|  |
| --- |
| INSTITUTO TECNOLÓGICO DE AERONÁUTICA **..\..\..\HD_D\Producao\logotipos\ita2.gif**  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  Trabalho de Graduação  *2013*  Computação |

Número da CDU (tamanho 10)

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

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**

**Abstract**

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

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

Lista de Abreviaturas, Siglas e Símbolos

Sumário

[**1.** **Introdução** 14](#_Toc372608830)

[1.1. Motivação 14](#_Toc372608831)

[1.2. Objetivo 15](#_Toc372608832)

[1.3. Estrutura do Trabalho 16](#_Toc372608833)

[**2.** **Problema Abordado** 16](#_Toc372608834)

[2.1. Planta 16](#_Toc372608835)

[2.2. Sensores 18](#_Toc372608836)

[2.3. Roteadores 19](#_Toc372608837)

[2.4. Obstáculos 20](#_Toc372608838)

[2.5. Espaço de Busca 23](#_Toc372608839)

[**3.** **Algoritmo Genético** 24](#_Toc372608840)

[3.1. Introdução 24](#_Toc372608841)

[3.2. População Inicial 25](#_Toc372608842)

[3.3. Função de Aptidão 25](#_Toc372608843)

[3.3.1. Apresentação 25](#_Toc372608844)

[3.3.2. Cálculo 25](#_Toc372608845)

[3.3.3. Desempenho 27](#_Toc372608846)

[3.4. Escolha dos mais aptos 28](#_Toc372608847)

[3.5. Função de Cruzamento 29](#_Toc372608848)

[3.6. Mutação 29](#_Toc372608849)

[3.7. Tamanho da população 30](#_Toc372608850)

[3.8. Número de iterações 30](#_Toc372608851)

[3.9. Limitações 30](#_Toc372608852)

[**4.** **Algoritmos Alternativos** 33](#_Toc372608853)

[4.1. Introdução 33](#_Toc372608854)

[4.2. Algoritmo de Busca Aleatório 33](#_Toc372608855)

[4.3. Algoritmo de Busca Gulosa 33](#_Toc372608856)

[**5.** **Resultados** 34](#_Toc372608857)

[5.1. Variáveis de ambiente 34](#_Toc372608858)

[5.2. Cenários de testes 34](#_Toc372608859)

[5.3. Algoritmo Genético 39](#_Toc372608860)

[5.4. Busca Aleatória 62](#_Toc372608861)

[5.5. Busca Gulosa 62](#_Toc372608862)

[6. Conclusão 63](#_Toc372608863)

[7. Referencias 63](#_Toc372608864)

[8. Apêndice 64](#_Toc372608865)

1. **Introdução**

Este trabalho tem por finalidade aplicar o conceito de Algoritmos Genéticos(AG) em um problema real de cobertura de sensores em uma rede sem fio de forma a gerar uma solução de roteamento barata e mais eficiente possível de acordo com os objetivos de otimização inseridos.

O completo desenvolvimento deste Trabalho de Graduação exigiu sólido conhecimento das disciplinas aprendidas no curo de Engenharia de Computação e conhecimento multidisciplinar. Primeiramente, foi necessário conhecer detalhadamente o cenário da problemática tratada. Em seguida, era contingencial definir o algoritmo que a resolvesse. Finalmente, a resposta gerada pelo software desenvolvido deveria ser analisada e criticada.

Para entender o problema era preciso modelar os efeitos dos obstáculos na área de cobertura de um roteador sem fio. Um modelo simples baseado apenas na atenuação do sinal foi usado. Fatores como reflexões e interferências não foram considerados neste trabalho.

O Algoritmo escolhido deveria ser capaz de otimizar a cobertura de pontos definidos do mapa e, ao mesmo tempo, prover um alto grau de conexão entre os roteadores de forma que, idealmente, todos se encontrem em uma componente conexa.

Os resultados encontrados foram analisados à luz do valor de aptidão da solução. Este valor é definido por uma função que, ao mesmo tempo, precisa ser rápida para não comprometer a performance e complexa para definir claramente os anseios com relação a saída desejada ao final da execução.

* 1. Motivação

Os Algoritmos Genéticos estão em alta no mundo da computação. Uma rápida busca no site *Scholar.Google.com* pelo termo “*Genetic Algorithms”* revela cerca de 1,8 milhão de resultados. Aproximadamente o mesmo número que “*Artificial Intelligence”* e “*Wireless Networks”*.

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).

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.

Indo em direção ao campo das redes sem fio. Até fevereiro de 2013, cerca de 1,5 bilhão de *smartphones* foram vendidos mundialmente (Business Insider, 2013). Atualmente, há vários tipos e usos para dispositivos sem fio. Ademais, existem iniciativas públicas e privadas ao redor do mundo para a instalação de redes sem fio em cidades inteiras.

* 1. Objetivo

Tendo em vista que o trabalho desenvolvido é baseado na criação de um programa de computador, os objetivos aqui listados se parecem com requisitos funcionais e não-funcionais de um software.

O objetivo deste trabalho é obter uma solução eficiente para a colocação de 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.

Como regra de conduta, tentou-se aplicar ao máximo as técnicas aprendidas durante os 5 anos do curso de Engenharia de Computação do ITA. Boas práticas de programação são sempre bem vindas no desenvolvimento de software, mesmo em ambiente acadêmico. Procurou-se utilizar conceitos de Engenharia de software para minimizar a ocorrência de código duplicado e nomes inapropriados de variáveis e funções. Fez se uso do sistema de versionamento *GitHub*, uma das ferramentas de auxílio ao desenvolvimento de software mais famosas do mundo.

* 1. Estrutura do Trabalho

Primeiramente, 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**
   1. Planta

A planta é constituída por um mapa quadrado de lado . *Grain* é uma variável interna do programa que define o grau de refinamento dos cromossomos. Qualquer entidade, seja ela sensor, roteador ou obstáculo terá suas propriedades expressas por cadeias de bits de tamanho *grain*, i.e, números inteiros variando no intervalo .



Figura 2.1 – 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

Os sensores são representados por pequenos símbolos vermelhos na planta. A colocação pelo usuário é manual, mas há a possibilidade de se passar uma lista de sensores pré-colocados.

O significado destas entidades pode ser abstraído como sendo pontos em uma área que precisa ser coberta, estações de trabalho ou sensores remotos para aquisição de dados.

Por simplificação, optou-se por garantir cobertura apenas no ponto, i.e, os sensores são estáticos. Casos com sensores se movimentando em uma dada região não foram abordados neste trabalho.

O alcance da comunicação em visada direta entre roteador e sensor é dada pela variável .

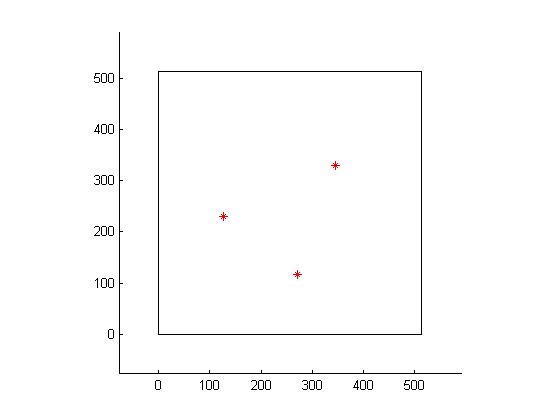


Figura 2.2 – Representação dos Sensores na planta.

* 1. Roteadores

Os roteadores são representados por símbolos azuis na planta. O posicionamento destas entidades é, em suma, o *output* do programa.

O alcance da comunicação em visada direta entre dois roteadores é dado pela variável .

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.

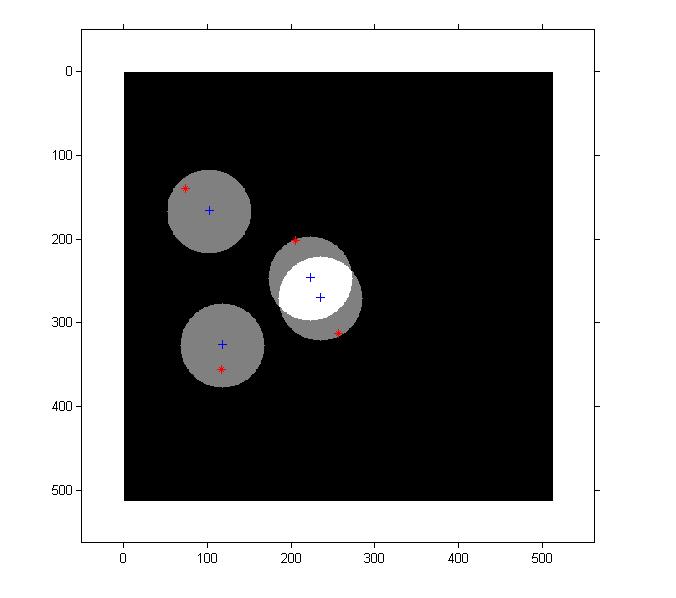


Figura 2.3 – Quatro Roteadores(Azul) cobrindo 4 sensores em um exemplo sem obstáculos e com .

* 1. Obstáculos

Os obstáculos são representados por segmentos de reta na planta. Pretos na planta de entrada e verdes na planta de visualização da resposta.

Por simplificação, existe somente obstáculos retos. Outras formas não são suportadas.

Os obstáculos são responsáveis pela atenuação do sinal. O grau de atenuação é dado pelo valor da variável . Este nome refere-se a equação de raio do sinal após passar por 1 obstáculo.

Onde e representam, respectivamente, os raios antes e depois de atravessar a barreira. Para casos de atenuação do sinal, . Se não há atenuação e para haveria amplificação no sinal.

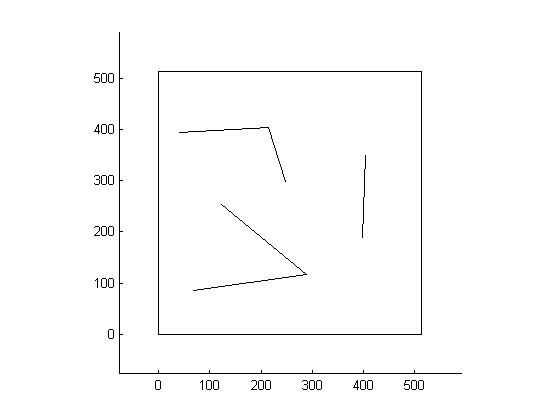


Figura 2.4 – Obstáculos

A inserção de possíveis barreiras no problema de posicionamento cria uma complexidade extra. As Fig. 2.5 e 2.6 mostram cenários com a inclusão um número crescente de barreiras ao problema.

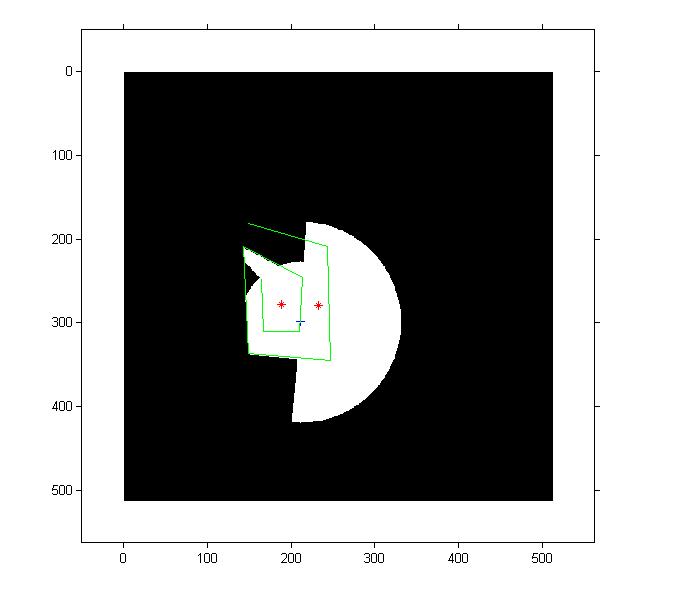


Figura 2.5 – Um Roteador(Azul) cobrindo 2 sensores em um exemplo com obstáculos, , .

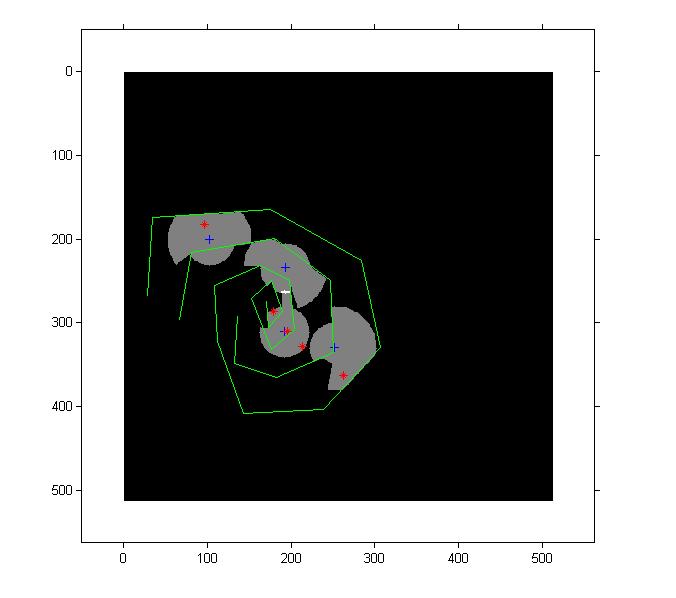


Figura 2.6 – Quatro Roteadores cobrindo 5 sensores em um exemplo com obstáculos, , .

Na Fig. 2.5, 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. 2.6, o grande número de barreiras e sensores torna o processo bem complexo. Ela mostram o quão difícil é a análise do posicionamento quando muitos obstáculos são incorporados ao mapa.

Ainda na Fig. 2.6 é possível ver um roteador que não cobre sensor algum. No entanto, na Fig. 2.7 fica evidente a sua função de ponte para construir uma configuração de roteadores interconectados, i.e, uma componente conexa de tamanho igual ao número de roteadores.

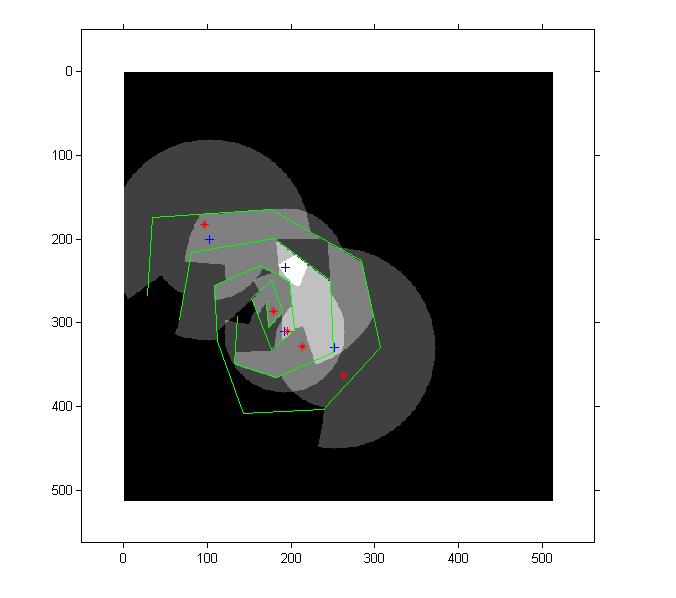


Figura 2.7 – Quatro Roteadores formando uma única componente conexa em um exemplo com obstáculos, , .

* 1. Espaço de Busca

Toda tarefa de busca e otimização possui vários componentes, entre eles: o espaço de busca, onde são consideradas todas as possibilidades de solução de um determinado problema e a função de avaliação (ou função de custo), uma maneira de avaliar os membros do espaço de busca. Existem muitos métodos de busca e funções de avaliação.

O problema estudado nesse trabalho pertence à classe de problemas NP-Difícil (Mohamed Younis, 2007).

O Tamanho do espaço de busca é dado por:

A seguir encontram-se alguns resultados da ordem de grandeza para uma determinada faixa de valores de e número de roteadores.

|  |  |  |
| --- | --- | --- |
|  | Número de Roteadores | Espaço de busca |
| 8 | 10 |  |
| 9 | 20 |  |
| 10 | 30 |  |

Tabela 2.1 – Tamanho do espaço de busca para diferentes configurações de planta.

Estes resultados demonstram a ineficiência de qualquer método de busca baseado em analisar todas as possíveis soluções.

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**
   1. Introdução

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.

As técnicas de busca e otimização tradicionais iniciam-se com um único candidato que, iterativamente, é manipulado utilizando algumas heurísticas (estáticas) diretamente associadas ao problema a ser solucionado. Geralmente, estes processos heurísticos não são algorítmicos e sua simulação em computadores pode ser muito complexa. Apesar destes métodos não serem suficientemente robustos, isto não implica que eles sejam inúteis. Na prática, eles são amplamente utilizados, com sucesso, em inúmeras aplicações.

Por outro lado, as 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. Indivíduos

Um indivíduo representa um conjunto de dados candidato a solução do problema. No escopo deste trabalho, o indivíduo é composto por um vetor de tamanho 2 vezes o número de 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.

* 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 irá conter um cromossomo 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
     1. Apresentação

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

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 3.1.

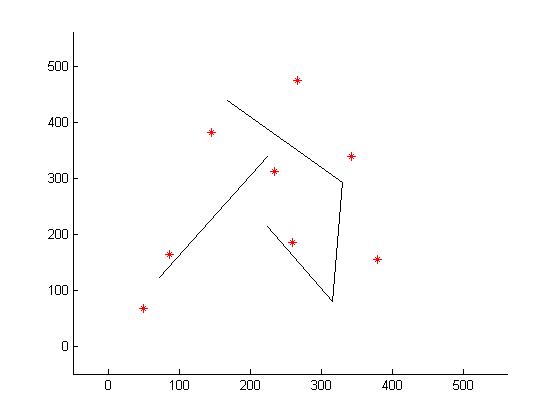


Figura 3.1 – Exemplo com 8 sensores para teste de velocidade.

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

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 se escolhido para reproduzir.

Figura 3.2 – 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 de um ou mais cortes na cadeia de bits dos pais e a junção cruzada dos pedaços cortados (Fig. 3.3).

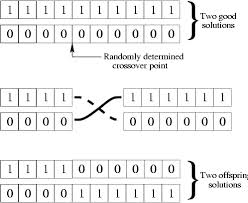


Figura 3.3 – Cruzamento por corte único.

O número de cortes em um indivíduo determina o tipo de cruzamento. Há duas possibilidades: Corte único e multicorte.

Neste trabalho, cada cromossomo é cortado uma única vez. No entanto, cada indivíduo possui ao menos 2 cromossomos (quando apenas a solução com 1 roteador é testada). Logo, tem-se uma função de cruzamento multicorte.

* 1. Mutação

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

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

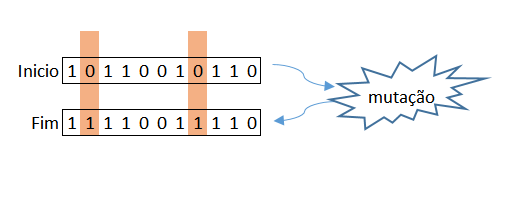


Figura 3.4 – Individuo sofrendo mutação de dois bits.

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

Nas Figs. 3.5 , 3.6 , 3.7 e 3.8 são mostrados os comportamentos dos valores máximo e médio da função de aptidão com o número de iterações para, respectivamente, probabilidades de mutação 50% ,5% , 0,5% e 0,1% por bit. O número de bits do indivíduo em questão é 162(9 roteadores).

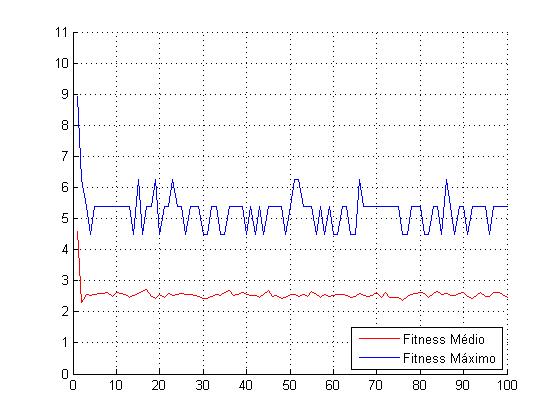


Figura 3.5 – Aptidão máxima e média para probabilidade de mutação igual a 50% por bit.

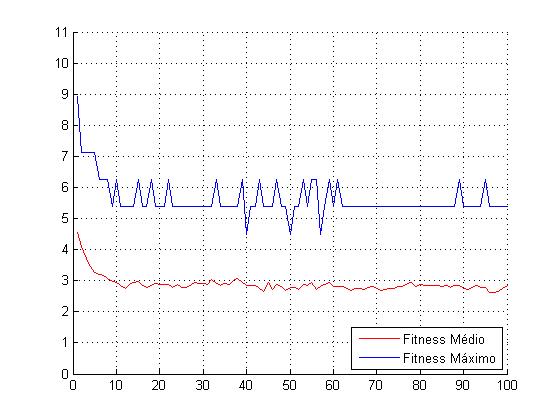


Figura 3.6 – Aptidão máxima e média para probabilidade de mutação igual a 5% por bit.

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

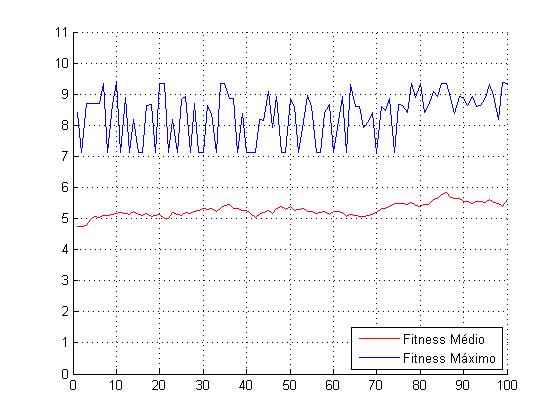


Figura 3.7 – Aptidão máxima e média para probabilidade de mutação igual a 0,5% por bit.

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

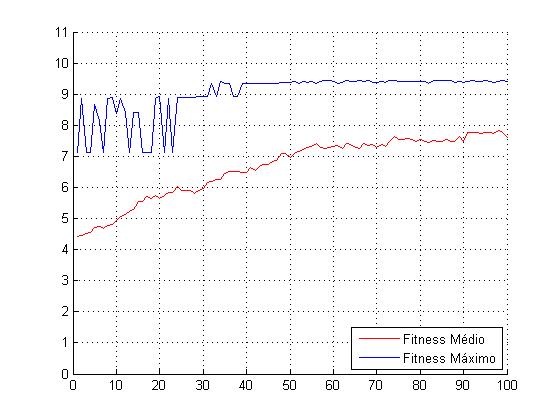


Figura 3.8 – Aptidão máxima e média para probabilidade de mutação igual a 0,1% por bit.

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 geneticos, 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 incorporado 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: Determinado valor de aptidão encontrado;

3: Convergência.

É possivel, 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).

Todavia, iterações excessivas podem levar o algoritmo a executar muito tempo sob condição de convergência alcançada e demorar muito mais que o necessário.

Cancelar a busca ao encontrar um indivíduo com determinado valor de aptidão diminui a necessidade de se efetuar cálculos desnecessários. Infelizmente, é necessário assegurar a qualidade desta entidade. No trabalho em questão não foi usada esta abordagem pois é impossível saber, dada uma planta mais complexa, o valor máximo global da função de aptidão.

Uma solução convergente para um Algoritmo Genético significa que as chances de se descobrir uma solução melhor são muito remotas a partir daquele ponto. Muito deste efeito se dá pela perda de diversidade da população e também influencias de máximos locais da função de aptidão.

Para este trabalho utilizou-se um número de iterações manual e critério de convergência igual a 0,3 comparando-se o melhor indivíduos de toda a execução com a média da iteração corrente.

* 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. 3.9, os sensores estão posicionados de maneira que a região que deve ser ocupada pelo roteador é muito maior do que a representada na Fig. 3.8.

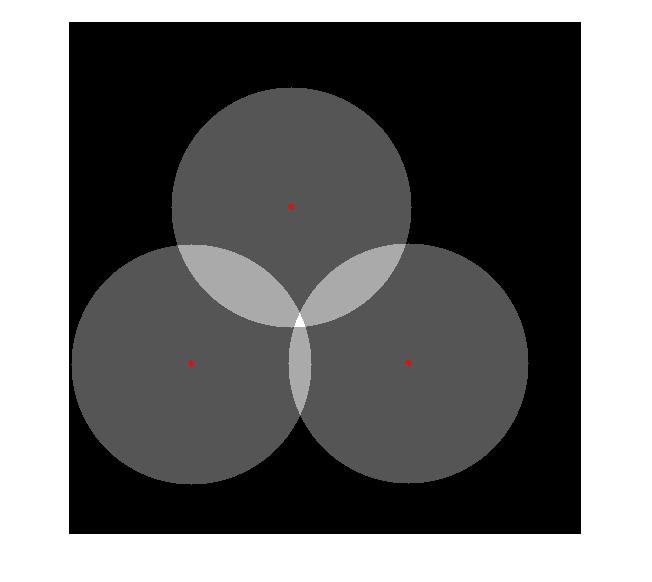


Figura 3.9 – Região de cobertura de três sensores em uma planta sem obstáculos.

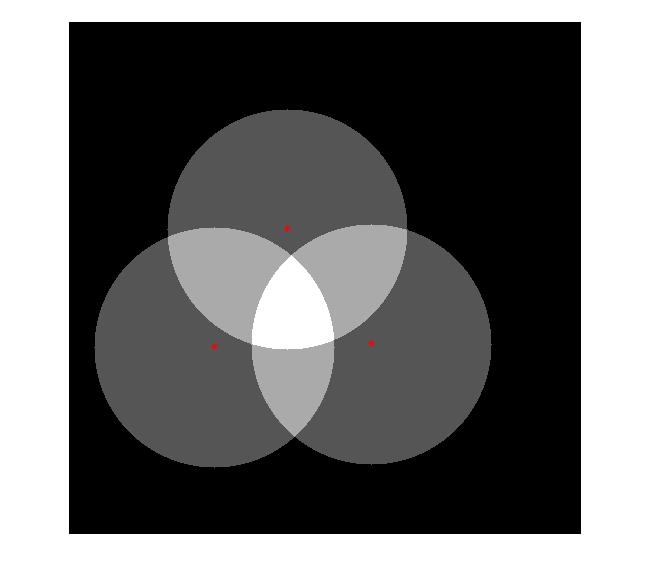


Figura 3.10 – Região de cobertura de três sensores em uma planta sem obstáculos. Agora um pouco mais próximos em relação a Fig. 3.5.

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**
   1. Introdução

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 sensores entrados 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**
   1. Variáveis de ambiente

As variáveis de ambiente são valores de entrada estáticos para 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 casos 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.

Cada par te testes é composto pela mesma dispersão de sensores. No entanto, os testes pares não possuem obstáculos.

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

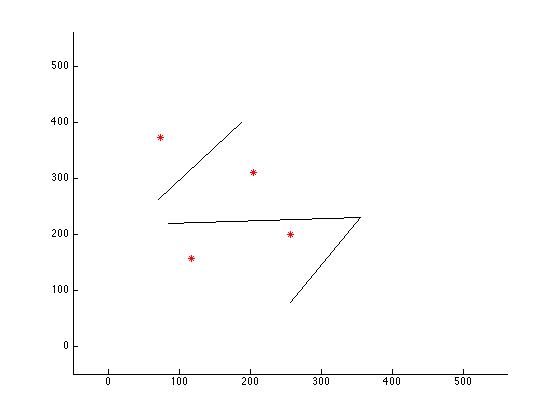


Figura 5.1 - Teste 1

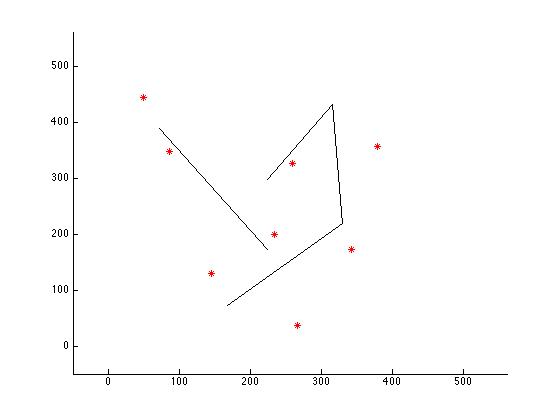


Figura 5.2 - Teste 3

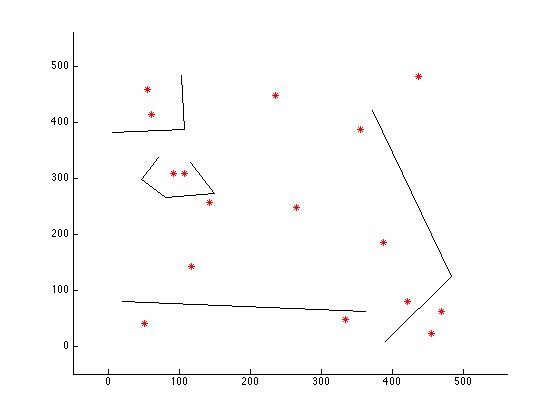


Figura 5.3 - Teste 5

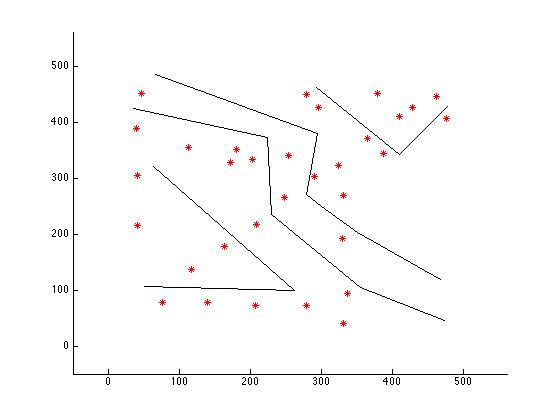


Figura 5.4 - Teste 7

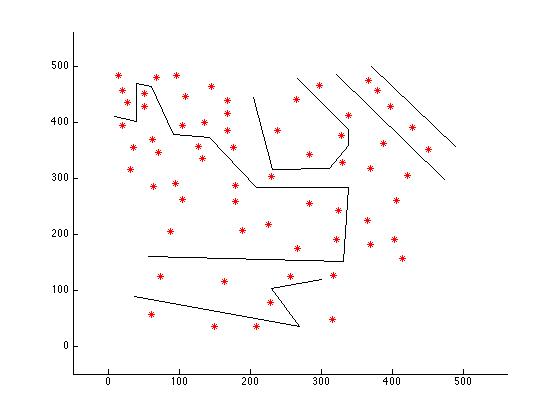


Figura 5.5 - Teste 9

* 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 figuras 5.6 a 5.38 a seguir apresentam os resultados dos testes 1 a 10.

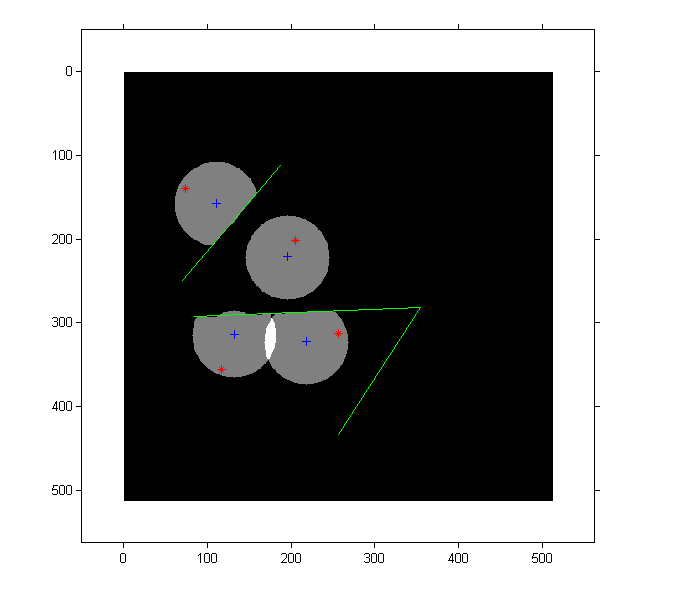
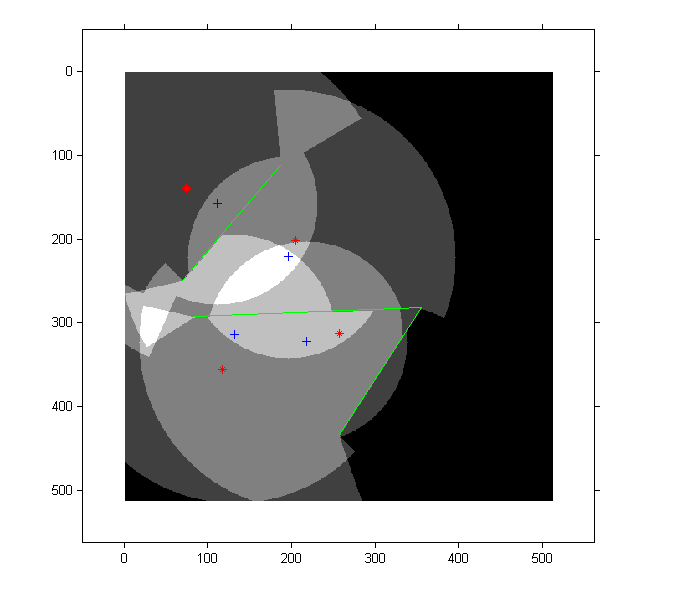
Figura 5.6 – Mapa de sensores cobertos para o teste 1.

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

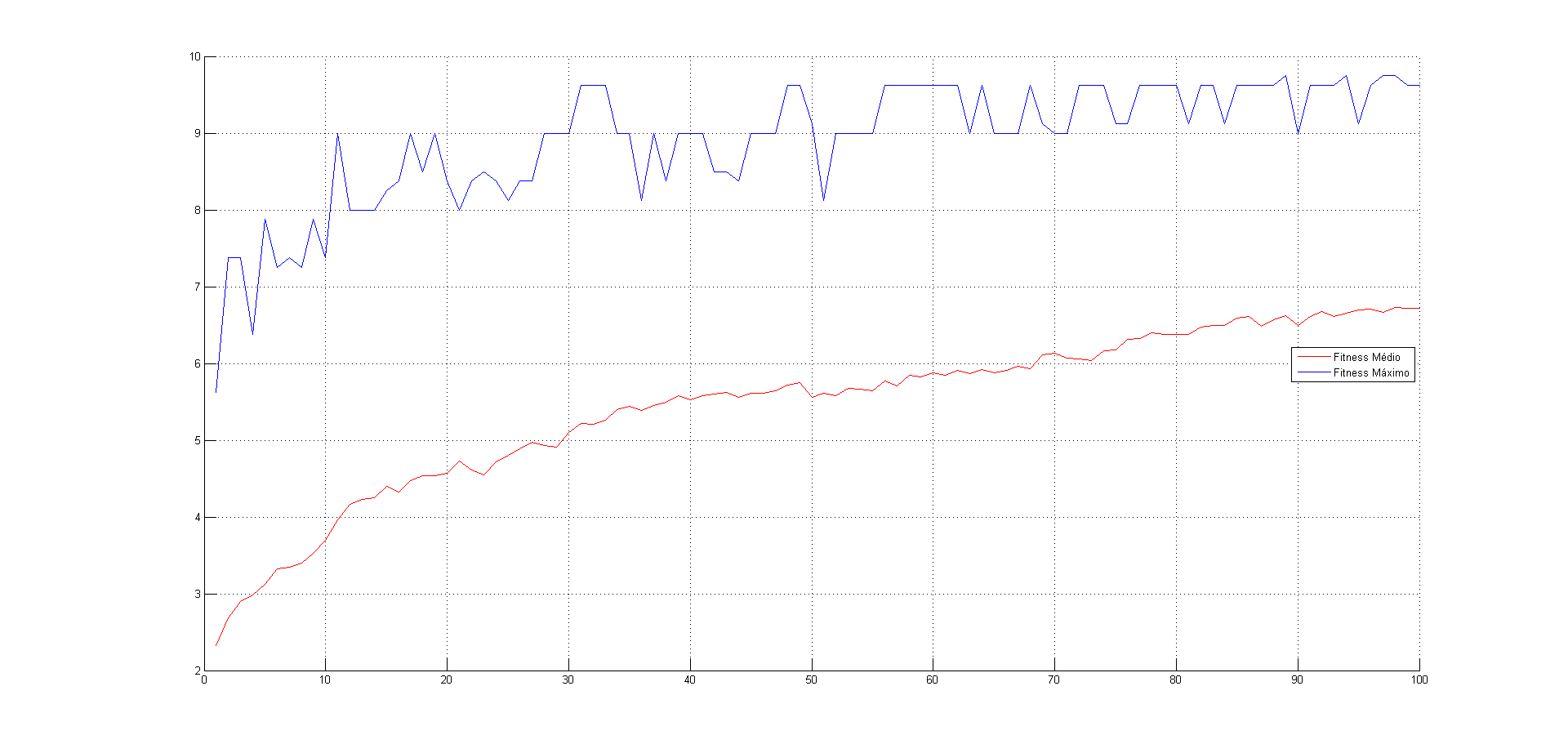
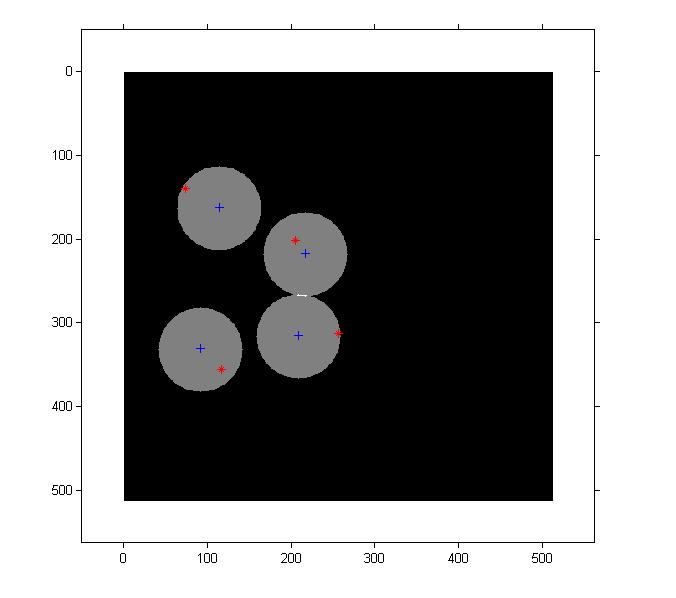


Figura 5.8 - Evolução da aptidão do teste 1.

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

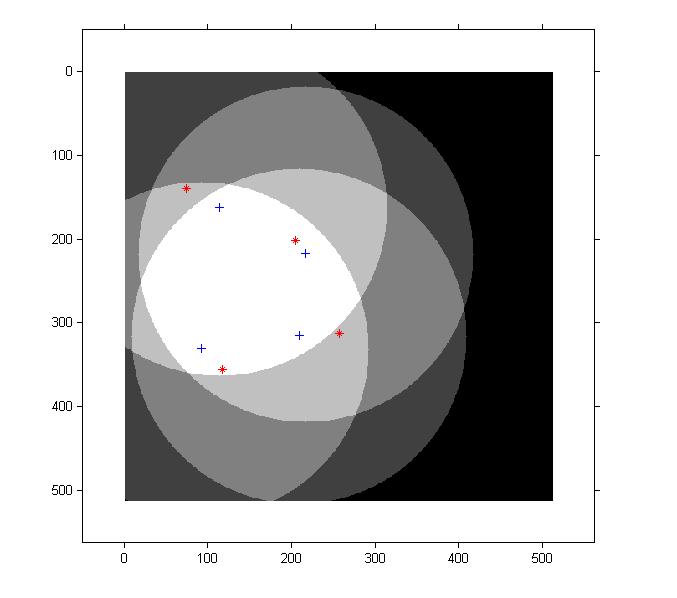


Figura 5.10 – Mapa para visualização de comunicação entre roteadores, considerando para o teste 2.

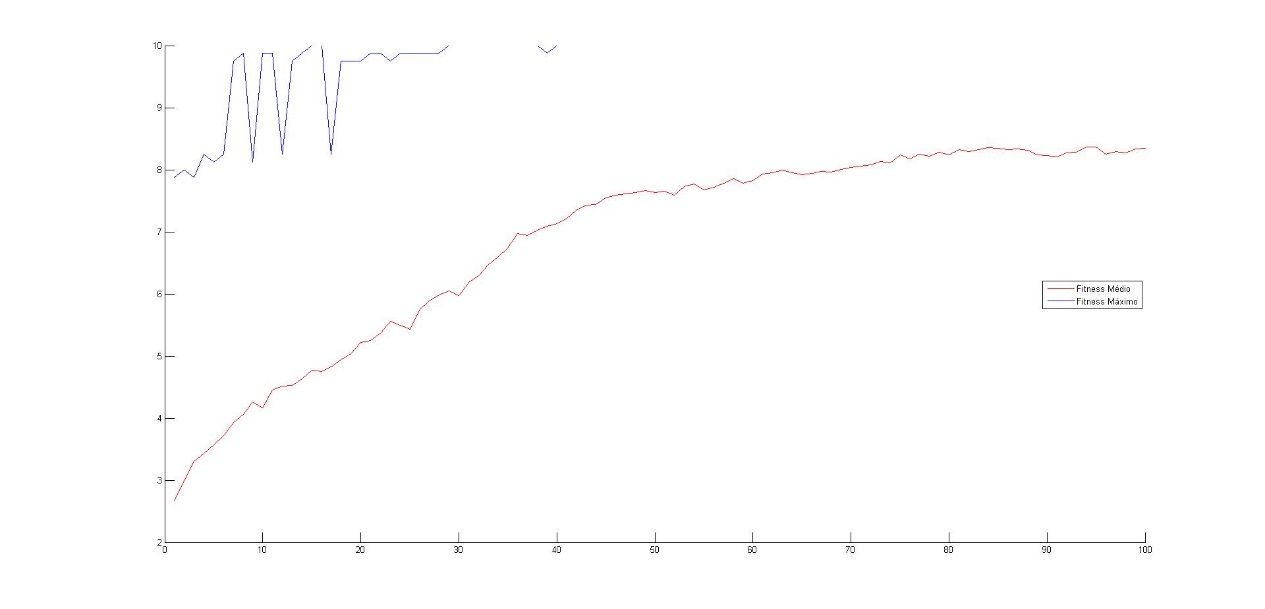


Figura 5.11 - Evolução da aptidão do teste 2.

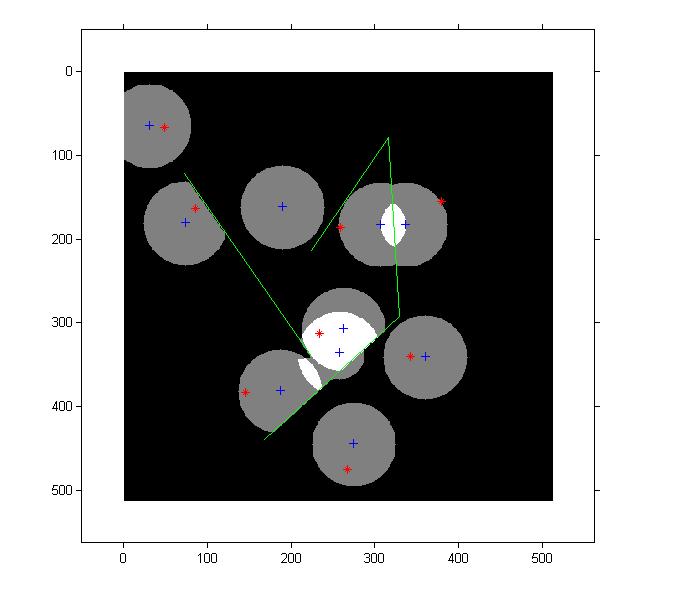


Figura 5.12 - Figura 5.9 – Mapa de sensores cobertos para o teste 3.

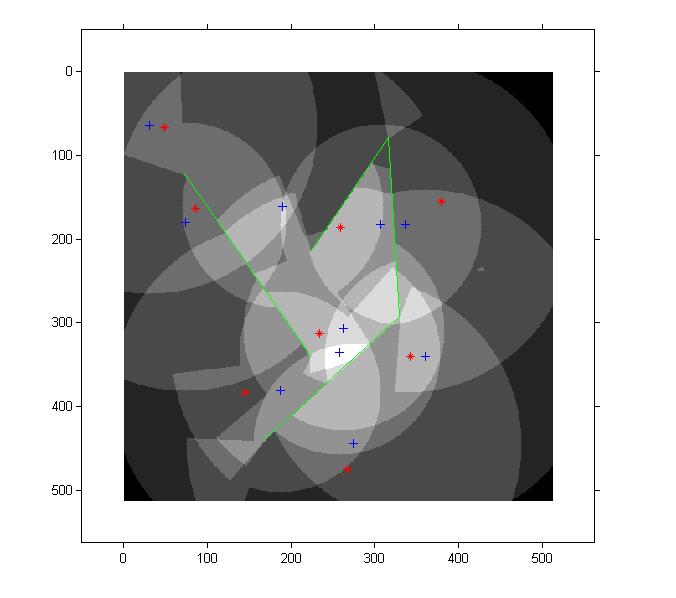


Figura 5.13 - Mapa para visualização de comunicação entre roteadores, considerando para o teste 3.

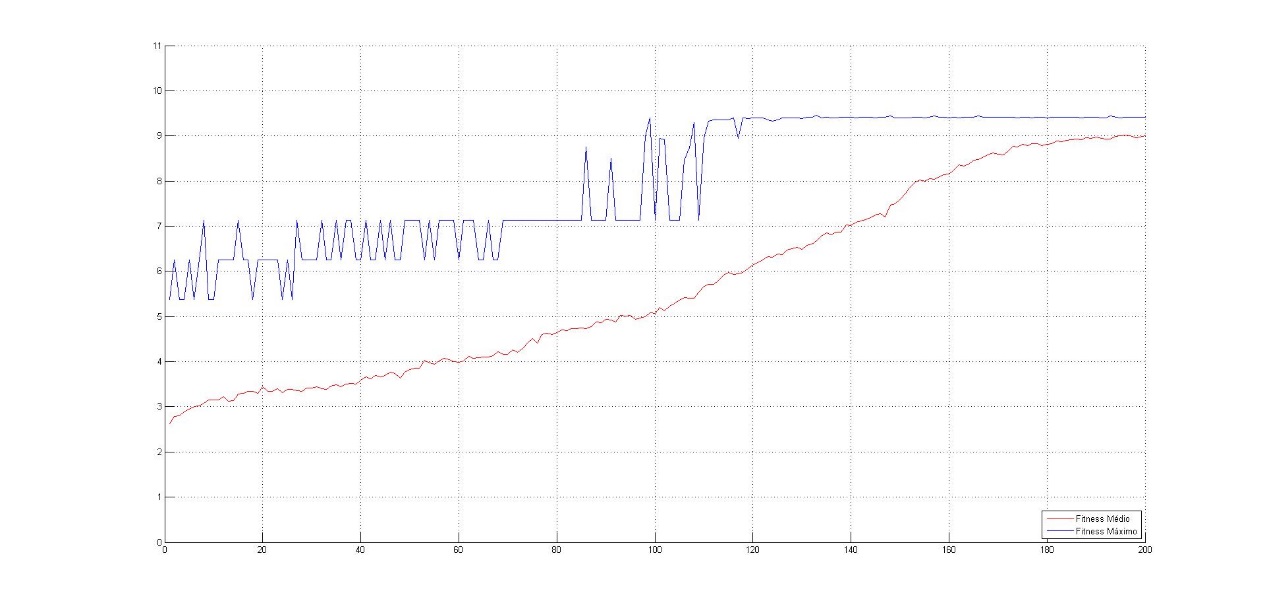


Figura 5.14 – Evolução da aptidão para o teste 3.

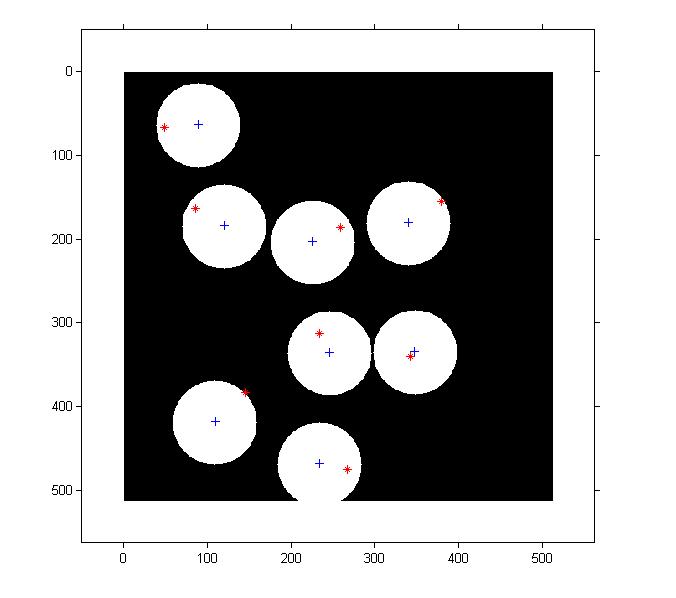


Figura 5.15 - Figura 5.9 – Mapa de sensores cobertos para o teste 4.

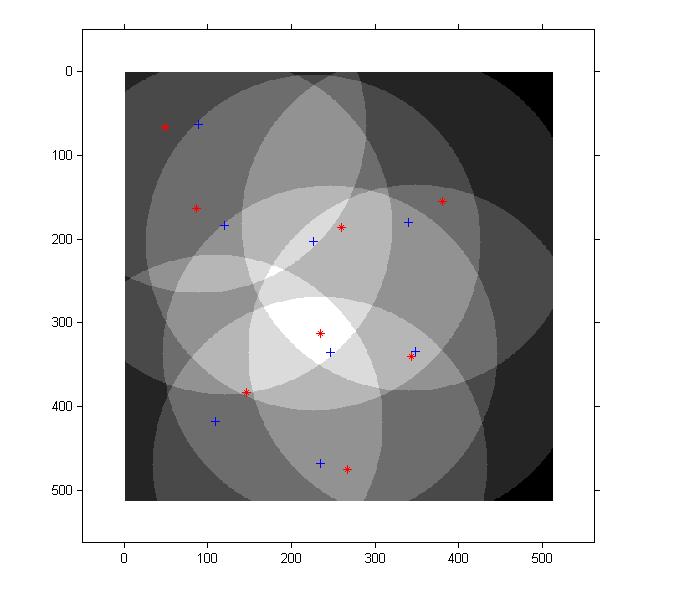


Figura 5.16 - Mapa para visualização de comunicação entre roteadores, considerando para o teste 4.

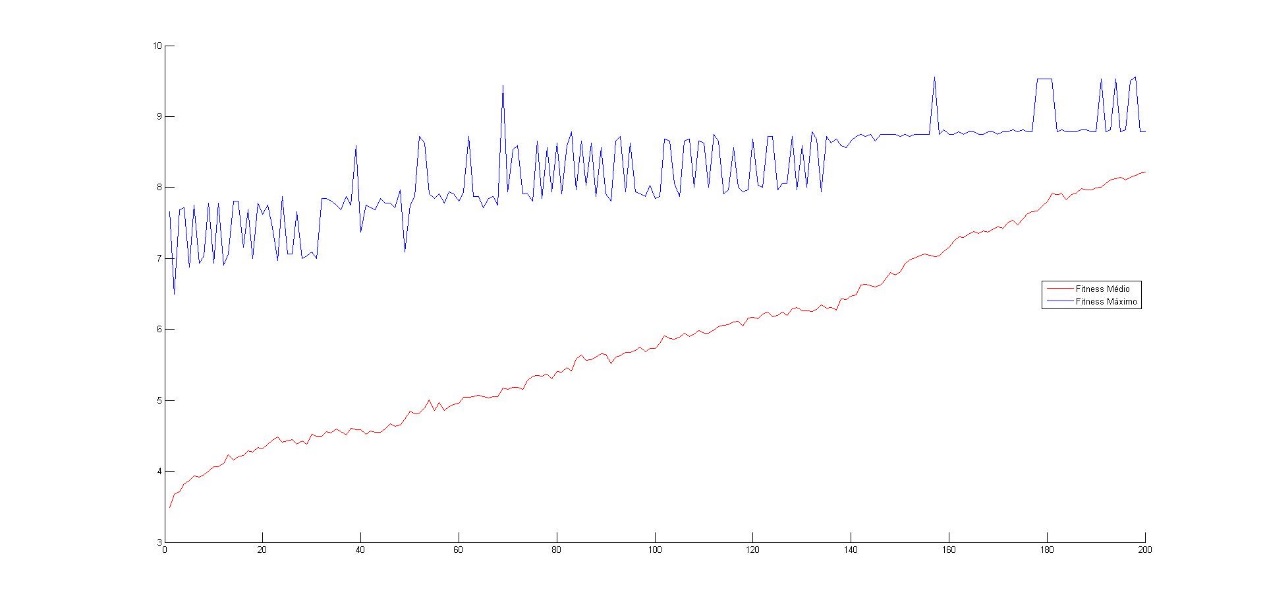


Figura 5.17 – Evolução da aptidão para o teste 4.

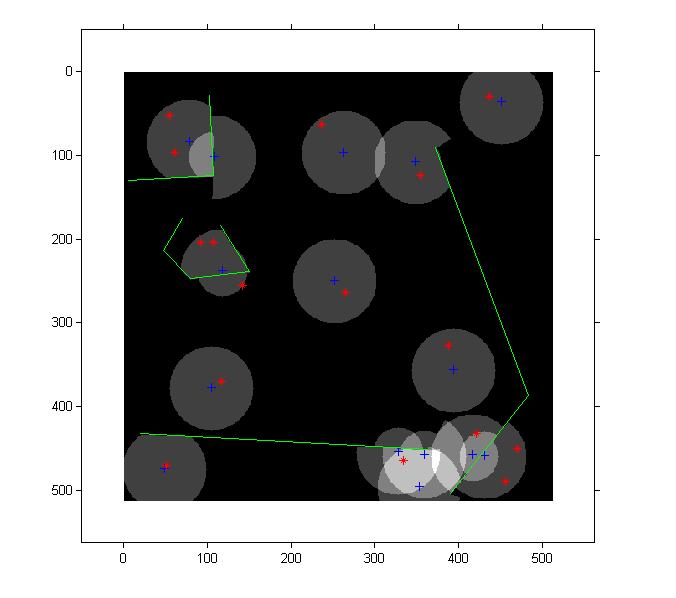


Figura 5.18 - Mapa de sensores cobertos para o teste 5.

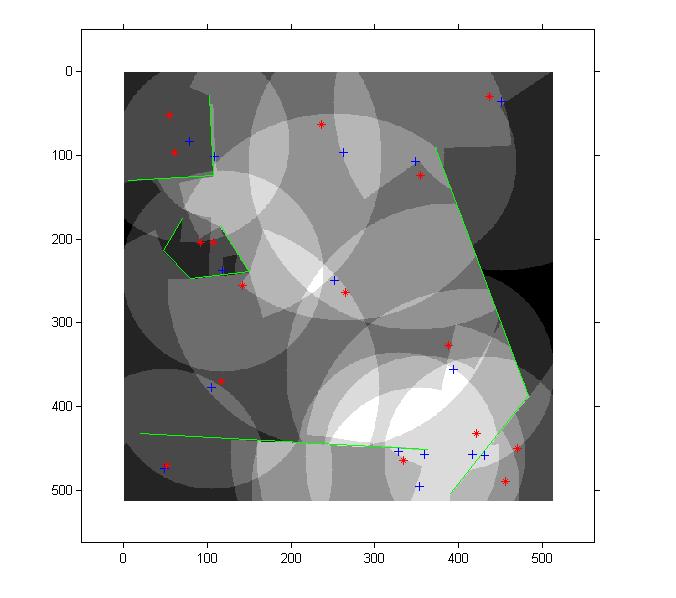


Figura 5.19 - Mapa para visualização de comunicação entre roteadores, considerando para o teste 5.

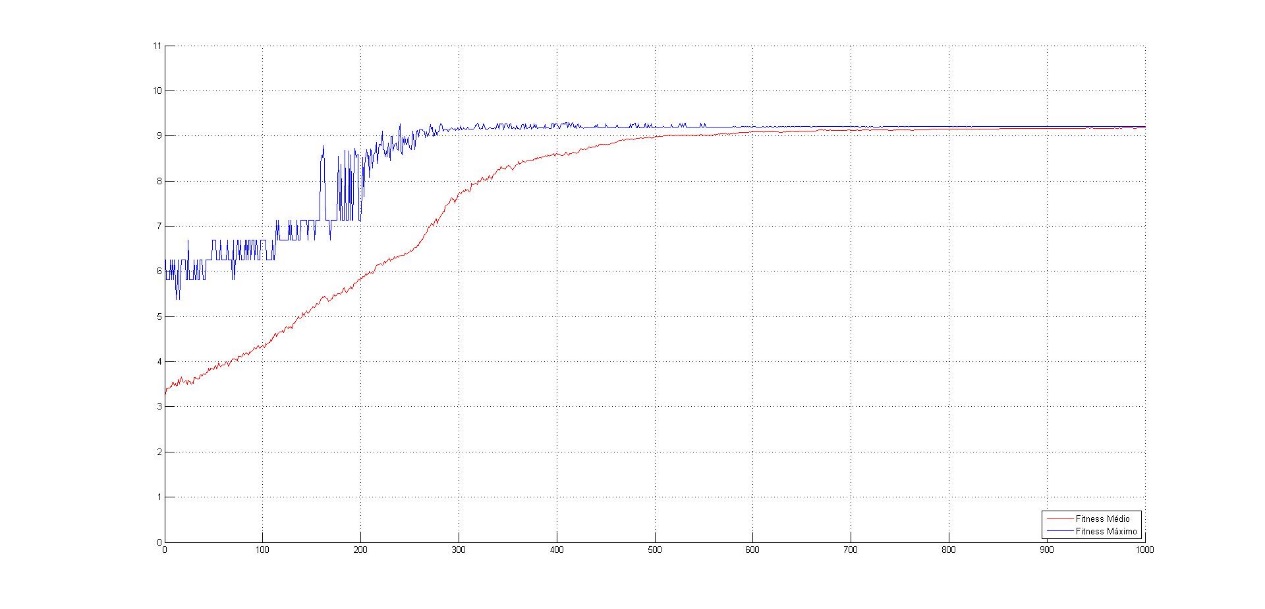


Figura 5.20 – Evolução da aptidão para o teste 5.

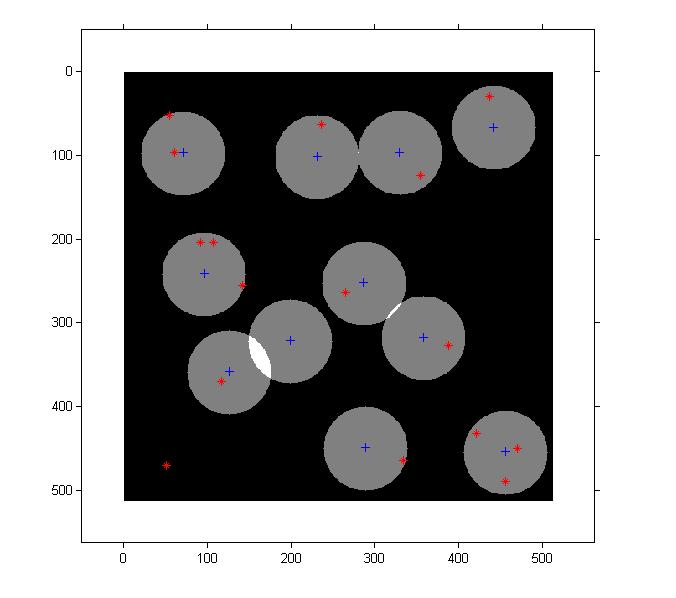
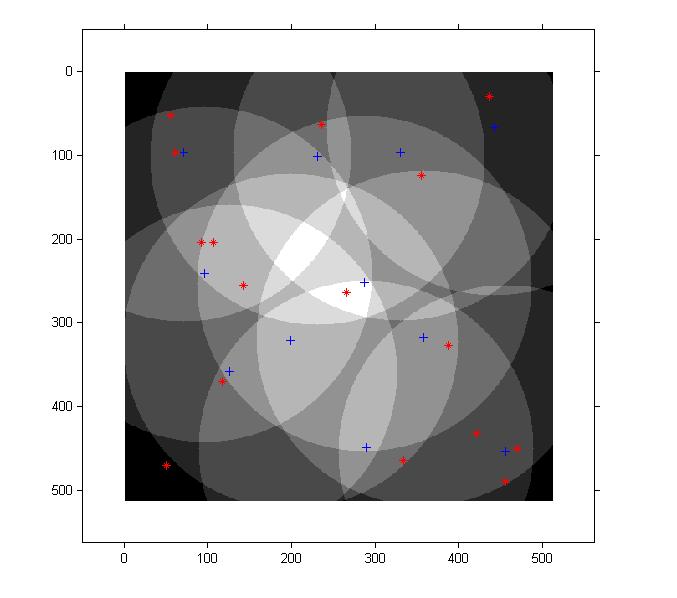


Figura 5.21 - Figura 5.9 – Mapa de sensores cobertos para o teste 6.

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

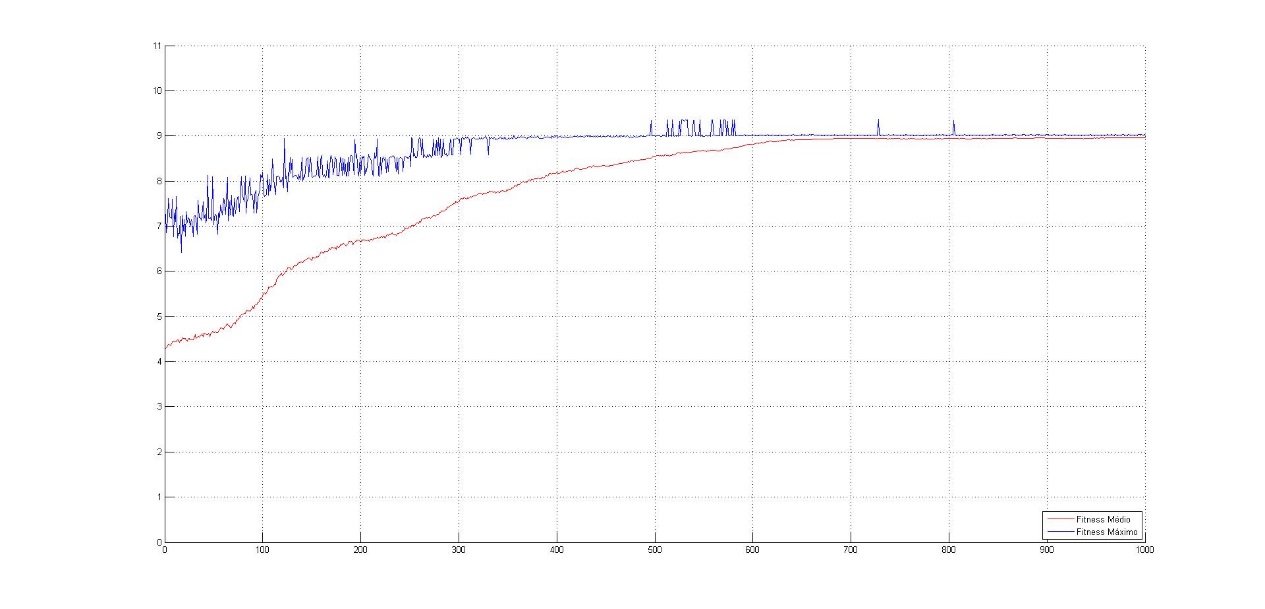


Figura 5.23 – Evolução da aptidão para o teste 6.

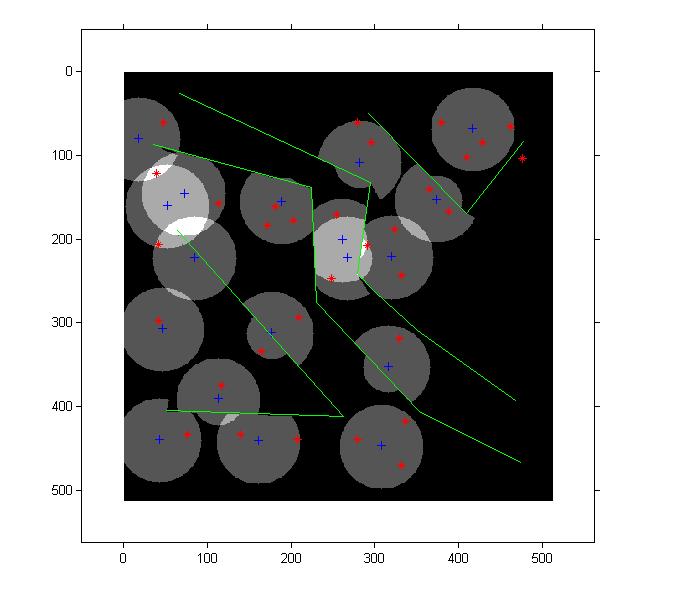


Figura 5.24 - Mapa de sensores cobertos para o teste 7.

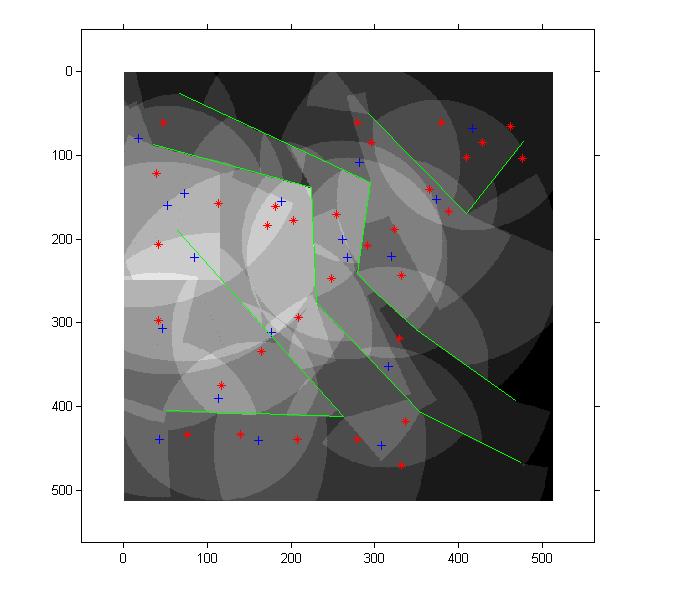


Figura 5.25 - Mapa para visualização de comunicação entre roteadores, considerando para o teste 7.

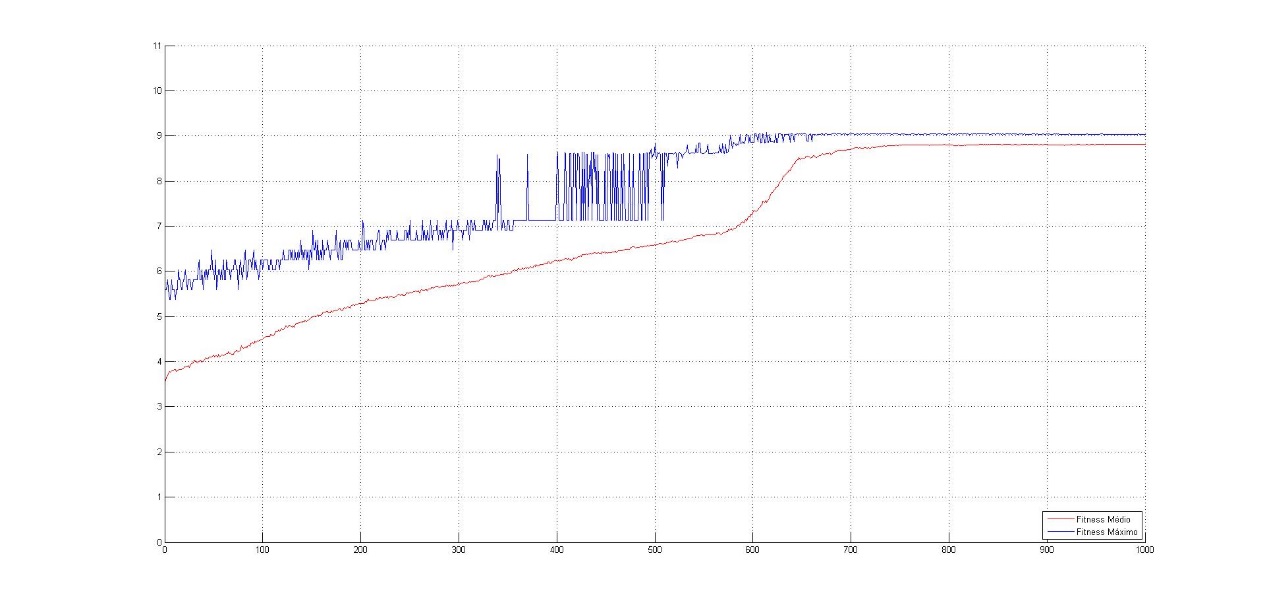


Figura 5.26 – Evolução da aptidão para o teste 7.

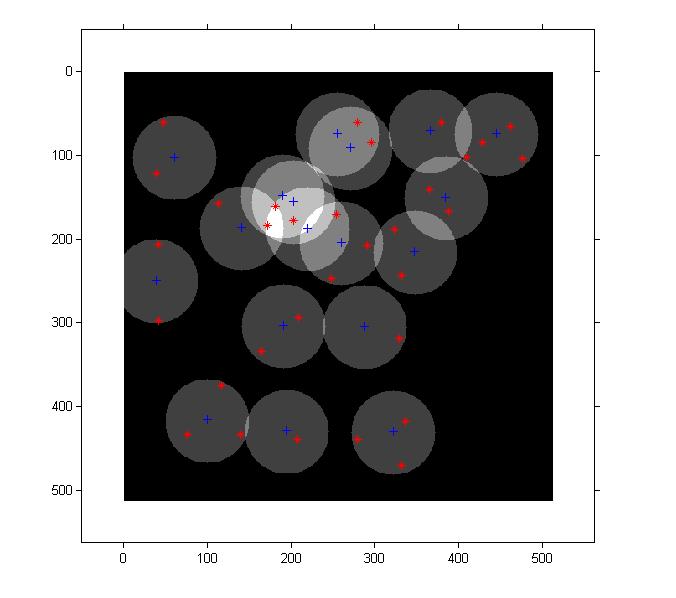


Figura 5.27 - Mapa de sensores cobertos para o teste 8.

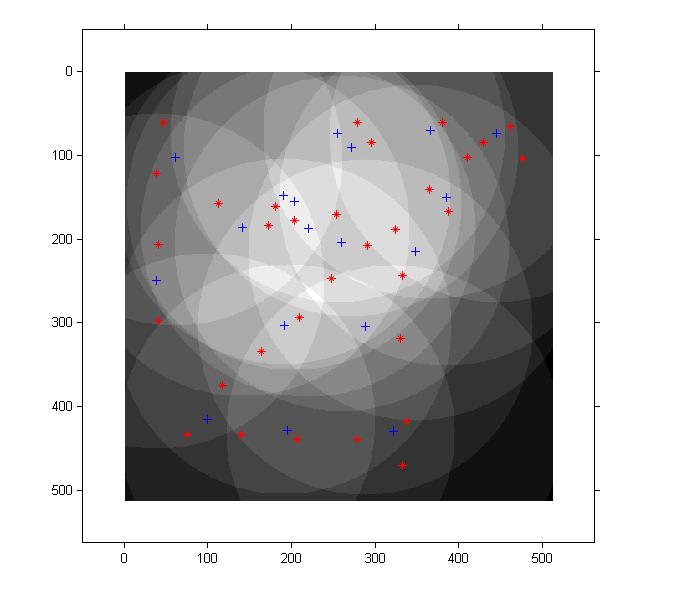


Figura 5.28 - Mapa para visualização de comunicação entre roteadores, considerando para o teste 8.

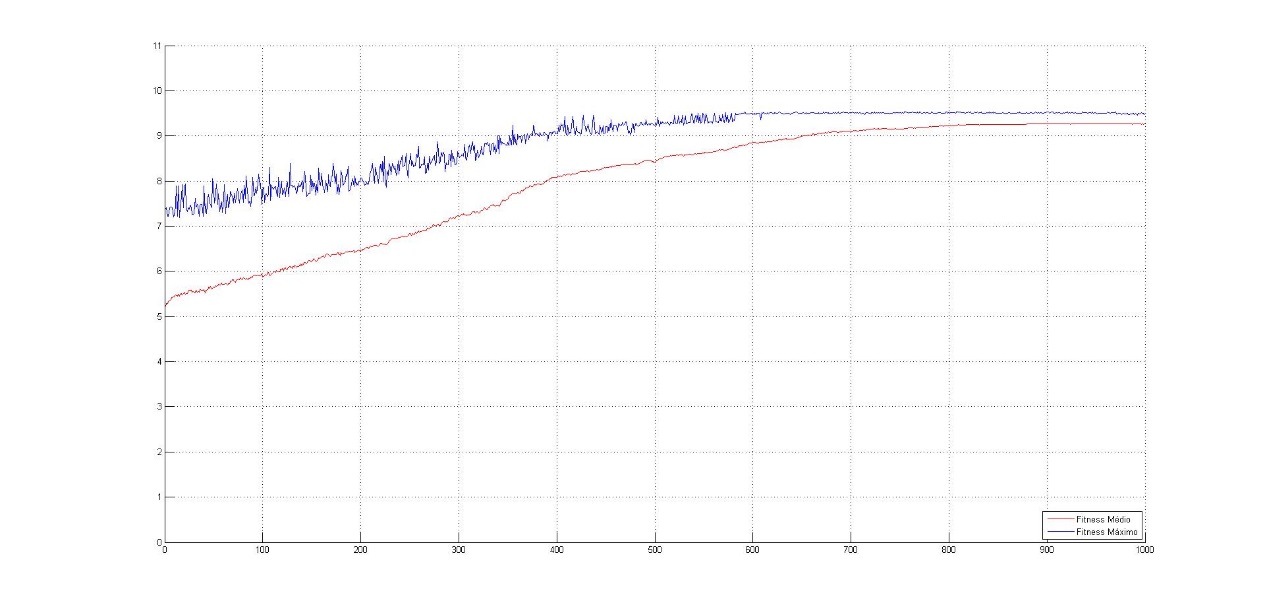


Figura 5.29 – Evolução da aptidão para o teste 8.

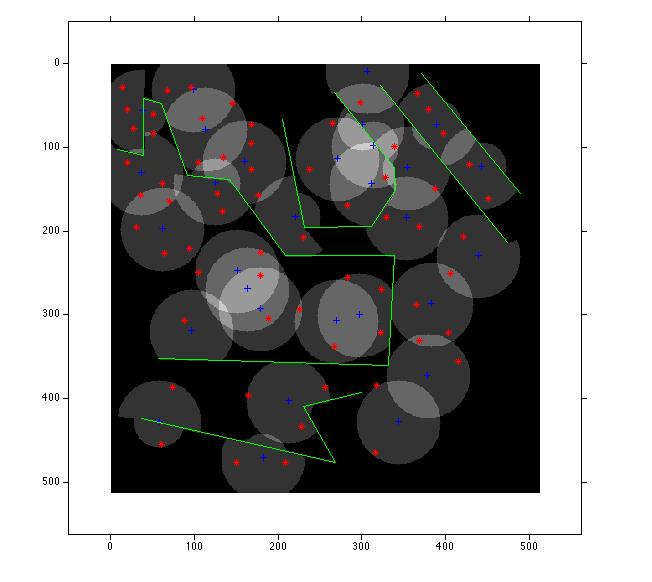


Figura 5.33 - Mapa de sensores cobertos para o teste 9.

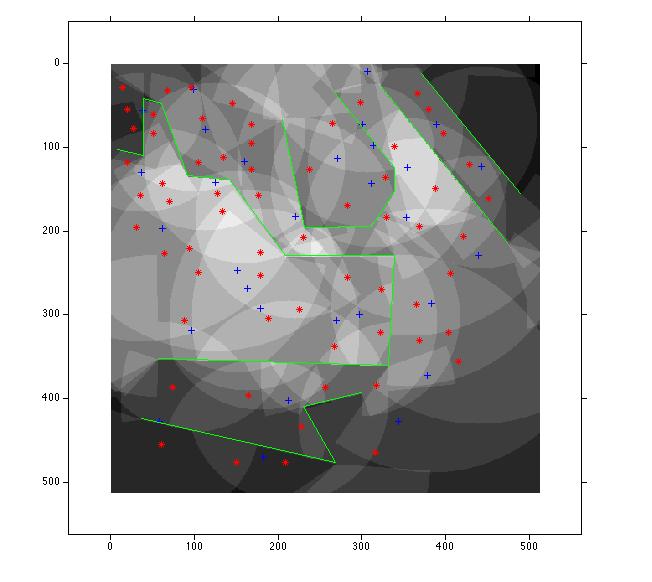


Figura 5.34 - Mapa para visualização de comunicação entre roteadores, considerando para o teste 9.

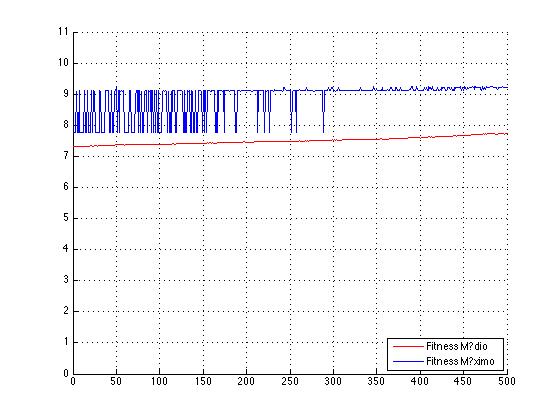
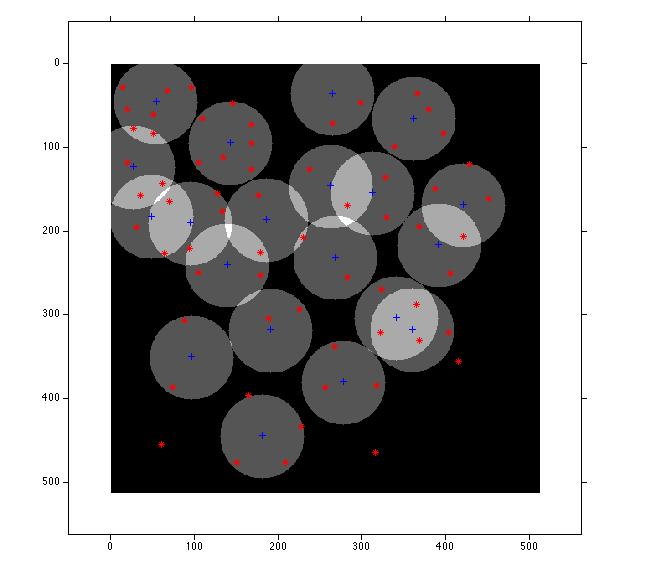


Figura 5.35 – Evolução da aptidão para o teste 9.

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

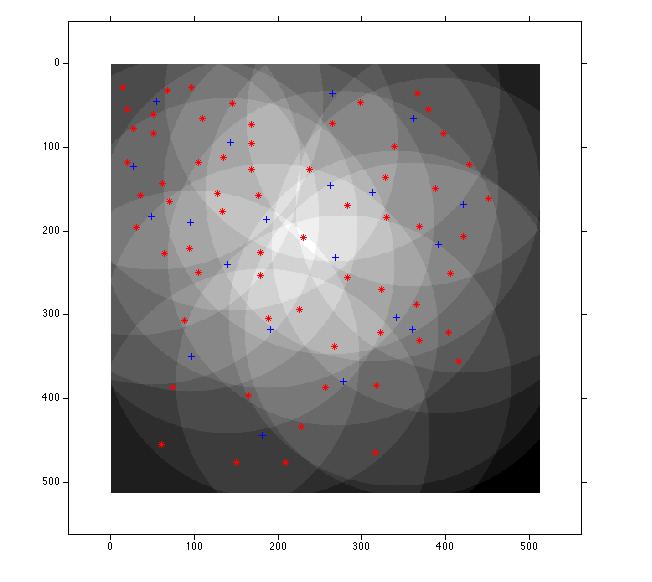


Figura 5.37 - Mapa para visualização de comunicação entre roteadores, considerando para o teste 10.

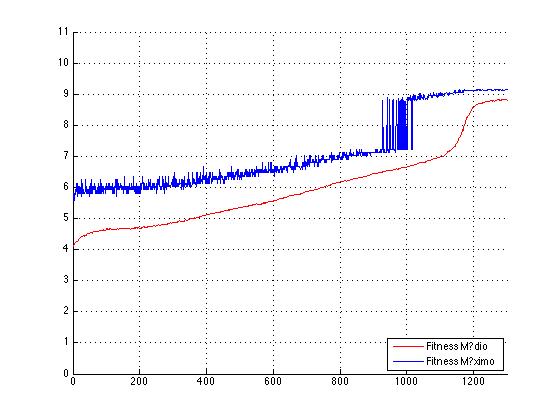
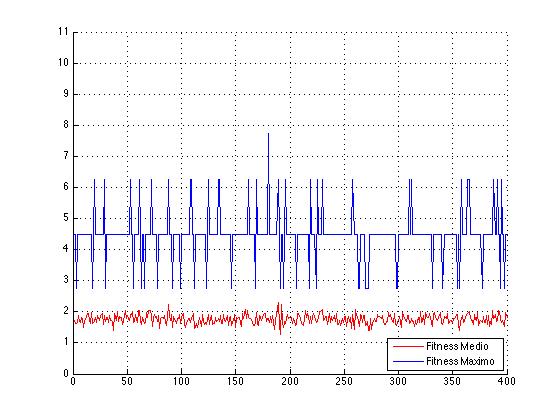


Figura 5.38 – Evolução da aptidão para o teste 10.

* 1. Busca Aleatória

Figura 5.38 – Evolução da aptidão para o teste 1.

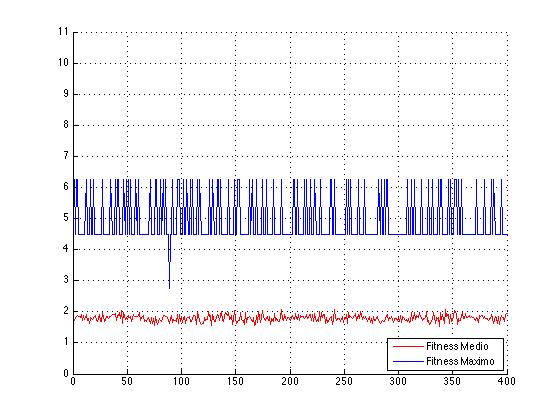


Figura 5.39 – Evolução da aptidão para o teste 2.

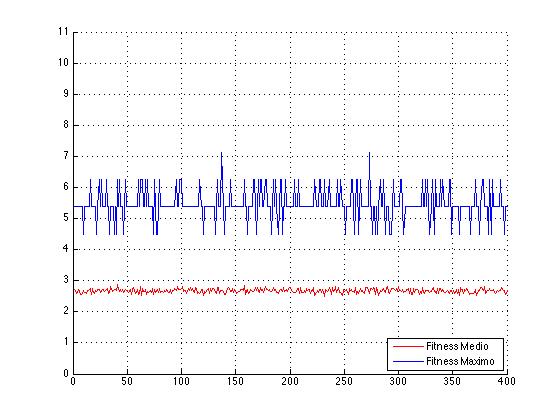


Figura 5.40 – Evolução da aptidão para o teste 3.

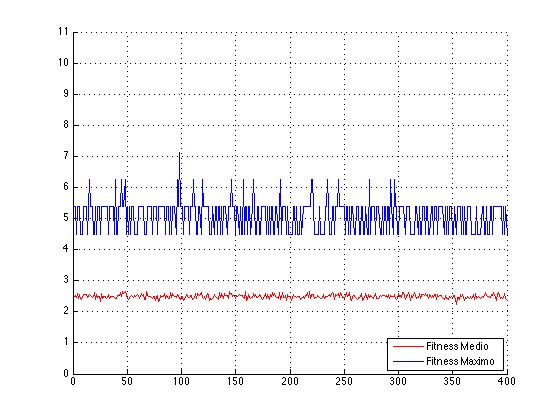


Figura 5.41 – Evolução da aptidão para o teste 4.

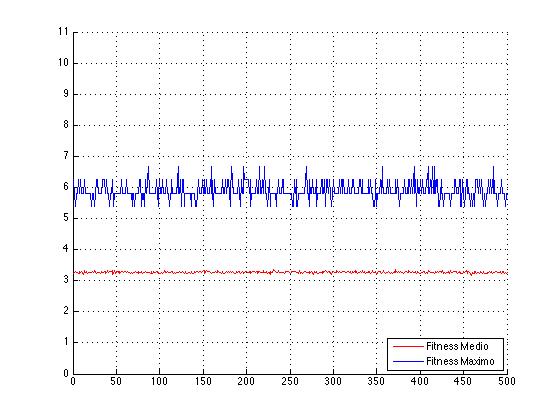


Figura 5.42 – Evolução da aptidão para o teste 5.

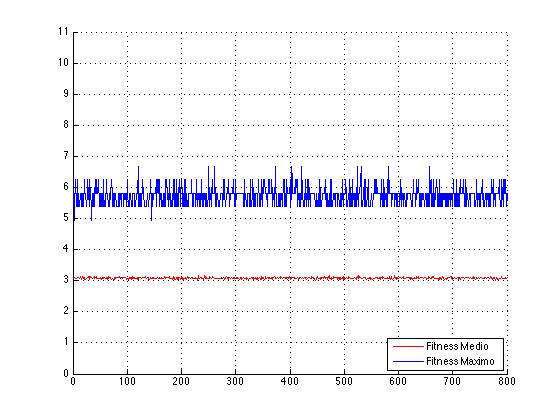


Figura 5.43 – Evolução da aptidão para o teste 6.

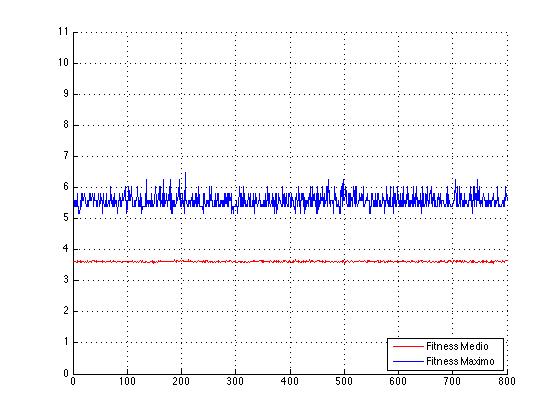


Figura 5.44 – Evolução da aptidão para o teste 7.

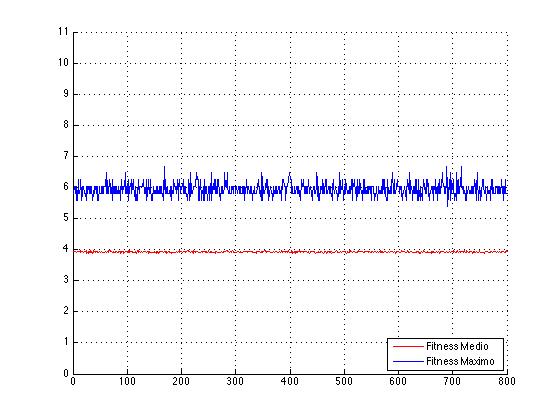


Figura 5.45 – Evolução da aptidão para o teste 8.

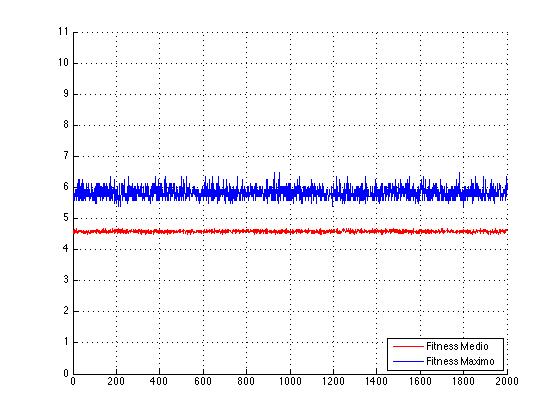


Figura 5.46 – Evolução da aptidão para o teste 9.

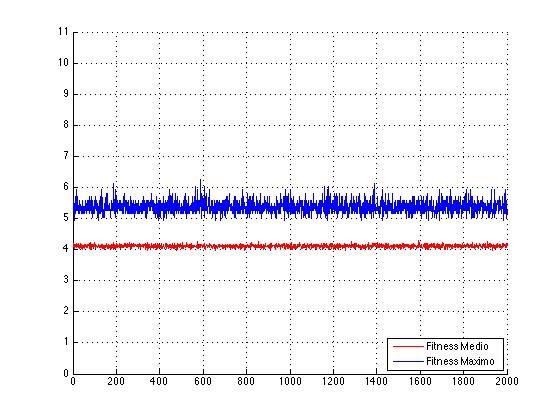


Figura 5.47 – Evolução da aptidão para o teste 10.

Percebe-se, como já foi dito no capítulo 4, que não há um direcionamento dos indivíduos, dessa forma, o surgimento de um indivíduo mais apto está condicionado a um evento probabilistico.

* 1. Busca Gulosa

|  |  |  |
| --- | --- | --- |
| **Teste** | **Roteadores** | **Aptidao** |
| 1 | 4 | 8,875 |
| 2 | 4 | 9,75 |
| 3 | 8 | 8,8438 |
| 3 | 9 | 9,4074 |
| 3 | 10 | 9,4 |
| 4 | 8 | 9,4688 |
| 5 | 15 | 9,2089 |
| 6 | 11 | 9,3223 |
| 6 | 12 | 9,3611 |
| 7 | 18 | 9,2716 |
| 8 | 18 | 9,4074 |
| 9 | 26 | 8,7933 |
| 10 | 20 | 9,3656 |

1. Conclusão
2. 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.

1. Apêndice

Aqui você põe as tabelas de contas, os códigos-fonte, etc.

|  |  |  |  |
| --- | --- | --- | --- |
| 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** | | | |