

**LOCALIZAÇÃO DE SENSORES  
CONSIDERANDO CUSTO MÍNIMO**



ANGELO FERREIRA ASSIS

**LOCALIZAÇÃO DE SENSORES  
CONSIDERANDO CUSTO MÍNIMO**

Dissertação apresentada ao Programa de Pós-Graduação em Ciência da Computação do Instituto de Ciências Exatas da Universidade Federal de Minas Gerais. Departamento de Ciência da Computação como requisito parcial para a obtenção do grau de Mestre em Ciência da Computação.

ORIENTADOR: LUIZ FILIPE MENEZES VIEIRA

Belo Horizonte  
Janeiro de 2013

© 2013, Angelo Ferreira Assis.  
Todos os direitos reservados.

Assis, Angelo Ferreira

A848l      Localização de sensores considerando custo mínimo /  
Angelo Ferreira Assis. — Belo Horizonte, 2013  
xvi, 57 f. : il. ; 29cm

Dissertação (mestrado) — Universidade Federal de  
Minas Gerais. Departamento de Ciência da  
Computação

Orientador: Luiz Filipe Menezes Vieira

1. Redes de Sensores Sem Fio. 2. Localização.  
3. Algoritmo Genético. I. Título.

CDU 519.6\*22 (043)



UNIVERSIDADE FEDERAL DE MINAS GERAIS  
INSTITUTO DE CIÊNCIAS EXATAS  
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

## FOLHA DE APROVAÇÃO

Localização de sensores considerando custo mínimo

**ANGELO FERREIRA ASSIS**

Dissertação defendida e aprovada pela banca examinadora constituída pelos Senhores:

PROF. LUIZ FILIPE MENEZES VIEIRA - Orientador  
Departamento de Ciência da Computação - UFMG

PROF. GERALDO ROBSON MATEUS  
Departamento de Ciência da Computação - UFMG

PROFA. GISELE LOBO PAPP  
Departamento de Ciência da Computação - UFMG

PROF. RICARDO AUGUSTO RABELO OLIVEIRA  
Departamento de Computação - UFOP

Belo Horizonte, 01 de março de 2013.



# Agradecimentos

Toda lista de agradecimentos peca pela ausência de uma ou outra pessoa que teve papel, ainda que pequeno, na conclusão de um trabalho como este. Mesmo assim, tento registrar aqui, em poucas linhas, os meus agradecimentos, sabendo que não são suficientes para expressar a importância de todos nessa minha jornada. Agradeço a todos que contribuíram direta, ou indiretamente, para a concretização deste trabalho:

- A Deus pela minha vida, minha saúde e por guiar meu caminho na conquista de mais um objetivo.
- A minha mãe, que sempre acreditou na minha capacidade e com muita paciência soube aturar as noites em claro que passei durante toda a caminhada.
- Ao meu pai, que apesar de morar longe, nunca deixou de se preocupar com meus estudos, meu trabalho, minha vida.
- A Flavinha, pelo amor, carinho e apoio em todos os momentos que precisei.
- Aos meus amigos que me acompanharam durante a jornada nos momentos de concentração e também descontração, e sempre torceram pelo meu sucesso. Em especial Marco Tulio que além do apoio, fazia sempre um arroz bom demais!
- Aos professores das disciplinas que tive o prazer de cursar. Todos participaram dessa jornada comigo, ou antes dela forneceram-me os meios para a travessia. Em especial agradeço ao meu orientador Luiz Filipe pela paciência e esforço na orientação desse trabalho.





# Resumo

Localização é uma das questões essenciais em Redes de Sensores Sem Fio. Sua utilização é indiscutivelmente importante em diferentes aplicações. Entretanto, é preciso minimizar o custo de localização de toda a rede. Isto pode ser feito definindo alguns nós como âncora, que servirão de referência para os outros nós se localizarem. Várias das soluções existentes focam principalmente em localizar o maior número de nós possível dado um conjunto estático de nós âncoras, fornecendo a cada um destes um recurso como, por exemplo, o GPS para definir sua localização. Porém, isso é inviável para muitas das redes de sensores, devido ao custo e/ou complexidade de implantação. O problema de otimização deste trabalho consiste em encontrar o menor conjunto de nós âncoras necessários para a localização de toda a Rede de Sensores Sem Fio. Outra abordagem é encontrar o menor caminho entre os nós âncoras. Dessa forma pode-se utilizar um robô que tem a função de visitar e definir a posição exata dos nós que funcionarão como âncoras. Nesse caso é preciso minimizar apenas o caminho, pois o custo da localização é o combustível do robô. Aqui é apresentada uma modelagem para o problema usando Algoritmos Genéticos, a fim de criar uma melhor solução para o problema. Foram realizados diversos testes que mostram a eficácia da estratégia com base no número de nós âncoras necessários para localizar toda a rede. Os resultados apresentados mostram que o algoritmo genético alcançou uma solução em média 50% melhor que um algoritmo guloso recente encontrado na literatura, com um tempo de execução viável.

**Palavras-chave:** Redes de Sensores Sem Fio, Localização, Algoritmo Genético.



# Abstract

Localization is one of the key issues in Wireless Sensor Networks. Its use is undoubtedly important in many different applications. However, it's often required to minimize the localization cost in a network. This can be done by setting some nodes as anchors, which are used as reference to other nodes. Current literature solutions focus on finding as many nodes as possible given a static set of anchor nodes, providing each one of these nodes resources such as a GPS to determine its location. On the other hand, this may be unfeasible for many sensor networks due to the high cost and/or implementation complexity. The optimization problem presented in this work consists of finding the minimum set of anchor nodes needed to locate all nodes in the network. Another approach is to find the shortest path between the anchors. This way a robot can be used to visit and set the exact position of the nodes that work as anchors, that is, it is only necessary to minimize the path, since the localization cost is the robot fuel. Here it is presented a model for the problem using Genetic Algorithms in order to create this problem a better solution. Several tests were performed to show the effectiveness of the strategy based on the number of anchors required to locate the entire network. The results have shown that the genetic algorithm reached, on average, a 50%-better solution than the greedy algorithm, having a feasible runtime.

**Keywords:** Wireless Sensor Networks, Localization, Genetic Algorithm.



# Lista de Figuras

1.1	Aplicações de RSSF . . . . .	2
3.1	Exemplo do método Trilateração . . . . .	15
3.2	Exemplo do método Bouding Box. Fonte: Oliveira [2008] . . . . .	16
4.1	Exemplo de varredura local. Fonte: Huang et al. [2011] . . . . .	23
4.2	Exemplo de cruzamento . . . . .	27
4.3	Exemplo de mutação . . . . .	27
4.4	Exemplo de solução para minimização de âncoras . . . . .	28
4.5	Exemplo de solução para minimização do caminho . . . . .	31
5.1	Ambiente do Sinalgo . . . . .	34
5.2	Resultado da execução para minimização de âncoras sem varredura local . . . . .	36
5.3	Tempo de Execução - Minimização de âncoras sem varredura local . . . . .	37
5.4	Evolução do Algoritmo Genético para minimização de âncoras sem varredura local . . . . .	38
5.5	Resultado da execução para minimização de âncoras com varredura local . . . . .	39
5.6	Tempo de Execução - Minimização de âncoras com varredura local . . . . .	40
5.7	Evolução do Algoritmo Genético para minimização de âncoras com varredura local . . . . .	42
5.8	Impacto da variação do raio de comunicação dos nós sem varredura local . . . . .	43
5.9	Impacto da variação do raio de comunicação dos nós com varredura local . . . . .	44
5.10	Resultado da execução para minimização do caminho sem varredura local . . . . .	45
5.11	Tempo de Execução - Minimização de caminho sem varredura local . . . . .	46
5.12	Evolução do Algoritmo Genético para minimização de caminho sem varredura local . . . . .	47
5.13	Resultado da execução para minimização do caminho com varredura local . . . . .	48
5.14	Tempo de Execução - Minimização de caminho com varredura local . . . . .	49

5.15 Evolução do Algoritmo Genético para minimização de caminho com varredura local . . . . .	50
---	----

# Sumário

<b>Agradecimentos</b>	<b>vii</b>
<b>Resumo</b>	<b>ix</b>
<b>Abstract</b>	<b>xi</b>
<b>Lista de Figuras</b>	<b>xiii</b>
<b>1 Introdução</b>	<b>1</b>
1.1 Motivação . . . . .	1
1.2 Objetivos . . . . .	3
1.3 Principais Contribuições . . . . .	4
1.4 Organização do texto . . . . .	5
<b>2 Trabalhos Relacionados</b>	<b>7</b>
2.1 Localização em Redes de Sensores Sem Fio . . . . .	7
2.2 Algoritmo Genético para Localização . . . . .	9
2.3 Problema de Localização com Custo Mínimo . . . . .	9
2.4 Localização com âncoras móveis . . . . .	10
2.5 Coleta de dados com robôs . . . . .	11
<b>3 Definição do Problema</b>	<b>13</b>
3.1 Localização em Redes de Sensores Sem Fio . . . . .	13
3.1.1 Estimativa de ângulo e distância . . . . .	14
3.1.2 Métodos de cálculo de posição . . . . .	15
3.2 Problema de Minimização de Nós Âncoras . . . . .	17
3.2.1 Definição do problema . . . . .	17
3.3 Problema de Minimização de Caminho . . . . .	18
3.3.1 Definição do problema . . . . .	18

<b>4</b>	<b>Definição dos Algoritmos</b>	<b>21</b>
4.1	Algoritmo Guloso . . . . .	21
4.2	Algoritmo Genético . . . . .	24
4.2.1	Seleção por torneio . . . . .	26
4.2.2	Cruzamento ponto a ponto . . . . .	26
4.2.3	Mutação . . . . .	27
4.3	Problema de Minimização de Nós Âncoras . . . . .	27
4.3.1	Algoritmo Genético com Trilateração . . . . .	29
4.3.2	Algoritmo Genético com Trilateração e Varredura Local . . . . .	29
4.4	Problema de Minimização de Caminho . . . . .	30
4.4.1	Algoritmo Genético com Trilateração . . . . .	31
4.4.2	Algoritmo Genético com Trilateração e Varredura Local . . . . .	32
<b>5</b>	<b>Experimentos e Resultados</b>	<b>33</b>
5.1	Problema de Minimização de Nós Âncoras . . . . .	35
5.1.1	Algoritmo Genético com Trilateração . . . . .	35
5.1.2	Algoritmo Genético com Trilateração e Varredura Local . . . . .	39
5.1.3	Impacto da variação do raio de comunicação dos nós . . . . .	43
5.2	Problema de Minimização de Caminho . . . . .	45
5.2.1	Algoritmo Genético com Trilateração . . . . .	45
5.2.2	Algoritmo Genético com Trilateração e Varredura Local . . . . .	48
<b>6</b>	<b>Considerações Finais</b>	<b>51</b>
6.1	Conclusões . . . . .	51
6.2	Trabalhos Futuros . . . . .	52
	<b>Referências Bibliográficas</b>	<b>53</b>



# Capítulo 1

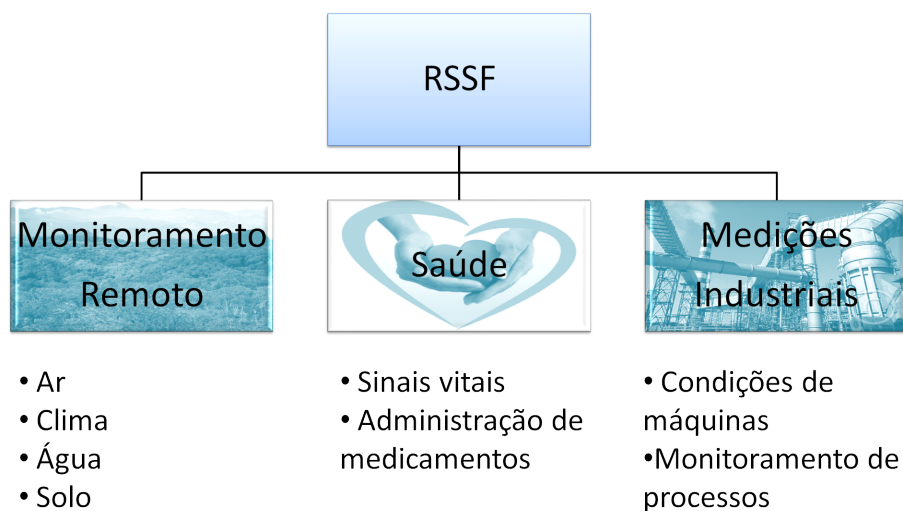
## Introdução

As redes de sensores sem fio (RSSF) têm recebido considerável atenção da comunidade científica nos últimos anos. Isso porque mesmo com a capacidade de memória e computação limitadas dos sensores, é possível monitorar vários aspectos do ambiente como temperatura, umidade, pressão, movimento, etc [Loureiro et al., 2003; Ruiz et al., 2004].

Devido à grande aplicabilidade, as RSSF se tornaram populares e envolvem diferentes áreas como meio ambiente, indústrias, medicina, entre outros, como mostrado na Figura 1.1. Por exemplo, na área médica, as RSSFs podem ser utilizadas para monitorar o funcionamento de órgãos como o coração, detectar a presença de substâncias que indicam a existência ou o surgimento de algum problema biológico, seja no corpo humano ou em animais. Na área ambiental, as redes de sensores são importantes na prevenção e no auxílio às vítimas de catástrofes naturais tais como terremotos, vulcões, tsunamis, furacões, queimadas, etc, e ainda podem auxiliar na previsão do tempo [Yawut & Kilaso, 2011]. Além disso, as redes de sensores sem fio são capazes de garantir o controle de dados em áreas de difícil acesso ou perigosas. Enfim, em aplicações desses tipos se torna fundamental o conhecimento da posição de cada um dos sensores.

### 1.1 Motivação

Um dos principais problemas em RSSF é a localização dos nós sensores. Isso se deve à natureza intrínseca das aplicações de redes de sensores. Informações de localização podem ser usadas em muitas aplicações de redes de sensores sem fio [Akyildiz et al., 2002]. Em aplicações típicas, os sensores medem alguma grandeza física em um ponto do espaço, de forma que o valor medido daquela grandeza somente terá algum significado se



**Figura 1.1.** Aplicações de RSSF

acompanhada da posição em que tal medida foi feita. Além disso, informações de localização podem beneficiar protocolos de rede para melhorar o desempenho de redes de sensores de diferentes maneiras, como é o caso de algoritmos geográficos [Karp & Kung, 2000]. Portanto, é fundamental que algumas redes de sensores tenham conhecimento da posição de cada um dos seus nós sensores.

Um possível método para determinar a localização de toda a rede é configurar manualmente a posição dos nós sensores. No entanto, não é uma solução viável para implementações em grande escala e em cenários onde os nós se movem frequentemente. Outra alternativa é equipar cada sensor com um Sistema de Posicionamento Global (GPS) [Hofmann-Wellenhof et al., 2001]. Porém, isso é inviável para a maioria das redes de sensores, devido ao custo e/ou complexidade de implantação, maior consumo de energia, aumento do tamanho dos sensores, entre outros.

A principal tarefa de localização em redes de sensores sem fio é obter a localização exata de cada sensor em um plano bidimensional (2D). Sendo assim, vários trabalhos propõem determinar a localização de todos os nós da rede onde apenas alguns nós especiais, que são chamados de nós âncoras, ou nós beacons, conhecem sua posição [He et al., 2003; Savvides et al., 2001; Moore et al., 2004]. Os demais sensores irão determinar sua localização através da medição das distâncias aos seus vizinhos, usando diferentes métodos (como por exemplo a intensidade do sinal de comunicação [Rodrigues et al., 2011]). Entretanto, é importante que o número de nós âncoras seja o menor possível, pois são nós mais caros em relação aos demais. Sendo assim, esse trabalho propõe um algoritmo para resolver o problema de minimização do número de nós âncoras para que o custo de localização da rede seja minimizado. Uma abordagem para

esse problema é detalhada no trabalho de Huang et al. [2011], e será discutida na Seção 2.

Uma outra opção para minimizar custos é definir um trajeto para um robô que tem a função de visitar e definir a posição exata dos nós que funcionarão como âncoras. Dessa forma, é preciso considerar o custo do trajeto do robô, em vez do número de âncoras. Esse trabalho propõe também outro algoritmo para resolver o problema dessa forma, minimizando o caminho do robô, em vez do número de nós âncoras. Esse trajeto deve ser o menor possível, independente do número de nós visitados, pois nesse caso o custo de localização é proporcional ao tempo e/ou distância gasto pelo robô. O problema do caixeiro viajante (PCV), bem conhecido na literatura [Applegate et al., 2007], procura pelo trajeto de menor distância que visite todas as cidades em um conjunto. Esse trajeto se inicia em qualquer cidade, visita cada uma apenas uma vez e retorna à cidade inicial. Caso seja possível calcular a distância do robô a um determinado nó, assim como definir o ângulo do sinal recebido desse nó, o robô não precisa visitar a localização exata do sensor. Em vez disso, pode-se visitar um ponto dentro do raio de comunicação do nó, para verificar se este já conhece sua posição através de outros sensores<sup>1</sup>. O nó deverá ser visitado pelo robô apenas se não conhecer ainda sua localização. Uma variação do PCV é o PCV com vizinhança (em inglês TSP with Neighborhoods - TSPN), que se assemelha ao problema de visitação da região de raio de comunicação do nó. Esse problema também é conhecido, em inglês, como Covering Salesman Problem (CSP), Group-TSP (GTSP), One-of-a-Set-TSP, ou Errand Scheduling problem, como mostrado no trabalho de Safra & Schwartz [2006]. Enfim, o PCV com vizinhança é uma generalização do PCV, e como definido no trabalho de Current & Schilling [1989], deve-se identificar o menor trajeto em um subconjunto de nós, de forma que todo nó é de alguma forma coberto por um outro nó que pertence ao trajeto.

## 1.2 Objetivos

Neste trabalho é proposto um algoritmo eficaz para resolver o problema de localização com custo mínimo com duas abordagens. Uma das abordagens visa determinar o número mínimo de nós âncoras necessários em uma rede de sensores sem fio para que seja possível determinar a localização de todos os outros nós da rede utilizando dois métodos diferentes, o de trilateração e o método da trilateração com varredura local. A outra abordagem visa determinar quais nós serão âncoras considerando o

---

<sup>1</sup>Essa é uma aplicação que depende das características do sinal, qualidade do sinal e do ambiente e nível de consumo de energia.

menor caminho para o robô entre os nós escolhidos, de forma que seja possível localizar toda a rede. Novamente, a localização da rede pode ser feita com dois métodos, o de trilateração e o método da trilateração com varredura local.

O uso de algoritmos genéticos e várias outras técnicas em problemas de redes de sensores sem fio tem sido explorado com sucesso [Gao et al., 2010; Duckett, 2003]. A desvantagem da maioria dessas abordagens é que resolvem o problema de localização de uma rede de sensores sem fio sem se preocupar em encontrar a solução para o problema de forma que minimize o custo de localização.

### 1.3 Principais Contribuições

Essa dissertação procura minimizar custos de localização em uma rede de sensores sem fio. As principais contribuições são:

- As definições formais para o problema de acordo com o custo que se deseja minimizar. Definições de trabalhos existentes até então não focam na minimização do custo ou então focam apenas em minimizar o número de âncoras, desconsiderando outras possibilidades de redução de custo de localização.
- Discussão da semelhança entre o problema de minimização de caminho de âncoras móveis com problemas clássicos da literatura. O Problema do Caixeiro Viajante já é bem definido e estudado, possuindo vários algoritmos eficientes para alcançar soluções muito boas. Definir sua relação com o problema de localização com custo mínimo auxilia na resolução do problema em questão.
- Desenvolvimento de um Algoritmo Genético, que utiliza recursos de localização já conhecidos. O algoritmo implementado usa a técnica da trilateração com varredura local para obter resultados melhores que os encontrados na literatura. O algoritmo é adaptado para minimizar o número de âncoras ou minimizar o caminho entre nós âncoras.
- Várias simulações foram realizadas para avaliar o algoritmo proposto. Os resultados ilustram bem o funcionamento do algoritmo genético, mostrando que a solução evolui ao passar das gerações e alcança ao final bons resultados. As simulações mostram ainda o impacto da variação de alguns parâmetros de rede na solução.
- São apresentadas comparações do algoritmo genético proposto com um algoritmo guloso recente e eficiente encontrado na literatura. Os resultados mostram que

o algoritmo genético para o problema de minimização de nós âncoras alcançou soluções em média 50% melhores em simulações realizadas para redes de 1500 a 3000 nós.

## 1.4 Organização do texto

O restante dessa dissertação está organizado da seguinte forma:

- **Capítulo 2 [Trabalhos Relacionados]:** Apresenta um estudo bibliográfico de trabalhos relacionados ao problema de localização, algoritmos genéticos e roteamento de robôs. São mostrados os aspectos positivos de cada um dos trabalhos, destacando as diferenças existentes com o problema proposto na dissertação.
- **Capítulo 3 [Definição do Problema]:** Esse capítulo apresenta aspectos importantes para o problema de localização, além de definições formais de cada abordagem do problema de localização de custo mínimo considerada neste trabalho. A primeira abordagem trata do problema de minimização dos âncoras, enquanto a outra trata do problema de minimização do caminho entre os âncoras.
- **Capítulo 4 [Definição dos Algoritmos]:** É apresentado o algoritmo proposto neste trabalho para o problema, assim como detalhes sobre a modelagem para as diferentes abordagens tratadas.
- **Capítulo 5 [Experimentos e Resultados]:** São discutidos os experimentos e os resultados alcançados. Para cada uma das abordagens é mostrada a evolução do algoritmo genético e uma comparação do resultado final com o algoritmo guloso.
- **Capítulo 6 [Considerações Finais]:** São apresentadas nesse capítulo as conclusões, observações finais e possíveis extensões do presente trabalho.



# Capítulo 2

## Trabalhos Relacionados

Nos últimos anos, o problema de localização em RSSF tem sido consideravelmente estudado. Esses estudos resultaram em diferentes técnicas que objetivam a localização do maior número de sensores possível.

Estudos mais recentes já propõem a minimização do número de nós âncoras necessários para a localização total da rede. Serão apresentados trabalhos relevantes na área de localização em RSSF, assim como trabalhos sobre o algoritmos genéticos e roteamento de robôs. São mostrados ainda os aspectos positivos de cada um dos trabalhos, destacando as diferenças existentes com o problema proposto na dissertação.

### 2.1 Localização em Redes de Sensores Sem Fio

No trabalho de Aspnes et al. [2006], são destacados fundamentos teóricos para o problema de localização. Esse problema é basicamente definido como uma rede onde alguns nós já sabem sua localização e outros nós determinam suas localizações através de medições das distâncias de seus vizinhos. Nesse caso os nós utilizados como âncoras são estáticos, porém a localização também pode ser realizada com nós âncoras móveis, como mostrado no trabalho de Ssu et al. [2005]. Ainda no trabalho de Aspnes et al., são identificadas algumas questões fundamentais como quais as condições para localização da rede, ou qual a complexidade computacional da localização. É mostrado que uma rede tem uma localização única se e somente se possui um grafo correspondente que é globalmente rígido. Esse conceito é apresentado pelos autores, e mostra que o problema é NP-Completo. No grafo correspondente cada vértice representa a um nó sensor, e dois vértices são conectados se a distância entre os dois é conhecida; isto é, quando a distância entre dois nós sensores pode ser medida ou quando os dois nós sensores

são âncoras e a distância é implicitamente conhecida. A complexidade computacional analisada para esse tipo de grafo mostra que o problema de localização é NP-difícil.

Shih & Marron [2010] utilizam o método da trilateração para realizar a localização de uma rede de sensores 3D. O objetivo do trabalho é reduzir a complexidade da trilateração 3D para uma trilateração 2D através de nós que os autores chamaram de super âncoras. Na trilateração 2D são necessários no mínimo 3 nós vizinhos com posição conhecida e distância já determinada ao nó que se deseja localizar no momento. Os resultados mostram que é possível diminuir o custo computacional da localização e alcançar melhores resultados. Porém o foco dos autores é a rede de sensores tridimensional.

Vários trabalhos como o de Alves [2012] propõem excelentes algoritmos para localização, além de estudos detalhados de técnicas existentes. O diferencial do trabalho aqui proposto é a minimização do custo de localização da rede, seja considerando o número de âncoras, ou o caminho entre eles.

No trabalho de Rodrigues et al. [2011] são utilizadas diferentes tecnologias de rede sem fio para localização, como por exemplo, Wi-Fi, Bluetooth e Zigbee. São investigadas as vantagens e desafios da utilização de múltiplas tecnologias de radiofrequência para realizar localização em ambientes internos. As duas principais vantagens do uso dessa técnica são a possibilidade de aumentar a exatidão e a disponibilidade do serviço de localização. Os dois principais desafios são descobrir como integrar as múltiplas tecnologias e como evitar que efeitos adversos dessa integração possam degradar a qualidade da localização obtida. Trabalhos como esse são importantes por tratarem o problema focando na precisão da localização, entretanto o trabalho proposto nessa dissertação foca em minimizar o custo, sem se preocupar com as técnicas de localização.

No trabalho de Doherty et al. [2001] é proposto um método para estimar a posição dos nós desconhecidos em uma RSSF baseando-se em restrições deduzidas da conectividade entre os nós, através de um modelo conhecido de comunicação par-a-par. Segundo os autores tais restrições são fortes o suficiente para estimar a posição dos nós e simulações mostram que tais posições estimadas são próximas às posições reais dos nós.

Niewiadowska-Szynkiewicz & Marks [2009] apresentam um método baseado em trilateração e meta-heurísticas (*Simulated Annealing* e Algoritmo Genético). Um primeiro posicionamento da rede é feito através de trilateração e em seguida é aplicada uma das meta-heurísticas para refinamento da posição. Uma das formas de minimizar o erro encontrado consiste em substituir os nós de referência da trilateração caso o erro fique acima de um limiar aceitável. Eles comparam os resultados encontrados pelo algoritmo com técnicas de SDP (*Semidefinite Programming*) e com outro trabalho



que também aplica *Simulated Annealing* e mostram que alcançam melhores resultados tanto para redes com âncoras bem distribuídos quanto para redes com âncoras concentrados em determinadas regiões da rede. Esse é um trabalho que alcança ótimos resultados para a minimização do erro da localização, já o objetivo do trabalho nessa dissertação é minimizar o custo de localização considerando técnicas já existentes para tratar da precisão dessa localização.

## 2.2 Algoritmo Genético para Localização

Em Gao et al. [2010] é tratado um problema de localização em que é dada uma região com nós sensores distribuídos em posições previamente não definidas. Dentre esses nós, é determinado um subconjunto de nós âncoras que possuem suas posições definidas de maneira precisa. A partir da posição dos sensores desse subconjunto são determinadas as coordenadas dos nós cujas posições geográficas são ainda desconhecidas.

No trabalho de Zhang et al. [2008] é proposta uma localização baseada em algoritmo genético, onde os operadores de mutação e cruzamento do algoritmo genético funcionam de acordo com a estrutura de vizinhança da rede. Esse trabalho mostra a eficiência de algoritmos gulosos se comparado a outras heurísticas encontradas na literatura. Já no trabalho de Vecchio et al. [2012] é proposto um algoritmo evolucionário multi-objetivo que trata concorrentemente a precisão da localização e certas restrições topológicas deduzidas a partir da conectividade dos nós.

Essas abordagens resolvem o problema de localização proposto em cada uma, porém não visam encontrar a solução para o problema de forma que minimize o custo total de localização, tendo como foco a precisão da localização dos sensores, assim como no trabalho de Niewiadomska-Szynkiewicz & Marks [2009].

## 2.3 Problema de Localização com Custo Mínimo

O trabalho de Huang et al. [2011] trabalha com o Problema de Localização de Custo Mínimo (Minimum Cost Localization Problem - MCLP). MCLP é um problema de otimização com o objetivo de localizar toda uma RSSF usando o menor número possível de nós âncoras.

Para a definição desse problema considera-se que uma rede de sensores seja modelada como um grafo  $G = (V, E)$ , onde  $V$  é um conjunto de  $n$  sensores  $v_1, \dots, v_n$  e  $E$  é um conjunto de enlaces. Se um enlace  $v_i v_j \in E$ , a distância entre os sensores  $v_i$  e  $v_j$  pode ser medida ou estimada através de comunicação sem fio. Cada sensor pode ter raio

de comunicação diferente. Supõe-se também que três sensores não são colineares. Um subconjunto de sensores  $B \subset V$  são nós âncoras cujas posições são conhecidas no início da localização. Os sensores restantes dependerão de medições de distância em  $E$  e das posições dos nós âncoras  $B$  para determinar as suas localizações durante o processo de localização. O objetivo do problema de localização de custo mínimo é encontrar um subconjunto de sensores  $B$  para serem nós âncoras tal que todos os sensores possam ser localizados e suas posições possam ser calculadas, dado o grafo, o comprimento de todos os enlaces, e as posições de todos os nós âncoras; e o número total de nós âncoras  $|B|$  seja minimizado.

Dada a definição do problema, foram definidos quatro algoritmos gulosos baseados em trilateração para alcançar o objetivo. A ideia básica de todos os algoritmos é a seguinte: (1) Todos os nós com grau menor que três são obrigatoriamente âncoras, pois não podem ser localizados a partir de outros nós. (2) Em cada passo, é selecionado de forma gulosa um nó que ainda não conhece a localização, e que mais pode beneficiar o procedimento de localização no próximo passo se for selecionado como âncora, e então define-se este nó como âncora. Todos esses algoritmos são definidos de forma centralizada, entretanto, podem também ser implementados de forma distribuída. Após inúmeras simulações, os autores demonstraram uma eficiência satisfatória dos algoritmos propostos. Entretanto o problema é ainda desafiador, considerando que verificar se uma solução (conjunto de nós âncoras) pode localizar todos os sensores é um problema NP-difícil.

O trabalho de Khan et al. [2009] propõe um algoritmo para localizar nós sensores definindo um mínimo de âncoras necessários para tal tarefa. As posições iniciais dos nós são estimadas através de uma suposição aleatória, por exemplo. O trabalho sugere uma fórmula matemática para o limite inferior de nós âncoras em uma rede de sensores com  $m$  dimensões. A equação que define esse limite é dada por  $m + 1 \leq |B|$ .

## 2.4 Localização com âncoras móveis

Âncoras móveis são nós que conhecem a própria posição (por exemplo, com auxílio de um GPS) e que tem a habilidade de se locomover pelo campo de sensoriamento. Esse âncora pode ser um humano, um veículo não tripulado, um robô, etc.

O trabalho de Kuo et al. [2009] aborda o problema de localização com âncoras móveis de uma maneira diferente. O problema proposto pelos autores assume que podem existir alterações nas posições dos âncoras e deve-se identificar âncoras instáveis observando apenas os âncoras estáveis. Esses nós instáveis devem ser removidos do

sistema de localização.

Em Ssu et al. [2005], são utilizados nós âncoras móveis para determinar a posição de outros sensores na rede. São adicionados nós virtuais com localização conhecida durante a execução do algoritmo. Um nó virtual representa a localização instantânea de um nó âncora durante seu movimento. O trabalho de Sichitiu & Ramadurai [2004] também utiliza âncoras móveis para auxiliar os nós de uma RSSF a estimar suas posições. Entretanto, é preciso que o caminho percorrido pelo âncora seja minimizado para diminuir o custo total da localização.

## 2.5 Coleta de dados com robôs

Nos trabalhos de Bhadauria et al. [2011] e Yuan et al. [2007] são apresentados sistemas robóticos para coleta de dados em Redes de Sensores Sem Fio. O primeiro deles utiliza robôs que percorrem a rede e reúnem os dados dos nós da RSSF. O problema, denominado Problema de Coleta de Dados (Data Gathering Problem - DGP), foi considerado para realizar o planejamento de caminhos de vários robôs e ainda coletar os dados de todos os sensores com o menor tempo possível. É considerado que o tempo total não depende apenas do tempo de percurso do robô, mas também do tempo para realizar a coleta dos dados de cada sensor, e do número de sensores atribuídos a cada robô. Já o segundo tem o objetivo principal de minimizar a distância percorrida pelo robô. Porém consideram sempre um mesmo conjunto de nós a serem visitados. Essa dissertação trata de um problema onde o caminho também deve ser minimizado, porém o conjunto de nós pode variar para contribuir com a minimização desejada.

Ambos os trabalhos mostram uma relação entre o DGP e o Problema do Caixeiro Viajante com Vizinhança (Traveling Salesman Problem with Neighborhoods - TSPN). Com isso, suas simulações não fazem com que o robô visite exatamente a posição do nó sensor. Bhadauria et al. [2011] se baseiam em algoritmos do trabalho de Mitchell [2007]. O trabalho de Dumitrescu & Mitchell [2003] destaca essa generalização do PCV, denominada PCV com vizinhança, também NP-difícil. Ainda são apresentados algoritmos que garantem um fator constante para o caso de discos de vizinhança iguais na rede.



# Capítulo 3

## Definição do Problema

Nesse capítulo serão apresentadas aspectos importantes para o problema geral de localização, além de definições formais para o problema proposto nessa dissertação. O problema possui duas abordagens diferentes, serão apresentadas as diferenças entre essas abordagens. Em resumo, a primeira abordagem trata do problema de minimização dos âncoras, enquanto a outra trata do problema de minimização do caminho entre os âncoras.

### 3.1 Localização em Redes de Sensores Sem Fio

Localização é o processo pelo qual são determinadas as posições dos sensores de uma rede. É um problema relevante em diversas áreas da computação [Win et al., 2011], e em Redes de Sensores Sem Fio se torna essencial na maioria das aplicações. A localização pode ser obtida através de configuração manual de cada um dos sensores, ou através de equipamentos como o GPS [Hofmann-Wellenhof et al., 2001], porém esses são recursos limitados. Em redes de difícil acesso, por exemplo, a configuração manual pode ser praticamente impossível, devido à dificuldade em alcançar cada nó sensor e definir exatamente sua posição. Ainda devem ser consideradas grandes redes de sensores, com milhares de nós, onde se torna inviável equipar cada um dos sensores com GPS, devido ao custo de implantação desse recurso. É preciso recorrer a outras técnicas que facilitam esse trabalho de localização, tornando o processo mais barato, prático e confiável.

No trabalho de Oliveira [2008] são destacados três componentes importantes para sistemas de localização:

- Estimativa de ângulo e distância

- Cálculo de posição
- Algoritmo de Localização

Nas próximas seções serão apresentados detalhes de algumas técnicas e abordagens para a estimativa de ângulo e distância e para o cálculo de posição. Algoritmos de localização relevantes para o trabalho foram apresentados na Seção 2 - Trabalhos Relacionados.

### 3.1.1 Estimativa de ângulo e distância

Este componente é responsável por calcular de forma estimada a distância e/ou ângulo entre cada par de nós. Essa estimativa será utilizada pelos demais componentes do sistema de localização. No trabalho de Boukerche et al. [2007] são apresentadas técnicas que mostram como essas estimativas podem ser obtidas:

- **RSSI (Received Signal Strength Indicator):** Esse é um dos métodos mais comuns para estimativa de distância. Em teoria, a potência do sinal é inversamente proporcional ao quadrado da distância. Sendo assim pode ser utilizado um modelo conhecido de propagação de sinal de rádio para converter a potência do sinal em distância. É uma técnica de baixo custo, porém com baixa precisão na estimativa, pois vários são os fatores que podem influenciar na potência do sinal.
- **ToA (Time-of-Arrival):** Essa técnica se utiliza do tempo para encontrar a distância. Nesta abordagem calcula-se o tempo que o sinal demora para percorrer a distância entre emissor e receptor. Como a velocidade de propagação do sinal é conhecida e constante, a distância entre emissor e receptor pode ser calculada no receptor. O problema desse tipo de medição é a necessidade de sincronismo entre os nós, além da influência de fatores externos que podem atrapalhar a transmissão de mensagens.
- **TDoA (Time-difference-of-Arrival):** Mais uma técnica baseada em tempo. Pode ter duas abordagens. Na primeira delas é medida a diferença do tempo que um sinal de um nó chega a três ou mais nós. Na segunda é medida a diferença do tempo onde vários sinais são enviados por um único nó a um outro nó.
- **AoA (Angle-of-Arrival):** É possível também medir o ângulo de chegada de sinal em um nó através de antenas direcionais, ou através de vários receptores

uniformemente separados. A principal desvantagem dessa técnica é o alto custo de hardware adicional.

Nesse trabalho não será tratado o problema de estimativa de distância entre os nós. Portanto, esse é um parâmetro de entrada para o algoritmo proposto e é extremamente importante para o seu perfeito funcionamento. As métricas aqui apresentadas são em geral imprecisas, logo para a implementação e aplicação dessa solução em uma Rede de Sensores Sem Fio deve-se considerar esse fato.

### 3.1.2 Métodos de cálculo de posição

#### 3.1.2.1 Trilateração

Em uma trilateração 2D, são conhecidas as posições de três pontos, assim como as distâncias entre cada um desses pontos e o objeto a ser localizado. Para isso, basta encontrar um ponto de interseção entre três círculos. A Figura 3.1 mostra que caso fossem conhecidos apenas os pontos  $r$  e  $s$  e suas respectivas distâncias a  $v$ , existiriam duas possíveis posições para o ponto  $v$ . Porém, como o ponto  $t$  também possui a informação de sua localização e distância a  $v$ , existe apenas um ponto que satisfaz a todas as condições, e o ponto representado por  $v'$  é descartado.

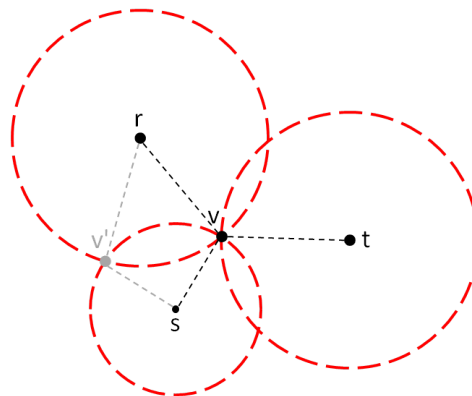


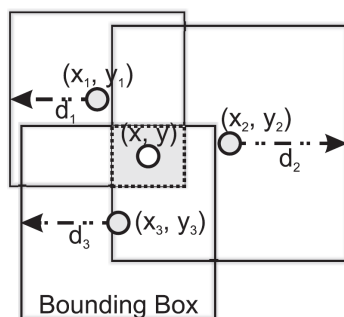
Figura 3.1. Exemplo do método Trilateração

#### 3.1.2.2 Multilateração

A trilateração ocorre com exatamente três pontos de referência. Quando existe um maior número de pontos de referência é possível utilizar a multilateração. Nesse caso é utilizado um sistema de equações sobredeterminado, aumentando a complexidade dessa técnica. A vantagem desse método é a melhora da precisão da solução [Erol et al., 2008].

### 3.1.2.3 Bounding Box

O método *Bounding Box* proposto por Simic & Sastry [2002] utiliza quadrados - ao invés de círculos como ocorre na trilateração - para limitar as possíveis posições de um nó. Para cada nó de referência é definido um quadrado com esse nó no centro, e seus lados de tamanho  $2d_i$ , onde  $d_i$  é a distância estimada do nó ao vizinho em questão. A interseção de todos os retângulos pode ser facilmente calculada. A posição do nó é definida como o centro do retângulo gerado por essa interseção. A Figura 3.2 ilustra esse método. O novo nó a ser localizado possui três vizinhos, dessa forma o quadrado sombreado representa as interseções dos três quadrados que limitam a área onde esse novo nó pode estar. Esse método é simples e rápido, porém a posição estimada não é precisa.



**Figura 3.2.** Exemplo do método Bouding Box. Fonte: Oliveira [2008]

### 3.1.2.4 Triangulação

Triangulação é o método geométrico que permite calcular a posição de um objeto baseado nos ângulos formados entre esse nó e outros nós de posição conhecida. A vantagem deste método é que nenhum tipo de sincronização é necessária para que ele funcione.

### 3.1.2.5 Abordagens probabilísticas

A imprecisão de estimativas de distâncias motivou o aparecimento de abordagens probabilísticas para cálculo de posição de nós. A ideia básica dessa abordagem é que se um nó desconhecido recebe um pacote de um nó de referência ele pode estar em qualquer posição ao redor do nó de referência com igual probabilidade. Ao receber a informação de outro nó, as probabilidades são atualizadas. Quando uma nova informação de posição é recebida a partir de outros nós é possível que o nó desconhecido calcule



sua posição. A principal desvantagem dessa abordagem é o alto custo computacional requerido.

## 3.2 Problema de Minimização de Nós Âncoras

A entrada consiste de um conjunto (rede) de  $n$  nós sensores dispostos em um campo de sensoriamento. A rede de sensores é modelada como um grafo  $G = (V, E)$ , onde  $V$  é o conjunto de  $n$  nós sensores  $v_1, \dots, v_n$  e  $E$  é o conjunto de enlaces. Se um enlace  $v_i v_j \in E$ , a distância entre os nós sensores  $v_i$  e  $v_j$  pode ser medida ou estimada por comunicação sem fio. Todos os nós são idênticos, isso significa que o alcance e a taxa de transmissão são idênticos. É definido um subconjunto  $B \subset V$  de nós âncoras, cuja localização deve ser previamente conhecida. As coordenadas dos demais nós não são previamente conhecidas, porém assume-se que não existem três sensores colineares. Os nós sensores que não pertencem ao conjunto de nós âncoras dependerão da medição das distâncias em  $E$  e da posição dos nós  $\in B$  para determinarem as suas posições durante o procedimento. É importante destacar que para um nó determinar sua posição são necessários três vizinhos que já possuem posição conhecida, para então utilizar a trilateração. Após a definição da sua posição, este nó passa a atuar como âncora e pode auxiliar seus vizinhos no cálculo de suas posições.

O problema pode ser definido da seguinte forma:

### 3.2.1 Definição do problema

**Problema de Minimização de Nós Âncoras:** *Dada uma rede de nós sensores  $G$ , encontrar um subconjunto de nós sensores  $B$  para serem nós âncoras tal que (1) todos os nós sensores possam ser localizados e suas posições possam ser calculadas, dado o grafo, o comprimento de todos os enlaces, e as posições de todos os nós âncoras, e (2) o número total de âncoras  $|B|$  seja minimizado.*

Sendo assim, esse trabalho propõe um algoritmo genético para o problema de localização de custo mínimo. É fácil perceber que o problema sempre tem uma solução viável, uma vez que no pior caso, cada sensor é selecionado como nó âncora, ou seja,  $B = V$ . No entanto, encontrar a melhor solução,  $B_{opt}$ , é um problema NP-Completo [Aspnes et al., 2006]. É óbvio que todos os sensores com grau menor que 3 devem ser incluídos em  $B$ . Quando um sensor possui apenas dois vizinhos, ele não pode determinar sua localização a partir de outros. Seja  $V_{<3}$  a união desses sensores, então  $|V_{<3}|$  é obviamente um limite inferior para  $|B_{opt}|$ . Por outro lado, se todos os nós da rede possuem grau maior ou igual a 3, o limite superior pode ser dado pelo tamanho

do conjunto 3-dominante mínimo  $M3DS_{opt}$  de  $V$ . Nesse caso, o conjunto  $k$ -dominante mínimo de  $V$  (denominado por Minimum  $k$ -Dominating Set, ou  $MkDS$ ) é um subconjunto de nós tal que (1) todo nó em  $MkDS_{opt}$  deve ter  $k$  vizinhos em  $MkDS_{opt}$ ; e (2)  $|MkDS_{opt}|$  é minimizado. Encontrar o  $MkDS$  é um problema NP-difícil, sendo assim, a análise da solução ótima do MCLP é extremamente difícil [Huang et al., 2011]. A melhor aproximação conhecida para o problema  $MkDS$  é  $\max\{\frac{5}{k}, 1\}$  do trabalho de Shang et al. [2008].

### 3.3 Problema de Minimização de Caminho

A entrada consiste de um conjunto de  $n$  nós sensores dispostos em um campo de sensoriamento. Um robô é encarregado de percorrer o campo visitando um subconjunto dos nós, afim de descobrir a localização dos mesmos, sendo que esses sensores passarão a atuar como âncoras. A partir dos nós âncoras deve ser possível descobrir a localização de todos os outros nós. O caminho do robô entre os nós âncoras deve ser o menor possível. Da mesma forma que a definição anterior (Seção 3.2), a rede de sensores é modelada como um grafo  $G = (V, E)$ , onde  $V$  é o conjunto de  $n$  nós sensores  $v_1, \dots, v_n$  e  $E$  é o conjunto de enlaces. Se um enlace  $v_i v_j \in E$ , a distância entre os sensores  $v_i$  e  $v_j$  pode ser medida ou estimada por comunicação sem fio. Todos os nós são idênticos. Isso significa que o alcance e a taxa de transmissão são idênticos. As coordenadas dos nós não são previamente conhecidas. Porém assume-se que não existem três sensores colineares. Um subconjunto de sensores  $B \subset V$  são nós âncora, cuja localização deve ser definida pelo robô. Os demais sensores dependerão da medição das distâncias em  $E$  e da posição dos nós  $B$  para determinar suas localizações durante o procedimento. É importante destacar que para um nó determinar sua posição são necessários três vizinhos que já possuem posição conhecida, para então utilizar a trilateração. Após a definição da sua posição, este nó passa a atuar como âncora e pode auxiliar seus vizinhos no cálculo de suas posições.

Então o problema pode ser formalmente definido como:

#### 3.3.1 Definição do problema

**Problema de Minimização de Caminho:** Dada uma rede de sensores  $G$ , encontrar um subconjunto de nós sensores  $B$  para serem nós âncoras e definir um percurso de um robô para visitar esses nós tal que (1) todos os nós sensores possam ser localizados e suas posições possam ser calculadas, dado o grafo, o comprimento de todas as arestas

*(distâncias entre os nós), e as posições de todos os nós âncoras, (2) o caminho do robô seja minimizado.*

Sendo assim, esse trabalho também propõe um algoritmo genético para o problema de caminho mínimo para localização da rede de sensores.

Para essa abordagem do problema é possível perceber que existe uma grande semelhança com o Problema do Caixeiro Viajante, que procura pelo trajeto de menor distância em um conjunto de cidades, visitando todas elas exatamente uma vez. A diferença nesse caso é que a rede possui vários nós que não precisam ser visitados pelo robô. Porém o menor caminho deve ser definido apenas entre os nós âncoras. Basta então considerar um grafo contendo apenas os âncoras e a distância entre cada par de nós. Dessa forma o problema se torna exatamente o mesmo do caixeiro viajante, que é um problema NP-difícil e possui várias soluções eficientes conhecidas [Applegate et al., 2007].



# Capítulo 4

## Definição dos Algoritmos

Diferentes estratégias foram utilizadas para resolver o problema de localização de custo mínimo. Nesse capítulo serão apresentados todos os algoritmos implementados para essa dissertação. Primeiramente são mostrados aspectos gerais do algoritmo guloso que serviu de base para o algoritmo genético que é apresentado em seguida. Após a apresentação dos algoritmos, são discutidos os detalhes que diferenciam cada abordagem do problema na modelagem proposta.

### 4.1 Algoritmo Guloso

O algoritmo genético proposto nesse trabalho se utiliza do algoritmo guloso apresentado no trabalho de Huang et al. [2011]. Para explicar o algoritmo de uma forma mais clara, foram definidos status para cada nó sensor usando 3 cores: branco, preto e verde. Um nó branco é um nó cuja posição ainda não é possível de ser descoberta. Um nó preto representa um nó âncora. E finalmente, um nó verde é um nó não-âncora, mas com posição possível de ser obtida através de métodos de localização. Inicialmente todos os nós são brancos. O propósito nesse caso é encontrar o menor número de nós pretos de forma que seja possível descobrir a posição de todos os outros nós, colorindo-os de verde.

O algoritmo guloso é baseado no método da trilateração, explicada na Seção 3.1.2.1. Nesse caso, a posição de um nó pode ser calculada se este possui três vizinhos com posições previamente conhecidas, ou possíveis de serem obtidas. Para cada nó  $v$ , é mantido um rank  $r(v)$  que indica o número de vizinhos desse nó com posição conhecida. Quando um nó é marcado pelo algoritmo como preto ou verde, o rank de todos os seus vizinhos são atualizados, e estes empregam a trilateração para tentar localizar outros nós. Quando um nó  $v$  possui 3 vizinhos localizados, pode ser marcado como verde. O

procedimento para definir a cor do nó pode ser feito recursivamente, como mostrado no Algoritmo 1.

---

**Algoritmo 1:** MARK( $u, cor$ )
 

---

```

1  $s(u) = cor$ ;
2 for todo  $v$  vizinho de  $u$  e  $s(v) = branco$  do
3    $r(v) = r(v) + 1$ 
4 if qualquer  $v$  vizinho branco de  $u$  e  $r(v) \geq 3$  then
5   if  $cor = preto$  ou  $verde$  then
6      $MARK(v, verde)$ ;
7   if  $cor = azul$  then
8      $MARK(v, azul)$ ;

```

---

O algoritmo guloso é construtivo e a ideia básica é a seguinte: (1) Todos os nós com grau menor que três são coloridos como pretos, pois não podem ser localizados a partir de outros nós. (2) Em cada passo, é selecionado de forma gulosa o nó branco que mais pode beneficiar o procedimento de localização se for marcado como preto, e então colore-se este nó de preto. O algoritmo termina quando não existem mais nós brancos. O procedimento completo é mostrado no Algoritmo 2.

---

**Algoritmo 2:** Algoritmo Guloso para Localização
 

---

```

1 for cada  $v \in V$  do
2    $s(v) = branco$  e  $r(v) = 0$ ;
3 for cada  $v \in V$  do
4   if o grau de  $v \leq 2$  then
5      $MARK(v, preto)$ ;
6 while  $\exists v$  tal que  $s(v) = branco$  do
7    $u = SELEÇÃO-GULOSA$ ;
8    $MARK(u, preto)$ ;

```

---

Para definir qual o próximo nó branco a ser colorido de preto deve-se analisar qual dos nós trará uma maior contribuição para a localização rede. Para isso, o algoritmo executa um falso MARK (Algoritmo 1) em cada nó branco  $v$ , colorindo-o como azul e recursivamente outros nós também como azuis usando trilateração. É contado o número de nós azuis marcados pelo nó  $v$ , e este número é representado por  $c(v)$ . O nó com maior  $c(v)$  da rede será selecionado como o próximo nó âncora. Esse procedimento é mostrado no Algoritmo 3.



outros nós. Já na Figura 4.1(b) existem duas combinações possíveis. Nesse caso, ainda não é possível determinar as posições desses dois nós. O trabalho de Goldenberg et al. [2006] apresenta um estudo detalhado sobre algoritmos de varredura para localização em Redes de Sensores Sem Fio. O procedimento utilizado nesse trabalho é mostrado no Algoritmo 4.

---

**Algoritmo 4:** MARK2( $u, cor$ )

---

```

1 Todas as linhas (Linhas 1-8) do Algoritmo 1 alterando MARK para MARK2.
  if qualquer  $v$  vizinho branco de  $u$  e qualquer  $w$  vizinho branco de  $v$  satisfazem
   $r(v) = r(w) = 2$  then
2   if ambos  $v$  e  $w$  tem únicas posições para garantir a consistência da
   distância medida then
3     if  $cor = preto$  ou  $verde$  then
4       MARK2( $v, verde$ );
5       MARK2( $w, verde$ );
6     if  $cor = azul$  then
7       MARK2( $v, azul$ );
8       MARK2( $w, azul$ );

```

---

## 4.2 Algoritmo Genético

A solução proposta nesse trabalho é baseada em Algoritmos Genéticos. Os Algoritmos Genéticos (AG) foram concebidos a partir da analogia com a seleção natural, sendo portanto um dos ramos da computação evolutiva. A computação evolutiva originou-se a partir da década de 50 e se baseia na evolução natural das espécies. John Holland e sua equipe desenvolveram e aperfeiçoaram a ideia, formalizando os algoritmos genéticos. Baseando-se na Teoria da Evolução de Darwin, os AGs possuem uma forma de funcionamento como na evolução de espécies de seres vivos e, através do estudo do Darwinismo, são feitas algumas abstrações análogas para mecanizar a ideia.

Cada uma das possíveis soluções do problema é tratada como um indivíduo, pertencente a uma população. A cada geração, alguns indivíduos são selecionados de maneira probabilística e sofrem operações genéticas, de maneira análoga à seleção natural. A evolução desses indivíduos, causada pelas operações realizadas sobre eles, é o principal fator de modificação e de eficácia do AG. A inteligência do método provém da maneira como são selecionados quais indivíduos sofrerão as operações genéticas: quanto melhor for o indivíduo, maior a chance de ele ser selecionado.



O processo de evolução, baseado na seleção, cruzamento e mutação é repetido até que uma condição de parada seja atingida. Essa condição pode ser um limite no número de gerações, ou um valor satisfatório para a solução encontrada, entre outras.

A *fitness* é o atributo do indivíduo que indica o quão adequado ele é para a solução do problema. Para o valor da *fitness* podemos ter as seguintes situações: se um problema é de minimização (problema de minimizar os custos totais de um determinado processo, por exemplo), quanto menor o valor da *fitness*, melhor é esse indivíduo como solução para o problema. No entanto, se o problema é de maximização (por exemplo, problema de maximizar o lucro de uma organização), o valor da *fitness* é proporcional ao quanto ele é bom como solução para o problema. Nesse trabalho foram implementados dois diferentes modos de cálculo de *fitness*, que são explicados nas Seções 4.3 e 4.4.

Vários parâmetros influenciam na execução de um Algoritmo Genético. Os principais são taxa de cruzamento, taxa de mutação, número de gerações, e número de indivíduos. A taxa de cruzamento indica o quanto os indivíduos serão combinados. A taxa de mutação, por sua vez, indica a probabilidade de que cada indivíduo seja mutado. O número de gerações indica quantas vezes ocorrerá o ciclo do Algoritmo Genético e influencia consideravelmente no resultado pois, assumindo que existe uma diversidade de indivíduos, quanto mais gerações forem criadas melhor será sua solução, visto que é verificado um maior número de soluções no espaço de busca. O número de indivíduos indica quantas soluções serão analisadas em cada geração.

Para o problema em questão foi definido um AG com a estrutura mostrada no Algoritmo 5.

---

**Algoritmo 5: ALGORITMO-GENÉTICO**

---

```
1 Inicializa a população;  
2 while geracaoAtual < limiteGeracoes do  
3   | Avalia cada um dos indivíduos;  
4   | Elege os melhores indivíduos;  
5   | Cria nova população através de cruzamento e mutação;  
6   | geracaoAtual ++;  
7 Avalia novamente cada indivíduo;
```

---

É importante ressaltar que a modelagem apresentada nesse trabalho garante que o algoritmo não trabalhe com soluções inviáveis, ou seja, soluções onde podem existir nós não descobertos. Isso acontece porque, para cada solução, caso ainda existam nós não descobertos, estes são transformados em âncora no momento da avaliação do indivíduo.

Cada indivíduo foi considerado como uma sequência de 0 e 1, de tamanho igual ao número de nós, onde 1 indica que o nó é âncora e 0 caso contrário. A população inicial foi criada com 1% de indivíduos gerados a partir do algoritmo guloso e os 99% restantes de forma não-determinística.

Em resumo, o algoritmo segue os seguintes passos:

- Inicialmente é definida uma população inicial. Aqui foi utilizada a estratégia não-determinística, combinada com uma pequena parte da população gerada por algoritmo guloso.
- Avalia-se toda a população de indivíduos segundo o critério determinado pela função de *fitness*.
- Em seguida, através do operador de seleção, escolhem-se os indivíduos de melhor valor como base para a criação de um novo conjunto de possíveis soluções, que será a próxima geração.
- Esta nova geração é obtida aplicando-se sobre os indivíduos selecionados operações que misturem suas características, através dos operadores de cruzamento e mutação.
- Estes passos são repetidos até que o número predeterminado de passos seja atingido.

#### 4.2.1 Seleção por torneio

Nesse trabalho, a seleção é utilizada toda vez que acontece um cruzamento ou mutação. Essa seleção foi realizada utilizando o método do torneio. Nesse método são selecionados aleatoriamente  $k$  indivíduos e é escolhido o melhor dentre eles. No caso, foi definido  $k = 5$ , e para escolher o melhor entre os  $k$  indivíduos é consultada a *fitness* de cada um deles. O trabalho de Blickle & Thiele [1995] apresenta uma análise matemática e maiores detalhes sobre este método de seleção.

#### 4.2.2 Cruzamento ponto a ponto

Para realizar o cruzamento foi definida uma taxa de probabilidade de troca de informação entre dois indivíduos. Em outras palavras, são gerados dois diferentes filhos a partir de dois pais, trocando-se alguns genes de acordo com a probabilidade definida. A Figura 4.2 ilustra esse passo.

Os dois pais são escolhidos através da seleção descrita na Seção 4.2.1.

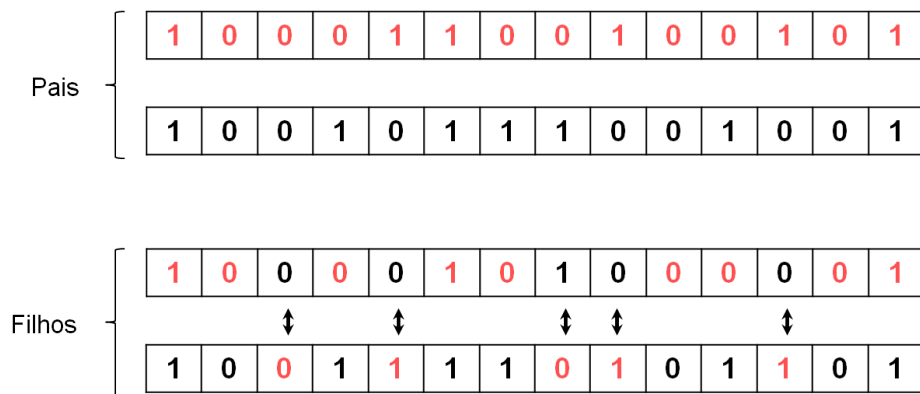


Figura 4.2. Exemplo de cruzamento

### 4.2.3 Mutação

Finalmente, a mutação é realizada de acordo com uma taxa de mutação, onde o nó que é âncora passa a ser não âncora e vice-versa. O indivíduo a ser mutado também é escolhido através da seleção descrita na Seção 4.2.1. Assim como o cruzamento, a mutação também ocorre ponto a ponto, como mostrado na Figura 4.3

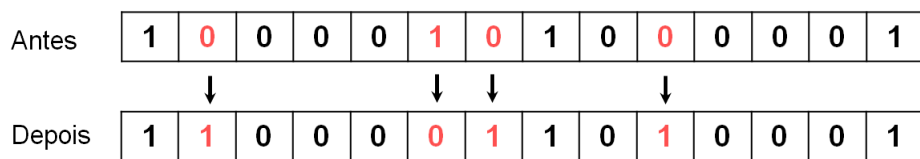
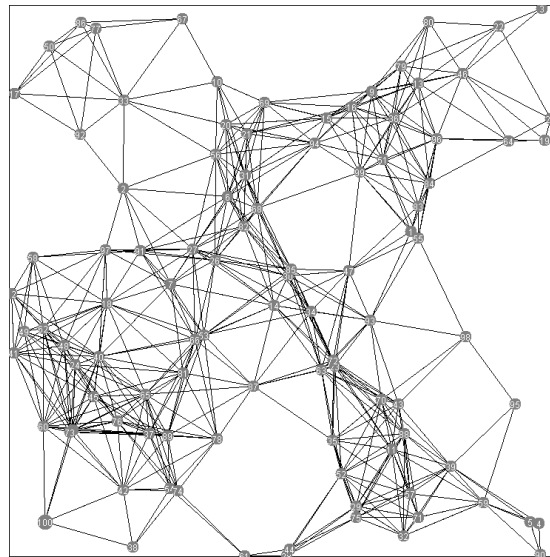


Figura 4.3. Exemplo de mutação

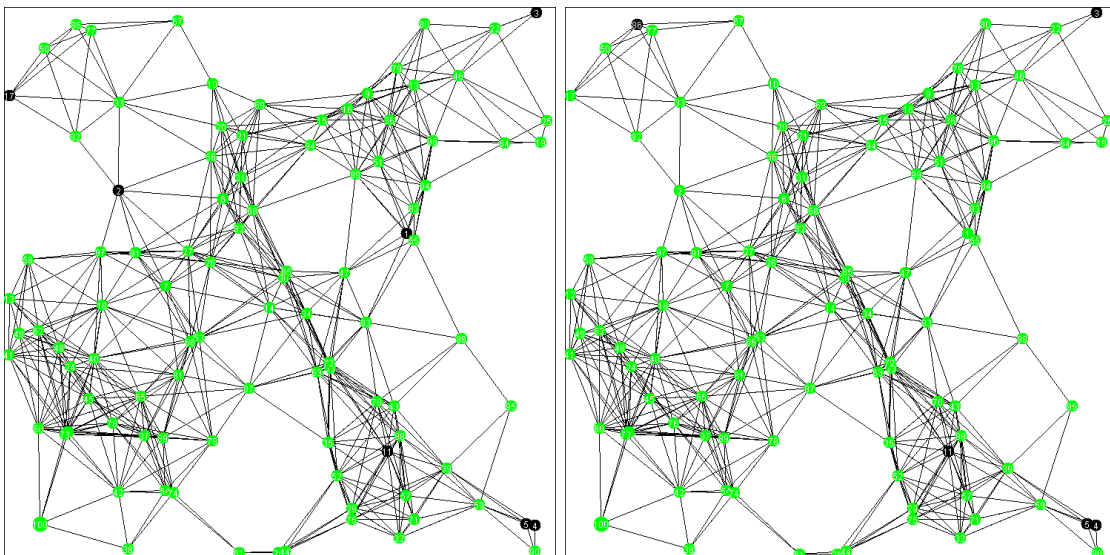
## 4.3 Problema de Minimização de Nós Âncoras

Como definido na Seção 3.2, o objetivo desse trabalho é encontrar o menor número possível de nós âncoras necessários para localizar toda a rede. A Figura 4.4 ilustra o problema.

Na Figura 4.4(a) é mostrada uma rede de sensores simulada no Sinalgo [Zurich, 2013] antes da execução de qualquer algoritmo de localização com custo mínimo. É simulada uma rede com 100 nós, em um espaço de 300x300, e com raio de comunicação de cada nó sendo 60. É possível perceber que essa rede pode ser representada por um grafo conexo. Na Figura 4.4(b) é mostrado o resultado da execução do algoritmo guloso sem varredura local apresentado, que encontrou uma solução com 7 nós âncoras. Já na Figura 4.4(c) é mostrado o resultado da execução do algoritmo genético proposto



(a) Rede de Sensores antes da execução dos algoritmos



(b) Solução do Algoritmo Guloso (7 nós âncoras)

(c) Solução do Algoritmo Genético (5 nós âncoras)

**Figura 4.4.** Exemplo de solução para minimização de âncoras

nessa dissertação com trilateração sem varredura local, que encontrou uma solução com apenas 5 nós âncoras.

Para resolver esse problema foi implementado o algoritmo genético (Algoritmo 5) detalhado na seção anterior com alguns detalhes específicos.

A métrica de avaliação utilizada foi o número de nós âncoras na solução. No caso, quanto menor esse valor, melhor é o indivíduo. A função de *fitness* é definida

pela equação 4.1.

$$fitness_i = \sum_{j=1}^{nNos} n_{ij} \quad (4.1)$$

Na equação 4.1,  $fitness_i$  representa a *fitness* do indivíduo  $i$ . A variável  $n_{ij}$  possui sempre o valor 0 ou 1 e representa o nó  $j$  do indivíduo  $i$ , sendo que  $j$  varia de 1 a  $nNos$ , e  $nNos$  representa o número de nós. Sendo assim, basta somar esses valores para obter o número de âncoras da solução e posteriormente comparar a qualidade com outras soluções.

Foram detalhados dois métodos de cálculo de posição dos nós (Seção 4.1), e cada um deles foi usado separadamente no algoritmo genético para o problema de localização de custo mínimo.

### 4.3.1 Algoritmo Genético com Trilateração

A principal particularidade desse algoritmo genético se deve à maneira de como é realizada a avaliação de cada indivíduo. Para cada indivíduo é executado o método MARK (Algoritmo 1). Dessa forma, é possível verificar se ao final da execução todos os nós são localizáveis. Caso o indivíduo represente uma solução onde não é possível definir a posição de todos os nós, estes serão definidos como nós âncora nessa solução.

Outro detalhe que diferencia esse algoritmo é a geração da população inicial. Nesse caso, a parte da população inicial gerada a partir do algoritmo guloso utiliza também o método MARK, ou seja, utiliza apenas a trilateração, sem varredura local.

### 4.3.2 Algoritmo Genético com Trilateração e Varredura Local

O algoritmo com trilateração e varredura local segue os mesmos princípios do algoritmo que utiliza apenas a trilateração. Porém, se diferenciam novamente na avaliação dos indivíduos. Para cada indivíduo é executado o método MARK2 (Algoritmo 4). Esse algoritmo se mostra mais eficiente na localização dos nós. Assim como antes, caso não seja possível descobrir a posição de um nó, este se torna âncora. Além disso, caso um nó seja considerado como âncora pela solução, mas durante a execução o algoritmo perceba que este nó já possui a capacidade de determinar sua posição a solução é alterada e este nó deixa de ser âncora.

A população inicial gerada pelo algoritmo guloso utiliza nesse caso o método MARK2, sendo beneficiada pela varredura local, além da trilateração

## 4.4 Problema de Minimização de Caminho

Relembrando a Seção 3.3, esse problema procura encontrar o menor caminho entre os nós de um conjunto de âncoras. A principal restrição é que toda a rede seja localizada, sem se importar com o número de âncoras necessários. É importante ressaltar que para esse tipo de problema devem ser consideradas como entrada apenas redes de sensores que resultem em grafos conexos<sup>1</sup>.

Nesse caso é necessário alterar a métrica de avaliação dos indivíduos do algoritmo genético. Já que o objetivo é minimizar o caminho entre os nós é preciso determinar uma heurística que encontre de forma prática e rápida um caminho que passe por todos os nós âncoras da solução. Esse subproblema se assemelha ao já bem conhecido Problema do Caixeiro Viajante (PCV).

Podemos então definir que a nova maneira de se obter a *fitness* de um indivíduo do algoritmo genético é a utilização de uma heurística, como mostrado na equação 4.2.

$$fitness_i = HeuristicaTSP(B) \quad (4.2)$$

A maioria das estratégias de solução para problemas como o PCV utiliza como dado de entrada um grafo completo, onde todos os nós possuem a distância conhecida para todos os outros nós. Além disso, todos os nós do grafo devem ser visitados para a solução do PCV. Para definir o caminho na rede de sensores considerada é necessária uma pequena adaptação no grafo que a representa, e que será utilizado no processamento. Serão considerados apenas os nós âncoras, mesmo que os outros funcionem como um elo de ligação entre dois nós âncoras. Para cada par de nós âncoras é calculado, utilizando o algoritmo de Dijkstra [Cormen et al., 2005], a menor distância utilizando todos os nós da rede. E assim é formado um grafo completo com todas as arestas conhecidas, para ser utilizado no algoritmo que determina o menor caminho. Com esse grafo é calculado o caminho utilizando uma heurística do vizinho mais próximo. Primeiramente é selecionado um nó aleatoriamente e a partir de uma heurística novamente construtiva e gulosa é selecionado o próximo nó observando todas as possibilidades de caminho para cada vizinho que não fecham um ciclo e selecionando a menor delas. Essa é uma das heurísticas mais simples para a resolução do PCV.

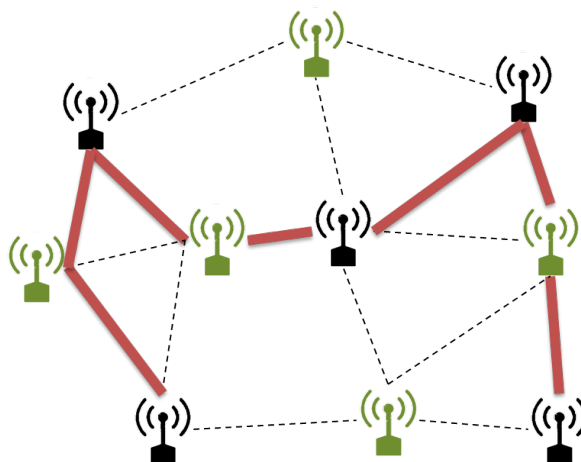
Outras heurísticas podem ser implementadas e facilmente utilizadas pelo algoritmo genético. Porém é necessário considerar que nesse momento do processamento as posições dos nós não são conhecidas. A única informação que se tem é a distância entre os nós vizinhos. Dessa forma não é possível, por exemplo, utilizar algumas

---

<sup>1</sup>Um grafo é conexo se há pelo menos um caminho entre cada par de vértices deste grafo.

heurísticas de inserção conhecidas, pois seria necessário calcular o custo de inserção de cada nó na construção do circuito, e para isso é preciso conhecer a posição de todos os nós. Além disso, heurísticas mais sofisticadas geralmente demandam um tempo maior de execução e o Algoritmo Genético também gasta um maior tempo para alcançar a solução com relação a outras heurísticas como algoritmos gulosos. Por isso, para essa dissertação foi escolhida a heurística do vizinho mais próximo, já explicada, que é mais simples e rápida.

A Figura 4.5 ilustra uma rede onde os nós pretos são os nós âncoras e os verdes são os nós que serão localizados a partir dos demais. A linha vermelha ilustra um possível caminho para o robô entre os âncoras. Dessa forma, a cada visita o robô define a posição do nó e segue para o próximo nó. Ao final do trajeto do robô será possível definir a posição de todos os nós através de cálculos que utilizam a trilateração.



**Figura 4.5.** Exemplo de solução para minimização do caminho

Novamente, cada um dos métodos de cálculo de posição de nós descrito foi utilizado separadamente no algoritmo genético, dessa vez, para o problema de definição da trajetória de âncoras móveis para localização com custo mínimo.

#### 4.4.1 Algoritmo Genético com Trilateração

Além do cálculo da *fitness*, outro detalhe que diferencia esse algoritmo dos demais é o uso desse método de avaliação em conjunto com o Algoritmo 1, MARK. O algoritmo MARK fica responsável por validar a solução, conferindo se os nós âncoras definidos são suficientes para localizar toda a rede, enquanto a heurística para encontrar o caminho fica responsável por avaliar a qualidade da solução.

A parte da população inicial gerada por algoritmo guloso também utiliza o método MARK, ou seja, utiliza apenas a trilateração, sem varredura local.

#### 4.4.2 Algoritmo Genético com Trilateração e Varredura Local

Finalmente, esse algoritmo utiliza como cálculo da *fitness* o menor caminho entre os âncoras. Entretanto, para validar a solução, é utilizado o algoritmo guloso MARK2.

A varredura local também é utilizada na geração da população inicial. Sendo assim, a parte da população inicial que é gerada a partir do algoritmo guloso, utiliza o método MARK2.



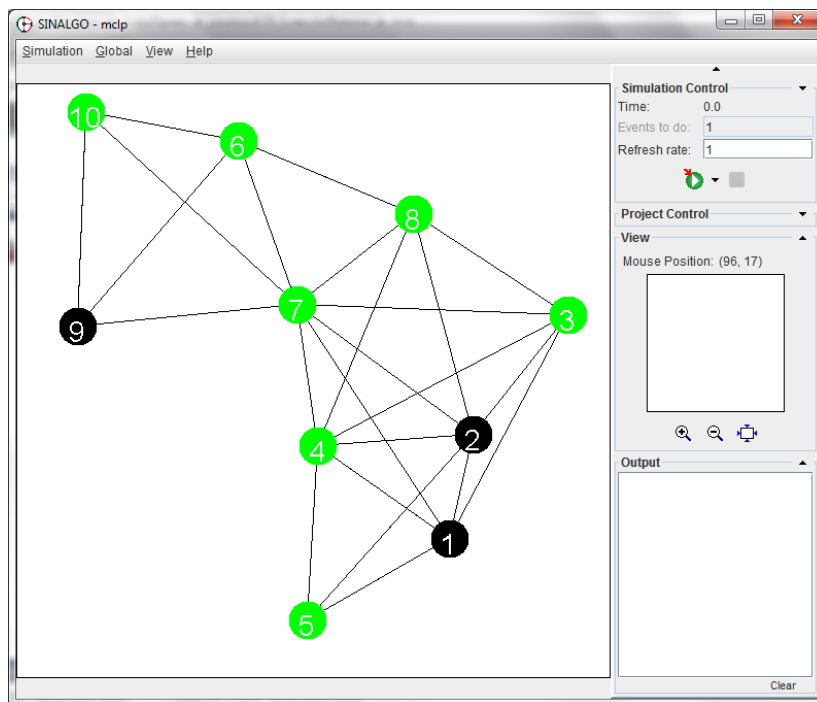
# Capítulo 5

## Experimentos e Resultados

Para realizar testes do algoritmo em diferentes redes foi utilizado o simulador Sinalgo [Zurich, 2013]. Sinalgo é um framework para testes e validações de algoritmos de rede. Foi projetado para as redes sem fio, mas não é limitado apenas a estas, além disso, oferece suporte para redes 2D e 3D. Oferece ainda um amplo conjunto de condições da rede, sendo possível variar a conectividade dos nós, a mobilidade, o modelo de distribuição dos nós, definir como funcionará a interferência na transmissão de mensagens, etc. O projeto é escrito em Java, é gratuito, e publicado sob a licença BSD. A figura 5.1 mostra como é o ambiente de simulação do Sinalgo apresentando uma rede com 10 nós, sendo que os nós 1, 2 e 9 foram considerados os âncoras para essa execução do algoritmo.

Para avaliar o algoritmo proposto, foram conduzidos experimentos em redes de sensores geradas aleatoriamente pelo sinalgo. O algoritmo foi executado em redes com 500, 1000, 1500, 2000, 2500 e 3000 nós distribuídos uniformemente em uma região retangular fixa de 1200 x 1000. Foi definido o raio de comunicação de cada nó com o valor fixo 80. Isso significa que, se a distância entre dois sensores é menor ou igual a 80, os nós são vizinhos e é possível medir a distância entre eles. No simulador essa vizinhança é definida através do modelo de conectividade. O modelo definido foi o UDG (Unit Disk Graph). Esse modelo é puramente geométrico e realiza a conexão entre dois nós exatamente da forma como a vizinhança é proposta pelo problema, ou seja, dois nós estabelecem comunicação se e somente se a distância entre eles é menor que um limite definido. Além disso, cada uma das arestas formadas entre dois nós, indicando que existe uma vizinhança, é bidirecional.

O sinalgo oferece várias outras opções de configuração para gerar redes de sensores e simular algoritmos nessas redes. Diante dessas opções, foi definido que nenhum dos nós possui mobilidade dentro da rede. Isso significa que todos os nós permanecem



**Figura 5.1.** Ambiente do Sinalgo

estáticos, na mesma posição, durante toda a execução do algoritmo.

O algoritmo genético, como explicado na Seção 4.2 tem os seguintes parâmetros:

- Tamanho da população: 100
- Número de gerações: 50
- Taxa de mutação: 30%
- Taxa de cruzamento: 60%
- Taxa gulosa da população inicial: 1% (percentual de indivíduos gerados pelo algoritmo guloso)
- Elitismo: 10 indivíduos

Para os experimentos foram implementados os quatro algoritmos propostos. São apresentadas comparações com o algoritmo guloso do trabalho de Huang et al. [2011]. As métricas de avaliação utilizadas dependem do problema. Para a localização com custo mínimo é considerado o número de nós âncoras na solução de cada algoritmo. Para o problema da trajetória é considerado o menor caminho entre os nós âncoras. Cada um dos experimentos com o algoritmo genético foi repetido 30 vezes e os resultados apresentados são a média dessas 30 execuções. A comparação do algoritmo

genético é feita com o algoritmo guloso considerando uma rede idêntica, com exatamente as mesmas características. O sinalgo permite esse tipo de simulação quando se define que a criação da rede será através de uma semente fixa utilizada para gerar números aleatórios. A semente utilizada para a execução da simulação foi o valor 123456. O algoritmo genético possui uma semente de geração de números aleatórios diferente, definida pelo tempo do sistema. Isso faz com que o resultado seja diferente a cada execução. Essa estratégia foi utilizada para que a comparação entre o algoritmo genético e o algoritmo guloso fosse realizada de forma justa.

## 5.1 Problema de Minimização de Nós Âncoras

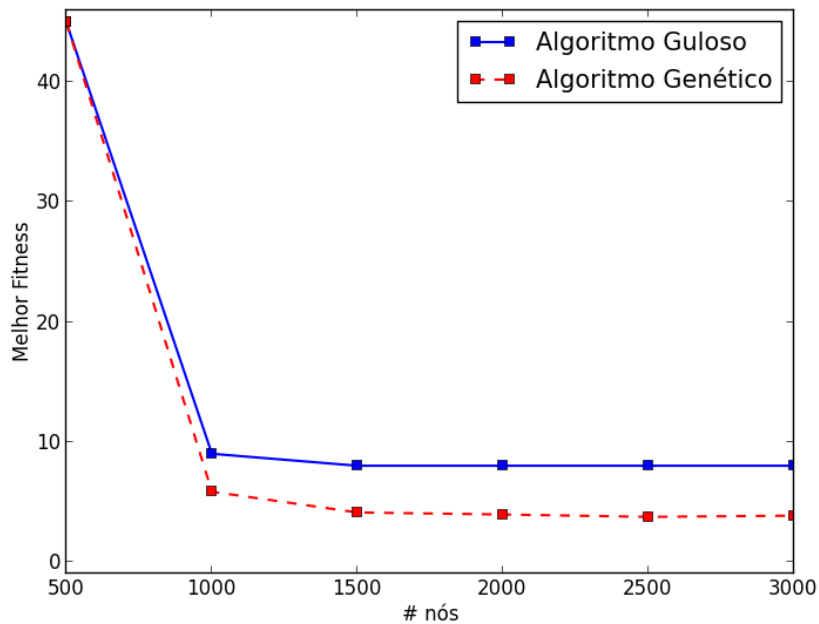
Para esse problema a métrica de qualidade é o número de nós âncoras na solução. Quanto menor esse valor, melhor é a solução. Foram executados testes para a trilateração com e sem varredura local. Os gráficos apresentados a seguir reúnem, de maneira resumida, todos os experimentos realizados.

### 5.1.1 Algoritmo Genético com Trilateração

A Figura 5.2 mostra o resultado final do experimento com os algoritmos genético e guloso para cada uma das redes de sensores onde foram executados. É possível observar que o algoritmo genético proposto alcança sempre no máximo o mesmo resultado que o algoritmo guloso. Isso acontece devido à duas características fundamentais propostas na modelagem. A primeira delas é que uma pequena parte da população inicial é gerada a partir do mesmo algoritmo guloso que aparece na comparação. A segunda característica da modelagem é o elitismo que ocorre entre as gerações de indivíduos. Essa estratégia faz com que os melhores indivíduos sempre sobrevivam. Dessa forma, a melhor solução encontrada pelo algoritmo genético nunca será pior que a solução do algoritmo guloso.

Através da Figura 5.2 ainda é possível observar que à medida que o número de nós aumenta, a tendência é que o número de âncoras necessários diminua, pois a área de distribuição dos nós continua fixa. Além disso, a diferença entre as soluções do algoritmo genético e guloso se destaca em redes com um maior número de nós. O intervalo de confiança dos valores para o algoritmo genético é muito pequeno, o que aumenta a confiabilidade dos resultados.

Outra observação é que o algoritmo genético alcançou resultados bem próximos do ótimo para as redes com 1500, 2000, 2500 e 3000. Sabemos que nenhuma rede é possível de ser localizada se possuir menos de três âncoras. Logo, quando o algoritmo



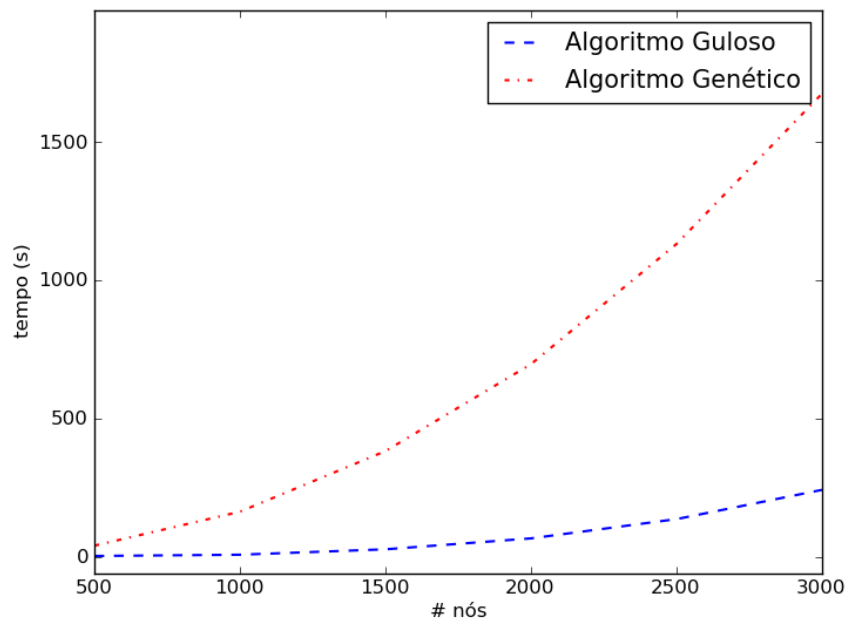
**Figura 5.2.** Resultado da execução para minimização de âncoras sem varredura local

genético alcança esse resultado é possível afirmar que encontrou a solução ótima para o problema. Isso aconteceu em várias execuções do algoritmo genético, fazendo com que a média do número de âncoras do melhor indivíduo ao final das 50 gerações fosse bem próxima de 3, enquanto o algoritmo guloso nessas redes alcançou uma solução com 8 âncoras.

A Figura 5.3 mostra a diferença do tempo de execução entre os algoritmos guloso e genético. Percebe-se que o algoritmo genético precisa de um tempo maior para alcançar sua solução.

A Figura 5.4 mostra a evolução do algoritmo genético durante as 50 gerações para cada uma das redes simuladas, variando apenas o número de nós em cada um dos gráficos. É importante observar que em todos os gráficos foi traçado o resultado do algoritmo guloso que, por ser determinístico, alcança sempre o mesmo resultado, não sendo necessária a apresentação de uma média de várias execuções, como o algoritmo genético. Foi traçada também a média da média de todos os indivíduos, além da média do melhor indivíduo nas 30 execuções.

A Figura 5.4(a) mostra que o resultado não sofre muita variação com o passar das gerações. Isso ocorre porque essa rede possui pouca conectividade entre os nós, fazendo com que a melhor solução seja sempre a mesma encontrada pelo algoritmo

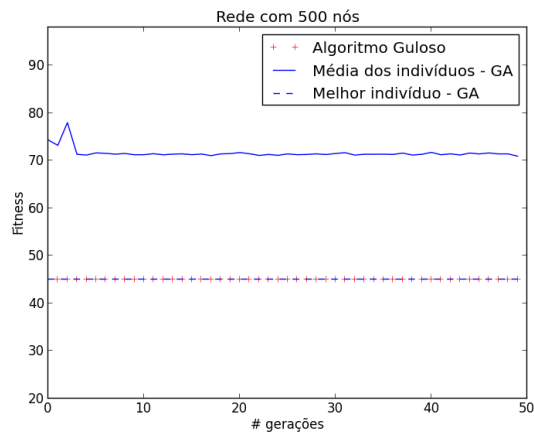


**Figura 5.3.** Tempo de Execução - Minimização de âncoras sem varredura local

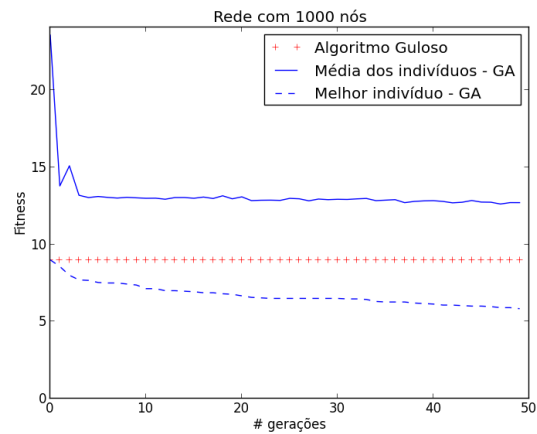
guloso. Apenas a média dos indivíduos sofre variações durante as gerações.

Já as Figuras 5.4(b) - 5.4(f) mostram que o algoritmo genético evolui à medida que passam as gerações. A média da melhor solução começa sempre igual ou próxima ao valor encontrado pelo algoritmo guloso e diminui com o tempo. Nas redes com 2500 e 3000 nós o resultado final do algoritmo alcança inclusive valores bem próximos do ótimo aproximadamente na geração 30.

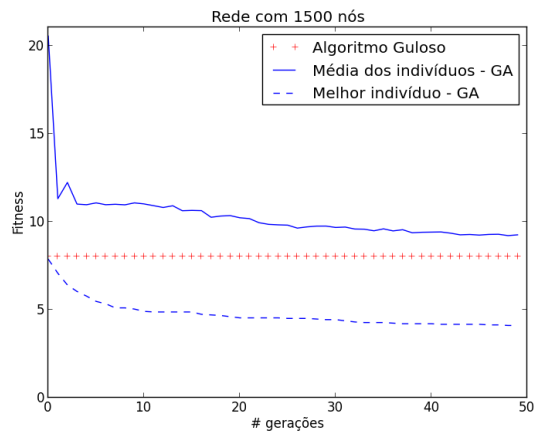
A distância entre os valores da média dos indivíduos e do melhor indivíduo mostra que a população do algoritmo genético representa soluções bem diferentes. Essa heterogeneidade da população é uma vantagem do algoritmo, pois permite que sejam exploradas diversas soluções diferentes, escapando muitas vezes de valores mínimos locais que também satisfazem como solução do problema. Para as redes com 1500, 2000, 2500 e 3000 nós os gráficos mostram que a média dos indivíduos alcança valores próximos da solução encontrada pelo algoritmo genético. Isso mostra que mesmo com a variedade dos indivíduos, ocorre uma evolução geral da população.



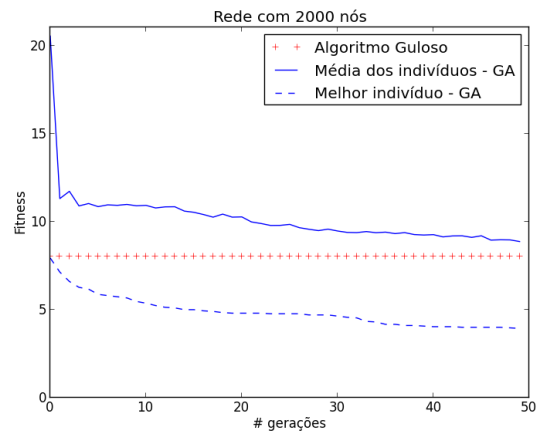
(a) Rede com 500 nós



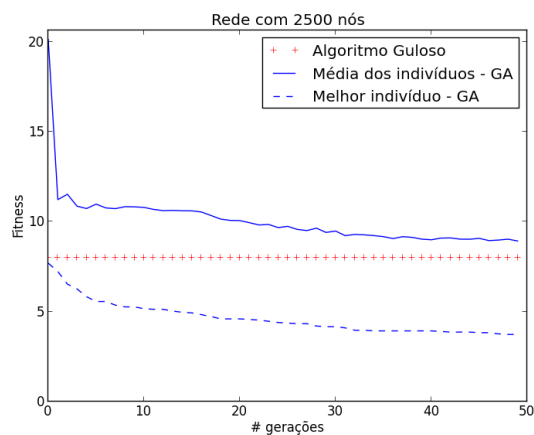
(b) Rede com 1000 nós



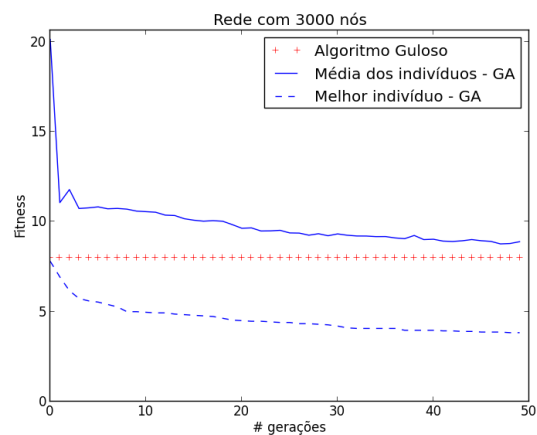
(c) Rede com 1500 nós



(d) Rede com 2000 nós



(e) Rede com 2500 nós



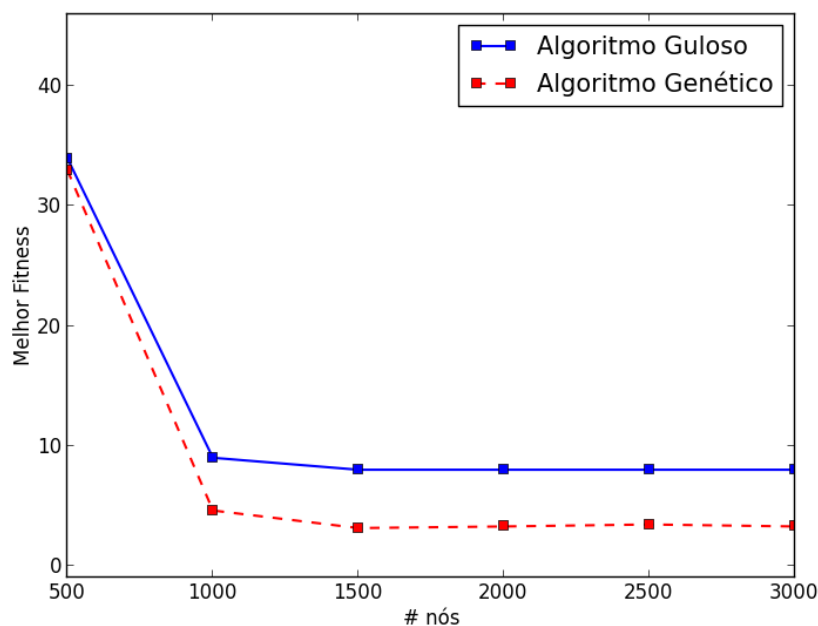
(f) Rede com 3000 nós

**Figura 5.4.** Evolução do Algoritmo Genético para minimização de âncoras sem varredura local

### 5.1.2 Algoritmo Genético com Trilateração e Varredura Local

A Figura 5.5 mostra o resultado final da simulação dos algoritmos genético e guloso para a minimização de âncoras com trilateração e varredura local. Novamente é possível perceber que o algoritmo genético sempre alcança resultados que são iguais ou melhores que o algoritmo guloso, por causa da população inicial e do elitismo, como explicado na seção anterior.

Para o algoritmo com varredura local aplicado na rede com 500 nós a solução teve uma melhora de aproximadamente 25% se comparada com o algoritmo sem varredura local. Essa melhora ocorre porque a varredura garante que vários nós sejam localizados sem a necessidade de possuir três vizinhos com localização já conhecida. Para os demais valores de número de nós a solução não diferencia com a mesma proporção porque são redes com grande conectividade entre os nós, fazendo com que soluções muito boas sejam encontradas pelos dois algoritmos.

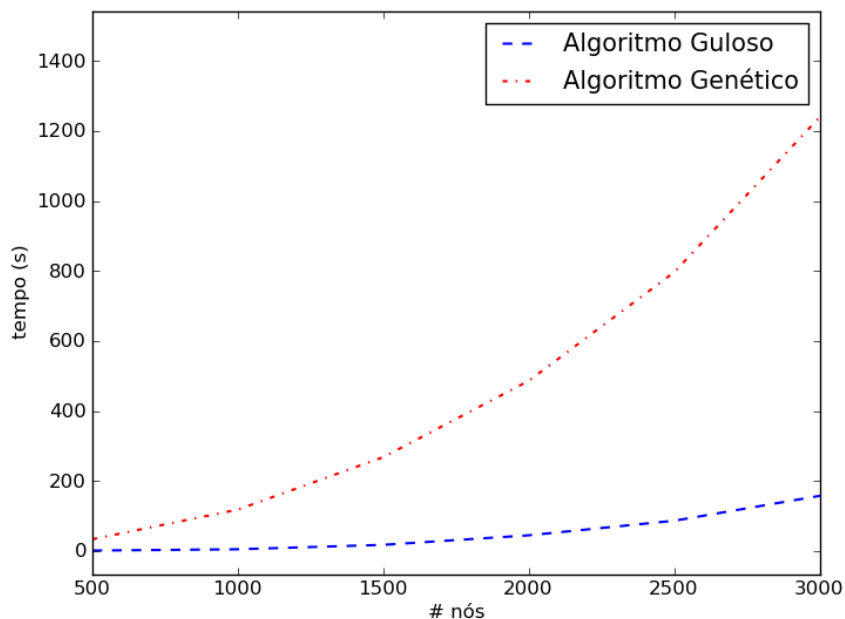


**Figura 5.5.** Resultado da execução para minimização de âncoras com varredura local

Na Figura 5.5 também é possível observar a tendência de diminuição de nós âncoras à medida que se aumenta o número de nós na rede. O aumento do número de nós também deixa mais clara a diferença entre as soluções do algoritmo genético e o algoritmo guloso, pois para as redes com 1500, 2000, 2500 e 3000 o algoritmo genético alcançou resultados bem próximos do ótimo. Logo, podemos afirmar que em

várias execuções a solução encontrada foi a ótima, de apenas 3 âncoras necessários para localizar toda a rede, enquanto o algoritmo guloso encontrou uma solução com 8 âncoras. Novamente, o intervalo de confiança dos valores para o algoritmo genético é muito pequeno, mostrando que os valores dos resultados não possuem grande variação.

A Figura 5.6 mostra a diferença do tempo de execução entre os algoritmos guloso e genético. Assim como para a solução sem varredura local, o algoritmo genético gasta um tempo maior que o algoritmo guloso, porém alcança sempre melhores resultados.



**Figura 5.6.** Tempo de Execução - Minimização de âncoras com varredura local

A Figura 5.7 mostra a evolução do algoritmo genético durante as 50 gerações para cada uma das redes simuladas, variando apenas o número de nós em cada um dos gráficos. Novamente, é importante observar que em todos os gráficos foi traçado o resultado do algoritmo guloso que, por ser determinístico, alcança sempre o mesmo resultado, não sendo necessária a apresentação de uma média de várias execuções, como o algoritmo genético. Foi traçada também a média da média de todos os indivíduos, além da média do melhor indivíduo nas 30 execuções.

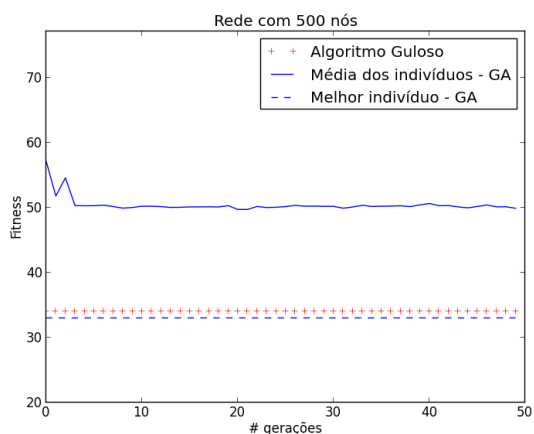
A Figura 5.4(a) mostra um que o algoritmo genético alcança um resultado bem constante, com pouca variação com o passar das gerações. Essa rede possui pouca conectividade entre os nós, fazendo com que a melhor solução seja sempre próxima à solução encontrada pelo algoritmo guloso, com pouca melhora. Apenas a média dos indivíduos sofre variações durante as gerações.



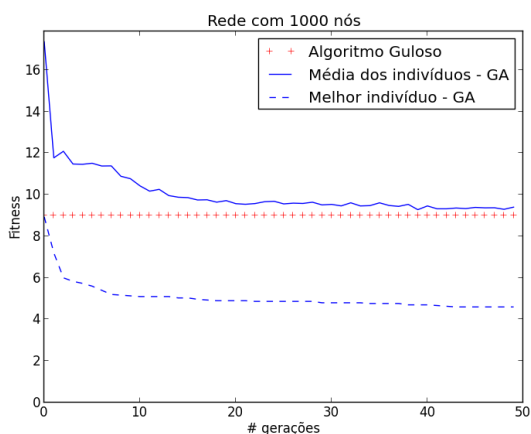
Já as Figuras 5.4(b) - 5.4(f) mostram que o algoritmo genético evolui muito bem à medida que passam as gerações. A média do melhor indivíduo começa sempre igual ou próximo ao valor encontrado pelo algoritmo guloso e diminui com o tempo. Nas redes com 1500, 2000, 2500 e 3000 nós o resultado final do algoritmo alcança inclusive valores bem próximos do ótimo aproximadamente na geração 30. Isso mostra que a varredura local teve uma maior influência no resultado das redes com 500, 1000, 1500 e 2000 nós, fazendo a solução melhorar consideravelmente.

A distância entre os valores da média dos indivíduos e do melhor indivíduo mostra novamente que a população do algoritmo genético representa soluções bem diferentes. Essa diferença entre os indivíduos da população garante que um grande número de diferentes soluções será analisado e avaliado.

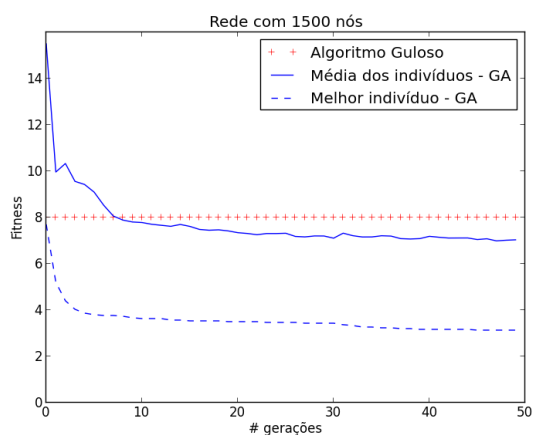
Para as redes com 1500, 2000, 2500 e 3000 nós os gráficos mostram que a média dos indivíduos alcança valores melhores que a solução encontrada pelo algoritmo genético. Esse resultado é importante para o algoritmo, pois boa parte da população é melhor que o algoritmo guloso, o que mostra que realmente é possível conseguir bons resultados através de heurísticas como Algoritmo Genético.



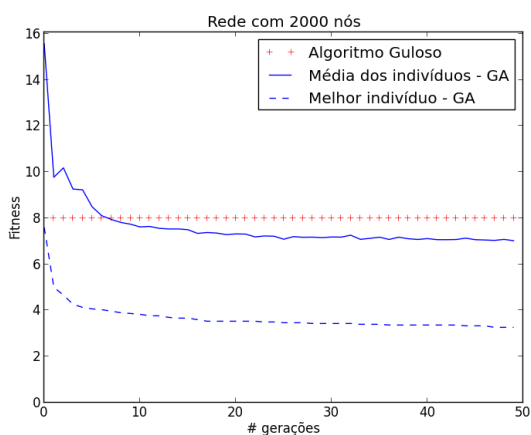
(a) Rede com 500 nós



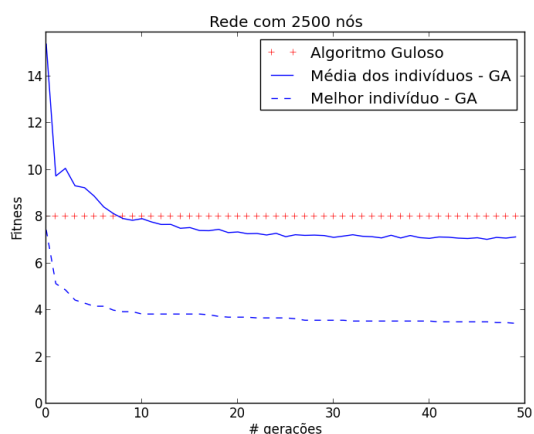
(b) Rede com 1000 nós



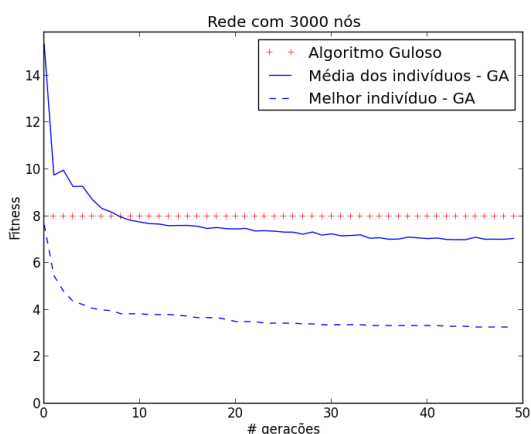
(c) Rede com 1500 nós



(d) Rede com 2000 nós



(e) Rede com 2500 nós



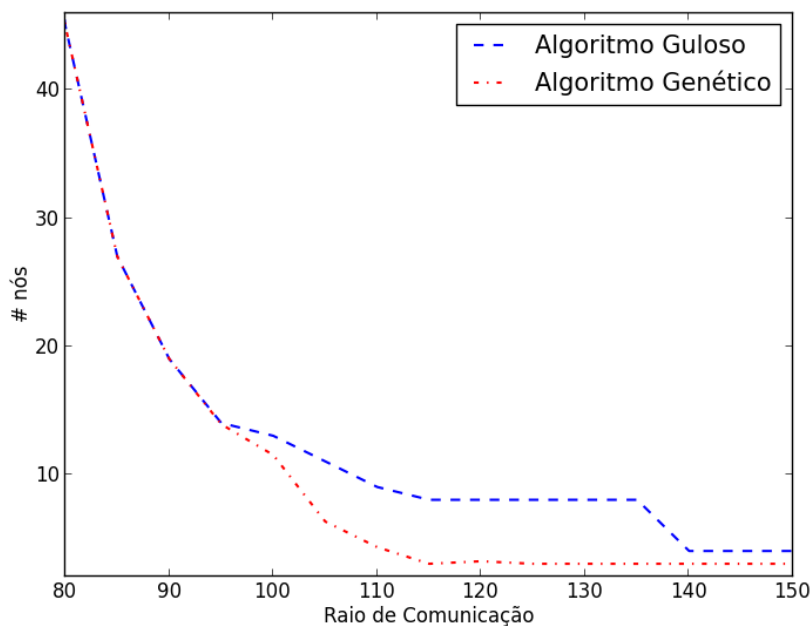
(f) Rede com 3000 nós

**Figura 5.7.** Evolução do Algoritmo Genético para minimização de âncoras com varredura local

### 5.1.3 Impacto da variação do raio de comunicação dos nós

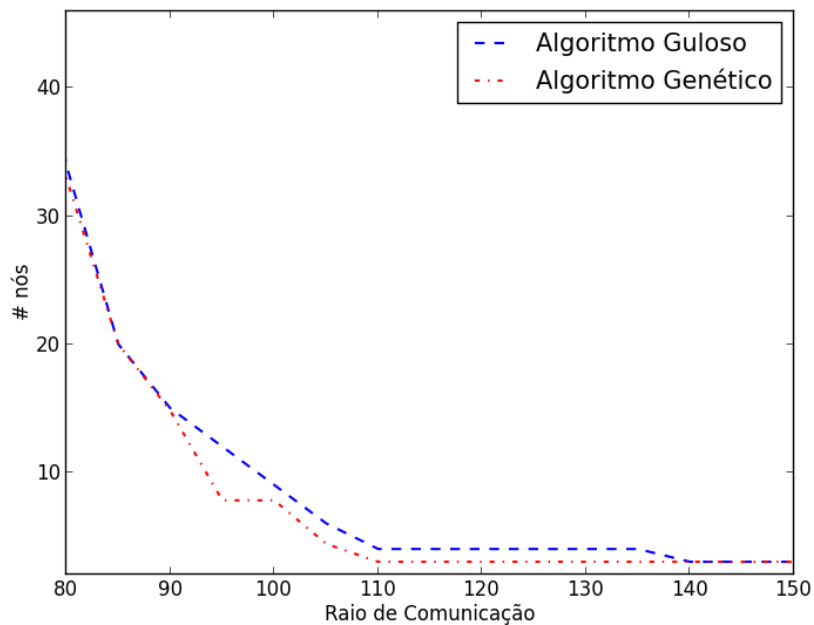
Nesse experimento é mostrado o impacto da variação do raio de comunicação de transmissão dos nós. Todos os parâmetros do algoritmo, da rede e do simulador foram conservados, com exceção do raio de comunicação, que variou de 80 até 150 (de 5 em 5). Quando uma rede mantém o mesmo número de nós nas mesmas posições e aumenta o raio de comunicação, é aumentado o raio de vizinhança de cada nó. Quanto mais vizinhos os nós tiverem em média, mais fácil será para localizar toda a rede. Os testes foram executados em uma rede com 500 nós para os algoritmos com e sem varredura local.

A Figura 5.8 mostra o resultado da execução para os algoritmos guloso e genético sem varredura local. Para os valores 80, 85, 90 e 95 os algoritmos encontraram exatamente a mesma solução. A partir do raio igual a 100 o algoritmo genético se destacou encontrando sempre melhores soluções. Apesar da média do algoritmo genético ser muito boa, e sempre melhor que o algoritmo guloso, percebe-se que para os valores 140, 145 e 150 o algoritmo guloso se aproxima novamente da solução do algoritmo genético. Isso ocorre devido ao fato da rede estar com uma conectividade maior, facilitando a trilateração e fazendo com que seja mais fácil para qualquer algoritmo encontrar boas soluções.



**Figura 5.8.** Impacto da variação do raio de comunicação dos nós sem varredura local

A Figura 5.9 mostra o resultado da execução para os algoritmos guloso e genético utilizando a varredura local. Analisando o gráfico, podemos ver que para os valores 80, 85 e 90 os dois algoritmos são igualmente eficientes, pois existe pouca conectividade entre os nós, fazendo com que existam também poucas soluções. Para os valores 140, 145 e 150 novamente os algoritmos são igualmente eficientes. Porém, dessa vez existe uma grande conectividade entre os nós, colaborando para a trilateração e mais ainda para a varredura local. Isso faz com que os algoritmos encontrem sempre a solução ótima para o problema que é uma solução com 3 âncoras. Para os demais valores, o algoritmo genético alcançou melhores resultados.



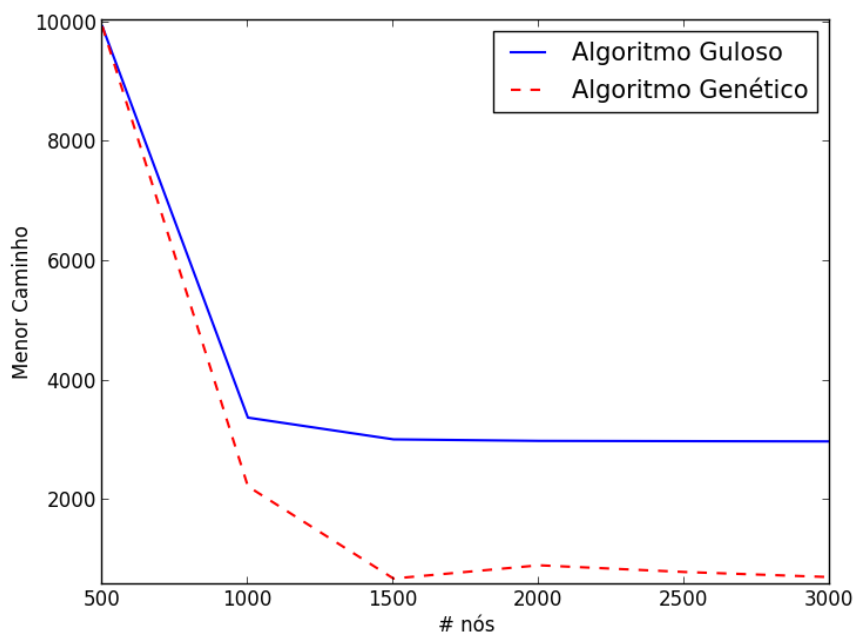
**Figura 5.9.** Impacto da variação do raio de comunicação dos nós com varredura local

## 5.2 Problema de Minimização de Caminho

Para esse problema a métrica de avaliação é o caminho mínimo entre os nós âncoras. Quanto menor o caminho, melhor é a solução. Assim como na abordagem anterior, foram executados testes para os algoritmos que utilizam a trilateração com e sem varredura local. Os gráficos apresentados a seguir reúnem, de maneira resumida, todas as simulações realizadas.

### 5.2.1 Algoritmo Genético com Trilateração

A Figura 5.10 mostra o resultado da simulação dos algoritmos genético e guloso para redes de 500 a 3000 nós. Um fato interessante de ser observado é que a média do melhor indivíduo cresce ligeiramente de 1500 para 2000 nós. Porém nesse caso deve-se observar o intervalo de confiança, que para esse problema se torna um pouco maior.



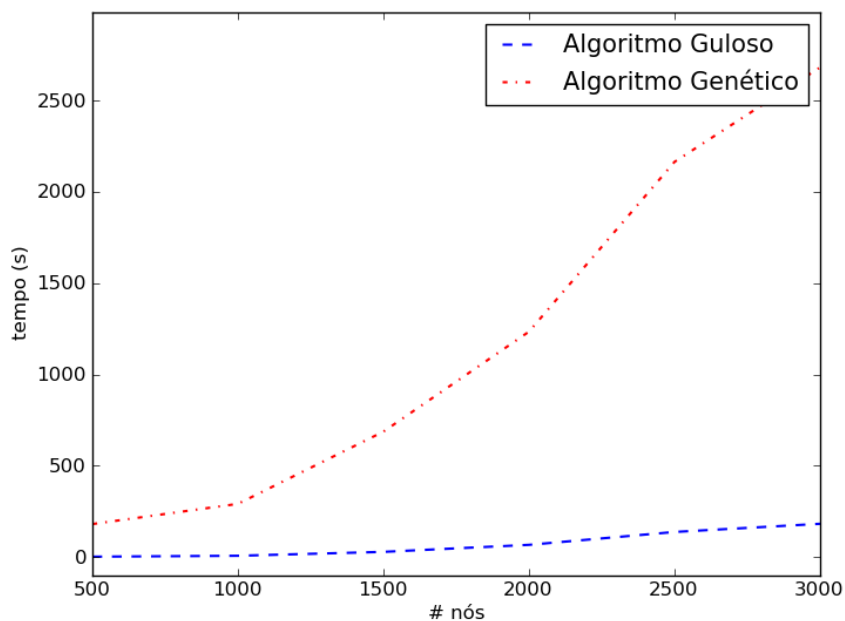
**Figura 5.10.** Resultado da execução para minimização do caminho sem varredura local

É possível observar também que o resultado do algoritmo genético é sempre igual ou melhor que o algoritmo guloso, pelo fato de contar com as estratégias de elitismo e população inicial com indivíduos gerados pelo algoritmo guloso.

Nessa abordagem a diferença entre as soluções dos algoritmos genético e guloso é maior porque o algoritmo guloso não tem como objetivo principal minimizar a distância

entre os nós âncoras. Outro detalhe importante dessa abordagem do problema é que não é trivial encontrar um limite inferior como na abordagem anterior.

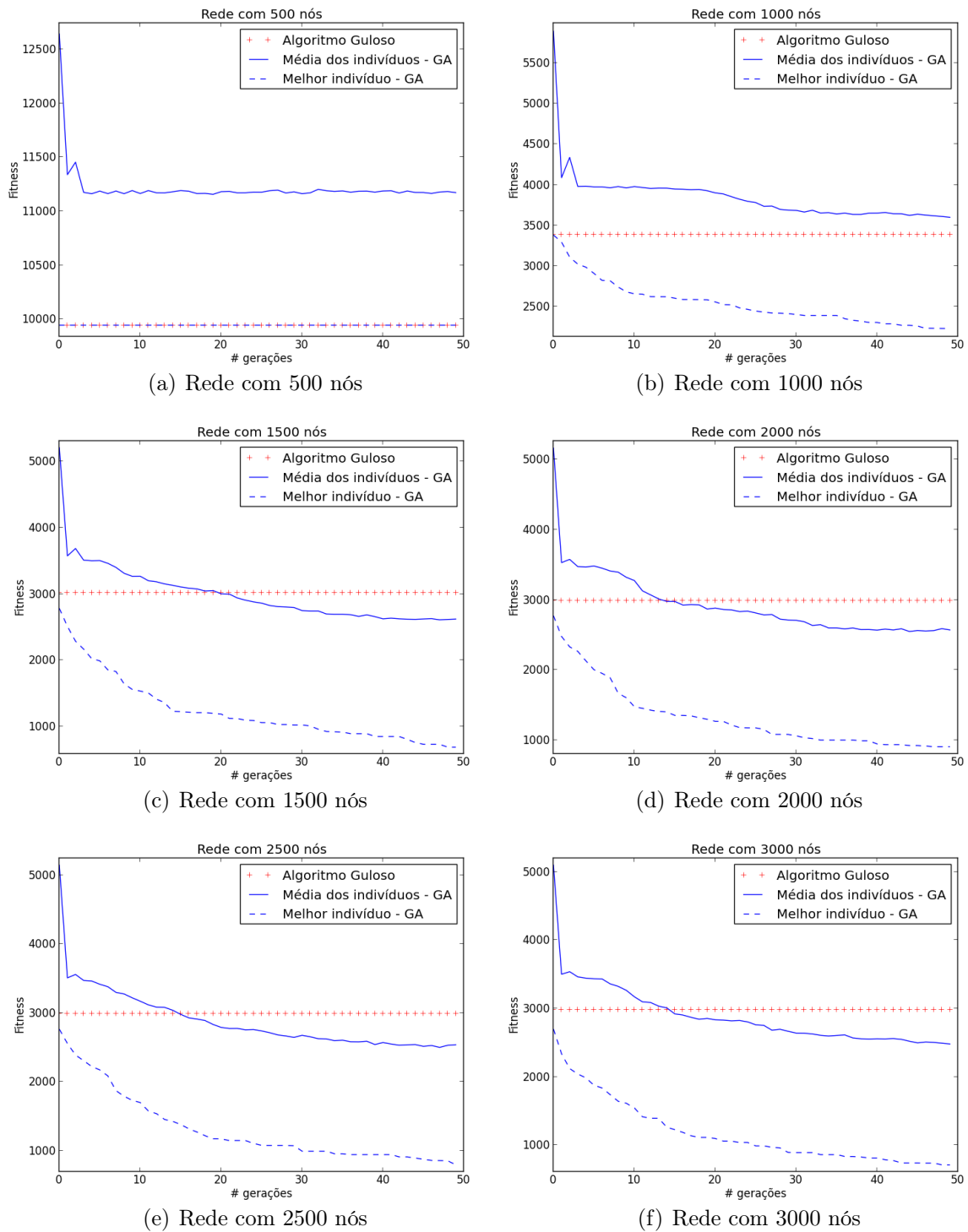
A Figura 5.11 mostra a diferença do tempo de execução entre os algoritmos guloso e genético. É possível perceber que o algoritmo genético gasta um tempo muito maior que o algoritmo guloso. Além disso, o número de nós tem um impacto muito maior no tempo de execução do algoritmo genético.



**Figura 5.11.** Tempo de Execução - Minimização de caminho sem varredura local

A Figura 5.12 mostra a evolução do algoritmo genético com o decorrer das gerações. Apenas para a rede com 500 nós que a solução permanece sempre a mesma. Esse fato ocorre porque a conectividade entre os nós nesse caso é pequena, fazendo com que a solução encontrada não possa ser melhorada. Os gráficos das redes com 1500, 2000, 2500 e 3000 mostram que até mesmo a média das soluções do algoritmo genético supera o menor caminho encontrado na solução do algoritmo guloso.

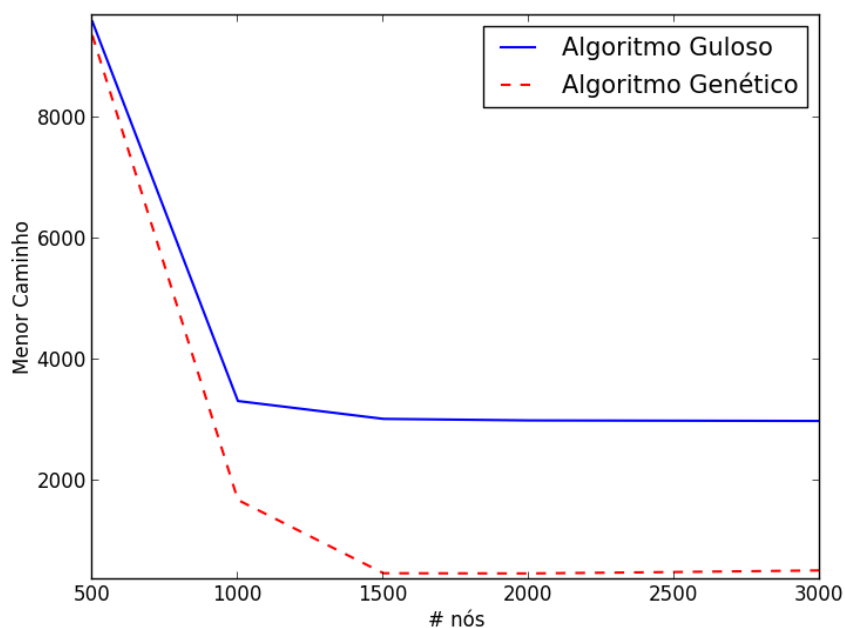
A grande variação entre os valores de melhor indivíduo e média dos indivíduos ilustra a heterogeneidade da população do algoritmo genético. Como já explicado, essa diversidade da população garante que um espaço maior de soluções seja avaliado.



**Figura 5.12.** Evolução do Algoritmo Genético para minimização de caminho sem varredura local

### 5.2.2 Algoritmo Genético com Trilateração e Varredura Local

A Figura 5.13 mostra o resultado final da simulação do algoritmo para minimização do caminho entre os âncoras contando com trilateração e varredura local. Novamente é possível perceber uma grande diferença entre os resultados alcançados pelos dois algoritmos, sendo o genético melhor que o guloso.



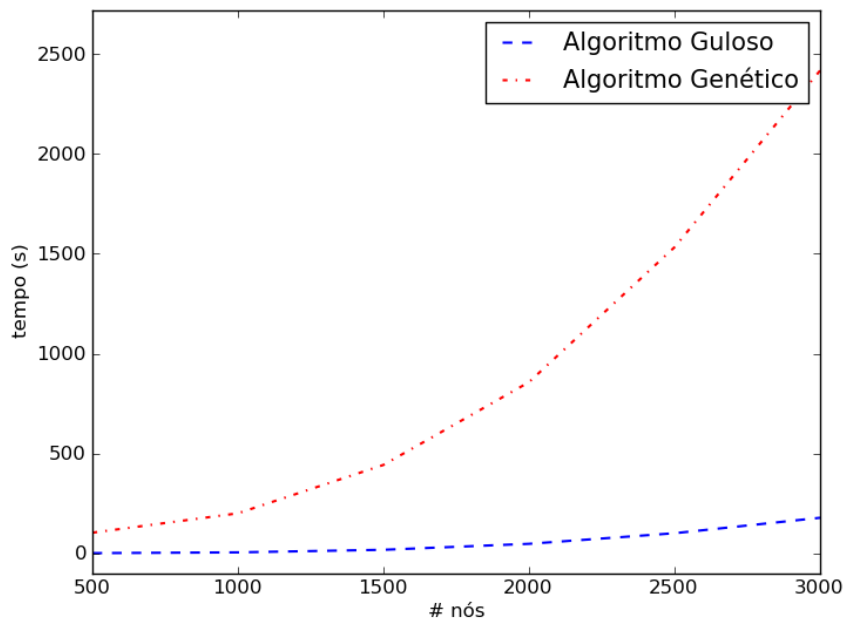
**Figura 5.13.** Resultado da execução para minimização do caminho com varredura local

Também é possível observar na Figura 5.13 que o aumento do número de nós na rede faz a diferença entre as soluções do algoritmo ficar mais clara. Novamente, é possível observar que o intervalo de confiança é um pouco maior, porém mesmo os valores mais altos ainda são melhores que o algoritmo guloso.

A Figura 5.14 mostra a diferença do tempo de execução entre os algoritmos guloso e genético. Pode-se concluir que para todos os casos apresentados o algoritmo genético gasta um tempo muito maior que o algoritmo guloso. Porém, o algoritmo genético alcança sempre melhores resultados. O tempo nesse caso é bastante influenciado pela heurística utilizada para o cálculo da *fitness*, sendo assim talvez seja possível apenas utilizar heurísticas mais rápidas para melhorar esse aspecto do algoritmo.

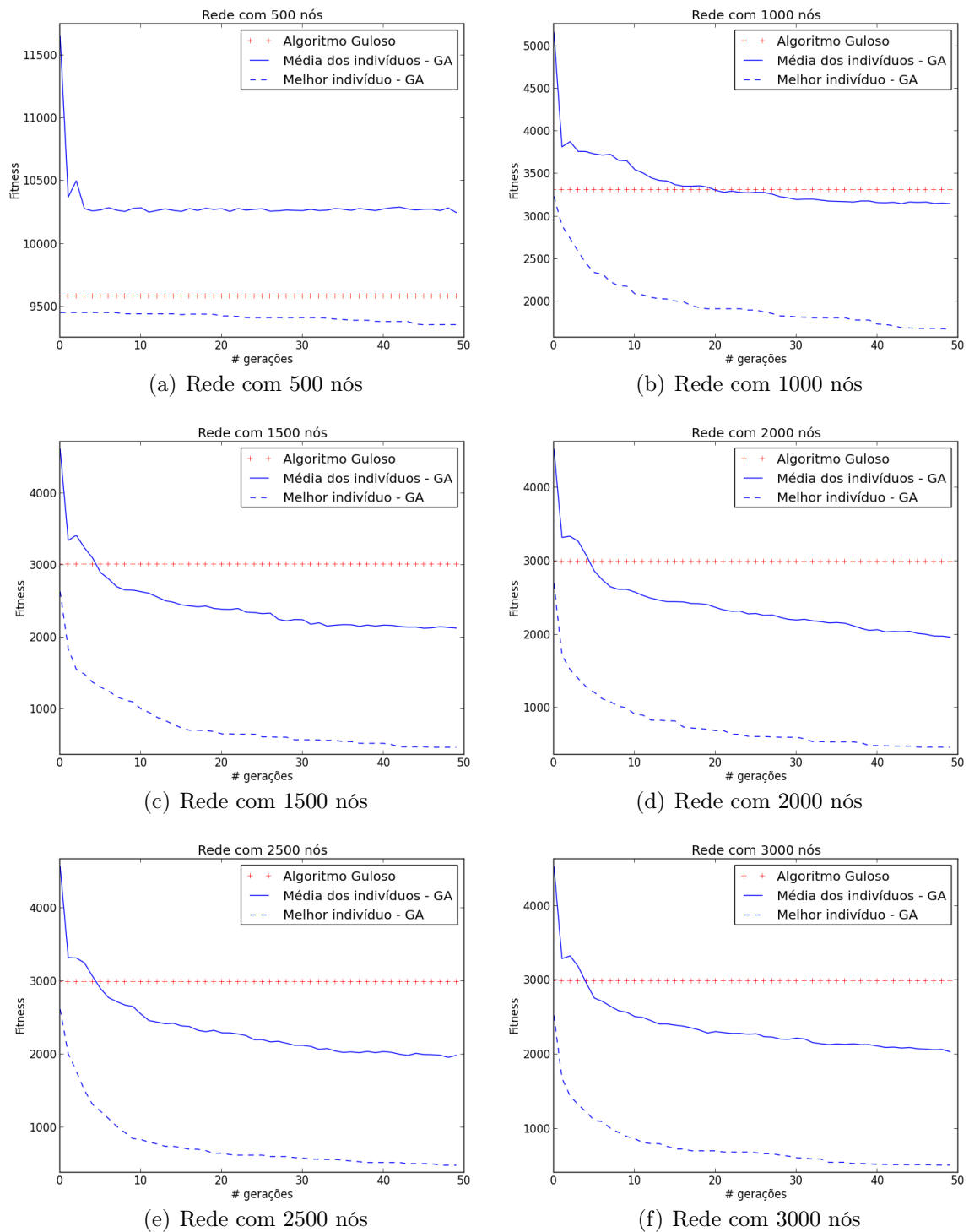
A Figura 5.15 mostra a evolução da população do algoritmo genético. Diferentemente dos gráficos anteriores, houve uma melhora para a rede com 500 nós nessa abordagem. Essa evolução que ocorreu ilustra bem a importância de usar métodos





**Figura 5.14.** Tempo de Execução - Minimização de caminho com varredura local

de localização além da trilateração. Outro fato que destaca a vantagem da varredura local é que na rede com 1000 nós, sem varredura local (Figura 5.12(b)) a média dos indivíduos não superou o resultado do algoritmo guloso, enquanto com varredura local aproximadamente na 20<sup>a</sup> geração até mesmo a média dos indivíduos do algoritmo genético já é melhor que a solução do algoritmo guloso. Para 1500, 2000, 2500 e 3000 nós antes mesmo de 10 gerações a média dos indivíduos já é melhor que o resultado do algoritmo guloso.



**Figura 5.15.** Evolução do Algoritmo Genético para minimização de caminho com varredura local

# Capítulo 6

## Considerações Finais

### 6.1 Conclusões

Nesta dissertação foi investigado o problema de localização de custo mínimo para redes de sensores sem fio com duas abordagens diferentes. A primeira abordagem foca em minimizar o número de nós âncoras, enquanto que a segunda abordagem minimiza o trajeto que inclui todos os âncoras definidos. Para ambos os casos é sempre considerado que toda a rede deve ser localizada.

Foram apresentadas definições formais que tratam das particularidades de cada uma das abordagens desse problema. A importância dessa definição se deve ao fato de especificar qual é o verdadeiro custo que se deseja minimizar no sistema de localização da rede.

Para o problema de minimização do caminho entre os nós âncoras foi discutida a semelhança entre este problema e o Problema do Caixeiro Viajante. Essa maneira de tratar o problema faz com que diversas opções de heurísticas possam ser utilizadas na avaliação das soluções dos algoritmos.

Foi definido um algoritmo genético para encontrar a solução desejada em cada caso. Esse algoritmo se mostrou eficaz, baseado na trilateração com ou sem o uso de varredura local. Os resultados apresentados mostram claramente que o algoritmo genético para o problema de minimização de nós âncoras alcançou soluções em média 50% melhores em simulações realizadas para redes de 1500 a 3000 nós. Ainda é possível perceber que até mesmo a média da *fitness* dos indivíduos do algoritmo genético supera o algoritmo guloso para as redes citadas.

## 6.2 Trabalhos Futuros

Uma possível melhoria seria encontrar um conjunto ideal de valores para todos os parâmetros do algoritmo genético. Para isso é preciso avaliar o impacto de cada um dos parâmetros do algoritmo, como taxa de cruzamento, taxa de mutação, número de gerações, e número de indivíduos. Alterações como aumento do número de indivíduos fazem com que um maior número de soluções seja analisado a cada geração, melhorando a solução final. Porém, esse tipo de alteração também aumenta significativamente o tempo de execução do algoritmo.

Também é possível ainda utilizar de outras métricas para a avaliação das soluções, ou seja, para o cálculo da *fitness*. Nesse caso, é possível realizar uma avaliação do impacto do uso de diferentes heurísticas na busca do menor caminho entre os nós âncoras. Sempre seguindo a restrição de que toda a rede seja localizada. Implementações e avaliações sobre essa nova abordagem foram deixadas como trabalhos futuros.

Outra abordagem futura desse trabalho é a utilização de múltiplos robôs como âncoras móveis. Abordagens desse tipo aparecem em trabalhos recentes e mostram que podem ser eficientes se o foco for a minimização do caminho total de todos os âncoras móveis [Bhadauria et al., 2011].

Como comentado na Seção 5, o modelo de conectividade dos nós é um dos modelos mais simples. Para avaliar melhor o algoritmo proposto, um futuro trabalho é realizar as simulações em redes com o modelo de conectividade quasi-UDG. Esse modelo se aproxima mais da realidade pois o modelo não é puramente geométrico, ou seja, a conectividade dos nós não depende apenas da distância entre eles, e sim de uma função de probabilidade que define se existirá um canal de comunicação (vizinhança).

# Referências Bibliográficas

- Akyildiz, I. F.; Su, W.; Sankarasubramaniam, Y. & Cayirci, E. (2002). Wireless sensor networks: a survey. *Computer Networks*, 38(4):393 – 422.
- Alves, J. C. (2012). Algoritmo para Localização em Redes de Sensores Sem Fio. Dissertação de mestrado, (UFMG) Universidade Federal de Minas Gerais, DCC Departamento de Ciência da Computação.
- Applegate, D. L.; Bixby, R. E.; Chvatal, V. & Cook, W. J. (2007). *The Traveling Salesman Problem: A Computational Study (Princeton Series in Applied Mathematics)*. Princeton University Press, Princeton, NJ, USA.
- Aspnes, J.; Eren, T.; Goldenberg, D. K.; Morse, A. S.; Whiteley, W.; Yang, Y. R.; Anderson, B. D. O. & Belhumeur, P. N. (2006). A theory of network localization. *IEEE Transactions on Mobile Computing*, 5(12):1663--1678.
- Bhadauria, D.; Tekdas, O. & Isler, V. (2011). Robotic data mules for collecting data over sparse sensor fields. *J. Field Robot.*, 28:388--404.
- Blickle, T. & Thiele, L. (1995). A mathematical analysis of tournament selection. Em *Proceedings of the Sixth International Conference on Genetic Algorithms*, pp. 9--16. Morgan Kaufmann.
- Boukerche, A.; Oliveira, H.; Nakamura, E. & Loureiro, A. (2007). Localization systems for wireless sensor networks. *Wireless Communications, IEEE*, 14(6):6 –12.
- Cormen, T. H.; Leiserson, C. E.; Rivest, R. L. & Stein, C. (2005). Dijkstra's algorithm. Em *Introduction to Algorithms 2nd edition*, capítulo 24, pp. 595--599. MIT Press.
- Current, J. R. & Schilling, D. A. (1989). The covering salesman problem. *Transportation Science*, 23(3):208–213.
- Doherty, L.; pister, K. & El Ghaoui, L. (2001). Convex position estimation in wireless sensor networks. Em *INFOCOM 2001. Twentieth Annual Joint Conference of the*

- IEEE Computer and Communications Societies. Proceedings. IEEE*, volume 3, pp. 1655–1663 vol.3.
- Duckett, T. (2003). A genetic algorithm for simultaneous localization and mapping. Em *in proceedings of the IEE International Conference on Robotics and Automation (ICRA '2003)*, pp. 434–439.
- Dumitrescu, A. & Mitchell, J. S. B. (2003). Approximation algorithms for tsp with neighborhoods in the plane. *J. Algorithms*, 48(1):135–159.
- Erol, M.; Vieira, L.; Caruso, A.; Paparella, F.; Gerla, M. & Oktug, S. (2008). Multi stage underwater sensor localization using mobile beacons. Em *Sensor Technologies and Applications, 2008. SENSORCOMM '08. Second International Conference on*, pp. 710 –714.
- Gao, Y.; Zhuang, Y.; Ni, T.; Yin, K. & Xue, T. (2010). An improved genetic algorithm for wireless sensor networks localization. Em *BIC-TA*, pp. 439–443. IEEE.
- Goldenberg, D. K.; Bihler, P.; Cao, M.; Fang, J.; Anderson, B. D. O.; Morse, A. S. & Yang, Y. R. (2006). Localization in sparse networks using sweeps. Em *Proceedings of the 12th annual international conference on Mobile computing and networking, MobiCom '06*, pp. 110–121, New York, NY, USA. ACM.
- He, T.; Huang, C.; Blum, B. M.; Stankovic, J. A. & Abdelzaher, T. (2003). Range-free localization schemes for large scale sensor networks. Em *Proceedings of the 9th annual international conference on Mobile computing and networking, MobiCom '03*, pp. 81–95, New York, NY, USA. ACM.
- Hofmann-Wellenhof, B.; Lichtenegger, H. & Collins, J. (2001). *Global Positioning System: Theory and Practice*. Springer, 5th edição.
- Huang, M.; Chen, S. & Wang, Y. (2011). Minimum cost localization problem in wireless sensor networks. *Ad Hoc Networks*, 9:387–399.
- Karp, B. & Kung, H. T. (2000). Gpsr: Greedy perimeter stateless routing for wireless networks. Em *MOBICOM*, pp. 243–254.
- Khan, U.; Kar, S. & Moura, J. (2009). Distributed sensor localization in random environments using minimal number of anchor nodes. *Signal Processing, IEEE Transactions on*, 57(5):2000 –2016.

- Kuo, S.-P.; Kuo, H.-J. & Tseng, Y.-C. (2009). The beacon movement detection problem in wireless sensor networks for localization applications. *Mobile Computing, IEEE Transactions on*, 8(10):1326 –1338.
- Loureiro, A. A. F.; Nogueira, J. M. S.; Ruiz, L. B.; Mini, R. A.; Nakamura, E. F. & Figueiredo, C. M. S. (2003). Wireless sensors networks (in portuguese). Em *Proceedings of the 21st Brazilian Symposium on Computer Networks (SBRC'03)*, pp. 179--226, Natal, RN, Brazil. Tutorial.
- Mitchell, J. S. B. (2007). A ptas for tsp with neighborhoods among fat regions in the plane. Em *Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms, SODA '07*, pp. 11--18, Philadelphia, PA, USA. Society for Industrial and Applied Mathematics.
- Moore, D.; Leonard, J.; Rus, D. & Teller, S. (2004). Robust distributed network localization with noisy range measurements. Em *Proceedings of the 2nd international conference on embedded networked sensor systems, SenSys '04*, pp. 50--61, New York, NY, USA. ACM.
- Niewiadomska-Szynkiewicz, E. & Marks, M. (2009). Optimization schemes for wireless sensor network localization. *Int. J. Appl. Math. Comput. Sci.*, 19(2):291--302.
- Oliveira, H. A. B. F. D. (2008). *Localização no tempo e no espaço em Redes de Sensores Sem Fio*. Phd thesis, UFMG - Universidade Federal de Minas Gerais, Brazil, DCC - Departamento de Ciência da Computação.
- Rodrigues, M.; Vieira, L. & Campos, M. (2011). Fingerprinting-based radio localization in indoor environments using multiple wireless technologies. Em *Personal Indoor and Mobile Radio Communications (PIMRC), 2011 IEEE 22nd International Symposium on*, pp. 1203 –1207.
- Ruiz, L. B.; Correia, L. H. A.; Vieira, L. F. M.; Macedo, D. F.; Nakamura, E. F.; Figueiredo, C. M. S.; Vieira, M. A. M.; Maia, E. H. B.; Câmara, D.; Loureiro, A. A. F.; Nogueira, J. M. S.; da Silva Jr., D. C. & Fernandes, A. O. (2004). Arquiteturas para redes de sensores sem fio. Em *Proceedings of the 22nd Brazilian Symposium on Computer Networks (SBRC'04)*, pp. 167--218, Gramado, RS, Brazil. Tutorial. ISBN: 85-88442-82-5.
- Safra, S. & Schwartz, O. (2006). On the complexity of approximating tsp with neighborhoods and related problems. *Computational Complexity*, 14:281–307. 10.1007/s00037-005-0200-3.

- Savvides, A.; Han, C.-C. & Strivastava, M. B. (2001). Dynamic fine-grained localization in ad-hoc networks of sensors. Em *Proceedings of the 7th annual international conference on Mobile computing and networking*, MobiCom '01, pp. 166–179, New York, NY, USA. ACM.
- Shang, W.; Yao, F.; Wan, P. & Hu, X. (2008). On minimum m-connected k-dominating set problem in unit disc graphs. *Journal of Combinatorial Optimization*, 16:99–106.
- Shih, C.-Y. & Marron, P. J. (2010). Cola: Complexity-reduced trilateration approach for 3d localization in wireless sensor networks. *Sensor Technologies and Applications, International Conference on*, 0:24–32.
- Sichitiu, M. & Ramadurai, V. (2004). Localization of wireless sensor networks with a mobile beacon. Em *Mobile Ad-hoc and Sensor Systems, 2004 IEEE International Conference on*, pp. 174 – 183.
- Simic, S. & Sastry, S. S. (2002). Distributed localization in wireless ad hoc networks. Relatório técnico UCB/ERL M02/26, EECS Department, University of California, Berkeley.
- Ssu, K.; ho Ou, C. & Jiau, H. C. (2005). Localization with mobile anchor points in wireless sensor networks. *IEEE Transactions on Vehicular Technology*, pp. 1187–1197.
- Vecchio, M.; López-Valcarce, R. & Marcelloni, F. (2012). A two-objective evolutionary approach based on topological constraints for node localization in wireless sensor networks. *Appl. Soft Comput.*, 12(7):1891--1901.
- Win, M.; Conti, A.; Mazuelas, S.; Shen, Y.; Gifford, W.; Dardari, D. & Chiani, M. (2011). Network localization and navigation via cooperation. *Communications Magazine, IEEE*, 49(5):56 –62.
- Yang, Z.; Liu, Y. & Li, X.-Y. (2009). Beyond trilateration: On the localizability of wireless ad-hoc networks. Em *INFOCOM 2009, IEEE*, pp. 2392 –2400.
- Yawut, C. & Kilaso, S. (2011). A wireless sensor network for weather and disaster alarm systems. Em *Proceedings 2011 International Conference on Information and Electronics Engineering (ICIEE 2011)*, volume 6, pp. 155–160.
- Yuan, B.; Orłowska, M. & Sadiq, S. (2007). On the optimal robot routing problem in wireless sensor networks. *Knowledge and Data Engineering, IEEE Transactions on*, 19(9):1252 –1261.



Zhang, Q.; Wang, J.; Jin, C.; Ye, J.; Ma, C. & Zhang, W. (2008). Genetic algorithm based wireless sensor network localization. Em *Natural Computation, 2008. ICNC '08. Fourth International Conference on*, volume 1, pp. 608 –613.

Zurich, D. C. G. A. E. (2013). Sinalgo - simulator for network algorithms. <http://dcg.ethz.ch/projects/sinalgo/>, visitado em 30/01/2013.