ( $^\circ$ )	Frequência de ressonância (GHz)
0	2.122
30	2.1087
60	2.1283

Tabela 4 - Estudo da frequência de ressonância para o tripolo gerado em malha triangular em função do ângulo de incidência  $\theta$ .

Observando os valores obtidos, verificamos que a frequência de ressonância não varia tão acentuadamente com a variação do ângulo de incidência, e por isso, constatamos que este tipo de tripolo funciona bem e é um padrão aceitável. A utilização de uma malha triangular para se resolver problemas de FSS é aceitável do ponto de vista eletromagnético.

### 5.2.2 Malha quadrada e simetria $n$ -fold

Por comodidade, e como apresentado em várias publicações sobre AGs que utilizam malha quadrada, começamos a construir nosso algoritmo para este tipo de malha durante a codificação dos cromossomos. De fato, a utilização de malha quadrada já demanda um tempo considerável, o que nos fez desistir da idéia da utilização da malha triangular visto que esta necessita de quatro vezes o número de bits da malha quadrada.

Mostraremos um estudo de caso para uma estrutura de malha quadrada de ordem  $N = 8$ . Esta estrutura apresenta, então,  $8 \times 8 = 64$  bits (genes), o que corresponde a  $2^{64}$  soluções diferentes, o que é impraticável de se analisar. Utilizamos então algumas técnicas para se reduzir o nosso espaço de busca, começando por escolher um tipo de simetria. (Figura 5.18).

Neste caso, para uma simetria  $8$ -fold (Figura 5.18c), codificamos apenas 10 bits (selecionados em vermelho) invés dos 64 da estrutura inteira. Neste caso, o cromossomo é composto por uma cadeia de 10 bits apenas, distribuídos de 1 a 10 conforme enumerado na

Figura 5.18c, os bits 1 e 0 representam a presença e ausência de metal respectivamente. Para se criar a estrutura inteira, rebatemos os 10 bits selecionados em vermelho (como em um espelho) como mostra o passo 1, formando um quadrado. Em seguida, rebate-se este quadrado seguindo o passo 2 formando um retângulo. Por fim, rebatemos este retângulo conforme o passo 3 para concluir a estrutura. Assim, representamos uma célula básica completa codificando apenas 10 bits, e o espaço de busca se reduzirá a  $2^{10}$  soluções. Outra grande vantagem da simetria *8-fold* é apresentar simetria nos eixos *x* e *y*.

Podemos reduzir ainda mais este espaço de busca, sabendo que em condições ideais, o coeficiente de reflexão de uma estrutura é igual ao coeficiente de transmissão de sua estrutura complementar (segundo o Princípio do Complemento de Babinet (1)), assim, necessitamos analisar somente a metade deste espaço. Então, o espaço de busca é reduzido a  $2^9$  soluções.

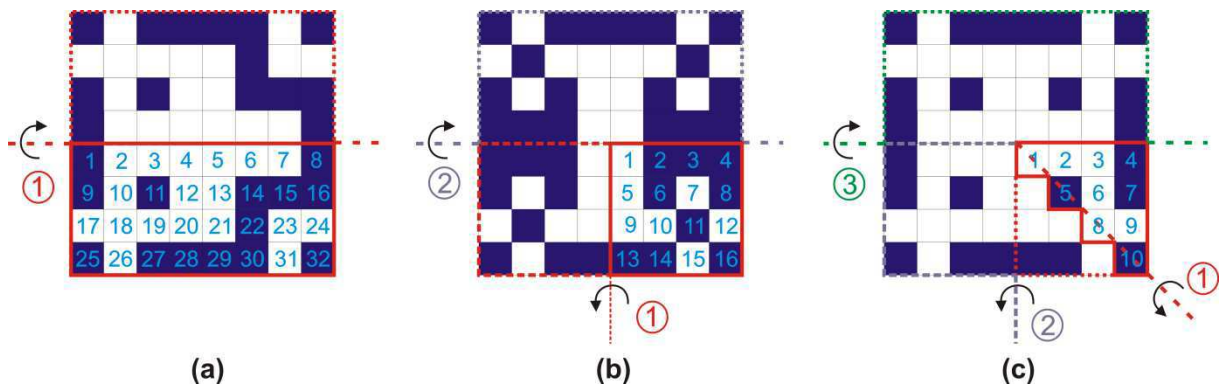


Figura 5.18 - Construção de uma célula elementar e codificação utilização simetria: (a) 2-fold ; (b) 4-fold ; (c) 8-fold ; Os quadrados em azul correspondem à presença de metal (bit 1).

Para se calcular o número de total de genes relacionados ao desenho da estrutura, utiliza-se a fórmula do número triangular:

$$T = \frac{(n)(n + 1)}{2} \quad (E 5.4)$$

Onde *n* é inteiro e igual a *N*/2. Por exemplo, para *N*=8, *n* = 4 então *T* = 10, como mostrado na Figura 5.18c.

### 5.3 Estrutura geral do Algoritmo Genético construído

Antes de se buscar o objetivo final, que corresponde a um filtro que bloqueia as frequências a 900 MHz e 1.8 GHz, insensível a variação do ângulo de incidência  $\theta$ , realizamos estudo para três outros filtros mais simples a título de aprendizado e para obtenção dos parâmetros do AG. Os filtros selecionados inicialmente foram:

- Filtro bloqueia faixa a 900 MHz.
- Filtro bloqueia faixa a 900 MHz insensível à variação do ângulo de incidência  $\theta$ .
- Filtro bloqueia faixas a 900 MHz e 1.8 GHz

Para começar a realização do AG, foi necessário encontrar parâmetros adequados. Para se selecionar os pais, este algoritmo utilizou seleção por torneio com  $k = 2$  elementos, crossover a dois pontos com probabilidade  $p_{\text{cross}} = 0.6$  em conjunto com taxa de mutação  $p_{\text{mut}} = 0.01$  e simetria *8-fold*. Tais parâmetros do AG, normalmente são levantados de forma empírica. Para todas as aplicações, escolhendo uma matriz de ordem  $N$  ímpar, por permitir desenho de estruturas como as cruzes afiladas e dipolos centrados na sua célula, diferentemente de uma estrutura para um  $N$  par (Figura 5.19). Ela permite também construir estruturas do tipo caminho fechado como os quadrados.

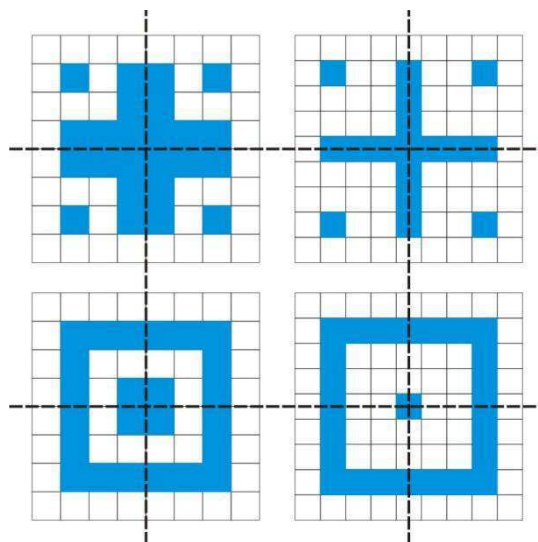


Figura 5.19 - Diferença entre uma estrutura  $N$  par (à esquerda) e ímpar (à direita). A linha tracejada representa os eixos de simetria

Escolhemos primeiramente um tamanho de população mínima igual a 20, pois funciona bem para a maioria dos problemas práticos e permite um desempenho computacional mais rápido (9).

### 5.3.1 Função de Avaliação

Como já explicado na seção 3.3.3, a função de avaliação representa o maior desafio em um Algoritmo Genético. Para as aplicações desejadas, utilizamos uma função do Matlab chamada *findpeaks*, que encontra os valores máximos (picos) de uma curva e retornam dois vetores, o primeiro com os valores das abscissas e outro com os valores das ordenadas correspondentes.

De fato, para a curva dos coeficientes de transmissão que estudamos, os picos são representados por valores mínimos, e por isso, devemos enviar como parâmetro à função

*findpeaks* o inverso da curva, por exemplo, se Y corresponde a curva do coeficiente de transmissão, chamamos a função da seguinte forma: *findpeaks* (1./Y).

Se existem picos de atenuação inferior a um valor pré-determinado na curva do coeficiente de transmissão (-30dB para o modo TE, por exemplo), verificamos se o pico se localiza em uma faixa em torno da freqüência desejada ( $f_o$ ) (a largura desta faixa é dada por uma porcentagem de tolerância ( $\rho bf$ ), que pode ser ajustada no algoritmo). Em caso positivo (Figura 5.20) a freqüência correspondente a esse pico será atribuída à variável *picMin*, e ao valor de *fitness* (*fit*) será atribuído utilizando um desvio médio como mostrado abaixo:

$$fit = 1 - \frac{|f_o - picMin|}{f_o} \quad (E 5.5)$$

Para o filtro a duas faixas, insensível à variação do ângulo de incidência  $\theta$ , verificamos a existência de bons picos em torno das duas freqüências de interesse, e calculamos o valor de *fitness* utilizando a mesma fórmula para os dois picos e realizando uma soma ponderada dos valores das duas freqüências tanto para o modo TE quanto TM, avaliados para ângulo de incidência igual a  $0^\circ$ . Caso existam bons picos para ambas as freqüências, ou seja, os picos estão localizados dentro das áreas mostradas em azul na Figura 5.20, esta solução estará apta a ser simulada para  $\theta = 30^\circ$ , calculando-se novamente um novo valor de *fitness* ponderado, que será somado ao valor anterior. Também foi atribuída uma pequena bonificação ao *fitness* de estruturas que levam menos metal na sua composição por representar menor custo. Um fluxograma deste algoritmo é mostrado na Figura 5.21.

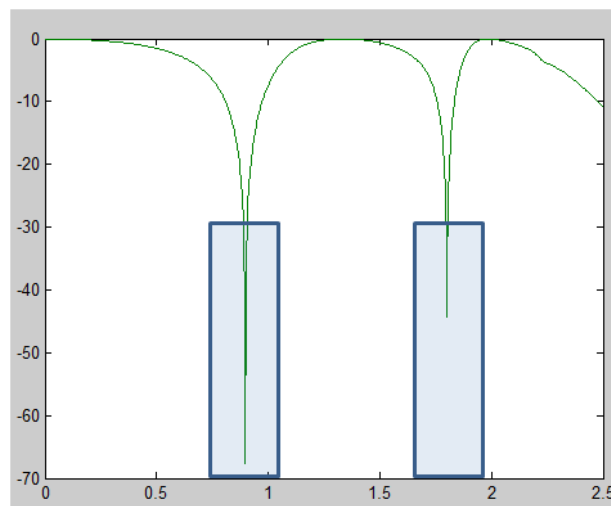


Figura 5.20 - Verificação da presença de bons picos na curva do coeficiente de transmissão. O pico é bom se está localizado na área em azul (em torno da freqüência de interesse e atenuação inferior a -30dB).

Vale lembrar que a dinâmica do AG implementado se desenrola no Matlab. Ele gera os indivíduos aleatoriamente, realiza as operações genéticas. Para se obter o *fitness* de cada indivíduo, utiliza-se rotinas com os comandos do CST para o desenho da célula básica do indivíduo dada sua cadeia de bits (cromossomo) e realizar os rebatimentos (comando

*Mirror*) similar ao procedimento apresentado na Figura 5.18c. Uma vez realizada esta interface, a curva do coeficiente de transmissão é calculada e devolvida ao Matlab para fazer o tratamento dos dados, inclusive, calcular o valor de *fitness*.

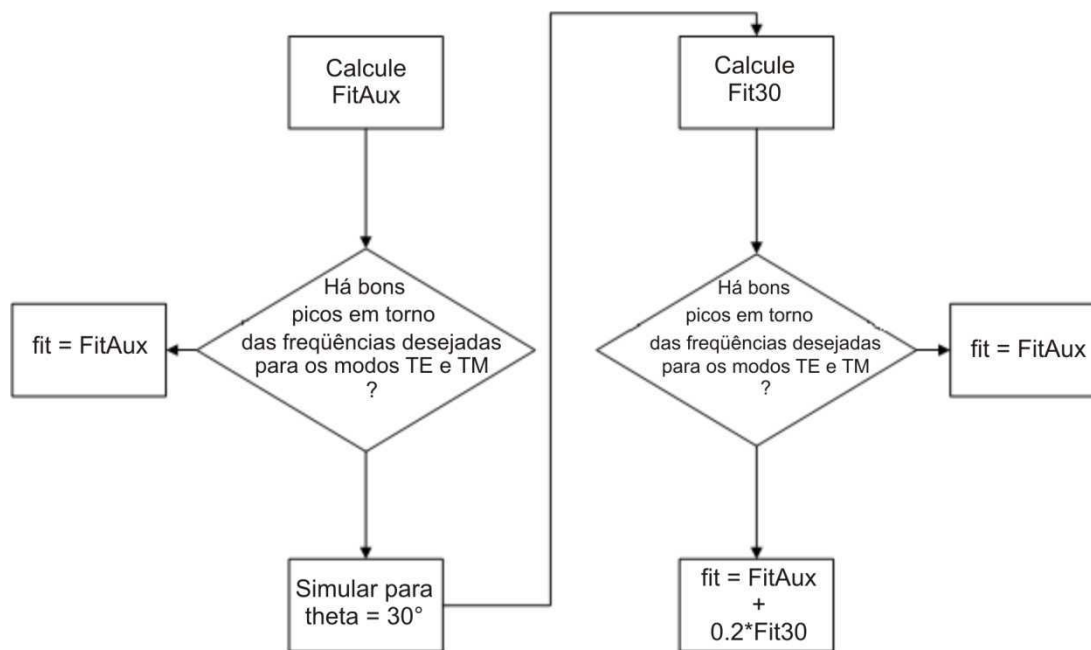


Figura 5.21 - Esquema demonstrativo para atribuição de *fitness* no algoritmo construído

Para cada um dos quatro modelos de filtro, escolheu-se uma dimensão para o menor elemento (pixel), que corresponde a um bit, de forma empírica. Na verdade, através da expressão  $c = \lambda.f$ , pode-se ter uma idéia da ordem de grandeza do comprimento de onda. Tomamos um padrão simples (loop quadrado Figura 5.22) e calculamos a dimensão do pixel necessário para fazer ressoar uma onda de frequência desejada (a de menor ordem para o caso de duas frequências).

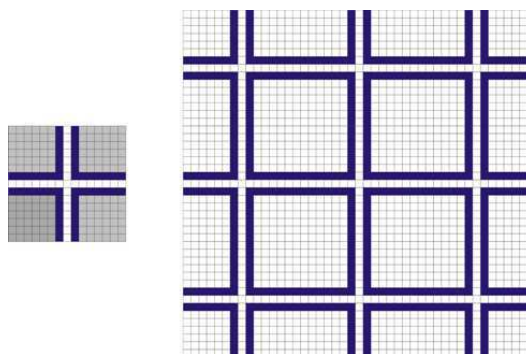


Figura 5.22 - Padrão de testes e Estrutura FSS para obtenção do comprimento de pixel.

O fato de se testar a menor das duas frequências de interesse é porque acredita-se que possa ser gerado alguma subestrutura nas regiões em cinza capazes de fazer ressoar

freqüências de ordem maior. Logicamente, freqüência é inversamente proporcional ao comprimento de onda.

## 5.4 Conclusão

Esta seção abordou a metodologia empregada na construção do algoritmo genético para tentar encontrar padrões capazes de filtrar os sinais desejados. Foi visto que a implementação de um AG é simples, mas se deve dar especial atenção aos parâmetros do AG para evitar lançar longas simulações sem perspectivas de sucesso. Apresentou-se técnicas importantes, como o uso de simetria, para reduzir consideravelmente o espaço de busca e tornar o algoritmo mais eficiente.

Foi dado também uma maior ênfase ao *software* CST, e como ele foi utilizado para gerar em conjunto com o Matlab, a principal preocupação do AG, ou seja, a função de avaliação.

## 6 Resultados

Inicialmente apresentamos os resultados para os três filtros mais simples, onde se utilizou uma população reduzida, condição de parada de apenas 15 gerações, e faixa de frequência de simulação menor, o que representa a simulação de apenas 320 estruturas, lançadas sobre um processador Intel®Core 2 Duo. Apresentamos os parâmetros utilizados, a célula básica, estrutura FSS periódica, e coeficiente de transmissão para cada filtro.

### 6.1.1 Filtro Rejeita-Faixa GSM (900 MHz)

Para esta primeira aplicação, escolhemos os seguintes parâmetros:

Tamanho da População (npop)	20
Ordem da matriz (N)	15
Condição de parada	Número de gerações = 15
Dimensão do pequeno quadrado (bit) (lp)	12.47 mm
Período da célula elementar	$T_x = T_y = 187,05$ mm

Resultados:

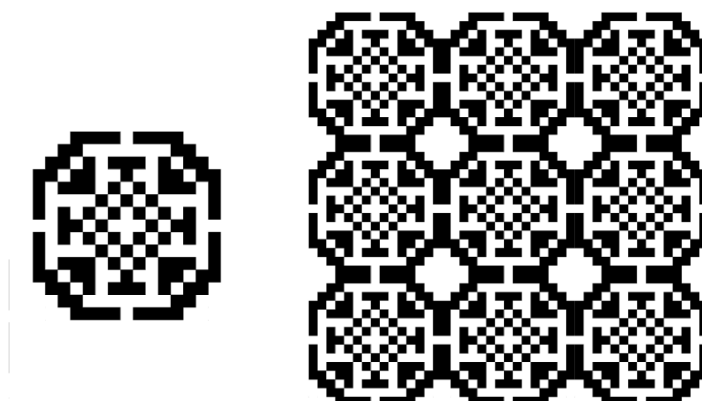


Figura 6.1 - Célula elementar e estrutura FSS correspondente para um filtro rejeita-faixa GSM (900MHz).

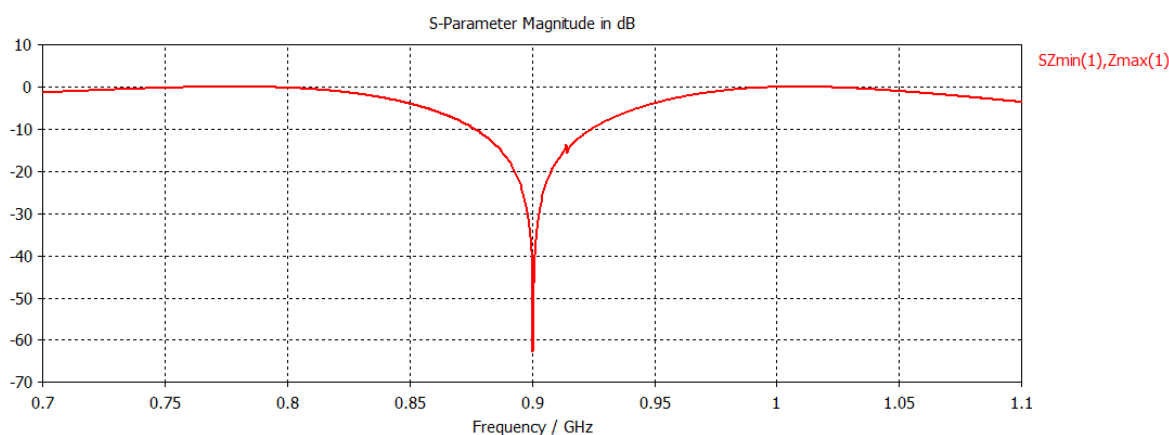


Figura 6.2 - Coeficiente de Transmissão para um filtro rejeita-faixa GSM (900MHz).



### 6.1.2 Filtro rejeita-faixas GSM (900 MHz) e (1,8 GHz)

Parâmetros escolhidos:

Tamanho da População (n <sub>pop</sub> )	20
Ordem da matriz (N)	15
Condição de parada	Número de gerações = 15
Dimensão do pixel (1 bit) (l <sub>p</sub> )	10.35 mm
Período da célula elementar	T <sub>x</sub> = T <sub>y</sub> = 115.25 mm

Resultados:

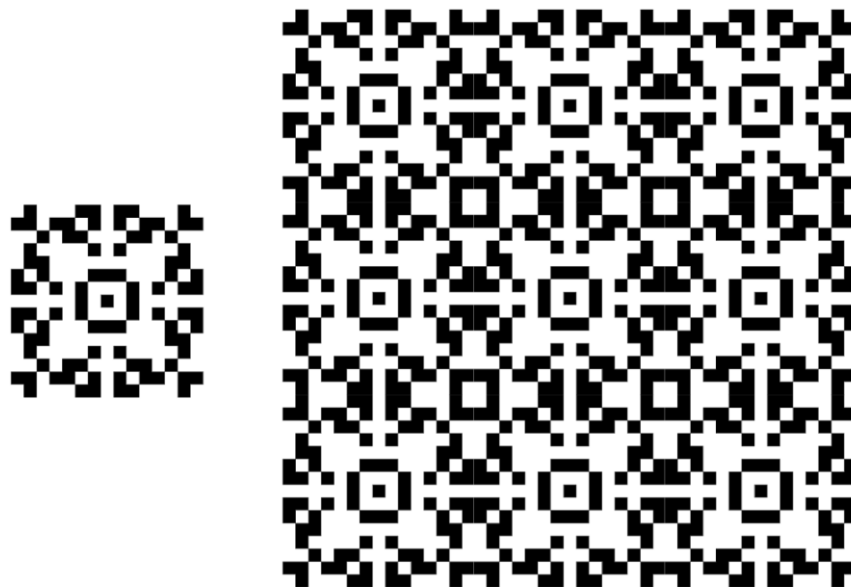


Figura 6.3 - Célula elementar e estrutura FSS correspondente para um filtro rejeita-faixas GSM (900MHz) e (1.8GHz).

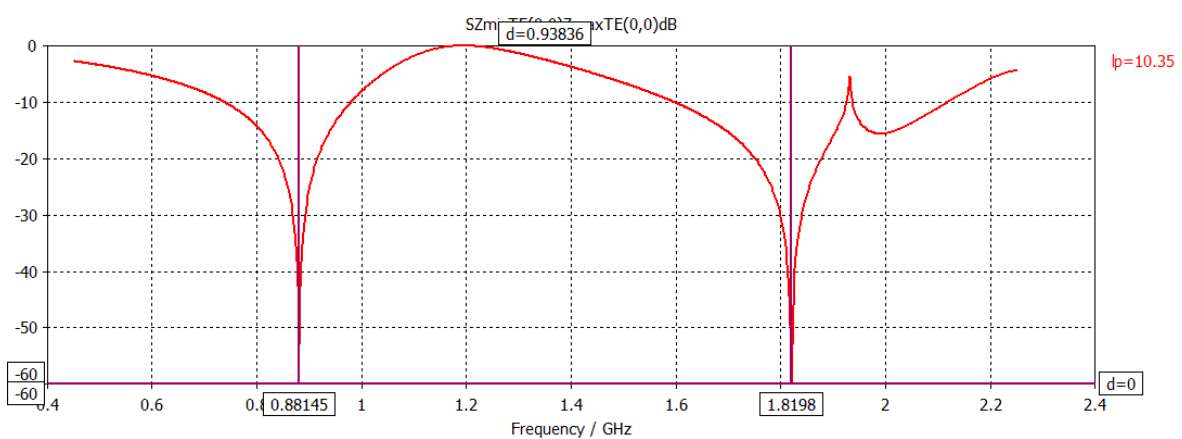


Figura 6.4 - Coeficiente de Transmissão para um filtro rejeita-faixas GSM (900MHz) e (1.8GHz)

### 6.1.3 Filtro Rejeita-Faixa GSM (900 MHz) insensível ao ângulo de incidência $\theta$

Para esta aplicação, os parâmetros escolhidos foram:

Tamanho da População (npop)	20
Ordem da matriz (N)	15
Condição de parada	Número de gerações = 15
Dimensão do pixel (1 bit) (lp)	12.47 mm
Período da célula elementar	$T_x = T_y = 187.25$ mm

Resultados:



Figura 6.5 - Célula elementar e estrutura FSS correspondente para um filtro rejeita-faixa GSM (900MHz) pouco sensível ao ângulo de incidência  $\theta$ .

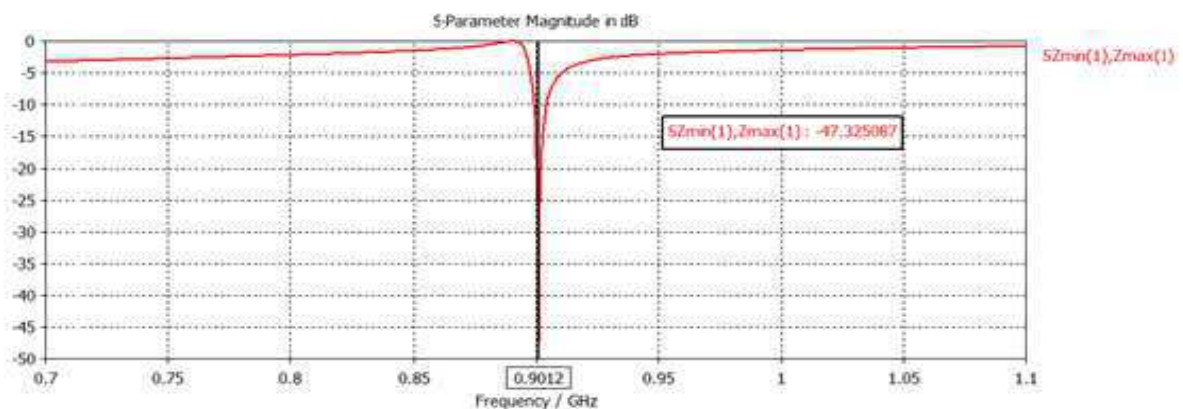


Figura 6.6 - Coeficiente de Transmissão ( $\tau$ ) para um filtro rejeita-faixa GSM (900MHz) para ângulo de incidência  $\theta = 0^\circ$

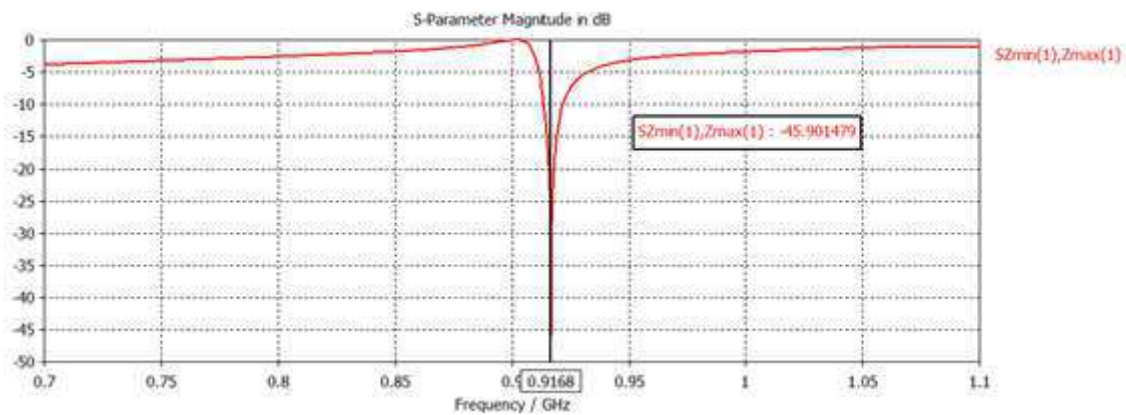


Figura 6.7 - Coeficiente de Transmissão ( $\tau$ ) para um filtro rejeita-faixa GSM (900MHz) para ângulo de incidência  $\theta = 30^\circ$

### 6.1.4 Filtro rejeita-faixas GSM (900 MHz) e (1,8 GHz) pouco sensível à variação do ângulo de incidência $\theta$ .

Para este filtro, aumentou-se a ordem da matriz N de 15 para 17, com isso se aumentou o número de bits do cromossomo de 36 para 45, o que representou um aumento no número da população em  $2^9$  vezes, por isso, exigiu-se um aumento na população inicial e número de gerações, para se obter soluções aceitáveis. As simulações foram lançadas sobre processador Intel Core i7-2600 @ 3.40 GHz.

Tamanho da População (npop)	30
Ordem da matriz (N)	17
Condição de parada	Número de gerações = 25
Dimensão do pixel (1bit) (lp)	8 mm
Período da célula elementar	$T_x = T_y = 136$ mm

#### Resultados:

Depois de findada todas as simulações, se encontrou um resultado bastante promissor. Sua célula básica, e por conseqüência sua estrutura periódica apresenta bem menos áreas de metal que vácuo (Figura 6.8), o que é interessante para se reduzir custos.

Analisando-se a curva do coeficiente de transmissão tanto para  $\theta = 0^\circ$  quanto  $\theta = 30^\circ$ , verificamos a presença de picos bem definidos bastante próximos das freqüências de interesse, e o que é importante: atenuações inferiores a -30dB para 0.9 GHz e 1.8 GHz (Figura 6.9 e Figura 6.10) e quase nenhuma diferença entre as duas curvas, ou seja, a estrutura apresenta excelente imunidade à variação de  $\theta$ .

Uma vez escolhida a melhor solução dentre todas, podemos então lançar simulações com maior precisão ativando o refinamento da malha e aumentando sua resolução. Agora foi analisado o comportamento da solução de 0 GHz a 5GHz (Figura 6.11).

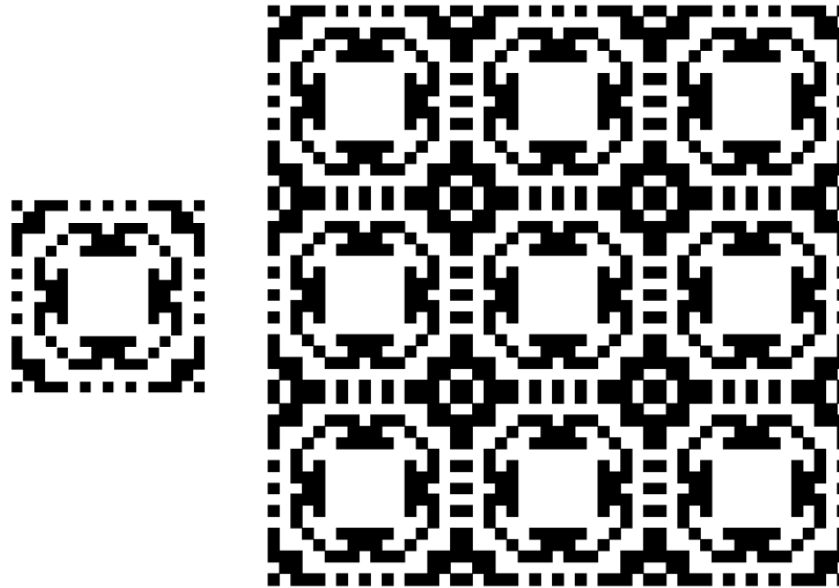


Figura 6.8 - Célula elementar e estrutura FSS correspondente para um filtro rejeita-faixas GSM (900MHz e 1.8GHz) com pouca sensibilidade à variação do ângulo de incidência  $\theta$ .

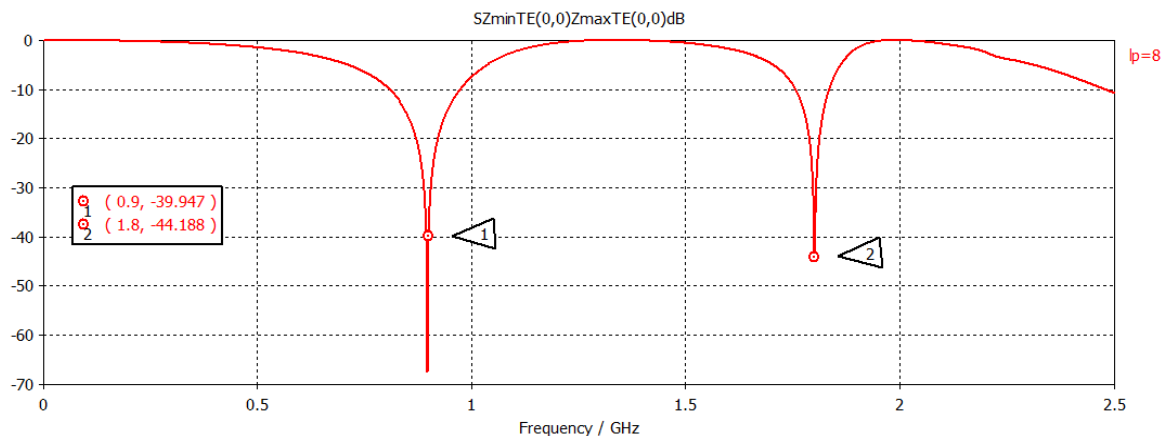


Figura 6.9 - Coeficiente de Transmissão ( $\tau$ ) para um filtro rejeita-faixas GSM (900MHz e 1.8GHz) para  $\theta = 0^\circ$

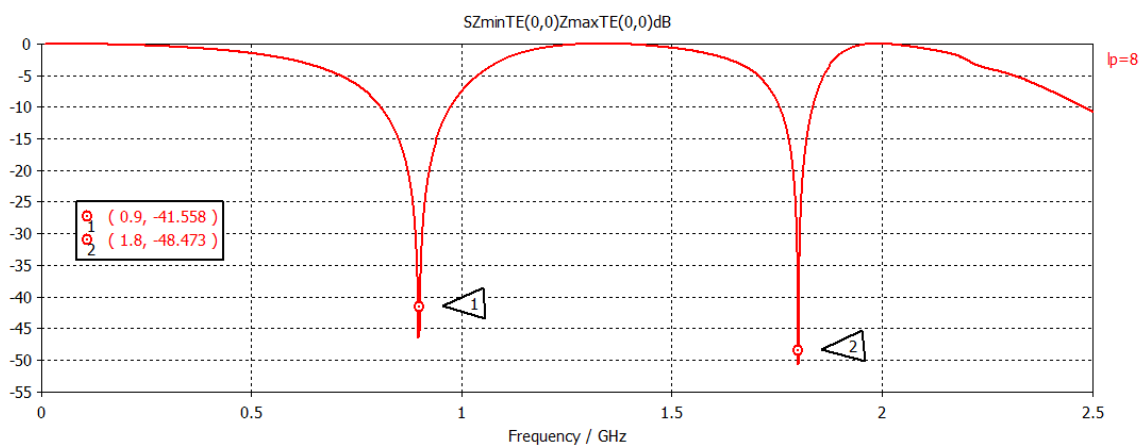


Figura 6.10 - Coeficiente de Transmissão ( $\tau$ ) para um filtro rejeita-faixas GSM (900MHz e 1.8GHz) para  $\theta = 30^\circ$

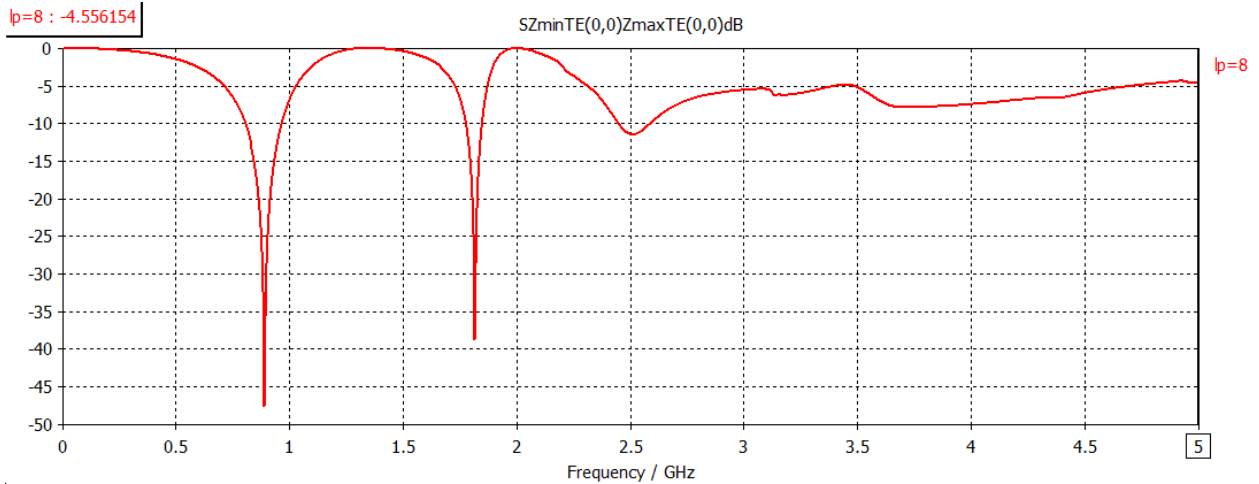


Figura 6.11 - Coeficiente de Transmissão ( $\tau$ ) para uma faixa de frequência de 0 a 5 GHz

Para essa nova análise verificou-se que o filtro possui uma pequena atenuação (aproximadamente -12dB) em torno de 2.5 GHz, o que não chega a ser considerável, e o filtro continua atendendo às suas especificações.

Mostraremos em seguida a título de ilustração um vista das magnitudes de campo elétrico tanto para a frequência de máxima atenuação quanto para uma de pouca atenuação.

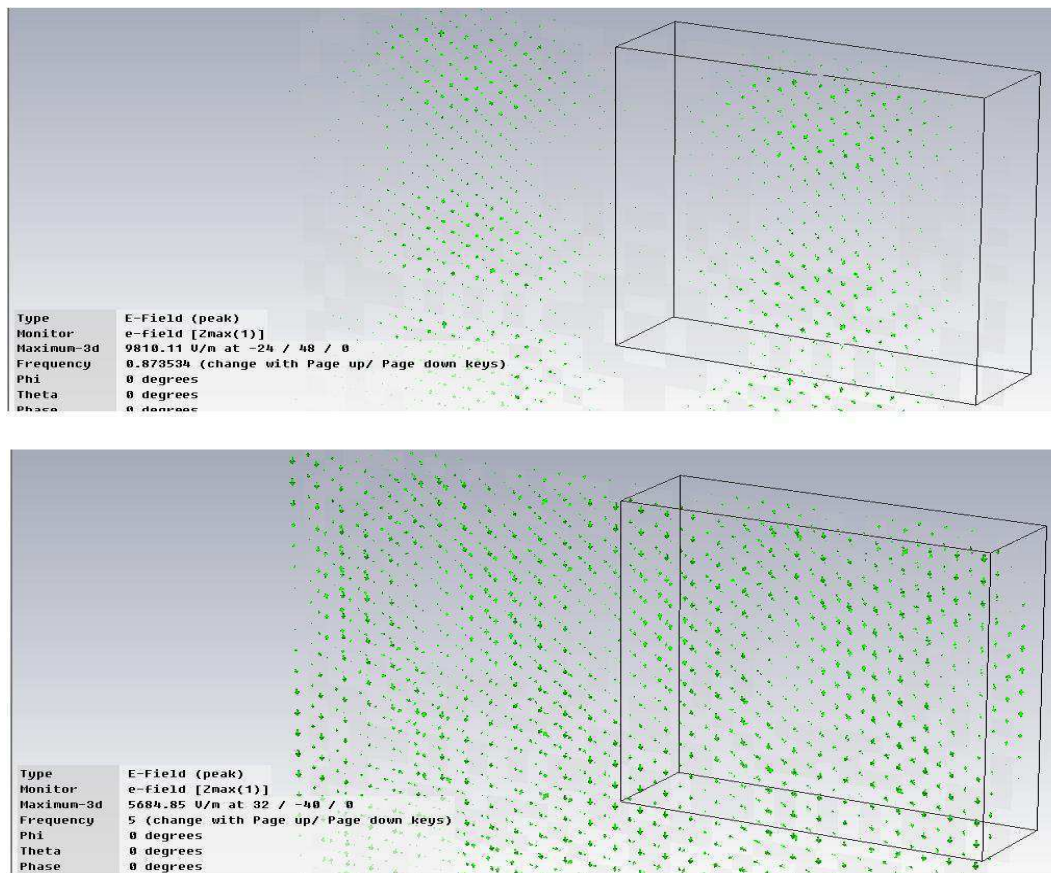


Figura 6.12 - Vista das magnitudes de campo elétrico à 0.9 GHz (acima) e 5 GHz (abaixo)

Em problemas de AG, é interessante se verificar a evolução dos indivíduos, ou seja, se está ocorrendo um "melhoramento genético" com o passar das iterações ou se a mesma está estagnada. Esta curva vai sendo incrementada a cada geração simulada e verifica qual o melhor indivíduo até o momento. Para o nosso algoritmo, a curva de evolução obtida está mostrada na Figura 6.13.

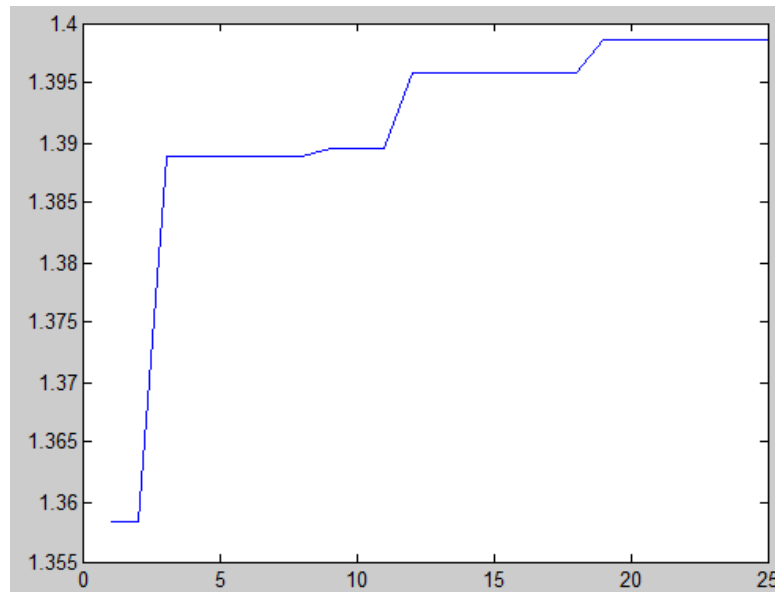


Figura 6.13 - Curva de evolução do valor de fitness para cada geração

## 7 Conclusões e Perspectivas

O tema deste trabalho aborda a utilização dos algoritmos genéticos para resolução de problemas em eletromagnetismo, em especial os filtros espaciais utilizando Superfícies Seletivas em Frequência.

Um estudo sobre as estruturas FSS foi realizado, e foram apresentadas suas propriedades, tipos, e comportamento sobre a influência de diversos fatores, como a presença de um dielétrico, variação do ângulo de incidência, tamanho da estrutura, etc., o que nos permitiu adquirir grande conhecimento no software de simulação eletromagnética CST® *Microwave Studio*.

Verificamos através do estudo sobre os Algoritmos Genéticos, que esta ferramenta se mostra bastante eficiente para o problema do tipo tentativa e erro como as FSS. Com o conhecimento adquirido, será possível resolver vários problemas fora do contexto das FSS, permitindo-se encontrar ótimas soluções analisando uma pequena amostra do espaço de busca relativamente em pouco tempo.

Os resultados obtidos primeiramente para os filtros mais simples foram satisfatórios visto que foram simulados com população reduzida, baixo número de gerações, ordem de matriz menor e lançados em um computador de baixo poder de processamento (Core 2 Duo). Estes serviram para aprimorar o algoritmo, bem como estudar a influência da cada parâmetro na simulação, tais como as dimensões de malha e probabilidades do próprio AG antes de se começar a estudar o filtro correspondente ao objetivo principal. Para este filtro rejeita-faixas a 900MHz e 1.8GHz insensível a variação do ângulo de incidência, os resultados foram considerados excelentes o que nos motiva a continuar aperfeiçoando o algoritmo para resolver problemas ainda mais complexos.

Como perspectivas para este projeto, propõe-se a utilização de AG para se obter um filtro capaz de bloquear três ou quatro sinais, incluindo-se Wi-fi (2.4 GHz e 3.6GHz), por exemplo, e que possua comportamento semelhante em qualquer tipo de parede (substrato). É interessante a criação de uma interface com o usuário para se modelar o filtro conforme suas exigências, podendo-se escolher a quantidade de sinais a ser bloqueados, parâmetros de simulação, presença e espessura de substrato.

Por fim, a realização deste estágio aliado à experiência única de um intercâmbio cultural permitiu uma ampla aquisição de conhecimentos técnicos e um enriquecimento grandioso na formação intelectual, profissional e pessoal.

## 8 Bibliografia

1. **Munk, B. A.** *Frequency Selective Surfaces: Theory and Design*. New : Wiley, 2000.
2. **Arnaut, L. R. e Blackburn, J. F.** *High-Q Frequency Selective Surfaces for Improved Spectral Efficiency*. Teddington, UK : s.n., 2004. S155.
3. **Systems, Cisco.** *20 Myths of Wi-Fi Interferences: Dispel Myths to Gain High-Performing and Reliable Wireless*. San Jose, CA : s.n., 2007.
4. **Gustafsson, Mats, et al.** Desing of frequency selective windows for improved indoor outdoor communication. *Antennas and Propagation, IEEE Transactions*. 2006.
5. **Collin, Robert E.** *Foundations for Microwave Engineering*. s.l. : McGraw-Hill International Editions, 1992.
6. **Hooberman, Benjamin.** *Everything You Ever Wanted to Know About Frequency-Selective Surface Filters but Were Afraid to Ask*. New York : University of Columbia, 2005.
7. **Holland, John.** *Adaptation in Natural and Artificial System*. Michigan : University of Michigan Press. , 1975.
8. **Koza, John.** *Genetic Programming: On the Programming of Computers by Means of Natural Selection*. Cambridge, MA : The MIT Press, 1992.
9. **Johnson, J. Michael e Rahmat-Samii, Yahya.** Genetic Algorithms in Engineering Electromagnetics. *IEEE Antennas and Propagation Magazine*. 1997, Vol. 39, 4.
10. **Marczyk, Adam.** Genetic Algorithms and Evolutionary Computation. *IEEE Transactions on Evolutionary Computation*. 2004, Vol. 3, 4.
11. **Forrest, Stephanie.** Genetics Algorithms: principles of natural selection applied to computation. *Science*. 1993, Vol. 261, pp. 872-878.
12. **Weiland, T.** A Discretization Method for the Solution of Maxwell's Equations for Six-Component Fields. *Electronics and Communications AEUE*. 1977, Vol. 31, 3, pp. 116–120.
13. **Kern, D. J. e Werner, D. H.** A GENETIC ALGORITHM APPROACH TO THE DESIGN OF ULTRA-THIN ELECTROMAGNETIC BANDGAP ABSORBERS. *MICROWAVE AND OPTICAL TECHNOLOGY LETTERS*. 2003, Vol. 38, 1.
14. **Ohira, Masataka, et al.** Multiband Single-Layer Frequency Selective Surface Designed by Combination of Genetic Algorithm and Geometry-Refinement Technique. *IEEE TRANSACTIONS ON ANTENNAS AND PROPAGATION*. NOVEMBER de 2004, Vol. 58, 11, pp. 2925-2931.



15. **Weile, Daniel S. e Michielssen, Eric.** *The use of domain decomposition genetic algorithms exploiting model reduction for the design of frequency selective surfaces.* Urbana : Elsevier, 1999. 186.
16. **Jun, Lu, Jian-Bo, Wang e Guan-Cheng, Sun.** Frequency selective surface structure optimized by genetic algorithm. *Chinese Physics B.* 7 de November de 2008, pp. 1598-1600.
17. **Goldberg, D. E. e Deb, K.** A Comparative Analysis of Selection Schemes Used in Genetic Algorithms. [A. do livro] Morgan Kaufmann. *Foundations of Genetic Algorithms.* 1991, pp. 69-93.
18. **Johnson, J. M.; Rahmat-Samii, Y.** Genetic algorithms in engineering electromagnetics. *IEEE Antennas Propagat. Mag.* 1997, Vol. 39, pp. 7-25.

## 9 Anexos

Algoritmo construído em Matlab:

---

**Main ( )**

---

```
clear all

clc

saveSol = zeros(500,2); % variable responsable de sauvegarder toutes les
solutions
contInd = 1; %index croissant qui compte le nombre de solutions essayees
sol = 0; %variable responsable pour l'identification d'une solution
pot2 = 1; %variable auxiliare pour changer la base 2 -> 10
bestFIT = 0; %sauvegarder le meilleur fit du moment
cont1 = 0; % compter le nombre de 1 dans une matrice

%Ouverture d'un model MWS
cst = actxserver('CSTStudio.application');
mws = invoke(cst, 'NewMWS');
app = invoke(mws, 'GetApplicationName');
ver = invoke(mws, 'GetApplicationVersion');
invoke(mws, 'OpenFile', 'C:\temp\MultiBand0918.cst');
%

tic %
N = 17; %ordre de la matrice
npop = 30; %taille de la population
Ni = ceil(N/2); %moitie de N
n_genes_tot=(Ni*(Ni+1)/2); %nombre de bits (genes)de chaque individu

pcross=0.6; % taux de cross over
pmut=0.01; % taux de mutation
n_el_tournoi=2; % nombre d'individus en chaque tournoi (selection de
parents par tournoi)
n_max_generations=15;% nombre max de generations (condition d'arret)
p_elit=0.1; % pourcentage d'individus selectionnes par l'elitisme
population = round(rand(npop,n_genes_tot)); % genere une population
initiale

for i=1:npop % evaluation de chaque individu de la population initiale
    i,1 % montre a l'ecran l'index pour controle

    for c = 1 : n_genes_tot %verifie le nombre de '1' dans la matrice
        if population(i,c) == 1
            cont1 = cont1 + 1;
        end
    end
end
```

```

    if cont1 >= (n_genes_tot)/2 %s'il y a plus de '1' que de '0', prendre
le complement de Babinet
        for m = 1 : n_genes_tot
            population(i,m) = ~population(i,m);
        end
    end

    x=+population(i,:); %Variable x recupere l'individu du moment presente
dans la population

    for n = 1:n_genes_tot %identification de la solution par son
correspondant de base 10
        sol = sol + x(n)*pot2;
        pot2 = pot2 * 2;
    end

    pot2 = 1; % reinitialization de la variable
fitness(i) = t2(cst,mws,x,N); % evalue le fitness pour l'individu i
contInd = contInd + 1; %sauvegarder les solutions
saveSol(contInd,2) = fitness(i);%
saveSol(contInd,1) = sol; %

    sol = 0; %reinitialisation des variables
    cont1 = 0;
end

[rank,ordre] = sort(fitness,'descend'); % trie les individus par le fitness
(ordre decroissant)

for n_generation = 1 : n_max_generations; %Methode de Creation des
nouvelles generations
    for i = 1 : ceil(p_elit*npop) %elitime -> selectionne les p_elit*npop
pour composer automatiquement la generation
        nouvel_population(i,:)=population(ordre(i),:) ;
    end
    i = ceil(p_elit*npop); % index doit commencer apres les elitistes

    while (i < npop)
        i = i + 1;

        %%%%selection des parents%%%%%%%%
        el_tournoi = ceil(npop*rand(1,n_el_tournoi)); % selectionne les
elements de la population pour le premier tournoi
        [v_fit_tournoi,vainqueur] = max(fitness(el_tournoi));% vainqueur =
index du meilleur individu du tournoi (meilleur fitness)
        parent1 = el_tournoi(vainqueur); %parent1= index de la population
        parent2 = parent1;
        while parent2 == parent1 % nouveau tournoi
            el_tournoi = ceil(npop*rand(1,n_el_tournoi));
            [v_fit_tournoi,vainqueur] = max(fitness(el_tournoi));
            parent2 = el_tournoi(vainqueur);
        end
        % parent1 et parent2 selectionnés pour le reproduction

        %%%%crossover%%%%%%%%%%%%

        if rand < pcross; %si rand < pcross, realiser crossover a 2 points

```

```

        % selection des 2 points de crossover
        cross_point1 = ceil((n_genes_tot - 1)*rand);
        cross_point2 = ceil((n_genes_tot - 1)*rand);
        if cross_point1 > cross_point2
            cross_point1 = cross_point2;
        end
        % changement de genes entre le 2 parents
        desc1(1:cross_point1) = population(parent1,1:cross_point1);
        desc1(cross_point1+1:cross_point2) =
population(parent2,cross_point1+1:cross_point2);
        desc1(cross_point2+1:n_genes_tot) =
population(parent1,cross_point2+1:n_genes_tot);
        else % si rand > pcross, le descendant rest egal au premier parent
            desc1 = population(parent1,:);
        end

        %%%%%%%%%MUTATION%%%%%%%%%
        for k = 1 : n_genes_tot; % pour chaque gene, si rand < pmut change le
gene
            if rand < pmut;
                desc1(k) = ~desc1(k);

            end
        end
        %%%%%%%%%%

        acc=0; % initialization de variable de controle de individus
similaires
        for m=1:(i-1) % verifie si il n'y a pas d'individuo egal dans la
population

            if desc1 == nouvl_population(m,:)
                acc = acc+1;
            end
        end

        if acc == 0 % si le descendant est different, il va composer la
nouvelle popoulation, si non, il sera genere a nouveau depuis la selection
            nouvl_population(i,:) = desc1;

        else
            i = i - 1;
        end

    end

    population = nouvl_population; %le variable popualtion peut maintenant
recevoir la nouvelle population genere
    for i = 1:npop % evalue la population
        fprintf ('Index et generation\n');
        i,n_generation

        for c = 1 : n_genes_tot %verifie le quantité de 1s dans la
matrice
            if population(i,c) == 1
                cont1 = cont1 + 1;
            end
        end
    end
end

```

```

        if cont1 >= (n_genes_tot)/2 %si il y a plus des 1s que 0s, prenda
le complement de Babinet
            for m = 1 : n_genes_tot
                population(i,m) = ~population(i,m);
            end
        end

        x+=population(i,:); %Variable x recupere l'individu du moment
presente dans la population
        for n = 1:n_genes_tot %identification de la solution par son
correspondant de base 10
            sol = sol + x(n)*pot2;
            pot2 = pot2 * 2;
        end

        pot2 = 1; % reinitialization de la variable

        fitness(i) = t2(cst,mws,x,N); % evalue le fitness pour l'individu i

        contInd = contInd + 1;           %sauvegarde les solutions
        saveSol(contInd,2) = fitness(i);%
        saveSol(contInd,1) = sol;       %

        sol = 0;                         %reinitialization des variables
        cont1 = 0;

    end

    [rank,ordre] = sort(fitness,'descend'); % trie les individus par le
fitness (du grande au petit)
    fitness(ordre(1)) %plot l'evolution du fitness par contre le nombre de
generations
    hold off
    evolution(n_generation)=max(fitness); % montre le meilleur resultat
dans chaque generation

    figure(1)
    plot(evolution)
    drawnow

    save populationFSS; % sauvegarde le workspace pour futures consultes
end
toc

x +=population(ordre(1),:);% recupere le meilleur individu a la fin

fitness(i) = t2(cst,mws,x,N); % lance le simulation a nouveau

```

---

```
function [fit] = t2(cst,mws,genes,N)
```

---

```
tic %initialization du horloge
cont = 1;
fit = 0; %valeur de fitness du individu
fo1 = 0.9; %premiere frequence d'interet
fo2 = 1.8; %deuxieme frequence d'interet
N_2 = ceil(N/2); %moitie de N pour dessiner matrice
[ma,numls] = doMatrix(N, genes); %apell pour la fonction qui genere la
matrice FSS

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% variables pour garder les valeurs des pics  trouves %%%
picMinFo1TE = 0; %pic minimale proche a fo1 (Mode TE)
picMinFo1TM = 0; %pic minimale proche a fo1 (Mode TM)
picMinFo1TE2 = 0; %pic minimale proche a fo1 (Mode TE, Theta = 30)
picMinFo2TE = 0; %pic minimale proche a fo2 (Mode TE)
picMinFo2TM = 0; %pic minimale proche a fo2 (Mode TM)
picMinFo2TE2 = 0; %pic minimale proche a fo2 (Mode TE, Theta = 30)

pbf = 0.2; %pourcentage de bande de frequence (tolerance)

%Instanciation d'un Composant Brick
invoke(mws, 'Rebuild');
brick = invoke(mws, 'Brick');
invoke(brick, 'Reset');

%Reset the angle of incidence
theta = invoke(mws, 'Boundary');
invoke(theta, 'SetPeriodicBoundaryAngles', '0', '0');

%Instanciation d'un Composant Solid
solid = invoke(mws, 'solid');
mat='PEC'; %Type du materiaux conducteur

%dessin de la matrice
for i = 1 : N_2
    for j = 1 : N_2

        if (ma(i,j) == 1) %si le bit == 1, dessine metal

            is = int2str(i);
            js = int2str(j);
            cs = int2str(cont);
            csSolid = strcat('solid',cs);

            %Les 4 limites du petit carre a dessiner (lp = longueur du
            %petit carre)
            xmin = strcat('(',js,'-1)*lp');
            xmax = strcat('(',js,'*lp');
            ymin = strcat('-',is,'*lp');
            ymax = strcat('-',is,'-1)*lp');

            %Commande pour dessiner un carre sur CST
            invoke(brick, 'Name', csSolid);
            invoke(brick, 'Component', 'component1');
```

```

        invoke(brick, 'Material', mat);
        invoke(brick, 'Xrange', xmin, xmax);
        invoke(brick, 'Yrange', ymin, ymax);
        invoke(brick, 'Zrange', '0', '0');
        invoke(brick, 'Create');

        % Commande pour ajouter des carres a chaque iteration de la
boucle
        if cont > 1
            cont = 1;

        invoke(solid, 'add', 'component1:solid1', 'component1:solid2');
        end

        cont = cont + 1;
    end

end

end

%Transformations type miroir

%Si N est impair, decaler l'axe de lp/2 pour assurer une matrice impaire
%a la fin des operations de miroir
if (mod(N,2) == 1)
    transf = invoke(mws, 'WCS');
    invoke(transf, 'ActivateWCS', 'local');
    invoke(transf, 'MoveWCS', 'local', 'lp/2', '0', '0');
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
end

%Transformation type miroir axe x
comp = invoke(mws, 'Transform');
invoke (comp, 'Reset');
invoke (comp, 'Name', 'component1');
invoke (comp, 'Origin', 'Free');
invoke (comp, 'Center', '0', '0', '0');
invoke (comp, 'PlaneNormal', '180', '0', '0');
invoke (comp, 'MultipleObjects', 'True');
invoke (comp, 'GroupObjects', 'False');
invoke (comp, 'Repetitions', '1');
invoke (comp, 'MultipleSelection', 'False');
invoke (comp, 'Component', '');
invoke (comp, 'Material', '');
invoke (comp, 'MirrorAdvanced');
invoke(solid, 'add', 'component1:solid1', 'component1:solid1_1 ');

%si N est impair, decaler l'axe pour realiser le miroir
if (mod(N,2) == 1)
    invoke(transf, 'MoveWCS', 'local', '0', '-lp/2', '0');
end

%Transformation type miroir axe y
comp = invoke(mws, 'Transform');
invoke (comp, 'Reset');
invoke (comp, 'Name', 'component1');
invoke (comp, 'Origin', 'Free');
invoke (comp, 'Center', '0', '0', '0');
invoke (comp, 'PlaneNormal', '0', '180', '0');

```

```

invoke (comp, 'MultipleObjects', 'True');
invoke (comp, 'GroupObjects','False');
invoke (comp, 'Repetitions','1');
invoke (comp, 'MultipleSelection', 'False');
invoke (comp, 'Component','');
invoke (comp, 'Material','');
invoke (comp, 'MirrorAdvanced');
invoke(solid,'add','component1:solid1','component1:solid1_1' );
release(brick);

%Debut de la simulation
solver = invoke(mws, 'FDSolver');
invoke(solver, 'Start');

%Recuperer les resultats dans un fichier (mode TE)
result = invoke(mws, 'Result1D', 'dZmin(1)Zmax(1)');
invoke(result, 'Save', 'c:\temp\resultatTE.txt');
S12TE = importdata('C:\temp\resultatTE.txt', ' ', 4);

%Recuperer les resultats dans un fichier (mode TM)
result = invoke(mws, 'Result1D', 'dZmin(2)Zmax(2)');
invoke(result, 'Save', 'c:\temp\resultatTM.txt');
S12TM = importdata('C:\temp\resultatTM.txt', ' ', 4);

%Les variables reçoivent les resultats suivants
XTE = S12TE.data(:,1);
YTE = S12TE.data(:,2);

XTM = S12TM.data(:,1);
YTM = S12TE.data(:,2);

%Chercher les pics dans les courbes des coefficients de transmission
[pks,locs] = findpeaks(1./YTE);

[pksTM,locSTM] = findpeaks(1./YTM);

figure(2)

%Dessiner les courbes
plot(XTE,YTE,XTM,YTM)
drawnow

%Variables reçoivent les valeurs liees aux pics (frequence et attenuation)
freqPicTE = XTE(locs);
dbPicTE = YTE(locs);
freqPicTM = XTM(locSTM);
dbPicTM = YTM(locSTM);

%Verifier s'il y a des bons pics dans la pbf (Mode TE)
for i = 1: length(locs)

    if (dbPicTE(i) < -30)
        if (freqPicTE(i) > (fo1-(fo1*pbf))) && (freqPicTE(i) <
fo1+(fo1*pbf))
            picMinFo1TE = freqPicTE(i);
        end
        if (freqPicTE(i) > (fo2-(fo1*pbf))) && (freqPicTE(i) <
fo2+(fo1*pbf))
            picMinFo2TE = freqPicTE(i);
        end
    end
end

```



```

        end

    end
end
%Verifier s'il y a des bons pics dans la pbf (Mode TM)
for i = 1: length(locsTM)

    if (dbPicTM(i) < -20)
        if (freqPicTM(i) > (fo1-(fo1*pbf))) && (freqPicTM(i) <
fo1+(fo1*pbf))
            picMinFo1TM = freqPicTM(i)
        end
        if (freqPicTM(i) > (fo2-(fo1*pbf))) && (freqPicTM(i) <
fo2+(fo1*pbf))
            picMinFo2TM = freqPicTM(i)
        end
    end
end

%Attribution de la valeur du fit
fitAux = (1-abs(fo1-picMinFo1TE)/fo1)*0.25 + (1-abs(fo1-
picMinFo1TM)/fo1)*0.25 + (1-abs(fo2-picMinFo2TE)/fo2)*0.25 + (1-abs(fo2-
picMinFo2TM)/fo2)*0.25;

%Si le motif est bon, on lance une simulation avec theta = 30°
%la procede est pareil pour le theta = 0 vu precedemment
if (fitAux > 0.80)

    invoke(theta, 'SetPeriodicBoundaryAngles', '30', '0');

    invoke(solver, 'Start');

    result2 = invoke(mws, 'Result1D', 'dZmin(1) Zmax(1)');
    invoke(result2, 'Save', 'c:\temp\resultat2.txt');
    S12TE = importdata('C:\temp\resultat2.txt', ' ', 4);

    XTE = S12TE.data(:,1);
    YTE = S12TE.data(:,2);

    [pks,locs] = findpeaks(1./YTE);

    result = invoke(mws, 'Result1D', 'dZmin(2) Zmax(2)');
    invoke(result, 'Save', 'c:\temp\resultatTM2.txt');
    S12TM = importdata('C:\temp\resultatTM2.txt', ' ', 4);

    XTM = S12TM.data(:,1);
    YTM = S12TM.data(:,2);

    [pksTM,locsTM] = findpeaks(1./YTM);

    fprintf('pics para Theta = 30\n');
    freqPicTE = XTE(locs);
    dbPicTE = YTE(locs);
    freqPicTM = XTM(locsTM);
    dbPicTM = YTM(locsTM);

```

```

for i = 1: length(locs)

    if dbPicTE(i) < -30
        if (freqPicTE(i) > (fo1-(fo1*pbf))) && (freqPicTE(i) <
fo1+(fo1*pbf))
            picMinFo1TE2 = freqPicTE(i)
        end
        if (freqPicTE(i) > (fo2-(fo1*pbf))) && (freqPicTE(i) <
fo2+(fo1*pbf))
            picMinFo2TE2 = freqPicTE(i)
        end
    end
end

%si cette simulation presente des bons pics dans la bande
%pbf, le fit doit recevoir un increment

if ((picMinFo1TE2 > 0) && (picMinFo2TE2 > 0))
    fit = fitAux + ((1 - abs(picMinFo1TE - picMinFo1TE2)/picMinFo1TE)
+ (1 - abs(picMinFo2TE - picMinFo2TE2)/picMinFo2TE))*0.2;
else
    fit = fitAux;
end

else %
    fit = fitAux;
end

toc

```

---

```
function [mat,compt1] = doMatrix(N,vetGenes)
```

---

```
%genere une matrice carre composee para '1' et '0'
moitie = round(N/2); %prendre la moitie de l'ordre de la matrice
lim = N+1; %limite pour le dessin
compt1 = 0; %compteur de '1'
Nimp = ceil(N/2); %prendre la moitie de l'ordre de la matrice
compt = 0; %compteur de genes
mat = rand(N,N); %instanciation d'une matrice NxN

%Procede pour remplir une partie de la matrice en prennant le chromosome
% (vecteur)
for i = 1 : Nimp
    for j = i : Nimp
        compt = compt + 1;
        mat(i,j) = vetGenes(compt);
    end
end

%Methode pour remplir la matrice
for i = 1 : N
    for j = 1 : N

        if (i <= moitie)

            if ((j <= moitie))
                if (j < i)          %remplir le premier quadrant
                    mat(i,j) = mat(j,i);
                end
            end

            if ((j > moitie))
                if ((i + j) <= lim)
                    mat(i,j) = mat(i,lim-j); %remplir le deuxieme
quadrant
                end

                if (i + j > lim)
                    mat(i,j) = mat(lim - j,i);
                end
            end

        end
        if (i > moitie)

            if (i == j)

                mat(i,j) = mat(lim-i,lim-j);
            end
            if (i + j == lim)

                mat(i,j) = mat(j,j);
            end
            if (j <= moitie)
```

```

    if (i + j < lim)
        mat(i,j) = mat(j,lim - i);
    end
    if (i + j > lim)
        mat(i,j) = mat(lim - i,j);
    end

end

if (j > moitie)

    if (i > j)
        mat(i,j) = mat(lim - i,lim - j);
    end
    if (i < j)

        mat(i,j) = mat(lim - j,lim - i);
    end

end

end

end

end

end

```