

# Evolução de Redes Neurais para Localização de Robôs Móveis usando Redes Sem Fio

Gustavo Pessin<sup>1</sup>, Fernando Osório<sup>1</sup>, Denis Wolf<sup>1</sup>, Patrícia Vargas<sup>2</sup>, Jó Ueyama<sup>1</sup>

<sup>1</sup>Instituto de Ciências Matemáticas e de Computação (ICMC)  
Universidade de São Paulo (USP) – São Carlos, SP, Brasil

<sup>2</sup>School of Mathematical and Computer Sciences (MACS)  
Heriot-Watt University – Edinburgh, UK

{pessin, fosorio, denis, joueyama}@icmc.usp.br, p.a.vargas@hw.ac.uk

**Resumo.** *O uso de robôs móveis em ambientes internos usualmente tem como dependência a capacidade de localização, que é uma informação fundamental para o desenvolvimento de diversas tarefas úteis, como navegação autônoma. Diversos métodos para localização em ambientes internos têm sido propostos, entretanto, ainda apresentam sérias deficiências (em termos de qualidade) ou exigências (em termos de tempo de processamento e uso de recursos). O uso de redes sem fio (p. ex. Wireless LAN IEEE 802.11 b/g), por meio da potência do sinal, pode ser um método simples e barato para obtenção de localização, sendo utilizado como uma metáfora do sistema de GPS. Neste artigo, apresentamos uma investigação sobre a evolução de redes neurais utilizadas para o aprendizado da localização de um robô móvel. As entradas da rede neural são a potência obtida de pontos de acesso. Mostramos como um método evolutivo pode auxiliar na obtenção de redes neurais eficientes para prover localização.*

## 1. Introdução

Localização é uma informação fundamental para a realização de diversas tarefas nas áreas de robótica móvel [Siegwart and Nourbakhsh 2004] e de redes de sensores sem fio [Dargie and Poellabauer 2010]. Ainda, o conhecimento sobre localização pode também ser usado para rastrear animais, pessoas (p. ex. praticando esportes) ou veículos (p. ex. em um estacionamento). Embora sendo um sistema de localização popular e podendo ser utilizado para a determinação da posição absoluta, o *Global Positioning System* (GPS) é adequado para ambientes externos não possuindo um bom desempenho em ambientes internos [Gwon et al. 2004].

Em ambientes internos, sensores como *lasers* e câmeras podem ser utilizados para a estimação de posição [Napier et al. 2010]. Entretanto, tais sensores usualmente necessitam de marcos (*landmarks*) no ambiente e dependem de algoritmos que necessitam de grande poder computacional. Ainda, tais sensores têm campos de visão limitados, o que dificulta a tarefa de obtenção de posição. No caso de câmeras, a variação na iluminação é ainda um outro grande problema. Um tipo de sensor que não depende de grande poder computacional são os do tipo *encoders* [Martinelli 2002], que provêm odometria. Odometria é realmente útil em diversos casos porém apresenta um erro incremental que não pode ser ignorado na maioria dos sistemas reais. Uma alternativa que tem recebido grande atenção nos últimos anos é a odometria visual, que tem provido algoritmos cada vez mais

eficientes, embora apresentem custo computacional altíssimo, dependam de câmeras de alta resolução e também sofram com os problemas de iluminação. O estudo de localização interna, usando redes sem fio, tem sido uma área de pesquisa com crescente atenção. O uso de diferentes tecnologias (p. ex. *Wireless LAN*, *Bluetooth*, *ZigBee*) e diferentes métricas (p. ex. tempo, potência) tem provido soluções cada vez mais interessantes. É mais comum o estudo de localização em grandes áreas, como prédios, onde o objetivo é saber em que sala um dispositivo se encontra. Tal abordagem, em geral, utiliza o sinal da rede sem fio utilizado como uma metáfora do modelo de GPS. Além disso, redes sem fio convencionais são dispositivos baratos e que poderiam ser uma fonte eficiente de localização. Por fazer uso de algoritmos relativamente simples, o custo computacional esperado é baixo.

Diversos trabalhos realizados em laboratórios de robótica apresentam o uso e a avaliação de redes neurais artificiais aplicadas nas diversas áreas da robótica. Entre estes trabalhos podemos citar trabalhos sobre mapeamento [Hata et al. 2009], trabalhos sobre localização [Pessin et al. 2011], e trabalhos sobre navegação [Souza et al. 2013]. A avaliação de diferentes redes neurais se faz importante pelo fato de que o aprendizado é relacionado à topologia da rede. Nem toda rede neural apresenta capacidade de aprendizado e nem toda rede neural grande é melhor que uma rede menor. Entretanto, escolher e avaliar topologias de forma manual tem se mostrado um trabalho relativamente árduo e que consome tempo considerável. Isso motivou o desenvolvimento e a investigação de um algoritmo genético para busca (evolução) de topologias. O objetivo principal é investigar o comportamento do AG na evolução de topologias e compará-lo com as topologias previamente obtidas pelo processo manual. Embora sendo um processo estocástico, o AG desenvolvido deve ser capaz de evoluir redes neurais com topologias estatisticamente similares.

Assim, este artigo tem como objetivo (i) a proposta e avaliação de uma arquitetura para localização de robôs móveis usando o valor da potência de redes tipo *Wireless LAN IEEE 802.11 b/g* e aprendizado de máquina (por meio de RNAs), (ii) a investigação sobre evolução de topologias de RNAs, e (iii) a investigação sobre evolução de pesos de conexões nas RNAs. O restante do artigo é dividido na seguinte estrutura: A primeira seção descreve sucintamente aspectos de aprendizado de máquina. A segunda seção apresenta trabalhos relacionados seguido da metodologia. Em seguida são descritos os resultados e as discussões, detalhando as investigações realizadas. Por fim, são apresentadas as considerações finais e os trabalhos futuros.

## 2. Inteligência computacional

Algoritmos Genéticos (AG) [Holland 1975, Mitchell 1998] são técnicas de otimização que tem como inspiração a teoria da evolução pela seleção natural. O primeiro passo em um AG típico é a geração de uma população inicial, usualmente formada por um conjunto aleatório de valores que representam possíveis soluções do problema. Durante o processo evolutivo, cada indivíduo da população é avaliado por meio de uma função de aptidão (*fitness*) que, em geral, deve permitir manter os indivíduos mais aptos e descartar os menos aptos. Nos indivíduos selecionados, é feito cruzamento e mutação, gerando descendentes para a próxima geração [Goldberg 1989]. Este processo é repetido até que a condição de parada seja satisfeita. AGs são estruturados de forma que as informações referentes a um determinado sistema possam ser codificadas de maneira análoga a cro-

mossomos biológicos, como uma sequência de valores, onde usualmente cada fragmento da sequência representa uma variável. Aplicações com AGs podem ser vistas nos seguintes trabalhos: [Orlando et al. 2010] apresenta o desenvolvimento de AGs para realizar planejamento de trajetória de dispositivos como braços robóticos com diversas juntas. O trabalho [Jun and Qingbao 2010] apresenta o desenvolvimento de AGs para realizar navegação robótica.

Uma Rede Neural Artificial (RNA) é representada por uma coleção de unidades (neurônios) conectadas por ligações com pesos (sinapses). Unidades de entrada e saída recebem e transferem os sinais do ambiente e para o ambiente. Unidades internas são chamadas de ocultas, dado que elas não tem contato com o ambiente externo da RNA [Nolfi and Floreano 2000]. Os atributos básicos de uma rede neural podem ser divididos em arquitetura e neurodinâmica. A arquitetura determina a estrutura da rede, ou seja, o número de neurônios e sua interconectividade e a neurodinâmica, por sua vez, define as propriedades funcionais da rede, ou seja, como ela aprende, recupera, associa e compara novas informações com o conhecimento já armazenado [Kartalopoulos 1995]. Matematicamente, RNAs são aproximadores universais, que realizam mapeamentos em espaços de funções multivariáveis [Hornik et al. 1989]. O número de neurônios na camada oculta de uma rede neural tem relação com a sua capacidade de aprendizado e generalização. Poucos neurônios podem conduzir ao chamado *underfitting* (deficiência no aprendizado) enquanto o excesso de neurônios pode conduzir ao *overfitting* (aprendizado excessivo com perda de capacidade de generalização – a.k.a. decorar os dados) além de desperdício de recursos computacionais [Teoh et al. 2006, Huang 2003]. É comum encontrar aplicações de RNAs em problemas que necessitem aproximação, predição ou classificação, como nos exemplos citados na introdução.

### 3. Trabalhos relacionados com localização

[Elnahrawy et al. 2004] apresenta que a obtenção de boas performances em localização usando redes sem fio depende de configurações específicas do ambiente. Isso significa que diferentes abordagens devem ser utilizadas em diferentes ambientes, como usando diferentes tipos de sinais ou filtros. Localização em grandes prédios podem envolver diferentes requisitos, mais do que no caso da localização em pequenas áreas. Essas dificuldades podem ser relacionadas com a atenuação e reflexão do sinal em paredes e objetos móveis bem como outras fontes de interferência [Ladd et al. 2004]. Usualmente, o sinal de uma rede sem fio propaga em uma forma radial. O sinal pode ser direcional dependendo do tipo de antena que está sendo usado. A potência do sinal decresce de acordo com a distância da fonte. Por meio de trilateração e usando pelo menos três fontes, pode ser utilizada uma metáfora do modelo de GPS para obter a posição de um nodo e, da mesma forma, estimar outras informações como orientação e velocidade caso o objeto esteja em movimento. Entretanto, ao contrário do GPS, o sinal da rede sem fio apresenta grande instabilidade e sofre mais com interferências [Gast 2002].

[Maisano et al. 2008] descreve aspectos técnicos de um método chamado de *indoor GPS* (iGPS). A abordagem proposta apresenta como leituras com base em luz estruturada podem ser utilizadas para rastrear elementos em uma área delimitada. O sistema tem como princípio básico de funcionamento o cálculo da posição por meio do ângulo relativo (ângulo horizontal – azimute e ângulo vertical – elevação) dos emissores aos receptores. Baseado no conhecimento da posição inicial dos emissores, ob-

tidas na fase de inicialização/calibragem do sistema o sistema pode calcular a posição dos sensores (fixado junto ao objeto a ser localizado). Alguns pontos críticos do sistema são a necessidade de contato visual entre emissores e receptores e uma fase inicial de calibragem relativamente trabalhosa. [Franceschini et al. 2009] apresenta um sistema de localização/rastreamento chamado de *Mobile Spatial Coordinate Measuring System* (MScMS) proposto para localização de objetos de larga escala (dimensões na ordem de dezenas de metros). Apresenta uma natureza distribuída, com uma coleção de sensores *Bluetooth*, e faz uso de *Time Difference of Arrival* (TDoA) [Gustafsson and Gunnarsson 2003] para calcular a posição por meio de uma série de equações. Nossa abordagem utilizando Wi-Fi usa um ambiente relativamente similar aos propostos por [Maisano et al. 2008, Welch et al. 2001, Franceschini et al. 2009] entretanto, nossa abordagem faz uso de robôs móveis e pretende ser uma opção de maior acuracidade e também de calibragem automática pelo robô móvel. Buscamos avaliar o uso de Wi-Fi pelo fato do mesmo ser mais barato e prático de implementar.

[Perkins et al. 2011] apresenta uma comparação entre resoluções obtidas com uso de TDoA e o *Received Signal Strength Indication* (RSSI). Utilizando TDoA a resolução obtida ficou em 1,1 cm e utilizando RSSI a resolução obtida ficou em 2,4 cm. Entretanto, [Hara and Anzai 2008] apresenta outra comparação entre TDoA e RSSI para localização, onde a conclusão aponta que o RSSI é menos sensível a mudanças do ambiente, sendo mais indicado para ambientes dinâmicos. Nossas avaliações fazem uso do RSSI pelos seguintes motivos: (i) é um valor disponível em praticamente todas as placas de redes atuais, (ii) não depende de hardware adicional, (iii) apresentou melhores resultados em ambientes dinâmicos em comparação com o TDoA [Hara and Anzai 2008], (iv) apresentou melhores resultados em relação a outras métricas simples de tempo [Marinello et al. 2012].

#### 4. Metodologia

O modelo proposto para localização tem como objetivo o aprendizado da posição ( $x, y$ ) de um robô por meio de uma rede neural em que as entradas são o sinal da potência de APs. A Figura 1 apresenta a configuração proposta. Embora utilizando neste experimento o robô NAO (Aldebaran Robotics), esta metodologia pode ser aplicada em qualquer outro tipo de dispositivo que suporte rede *wireless*. O sinal obtido da rede sem fio é o *Received Signal Strength Indication* (RSSI). Este valor é obtido por meio do comando do GNU/Linux *iwlist* (usado como *iwlist <interface> scanning*). Como usamos o comando *iwlist*, não existe a necessidade de conexão ou mesmo qualquer tipo de *login* nos diferentes APs. A leitura do sinal das redes com independência de conexão provê informação suficiente para o sistema e torna o sistema mais fácil de usar, leve e flexível. RSSI é uma métrica de potência presente em sinais recebidos de fontes de rádio. Praticamente qualquer dispositivo de rádio implementa em seu circuito a possibilidade de entrega deste valor sem necessidade de desenvolvimento ou adaptação adicional [Gast 2002]. A coleta de dados se dá utilizando o robô dentro do plano apresentado (Figura 1(b)) e coletando 3 minutos de leituras (i.e.  $\approx 180$  leituras) em cada ponto marcado. Com um deslocamento de 60 cm, mapeando um plano de 180 cm x 180 cm, isso significa 16 pontos para ler resultando num total de  $\approx 2880$  leituras.

Nossa abordagem se baseia na capacidade de aprendizado e generalização de redes neurais artificiais a fim de reduzir o efeito de instabilidade nas entradas. Entretanto, os valores obtidos dos APs são muito ruidosos, o que, em avaliações iniciais, apresentou que

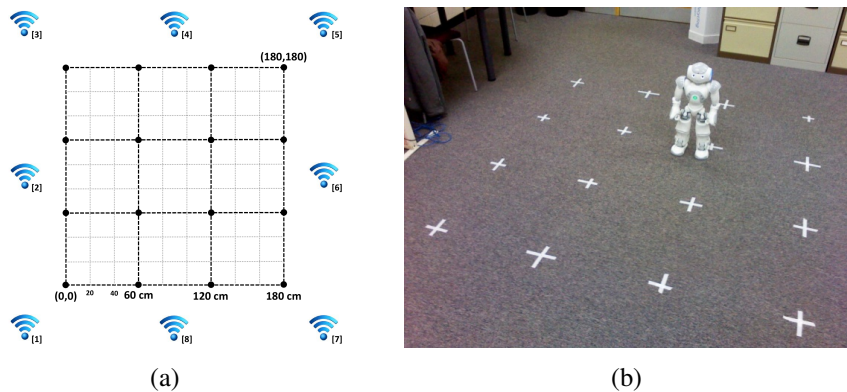


Figura 1. (a) Representação gráfica da área de trabalho considerando uma área de 180 cm x 180 cm. (b) Foto parcial do ambiente com o robô, similar a representação gráfica da Figura 1(a). Cada pequena cruz está posicionada a 60 cm uma da outra. Em (a), a posição dos APs é aproximada.

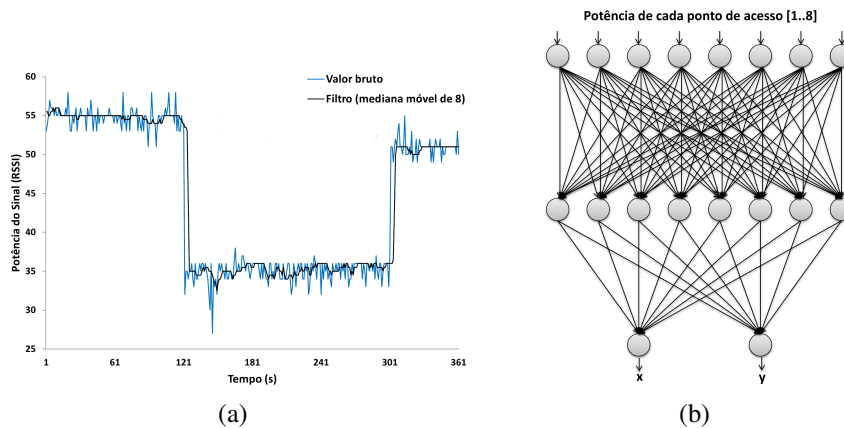
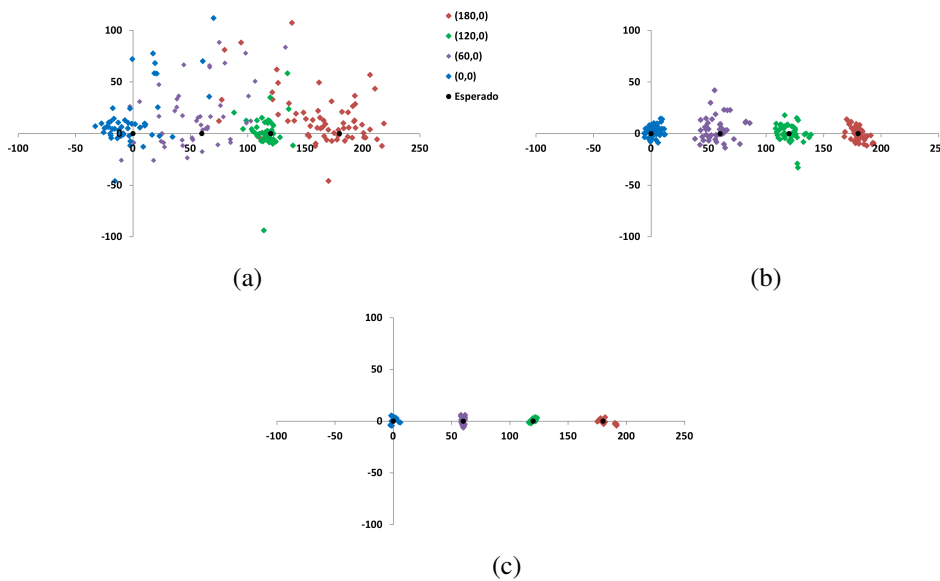


Figura 2. (a) Exemplo de comportamento do filtro de mediana móvel – A linha azul apresenta o valor bruto da leitura de um ponto de acesso. A linha preta apresenta como o filtro de mediana móvel remove parte dos ruídos (janela de 8 leituras). (b) Exemplo de topologia de RNA utilizada.

o aprendizado da rede obteve baixa acuracidade. Isso motivou o estudo e avaliação de filtros. Uma avaliação inicial de filtros de média móvel e mediana móvel realizada por [Pessin et al. 2011] apresentou que o filtro de mediana, com mesmo tamanho de janela do filtro de média, permitia um aprendizado um pouco mais eficiente da RNA. Acredita-se que isso seja devido a capacidade da mediana de ignorar *outliers*. O comportamento do filtro pode ser entendido com a Figura 2(a), onde duas linhas representam (em azul) a leitura do valor bruto de um ponto de acesso e (em preto) o resultado do filtro de mediana móvel com janela de 8 leituras.

Em relação ao filtro, a Figura 3 apresenta uma secção do plano (Figura 1) com valores esperados e obtidos usando diferentes janelas de tempo no filtro de mediana móvel. São apresentados os valores usando o valor bruto, usando filtro de mediana de 10 leituras e usando filtro de mediana de 60 leituras. Podemos ver o resultado dos erros no plano de trabalho proposto. Assim podemos ver que o filtro permite que a RNA tenha um aprendizado cada vez mais acurado. Este gráfico é obtido de [Pessin 2013], onde uma



**Figura 3. Secção do plano (Figura 1) com valores esperados e obtidos usando diferentes janelas de tempo na mediana. (a) Valor bruto. (b) Usando filtro de mediana de 10 leituras. (c) Usando filtro de mediana de 60 leituras. Os eixos  $x$  e  $y$  estão em centímetros.**

avaliação sobre filtros também é apresentada.

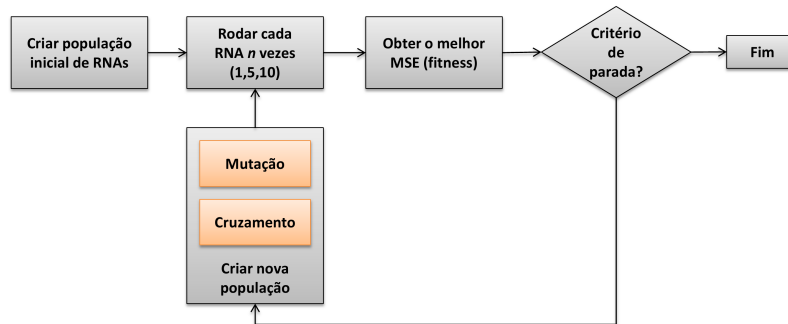
O número de neurônios na camada de entrada é equivalente ao número de APs disponíveis. Assim, como utilizamos 8 APs, existem 8 neurônios de entrada na RNA, como mostra a Figura 2(b). No nosso modelo, a ordem de entrada é importante, assim, o ponto de acesso um é ligado ao neurônio um, o ponto de acesso dois ligado ao neurônio dois e assim por diante. Esse controle é feito por meio do endereço físico (MAC) dos APs. Tal informação provém do escaneamento dos APs. As saídas da rede são dois valores, as coordenadas  $(x, y)$  da posição do robô. O erro é medido por meio da distância euclidiana ( $d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$ ). O valor  $d$  é o erro (distância, em centímetros),  $(x_1, y_1)$  são os valores esperados da base de validação da RNA e  $(x_2, y_2)$  são os valores obtidos enquanto usando a RNA. Um exemplo de topologia pode ser visto na Figura 2(b). Aspectos pontuais de metodologia serão descritos no início das seções de investigação, detalhadas na sequência.

## 5. Evolução de topologias de redes neurais para localização

Avaliamos um método evolutivo (algoritmo genético) para buscar diferentes topologias de redes neurais de forma automática para o problema de localização usando rede sem fio<sup>1</sup>. A ideia básica deste AG pode ser entendida com a Figura 4. Neste AG evoluímos topologia da RNA, ou seja, o número de neurônios e o número de camadas ocultas. Os parâmetros padrão para evolução são de 1 a 3 camadas de neurônios e de 0 a 50 neurônios em cada camada. Desta forma, o cromossomo é representado por um vetor de três posições. Tal cromossomo simples tem como objetivo tornar o espaço de busca menor. Para ter uma melhor dispersão na inicialização da população, a mesma é realizada usando a seguinte heurística: (i) um terço da população é inicializado com uma camada, ou seja,

<sup>1</sup>Código-fonte disponível em <http://goo.gl/jubGr>

um número de 1 a 50 na primeira camada e zero nas outras; (ii) outro um terço da população é inicializado com duas camadas, ou seja, números entre 1 e 50 nas duas primeira camadas e zero na terceira; (iii) o um terço restante é inicializado com 3 camadas.



**Figura 4. Fluxograma do AG responsável pela evolução da topologia da RNA.**

Quanto as configurações do AG, utilizamos 15 indivíduos e 20 gerações. Para uma melhor dispersão na inicialização, além da heurística de 1/3 para cada camada, também fizemos a inicialização da rede com múltiplos de 5. Quanto ao cromossomo, um vetor com {10, 5, 25} produzira uma rede neural com 3 camadas ocultas na respectiva ordem do vetor. E, um vetor com {10, 0, 5} produzira uma rede neural com 2 camadas, desconsiderando o zero. O AG é responsável pela topologia da RNA sendo o treino da rede (atualização dos pesos) realizado pelo SNNS<sup>2</sup>. Assim, para cada indivíduo – que é a representação de uma topologia – o processo de obtenção do *fitness* depende de chamadas ao SNNS. O *fitness* que o AG usa como parâmetro de evolução é o MSE da base de validação das redes treinadas. Inicialmente, utilizamos no SNNS o algoritmo de aprendizado *rprop* por 20.000 ciclos e fazemos validação a cada 200 ciclos. Percorremos os valores de validação procurando pelo ponto ótimo de generalização.

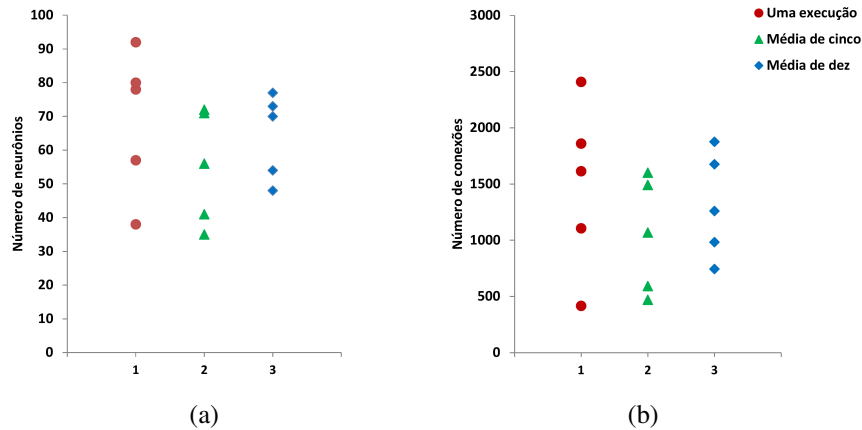
O critério de parada do AG é o número de gerações. O esquema de mutação desenvolvido é o modelo gaussiano, onde uma mudança no número de neurônios ocorre de acordo com uma distribuição normal com média no valor do gene (quantidade de neurônios) e um desvio padrão baixo. O que faz com que novos valores caiam na vizinhança do valor, evitando mudanças muito bruscas. A probabilidade de mutação é definida em 20 por cento. O esquema de cruzamento (*crossover*) é feito em três diferentes formas: com probabilidade randômica, um indivíduo pode trocar genes com o (i) melhor de todos, (ii) com o segundo melhor ou (iii) pode fazer a média entre seus neurônios e os neurônios do melhor indivíduo. O cruzamento ocorre em todos os indivíduos, com exceção dos dois melhores de cada geração, onde usamos o conceito de elitismo – mantendo os dois melhores indivíduos da população inalterados.

Considerando a existência de 3 camadas ocultas com até 50 neurônios em cada camada, poderíamos ter 50<sup>3</sup> (125.000) diferentes topologias. Usando o AG podemos reduzir e espaço de busca, neste caso, 15 indivíduos e 20 gerações resulta em 15×20 comparações, um total de 300 avaliações. A questão que fica em aberto aqui é: quantas vezes deve-se executar a rede neural a fim de minimizar a influência dos números aleatórios da inicialização dos pesos? Para minimizar a influência da inicialização, avaliamos três tipos

<sup>2</sup><http://www.ra.cs.uni-tuebingen.de/SNNS>

**Tabela 1. Topologias obtidas pela evolução. Resultado de 5 rodadas com cada conjunto de configurações.**

		Topologia obtida
AG usando <i>fitness</i> como uma única execução da RNA	$E_{1a}$	32 x 25
	$E_{1b}$	50 x 35 x 07
	$E_{1c}$	13 x 30 x 35
	$E_{1d}$	30 x 40 x 10
	$E_{1e}$	10 x 28
AG usando <i>fitness</i> como média de 5 execuções	$E_{5a}$	47 x 25
	$E_{5b}$	50 x 21
	$E_{5c}$	25 x 10
	$E_{5d}$	30 x 11
	$E_{5e}$	33 x 23
AG usando <i>fitness</i> como média de 10 execuções	$E_{10a}$	18 x 30
	$E_{10b}$	35 x 19
	$E_{10c}$	41 x 36
	$E_{10d}$	20 x 50
	$E_{10e}$	45 x 28



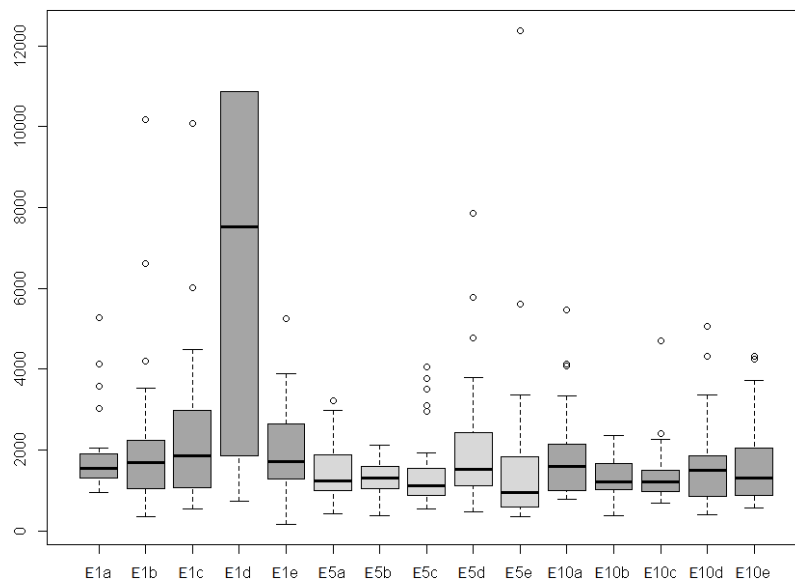
**Figura 5. Topologias obtidas pela evolução. (a) Gráfico de pontos do número de neurônios. (b) Gráfico de pontos do número de conexões. Podemos ver que em ambos os valores, a evolução baseada em apenas uma execução da rede neural apresentou a maior dispersão nos resultados.**

de evolução: (i) AG evoluindo baseado em apenas uma execução de cada RNA; (ii) AG evoluindo baseado em 5 execuções de cada RNA – o *fitness* então não é mais o MSE da rede, mas sim a média das 5 execuções; (iii) AG evoluindo baseado em 10 execuções de cada rede – o *fitness* então passa a ser a média de 10 execuções.

O uso de média de cinco ou dez execuções da rede dá mais importância para redes com menor dispersão no resultados, diminuindo a influência da aleatoriedade nos resultados. Enquanto o uso de apenas uma execução dá mais importância a uma única rede boa, mas que pode apresentar alta dispersão nos resultados quando treinada novamente com outra semente de inicialização de pesos. Buscamos avaliar dois aspectos chaves nessa avaliação: (i) quantas vezes devemos rodar cada RNA a fim de ter resultados mais homogêneos; e (ii) quantos ciclos de treino e teste são necessários para obter resultados satisfatórios.

As topologias obtidas pela evolução podem ser vistas na Tabela 1. Podemos ver

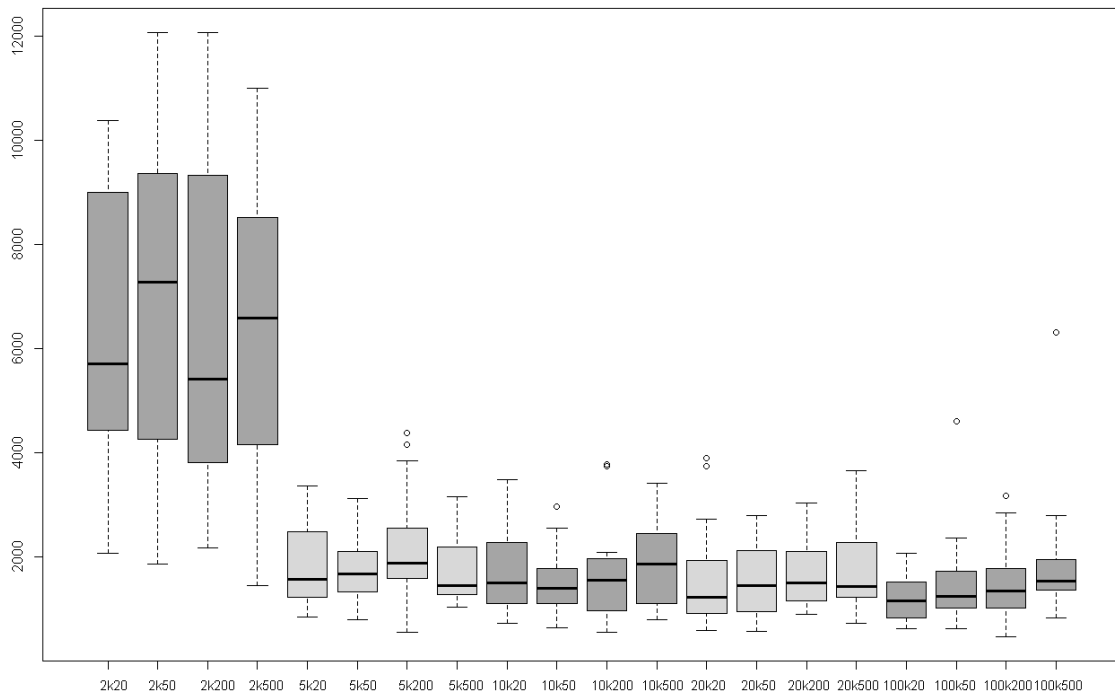




**Figura 6. Resultados de 30 execuções de cada RNA obtida pela evolução – redes apresentadas na Tabela 1. O erro (eixo y) está mostrado como MSE.**

que os resultados de uma única execução da rede teve resultados com redes de 2 e 3 camadas. Enquanto o resultado de média de cinco e dez apresentou resultados apenas com duas camadas. A Figura 5 apresenta o gráfico de pontos do número de neurônios e de conexões resultantes da evolução. Podemos ver que em ambos os valores, a evolução baseada em apenas uma execução da rede neural apresentou a maior dispersão nos resultados. Buscando uma comparação estatisticamente válida sobre os resultados da evolução, executamos cada rede neural 30 vezes. O resultado pode ser visto na Figura 6. Podemos ver que os resultados das redes neurais que foram evoluídas com base em uma única execução são os que apresentam maior dispersão. A análise estatística (*Kruskal-Wallis rank sum test*) sobre o conjunto de dados de 5 e 10 (Figura 6) apresenta p-valor de 0,075 ( $>0,05$ ). Ou seja, os conjuntos 5 e 10 não apresentam diferenças significativas usando 95% de confiança. Mas, considerando o conjunto dos resultados com uma execução, o valor-p resultante do teste (*Kruskal-Wallis rank sum test*) é de  $1,838e^{-09}$  ( $<0,05$ ). Ou seja, existem diferenças significativas entre os conjuntos de 1, 5 e 10 execuções, mas não entre 5 e 10 execuções.

Embora similares, os resultados usando média de 5 e 10 execuções que apresentaram as redes neurais com menor dispersão e menores médias e medianas são  $E_{5b}$  e  $E_{10b}$ . Essas redes têm respectivamente  $50 \times 21$  e  $35 \times 19$  neurônios e 1492 e 983 conexões cada (considerando oito entradas e duas saídas). Como descrito anteriormente, tais redes tiveram suas topologias evoluídas usando *rprop* por 20.000 ciclos e sendo validada a cada 200 ciclos. A questão que gostaríamos de responder agora é: faz diferença aumentar ou diminuir estes números? Para isso, alteramos o treino da rede para os seguintes ciclos: {2k, 5k, 10k, 20k e 100k}. E o número de ciclos de validação para {20, 50, 200, 500}. Selecionamos então a rede  $E_{10b}$  para o próximo passo de avaliação devido ao fato de possuir menos neurônios e conexões que a rede  $E_{5b}$ . Executamos 30 vezes a RNA com a nova combinação de ciclos de treino/validação. Os resultados podem ser vistos na Figura 7. Podemos ver que o resultado utilizando 2.000 ciclos (2k) de treino apresentou os

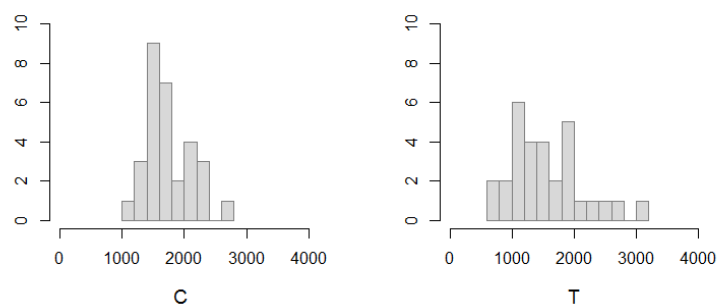


**Figura 7. Resultados de 30 execuções com variação no número de ciclos de treino e validação sobre a rede  $E_{10b}$ . O erro (eixo y) está mostrado como MSE. As referências no eixo x representam número de ciclos de treino e validação, por exemplo, 2k20 representa 2.000 ciclos de treino e validação a cada 20 ciclos.**

piores resultados. Isso leva a crer que esse número é muito pequeno para a rede apresentar aprendizado.

A análise estatística (*Kruskal-Wallis rank sum test*) sobre o conjunto de dados (desconsiderando os conjuntos com 2.000 ciclos) apresenta valor- $p$  de 0,00027 ( $<0,05$ ). Ou seja, os conjuntos apresentam diferenças significativas usando 95% de confiança. Dessa forma, buscamos avaliar quais membros da população seriam diferentes. Ainda, podemos ver na Figura 7 que o conjunto que apresentou menor mediana e também menor dispersão é o conjunto que utilizou 100.000 ciclos (100k) de treino com validação a cada 20 ciclos. Dessa forma, usamos esse valor como melhor valor e comparamos os demais valores com esse conjunto. Um teste de comparação estatístico, entre o aparente melhor conjunto (100k20) e todos os demais foi realizado. Foi possível ver que existe rejeição da igualdade entre os resultados de 100k20 com todas as redes treinadas por 5.000 ciclos (5k). Ainda, o mesmo ocorre com as redes treinadas por 10.000 ciclos (10k) com exceção da rede validada a cada 50 ciclos. Entretanto, tal valor- $p$  está muito próximo do limite de corte (0,051). Assim, consideramos como aceitável a rede treinada por 20k ciclos, que é cerca de 5 vezes mais rápida que a rede treinada por 100k ciclos. Algo como de 30 minutos ao contrário de 150 minutos. Assim, a não diferença entre os resultados de 100k20 e 20k20 nos faz crer que, para estes dados, o treino de 20k ciclos, validado a cada 20 ciclos é suficiente.

No passo final da avaliação sobre evolução de topologias buscamos comparar a rede evoluída neste processo com a rede obtida por processo manual em [Pessin et al. 2011]; para tanto, treinamos a topologia proveniente de [Pessin et al. 2011]



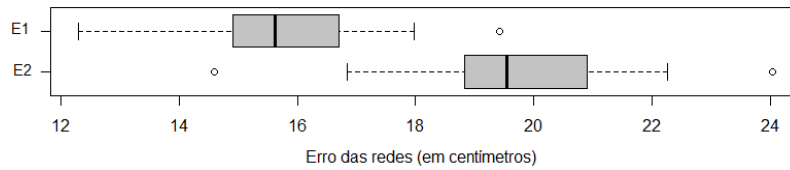
**Figura 8. Resultado da comparação entre a rede obtida em [Pessin et al. 2011] (C) e a rede evoluída neste trabalho (T). Resultado de 30 execuções. Erro (eixo x) apresentado como MSE. Resultados estatisticamente semelhantes.**

(referenciada como C) com o *rprop* com 20k ciclos sendo a validação feita a cada 20 ciclos (de acordo com o obtido nos passos anteriores) para comparar com a rede de topologia evoluída (referenciada como T). A Figura 8 apresenta o resultado da comparação. São apresentados resultados de 30 execuções de cada RNA, sendo o erro apresentado como MSE. O teste de adequação de normalidade (*Shapiro-Wilk normality test*) apresentou os seguintes valores-*p* para T e C: 0,334 e 0,085. Sendo assim, ambos os conjuntos são aceitos como distribuições normais e permitem um teste de comparação paramétrico, como o teste-*t* (*Welch Two Sample t-test*). O teste-*t* apresenta valor-*p* de 0,104. Ou seja, usando 95% de confiança não existe diferença entre as amostras analisadas. Embora estatisticamente similares, podemos ver que a rede de topologia evoluída (T) apresenta valores mínimos menores que a rede do trabalho [Pessin et al. 2011] (C). No histograma podemos ver que apenas a rede de topologia evoluída apresenta valores inferiores a 1.000 (MSE). Embora a rede de topologia evoluída é a que apresenta a maior dispersão nos resultados.

## 6. Evolução de pesos de conexões em redes neurais para localização

Os trabalhos [Pessin et al. 2012] e [Souza et al. 2012] fazem uma extensa busca por parâmetros em métodos de evolução a fim de apresentar resultados de evolução de redes neurais no que se refere ao peso de conexões (respectivamente, fazendo uso de otimização por enxame de partículas e algoritmos genéticos). Diferente do apresentado na seção anterior, a evolução se dá sobre os pesos das conexões e não sobre a estrutura (topologia).

Os dados utilizados na evolução dos pesos em [Souza et al. 2012] são os mesmos dados utilizados no treinamento da rede utilizando *rprop* em [Souza et al. 2013]. Pôde-se perceber, em uma análise superficial, que a rede tendo os pesos evoluídos apresentou resultados piores que a rede treinada com o algoritmo *resilient backpropagation* (*rprop*), tanto em termos de tempo como de acuracidade. Para a rede evoluída em [Pessin et al. 2012] apresentamos nessa seção a comparação com o treino usando o algoritmo *rprop*. A Figura 9 apresenta o resultado de 30 execuções da rede neural treinada neste trabalho ( $E_1$ ) e a rede com pesos evoluídos, de [Pessin et al. 2012] ( $E_2$ ). O teste de comparação (*Welch Two Sample t-test*) apresenta valor-*p* inferior a 0,000, ou seja, usando 95% de confiança, os conjuntos não são similares. Os valores-*p* do teste de adequação (*Shapiro-Wilk normality test*) para  $E_2$  e  $E_1$  são respectivamente 0,925 e 0,915. Além de apresentar resultados melhores, o uso do *rprop* é consideravelmente mais rápido. Foram necessárias 10.000 gerações e 1.000 indivíduos para obter os resultados de  $E_2$  (i.e.



**Figura 9. Resultados para a rede com evolução de pesos ( $E_2$ ) e para a rede treinada ( $E_1$ ). A rede treinada apresenta resultados  $\approx 20\%$  melhores do que a rede com evolução de pesos.**

10.000.000 de avaliações), enquanto que para o *rprop* foi utilizado 100.000 ciclos. Assim, podemos concluir com este resultado que é melhor usar um bom algoritmo de aprendizado, como o *rprop* ao contrário de um método evolutivo para a correção dos pesos das conexões.

## 7. Conclusões e trabalhos futuros

Apresentamos uma metodologia para obtenção de localização por meio de uso de redes neurais onde as entradas são a potência de pontos de acesso. Avaliamos um processo evolucionário na busca de (i) topologias e (ii) pesos de conexões em redes neurais. Foram obtidas topologias de redes neurais estatisticamente similares com as redes obtidas por processo manual. Pudemos ver que o uso de apenas uma execução de cada rede apresentou resultados sensivelmente piores que utilizando média de 5 ou 10 execuções. Estatisticamente, o conjunto resultante de uma única execução não é aceito como similar às execuções utilizando média de 5 ou 10. Execuções considerando médias de 5 ou 10 execuções apresentaram resultados mais homogêneos. Ainda, pudemos ver que é estatisticamente semelhante, neste caso, a execução do treino da rede por 20.000 ciclos ou 100.000 ciclos. Esta descoberta é bastante importante visto que o treino de uma rede neural por um número menor de ciclos permite uso mais rápido do sistema e, por ser uma tarefa que usa bastante poder computacional, recursos como bateria, memória e CPU são poupados. Em relação a evolução de pesos, o uso do método evolucionário não se mostrou melhor que o *rprop*; além do *rprop* ser consideravelmente mais rápido ele também permitiu obter resultados melhores na execução da RNA. Assim, conclui-se que (i) o método evolutivo é uma boa abordagem para a descoberta de topologias e que (ii) quando possível, o algoritmo de aprendizado *rprop* deve ser priorizado em relação ao método evolutivo para evolução dos pesos.

Trabalhos descrevendo uma avaliação de filtros e de diferentes quantidades de pontos de acesso podem ser vistos em [Pessin 2013]. Como trabalhos futuros, é esperado o desenvolvimento de propostas para (i) tolerância a falhas e para (ii) uma maior autonomia no uso do sistema de localização proposto.

## Agradecimentos

Agradeço ao professor T. Braun e aos colegas J. R. Souza, F. G. Costa e D. F. Sciotti pelo auxílio e por diversas inspirações neste trabalho. Agradeço ao apoio financeiro do CNPq e FAPESP por meio do INCT-SEC (National Institute of Science and Technology – Critical Embedded Systems – Brazil), processos 573963/2008-8 e 08/57870-9. Ainda, agradeço a CAPES pelo apoio financeiro por meio de bolsa de doutorado nacional e sanduíche (processo BEX 4202-11-2).

## Referências

- Dargie, W. and Poellabauer, C. (2010). *Fundamentals of Wireless Sensor Networks: Theory and Practice*. Wiley.
- Elnahrawy, E., Li, X., and Martin, R. (2004). The limits of localization using signal strength: a comparative study. In *Sensor and Ad Hoc Communications and Networks, First Annual IEEE Communications Society Conference on*, pages 406–414.
- Franceschini, F., Galetto, M., Maisano, D., and Mastrogiacomo, L. (2009). Mobile spatial coordinate measuring system (mscms) – introduction to the system. *The International Journal of Advanced Manufacturing Technology*, 47:3867 –3889.
- Gast, M. S. (2002). *802.11 Wireless Networks: The Definitive Guide*. O’Reilly.
- Goldberg, D. E. (1989). *Genetic Algorithms in Search, Optimization, and Machine Learning*. Addison-Wesley Professional.
- Gustafsson, F. and Gunnarsson, F. (2003). Positioning using time-difference of arrival measurements. In *Acoustics, Speech, and Signal Processing, 2003. Proceedings. (ICASSP ’03). 2003 IEEE International Conference on*, volume 6.
- Gwon, Y., Jain, R., and Kawahara, T. (2004). Robust indoor location estimation of stationary and mobile users. In *INFOCOM 2004. Twenty-third Annual Joint Conference of the IEEE Computer and Communications Societies*, volume 2, pages 1032–1043.
- Hara, S. and Anzai, D. (2008). Experimental performance comparison of rssi- and tdoa-based location estimation methods. In *Vehicular Technology Conference, 2008. VTC Spring 2008. IEEE*, pages 2651 –2655.
- Hata, A. Y., Wolf, D. F., Pessin, G., and Osório, F. S. (2009). Terrain mapping and classification using neural networks. In *International Conference on Hybrid Information Technology (ICHIT)*, volume 321, pages 438–442. New York: ACM.
- Holland, J. H. (1975). *Adaptation in Natural and Artificial Systems*. Univ. of Michigan.
- Hornik, K., Stinchcombe, M., and White, H. (1989). Multilayer feedforward networks are universal approximators. *Neural Networks*, 2:359–366.
- Huang, G.-B. (2003). Learning capability and storage capacity of two-hidden-layer feedforward networks. *Neural Networks, IEEE Transactions on*, 14(2):274 – 281.
- Jun, H. and Qingbao, Z. (2010). Multi-objective mobile robot path planning based on improved genetic algorithm. In *Int. Conf. on Int. Comp. Technology and Automation*.
- Kartalopoulos, S. V. (1995). *Understanding Neural Networks and Fuzzy Logic: Basic Concepts and Applications*. Wiley.
- Ladd, A., Bekris, K., Rudys, A., Wallach, D., and Kavraki, L. (2004). On the feasibility of using wireless ethernet for indoor localization. *Robotics and Automation, IEEE Trans. on*, 20(3):555 – 559.
- Maisano, D. A., Jamshidi, J., Franceschini, F., Maropoulos, P., Mastrogiacomo, L., Mileham, A., and Owen, G. (2008). Indoor gps: System functionality and initial performance evaluation. *International Journal of Manufacturing Research*, 3(3):335–349.

- Marinello, S. H. M., Ueyama, J., Pessin, G., Osório, F. S., and Vargas, P. A. (2012). Determinando posições relativas de nós móveis por meio de bluetooth e aprendizado de máquina. *Revista de Exatas e Tecnológicas (RETEC)*, 3:1–10.
- Martinelli, A. (2002). The odometry error of a mobile robot with a synchronous drive system. *Robotics and Automation, IEEE Trans. on*, 18(3):399–405.
- Mitchell, M. (1998). *An Introduction to Genetic Algorithms*. The MIT Press.
- Napier, A., Sibley, G., and Newman, P. (2010). Real-time bounded-error pose estimation for road vehicles using vision. In *IEEE Conf. on Intelligent Transp. Systems*.
- Nolfi, S. and Floreano, D. (2000). *Evolutionary Robotics: The Biology, Intelligence, and Technology of Self-Organizing Machines*. The MIT Press.
- Orlando, M., Akolkar, H., Dutta, A., Saxena, A., and Behera, L. (2010). Optimal design and control of a hand exoskeleton. In *IEEE Conf. on Robotics Aut. and Mechs*.
- Perkins, C., Lei, L., Kuhlman, M., Lee, T., Gateau, G., Bergbreiter, S., and Abshire, P. (2011). Distance sensing for mini-robots: Rssi vs. tdoa. In *Circuits and Systems (ISCAS), 2011 IEEE International Symposium on*, pages 1984 –1987.
- Pessin, G. (2013). *Estratégias inteligentes aplicadas em robôs móveis autônomos e em coordenação de grupos de robôs*. PhD thesis, Universidade de São Paulo.
- Pessin, G., Osório, F. S., Souza, J. R., Costa, F. G., Ueyama, J., Wolf, D. F., Braun, T., and Vargas, P. A. (2012). Evolving an indoor robotic localization system based on wireless networks. In *13th International Conference on Engineering Applications of Neural Networks (EANN)*, volume 311, pages 61–70.
- Pessin, G., Osório, F. S., Ueyama, J., Souza, J. R., Wolf, D. F., Braun, T., and Vargas, P. A. (2011). Evaluating the impact of the number of access points in mobile robots localization using artificial neural networks. In *The Fifth International Conference on COMmunication System softWARE and MiddlewaRE (COMSWARE)*, pages 10:1–10:9.
- Siegwart, R. and Nourbakhsh, I. R. (2004). *Introduction to Autonomous Mobile Robots*. The MIT Press.
- Souza, J. R., Pessin, G., Osório, F. S., Wolf, D. F., and Vargas, P. A. (2012). Combining evolution and training in a robotic controller for autonomous vehicle navigation. In *Towards Autonomous Robotic Systems (TAROS), Advances in Autonomous Robotics, Lecture Notes in Computer Science (LNCS)*, volume 7429, pages 426–427.
- Souza, J. R., Pessin, G., Shinzato, P. Y., Osório, F. S., and Wolf, D. F. (2013). Vision-based waypoint following using templates and artificial neural networks. *Neurocomputing*, 107:77–86.
- Teoh, E., Tan, K., and Xiang, C. (2006). Estimating the number of hidden neurons in a feedforward network using the singular value decomposition. *Neural Networks, IEEE Transactions on*, 17(6):1623 –1629.
- Welch, G., Bishop, G., Vicci, L., Brumback, S., Keller, K., and Colucci, D. (2001). High-performance wide-area optical tracking: The hiball tracking system. *Presence: Teleoper. Virtual Environ.*, 10(1):1–21.