

Um Agente Racional para a Localização de Usuários em Ambientes de Redes Sem Fio

Filipe M. Roberto, Jeandro de M. Bezerra, Gustavo A. L. de Campos, Marcial P. Fernandez, Joaquim Celestino Jr, Ana Luiza B. de Paula.

Laboratório de Redes de Comunicação e Segurança da Informação (LARCES) –
Universidade Estadual do Ceará (UECE)
Av. Paranjana 1700 – Itaperi - 60740-903 – Fortaleza – CE

{filipe, jeandro, gustavo, marcial, celestino, analuiza}@larces.uece.br

Abstract. *With the increase of context-aware applications, becomes advisable to have the location of mobile device to adapt according the environment. This paper purpose an approach based on concepts of Artificial Intelligence for the problem of location of mobile devices in indoor environments using IEEE 802.11 technology. This approach is based on the notion of rational agents and artificial neural networks to implement the location procedure. Two architectures of agents were explored, a multilayered neural network with back-propagation algorithm and a specialized version of this algorithm for definition of location. A prototype was implemented and the results demonstrate that the approach is efficient for the definition locations in dynamic environments.*

Resumo. *Com o crescimento das aplicações baseados no contexto, torna-se aconselhável a localização do dispositivo para adaptação da aplicação ao ambiente. Este artigo consiste em uma abordagem fundamentada em conceitos de Inteligência Computacional para o problema de localização de dispositivos móveis em ambientes indoor utilizando a tecnologia IEEE 802.11. Esta abordagem fundamenta-se na noção de agentes racionais e de redes neurais artificiais para a concretização desta noção. Neste contexto, foram exploradas duas arquiteturas de agentes, a rede neural multicamadas com algoritmo back-propagation e uma versão especializada deste algoritmo para o cálculo de localizações. Foi desenvolvido um protótipo e os resultados obtidos demonstram que a abordagem é eficiente para o cálculo de localizações em ambientes dinâmicos.*

1. Introdução

Recentemente, tem-se observado a necessidade de aplicações para redes sem fios que funcionem conforme o ambiente em que os dispositivos móveis estão localizados. Para um funcionamento adequado, este tipo de aplicação, chamada de aplicação baseada no contexto, requisita a localização física do dispositivo. Apesar dos avanços na tecnologia GPS, uma visão geral sobre o GPS pode ser lida em Dana (2000), e sua ampla utilização, esta tecnologia não funciona, ou tem o comportamento degradado, quando é utilizada em ambientes fechados. Surge a partir daí a necessidade de projetos de sistemas de localização, denominados LPS (*Local Position System*).

Vale ressaltar, o padrão IEEE 802.11 das WLAN's nesses ambientes fechados, utilizam sinais de radio frequência. Esses sinais possuem a virtude de não exigir que haja uma linha de visada entre um emissor e um receptor, porém o sinal sofre interferência devido à reflexão, refração e difração. Durante a propagação do sinal no ambiente, estas propriedades físicas atuam de forma dinâmica sobre o sinal. Com isto, variam também os parâmetros de caracterização do ambiente, por exemplo: potência, relação sinal/ruído e qualidade do canal. Estas características dificultam o projeto eficaz de sistemas LPS. Para o estudo das características de uma rede sem fio é indicada a leitura de Rappaport (2002).

Este artigo propõe uma abordagem fundamentada em noções de Inteligência Computacional para o problema de localização de dispositivos moveis em redes sem fio que operam em ambientes fechados (*indoor*). A finalidade da abordagem consiste em estabelecer uma arquitetura fundamentada em agentes racionais, que estruture um procedimento de localização robusto e eficaz, independente de hardware, baseando-se somente na implementação em nível de software e uso da infra-estrutura tradicional de uma WLAN.

Como objetivo de mais longo prazo, a abordagem visa dar origem a um *framework* para a implementação de agentes racionais localizadores. Para entender melhor a idéia, vale imaginar um mundo formado por duas entidades que interagem entre si: um ambiente, em analogia ao ambiente físico de alcance da rede, e um programa agente, em analogia ao agente humano, que calcula aproximadamente a posição de objetos em um espaço determinado. Neste mundo, o programa agente recebe as percepções do ambiente que são relevantes para a localização de um dispositivo, seleciona e calcula uma localização, que deverá ser enviada para o ambiente.

Para validar a arquitetura proposta, foi concebido um protótipo de programa agente tomando como base duas idéias principais: potência do sinal, para caracterizar o ambiente, e as redes neurais artificiais, para implementação da arquitetura. Os resultados obtidos com esta proposta demonstram que a mesma é eficiente e que ela deve ser continuada, através de determinados refinamentos e de sua adaptação para ambientes abertos.

Além desta introdução, este artigo está organizado em mais cinco seções principais. A Seção 2 apresenta alguns dos trabalhos recentes envolvendo sistemas localizadores. A Seção 3 formaliza o problema de localização adequadamente e especifica a arquitetura abstrata do agente racional para resolver o problema. A Seção 4 concretiza esta arquitetura abstrata empregando redes neurais artificiais. A Seção 5 avalia o desempenho do agente concretizado. Finalmente, a Seção 6 apresenta algumas considerações sobre a abordagem testada e a direção em que a mesma deverá evoluir.

2. Sistemas para a Localização de Dispositivos Móveis

Nos últimos anos têm sido desenvolvidos vários sistemas de localização utilizando uma ampla variedade de tecnologias: GPS, infravermelho, sistemas de visão computacional e sistemas baseados em rádio frequência [Hightower and Borrielo, 2001]. Uma taxonomia para sistemas de localização feita por Yousief (2004) destaca o modo WLAN infra-estruturado, dando ênfase às tecnologias que utilizam a potência do sinal.

Bahl e Padmanabhan (2000) desenvolveram o RADAR, um trabalho que se tornou referência na pesquisa de localização em ambientes. Essa proposta utiliza a potência do sinal para a localização dos dispositivos móveis e a própria infra-estrutura de uma rede WLAN IEEE 802.11b para localizar dispositivos.

Por sua vez, Ni e Liu (2004) um outro sistema de localização denominado LANDMARC. Este sistema utiliza a tecnologia RFID (Radio Frequency Identification) para auxiliar na localização e para monitorar o ambiente. Esta tecnologia faz uso de hardware adicional.

Outros trabalhos têm como base a análise do comportamento da potência do sinal em um ambiente. Kaemarungsi et. al (2004) desenvolveu um *framework* para avaliação de um sistema de posicionamento simples. O *framework* produz uma análise do desempenho do sistema de localização sob o efeito da alteração do número de AP's e parâmetros de propagação do sinal.

RNAs (Redes Neurais Artificiais) têm sido utilizadas para executar tarefas complexas como classificação, otimização, controle e função de aproximação. A complexidade associada aos problemas de localização de dispositivos móveis em ambiente *indoor*, tem incentivado a utilização desta técnica. Por exemplo, Battiti e Nhat (2002) utilizaram a infra-estrutura da WLAN para realizar experimentos empregando um método fundamentado em RNAs multicamadas.

Nerguizian (2004) utilizou um ambiente de redes neurais artificiais proprietário para realizar diversos experimentos. Ele empregou a técnica de *fingerprinting* para construção de conjuntos de exemplos sobre o problema. Esta técnica envolve, primeiramente, a determinação de pontos de calibração (onde são colhidas medidas de potência do sinal) no ambiente e, posteriormente, a coleta de dados associados aos pontos que são relevantes para a localização.

3. Formalização do Problema de Localização e da Arquitetura do Agente

O problema de localização de usuários abordado neste artigo supõe um ambiente físico de dimensões máximas X_{max} e Y_{max} em \mathbf{R} . Considera-se que, neste ambiente existem r pontos de acesso. Neste contexto, são conhecidas as seguintes informações sobre o ambiente: um conjunto formado por n_c localizações de determinados pontos de calibração espalhados pelo ambiente, $\mathbf{Lc} = \{(x_1, y_1), \dots, (x_{n_c}, y_{n_c})\}$, e uma família de n_c conjuntos formados por r seqüências de n_r medidas de atenuação nas potências dos sinais, emitidos pelos r pontos de acesso e recebidos nos n_c pontos de calibração, $\mathbf{Pc} = \{\{pc_{111} \dots pc_{11(nr)}, \dots, pc_{1r1} \dots pc_{1r(nr)}\}, \dots, \{pc_{(nc)11} \dots pc_{(nc)1(nr)}, \dots, pc_{(nc)r1} \dots pc_{(nc)r(nr)}\}\}$. Assim, dado um novo conjunto de n_r seqüências de medidas de potências, dos sinais emitidos pelos mesmos r pontos e recebidos por um usuário em um ponto de teste \mathbf{p}_T no ambiente, $\mathbf{Pt} = \{p_{T11} \dots p_{T1(nr)}, \dots, p_{Tr1} \dots p_{Tr(nr)}\}$, a tarefa consiste em determinar a localização de \mathbf{p}_T , ou seja, (x_T, y_T) .

3.1. O Agente Padrão

A abordagem computacional proposta para calcular soluções para o problema formalizado acima é uma concretização da noção de agente racional, descrita por Russell e Norvig (1995). Ela tomou como base a arquitetura do agente padrão especializando-a para resolver problemas semelhantes ao descrito acima e um

refinamento desta especialização para ser utilizado nos casos em que o ambiente físico pode ser dividido no formato de salas interligadas. A Figura 1 apresenta a arquitetura do agente padrão [Weiss 1999].

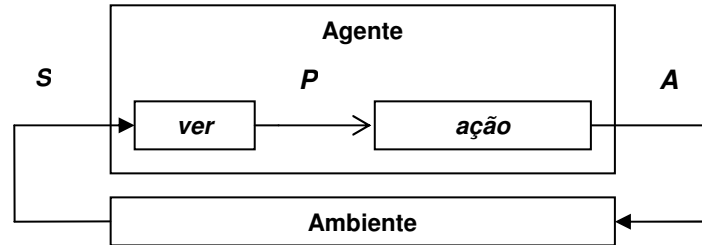


Figura 1. Arquitetura do agente padrão

Esta arquitetura supõe que em qualquer instante:

- (1) por meio de sensores o agente recebe informações do ambiente que são seqüências de estados definidos em um conjunto, $S = \{s_1, \dots, s_n\}$, de n estados possíveis para o ambiente (E);
- (2) um subsistema de percepção, $ver: S \rightarrow P$, processa cada estado de uma seqüência S^* e mapeia em uma de m percepções, $P = \{p_1, \dots, p_n\}$, que são representações de aspectos dos estados de S que estão acessíveis ao agente para a tomada de decisão;
- (3) dependendo do problema, o agente mantém internamente algum conhecimento a respeito do comportamento do ambiente;
- (4) um subsistema de tomada de decisão, $ação: P^* \rightarrow A$, processa as seqüências perceptivas P^* , resultantes de S^* , e seleciona uma de l ações do conjunto de ações possíveis para o agente, $A = \{a_1, \dots, a_l\}$;
- (5) por meio de atuadores o agente envia a ação selecionada para o ambiente;
- (6) a racionalidade da ação selecionada pelo agente é avaliada através de uma medida de desempenho G previamente estabelecida, adequada ao problema que o agente irá resolver.

Quanto à especialização da arquitetura para o problema de localização formalizado e, mais especificamente, aos aspectos necessários à implementação da função *ação* do agente, pode-se dizer mais concretamente que:

- (1-2) apesar dos estados do ambiente conterem diversas informações, inclusive, a respeito de outros objetos anteriormente localizados e da presença de obstáculos ativos e passivos, considera-se que o subsistema de percepção, além de disponibilizar as dimensões máximas do ambiente físico, (X_{max}, Y_{max}) , auxilia no mapeamento de seqüências de estados do ambiente, capturados em um ponto de teste p_T , em um conjunto formado por r seqüências perceptivas de medidas de atenuação das potências dos sinais que têm origem nos r pontos de acesso, $P_t = \{p_{T11} \dots p_{T1(nr)}, \dots, p_{Ts1} \dots p_{Tr(nr)}\}$;
- (3) as informações a respeito do ambiente do problema estão disponíveis no formato de um subconjunto de exemplos definido em $P_c \times L_c$, ou seja, $\Psi = \{(\{pc_{111} \dots pc_{11(nr)}, \dots, pc_{1r1} \dots pc_{1r(nr)}\}, (x_1, y_1)), \dots, (\{pc_{(nc)11} \dots pc_{(nc)1(nr)}, \dots, pc_{(nc)r1} \dots pc_{(nc)r(nr)}\}, (x_{nc},$

$y_{nc})\}$, formado por n_c pares ordenados, cada par formado por r seqüências perceptivas de n_r medidas e uma localidade correspondente;

- (4-5) a função **ação** para o agente deve ser capaz de fazer uso das informações disponíveis em Ψ e em P_t , e calcular a localização (x_T, y_T) de p_T que será enviada para o ambiente; e
- (6) a medida de avaliação de desempenho, ou meta do agente, é a distância euclidiana mínima entre as coordenadas de uma localização calculada, (x_T, y_T) em $[0, X_{max}] \times [0, Y_{max}]$, e as coordenadas de uma localização desejada, (x^*, y^*) em $[0, X_{max}] \times [0, Y_{max}]$, ou seja, $D_{min} = [(x_T - x^*)^2 + (y_T - y^*)^2]^{0.5}$.

É pertinente, no que diz respeito às informações sobre o ambiente, que no conjunto de exemplos algumas seqüências perceptivas são semelhantes, ou seja, de valores de atenuação de potência de sinal semelhantes são medidas em lugares diferentes. Da mesma forma, sabe-se que, dependendo das condições ambientais, seqüências perceptivas diferentes podem ser recebidas em uma mesma localidade ou muito próximo desta localidade. Isto ocorre principalmente devido a problemas de radio propagação em ambientes fechados.

Assim, além de ser um ambiente contínuo, já que o domínio das percepções e das ações são respectivamente subconjuntos dos números reais, do ponto de vista do agente é também um ambiente não-determinístico, já que para uma mesma seqüência perceptiva pode existir mais de uma localização, $amb: P^* \rightarrow 2^A$. Some-se a isto a falta de acessibilidade do ambiente dificulta ainda mais a tarefa de concepção do agente. De maneira a realizar a meta G neste tipo de ambiente, um agente racional deve ser capaz de aprender e utilizar em seu mecanismo de seleção de ações informações a respeito do padrão de comportamento do ambiente.

3.2. Agente Padrão Refinado

Neste contexto, visando acrescentar à arquitetura descrita na Figura 1 um componente a mais para lidar com a complexidade do ambiente, esta primeira etapa da abordagem experimentou um refinamento da função **ação** do agente padrão. A Figura 2 apresenta este primeiro refinamento na arquitetura do agente padrão especializado para o problema de localização.

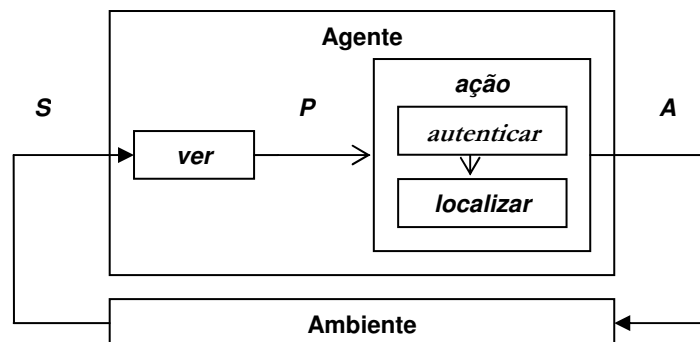


Figura 2. Arquitetura refinada do agente padrão

Vale ressaltar, este refinamento tem como princípio a idéia de dividir para conquistar. De um lado, o ambiente é decomposto em salas interligadas e do outro o

subsistema de tomada de decisão do agente é decomposto em dois outros subsistemas. Assim, esta nova figura especifica duas das concretizações que foram realizadas na especialização do agente padrão para o problema de localização, ou seja:

(2.1) a função **ver**, além de disponibilizar as dimensões máximas do ambiente físico, decompõe este ambiente em um conjunto de n_z salas interligadas, $Salas = \{sala_1, \dots, sala_{n_z}\}$, e disponibiliza para o agente o esquema de interligação entre salas e um conjunto contendo os pares de dimensões máximas destas salas, $Dsalas = \{(X_{max(sala1)}; Y_{max(sala1)}), \dots, (X_{maxsala(nz)}; Y_{maxsala(nz)})\}$; e

(4.1) a função **ação** engloba duas outras funções, ou seja: (a) uma função **autenticar** deve fazer uso das informações contidas em Ψ , $Dsalas$ e em P_t , e do esquema de interligação entre as salas em S , e identificar a $sala_T$ em que p_T está localizado; e (b) fazendo uso também desta nova informação, uma função **localizar** deve calcular a localização (x_T, y_T) de p_T .

A Figura 2 não indica a maneira como os resultados produzidos pelas duas novas funções são combinados para a localização dos pontos de teste. Entretanto, esta nova possibilidade refina o mecanismo de seleção de ações do agente, possibilitando a concepção de lógicas de seleção mais racionais para ambientes complexos como os de redes sem fio. A próxima seção apresenta uma maneira adequada de se concretizar as duas arquiteturas. Conforme mencionado na seção de introdução, este artigo estudou a possibilidade e os problemas que decorrem da utilização de redes de neurônios multicamadas diretamente alimentadas com algoritmo de treinamento *back-propagation* [Haykin 1999].

4. As Redes Neurais Artificiais e o Mecanismo de Localização do Agente

A maneira como as percepções em P , as ações em A e as informações sobre o comportamento do ambiente em Ψ estão disponíveis para o agente no problema formalizado, restringe as possibilidades de concretização da função **ação** nas arquiteturas propostas [Kasabov 1995]. Assim, considerando que a informação está disponível no formato de um conjunto não muito extenso de dados, implementou-se a função **ação** fundamentada na noção de redes neurais artificiais (RNA). Além do mais, considerando que cada exemplo em Ψ é um par ordenado definido em $P \times A$, e que a função **ação** do agente deve aproximar o mapeamento ideal $P^* \rightarrow A$ para um certo ambiente, empregou-se mais especificamente as RNAs multicamadas diretamente alimentadas com algoritmo de aprendizado supervisionado *back-propagation* [Haykin 1999].

As duas arquiteturas foram concretizadas através de uma mesma estrutura de RNA multicamada, ou seja, formada por uma camada de entrada composta por $r+1$ neurônios, uma camada oculta de neurônios, e uma camada de saída composta por 2 neurônios. A Figura 3 apresenta esta estrutura, supostamente treinada, em etapa de operação, podendo funcionar como **ação** na Figura 1, ou como **autenticação** e **localização** na Figura 2. Nos casos em que a Figura 3 representar a função **ação** ou **localização**, a saída da rede é um par ordenado (x_T, y_T) em $[0, X_{max}] \times [0, Y_{max}]$. Nos casos em que representar a função **autenticação**, a saída da rede é uma indicação de autenticação, ou não, da sala identificada pela rede. Neste trabalho, considerou-se que,

uma sala deve ser autenticada para localização se a saída de sua RNA se aproximar de $(1,0)$; e que, a sala não deve ser autenticada se a saída se aproximar de $(0, 1)$.

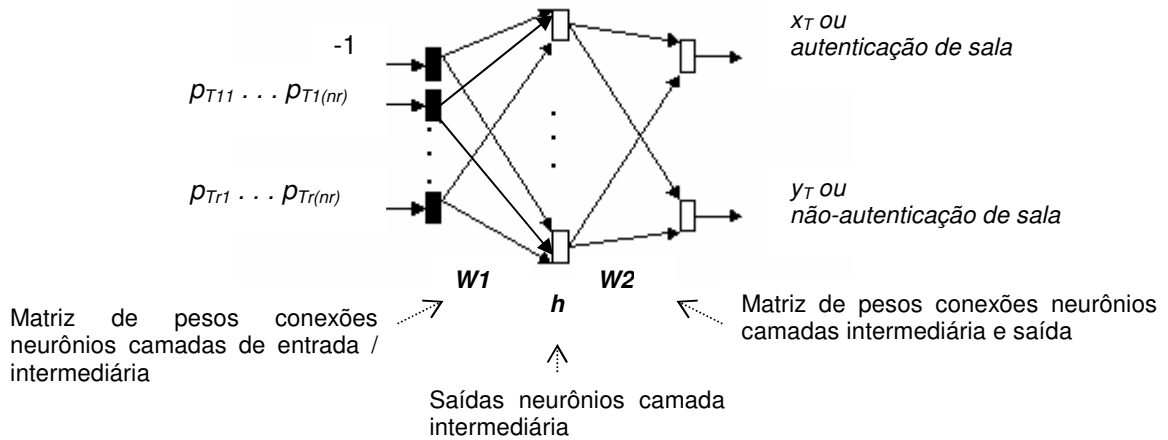


Figura 3. Modelo de uma RNA multicamadas diretamente alimentada

Todos os neurônios do modelo bloco construtor descrito acima foram implementados com função de ativação sigmóide, ou seja, $g(v) = 1/(1+e^{-v})$ onde v é o resultado da integração espacial dos estímulos nas conexões de um neurônio [Haykin 1999]. Considerando esta função e a medida de avaliação G do agente, todas as RNAs foram treinadas buscando parâmetros $W1$ e $W2$ ótimos, ou seja, que minimizem o erro médio quadrático entre localizações calculadas, (x_T, y_T) e localizações desejadas, (x^*, y^*) , e, conseqüentemente, minimizem a distância euclidiana média entre estas localizações.

Nas duas arquiteturas foram testados o algoritmo de aprendizado *back-propagation* convencional e uma alteração deste algoritmo, *back-propagation-alterado*, desenvolvida com o objetivo de verificar a influência das dimensões máximas do ambiente de desempenho do agente na eficiência da aproximação do mapeamento ideal. Assim, a função **ação** da primeira arquitetura foi concretizada de duas maneiras diferentes, ou seja, através de uma RNA *back-propagation* ou de uma RNA *back-propagation-alterado*. Quanto às funções da segunda arquitetura, a função **autenticar** foi implementada por n_z RNAs *back-propagation*, ou seja, uma rede de autenticação para cada uma das salas no conjunto *Salas*, e a função **localizar** foi implementada por n_z RNAs *back-propagation-alterado*, ou seja, uma rede de localização para cada sala de dimensões máximas definidas em *Dsalas*.

Mais especificamente, nas redes de concretização da função **ação** da Figura 1 com *back-propagation* convencional, o treinamento ocorreu buscando minimizar o erro na saída das redes, ou seja:

$$E(W) = 0.5 * [(g(\sum_i w2_{i1} * h_i) - x^*/X_{max})^2 + ((g(\sum_i w2_{i2} * h_i) - y^*/Y_{max})^2)] \quad (1)$$

onde i identifica os neurônios na única camada intermediária do modelo de RNA bloco construtor. Nestes casos, nos momentos de operação de uma rede as saídas calculadas foram multiplicadas pelas dimensões máximas do ambiente, ou seja,

$$x_T = g(\sum_i w2_{i1} * h_i) * X_{max} \quad (2)$$

e

$$y_T = g\left(\sum_i w_{i2} * h_i\right) * Y_{max}. \quad (3)$$

Por outro lado, nas redes de concretização das funções nas Figuras 1 e 2 com *back-propagation-alterado*, em vez de calcular o erro levando em consideração as saídas calculadas pelos neurônios de saída, o treinamento considerou o erro real no ambiente físico total, para o caso da implementação da função **ação** na Figura 1, e o erro real em cada uma das salas que compõem o ambiente, para o caso da função **localização** na Figura 2. De uma maneira geral, pode-se dizer que nestes casos o treinamento buscou minimizar:

$$E^{(z)}(W) = 0.5 * [g\left(\sum_i w_{i1} * h_i\right) * X_{max}^{(z)} - x^*]^2 + ((g\left(\sum_i w_{i2} * h_i\right) * Y_{max}^{(z)} - y^*)^2] \quad (4)$$

tal que quando $z = 0$ então (X_{max}, Y_{max}) e $E^{(0)}$ identificam respectivamente as dimensões máximas do ambiente e o erro meta para o treinamento de redes para concretização da função **ação** do agente padrão; e quando $z = 1, \dots, n_z$, então $(X_{max}^{(z)}, Y_{max}^{(z)})$ e $E^{(z)}$ identificam respectivamente as dimensões máximas de uma sala, $sala_z \in Salas$, e o erro meta para treinamento de redes para a concretização da função **localização** do agente padrão refinado.

Assim, nestes casos, no momento da operação de uma rede z , as saídas foram calculadas conforme as Expressões (2) e (3), substituindo-se (X_{max}, Y_{max}) por $(X_{max}^{(z)}, Y_{max}^{(z)})$ nos casos em que $z > 0$. Esta nova medida de erro descrita na Expressão (4) impôs uma alteração em alguns termos presentes na regra delta presente no algoritmo *back-propagation* convencional [Haykin 1999]. Esta alteração pode ser percebida deduzindo-se a fórmula para o cálculo, na iteração $k+1$ do aprendizado de uma sala, do peso na conexão entre um neurônio i da camada intermediária e um neurônio localizador j da camada de saída, ou seja, deduzindo-se:

$$w_{ij}^{(k+1)} = w_{ij}^{(k)} - \eta * \partial E^{(z)}(W2^{(k)}) / \partial w_{ij}^{(k)} \quad (5)$$

onde:

$$\partial E^{(z)}(W2^{(k)}) / \partial w_{ij}^{(k)} = -h_i * \delta 2_j^{(z)};$$

para $j = 1, 2$. Assim, nesta dedução, representando as saídas dos neurônios da camada de saída por $x' = g\left(\sum_i w_{i1} * h_i\right)$ e $y' = g\left(\sum_i w_{i2} * h_i\right)$, então

$$\delta 2_1^{(z)} = X_{max}^{(z)} * (x^* - x' * X_{max}^{(z)}) * x' * (1 - x') \quad (6)$$

e

$$\delta 2_2^{(z)} = Y_{max}^{(z)} * (y^* - y' * Y_{max}^{(z)}) * y' * (1 - y'). \quad (7)$$

Comparando-se as Expressões (6) e (7) com as obtidas para o caso do *back-propagation* convencional [Haykin 1999], é possível determinar a seguinte relação entre os componentes de aprendizagem associados aos dois neurônios localizadores componentes da camada de saída:

$$\delta 2_{alterado1}^{(z)} = (X_{max}^{(z)})^2 * \delta 2_{convencional1}^{(z)} \quad (8)$$

e

$$\delta\mathcal{L}_{alterado2}^{(z)} = (Y_{max}^{(z)})^2 * \delta\mathcal{L}_{convencional2}^{(z)}. \quad (9)$$

Assim, de acordo com o princípio da retropropagação, este tipo de alteração propaga-se para a atualização dos pesos nas conexões entre os neurônios da camada de entrada e intermediária. Conforme pode ser percebido nos resultados dos testes realizados com os dois algoritmos concretizando as duas arquiteturas, o *back-propagation-alterado* refinou levemente, na maioria dos casos, o cálculo das localizações realizadas pelo *back-propagation* convencional. A próxima seção apresenta os experimentos que foram realizados e seus principais resultados.

5. Avaliação do Protótipo

O conjunto de exemplos empregado para o treinamento e teste das redes neurais das arquiteturas concretas, foi coletado em um ambiente físico conforme representado na Figura 4. Além dos três pontos de acesso, $r = 3$, este ambiente está dividido em três salas separadas por divisórias de madeira. Os exemplos foram coletados em sessenta e oito localizações diferentes, ou seja, vinte e três na sala da esquerda, $sala_1$, vinte e três na sala central, $sala_2$, e vinte e dois na sala da direita, $sala_3$. Nos experimentos realizados, sessenta localizações foram consideradas pontos de calibração e o restante foi utilizado como pontos de teste, $n_t = 8$.

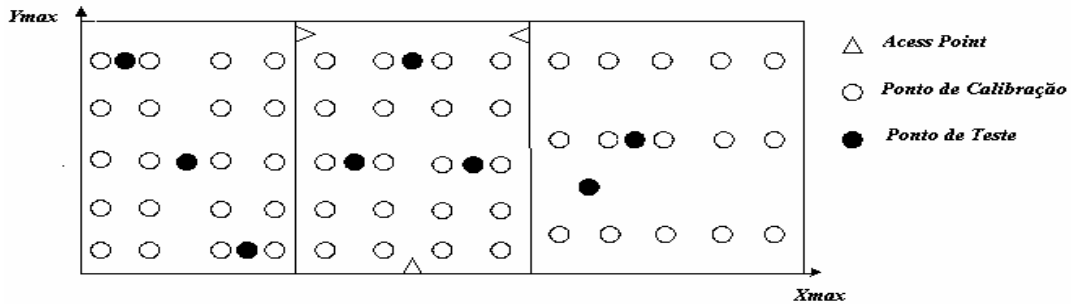


Figura 4. Ambiente físico de teste

Em cada uma das localizações, foram medidas três seqüências de vinte medidas de atenuação nas potências dos sinais emitidos pelos três pontos de acesso, $n_3 = 20$. Por exemplo, a Tabela 1 apresenta três seqüências de medidas obtidas no ponto de calibração 8, situado nas coordenadas (4.6, 2.6) do ambiente de teste.

Tabela 1. Três seqüências de vinte medidas obtidas no ponto de calibração 8

$pc_{811} \dots pc_{81(20)}$	-39 -39 -39 -40 -40 -41 -40 -40 -39 -39 -39 -40 -39 -39 -41 -40 -41 -40 -40 -41
$pc_{821} \dots pc_{82(20)}$	-8 -11 -10 -11 -11 -11 -16 -20 -20 -20 -21 -21 -21 -20 -21 -20 -20 -19 -20 -19
$pc_{831} \dots pc_{83(20)}$	-42 -43 -42 -43 -44 -44 -43 -42 -43 -43 -44 -43 -44 -45 -45 -44 -44 -43 -45 -44

O desempenho dos agentes foi avaliado considerando três conjuntos de exemplos formados entre os sessenta pontos de calibração:

Ψ	todos os sessenta exemplos coletados nas três salas, $n_c = 60$;
Ψ_1	trinta exemplos, $n_c = 30$, ou seja, dez coletados na $sala_1$, dez na $sala_2$ e dez na $sala_3$;
Ψ_2	trinta exemplos restantes de Ψ que não pertencem a Ψ_1 , $n_c = 30$, ou seja, dez coletados na $sala_1$, dez na $sala_2$ e dez na $sala_3$.

Apesar do desempenho do agente proposto depender fortemente do número e da qualidade dos exemplos disponíveis, nesta primeira etapa da abordagem não foi adotado um critério rigoroso para a seleção dos exemplos componentes de cada um dos conjuntos acima. Apenas, os três primeiros experimentos com o agente padrão especializado, descrito na Seção 3, e o *back-propagation* convencional objetivaram escolher entre Ψ , $\Psi1$ e $\Psi2$, o melhor conjunto para ser utilizado nos dois últimos experimentos, ou seja, um empregando a arquitetura do agente padrão e o *back-propagation-alterado*, e outro empregando a arquitetura do agente padrão refinada e este mesmo algoritmo. Conforme pode ser observado, os três conjuntos foram suficientes para a geração das primeiras conclusões e comparação entre algoritmos e arquiteturas.

Assim, considerando Ψ , $\Psi1$ e $\Psi2$, o *back-propagation* convencional e *back-propagation-alterado*, foram realizados cinco experimentos para avaliação das propostas. Todas as redes neurais treinadas foram constituídas de uma única camada intermediária. Depois de algumas verificações, decidiu-se implementar esta camada em alguns casos com três neurônios e em outros casos com cinco neurônios. A taxa de aprendizado e o número máximo de iterações para as redes variaram de acordo com o experimento. A avaliação das redes neurais levou em consideração: o erro médio quadrático na camada de saída das redes devido a todos os pontos de calibração treinados, E_{Treino} , o maior erro associado a algum dos pontos de calibração treinados, E_{Max} , o erro médio quadrático e a distância euclidiana média devida aos 8 pontos testados, E_{Teste} e D_{Teste} . As próximas subseções destacam alguns dos resultados obtidos nos cinco experimentos.

5.1. Resultados do Agente Padrão

Os três primeiros experimentos foram realizados com a arquitetura do agente padrão descrita na Figura 1, com o *back-propagation* convencional e os três conjuntos de exemplos. As redes neurais treinadas continham cinco neurônios na camada intermediária. A taxa de aprendizado foi igual a 0.01 e o número máximo de iterações foi igual 10^5 . Conforme mencionado, estes experimentos tiveram o propósito de observar os conjuntos de exemplos propostos e selecionar aquele conjunto com o menor E_{Teste} , para ser utilizado nos próximos dois experimentos. A Tabela 2 sintetiza E_{Treino} , E_{Max} e E_{Teste} depois que a rede foi treinada com os três conjuntos de exemplos.

Tabela 2. Resultados do treinamento com $\Psi1$, $\Psi2$ e Ψ

	Conjuntos de Exemplos		
	$\Psi1$	$\Psi2$	Ψ
E_{Treino}	5.58	2.42	6.26
(E_{Max}, ponto)	(82.71, 1)	(20.77, 2)	(71.76, 1)
E_{Teste}	11.16	21.87	7.09

Conforme pode ser percebido na Tabela, o conjunto de exemplos total, Ψ , apresentou o menor erro médio quadrático, E_{Teste} , devido aos oito pontos de teste. Entre todos os pontos de calibração treinados, o ponto de calibração 1, situado na coordenada $(1, 1)$, apresentou o maior erro durante o processo de treinamento da rede. A Tabela 3 apresenta as localizações calculadas pelas redes treinadas com os três conjuntos de exemplos.

Tabela 3. Localizações calculadas por RNAs treinadas com $\Psi 1$, $\Psi 2$ e Ψ

		Conjuntos de Exemplos						
		$\Psi 1$		$\Psi 2$		Ψ		
\bar{x}	\bar{y}	x_T	y_T	x_T	y_T	x_T	y_T	D_T
4.2	1	4.93	2.72	11.99	2.83	6.19	2.70	2.62
2.8	2.6	10.26	3.01	6.99	2.32	8.95	3.20	6.18
1.4	5	7.71	4.91	1.98	3.79	3.65	3.90	2.51
9.4	2.6	5.53	2.07	10.93	2.69	5.78	2.48	3.62
8.5	4.2	4.36	3.51	16.09	5.52	8.42	4.56	0.37
6.9	2.6	5.92	3.68	16.07	5.29	7.11	4.00	1.42
13.02	2.10	12.13	1.66	5.47	2.09	10.61	2.53	2.45
13.77	3.15	7.37	1.58	6.39	1.74	7.45	2.16	6.38

Na tabela 3 as células das colunas referentes aos conjuntos de exemplos Ψ , $\Psi 1$ e $\Psi 2$ representam as localizações calculadas pelas redes e as duas colunas mais à esquerda representam as localizações desejadas. As células tracejadas, cujos valores estão em negrito, nas colunas referentes aos conjuntos representam os melhores resultados obtidos nos três experimentos. Conforme E_{teste} na Tabela 2 e conforme pode ser observado na tabela acima, no geral, os melhores resultados foram produzidos pelo conjunto Ψ . A coluna mais à direita da tabela destaca as distâncias associadas aos valores calculados pela RNA treinada com este conjunto.

A princípio os parâmetros $W1$ e $W2$ obtidos com o treinamento da rede não providenciaram a generalização esperada. Apesar do resultado com *back-propagation* convencional ainda poder ser melhorado, por exemplo, através de um refinamento do processo de seleção de exemplos para treinamento, decidiu-se comparar este melhor resultado com o resultado obtido por uma rede de mesma estrutura, mas com *back-propagation-alterado*, e deixar para o futuro a elaboração de um esquema de mineração de exemplos. A Tabela 4 apresenta os novos resultados obtidos com este novo algoritmo e o conjunto de exemplos Ψ .

Tabela 4. Resultados obtidos após o treinamento da rede com Ψ

	<i>Back-convencional</i>	<i>Back-alterado</i>
E_{Treino}	6.26	7.38
(E_{Max}, ponto)	(71.76, 1)	(100.17, 1)
D_{Teste}	3.19	2.62

Conforme pode ser percebido, apesar do erro médio obtido com os exemplos de treinamento, E_{Treino} , e do maior erro associado ao ponto de calibração 1 serem menores para o algoritmo convencional, E_{Max} , o *back-propagation-alterado* reduziu a distância euclidiana média relacionada aos oito pontos de teste. A Tabela 5, a seguir, destaca as distâncias.

Tabela 5. Localizações calculadas pela rede treinada com Ψ

		Conjunto de Exemplos Ψ		
x	y	x_T	y_T	D_T
4.2	1	11.33	1.97	7.20
2.8	2.6	6.25	3.62	3.60
1.4	5	3.46	3.27	2.69
9.4	2.6	10.79	2.03	1.50
8.5	4.2	8.95	4.21	0.45
6.9	2.6	6.16	3.75	1.36
13.02	2.10	11.80	2.59	1.32
13.77	3.15	11.15	1.96	2.88

As células tracejadas, cujos valores estão em negrito, na coluna referente à distância euclidiana, demonstram que o *back-propagation-alterado* melhorou o cálculo das localizações de teste descritas na Tabela 3. Apesar da distância euclidiana média ser somente um pouco menor neste último caso, ela poderá diminuir ainda mais com a mineração dos exemplos para treinamento. Nesta etapa, conforme aconteceu no experimento anterior, decidiu-se comparar este último resultado com aquele produzido pelo agente padrão com arquitetura refinada, ou seja, implementado conforme a segunda arquitetura abstrata e com *back-propagation alterado*. A próxima subseção apresenta os dados coletados desta nova concretização.

5.2. Resultados do Agente Padrão Refinado

Conforme a arquitetura refinada do agente padrão, neste último experimento o conjunto de exemplos Ψ foi dividido em três conjuntos, cada conjunto formado por vinte pontos de calibração coletados em uma das três salas que compõem o ambiente. Assim, cada conjunto associado a uma sala foi utilizado para treinar duas redes neurais, ou seja, uma rede para autenticar a sala do ambiente em que o ponto de teste está localizado e outra para localizar o ponto sobre consideração na sala que foi autenticada. Todas as seis redes neurais treinadas continham três neurônios na camada intermediária. A taxa de aprendizado foi igual a 0.001 e o número máximo de iterações foi igual 10^5 . A Tabela 6 sintetiza o erro médio quadrático, E_{Treino} , obtido ao final do treinamento das redes.

Tabela 6. Erro médio quadrático ao final do treinamento das RNAs

	sala₁		sala₂		sala₃	
	RNA_{Aut}	RNA_{Loc}	RNA_{Aut}	RNA_{Loc}	RNA_{Aut}	RNA_{Loc}
E_{Treino}	0.10	1.84	0.13	1.52	0.08	1.83

Conforme mencionado anteriormente, o treinamento das três redes de autenticação foi interrompido depois do número máximo de iterações. A princípio, percebeu-se que as medidas de erros obtidas ao final do treinamento eram suficientes para a função de autenticação das salas no ambiente. Percebeu-se também que, acima do número máximo de iterações pré-estabelecido, os erros diminuem consideravelmente, entretanto as redes memorizam as informações sobre os pontos de calibração treinados, dificultando a generalização dos resultados para pontos de teste que não fazem parte dos conjuntos de treinamento. A Tabela 7 apresenta os resultados de autenticação quando foram apresentados às redes os oito pontos de teste.

Tabela 7. Resultados produzidos pelas RNAs de autenticação

	<i>sala₁</i>		<i>sala₂</i>		<i>sala₃</i>		<i>sala correta</i>
	<i>Aut</i>	<i>N-Aut</i>	<i>Aut</i>	<i>N-Aut</i>	<i>Aut</i>	<i>N-Aut</i>	
<i>p_{T1}</i>	0.18	0.82	0.78	0.22	0.00	1.00	<i>sala₁</i>
<i>p_{T2}</i>	0.12	0.88	0.08	0.92	0.24	0.76	<i>sala₁</i>
<i>p_{T3}</i>	0.64	0.36	0.15	0.85	0.00	1.00	<i>sala₁</i>
<i>p_{T4}</i>	0.65	0.35	0.34	0.66	0.00	1.00	<i>sala₂</i>
<i>p_{T5}</i>	0.06	0.94	0.79	0.21	0.04	0.96	<i>sala₂</i>
<i>p_{T6}</i>	0.13	0.87	0.57	0.43	0.00	1.00	<i>sala₂</i>
<i>p_{T7}</i>	0.09	0.90	0.07	0.93	0.65	0.35	<i>sala_{3s}</i>
<i>p_{T8}</i>	0.05	0.95	0.89	0.10	0.00	1.00	<i>sala₃</i>

As colunas das células que estão tracejadas, cujos valores estão em negrito, correspondem às salas que foram autenticadas para cada um dos pontos testados. Se na RNA de uma determinada sala, a saída *Aut* for maior que 0.5 e a saída *N-Aut* for menor que 0.5, então a sala deve ser autenticada. Entretanto, se a saída *N-aut* de uma RNA for maior que 0.5 então a sala correspondente não deve ser autenticada. Neste contexto, conforme pode ser percebido na tabela acima, para os oito pontos de teste apresentados cinco foram autenticados corretamente. Estes resultados parciais eram esperados visto que o conjunto de exemplos total disponível para treinamento das redes apresentou problemas em todos os experimentos que foram realizados. Mesmo assim, o experimento foi continuado e, desta vez, as RNAs de localização das salas foram testadas. A Tabela 8 compara os resultados com aqueles que foram obtidos no último experimento com o agente padrão.

Tabela 8. Comparação entre os resultados das duas arquiteturas

	<i>Agente Padrão</i>	<i>Agente Refinado</i>
<i>E_{Treino}/D_{Treino}</i>	7.38 / 2.85	1.74 / 1.59
<i>(E_{Max}, ponto)</i>	(100.17, 1)	(10.14, 55)
<i>E_{Teste}/D_{Teste}</i>	5.39 / 2.62	3.18 / 2.30

A tabela demonstra que o refinamento da arquitetura melhorou a eficiência no cálculo das localizações. Este refinamento produziu os melhores resultados considerando todos os critérios. Isto indica que a estratégia de seleção de ação em duas etapas deve ser intensificada. Por exemplo, vale perceber as localizações calculadas para todos os pontos de teste. A Tabela 9 apresenta estes resultados.

Tabela 9. Resultados obtidos pelas RNAs de localização

		<i>RNAsala₁</i>		<i>RNAsala₂</i>				<i>RNAsala₃</i>			
<i>x*</i>	<i>y*</i>	<i>x_T</i>	<i>y_T</i>	<i>x*</i>	<i>y*</i>	<i>x_T</i>	<i>y_T</i>	<i>x*</i>	<i>y*</i>	<i>x_T</i>	<i>y_T</i>
4.2	1	0.83	1.24	9.4	2.6	7.44	1.40	13.02	2.10	13.00	1.74
2.8	2.6	1.17	1.45	8.5	4.2	7.39	1.70	13.77	3.15	11.09	1.14
1.4	5	2.00	1.87	6.9	2.6	7.09	1.48				

Comparando-se estes resultados com aqueles produzidos pela arquitetura do agente padrão, pode-se confirmar o aumento na eficiência da localização. A Tabela 10 apresenta os resultados desta comparação disponibilizando as medidas de distância obtidas nos dois casos.

Tabela 10. Resultados obtidos pelas RNAs de localização

<i>Conjunto de Exemplos Ψ</i>	
<i>D_T – Agente Padrão</i>	<i>D_T Agente Refinado</i>
7.20	3.38
3.60	1.99
2.69	3.18
1.50	2.29
0.45	2.74
1.36	1.14
1.32	0.36
2.88	3.35

As células tracejadas, cujos valores estão em negrito, destacam os pontos de teste em que a arquitetura refinada obteve os melhores resultados. Conforme pode ser percebido, ela foi melhor em quatro dos oito pontos de teste apresentados. Mesmo assim, em todos os pontos de teste as diferenças entre os resultados dos dois casos, foram mínimas. Vale ressaltar, estas diferenças deverão ser notadas nas situações em que o ambiente de teste tem dimensões máximas bem maiores que as do ambiente proposto. Nestes casos, o ambiente de teste poderá ser dividido em um número adequado de salas e a eficiência das redes de localização poderá ser aumentada. Em qualquer um dos casos, deve-se investir numa seleção criteriosa dos elementos pertencentes aos conjuntos de treinamentos associados a cada uma das salas que compõem o ambiente.

5.3. Análise dos Resultados

Os resultados dos experimentos envolvendo RNAs na concretização das arquiteturas de agentes que foram propostas, e a comparação destes com resultados obtidos em outros trabalhos, indicam que a abordagem deve ser continuada. Por exemplo, Nerguizian (2004) obteve uma precisão de 0,5 metros em 90% dos pontos que foram treinados e em torno de 5 metros em 45% dos pontos de teste, ou seja, de pontos não treinados. Vale ressaltar, mesmo sem refinamento do processo de seleção de exemplos, esta abordagem conseguiu, na maioria dos casos testados, uma precisão melhor.

Comparando esta abordagem com o trabalho de Battiti e Nhat (2002), os resultados obtidos estão bem próximos, apesar destes autores terem buscado uma estrutura ótima de RNA para o problema. Assim, visando empregar a proposta em casos reais de localização, espera-se que a mesma evolua e dê origem a agentes que sejam realmente racionais, ou seja, que realizem em diversos tipos de ambientes a medida de avaliação de desempenho que foi proposta.

Neste sentido, conforme os resultados obtidos em todos os experimentos, primeiramente, deve-se investir no processo de mineração dos exemplos para treinamento das diversas redes e, posteriormente, conforme os resultados específicos obtidos com a especialização do algoritmo *back-propagation* para problemas de localização, deve-se investir em outras variações deste algoritmo para o problema. Apesar de não comentado, durante os experimentos, foram testadas duas variações promissoras do *back-propagation-alterado*, que envolviam outros parâmetros além das dimensões máximas dos ambientes físicos treinados. Entretanto estas formulações ainda precisam ser testadas intensivamente até que estejam adequadas para publicação.

6. Conclusões e Trabalhos Futuros

De uma maneira geral, os resultados obtidos demonstram que a abordagem proposta neste artigo é promissora. Além dos melhores resultados terem aparecido de maneira evolutiva, a proposta é bastante flexível, principalmente, no que diz respeito à decomposição do ambiente em células bem posicionadas, à segurança e quanto à sua própria implementação.

No que diz respeito à decomposição de ambientes extensos em células menores, esta atividade pode ser realizada em função dos resultados obtidos a partir de diferentes decomposições, aceitando-se uma decomposição do ambiente particular nos casos em que o treinamento de uma rede com esta decomposição produzir um alto grau de generalização.

Em relação aos aspectos de segurança, dispositivos requisitando localização não precisam recorrer a entidades externas, ou seja, o próprio dispositivo pode ser adaptado para determinar sua localização. Por outro lado é possível fazer uso da abordagem para monitorar as posições dos dispositivos móveis baseado nas seqüências de medidas de potências recebidas nos dispositivos.

Quanto aos aspectos de implementação, as arquiteturas abstratas dos agentes foram concretizadas por meio de redes neurais codificadas em C. Entretanto, outras técnicas podem ser utilizadas para a mesma finalidade. O que vai determinar a concretização, conforme aconteceu para esta abordagem específica, é a forma em que as informações estão disponíveis. Do ponto de vista da Inteligência Computacional existem várias possibilidades.

Assim, além da seleção adequada de conjuntos de exemplos e do investimento em algoritmos de treinamento especializados para o problema de localização, esta abordagem deverá evoluir de maneira a refinar o trabalho iniciado visando, principalmente, a concepção de agentes híbridos e, conforme os resultados, dar origem ao *framework* para a implementação de agentes localizadores, proposto na Introdução como objetivo de longo prazo deste trabalho.

Referências

- Bahl, P. and Padmanabhan, V. N. (2000), Radar: An in-building rf-based user location and tracking system. In IEEE INFOCOM.
- Battiti, R. and Brunato, M. and Villani, A. (2002), Statistical Learning Theory for Location Fingerprinting in Wireless Lans. In Technical report DIT-02-0086.
- Battiti, R and Nhat, T.L. and Villani, A (2002). Neural network models for intelligent networks: deriving the location from signal patterns. In Proceedings of AINS.
- Dana, P. H. (2000). "Global Positioning System overview", <http://www.colorado.edu/geography/gcraft/notes/gps/gps.html>
- Haykin, S. (1999), Neural Networks: A Comprehensive Foundation. New Jersey: Ed. Prentice Hall, 2o Ed.
- Hightower, J. and Borrielo, G. (2001). A survey and taxonomy of location systems

for ubiquitous computing. Technical Report UW-CSE 01-08-03, University of Washington.

Kaemarungsi, K. and Krisnamurthy, P. (2004), Modeling of indoor positioning systems based on location fingerprint. In IEEE INFOCOM.

Kasabov, N. K. (1998), Foundations of Neural Networks, Fuzzy Systems, and Knowledge Engineering. The MIT Press, 2^o Ed.

Neguizian, C. and Despins, C. and AFFÈS, S. (2004), Indoor Geolocation with Received Signal Strength Fingerprinting Technique and Neural Networks. In ICT.

Ni, L. M. and Liu, Y. (2004), Landmarc: Indoor location sensing using active rfid. Wireless Networks, 10.

Rappaport, T.S. (2002), Wireless Communications Principles and Practice, Pearson education, second edition.

Russel, S. e Norvig, P (1995), Artificial Intelligence: A Modern Approach. New Jersey: Ed. Prentice-Hall.

Weiss, G (1999), Multiagent Systems: A modern Introduction to Distributed Artificial Intelligence. Cambridge: MIT Press.

Yousief, M (2004), Horus: A Wlan-Based Indoor Location Determination System. Doctoral thesis. University of Maryland.