|  |
| --- |
| INSTITUTO TECNOLÓGICO DE AERONÁUTICA C:\Users\Gabriel\Desktop\ita.jpg  Carlos Mauricio de Lemos Sobrinho  Gabriel Lucas Gil Secco  Algoritmo Genético para Posicionamento de Roteadores em Redes de Sensores Sem Fio em cenário com Obstáculos  Trabalho de Graduação  *2013*  Computação |

Número da CDU (tamanho 10)

Carlos Mauricio de Lemos Sobrinho

Gabriel Lucas Gil Secco

Algoritmo Genético para Posicionamento de Roteadores em Redes de Sensores Sem Fio em cenário com Obstáculos

Orientador

Prof.ª Dr.ª Cecília de Azevedo Castro Cesar (ITA)

Engenharia de Computação

São José dos Campos

Departamento de Ciência e Tecnologia Aeroespacial

Instituto Tecnológico de Aeronáutica

2010

Dados Internacionais de Catalogação-na-Publicação (CIP)

Divisão de Informação e Documentação

|  |
| --- |
| Carlos Mauricio de Lemos Sobrinho, Gabriel Lucas Gil Secco  Aplicação de Algoritmo evolutivo para otimização multiobjetivo em cobertura e componente conexa de redes sem fio considerando cenário com obstáculos  São José dos Campos, 2010.  xxf.  Trabalho de Graduação – Divisão de Ciência da Computação – Instituto Tecnológico de Aeronáutica,  2013. Orientador: Prof.ª Dr.ª Cecília de Azevedo Castro Cesar (ITA)  1. Assunto1. 2. Assunto2. 3. Assunto3. I. Comando-Geral de Tecnologia Aeroespacial. II. Instituto Tecnológico de Aeronáutica. Divisão de Ciência da Computação. III. Desenvolvimento de Trabalhos Acadêmicos |

REFERÊNCIA BIBLIOGRÁFICA

SOBRINHO, Carlos Mauricio de Lemos; SECCO, Gabriel Lucas Gil. Aplicação de Algoritmo Evolutivo para Otimização Multiobjetivo em Cobertura e Componente conexa de redes sem fio considerando cenário com obstáculos. 2013. xxf. Trabalho de Conclusão de Curso (Graduação) – Instituto Tecnológico de Aeronáutica, São José dos Campos.

CESSÃO DE DIREITOS

NOME DOS AUTORES: Carlos Mauricio de Lemos Sobrinho, Gabriel Lucas Gil Secco

TÍTULO DO TRABALHO: Aplicação de Algoritmo Evolutivo para Otimização Multiobjetivo em Cobertura e Componente conexa de redes sem fio considerando cenário com obstáculos

TIPO DO TRABALHO/ANO: Graduação / 2013

É concedida ao Instituto Tecnológico de Aeronáutica permissão para reproduzir cópias deste trabalho de graduação e para emprestar ou vender cópias somente para propósitos acadêmicos e científicos. O autor reserva outros direitos de publicação e nenhuma parte desta monografia de graduação pode ser reproduzida sem a autorização do autor.

Carlos Mauricio de Lemos Sobrinho

Endereço do autor

**D**

Aplicação de Algoritmo Evolutivo para Otimização Multiobjetivo em Cobertura e Componente conexa de redes sem fio considerando cenário com obstáculos

Essa publicação foi aceita como Relatório Final de Trabalho de Graduação

Carlos Mauricio de Lemos Sobrinho

Autor

Gabriel Lucas Gil Secco

Autor

Prof.ª Dr.ª Cecília de Azevedo Castro Cesar (ITA)

Orientador

Prof. Dr. Fábio Carneiro Mokarzel

Coordenador do Curso de Engenharia de Computação

São José dos Campos, XX de xxxxxxxxxx de 2010

Dedico este trabalho a A, B e C

**Agradecimentos**

A X, Y e Z.

“O fracasso não existe, o que existe é o sucesso dos outros.”

Aviões do Forró

**Resumo**

Redes sem fio vem se mostrando cada vez mais competitivas quando comparadas a  a redes cabeadas. Nesse contexto, um problema de difícil solução e de grande importância é realizar um posicionamento eficiente dos roteadores. Neste trabalho é desenvolvido um algoritmo genético que leva em consideração um modelo simplificado de obstáculos, para posicionar roteadores de modo a se obter cobertura total de um conjunto sensores fixos e total conectividade entre os roteadores.

**Abstract**

Wireless networking is proving increasingly competitive when compared to wired networks. In this context, a difficult  and very important problem is to perform an efficient routers placement. In this graduation work we developed a genetic algorithm that takes into account a simplified model of barriers to place routers in order to obtain full coverage of a fixed set of sensors and full connectivity between the routers.

Lista de Figuras

Figura 1: Planta para . 20

Figura 2: Representação dos Sensores na planta. 22

Figura 3 - Quatro Roteadores(Azul) cobrindo 4 sensores em um exemplo sem obstáculos e com . 23

Figura 4 – Obstáculos 24

Figura 5 – Um Roteador(Azul) cobrindo 2 sensores em um exemplo com obstáculos, , . 25

Figura 6 – Quatro Roteadores cobrindo 5 sensores em um exemplo com obstáculos, , . 26

Figura 7 – Quatro Roteadores formando uma única componente conexa em um exemplo com obstáculos, , . 27

Figura 8 – Exemplo com 8 sensores para teste de velocidade. 31

Figura 9 – Representação das probabilidades de escolha em um gráfico tipo pizza. 33

Figura 10 – Cruzamento do tipo Corte Único. 33

Figura 11 – Cruzamento do tipo Multi Corte. 34

Figura 12 – Individuo sofrendo mutação de dois bits. 35

Figura 13 – Aptidão máxima e média para probabilidade de mutação igual a 50% por bit. 35

Figura 14 – Aptidão máxima e média para probabilidade de mutação igual a 5% por bit. 36

Figura 15 37

Figura 16 38

Figura 17 – Região de cobertura de três sensores em uma planta sem obstáculos. 41

Figura 18 – Região de cobertura de três sensores em uma planta sem obstáculos. Agora um pouco mais próximos em relação a Fig. 17. 42

Figura 19 45

Figura 20 46

Figura 21 47

Figura 22 48

Figura 23 49

Figura 24 50

Figura 25 52

Figura 26 53

Figura 27 54

Figura 28 55

Figura 29 56

Figura 30 57

Figura 31 58

Figura 32 59

Figura 33 60

Figura 34 61

Figura 35 61

Figura 36 62

Figura 37 63

Figura 38 63

Figura 39 64

Figura 40 65

Figura 41 65

Figura 42 66

Figura 43 67

Figura 44 67

Figura 45 68

Figura 46 68

Figura 47 69

Figura 48 70

Figura 49 71

Figura 50 72

Figura 51 73

Figura 52 74

Figura 53 75

Figura 54 76

Figura 55 77

Figura 56 78

Figura 57 79

Figura 58 80

Figura 59 81

Figura 60 82

Figura 61 83

Figura 62 84

(figuras)

**Nenhuma entrada de índice de ilustrações foi encontrada.** (tabelas)

Lista de Abreviaturas, Siglas e Símbolos

Sumário

[Lista de Figuras 23](#_Toc372875479)

[Lista de Abreviaturas, Siglas e Símbolos 25](#_Toc372875480)

[**1.** **Introdução** 14](#_Toc372875481)

[1.1. Estrutura do Trabalho 15](#_Toc372875482)

[**2.** **Problema Abordado** 15](#_Toc372875483)

[2.1. Complexidade 16](#_Toc372875484)

[2.2. Planta 17](#_Toc372875485)

[2.3. Sensores 17](#_Toc372875486)

[2.4. Roteadores 18](#_Toc372875487)

[2.5. Obstáculos 19](#_Toc372875488)

[2.6. Espaço de Busca 23](#_Toc372875489)

[2.7. Método de Solução 24](#_Toc372875490)

[**3.** **Algoritmo Genético** 24](#_Toc372875491)

[3.1. Definições Gerais e nomenclatura 25](#_Toc372875492)

[3.1.1. Termos referentes ao Algoritmo Genético 25](#_Toc372875493)

[3.1.2. Termos referentes a modelagem 26](#_Toc372875494)

[3.2. Modelagem 26](#_Toc372875495)

[3.2.1. Planta 27](#_Toc372875496)

[3.3. População Inicial 28](#_Toc372875497)

[3.4. Função de Aptidão 29](#_Toc372875498)

[3.4.1. Apresentação 29](#_Toc372875499)

[3.4.2. Cálculo 30](#_Toc372875500)

[3.4.3. Desempenho 31](#_Toc372875501)

[3.5. Escolha dos mais aptos 32](#_Toc372875502)

[3.6. Função de Cruzamento 33](#_Toc372875503)

[3.7. Mutação 34](#_Toc372875504)

[3.8. Condições de parada 39](#_Toc372875505)

[3.9. Limitações 40](#_Toc372875506)

[**4.** **Algoritmos Alternativos** 43](#_Toc372875507)

[4.1. Algoritmo de Busca Aleatório 43](#_Toc372875508)

[4.2. Algoritmo de Busca Gulosa 43](#_Toc372875509)

[**5.** **Resultados** 44](#_Toc372875510)

[5.1. Variáveis de ambiente 44](#_Toc372875511)

[5.2. Cenários de testes 44](#_Toc372875512)

[5.3. Resultados dos Testes 49](#_Toc372875513)

[5.3.1. Algoritmo Genético 49](#_Toc372875514)

[5.3.2. Busca Aleatória 71](#_Toc372875515)

[5.3.3. Busca Gulosa 77](#_Toc372875516)

[6. Conclusão 77](#_Toc372875517)

[7. Referencias 78](#_Toc372875518)

[8. Apêndice 79](#_Toc372875519)

1. **Introdução**

Conceitos de algoritmos genéticos executados em computadores apareceram em 1954 quando Nils Aall Barricelli, pela primeira vez, simulou a teoria da evolução. Em 1975, John Henry Holland publicou “*Adaptation in Natural and Artificial Systems*”. Trabalho considerado como sendo a bíblia dos AG’s.

Atualmente, Algoritmos de Genéticos são aplicados na resolução de inúmeros problemas. Existem soluções com AGs para otimização de parâmetros de controle, movimentos no xadrez, robótica, logística, entre outros.

Com esta ferramenta, é possível simular problemas complexos e com infindável número de possibilidades de soluções usando um conceito biológico simples: “Apenas os indivíduos mais aptos irão propagar suas características aos seus descendentes” (Charles Darwin,1858).

É comum cenários nos quais existem diversos dispositivos, fixos ou móveis, os quas devem estar conectados a um ponto, host central, computador, assim por diante. Uma abordagem mais simplista é criar uma rede cabeada de modo a conectá-los.

O surgimento de tecnologias sem fio possibilitou o surgimento de dispositivos móveis e facilitou a implementação de redes. Frente a um enorme potencial de mercado, características como taxa de dados, tempo de resposta e segurança tiveram melhorias significativas. Cada vez mais redes sem fio são utilizadas nos mais diversos cenários.

Apesar de baratas e eficientes, existe uma dependência com o local de instalação não tratada pelos protocolos. Fica a cargo do projeto de instalação a tarefa de posicionar os dispositivos roteadores sem fio de modo a atender os requisitos. O tratamento desse problema é de fundamental importância, pois um procedimento de posicionamento muito simplista, ou até mesmo arbitrário, pode demandar um número elevado de dispositivos, causando aumento desnecessário no custo de implementação da rede.

Na comunidade cientifica é possível encontrar diversos tipos de soluções para esta problemática. Junfang Wang(2007), por exemplo, utiliza uma heurística para determinação de cobertura dos sensores e conectividade baseada em busca gulosa e agregação. Não há tratamento de obstáculos.

O objetivo deste trabalho foi obter uma solução eficiente para o posicionamento roteadores em uma rede de sensores sem fio que garanta a cobertura dos sensores e a conectividade da rede garantindo a entrega da demanda de tráfego para a Internet. O posicionamento foi obtido por meio da execução de um Algoritmo Genético.

* 1. Estrutura do Trabalho

Primeiramente, no Cap. 2, falaremos acerca do problema abordado. Serão apresentados todos os seus componentes, valores possíveis e espaço de busca.

No Cap. 3, o algoritmo implementado será discutido. Serão apresentadas as suas estruturas internas e uma breve análise da escolha de parâmetros utilizada.

No Cap. 4, duas soluções alternativas implementadas serão apresentadas bem como suas vantagens, desvantagens, limitações e estrutura interna.

No Cap. 5 serão definidos os casos de testes nos quais o programa e as soluções alternativas serão testadas bem como os resultados extraídos de cada um dos algoritmos testados. Ainda neste capitulo, as soluções extraídas são analisadas à luz do objetivo do trabalho evidenciado no Cap. 1.

1. **Problema Abordado**

O problema de posicionamento de roteadores é a determinação de um conjunto mínimo de posições, entre as posições candidatas, de tal maneira que os roteadores situados nestas posições possam cobrir uma determinada região, manter o plena conectividade para o Gateway e atender a demanda de tráfego. (Junfang Wang,2007).

Neste trabalho, a conectividade para o Gateway não foi abordada.

Por plena conectividade entenda-se que dados quaisquer dois roteadores posicionados, existe um caminho de passagem de pacotes de rede entre eles. A Cobertura é definida pela quantidade total de sensores conectados.

Considere uma planta onde todos os componentes estão localizados. Existe um conjunto de sensores e um conjunto de obstáculos. O problema é, considerando as atenuações causadas pelas barreiras, calcular e posicionar o melhor número de roteadores necessários para cobrir todos estes sensores (cobertura completa) e, adicionalmente, possuírem conectividade.

* 1. Considerações

1 - todos os elementos de entrada (sensores, obstáculos) são fixos.

2 - sensores se comunicam somente com roteadores.

3 - roteadores tem a capacidade de se comunicar entre si.

4 - existem somente obstáculos retos.

5 - Barreias causam atenuação constante e independe do angulo de incidência no sinal eletromagnético.

* 1. Complexidade

Considere o problema de posicionar roteadores em uma planta genérica onde existe um número de posições disponíveis. Então, o número possível de posicionamentos distintos, ou Espaço de Busca, é dado por:

Para exemplificar, considere uma planta com na qual se deseja colocar roteadores. A Tabela 1 contém os valores de para o número de posicionamentos possíveis.

|  |  |  |
| --- | --- | --- |
|  |  |  |
| 1 | 100 | 100 |
| 2 | 100 | 4950 |
| 3 | 100 | 161700 |
| 4 | 100 | 3921225 |
| 5 | 100 | 75287520 |
| 6 | 100 | 1192052400 |
| 7 | 100 | 16007560800 |
| 8 | 100 | 186087894300 |
| 9 | 100 | 1902231808400 |
| 10 | 100 | 17310309456440 |

Tabela 1 – Numero de posicionamentos possíveis.

A alta complexidade motiva-nos a encontrar uma abordagem eficaz para o posicionamento de roteadores. (Junfang Wang,2007).

O cruzamento das restrições de posições com os objetivos de cobertura e conectividade leva a criação de um problema de otimização NP-Difícil.

No caso deste trabalho, iremos utilizar o algoritmo genético. Este tipo de algoritmo pertence à classe dos algoritmos de busca randômicos guiados (Fatos Xhafa, 2010).

* 1. Método de Solução

A solução será o *output* de uma aplicação em computador que, pelo uso de um algoritmo genético em uma planta entrada pelo usuário, calcula o menor número de roteadores necessários para cobrir todos os pontos entrados. O programa vai, iterativamente, adicionando mais roteadores, se necessário, até que a maior componente conexa tenha o tamanho igual ao número de roteadores posicionados. Em Algoritmos Genéticos, a função de aptidão é responsável por traduzir as características desejadas. Esta função é apresentada mais detalhadamente na Seção 3.4.

Os dados serão entrados por meio de uma interface gráfica que possibilite o acesso intuitivo e simples à ferramenta de criação de cenário. Ademais, o programa fornecerá gráficos da dinâmica de atenuação, componente conexa, cobertura dos pontos e gráfico histórico dos valores de aptidão do melhor indivíduo de uma dada geração e a média. Estes elementos servirão de apoio à validação da solução.

1. **Algoritmo Genético**

Algoritmos Genéticos são algoritmos de otimização global, baseados nos mecanismos de seleção natural. Eles empregam uma estratégia de busca voltada em direção ao reforço dos pontos de “alta aptidão”, ou seja, pontos nos quais a função a ser minimizada (ou maximizada) tem valores relativamente baixos (ou altos).

Apesar de possuírem eventos probabilísticos na sua execução, eles apresentam dinâmica direcionada, pois exploram informações históricas para encontrar novos pontos de busca onde são esperados melhores desempenhos.

Técnicas de computação evolucionária operam sobre uma população de candidatos em paralelo. Assim, elas podem fazer a busca em diferentes áreas do espaço de solução, alocando um número de membros apropriado para a busca em várias regiões.

Além de ser uma estratégia de gerar-e-testar muito elegante, por serem baseados na evolução biológica, são capazes de identificar e explorar fatores ambientais e convergir para soluções ótimas, ou aproximadamente ótimas em níveis globais. Embora possam parecer simplistas do ponto de vista biológico, estes algoritmos são suficientemente complexos para fornecer mecanismos de busca adaptativo poderosos e robustos.

* 1. Definições Gerais, Considerações e nomenclatura
     1. Termos referentes ao Algoritmo Genético

Aqui se encontram os elementos do AG implementado e suas respectivas definições:

1 – Indivíduo: cadeia de bits que representa uma solução possível para o problema.

2 – População: conjunto de pontos (indivíduos) no Espaço de Busca

3 – Função Objetivo (Função de aptidão): Avalia um indivíduo de acordo com os objetivos desejados.

4 – Geração: iteração completa do AG que gera uma nova população.

5 – Aptidão: saída gerada pela função objetivo para um indivíduo da população.

6 – Mutação: Evento aleatório que pode ou não modificar os bits de um indivíduo. Cada bit possui uma probabilidade de ser modificado.

* + 1. Termos referentes a modelagem

Aqui se encontram os elementos da modelagem realizada e suas respectivas definições:

1 – : número de bits da representação binaria de todas as entidades envolvidas.

2 – Planta: Mapa quadrado de lado onde todos os elementos estão inseridos.

3 – Sensor: Elemento de rede presente na planta. Localização denotada por um ‘\*’ vermelho.

4 – Roteador: Elemento de rede presente na planta. Localização denotado por um ‘+’ azul.

5 – Obstáculo: representados por segmentos de reta na planta. Pretos na planta de entrada e verdes na planta de visualização da resposta.

6 – : O alcance da comunicação em visada direta entre roteador e sensor.

7 - : O alcance da comunicação em visada direta entre roteadores.

8 - : Coeficiente de atenuação de sinal respectivo a um obstáculo.

* 1. Modelagem

Para realizar a execução do AG foi necessário modelar o problema. O tratamento algorítmico do cenário é considerar um plano cartesiano discreto, desse modo as posições dos sensores, roteadores, início e fim de barreiras são representados por pontos nesse plano.

Um ponto do mapa é representado por um par ordenado , onde e são números inteiros de bits.

* + 1. Planta

A planta é constituída por um mapa quadrado de lado .



Figura 1: Planta para .

Considere como sendo o Fator de Granularidade da planta. é dado pela razão entre a área da planta original e o número de quadrados gerados pela quantização das medidas para valores possíveis.

O intuito é fazer a correspondência com uma planta real. Quanto maior , maior será a granularidade, i.e, maior será a precisão do modelo. Como exemplo, considere um terreno com medidas . Se então, o terreno de está dividido em quadrados. Logo,

Comparando-se o fator de granularidade com o alcance dos roteadores é possível analisar a confiabilidade da solução apresentada pelo programa. Este tipo de análise não será feita neste trabalho.

* + 1. Sensores

O significado destas entidades pode ser abstraído como sendo pontos em uma área que precisam ser cobertos. Por exemplo: estações de trabalho ou sensores remotos para aquisição de dados.

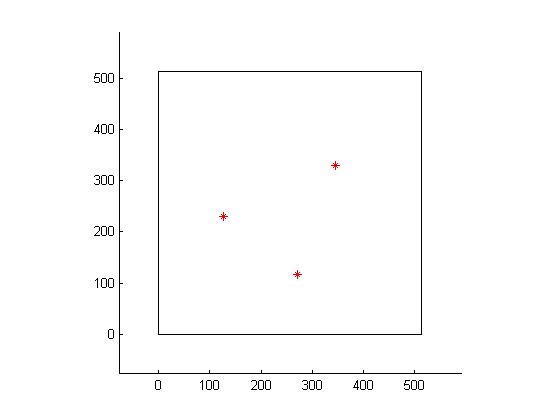


Figura 2: Representação dos Sensores na planta.

* + 1. Roteadores

Não necessariamente . Sensores remotos costumar ter limitações de energia para seu funcionamento enquanto que um roteador está ligado diretamente à rede elétrica e, portanto, não possui restrições de consumo tão apertadas.

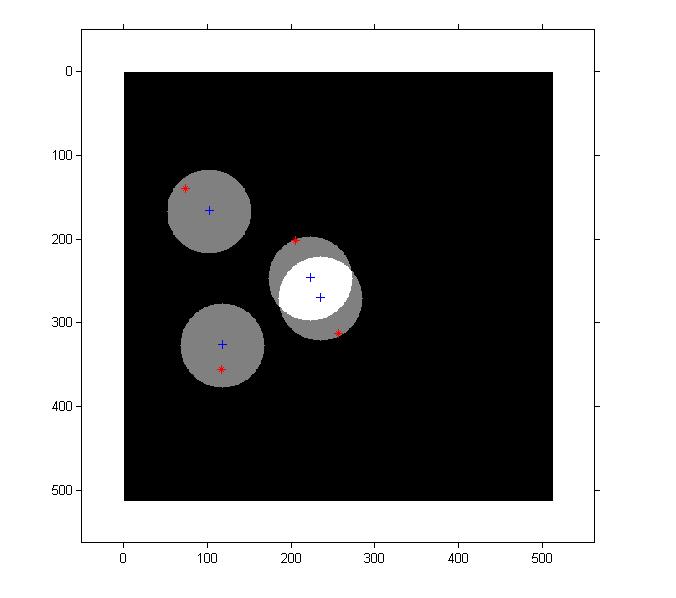


Figura 3 - Quatro Roteadores(Azul) cobrindo 4 sensores em um exemplo sem obstáculos e com .

* + 1. Obstáculos

Os obstáculos são responsáveis pela atenuação do sinal. O grau de atenuação é dado pelo valor da variável . A cada passagem por um barreira a intensidade do sinal eletromagnético é multiplicada por este valor. Após passagens, tem-se :

Onde e representam, respectivamente, os alcances antes e depois de atravessar a barreira. Para casos de atenuação do sinal, .

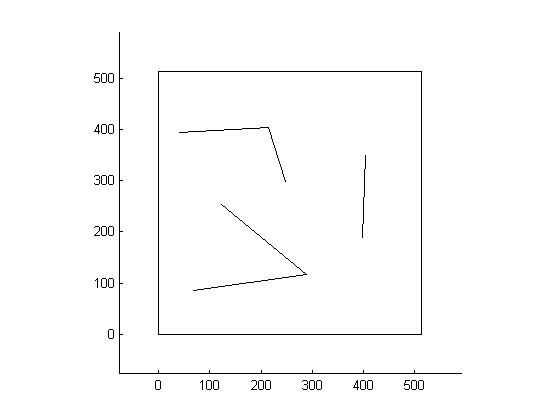


Figura 4 – Obstáculos

A inserção de possíveis barreiras no problema de posicionamento cria uma complexidade extra. A Figura 5 e a Figura 6 mostram cenários com a inclusão de um número crescente de barreiras no problema.

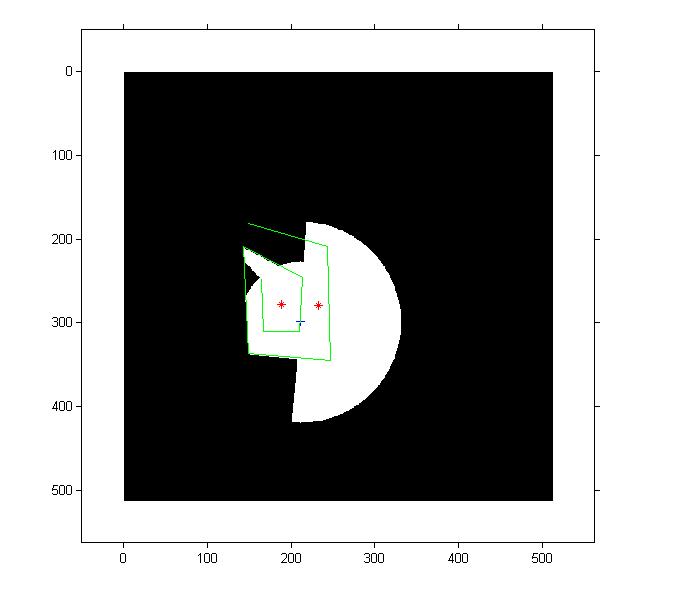


Figura 5 – Um Roteador(Azul) cobrindo 2 sensores em um exemplo com obstáculos, , .

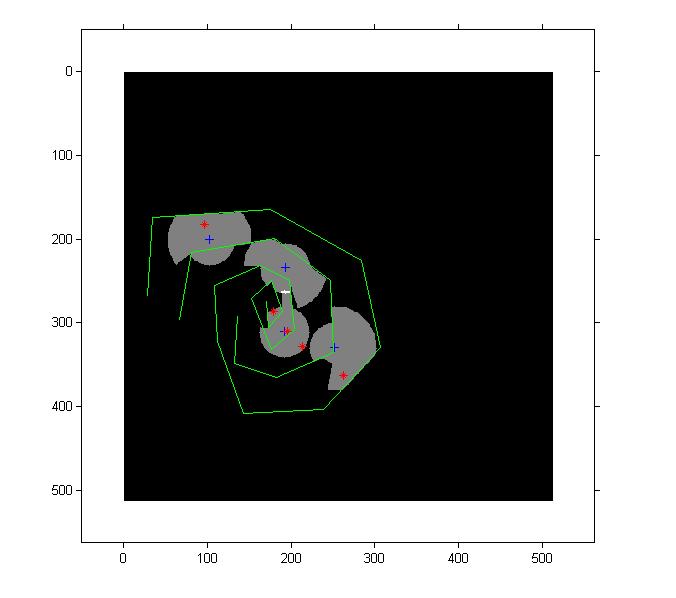


Figura 6 – Quatro Roteadores cobrindo 5 sensores em um exemplo com obstáculos, , .

Na Fig. 5, há poucos sensores e barreiras, é fácil para um humano conceber a localização ideal de um roteador que cubra os dois sensores. No entanto, na Fig. 6, o grande número de barreiras e sensores torna o processo mais complexo. Estes exemplos mostram o quão difícil é a análise do posicionamento quando muitos obstáculos são incorporados ao mapa.

Ainda na Fig. 6 é possível ver um roteador que não cobre sensor algum. No entanto, na Fig. 7 fica evidente a sua função de ponte para construir uma configuração de roteadores interconectados.

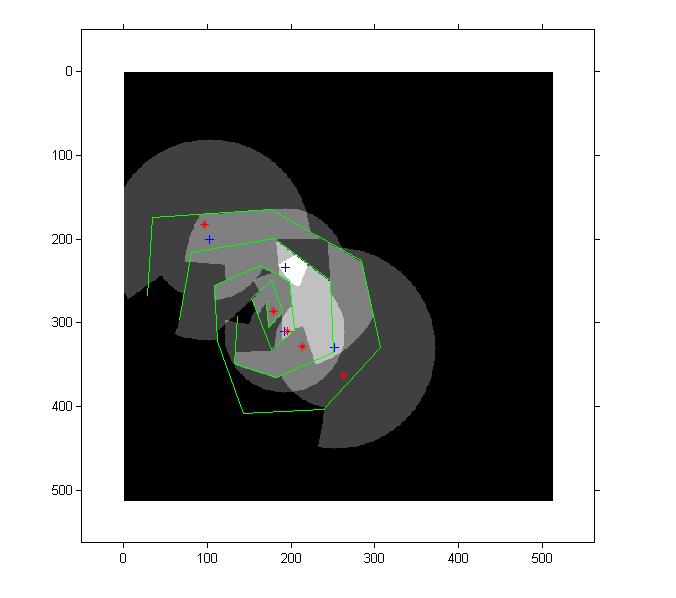


Figura 7 – Quatro Roteadores formando uma única componente conexa em um exemplo com obstáculos, , .

* + 1. Grafo de conectividade

Seja o número de barreiras entre dois dispositivos e . Eles estão conectados se, e somente se, onde é o raio de comunicação entre e com visada direta e é o operador distância euclidiana.

Define-se um grafo não orientado , no qual cada roteador é um vertice e a aresta existe se, e somente se, está conectado com , conforme a descrição de conectividade acima.

* 1. Objetivos após a modelagem

Objetivo agora é encontrar as posições de roteadores de modo que:

1 - Todo sensor esteja conectado a pelo menos um roteador.

2 - O tamanho da componente conexa do grafo seja igual ao número total de roteadores.

* 1. Indivíduo

Um indivíduo representa um candidato à solução do problema, i.e, um conjunto de posições no mapa referentes às localizações dos roteadores. No escopo deste trabalho, o indivíduo é composto pela simples concatenação dos pares ordenados que representam as posições dos roteadores.

Considere um indivíduo que representa uma solução com 4 roteadores. Este indivíduo possui a seguinte configuração.

Cada par ordenado representa uma posição na planta.

A posição dos roteadores configura a saída do programa.

* 1. População Inicial

A presença de soluções convencionais geradas por heurística na população inicial do Algoritmo Genético conduz a resultados estatisticamente melhores(Yow-Yuh Leu, 2007).

Quanto maior o tamanho da população maior é a chance de que o estado inicial da população contenha o indivíduo que representa a solução ideal (STANLEY GOTSHALL, 2010).

No entanto, populações demasiadamente grandes tornam a execução muito lenta. Durante os testes, o número de indivíduos foi configurado de forma a propiciar resultados em menos de 8 horas de execução.

Tendo em vista a conclusão de Yow-Yuh Leu, uma heurística de geração de população inicial foi desenvolvida. À solução da heurística, replicada para o tamanho da população, foi adicionado um ruído para criar uma população inicial com maior diversidade. O ruído utilizado foi de do tamanho da planta para cada direção das duas componentes referentes a posição do roteador.

A diversidade é importante em algoritmos genéticos, pois cruzando-se indivíduos de uma população homogênea não se produz novas soluções.

* 1. Função de Aptidão

Seja um individuo de uma população . A função de aptidão é dada por:

O número real retornado define o grau de aptidão de um indivíduo. A função de aptidão é responsável pela avaliação dos indivíduos e define quais são mais ou menos aptos à reproduzir e transmitir suas características as novas gerações.

A função de aptidão deve traduzir claramente os objetivos desejados, caso contrário a solução encontrada pode não ser a esperada. Definir claramente o que é almejado pode levar a criação de funções muito complexas.

A cada iteração, todos os indivíduos passam pela função de aptidão. Caso esta função seja lenta toda a execução do programa será afetada.

* + 1. Cálculo

Para o problema abordado neste trabalho foram identificados 3 indicadores que traduzem os objetivos desejados para uma solução satisfazer:

1 - Fração dos sensores cobertos ();

2 - Fração dos roteadores na maior componente conexa (.

3 - Número de ligações entre roteadores ().

As primeiras componentes reiteram o que fora dito na Seção 1.2 acerca dos objetivos almejados pela execução do programa. A última foi incluída, pois propicia uma maior conectividade entre os roteadores, ajudando na divisão de carga de pacotes transmitidos em rede, mas não faz parte dos objetivos primários do programa.

Sejam:

o número total de sensores na planta;

o número total de roteadores na planta;

a matriz de adjacências que representa a conexão entre os roteadores;

o elemento da linha e coluna de ;

o número de sensores cobertos;

o número de roteadores na maior componente conexa.

Então, temos definidas as variáveis abaixo para o cálculo da função de aptidão.

Cada uma das componentes criadas é normalizada. Foram atribuídos pesos 7:2:1, respectivamente, para a criação do valor de aptidão ()

* + 1. Desempenho

Para indivíduos maiores, o cálculo de é bem mais lento em comparação a . Assim definida, a função de aptidão realizaria todos os cálculos para qualquer indivíduo da população. Tal tarefa demandaria muito tempo para os exemplos mais complexos.

Foi criada uma variável chamada, tal que . O programa somente calculará as duas últimas componentes de se .

Esta pequena mudança causou um enorme impacto no tempo total de execução do algoritmo. Para exemplificar, considere o cenário da Figura 8.

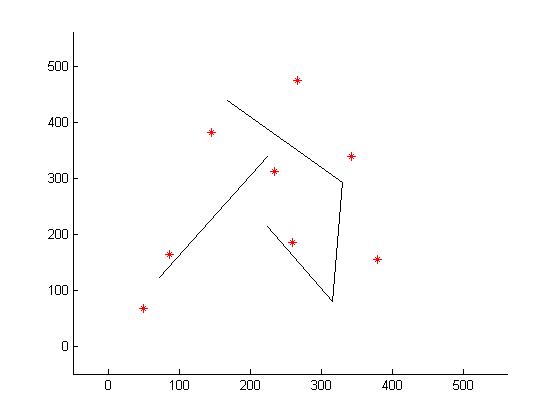


Figura 8 – Exemplo com 8 sensores para teste de velocidade.

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Teste | | minCoverage | Roteadores | População | | Iterações | | Aptidão média | Melhor aptidão | Tempo(s) | |
| 1 | 0,9 | | 10 | 400 | 200 | | 8,99 | | 9,46 | 120 |
| 2 | Não usada | | 10 | 400 | 200 | | 8,11 | | 9,44 | 625 |

Tabela 3.1 – Tempo de execução observado para uma mesma planta com e sem o uso da variável .

O tempo total caiu para menos de um quinto do original. Outros testes retornaram valores diferentes para a razão de tempo gasto uma vez que existe probabilidades envolvidas de forma que um conjunto razoável de indivíduos de uma população tenha maior que

* 1. Escolha dos mais aptos

Uma “alta aptidão” não é garantia de sucesso para um dado indivíduo. No entanto, ela define a probabilidade deste indivíduo participar da reprodução e propagar suas características. Este ambiente que propicia maior probabilidade aos mais aptos é chamado de elitista. Algoritmos genéticos podem ser de zero a 100% elitistas.

|  |  |  |  |
| --- | --- | --- | --- |
| ID | Individuo | Aptidão Absoluta | Aptidão Relativa |
| 1 | 11010 | 5 | 0,108695652 |
| 2 | 11011 | 7 | 0,152173913 |
| 3 | 10001 | 11 | 0,239130435 |
| 4 | 10101 | 3 | 0,065217391 |
| 5 | 10111 | 20 | 0,434782609 |

Tabela 3.2 – Valores fictícios de aptidão para o cálculo da probabilidade de ser escolhido para reproduzir.

Figura 9 – Representação das probabilidades de escolha em um gráfico tipo pizza.

* 1. Função de Cruzamento

A função de cruzamento é responsável por misturar as características dos pais. Este efeito ocorre por meio de um ou mais cortes na cadeia de bits dos pais e a junção cruzada dos pedaços cortados (Fig. 10).

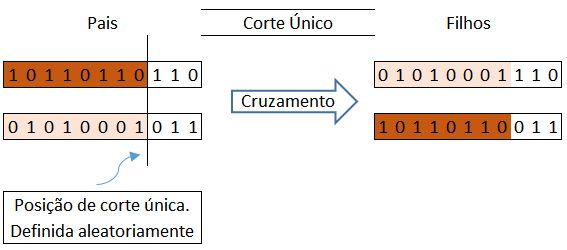


Figura 10 – Cruzamento do tipo Corte Único.

O número de cortes em um indivíduo determina o tipo de cruzamento. Há duas possibilidades: Corte único e multi corte. Nesse trabalho utilizou-se função de cruzamento multi corte conforme mostrado na Figura 11.

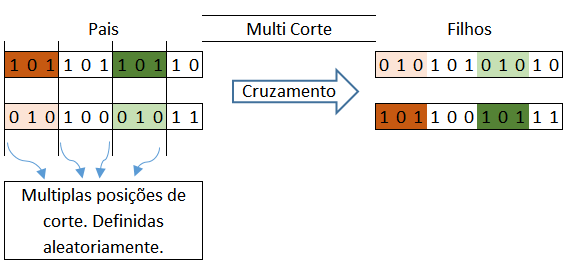


Figura 11 – Cruzamento do tipo Multi Corte.

* 1. Mutação

Em algoritmos genéticos, a mutação é frequentemente considerada como um operador de fundo cuja única importância é evitar que o algoritmo prematuramente convirja para uma solução sub-ótima. (Thomas Bäck, 1993)

A mutação consiste de um evento puramente aleatório que troca o valor de alguns bits dos filhos gerados do cruzamento.

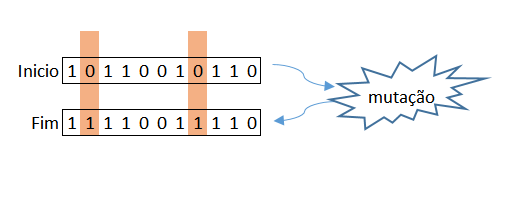


Figura 12 – Individuo sofrendo mutação de dois bits.

A probabilidade deve ser balanceada uma vez que valores muito baixos podem deixar a população presa a máximos locais da função de aptidão. Em contrapartida, valores muito altos podem conduzir a dinâmica aleatória não direcionada.

Nas Figs. 13, 14, 15 e 16 são mostrados as evolução dos valores máximo e médio da função de aptidão para probabilidades de mutação 50% ,5% , 0,5% e 0,1% por bit. O número de bits do indivíduo em questão é 162(9 roteadores).

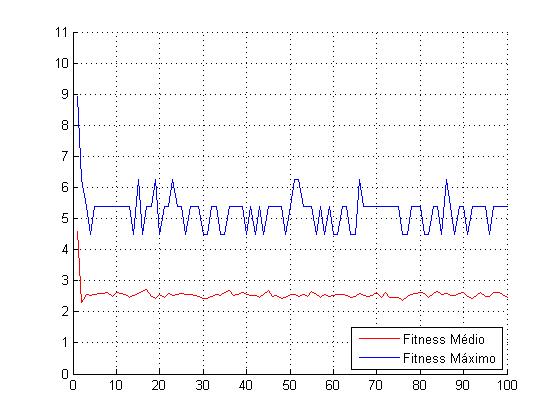


Figura 13 – Aptidão máxima e média para probabilidade de mutação igual a 50% por bit.

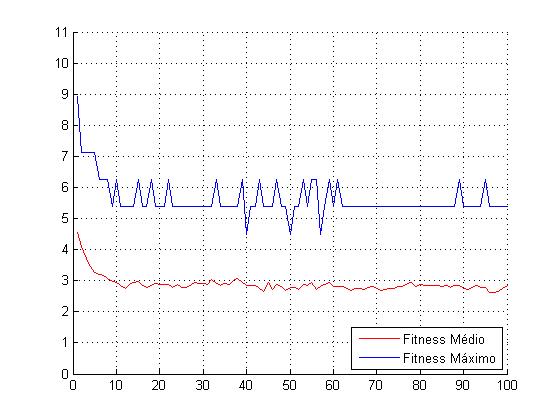


Figura 14 – Aptidão máxima e média para probabilidade de mutação igual a 5% por bit.

Nos dois casos acima, o nível de mutação é tão grande que mesmo o direcionamento pela função de aptidão não é capaz de manter a qualidade da população.

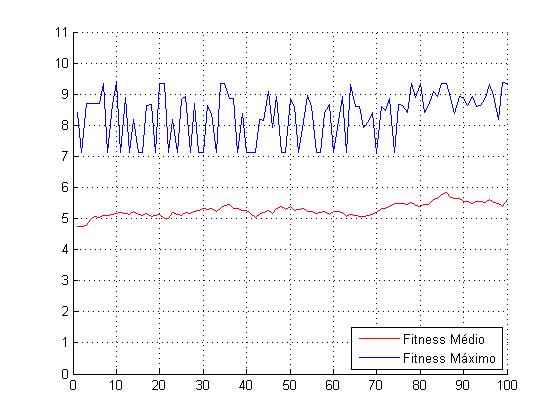


Figura 15

Figura 3.7 – Aptidão máxima e média para probabilidade de mutação igual a 0,5% por bit.

Neste caso, a variação para a aptidão máxima ainda é muito grande. No entanto, a média é consideravelmente maior do que nos casos anteriores.

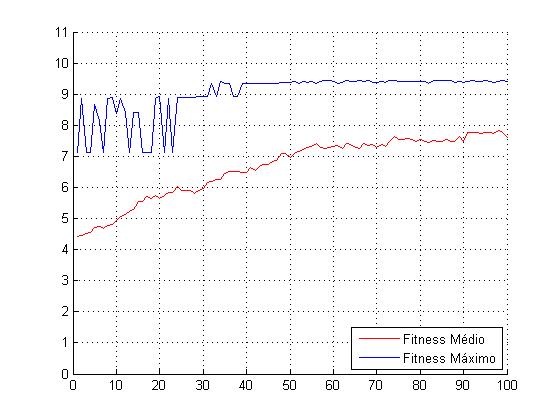


Figura 16

Figura 3.8 – Aptidão máxima e média para probabilidade de mutação igual a 0,1% por bit.

Aqui, o comportamento evolutivo pode ser melhor observado. A aptidão média é a maior dentre os quatro casos analisados.

Com o número de bits e a probabilidade de bits podemos calcular o valor esperado de bits modificados por indivíduo.

Onde representa o tamanho do individuo em numero de bits e a porcentagem de mutação por bit.

|  |  |  |
| --- | --- | --- |
|  |  |  |
| 162 | 0,5 | 81 |
| 162 | 0,05 | 8,1 |
| 162 | 0,005 | 0,81 |
| 162 | 0,001 | 0,162 |

Tabela 3.3 – Valor esperado de bits modificados por indivíduo.

Em algoritmos genéticos, a probabilidade de mutação mais recorrentemente aplicada é de (Thomas Bäck,1993).

Para o valor de recomendado por Bäck o valor esperado pode ser calculado com o uso da expressão anterior.

De fato, o caso mais próximo de apresentou um comportamento interessante e com bom valor de aptidão máximo. Entretanto, a curva de melhor performance foi obtida para próximo de 0,2.

Para obter para todos os tamanhos de indivíduos tratados durante a execução do programa, usou-se a seguinte formula para a determinação dinâmica de :

Subistituindo na formula para tem-se:

Sendo o numero de bits do individuo, a partir do que fora dito na Seção 3.2 tem-se:

Logo, a expressão para fica:

* 1. Condições de parada

Classicamente, podemos definir três tipos de condições de parada para algoritmos genéticos(Martín Safe,2004):

1: Numero de iterações;

2: Determinada característica ou valor de aptidão encontrados;

3: Convergência.

É possível, obviamente, utilizar mais de um critério

Quanto maior for o número de iterações maior é a chance de convergência dos valores de aptidão da população (STANLEY GOTSHALL, 2010).

Não foram feitas muitas elucubrações a respeito do número ideal de iterações, uma vez que o tempo de execução não é um requisito do problema.

Desse modo utilizou-se o critério 2. A característica desejada é cobertura e conectividade totais. A busca pode convergir para soluções não desejadas, nesse caso seria interessante implementar uma parada por convergência, por facilidade foi implementada a parada por número de iterações.

Em resumo, para este trabalho utilizou-se um número de iterações manual e critério de parada por característica encontrada.

* 1. Limitações

Considere o problema de cobrir 3 sensores em uma planta sem obstáculos com apenas um roteador. Obviamente, para que seja possível a existência de tal solução, é necessário que existam pontos da planta tais que a distância destes até os sensores seja menor que o raio de cobertura do roteador. No entanto, o número de pontos com esta característica pode variar muito.

Na Fig. 18, os sensores estão posicionados de maneira que a região que deve ser ocupada pelo roteador é muito maior do que a representada na Fig. 17.

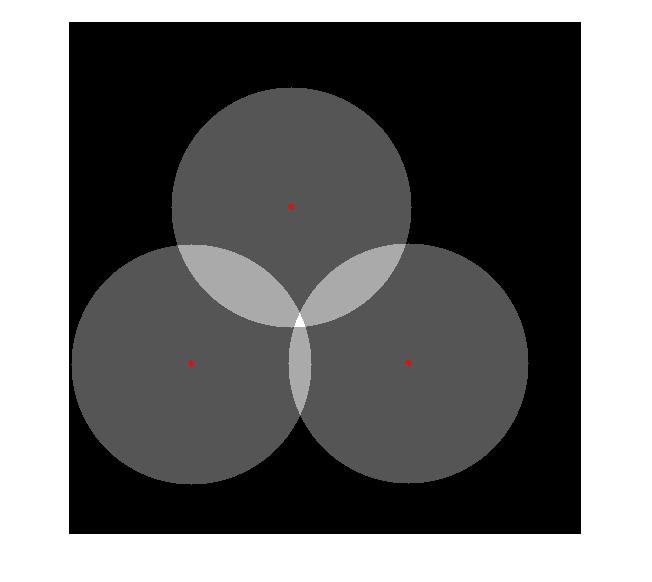


Figura 17 – Região de cobertura de três sensores em uma planta sem obstáculos.

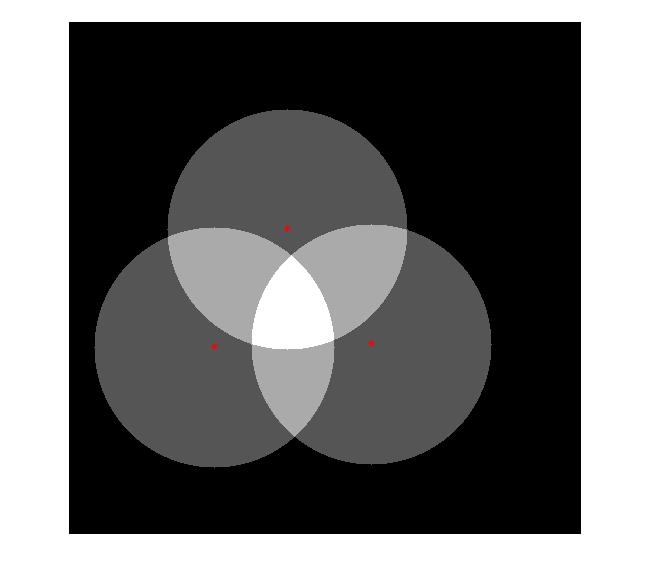


Figura 18 – Região de cobertura de três sensores em uma planta sem obstáculos. Agora um pouco mais próximos em relação a Fig. 17.

Em ambas as configurações, a colocação de um roteador na região mais clara proporciona cobertura completa. Entretanto existe uma dificuldade muito maior em se encontrar uma solução para a primeira configuração uma vez que a relação entre número de soluções ideais e totais é muito menor.

Problemas desta natureza podem ser contornados com auxílio de funções objetivo mais complexas e com maior número de valores possíveis. Outro tipo de solução possível é considerar estas regiões na população inicial por meio do uso de heurísticas de geração da população inicial em algoritmos genéticos. A segunda abordagem foi realizada neste trabalho. Logo, espera-se que a existência de pequenas regiões com cobertura privilegiada não configure um teste estatisticamente impossível para o programa desenvolvido.

1. **Algoritmos Alternativos**

Conforme citado na Seção 2.5, o problema tratado por este trabalho é NP-Difícil. O Espaço de busca para os testes realizados é muito maior que o número de soluções testadas.

Os algoritmos testados neste capitulo foram implementados pelos autores. Duas heurísticas de busca foram usadas: Aleatória e Gulosa.

* 1. Algoritmo de Busca Aleatório

O algoritmo implementado prevê um valor aleatório para cada uma das componentes bidimensionais da localização do roteador.

O algoritmo aleatório possui uma característica interessante de, ao mesmo tempo, testar opções diferentes e com perfil espalhado dos roteadores. O que contribui para uma maior eficiência na cobertura total.

No entanto, o número total de possibilidades inviabiliza seu uso em comparação ao algoritmo genético uma vez que neste há o direcionamento dos indivíduos de uma população por meio da função de fitness.

* 1. Algoritmo de Busca Gulosa

De uma forma geral, algoritmos gulosos tomam decisões baseadas na informação da iteração corrente, e essa decisão otimiza um subproblema, mas nem sempre encontra a solução correta.

Tendo isto em vista, seja o conjunto de todos os sensores, o algoritmo decide por colocar um roteador em um ponto que cobrirá o maior subconjunto , o próximo roteador será colocado no local que cobre o maior subconjunto de , assim por diante.

A implementação desse algoritmo baseia-se na construção de uma *lookup* *table* que contem entradas, cada uma delas correspondendo a uma célula da planta, para cada entrada é armazenada uma lista dos sensores a serem cobertos caso um roteador seja posicionado na célula em questão. Para cada novo roteador a ser posicionado essa tabela é refeita. Caso o número de roteadores entrado seja maior que o mínimo necessário para cobertura total o algoritmo irá posicionar os roteadores adicionais aleatoriamente.

O algoritmo tem desempenho satisfatório para a otimização de cobertura, porém não é capaz de maximizar a componente conexa da solução.

A solução desta heurística de busca foi usada para a geração dos indivíduos da primeira iteração do algoritmo genético.

1. **Resultados**

Neste capítulo serão mostrados os testes de execução para o Algoritmo Genético implementado e para as duas heurísticas de busca alternativas citadas no Cap. 4.

Em seguida, os resultados de cada heurística serão confrontados com os do Algoritmo genético.

Os testes serão executados sobre o modelo apresentado. Necessita-se ainda definir as variáveis de ambiente relativas aos cenários de testes criados. Em seguida, sensores e obstáculos serão posicionados definindo-se assim os casos de testes.

* 1. Variáveis de ambiente

As variáveis de ambiente são valores de entrada estáticos necessários para a construção do cenário de testes. A tabela a seguir contém estas variáveis e seus valores para os testes realizados.

|  |  |
| --- | --- |
| **Variável** | **Valor** |
|  | 0,6 |
|  | 50 |
|  | 200 |
|  | 9 |

Tabela 5.1 – Variáveis de ambiente.

* 1. Cenários de testes

Foram confeccionados 10 casos de testes. Dois para cada tipo de dispersão de sensores na planta. Os tipos de dispersão foram interpretados como variações na densidade de pontos.

|  |  |  |
| --- | --- | --- |
| **Densidade** | **Número de sensores** | **Testes** |
| Baixíssima | 4 | 1 e 2 |
| Baixa | 8 | 3 e 4 |
| Média | 16 | 5 e 6 |
| Alta | 32 | 7 e 8 |
| Altíssima | 64 | 9 e 10 |

Tabela 5.2 – Classificação dos casos de teste quando a densidade/número de sensores na planta.

O intuito desta classificação é criar cenários com complexidade crescente. Cada par de testes é composto pela mesma dispersão de sensores, um com e outro sem barreiras.

As figuras 5.1 a 5.5 a seguir ilustram os casos de testes impares.

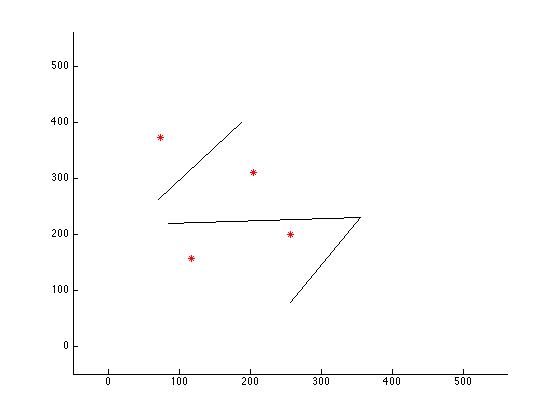


Figura 19 - Teste 1

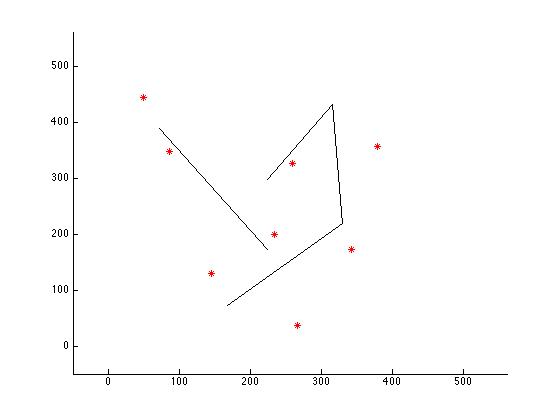


Figura 20 - Teste 3

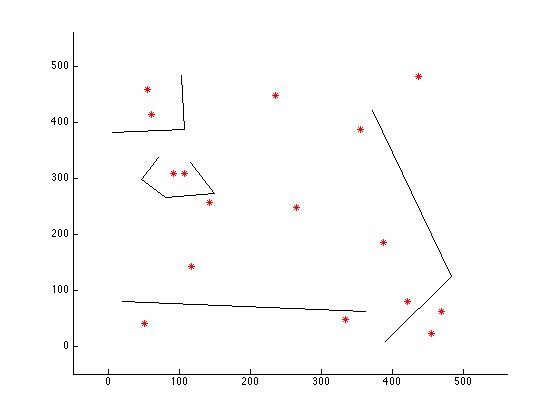


Figura 21 - Teste 5

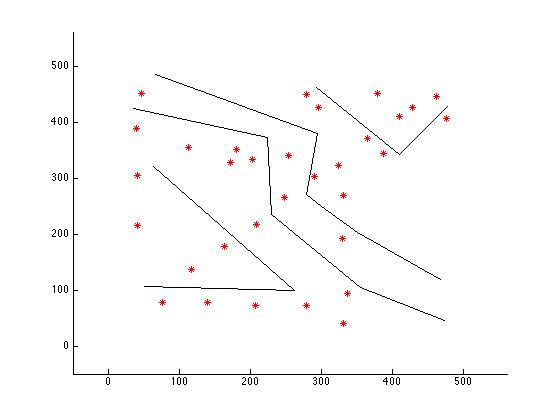


Figura 22 - Teste 7

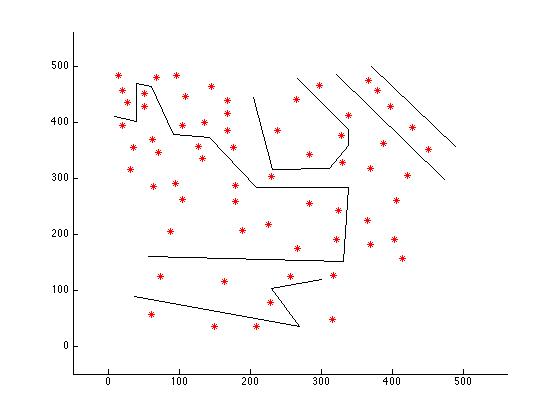


Figura 23 - Teste 9

* 1. Resultados dos Testes

Os algoritmos foram comparados a luz da função de aptidão do melhor indivíduo gerado. Visando uma comparação minimamente justa, a quantidade de soluções aleatoriamente geradas foram iguais ao número total de indivíduos avaliados durante o algoritmo genético caso ocorresse somente a parada por número de iterações. Na comparação com a heurística gulosa escolheu-se a melhor dentre 100 execuções para um mesmo teste.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Teste** | **Roteadores** | **Algoritmo Genético** | **Busca Gulosa** | **Busca Aleatória** |
| 1 | 4 | 9,625 | 8,3125 | 7,75 |
| 2 | 4 | 10 | 9,5 | 6,25 |
| 3 | 8 | 9,44 | 8,7344 | 7,125 |
| 4 | 8 | 9,5625 | 9,2969 | 7,125 |
| 5 | 15 | 9,29 | 8,4756 | 7,75 |
| 6 | 12 | 9,375 | 8,5556 | 6,8 |
| 7 | 18 | 9,07 | 7,1813 | 6,5 |
| 8 | 18 | 9,5494 | 9,0185 | 6,68 |
| 9 | 26 | 9,2 | 6,8531 | 6,4 |
| 10 | 20 | 9,45 | 7,2606 | 6,1 |

Tabela x – Resultado dos testes com Formatação condicional

* + 1. Algoritmo Genético

Os valores da tabela a seguir referem-se a execução do algoritmo genético para pelo menos uma repetição de cada caso de teste.

As ilustrações a seguir apresentam os resultados dos testes 1 a 10.

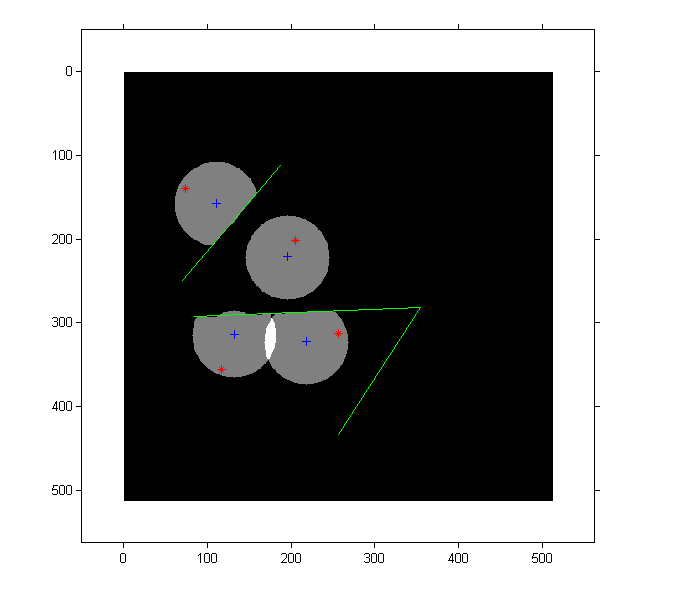


Figura 24

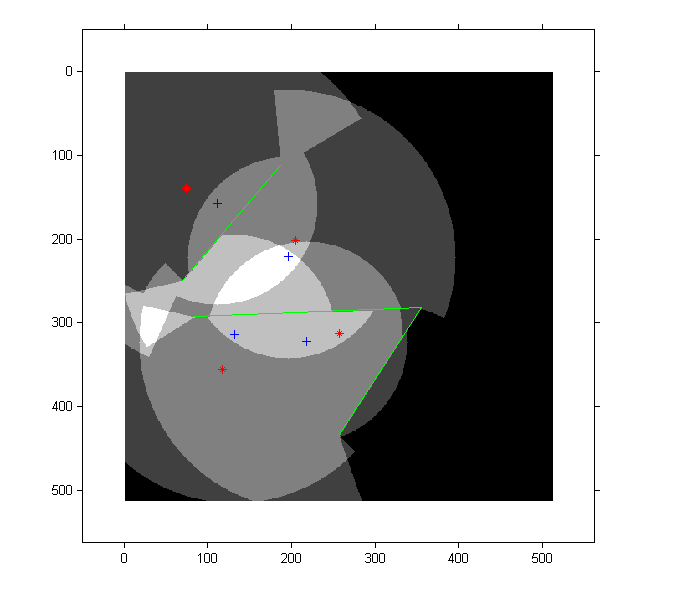
Figura 5.6 – Mapa de sensores cobertos para o teste 1.

Figura 5.7 – Mapa para visualização de comunicação entre roteadores, considerando para o teste 1.

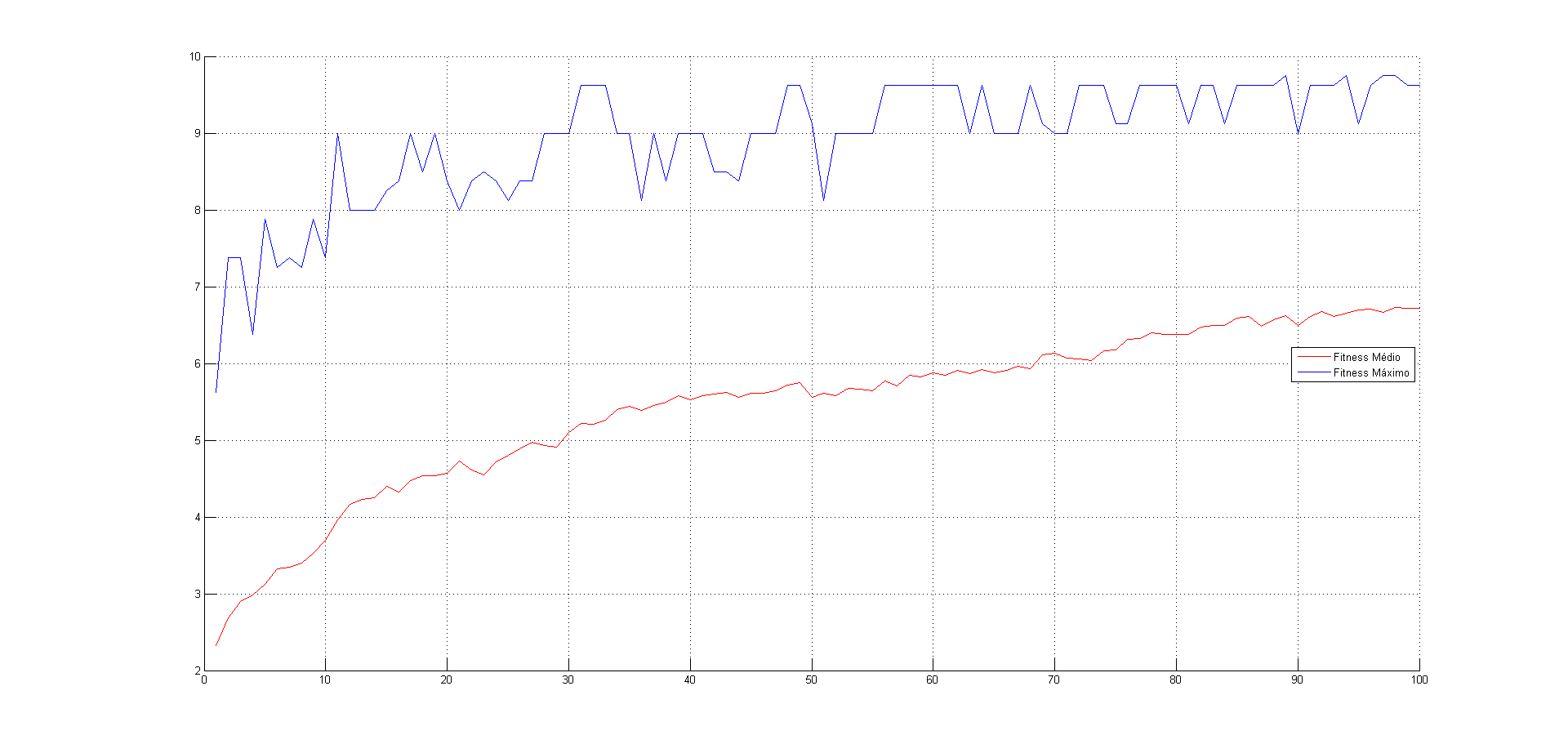


Figura 25

Figura 5.8 - Evolução da aptidão do teste 1.

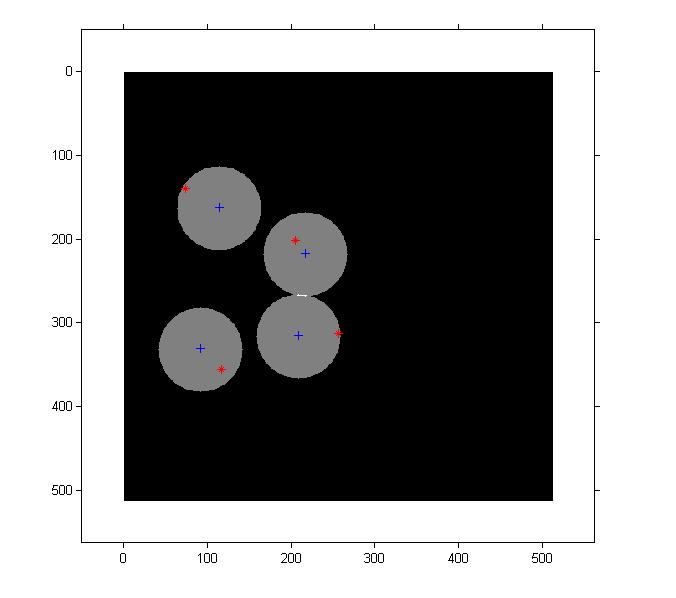


Figura 26

Figura 5.9 – Mapa de sensores cobertos para o teste 2.

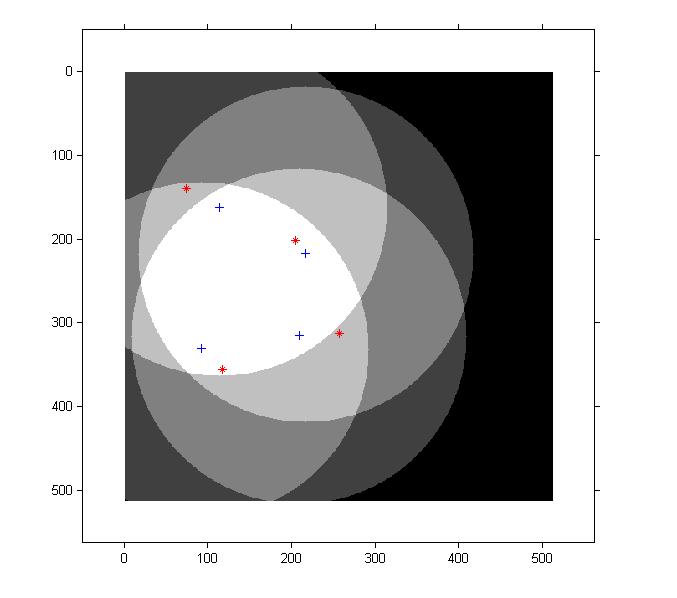


Figura 27

Figura 5.10 – Mapa para visualização de comunicação entre roteadores, considerando para o teste 2.

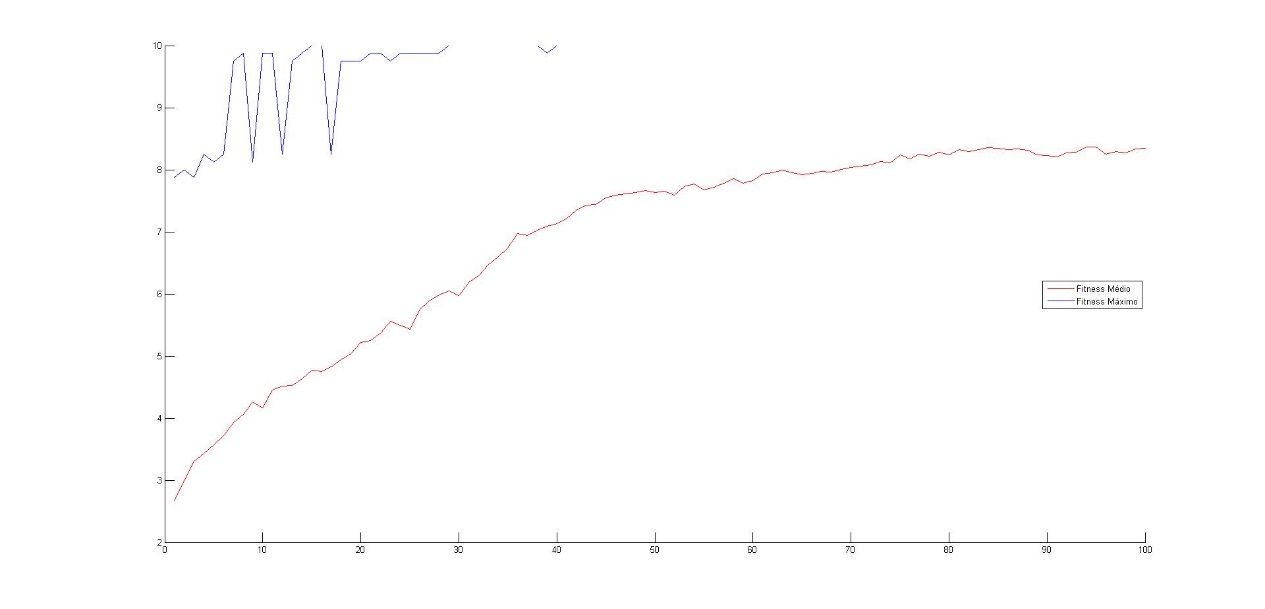


Figura 28

Figura 5.11 - Evolução da aptidão do teste 2.

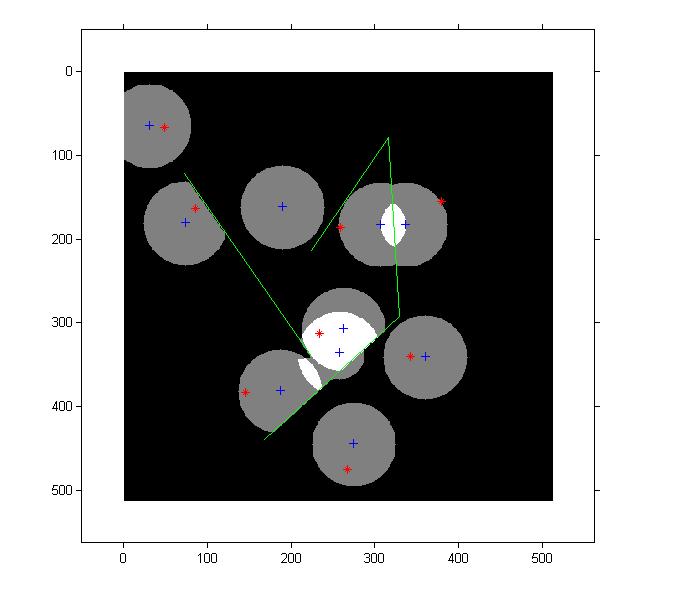


Figura 29

Figura 5.12 - Figura 5.9 – Mapa de sensores cobertos para o teste 3.

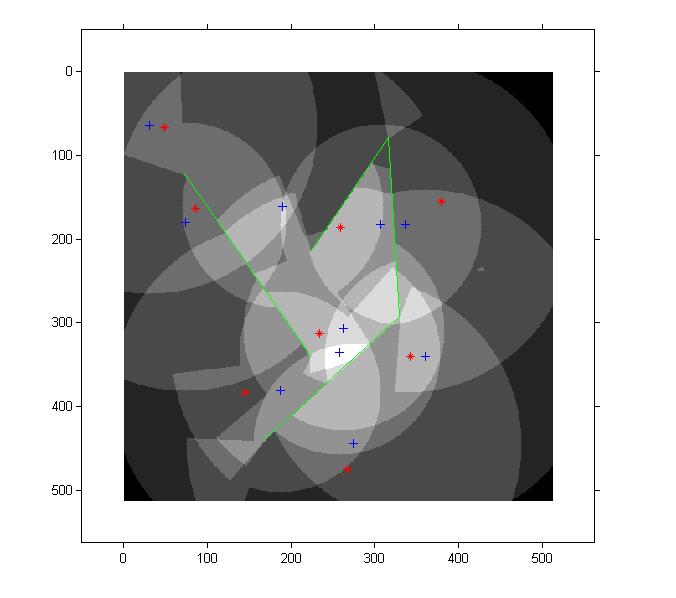


Figura 30

Figura 5.13 - Mapa para visualização de comunicação entre roteadores, considerando para o teste 3.

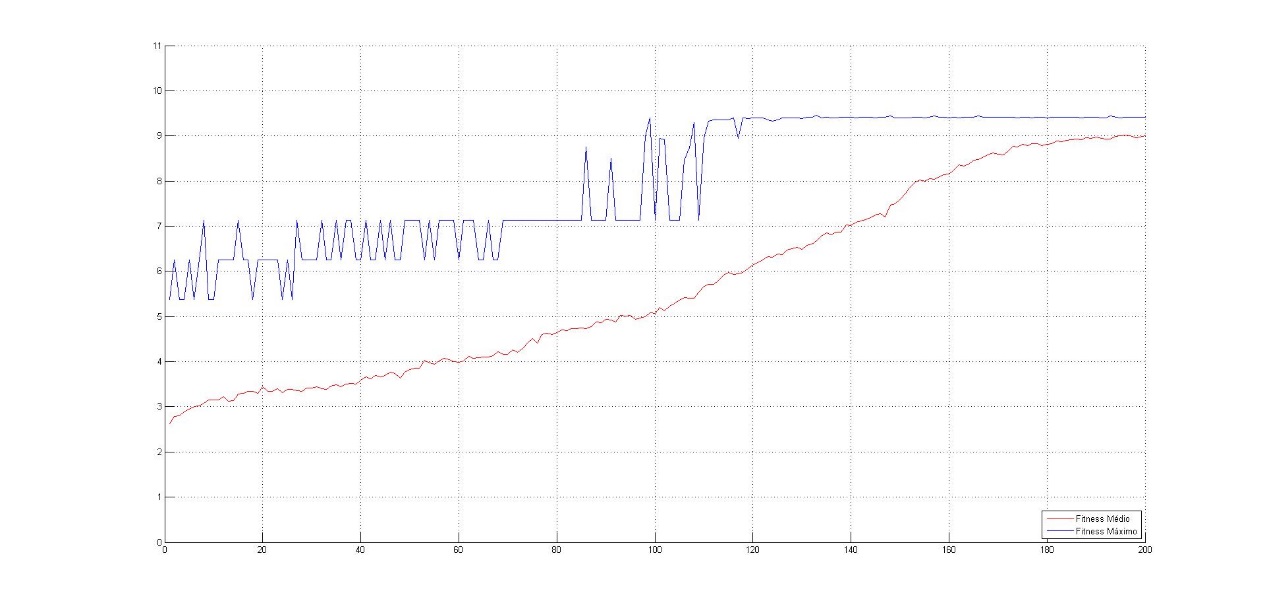


Figura 31

Figura 5.14 – Evolução da aptidão para o teste 3.

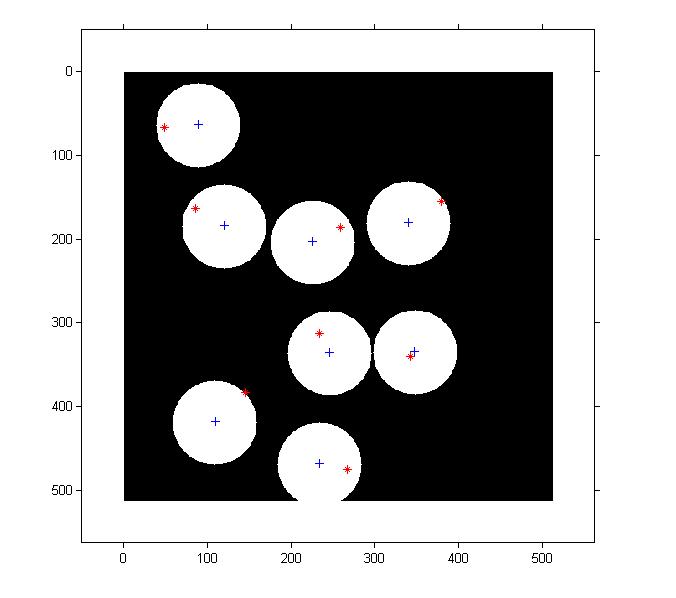


Figura 32

Figura 5.15 - Figura 5.9 – Mapa de sensores cobertos para o teste 4.

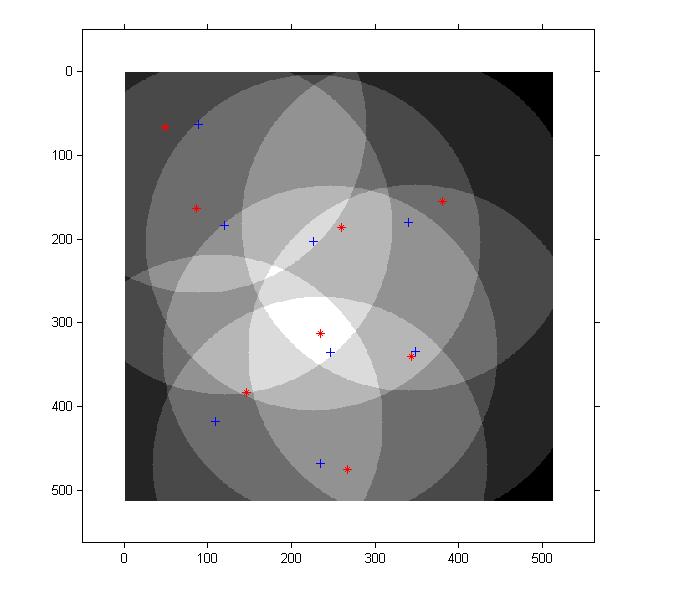


Figura 33

Figura 5.16 - Mapa para visualização de comunicação entre roteadores, considerando para o teste 4.

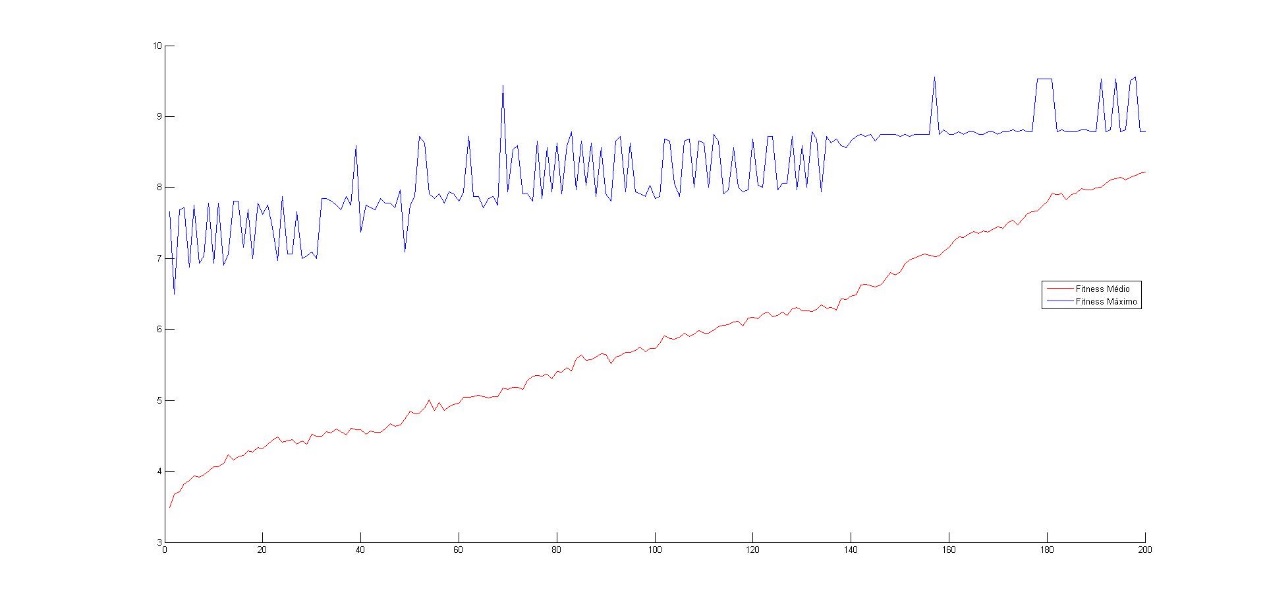


Figura 34

Figura 5.17 – Evolução da aptidão para o teste 4.

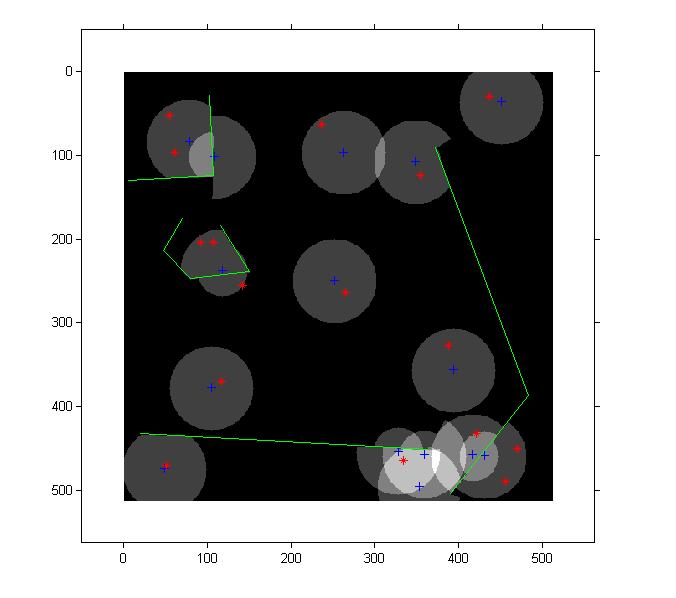


Figura 35

Figura 5.18 - Mapa de sensores cobertos para o teste 5.

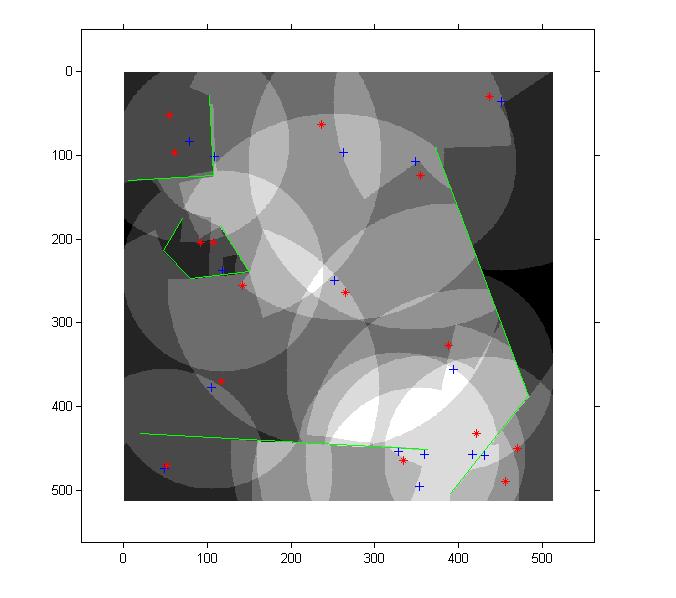


Figura 36

Figura 5.19 - Mapa para visualização de comunicação entre roteadores, considerando para o teste 5.

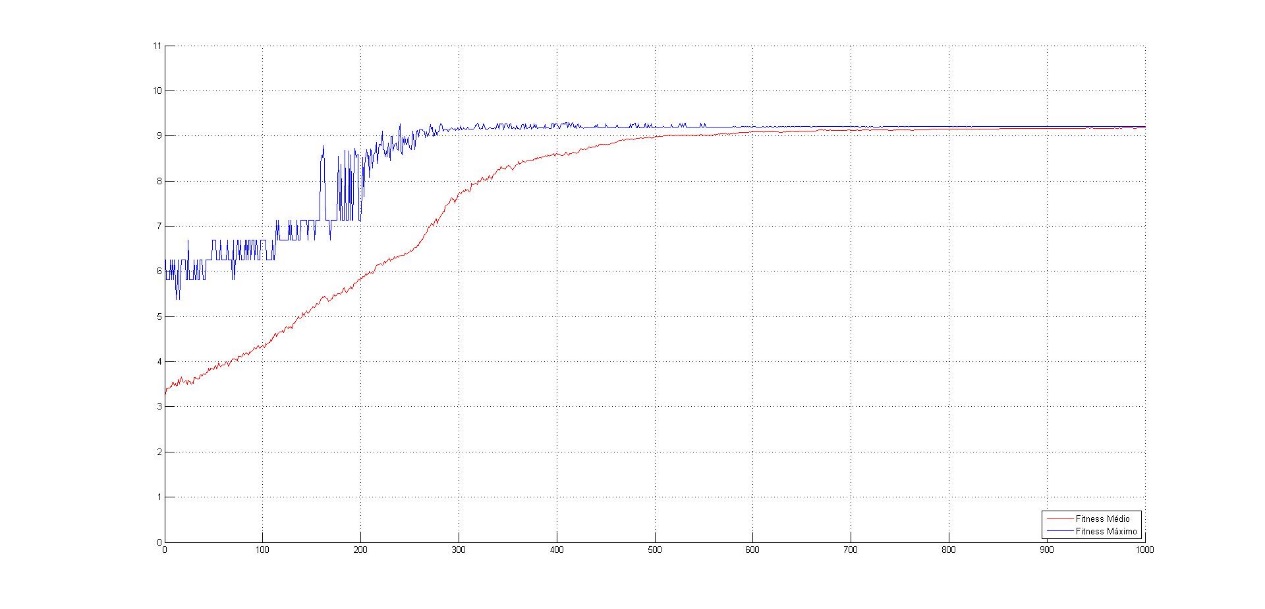


Figura 37

Figura 5.20 – Evolução da aptidão para o teste 5.

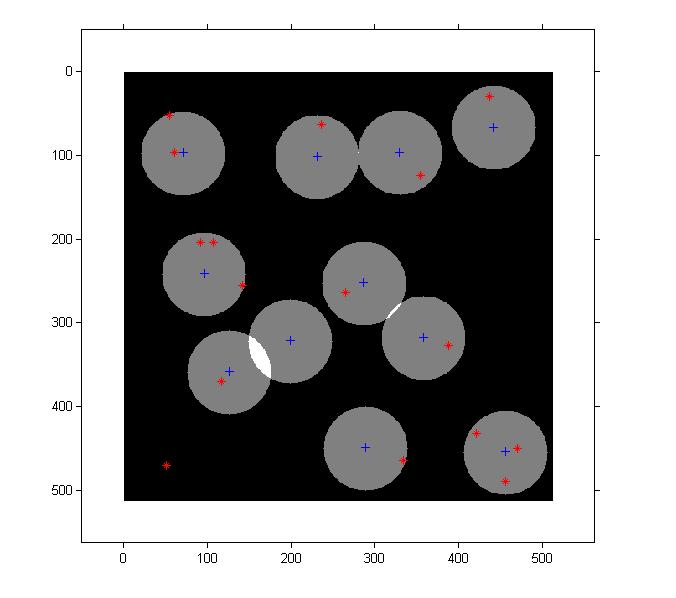


Figura 38

Figura 5.21 - Figura 5.9 – Mapa de sensores cobertos para o teste 6.

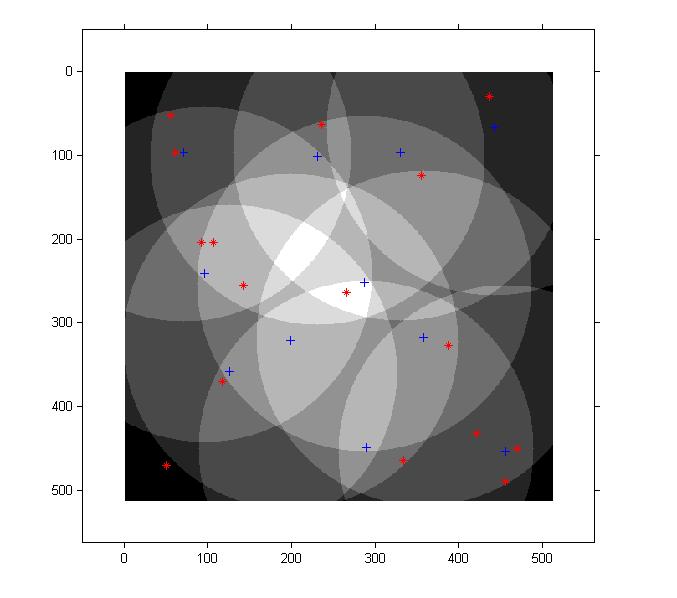


Figura 39

Figura 5.22 - Mapa para visualização de comunicação entre roteadores, considerando para o teste 6.

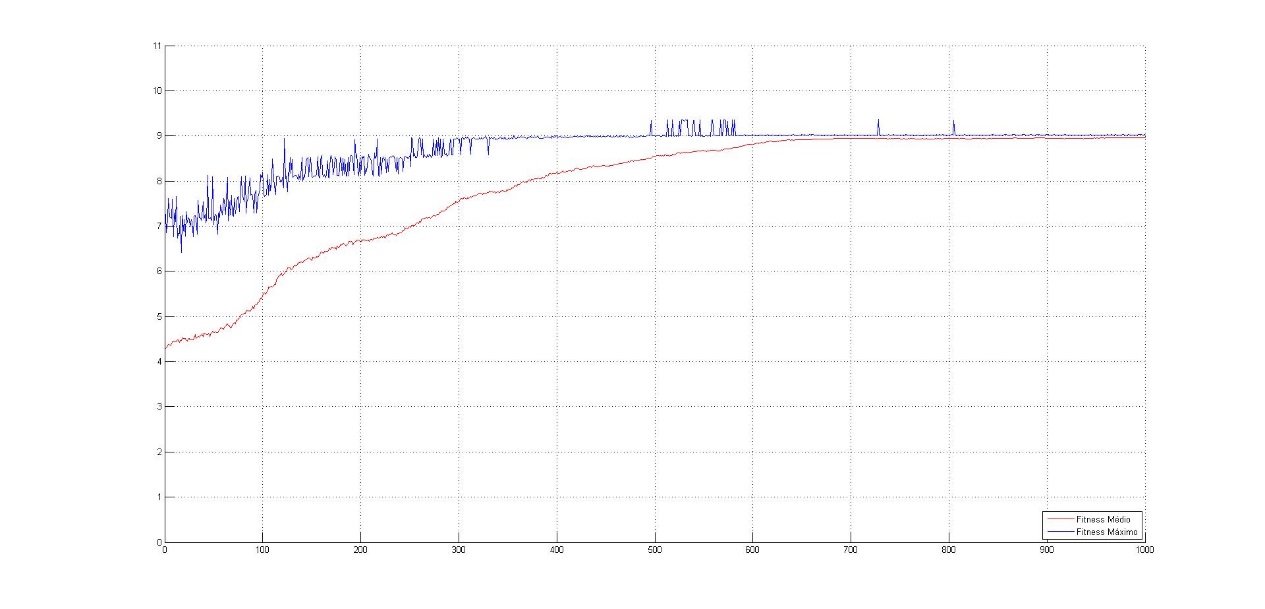


Figura 40

Figura 5.23 – Evolução da aptidão para o teste 6.

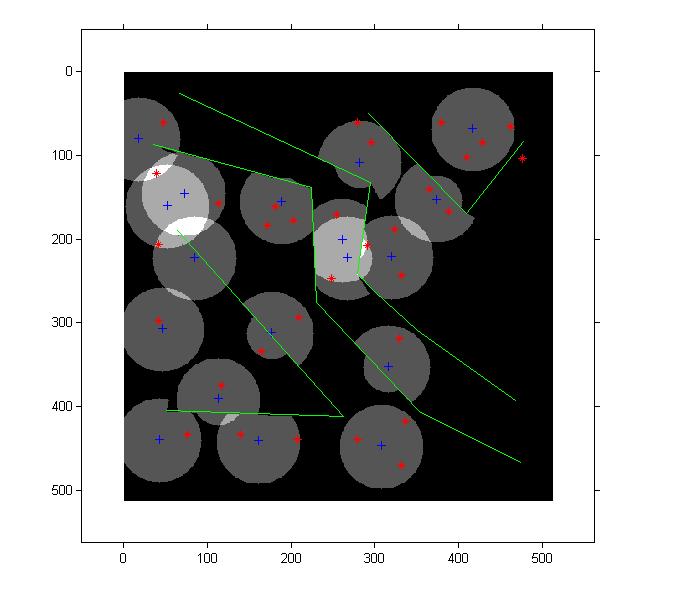


Figura 41

Figura 5.24 - Mapa de sensores cobertos para o teste 7.

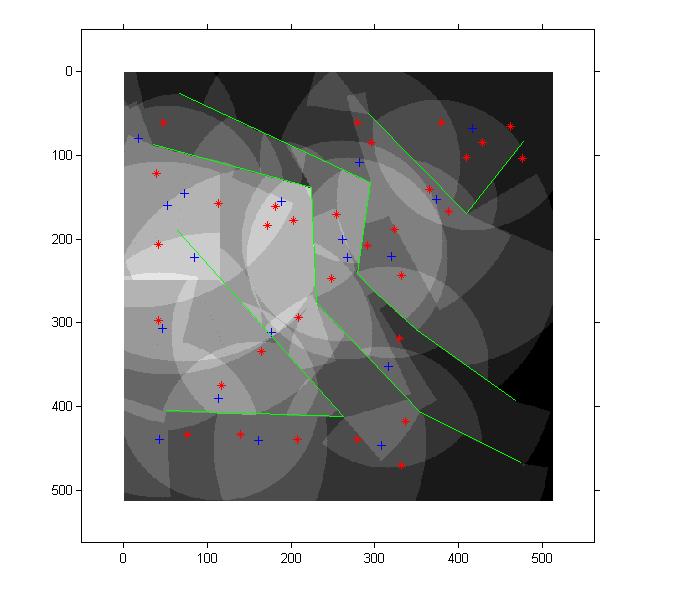


Figura 42

Figura 5.25 - Mapa para visualização de comunicação entre roteadores, considerando para o teste 7.

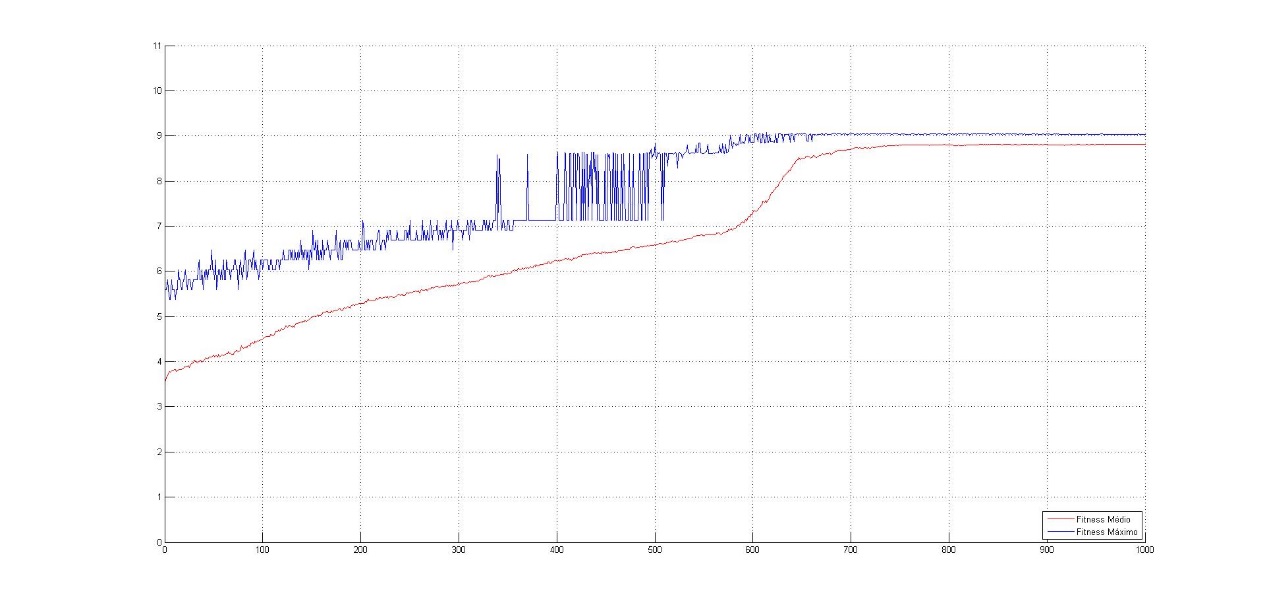


Figura 43

Figura 5.26 – Evolução da aptidão para o teste 7.

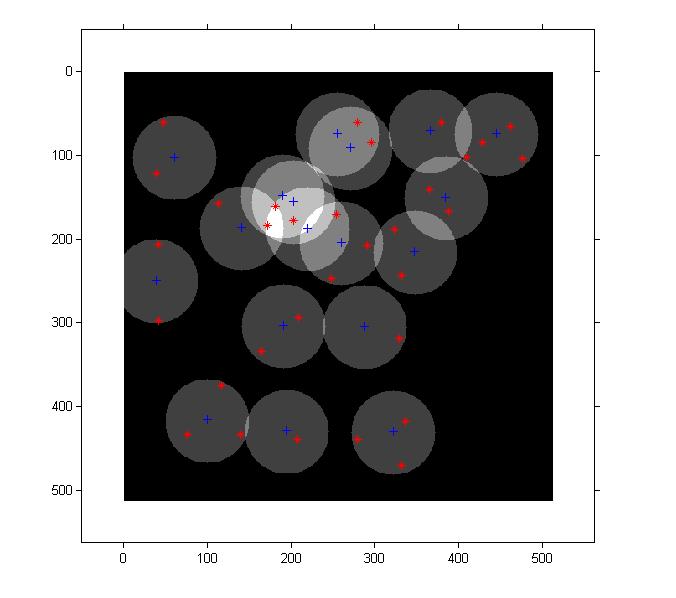


Figura 44

Figura 5.27 - Mapa de sensores cobertos para o teste 8.

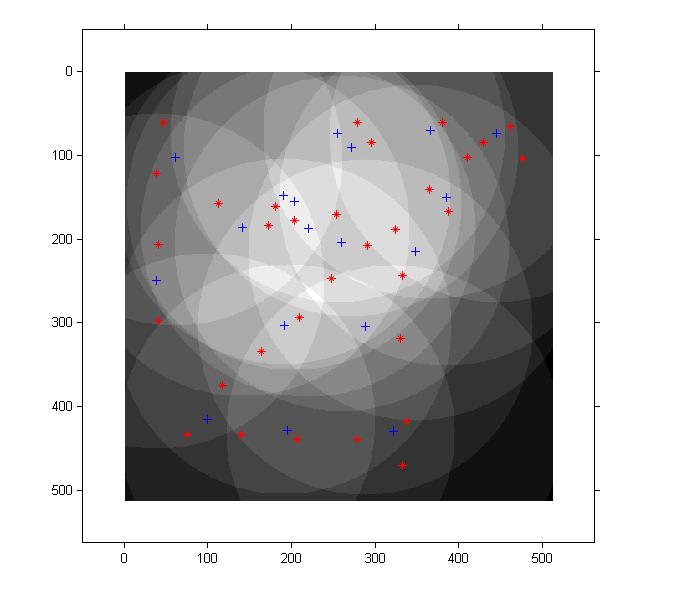


Figura 45

Figura 5.28 - Mapa para visualização de comunicação entre roteadores, considerando para o teste 8.

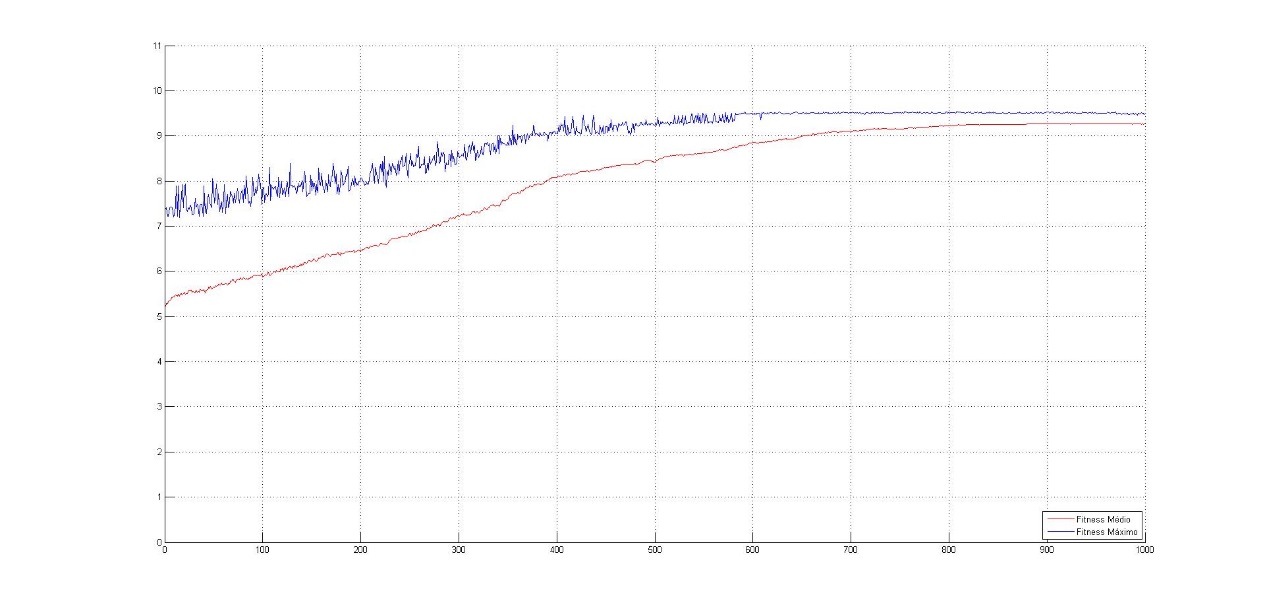


Figura 46

Figura 5.29 – Evolução da aptidão para o teste 8.

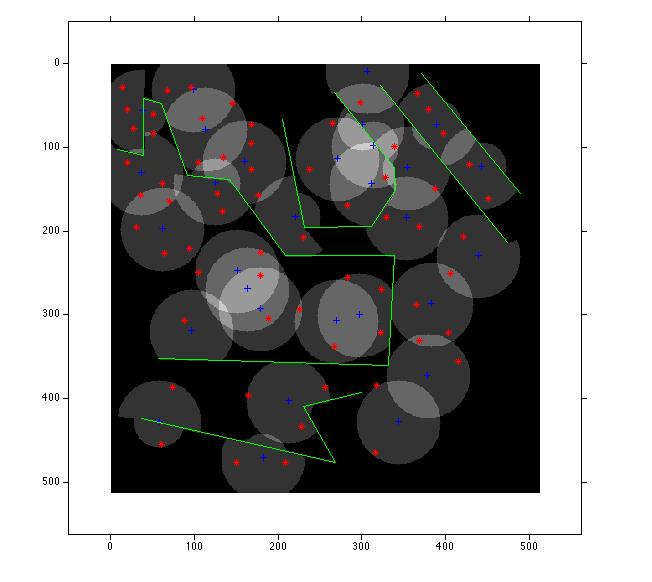


Figura 47

Figura 5.33 - Mapa de sensores cobertos para o teste 9.

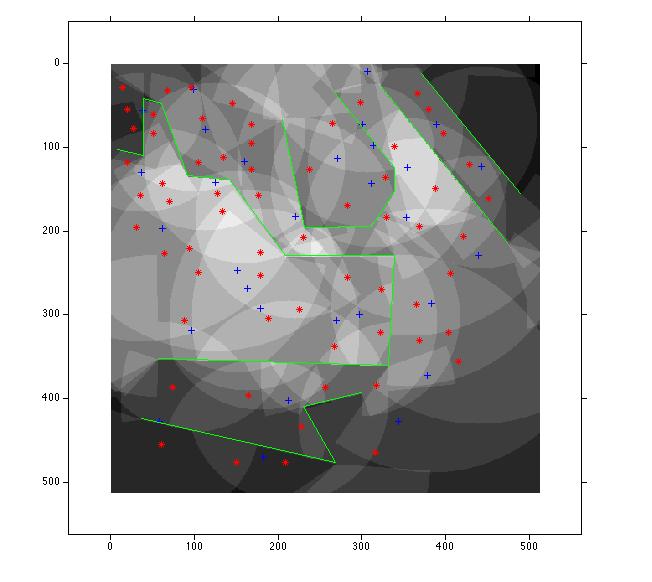


Figura 48

Figura 5.34 - Mapa para visualização de comunicação entre roteadores, considerando para o teste 9.

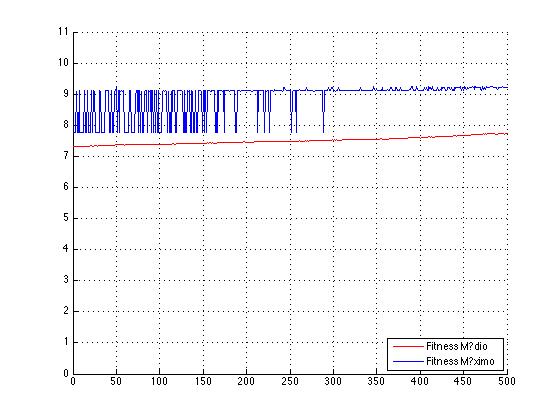


Figura 49

Figura 5.35 – Evolução da aptidão para o teste 9.

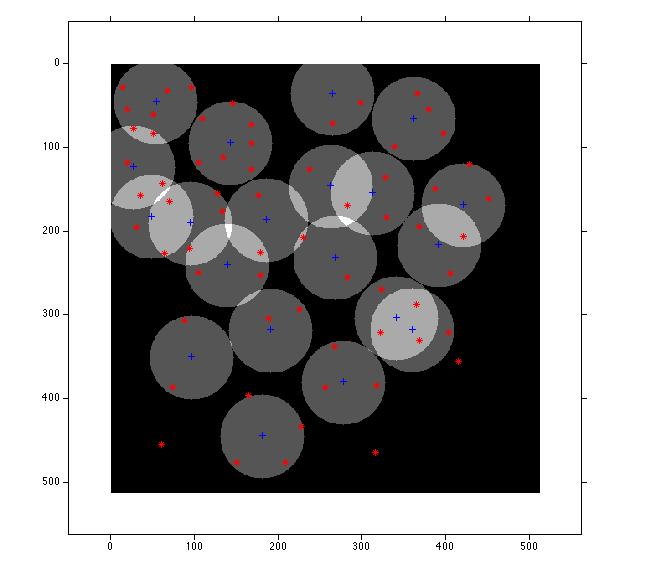


Figura 50

Figura 5.36 - Mapa de sensores cobertos para o teste 10.

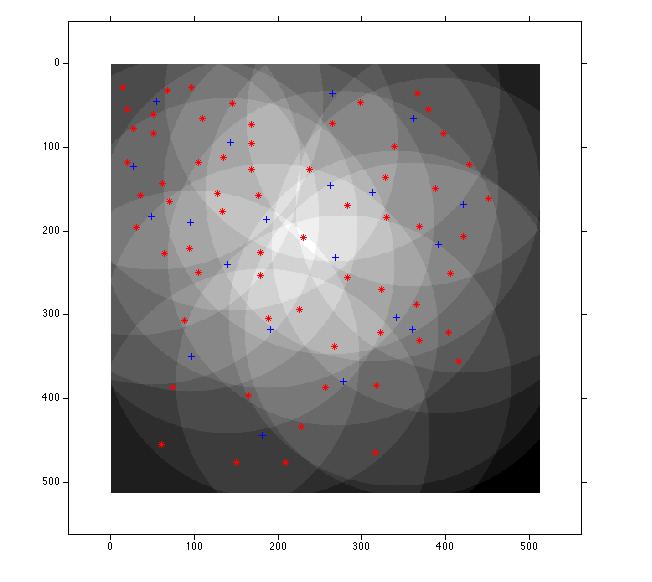


Figura 51

Figura 5.37 - Mapa para visualização de comunicação entre roteadores, considerando para o teste 10.

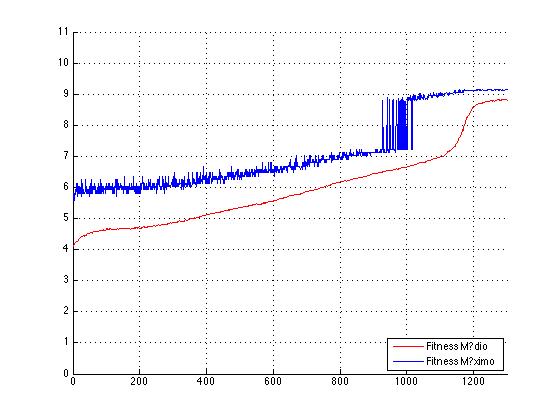


Figura 52

Figura 5.38 – Evolução da aptidão para o teste 10.

* + 1. Busca Aleatória

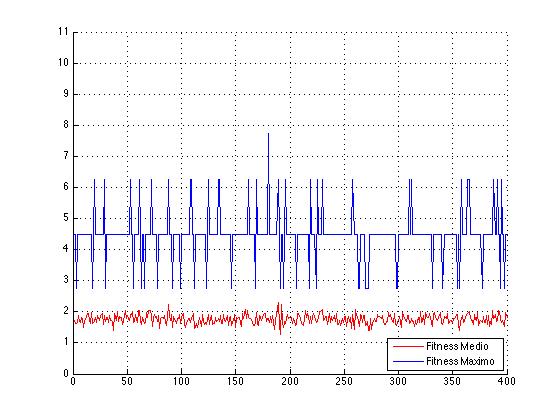


Figura 53

Figura 5.38 – Evolução da aptidão para o teste 1.

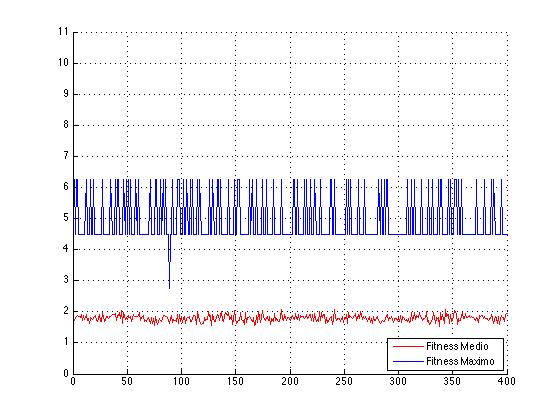


Figura 54

Figura 5.39 – Evolução da aptidão para o teste 2.

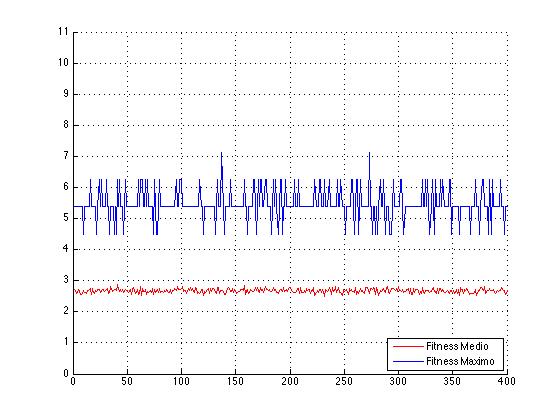


Figura 55

Figura 5.40 – Evolução da aptidão para o teste 3.

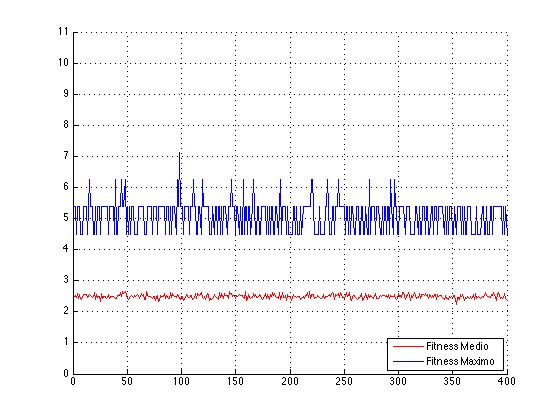


Figura 56

Figura 5.41 – Evolução da aptidão para o teste 4.

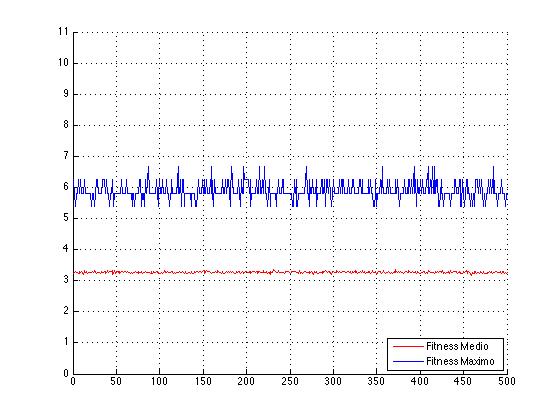


Figura 57

Figura 5.42 – Evolução da aptidão para o teste 5.

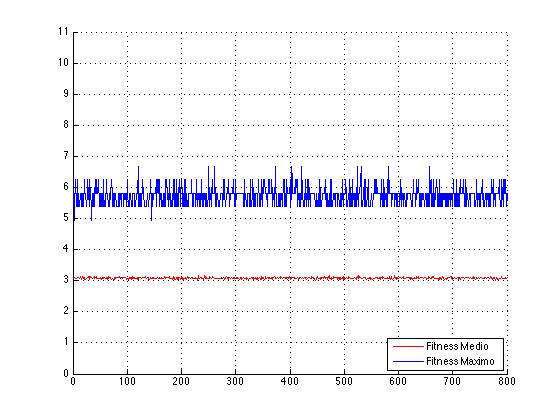


Figura 58

Figura 5.43 – Evolução da aptidão para o teste 6.

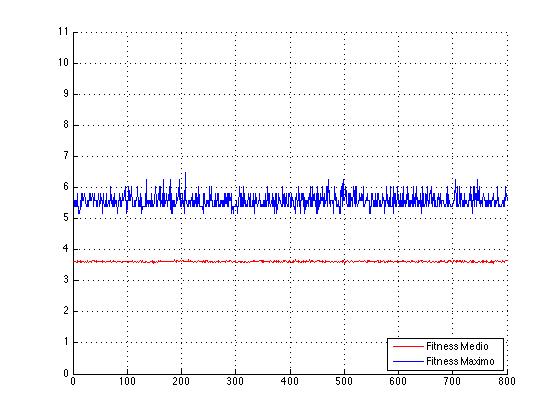


Figura 59

Figura 5.44 – Evolução da aptidão para o teste 7.

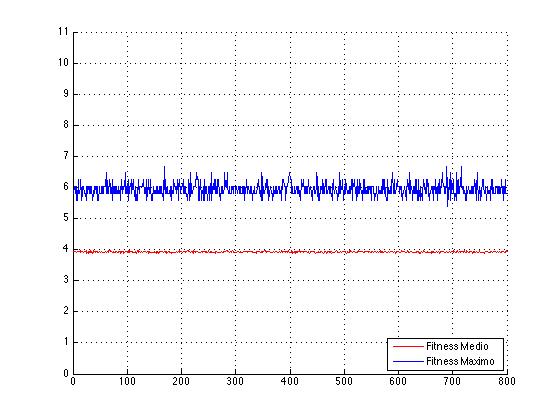


Figura 60

Figura 5.45 – Evolução da aptidão para o teste 8.

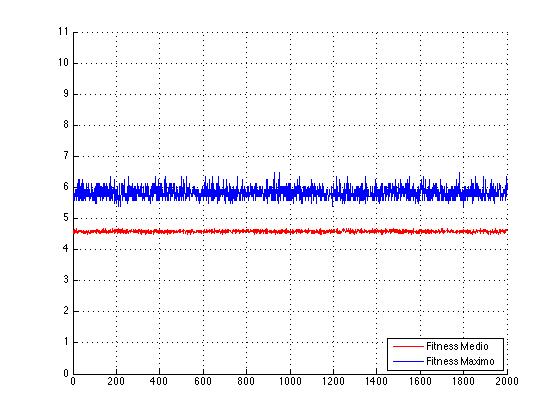


Figura 61

Figura 5.46 – Evolução da aptidão para o teste 9.

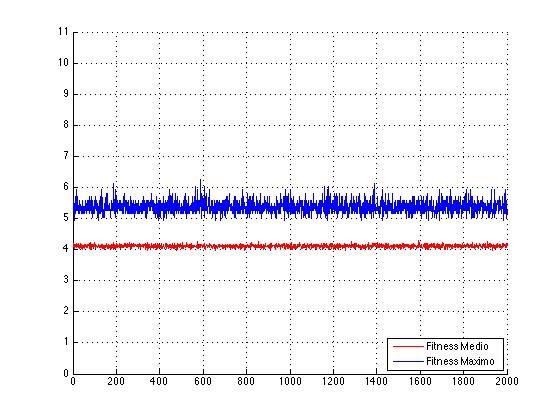


Figura 62

Figura 5.47 – Evolução da aptidão para o teste 10.

Percebe-se, como já foi dito no capítulo 4, que não há um direcionamento, evolução ou melhora das aptidão. O surgimento de um indivíduo mais apto está condicionado a um evento probabilístico.

* + 1. Busca Gulosa

|  |  |  |
| --- | --- | --- |
| **Teste** | **Roteadores** | **Maior aptidão** |
| 1 | 4 | 8,3125 |
| 2 | 4 | 9,5 |
| 3 | 8 | 8,7344 |
| 4 | 8 | 9,2969 |
| 5 | 15 | 8,4756 |
| 6 | 12 | 8,5556 |
| 7 | 18 | 7,1813 |
| 8 | 18 | 9,0185 |
| 9 | 26 | 6,8531 |
| 10 | 20 | 7,2606 |

Tabela x – Resultados do Algoritmo de Busca Gulosa

1. **Conclusão**

Este Trabalho de Graduação teve por objetivo obter uma solução eficiente para o posicionamento, em cenário com obstáculos, de roteadores em uma rede de sensores sem fio que garanta a cobertura dos sensores e conectividade entre os roteadores.

O Algoritmo Genético desenvolvido foi capaz de posicionar roteadores de maneira eficiente. A rede sem fio resultante foi, em todos os casos, coberta e conexa. Secundariamente, mostrou-se que algoritmos baseados em heurísticas aleatórias não dirigidas ou gulosas não são capazes de realizar esta tarefa.

Os autores deste trabalho acreditam que Algoritmos Genéticos possam ser amplamente utilizados no posicionamento de dispositivos em redes sem fio.

Uma vez não há cálculo automático do número de iterações ou indivíduos, recomenda-se o ajuste manual das variáveis do Algoritmo Genético para exemplos específicos diferentes dos tratados neste trabalho.

Os autores deste trabalho sugerem o uso de computação paralela para a execução de cenários ainda mais complexo em menor tempo e testando um maior número de casos.

1. **Referencias**

Holland, J., Adaptation in Natural and Artificial Systems, Ann Arbor, MI, University of Michigan Press, 1975.

Goldberg, D.E. (1989b). Sizing Populations for Serial and Parallel Genetic Algorithms, Proceedings of the Third International Conference on Genetic Algorithms. San Mateo, CA: Morgan Kaufman. pp. 70-79

Rylander, B., Foster, J., GA-hard Problems, Proc. On Genetic and Evolutionary Computation Conference, 2000.

Gotshall, S., Rylander, B., Optimal Population Size and the Genetic Algorithm, University of Portland.

Leu, Y.-Y., Matheson, L. A. and Rees, L. P. (1994), Assembly Line Balancing Using Genetic Algorithms with Heuristic-Generated Initial Populations and Multiple Evaluation Criteria. Decision Sciences, 25: 581–605. doi: 10.1111/j.1540-5915.1994.tb00822.x

Bäck, Thomas, Optimal Mutation Rates in Genetic Search,Department of Computer Science.University of Dortmund.

Junfang Wang; Bin Xie; Kan Cai; Agrawal, D.P., "Efficient Mesh Router Placement in Wireless Mesh Networks," *Mobile Adhoc and Sensor Systems, 2007. MASS 2007. IEEE Internatonal Conference on* , vol., no., pp.1,9, 8-11 Oct. 2007  
doi: 10.1109/MOBHOC.2007.4428616

|  |  |  |  |
| --- | --- | --- | --- |
| FOLHA DE REGISTRO DO DOCUMENTO | | | |
| 1. CLASSIFICAÇÃO/TIPO  TC | 2. DATA | 3. REGISTRO N° | 4. N° DE PÁGINAS  XX |
| 5. TÍTULO E SUBTÍTULO:  Estudo Sobre o Desenvolvimento de Trabalhos Acadêmicos em Cima da Hora | | | |
| 6. AUTOR(ES):  Fulano de Tal | | | |
| 1. INSTITUIÇÃO(ÕES)/ÓRGÃO(S) INTERNO(S)/DIVISÃO(ÕES):   Instituto Tecnológico de Aeronáutica – ITA | | | |
| 8. PALAVRAS-CHAVE SUGERIDAS PELO AUTOR: | | | |
| 9.PALAVRAS-CHAVE RESULTANTES DE INDEXAÇÃO: | | | |
| 10. APRESENTAÇÃO:  **X Nacional Internacional**  ITA, São José dos Campos. Curso de Graduação em Engenharia de Computação. Orientador: Chuck Norris. Publicado em 2010. | | | |
| 11. RESUMO: | | | |
| 12. GRAU DE SIGILO:  **(X ) OSTENSIVO ( ) RESERVADO ( ) CONFIDENCIAL ( ) SECRETO** | | | |