

UNIVERSIDADE FEDERAL DE SANTA MARIA
CENTRO DE TECNOLOGIA
PROGRAMA DE PÓS-GRADUAÇÃO EM INFORMÁTICA

LOCALIZAÇÃO DE NODOS EM REDES DE
SENSORES SEM FIO UTILIZANDO REDES
NEURAIS ARTIFICIAIS E METAHEURÍSTICAS

DISSERTAÇÃO DE MESTRADO

Stephan Hermes Chagas

Santa Maria, RS, Brasil

2012

**LOCALIZAÇÃO DE NODOS EM REDES DE
SENSORES SEM FIO UTILIZANDO REDES
NEURAIS ARTIFICIAIS E METAHEURÍSTICAS**

por

Stephan Hermes Chagas

Dissertação apresentada ao Programa de Pós-Graduação em Informática,
Área de concentração em Microeletrônica e Processamento de Sinais, da
Universidade Federal de Santa Maria (UFSM, RS), como requisito parcial
para obtenção do grau de
Mestre em Computação.

Orientador: Prof. Dr. João Baptista dos Santos Martins

Santa Maria, RS, Brasil

2012

Ficha catalográfica elaborada através do Programa de Geração Automática da Biblioteca Central da UFSM, com os dados fornecidos pelo(a) autor(a).

Hermes Chagas, Stephan
Localização de Nodos em Redes de Sensores Sem Fio
Utilizando Redes Neurais Artificiais e Metaheurísticas /
Stephan Hermes Chagas.-2012.
218 p. ; 30cm

Orientador: João Baptista dos Santos Martins
Dissertação (mestrado) - Universidade Federal de Santa
Maria, Centro de Tecnologia, Programa de Pós-Graduação em
Informática, RS, 2012

1. Redes de Sensores Sem Fio 2. Redes Neurais
Artificiais 3. Metaheurísticas 4. Localização de Nodos 5.
Otimização I. dos Santos Martins, João Baptista II. Título.

Universidade Federal de Santa Maria
Centro de Tecnologia
Programa de Pós-Graduação em Informática

A Comissão Examinadora, abaixo assinada,
aprova a Dissertação de Mestrado

**LOCALIZAÇÃO DE NODOS EM REDES DE SENSORES SEM
FIO UTILIZANDO REDES NEURAIIS ARTIFICIAIS E
METAHEURÍSTICAS**

elaborada por
Stephan Hermes Chagas

como requisito parcial para obtenção do grau de
Mestre em Computação

COMISSÃO EXAMINADORA:

João Baptista dos Santos Martins, Dr.
(Presidente/Orientador)

Leonardo Londero de Oliveira, Dr. (UFSM)

Rolf Fredi Molz, Dr. (UNISC)

Santa Maria, 25 de Outubro de 2012

AGRADECIMENTOS

Costumo dizer que dificilmente fazemos algo em nossas vidas sem a ajuda de outras pessoas. Podemos até tentar nos enganar, achando que fizemos algo totalmente sós e que temos todo o mérito, provavelmente apenas para satisfazer um ego pouco evoluído. Porém, se pararmos para refletir, certamente perceberemos que em algum momento recebemos a ajuda de alguém, direta ou indireta.

Ao contrário do que pensam alguns, vitoriosos são os que têm pessoas com as quais podem dividir suas conquistas, e não os que comemoram sós. Tive a sorte de ter ao meu lado pessoas que fizeram toda a diferença. Algumas delas, mesmo de longe, deram apoio imprescindível para que este momento fosse possível. Meu pai, Clóvis Chagas, é um grande exemplo disso.

Sendo assim, agradeço:

À minha família, em especial aos meus pais, Clóvis Chagas e Fátima Hermes, bem como às minhas irmãs, Danielle Chagas e Mariana Chagas. Não posso deixar de citar também meus tios, Sérgio e Elizabeth Ávila, Beatriz Chagas e Maria de Lourdes Hermes, pelo imenso apoio nos mais variados momentos de minha vida. Da mesma forma, agradeço ao meu cunhado, Mário dos Santos, por sua ajuda em diversas ocasiões.

À Pauline Neutzling Fraga, pela força e incentivo que me foram dados, especialmente nas inúmeras vezes em que tive vontade de desistir.

Aos meus amigos e companheiros nas "empreitadas musicais", Estevan Poll e Preto Cardoso, que tanto admiro, por tornarem o meu caminho mais suave e por me darem a honra de ouví-los e acompanhá-los em algumas peças musicais.

Aos amigos Max Henrique Braunstein, Rodrigo Comassetto, Alexandre Bolzan, Argos Miron e Gregor Coelho, pelo companheirismo de sempre. Um agradecimento especial ao amigo Tiago Oliveira Weber, por todas as discussões acerca deste trabalho e também por aquelas de cunho mais filosófico.

À Roberta Centofante, Rafaela de Maman, Stephen Souza Michel e Renata Celante, pelo incentivo, companheirismo e boas risadas, contribuindo para afastar momentanea-

Agradecimentos

mente as minhas preocupações relacionadas a esta dissertação.

Ao meu orientador, Prof. Dr. João Baptista dos Santos Martins, pelas oportunidades dadas a mim e por sua confiança em meu trabalho.

Ao Prof. Dr. Leonardo Londero de Oliveira, pelas inúmeras e proveitosas discussões sobre diversos pontos deste trabalho, bem como sobre os artigos publicados e demais temas relacionados à pesquisa.

Aos professores e alunos do Gmicro, por suas contribuições diretas e indiretas para a realização deste trabalho.

RESUMO

Dissertação de Mestrado
Programa de Pós-Graduação em Informática
Universidade Federal de Santa Maria, RS, Brasil

LOCALIZAÇÃO DE NODOS EM REDES DE SENSORES SEM FIO UTILIZANDO REDES NEURAIS ARTIFICIAIS E METAHEURÍSTICAS

AUTOR: STEPHAN HERMES CHAGAS

ORIENTADOR: JOÃO BAPTISTA DOS SANTOS MARTINS

Local da Defesa e Data: Santa Maria, 25 de Outubro de 2012.

Localização de nodos em redes de sensores sem fio (RSSF) tem importância em diversas aplicações, tais como monitoramento ambiental, robótica, vigilância militar, automação residencial, rastreamento de animais e muitas outras. A percepção da posição espacial pelos elementos da rede é um tema importante uma vez que os dados transmitidos/recebidos pelos nodos podem tornar-se inúteis se desprovidos dessa informação. Este procedimento é especialmente delicado quando a aplicação tem fortes restrições de energia, o que é situação comum em aplicações de RSSF. O uso de abordagens sem sistema de posicionamento global (GPS) é, então, necessário para fornecer um sistema de baixo custo e energeticamente eficiente. Métodos que usam informação indireta tal como RSSI (do inglês, *Received Signal Strength Indicator*) são utilizados para este tipo de aplicação. Este trabalho apresenta uma abordagem para o problema de localização de nodos em RSSF utilizando Redes Neurais Artificiais (RNAs) com estruturas otimizadas através das metaheurísticas Algoritmos Genéticos (AGs) e Simulated Annealing (SA). No AG, uma população composta por diversas RNAs é criada inicialmente, na qual cada indivíduo tem suas características estruturais principais codificadas em um código genético. A cada geração, os indivíduos são avaliados através do treinamento das RNAs e posterior cálculo da raiz do erro quadrático médio (REQM) para todo o conjunto de teste. De forma similar, a otimização é realizada com SA para melhorias sucessivas em um indivíduo (estrutura da RNA). Uma abordagem utilizando a concatenação de duas RNAs também foi proposta. Uma rede tem sua estrutura otimizada e parâmetros treinados para determinar a posição dos nodos e, em sequência, uma rede auxiliar é utilizada para corrigir o erro da primeira. Em todos os estudos, as medições de RSSI foram utilizadas como entradas para as RNAs visando localizar nodos. Os métodos foram testados usando o simulador de redes sem fio probabilístico chamado *Prowler*, que é baseado no software *MatLab*, para coletar os dados de entrada de um ambiente de rede estático interno. O ambiente de testes foi de 26x26 metros com quantidade e posições variadas para os âncoras, *i.e.*, nodos com percepção de suas posições. Simulações com diferente número de nodos-treino também foram realizadas. As ferramentas de algoritmos genéticos e RNAs do *MatLab* foram utilizadas. O melhor resultado obteve REQM de 0,23 metro, um erro máximo de 0,65 metro e um erro mínimo de 0,014 metro, utilizando 3 nodos-âncora e 64 nodos-treino.

Palavras-chave: Redes de Sensores Sem Fio; Redes Neurais Artificiais; Metaheurísticas; Algoritmos Genéticos; Localização; Otimização.

ABSTRACT

Master's Dissertation

Programa de Pós-Graduação em Informática
Federal University of Santa Maria, RS, Brazil

NODE LOCALIZATION IN WIRELESS SENSOR NETWORKS USING ARTIFICIAL NEURAL NETWORKS AND METAHEURISTICS

AUTHOR: STEPHAN HERMES CHAGAS

ADVISOR: JOÃO BAPTISTA DOS SANTOS MARTINS

Place and Date: Santa Maria, October 25th, 2012.

Node localization in wireless sensor networks (WSNs) is important in several applications, such as environmental monitoring, robotics, military surveillance, domotics, animal tracking and many others. Position awareness is an important issue because the collected/transmitted data by the network nodes could become meaningless without correct positioning information. This procedure is specially difficult when the application has energy saving requirements which is commonly the case in WSNs. The use of GPS-less approach is required to provide a low-cost and energy efficient system. Methods that use indirect information such as RSSI (Received Signal Strength Indicator) are frequently used for this kind of application. This work presents an artificial neural networks (ANNs) approach to localization in wireless sensor network through the adjustment of the ANNs structures using metaheuristics such as Genetic Algorithms (AGs) and Simulated Annealing (SA). In the AG several artificial neural networks compose a population containing their structure in a genetic code. At each generation, the individuals are evaluated through the training of the artificial neural networks and further calculation of their root mean square error for all the testing set. In a similar way, the optimization was accomplished using SA with successive improvements of a single individual (network structure). An approach using two neural networks was also proposed. One network is optimized to detect the node position and the next one aims to correct the localization errors of the previous. The RSSI measurements were used as the artificial neural networks inputs to localize the nodes. The method was tested using the MatLab-based Probabilistic Wireless Network Simulator (Prowler) to collect the artificial neural networks input data, under simulated static indoor network environment of 26x26 meters with several anchor-node topologies and variable sample size for training. The MatLab's genetic algorithms and artificial neural networks toolboxes were used. Results using the best artificial neural network structure found after optimization had a root mean square error of 0.23 meter, a maximum error of 0.65 meter and a minimum error of 0.014 meter using a 3-anchor topology with 64 training-nodes acquiring samples.

Keywords: Wireless Sensor Networks; Artificial Neural Networks; Metaheuristics; Genetic Algorithms; Localization; Optimization.

LISTA DE FIGURAS

Figura 1	Rede de sensores sem fio.	p. 33
Figura 2	Estrutura de um nodo.	p. 35
Figura 3	Estrutura mínima para aplicação de lateração em duas dimensões.	p. 44
Figura 4	Estrutura mínima para aplicação de triangulação.	p. 45
Figura 5	Regiões de localização em diferentes granularidades.	p. 47
Figura 6	Modelo de neurônio artificial.	p. 49
Figura 7	Função degrau.	p. 50
Figura 8	Função degrau bipolar.	p. 50
Figura 9	Função rampa simétrica	p. 51
Figura 10	Função linear.	p. 52
Figura 11	Função gaussiana.	p. 52
Figura 12	Função logística.	p. 53
Figura 13	Função tangente hiperbólica.	p. 53
Figura 14	Rede neural artificial do tipo <i>feedforward</i>	p. 55
Figura 15	Função logsig contínua e utilizando 32 intervalos.	p. 59
Figura 16	Erro entre a função aproximada e a contínua para a função logsig.	p. 59
Figura 17	Função tansig contínua e utilizando 32 intervalos.	p. 60
Figura 18	Erro entre a função aproximada e a contínua para a função tansig.	p. 60
Figura 19	Função radbas contínua e utilizando 32 intervalos.	p. 61
Figura 20	Erro entre a função aproximada e a contínua para a função radbas.	p. 62
Figura 21	Função purelin contínua e utilizando 32 intervalos.	p. 62
Figura 22	Estrutura da RALUT.	p. 63
Figura 23	Mínimos locais e mínimo global.	p. 68
Figura 24	Exemplo de função custo deceptiva.	p. 69
Figura 25	Exemplos de tipos de <i>Crossovers</i>	p. 71
Figura 26	Mínimos locais e mínimo global no <i>Simulated Annealing</i>	p. 75
Figura 27	Arranjos de nodos adotados para testes com o algoritmo <i>Simulated Annealing</i>	p. 88

Figura 28	Arranjos de nodos adotados para testes do primeiro estágio do sistema de localização com duas RNAs.	p. 89
Figura 29	Conjunto de testes para o segundo estágio do sistema de localização com duas RNAs.	p. 89
Figura 30	Variações no tamanho das amostras de treinamento.	p. 90
Figura 31	Disposições de nodos-âncora utilizadas.	p. 91
Figura 32	Disposições alternativas de nodos-âncora.	p. 94
Figura 33	Arranjo de nodos-treino sem repetição de coordenadas.	p. 95
Figura 34	Representação de uma RNA utilizada no algoritmo SA.	p. 97
Figura 35	Otimização baseada em <i>Simulated Annealing</i>	p. 98
Figura 36	Código genético para representação de RNAs utilizado.	p. 98
Figura 37	Otimização baseada em algoritmo genético.	p. 99
Figura 38	Influência do tipo de solução inicial do <i>Simulated Annealing</i> nas estimativas de localização para 8 nodos-âncora.	p. 103
Figura 39	Comparação dos motores de localização de um e dois estágios usando 9 nodos-âncora.	p. 105
Figura 40	Influência da quantidade de âncoras no erro de localização usando 16 nodos-treino.	p. 108
Figura 41	Influência da quantidade de âncoras no erro de localização usando 16 nodos-treino.	p. 109
Figura 42	Influência da quantidade de âncoras nas estimativas de localização usando 16 nodos-treino.	p. 110
Figura 43	Influência da quantidade de âncoras nas fronteiras de Pareto usando 16 nodos-treino.	p. 111
Figura 44	Influência da quantidade de âncoras no erro de localização usando 25 nodos-treino.	p. 113
Figura 45	Influência da quantidade de âncoras no erro de localização usando 25 nodos-treino.	p. 114
Figura 46	Influência da quantidade de âncoras nas estimativas de localização usando 25 nodos-treino.	p. 115
Figura 47	Influência da quantidade de âncoras nas fronteiras de Pareto usando 25 nodos-treino.	p. 116
Figura 48	Influência da quantidade de âncoras no erro de localização usando 36 nodos-treino.	p. 118
Figura 49	Influência da quantidade de âncoras no erro de localização usando 36 nodos-treino.	p. 119

Figura 50	Influência da quantidade de âncoras nas estimativas de localização usando 36 nodos-treino.	p. 120
Figura 51	Influência da quantidade de âncoras nas fronteiras de Pareto usando 36 nodos-treino.	p. 121
Figura 52	Influência da quantidade de âncoras no erro de localização usando 49 nodos-treino.	p. 123
Figura 53	Influência da quantidade de âncoras no erro de localização usando 49 nodos-treino.	p. 124
Figura 54	Influência da quantidade de âncoras nas estimativas de localização usando 49 nodos-treino.	p. 125
Figura 55	Influência da quantidade de âncoras nas fronteiras de Pareto usando 49 nodos-treino.	p. 126
Figura 56	Influência da quantidade de âncoras no erro de localização usando 64 nodos-treino.	p. 128
Figura 57	Influência da quantidade de âncoras no erro de localização usando 64 nodos-treino.	p. 129
Figura 58	Influência da quantidade de âncoras nas estimativas de localização usando 64 nodos-treino.	p. 130
Figura 59	Influência da quantidade de âncoras nas fronteiras de Pareto usando 64 nodos-treino.	p. 131
Figura 60	Influência da quantidade de âncoras no erro de localização usando 81 nodos-treino.	p. 133
Figura 61	Influência da quantidade de âncoras no erro de localização usando 81 nodos-treino.	p. 134
Figura 62	Influência da quantidade de âncoras nas estimativas de localização usando 81 nodos-treino.	p. 135
Figura 63	Influência da quantidade de âncoras nas fronteiras de Pareto usando 81 nodos-treino.	p. 136
Figura 64	Comportamento da REQM em função do aumento de nodos-âncora pra todas as amostras simuladas	p. 137
Figura 65	Influência do tamanho da amostra de treinamento no erro de localização para 3 nodos-âncora.	p. 139
Figura 66	Influência do tamanho da amostra de treinamento no erro de localização para 3 nodos-âncora.	p. 140
Figura 67	Influência do tamanho da amostra de treinamento nas estimativas de localização para 3 nodos-âncora.	p. 141
Figura 68	Influência do tamanho da amostra de treinamento nas fronteiras de Pareto para 3 nodos-âncora.	p. 142

Figura 69	Influência do tamanho da amostra de treinamento no erro de localização para 4 nodos-âncora.	p. 144
Figura 70	Influência do tamanho da amostra de treinamento no erro de localização para 4 nodos-âncora.	p. 145
Figura 71	Influência do tamanho da amostra de treinamento nas estimativas de localização para 4 nodos-âncora.	p. 146
Figura 72	Influência do tamanho da amostra de treinamento nas fronteiras de Pareto para 4 nodos-âncora.	p. 147
Figura 73	Influência do tamanho da amostra de treinamento no erro de localização para 5 nodos-âncora.	p. 149
Figura 74	Influência do tamanho da amostra de treinamento no erro de localização para 5 nodos-âncora.	p. 150
Figura 75	Influência do tamanho da amostra de treinamento nas estimativas de localização para 5 nodos-âncora.	p. 151
Figura 76	Influência do tamanho da amostra de treinamento nas fronteiras de Pareto para 5 nodos-âncora.	p. 152
Figura 77	Influência do tamanho da amostra de treinamento no erro de localização para 6 nodos-âncora.	p. 154
Figura 78	Influência do tamanho da amostra de treinamento no erro de localização para 6 nodos-âncora.	p. 155
Figura 79	Influência do tamanho da amostra de treinamento nas estimativas de localização para 6 nodos-âncora.	p. 156
Figura 80	Influência do tamanho da amostra de treinamento nas fronteiras de Pareto para 6 nodos-âncora.	p. 157
Figura 81	Influência do tamanho da amostra de treinamento no erro de localização para 7 nodos-âncora.	p. 159
Figura 82	Influência do tamanho da amostra de treinamento no erro de localização para 7 nodos-âncora.	p. 160
Figura 83	Influência do tamanho da amostra de treinamento nas estimativas de localização para 7 nodos-âncora.	p. 161
Figura 84	Influência do tamanho da amostra de treinamento nas fronteiras de Pareto para 7 nodos-âncora.	p. 162
Figura 85	Influência do tamanho da amostra de treinamento no erro de localização para 8 nodos-âncora.	p. 164
Figura 86	Influência do tamanho da amostra de treinamento no erro de localização para 8 nodos-âncora.	p. 165
Figura 87	Influência do tamanho da amostra de treinamento nas estimativas de localização para 8 nodos-âncora.	p. 166

Figura 88	Influência do tamanho da amostra de treinamento nas fronteiras de Pareto para 8 nodos-âncora.	p. 167
Figura 89	Influência do tamanho da amostra de treinamento no erro de localização para 9 nodos-âncora.	p. 169
Figura 90	Influência do tamanho da amostra de treinamento no erro de localização para 9 nodos-âncora.	p. 170
Figura 91	Influência do tamanho da amostra de treinamento nas estimativas de localização para 9 nodos-âncora.	p. 171
Figura 92	Influência do tamanho da amostra de treinamento nas fronteiras de Pareto para 9 nodos-âncora.	p. 172
Figura 93	Comportamento da REQM em função do aumento da amostra de treinamento, para todos os números de nodos-âncora simulados.	p. 173
Figura 94	Influência da posição de 3 âncoras usando 36 nodos-treino.	p. 176
Figura 95	Influência da posição de 3 âncoras usando 81 nodos-treino.	p. 177
Figura 96	Influência da posição de 4 âncoras usando 36 nodos-treino.	p. 179
Figura 97	Influência da posição de 4 âncoras usando 81 nodos-treino.	p. 180
Figura 98	Influência da posição de 5 âncoras usando 36 nodos-treino.	p. 182
Figura 99	Influência da posição de 5 âncoras usando 81 nodos-treino.	p. 183
Figura 100	Influência da posição de 6 âncoras usando 36 nodos-treino.	p. 185
Figura 101	Influência da posição de 6 âncoras usando 81 nodos-treino.	p. 186
Figura 102	Influência da posição dos âncoras usando 25 nodos-treino.	p. 188
Figura 103	Influência da posição dos âncoras usando 81 nodos-treino.	p. 189
Figura 104	Influência da posição dos âncoras usando 25 nodos-treino.	p. 191
Figura 105	Influência da posição dos âncoras usando 81 nodos-treino.	p. 192
Figura 106	Influência da posição de 4 nodos-âncora nos resultados com 16 nodos-treino.	p. 194
Figura 107	Influência da posição de 4 nodos-âncora nos resultados com 25 nodos-treino.	p. 195
Figura 108	Influência da posição de 4 nodos-âncora nos resultados com 36 nodos-treino.	p. 196
Figura 109	Influência da posição de 4 nodos-âncora nos resultados com 49 nodos-treino.	p. 197
Figura 110	Influência da posição de 4 nodos-âncora nos resultados com 64 nodos-treino.	p. 198
Figura 111	Influência da posição de 4 nodos-âncora nos resultados com 81 nodos-treino.	p. 199

Figura 112 Influência da posição de 25 nodos-treino nos resultados com 3 âncoras.p. 201

LISTA DE TABELAS

Tabela 1	Especificações do mote comercial <i>MICAz</i>	p. 38
Tabela 2	<i>Lookup table</i> para função $\text{seno}(x\pi/8)$	p. 57
Tabela 3	Parâmetros Principais de Configuração do Algoritmo Genético . . .	p. 72
Tabela 4	Parâmetros de configuração do algoritmo de treinamento das RNAs	p. 85
Tabela 5	Limites das variáveis para otimização da estrutura das redes neurais artificiais	p. 85
Tabela 6	Parâmetros de configuração dos algoritmos genéticos	p. 86
Tabela 7	Parâmetros de configuração do algoritmo <i>Simulated Annealing</i> . . .	p. 87
Tabela 8	Resultados para amostra de 16 nodos de treinamento variando-se a quantidade de âncoras - ordenação por REQM	p. 107
Tabela 9	Resultados para amostra de 16 nodos de treinamento variando-se a quantidade de âncoras - ordenação por âncoras	p. 107
Tabela 10	Resultados para amostra de 25 nodos de treinamento variando-se a quantidade de âncoras - ordenação por REQM	p. 112
Tabela 11	Resultados para amostra de 25 nodos de treinamento variando-se a quantidade de âncoras - ordenação por âncoras	p. 112
Tabela 12	Resultados para amostra de 36 nodos de treinamento variando-se a quantidade de âncoras - ordenação por REQM	p. 117
Tabela 13	Resultados para amostra de 36 nodos de treinamento variando-se a quantidade de âncoras - ordenação por âncoras	p. 117
Tabela 14	Resultados para amostra de 49 nodos de treinamento variando-se a quantidade de âncoras - ordenação por REQM	p. 122
Tabela 15	Resultados para amostra de 49 nodos de treinamento variando-se a quantidade de âncoras - ordenação por âncoras	p. 122
Tabela 16	Resultados para amostra de 64 nodos de treinamento variando-se a quantidade de âncoras - ordenação por REQM	p. 127
Tabela 17	Resultados para amostra de 64 nodos de treinamento variando-se a quantidade de âncoras - ordenação por âncoras	p. 127
Tabela 18	Resultados para amostra de 81 nodos de treinamento variando-se a quantidade de âncoras - ordenação por REQM	p. 132
Tabela 19	Resultados para amostra de 81 nodos de treinamento variando-se a quantidade de âncoras - ordenação por âncoras	p. 132

Tabela 20	Resultados para 3 âncoras variando-se o tamanho da amostra de treinamento - ordenação por REQM	p. 138
Tabela 21	Resultados para 3 âncoras variando-se o tamanho da amostra de treinamento - ordenação por âncoras	p. 138
Tabela 22	Resultados para 4 âncoras variando-se o tamanho da amostra de treinamento - ordenação por REQM	p. 143
Tabela 23	Resultados para 4 âncoras variando-se o tamanho da amostra de treinamento - ordenação por âncoras	p. 143
Tabela 24	Resultados para 5 âncoras variando-se o tamanho da amostra de treinamento - ordenação por REQM	p. 148
Tabela 25	Resultados para 5 âncoras variando-se o tamanho da amostra de treinamento - ordenação por âncoras	p. 148
Tabela 26	Resultados para 6 âncoras variando-se o tamanho da amostra de treinamento - ordenação por REQM	p. 153
Tabela 27	Resultados para 6 âncoras variando-se o tamanho da amostra de treinamento - ordenação por âncoras	p. 153
Tabela 28	Resultados para 7 âncoras variando-se o tamanho da amostra de treinamento - ordenação por REQM	p. 158
Tabela 29	Resultados para 7 âncoras variando-se o tamanho da amostra de treinamento - ordenação por âncoras	p. 158
Tabela 30	Resultados para 8 âncoras variando-se o tamanho da amostra de treinamento - ordenação por REQM	p. 163
Tabela 31	Resultados para 8 âncoras variando-se o tamanho da amostra de treinamento - ordenação por âncoras	p. 163
Tabela 32	Resultados para 9 âncoras variando-se o tamanho da amostra de treinamento - ordenação por REQM	p. 168
Tabela 33	Resultados para 9 âncoras variando-se o tamanho da amostra de treinamento - ordenação por âncoras	p. 168
Tabela 34	Comparação de desempenho entre os métodos propostos e outros da literatura	p. 203

LISTA DE SIGLAS E ABREVIATURAS

RSSF	Rede de Sensor Sem Fio
GPS	<i>Global Positioning System</i>
RNA	Rede Neural Artificial
CC	Corrente Contínua
IEEE	<i>Institute of Electrical and Electronics Engineers</i>
ToA	<i>Time of Arrival</i>
RSSI	Received Signal Strength Indicator
AoA	<i>Angle of Arrival</i>
CM	<i>Connectivity Metric</i>
TANSIG	Função de Transferência Tangente Hiperbólica
LOGSIG	Função de Transferência Logística
RADBAS	Função de Transferência Gaussiana
PURELIN	Função de Transferência Linear
LUT	<i>Lookup Table</i>
RALUT	<i>Range Addressable Lookup Table</i>
PWL	<i>Piece-wise Linear Functions</i>
ROM	<i>Read-only Memory</i>
XOR	Porta lógica do tipo Ou-exclusivo
MIT	<i>Massachusetts Institute of Technology</i>
MLP	<i>Multi-Layer Perceptron</i>
RBF	<i>Radial Basis Function</i>
RRBF	<i>Reduced Radial Basis Function</i>
RNN	<i>Recurrent Neural Network</i>
RMSE	<i>Root Mean Square Error</i>

Lista de Siglas e Abreviaturas

REQM	Raiz do Erro Quadrático Médio
MLE	<i>Maximum Likelihood Estimation</i>
AG	Algoritmo Genético
DNA	Ácido desoxirribonucleico
SA	<i>Simulated Annealing</i>
AND	Porta lógica do tipo E
OR	Porta lógica do tipo Ou
NOT	Porta lógica do tipo Inversora
SINR	<i>Signal to Interference and Noise Ratio</i>
LVQ	<i>Learning Vector Quantization</i>
REQMp	Raiz do Erro Quadrático Médio Proporcional
2 MLP-AG	Rede Neural do tipo <i>Multi-Layer Perceptron</i> otimizada por algoritmo genético
MLP-SA	Rede Neural do tipo <i>Multi-Layer Perceptron</i> otimizada por <i>Simulated Annealing</i>
MLP-AGMO	Rede Neural do tipo <i>Multi-Layer Perceptron</i> otimizada por algoritmo genético multi-objetivo
PV	<i>Position-velocity</i>
PVA	<i>Position-velocity-Acceleration</i>

SUMÁRIO

1	Introdução	p. 29
1.1	Motivação	p. 30
1.2	Objetivos	p. 31
1.3	Contribuições do Trabalho	p. 31
1.4	Organização da Dissertação	p. 32
2	Fundamentação Teórica	p. 33
2.1	Redes de Sensores Sem Fio	p. 33
2.1.1	Introdução	p. 33
2.1.2	Características dos Nodos Sensores	p. 34
2.1.2.1	Tamanho Reduzido	p. 34
2.1.2.2	Baixo Custo	p. 34
2.1.2.3	Baixo Consumo	p. 34
2.1.2.4	Comunicação Sem Fio	p. 35
2.1.2.5	Escalabilidade e Auto-Organização	p. 35
2.1.2.6	Estrutura Interna	p. 35
2.1.2.7	Mote Comercial <i>MICAz</i>	p. 37
2.1.3	Características das Redes de Sensores	p. 37
2.1.4	Exemplos de Aplicações	p. 38
2.1.4.1	Militar	p. 39
2.1.4.2	Domótica	p. 39
2.1.4.3	Saúde	p. 39

Sumário

2.1.4.4	Logística	p. 39
2.1.4.5	Vigilância	p. 40
2.1.4.6	Monitoramento Ambiental	p. 40
2.2	Localização em Redes de Sensores Sem Fio	p. 40
2.2.1	Introdução	p. 40
2.2.2	Classificação das Técnicas de Localização	p. 41
2.2.2.1	Conforme Tipo de Observação	p. 41
2.2.2.2	Conforme Caráter Individual/Coletivo do Processamento da Localização	p. 42
2.2.2.3	Conforme o Local de Processamento das Observações	p. 42
2.2.3	Algoritmos de Localização Encontrados na Literatura	p. 42
2.3	Redes Neurais Artificiais	p. 47
2.3.1	Introdução	p. 47
2.3.1.1	Características Principais	p. 48
2.3.2	Neurônio Artificial	p. 48
2.3.2.1	Funções de Transferência Parcialmente Diferenciáveis	p. 50
2.3.2.2	Funções de Transferência Totalmente Diferenciáveis	p. 51
2.3.3	Processos de Treinamento	p. 54
2.3.4	Arquiteturas de RNAs	p. 54
2.3.5	Aplicações	p. 55
2.3.6	Estudo Sobre Implementações das Funções de Transferência	p. 57
2.3.6.1	Formas de Implementação em Hardware	p. 61
2.4	Localização Utilizando Redes Neurais Artificiais	p. 63
2.4.1	Introdução	p. 63
2.4.2	Trabalhos Encontrados na Literatura	p. 64
2.5	Metaheurísticas	p. 66
2.5.1	Introdução	p. 66

Sumário

2.5.2	Problema de Otimização	p. 67
2.5.3	Problema com Múltiplos Objetivos	p. 69
2.5.4	Algoritmos Genéticos	p. 70
2.5.5	<i>Simulated Annealing</i>	p. 72
2.5.6	Aplicações	p. 75
2.5.6.1	Roteamento de Veículos	p. 75
2.5.6.2	Problema da Mochila	p. 75
2.5.6.3	Problema do Caixeiro Viajante	p. 75
2.5.6.4	Problema da Satisfatibilidade Booleana	p. 76
2.6	Software <i>MatLab</i>	p. 76
2.6.1	Ambiente de simulação - <i>Prowler</i>	p. 76
2.6.2	Toolbox de Redes Neurais Artificiais	p. 78
2.6.3	Toolbox de Otimização Global	p. 80
3	Metodologia	p. 83
3.1	Introdução	p. 83
3.2	Métodos Propostos	p. 83
3.3	Cenário	p. 83
3.4	Configuração de Ferramentas	p. 84
3.4.1	<i>Prowler</i>	p. 84
3.4.2	<i>Toolbox</i> de Redes Neurais Artificiais	p. 84
3.4.3	<i>Toolbox</i> de Algoritmo Genético Multi-Objetivo	p. 86
3.4.4	Algoritmo <i>Simulated Annealing</i>	p. 86
3.5	Topologias para Testes Iniciais	p. 87
3.5.1	<i>Simulated Annealing</i>	p. 88
3.5.2	Sistema de dois estágios	p. 88
3.6	Topologias para Testes Principais	p. 89

Sumário

3.6.1	Variações no Tamanho das Amostras de Treinamento	p. 90
3.6.2	Variações na Quantidade de Nodos-âncora	p. 90
3.7	Topologias para Testes Complementares	p. 91
3.7.1	Variações nas Posições dos Nodos-âncora	p. 92
3.7.1.1	Arranjo Alternativo - 3 Âncoras	p. 92
3.7.1.2	Arranjo Alternativo - 4 Âncoras	p. 92
3.7.1.3	Arranjo Alternativo - 5 Âncoras	p. 92
3.7.1.4	Arranjo Alternativo - 6 Âncoras	p. 92
3.7.1.5	Arranjo Circular Alternativo - 8 Âncoras	p. 93
3.7.1.6	Arranjo Circular Alternativo - 9 Âncoras	p. 93
3.7.1.7	Arranjo Triangular Alternativo - 4 Âncoras	p. 93
3.7.2	Varição nas Posições dos Nodos de Treinamento	p. 95
3.8	Aquisição de dados	p. 95
3.9	Algoritmo Principal	p. 96
3.9.1	Dados de Entrada	p. 96
3.9.2	Processos de Otimização	p. 96
3.9.2.1	<i>Simulated Annealing</i>	p. 96
3.9.2.2	Algoritmo Genético Multi-Objetivo	p. 97
3.9.3	Dados de Saída	p. 98
4	Resultados	p. 101
4.1	Introdução	p. 101
4.2	Testes Iniciais	p. 102
4.2.1	<i>Simulated Annealing</i>	p. 102
4.2.2	Motor de localização de 2 estágios	p. 104
4.3	Testes Principais	p. 106
4.3.1	Varição na Quantidade de Nodos-Âncora	p. 106

Sumário

4.3.1.1	Amostra composta por 16 Nodos-Treino	p.107
4.3.1.2	Amostra composta por 25 Nodos-Treino	p.112
4.3.1.3	Amostra composta por 36 Nodos-Treino	p.117
4.3.1.4	Amostra composta por 49 Nodos-Treino	p.122
4.3.1.5	Amostra composta por 64 Nodos-Treino	p.127
4.3.1.6	Amostra composta por 81 Nodos-Treino	p.132
4.3.1.7	Análise do REQM em função do número de âncoras .	p.137
4.3.2	Variação no Tamanho da Amostra de Treinamento	p.138
4.3.2.1	Topologias com 3 Nodos-Âncora	p.138
4.3.2.2	Topologias com 4 Nodos-Âncora	p.143
4.3.2.3	Topologias com 5 Nodos-Âncora	p.148
4.3.2.4	Topologias com 6 Nodos-Âncora	p.153
4.3.2.5	Topologias com 7 Nodos-Âncora	p.158
4.3.2.6	Topologias com 8 Nodos-Âncora	p.163
4.3.2.7	Topologias com 9 Nodos-Âncora	p.168
4.3.2.8	Análise do REQM em função do tamanho das amostras	p.173
4.4	Testes complementares	p.174
4.4.1	Variação na Posição de Nodos-Âncora e Nodos-Treino	p.174
4.4.1.1	Arranjo Alternativo - 3 Nodos-Âncora	p.175
4.4.1.2	Arranjo Alternativo - 4 Nodos-Âncora	p.178
4.4.1.3	Arranjo Alternativo - 5 Nodos-Âncora	p.181
4.4.1.4	Arranjo Alternativo - 6 Nodos-Âncora	p.184
4.4.1.5	Arranjo Alternativo - 8 Nodos-Âncora em Círculo . .	p.187
4.4.1.6	Arranjo Alternativo - 9 Nodos-Âncora em Círculo . .	p.190
4.4.1.7	4 Nodos-Âncora em Arranjo Triangular	p.193
4.4.1.8	Variação na Posição de 25 Nodos-Treino - 3 Âncoras .	p.200

Sumário

4.5	Comparação com Outras Abordagens	p. 202
5	Conclusão	p. 205
5.1	Trabalhos Futuros	p. 207
	Apêndice A – Artigos Publicados Pelo Autor	p. 209
	Apêndice B – Triangulação e Trilateração	p. 211
	Referências	p. 215

1 INTRODUÇÃO

Rede de sensores sem fio (RSSF) é uma sub-classe das redes *ad-hoc*. Basicamente, é uma rede que visa monitorar algum tipo de fenômeno onde cada dispositivo (chamado usualmente de nodo ou terminal) geralmente possui um ou mais sensores, um processador, uma unidade de armazenamento, um transceptor de radiofrequência e uma bateria.

Em uma RSSF, todos os nodos cooperam entre si para desempenhar uma tarefa específica, que pode variar desde monitoramento ambiental até aplicações industriais e militares. Tendo em vista a natureza das aplicações das RSSFs, a localização de nodos é uma área de pesquisa que tem atraído bastante a atenção da comunidade científica em âmbito mundial. Geralmente, os sensores medem alguma grandeza física em algum ponto do espaço (por exemplo, temperatura do ar, luminosidade, etc.), de modo que estas medições não seriam muito úteis se os locais onde foram tomadas não fossem informados.

Portanto, as redes de sensores sem fio devem ser capazes de determinar as posições de seus terminais. Na maioria dos casos, existe nas redes um tipo de nodo especial. Este tipo, chamado geralmente de nodo-âncora ou nodo de referência, possui conhecimento sobre sua própria localização. Além disso, em alguns casos, tal tipo de nodo ainda tem maior capacidade de processamento e, em virtude disto, uma maior disponibilidade de energia. Os nodos que não possuem informações sobre suas próprias coordenadas, são usualmente denominados nodos desconhecidos.

As técnicas de localização podem utilizar mais de um tipo de informação para inferir a posição dos nodos desconhecidos. A classificação mais usual separa os métodos em *range-based* e *range-free*.

No tipo *range-based*, as técnicas usualmente baseiam-se em medições de distâncias ou ângulos, para posteriormente aplicar os métodos de lateração ou triangulação. Na lateração, a localização do terminal de posição desconhecida é feita com base na distância entre este e três ou mais terminais de posição conhecida (para casos de localização em duas dimensões). Já no método de triangulação, a localização é feita com base nos ângulos

de retas que ligam o terminal desconhecido aos terminais cujas posições são conhecidas (terminais de referência).

No tipo *range-free*, apenas as informações de conectividade são usadas para derivar a posição dos terminais desconhecidos. Informações retiradas das mensagens trocadas entre os nodos são suficientes para aplicar este tipo de técnica.

Tem-se investigado também a utilização de redes neurais artificiais (referenciadas no texto através da sigla RNAs) para o problema da localização. No entanto, poucos métodos utilizam RNAs como princípio de funcionamento, por exemplo (SHAREEF; ZHU; MUSAVI, 2007), (RAHMAN; PARK; KIM, 2009) e (TIAN; SHI, 2007). Todos eles utilizam como entrada de suas RNAs dados do tipo *range-based*, ou seja, medições de grandezas tais como distâncias e potência de sinais de radiofrequência.

1.1 Motivação

Devido às limitações de custo, consumo de energia e tamanho físico dos nodos utilizados nas RSSFs, a necessidade de localização de nodos sem o uso de GPS (*Global Positioning System*) fica evidente. Muitos métodos para a resolução deste problema foram propostos ao longo dos anos. Na grande maioria dos casos, métodos analíticos foram empregados como possíveis fontes de solução. Porém, tais abordagens são pouco efetivas quando aplicadas em ambientes internos e/ou casos em que os terminais não se encontram em situação de visada direta, ou seja, sem obstáculo algum entre eles.

Redes neurais artificiais podem ser utilizadas para diversas finalidades, especialmente reconhecimento de padrões e modelagem de funções. Pela natureza do aprendizado de máquina, tais redes alcançam melhores resultados onde métodos analíticos (PRIYANTHA; CHAKRABORTY; BALAKRISHNAN, 2000), (AIPING et al., 2009), (ABDELSALAM; OLARIU, 2009), não são capazes de aproximar a posição de nodos com precisão razoável. Uma rede neural artificial ao ser treinada para uso em um ambiente específico, leva em consideração todos os fatores presentes neste, incluindo-se obstáculos entre nodos e ruídos nas leituras dos sinais utilizados na inferência da localização dos nodos. Alguns trabalhos foram publicados corroborando o bom desempenho de redes neurais artificiais na localização de nodos, como por exemplo em (SHAREEF; ZHU; MUSAVI, 2007) e (RAHMAN; PARK; KIM, 2009), mas em menor número quando comparados aos trabalhos que abordam métodos analíticos ((AIPING et al., 2009), (ABDELSALAM; OLARIU, 2009)).

Porém, uma desvantagem que surge é a necessidade de refazer o treinamento das

RNAs caso o ambiente onde estas estão sendo aplicadas sofra mudanças no sentido de afastar muito as condições de operação das condições do treino aplicado. Quanto mais as condições de operação das RNAs se afastarem das condições de treino, menos precisão terão as estimativas de localização geradas por este sistema. Este fato pode explicar o baixo número de trabalhos publicados utilizando RNAs para localização de nodos.

Logo, é possível notar que existe bastante espaço para novas pesquisas relacionando estes temas, especialmente no que tange a seleção da melhor estrutura de rede neural artificial aplicável para cada tipo específico de rede de sensores sem fio. Analisando este quesito, fica evidente a necessidade de um método mais prático e preciso que execute esta tarefa. Nos trabalhos analisados, este procedimento era de responsabilidade do projetista, que tomava decisões após vários ciclos de tentativa e erro durante a configuração dos parâmetros da rede neural artificial.

Sendo assim, a motivação para a realização deste trabalho reside no bom desempenho e pouca exploração de redes neurais artificiais na localização de nodos em redes de sensores sem fio. Aliado a isto, é notável o fato de não existir uma metodologia clara ou algum sistema que seja capaz de selecionar a melhor estrutura de RNA a ser aplicada para localização em uma determinada RSSF.

1.2 Objetivos

Este trabalho objetiva o estudo do desempenho de redes neurais artificiais aplicadas ao problema de localização de nodos em redes de sensores sem fio. Os trabalhos presentes na literatura não apresentam uma metodologia clara no que diz respeito a seleção dos parâmetros estruturais das redes neurais artificiais (número de camadas, número de neurônios artificiais, funções de transferência) aplicadas em sistemas de localização.

Por esta razão, outro objetivo é propor novas abordagens para o tema, especialmente no que se refere ao método de seleção dos parâmetros para a estruturação das RNAs, tarefa que neste trabalho é realizada por metaheurísticas.

1.3 Contribuições do Trabalho

As contribuições presentes neste trabalho podem ser resumidas em:

- proposta de aplicação de metaheurísticas objetivando automatizar o processo de

seleção da melhor estrutura de RNA aplicável para uma determinada RSSF;

- proposta de um sistema de localização composto por duas RNAs em cascata, sendo cada uma delas otimizada individualmente através de metaheurísticas;
- realização de uma análise da influência exercida pela quantidade e posição dos nodos-âncora sobre as estruturas de RNAs geradas e seus desempenhos quanto ao erro das estimativas de localização;
- realização de uma análise da influência exercida pelo tamanho das amostras de treinamento sobre as estruturas de RNAs geradas e seus desempenhos quanto ao erro das estimativas de localização.

Estas contribuições representam um avanço para o projeto de sistemas de localização utilizando RNAs, visto que estes pontos não são abordados nos trabalhos presentes na literatura.

1.4 Organização da Dissertação

Esta dissertação encontra-se organizada da seguinte forma:

- Capítulo 2: Apresenta uma fundamentação teórica sobre os elementos envolvidos neste trabalho, fazendo uma revisão sobre sistemas de localização com e sem a utilização de redes neurais artificiais;
- Capítulo 3: Detalha a metodologia empregada para conduzir as simulações utilizadas para validar as técnicas de localização propostas neste trabalho, passando pela configuração das ferramentas de software utilizadas;
- Capítulo 4: Apresenta os resultados das simulações descritas no capítulo de metodologia fazendo uso de gráficos, tabelas e figuras. Também apresenta discussões relacionando as variáveis presentes nas diversas topologias simuladas, bem como comentários a respeito dos resultados obtidos em termos de erro de localização;
- Capítulo 5: Neste capítulo são apresentadas as conclusões baseadas nas análises feitas nos capítulos anteriores, bem como propostas para trabalhos futuros.

2 FUNDAMENTAÇÃO TEÓRICA

2.1 Redes de Sensores Sem Fio

2.1.1 Introdução

Os avanços no campo da microeletrônica permitiram o desenvolvimento de novos microcontroladores e sensores, sendo estes cada vez menores e com consumo de energia reduzido, além de baixo custo. Estas características dos novos dispositivos originados a partir das novas tecnologias são pré-requisitos para aplicações em redes de sensores sem fio.

Uma rede de sensores sem fio é formada por um grande número de nodos sensores, tendo como objetivo, geralmente, o monitoramento de grandezas físicas (por exemplo: temperatura, umidade, pressão, etc.) em algum ponto do espaço. Os terminais da rede devem estar dispostos de forma que possam comunicar-se entre si e propagar os dados obtidos através de seus transdutores até uma central, onde serão utilizados. A estrutura básica de uma rede de sensores sem fio pode ser vista na figura 1.

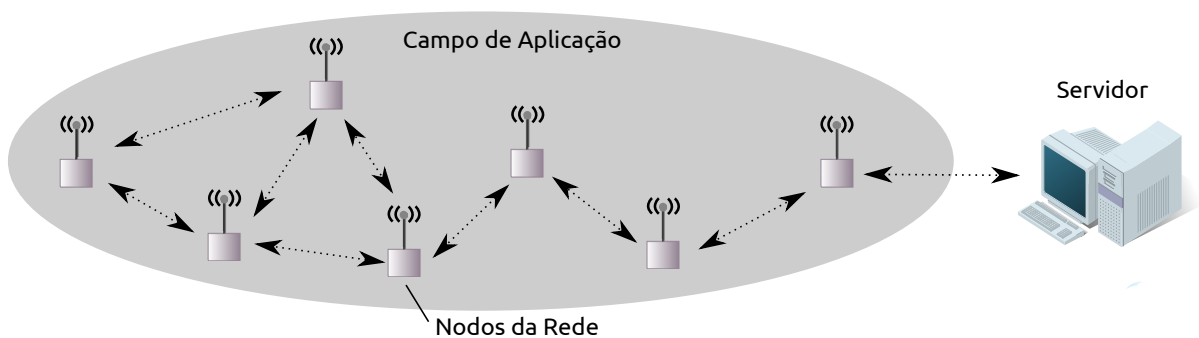


Figura 1: Rede de sensores sem fio.

2.1.2 Características dos Nodos Sensores

Os nodos sensores presentes em RSSFs são dispositivos capazes de coletar informações através de um ou mais transdutores, realizar um leve processamento de tais dados, armazenando-os e/ou transmitindo-os para nodos vizinhos de forma a chegarem em seu destino para posterior análise. As principais características dos nodos sensores estão descritas a seguir.

2.1.2.1 Tamanho Reduzido

Especialmente em algumas aplicações, o tamanho físico é um fator muito importante devido à necessidade de se embutir os nodos sensores em outros objetos ou até mesmo acoplá-los em animais para estudos ambientais. A consequência deste fato em nível mais baixo é o compartilhamento de hardware (placas de circuito impresso ou até mesmo uma única pastilha de silício) pelos elementos lógicos principais de um nodo: microcontrolador, sensores e transceptor de rádio.

2.1.2.2 Baixo Custo

Uma rede de sensores sem fio pode ser composta por centenas ou até mesmo milhares de nodos, sendo imperativo o baixo custo destes. A necessidade de manter o baixo custo dos nodos implica na utilização de microcontroladores com pouca memória de dados e de programa, bem como um reduzido e muitas vezes insubstituível suprimento de energia.

Apesar da baixa capacidade de processamento e autonomia energética dos nodos sensores, a quantidade de memória disponível e o processamento da rede como um todo pode ser considerada grande, tendo em vista o grande número de terminais trabalhando em paralelo.

2.1.2.3 Baixo Consumo

O consumo de energia geralmente é uma das principais preocupações durante o projeto de uma RSSF, visto que os nodos geralmente operam por meio de baterias que, na maioria das vezes, não são recarregáveis ou substituíveis devido às localizações de difícil acesso dos nodos. Outro empecilho para resolver o problema de renovação da fonte de energia reside no fato de que as RSSFs podem ser compostas de milhares de nodos, tornando inviável a substituição de baterias ou a incorporação de elementos de recarga (alto custo).

A redução de consumo pode ser alcançada através de hardware de baixo consumo e também através da racionalização das tarefas impostas aos nodos sensores. Todos os elementos do nodo podem permanecer a maior parte do tempo em *sleep mode*, ou seja, modo de baixo consumo, sendo ativados periodicamente para a execução de suas tarefas de medição e transmissão / armazenamento de dados.

2.1.2.4 Comunicação Sem Fio

Os ambientes nos quais os nodos sensores são geralmente utilizados são desprovidos de infra-estrutura para comunicação em rede, fazendo da comunicação sem fio um fator chave do sistema, especialmente se terminais móveis são utilizados.

2.1.2.5 Escalabilidade e Auto-Organização

Os nodos de redes de sensores sem fio devem ser capazes de adaptar-se às eventuais mudanças no número de nodos que as compõem. Esta característica é muito importante pois alguns terminais podem ficar inativos devido a falhas ou falta de energia e, mesmo nestas condições, o tráfego de dados da rede não deve ser prejudicado. Além disso, a conectividade da rede pode sofrer variações devido ao deslocamento de nodos móveis, gerando uma nova rota para o tráfego de dados e impondo uma nova adaptação da rede em resposta a estes eventos.

2.1.2.6 Estrutura Interna

Um terminal de uma rede de sensores sem fio é composto por quatro blocos principais, os quais podem ser vistos na figura 2:

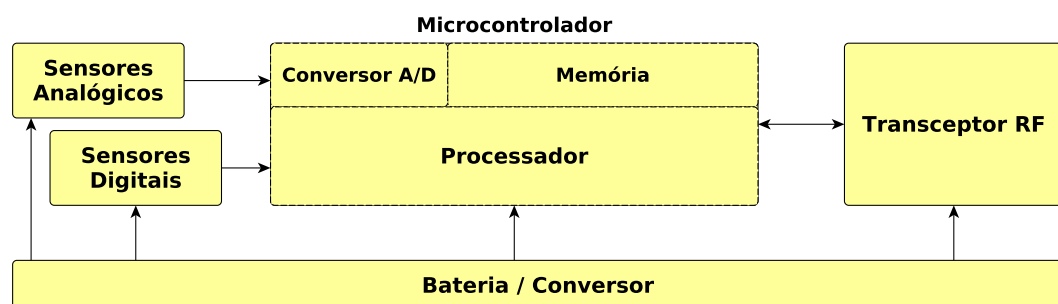


Figura 2: Estrutura de um nodo.

- **Unidade de sensoriamento:** Um sensor é um elemento que converte uma grandeza física em um sinal elétrico. Os sensores podem ser classificados de diversas formas, sendo uma das mais utilizadas a classificação como ativo ou passivo. Os transdutores que necessitam de uma fonte de alimentação externa são classificados como ativos, e gastam energia mesmo quando nada está sendo medido. Já os sensores passivos conseguem gerar seus sinais elétricos sem o uso de alimentação externa. A maioria dos sensores necessita de circuitos condicionadores de sinais, amplificando-os, filtrando-os, transpondo níveis de tensão e fazendo o casamento de impedâncias para que possam ser utilizados nas entradas dos microcontroladores.
- **Unidade de processamento:** Esta unidade é responsável por coordenar a coleta de dados dos sensores, o processamento das leituras e a comunicação com o meio externo, e sempre levando em consideração as restrições de energia do terminal. Existem vários modelos de microcontroladores, microprocessadores e FPGAs que podem ser integrados nos nodos sensores, o que permite uma certa flexibilidade neste sentido.
- **Unidade de comunicação:** A unidade de comunicação permite que os dados extraídos dos sensores sejam enviados aos demais nodos da rede e, conseqüentemente, ao seu destino final. Como a transmissão e a recepção dos dados é baseada em ondas eletromagnéticas, o projeto e o tamanho das antenas é um dos principais desafios. A comunicação via radiofrequência entre os nodos possui vantagens, visto que é fácil de usar, de ser integrada aos dispositivos e por ser uma tecnologia já bem estabelecida. A energia gasta depende do tipo de modulação utilizada, da potência de transmissão e da taxa de transferência de dados. Um ponto importante a ser observado é que o transceptor de rádio, quando em estado de espera, tem o mesmo consumo de uma recepção de dados. Logo, os algoritmos utilizados devem atuar no sentido de evitar este desperdício de energia.
- **Unidade de energia:** Fornece energia para o nodo sensor, sendo constituída geralmente por uma bateria e um conversor CC-CC. Para manter o baixo custo dos nodos, as baterias geralmente não possuem sistema para recarga, salvo em casos especiais como a alimentação de nodos do tipo âncora, por exemplo. Devido ao tamanho reduzido dos sensores, as baterias devem ser pequenas o suficiente para cumprir os requisitos de suas aplicações.

2.1.2.7 Mote Comercial *MICAz*

Este trabalho tem como referência para implementações de redes de sensores o mote *MICAz*, da empresa *Crossbow*. Suas características foram introduzidas nas simulações para que estas reproduzissem cenários reais envolvendo aplicações com este mote. Ele é um módulo de 2,4 GHz de frequência com foco em aplicações com baixo consumo de energia, sendo utilizado em segurança e monitoramento em ambientes fechados, redes de sensores de grande escala e aplicações que monitoram dados de diversas fontes, tais como acústica, vídeo e vibração.

O transceptor respeita o protocolo IEEE 802.15.4 e opera entre 2,4 até 2,48 GHz. O MPR2400CA tem características como:

- **sequência direta de espalhamento do espectro:** consegue densidade espectral de potência baixa através do espalhamento da potência do sinal em uma banda larga. Esta técnica provê resistência ao ruído e favorece a utilização de criptografia através de saltos pseudo aleatório de frequência;
- **alta taxa de transmissão de dados:** capaz de transmitir até 250 kbps, o que para uma aplicação de rede de sensores costuma satisfazer às demandas de velocidade;
- **utilização da plataforma *Mote Works*:** baseado no sistema operacional *TinyOS*, oferece capacidades de rede de malha *ad-hoc*, ferramentas de desenvolvimento e softwares para integração entre a aplicação embarcada e interface de clientes para análise de dados e configuração da rede.

A tabela 1 ilustra as especificações principais do mote. As características que influem diretamente na simulação das redes são a potência de transmissão (-24 dBm até 0 dBm), a sensibilidade do receptor (-94 dBm) e a taxa de transmissão (250 kbps).

2.1.3 Características das Redes de Sensores

Em uma rede de sensores sem fio típica, os nodos estão espacialmente distribuídos em uma área e cooperam para monitorar algumas condições físicas e ambientais. Usualmente, uma RSSF e algumas de suas características são dependentes da aplicação pretendida. Porém, algumas características podem ser citadas como sendo intrínsecas a RSSFs.

- Limitações de energia, memória, processamento e potência de transmissão;

Tabela 1: Especificações do mote comercial *MICAz*

Parâmetro	Especificação
Processador	
Memória de Programa (Flash)	128 Kbytes
Memória para Medidas (Flash)	512 Kbytes
Memória de Configuração (EEPROM)	4 Kbytes
Conversor A/D	10 bits
Interfaces	E/S digital, I2C, SPI
Corrente no modo ativo	8 mA
Corrente no modo de repouso	$< 15 \mu A$
Transceptor	
Frequência	2400 MHz até 2483,5 MHz
Taxa de transmissão	250 kbps
Potência RF	-24 dBm até 0 dBm
Alcance em uso externo	75 m até 100 m
Alcance em uso interno	20 m até 30 m
Corrente no modo recepção	19,7 mA
Corrente no modo transmissão	11mA até 17,4 mA
Corrente no modo de espera	$20 \mu A$
Corrente no modo de repouso	$1 \mu A$

- Transceptores de rádio com curto alcance, acarretando em comunicação P2P;
- Alta densidade de nodos e esforço cooperativo destes para monitorar o ambiente;
- Mudanças frequentes na conectividade devido à falhas e movimentação de nodos;
- Capacidade de auto-organização;
- Necessidade de conhecimento da posição dos terminais.

2.1.4 Exemplos de Aplicações

As redes de sensores sem fio foram originalmente concebidas para aplicações militares, porém aplicações em outras áreas são perfeitamente possíveis. Alguns exemplos de utilização de RSSFs são mostrados a seguir.

2.1.4.1 Militar

Para fins de monitoramento de um campo de batalha, diversos nodos podem ser distribuídos por vias aérea e terrestre. Após estabelecida a comunicação entre os terminais, pode-se detectar a presença de unidades militares hostis, por exemplo. Outra aplicação para fins militares pode ser vista em, onde diversos terminais com sensores acústicos de baixo custo podem ser usados para inferir a posição de atiradores escondidos, bem como a trajetória das balas.

Geralmente as RSSFs fazem parte dos sistemas de Comando, Controle, Comunicações, Computação, Inteligência, Vigilância, Reconhecimento e Mira (*C⁴ ISRT*, em inglês). Ainda é possível fazer o reconhecimento de forças amigas, controle de equipamento e munição, avaliação de danos em batalhas e detecção de ataques biológicos, nucleares ou químicos.

2.1.4.2 Domótica

A possibilidade de integração de nodos sensores em objetos de uso cotidiano permite automatizar uma série de tarefas usuais e dispositivos relacionados com sistemas de segurança e ar-condicionado, por exemplo. O monitoramento contínuo dos ambientes da residência permite uma rápida resposta dos sistemas eletrônicos que controlam equipamentos responsáveis pela segurança e conforto do local.

2.1.4.3 Saúde

O uso de RSSFs na medicina pode ser exemplificado por sistemas de monitoramento de pacientes, sendo que os dados sobre os sinais vitais da pessoa (pressão do sangue, frequência cardíaca, temperatura do corpo, etc.) podem ser enviados diretamente para o banco de dados do médico através da tecnologia sem fio e pela internet. A aplicação deste tipo de sistema ajuda no diagnóstico de doenças e até mesmo na predição destas.

2.1.4.4 Logística

Algumas mercadorias devem ser transportadas dentro de devidas condições de temperatura e umidade, caso contrário podem sair dos padrões de qualidade. Além disso, grandes empresas com sedes em vários países podem rastrear o caminho trilhado e as condições de seus produtos durante o transporte e armazenagem.

2.1.4.5 Vigilância

Além de monitoramento de campos de batalha em aplicações militares, também é possível monitorar acesso a determinadas propriedades privadas e equipamentos. Empresas de telecomunicações e concessionárias de energia elétrica podem, por exemplo, fazer a vigilância de suas infraestruturas através de redes de sensores sem fio, tarefa esta que seria inviável de ser feita por humanos devido ao alto custo.

2.1.4.6 Monitoramento Ambiental

Com diversos sensores distribuídos em uma determinada área geográfica é possível detectar com antecedência, por exemplo, abalos sísmicos, atividades vulcânicas, tempestades, furacões, tornados e incêndios. Além disso, é possível que seja feito um monitoramento a respeito do comportamento de alguns animais integrando sensores a eles.

2.2 Localização em Redes de Sensores Sem Fio

2.2.1 Introdução

O conhecimento das posições dos nodos em uma rede de sensores sem fio é um requisito inerente a muitas aplicações. Geralmente, as RSSFs têm como objetivo fazer o sensoramento de alguma grandeza física em alguma área geográfica. Deste modo, tomando como exemplo um sistema de monitoramento da umidade do solo em uma determinada área agrícola, as medições feitas pelos nodos sensores teriam pouca utilidade se não estivessem acompanhadas da localização dos terminais.

Idealmente, a posição dos nodos poderia ser proveniente de um receptor GPS. Porém, como os terminais utilizados nas redes de sensores são de baixo custo, capacidade limitada de processamento e disponibilidade limitada de energia, o uso de GPS torna-se inviável. Desta forma, muitos esforços têm sido feitos em pesquisas direcionadas a métodos de localização que não necessitem de módulos de *hardware* adicionais e, ao mesmo tempo, que sejam de baixo consumo e de baixa complexidade computacional.

O objetivo de um algoritmo de localização é estimar as coordenadas dos nodos em uma rede. Para desempenhar tal ação, os métodos precisam utilizar como insumos os dados ou medições de sinais presentes na rede de sensores sem fio. Como os nodos são conectados através de ondas de radiofrequência, é possível que a inferência de localização seja baseada em dados como, por exemplo, a força do sinal referente a um pacote de dados transmitido

por um nodo de referência e recebido por um nodo desconhecido. A força do sinal de radiofrequência que chega ao receptor está relacionada com a distância entre este e o transmissor. Desta forma, partindo-se das distâncias entre o nodo desconhecido e os nodos de referência, é possível estimar a posição dos nodos desconhecidos. Porém, a medição da força do sinal recebido é distorcida em função do respectivo canal de propagação, resultando em estimativas erradas de distâncias. Sendo assim, para casos em que medições de alguma grandeza são utilizadas para inferir posição de nodos, as técnicas de localização visam minimizar as distorções nestes sinais.

2.2.2 Classificação das Técnicas de Localização

A grande quantidade de trabalhos acerca do problema de localização permite uma série de classificações, porém as mais pertinentes para este estudo são:

- Quanto ao tipo de observação (grandeza física ou dado de conectividade) utilizada na localização;
- Quanto à forma de processamento da localização
- Quanto ao local de processamento das observações

2.2.2.1 Conforme Tipo de Observação

As técnicas de localização geralmente são classificadas em duas grandes categorias no que diz respeito ao tipo de observação feita para estimar as posições dos nodos: *range-based* e *range-free*.

- Categoria *range-based*

Nesta categoria se enquadram todos os métodos que empregam medições de grandezas físicas para estimar a posição dos nodos na rede (ALIPPI; MOTTARELLA; VANINI, 2005). Tais grandezas podem ser, por exemplo, a força do sinal de radiofrequência recebido, o tempo de chegada do sinal recebido e o ângulo de chegada do sinal recebido.

- Categoria *range-free*

Os algoritmos pertencentes a esta categoria se utilizam apenas de dados de conectividade para derivar as posições dos terminais na rede (OLIVEIRA et al., 2010) e

(BULUSU; HEIDEMANN; ESTRIN, 2000). Estes dados estão presentes nos pacotes de dados trocados entre os nodos da rede, e podem ser, por exemplo, as coordenadas cartesianas dos nodos-âncoras acompanhadas da quantidade de pacotes recebidos deste mesmo nodo.

2.2.2.2 Conforme Caráter Individual/Coletivo do Processamento da Localização

A localização dos nodos pode ser feita de forma individual e local, como por exemplo em (OLIVEIRA et al., 2010) e (BULUSU; HEIDEMANN; ESTRIN, 2000) onde cada nodo é responsável por inferir sua própria localização. Também, o processo de localização pode ser cooperativo como visto em (BAGGIO; LANGENDOEN, 2006) e (HU; EVANS, 2004), de modo que mais informações possam ser incorporadas no processo, reduzindo assim o erro das estimativas.

2.2.2.3 Conforme o Local de Processamento das Observações

Partindo deste ponto de análise, o processamento das observações que são a base para a estimativa da localização dos nodos, pode ser feito de forma centralizada ou distribuída. Na modalidade centralizada, todas as observações são enviadas para uma central que as processa e em seguida retorna os resultados para os nodos, como por exemplo em (DOHERTY; PISTER; GHAOUI, 2001) e (SHANG et al., 2003). Já na modalidade de processamento distribuído, cada nodo é responsável pelo processamento de suas próprias observações, como pode ser visto em (SHANG; RUMMLER, 2004) e (JI; ZHA, 2003).

2.2.3 Algoritmos de Localização Encontrados na Literatura

Esta seção tem como objetivo comentar algumas das abordagens encontradas na literatura para a localização de nodos em redes de sensores sem fio sem a utilização de redes neurais artificiais.

Como dito anteriormente, os algoritmos representantes da categoria *range-based* se utilizam de medições de distâncias ou ângulos entre nodos para calcular a localização dos nodos.

O tempo que um sinal transmitido por um terminal âncora leva para atingir a antena de um terminal desconhecido pode ser utilizado para inferir a distância entre eles. Desta forma, um terminal desconhecido pode ter sua posição inferida através da observação do

tempo de chegada (ToA, do inglês *Time of Arrival*) de sinais transmitidos por nodos-âncoras. Supondo um sinal saindo de um determinado terminal âncora e chegando em um dado terminal desconhecido com um ToA sendo t , a distância entre eles pode ser encontrada através da relação:

$$\hat{d} = t \cdot v_p \quad (2.1)$$

sendo que v_p representa a velocidade da propagação da onda (CAPKUN; HAMDI; HUBAUX, 2001).

As abordagens que se utilizam do tempo de chegada como princípio de funcionamento apresentam algumas desvantagens em relação às demais. A necessidade de sincronia entre os nodos da rede é uma delas, acompanhada da sensibilidade do método aos erros de medição nos tempos de chegada. Ocorre uma degradação da relação sinal-ruído do sinal recebido devido às componentes multipercurso do canal de propagação, sendo esta uma causa do aumento do erro nas estimativas de localização.

Outra maneira de se inferir distâncias entre nodos e então utilizá-las para estimar posições é através do valor médio da potência recebida, comumente referenciada como RSSI (do inglês *Received Signal Strength Indicator*) sendo este modelada como sendo inversamente proporcional à distância entre o transmissor e o receptor. Para potência medida em decibéis, tem-se:

$$\bar{P}_{dB} = P_{0,dB} - 10\eta \log_{10} \frac{d}{d_0} \quad (2.2)$$

sendo \bar{P}_{dB} a potência média recebida, d a distância entre transmissor e receptor, $P_{0,dB}$ a potência recebida a uma distância de referência d_0 e η o expoente de perda de percurso, que é dependente do meio de propagação. Com os parâmetros $P_{0,dB}$, d_0 e η conhecidos, é possível determinar a distância entre dois terminais (um âncora e um desconhecido) através da relação:

$$\hat{d} = d_0 10^{\frac{P_{0,dB} - \bar{P}_{dB}}{10 \cdot \eta}} \quad (2.3)$$

A partir das inferências das distâncias de um nodo desconhecido em relação a três ou mais nodos-âncoras, é possível estimar sua posição através do método da lateração (Apêndice B), sendo que uma topologia básica na qual a lateração pode ser empregada é

mostrada na figura 3.

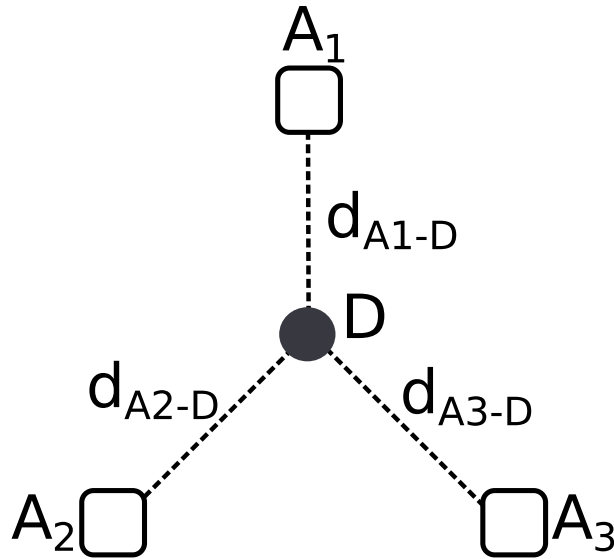


Figura 3: Estrutura mínima para aplicação de lateração em duas dimensões.

A vantagem da utilização da análise da potência média do sinal de radiofrequência recebido é que este hardware é barato e geralmente está presente na maioria dos nodos sensores acadêmicos e comerciais, a exemplo do MICAz fabricado pela Crossbow.

Como desvantagem, tem-se os erros nas estimativas das distâncias, que são a base do método de localização. Estas distorções são ocasionadas pelo desvanecimento do sinal imposto pelo canal de propagação. A presença de obstáculos entre transmissor e receptor pode acarretar em variações na potência recebida e, conseqüentemente, induzir erros nas estimativas de distância. A modelagem da influência do desvanecimento do sinal recebido geralmente é feita somando-se à potência média uma variável aleatória χ de distribuição gaussiana, possuindo média zero e variância σ_{dB}^2 , resultando na expressão:

$$\bar{P}_{dB} = P_{0,dB} - 10\eta \log_{10} \frac{d}{d_0} + \chi \quad (2.4)$$

Os métodos de localização que são baseados no uso de radiofrequência foram amplamente discutidos, por exemplo, em (GIROD; ESTRIN, 2001), (ALIPPI; MOTTARELLA; VANINI, 2005), (HIGHTOWER et al., 2000), (LORINCZ; WELSH, 2004) e (NI et al., 2004).

Além de técnicas de localização que empregam o ToA e o RSSI para estimar distâncias e posteriormente posições, também é muito conhecida a técnica que emprega o ângulo de chegada do sinal (AoA, do inglês *Angle of arrival*). Os sinais transmitidos pelos terminais desconhecidos chegam aos terminais âncoras de modo que são conhecidas informações direcionais (com relação a uma referência) destes. Desta forma, é possível estimar a

posição dos terminais desconhecidos através da triangulação (Apêndice B), sendo que a figura 4 é mostra uma topologia mínima para aplicação deste método.

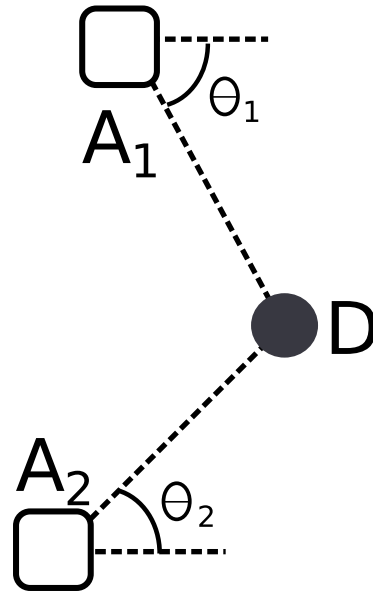


Figura 4: Estrutura mínima para aplicação de triangulação.

É notável que conhecendo-se a posição dos terminais âncora A_1 e A_2 bem como os ângulos θ_1 e θ_2 é possível determinar a posição do nodo desconhecido D . Porém, assim como nos métodos descritos anteriormente, o AoA (NICULESCU; NATH, 2003) também sofre distorções causadas pelo canal de propagação. Neste caso, estas distorções são modeladas como ruído aditivo de distribuição gaussiana, possuindo média igual ao valor do ângulo de chegada e desvio padrão σ_α .

A desvantagem deste método é facilmente notada observando-se o hardware necessário para sua operação. Para identificar o ângulo de chegada do sinal é necessário um arranjo de antenas ou antenas direcionais no receptor, prejudicando o tamanho e o consumo de tais nodos.

Já os algoritmos que utilizam dados do tipo *range-free* não utilizam medições de grandezas físicas, mas sim dados de conectividade, isto é, informações contidas nas mensagens trocadas entre os nodos. Essas abordagens são eficientes em termos de consumo de energia, porém são mais apropriadas para redes com alta conectividade. O exemplo mais difundido desta categoria de algoritmos de localização é o método proposto em (BULUSU; HEIDEMANN; ESTRIN, 2000).

Este algoritmo, chamado de Centroid, efetua a localização estimada de um nodo desconhecido utilizando sinais recebidos de nodos âncora, os quais têm conhecimento sobre suas posições devido ao fato de possuírem GPS integrado ou por programação

prévia. Os âncoras são responsáveis por enviar, periodicamente em um tempo T e para todos os nodos da rede que estejam em seu raio (R) de alcance (modo *broadcast*), uma mensagem (também chamada de *beacon*) contendo suas coordenadas x e y , além de suas identificações i na rede (XA_i, YA_i).

Os nodos desconhecidos, ao receberem estas mensagens de *broadcast* enviadas pelos terminais âncora, iniciam o processamento que resultará na estimativa de localização. Durante um certo período de tempo t , cada mensagem recebida incrementa um contador específico para cada remetente. Cada nodo âncora deve transmitir um determinado número de mensagens, denominado S (tamanho da amostra), durante o período de amostragem t .

Os dados coletados serão utilizados para estimar as coordenadas do nodo desconhecido ao término do tempo t . O primeiro passo é calcular um valor chamado métrica de conectividade, que vai apontar quais âncoras (e suas respectivas mensagens) devem influir no cálculo da posição estimada. E expressão para o cálculo desta métrica (CM_i , do inglês *Connectivity Metric*) é:

$$CM_i = \frac{N_{recv(i,t)}}{N_{sent(i,t)}} \cdot 100 \quad (2.5)$$

sendo que $N_{recv(i,t)}$ é o número de mensagens que foram enviadas pelo nodo âncora i e recebidas no período de tempo t e $N_{sent(i,t)}$ é o número de mensagens que foram enviadas pelo nodo-âncora i no período de tempo t .

Os âncoras que excederem um valor determinado pelo usuário (CM_{thresh}) para a métrica de conectividade farão parte do cálculo final das coordenadas estimadas (X_{est}, Y_{est}), o qual é mostrado na fórmula

$$(X_{est}, Y_{est}) = \left(\frac{XA_{i_1} + \dots + XA_{i_k}}{k}, \frac{YA_{i_1} + \dots + YA_{i_k}}{k} \right) \quad (2.6)$$

sendo que k representa o número de nodos âncora que ultrapassaram o valor de (CM_{thresh}).

Assim, a localização do nodo desconhecido é considerada como sendo a intersecção das regiões de conectividade dos nodos-âncora que fizeram parte dos cálculos, sendo que esta posição é definida como sendo o centro dos pontos (nodos-âncora) tomados como referência.

Este algoritmo possui alguns parâmetros que podem ser configurados pelo usuário e, preferencialmente, levando em conta a aplicação final do método. Como exemplos destes

parâmetros configuráveis temos T , t e S , além do próprio raio (R) de alcance do nodo, sendo que este último depende do hardware empregado. A relação entre o tempo de amostragem t , o tamanho da amostra S e o período T é expressa pela fórmula

$$t = (S + 1 - \epsilon) \cdot T (0 < \epsilon \ll 1) \quad (2.7)$$

sendo que o termo ϵ é muito próximo de zero, porém seu significado não foi deixado claro em (BULUSU; HEIDEMANN; ESTRIN, 2000).

Como vantagens, o método Centroid traz a simplicidade de cálculos e o fato de não precisar transmitir mensagens (*receiver-based*), sendo uma desvantagem a dependência da sobreposição dos alcances dos rádios dos nodos-âncoras. A figura 5 mostra as diferentes regiões de localização baseadas na granularidade gerada pelos pontos de referência.

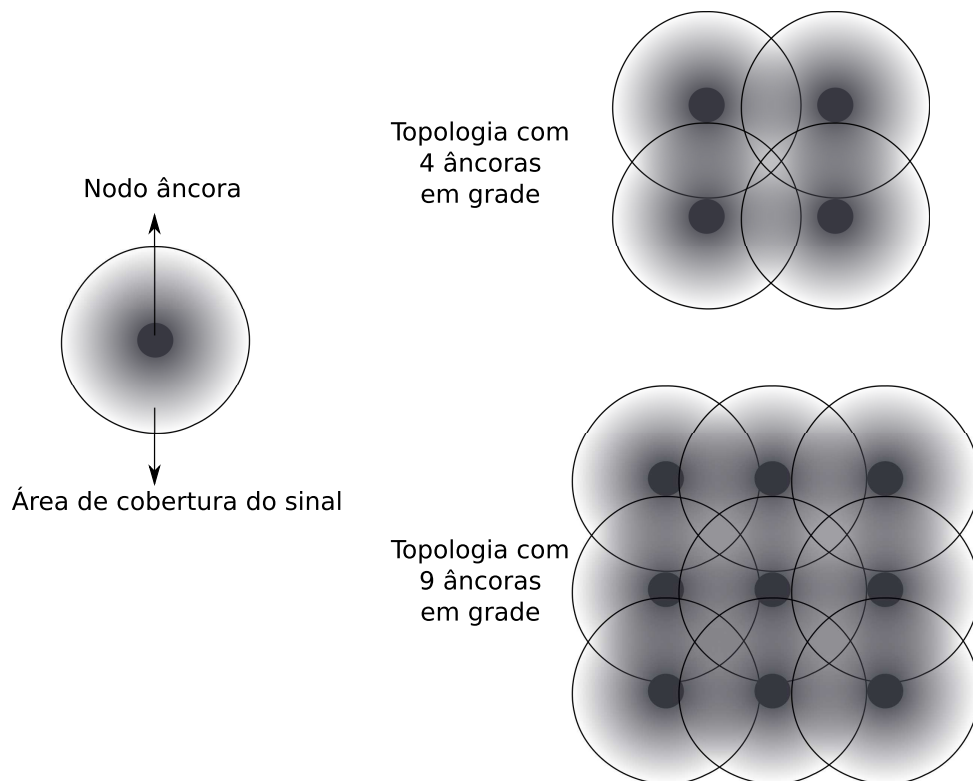


Figura 5: Regiões de localização em diferentes granularidades.

2.3 Redes Neurais Artificiais

2.3.1 Introdução

As redes neurais artificiais (RNAs) (SILVA; SPATTI; FLAUZINO, 2010) foram inspiradas no sistema nervoso de seres vivos, e são modelos matemáticos capazes de classificar padrões

ou aproximar funções, por exemplo. São formadas por neurônios artificiais dotados de entradas e saídas, além de pesos para ponderação das entradas. Tais neurônios podem ser interligados de diversas maneiras, mudando assim a nomenclatura e funcionalidade das redes. São capazes de adquirir conhecimento mediante treinamento, tornando-se aptas a exercer suas funções após este procedimento.

2.3.1.1 Características Principais

As redes neurais artificiais possuem diversas características que as fazem ferramentas extremamente úteis para uma gama diversa de aplicações. Elas podem ser facilmente prototipadas, uma vez que, após o período de treinamento, o resultado para o problema em questão estará na saída ao custo da execução de algumas operações matemáticas elementares.

No período de treinamento a rede é apresentada a um conjunto de entradas sobre um determinado problema, bem como o seu respectivo conjunto respostas desejadas. Desta forma, a RNA pode aprender sobre o processo em questão e modelar a relação entre entradas e saídas de forma satisfatória. Mesmo perante conjuntos de entradas nunca antes vistos, a RNA, após treinada, consegue generalizar uma resposta.

Como geralmente é formada por mais de um neurônio artificial, uma RNA acaba por ser naturalmente tolerante a falhas, pois mesmo se alguma de suas subestruturas apresentar defeito, devido às inúmeras interconexões entre elas, as demais conseguem ainda assim completar a tarefa para a qual a rede foi projetada.

2.3.2 Neurônio Artificial

Os modelos de neurônios artificiais foram inspirados a partir da geração e propagação de impulsos elétricos pela membrana celular dos neurônios biológicos, e são modelos bem simplificados destes últimos. Basicamente, o neurônio artificial coleta os sinais de suas entradas ponderadas e produz uma saída aplicando ainda uma função de ativação. A representação de um neurônio artificial pode ser visto na figura 6.

Os sinais de entrada são representados na figura 6 pelo conjunto x_1, x_2, \dots, x_n , sendo que estes são análogos aos impulsos elétricos captados pelos neurônios biológicos. As ponderações que ocorrem no neurônio biológico são representadas no neurônio artificial pelo conjunto de pesos sinápticos w_1, w_2, \dots, w_n . Deste modo, a importância de cada uma das entradas x_i é ponderada através de suas multiplicações pelos respectivos pesos

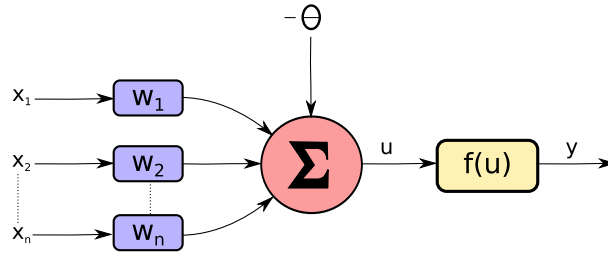


Figura 6: Modelo de neurônio artificial.

sinápticos w_i .

Além das entradas e dos pesos sinápticos, também podem ser vistos na figura 6 o combinador linear, o limiar de ativação, o potencial de ativação, a função de ativação e o sinal de saída propriamente dito. O combinador linear Σ produz um valor de potencial de ativação agregando todas as entradas já ponderadas. O limiar de ativação θ atua no sentido de selecionar o nível adequado para que a saída do combinador linear possa gerar um disparo. O potencial de ativação u , quando maior que zero, produz um potencial excitatório, sendo seu valor a diferença entre o combinador linear e o limiar de ativação. A função de ativação f apenas limita a saída do neurônio dentro de uma faixa de valores aceitáveis. Finalmente, a saída y é o valor final gerado por um neurônio artificial.

Nas equações 2.8 e 2.9 pode-se observar as duas expressões que representam o resultado produzido por um neurônio artificial.

$$u = \sum_{k=1}^n w_k \cdot x_k - \theta \quad (2.8)$$

$$y = f(u) \quad (2.9)$$

De forma mais direta, pode-se descrever como passos para o funcionamento de um neurônio artificial a apresentação do conjunto de entradas para determinado caso, a multiplicação de cada entrada por um respectivo peso, subtração do potencial de ativação da soma ponderada das entradas, aplicação da função de transferência visando limitar a saída do neurônio e, finalmente, determinar a real saída no neurônio baseando-se em seu potencial de ativação e sua função de ativação.

As funções de ativação, considerando-se o total de seus domínios de definição, geralmente são classificadas em dois grupos principais: funções parcialmente diferenciáveis e funções totalmente diferenciáveis.

2.3.2.1 Funções de Transferência Parcialmente Diferenciáveis

São as funções que possuem pontos em que suas derivadas de primeira ordem são inexistentes.

- Função degrau (*heavyside/hard limiter*)

A função degrau assumirá valor unitário positivo quando o potencial de ativação do neurônio for maior ou igual a zero. Caso contrário, ela assumirá valor nulo. A representação gráfica desta função é ilustrada na figura 7.

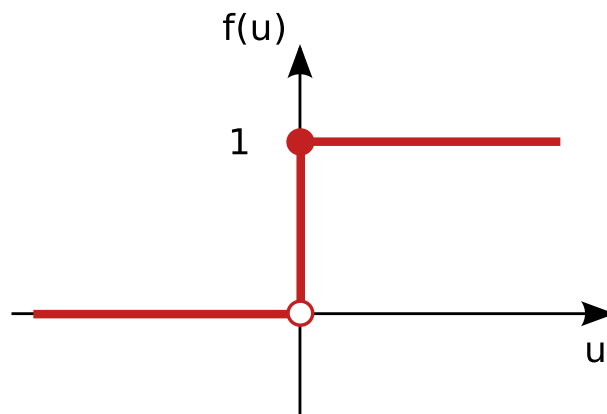


Figura 7: Função degrau.

- Função degrau bipolar (*symmetric hard limiter*)

Quando o potencial de ativação do neurônio for maior positivo, a função assumirá valor unitário positivo. Quando o potencial for nulo, o resultado também será nulo. Quando o potencial for negativo, a saída apresentará resultado unitário negativo. A representação gráfica é apresentada na figura 8.

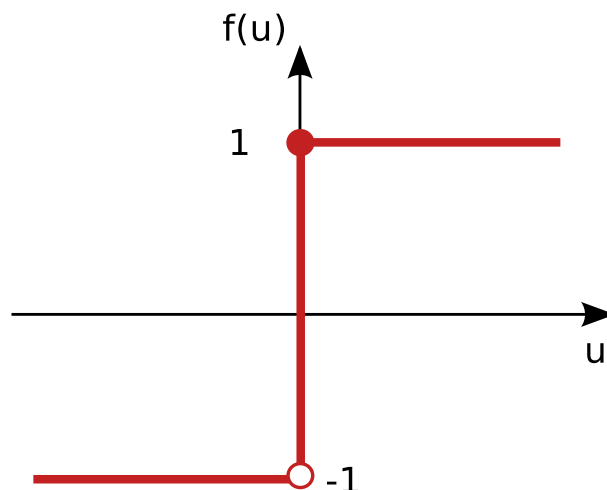


Figura 8: Função degrau bipolar.

- Função rampa simétrica

Os valores retornados serão iguais aos próprios valores dos potenciais de ativação quando eles estiverem definidos no intervalos $[-a, a]$, restringindo-se aos valores limites em caso contrário. A figura 9 ilustra seu comportamento.

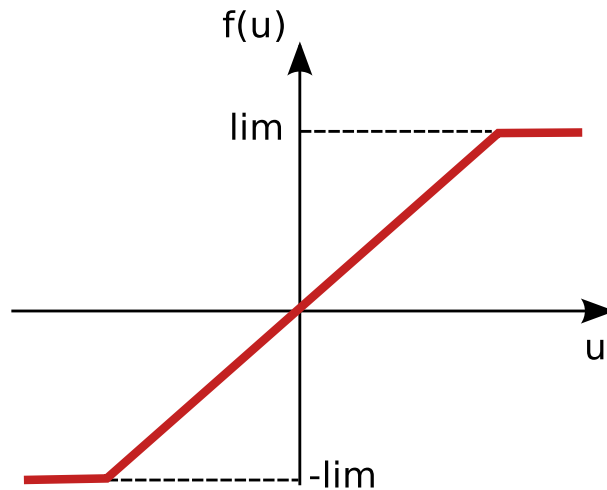


Figura 9: Função rampa simétrica

2.3.2.2 Funções de Transferência Totalmente Diferenciáveis

São as funções que possuem derivadas de primeira ordem em todos os pontos de seu domínio de definição.

- Função linear

Produz saídas idênticas aos valores do potencial de ativação. Também é chamada de identidade, sendo sua expressão matemática definida pela equação 2.10 e seu comportamento ilustrado na figura 10.

$$f(u) = u \quad (2.10)$$

Uma das aplicabilidades da função linear está na utilização de redes neurais artificiais como aproximadora universal de funções, onde o objetivo é mapear o comportamento entre as variáveis de entrada e saída de um processo.

- Função gaussiana

A saída da função de ativação gaussiana produzirá resultados iguais para aqueles valores de potencial de ativação u que estejam posicionados a uma mesma distância de seu centro (média), havendo simetria em relação a este. A função de ativação gaussiana

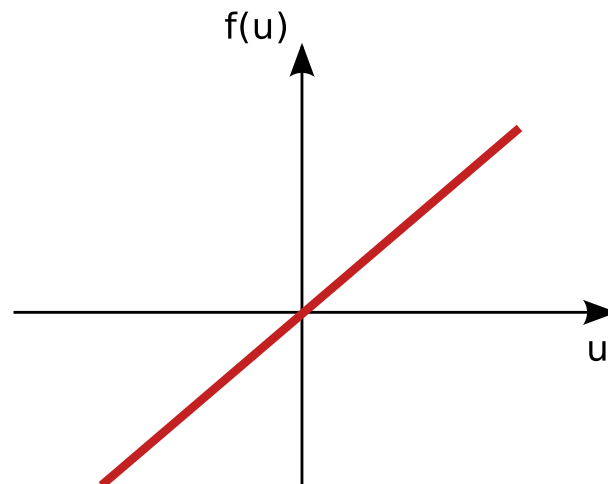


Figura 10: Função linear.

é dada pela equação 2.11.

$$f(u) = e^{-\frac{(u-c)^2}{2\sigma^2}} \quad (2.11)$$

O parâmetro c define o centro da função gaussiana e σ demonstra o desvio padrão associado à mesma, ou seja, o quão dispersos estão os valores em relação ao seu centro. A representação gráfica desta função é ilustrada na figura 11.

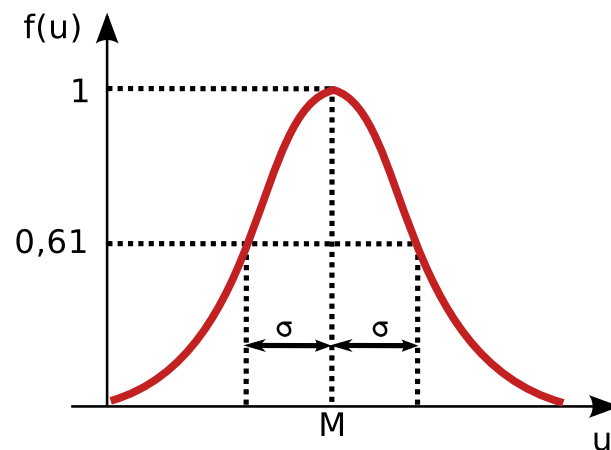


Figura 11: Função gaussiana.

- Função logística

A saída desta função assumirá sempre valores reais entre zero e um, sendo que sua expressão matemática é dada pela equação 2.12.

$$f(u) = \frac{1}{1 + e^{-\beta \cdot u}} \quad (2.12)$$

O valor β é uma constante real relacionada ao nível de inclinação da função logística frente ao seu ponto de inflexão. Sua representação gráfica pode ser vista na figura 12.

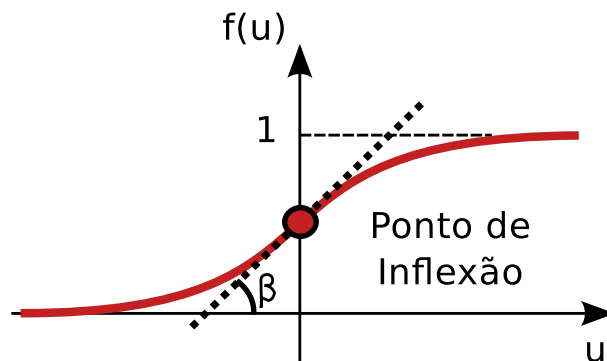


Figura 12: Função logística.

- Função tangente hiperbólica

O resultado de saída sempre assumirá valores reais entre -1 e 1 . A expressão matemática desta função é definida pela equação 2.13.

$$f(u) = \frac{1 - e^{-\beta \cdot u}}{1 + e^{-\beta \cdot u}} \quad (2.13)$$

Como é possível notar, β também estará associado ao nível de inclinação da função tangente hiperbólica em relação ao seu ponto de inflexão. A representação gráfica desta função pode ser vista na figura 13.

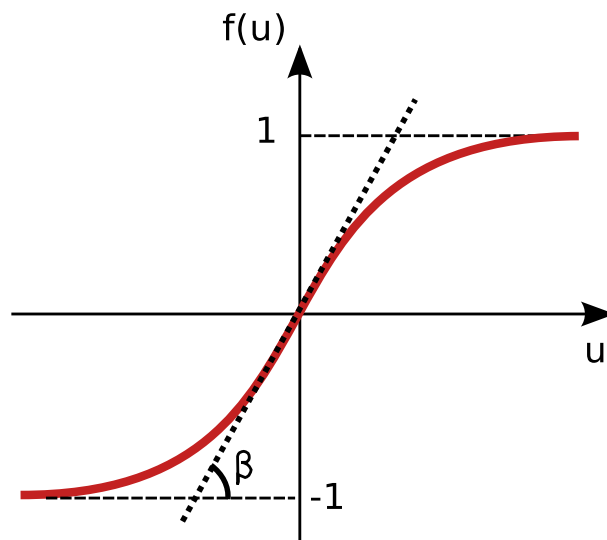


Figura 13: Função tangente hiperbólica.

Tanto a função logística como a função tangente hiperbólica pertencem à família das funções denominadas de sigmoidais.

2.3.3 Processos de Treinamento

O treinamento de uma rede neural artificial pode ser descrito como a aplicação de um conjunto de passos ordenados visando ajustar os pesos sinápticos e limiares de ativação dos neurônios de modo a melhor se adaptar ao processo em foco. Este procedimento objetivando ajustar o melhor ponto de trabalho da rede em um determinado contexto, é chamado de algoritmo de aprendizagem.

O procedimento de treinamento pode ser supervisionado ou não-supervisionado, sendo que para a realização do primeiro tipo deve-se ter à disposição os sinais de saída correspondentes para cada conjunto de entradas aplicado. No treinamento não-supervisionado, as saídas desejadas para cada conjunto de entradas não estão disponíveis. Desta forma, a rede deve se auto-organizar baseada nas particularidades presentes entre os componentes do conjunto total de amostras, criando grupos de dados semelhantes.

2.3.4 Arquiteturas de RNAs

A maneira como os neurônios artificiais estão dispostos uns em relação aos outros é definida pela arquitetura da rede neural artificial. Esta disposição de neurônios é estruturada através das ligações sinápticas dos neurônios. A topologia de uma rede neural, dentro de uma determinada arquitetura, pode ser definida como sendo as diferentes formas de composições estruturais passíveis de serem assumidas.

É possível que se tenha duas topologias pertencentes a uma mesma arquitetura, sendo que uma é composta por um número x de neurônios e a outra por um número y . Estas topologias, ainda que com o mesmo número de neurônios, podem ser diferenciadas baseando-se no tipo de função de ativação dos neurônios.

Uma rede neural artificial pode ser dividida em três partes, denominadas camadas, as quais são:

- Camada de entrada

É a camada que coleta os dados, sinais ou medições do meio externo, sendo que tais dados são geralmente normalizados para facilitar o processamento do algoritmo de treinamento, por exemplo. A normalização também contribui para facilitar as operações matemáticas da rede como um todo.

- Camadas escondidas, intermediárias ou ocultas

São aquelas camadas que possuem como responsabilidade a função de aprender sobre o processo em foco, de modo que praticamente todo o processamento interno da rede é realizado nestas camadas.

- Camadas de saída

É responsável pela apresentação dos resultados finais da rede, os quais foram produzidos baseados nas camadas anteriores.

As principais arquiteturas de redes neurais artificiais podem ser divididas em: redes *feedforward* (alimentação à frente) de camada simples, redes *feedforward* de camadas múltiplas, redes reticuladas e redes recorrentes. Tais classificações são advindas da disposição dos neurônios, bem como das interligações entre eles e a interligação das camadas.

Na figura 14 é possível observar todas as camadas que podem compor uma rede neural artificial do tipo *feedforward*.

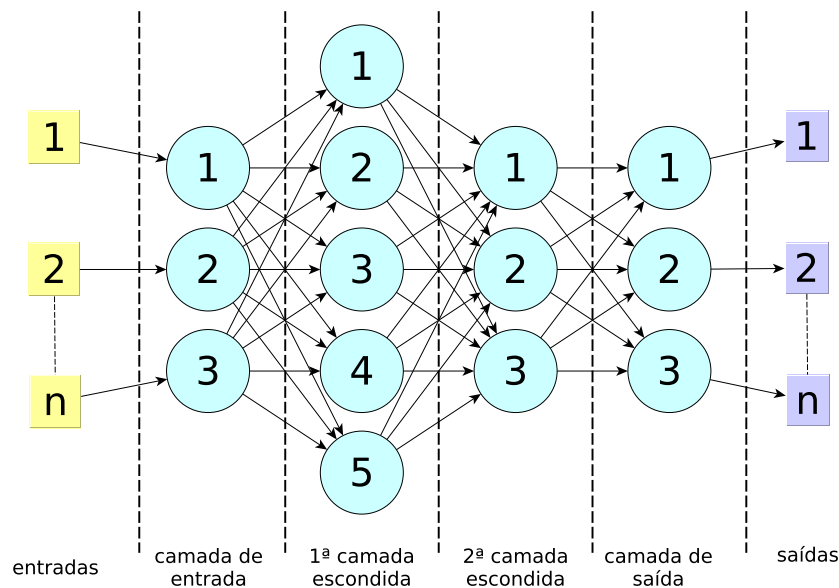


Figura 14: Rede neural artificial do tipo *feedforward*.

2.3.5 Aplicações

As redes neurais artificiais podem ser aplicadas com êxito em diversas áreas do conhecimento, tais como reconhecimento de faces, análise de imagens captadas por satélite, classificação de padrões de fala, previsão para ações do mercado financeiro, entre outras.

Além de tais aplicações, No caso da medicina, por exemplo, as RNAs são utilizadas na

análise de perfis genéticos de indivíduos para classificações e predições de câncer (KHAN et al., 2001).

Na biologia, as RNAs já foram utilizadas em aplicações visando identificar espécies de morcegos (PARSONS; JONES, 2000) a partir dos sinais emitidos por estes (chamado de biosonar). Tais sinais são emitidos durante os voos com a função de localização. Outra abordagem semelhante pode ser vista em (TIAN; SHANG, 2006), onde uma classificação de espécies ratos através dos sons produzidos pelos mesmos foi implementada.

Trabalhos registrados na área da química mostram a utilização de redes neurais artificiais para a obtenção de novos compostos poliméricos (ZHANG; FRIEDRICH, 2003), além de aplicação em sistemas de controle para tratamento de água (ZHANG; STANLEY, 1999). Neste último exemplo, o poder das RNAs em mapear processos não-lineares é explorado.

Devido à capacidade de tratamento de não-linearidades intrínsecas, as RNAs são também muito usadas na área de finanças e economia, devido ao comportamento não-linear destes sistemas (COAKLEY; BROWN, 2000).

As redes neurais artificiais já contribuíram para a elaboração de fórmulas, indicando, por exemplo, se o medicamento deverá ser fabricado através de um processo ou outro. Por exemplo, por intermédio de micro-emulsão ou dispersão sólida (MENDYK; JACHOWICZ, 2007).

Já no setor aeroespacial e automotivo, pode-se observar a utilização de redes neurais artificiais auxiliando no mapeamento de processos que envolvem estimativas de variáveis de controle e parâmetros de projeto. Em (CHO et al., 2006), estratégias de controle para veículos aéreos não-tripulados foram propostos, bem como esquemas de modelagem. Em uma aplicação formulada em (ORTEGA; SILVA, 2008), RNAs são responsáveis pela otimização de projetos de *brake-lights* automotivos construídos com diodos emissores de luz.

Na acústica, o projeto de salas de cinemas e ambientes sensíveis a ruídos externos necessita determinar a impedância acústica de cada ambiente. Tal aplicação emprega o uso de RNAs como pode ser visto em (TOO; CHEN; HWANG, 2007).

Como pôde ser visto através de vários exemplos de aplicações, a partir de medidas, amostras ou padrões, as RNAs alcançam ótimos resultados mapeando sistemas, inclusive não-lineares.

Tabela 2: *Lookup table* para função $\text{seno}(x\pi/8)$.

Índice	Resultado
0000	0
0001	0.38
0010	0.71
0011	0.92
0100	1
0101	0.92
0110	0.71
0111	0.38
1000	0
1001	-0.38
1010	-0.71
1011	-0.92
1100	-1
1101	-0.92
1110	-0.71
1111	-0.38

2.3.6 Estudo Sobre Implementações das Funções de Transferência

Os nodos de cada camada das redes neurais artificiais efetuam a operação descrita em sua função de transferência. Tais funções, no entanto, requerem um esforço computacional significativo e podem ter um tempo de processamento longo. Desta forma, para implementá-las corretamente em dispositivos com recursos limitados, pode-se implementá-las através de *lookup tables* (LUTs).

As *lookup tables* são tabelas indexadas com valores de referência para substituir a implementação de uma função. Elas são utilizadas em casos nos quais existem funções de difícil implementação. Em vez de um determinado valor ser utilizado como entrada de um bloco de processamento da função, ele serve como índice para esta tabela que fornece o resultado que está previamente memorizado. Assim, adquire-se velocidade em troca memória.

A tabela 2 mostra um exemplo de LUT para a função $\text{seno}(x\pi/8)$, com 4 bits para indexação. A entrada da equação é usada diretamente como índice da tabela, retornando o valor correspondente.

No caso de uma função real, com um número infinito de pontos, é necessário converter a simples busca por índice em uma busca por intervalos. O pseudocódigo 1 mostra como é a descrição funcional deste tipo de procedimento.

A maior vantagem da utilização das LUTs é a velocidade de processamento. As

```

1  função y = lut ( tabela , x )
2  Para i de 2 até o tamanho da tabela
3      Se x < tabela ( i , 1 )
4          y = tabela ( i - 1 , 2 ) ;
5          quebra ;
6      Fim Se
7  Fim para
8  y = lut ( i , 2 ) ;
9  Fim da função

```

Programa 1: Pseudocódigo da implementação de uma LUT.

aproximações por segmentos lineares (PWL, do inglês *piece-wise linear functions*), por exemplo, necessitam o uso de multiplicadores no projeto e, portanto, acabam sendo muito mais lentas que as tabelas.

Apesar da utilização de *lookup tables* tradicionais (com intervalos fixos) ser eficiente, em alguns casos a utilização de tabelas com intervalos variáveis pode diminuir consideravelmente o erro associado a esta discretização. Nestes casos, cria-se tabelas com intervalos proporcionais a taxa de variação da função, que resultam nas chamadas *range addressable lookup tables* (RALUT). Estas tabelas acompanham mais adequadamente a função contínua pois realizam mais transições nos momentos em que a função tem uma grande taxa de variação do resultado.

Neste trabalho, foi efetuado um estudo da aproximação das funções de transferência utilizadas para redes neurais quanto a utilização de LUTs e RALUTs. Ainda, a análise do erro para a variação do número de intervalos foi realizada para averiguar o valor adequado de intervalos para a aplicação e, para estes casos, se é mais interessante o uso da técnica de LUTs ou RALUTs.

A função logsig é descrita através da equação 2.14. A figura 15 mostra o comportamento da função para o intervalo de entrada $[-3,3]$, para o caso da função contínua, da implementação usando LUT e RALUT com 32 intervalos. Observa-se que a versão utilizando RALUT desempenha-se melhor em acompanhar a função contínua, uma vez que utiliza a dimensão dos intervalos de forma mais apropriada.

$$y(x) = \frac{1}{1 + e^{-x}} \quad (2.14)$$

A análise do erro entre as funções aproximadas e a contínua para a função logsig é mostrada na figura 16. Observa-se que o erro da RALUT é inferior a LUT, especialmente quando o número de intervalos é reduzido. A partir de 16 intervalos, a diferença de erro

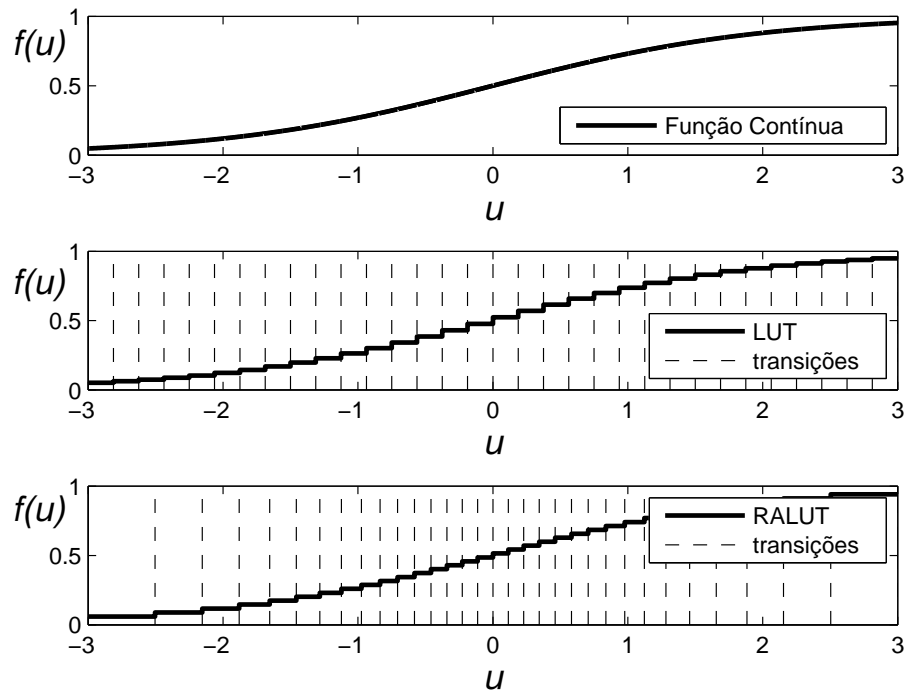


Figura 15: Função logsig contínua e utilizando 32 intervalos.

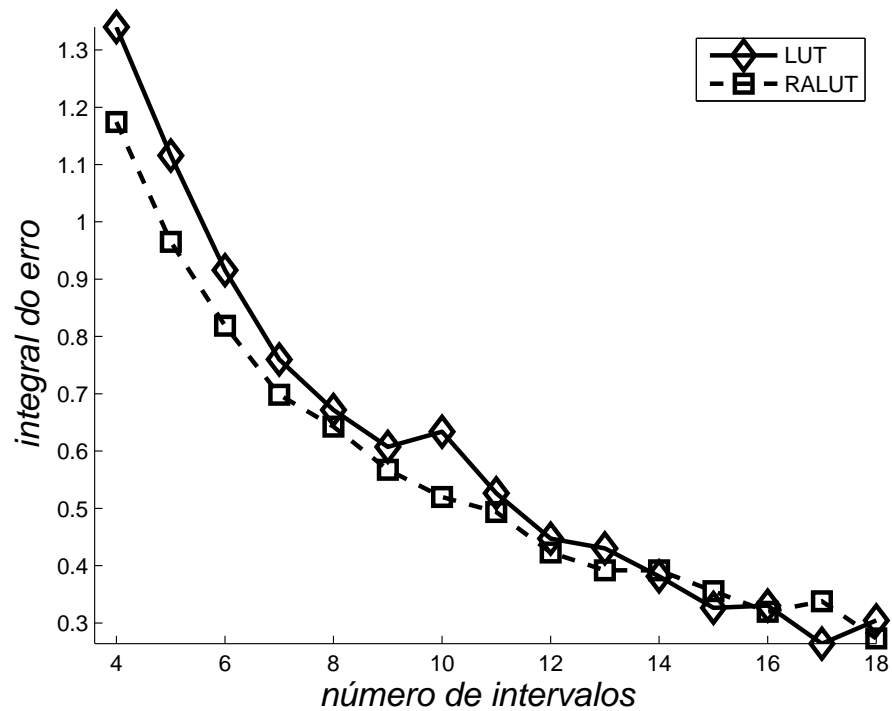


Figura 16: Erro entre a função aproximada e a contínua para a função logsig.

torna-se pequena e por vezes a LUT é melhor que a RALUT.

A mesma análise é realizada para as demais funções. A equação 2.15 descreve a função tangente sigmóide (similar a tangente hiperbólica). A figura 17 mostra a função contínua e a implementação com LUT e RALUT com 32 intervalos. O erro para diversos números

de intervalos pode ser visto na figura 18.

$$y(x) = \frac{2}{1 + e^{-2x}} - 1 \quad (2.15)$$

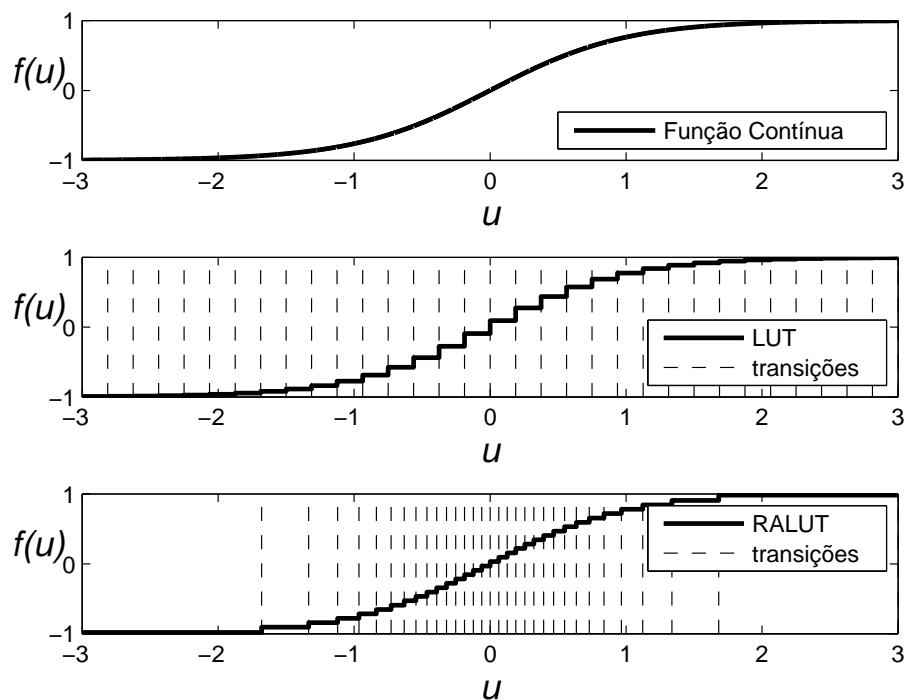


Figura 17: Função tansig contínua e utilizando 32 intervalos.

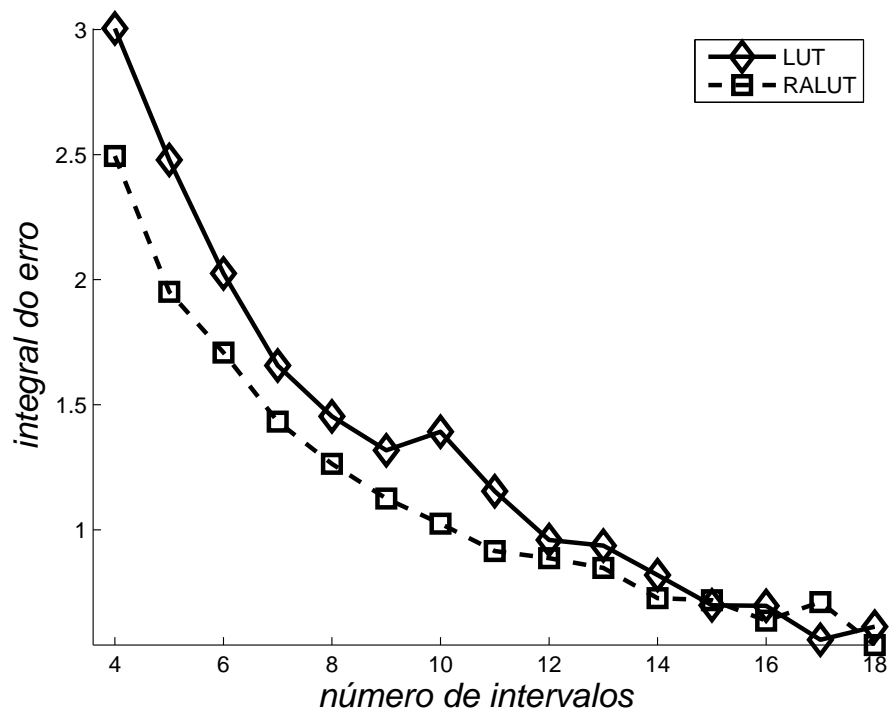


Figura 18: Erro entre a função aproximada e a contínua para a função tansig.

A equação 2.16 descreve a função *radial basis*. A figura 19 mostra a função contínua e a implementação com LUT e RALUT com 32 intervalos. O erro para diversos números de intervalos pode ser visto na figura 20.

$$y(x) = e^{(-2x)} \quad (2.16)$$

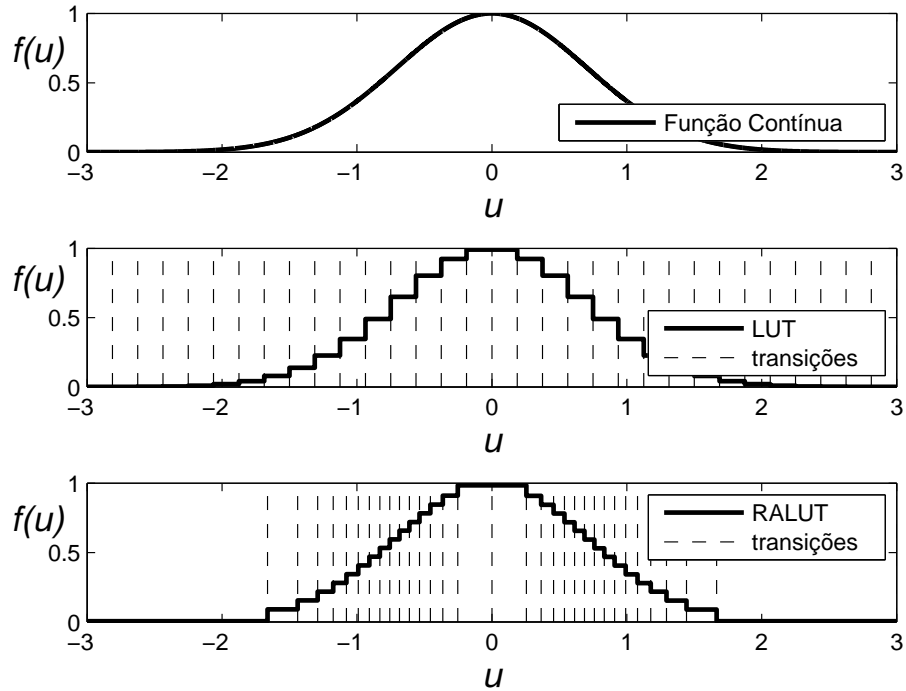


Figura 19: Função radbas contínua e utilizando 32 intervalos.

Por fim, a equação 2.17 descreve a função linear. Neste caso, não há distinção entre o uso de LUT e RALUT pois a taxa de variação é constante. A figura 21 mostra a função contínua e a implementação com LUT e RALUT com 32 intervalos.

$$y(x) = x \quad (2.17)$$

Pode-se observar que após aproximadamente 16 intervalos, os métodos tornam-se similares. Ainda assim, como pôde ser visto nas construções para 32 intervalos, o método RALUT oferece um comportamento mais próximo da função original.

2.3.6.1 Formas de Implementação em Hardware

As LUTs e RALUTs podem ser implementadas em hardware de duas formas: através do uso de memória ROM (do inglês, *Read Only Memory*) ou através de lógica combi-

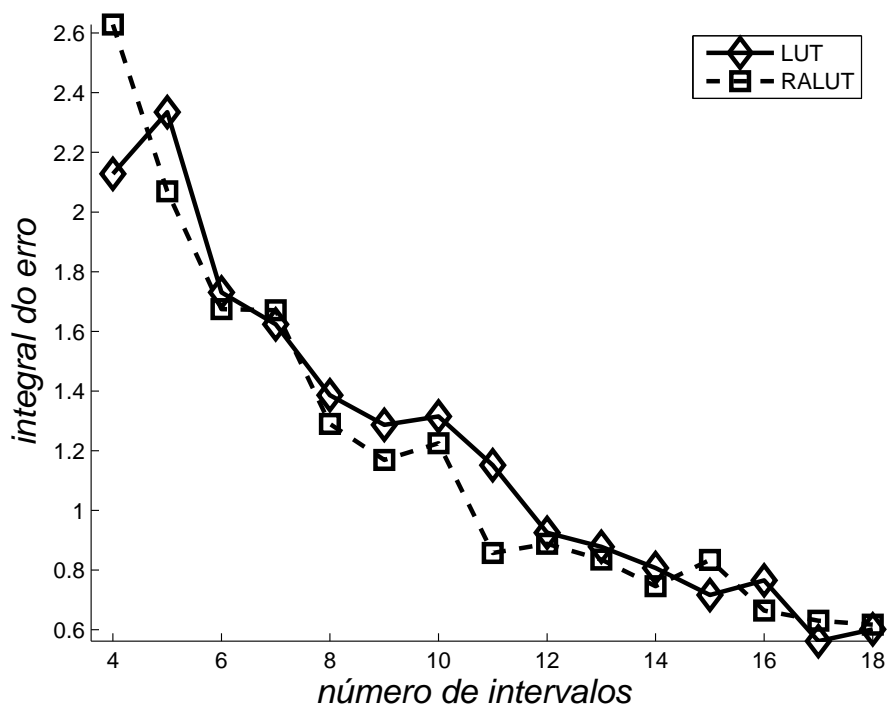


Figura 20: Erro entre a função aproximada e a contínua para a função radbas.

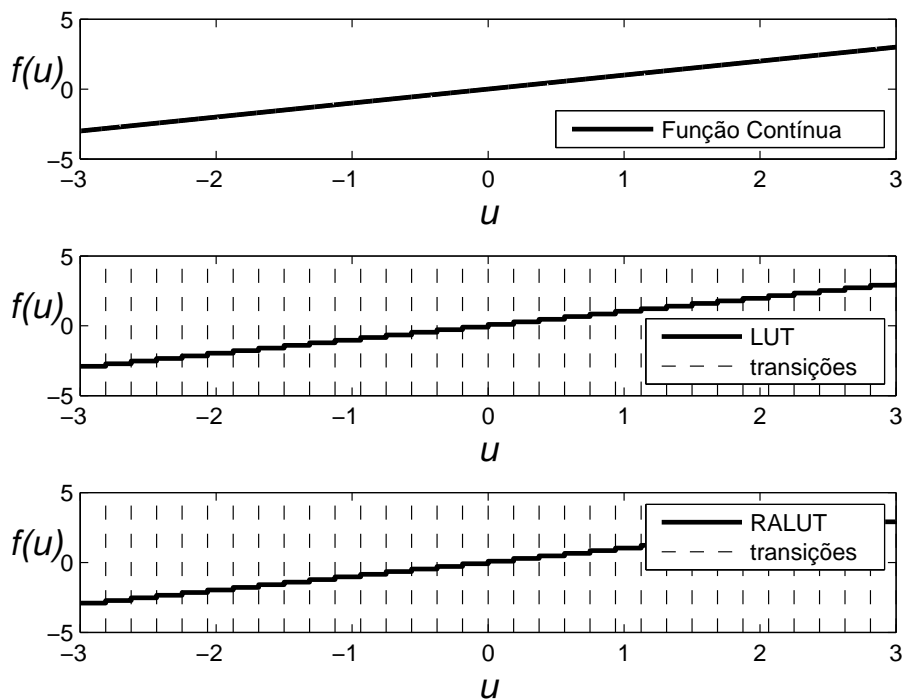


Figura 21: Função purelin contínua e utilizando 32 intervalos.

nacional. Este último método é especialmente interessante para aplicações em hardware no qual a mesma função será reutilizada diversas vezes, como é o caso das redes neurais. Um sintetizador lógico pode ser utilizado para reduzir o número de portas lógicas significativamente.

Em (MUSCEDERE et al., 2002), é proposta uma técnica de implementação de RALUTs. Esta implementação utiliza portas lógicas do tipo XOR (ou exclusivo) para detectar o intervalo no qual o dado de entrada está. Desta forma, a execução do processo ocorre de forma paralela, reduzindo o tempo de processamento. A figura 22 mostra a estrutura de uma RALUT. Observa-se o paralelismo do funcionamento, o que é proveitoso tanto na rapidez quanto na economia de energia, visto que o sistema executa rapidamente a operação e pode retornar ao estado de *stand-by*.

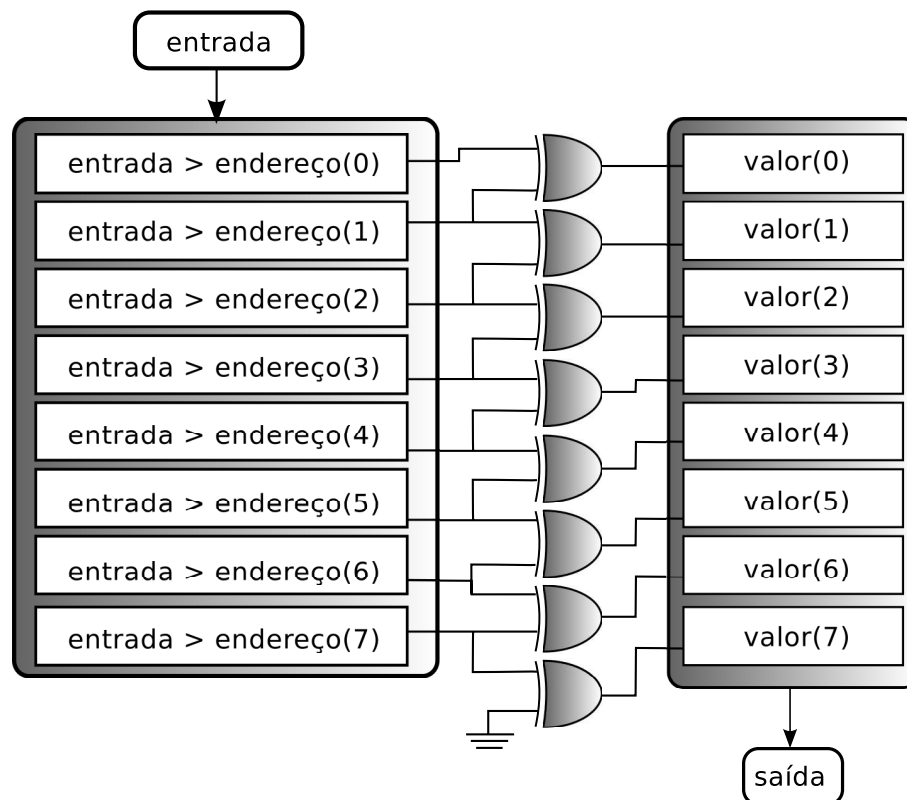


Figura 22: Estrutura da RALUT.

2.4 Localização Utilizando Redes Neurais Artificiais

2.4.1 Introdução

Apesar da existência de poucos trabalhos utilizando redes neurais artificiais diretamente na tarefa de localização de nodos em RSSF, os resultados publicados são promissores. Em todos os casos, os dados de entrada utilizados para treinar as RNAs foram provenientes de medições de grandezas, tais como o RSSI e as distâncias entre os nodos, sendo as saídas as próprias coordenadas estimadas.

Outro ponto em comum nestes trabalhos é a falta de uma metodologia para escolha

dos principais parâmetros das redes neurais artificiais empregadas. Todos os resultados foram obtidos com base na tentativa e erro, com tempo e eficiência variando de acordo com o projetista das RNAs. Segundo alguns autores, as simulações foram realizadas dentro do *MatLab*, porém não foram dados muitos detalhes a respeito de diversos parâmetros de simulação e validação.

2.4.2 Trabalhos Encontrados na Literatura

Em (SHAREEF; ZHU; MUSAVI, 2007), um experimento prático foi utilizado para avaliar o desempenho de três redes neurais artificiais, bem como para efetuar uma comparação com outros sistemas de localização baseados em filtros de Kalman (SHAREEF et al., 2007).

Os nodos utilizados foram os sensores *Cricket* do MIT (PRIYANTHA; CHAKRABORTY; BALAKRISHNAN, 2000), sendo medidas de distância utilizadas como entrada para as RNAs. Uma área de 3 x 3 metros foi dividida em quadrados de 0,3 metro, sendo que as medições de distância para treinamento das RNAs foram tomadas nas intersecções destes quadrados. Tais medidas foram tomadas em relação aos 3 nodos-âncora dispostos em 3 vértices da área de teste.

Devido às variações nas medições de distância que foram obtidas em cada ponto, um total de 1521 conjunto de medidas foi gerado para um total de 121 pontos. Estes dados foram usados no treinamento das RNAs sob teste, uma vez que pelas variações presentes nas medições de distância era possível capturar também o ruído no ambiente. Os dados para teste foram obtidos através da movimentação de um nodo por um caminho sinuoso dentro da área de 3 x 3 metros.

As RNAs testadas foram:

- MLP: Do inglês, *Multi-Layer Perceptron*;
- RBF: Do inglês, *Radial Basis Funtion*;
- RRBF: Do inglês, *Reduced Radial Basis Function*;
- RNN: Do inglês, *Recurrent Neural Network*.

Segundo os autores, a RNA do tipo RBF ainda não havia sido testada para localização. Esta RNA obteve os melhores resultados em termos de precisão das estimativas, mas utilizava muitos recursos computacionais por possuir 1521 neurônios em sua camada de

entrada. Já rede do tipo RRBf foi constituída por 121 neurônios em sua camada de entrada, sendo seu desempenho o pior entre todos os sistemas de localização testados.

A rede MLP utilizada foi composta por 9 neurônios na camada de entrada e dois na camada de saída, sendo que sua precisão foi ligeiramente menor do que a obtida pela rede do tipo RBF. Porém, o custo computacional da rede MLP é muito mais atrativo tendo em vista que esta possui apenas 9 neurônios. A rede do tipo RNN teve sua composição idêntica à MLP, porém as saídas de sua primeira camada foram utilizadas como entradas para cada um dos neurônios anteriores, criando uma realimentação.

Os resultados deste trabalho, medidos em função do RMSE (do inglês, *Root Mean Square Error*, ou REQM em português), indicam a rede MLP como sendo a melhor opção para localização de nodos, uma vez que esta possui uma relação custo-benefício superior aos demais tipos testados. É importante ressaltar que, para todos os tipos de RNAs testados, não foi descrita metodologia alguma para a seleção das melhores estruturas de RNAs a serem aplicadas.

Em (RAHMAN; PARK; KIM, 2009) foi utilizada uma MLP de duas camadas escondidas, sendo a primeira camada composta por 10 neurônios artificiais e a segunda por 4. Não ficou claro se a primeira camada escondida foi considerada como sendo a camada de entrada ou se, além das camadas escondidas, havia uma de entrada.

Nesta abordagem foram utilizados os dados de RSSI como entrada das RNAs, sendo utilizados 8 nodos-âncora dispostos uniformemente nas bordas de uma área quadrada que teve seu tamanho alterado de $20m^2$ até $200m^2$, em passos de $20m^2$. O autor informa que os testes foram feitos através do software *MatLab*, porém não são dadas muitas informações a respeito de como estas simulações foram feitas. Na área testada todos os nodos podiam se comunicar com todos, isto é, todos os nodos desconhecidos podiam receber pacotes de todos os nodos-âncora.

Para coletar dados visando o treinamento das RNAs, 81 nodos cujas coordenadas eram conhecidas foram dispostos na área de teste e armazenaram dados de RSSI em relação aos 8 nodos-âncora. Foi informado que o treinamento da rede era realizado periodicamente de modo a minimizar os erros nas estimativas, porém não fica claro como este procedimento pode ser feito pelos nodos da RSSF, se *online* ou *offline*.

Neste trabalho, (RAHMAN; PARK; KIM, 2009) afirma que a precisão das estimativas de localização (também avaliadas a partir do RMSE) depende da densidade da grade de treino utilizada e também que a precisão é aumentada à medida que se amplia o número de nodos-âncora. A primeira afirmação é corroborada por esta dissertação em sua seção

de simulações, porém a segunda, não.

Na abordagem apresentada por (TIAN; SHI, 2007) os dados para testes foram obtidos por software de simulação, porém não é dada informação alguma sobre este aspecto, e nem que tipo de dado foi usado (RSSI ou distância). É explicitado também que os testes foram desenvolvidos dentro do software *MatLab* e que a *toolbox* de redes neurais artificiais foi usada.

O ambiente principal dos testes era composto por 20 nodos-âncora dispostos aleatoriamente em uma área quadrada de 100 unidades. O alcance dos rádios foi ajustado para 40 unidades. A RNA usada foi uma MLP com duas camadas, sendo o número de neurônios na camada de entrada igual a 20 vezes a quantidade de nodos-âncora adotados. Este fato leva a uma RNA de centenas de neurônios artificiais, o que muitas vezes inviável ou energeticamente pouco eficiente quando se tem como alvo implementações em sistemas embarcados.

A quantidade de nodos-âncora foi variada de 10 até 24, sendo apresentado o dado de que acima de 22 nodos-âncora, a precisão das estimativas não é muito influenciada. São feitas comparações com um método de localização baseado em MLE (do inglês, *Maximum Likelihood Estimation*), sendo que o método utilizando RNAs se mostrou mais robusto em todas as situações.

2.5 Metaheurísticas

2.5.1 Introdução

Metaheurísticas são algoritmos para solução de problemas para os quais tem-se pouco conhecimento prévio a respeito de seu funcionamento. Assim, estes algoritmos utilizam estruturas que implementam diversas iterações a fim de aproximar-se da resposta correta gradualmente. Para tal, é necessário a existência uma forma de avaliação das soluções propostas pelo algoritmo, que é realizada através da função custo. Esta função recebe da metaheurística a descrição da solução (valores das variáveis) e retorna para o algoritmo um valor que informa quão próxima do ideal a solução está.

Diferentemente de métodos matemáticos criados especificamente para solução de um problema, as metaheurísticas são genéricas e podem ser aplicadas a uma vasta gama de aplicações. No entanto, a desvantagem decorrente disto é que não garantem encontrar o mínimo global (melhor solução possível para um problema), apesar de oferecerem um bom

custo/benefício no que diz respeito a soluções suficientemente boas em tempo aceitável.

Devido a sua característica genérica, as metaheurísticas são algumas vezes chamadas de otimização do tipo caixa-preta (do inglês, *black-box optimization*). Este nome dá ênfase a uma característica importante das metaheurísticas: elas não mostram a estrutura interna do problema, apenas conseguem encontrar a solução correta através do uso da função custo. Usualmente, nenhuma inferência a respeito das características do problema é necessária durante a implementação da metaheurística. Através de pequenas variações na solução inicial ou através de variações em uma população de soluções iniciais, o algoritmo guia as soluções à exploração do espaço de projeto, mantendo as que se desempenham melhor na função custo. Desta forma, grande parte do trabalho envolvido na aplicação destes algoritmos é a construção da função custo.

2.5.2 Problema de Otimização

O problema que a metaheurística precisa resolver pode ser descrito na forma de um problema de otimização. Em (RAO, 1996), a formulação matemática do problema de otimização é descrita pela equação 2.18:

$$\text{Encontrar } \vec{x} = \{x_1, x_2, \dots, x_n\} \quad \text{que minimiza } f(\vec{x}) \quad (2.18)$$

sujeita a:

$$g_j(\vec{x}) \leq 0, \quad j = 1, 2, \dots, m \quad (2.19)$$

$$l_j(\vec{x}) = 0, \quad j = 1, 2, \dots, p \quad (2.20)$$

onde $\vec{x} = [x_1, x_2, \dots, x_n]^T$ é um vetor contendo as variáveis de projeto, onde n é o número de variáveis, $f(\vec{x})$ é a função custo, $g_j : \mathbb{R}^n \rightarrow \mathbb{R}$ representa as restrições de desigualdade, $l_j : \mathbb{R}^n \rightarrow \mathbb{R}$ representa as restrições de igualdade, m e p são o número de restrições de desigualdade e igualdade respectivamente.

As restrições limitam as possibilidades de escolha de valores para as variáveis de entrada. Decorrente disto, surge o conceito de espaço de projeto viável \mathbf{X} , que é definido como o conjunto:

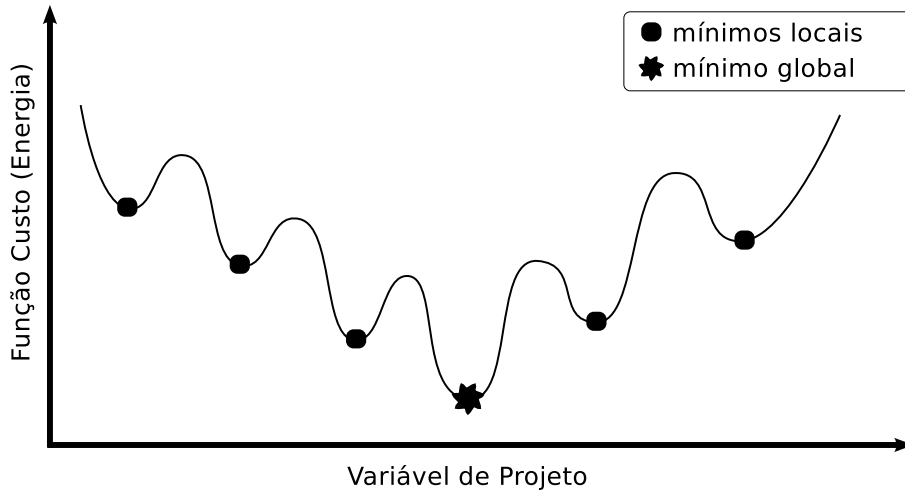


Figura 23: Mínimos locais e mínimo global.

$$\{\mathbf{X} | g_j(x) \leq 0, j = 1, 2, \dots, m; \quad e \quad l_j(x) = 0, j = 1, 2, \dots, p\}$$

O espaço de critério viável \mathbf{Z} é definido como o conjunto:

$$\{\mathbf{F}(x) | x \in \mathbf{X}\}$$

A figura 23 mostra um exemplo de comportamento da função custo (energia) para um problema com apenas uma variável. Como pode ser observado, movimentando a variável continuamente, haverá momentos no qual uma solução é melhor que todas as vizinhas, apesar de não ser a melhor de todo o espaço de projeto. As regiões nas quais isto ocorre são chamadas mínimos locais. Para um algoritmo que tenta gradualmente encontrar a solução ideal, estas regiões oferecem grande dificuldade pois o algoritmo tende a julgar que encontrou a melhor solução existente, quando na verdade o mínimo global ainda não foi encontrado.

Para evitar mínimos locais, cada metaheurística tem uma estratégia diferente. Em geral, o uso de variáveis randômicas nestes algoritmos permite que eles movimentem as soluções para locais contrários ao gradiente, diferentemente do que fazem as técnicas “gulosas” e também os métodos clássicos de otimização. Esta característica torna as metaheurísticas atraentes para o uso em aplicações com grande número de mínimos locais. Ainda, são mais robustas para aplicações nas quais não se sabe como a função custo trabalhará.

Apesar da grande flexibilidade destes algoritmos, existe um tipo de função custo as quais eles tem dificuldade em otimizar. Estas são as funções nas quais o gradiente de

diminuição de energia oferece pistas erradas para o otimizador. O mínimo global encontra-se em uma região afastada do ponto para o qual o gradiente da função custo aponta, ou seja, desfavorece a melhoria gradual da solução. A figura 24 mostra o caso de uma função custo deceptiva.

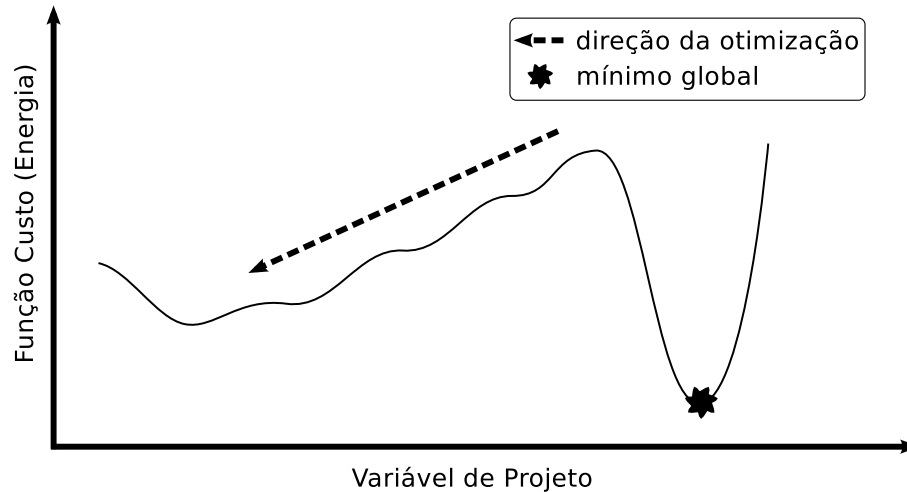


Figura 24: Exemplo de função custo deceptiva.

2.5.3 Problema com Múltiplos Objetivos

Alguns problemas podem ter mais de um objetivo a serem otimizados. Estes problemas são chamados multi-objetivo e necessitam algoritmos especiais para sua solução. A particularidade deste tipo de problema é devido a não ser sempre possível considerar uma solução melhor que a outra. Como existe mais de um objetivo, é possível que um deles seja melhor atendido por uma solução, e outro por outra. Neste casos, ambas as soluções devem ser mantidas pelo algoritmo. A equação 2.21 descreve matematicamente o problema multi-objetivo.

$$\text{Min } \mathbf{F}(\vec{x}) = \{f_1(\vec{x}), f_2(\vec{x}), \dots, f_k(\vec{x})\} \quad (2.21)$$

onde $\vec{x} = [x_1, x_2, \dots, x_n]^T$ é o vetor das variáveis de projeto, f_i é a função objetivo, e $i = 1, \dots, k$ são as funções objetivo para cada especificação.

Nestes problemas, define-se fronteira de pareto como sendo o a fronteira formada pelo conjunto de soluções que não podem ser consideradas dominadas por nenhuma outra. Uma solução dominada é alguma que seja pior que outra em todos os objetivos. Em um processo de otimização, algumas soluções serão consideradas melhor que todas as outras em pelo menos um objetivo. Isto a coloca no conjunto de pareto, que é composto pelas

soluções que descrevem as trocas de projeto, ou seja, as decisões que precisam ser feitas em detrimento de um objetivo ou outro visto que não pode-se contemplar todos em apenas uma solução.

Dados dois vetores $\vec{x}, \vec{y} \in \mathbb{R}^k$ diz-se que $\vec{x} \leq \vec{y}$ se $x_i \leq y_i$ para $i = 1, \dots, k$, e que \vec{x} domina \vec{y} se $\vec{x} \leq \vec{y}$ e $\vec{x} \neq \vec{y}$. Ainda, diz-se que um vetor de variáveis $\vec{x}^* \in F \subset \mathbb{R}^n$ (F é a região viável) é um Pareto-ótimo se ele não for dominado com respeito a F .

O vetor $\vec{x} = [x_1, x_2, \dots, x_n]^T$ das variáveis utilizadas durante a otimização depende da aplicação. No caso da modificação da estrutura das redes neurais, estas variáveis são as seguintes: número de camadas, número de nós em cada camada e função de transferência de cada camada. Uma alteração vai modificar uma das variáveis anteriores e incrementar ou decrementar seu valor. Estes valores serão utilizados para efetuar a criação da rede de sensores.

Nesta seção, as metaheurísticas utilizadas neste trabalho serão descritas. As subseção 2.5.4 descreve os princípios de funcionamento dos algoritmos genéticos e a seção 2.5.5 fala sobre o algoritmo de *Simulated Annealing*.

2.5.4 Algoritmos Genéticos

Os algoritmos genéticos são uma técnica de busca que tem seus passos baseados no processo de evolução das espécies. Dessa forma, o conceito básico consiste em gerar uma população de soluções para determinado problema e avaliar quais destas são as melhores e, conseqüentemente, devem ter mais chance de serem reproduzidas para gerar a nova geração da população. O processo é iterado diversas vezes de forma a gradualmente aperfeiçoar a população, buscando assim o ótimo global.

Através de (HOLLAND, 1975), os algoritmos genéticos foram popularizados. Atualmente, eles são utilizados em uma vasta gama de aplicações devido a sua robustez e flexibilidade. Uma vez que trata-se de um algoritmo populacional, no qual um grande número de soluções são utilizadas a cada geração, mesmo que diversas destas esteja inicialmente com desempenho apenas razoáveis, existe uma maior probabilidade que alguma delas possua um resultado levemente satisfatório em relação a um algoritmo não populacional. Estas soluções vão evoluindo com o passar das gerações, até que exista a convergência para um mínimo.

As etapas básicas presentes na técnica de algoritmos genéticos são:

- **Avaliação:** processo no qual cada indivíduo da população é avaliado quanto a

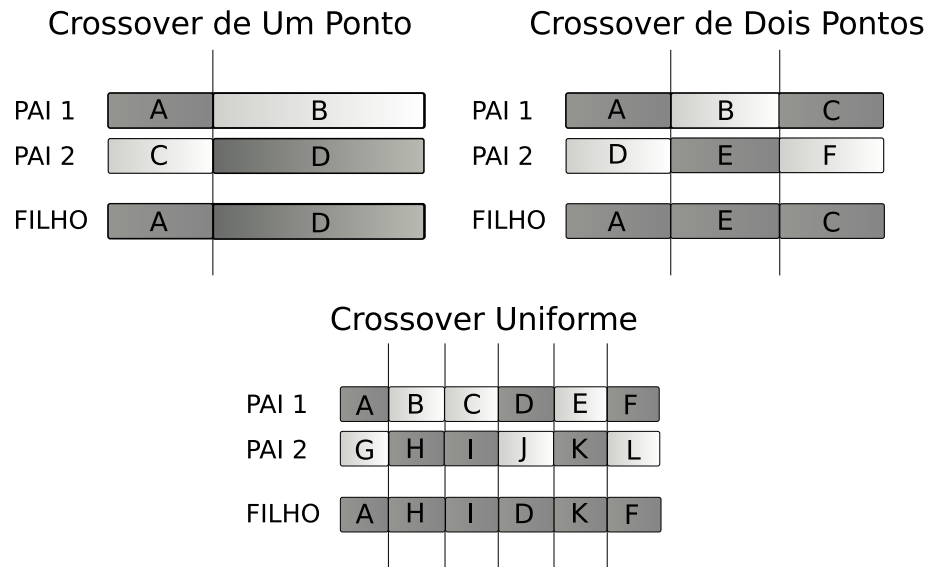


Figura 25: Exemplos de tipos de *Crossovers*.

proximidade do objetivo desejado. Ou seja, é a aplicação da função custo em todas as soluções existentes na população;

- **Seleção (Reprodução):** consiste na escolha de quais *strings* de dados (em analogia ao DNA) serão utilizadas para que sobre eles ocorra crossover com outras *strings*. Cada uma destas *strings* constitui uma solução, ou seja, um indivíduo. Existem diversos métodos de seleção de cromossomos para *crossover*. Algumas delas são: seleção do tipo roleta, seleção do tipo Boltzmann, seleção do tipo Torneio, seleção por *Ranking*, entre outros;
- **Crossover:** consiste na mistura e combinação de partes de diferentes pais (soluções pertencentes a geração anterior) para a formação de filhos (soluções pertencentes a nova geração). Existem diversos tipos de *crossovers*, como *crossover* uniforme, *crossover* de um ponto, *crossover* de dois pontos, entre outros. A figura 25 ilustra alguns destes tipos;
- **Mutação:** apenas o processo de crossover não é capaz de fazer uma exploração adequada do espaço de projeto, pois pode resultar em convergência prematura. Desta forma, a mutação é uma componente aleatória adicionada ao algoritmo para que exista a possibilidade de que novas soluções sejam geradas.

Os algoritmos genéticos tem a capacidade de realização de exploração genuína, através da operação de mutação, e movimentos de exploração direcionada, através do operador de *crossover*. Isto atribui a ele uma vantagem a diversos métodos, uma vez que conta com mais de um tipo de movimento.

Tabela 3: Parâmetros Principais de Configuração do Algoritmo Genético

Parâmetro	Descrição
Número de Soluções de Elite	Número de soluções que são mantidas de uma geração para outra sem serem alteradas. Estas soluções são as melhores de sua geração e, portanto, são mantidas para preservar respostas promissoras.
Número Máximo de Gerações	Número máximo de gerações até que ocorra o fim do algoritmo.
Fração de <i>Crossovers</i>	Fração da próxima geração que é criada através da função de <i>crossover</i>
Tamanho da População	Número de soluções na população.

No pseudocódigo 2, é ilustrado de forma simplificada os passos principais de um algoritmo genético. O algoritmo finaliza quando algum dos critérios de parada, que podem ser variados ocorre. No pseudocódigo mencionado utilizou-se apenas o número máximo de gerações como critério de parada, mas questões como a convergência das soluções para um único ponto ou tempo máximo de otimização também podem ser utilizados para cessar o processo. Ao fim, o algoritmo deve retornar a melhor solução da última geração, ou, possivelmente, todas as soluções presentes na última geração.

```

1 Para g=1 até G, onde G é o número máximo de gerações
2   Avaliar indivíduos da população
3   Criar vetor de probabilidade para seleção
4   Selecionar pais
5   Fazer crossover
6   Fazer mutação
7 Fim para

```

Programa 2: Pseudocódigo do Algoritmo Genético.

Em uma aplicação elaborada de algoritmos genéticos, uma série de opções pode ser considerada para adequar o algoritmo da melhor forma possível ao problema em questão. A tabela 3 mostra algumas destas opções, juntamente com suas descrições.

2.5.5 *Simulated Annealing*

O algoritmo *Simulated Annealing* (SA) (recozimento simulado, em português) é uma técnica de busca probabilística, não populacional (diferentemente dos algoritmos genéticos) e que mantém apenas uma solução na memória a cada iteração. A proposta do

algoritmo foi realizada por Kirkpatrick e desenvolvido por diversos cientistas nos anos 1980 (KIRKPATRICK; GELATT; VECCHI, 1983). O algoritmo é baseado no processo físico de recozimento, no qual eleva-se a temperatura de um material para depois gradualmente resfriá-lo. Durante o resfriamento, a estrutura cristalina torna-se progressivamente mais regular. O funcionamento do algoritmo é similar ao do *Hill-Climbing*, ou seja, existe uma solução que sofre alguma modificação e avalia-se se a solução gerada é melhor que a anterior. Caso seja, a nova é adotada e a anterior descartada. Caso contrário, ignora-se a solução gerada e se permanece com a solução anterior. No entanto, o que difere o *Simulated Annealing* do *Hill-Climbing* é a existência de uma probabilidade de aceitação de soluções mesmo que estas sejam piores que a anterior. No algoritmo, a temperatura é uma variável que afeta a probabilidade de aceitação de uma solução pior que a presente ser aceita, fazendo então a analogia com o caso físico. A vantagem de um algoritmo ter a capacidade de aceitar soluções piores em alguns momentos é que traduz-se na saída de regiões de mínimos locais. Caso uma solução inicial esteja próxima de um mínimo local, soluções piores (que caminham em sentido contrário ao mínimo local) podem ser escolhidas e levar a otimização para uma área próxima a um mínimo global. Variações do algoritmo inicial foram vastamente exploradas na literatura (SUMAN, 2004). No entanto, pode-se considerar como presente em todas estas as seguintes funções: função de geração, função de cronograma e função de aceitação.

- **Função de Geração:** responsável pela criação de uma nova solução a partir da anterior. Em problemas do tipo combinacional, esta tem a dimensão da menor alteração possível na solução anterior, mudando apenas qual variável será escolhida para sofrer alteração;
- **Função de Cronograma:** responsável pela diminuição e controle da temperatura;
- **Função de Aceitação:** quando existe uma solução com função custo pior que a atual, determina a probabilidade da mesma ser aceita.

No pseudocódigo 3, é ilustrado de forma simplificada os passos principais do algoritmo de *simulated annealing*.

A nova solução é selecionada baseada na decisão descrita na equação (2.22):

$$S_{k+1} = \begin{cases} N, & \text{se } \text{custo}(N) < \text{custo}(S) \quad \text{ou} \quad \tau \leq P(t_k, N, S_k) \\ S_k, & \text{caso contrário} \end{cases} \quad (2.22)$$

```

1  S ← solução inicial
2  melhor ← S
3  Enquanto (t ≥ tmin)
4      R ← Modificar(Copia(S))
5      Se (Custo(R) < Custo(S)) | (rand < P(t,R,S))
6          S ← R
7      Fim
8      Decrementar t
9      Se Custo(S) < Custo(melhor)
10         melhor ← S
11     Fim
12 Fim

```

Programa 3: Pseudocódigo do *Simulated Annealing*.

onde k é o índice da solução, N é a solução candidata, S_{k+1} é a próxima solução, τ é uma variável randômica uniformemente distribuída entre $[0, 1]$ e $P(t_k, R, S_k)$ é a probabilidade de aceitação de uma solução pior dada uma temperatura t^k e uma diferença de custo.

A probabilidade de aceitação de uma solução pior, também chamada de critério de aceitação, é a equação que, através da temperatura e diferença de custo, define a probabilidade de uma resposta pior ser aceita. O critério de Metropolis pode ser visto na equação 2.23:

$$P(t_k, N, S_k) = e^{-\frac{\text{Custo}(S_k) - \text{Custo}(N)}{t_k}} \quad (2.23)$$

A figura 26 mostra como o *Simulated Annealing* poderia comportar-se na função custo descrita como exemplo no começo desta seção. Através da aceitação de soluções piores que a atual, ocorre a fuga dos mínimos locais e aumentam as chances de encontrar o mínimo global.

O parâmetro de controle é reduzido utilizando um cronograma proporcional de temperatura:

$$t_{k+1} = \alpha t_k \quad (2.24)$$

onde α é um valor próximo e inferior a unidade. A função de geração é determinada pela menor alteração possível nas variáveis, que para a aplicação em questão trata-se da modificação em um degrau (incremento ou decremento de uma unidade) de uma das variáveis descritas anteriormente.

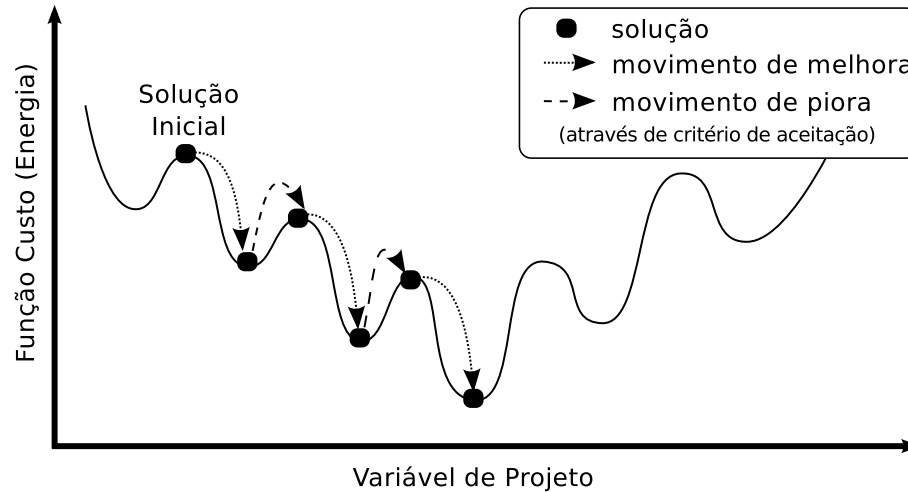


Figura 26: Mínimos locais e mínimo global no *Simulated Annealing*.

2.5.6 Aplicações

2.5.6.1 Roteamento de Veículos

Este é um dos mais estudados problemas de otimização combinatória. Uma determinada frota de veículos deve atender um conjunto de consumidores, sendo que tais veículos partem de um ou mais pontos denominados depósitos. A restrição é que cada veículo possui uma determinada capacidade, sendo que esta não deve ser excedida pelo somatório de todas as demandas dos consumidores atendidos por este veículo.

Este problema, apesar de aparentar certa simplicidade, possui elevada complexidade computacional, fato este que o faz interessante para a aplicação de Metaheurísticas.

2.5.6.2 Problema da Mochila

Este problema possui este nome por ele descrever uma situação onde é necessário o preenchimento de uma mochila com objetos de diferentes pesos e valores, sendo que o objetivo é que se preencha a mochila com o maior valor possível, mas sem exceder o peso máximo. Este modelo de problema pode ser aplicado em outras situações, como por exemplo empacotando e corte (de tecidos, papéis, etc.), carregamento de veículos, investimento de capital, entre outros.

2.5.6.3 Problema do Caixeiro Viajante

Apesar de parecer de fácil resolução, este é um problema muito pesquisado por cientistas de diversas áreas do conhecimento. Ele consiste em procurar um circuito com

menor distância possível, começando em uma cidade qualquer (entre várias) e visitando cada uma delas apenas uma vez, e regressando ao local inicial. O esforço computacional necessário para resolver este problema aumenta exponencialmente com seu tamanho.

2.5.6.4 Problema da Satisfatibilidade Booleana

Este problema consiste em, dada uma expressão booleana escrita apenas com operadores *AND*, *OR*, *NOT*, variáveis e parênteses, procurar uma combinação de valores binários para as variáveis tal que a expressão seja verdadeira. O problema da satisfatibilidade booleana é de extrema importância em diversos contextos, incluindo, por exemplo, teoria da computação e inteligência artificial.

2.6 Software *MatLab*

Todo o processo de simulação para coleta de dados, criação de redes neurais artificiais, utilização de metaheurísticas e processamento do algoritmo principal, foi feito dentro do software *MatLab* (*MATrix LABoratory*), que é um software voltado principalmente para cálculo numérico, processamento de sinais e construção de gráficos, possuindo pacotes de aplicações para diversas áreas do conhecimento, em especial engenharias e informática.

Para as tarefas que envolviam especificamente criação e manipulação de redes neurais artificiais e o uso de metaheurística populacional (algoritmo genético), foram utilizadas *toolboxes* (*Neural Network Toolbox* e *Global Optimization Toolbox*) nativas do próprio *MatLab*. Já a simulação de redes ficou a cargo do simulador probabilístico de redes sem fio, *Prowler* (*Probabilistic Wireless Network Simulator*), também enquadrado como uma *toolbox* e desenvolvido pela universidade norte-americana Vanderbilt.

2.6.1 Ambiente de simulação - *Prowler*

O *Prowler* (SIMON et al., 2003) é um simulador baseado em eventos, e pode ser configurado para operar tanto no modo determinístico (para produzir resultados replicáveis durante o teste de uma aplicação) quanto no modo probabilístico (para simular a natureza não-determinística do canal de comunicação e protocolo de comunicação de baixo nível existente entre os nodos). Neste trabalho foi empregado o modo probabilístico do simulador. Ele pode utilizar um número arbitrário de nodos, dispostos em topologia também arbitrárias (incluindo topologias dinâmicas), além de ter sido projetado para ser facilmente integrado em algoritmos de otimização. Desta forma, o *Prowler* constitui uma excelente

ferramenta para projetar e testar aplicações, agregando também ótimas capacidades de visualização da simulação.

O modelo de canal de propagação de radiofrequência determina a força de um sinal transmitido de um ponto específico do espaço para todos os nodos presentes na rede. Baseando-se nesta informação a respeito do sinal transmitido, as condições para a recepção deste por parte dos demais nodos podem ser então avaliadas e colisões podem ser detectadas.

A força do sinal de um transmissor para um receptor é determinada por uma função de propagação determinística (modelada pelo desvanecimento do sinal em função da distância), e pelos distúrbios aleatórios (modelando a natureza variante no tempo da força do sinal e outros erros diversos de transmissão).

A parte determinística da função de propagação pode ser uma função fornecida pelo usuário, mas um modelo de distância versus força do sinal razoável e utilizado com frequência é dado pela equação (2.25), onde $P_{rec,ideal}$ é a potência ideal de recepção do sinal, $P_{transmit}$ é a potência de transmissão, d é a distância entre o transmissor e o receptor, e γ é o parâmetro de atenuação com valores típicos entre $2 \leq \gamma \leq 4$.

$$P_{rec,ideal}(d) = P_{transmit} \frac{1}{1 + d^\gamma} \quad (2.25)$$

Sinais reais, contudo, se comportam de uma maneira muito diferente. A força do sinal pode variar de forma brusca à medida que a distância muda. O sinal pode variar mesmo com a distância entre transmissor e receptor se mantendo constante, devido a sua natureza variante no tempo. O efeito de desvanecimento é modelado por distúrbios randômicos no simulador. A força do sinal transmitido do nodo j para o nodo i é calculada com base na equação 2.25 modulando esta com as funções randômicas vistas na equação 2.26.

$$P_{rec}(i, j) = P_{rec,ideal}(d_{i,j}) \cdot [1 + \alpha(d_{i,j})] \cdot [1 + \beta(t)] \quad (2.26)$$

A variável randômica α depende exclusivamente da distância, logo, o simulador apenas a recalcula caso a posição do transmissor ou receptor mude. Já a variável β é recalculada a cada nova transmissão, visto que esta é dependente do tempo. No simulador, as variáveis randômicas α e β têm distribuições normais $N(0, \sigma_\alpha)$ e $N(0, \sigma_\beta)$ respectivamente, sendo os parâmetros (desvio padrão) σ_α e σ_β ajustáveis. Ainda, um parâmetro adicional p_{error} é responsável pela modelagem da probabilidade de erros de transmissão causados por efeitos diversos, tais como distúrbios externos e hardware não confiável.

Existem dois modelos para recepção de sinais e detecção de colisões atualmente usados no simulador. No primeiro, o sinal é recebido se a força do sinal é maior do que um parâmetro limite. O canal é percebido como desocupado se não houver sinal algum que possa ser recebido. Ocorre uma colisão se dois ou mais terminais transmitem ao mesmo tempo e um mesmo receptor está na área de alcance destes. No segundo modelo, cada receptor possui um parâmetro de variância de ruído σ_n^2 . A relação sinal/interferência-ruído (do inglês *signal to interference and noise ratio* - *SINR*) para um receptor i e um transmissor j é definida pela equação 2.27.

$$SINR = \frac{P_{rec}(i, j)}{\sigma_n^2 + \sum_{k \neq j} P_{rec}(i, k)} \quad (2.27)$$

A força total do sinal no nodo i é definida pela equação 2.28.

$$P_{tot}(i) = \sum_k P_{rec}(i, k) \quad (2.28)$$

O sinal é recebido se o *SINR* no receptor for maior do que o limite de recepção durante toda o tempo de transmissão. O canal é percebido como desocupado se a força total do sinal for menor do que um limite para canal liberado, o que depende da variância do receptor. Ocorre uma colisão quando o *SINR* no receptor se tornar menor do que o limite de recepção em qualquer instante durante a recepção.

A partir destas ferramentas, foram desenvolvidos ambientes simulados para permitir a implementação e teste de diversas topologias de redes de sensores sem fio, bem como a geração de dados para treinamento das redes neurais com mais fidelidade em relação aos dados de uma aplicação real.

2.6.2 Toolbox de Redes Neurais Artificiais

Este trabalho utilizou a *toolbox* de redes neurais artificiais do software *MatLab*, da empresa *Mathworks*. Através desta, é possível realizar a criação, treinamento e simulação deste tipo de rede. Esta *toolbox* permite que o projetista tenha um grande número de graus de liberdade, podendo utilizar funções próprias ou funções já existentes na *toolbox* para implementação.

Dentre as possibilidades existentes nesta ferramenta, citam-se algumas principais:

- **treinamento de redes supervisionadas:** utilizando estruturas do tipo *feed-*

forward (utilizada neste trabalho em conjunto com treinamento supervisionado), de base radial, de atraso de tempo, entre outras;

- **treinamento de redes não supervisionadas:** podendo ser com camadas competitivas ou mapas auto-organizados;
- **flexibilidade:** permite utilização de funções criadas pelo usuário além das funções desenvolvidas para o *toolbox*;
- **rotinas para evitar *overfitting*:** para evitar perda de generalização das soluções encontradas.

O treinamento e aprendizado dá-se através da seleção do algoritmo e escolha das opções básicas. Estes algoritmos visam encontrar os pesos corretos para a estrutura da rede. O algoritmo de treinamento trabalha em nível global, modificando os pesos de toda a rede, enquanto a função de aprendizado pode ser utilizada para nodos individuais. As opções de escolha destes algoritmos são diversas, citando-se: métodos baseados em gradiente descendente, métodos de gradiente conjugado, algoritmo de Levenberg-Marquardt, algoritmo de propagação reversa (*backpropagation*) resiliente.

As funções de treinamento podem ser: gradiente descendente, aprendizado Hebbiano, LVQ (do inglês, *learning vector quantization*), Widrow-Hoff e Kohonen.

Além das funções diretamente envolvidas no processo de criação e treinamento das redes, a ferramenta também provê ao usuário recursos de pré-processamento e pós-processamento. No que tange as entradas, estas funções visam adequá-las da rede para colocá-las no intervalo entre $[-1, 1]$, bem como redução da dimensionalidade das entradas quando isto é possível. Quanto a saída, realiza análises de regressão entre a resposta da rede e os alvos do treinamento. O usuário pode utilizar seu conjunto de dados na ferramenta sem preocupar-se com escalamento do vetor de entradas, visto que isto já é realizado automaticamente.

Ainda, a *toolbox* de redes neurais permite que o problema de *overfitting* seja evitado através de modificações regulares na função de avaliação de desempenho e através do uso de um conjunto de dados de entrada separados. O conjunto de treinamento é utilizado para que a escolha dos pesos seja realizada apropriadamente, e o conjunto de validação é utilizado para paralisar o treinamento quando o resultado do erro para este conjunto tornar-se muito grande.

2.6.3 Toolbox de Otimização Global

Da mesma forma que a *toolbox* de redes neurais, este trabalho utilizou a *toolbox* de otimização global do software *MatLab*, da empresa *Mathworks*. Esta ferramenta permite a busca de mínimos globais de problemas de otimização através de uma grande diversidade de algoritmos. Em especial, foi de uso deste trabalho os algoritmos genéticos, apesar da ferramenta poder trabalhar com qualquer dos algoritmos abaixo:

- busca global e de vários recomeços (*multistart*);
- algoritmo genéticos;
- algoritmo genético multiobjetivo;
- busca de padrões;
- *simulated annealing*;
- computação paralela.

Esta ferramenta permite a criação de processo de otimização com algoritmos genéticos tendo acesso aos elementos internos e grande número de opções para definição de parâmetros de controle. A ferramenta possui diversas funções previamente elaboradas para realizar a geração de soluções, seleção de cromossomos, *crossover* e controle de parâmetros, mas também dá acesso ao usuário criar os seus próprios para se adequarem ao problema específico que está sendo atacado.

As funções e opções disponíveis podem ser vistas abaixo:

- **Criação:** uniforme e viável(*feasible*);
- **Escalamento da avaliação:** baseada em ranqueamento, proporcional, truncamento, *shift* linear;
- **Seleção:** roleta, seleção uniforme estocástica, torneio, uniforme e *remainder*.
- **Crossover:** aritmético, heurístico, intermediário, disperso, de ponto único, de dois pontos;
- **Mutação:** viável adaptativo, gaussiano e uniforme.

A principal tarefa do projetista, além da escolha apropriada dos valores de parâmetros de controle e funções, é a criação da função de avaliação das soluções propostas pelo algoritmo genético (calcular o erro de localização, por exemplo). Ou seja, apesar de a otimização ser controlada pela ferramenta, cabe ao projetista definir como as respostas desta solução vão ser inseridas no problema real. A conversão do problema de otimização para o problema específico demanda mais esforços, visto que é a parte não genérica e aplicável a todos os problemas.

3 METODOLOGIA

3.1 Introdução

Neste capítulo serão descritas as simulações realizadas para avaliar os sistemas de localização propostos, bem como especificar as condições simuladas e as configurações das ferramentas utilizadas.

3.2 Métodos Propostos

Este trabalho apresenta métodos de localização de nodos em redes de sensores sem fio baseados em duas etapas. A primeira é responsável pela configuração da melhor estrutura para um sistema de localização (RNA mais adequada) visando uma determinada rede de sensores. Já a segunda, utilizando-se de dados de entrada (RSSI, neste caso), é responsável pela inferência da localização de nodos propriamente dita. É interessante ressaltar que a primeira etapa, que requer maior disponibilidade de energia, processamento e recursos de hardware, é feita *offline*. Ou seja, esta etapa ocorre fora dos nodos da rede, sendo apenas o resultado dela (RNA e seus parâmetros) utilizado diretamente nos nodos.

3.3 Cenário

O cenário utilizado para analisar o sistema de localização pode ser descrito como uma área quadrada (26 metros de lado em função da potência do MICAz, o nodo comercial usado como base para as simulações) em ambiente interno, totalizando $676m^2$. A escolha de um ambiente interno se justifica pois, de um modo geral, estes ambientes apresentam condições desfavoráveis para a localização de nodos devido às reflexões das ondas eletromagnéticas, por exemplo. Sendo assim, validando-se o sistema de localização em condições mais impróprias, provavelmente este será funcional também em condições (canal de propagação) mais amenas, como as encontradas em ambientes ao ar livre.

Neste espaço foram executados testes com até 9 nodos-âncoras (cientes de suas posições) e 81 nodos para treinamento das RNAs (chamados de nodos-treino, também cientes de suas posições). Os testes das RNAs geradas foram feitos usando um conjunto de 64 nodos de posições desconhecidas (chamados de nodos desconhecidos). Em todos os casos, tanto os nodos-âncora quanto os demais (nodos-treino e nodos desconhecidos) eram estáticos.

3.4 Configuração de Ferramentas

3.4.1 *Prowler*

De modo a melhor representar o cenário das simulações, o modelo matemático do canal de propagação utilizado pelo *Prowler* foi alterado. Ao invés de utilizar algum dos modelos nativos descritos nas seções anteriores, foi empregado o modelo de sombreamento log-normal (RAPPAPORT, 2001).

Este modelo pode ser visto na equação 3.1, onde L é a atenuação para a distância d , L_0 é a perda na distância de referência D_0 , γ é o expoente de perda de percurso e X_σ é uma variável gaussiana aleatória com média nula e desvio padrão σ . Os valores utilizados neste trabalho foram extraídos de medições empíricas feitas em (BARBERIS; BARBONI; VALLE, 2007). A distância de referência foi de 0.1 metro com potência de saída de $-25dBm$, L_0 igual a $30dBm$, γ igual a 2.5 e σ igual a 4.

$$L(d) = L_0 + 10\gamma \log_{10} \frac{d}{d_0} + X_\sigma \quad (3.1)$$

Além disso, a velocidade de transferência de dados usada no *Prowler* foi alterada de modo a adequá-lo para simular o MICAz, o modelo-alvo de nodo comercial utilizado como base neste trabalho. Este dispositivo trabalha com uma taxa de transferência de dados de $250kbps$, enquanto que a velocidade original modelada no referido simulador é de $40kbps$.

3.4.2 *Toolbox* de Redes Neurais Artificiais

Para uma geração mais dinâmica de diversas estruturas de RNAs do tipo *feedforward* e seus respectivos treinamentos com mais agilidade, foi utilizada esta *toolbox* do *MatLab*. A configuração utilizada para treinamento das RNAs foi estática, podendo ser vista na tabela 4.

Tabela 4: Parâmetros de configuração do algoritmo de treinamento das RNAs

Parâmetros	Valor
Algoritmo de treinamento	Levenberg-Marquardt
Épocas de treinamento	200
Erro-alvo	0.001
Máximo de falhas de validação	6

As estruturas das RNAs geradas foram controladas por metaheurísticas (algoritmo genético e *Simulated Annealing*, o que permitiu uma automação do processo de seleção das melhores estruturas para determinados tipos de RSSFs. Os parâmetros das RNAs passíveis de variação são o número de camadas intermediárias (sempre havendo uma de entrada e uma de saída), o número de neurônios artificiais em cada camada e também a função de transferência de cada camada da rede.

Como o objetivo é gerar RNAs para serem aplicadas em *firmwares* de nodos sensores comerciais, optou-se por limitar estes parâmetros de modo a não sobrecarregar os recursos de software e hardware dos microcontroladores embutidos nestes nodos. Os limites escolhidos (configuráveis) para os parâmetros das RNAs utilizados nos processos de otimização visando resultados para as RSSFs simuladas neste trabalho podem ser vistos na tabela 5.

Tabela 5: Limites das variáveis para otimização da estrutura das redes neurais artificiais

Parâmetro da estrutura	Limites
Número de camadas escondidas	0-3
Número de neurônios por camada	1-16
Função de transferência para cada camada	tansig ¹ , logsig ² , purelin ³ , radbas ⁴

¹Função tangente hiperbólica

²Função logística

³Função linear

⁴Função Gaussiana

3.4.3 *Toolbox* de Algoritmo Genético Multi-Objetivo

Para sua correta operação, o algoritmo genético multi-objetivo deve ter alguns parâmetros principais configurados. A tabela 6 mostra a combinação utilizada nas simulações apresentadas neste trabalho.

Tabela 6: Parâmetros de configuração dos algoritmos genéticos

Parâmetros	Valor
Tamanho da população	50
Número de gerações	20
Número de soluções de elite	1
Taxa de <i>Crossover</i>	0,8
Taxa de mutação	0,1

A geração da população de RNAs e o cruzamento entre os indivíduos ficou a cargo desta *toolbox*. As funções-custo utilizadas para avaliar o desempenho de cada indivíduo (RNA) serão descritas nas próximas seções. Cada RNA tem um código genético binário, sendo que todas as informações a respeito de sua estrutura (número de camadas, número de neurônios em cada camada e funções de transferência) são representadas.

A partir da manipulação destes códigos genéticos é possível a criação de indivíduos e o cruzamento entre eles.

3.4.4 Algoritmo *Simulated Annealing*

O algoritmo *Simulated Annealing* foi utilizado apenas na fase inicial deste trabalho, sendo apenas o seu resultado mais significativo reportado nas próximas seções. As configurações adotadas quando da utilização desta metaheurística podem ser vistas na tabela 7, sendo o seu código-fonte retirado de (KIRKPATRICK; GELATT; VECCHI, 1983).

Os parâmetros configuráveis do *Simulated Annealing* são:

- Função de Geração: Produz uma nova solução a partir de uma antiga. A solução é descrita em um vetor $\hat{x} = \{x_1, x_2, \dots, x_n\}$, contendo o valor de cada parâmetro variável x_i , com $i = 1, \dots, n$;

- Temperatura Inicial: Temperatura inicial do algoritmo. Pode ser qualquer valor positivo sendo 1 considerado o padrão;
- Temperatura Final: Temperatura de parada. Pode ser qualquer valor positivo menor do que a temperatura inicial;
- Valor de Parada: Menor valor procurado. As iterações são imediatamente interrompidas quando este valor é encontrado;
- Taxa de Resfriamento: Gera uma nova temperatura a partir da anterior. Pode ser qualquer função que gere um valor inteiro menor do que o de entrada;
- Máximo de Rejeições: Número máximo de rejeições consecutivas de soluções;
- Máximo de Tentativas: Número máximo de tentativas com a mesma temperatura;
- Máximo de Sucessos: Número máximo de sucessos com a mesma temperatura.

Tabela 7: Parâmetros de configuração do algoritmo *Simulated Annealing*

Parâmetros	Valor
Função de Geração	$x_i' = x_i + (1 + r(-2))$, com r igual a 0 ou 1
Temperatura Inicial	1
Temperatura Final	$1e^{-5}$
Valor de Parada	$-Inf$
Taxa de Resfriamento	$(0.8 \cdot T)$
Máximo de Rejeições	1000
Máximo de tentativas	10
Máximo de sucessos	8

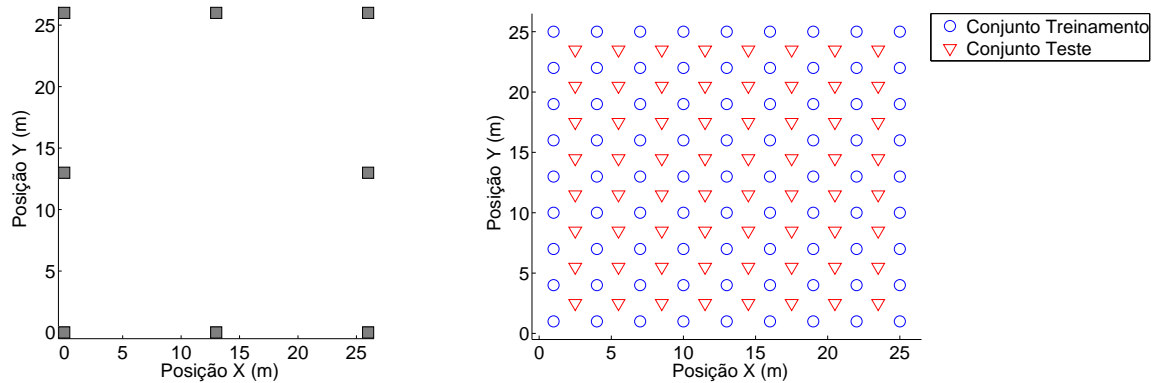
3.5 Topologias para Testes Iniciais

Na fase inicial deste trabalho foram executadas otimizações e simulações para dois casos. O primeiro foi idealizado como sendo um sistema de localização com apenas uma RNA tendo sua estrutura modelada pelo algoritmo *Simulated Annealing*. Já o segundo

teve como objetivo a geração de um sistema de localização com dois estágios, baseado em duas RNAs conectadas em cascata. O primeiro estágio foi utilizado para gerar as estimativas de localização e o segundo para reduzir o erro gerado pelo primeiro.

3.5.1 *Simulated Annealing*

Para verificar o desempenho do sistema de localização otimizado pelo algoritmo *Simulated Annealing* foi gerada uma topologia de rede contendo 8 nodos-âncora. A amostra para treinamento das RNAs foi composta por 81 nodos-treino, sendo os testes realizados por um conjunto de 64 nodos desconhecidos. Tanto o arranjo de nodos-âncora quanto os nodos para treino e teste das RNAs podem ser vistos na figura 27.



(a) Arranjo de nodos-âncoras.

(b) Conjuntos treino e teste.

Figura 27: Arranjos de nodos adotados para testes com o algoritmo *Simulated Annealing*.

A disposição dos nodos-teste em grade foi adotada visando cobrir o maior espaço possível da área simulada, porém dentro da grade composta pelos nodos-treino. A grade de teste foi posicionada de modo a não possuir pontos em comum com a grade de treinamento. Desta forma, no procedimento de teste, as RNAs foram expostas somente a informações novas, verificando assim a robustez das funções aproximadas para estimar localizações.

3.5.2 Sistema de dois estágios

Visando uma nova abordagem para o problema de localização em RSSFs utilizando RNAs, este trabalho apresenta e testa um sistema de localização composto por duas RNAs otimizadas individualmente. Ao conhecimento do autor, abordagens com mais de uma RNA em cascata ainda não haviam sido apresentadas na literatura. A otimização utilizada neste sistema foi baseada em algoritmo genético mono-objetivo.

A verificação de funcionamento do sistema de dois estágios foi realizada utilizando-se uma topologia contendo 9 nodos-âncora. Os arranjos de nodos-treino e nodos-teste para o primeiro estágio foi o mesmo utilizado no caso da otimização usando o algoritmo SA. A figura 28 mostra os nodos utilizados no primeiro estágio.

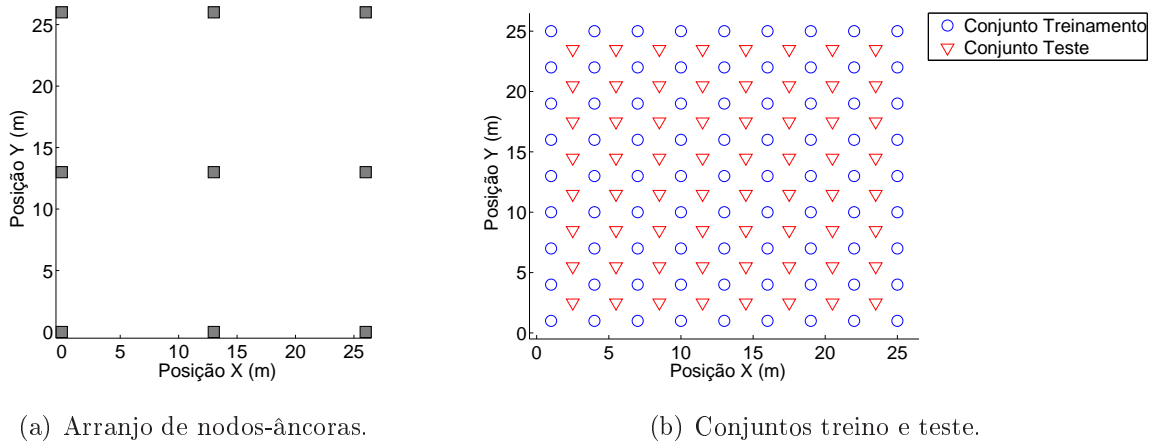


Figura 28: Arranjos de nodos adotados para testes do primeiro estágio do sistema de localização com duas RNAs.

O segundo estágio foi treinado com base nas estimativas de localização geradas pelo primeiro, e testado com um arranjo aleatório de nodos-teste para variar a forma de avaliação. Este arranjo pode ser visto na figura 29

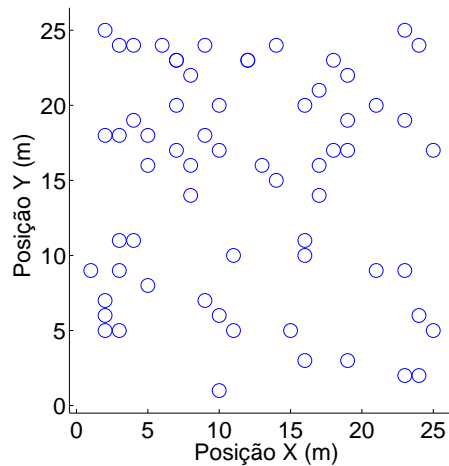


Figura 29: Conjunto de testes para o segundo estágio do sistema de localização com duas RNAs.

3.6 Topologias para Testes Principais

Os testes principais deste trabalho tomaram como base 7 conjuntos de nodos-âncora e 6 conjuntos de nodos-treino, utilizando topologias criadas por todas as combinações possíveis entre estes conjuntos (42 topologias distintas).

3.6.1 Variações no Tamanho das Amostras de Treinamento

Foram gerados 6 arranjos de nodos dispostos em formato de grade (figura 30). Tais arranjos foram compostos por 16, 25, 36, 49, 64 e 81 nodos cujas posições eram conhecidas.

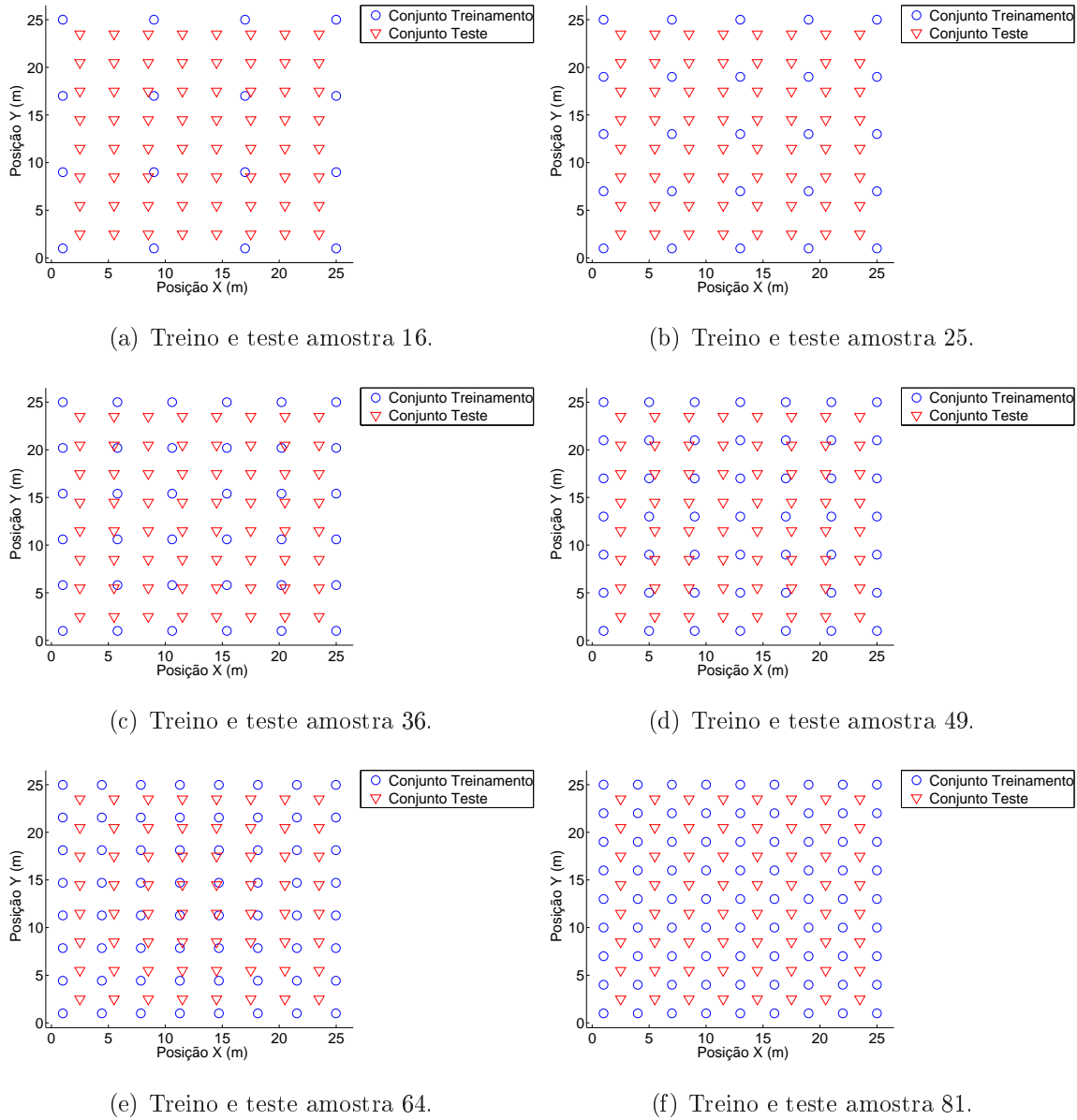
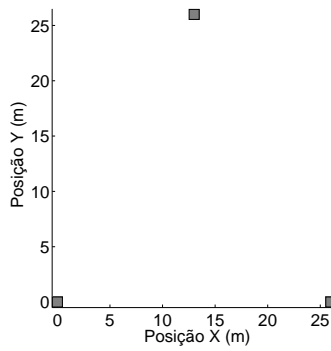


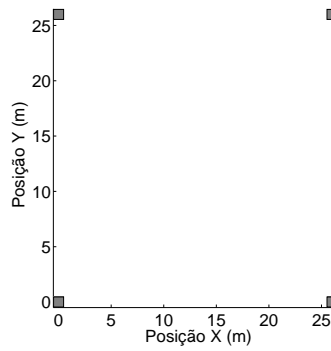
Figura 30: Variações no tamanho das amostras de treinamento.

3.6.2 Variações na Quantidade de Nodos-âncora

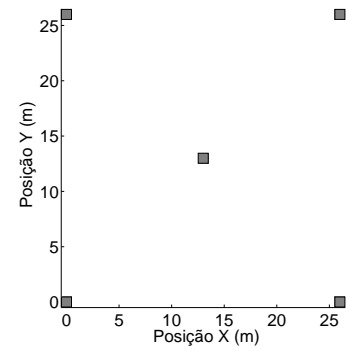
Com o objetivo de analisar o impacto da quantidade de nodos-âncora nas estimativas de localização, foram realizadas simulações variando a quantidade destes nodos de 3 até 9, com incrementos unitários (figura 31).



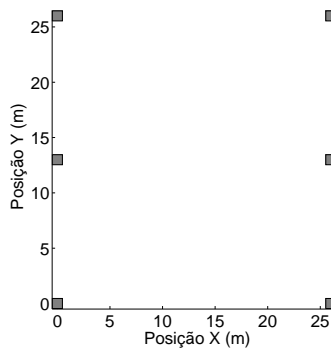
(a) Arranjo para 3 âncoras.



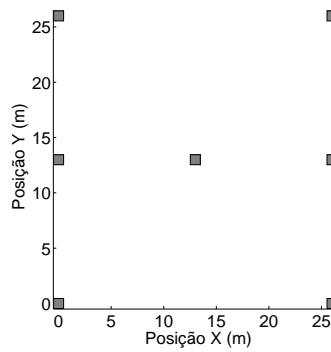
(b) Arranjo para 4 âncoras.



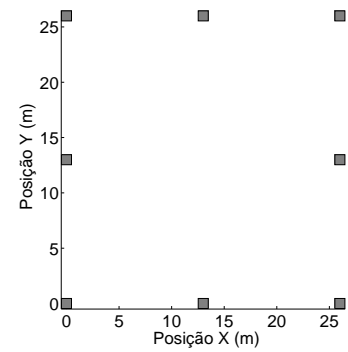
(c) Arranjo para 5 âncoras.



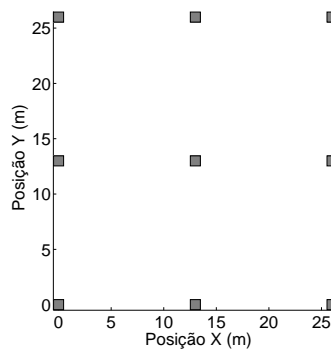
(d) Arranjo para 6 âncoras.



(e) Arranjo para 7 âncoras.



(f) Arranjo para 8 âncoras.



(g) Arranjo para 9 âncoras.

Figura 31: Disposições de nodos-âncora utilizadas.

3.7 Topologias para Testes Complementares

De modo a complementar os resultados obtidos através dos testes iniciais e principais, foram simuladas algumas topologias de RSSFs contemplando alterações nas posições de nodos-âncora e de nodos-treino.

3.7.1 Variações nas Posições dos Nodos-âncora

Com o objetivo de investigar o comportamento das RNAs diante de alterações nas posições dos nodos-âncora, este trabalho mostra algumas topologias alternativas às apresentadas nas seções anteriores. Com o arranjo de nodos-âncora em grade, a repetição de coordenadas (x ou y) foi inevitável. Desta maneira, surgiu a ideia de testar se as RNAs conseguem aproximar funções melhores utilizando-se de dados de nodos-âncora com coordenadas distintas entre si, isto é, sem repetir nenhuma coordenada x ou y .

Além da investigação acerca do número de coordenadas distintas (para âncoras) envolvidas na aproximação da melhor função para localização de nodos, foi investigada também a influência da colinearidade dos nodos-âncora.

3.7.1.1 Arranjo Alternativo - 3 Âncoras

A figura 32 mostra a disposição adotada para os nodos-âncora neste caso. Foram realizadas simulações de duas topologias distintas utilizando o mesmo arranjo de nodos-âncora, sendo uma com 36 nodos-treino e outra com 81 nodos-treino. A disposição dos nodos-treino seguem o padrão citado nos testes principais.

3.7.1.2 Arranjo Alternativo - 4 Âncoras

A figura 32 mostra a disposição adotada para os 4 nodos-âncora deste caso. Foram simuladas duas topologias distintas utilizando o mesmo arranjo de nodos-âncora, porém repetindo os arranjos de 36 e 81 nodos-treino mostrados nos testes principais.

3.7.1.3 Arranjo Alternativo - 5 Âncoras

As posições dos 5 nodos-âncora com todas as coordenadas distintas pode ser vista na figura 32. Para treino das RNAs foram usados os mesmos arranjos de 36 e 81 nodos-treino que figuram nos testes principais.

3.7.1.4 Arranjo Alternativo - 6 Âncoras

Para testar o comportamento das RNAs com 6 nodos-âncora de coordenadas distintas, foi tomado o arranjo visto na figura 32. Para treino das RNAs foram usados os mesmos arranjos de 36 e 81 nodos-treino que figuram nos testes principais.

3.7.1.5 Arranjo Circular Alternativo - 8 Âncoras

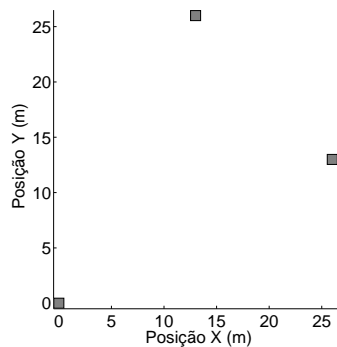
Para examinar o comportamento dos resultados gerados por nodos-âncora arranjados de maneira se aproximar de um formato circular, duas topologias foram simuladas utilizando os conjuntos de 25 e 81 nodos-treino que figuram nos testes principais. Os nodos-âncora podem ser vistos na figura 32

3.7.1.6 Arranjo Circular Alternativo - 9 Âncoras

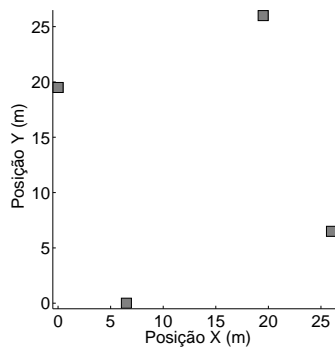
De modo a complementar o teste anterior, um nodo-âncora foi acrescentado ao centro da área simulada. Duas novas topologias foram simuladas utilizando os conjuntos de 25 e 81 nodos-treino que figuram nos testes principais. O arranjo de nodos-âncora podem ser vistos na figura 32

3.7.1.7 Arranjo Triangular Alternativo - 4 Âncoras

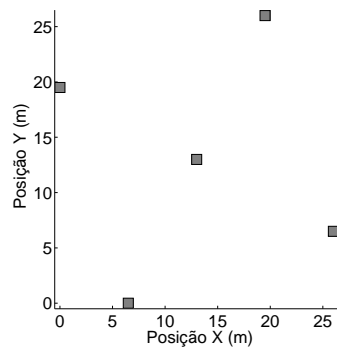
Partindo das simulações anteriores, inferiu-se que um arranjo alternativo contendo 4 nodos-âncora, sendo 3 em forma de triângulo e um posicionado ao centro da área de teste, teria possibilidades de obter bons resultados. Para testar esta hipótese este arranjo de nodos-âncora foi utilizado com as 6 topologias de nodos-treino apresentadas nos testes principais. A disposição dos nodos-âncora pode ser vista na figura 32.



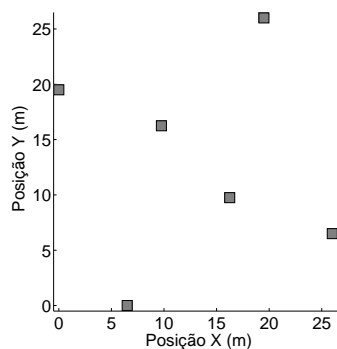
(a) Arranjo alternativo para 3 âncoras.



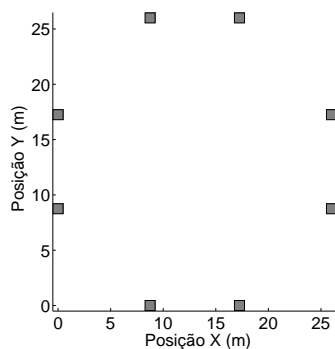
(b) Arranjo alternativo para 4 âncoras.



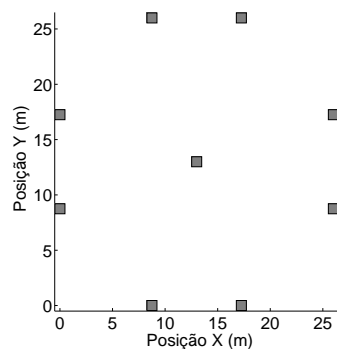
(c) Arranjo alternativo para 5 âncoras.



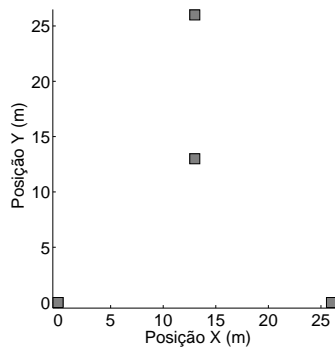
(d) Arranjo alternativo para 6 âncoras.



(e) Arranjo circular para 8 nodos-âncora.



(f) Arranjo circular para 9 nodos-âncora.



(g) Arranjo triangular para 4 nodos-âncora.

Figura 32: Disposições alternativas de nodos-âncora.

3.7.2 Variação nas Posições dos Nodos de Treinamento

Para complementar os testes de variação do tamanho das amostras para treinamentos das RNAs, foi realizada a simulação de uma topologia com 25 nodos-treino de modo que suas coordenadas fossem distintas, isto é, que não houvesse mais de um nodo com a mesma coordenada (tanto x quanto y). A topologia para treino nesses moldes pode ser vista na figura 33

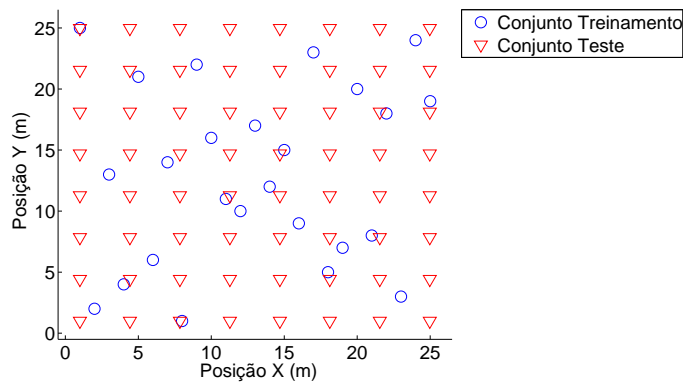


Figura 33: Arranjo de nodos-treino sem repetição de coordenadas.

3.8 Aquisição de dados

Como entrada para as redes neurais artificiais, os métodos propostos utilizam informações sobre a potência dos sinais de radiofrequência recebidos pelos nodos cujas posições são desconhecidas. A emissão destes sinais é feita pelos nodos-âncora, que possuem conhecimento sobre suas próprias localizações.

Para cada uma das topologias descritas nas seções anteriores, foram executadas simulações para coleta de dados para treino e teste das RNAs. Cada simulação gerou arquivos que foram respectivamente identificados, de modo que estes dados pudessem ser utilizados quantas vezes fosse necessário pelos algoritmos principais, evitando a necessidade de novas simulações.

Cada um dos nodos de treinamento presentes na grade coleta um conjunto de amostras. Um pacote de amostras contém o RSSI, a posição x e a posição y de todos os nodos-âncora. No ambiente de simulação desenvolvido, os nodos-âncoras enviam 30 mensagens cada, das quais as primeiras 10 são coletadas e salvas por cada um dos nodos da grade.

Visando obter dados para testes das RNAs, 64 nodos foram dispostos em formato de

grade de modo a cobrir a maior parte da área simulada e situando-se dentro das grades formadas pelas amostras de treinamento. Estes nodos-teste, da mesma forma que os nodos-treino, coletam 10 mensagens de cada nodo-âncora. As redes neurais artificiais, que foram treinadas sem acesso às posições destes nodos, são então requisitadas a estimar as posições dos nodos desconhecidos, testando assim o potencial de extrapolação das funções ocultas encontradas.

3.9 Algoritmo Principal

Todo o processo de otimização de RNAs, execução de treinamentos e testes, geração de figuras e gráficos, bem como a gravação dos dados de saída, são controlados por um algoritmo principal rodado dentro do *MatLab*. Após configurado, este algoritmo interage com todas as *toolboxes* necessárias dispensando qualquer ação do usuário.

3.9.1 Dados de Entrada

Para a realização dos processos de otimização (*Simulated Annealing* e algoritmo genético), o primeiro passo foi carregar os arquivos contendo os dados adquiridos nas simulações feitas no *Prowler*. Estes dados foram necessários para treino e teste das diversas RNAs geradas durante os processos.

3.9.2 Processos de Otimização

O próximo passo consistiu em operacionalizar a interação dos algoritmos de otimização com as estruturas das RNAs. Em tais estruturas, as características que eram passíveis de modificação durante a otimização eram o número de camadas escondidas, o número de neurônios artificiais presentes em cada camada e também as funções de transferência utilizadas por cada camada. Os limites inferior e superior para cada característica foram controlados de modo a coincidir com as configurações das RNAs descritas anteriormente.

3.9.2.1 *Simulated Annealing*

Para a aplicação do algoritmo (*Simulated Annealing*, as RNAs foram representadas por vetores de números inteiros, sendo cada posição deste vetor ocupada por uma característica da estrutura. Cada solução gerada por esta otimização era composta por um vetor de inteiros, representado então uma estrutura de uma RNA. Para viabilizar o início

do processo, foi necessário fornecer uma solução inicial (uma RNA em forma de vetor), sendo que foram executados dois casos partindo de soluções iniciais diferentes, as quais serão descritas na seção de resultados.

Cada alteração feita pelo algoritmo SA modificava de modo unitário apenas uma das posições do vetor representando a solução anterior, movendo-se desta forma dentro do espaço de soluções. A representação de uma RNA na forma de vetor utilizada no algoritmo SA pode ser vista na figura 34, onde o trecho A é responsável por indicar o número de camadas escondidas, trecho B responsável pela indicação da quantidade de neurônios em cada camada e o trecho C informa as funções de transferência de cada camada.

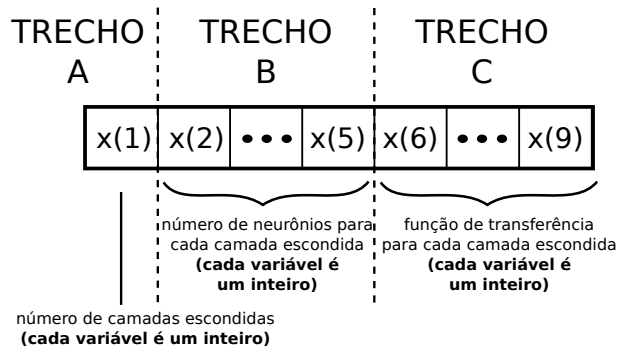


Figura 34: Representação de uma RNA utilizada no algoritmo SA.

A função-custo utilizada para avaliar cada solução sugerida consiste na geração do cálculo da raiz do erro quadrático médio (REQM) entre a posição real e a posição estimada para todas as amostras, onde i é o índice do nodo, x_i e y_i são as coordenadas reais, \hat{x}_i e \hat{y}_i são as coordenadas estimadas e n é o número total de nodos. O cálculo deste erro pode ser visto na equação 3.2.

$$REQM = \sqrt{\frac{1}{n} \sum_{i=1}^n [(x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2]} \quad (3.2)$$

O fluxo completo para o algoritmo principal utilizando o *Simulated Annealing* é mostrado na figura 35.

3.9.2.2 Algoritmo Genético Multi-Objetivo

De modo a viabilizar a manipulação das estruturas das RNAs pelo algoritmo genético, um código binário foi utilizado para representar cada uma delas. Desta forma, cada indivíduo (RNA) foi representado por um código genético contendo todas as informações

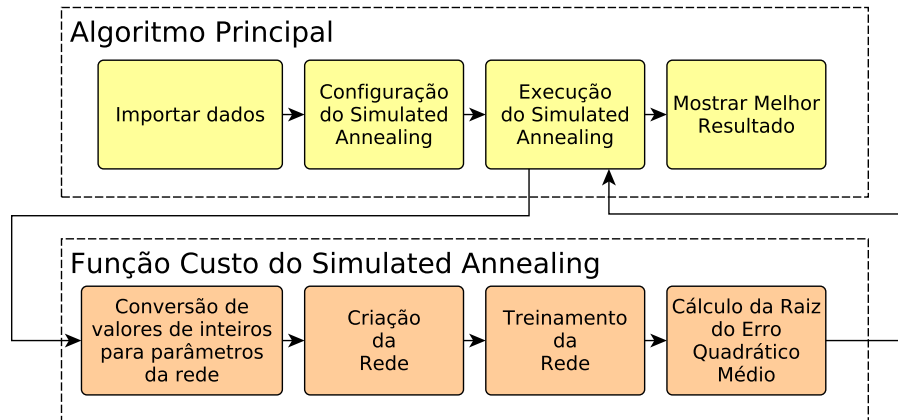


Figura 35: Otimização baseada em *Simulated Annealing*.

acerca de sua estrutura.

O modelo de estrutura genética dos indivíduos pode ser visto na figura 36, onde o trecho A é responsável por indicar o número de camadas escondidas (3 bits), trecho B responsável pela indicação da quantidade de neurônios em cada camada (4 bits por camada) e o trecho C informa as funções de transferência de cada camada (2 bits por camada).

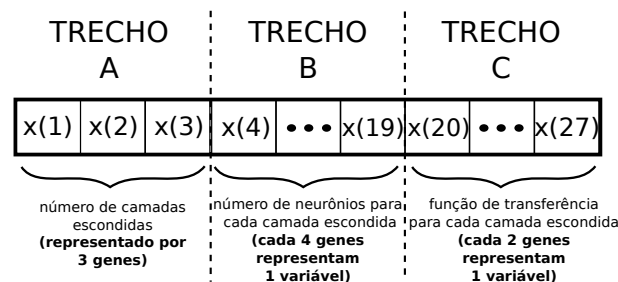


Figura 36: Código genético para representação de RNAs utilizado.

Uma população inicial de 50 indivíduos uniformemente distribuídos do limite inferior até o limite superior foi criada e evoluída por 20 gerações, sendo que ao longo deste processo cada uma das soluções propostas, uma rede neural artificial é criada, treinada e testada para avaliar o seu desempenho levando-se em consideração dois objetivos: Obter o menor REQM e o menor número de neurônios. O fluxo completo da otimização é similar ao apresentado no caso do algoritmo SA e pode ser visto na figura 37

3.9.3 Dados de Saída

Ao final da execução da otimização baseada no algoritmo SA, os arquivos de saída apresentaram a melhor estrutura de RNA encontrada no caminho percorrido no espaço de soluções, bem como os parâmetros de configuração desta. Entende-se por espaço

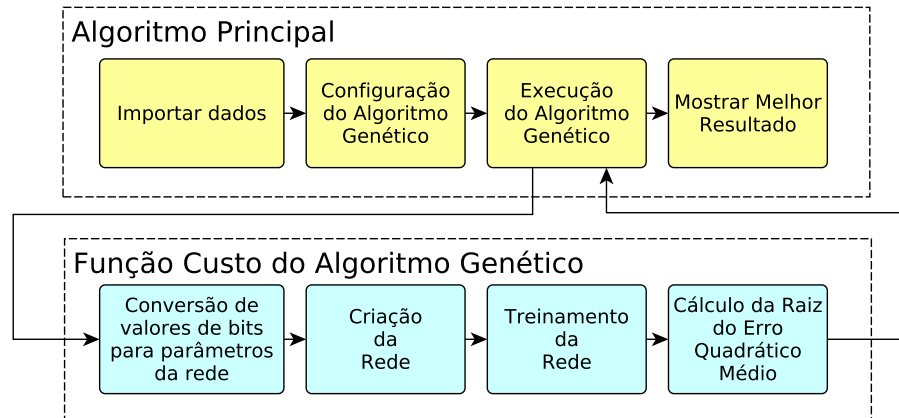


Figura 37: Otimização baseada em algoritmo genético.

de soluções todas as RNAs possíveis de serem geradas variando-se as características de suas estruturas conforme os limites expostos anteriormente. Além dos dados sobre a melhor RNA encontrada, são apresentadas também figuras com as melhores estimativas de localização obtidas e o erro em função da posição de cada nodo.

Após a conclusão do algoritmo baseado em otimização via algoritmo genético multi-objetivo, são apresentados os mesmos tipos de saída descritos no caso do algoritmo SA, porém novas informações são apresentadas: é gerado um gráfico com a fronteira de Pareto, sendo que as informações sobre estes pontos estão presentes nos arquivos de saída. Através das fronteiras de Pareto é possível tomar decisões de projeto de sistemas de localização com mais clareza e eficiência.

4 *RESULTADOS*

4.1 Introdução

Nesta seção serão apresentados e analisados os resultados das simulações executadas para os casos descritos no Capítulo 3, que trata da metodologia deste trabalho. Os dados serão expostos de forma a comparar as topologias simuladas, fazendo uso basicamente de figuras, gráficos e tabelas. Em seguida, será feita uma comparação dos resultados obtidos através dos métodos propostos neste trabalho com resultados de outras abordagens encontradas na literatura no que se refere ao erro presente nas estimativas de localização.

Conforme descrito no capítulo de metodologia, a principal ferramenta utilizada para realização das simulações foi o *MatLab*, sendo a utilização de suas *toolboxes* (Redes Neurais, Otimização Global e *Prowler*) de extrema importância para manter o dinamismo dos ensaios.

4.2 Testes Iniciais

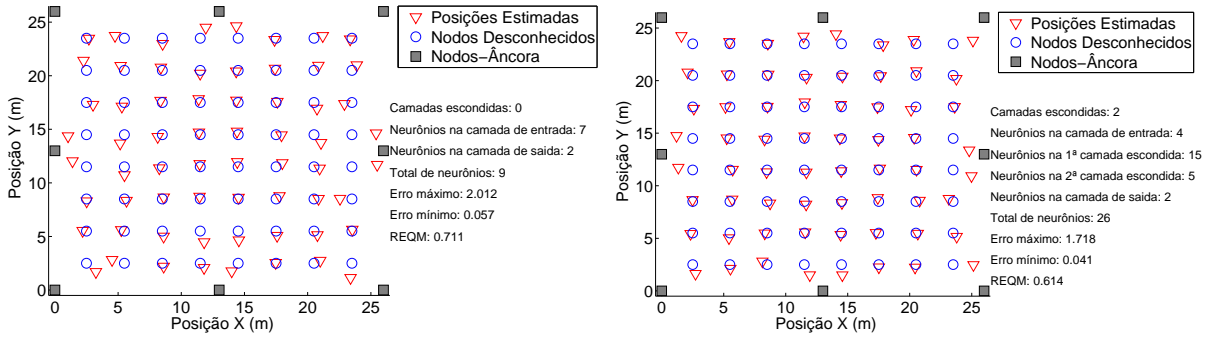
4.2.1 *Simulated Annealing*

Observando-se a figura 38(a) é possível notar que a simulação que partiu de uma solução inicial aleatória obteve REQM maior (0,711 metro), porém com uma RNA de apenas duas camadas (9 neurônios artificiais no total). Já o sistema derivado da solução inicial gerada por algoritmo genético (38(b)) obteve um REQM menor (0,614 metro), porém gerou uma RNA de 4 camadas (26 neurônios artificiais no total). Cabe ressaltar que, no segundo caso, apenas a solução inicial foi gerada por algoritmo genético, sendo a otimização principal baseada em *Simulated Annealing*.

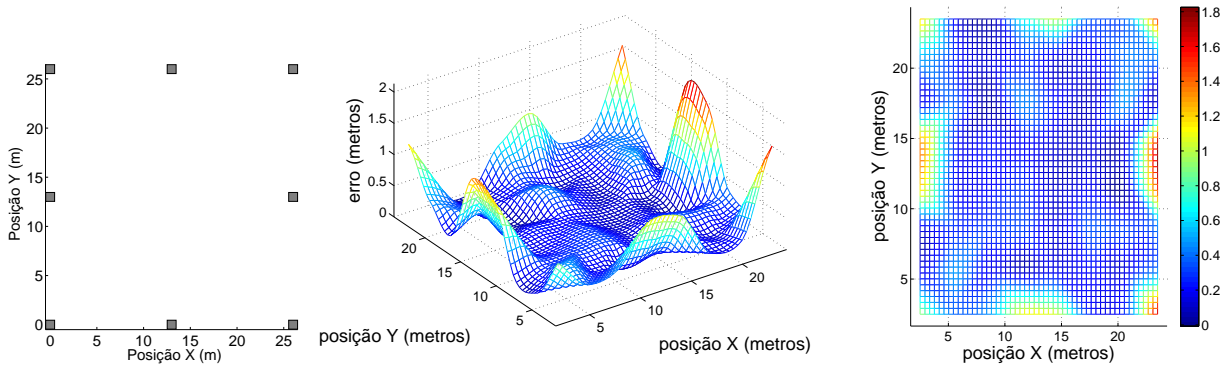
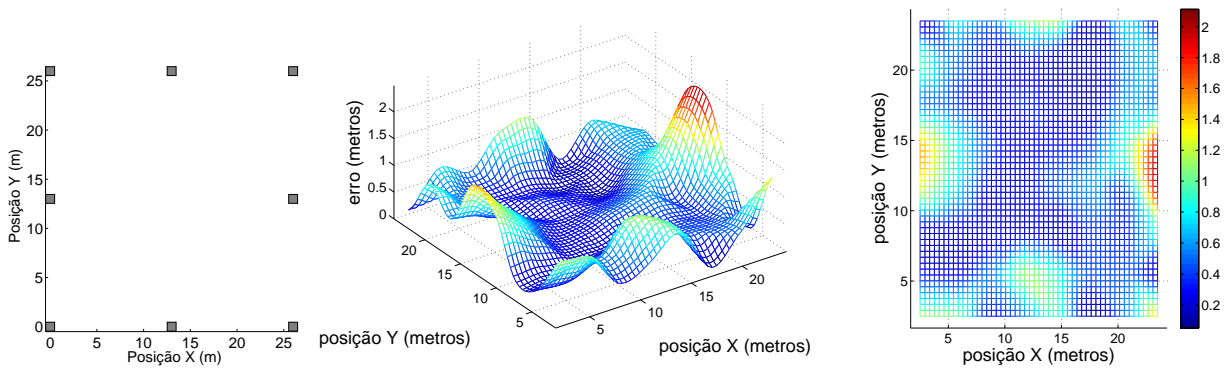
A partir da análise das figuras 38(c) e 38(d), é possível relacionar o comportamento do REQM com a posição dos nodos, sendo que os maiores valores para REQM ocorreram mais próximo aos nodos-âncora. Uma vez que não se tem acesso à função gerada pela RNA para aproximar a localização dos nodos, este comportamento do REQM nas proximidades dos nodos-âncora não possui justificativa clara.

Uma hipótese reside no fato de que, no centro da área de teste (mais distante dos nodos-âncora), os valores de RSSI medidos são mais semelhantes, sem valores discrepantes nos dados obtidos de um nodo-âncora em relação a outros. Nas bordas, os nodos-teste possuem valores altos referentes ao nodo-âncora mais próximo, o que vai contra a maioria das amostras obtidas como um todo.

Em (CHAGAS; MARTINS; OLIVEIRA, 2012a) é feita uma comparação entre o desempenho da RNA gerada por otimização *Simulated Annealing* com o desempenho de uma RNA gerada através de algoritmos genéticos.



(a) Estimativas partindo de solução inicial aleatória. (b) Estimativas partindo de solução inicial dada por AG.



(d) Erro em função da posição usando solução inicial dada por AG.

Figura 38: Influência do tipo de solução inicial do *Simulated Annealing* nas estimativas de localização para 8 nodos-âncora.

4.2.2 Motor de localização de 2 estágios

Na figura 39, para o caso do sistema de dois estágios, além da solução final (utilizando as duas RNAs em cascata), foram apresentados os resultados para a etapa intermediária (primeira RNA). Desta forma, ficou claro a diferença de desempenho quando se aplica o mesmo teste para um sistema de localização composto de uma RNA e outro composto de duas RNAs em cascata.

A REQM obtida e as estimativas de posição para todo o conjunto teste aplicando-se um sistema de localização com uma RNA é mostrado na figura 39(a). Os mesmos dados em relação ao desempenho do sistema de dois estágios, isto é, composto por duas RNAs em cascata, pode ser visto na figura 39(b). Nesta figura, a REQM explicitada refere-se ao sistema completo de dois estágios, ao passo que as informações sobre a RNA (número de camadas escondidas, número de neurônios artificiais, etc) referem-se apenas ao segundo estágio isoladamente.

A REQM para o sistema de um estágio foi de 0,46 metro, sendo que para este mesmo quesito o sistema de dois estágios apresentou o valor de 0,31 metro. Deve-se atentar para o fato de que o total de neurônios do sistema de dois estágios é formado pela soma dos neurônios das duas RNAs, totalizando 19 neurônios artificiais.

Nas figuras 39(c) e 39(d) é mostrada a REQM em função da posição dos nodos. As variações de REQM presentes no sistema de dois estágios foram menores em relação ao sistema de apenas um, corroborando a eficácia de uma RNA adicional visando a minimização do erro. Como as amostras de teste foram dispostas de modo aleatório, não foi possível observar uma relação clara entre o aumento do REQM em relação às posições dos nodos-âncora. Ainda assim, através da adição de um nodo-âncora no centro da área, é reforçada a hipótese de aumento do erro nas proximidades destes nodos. Da mesma forma que no caso anterior (*Simulated Annealing*) e para todos os casos seguintes, não foi possível ter acesso à função gerada pelas RNAs visando localização, de modo que não é possível dizer com certeza o porquê da REQM ser maior em certos locais da área de teste.

Em (CHAGAS; MARTINS; OLIVEIRA, 2012b) é descrito o projeto de um sistema de localização composto por uma RNA otimizada por algoritmo genético, tal qual foi feito para o primeiro estágio do sistema composto por duas RNAs.

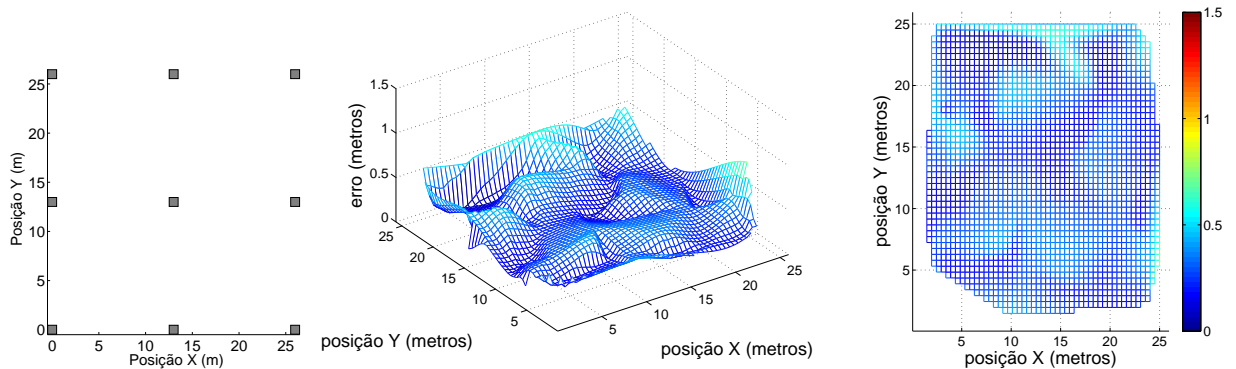
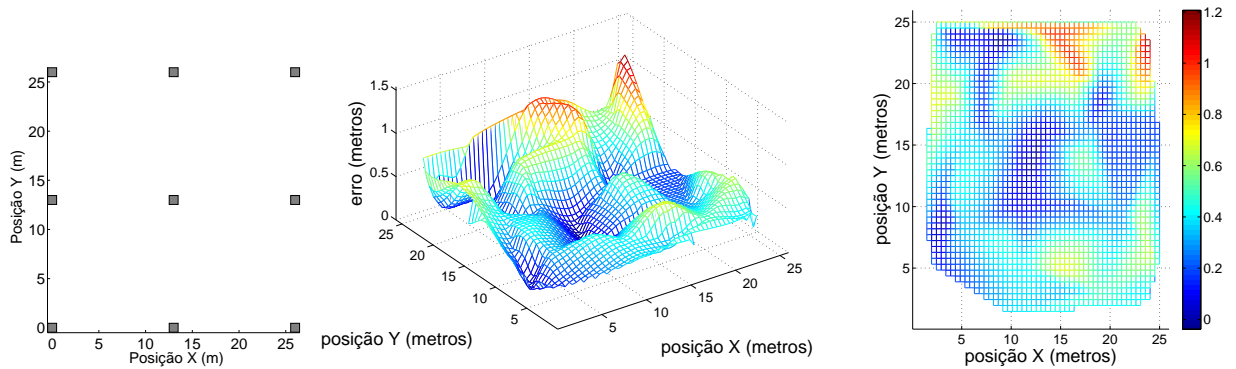
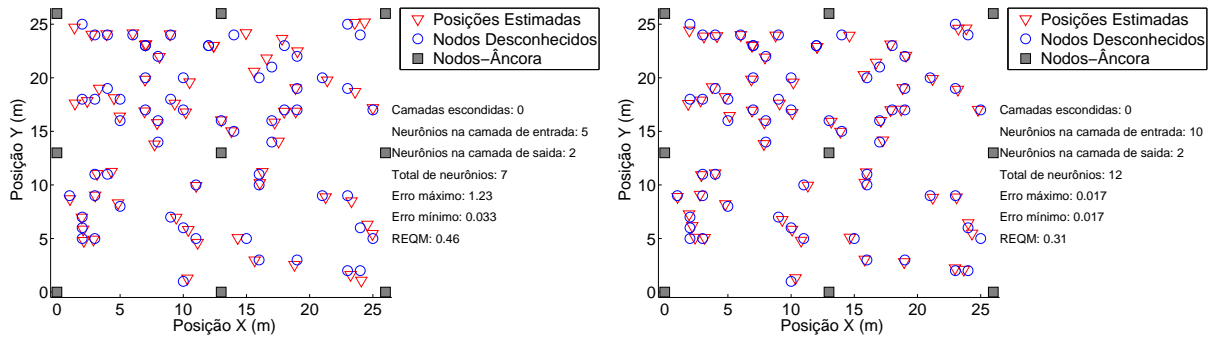


Figura 39: Comparação dos motores de localização de um e dois estágios usando 9 nodos-âncora.

4.3 Testes Principais

A partir dos resultados derivados das simulações principais (demonstrados a seguir), é possível definir de forma mais eficiente algumas variáveis básicas presentes em projetos de RSSFs tais como: número de nodos-âncora em função do erro de localização desejado, tamanho das amostras de treinamento visando um determinado patamar de erro de localização, entre outras.

Deste ponto em diante, todas as otimizações foram baseadas em algoritmo genético multi-objetivo, sendo que foram adotadas como funções-custo o número de neurônios artificiais e o indicador REQM. Os resultados serão apresentados baseados em 3 elementos gráficos:

- Figura demonstrando como a REQM se comporta em função da posição dos nodos-teste. É composta por duas vistas do mesmo resultado, sendo uma em 3 dimensões e outra em duas (vista superior da vista 3D). A vista superior do erro em função da posição tem como objetivo a melhor identificação dos pontos de pior desempenho do sistema de localização.
- Figura demonstrando as localizações corretas dos nodos-teste e as estimativas geradas pelo sistema de localização em questão para cada caso. Nesta mesma figura, estão descritas as características da RNA utilizada, mas relativas à melhor solução alcançada levando-se em conta apenas um objetivo: o menor valor possível para a REQM.
- Gráfico contendo a fronteira de Pareto, sendo importante ressaltar sua importância na análise de desempenho de um sistema de localização, especialmente quando há comparação entre abordagens distintas para o problema. Ao ocorrer um ganho de desempenho, a fronteira de Pareto mostra claramente o custo computacional (e consequentemente energético) desta melhora, permitindo assim inferir se a relação custo-benefício é aceitável.

Ainda, dentro dos testes principais, serão apresentadas tabelas resumindo os dados analisados.

4.3.1 Variação na Quantidade de Nodos-Âncora

Nesta seção serão mostrados os resultados separados por tamanho de amostra, variando-se o número de nodos-âncora.

4.3.1.1 Amostra composta por 16 Nodos-Treino

Como pode ser visto na figura 42, as RNAs formadas foram de baixa complexidade (poucos neurônios artificiais), com exceção dos resultados obtidos para 4 e 5 nodos-âncora. Porém, analisando-se as fronteiras de Pareto para cada caso (figura 43), é possível notar que, com exceção do caso para 3 nodos-âncora, obteve-se um erro entre 2 e 3 metros com RNAs de apenas 4 neurônios artificiais.

Através das figuras 40 e 41, apesar de não ser visto muito claramente devido ao tamanho reduzido desta amostra, o erro em função da posição indica que nas proximidades dos nodos-âncora é mais difícil obter boas estimativas.

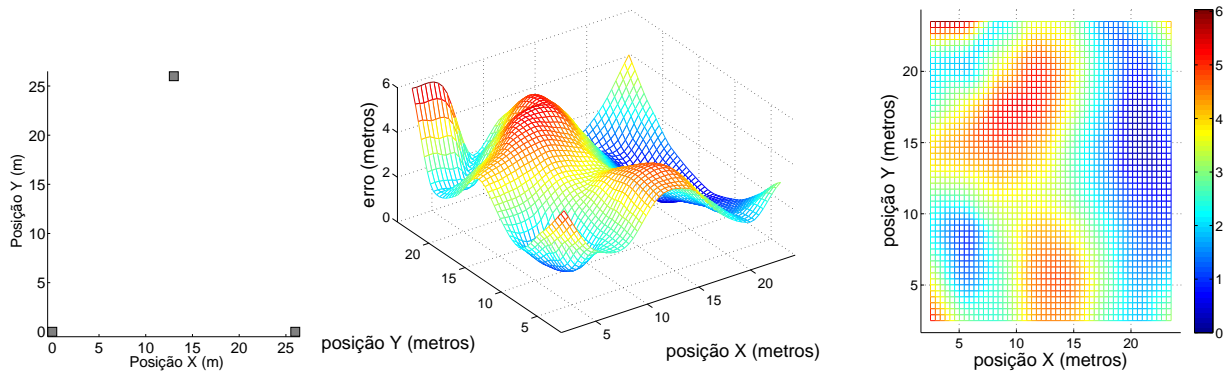
Nas tabelas 8 e 9 é possível analisar os resultados obtidos em termos de REQM e complexidade das RNAs para os casos de menor REQM obtido, ordenados por REQM e nodos-âncora.

Tabela 8: Resultados para amostra de 16 nodos de treinamento variando-se a quantidade de âncoras - ordenação por REQM

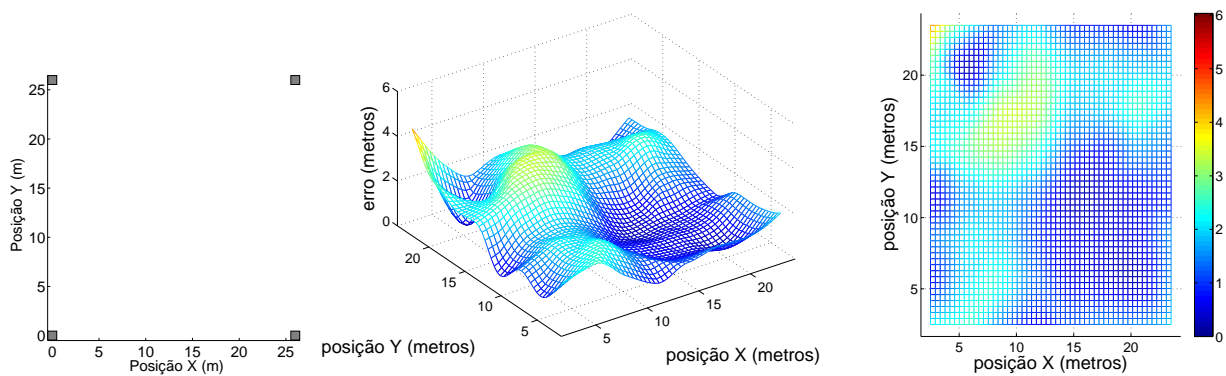
REQM (metros)	Nodos Âncora	Camadas Escondidas	Total de Neurônios	Distância Máxima (metros)
1.562	8	0	5	3.211
1.709	9	1	7	3.924
1.815	4	2	18	4.387
2.067	6	0	4	4.657
2.078	5	1	21	4.959
2.518	7	0	4	4.696
3.235	3	0	4	6.033

Tabela 9: Resultados para amostra de 16 nodos de treinamento variando-se a quantidade de âncoras - ordenação por âncoras

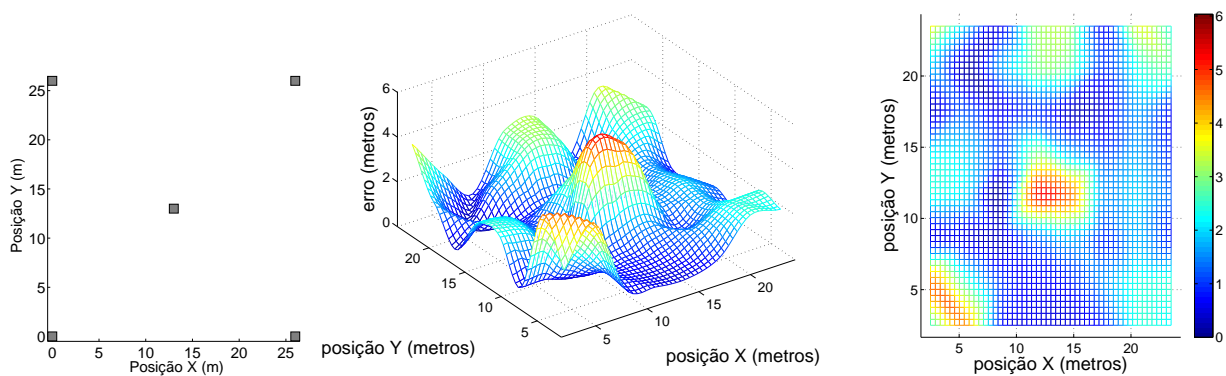
Nodos Âncora	REQM (metros)	Total de Neurônios	Camadas Escondidas	Distância Máxima (metros)
3	3.235	4	0	6.033
4	1.815	18	2	4.387
5	2.078	21	1	4.959
6	2.067	4	0	4.657
7	2.518	4	0	4.696
8	1.562	5	0	3.211
9	1.709	7	1	3.924



(a) Erro em função da posição usando 3 âncoras.

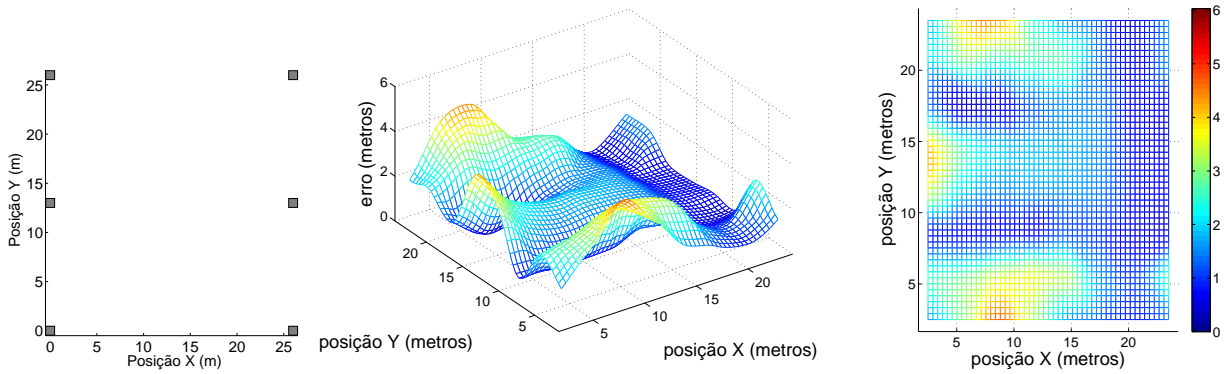


(b) Erro em função da posição usando 4 âncoras.

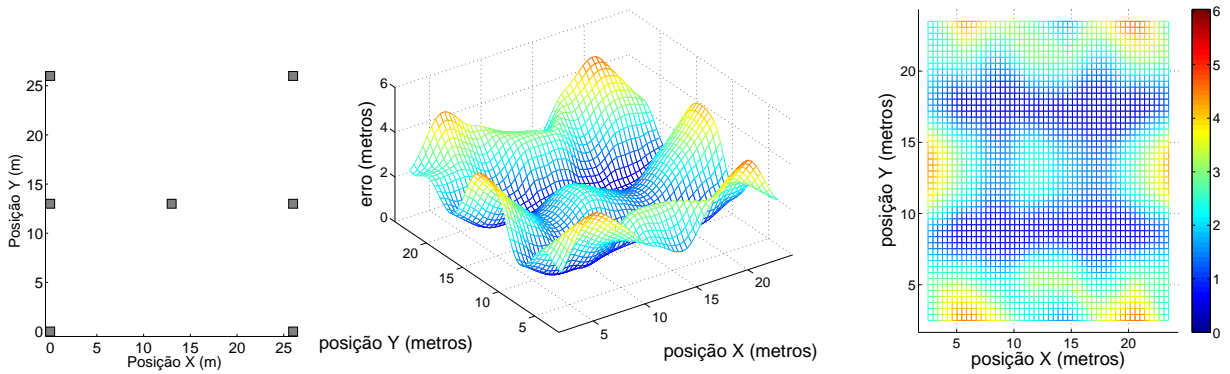


(c) Erro em função da posição usando 5 âncoras.

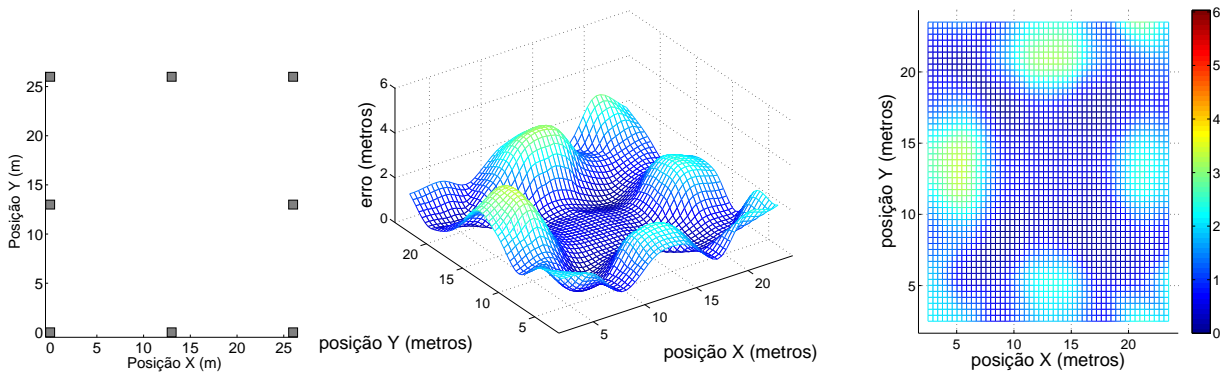
Figura 40: Influência da quantidade de âncoras no erro de localização usando 16 nodos-treino.



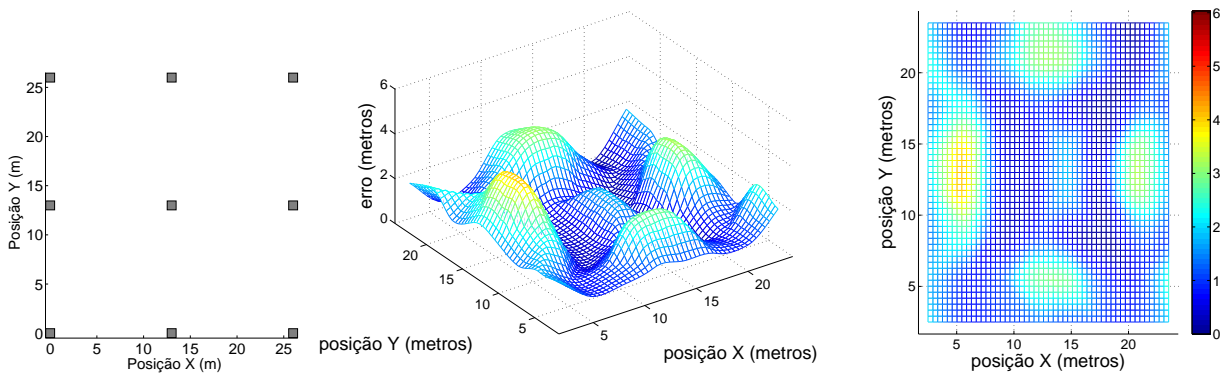
(a) Erro em função da posição usando 6 âncoras.



(b) Erro em função da posição usando 7 âncoras.

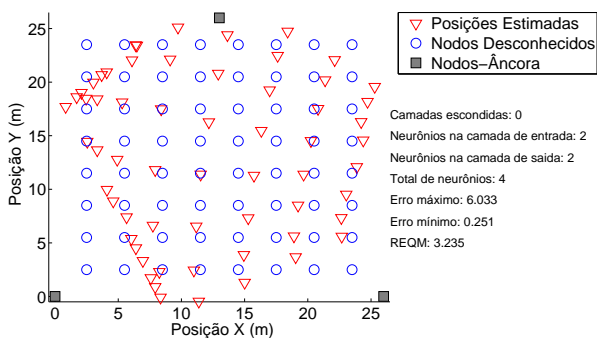


(c) Erro em função da posição usando 8 âncoras.

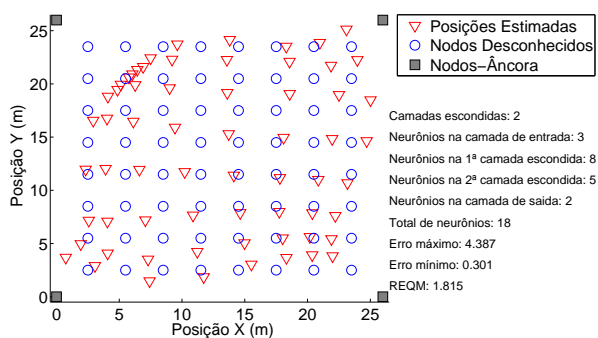


(d) Erro em função da posição usando 9 âncoras.

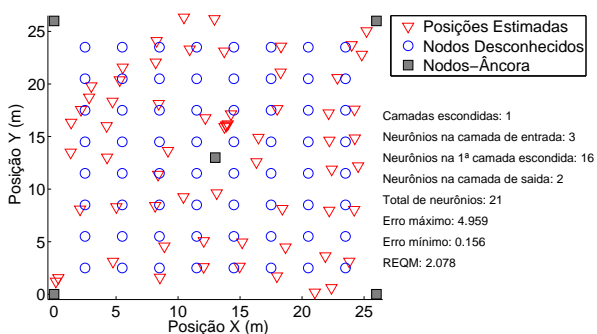
Figura 41: Influência da quantidade de âncoras no erro de localização usando 16 nodos-treino.



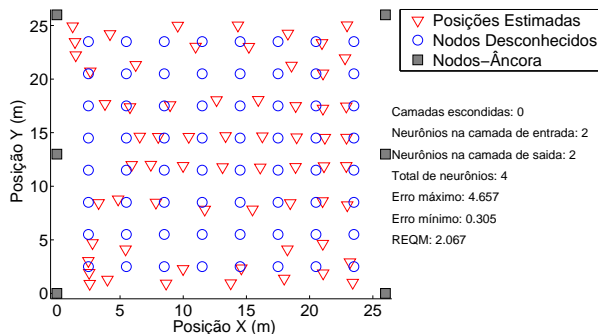
(a) Estimativas de localização usando 3 âncoras.



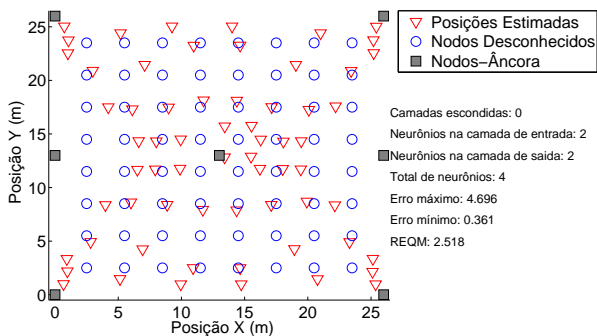
(b) Estimativas de localização usando 4 âncoras.



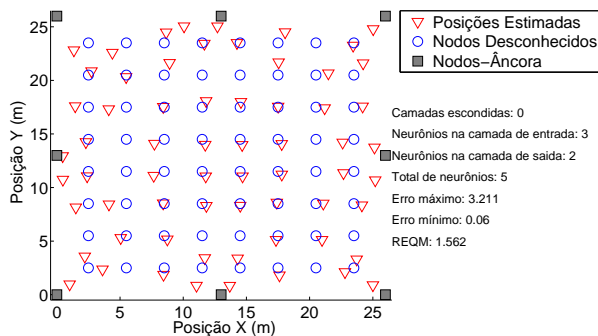
(c) Estimativas de localização usando 5 âncoras.



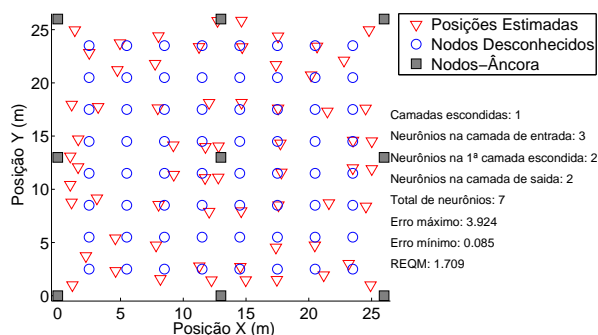
(d) Estimativas de localização usando 6 âncoras.



(e) Estimativas de localização usando 7 âncoras.

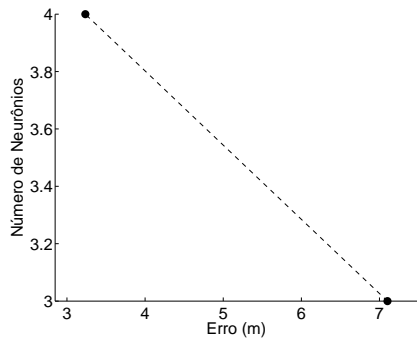


(f) Estimativas de localização usando 8 âncoras.

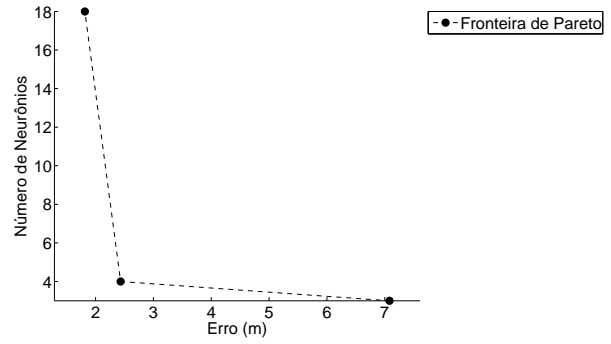


(g) Estimativas de localização usando 9 âncoras.

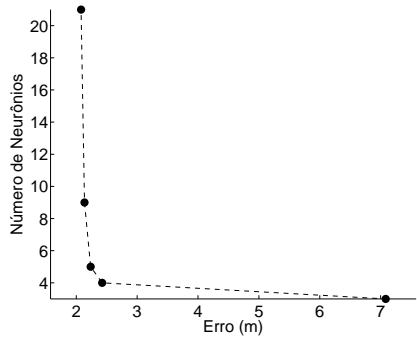
Figura 42: Influência da quantidade de âncoras nas estimativas de localização usando 16 nodos-treino.



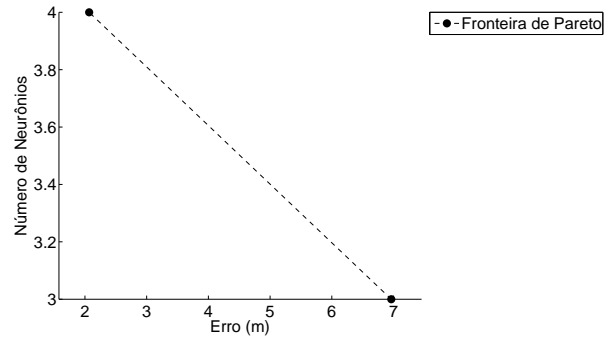
(a) Fronteira de Pareto usando 3 âncoras.



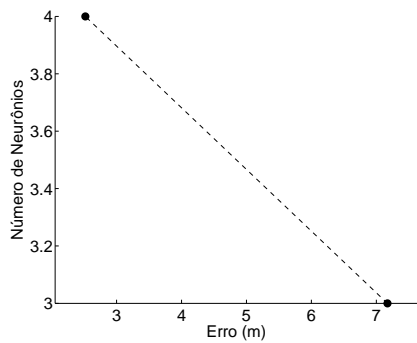
(b) Fronteira de Pareto usando 4 âncoras.



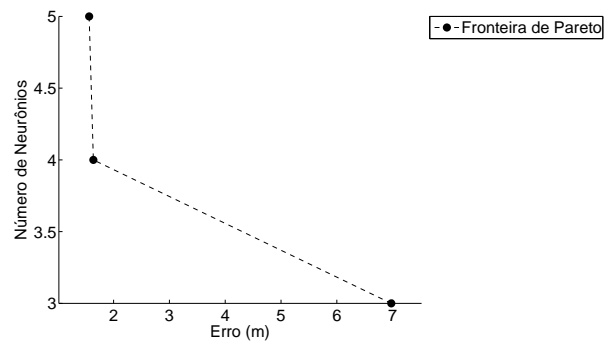
(c) Fronteira de Pareto usando 5 âncoras.



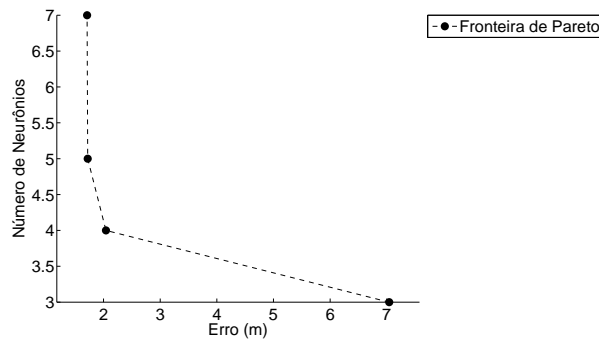
(d) Fronteira de Pareto usando 6 âncoras.



(e) Fronteira de Pareto usando 7 âncoras.



(f) Fronteira de Pareto usando 8 âncoras.



(g) Fronteira de Pareto usando 9 âncoras.

Figura 43: Influência da quantidade de âncoras nas fronteiras de Pareto usando 16 nodos-treino.

4.3.1.2 Amostra composta por 25 Nodos-Treino

Como pode ser visto na figura 46, quando comparadas ao caso anterior (16 nodos-treino), as estimativas foram melhores com o conjunto de amostras aumentado para 25 nodos-treino. É interessante observar que, em 100% dos casos com um nodo-âncora no centro da área de teste, não houve incidência de camadas escondidas nas RNAs. Analisando-se as fronteiras de Pareto para cada caso (figura 47), é possível notar que obteve-se um erro entre 1 e 2 metros com RNAs de aproximadamente 5 neurônios artificiais, sendo que os casos com 3 e 4 nodos-âncora apresentaram maior número de soluções.

As figuras 44 e 45 indicam, novamente, erros maiores nas proximidades dos nodos-âncora, denotando um aumento significativo a partir de 5 nodos-âncora.

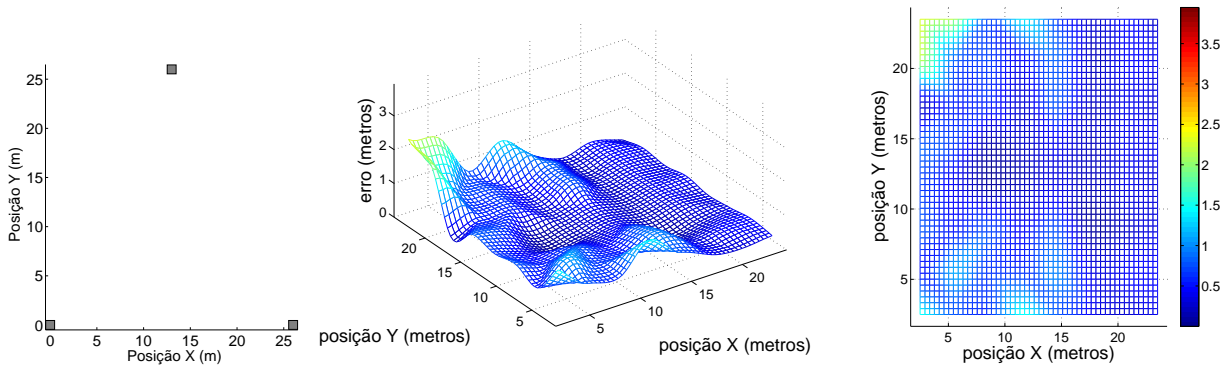
Nas tabelas 10 e 11 é possível analisar os resultados obtidos em termos de REQM e complexidade das RNAs para os casos de menor REQM obtido, ordenados por REQM e nodos-âncora.

Tabela 10: Resultados para amostra de 25 nodos de treinamento variando-se a quantidade de âncoras - ordenação por REQM

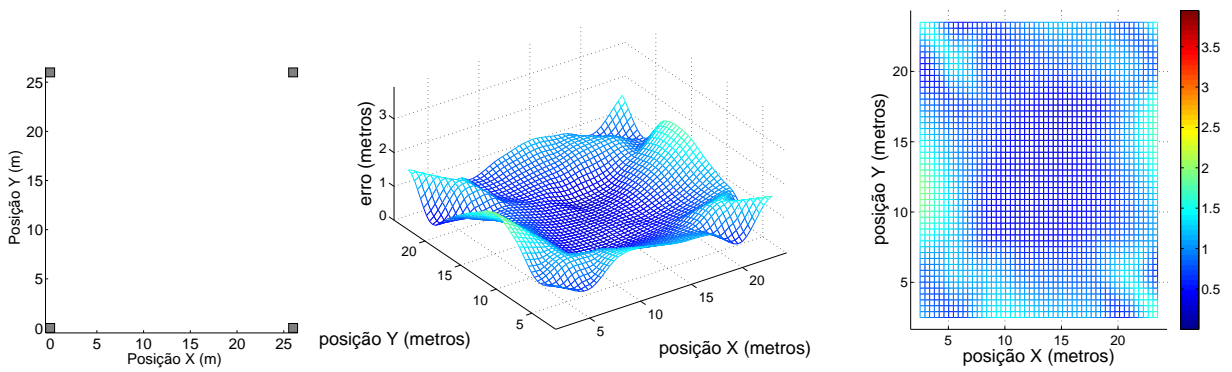
REQM (metros)	Nodos Âncora	Camadas Escondidas	Total de Neurônios	Distância Máxima (metros)
0.76	3	1	15	2.34
1.003	4	2	27	1.966
1.268	6	1	12	3.948
1.451	5	0	5	3.875
1.497	8	1	13	3.042
1.511	9	0	4	2.719
1.545	7	0	5	3.55

Tabela 11: Resultados para amostra de 25 nodos de treinamento variando-se a quantidade de âncoras - ordenação por âncoras

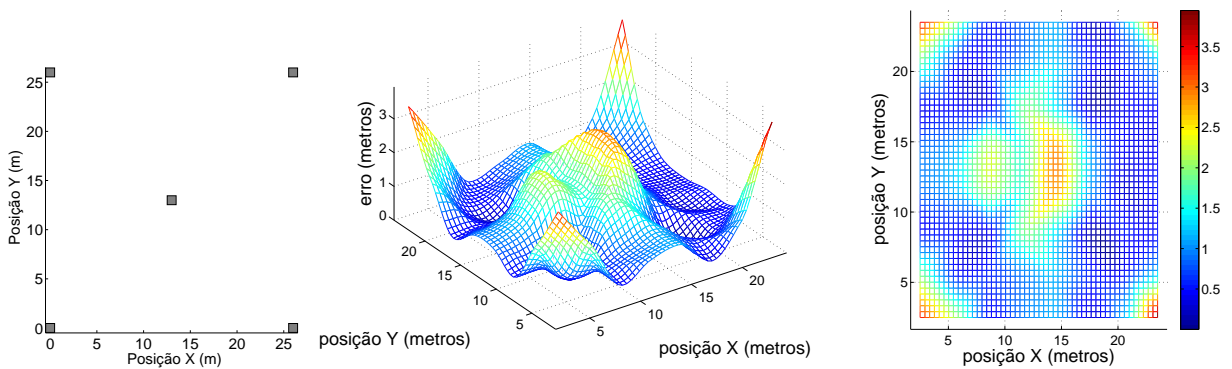
Nodos Âncora	REQM (metros)	Total de Neurônios	Camadas Escondidas	Distância Máxima (metros)
3	0.76	15	1	2.34
4	1.003	27	2	1.966
5	1.451	5	0	3.875
6	1.268	12	1	3.948
7	1.545	5	0	3.55
8	1.497	13	1	3.042
9	1.511	4	0	2.719



(a) Erro em função da posição usando 3 âncoras.

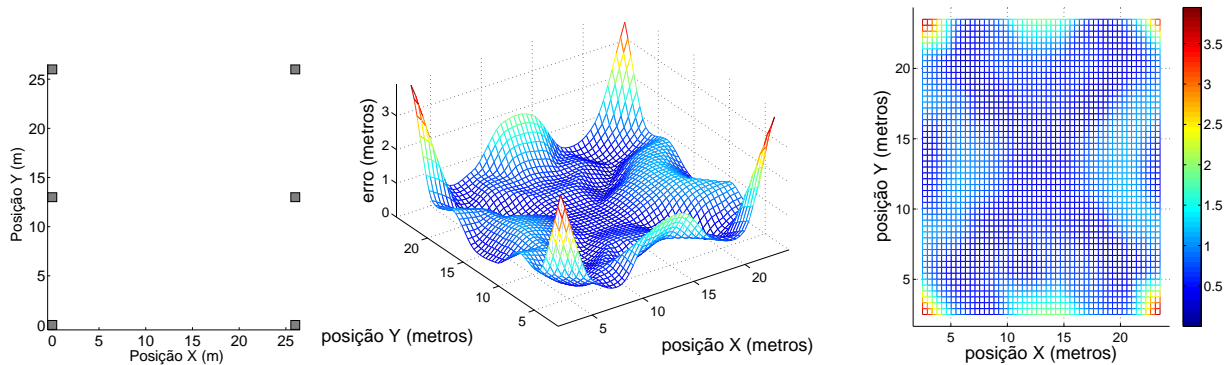


(b) Erro em função da posição usando 4 âncoras.

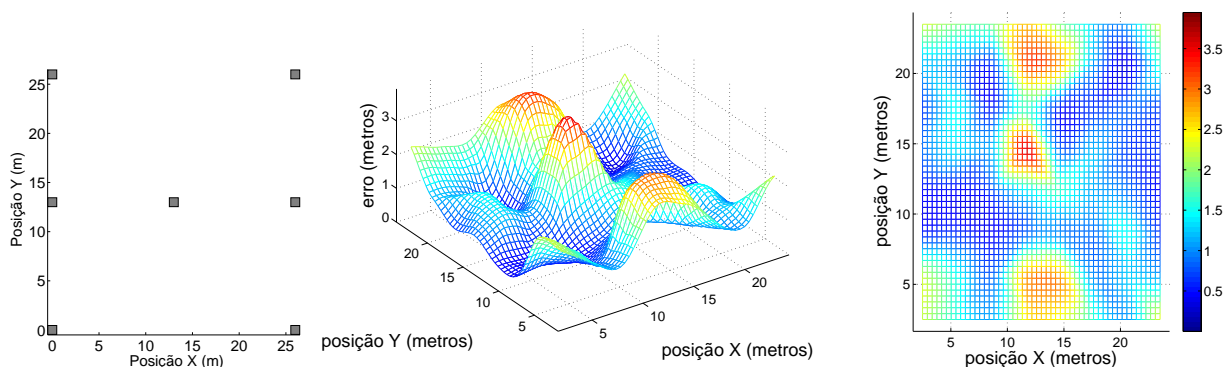


(c) Erro em função da posição usando 5 âncoras.

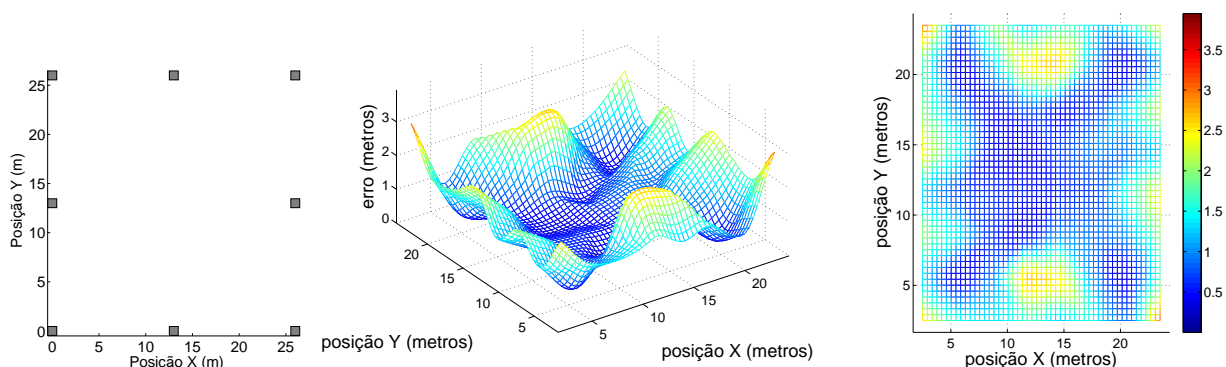
Figura 44: Influência da quantidade de âncoras no erro de localização usando 25 nodos-treino.



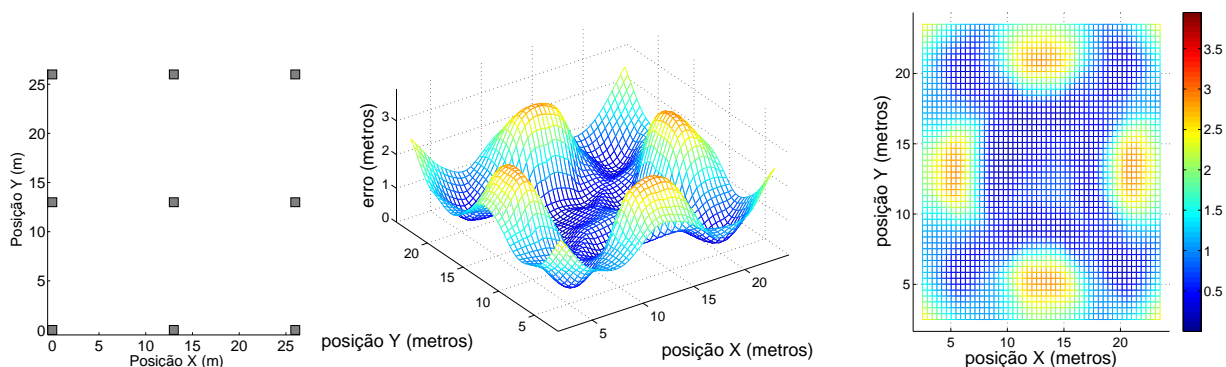
(a) Erro em função da posição usando 6 âncoras.



(b) Erro em função da posição usando 7 âncoras.

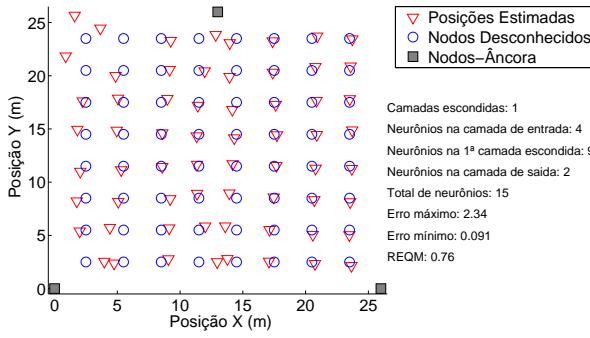


(c) Erro em função da posição usando 8 âncoras.

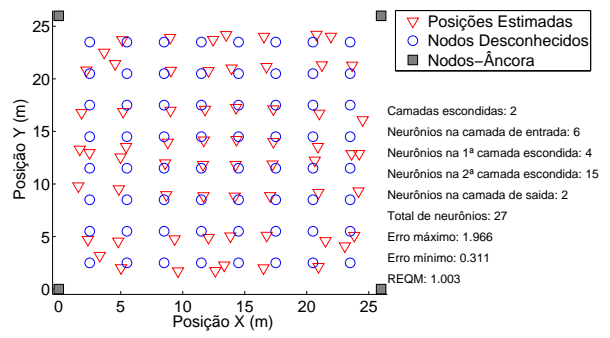


(d) Erro em função da posição usando 9 âncoras.

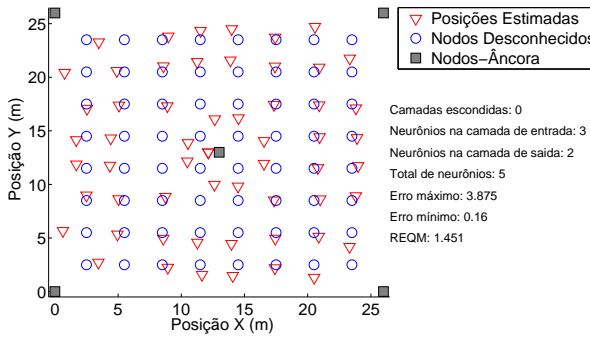
Figura 45: Influência da quantidade de âncoras no erro de localização usando 25 nodos-treino.



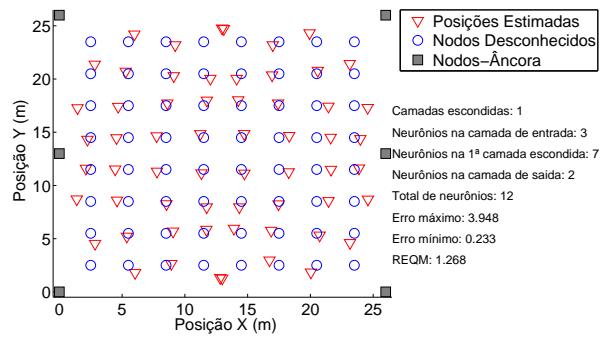
(a) Estimativas de localização usando 3 âncoras.



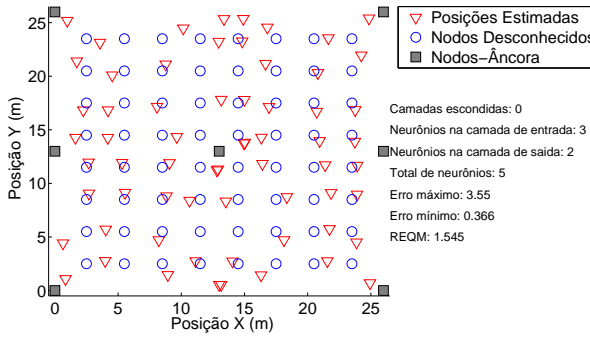
(b) Estimativas de localização usando 4 âncoras.



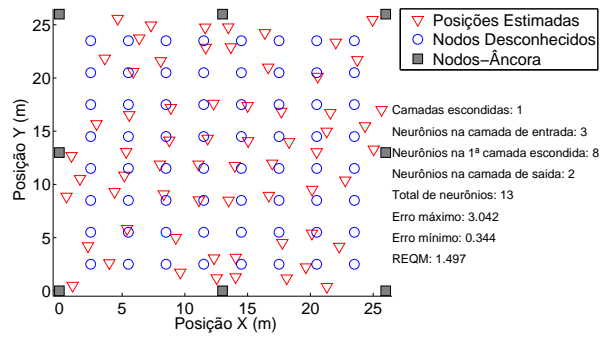
(c) Estimativas de localização usando 5 âncoras.



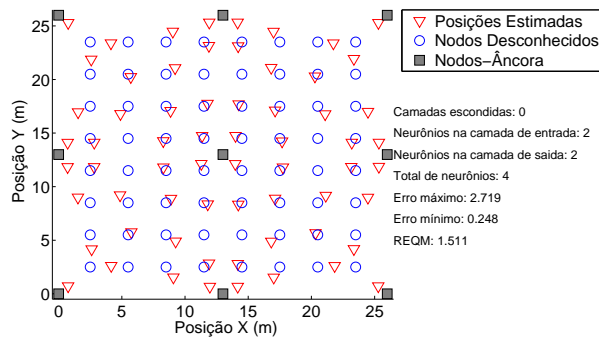
(d) Estimativas de localização usando 6 âncoras.



(e) Estimativas de localização usando 7 âncoras.

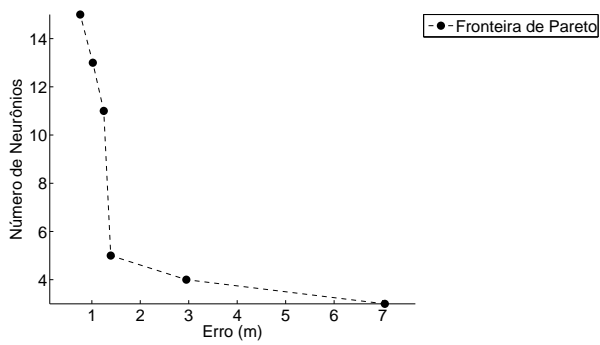


(f) Estimativas de localização usando 8 âncoras.

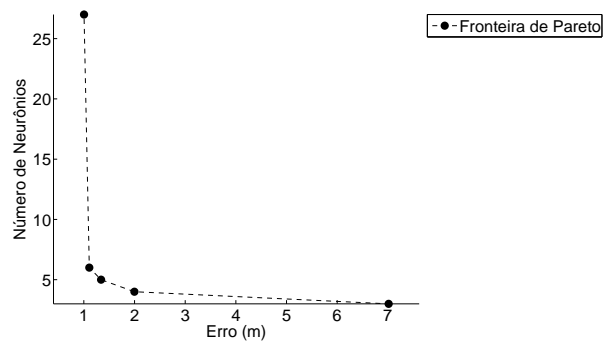


(g) Estimativas de localização usando 9 âncoras.

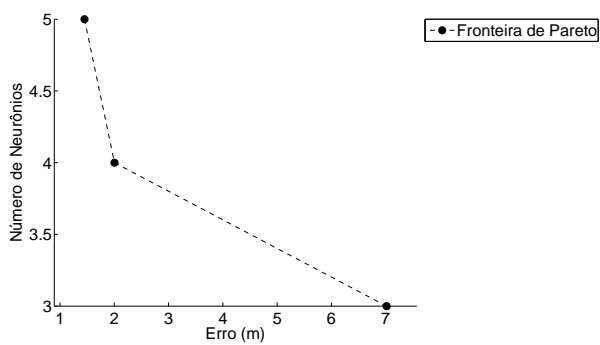
Figura 46: Influência da quantidade de âncoras nas estimativas de localização usando 25 nodos-treino.



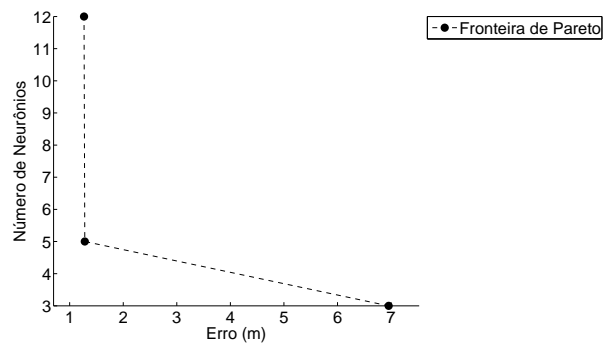
(a) Fronteira de Pareto usando 3 âncoras.



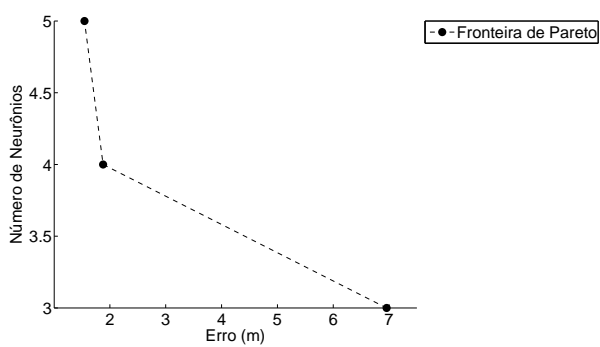
(b) Fronteira de Pareto usando 4 âncoras.



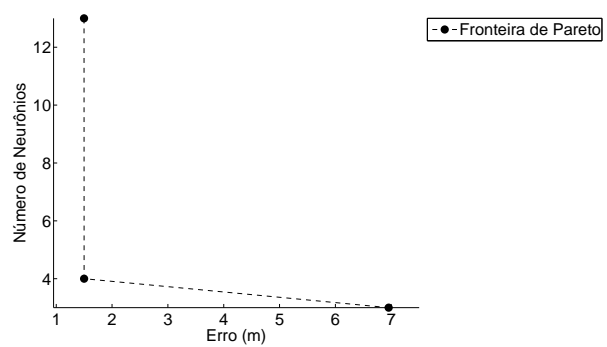
(c) Fronteira de Pareto usando 5 âncoras.



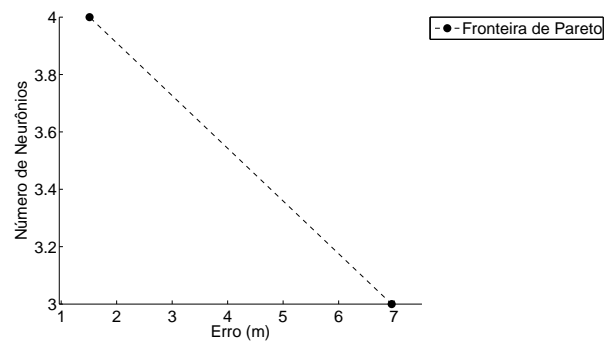
(d) Fronteira de Pareto usando 6 âncoras.



(e) Fronteira de Pareto usando 7 âncoras.



(f) Fronteira de Pareto usando 8 âncoras.



(g) Fronteira de Pareto usando 9 âncoras.

Figura 47: Influência da quantidade de âncoras nas fronteiras de Pareto usando 25 nodos-treino.

4.3.1.3 Amostra composta por 36 Nodos-Treino

Como pode ser visto na figura 50, quando comparadas ao caso anterior (25 nodos-treino), as estimativas foram melhores com o conjunto de amostras aumentado para 36 nodos-treino. Desta vez, em 66,66% dos casos com um nodo-âncora no centro da área de teste, não houve incidência de camadas escondidas nas RNAs. Analisando-se as fronteiras de Pareto para cada caso (figura 51), é possível notar que obteve-se um erro entre 1 e 2 metros com RNAs de aproximadamente 5 neurônios artificiais, sendo os casos com 3 e 6 nodos-âncora os que apresentaram maior número de soluções alternativas.

As figuras 48 e 49 indicam, novamente, uma tendência a erros maiores nas proximidades dos nodos-âncoras, sendo os arranjos com âncora central os que apresentaram pior desempenho.

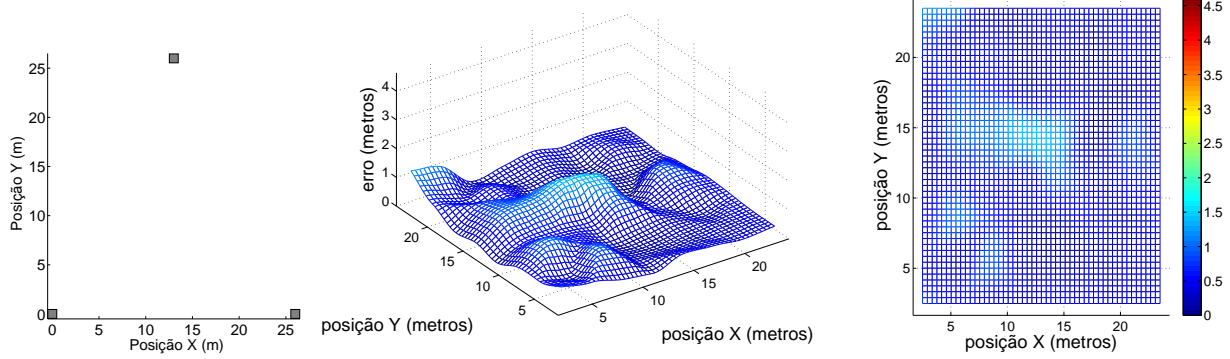
Nas tabelas 12 e 13 é possível analisar os resultados obtidos em termos de REQM e complexidade das RNAs para os casos de menor REQM obtido, ordenados por REQM e nodos-âncora.

Tabela 12: Resultados para amostra de 36 nodos de treinamento variando-se a quantidade de âncoras - ordenação por REQM

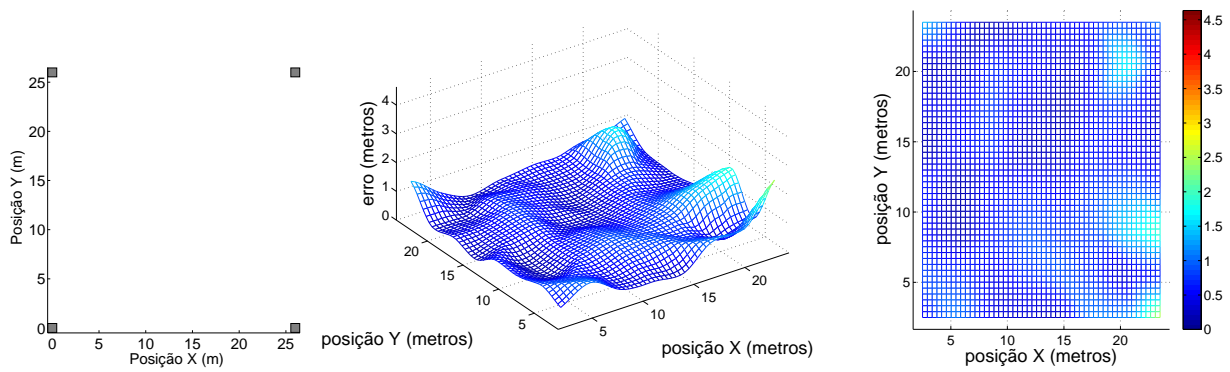
REQM (metros)	Nodos Âncora	Camadas Escondidas	Total de Neurônios	Distância Máxima (metros)
0.569	3	2	21	1.437
0.725	6	1	15	1.808
0.766	8	0	6	1.696
0.789	4	0	6	2.488
1.045	5	0	6	3.182
1.128	7	1	14	4.635
1.297	9	0	4	2.243

Tabela 13: Resultados para amostra de 36 nodos de treinamento variando-se a quantidade de âncoras - ordenação por âncoras

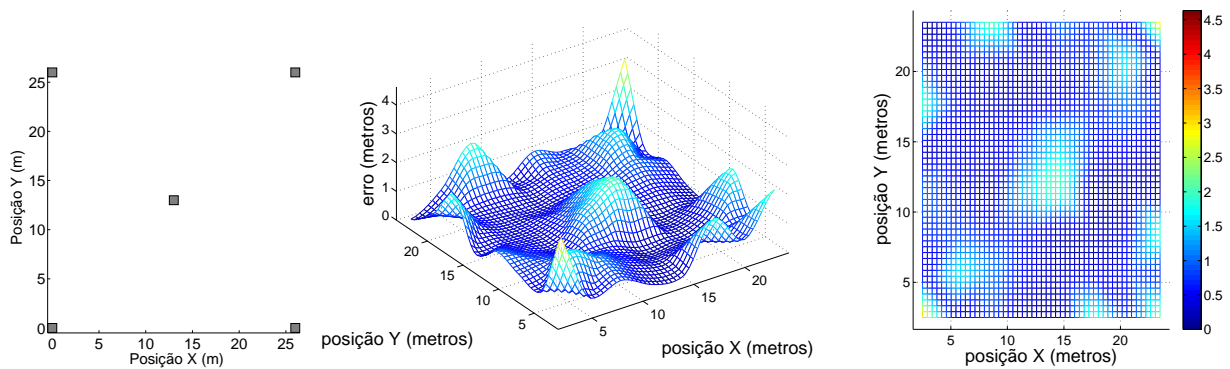
Nodos Âncora	REQM (metros)	Total de Neurônios	Camadas Escondidas	Distância Máxima (metros)
3	0.569	21	2	1.437
4	0.789	6	0	2.488
5	1.045	6	0	3.182
6	0.725	15	1	1.808
7	1.128	14	1	4.635
8	0.766	6	0	1.696
9	1.297	4	0	2.243



(a) Erro em função da posição usando 3 âncoras.

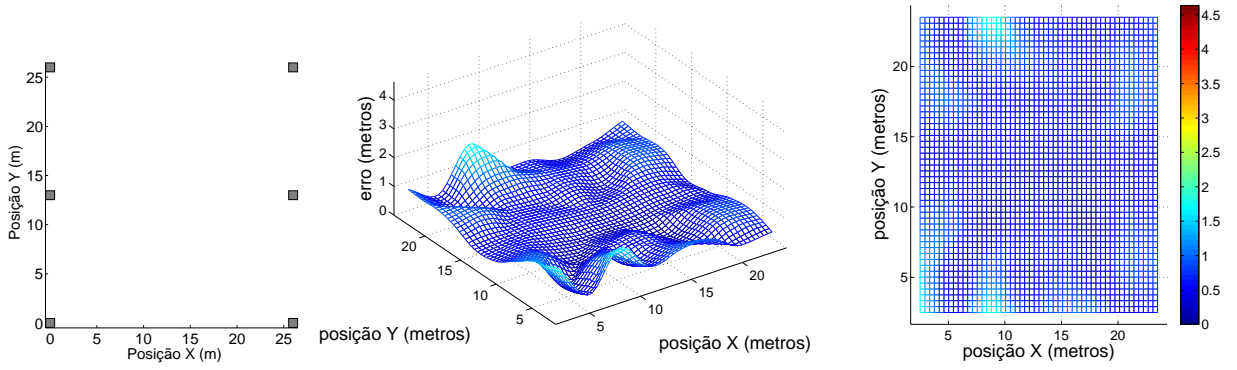


(b) Erro em função da posição usando 4 âncoras.

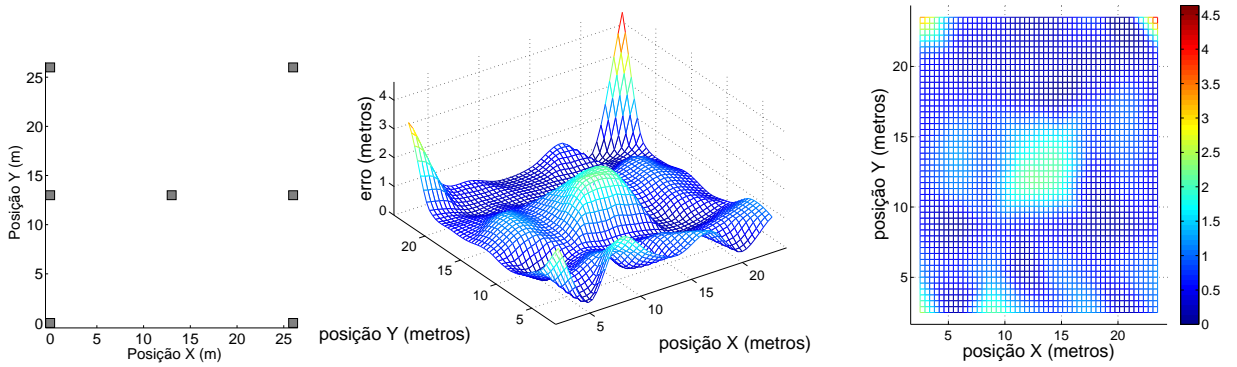


(c) Erro em função da posição usando 5 âncoras.

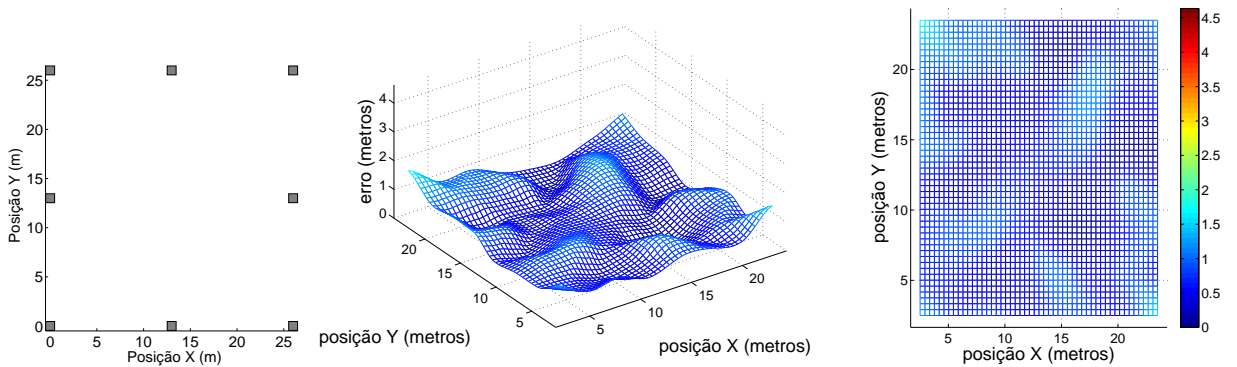
Figura 48: Influência da quantidade de âncoras no erro de localização usando 36 nodos-treino.



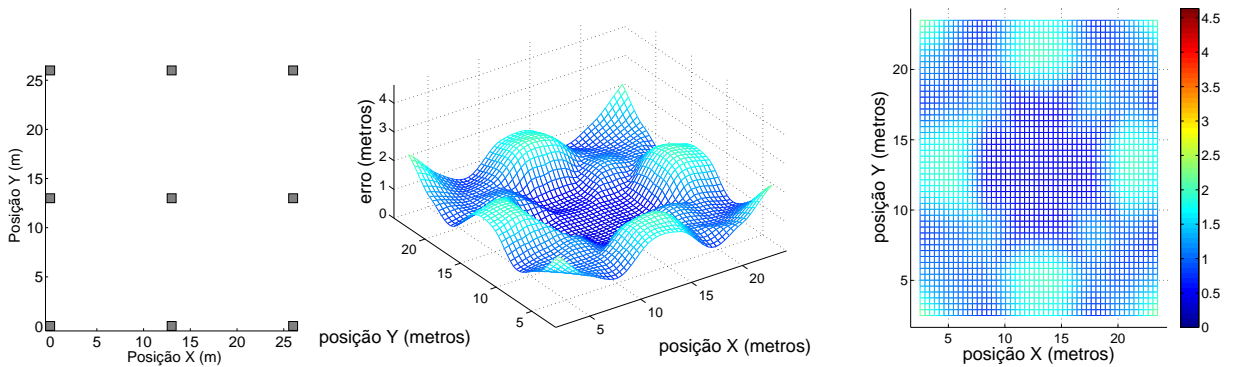
(a) Erro em função da posição usando 6 âncoras.



(b) Erro em função da posição usando 7 âncoras.

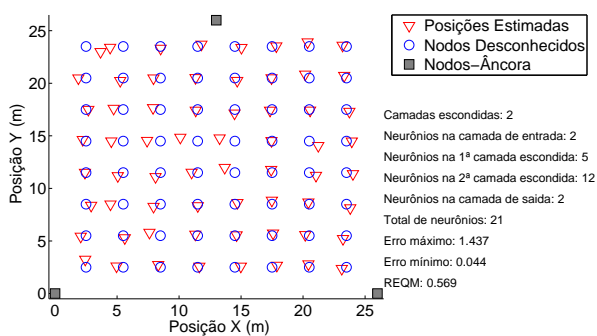


(c) Erro em função da posição usando 8 âncoras.

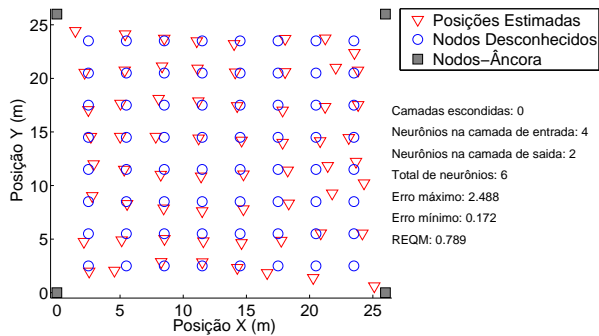


(d) Erro em função da posição usando 9 âncoras.

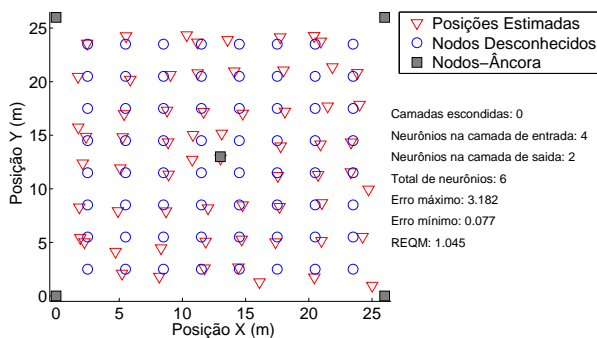
Figura 49: Influência da quantidade de âncoras no erro de localização usando 36 nodos-treino.



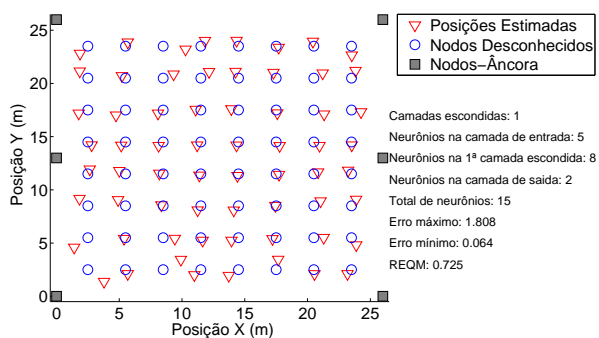
(a) Estimativas de localização usando 3 âncoras.



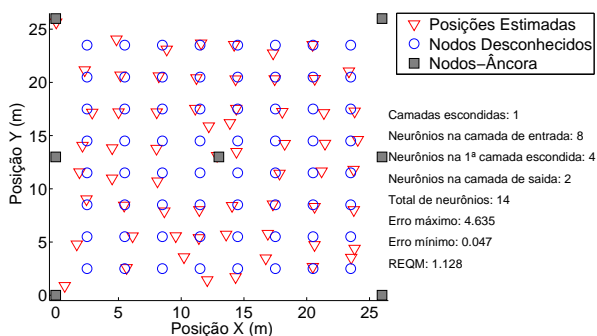
(b) Estimativas de localização usando 4 âncoras.



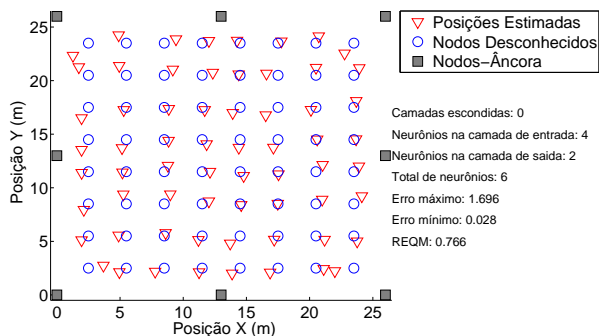
(c) Estimativas de localização usando 5 âncoras.



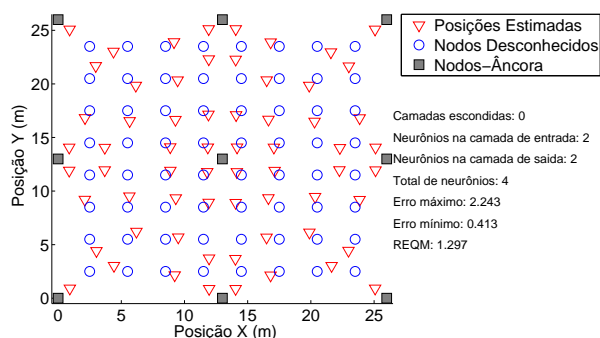
(d) Estimativas de localização usando 6 âncoras.



(e) Estimativas de localização usando 7 âncoras.

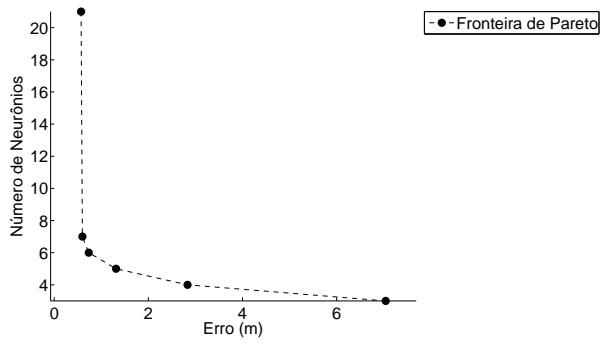


(f) Estimativas de localização usando 8 âncoras.

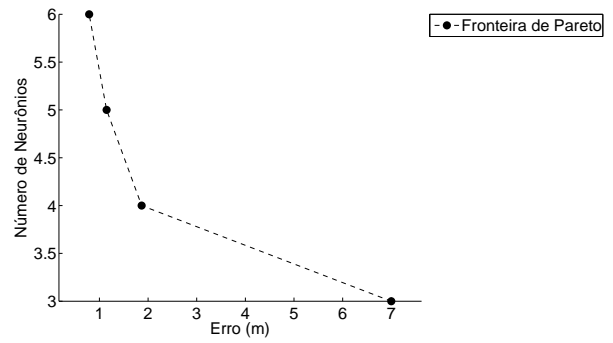


(g) Estimativas de localização usando 9 âncoras.

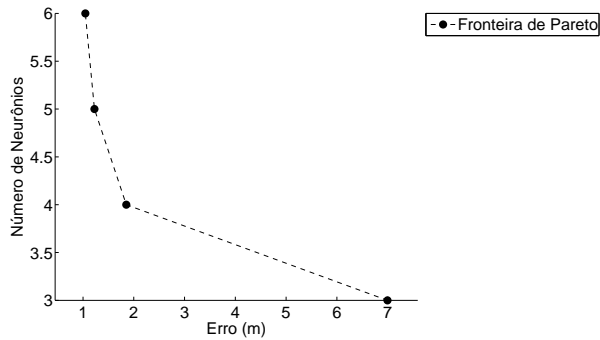
Figura 50: Influência da quantidade de âncoras nas estimativas de localização usando 36 nodos-treino.



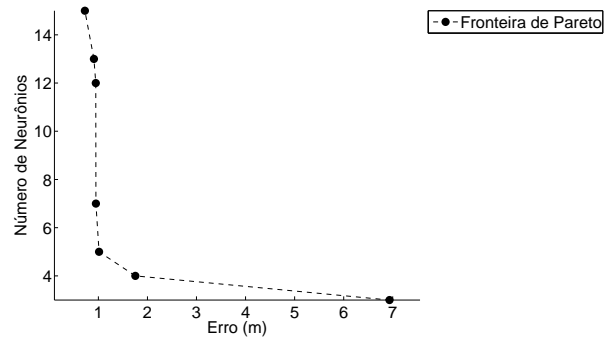
(a) Fronteira de Pareto usando 3 âncoras.



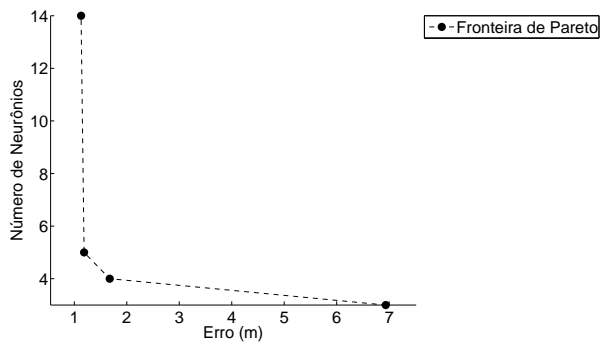
(b) Fronteira de Pareto usando 4 âncoras.



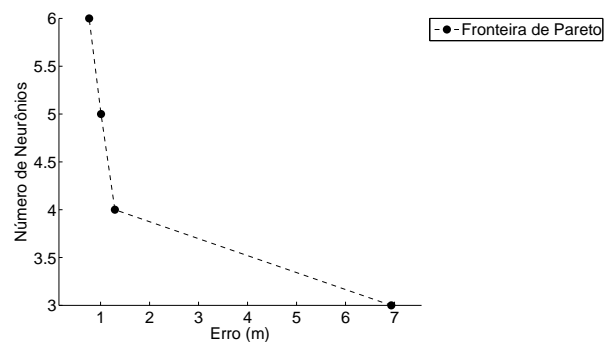
(c) Fronteira de Pareto usando 5 âncoras.



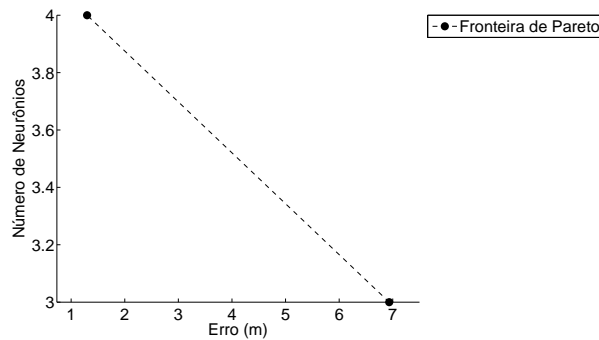
(d) Fronteira de Pareto usando 6 âncoras.



(e) Fronteira de Pareto usando 7 âncoras.



(f) Fronteira de Pareto usando 8 âncoras.



(g) Fronteira de Pareto usando 9 âncoras.

Figura 51: Influência da quantidade de âncoras nas fronteiras de Pareto usando 36 nodos-treino.

4.3.1.4 Amostra composta por 49 Nodos-Treino

Como pode ser visto na figura 54, quando comparadas ao caso anterior (36 nodos-treino), as estimativas foram melhores com o conjunto de amostras aumentado para 49 nodos-treino. Desta vez, em 100% dos casos com um nodo-âncora no centro da área de teste, houve incidência de camadas escondidas nas RNAs. Analisando-se as fronteiras de Pareto para cada caso (figura 55), é possível notar que obteve-se um erro de aproximadamente 1 metro com RNAs de aproximadamente 5 neurônios artificiais.

As figuras 52 e 53 indicam, novamente, uma tendência a erros maiores nas proximidades dos nodos-âncoras, porém com erro menor próximo ao nodo-âncora central. Os casos com 8 e 9 âncoras obtiveram os piores desempenhos.

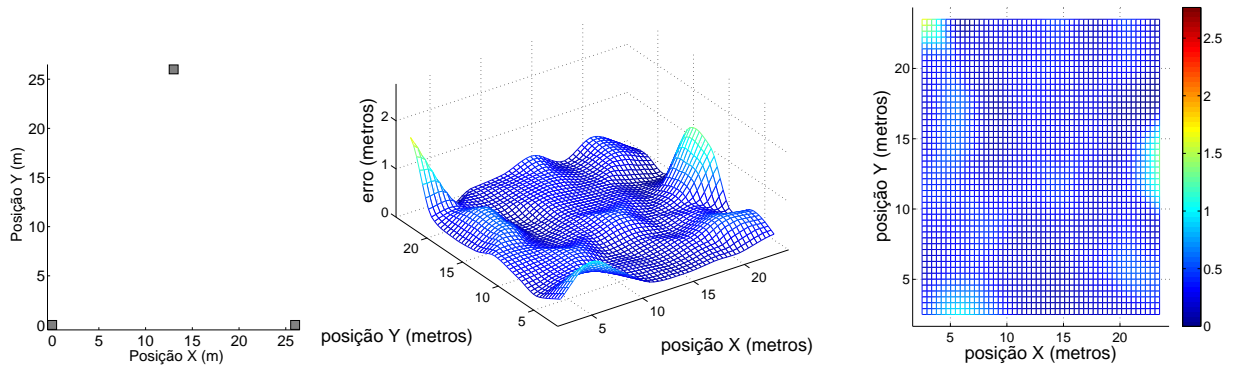
Nas tabelas 14 e 15 é possível analisar os resultados obtidos em termos de REQM e complexidade das RNAs para os casos de menor REQM obtido, ordenados por REQM e nodos-âncora.

Tabela 14: Resultados para amostra de 49 nodos de treinamento variando-se a quantidade de âncoras - ordenação por REQM

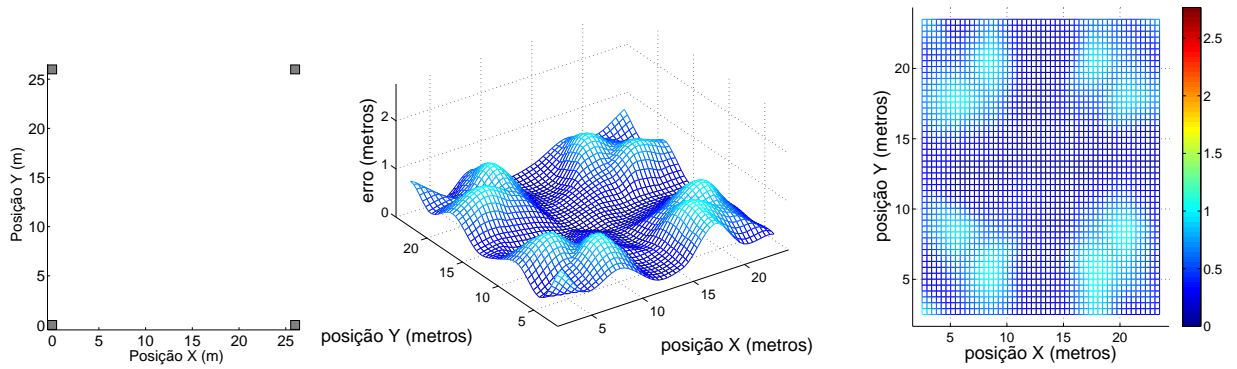
REQM (metros)	Nodos Âncora	Camadas Escondidas	Total de Neurônios	Distância Máxima (metros)
0.443	3	1	13	1.679
0.498	5	1	10	1.477
0.549	4	1	28	1.069
0.665	7	1	21	2.766
0.674	6	0	6	1.966
0.933	8	1	8	1.653
1.014	9	1	7	2.011

Tabela 15: Resultados para amostra de 49 nodos de treinamento variando-se a quantidade de âncoras - ordenação por âncoras

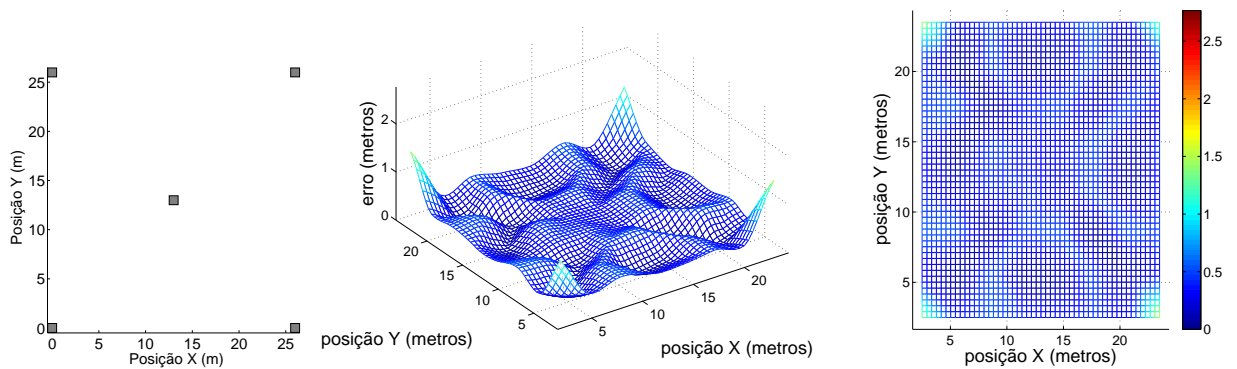
Nodos Âncora	REQM (metros)	Total de Neurônios	Camadas Escondidas	Distância Máxima (metros)
3	0.443	13	1	1.679
4	0.549	28	1	1.069
5	0.498	10	1	1.477
6	0.674	6	0	1.966
7	0.665	21	1	2.766
8	0.933	8	1	1.653
9	1.014	7	1	2.011



(a) Erro em função da posição usando 3 âncoras.

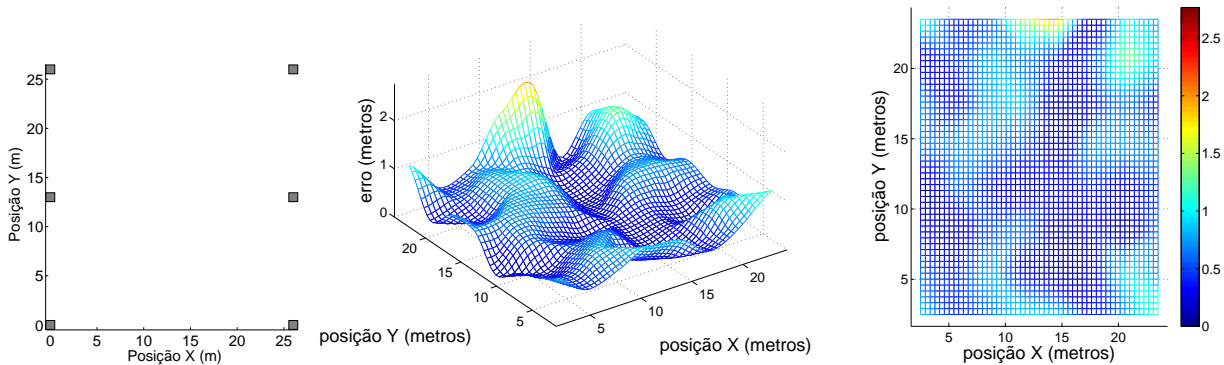


(b) Erro em função da posição usando 4 âncoras.

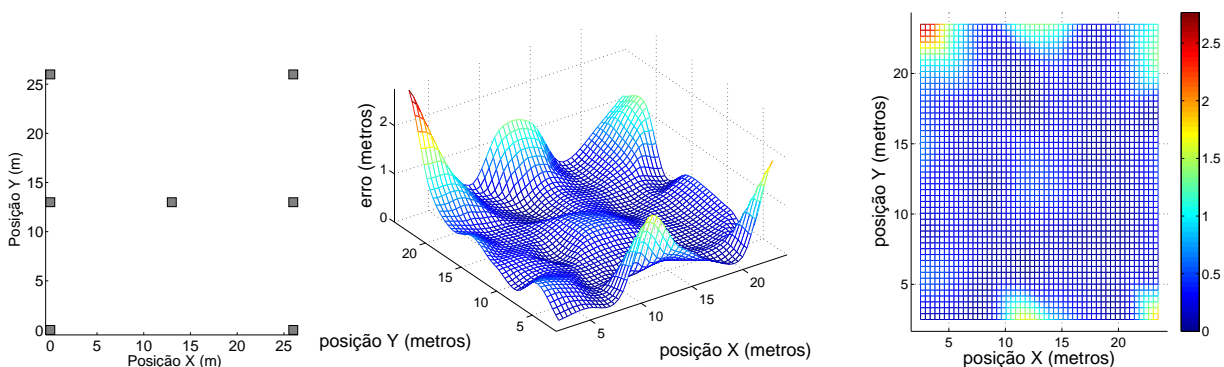


(c) Erro em função da posição usando 5 âncoras.

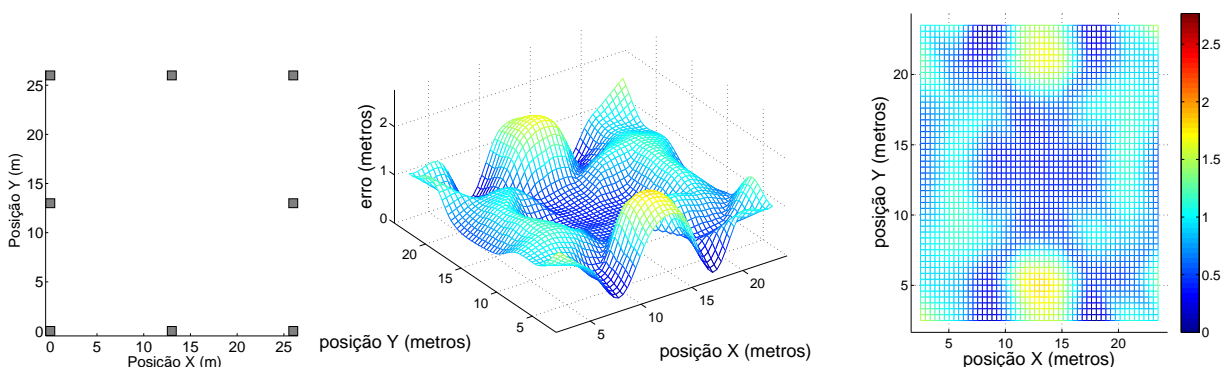
Figura 52: Influência da quantidade de âncoras no erro de localização usando 49 nodos-treino.



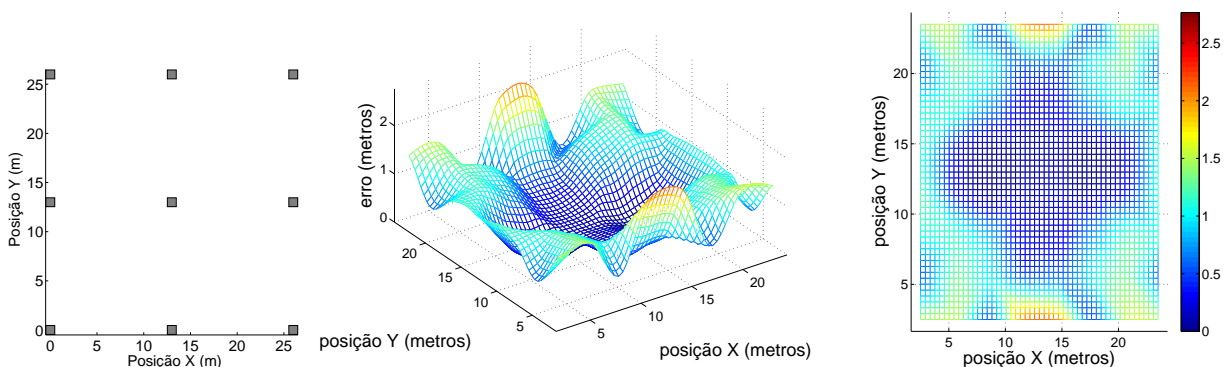
(a) Erro em função da posição usando 6 âncoras.



(b) Erro em função da posição usando 7 âncoras.

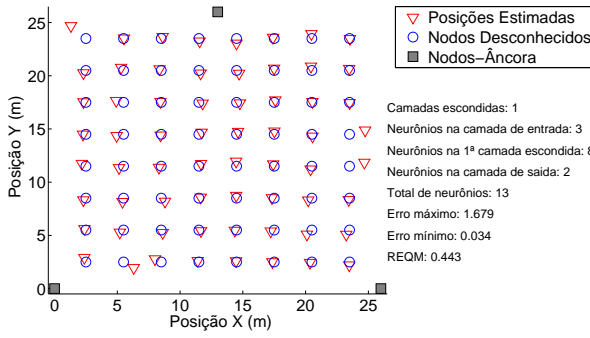


(c) Erro em função da posição usando 8 âncoras.

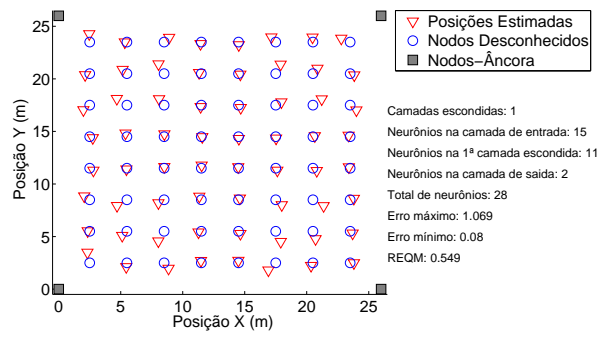


(d) Erro em função da posição usando 9 âncoras.

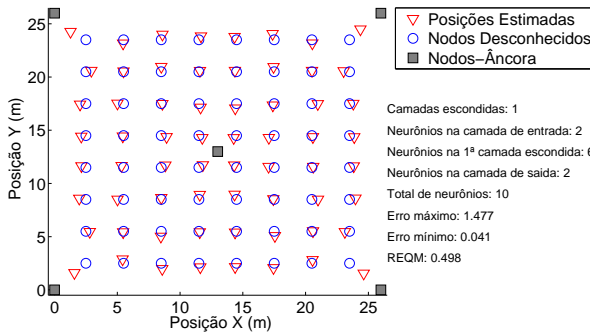
Figura 53: Influência da quantidade de âncoras no erro de localização usando 49 nodos-treino.



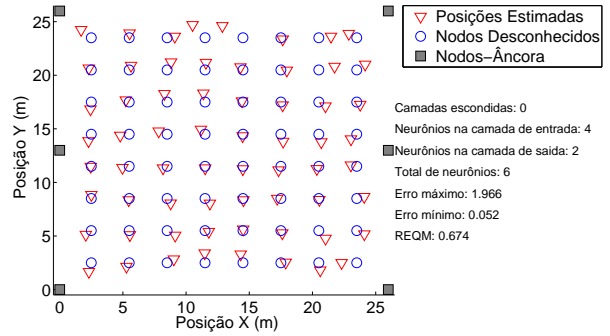
(a) Estimativas de localização usando 3 âncoras.



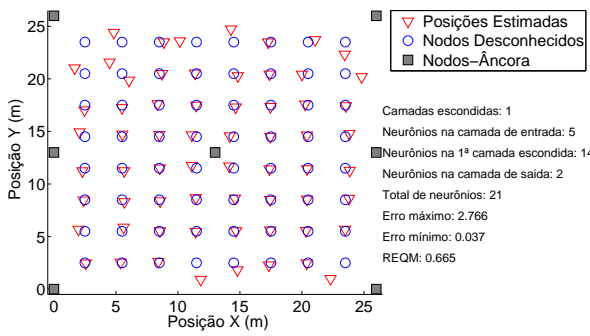
(b) Estimativas de localização usando 4 âncoras.



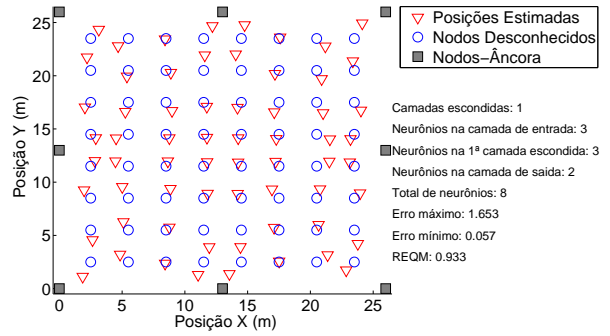
(c) Estimativas de localização usando 5 âncoras.



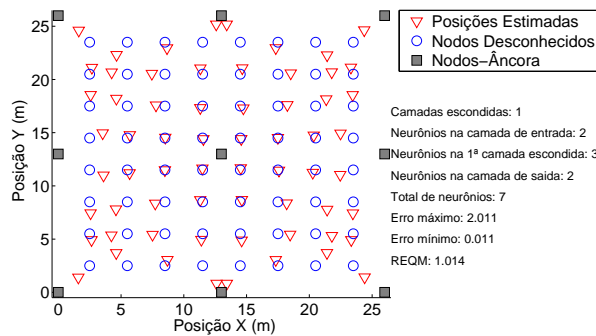
(d) Estimativas de localização usando 6 âncoras.



(e) Estimativas de localização usando 7 âncoras.

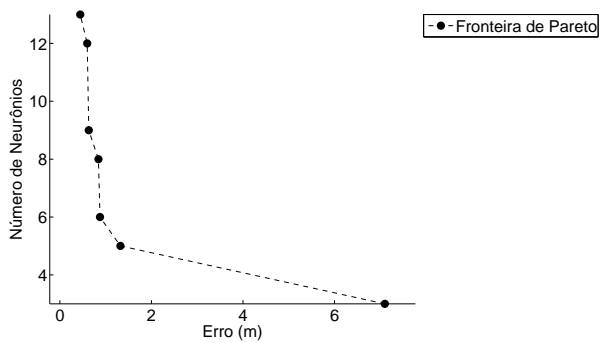


(f) Estimativas de localização usando 8 âncoras.

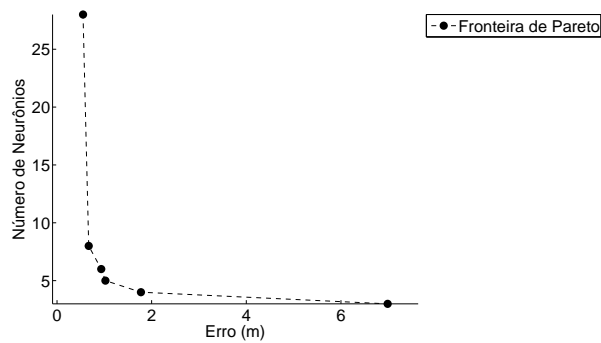


(g) Estimativas de localização usando 9 âncoras.

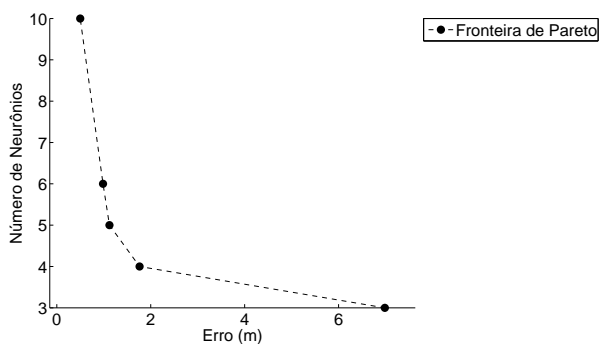
Figura 54: Influência da quantidade de âncoras nas estimativas de localização usando 49 nodos-treino.



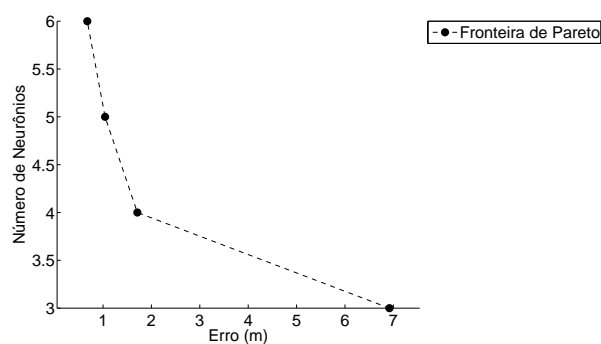
(a) Fronteira de Pareto usando 3 âncoras.



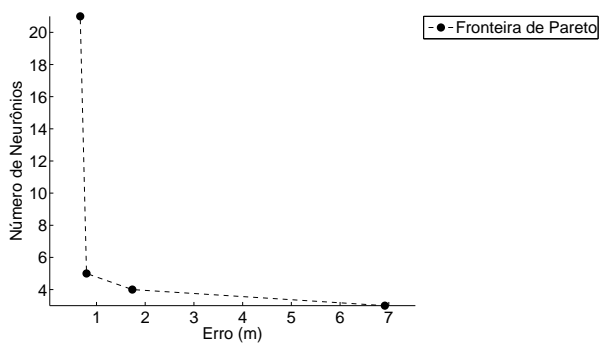
(b) Fronteira de Pareto usando 4 âncoras.



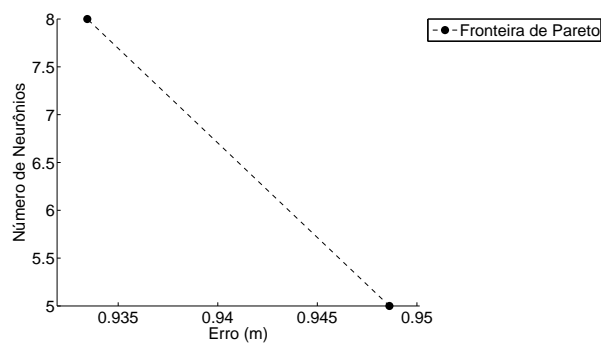
(c) Fronteira de Pareto usando 5 âncoras.



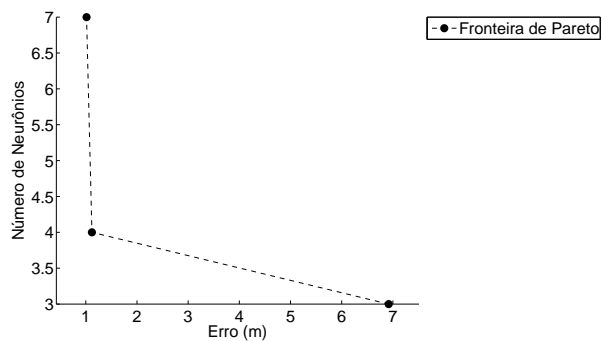
(d) Fronteira de Pareto usando 6 âncoras.



(e) Fronteira de Pareto usando 7 âncoras.



(f) Fronteira de Pareto usando 8 âncoras.



(g) Fronteira de Pareto usando 9 âncoras.

Figura 55: Influência da quantidade de âncoras nas fronteiras de Pareto usando 49 nodos-treino.

4.3.1.5 Amostra composta por 64 Nodos-Treino

Como pode ser visto na figura 58, quando comparadas ao caso anterior (49 nodos-treino), as estimativas foram melhores com o conjunto de amostras aumentado para 64 nodos-treino. Desta vez, em 66,66% dos casos com um nodo-âncora no centro da área de teste, não houve incidência de camadas escondidas nas RNAs. Analisando-se as fronteiras de Pareto para cada caso (figura 59), é possível notar que obteve-se um erro de aproximadamente 1 metro (em alguns casos menor) com RNAs de aproximadamente 5 neurônios artificiais, sendo que o caso com 3 nodos-âncora apresentou mais soluções alternativas.

As figuras 56 e 57 indicam, novamente, uma tendência a erros maiores nas proximidades dos nodos-âncoras, porém com erro menor próximo ao nodo-âncora central.

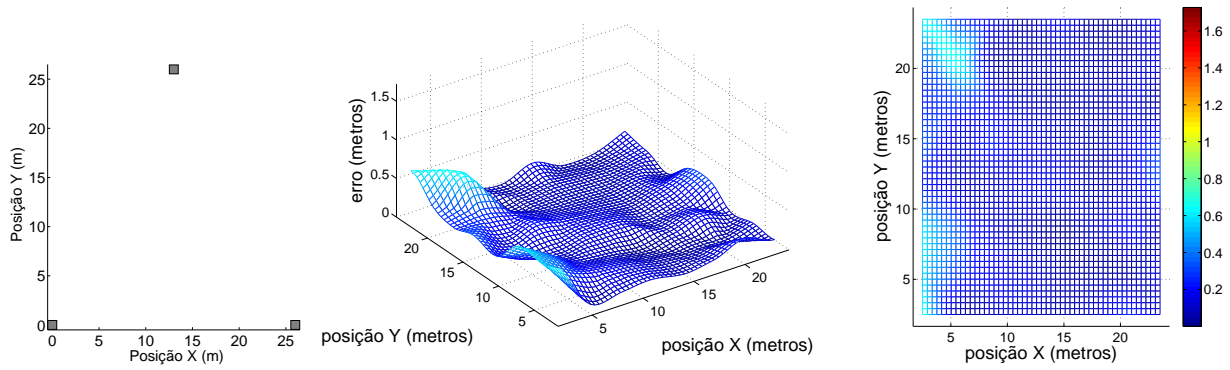
Nas tabelas 16 e 17 é possível analisar os resultados obtidos em termos de REQM e complexidade das RNAs para os casos de menor REQM obtido, ordenados por REQM e nodos-âncora.

Tabela 16: Resultados para amostra de 64 nodos de treinamento variando-se a quantidade de âncoras - ordenação por REQM

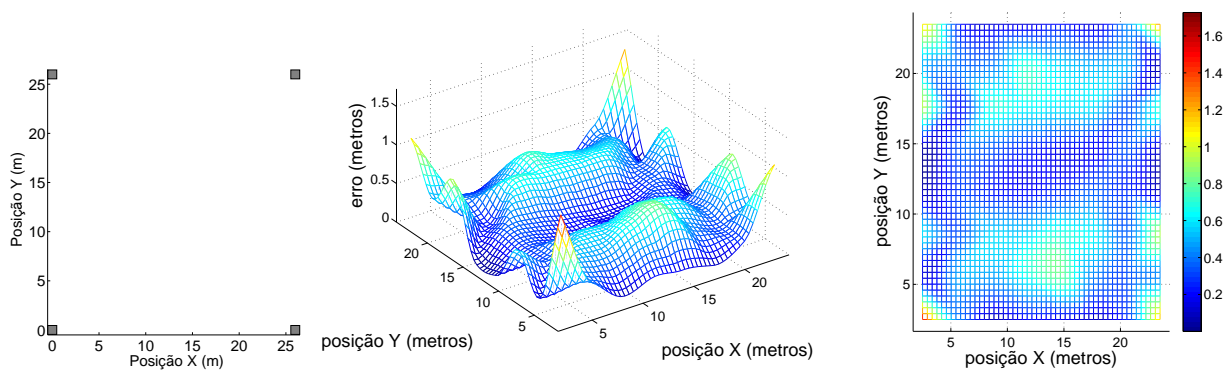
REQM (metros)	Nodos Âncora	Camadas Escondidas	Total de Neurônios	Distância Máxima (metros)
0.232	3	2	39	0.65
0.361	5	1	11	1.674
0.485	9	0	8	1.727
0.531	4	1	10	1.42
0.553	6	1	9	1.194
0.566	8	0	6	1.264
0.606	7	0	6	1.425

Tabela 17: Resultados para amostra de 64 nodos de treinamento variando-se a quantidade de âncoras - ordenação por âncoras

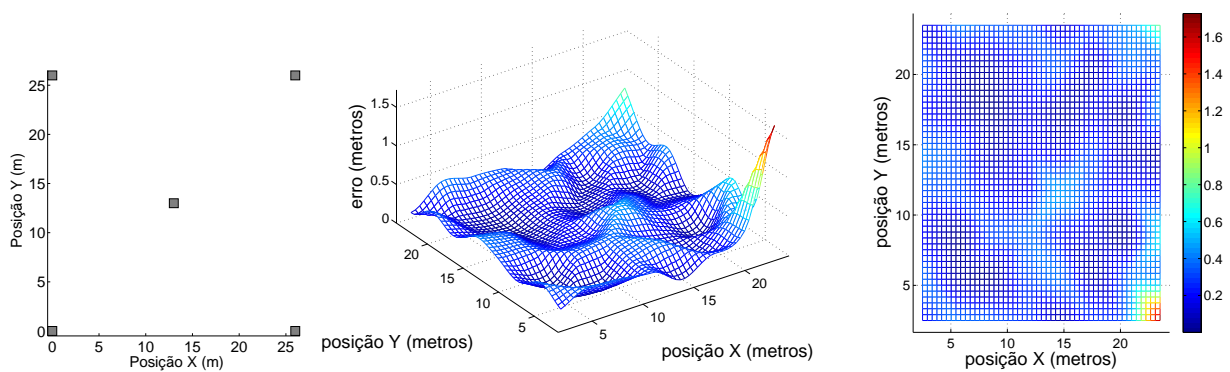
Nodos Âncora	REQM (metros)	Total de Neurônios	Camadas Escondidas	Distância Máxima (metros)
3	0.232	39	2	0.65
4	0.531	10	1	1.42
5	0.361	11	1	1.674
6	0.553	9	1	1.194
7	0.606	6	0	1.425
8	0.566	6	0	1.264
9	0.485	8	0	1.727



(a) Erro em função da posição usando 3 âncoras.

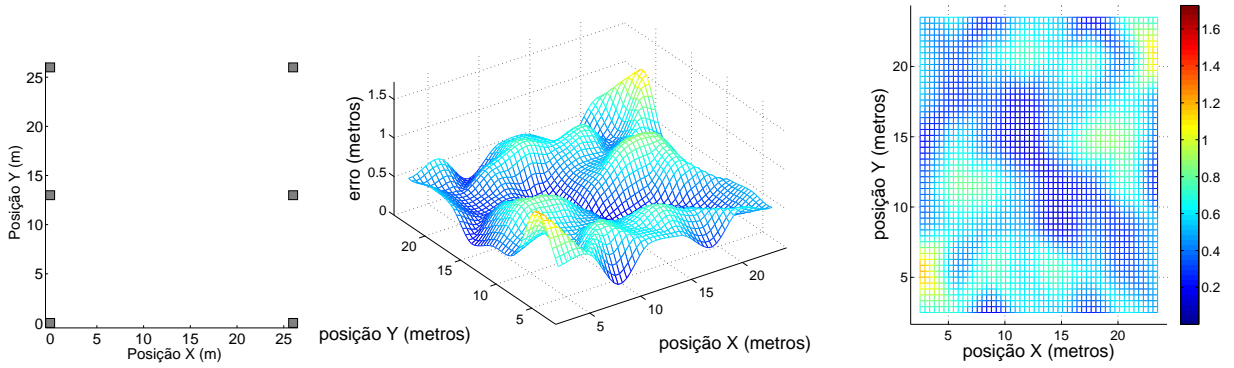


(b) Erro em função da posição usando 4 âncoras.

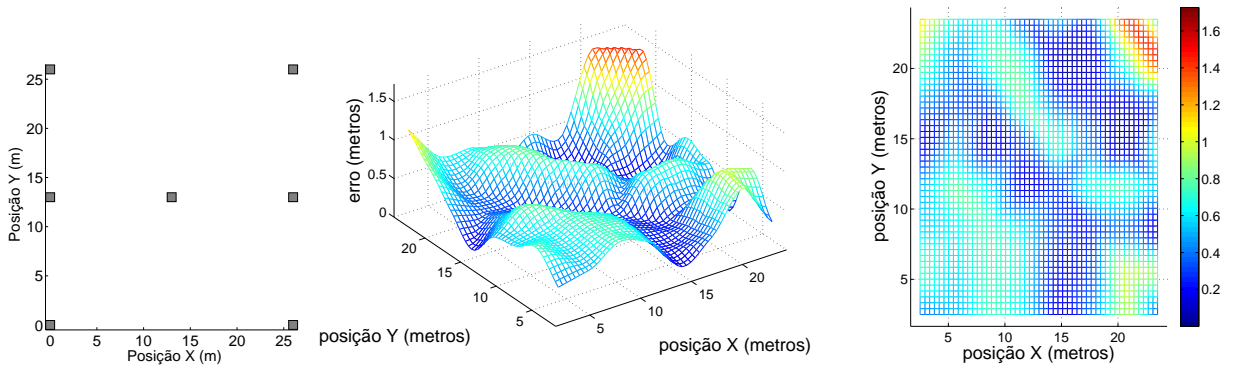


(c) Erro em função da posição usando 5 âncoras.

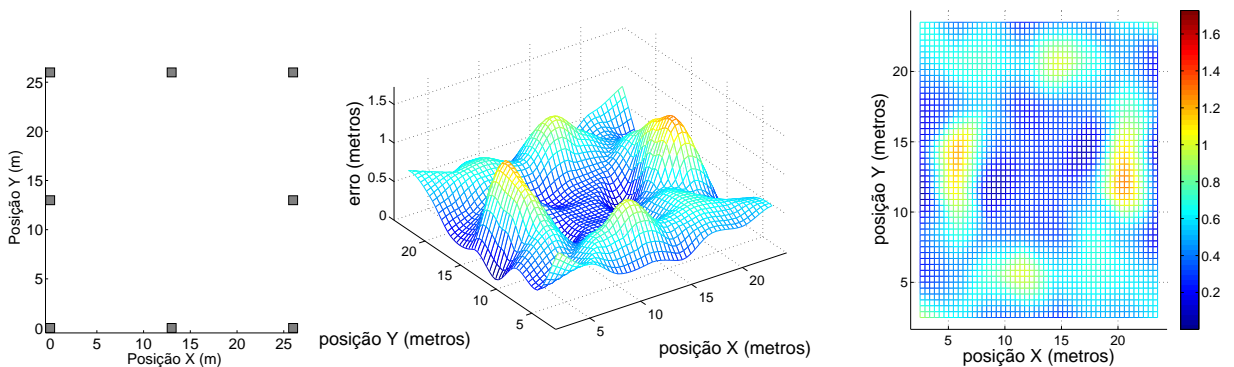
Figura 56: Influência da quantidade de âncoras no erro de localização usando 64 nodos-treino.



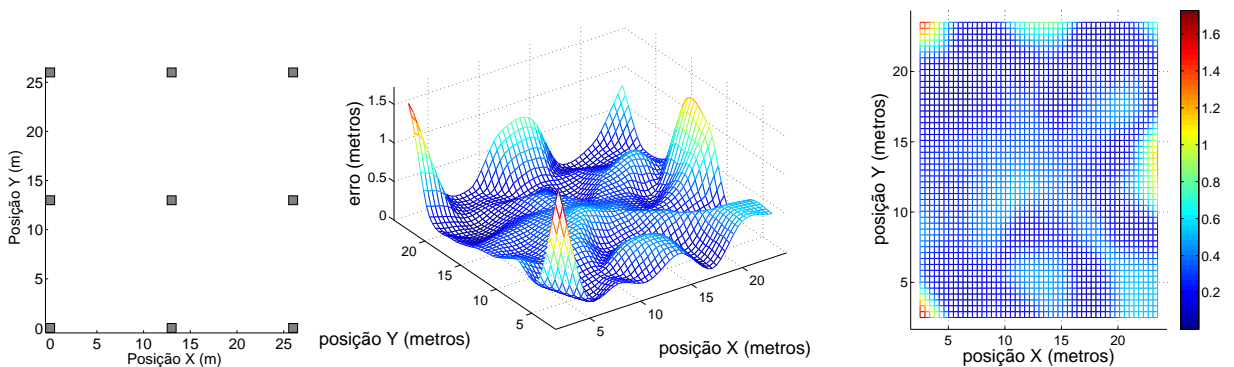
(a) Erro em função da posição usando 6 âncoras.



(b) Erro em função da posição usando 7 âncoras.

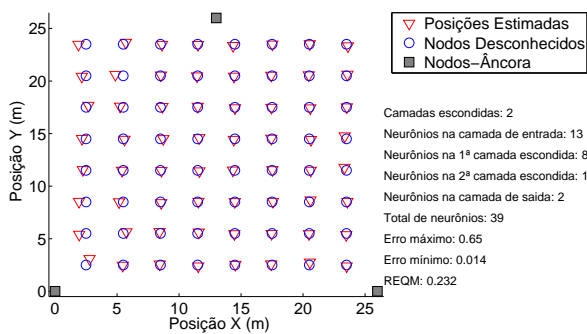


(c) Erro em função da posição usando 8 âncoras.

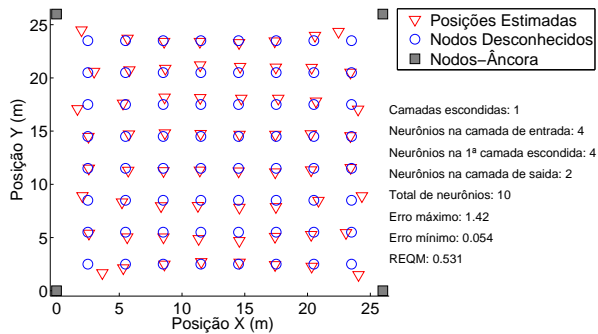


(d) Erro em função da posição usando 9 âncoras.

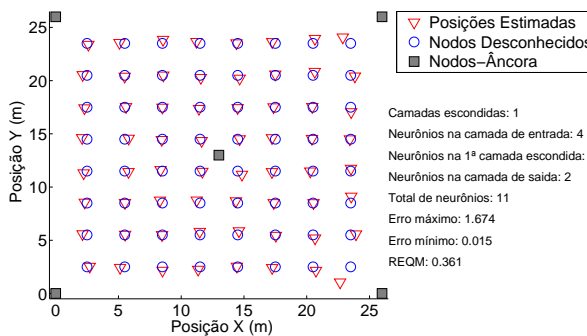
Figura 57: Influência da quantidade de âncoras no erro de localização usando 64 nodos-treino.



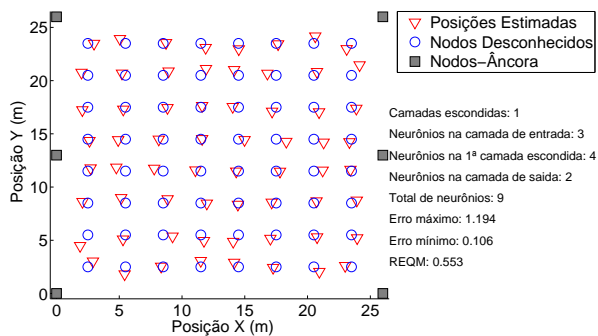
(a) Estimativas de localização usando 3 âncoras.



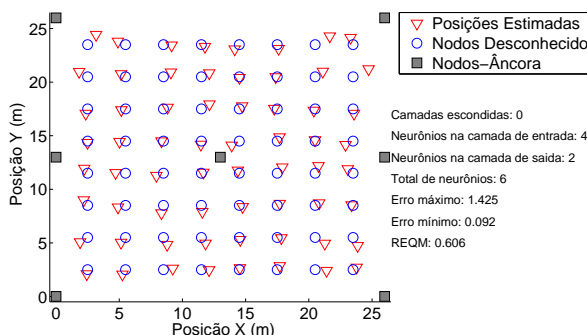
(b) Estimativas de localização usando 4 âncoras.



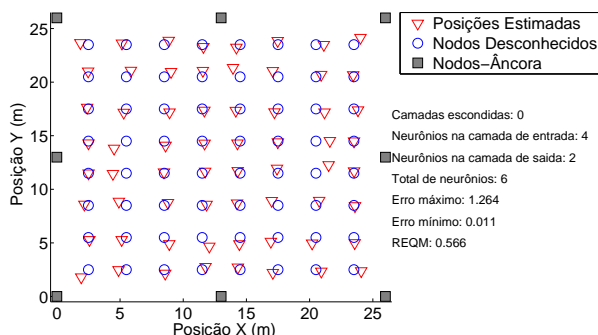
(c) Estimativas de localização usando 5 âncoras.



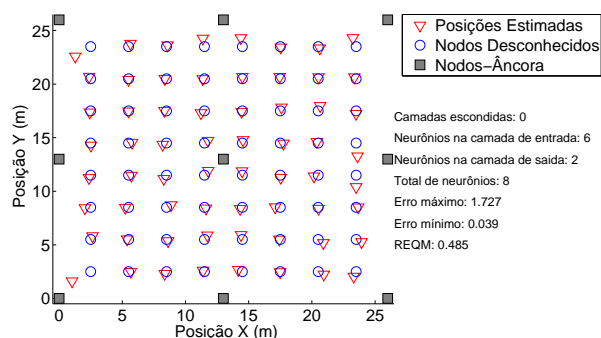
(d) Estimativas de localização usando 6 âncoras.



(e) Estimativas de localização usando 7 âncoras.

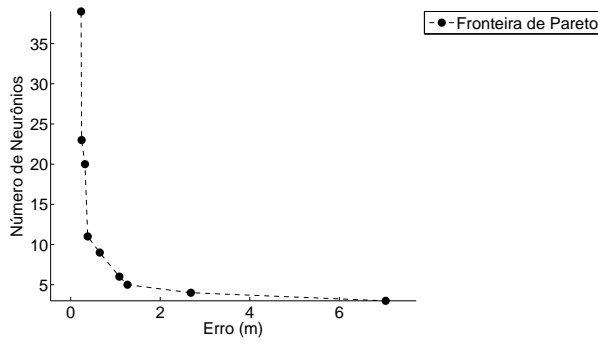


(f) Estimativas de localização usando 8 âncoras.

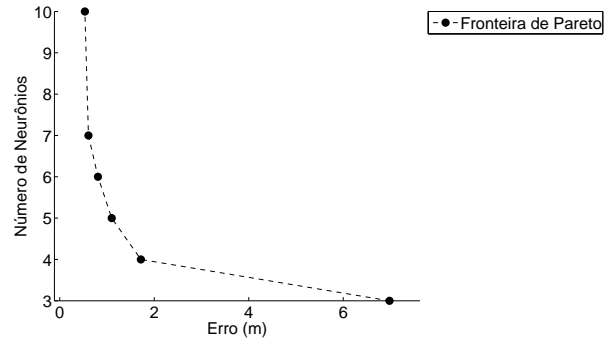


(g) Estimativas de localização usando 9 âncoras.

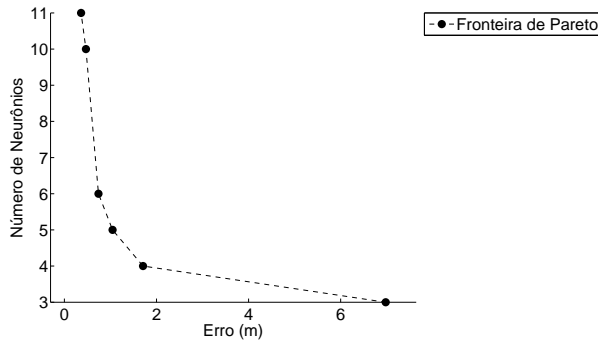
Figura 58: Influência da quantidade de âncoras nas estimativas de localização usando 64 nodos-treino.



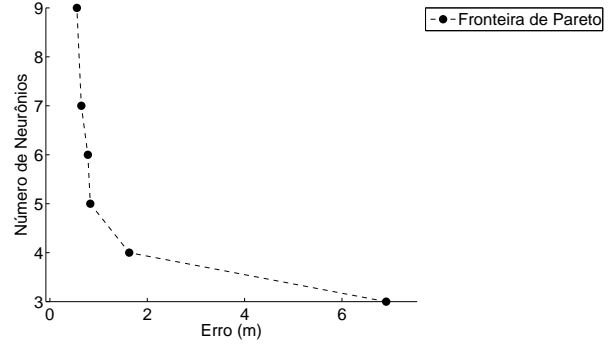
(a) Fronteira de Pareto usando 3 âncoras.



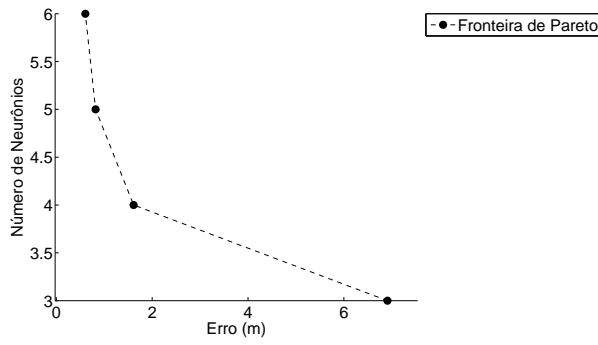
(b) Fronteira de Pareto usando 4 âncoras.



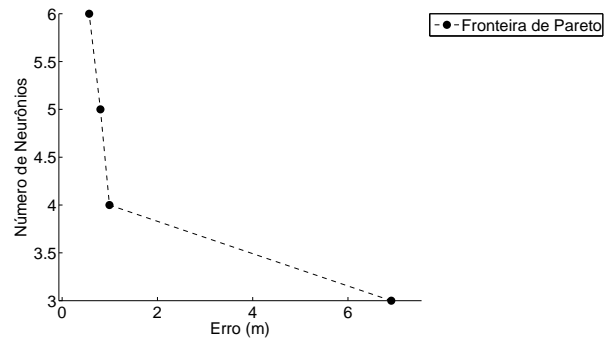
(c) Fronteira de Pareto usando 5 âncoras.



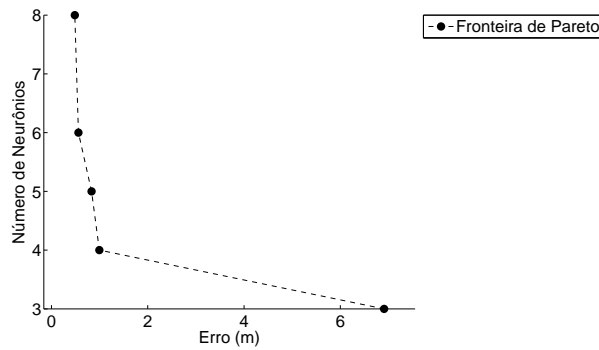
(d) Fronteira de Pareto usando 6 âncoras.



(e) Fronteira de Pareto usando 7 âncoras.



(f) Fronteira de Pareto usando 8 âncoras.



(g) Fronteira de Pareto usando 9 âncoras.

Figura 59: Influência da quantidade de âncoras nas fronteiras de Pareto usando 64 nodos-treino.

4.3.1.6 Amostra composta por 81 Nodos-Treino

Como pode ser visto na figura 62, quando comparadas ao caso anterior (64 nodos-treino), as estimativas foram melhores com o conjunto de amostras aumentado para 81 nodos-treino, a maior amostra testada. Em 66,66% dos casos com um nodo-âncora no centro da área de teste, não houve incidência de camadas escondidas nas RNAs. Analisando-se as fronteiras de Pareto para cada caso (figura 63), é possível notar que obteve-se um erro de aproximadamente 1 metro (em alguns casos menor) com RNAs de aproximadamente 5 neurônios artificiais.

As figuras 60 e 61 indicam, novamente, uma tendência a erros maiores nas proximidades dos nodos-âncoras, porém com erro menor próximo ao nodo-âncora central, sendo os casos com 8 e 9 âncoras os que apresentaram piores desempenhos.

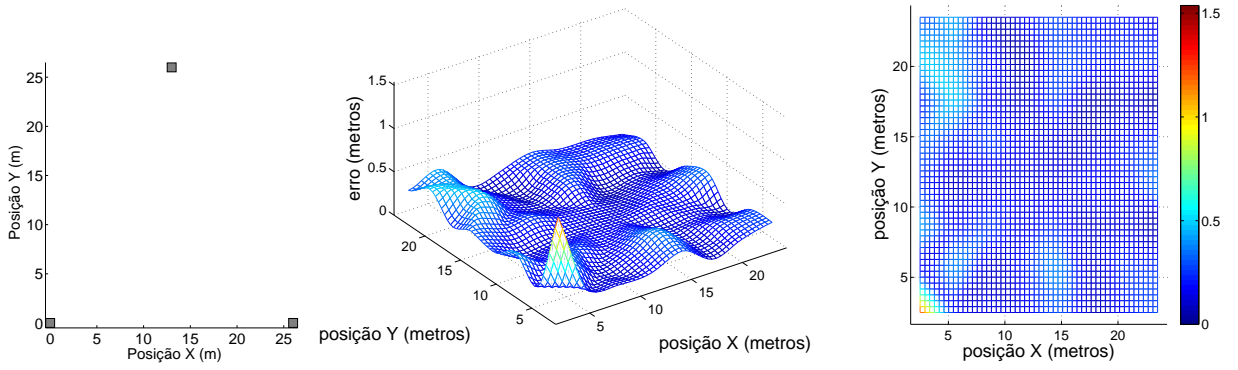
Nas tabelas 18 e 19 é possível analisar os resultados obtidos em termos de REQM e complexidade das RNAs para os casos de menor REQM obtido, ordenados por REQM e nodos-âncora.

Tabela 18: Resultados para amostra de 81 nodos de treinamento variando-se a quantidade de âncoras - ordenação por REQM

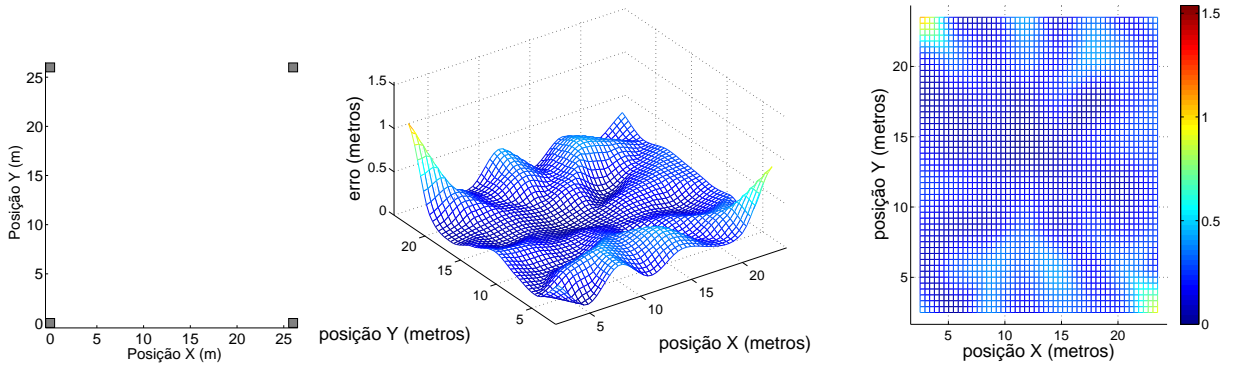
REQM (metros)	Nodos Âncora	Camadas Escondidas	Total de Neurônios	Distância Máxima (metros)
0.263	5	1	16	0.694
0.272	3	1	12	1.157
0.288	4	2	28	1.07
0.334	6	2	24	0.993
0.376	7	0	7	0.906
0.435	9	0	7	1.538
0.456	8	1	21	1.394

Tabela 19: Resultados para amostra de 81 nodos de treinamento variando-se a quantidade de âncoras - ordenação por âncoras

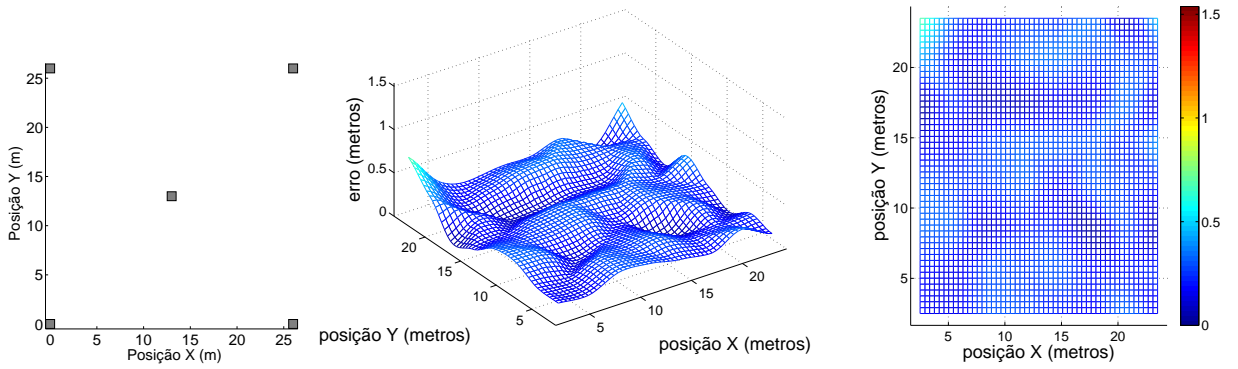
Nodos Âncora	REQM (metros)	Total de Neurônios	Camadas Escondidas	Distância Máxima (metros)
3	0.272	12	1	1.157
4	0.288	28	2	1.07
5	0.263	16	1	0.694
6	0.334	24	2	0.993
7	0.376	7	0	0.906
8	0.456	21	1	1.394
9	0.435	7	0	1.538



(a) Erro em função da posição usando 3 âncoras.

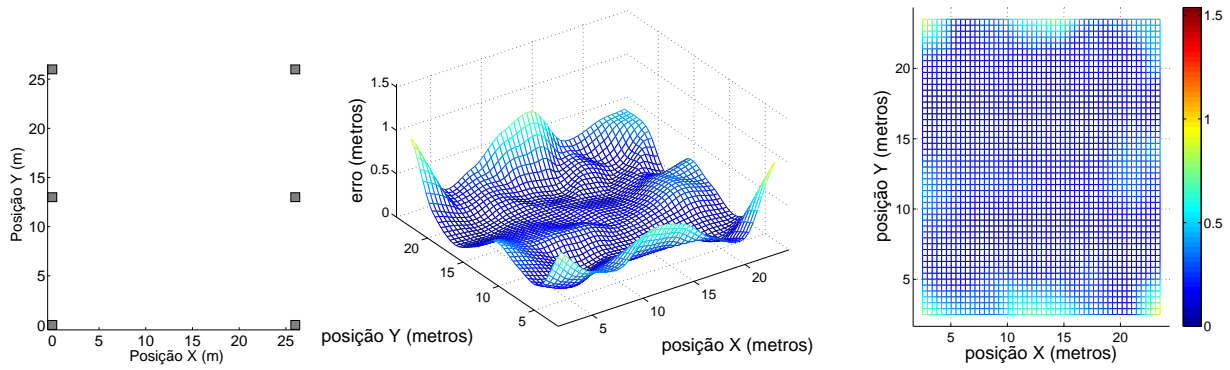


(b) Erro em função da posição usando 4 âncoras.

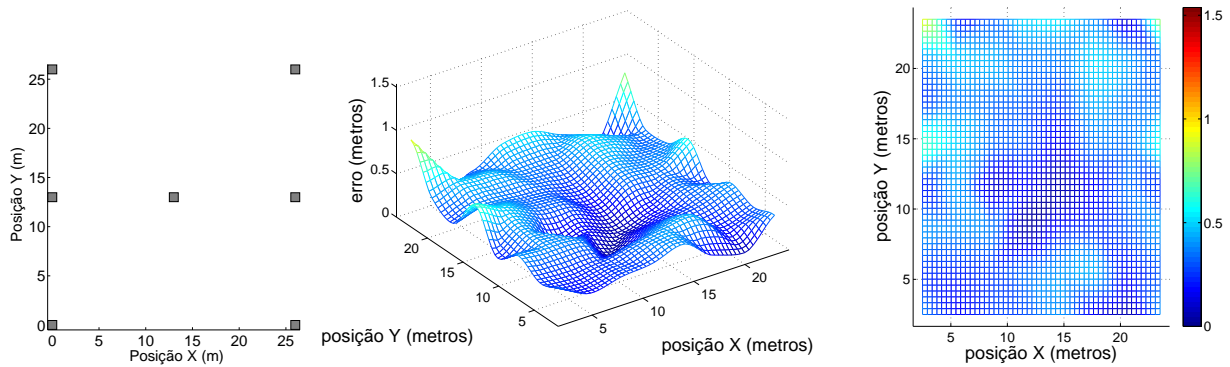


(c) Erro em função da posição usando 5 âncoras.

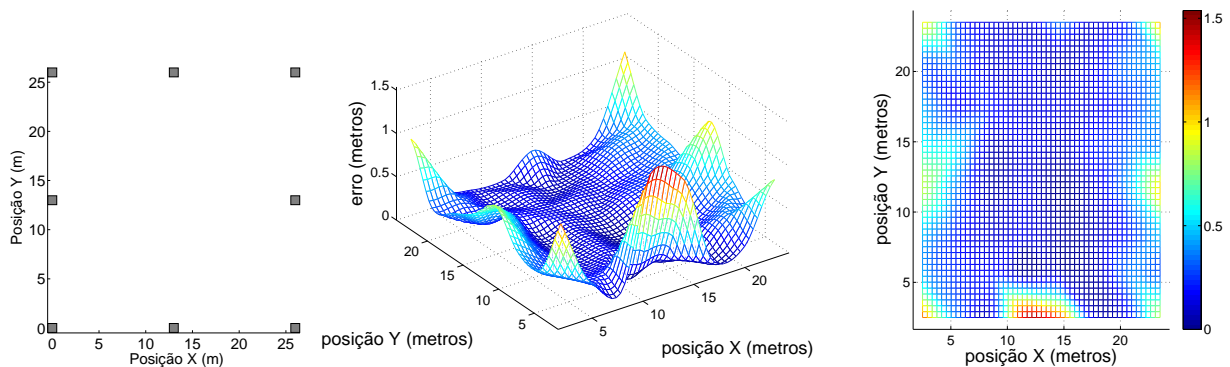
Figura 60: Influência da quantidade de âncoras no erro de localização usando 81 nodos-treino.



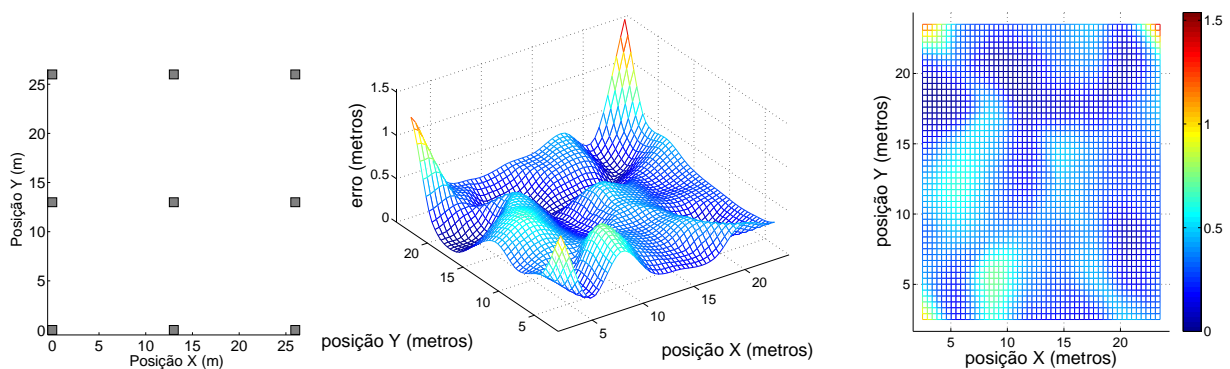
(a) Erro em função da posição usando 6 âncoras.



(b) Erro em função da posição usando 7 âncoras.

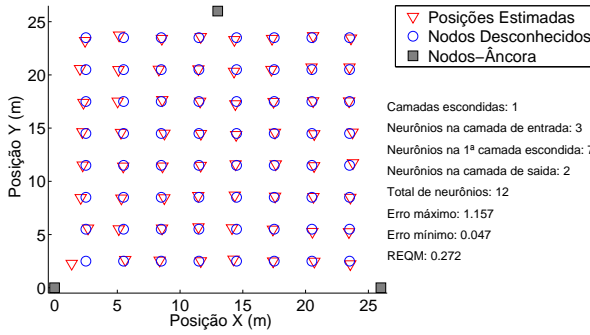


(c) Erro em função da posição usando 8 âncoras.

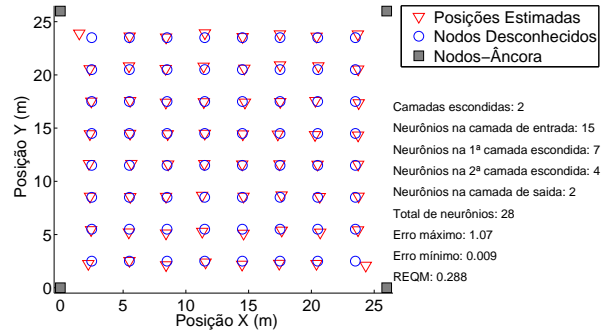


(d) Erro em função da posição usando 9 âncoras.

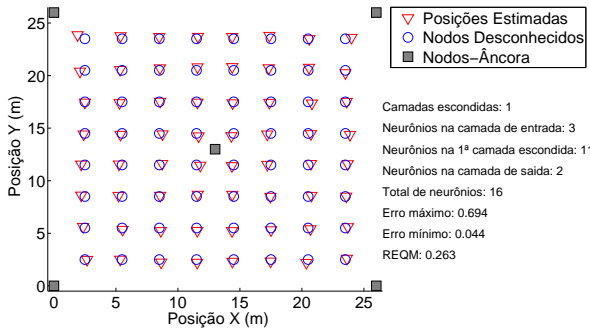
Figura 61: Influência da quantidade de âncoras no erro de localização usando 81 nodos-treino.



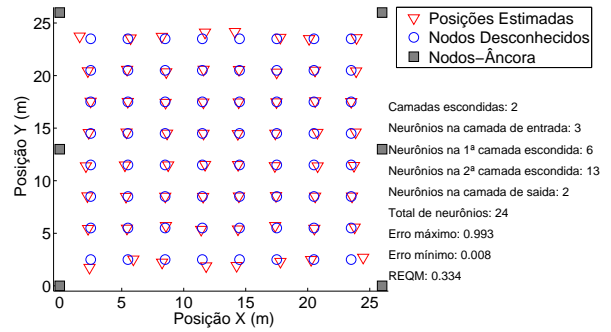
(a) Estimativas de localização usando 3 âncoras.



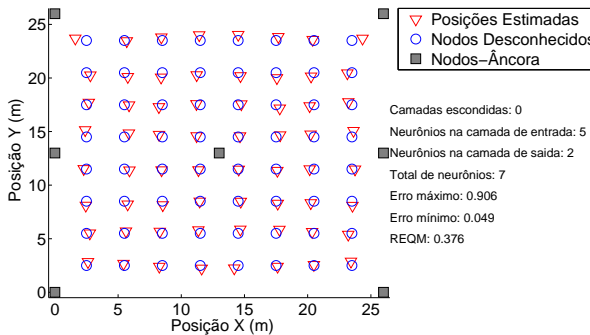
(b) Estimativas de localização usando 4 âncoras.



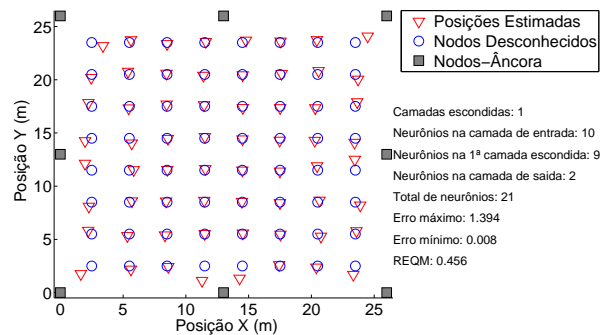
(c) Estimativas de localização usando 5 âncoras.



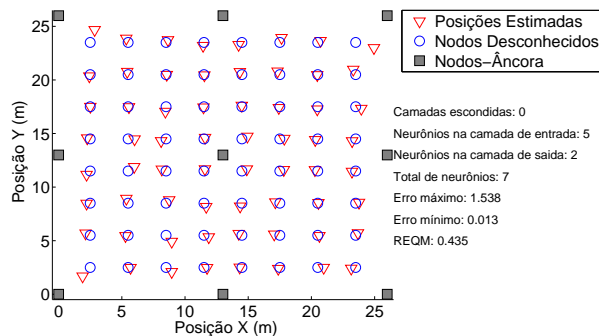
(d) Estimativas de localização usando 6 âncoras.



(e) Estimativas de localização usando 7 âncoras.

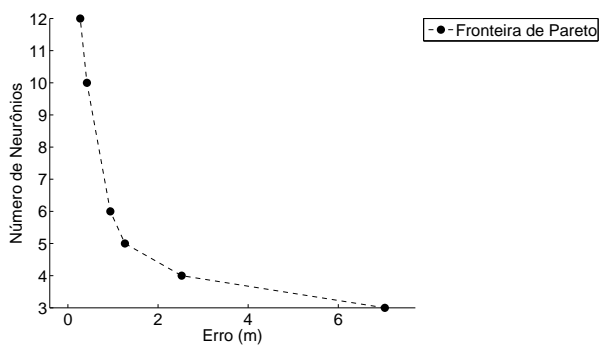


(f) Estimativas de localização usando 8 âncoras.

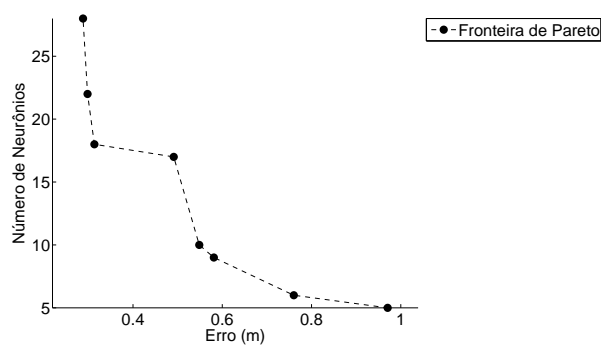


(g) Estimativas de localização usando 9 âncoras.

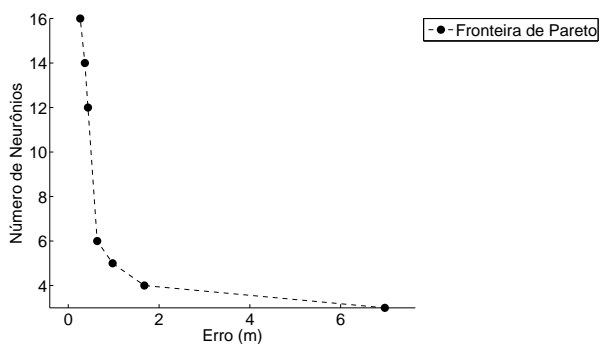
Figura 62: Influência da quantidade de âncoras nas estimativas de localização usando 81 nodos-treino.



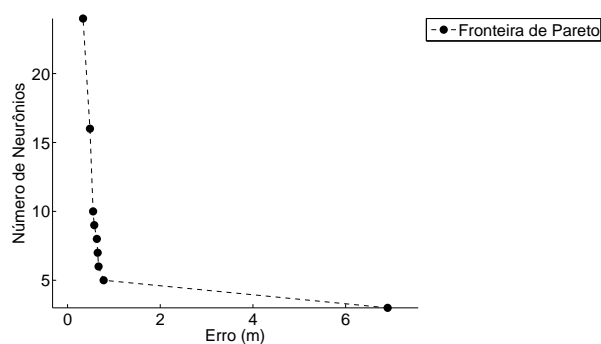
(a) Fronteira de Pareto usando 3 âncoras.



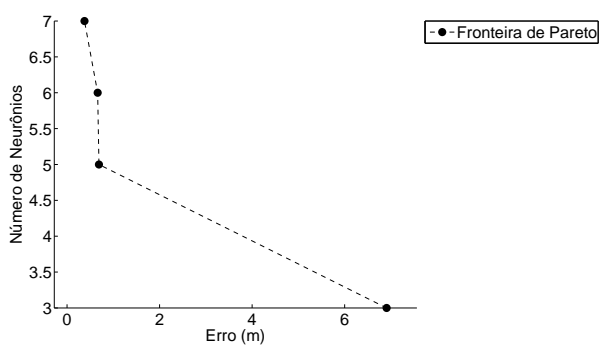
(b) Fronteira de Pareto usando 4 âncoras.



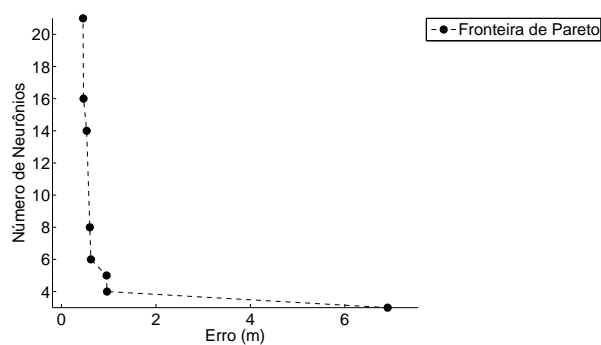
(c) Fronteira de Pareto usando 5 âncoras.



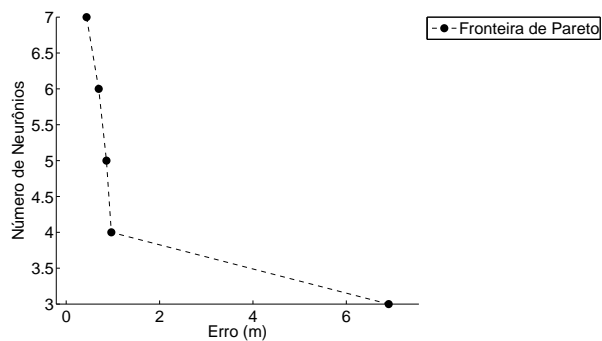
(d) Fronteira de Pareto usando 6 âncoras.



(e) Fronteira de Pareto usando 7 âncoras.



(f) Fronteira de Pareto usando 8 âncoras.



(g) Fronteira de Pareto usando 9 âncoras.

Figura 63: Influência da quantidade de âncoras nas fronteiras de Pareto usando 81 nodos-treino.

4.3.1.7 Análise do REQM em função do número de âncoras

Após a análise de todos os tamanhos de amostras simulados, a influência da quantidade de nodos-âncora na precisão das estimativas de localização pode ser vista de forma clara nos resultados das simulações principais. Ao contrário do comportamento apresentado pela maioria dos métodos de localização que não utilizam RNAs, uma quantidade maior de nodos-âncora não garante melhores estimativas de localização, piorando-as na maioria dos casos (figura 64). O sistema proposto em (BULUSU; HEIDEMANN; ESTRIN, 2000), por exemplo, apresenta aumento de desempenho quando é aumentada a quantidade de nodos-âncora na RSSF.

Este comportamento é compartilhado por diversas topologias simuladas com tamanhos de amostra de treinamento diferentes. Outro fato que merece atenção é que, as RSSFs simuladas com âncoras no centro da área de teste, possuem uma forte tendência a eliminar a necessidade de camadas intermediárias nas RNAs.

Deve-se levar em consideração que quanto menos camadas uma RNA possuir, provavelmente maior será a sua velocidade de processamento e, conseqüentemente, menor será o seu consumo de energia para entregar uma estimativa de localização. Conforme visto nas fronteiras de Pareto, para os casos de RNAs com 5 neurônios, não há presença de camadas intermediárias e ainda assim boas estimativas de localização.

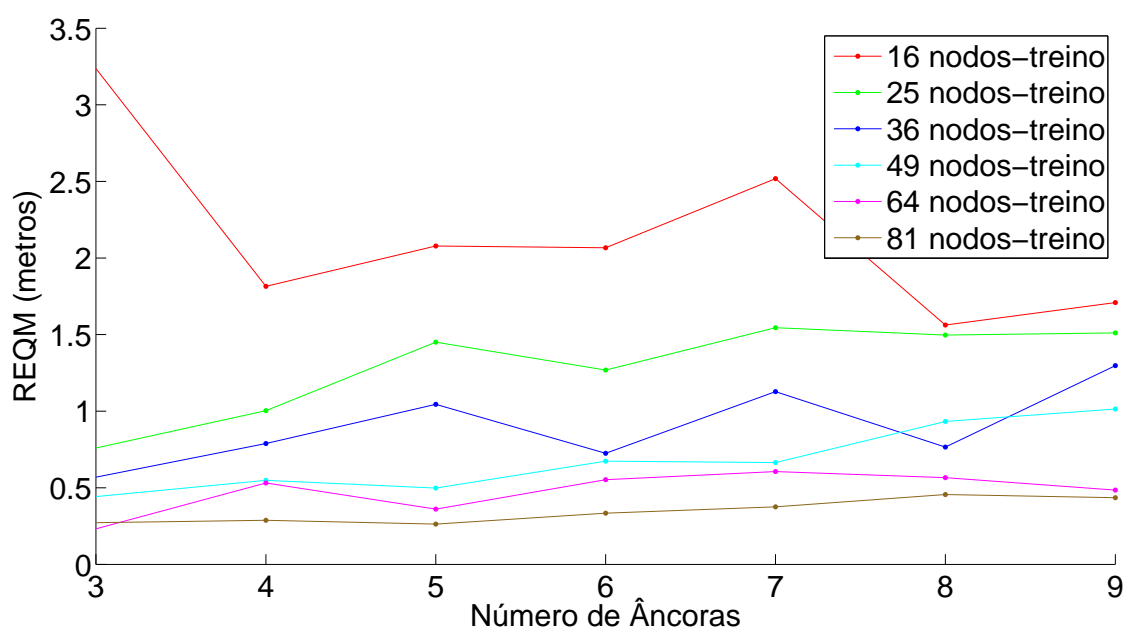


Figura 64: Comportamento da REQM em função do aumento de nodos-âncora pra todas as amostras simuladas .

4.3.2 Variação no Tamanho da Amostra de Treinamento

4.3.2.1 Topologias com 3 Nodos-Âncora

Como pode ser visto na figura 67, mantendo-se fixo o número de nodos-âncora e aumentando-se a quantidade de nodos-treino, o erro é reduzido. Nota-se claramente que a amostra composta por 16 nodos-treino não é suficiente para gerar boas estimativas. Outra constatação é a de que 83% dos casos apresentaram camadas escondidas nas RNAs.

Analisando-se as fronteiras de Pareto para cada caso (figura 68), é possível notar que pode-se reduzir substancialmente o número de neurônios das RNAs aumentando ligeiramente o erro.

As figuras 65 e 66 corroboram o fato de que a REQM é reduzida à medida que se aumenta o tamanho da amostra, sendo que a partir de 64 nodos-treino o erro é quase imperceptível frente ao pior caso (16 nodos-treino).

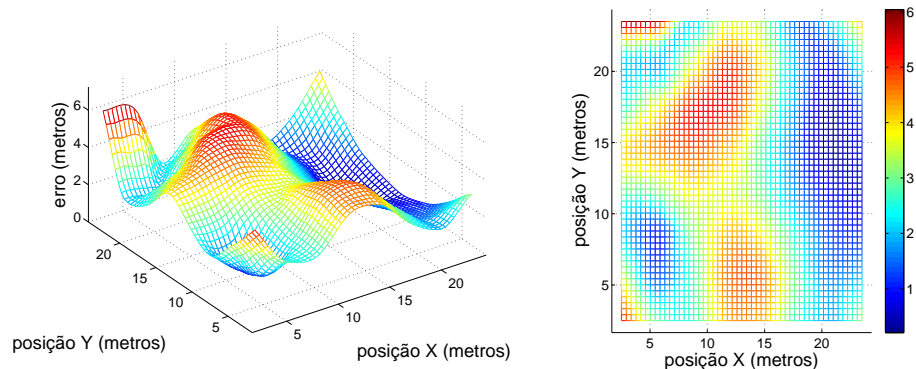
Nas tabelas 20 e 21 é possível analisar os resultados obtidos em termos de REQM e complexidade das RNAs para os casos de menor REQM obtido, ordenados por REQM e tamanho da amostra.

Tabela 20: Resultados para 3 âncoras variando-se o tamanho da amostra de treinamento - ordenação por REQM

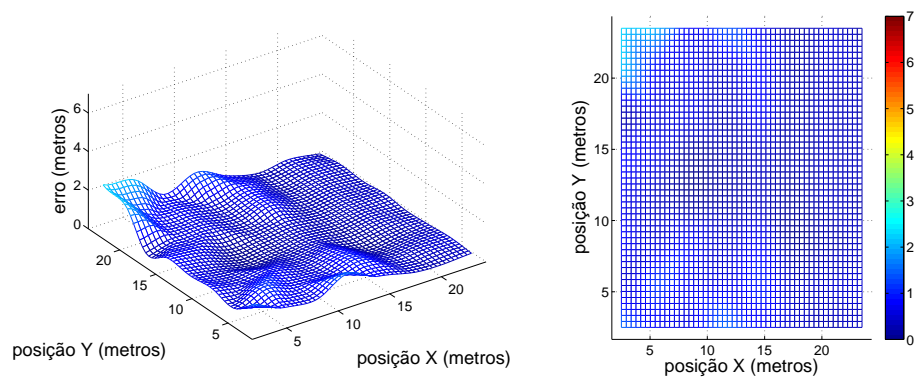
REQM (metros)	Tamanho da Amostra	Camadas Escondidas	Total de Neurônios	Distância Máxima (metros)
0.232	64	2	39	0.65
0.272	81	1	12	1.157
0.443	49	1	13	1.679
0.569	36	2	21	1.437
0.76	25	1	15	2.34
3.235	16	0	4	6.033

Tabela 21: Resultados para 3 âncoras variando-se o tamanho da amostra de treinamento - ordenação por âncoras

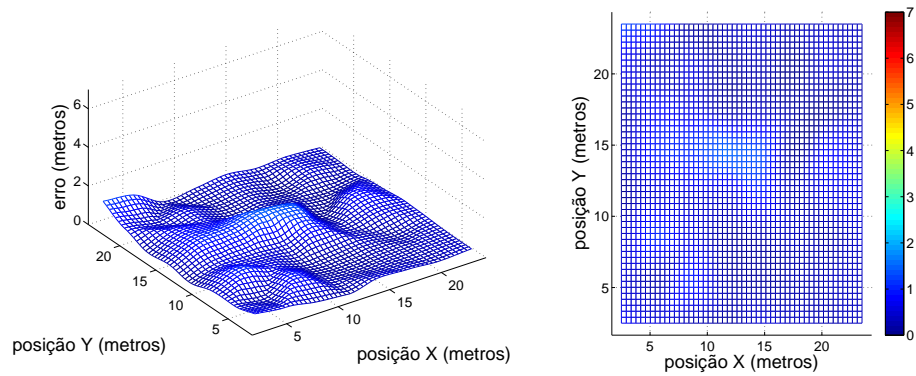
Tamanho da Amostra	REQM (metros)	Total de Neurônios	Camadas Escondidas	Distância Máxima (metros)
16	3.235	4	0	6.033
25	0.76	15	1	2.34
36	0.569	21	2	1.437
49	0.443	13	1	1.679
64	0.232	39	2	0.65
81	0.272	12	1	1.157



(a) Erro em função da posição usando 16 nodos-treino.

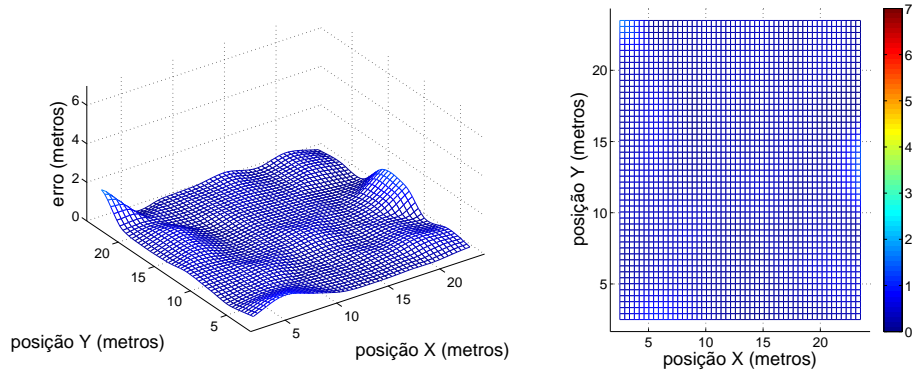


(b) Erro em função da posição usando 25 nodos-treino.

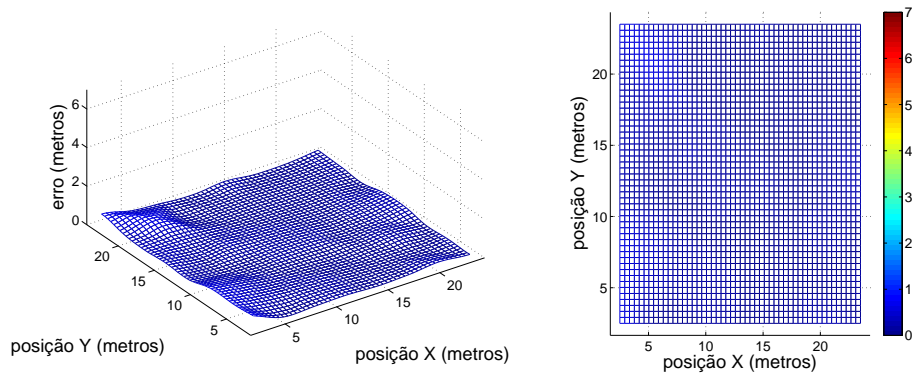


(c) Erro em função da posição usando 36 nodos-treino.

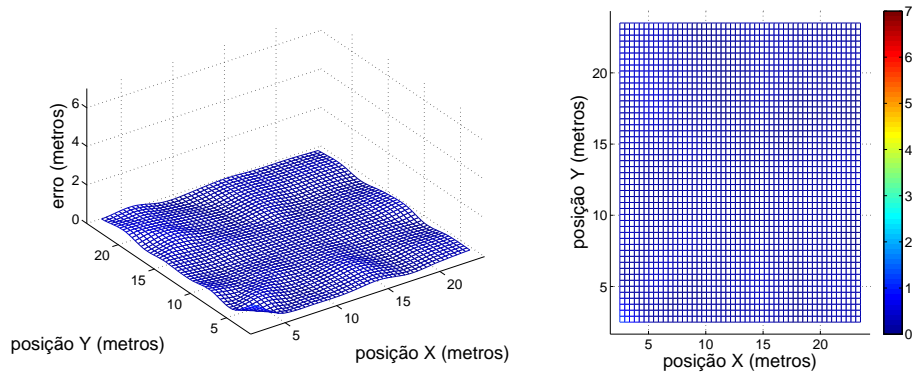
Figura 65: Influência do tamanho da amostra de treinamento no erro de localização para 3 nodos-âncora.



(a) Erro em função da posição usando 49 nodos-treino.

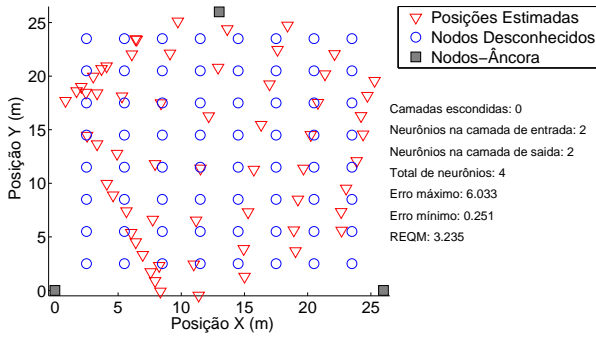


(b) Erro em função da posição usando 64 nodos-treino.

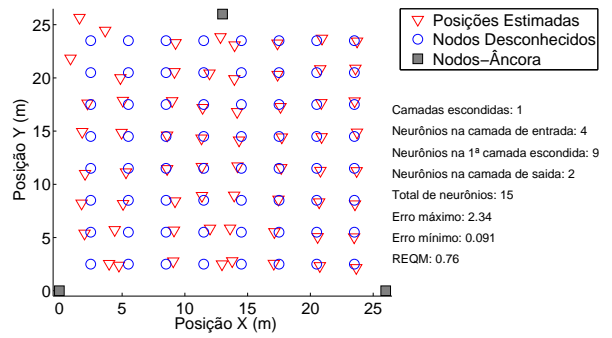


(c) Erro em função da posição usando 81 nodos-treino.

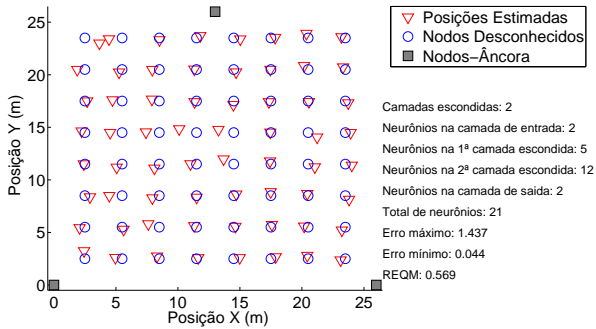
Figura 66: Influência do tamanho da amostra de treinamento no erro de localização para 3 nodos-âncora.



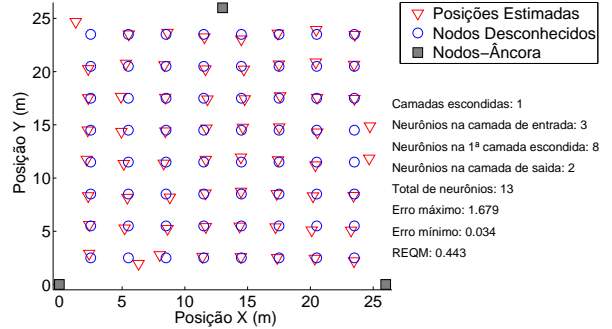
(a) Estimativas - amostra de 16 nodos-treino.



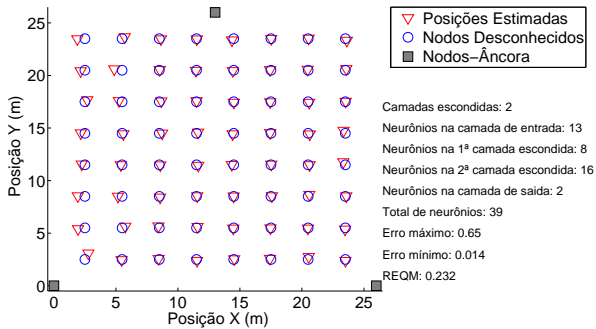
(b) Estimativas - amostra de 25 nodos-treino.



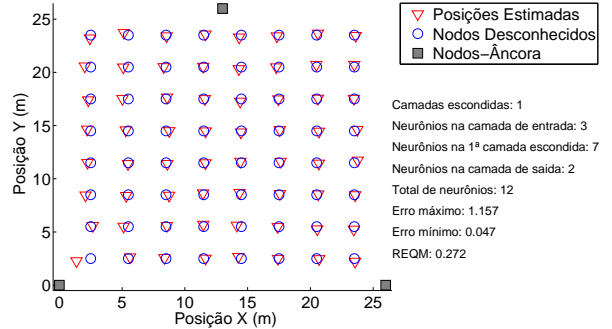
(c) Estimativas - amostra de 36 nodos-treino.



(d) Estimativas - amostra de 49 nodos-treino.

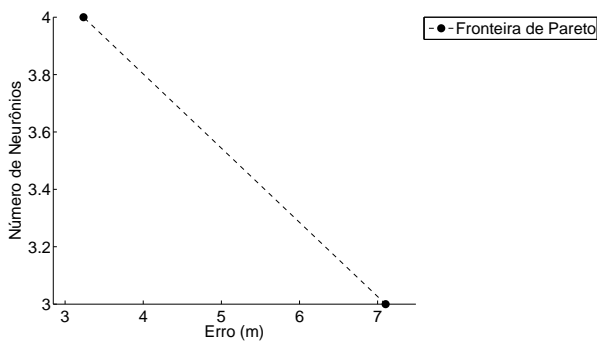


(e) Estimativas - amostra de 64 nodos-treino.

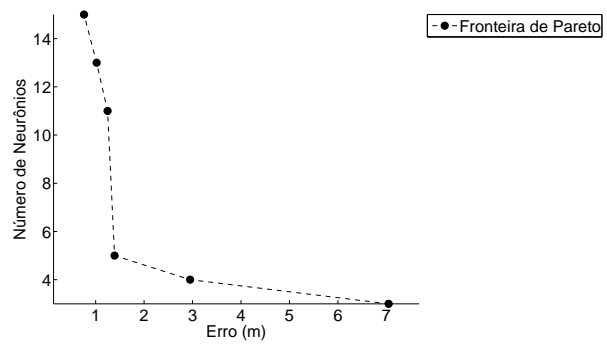


(f) Estimativas - amostra de 81 nodos-treino.

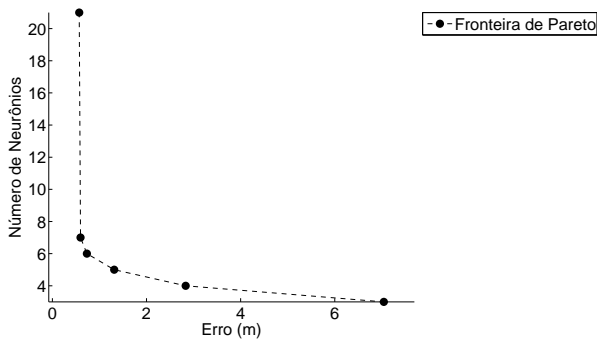
Figura 67: Influência do tamanho da amostra de treinamento nas estimativas de localização para 3 nodos-âncora.



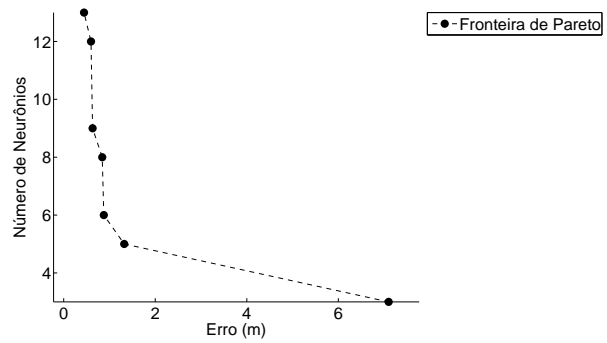
(a) Fronteira de Pareto usando 16 nodos-treino.



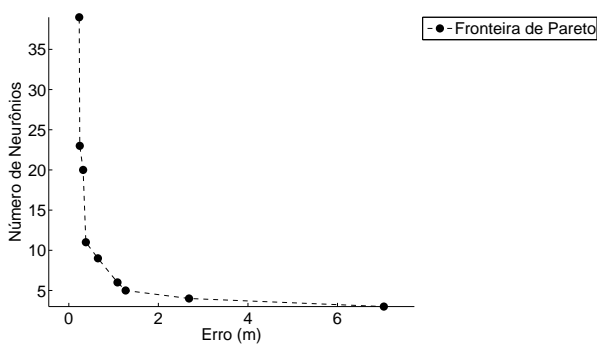
(b) Fronteira de Pareto usando 25 nodos-treino.



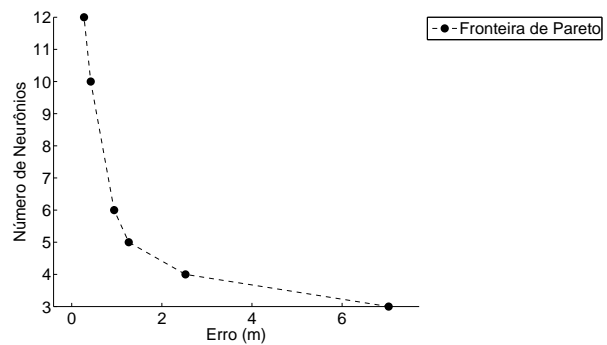
(c) Fronteira de Pareto usando 36 nodos-treino.



(d) Fronteira de Pareto usando 49 nodos-treino.



(e) Fronteira de Pareto usando 64 nodos-treino.



(f) Fronteira de Pareto usando 81 nodos-treino.

Figura 68: Influência do tamanho da amostra de treinamento nas fronteiras de Pareto para 3 nodos-âncora.

4.3.2.2 Topologias com 4 Nodos-Âncora

Como pode ser visto na figura 71, utilizando-se de 4 nodos-âncora e aumentando-se a quantidade de nodos-treino, o erro é reduzido. Nota-se claramente que a amostra composta por 16 nodos-treino continua não sendo suficiente para gerar boas estimativas. Outra constatação é a de que 83% dos casos apresentaram camadas escondidas nas RNAs.

Analisando-se as fronteiras de Pareto para cada caso (figura 72), é possível notar que o número de soluções com menos neurônios aumentou simultaneamente com o tamanho da amostra.

As figuras 69 e 70 indicam redução da REQM com o aumento dos nodos-treino e também que as incidências dos maiores erros encontram-se próximas aos nodos-âncora, sendo mais visíveis do que no caso com 3 nodos-âncora.

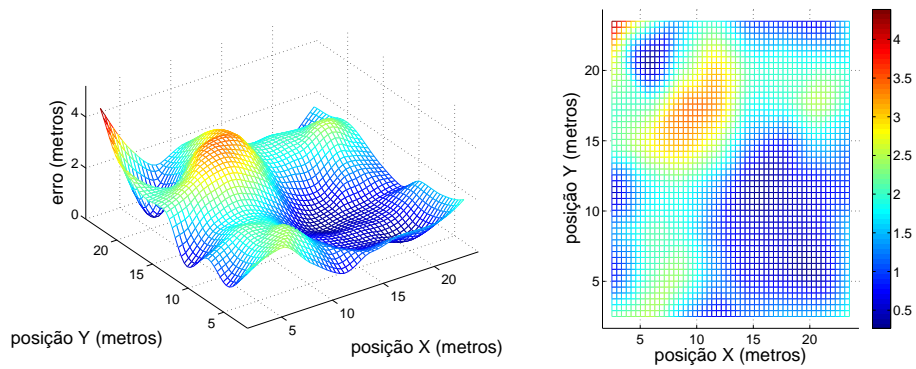
Nas tabelas 22 e 23 é possível analisar os resultados obtidos em termos de REQM e complexidade das RNAs para os casos de menor REQM obtido, ordenados por REQM e tamanho da amostra.

Tabela 22: Resultados para 4 âncoras variando-se o tamanho da amostra de treinamento - ordenação por REQM

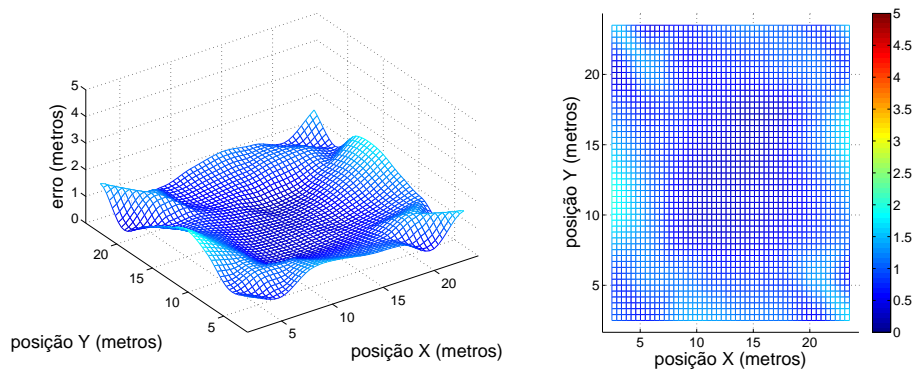
REQM (metros)	Tamanho da Amostra	Camadas Escondidas	Total de Neurônios	Distância Máxima (metros)
0.288	81	2	28	1.07
0.531	64	1	10	1.42
0.549	49	1	28	1.069
0.789	36	0	6	2.488
1.003	25	2	27	1.966
1.815	16	2	18	4.387

Tabela 23: Resultados para 4 âncoras variando-se o tamanho da amostra de treinamento - ordenação por âncoras

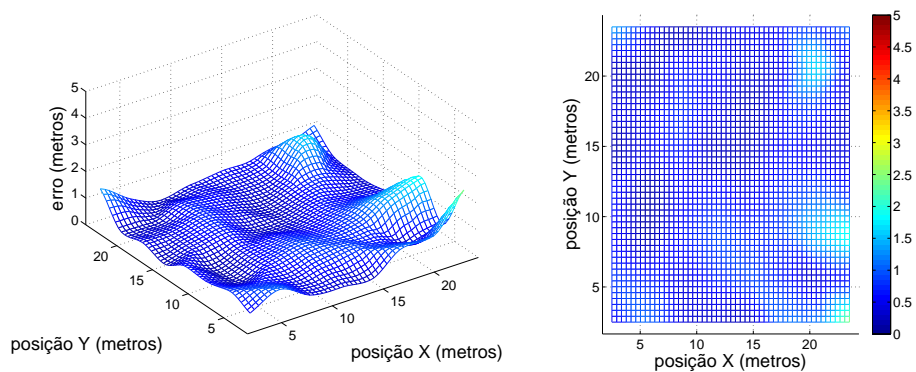
Tamanho da Amostra	REQM (metros)	Total de Neurônios	Camadas Escondidas	Distância Máxima (metros)
16	1.815	18	2	4.387
25	1.003	27	2	1.966
36	0.789	6	0	2.488
49	0.549	28	1	1.069
64	0.531	10	1	1.42
81	0.288	28	2	1.07



(a) Erro em função da posição usando 16 nodos-treino.

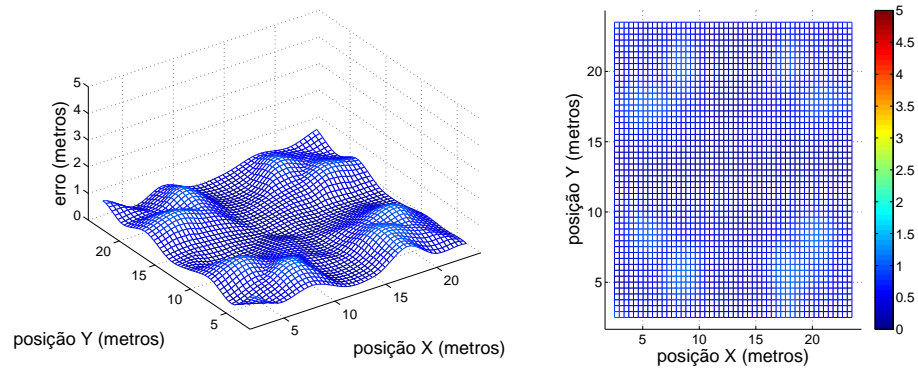


(b) Erro em função da posição usando 25 nodos-treino.

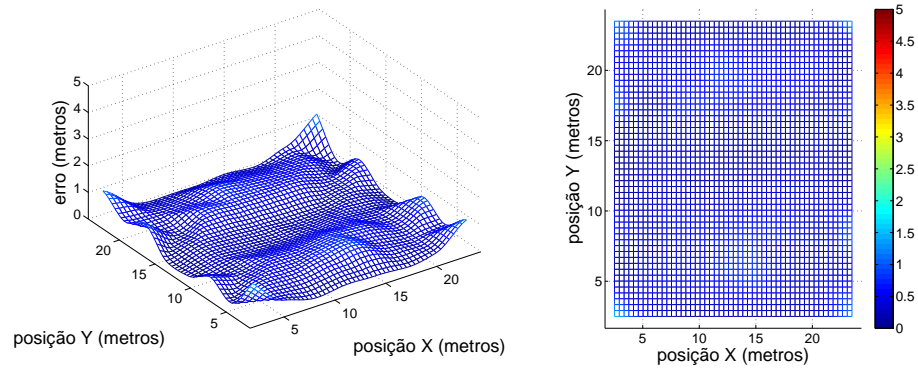


(c) Erro em função da posição usando 36 nodos-treino.

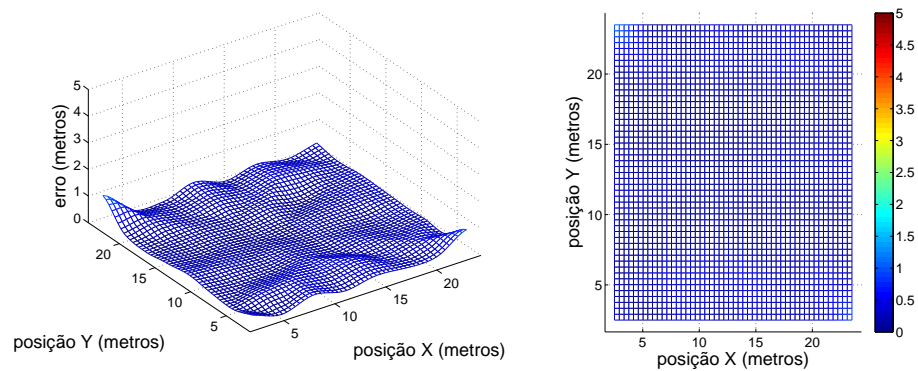
Figura 69: Influência do tamanho da amostra de treinamento no erro de localização para 4 nodos-âncora.



(a) Erro em função da posição usando 49 nodos-treino.

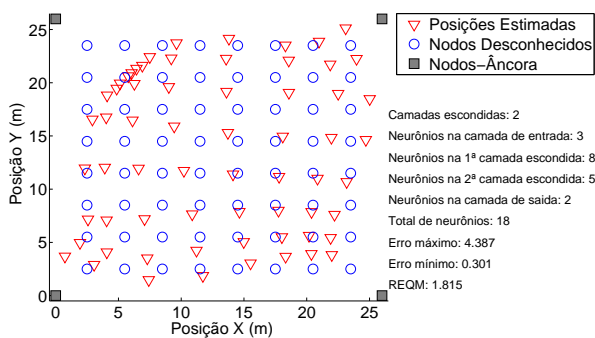


(b) Erro em função da posição usando 64 nodos-treino.

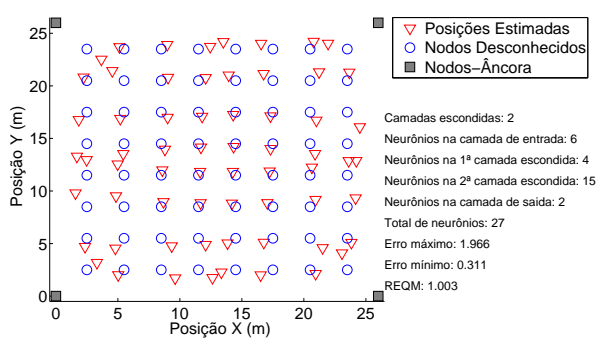


(c) Erro em função da posição usando 81 nodos-treino.

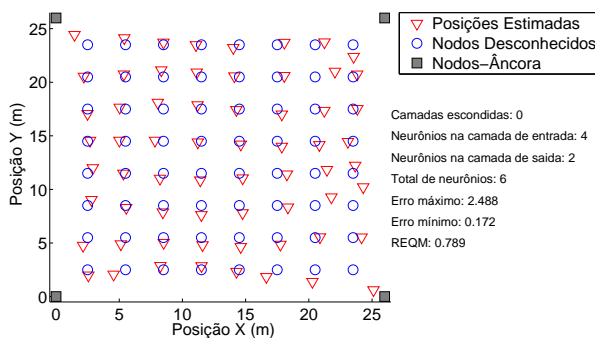
Figura 70: Influência do tamanho da amostra de treinamento no erro de localização para 4 nodos-âncora.



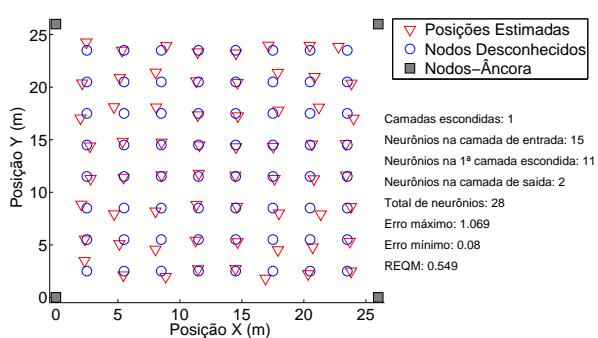
(a) Estimativas - amostra de 16 nodos-treino.



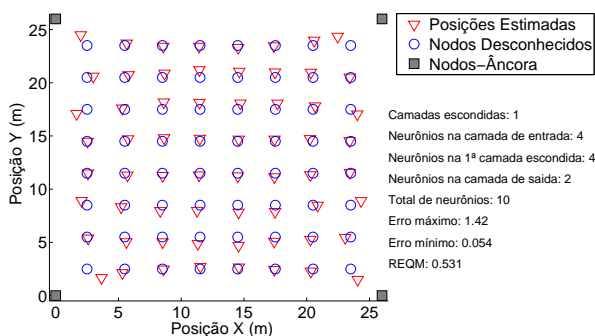
(b) Estimativas - amostra de 25 nodos-treino.



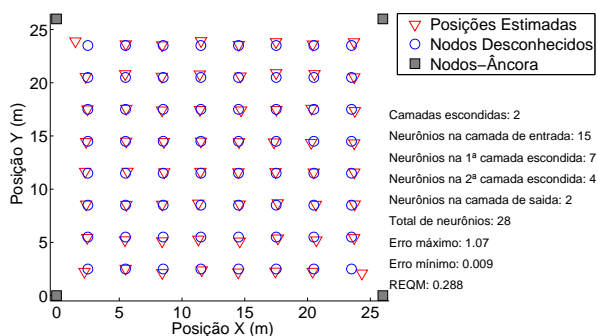
(c) Estimativas - amostra de 36 nodos-treino.



(d) Estimativas - amostra de 49 nodos-treino.

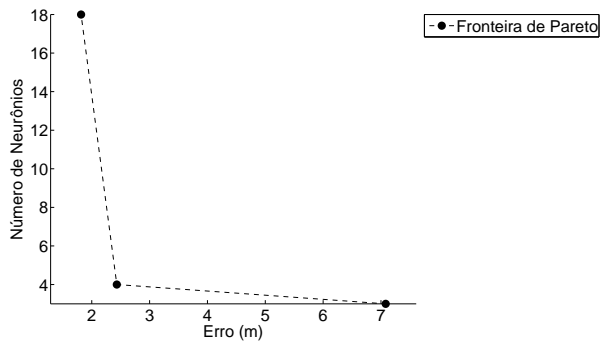


(e) Estimativas - amostra de 64 nodos-treino.

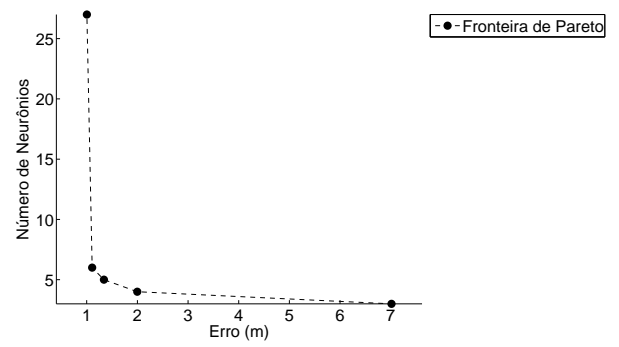


(f) Estimativas - amostra de 81 nodos-treino.

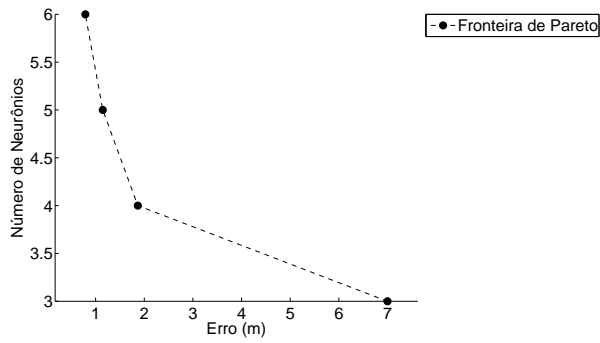
Figura 71: Influência do tamanho da amostra de treinamento nas estimativas de localização para 4 nodos-âncora.



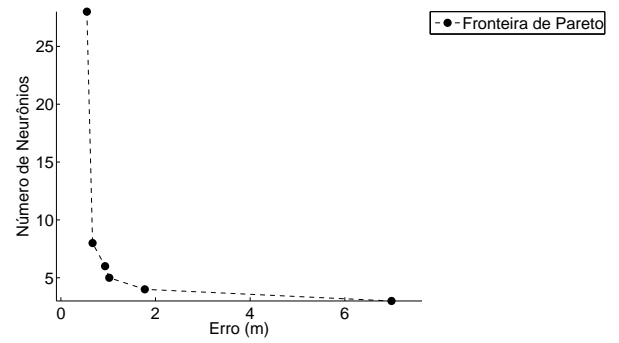
(a) Fronteira de Pareto usando 16 nodos-treino.



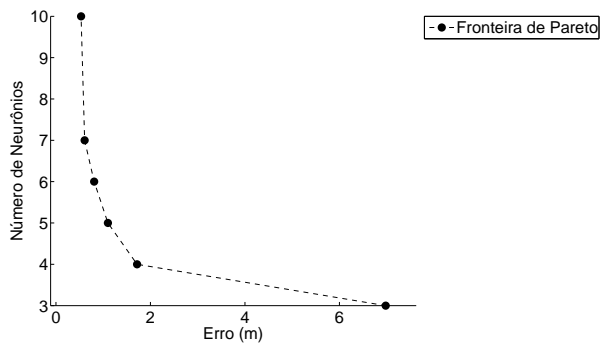
(b) Fronteira de Pareto usando 25 nodos-treino.



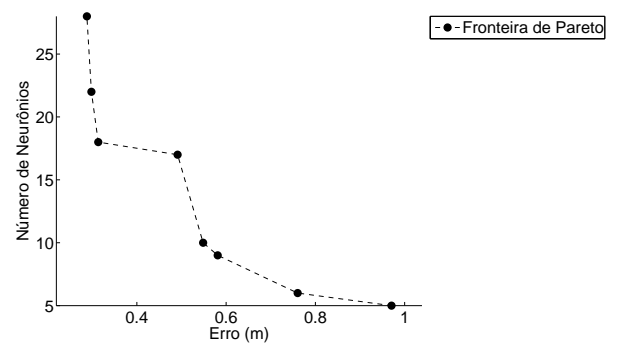
(c) Fronteira de Pareto usando 36 nodos-treino.



(d) Fronteira de Pareto usando 49 nodos-treino.



(e) Fronteira de Pareto usando 64 nodos-treino.



(f) Fronteira de Pareto usando 81 nodos-treino.

Figura 72: Influência do tamanho da amostra de treinamento nas fronteiras de Pareto para 4 nodos-âncora.

4.3.2.3 Topologias com 5 Nodos-Âncora

Como pode ser visto na figura 75, utilizando-se de 5 nodos-âncora e aumentando-se a quantidade de nodos-treino, o erro é reduzido. A amostra composta por 16 nodos-treino continua não sendo suficiente para gerar boas estimativas. Outra constatação é a de que 66,66% dos casos apresentaram camadas escondidas nas RNAs, provavelmente devido ao nodo-âncora central.

Analisando-se as fronteiras de Pareto para cada caso (figura 76), é possível notar que o número de soluções com menos neurônios aumentou simultaneamente com o tamanho da amostra.

As figuras 73 e 74 indicam redução da REQM com o aumento dos nodos-treino e também mostram aumento do erro na região central, ausente nos 2 casos anteriores.

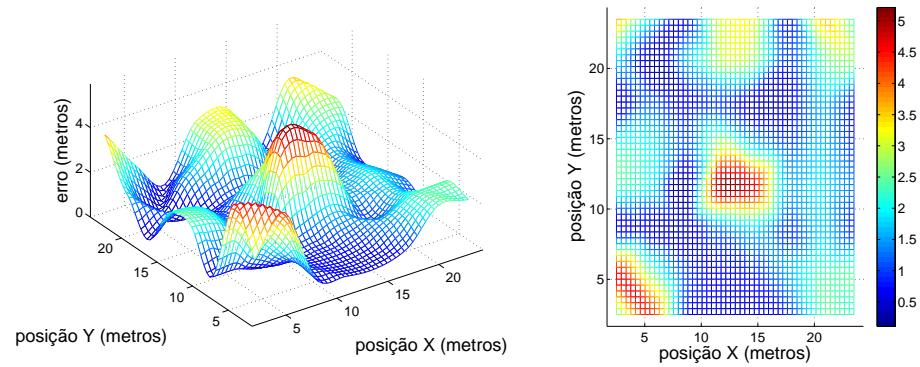
Nas tabelas 24 e 25 é possível analisar os resultados obtidos em termos de REQM e complexidade das RNAs para os casos de menor REQM obtido, ordenados por REQM e tamanho da amostra.

Tabela 24: Resultados para 5 âncoras variando-se o tamanho da amostra de treinamento - ordenação por REQM

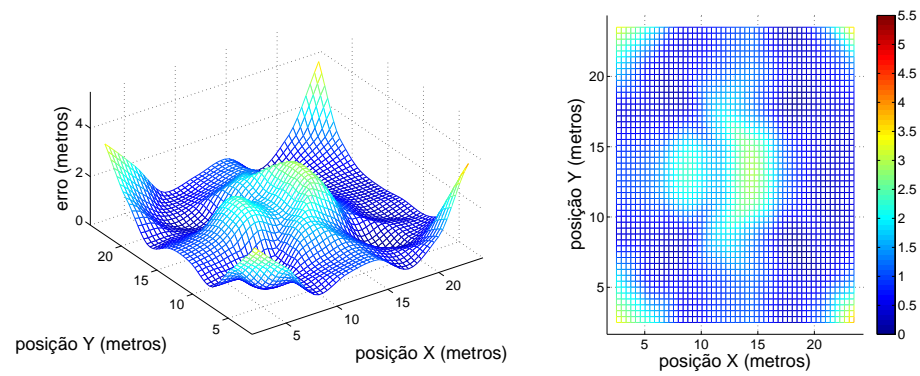
REQM (metros)	Tamanho da Amostra	Camadas Escondidas	Total de Neurônios	Distância Máxima (metros)
0.263	81	1	16	0.694
0.361	64	1	11	1.674
0.498	49	1	10	1.477
1.045	36	0	6	3.182
1.451	25	0	5	3.875
2.078	16	1	21	4.959

Tabela 25: Resultados para 5 âncoras variando-se o tamanho da amostra de treinamento - ordenação por âncoras

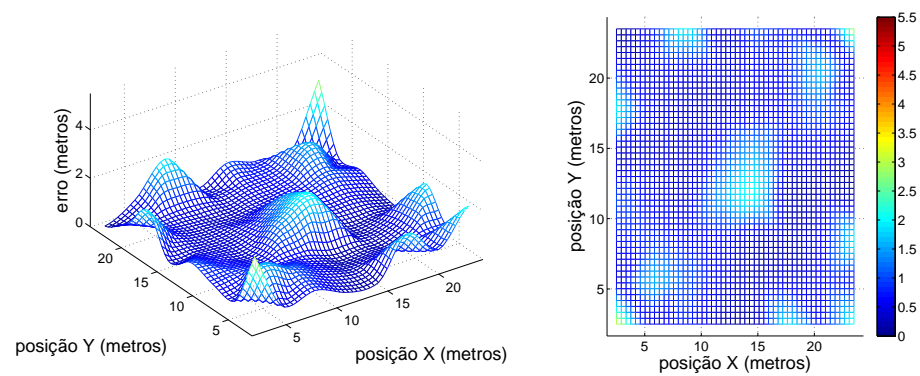
Tamanho da Amostra	REQM (metros)	Total de Neurônios	Camadas Escondidas	Distância Máxima (metros)
16	2.078	21	1	4.959
25	1.451	5	0	3.875
36	1.045	6	0	3.182
49	0.498	10	1	1.477
64	0.361	11	1	1.674
81	0.263	16	1	0.694



(a) Erro em função da posição usando 16 nodos-treino.

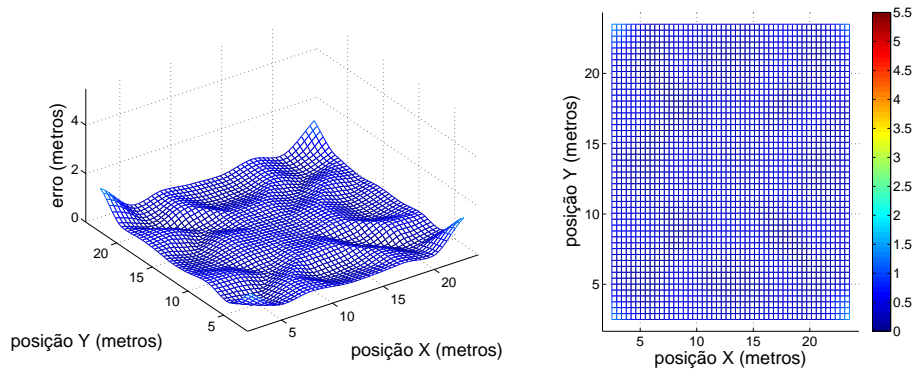


(b) Erro em função da posição usando 25 nodos-treino.

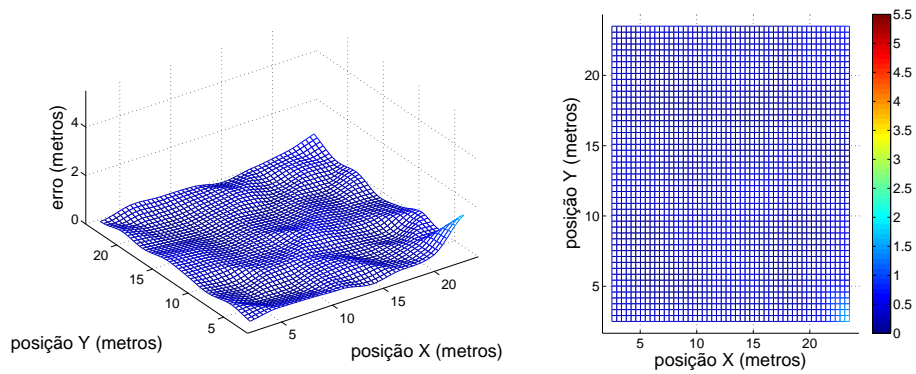


(c) Erro em função da posição usando 36 nodos-treino.

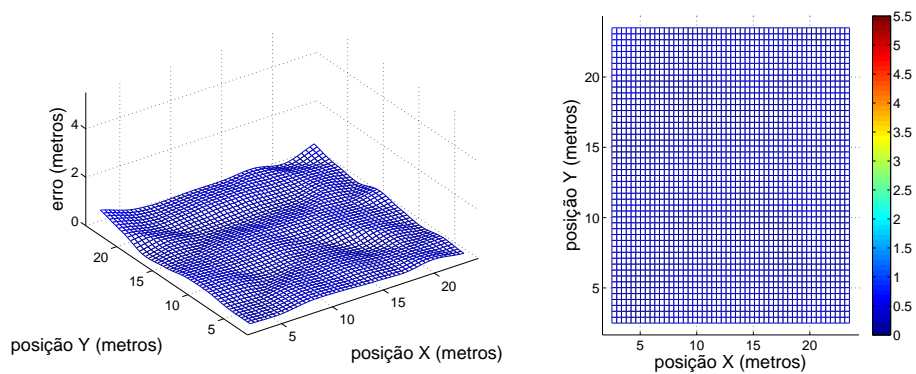
Figura 73: Influência do tamanho da amostra de treinamento no erro de localização para 5 nodos-âncora.



(a) Erro em função da posição usando 49 nodos-treino.

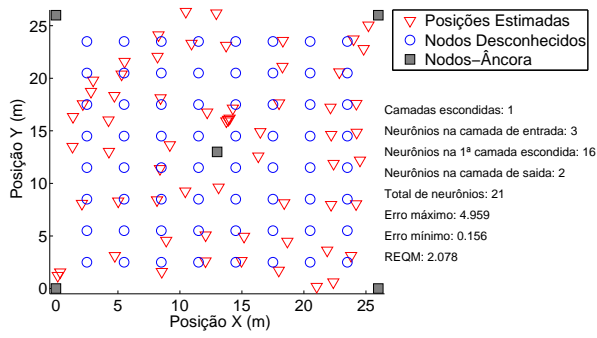


(b) Erro em função da posição usando 64 nodos-treino.

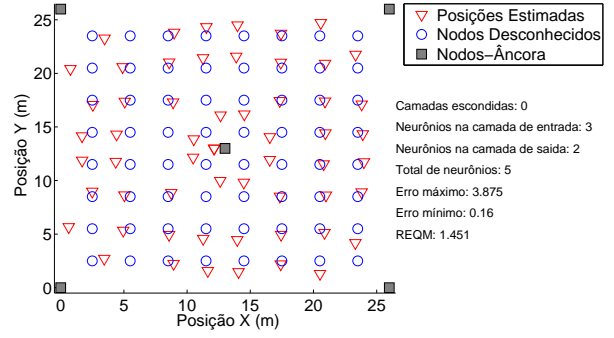


(c) Erro em função da posição usando 81 nodos-treino.

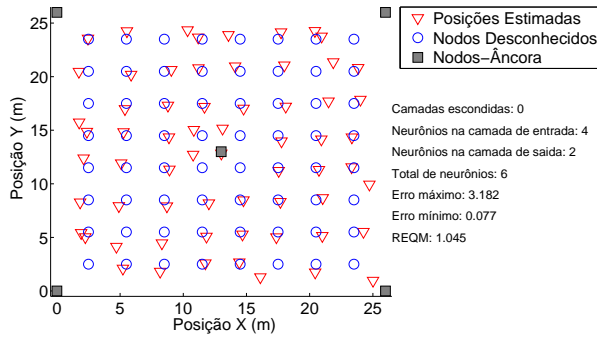
Figura 74: Influência do tamanho da amostra de treinamento no erro de localização para 5 nodos-âncora.



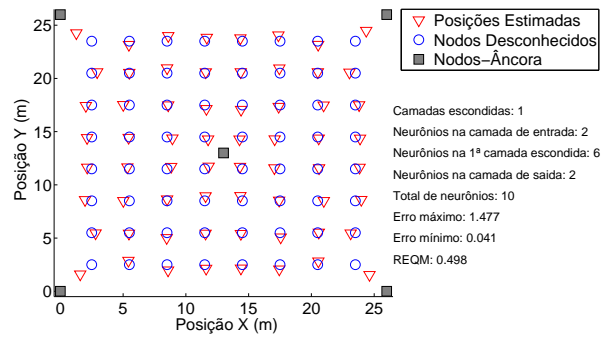
(a) Estimativas - amostra de 16 nodos-treino.



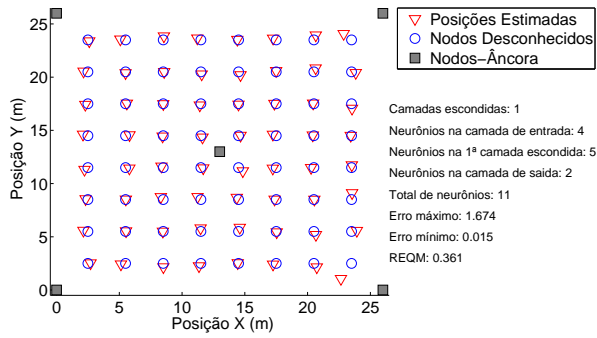
(b) Estimativas - amostra de 25 nodos-treino.



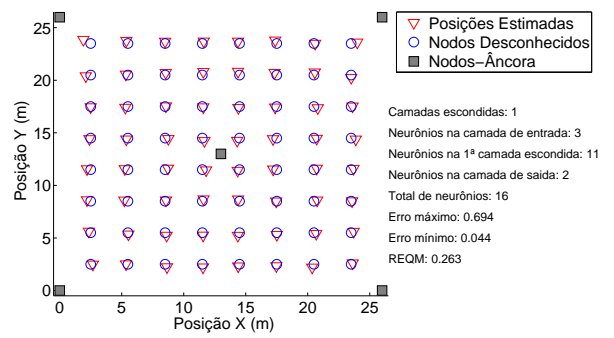
(c) Estimativas - amostra de 36 nodos-treino.



(d) Estimativas - amostra de 49 nodos-treino.

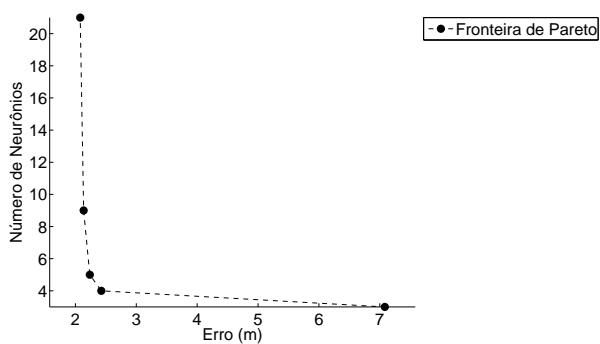


(e) Estimativas - amostra de 64 nodos-treino.

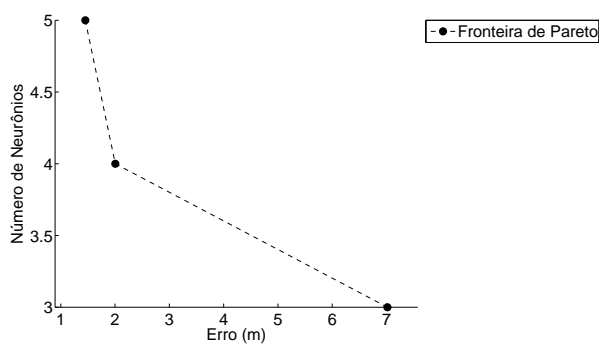


(f) Estimativas - amostra de 81 nodos-treino.

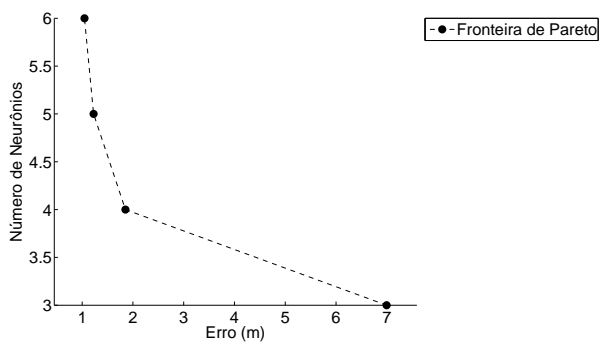
Figura 75: Influência do tamanho da amostra de treinamento nas estimativas de localização para 5 nodos-âncora.



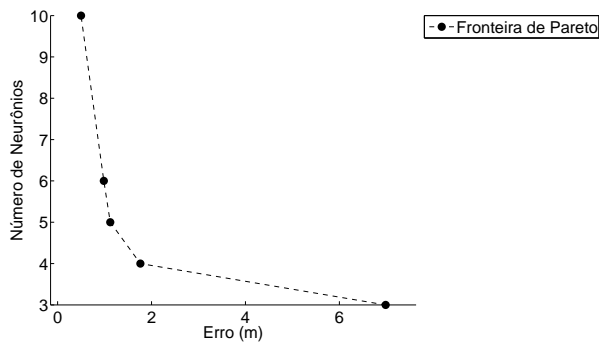
(a) Fronteira de Pareto usando 16 nodos-treino.



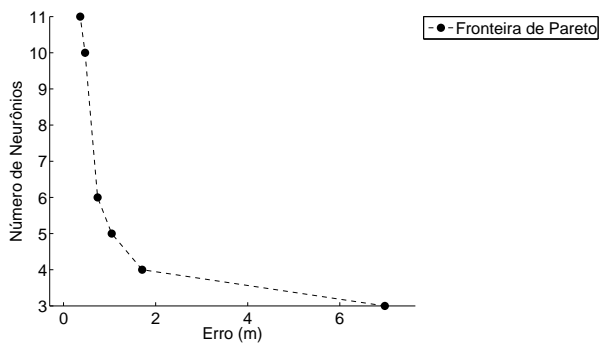
(b) Fronteira de Pareto usando 25 nodos-treino.



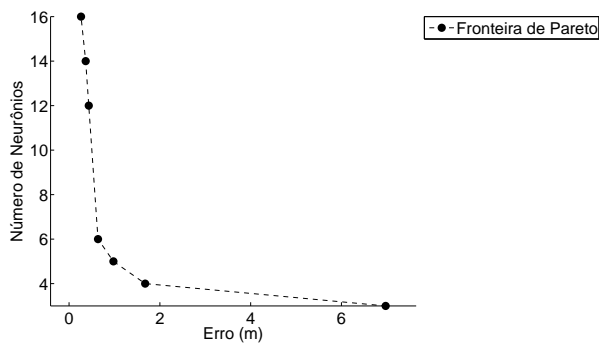
(c) Fronteira de Pareto usando 36 nodos-treino.



(d) Fronteira de Pareto usando 49 nodos-treino.



(e) Fronteira de Pareto usando 64 nodos-treino.



(f) Fronteira de Pareto usando 81 nodos-treino.

Figura 76: Influência do tamanho da amostra de treinamento nas fronteiras de Pareto para 5 nodos-âncora.

4.3.2.4 Topologias com 6 Nodos-Âncora

Como pode ser visto na figura 79, utilizando-se de 6 nodos-âncora e aumentando-se a quantidade de nodos-treino, o erro é reduzido. A amostra composta por 16 nodos-treino segue com o pior desempenho. Outra constatação é a de que 33,33% dos casos não apresentaram camadas escondidas nas RNAs.

Analisando-se as fronteiras de Pareto para cada caso (figura 80), é possível notar que o número de soluções com menos neurônios continua aumentando simultaneamente com o tamanho da amostra.

As figuras 77 e 78 indicam redução da REQM com o aumento dos nodos-treino, porém com aumento nas proximidades dos nodos-âncora.

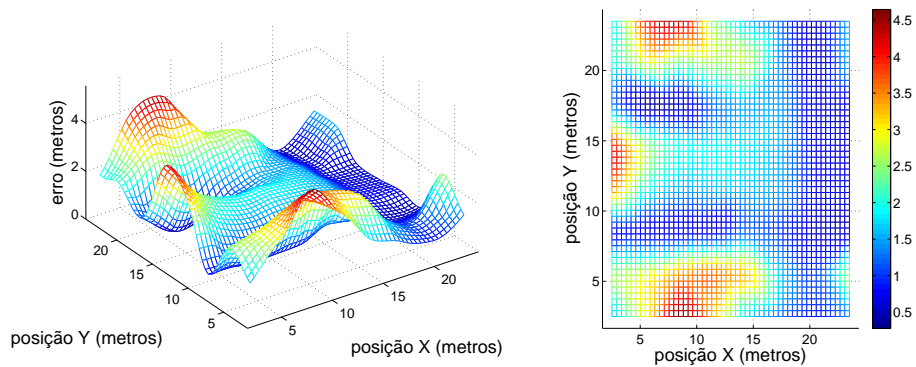
Nas tabelas 26 e 27 é possível analisar os resultados obtidos em termos de REQM e complexidade das RNAs para os casos de menor REQM obtido, ordenados por REQM e tamanho da amostra.

Tabela 26: Resultados para 6 âncoras variando-se o tamanho da amostra de treinamento - ordenação por REQM

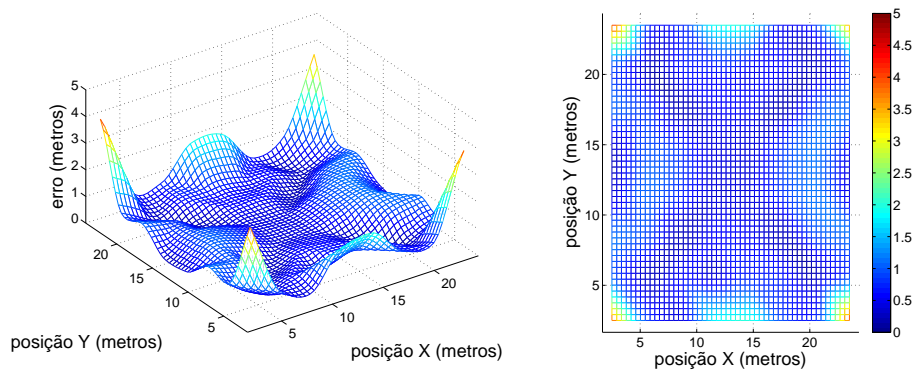
REQM (metros)	Tamanho da Amostra	Camadas Escondidas	Total de Neurônios	Distância Máxima (metros)
0.334	81	2	24	0.993
0.553	64	1	9	1.194
0.674	49	0	6	1.966
0.725	36	1	15	1.808
1.268	25	1	12	3.948
2.067	16	0	4	4.657

Tabela 27: Resultados para 6 âncoras variando-se o tamanho da amostra de treinamento - ordenação por âncoras

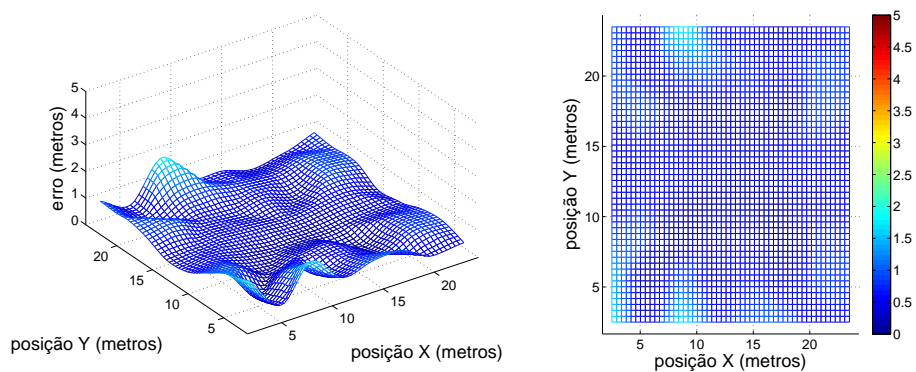
Tamanho da Amostra	REQM (metros)	Total de Neurônios	Camadas Escondidas	Distância Máxima (metros)
16	2.067	4	0	4.657
25	1.268	12	1	3.948
36	0.725	15	1	1.808
49	0.674	6	0	1.966
64	0.553	9	1	1.194
81	0.334	24	2	0.993



(a) Erro em função da posição usando 16 nodos-treino.

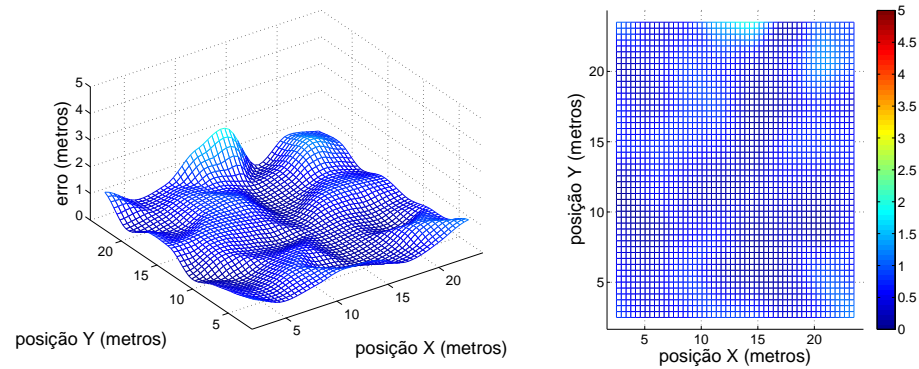


(b) Erro em função da posição usando 25 nodos-treino.

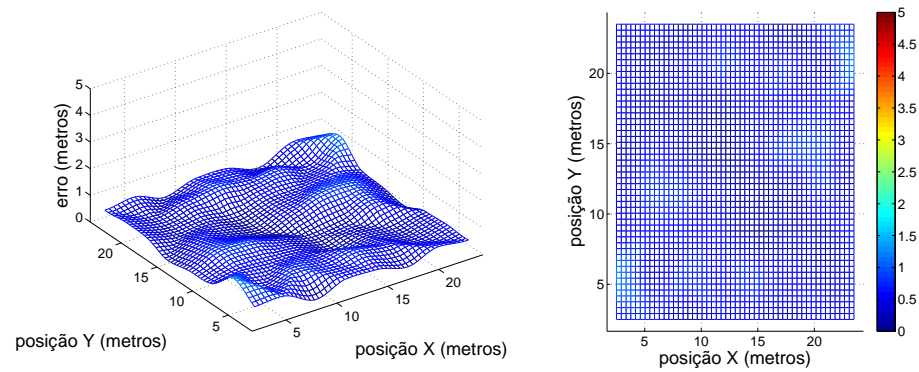


(c) Erro em função da posição usando 36 nodos-treino.

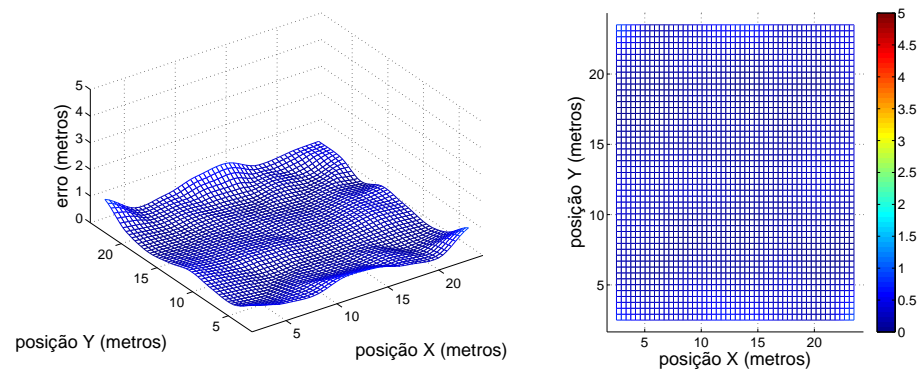
Figura 77: Influência do tamanho da amostra de treinamento no erro de localização para 6 nodos-âncora.



(a) Erro em função da posição usando 49 nodos-treino.

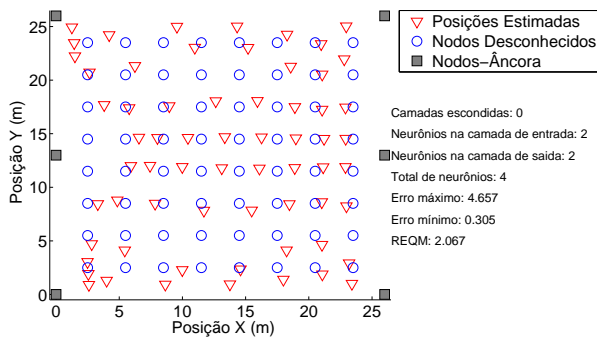


(b) Erro em função da posição usando 64 nodos-treino.

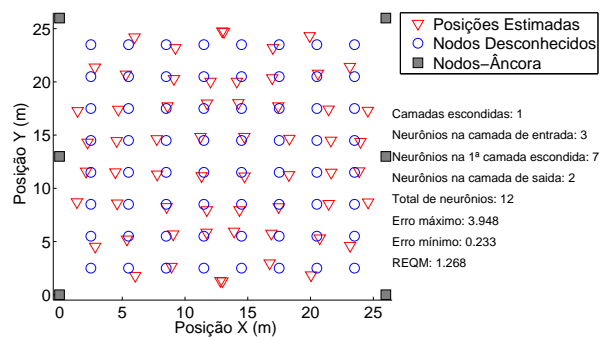


(c) Erro em função da posição usando 81 nodos-treino.

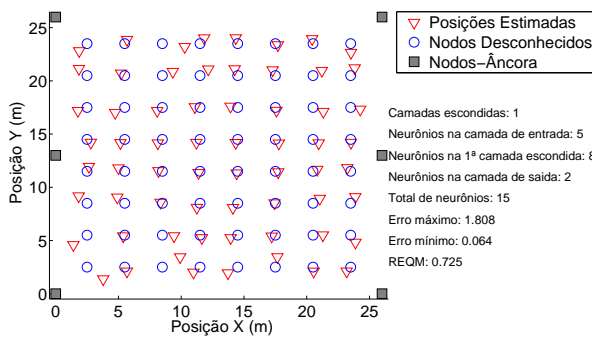
Figura 78: Influência do tamanho da amostra de treinamento no erro de localização para 6 nodos-âncora.



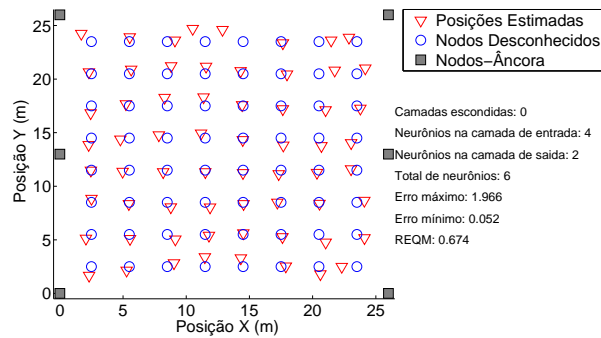
(a) Estimativas - amostra de 16 nodos-treino.



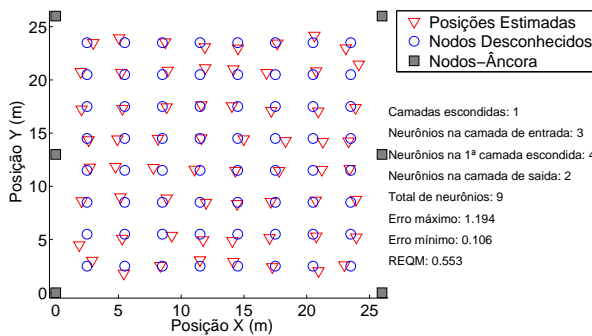
(b) Estimativas - amostra de 25 nodos-treino.



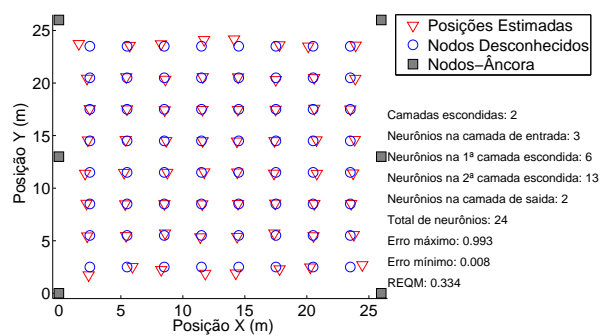
(c) Estimativas - amostra de 36 nodos-treino.



(d) Estimativas - amostra de 49 nodos-treino.

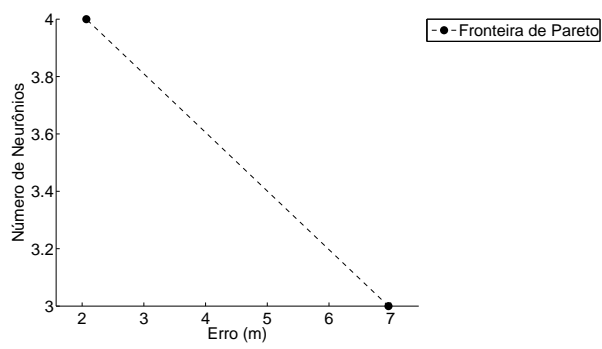


(e) Estimativas - amostra de 64 nodos-treino.

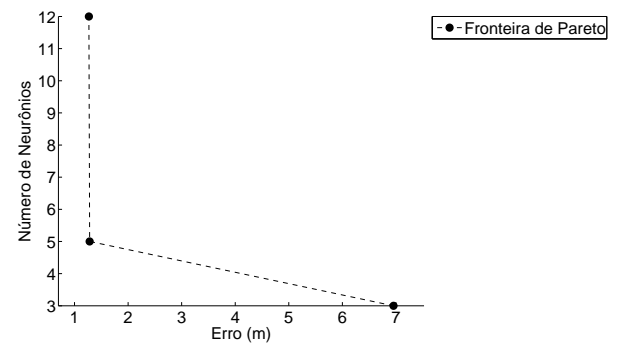


(f) Estimativas - amostra de 81 nodos-treino.

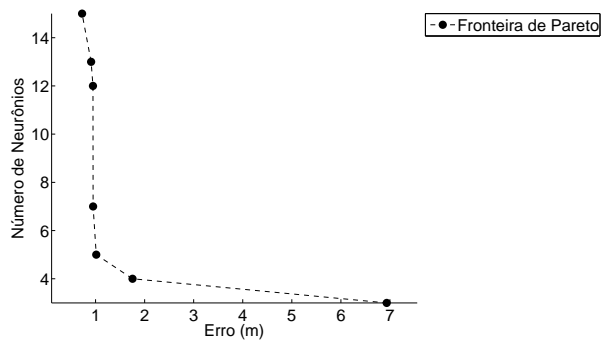
Figura 79: Influência do tamanho da amostra de treinamento nas estimativas de localização para 6 nodos-âncora.



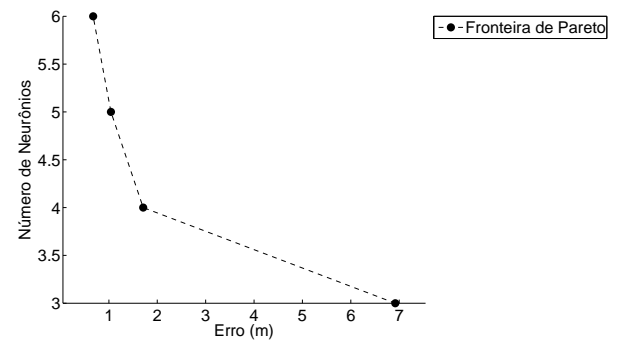
(a) Fronteira de Pareto usando 16 nodos-treino.



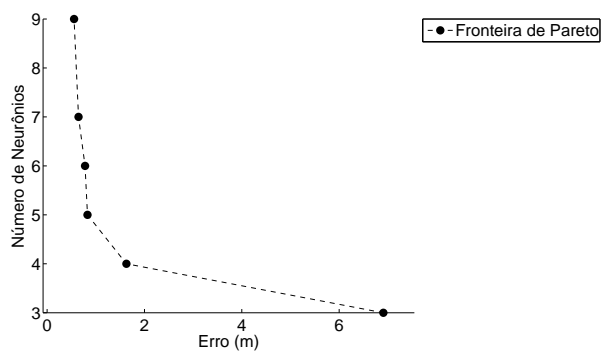
(b) Fronteira de Pareto usando 25 nodos-treino.



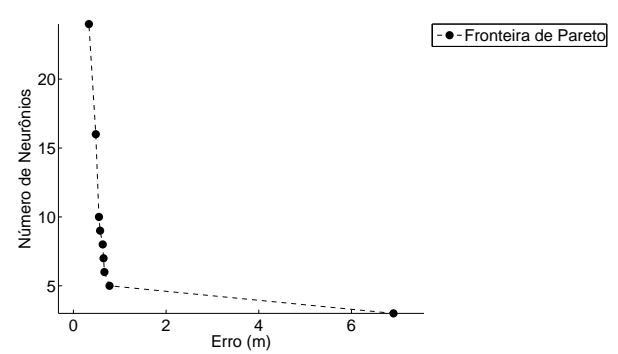
(c) Fronteira de Pareto usando 36 nodos-treino.



(d) Fronteira de Pareto usando 49 nodos-treino.



(e) Fronteira de Pareto usando 64 nodos-treino.



(f) Fronteira de Pareto usando 81 nodos-treino.

Figura 80: Influência do tamanho da amostra de treinamento nas fronteiras de Pareto para 6 nodos-âncora.

4.3.2.5 Topologias com 7 Nodos-Âncora

Como pode ser visto na figura 83, utilizando-se de 7 nodos-âncora e aumentando-se a quantidade de nodos-treino, o erro é reduzido. A amostra composta por 16 nodos-treino segue com o pior desempenho. Outra constatação é a de que 66,66% dos casos não apresentaram camadas escondidas nas RNAs, fato que indica relação deste dado com o nodo-âncora central.

Analisando-se as fronteiras de Pareto para cada caso (figura 84), é possível notar que o número de soluções com menos neurônios não apresenta aumento simultâneo com o tamanho da amostra.

As figuras 81 e 82 indicam redução da REQM com o aumento dos nodos-treino, porém com aumento nas proximidades dos nodos-âncora, inclusive no centro da área de teste.

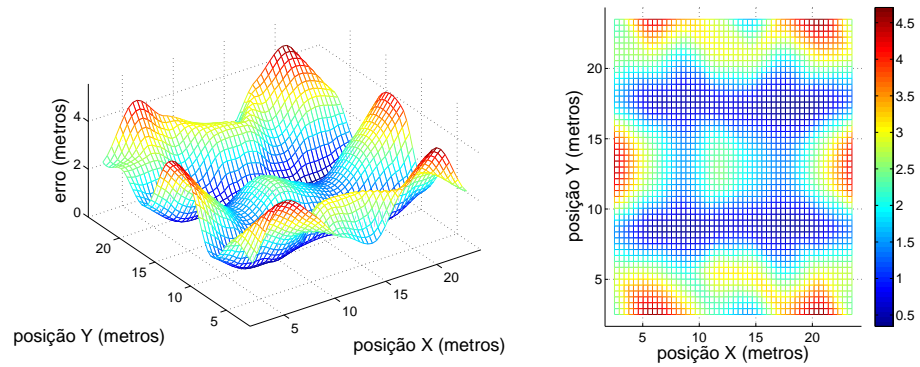
Nas tabelas 28 e 29 é possível analisar os resultados obtidos em termos de REQM e complexidade das RNAs para os casos de menor REQM obtido, ordenados por REQM e tamanho da amostra.

Tabela 28: Resultados para 7 âncoras variando-se o tamanho da amostra de treinamento - ordenação por REQM

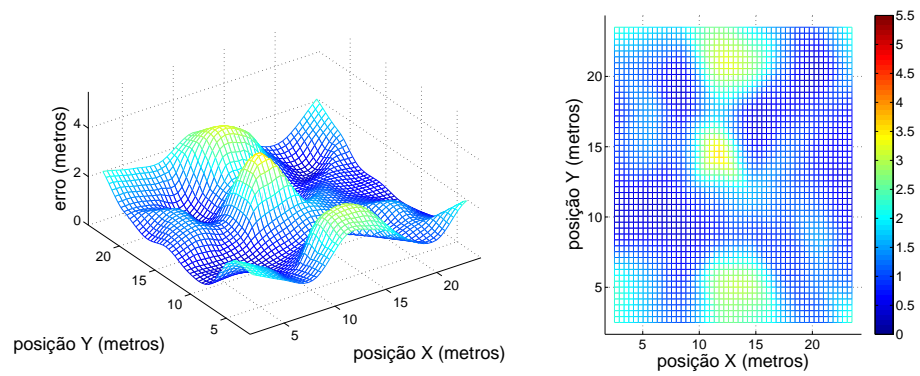
REQM (metros)	Tamanho da Amostra	Camadas Escondidas	Total de Neurônios	Distância Máxima (metros)
0.376	81	0	7	0.906
0.606	64	0	6	1.425
0.665	49	1	21	2.766
1.128	36	1	14	4.635
1.545	25	0	5	3.55
2.518	16	0	4	4.696

Tabela 29: Resultados para 7 âncoras variando-se o tamanho da amostra de treinamento - ordenação por âncoras

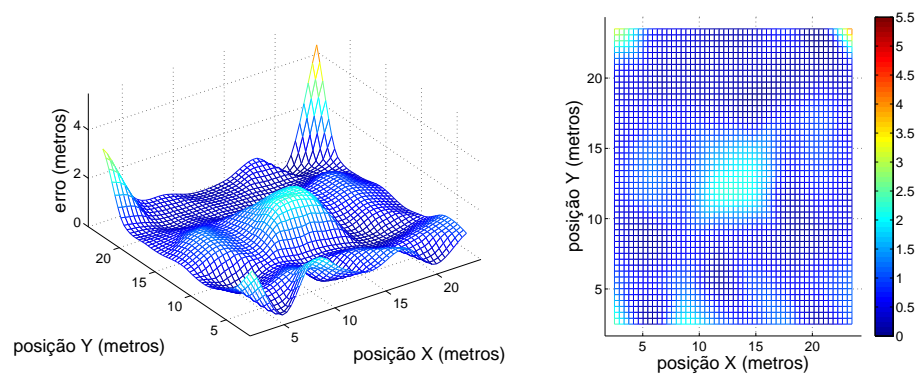
Tamanho da Amostra	REQM (metros)	Total de Neurônios	Camadas Escondidas	Distância Máxima (metros)
16	2.518	4	0	4.696
25	1.545	5	0	3.55
36	1.128	14	1	4.635
49	0.665	21	1	2.766
64	0.606	6	0	1.425
81	0.376	7	0	0.906



(a) Erro em função da posição usando 16 nodos-treino.

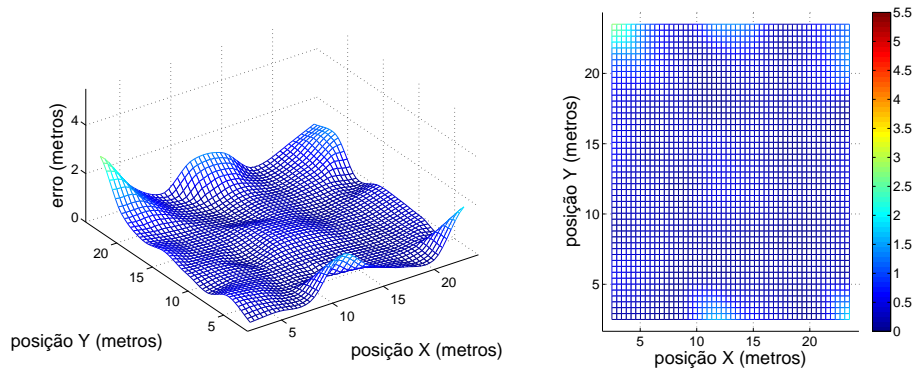


(b) Erro em função da posição usando 25 nodos-treino.

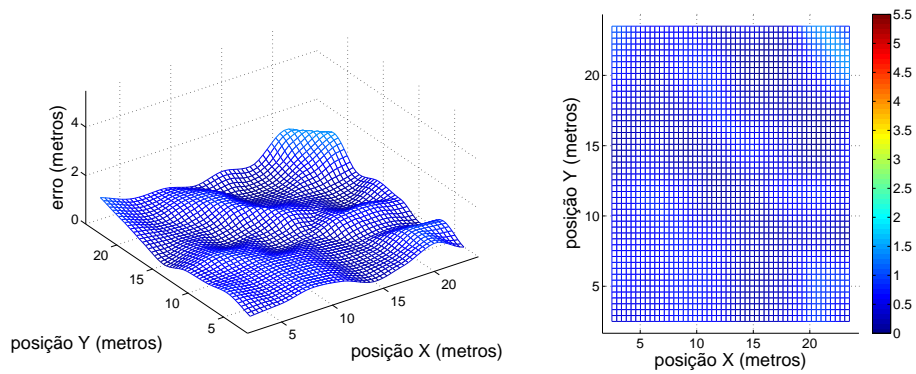


(c) Erro em função da posição usando 36 nodos-treino.

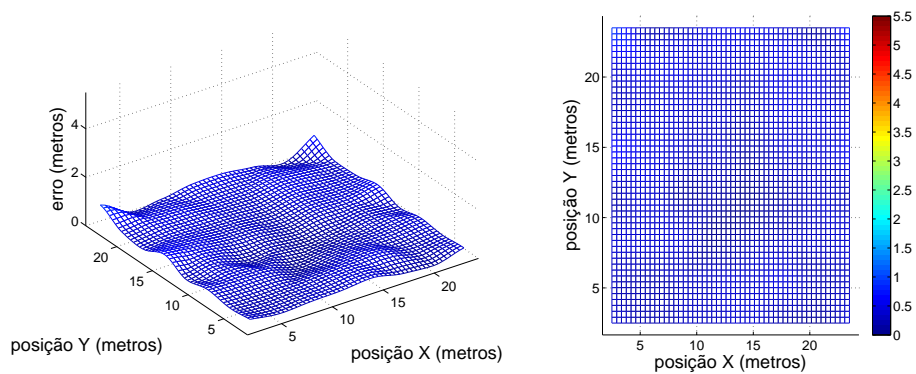
Figura 81: Influência do tamanho da amostra de treinamento no erro de localização para 7 nodos-âncora.



(a) Erro em função da posição usando 49 nodos-treino.

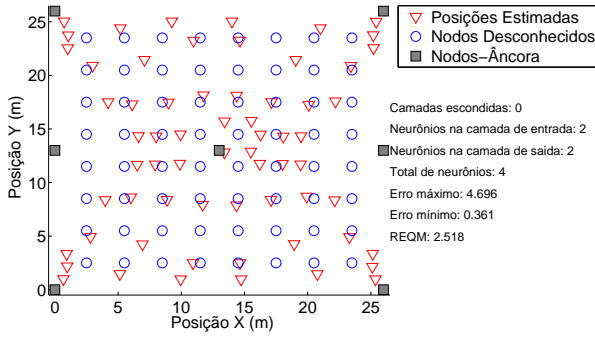


(b) Erro em função da posição usando 64 nodos-treino.

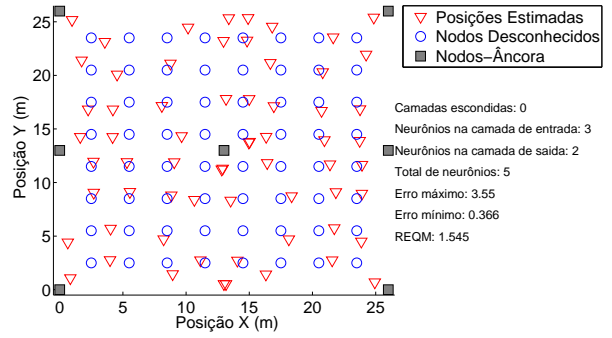


(c) Erro em função da posição usando 81 nodos-treino.

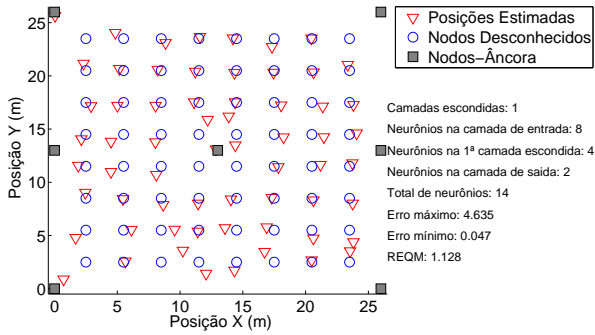
Figura 82: Influência do tamanho da amostra de treinamento no erro de localização para 7 nodos-âncora.



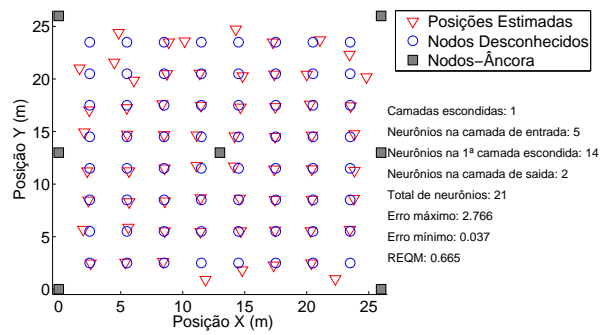
(a) Estimativas - amostra de 16 nodos-treino.



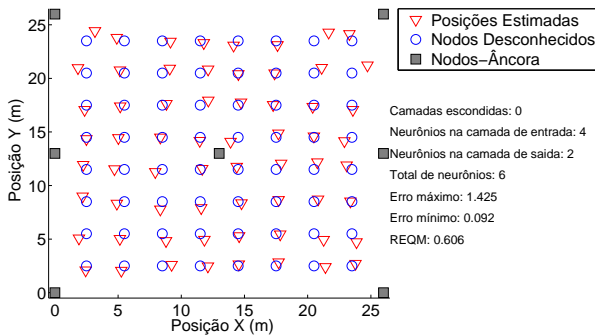
(b) Estimativas - amostra de 25 nodos-treino.



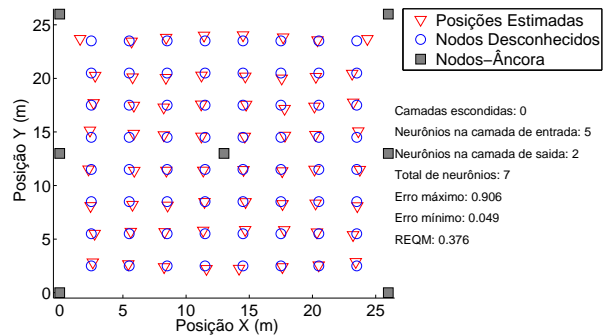
(c) Estimativas - amostra de 36 nodos-treino.



(d) Estimativas - amostra de 49 nodos-treino.

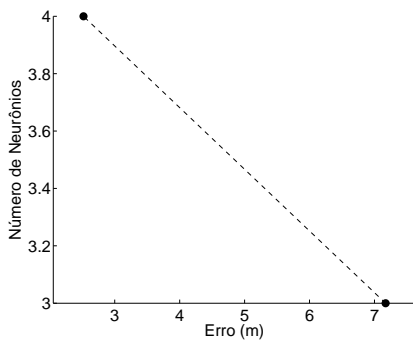


(e) Estimativas - amostra de 64 nodos-treino.

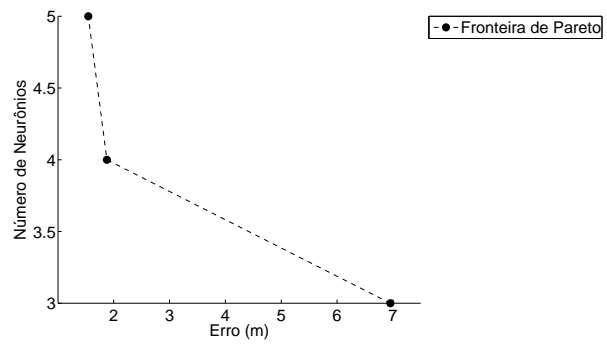


(f) Estimativas - amostra de 81 nodos-treino.

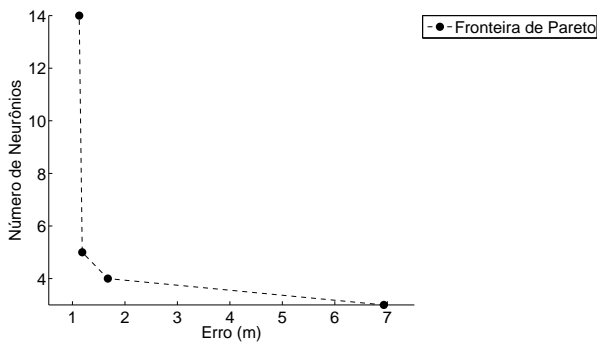
Figura 83: Influência do tamanho da amostra de treinamento nas estimativas de localização para 7 nodos-âncora.



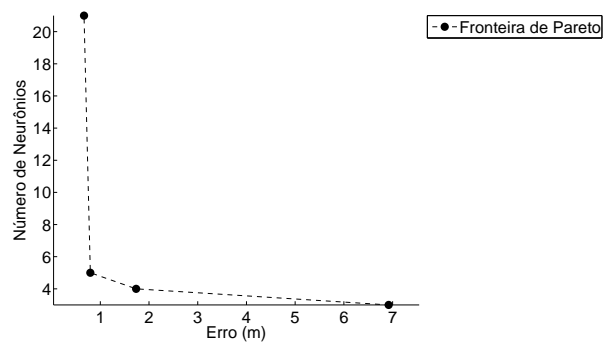
(a) Fronteira de Pareto usando 16 nodos-treino.



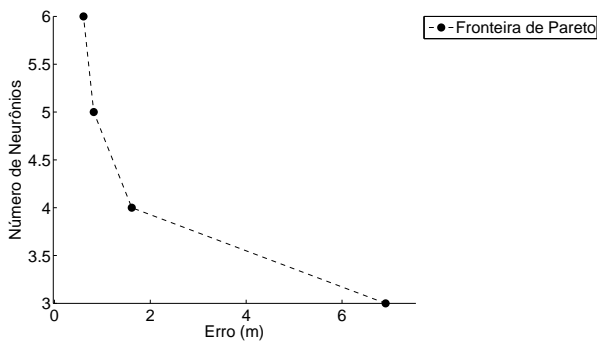
(b) Fronteira de Pareto usando 25 nodos-treino.



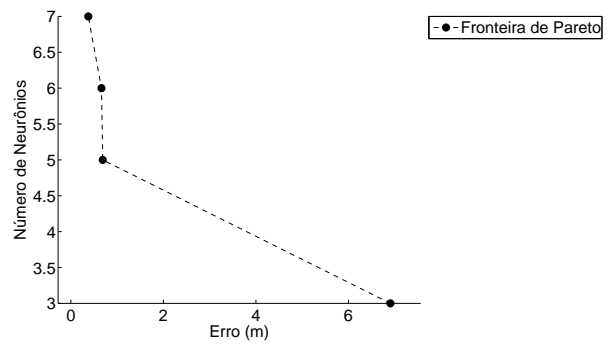
(c) Fronteira de Pareto usando 36 nodos-treino.



(d) Fronteira de Pareto usando 49 nodos-treino.



(e) Fronteira de Pareto usando 64 nodos-treino.



(f) Fronteira de Pareto usando 81 nodos-treino.

Figura 84: Influência do tamanho da amostra de treinamento nas fronteiras de Pareto para 7 nodos-âncora.

4.3.2.6 Topologias com 8 Nodos-Âncora

Como pode ser visto na figura 87, utilizando-se de 8 nodos-âncora e aumentando-se a quantidade de nodos-treino, o erro é reduzido. A amostra composta por 16 nodos-treino segue com o pior desempenho. Outra constatação é a de que 50% dos casos não apresentaram camadas escondidas nas RNAs.

Analisando-se as fronteiras de Pareto para cada caso (figura 88), é possível notar que o número de soluções com menos neurônios aumentou significativamente para a amostra de 81 nodos-treino.

As figuras 85 e 86 indicam redução da REQM com o aumento dos nodos-treino, porém com aumento nas proximidades dos nodos-âncora.

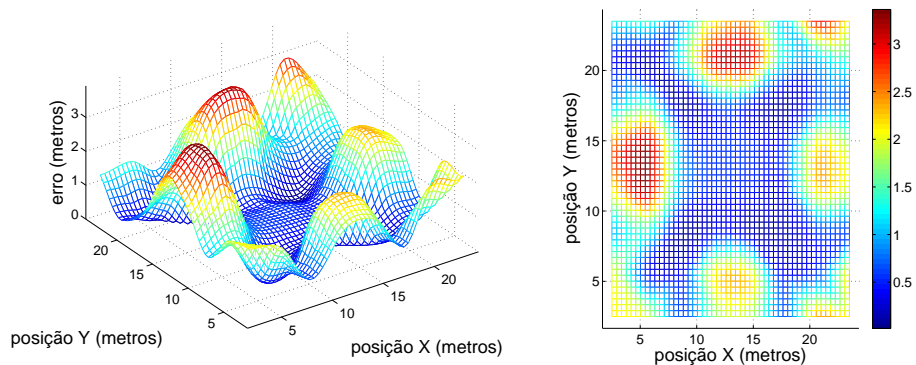
Nas tabelas 30 e 31 é possível analisar os resultados obtidos em termos de REQM e complexidade das RNAs para os casos de menor REQM obtido, ordenados por REQM e tamanho da amostra.

Tabela 30: Resultados para 8 âncoras variando-se o tamanho da amostra de treinamento - ordenação por REQM

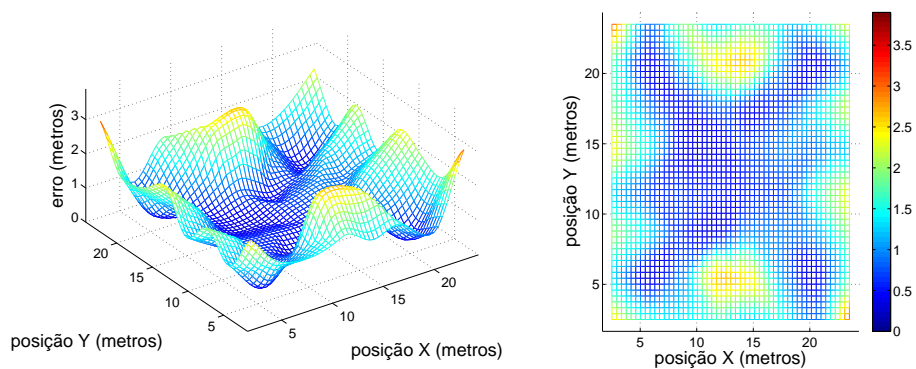
REQM (metros)	Tamanho da Amostra	Camadas Escondidas	Total de Neurônios	Distância Máxima (metros)
0.456	81	1	21	1.394
0.566	64	0	6	1.264
0.766	36	0	6	1.696
0.933	49	1	8	1.653
1.497	25	1	13	3.042
1.562	16	0	5	3.211

Tabela 31: Resultados para 8 âncoras variando-se o tamanho da amostra de treinamento - ordenação por âncoras

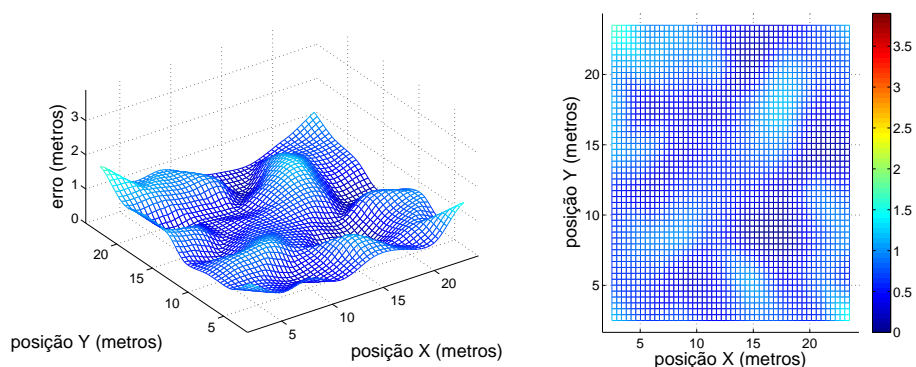
Tamanho da Amostra	REQM (metros)	Total de Neurônios	Camadas Escondidas	Distância Máxima (metros)
16	1.562	5	0	3.211
25	1.497	13	1	3.042
36	0.766	6	0	1.696
49	0.933	8	1	1.653
64	0.566	6	0	1.264
81	0.456	21	1	1.394



(a) Erro em função da posição usando 16 nodos-treino.

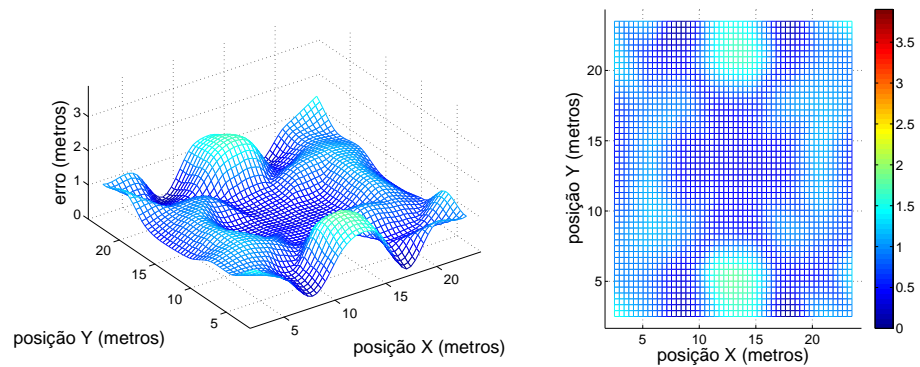


(b) Erro em função da posição usando 25 nodos-treino.

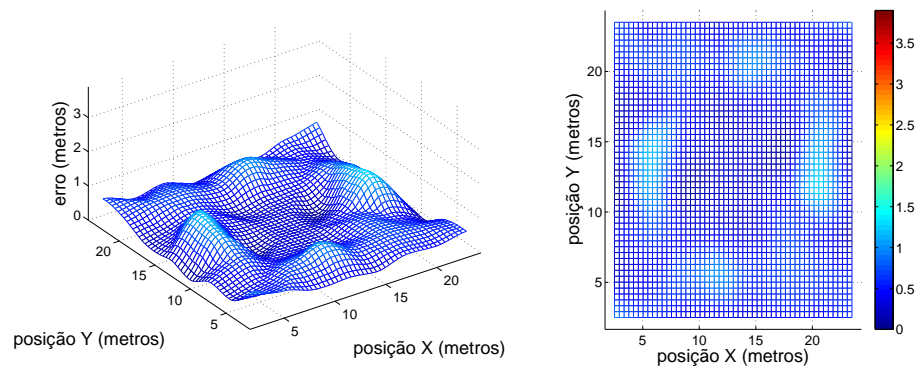


(c) Erro em função da posição usando 36 nodos-treino.

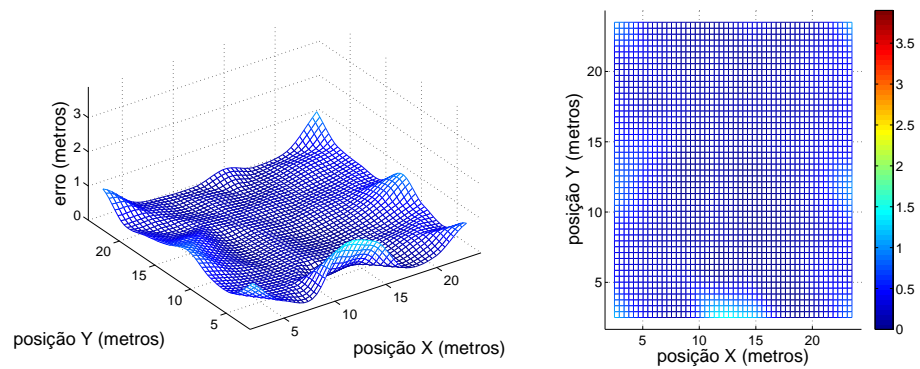
Figura 85: Influência do tamanho da amostra de treinamento no erro de localização para 8 nodos-âncora.



(a) Erro em função da posição usando 49 nodos-treino.

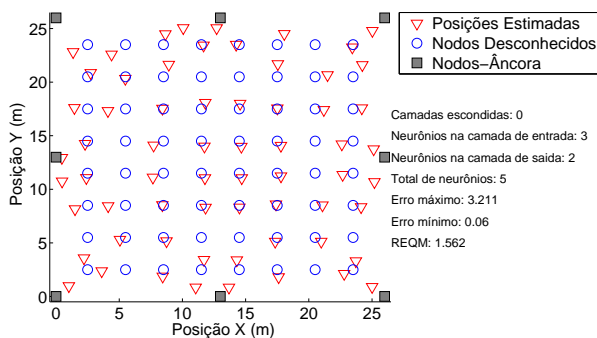


(b) Erro em função da posição usando 64 nodos-treino.

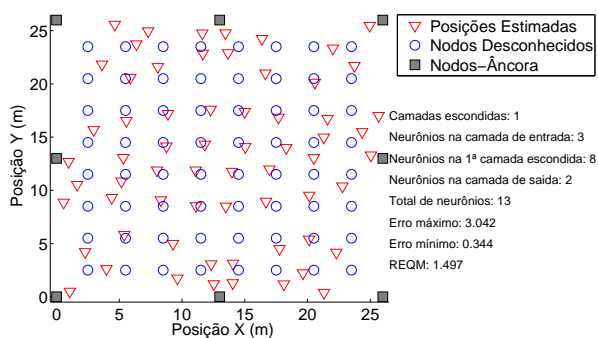


(c) Erro em função da posição usando 81 nodos-treino.

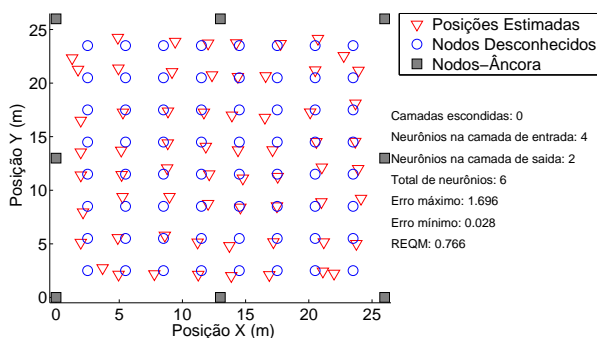
Figura 86: Influência do tamanho da amostra de treinamento no erro de localização para 8 nodos-âncora.



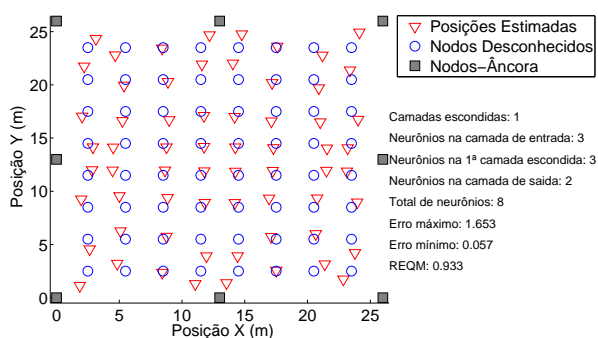
(a) Estimativas - amostra de 16 nodos-treino.



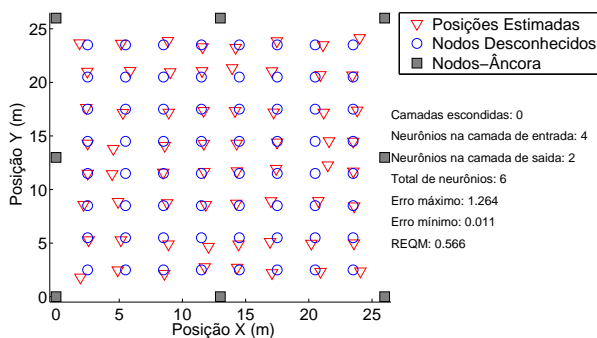
(b) Estimativas - amostra de 25 nodos-treino.



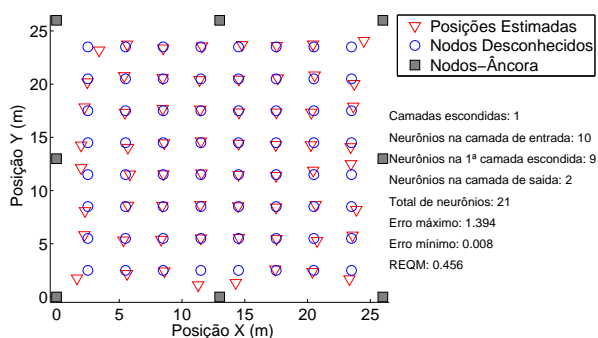
(c) Estimativas - amostra de 36 nodos-treino.



(d) Estimativas - amostra de 49 nodos-treino.

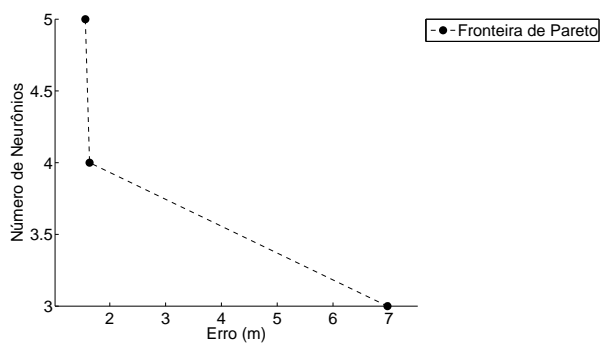


(e) Estimativas - amostra de 64 nodos-treino.

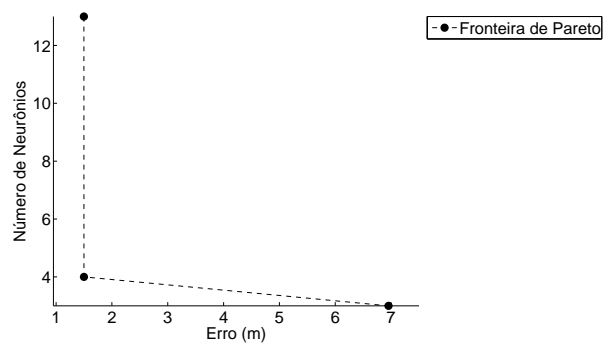


(f) Estimativas - amostra de 81 nodos-treino.

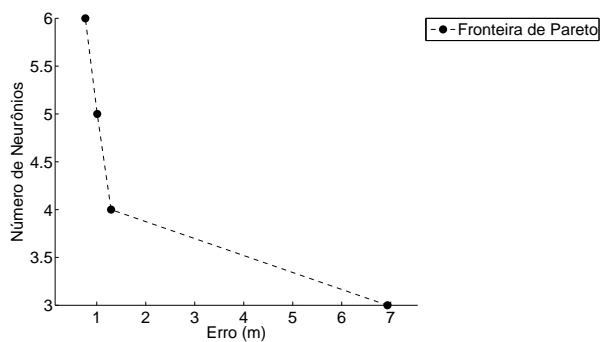
Figura 87: Influência do tamanho da amostra de treinamento nas estimativas de localização para 8 nodos-âncora.



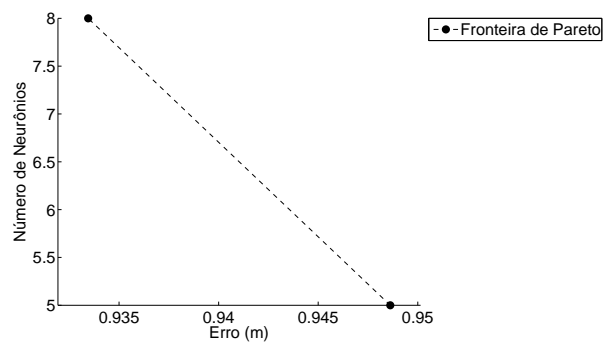
(a) Fronteira de Pareto usando 16 nodos-treino.



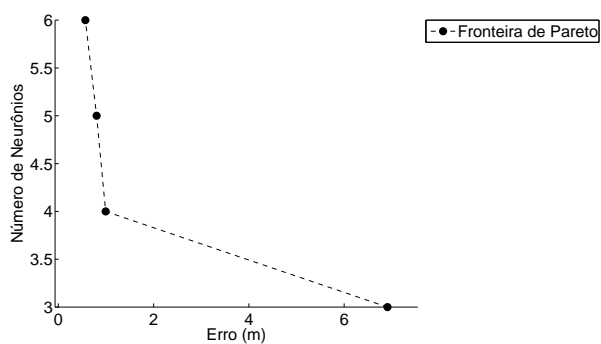
(b) Fronteira de Pareto usando 25 nodos-treino.



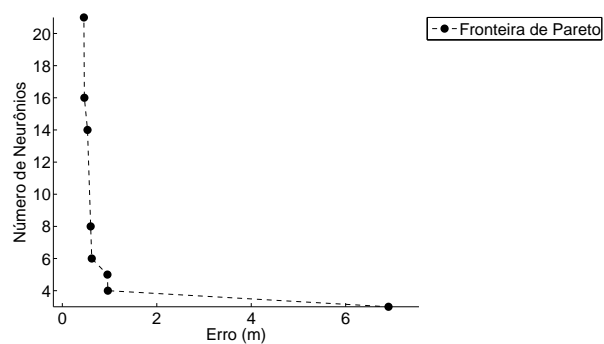
(c) Fronteira de Pareto usando 36 nodos-treino.



(d) Fronteira de Pareto usando 49 nodos-treino.



(e) Fronteira de Pareto usando 64 nodos-treino.



(f) Fronteira de Pareto usando 81 nodos-treino.

Figura 88: Influência do tamanho da amostra de treinamento nas fronteiras de Pareto para 8 nodos-âncora.

4.3.2.7 Topologias com 9 Nodos-Âncora

Como pode ser visto na figura 91, utilizando-se de 9 nodos-âncora e aumentando-se a quantidade de nodos-treino, o erro é reduzido. A amostra composta por 16 nodos-treino segue com o pior desempenho. Outra constatação é a de que 66,66% dos casos não apresentaram camadas escondidas nas RNAs, aumentando em relação ao caso com 8 nodos-âncora, provavelmente devido ao âncora central.

Analisando-se as fronteiras de Pareto para cada caso (figura 92), é possível notar que o número de soluções com menos neurônios apresentou um leve aumento para as amostras maiores.

As figuras 89 e 90 indicam redução da REQM com o aumento dos nodos-treino, porém com aumento nas proximidades dos nodos-âncora. Desta vez, não se observou aumento da REQM na região central da área de teste.

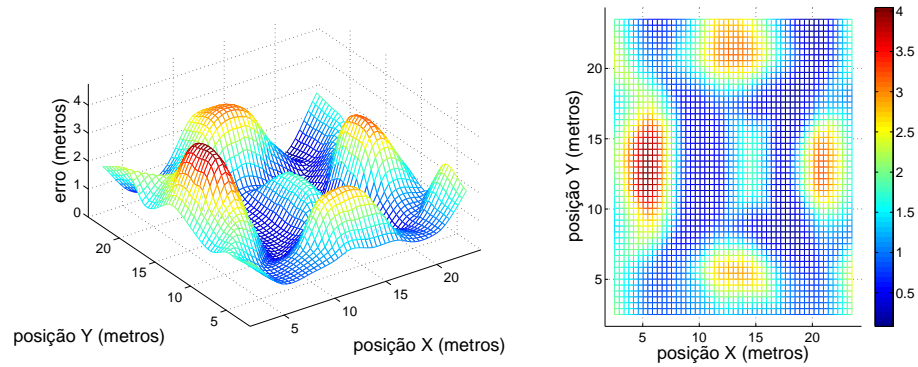
Nas tabelas 32 e 33 é possível analisar os resultados obtidos em termos de REQM e complexidade das RNAs para os casos de menor REQM obtido, ordenados por REQM e tamanho da amostra.

Tabela 32: Resultados para 9 âncoras variando-se o tamanho da amostra de treinamento - ordenação por REQM

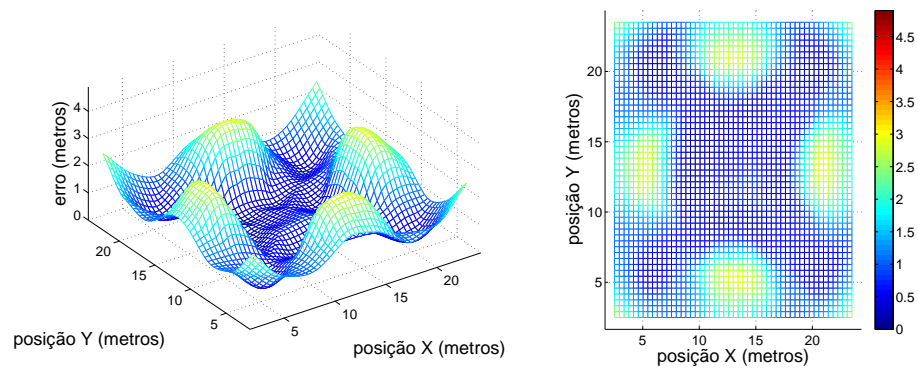
REQM (metros)	Tamanho da Amostra	Camadas Escondidas	Total de Neurônios	Distância Máxima (metros)
0.435	81	0	7	1.538
0.485	64	0	8	1.727
1.014	49	1	7	2.011
1.297	36	0	4	2.243
1.511	25	0	4	2.719
1.709	16	1	7	3.924

Tabela 33: Resultados para 9 âncoras variando-se o tamanho da amostra de treinamento - ordenação por âncoras

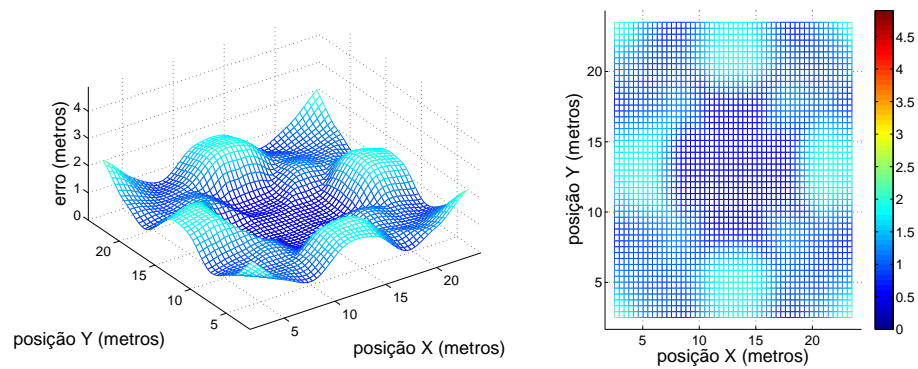
Tamanho da Amostra	REQM (metros)	Total de Neurônios	Camadas Escondidas	Distância Máxima (metros)
16	1.709	7	1	3.924
25	1.511	4	0	2.719
36	1.297	4	0	2.243
49	1.014	7	1	2.011
64	0.485	8	0	1.727
81	0.435	7	0	1.538



(a) Erro em função da posição usando 16 nodos-treino.

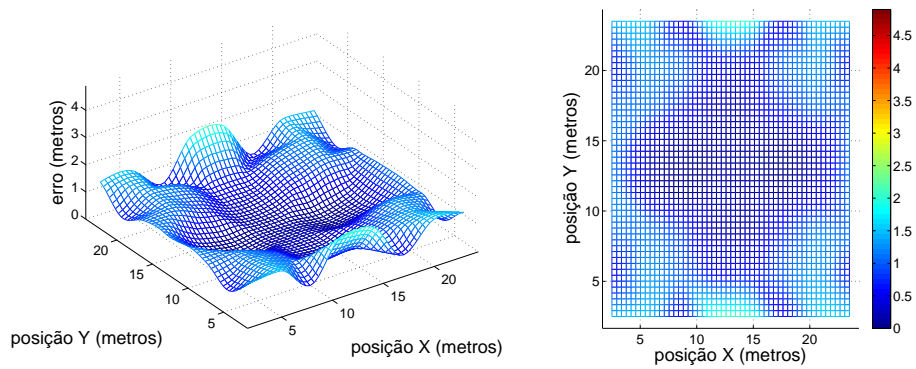


(b) Erro em função da posição usando 25 nodos-treino.

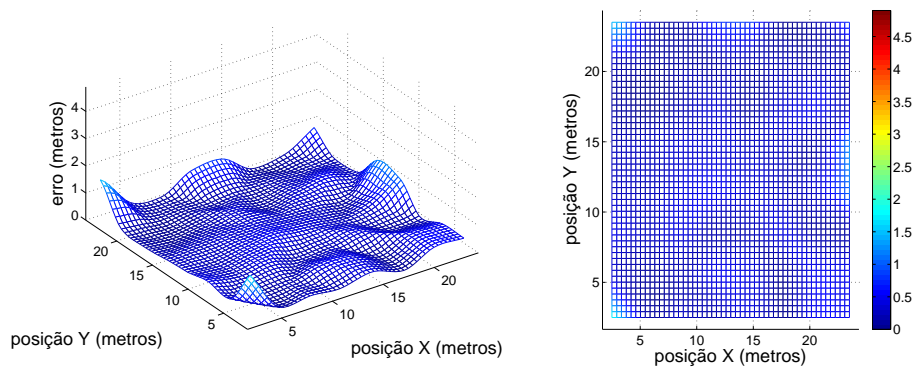


(c) Erro em função da posição usando 36 nodos-treino.

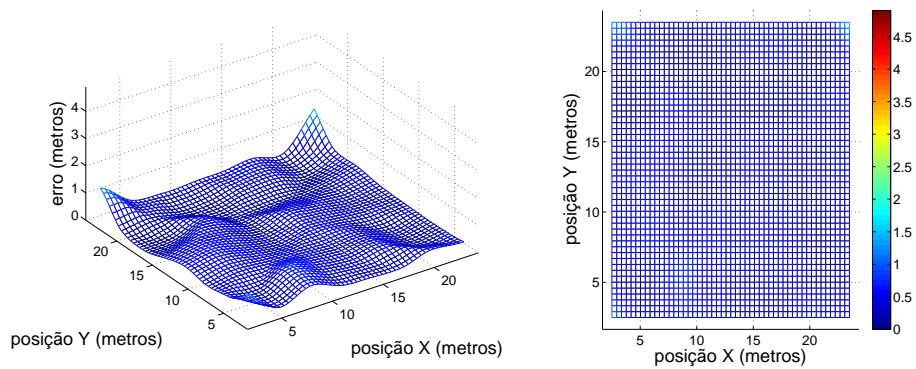
Figura 89: Influência do tamanho da amostra de treinamento no erro de localização para 9 nodos-âncora.



(a) Erro em função da posição usando 49 nodos-treino.

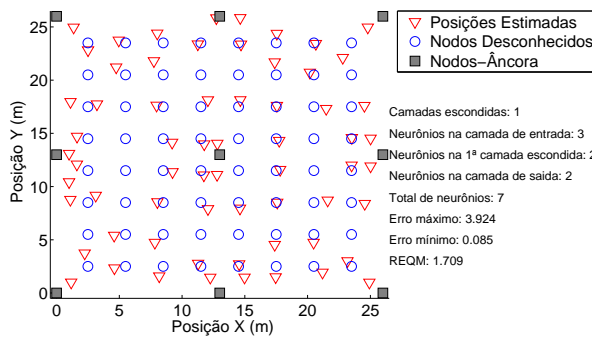


(b) Erro em função da posição usando 64 nodos-treino.

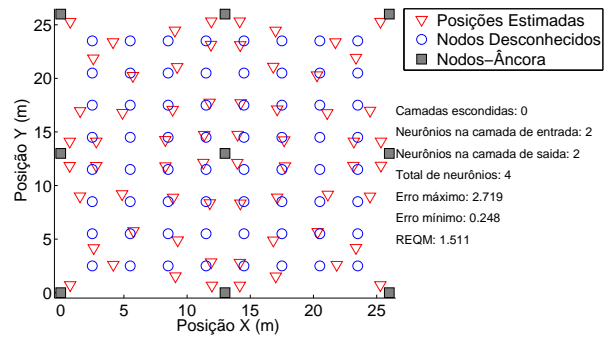


(c) Erro em função da posição usando 81 nodos-treino.

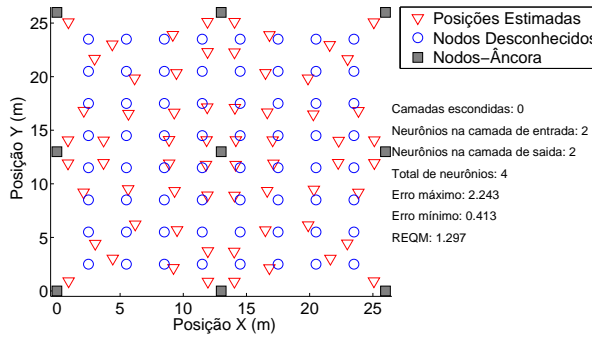
Figura 90: Influência do tamanho da amostra de treinamento no erro de localização para 9 nodos-âncora.



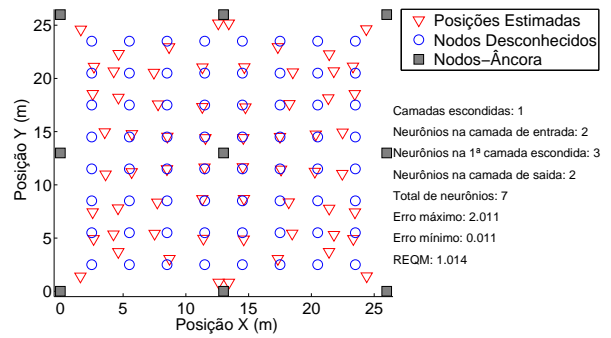
(a) Estimativas - amostra de 16 nodos-treino.



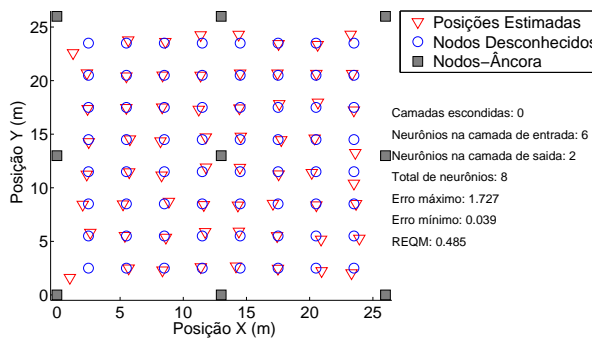
(b) Estimativas - amostra de 25 nodos-treino.



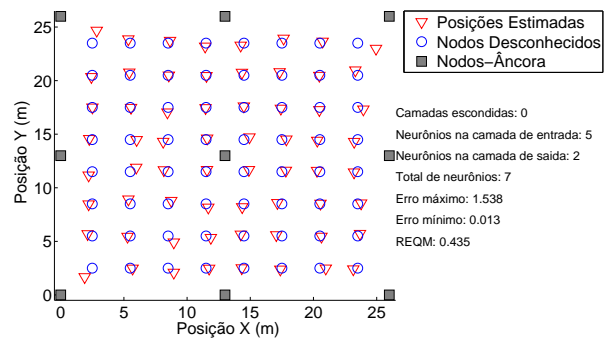
(c) Estimativas - amostra de 36 nodos-treino.



(d) Estimativas - amostra de 49 nodos-treino.

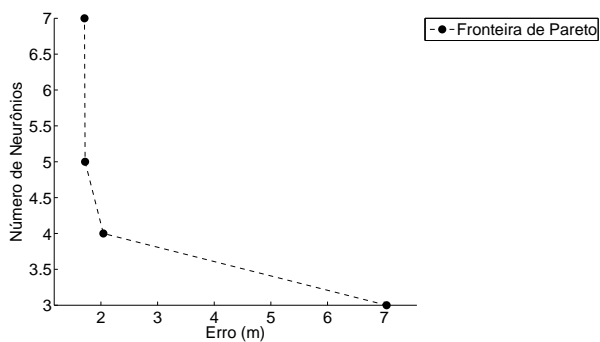


(e) Estimativas - amostra de 64 nodos-treino.

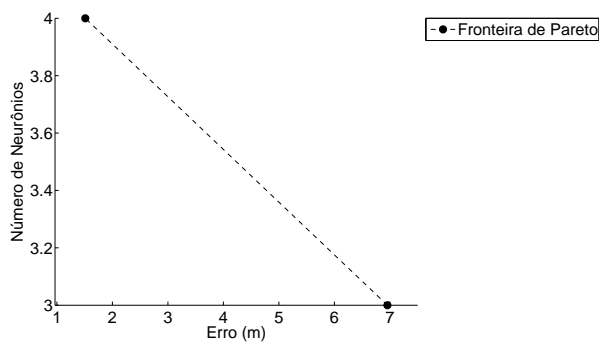


(f) Estimativas - amostra de 81 nodos-treino.

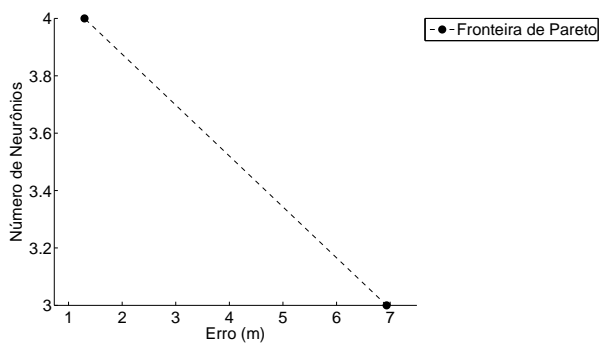
Figura 91: Influência do tamanho da amostra de treinamento nas estimativas de localização para 9 nodos-âncora.



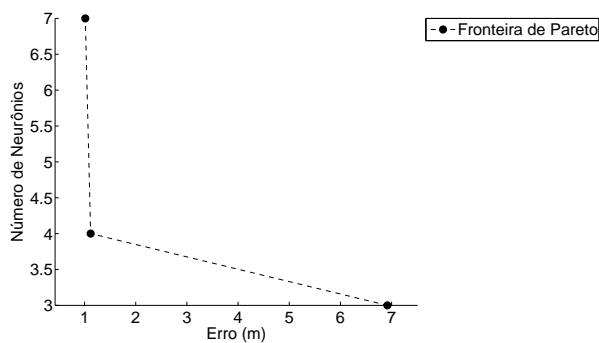
(a) Fronteira de Pareto usando 16 nodos-treino.



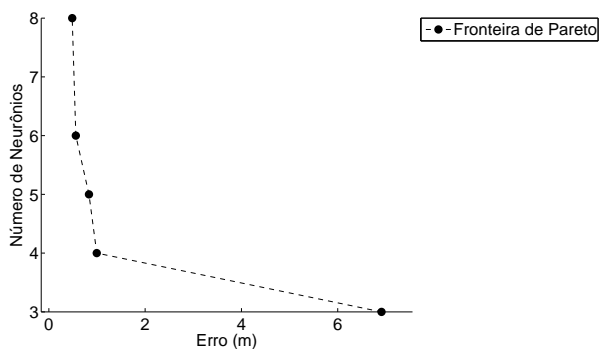
(b) Fronteira de Pareto usando 25 nodos-treino.



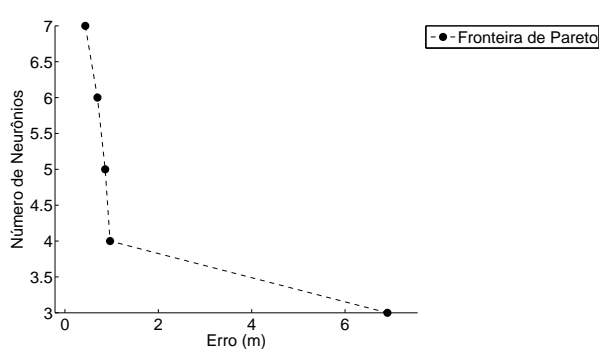
(c) Fronteira de Pareto usando 36 nodos-treino.



(d) Fronteira de Pareto usando 49 nodos-treino.



(e) Fronteira de Pareto usando 64 nodos-treino.



(f) Fronteira de Pareto usando 81 nodos-treino.

Figura 92: Influência do tamanho da amostra de treinamento nas fronteiras de Pareto para 9 nodos-âncora.

4.3.2.8 Análise do REQM em função do tamanho das amostras

Ao longo das simulações, foi possível inferir que o aumento na quantidade de nodos nas amostras de treinamento, na maioria dos casos, garante redução no erro das estimativas de localização (figura 93). É visível também que o nível de erro para amostras com 16 nodos-treino é discrepante em relação aos demais tamanhos de amostras, sendo maior.

Foi notável que nas topologias com apenas 3 nodos-âncora o erro foi menor e também apresentou redução menor quando eram aumentadas as amostras. Como as RNAs aproximam uma função para efetuar a localização dos nodos, provavelmente algo parecido com uma trilateração poderia estar sendo utilizado. Em casos de localização em duas dimensões, são necessários apenas 3 nodos-âncora para determinar a posição de um nodo desconhecido. Desta forma, arranjos com mais de 3 nodos-âncora poderiam confundir as RNAs, visto que em todas as simulações, incondicionalmente, sempre foram utilizadas todas as entradas simultaneamente. Esta é uma hipótese sobre a causa deste comportamento das RNAs.

As topologias com 8 nodos-âncora apresentaram variações mais bruscas e disformes no erro de localização, indicando que esta não é uma boa escolha a ser adotada. Como a REQM apresenta aumento na proximidade de nodos-âncora, conclui-se que quanto menos nodos-âncoras, menores as regiões com altos índices de erro.

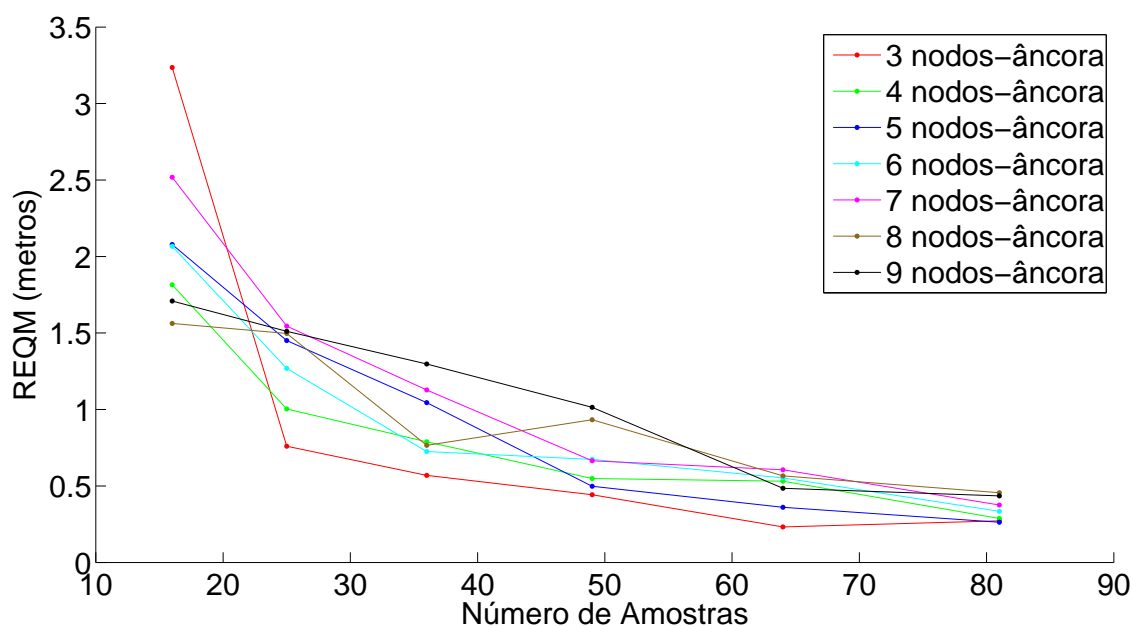


Figura 93: Comportamento da REQM em função do aumento da amostra de treinamento, para todos os números de nodos-âncora simulados.

4.4 Testes complementares

Todas as simulações de análise complementar foram feitas e comparadas aos casos de mesmo número de nodos-âncora mostrados nos testes principais, de modo a ter uma referência. Em geral, foram tomadas duas amostras de treinamento, sendo uma com poucos nodos-treino e outra com muitos. Desta forma foi possível observar se o comportamento apresentado era independente ou não do tamanho das amostras de treinamento.

Assim como mencionado ao iniciar a exposição dos resultados principais, a fronteira de Pareto mantém o seu papel nos testes complementares, informando a relação custo-benefício envolvida em cada simulação.

Já o caso de variação nas posições dos nodos-treino não foi comparado com algum dos testes principais, pois esse teste foi feito usando o arranjo alternativo de 3 nodos-âncora.

4.4.1 Variação na Posição de Nodos-Âncora e Nodos-Treino

A variação na posição dos nodos foi feita de modo a não haver repetições de coordenadas, visando amostrar uma parcela maior da área de teste. O objetivo destas variações é observar o comportamento das RNAs perante a falta de padrão no posicionamento dos nodos.

4.4.1.1 Arranjo Alternativo - 3 Nodos-Âncora

Foram simuladas duas topologias com 3 nodos-âncora deslocados em relação às suas posições originais (testes principais), sendo uma com 36 e outra com 81 nodos-treino. Conforme pode ser visto nas figuras 94 e 95, verifica-se uma queda no desempenho para a topologia com 36 nodos-treino e uma leve melhora para a topologia com 81 nodos-treino.

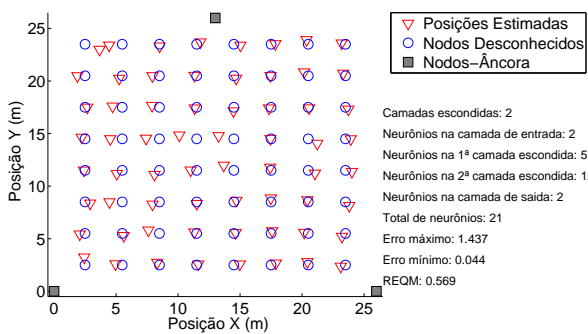
Observando-se as fronteiras de Pareto (figuras 94(c) e 94(d)) para a amostra de 36 nodos-treino, é possível notar a existência de uma solução ótima a mais para o arranjo convencional de nodos-âncora. Além disso, percebe-se um pequeno aumento no erro para as soluções oriúndas do arranjo alternativo de nodos-âncora.

Nas figuras 94(e) e 94(f) é possível notar que o maior índice de erro na área central ocorreu para o arranjo normal de nodos-âncora.

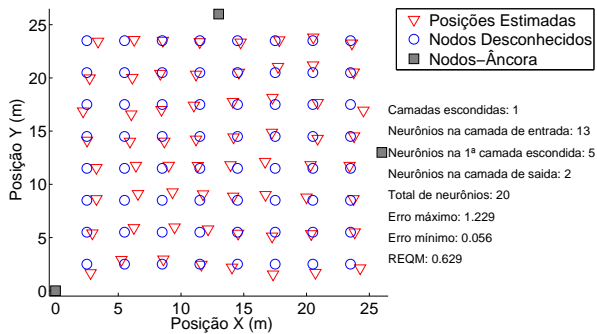
A topologia que utilizou 81 nodos-treino apresentou melhores estimativas de localização do que o caso que utilizou apenas 36. A REQM da topologia que aplicou o arranjo diferenciado de nodos-âncora foi menor, somado ao fato de possuir um neurônio artificial a menos em sua estrutura.

A partir da observação das fronteiras de Pareto (figuras 95(c) e 95(d)), é possível notar uma solução alternativa a mais presente na topologia com arranjo tradicional de nodos-âncora.

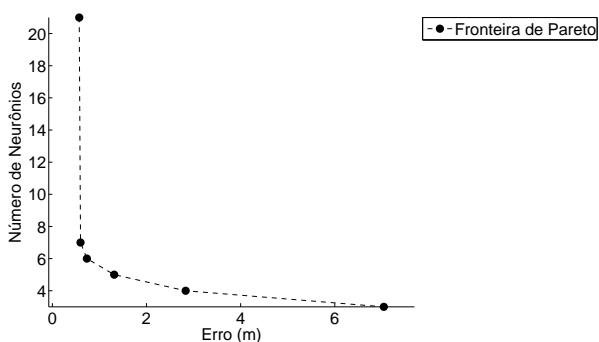
Já nas figuras 95(e) e 95(f) pode-se perceber que o maior pico na REQM ocorreu na topologia tradicional.



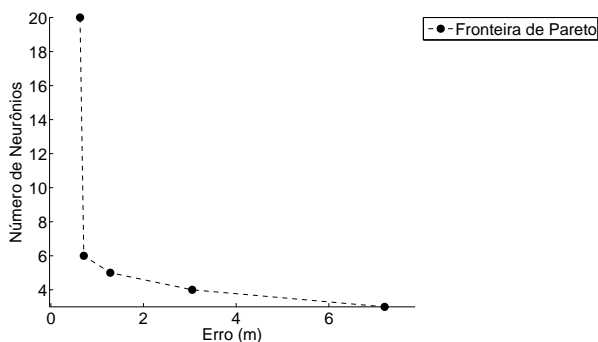
(a) Estimativas com 3 âncoras no arranjo usual.



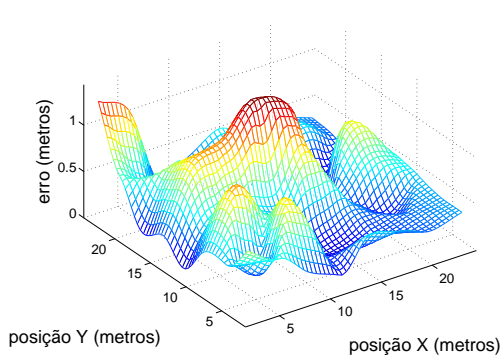
(b) Estimativas com 3 âncoras deslocados.



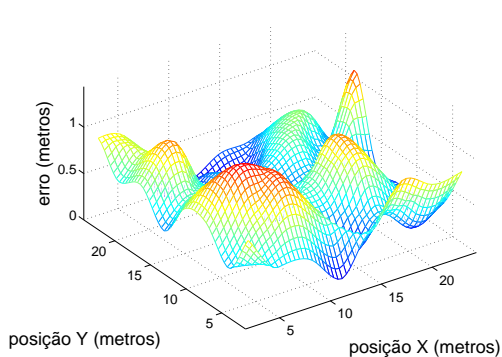
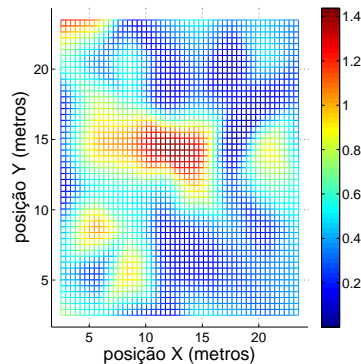
(c) Fronteira de Pareto - 3 âncoras no arranjo usual.



(d) Fronteira de Pareto - 3 âncoras deslocados.



(e) Erro com 3 âncoras no arranjo usual.



(f) Erro com 3 âncoras deslocados.

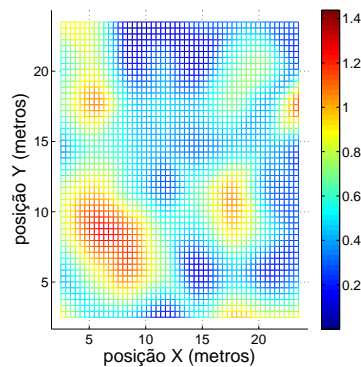
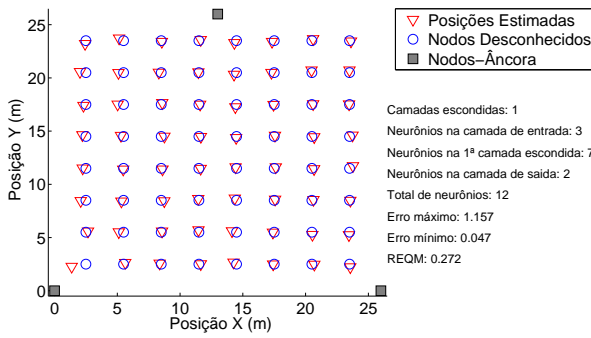
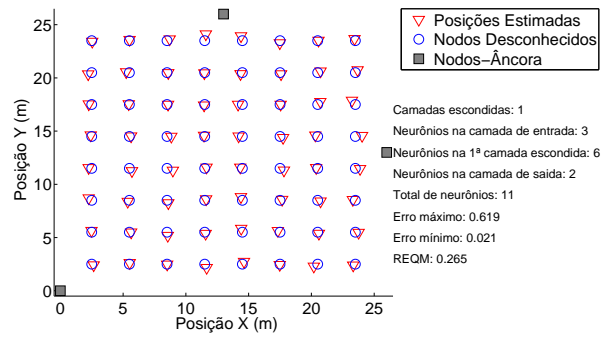


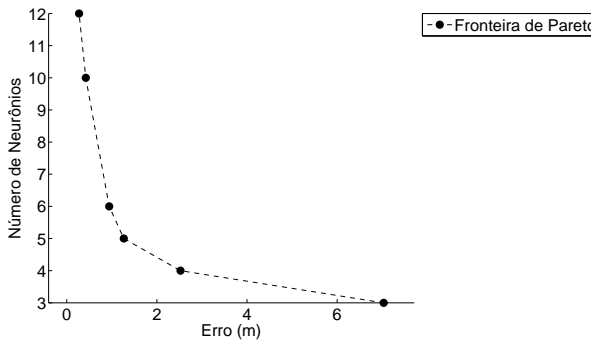
Figura 94: Influência da posição de 3 âncoras usando 36 nodos-treino.



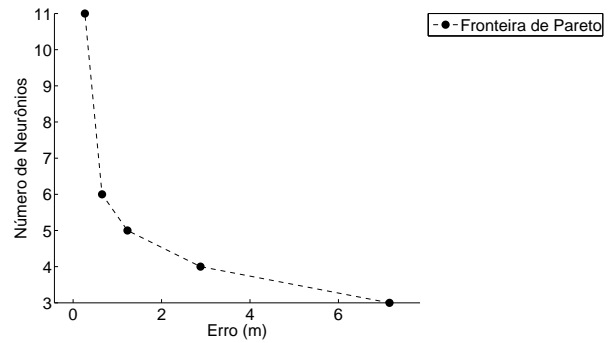
(a) Estimativas com 3 âncoras no arranjo usual.



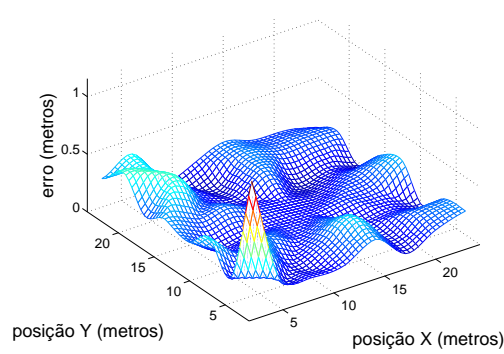
(b) Estimativas com 3 âncoras deslocados.



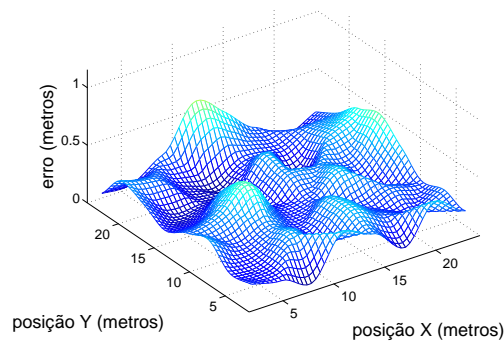
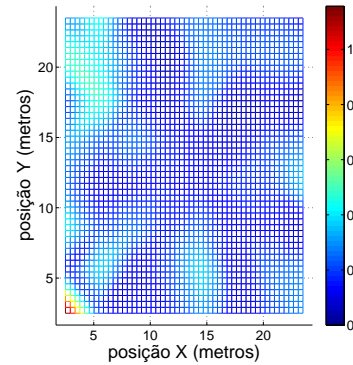
(c) Fronteira de Pareto - 3 âncoras no arranjo usual.



(d) Fronteira de Pareto - 3 âncoras deslocados.



(e) Erro com 3 âncoras no arranjo usual.



(f) Erro com 3 âncoras deslocados.

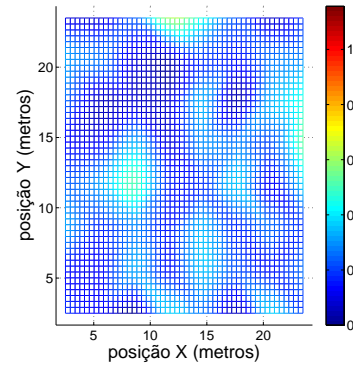


Figura 95: Influência da posição de 3 âncoras usando 81 nodos-treino.

4.4.1.2 Arranjo Alternativo - 4 Nodos-Âncora

Foram simuladas duas topologias com 4 nodos-âncora deslocados em relação às suas posições originais (testes principais), sendo uma com 36 e outra com 81 nodos-treino. Conforme pode ser visto nas figuras 96 e 97, de um modo geral, verifica-se uma queda no desempenho independente do tamanho da amostra.

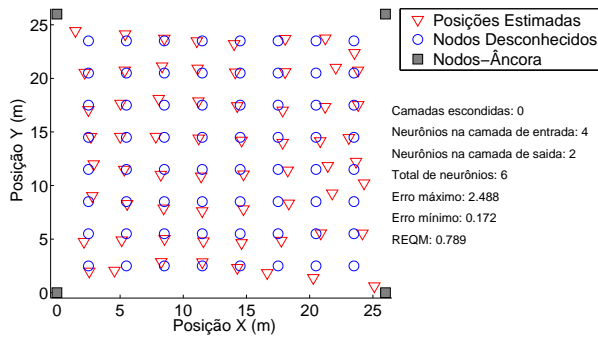
Observando-se as fronteiras de Pareto (figuras 96(c) e 96(d)) para a amostra de 36 nodos-treino, é possível notar a existência do dobro de soluções ótimas para o arranjo convencional frente ao alternativo.

Nas figuras 96(e) e 96(f) é possível notar que o maior índice de erro na área central ocorreu para o arranjo alternativo de nodos-âncora.

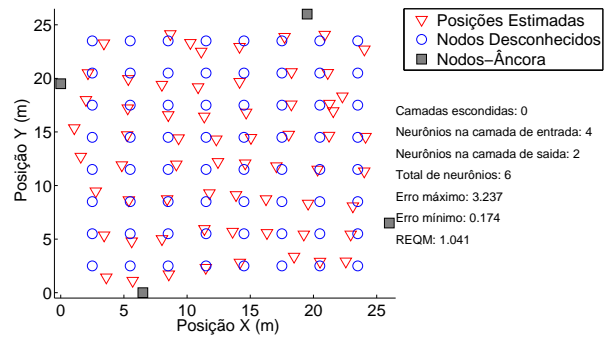
A topologia que utilizou 81 nodos-treino também apresentou piores estimativas de localização do que o caso dotado de arranjo de nodos-âncora usual. Porém, sua estrutura para a menor REQM apresentou um número menor de neurônios artificiais.

A partir da observação das fronteiras de Pareto (figuras 97(c) e 97(d)), é possível notar que todas as soluções ótimas referentes à topologia normal apresentaram resultados com REQM abaixo de 1 metro.

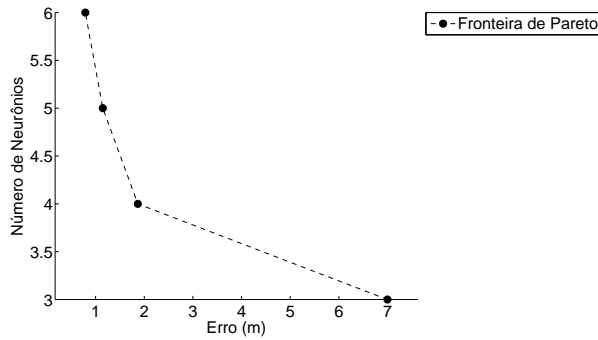
Já nas figuras 97(e) e 97(f) pode-se perceber que o maior pico na REQM ocorreu na topologia tradicional, apesar desta possuir menor erro médio.



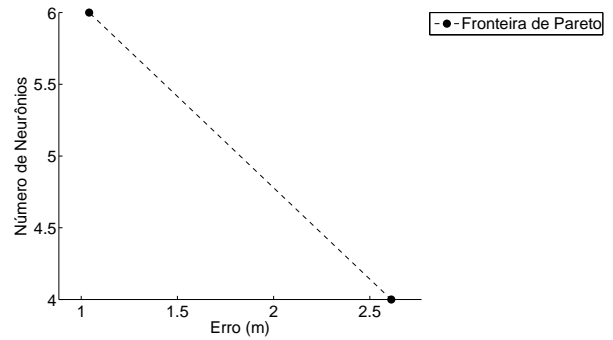
(a) Estimativas com 4 âncoras no arranjo usual.



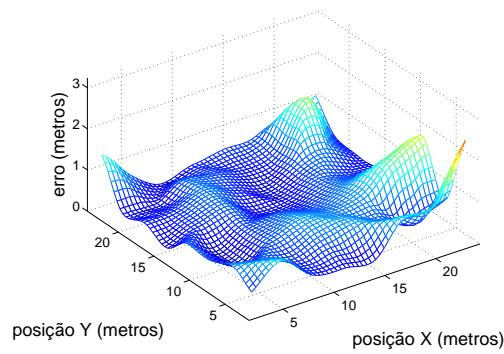
(b) Estimativas com 4 âncoras deslocados.



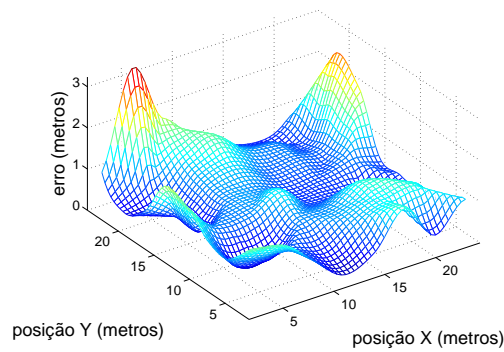
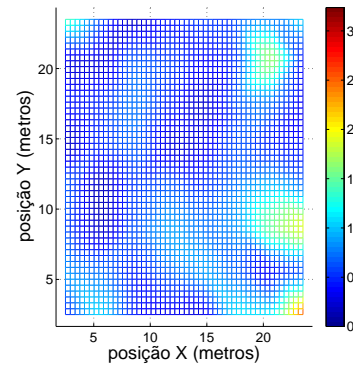
(c) Fronteira de Pareto - 4 âncoras no arranjo usual.



(d) Fronteira de Pareto - 4 âncoras deslocados.



(e) Erro com 4 âncoras no arranjo usual.



(f) Erro com 4 âncoras deslocados.

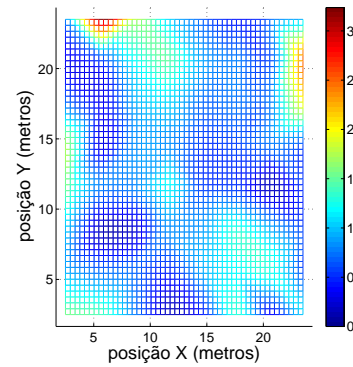
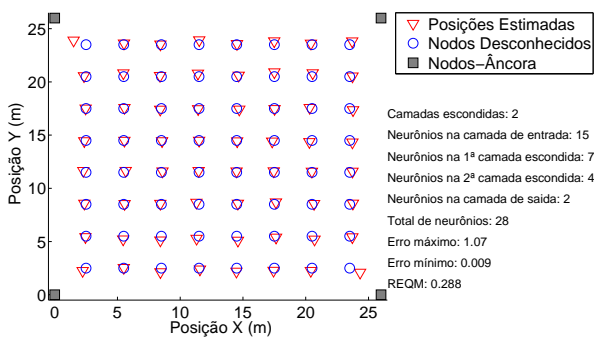
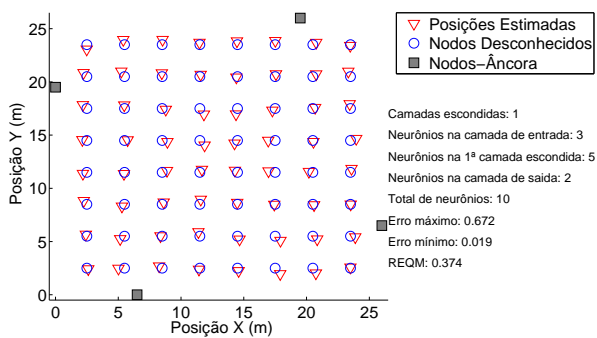


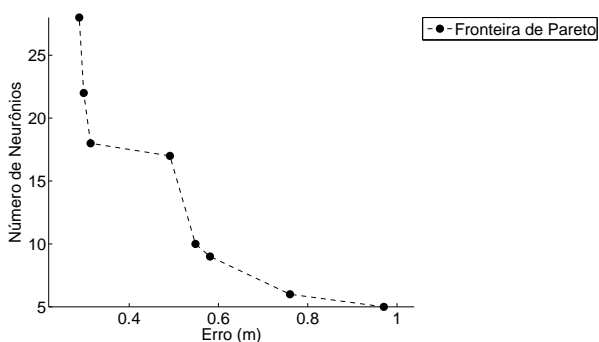
Figura 96: Influência da posição de 4 âncoras usando 36 nodos-treino.



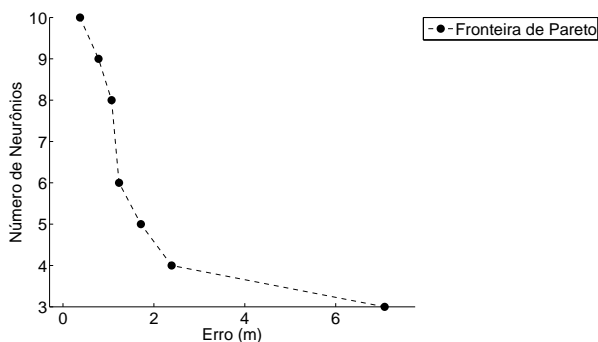
(a) Estimativas com 4 âncoras no arranjo usual.



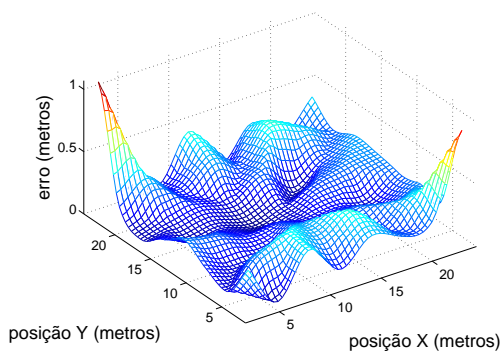
(b) Estimativas com 4 âncoras deslocados.



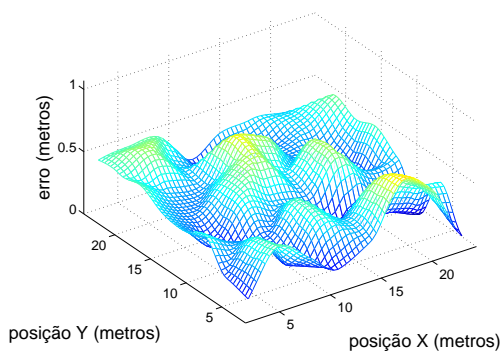
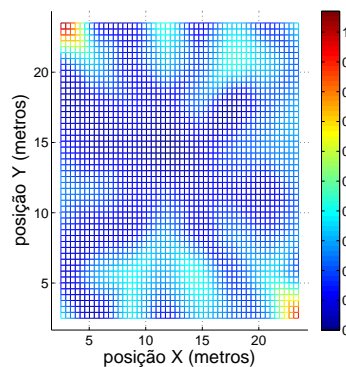
(c) Fronteira de Pareto - 4 âncoras no arranjo usual.



(d) Fronteira de Pareto - 4 âncoras deslocados.



(e) Erro com 4 âncoras no arranjo usual.



(f) Erro com 4 âncoras deslocados.

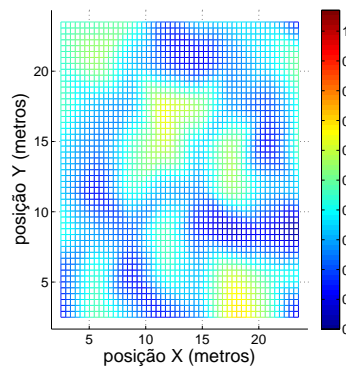


Figura 97: Influência da posição de 4 âncoras usando 81 nodos-treino.

4.4.1.3 Arranjo Alternativo - 5 Nodos-Âncora

Foram simuladas duas topologias com 5 nodos-âncora deslocados em relação às suas posições originais (testes principais), sendo uma com 36 e outra com 81 nodos-treino. Conforme pode ser visto nas figuras 98 e 99, de um modo geral, verifica-se uma queda no desempenho independente do tamanho da amostra, além de um aumento na quantidade de neurônios artificiais.

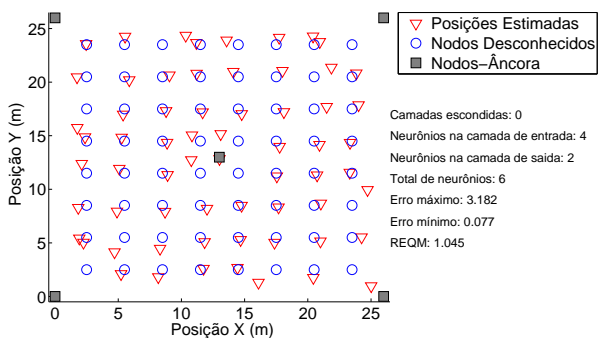
Observando-se as fronteiras de Pareto (figuras 98(c) e 98(d)) para a amostra de 36 nodos-treino, é possível notar que todas as soluções derivadas do arranjo convencional de nodos-âncora possuem 6 neurônios artificiais ou menos.

Nas figuras 98(e) e 98(f) é possível notar que o o arranjo com nodos-âncora deslocados obteve os maiores índices de erro.

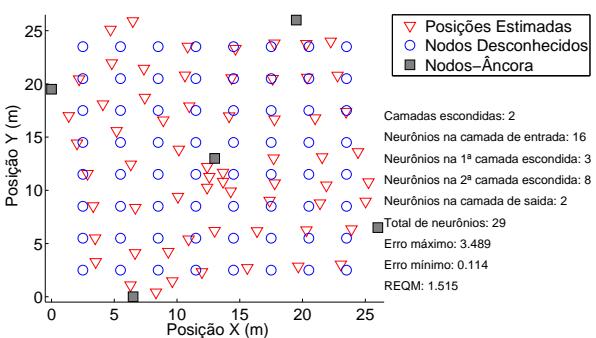
A topologia que utilizou 81 nodos-treino também apresentou piores estimativas de localização do que o caso dotado de arranjo de nodos-âncora usual. Porém, esta rede apresentou um erro mínimo menor.

A partir da observação das fronteiras de Pareto (figuras 99(c) e 99(d)), é possível notar que os resultados gerados pelas soluções ótimas são muito semelhantes.

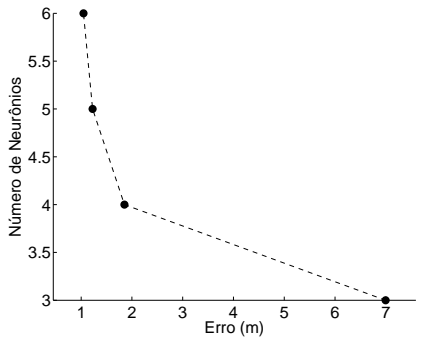
Nas figuras 99(e) e 99(f), com o aumento do tamanho da amostra, ficou mais visível a influência negativa do nodo-âncora central.



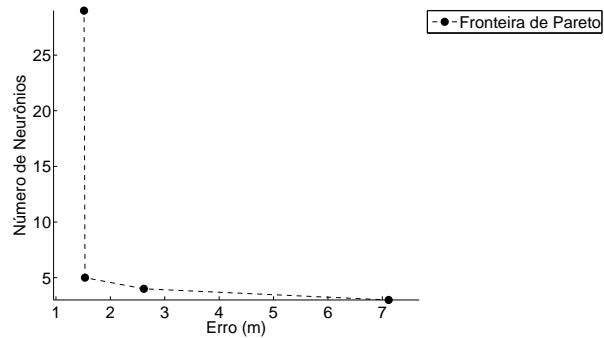
(a) Estimativas com 5 âncoras no arranjo usual.



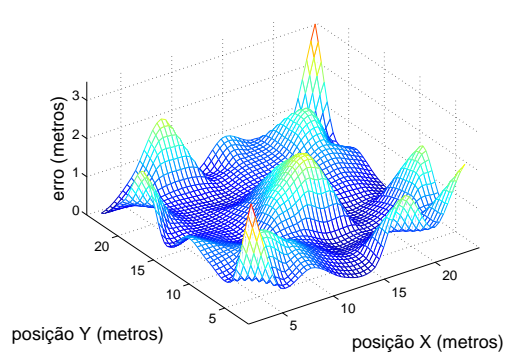
(b) Estimativas com 5 âncoras deslocados.



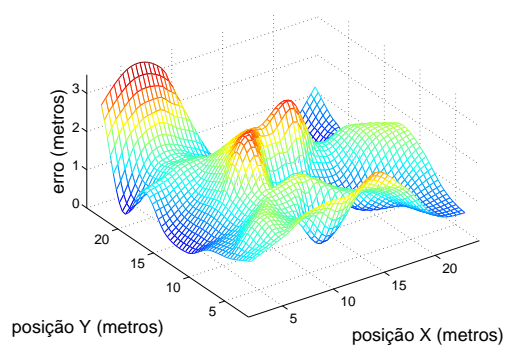
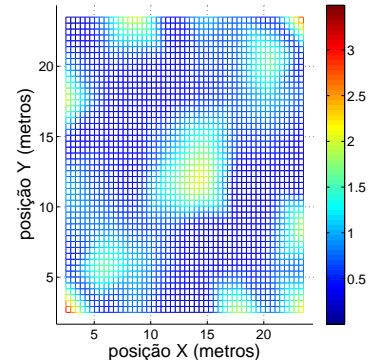
(c) Fronteira de Pareto - 5 âncoras no arranjo usual.



(d) Fronteira de Pareto - 5 âncoras deslocados.



(e) Erro com 5 âncoras no arranjo usual.



(f) Erro com 5 âncoras deslocados.

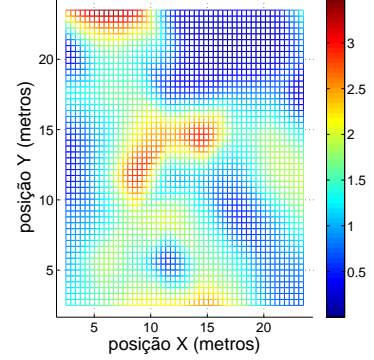
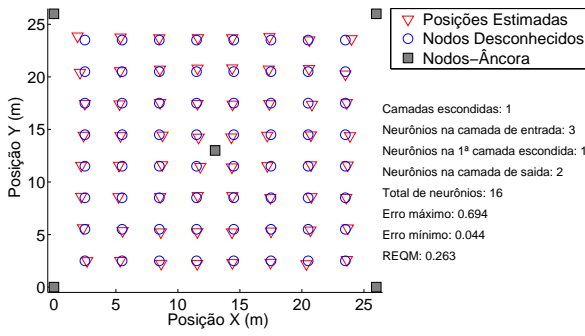
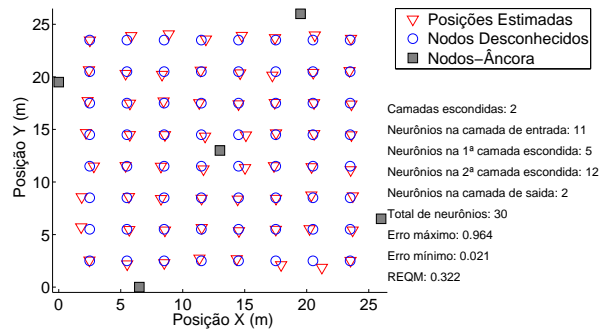


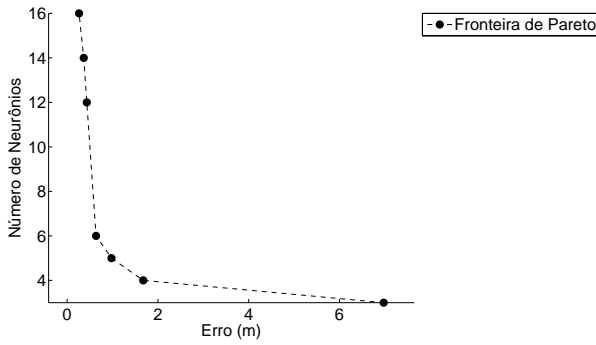
Figura 98: Influência da posição de 5 âncoras usando 36 nodos-treino.



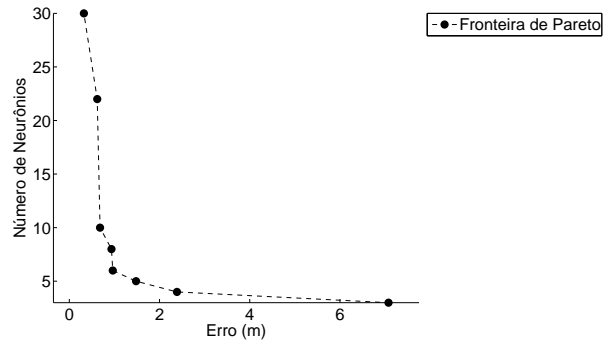
(a) Estimativas com 5 âncoras no arranjo usual.



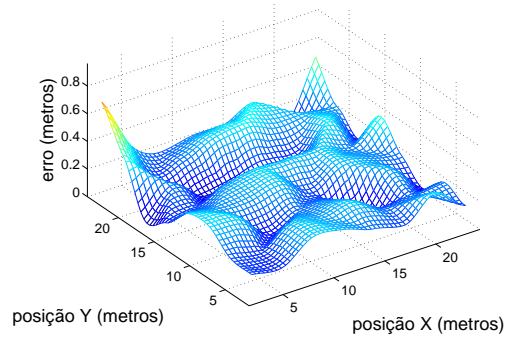
(b) Estimativas com 5 âncoras deslocados.



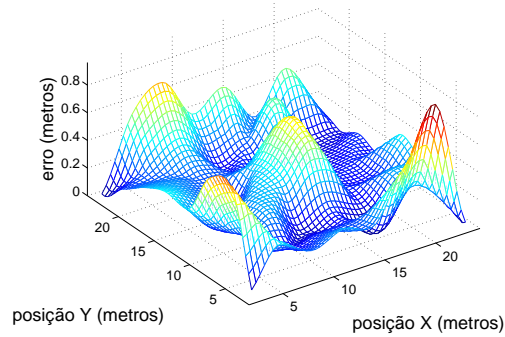
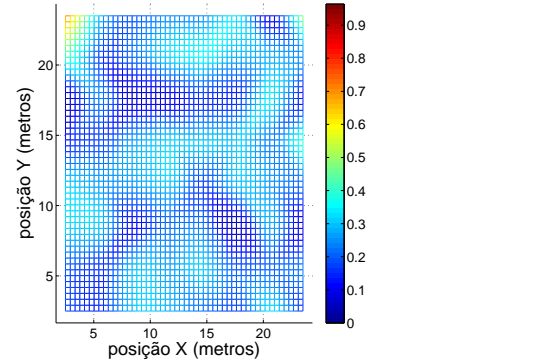
(c) Fronteira de Pareto - 5 âncoras no arranjo usual.



(d) Fronteira de Pareto - 5 âncoras deslocados.



(e) Erro com 5 âncoras no arranjo usual.



(f) Erro com 5 âncoras deslocados.

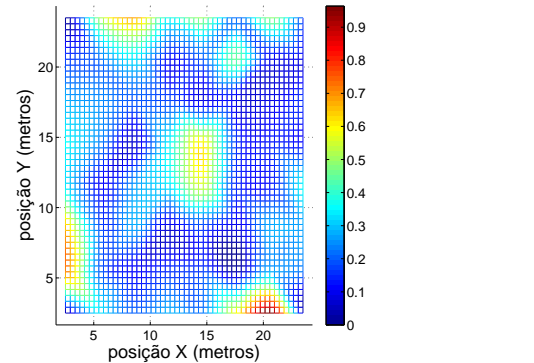


Figura 99: Influência da posição de 5 âncoras usando 81 nodos-treino.

4.4.1.4 Arranjo Alternativo - 6 Nodos-Âncora

Foram simuladas duas topologias com 6 nodos-âncora deslocados em relação às suas posições originais (testes principais), sendo uma com 36 e outra com 81 nodos-treino. Conforme pode ser visto nas figuras 100 e 101, de um modo geral, verifica-se uma grande queda no desempenho, especialmente para a amostra de 36 nodos-treino.

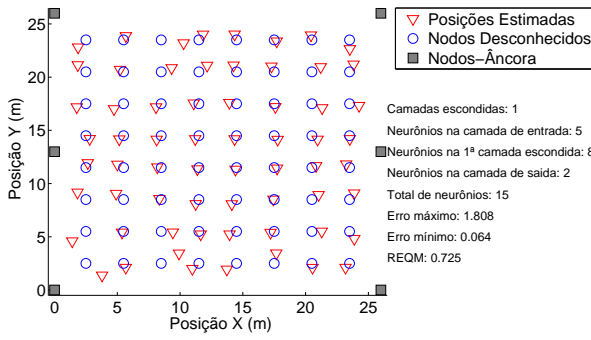
Observando-se as fronteiras de Pareto (figuras 100(c) e 100(d)) para a amostra de 36 nodos-treino, é possível notar que a maioria das soluções derivadas do arranjo convencional produziram resultados com erro menor do que 2 metros.

Nas figuras 100(e) e 100(f) é possível notar que o arranjo com nodos-âncora deslocados obteve os maiores índices de erro, sendo possível notar dois grandes focos nas áreas ocupadas pelos nodos-âncora centrais.

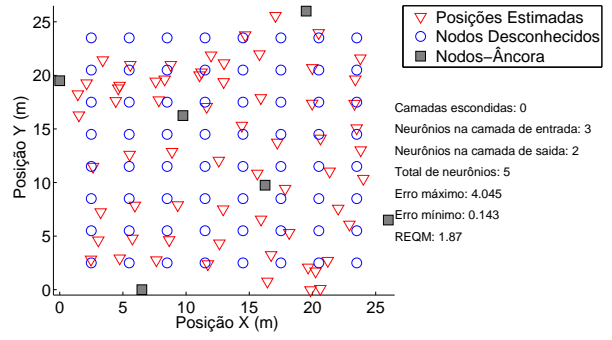
A topologia deslocada que utilizou 81 nodos-treino também apresentou piores estimativas de localização do que o caso dotado de arranjo de nodos-âncora usual. Além disso, o número de neurônios artificiais de suas soluções foi mais elevado.

Observando-se as fronteiras de Pareto (figuras 101(c) e 101(d)) para a amostra de 81 nodos-treino, é possível notar que todas as soluções derivadas do arranjo convencional produziram resultados com menos neurônios artificiais.

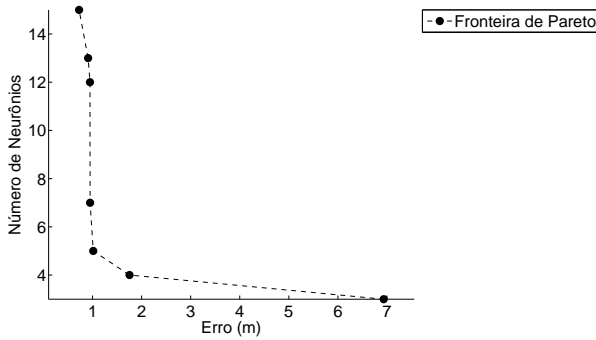
Nas figuras 101(e) e 101(f), com o aumento do tamanho da amostra, ficou mais visível a influência negativa dos nodos-âncora centrais.



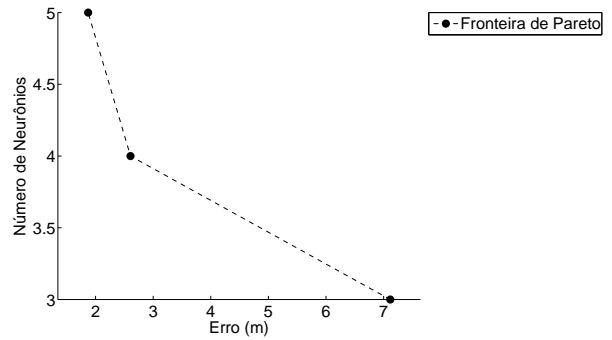
(a) Estimativas com 6 âncoras no arranjo usual.



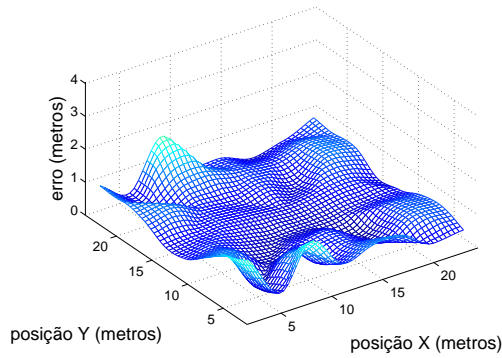
(b) Estimativas com 6 âncoras deslocados.



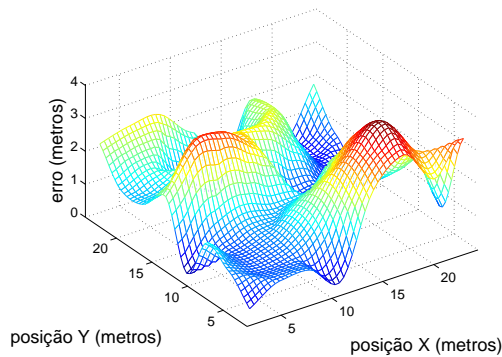
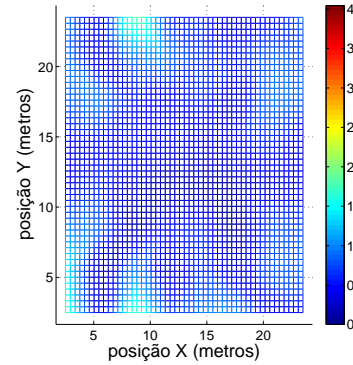
(c) Fronteira de Pareto - 6 âncoras no arranjo usual.



(d) Fronteira de Pareto - 6 âncoras deslocados.



(e) Erro com 6 âncoras no arranjo usual.



(f) Erro com 6 âncoras deslocados.

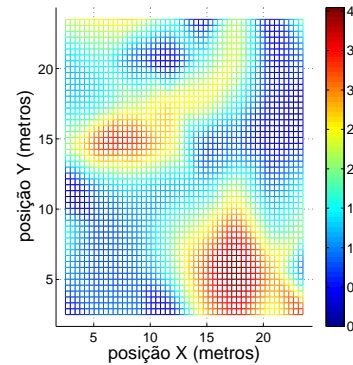
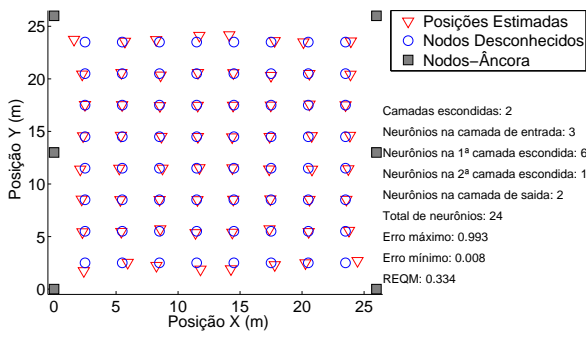
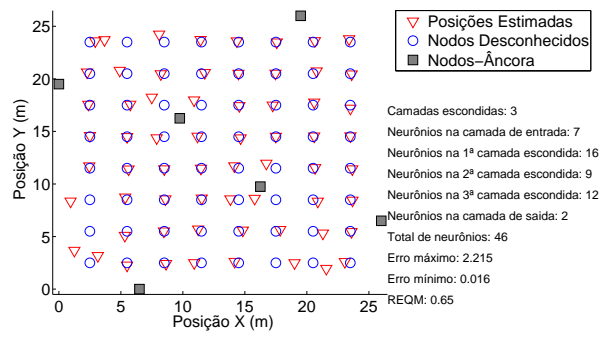


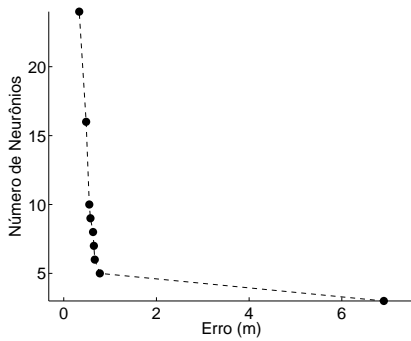
Figura 100: Influência da posição de 6 âncoras usando 36 nodos-treino.



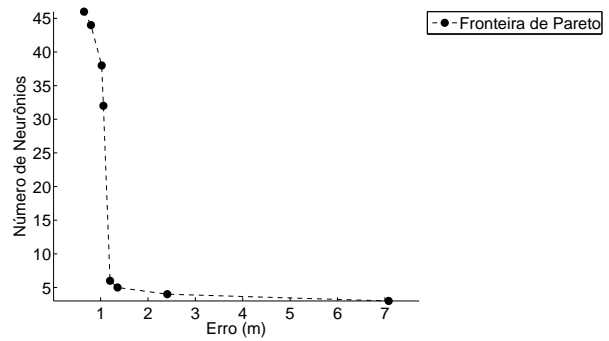
(a) Estimativas com 6 âncoras no arranjo usual.



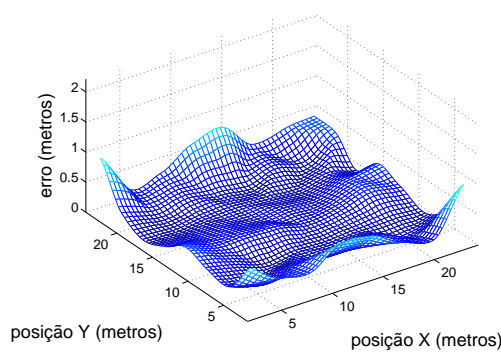
(b) Estimativas com 6 âncoras deslocados.



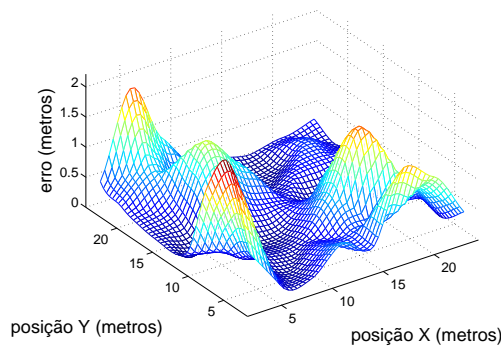
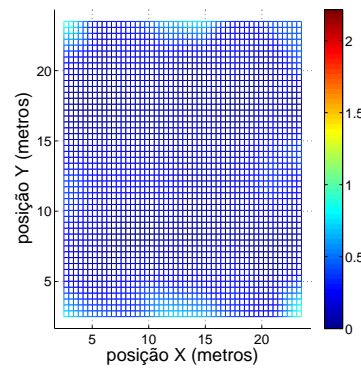
(c) Fronteira de Pareto - 6 âncoras no arranjo usual.



(d) Fronteira de Pareto - 6 âncoras deslocados.



(e) Erro com 6 âncoras no arranjo usual.



(f) Erro com 6 âncoras deslocados.

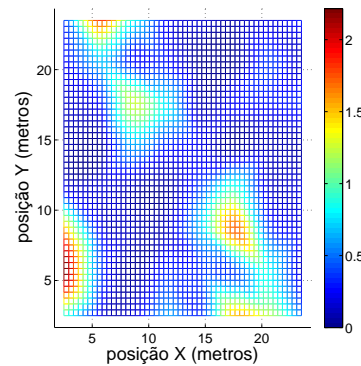


Figura 101: Influência da posição de 6 âncoras usando 81 nodos-treino.

4.4.1.5 Arranjo Alternativo - 8 Nodos-Âncora em Círculo

Visando testar o desempenho das RNAs com arranjos de nodos-âncora em formatos circulares, foram simuladas duas topologias sendo que a primeira utilizou-se de 25 nodos-treino e a segunda 81 nodos-treino. Observou-se uma melhora no desempenho da RSSF com 25 nodos-treino e uma piora para 81 nodos-treino (figuras 102 e 103).

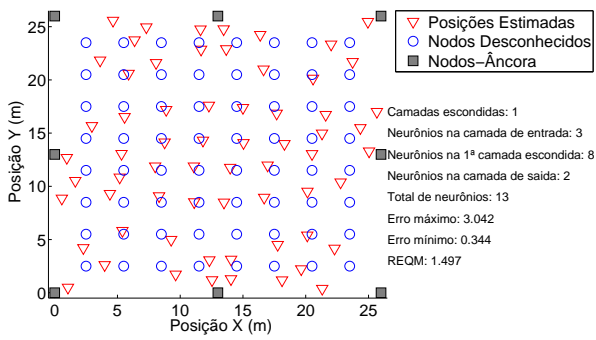
Observando-se as fronteiras de Pareto (figuras 102(c) e 102(d)) para a amostra de 25 nodos-treino, é possível notar que o arranjo circular apresentou mais soluções ótimas.

Nas figuras 102(e) e 102(f) é possível notar que o arranjo circular apresenta o menor erro médio, porém o maior erro máximo.

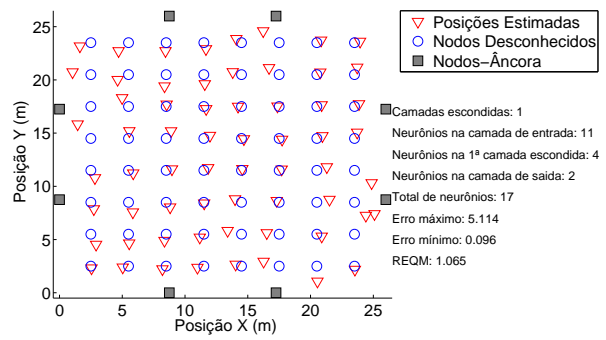
A topologia circular que utilizou 81 nodos-treino apresentou piores estimativas de localização do que o caso dotado de arranjo de nodos-âncora usual. Porém, o número de neurônios artificiais de suas soluções foi menor.

Observando-se as fronteiras de Pareto (figuras 103(c) e 103(d)) para a amostra de 81 nodos-treino, é possível notar que 87,5% das soluções derivadas do arranjo convencional produziram resultados com erro inferior a 2 metros.

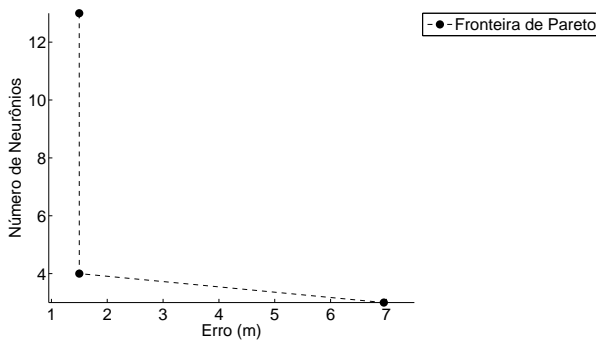
Nas figuras 103(e) e 103(f), com o aumento do tamanho da amostra, a topologia com o arranjo circular de nodos-âncora apresentou maiores erros, tanto médio quanto máximo.



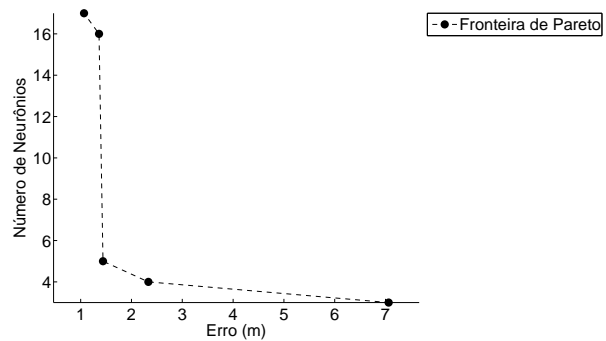
(a) Estimativas usando 8 âncoras alinhados.



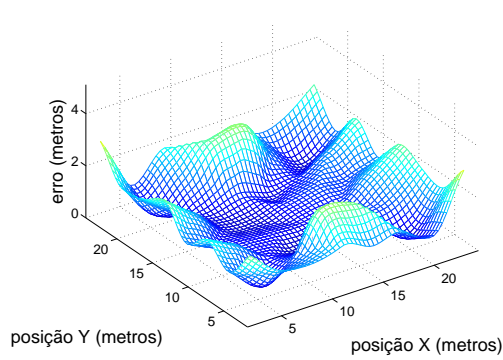
(b) Estimativas usando 8 âncoras circulares.



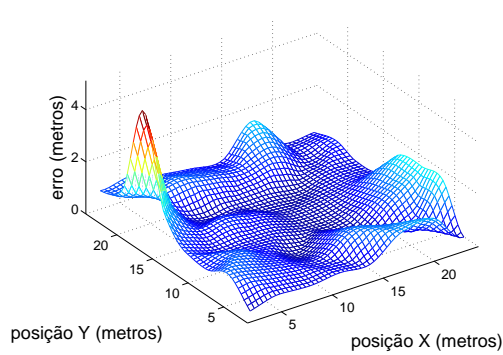
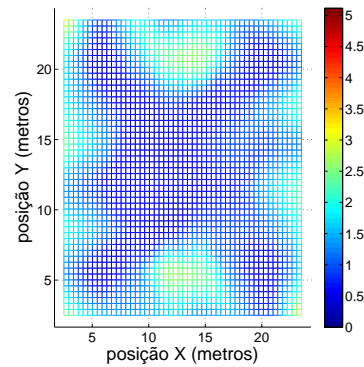
(c) Fronteira de Pareto - 8 âncoras alinhados.



(d) Fronteira de Pareto - 8 âncoras circulares.



(e) Erro usando 8 âncoras no arranjo normal.



(f) Erro usando 8 âncoras no arranjo circular.

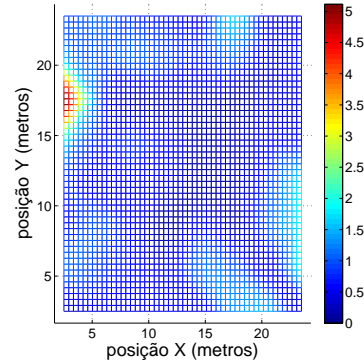
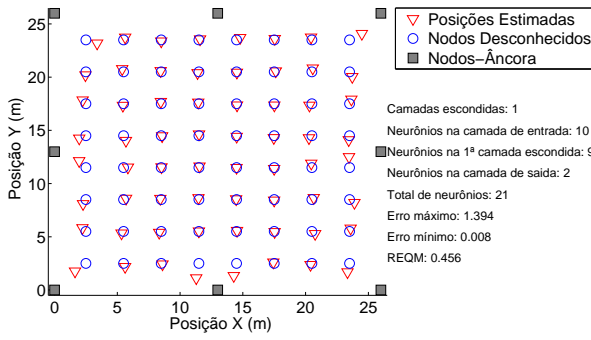
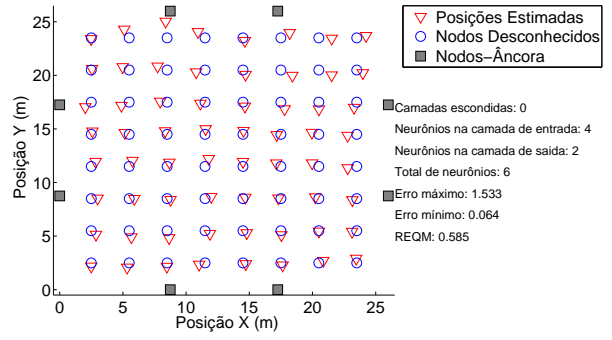


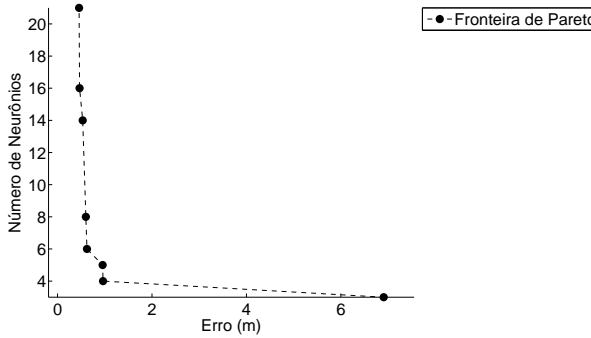
Figura 102: Influência da posição dos âncoras usando 25 nodos-treino.



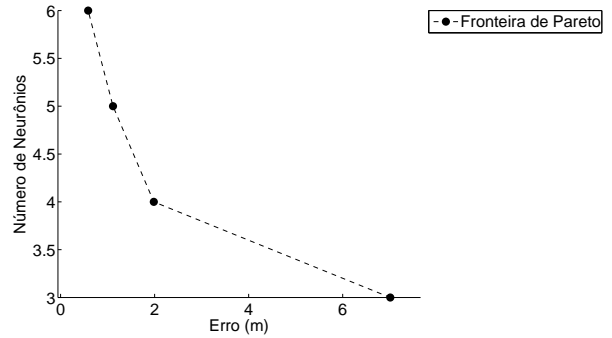
(a) Estimativas usando 8 âncoras alinhados.



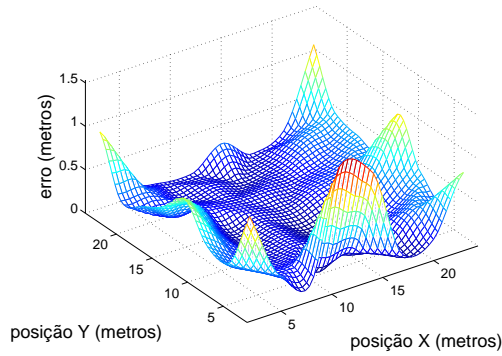
(b) Estimativas usando 8 âncoras circulares.



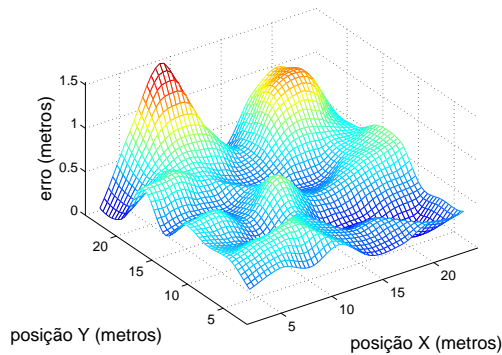
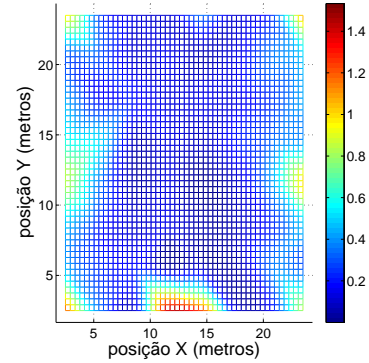
(c) Fronteira de Pareto - 8 âncoras alinhados.



(d) Fronteira de Pareto - 8 âncoras circulares.



(e) Erro usando 8 âncoras no arranjo normal.



(f) Erro usando 8 âncoras no arranjo circular.

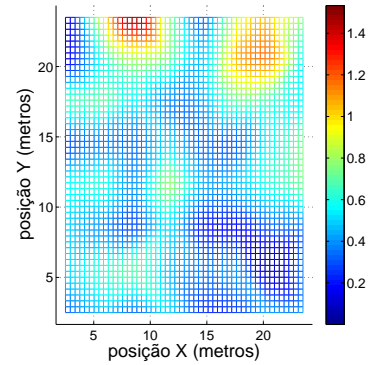


Figura 103: Influência da posição dos âncoras usando 81 nodos-treino.

4.4.1.6 Arranjo Alternativo - 9 Nodos-Âncora em Círculo

Para complementar o teste anterior, foram simuladas as mesmas topologias de rede, porém com um nodo-âncora adicional posicionado no centro da área de teste. A adição deste nodo-âncora central fez o arranjo circular superar o desempenho do arranjo em grade utilizado nos testes principais, tanto para a amostra de 25 nodos-treino quanto para a amostra de 81 nodos-treino (figuras 104 e 105), além de reduzir a complexidade das RNAs geradas.

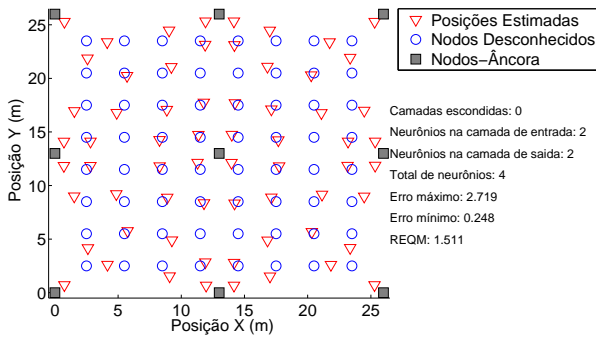
Observando-se as fronteiras de Pareto (figuras 104(c) e 104(d)) para a amostra de 25 nodos-treino, é possível notar que o arranjo circular apresentou uma solução ótima a mais.

Nas figuras 104(e) e 104(f) é possível notar que o arranjo circular apresenta assimetria na localização dos maiores índices de REQM, além de inserir erro na região central da área de teste.

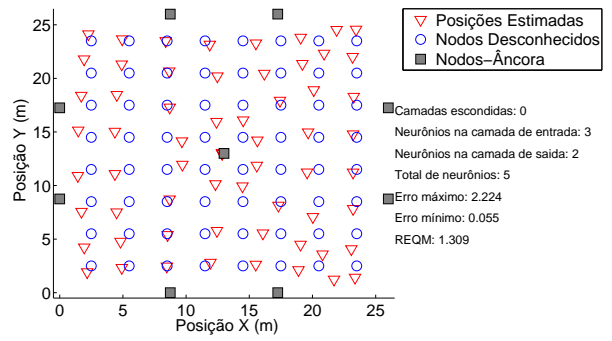
A topologia circular que utilizou 81 nodos-treino também apresentou melhores estimativas de localização do que o caso dotado de arranjo de nodos-âncora usual. Porém, esta topologia utilizou mais neurônios artificiais.

Observando-se as fronteiras de Pareto (figuras 104(c) e 104(d)) para a amostra de 81 nodos-treino, é possível notar que 100% das soluções derivadas do arranjo circular produziram resultados com erro igual ou inferior a 2 metros.

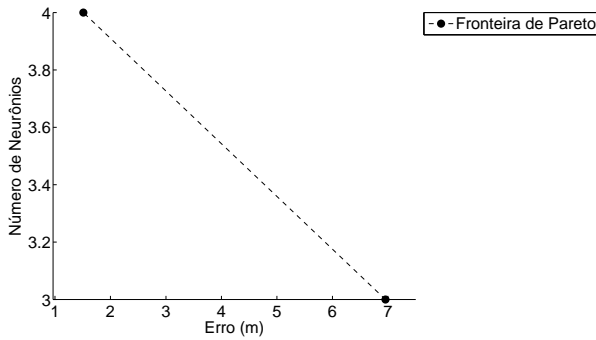
Nas figuras 104(e) e 104(f), com o aumento do tamanho da amostra, a topologia com o arranjo circular de nodos-âncora apresentou menores erros, tanto médio quanto máximo.



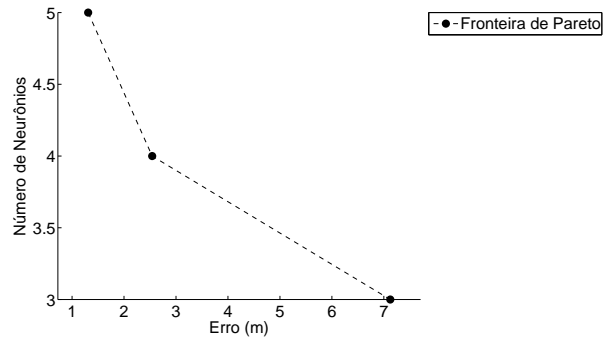
(a) Estimativas usando 9 âncoras alinhados.



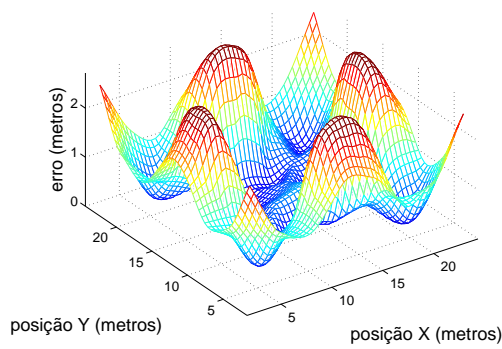
(b) Estimativas usando 9 âncoras circulares.



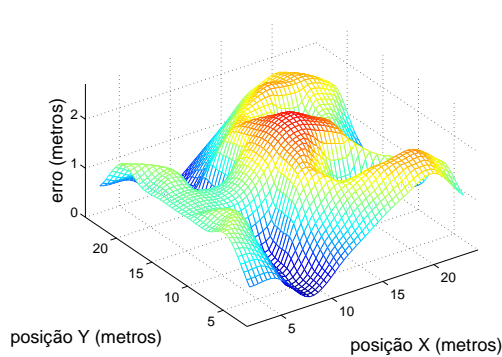
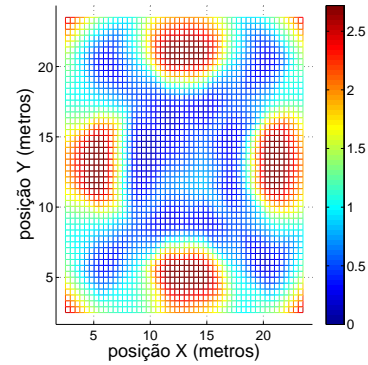
(c) Fronteira de Pareto - 9 âncoras alinhados.



(d) Fronteira de Pareto - 9 âncoras circulares.



(e) Erro usando 9 âncoras no arranjo normal.



(f) Erro usando 9 âncoras no arranjo circular.

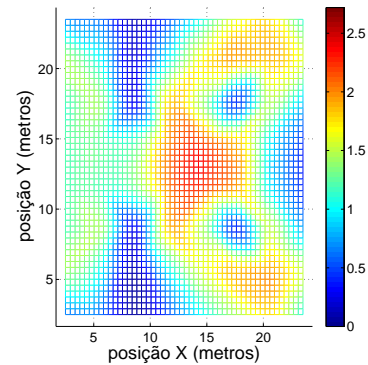
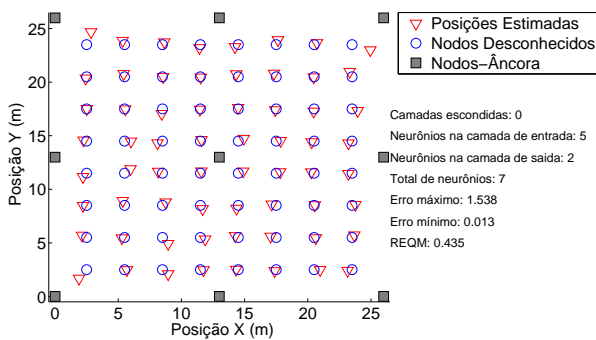
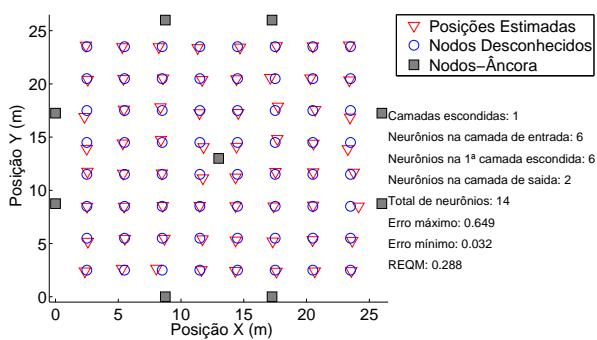


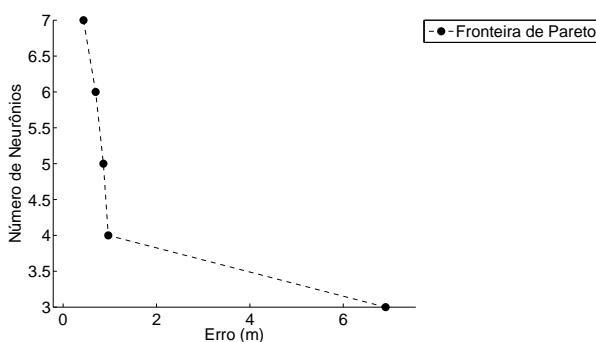
Figura 104: Influência da posição dos âncoras usando 25 nodos-treino.



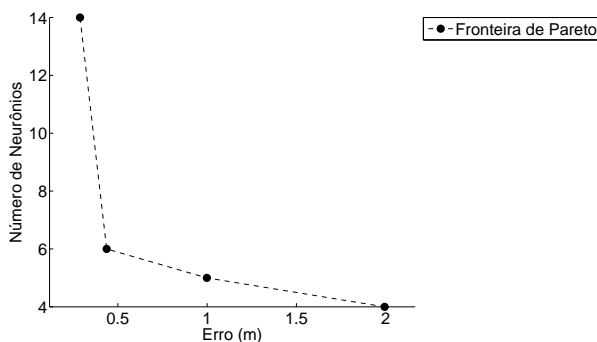
(a) Estimativas usando 9 âncoras alinhados.



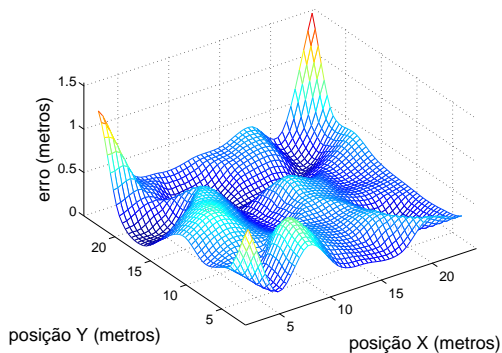
(b) Estimativas usando 9 âncoras circulares.



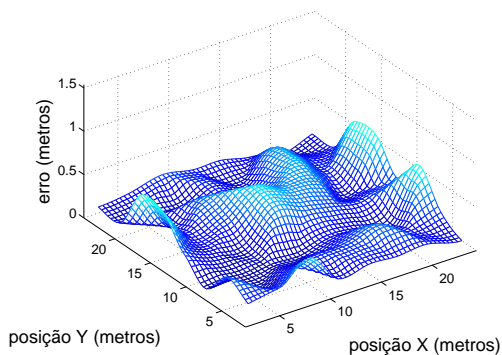
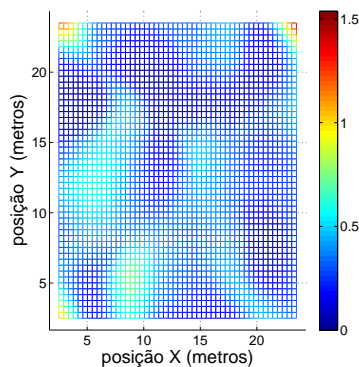
(c) Fronteira de Pareto - 9 âncoras alinhados.



(d) Fronteira de Pareto - 9 âncoras circulares.



(e) Erro usando 9 âncoras no arranjo normal.



(f) Erro usando 9 âncoras no arranjo circular.

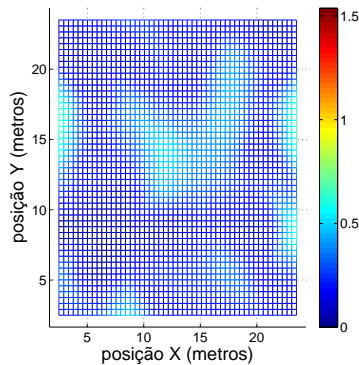


Figura 105: Influência da posição dos âncoras usando 81 nodos-treino.

4.4.1.7 4 Nodos-Âncora em Arranjo Triangular

A partir dos resultados dos testes principais foi possível perceber que, em geral, os melhores desempenhos foram alcançados pelas topologias com 3 nodos-âncora. Além disso, foi possível observar que um nodo-âncora central reduzia a complexidade da RNA, tendendo a eliminar a necessidade de camadas escondidas.

Logo, em uma tentativa de unir estas boas características, foi criada a topologia com 4 nodos-âncora, sendo 3 dispostos em forma de triângulo e um posicionado no centro da área de teste. O desempenho dessa topologia foi analisado para todos os tamanhos de amostras de treinamento (figuras 106 a 111), e se mostrou superior ao arranjo tradicional (4 nodos-âncora dispostos nos vértices da área de teste) a partir de 64 nodos-treino.

Para a amostra com 16 nodos-treino, a nova topologia de nodos-âncora obteve desempenho muito inferior, além de introduzir um grande nível de erro na região central da área de teste, como pode ser visto na figura 106(f).

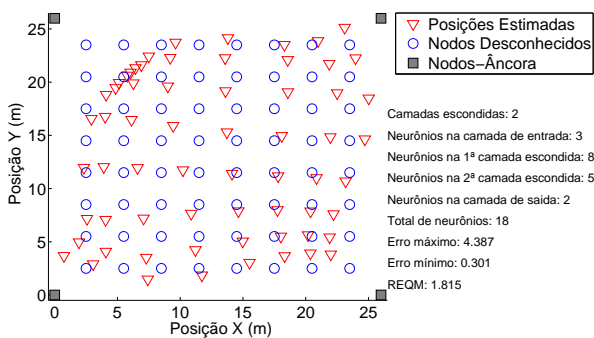
Na amostra composta por 25 nodos-treino, a nova topologia reduziu a complexidade da RNA gerada, mas novamente inseriu erro na região central da área de teste, conforme a figura 107(f).

Utilizando amostra de 36 nodos-treino, o erro médio gerado pela nova topologia já se aproxima mais do erro oriundo da topologia com nodos-âncora no arranjo usual. Porém, um grande pico no valor da REQM é visto no centro da área de teste, conforme ilustrado pela figura 108(f).

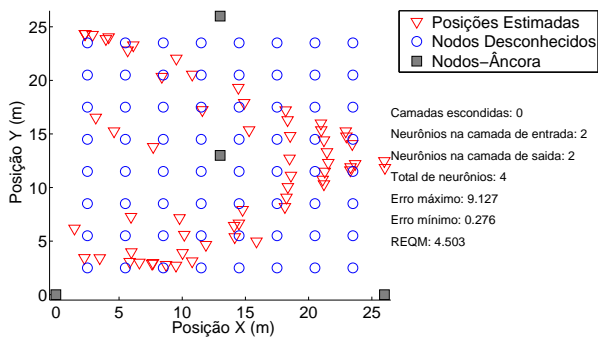
Para a amostra de 49 nodos-treino, o erro médio da topologia alternativa fica quase similar ao apresentado pela topologia normal. Além disso, com este tamanho de amostra, o erro central antes gerado não é mais apresentado (figura 109(f)).

A partir da amostra formada por 64 nodos-treino, o desempenho da topologia alternativa supera o alcançado pela topologia normal, apresentando o menor erro médio, bem como o menor erro máximo (figura 110(f)).

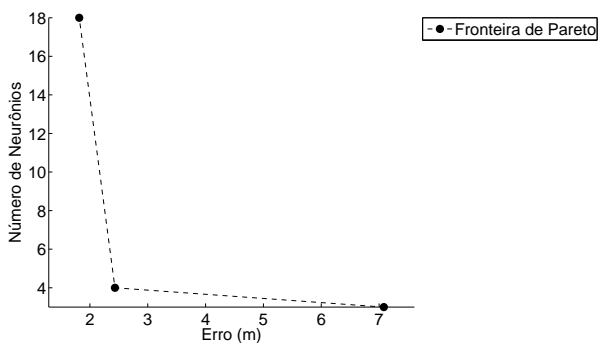
Utilizando 81 nodos âncora, o desempenho da topologia alternativa fica apenas um pouco superior ao apresentado pela topologia usual, porém, o erro na região central da área de teste aparece novamente (figura 111(f)). Além disso, a complexidade da RNA gerada pela topologia nova apresentou redução.



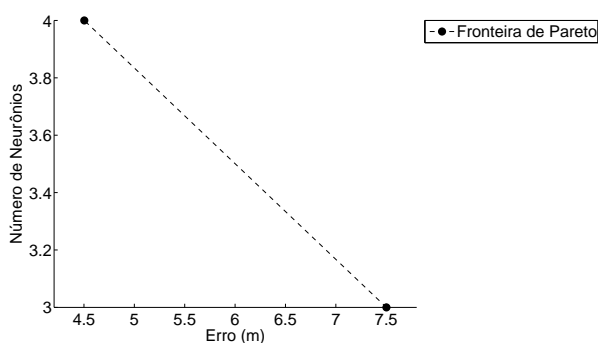
(a) Estimativas - 4 âncoras em grade.



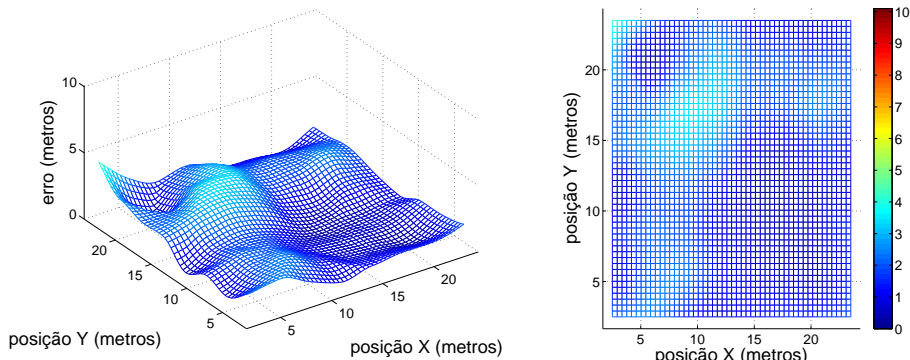
(b) Estimativas - 4 âncoras em novo arranjo.



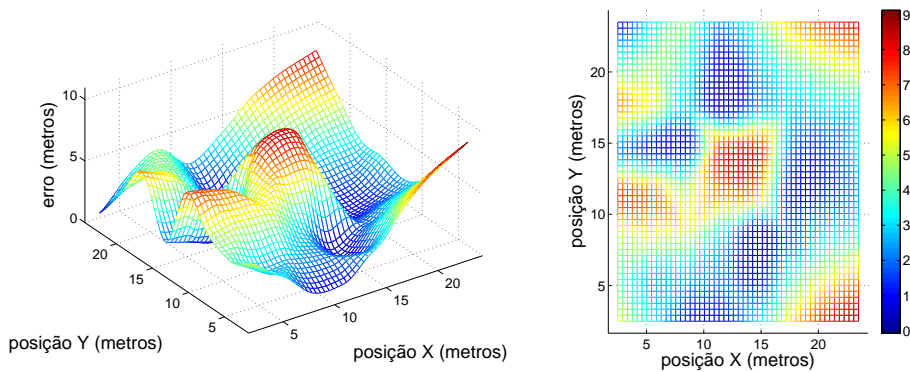
(c) Fronteira de Pareto - 4 âncoras em grade.



(d) Fronteira de Pareto - 4 âncoras em novo arranjo.

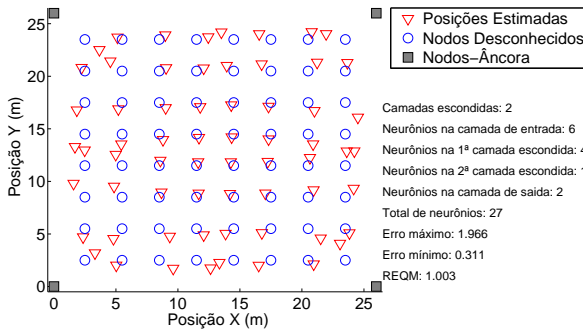


(e) Erro - 4 âncoras em grade.

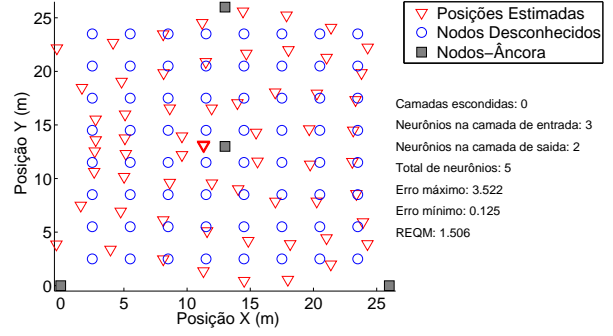


(f) Erro - 4 âncoras em novo arranjo.

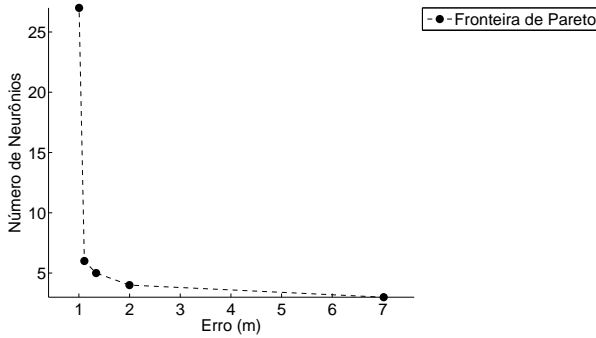
Figura 106: Influência da posição de 4 nodos-âncora nos resultados com 16 nodos-treino.



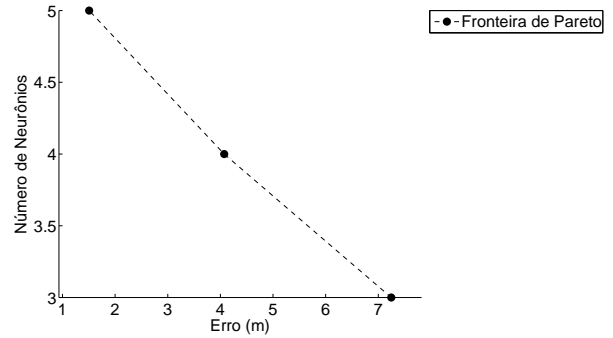
(a) Estimativas - 4 âncoras em grade.



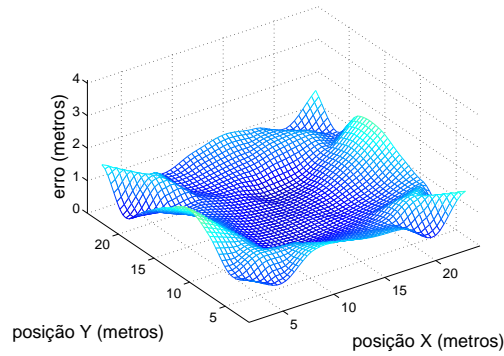
(b) Estimativas - 4 âncoras em novo arranjo.



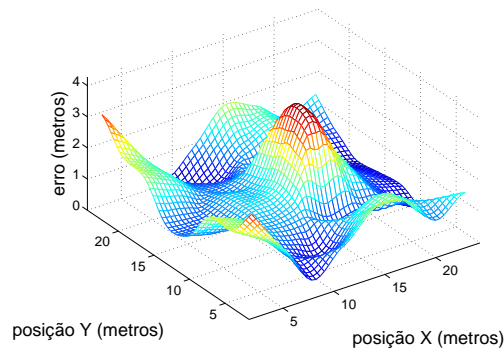
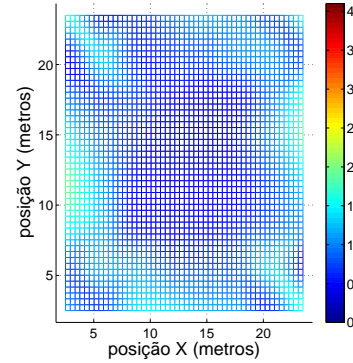
(c) Fronteira de Pareto - 4 âncoras em grade.



(d) Fronteira de Pareto - 4 âncoras em novo arranjo.



(e) Erro - 4 âncoras em grade.



(f) Erro - 4 âncoras em novo arranjo.

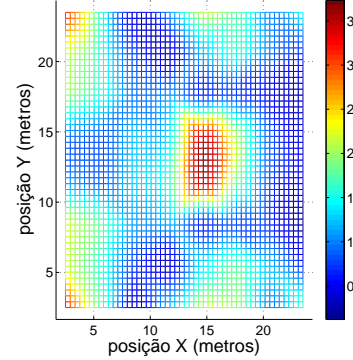
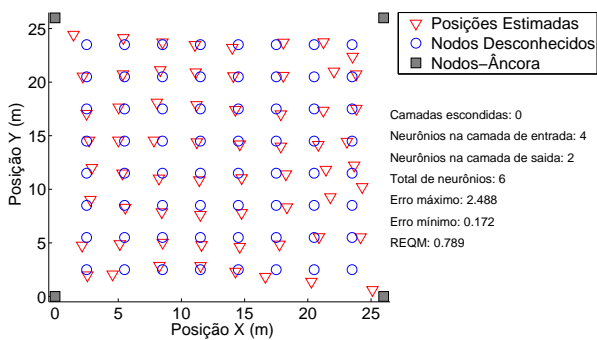
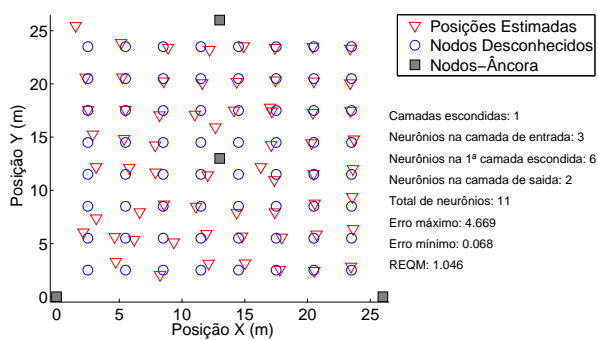


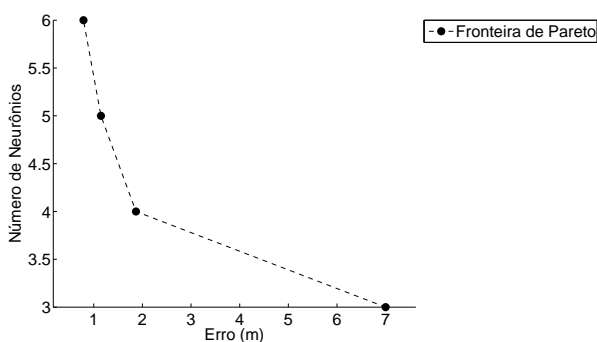
Figura 107: Influência da posição de 4 nodos-âncora nos resultados com 25 nodos-treino.



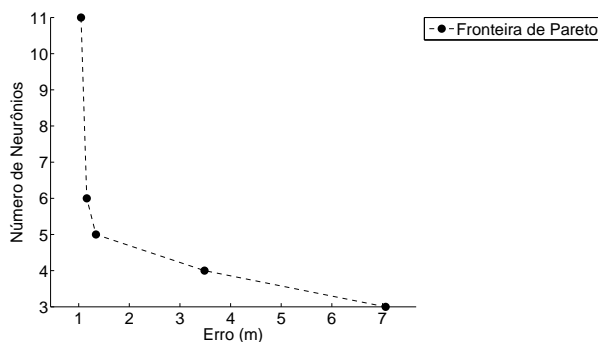
(a) Estimativas - 4 âncoras em grade.



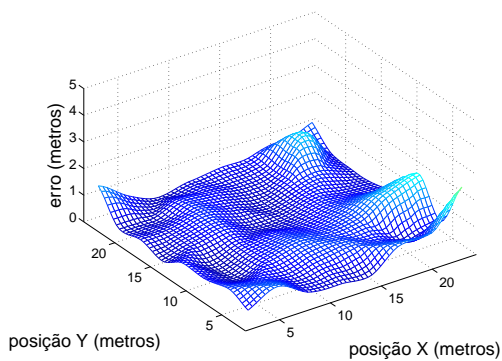
(b) Estimativas - 4 âncoras em novo arranjo.



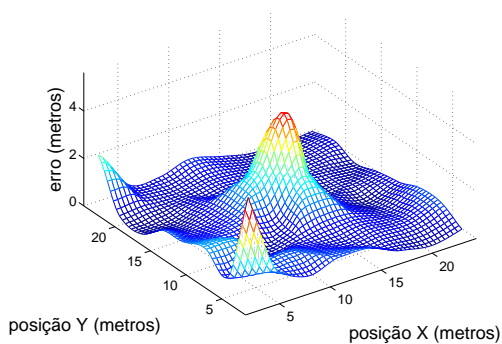
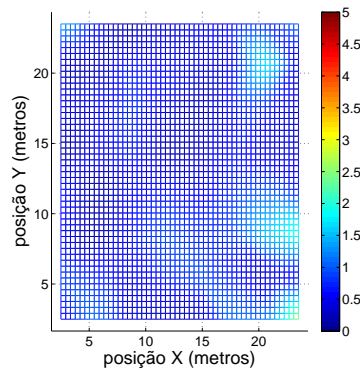
(c) Fronteira de Pareto - 4 âncoras em grade.



(d) Fronteira de Pareto - 4 âncoras em novo arranjo.



(e) Erro - 4 âncoras em grade.



(f) Erro - 4 âncoras em novo arranjo.

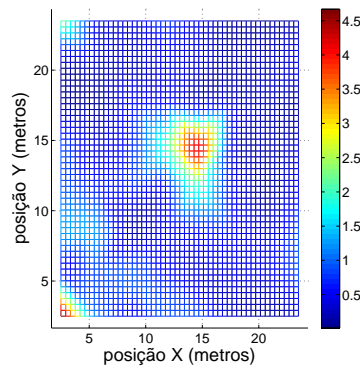
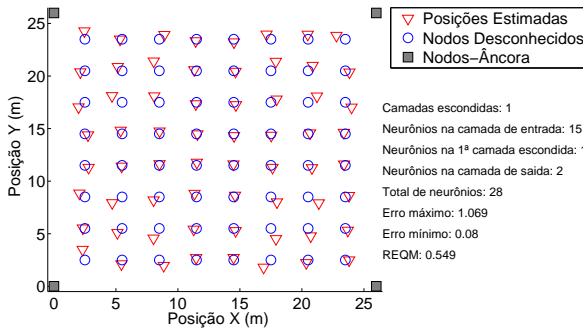
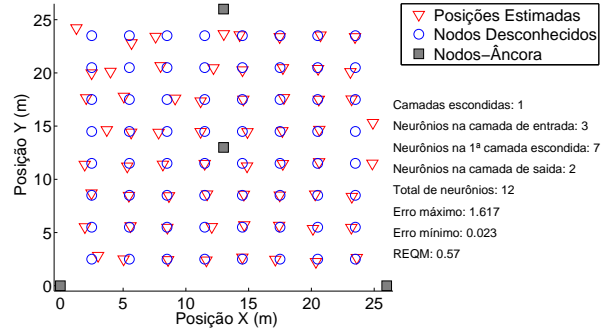


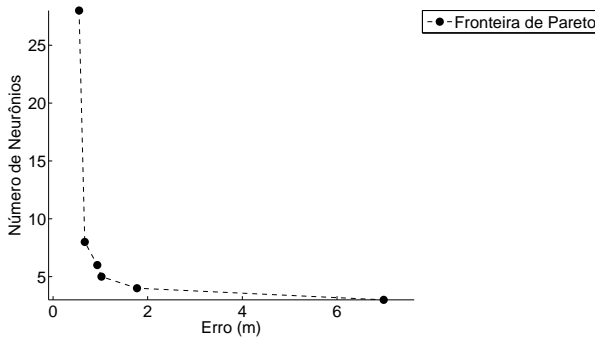
Figura 108: Influência da posição de 4 nodos-âncora nos resultados com 36 nodos-treino.



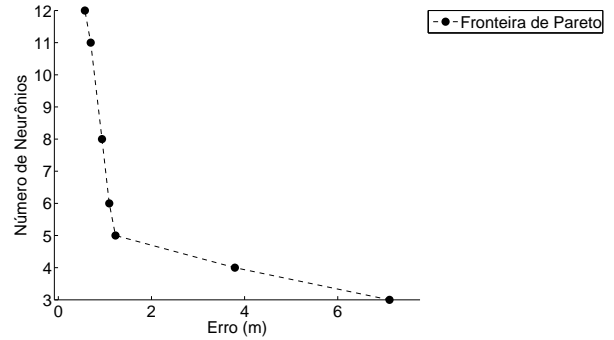
(a) Estimativas - 4 âncoras em grade.



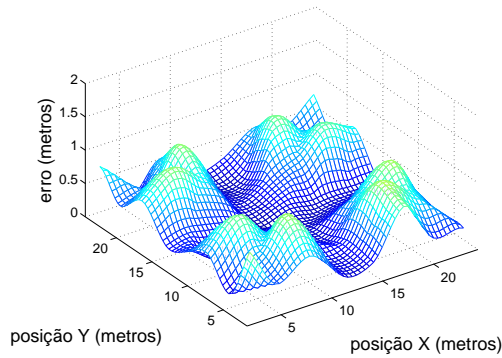
(b) Estimativas - 4 âncoras em novo arranjo.



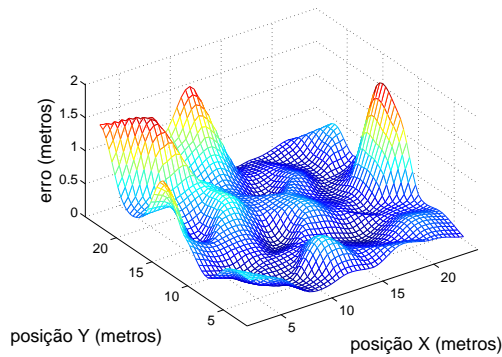
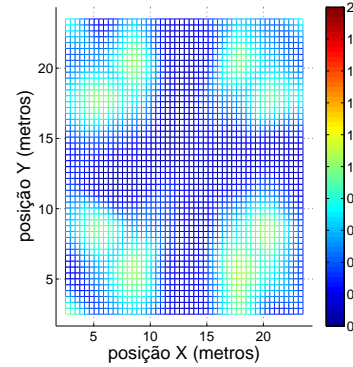
(c) Fronteira de Pareto - 4 âncoras em grade.



(d) Fronteira de Pareto - 4 âncoras em novo arranjo.



(e) Erro - 4 âncoras em grade.



(f) Erro - 4 âncoras em novo arranjo.

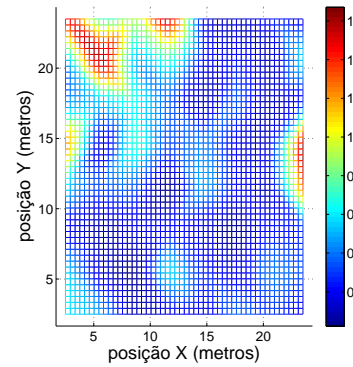
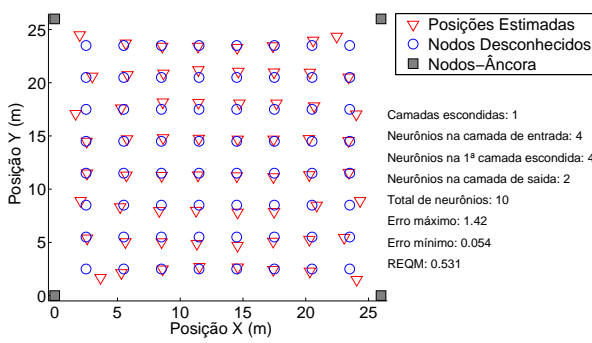
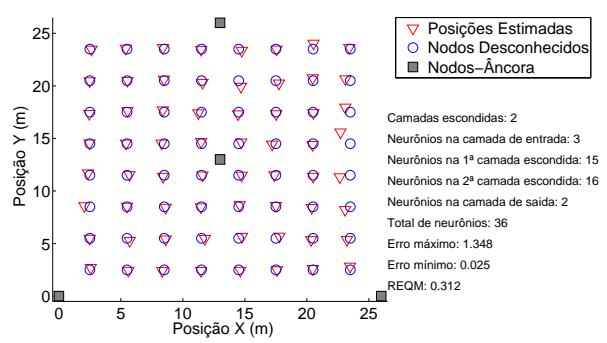


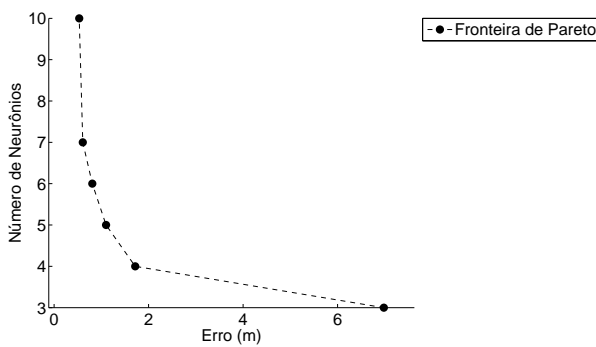
Figura 109: Influência da posição de 4 nodos-âncora nos resultados com 49 nodos-treino.



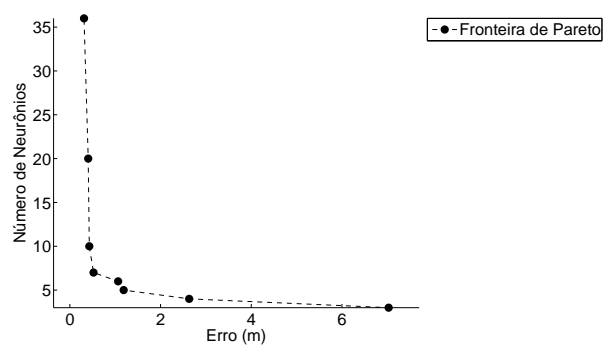
(a) Estimativas - 4 âncoras em grade.



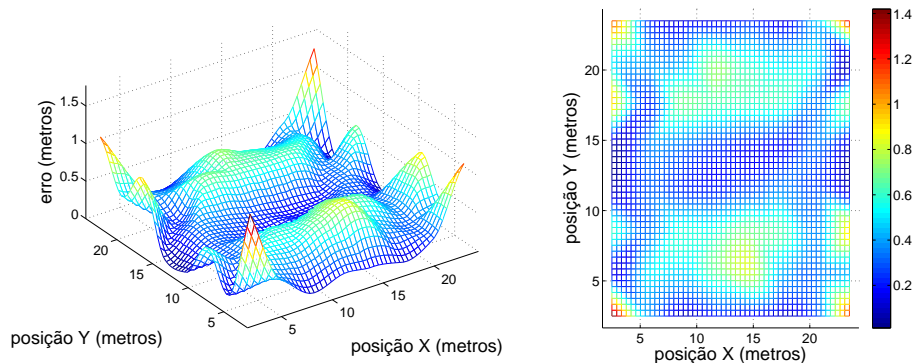
(b) Estimativas - 4 âncoras em novo arranjo.



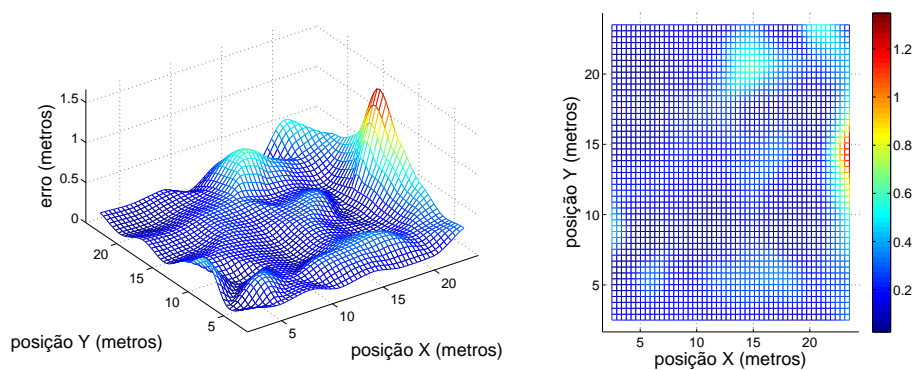
(c) Fronteira de Pareto - 4 âncoras em grade.



(d) Fronteira de Pareto - 4 âncoras em novo arranjo.

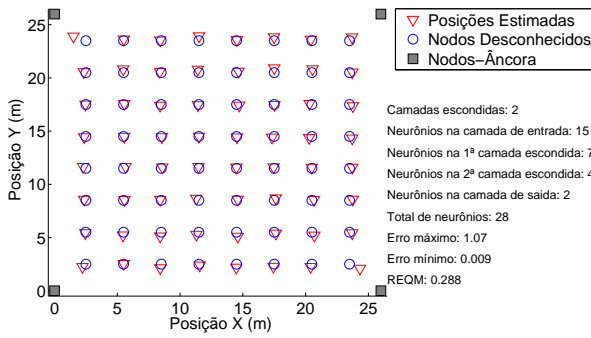


(e) Erro - 4 âncoras em grade.

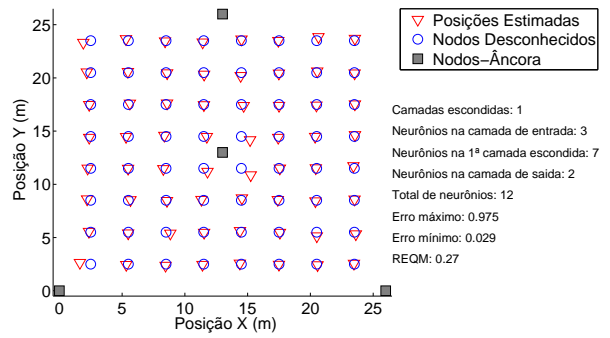


(f) Erro - 4 âncoras em novo arranjo.

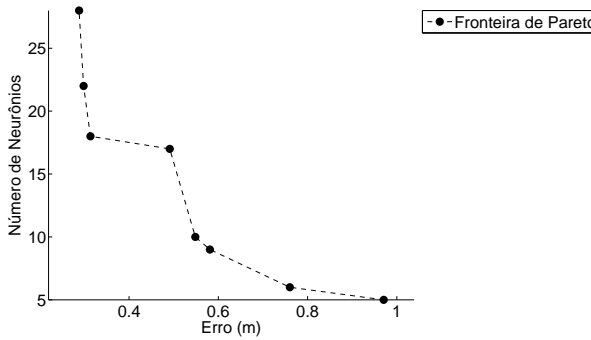
Figura 110: Influência da posição de 4 nodos-âncora nos resultados com 64 nodos-treino.



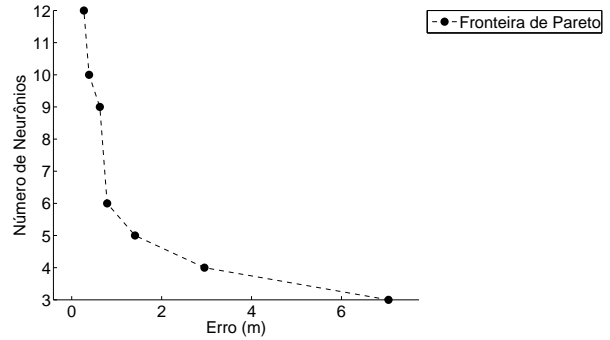
(a) Estimativas - 4 âncoras em grade.



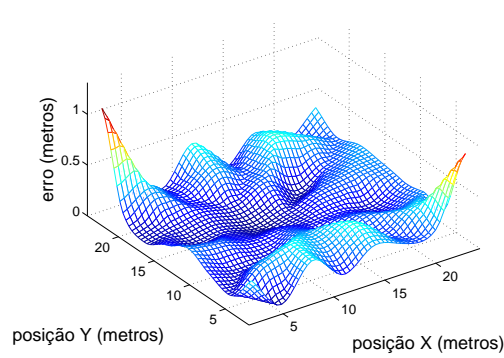
(b) Estimativas - 4 âncoras em novo arranjo.



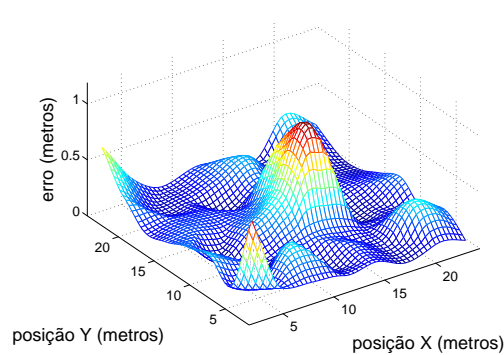
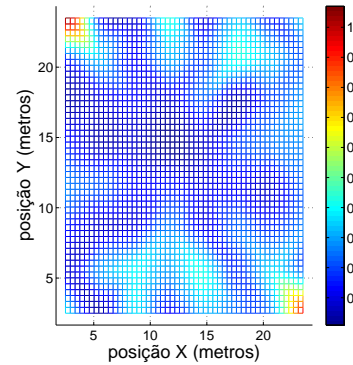
(c) Fronteira de Pareto - 4 âncoras em grade.



(d) Fronteira de Pareto - 4 âncoras em novo arranjo.



(e) Erro - 4 âncoras em grade.



(f) Erro - 4 âncoras em novo arranjo.

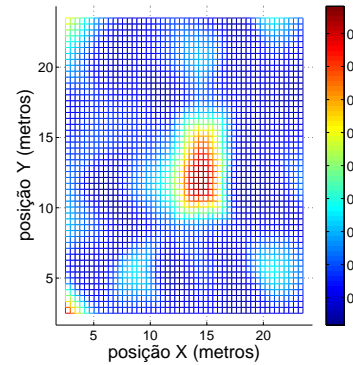
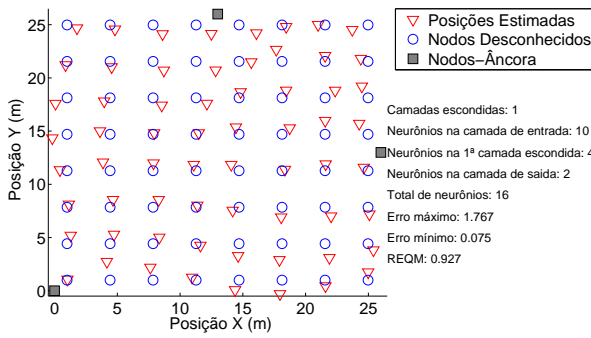


Figura 111: Influência da posição de 4 nodos-âncora nos resultados com 81 nodos-treino.

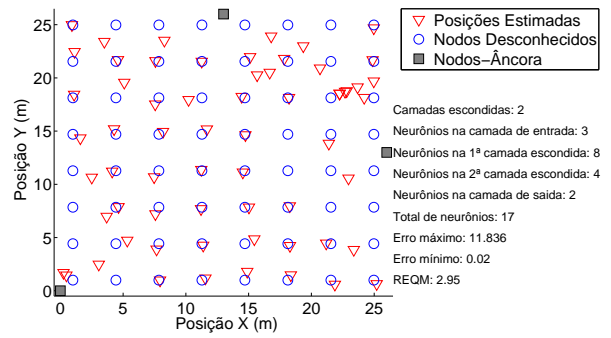
4.4.1.8 Variação na Posição de 25 Nodos-Treino - 3 Âncoras

Ainda, foi executada uma simulação com 25 nodos-treino em posições aleatórias, de modo que suas coordenadas (tanto x quanto y) não se repetissem. Deste modo, na ocasião do treinamento, a RNA seria exposta a mais coordenadas distintas da área de teste do que se fosse utilizado o arranjo em grade (que possui várias repetições de coordenadas), o que poderia resultar em aumento de desempenho.

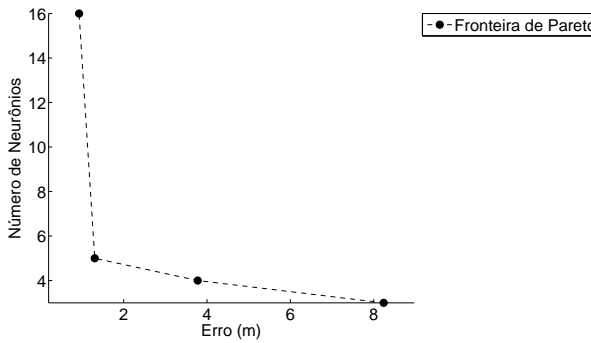
Esta hipótese estava errada, pois a REQM obtida foi aproximadamente o triplo da obtida com nodos-treino em dispostos em grade. Além disso, o erro máximo obtido através da topologia de nodos-treino sem repetição de coordenadas foi quase 7 vezes maior do que o normal, conforme é visto na figura 112(f).



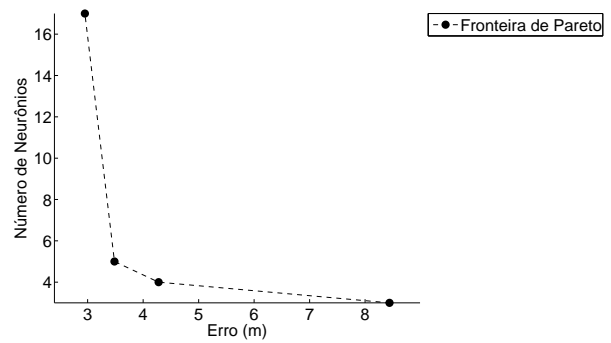
(a) Estimativas - Amostra em grade.



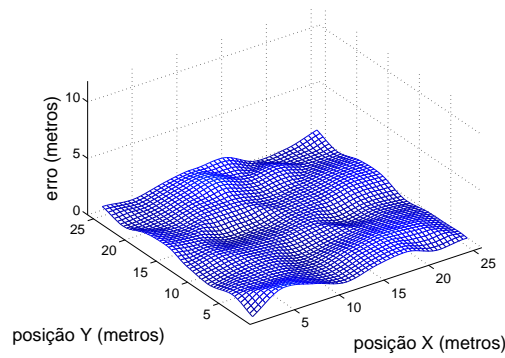
(b) Estimativas - Amostra aleatorizada.



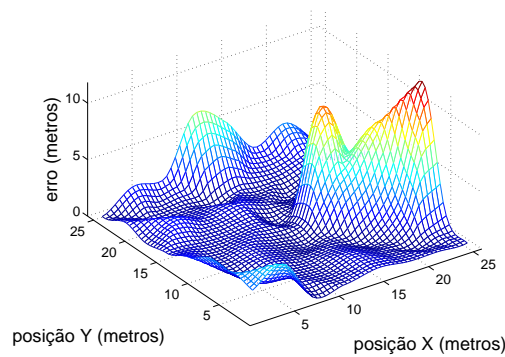
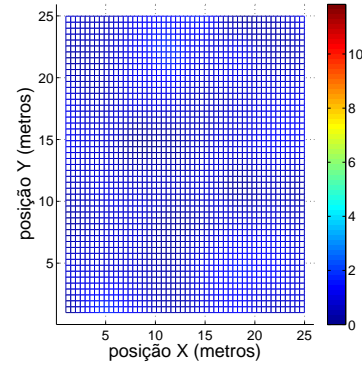
(c) Fronteira de Pareto - Amostra em grade.



(d) Fronteira de Pareto - Amostra aleatorizada.



(e) Erro - Amostra em grade.



(f) Erro - Amostra aleatorizada.

Figura 112: Influência da posição de 25 nodos-treino nos resultados com 3 âncoras.

4.5 Comparação com Outras Abordagens

Os sistemas de localização propostos na literatura são testados de diferentes formas e utilizando-se de diferentes métricas no que se refere ao erro de localização. Muitas variações também são encontradas no tamanho e tipo das áreas testadas, bem como nos métodos de simulação (poucas implementações práticas são encontradas) e no raio de alcance dos nodos utilizados (potência de transmissão). Aliado a estas adversidades, existe o fato de existirem poucas abordagens que se utilizam de RNAs como princípio de funcionamento de seus sistemas de localização em moldes similares ao exposto neste trabalho, dificultando ainda mais a tarefa de fazer comparações.

Ainda assim, a tabela 34 agrupou técnicas estudadas em alguns trabalhos e as comparou com os sistemas de localização propostos nesta dissertação. Objetivando uma comparação mais justa, foram selecionadas abordagens cujas condições de testes/simulações foram mais semelhantes às utilizadas neste trabalho. Em todos os casos comparados, o alcance dos rádios era tal que todos os nodos tinham capacidade de se comunicar entre si, ou seja, todos os nodos-âncora eram ouvidos pelos demais. Outra semelhança baseia-se no fato de que, nos trabalhos comparados, o RMSE/REQM (*Root Mean Square Error*/Raiz do Erro Quadrático Médio) foi adotado como métrica para avaliar a precisão da localização.

Como o tamanho das áreas nas quais os sistemas de localização foram testados variou de técnica para técnica, criou-se um figura de mérito relacionando o RMSE/REQM com o tamanho da área de teste, exprimindo um erro proporcional à maior distância encontrada em cada área testada. A expressão para cálculo deste erro é dada pela equação 4.1, onde $REQM_P$ representa o erro proporcional, $REQM$ é a raiz do erro quadrático médio e o termo $(aresta \cdot \sqrt{2})$ representa o comprimento em metros da diagonal do quadrado formado pela área de teste, uma vez que esta é a maior distância possível entre dois nodos em cada caso.

$$REQM_P = \frac{REQM}{aresta \cdot \sqrt{2}} \quad (4.1)$$

Os tipos de RNAs usados nesta comparação são os seguintes:

- MLP: Rede neural artificial do tipo *Multi-Layer Perceptron*;
- 2 MLP-AG: Dois estágios com uma MLP cada otimizados pelo algoritmo genético;
- RBF: Rede neural artificial do tipo *Radial Basis Function*;

- RRBf: Rede neural artificial do tipo *Reduced Radial Basis Function*;
- RNN: Rede neural artificial do tipo *Recurrent Neural Network*;
- MLP-SA: Rede neural artificial do tipo MLP otimizada pelo algoritmo *Simulated Annealing*;
- MLP-AGMO: Rede neural artificial do tipo MLP otimizada pelo algoritmo genético multi-objetivo;

Além de abordagens utilizando RNAs, estão também presentes na tabela comparativa dois sistemas de localização baseados em filtros de Kalman (SHAREEF et al., 2007) com modelos PV e PVA (do inglês, *Position-velocity* e *Position-velocity-Acceleration*) .

Tabela 34: Comparação de desempenho entre os métodos propostos e outros da literatura

Sistema de Localização	Área de Teste (metros)	Número de Âncoras	REQM (metros)	REQM Proporcional (%)
2 MLP-AG ¹	26 x 26	9	0,3100	0,8450
MLP-SA ¹	26 x 26	8	0,6100	1,6630
MLP-AGMO ¹	26 x 26	8	0,4500	1,2270
MLP ²	20 x 20	8	0,5400	1,9140
RBF ³	3,0 x 3,0	3	0,0727	1,7180
MLP ³	3,0 x 3,0	3	0,0754	1,7820
RNN ³	3,0 x 3,0	3	0,0757	1,7890
PV ³	3,0 x 3,0	3	0,1047	2,4750
PVA ³	3,0 x 3,0	3	0,1138	2,6903
RRBF ³	3,0 x 3,0	3	0,1397	3,3020
MLP-AGMO ¹	26 x 26	3	0,2300	0,6270

É importante ressaltar que, nos sistemas de localização propostos neste trabalho, a quantidade de nodos utilizados para treinamento das RNAs foi sempre igual (81 nodos-treino) ou inferior às empregadas nas abordagens de outros trabalhos presentes na tabela comparativa (81² e 121³ nodos-treino).

Através da tabela 34, é possível observar que os sistemas de localização propostos nesta dissertação apresentam ganho de desempenho em relação aos outros propostos na

¹Este trabalho

²(RAHMAN; PARK; KIM, 2009)

³(SHAREEF; ZHU; MUSAVI, 2007)

literatura. Além dos resultados melhores, as abordagens propostas nesta dissertação geram sistemas de localização de forma automática, adaptando-se de forma autônoma a diferentes redes de sensores sem fio.

5 CONCLUSÃO

Neste trabalho foi apresentado um método para seleção das melhores topologias de RNAs utilizando otimização através de metaheurísticas. Tanto a seleção de parâmetros da rede (número de camadas escondidas, número de neurônios em cada camada, função de transferência) quanto a seleção das condições de aplicação ideais (número de nodos-âncora, distribuição dos nodos-âncora, número de amostras de treinamento e distribuição de amostras de treinamento) foram abordados. A utilização de metaheurísticas para seleção de parâmetros constitui um avanço significativo para a área uma vez que os trabalhos na literatura envolvendo aplicações de RNAs em localização de nodos usualmente adotam estruturas fixas baseadas na experiência do pesquisador, sem investigação adicional acerca das variáveis que afetam o erro.

Foram realizadas três propostas de otimização das RNAs: utilização algoritmos genéticos (AGs) para criação de apenas uma rede, utilização de *Simulated Annealing* (SA) para criação de uma rede e utilização de AGs para a criação de duas RNAs. Neste último caso, a concatenação das redes visa eliminar o erro da primeira, sendo necessário para isso o treinamento isolado de cada uma.

Para cada proposta, foi desenvolvido um fluxo que permite ao desenvolvedor da RSSF encontrar os parâmetros ótimos desta sem a necessidade de basear-se em conhecimento prévio. No que diz respeito as condições de aplicações ideais, os resultados do trabalho servem como guia para seleção do número mínimo de nodos-treino para as redes e também número e disposição dos nodos-âncora.

Na criação das RNAs, as medições de RSSI foram utilizadas como entradas para localizar nodos. Os métodos foram testados usando o simulador de redes sem fio probabilístico chamado *Prowler*, que é baseado no software *MatLab*, para coletar os dados de entrada de um ambiente de rede estático interno. Os dados de transmissão e recepção foram todos baseados no mote comercial MicaZ, de forma a aproximar as simulações de uma aplicação real.

Os resultados indicam que a estrutura de rede varia consideravelmente para cada caso de âncoras e amostras. Portanto, é necessário realizar a otimização com o método proposto para cada novo caso, justificando o desenvolvimento deste trabalho. No entanto, no que diz respeito às condições de aplicação ideais, evidencia-se que amostras iguais ou inferiores a 16 nodos para treinamento resultam em erros demasiadamente grandes.

Recomenda-se aos projetistas utilizarem amostras para treinamento iguais ou superiores à 64 nodos, que neste trabalho foram os casos que resultaram em RNAs com melhor desempenho. Ainda, o número de nodos-âncora que obteve os melhores resultados para a maioria dos testes foi três (observando-se que os transmissores dos nodos-âncora abrangiam toda a área no qual os nodos desconhecidos estavam presentes). O resultado utilizando a melhor estrutura de rede neural artificial encontrada após o processo de otimização teve um erro médio quadrático de 0,23 metro, um máximo erro de 0,65 metro e um mínimo erro de 0,014 metro, utilizando 3 nodos-âncora e 64 nodos-treino.

A partir da análise dos resultados foi possível observar que os maiores erros nas estimativas de localização se encontravam nas proximidades dos nodos-âncora. Algumas hipóteses que se pode levantar sobre este comportamento do erro são:

- Saturação dos neurônios artificiais: Pode ter ocorrido, devido à proximidade excessiva dos nodos-âncora, a saturação dos neurônios artificiais em função da alta potência do sinal RF recebido. Mesmo com a utilizações de recursos para normalizar os dados de entrada, o *MatLab* pode não ter conseguido lidar muito bem com as entradas das RNAs;
- Modelo de propagação: Em algumas situações (nodos desconhecidos muito próximos a nodos-âncora), o modelo matemático adotado para representar o canal de propagação pode não ter sido adequado, gerando erros de localização nas proximidades dos nodos-âncora;
- Função de localização aproximada pelas RNAs: Como as RNAs aproximam funções relacionando a potência do sinal recebido de nodos-âncora com a localização de nodos desconhecidos e, a maioria das amostras de treinamento não estão muito próximas de nodos-âncora, a função aproximada se adequa mais aos pontos cujas características são semelhantes à maior parte da amostra utilizada no treino da RNA.

5.1 Trabalhos Futuros

De acordo com o estudo realizado e os resultados obtidos neste trabalho, observa-se a possibilidade de aprofundamentos de alguns pontos já explorados inicialmente e também investigação de novas linhas de pesquisa:

- Aplicar os resultados do presente trabalho em sistema de localização que utilizem nodos móveis;
- Aprofundar o estudo da influência do número e posição dos âncoras no desempenho e nas estruturas das RNAs para encontrar explicações teóricas fundamentadas sobre os resultados empíricos obtidos neste trabalho, objetivando relacionar as variáveis do problema;
- Estudar a influência da posição das amostras de treinamento no desempenho das RNAs para cada topologia de âncoras;
- Realizar testes da estrutura concatenada de RNAs (dois estágios) em aplicações com três âncoras, visto que este número de âncoras resultou nos melhores desempenhos;
- Implementar sistemas de localização gerados pelo método proposto neste trabalho em nodos MICAz visando analisar o consumo de energia e a velocidade de entrega das estimativas de posição, comparando-os com outras abordagens no que se refere a estes quesitos;
- Aprimoramento dos códigos-fonte utilizados para as simulações e desenvolvimento de uma interface de usuário visando a criação de uma ferramenta para auxiliar os projetistas de RSSFs;
- Testar os métodos propostos com outros modelos matemáticos para o canal de propagação, bem como para outros tamanhos de área de teste;
- Estudar a relação entre quantidade de neurônios artificiais / estrutura das RNAs e o custo computacional para realizar uma estimativa de localização.

APÊNDICE A - ARTIGOS PUBLICADOS PELO AUTOR

CHAGAS, S. H.; MARTINS, J. B. S; OLIVEIRA, L. L. *Review of Localization Schemes Using Neural Networks on Wireless Sensor Networks*. In: SIM 2011 - XXVI South Symposium on Microelectronics, 2011, Novo Hamburgo - Brasil.

CHAGAS, S. H.; MARTINS, J. B. S; OLIVEIRA, L. L. *An approach to Localization Scheme of Wireless Sensor Networks Based on Artificial Neural Networks and Genetic Algorithms*. In: IEEE NEWCAS Proceedings , 2012, Montreal - Canadá.

CHAGAS, S. H.; MARTINS, J. B. S; OLIVEIRA, L. L. *Node localization in Wireless Sensor Networks Using Artificial Neural Networks and Optimization Based on Simulated Annealing Algorithm*. In: SIM 2012 - XXVII South Symposium on Microelectronics, 2012, Ijuí - Brasil.

CHAGAS, S. H.; MARTINS, J. B. S; OLIVEIRA, L. L. *Genetic Algorithms and Simulated Annealing Optimization Methods in Wireless Sensor Networks Localization Using Artificial Neural Networks*. In: IEEE MWSCAS Proceedings, 2012, Boise - Estados Unidos da América.

APÊNDICE B - TRIANGULAÇÃO E TRILATERAÇÃO

Para efetuar o processo de triangulação e trilateração, dados de diversas referências são adicionados a um sistema linear para resolução. No caso da trilateração, são necessárias 3 ou mais referências, enquanto no caso da triangulação, são necessários 2 ou mais.

Em casos práticos, pode-se ter um número maior de referências (âncoras) com informações disponíveis para um determinado nodo. Desta forma, este é um sistema superdeterminado, ou seja, que tem um número de equações superior ao número de incógnitas. Os dados deste problema, no entanto, são obtidos experimentalmente e tem um grau de incerteza atrelado. Por tanto, pode-se resolver os mesmos através do método dos mínimos quadrados.

Os dados do problema de trilateração podem ser adicionados em um sistema linear, como o da equação B.1.

$$\left\{ \begin{array}{l} d_1^2 = (x - x_1)^2 + (y - y_1)^2 \\ d_2^2 = (x - x_2)^2 + (y - y_2)^2 \\ \vdots \\ d_n^2 = (x - x_n)^2 + (y - y_n)^2 \end{array} \right. \quad (\text{B.1})$$

Para resolver o problema por mínimos quadrados, coloca-se as equações na forma matricial:

$$\mathbf{Ax} = \mathbf{b} \quad (\text{B.2})$$

onde a matrix \mathbf{A} , \mathbf{b} e \mathbf{x} estão descritas logo abaixo:

$$\mathbf{A} = -2 \begin{bmatrix} (x_1 - x_n) & (y_1 - y_n) \\ \vdots & \vdots \\ (x_{n_1} - x_n) & (y_{n-1} - y_n) \end{bmatrix} \quad (\text{B.3})$$

$$\mathbf{b} = \begin{bmatrix} d_1^2 - d_n^2 - x_1^2 + x_n^2 - y_1^2 + y_n^2 \\ \vdots \\ d_{n-1}^2 - d_n^2 - x_{n-1}^2 + x_n^2 - y_{n-1}^2 + y_n^2 \end{bmatrix} \quad (\text{B.4})$$

$$\mathbf{x} = \begin{bmatrix} x \\ y \end{bmatrix} \quad (\text{B.5})$$

De forma similar, na triangulação, pode-se criar uma matriz com os dados das referências. Neste caso, converte-se a informação do ângulo para encontrar os elementos da equação da reta:

$$y = ax + b \quad (\text{B.6})$$

onde pode-se encontrar a e b:

$$a = \tan(\theta) \quad b = y - ax \quad (\text{B.7})$$

Assim, constrói-se o sistema linear dos dados de triangulação:

$$\begin{cases} y = a_1x + b_1 \\ y = a_2x + b_2 \\ \vdots \\ y = a_nx + b_n \end{cases} \quad (\text{B.8})$$

e converte-se para a forma matricial (eq. B.2), onde a matrix \mathbf{A} , \mathbf{b} e \mathbf{x} estão descritas logo abaixo:

$$\mathbf{A} = \begin{bmatrix} a_1 & -1 \\ \vdots & \vdots \\ a_n & -1 \end{bmatrix} \quad (\text{B.9})$$

$$\mathbf{b} = \begin{bmatrix} b_1 \\ \vdots \\ b_n \end{bmatrix} \quad (\text{B.10})$$

Uma vez colocados os dados no formato matricial, tanto para o caso da triangulação quanto para trilateração, pode-se resolver o problema através da solução dos mínimos quadrados, como pode ser visualizado na eq. B.11.

$$\hat{\mathbf{x}} = (\mathbf{A}^T \mathbf{A})^{-1} \mathbf{A}^T \mathbf{b} \quad (\text{B.11})$$

REFERÊNCIAS

- ABDELSALAM, H.; OLARIU, S. Towards enhanced rssi-based distance measurements and localization in wsns. In: *INFOCOM Workshops 2009, IEEE*. [S.l.: s.n.], 2009. p. 1–2.
- AIPING, P. et al. A distributed localization scheme for wireless sensor networks based on bounding box algorithm. In: *Electronic Measurement Instruments, 2009. ICEMI '09. 9th International Conference on*. [S.l.: s.n.], 2009. p. 2–984–2–988.
- ALIPPI, C.; MOTTARELLA, A.; VANINI, G. A RF map-based localization algorithm for indoor environments. In: *IEEE International Symposium on Circuits and Systems*. [S.l.: s.n.], 2005. p. 652–655.
- BAGGIO, A.; LANGENDOEN, K. Monte-carlo localization for mobile wireless sensor networks. *MSN 2006*, p. 317–328, 2006.
- BAHL, P.; PADMANABHAN, V. N. Radar: an in-building rf-based user location and tracking system. In: . [S.l.: s.n.], 2000. p. 775–784.
- BARBERIS, A.; BARBONI, L.; VALLE, M. Evaluating energy consumption in wireless sensor networks applications. In: *Digital System Design Architectures, Methods and Tools, 2007. DSD 2007. 10th Euromicro Conference on*. [S.l.: s.n.], 2007. p. 455–462.
- BULUSU, N.; HEIDEMANN, J.; ESTRIN, D. Gps-less low-cost outdoor localization for very small devices. *Personal Communications, IEEE*, v. 7, n. 5, p. 28–34, oct 2000. ISSN 1070-9916.
- CAPKUN, S.; HAMDI, M.; HUBAUX, J. GPS-free positioning in mobile Ad-Hoc networks. In: *In Proceedings of the 34th Hawaii International conference on System Sciences*. [S.l.: s.n.], 2001.
- CHAGAS, S.; MARTINS, J.; OLIVEIRA, L. de. Genetic algorithms and simulated annealing optimization methods in wireless sensor networks localization using artificial neural networks. In: *Circuits and Systems (MWSCAS), 2012 IEEE 55th International Midwest Symposium on*. [S.l.: s.n.], 2012. p. 928–931. ISSN 1548-3746.
- CHAGAS, S. H.; MARTINS, J. B.; OLIVEIRA, L. L. de. An approach to localization scheme of wireless sensor networks based on artificial neural networks and genetic algorithms. In: *New Circuits and Systems Conference (NEWCAS), 2012 IEEE 10th International*. [S.l.: s.n.], 2012. p. 137–140.
- CHENG, X. et al. Tps: A time-based positioning scheme for outdoor wireless sensor networks. In: *IEEE INFOCOM*. [S.l.: s.n.], 2004. p. 2685–2696.

CHO, J. et al. Modeling and inverse controller design for an unmanned aerial vehicle based on the self-organizing map. *Neural Networks, IEEE Transactions on*, v. 17, n. 2, p. 445–460, march 2006.

COAKLEY, J. R.; BROWN, C. E. Artificial neural networks in accounting and finance: modeling issues. *Intelligent Systems in Accounting, Finance and Management*, John Wiley and Sons, Ltd., v. 9, n. 2, p. 119–144, 2000. ISSN 1099-1174.

DOHERTY, L.; PISTER, K.; GHAOUI, L. E. Convex position estimation in wireless sensor networks. In: *INFOCOM 2001. Twentieth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*. [S.l.: s.n.], 2001. v. 3, p. 1655–1663 vol.3. ISSN 0743-166X.

GIROD, L.; ESTRIN, D. Robust range estimation using acoustic and multimodal sensing. In: *Proceedings of IROS*. [S.l.: s.n.], 2001.

HE, T. et al. Range-free localization schemes for large scale sensor networks. In: *Proceedings of the 9th annual international conference on Mobile computing and networking*. New York, NY, USA: ACM, 2003. (MobiCom '03), p. 81–95. ISBN 1-58113-753-2.

HIGHTOWER, J. et al. Spoton: An indoor 3d location sensing technology based on rf signal strength. Seattle,WA, February 2000.

HOLLAND, J. H. *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control and Artificial Intelligence*. [S.l.]: University of Michigan Press, 1975.

HU, L.; EVANS, D. Localization for Mobile Sensor Networks. In *Tenth Annual International Conference on Mobile Computing and Networking*, 2004.

JAUREGUI, S.; SILLER, M. A big picture on localization algorithms considering sensor logic location. In: *Systems, Man and Cybernetics, 2009. SMC 2009. IEEE International Conference on*. [S.l.: s.n.], 2009. p. 734–739. ISSN 1062-922X.

JI, X.; ZHA, H. Robust sensor localization algorithm in wireless ad-hoc sensor networks. In: *Computer Communications and Networks, 2003. ICCCN 2003. Proceedings. The 12th International Conference on*. [S.l.: s.n.], 2003. p. 527–532. ISSN 1095-2055.

KHAN, J. et al. Classification and diagnostic prediction of cancers using gene expression profiling and artificial neural networks. *Nature Medicine*, Nature Publishing Group, v. 7, n. 6, p. 673–679, 2001.

KIRKPATRICK, S.; GELATT, J. C. D.; VECCHI, M. P. Optimization by simulated annealing. *Science*, v. 220, p. 671–680, 1983.

LORINCZ, K.; WELSH, M. *A robust decentralized approach to RF-based location tracking*. [S.l.], 2004.

MENDYK, A.; JACHOWICZ, R. Unified methodology of neural analysis in decision support systems built for pharmaceutical technology. *Expert Syst. Appl.*, Pergamon Press, Inc., Tarrytown, NY, USA, v. 32, n. 4, p. 1124–1131, maio 2007.

- MUSCEDERE, R. et al. Efficient conversion from binary to multi-digit multi-dimensional logarithmic number systems using arrays of range addressable look-up tables. *Application-Specific Systems, Architectures and Processors, IEEE International Conference on*, IEEE Computer Society, Los Alamitos, CA, USA, v. 0, p. 130, 2002. ISSN 1063-6862.
- NI, L. M. et al. Landmarc: Indoor location sensing using active RFID. *Wireless Networks*, Kluwer Academic Publishers, Hingham, MA, USA, v. 10, n. 6, p. 701–710, 2004.
- NICULESCU, D.; NATH, B. Ad hoc positioning system (aps) using aoa. In: *INFOCOM 2003. Twenty-Second Annual Joint Conference of the IEEE Computer and Communications Societies. IEEE Societies*. [S.l.: s.n.], 2003. v. 3, p. 1734 – 1743 vol.3. ISSN 0743-166X.
- OLIVEIRA, L. L. d. et al. CentroidM: a Centroid-based Localization Algorithm for Mobile Sensor Networks. *Symposium on Integrated Circuits and Systems Design*, p. 204–209, 2010.
- ORTEGA, A.; SILVA, I. Neural network model for designing automotive devices using smd led. *International Journal of Automotive Technology*, The Korean Society of Automotive Engineers, v. 9, p. 203–210, 2008. ISSN 1229-9138.
- PARSONS, S.; JONES, G. Acoustic identification of twelve species of echolocating bat by discriminant function analysis and artificial neural networks. *Journal of Experimental Biology*, Company of Biologists, v. 203, n. Pt 17, p. 2641–2656, 2000.
- PRIYANTHA, N. B.; CHAKRABORTY, A.; BALAKRISHNAN, H. The cricket location-support system. In: *Proceedings of the 6th annual international conference on Mobile computing and networking*. New York, NY, USA: ACM, 2000. (MobiCom '00), p. 32–43. ISBN 1-58113-197-6. Disponível em: <<http://doi.acm.org/10.1145/345910.345917>>.
- RAHMAN, M. S.; PARK, Y.; KIM, K.-D. Localization of wireless sensor network using artificial neural network. In: *Proceedings of the 9th international conference on Communications and information technologies*. Piscataway, NJ, USA: IEEE Press, 2009. (ISCIT'09), p. 639–642. ISBN 978-1-4244-4521-9.
- RAO, S. S. *Engineering Optimization: Theory and Practice*. [S.l.]: New Age Publishers, 1996.
- RAPPAPORT, T. *Wireless Communications: Principles and Practice*. 2nd. ed. Upper Saddle River, NJ, USA: Prentice Hall PTR, 2001. ISBN 0130422320.
- SHANG, Y.; RUML, W. Improved mds-based localization. In: *INFOCOM 2004. Twenty-third Annual Joint Conference of the IEEE Computer and Communications Societies*. [S.l.: s.n.], 2004. v. 4, p. 2640 – 2651 vol.4. ISSN 0743-166X.
- SHANG, Y. et al. Localization from mere connectivity. In: *Proceedings of the 4th ACM international symposium on Mobile ad hoc networking & computing*. New York, NY, USA: ACM, 2003. (MobiHoc '03), p. 201–212. ISBN 1-58113-684-6. Disponível em: <<http://doi.acm.org/10.1145/778415.778439>>.
- SHAREEF, A.; ZHU, Y.; MUSAVI, M. Localization using neural networks in wireless sensor networks. In: *MOBILWARE'08: Proceedings of the 1st international conference on MOBILE Wireless MiddleWARE, Operating Systems, and Applications*. ICST, Brussels, Belgium, Belgium: ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 2007. p. 1–7. ISBN 978-1-59593-984-5.

- SHAREEF, A. et al. Comparison of mlp neural network and kalman filter for localization in wireless sensor networks. In: *Proceedings of the 19th IASTED International Conference on Parallel and Distributed Computing and Systems*. Anaheim, CA, USA: ACTA Press, 2007. (PDCS '07), p. 323–330. ISBN 978-0-88986-704-8. Disponível em: <<http://dl.acm.org/citation.cfm?id=1647539.1647599>>.
- SILVA, I. D.; SPATTI, D.; FLAUZINO, R. *Redes neurais artificiais para engenharia e ciências aplicadas - curso prático*. [S.l.]: ARTLIBER, 2010. ISBN 9788588098534.
- SIMON, G. et al. Simulation-based optimization of communication protocols for large-scale wireless sensor networks. In: *Aerospace Conference, 2003. Proceedings. 2003 IEEE*. [S.l.: s.n.], 2003. v. 3, p. 1339 – 1346. ISSN 1095-323X.
- SMITH, A. et al. Tracking Moving Devices with the Cricket Location System. In: *2nd International Conference on Mobile Systems, Applications and Services (Mobisys 2004)*. Boston, MA: [s.n.], 2004.
- SUMAN, B. Study of simulated annealing based algorithms for multiobjective optimization of a constrained problem. *Computers and Chemical Engineering*, April 2004.
- TIAN, H.; SHANG, Z. Artificial neural network as a classification method of mice by their calls. *Proceedings of Ultrasonics International (UI '05) and World Congress on Ultrasonics (WCU)*, v. 44, Supplement, n. 0, p. e275 – e278, 2006.
- TIAN, J.-P.; SHI, H.-C. Study of localization scheme base on neural network for wireless sensor networks. In: *Wireless, Mobile and Sensor Networks, 2007. (CCWMSN07). IET Conference on*. [S.l.: s.n.], 2007. p. 64 –67. ISSN 0537-9989.
- TOO, G.-P.; CHEN, S.; HWANG, S. Inversion for acoustic impedance of a wall by using artificial neural network. *Applied Acoustics*, v. 68, n. 4, p. 377 – 389, 2007. ISSN 0003-682X.
- WANT, R. et al. The active badge location system. *ACM Trans. Inf. Syst.*, ACM, New York, NY, USA, v. 10, p. 91–102, jan. 1992. ISSN 1046-8188.
- ZHANG, Q.; STANLEY, S. J. Real-time water treatment process control with artificial neural networks. *Journal of Environmental Engineering*, ASCE, v. 125, n. 2, p. 153–160, 1999.
- ZHANG, Z.; FRIEDRICH, K. Artificial neural networks applied to polymer composites: a review. *Composites Science and Technology*, v. 63, n. 14, p. 2029 – 2044, 2003. ISSN 0266-3538.