PUC Minas
A B A K Ó S
Instituto de Ciências Exatas e Informática

Licença Creative Commons Attribution-NonCommercial-NoDerivs 3.0 Unported

# Inteligência computacional aplicada à RSSF: controle de densidades de nós*

Computational intelligence applied to RSSF: density control

Diego Carvalho Pereira Macedo[1]
Harilton da Silva Araújo[2]
Aldir Silva Sousa[3]

**Resumo**

As redes de sensores sem fio (RSSF) são compostas por pequenos sensores capazes de coletar fenômenos do meio ambiente em que estão inseridos e de transmitir esses fenômenos ao observador. Os nós sensores são bastante limitados em seu poder de processamento e em capacidade de energia e de armazenamento. Quando se possui um número elevado de nós sensores em um ambiente, o consumo de energia aumenta. No sentido de resolver o controle de densidade dos nós nas redes de sensores sem fio, neste artigo apresentamos uma proposta para controle de densidades de nós nas redes de sensores sem fio. A proposta foi concebida aplicando-se a inteligência computacional de inferência *fuzzy* no algoritmo de roteamento *Inter Cluster Routing Algorithm* (ICA) a fim de melhorar o desempenho da rede e diminuir o consumo da energia residual dos nós. As simulações feitas neste trabalho mostram que a proposta é eficiente no que diz respeito à redução do consumo de energia, ao aumento no número de mensagens enviadas e recebidas, ao aumento na energia residual da rede, ao aumento da vida útil da rede, bem como à redução do custo de recebimento de mensagem. Esses foram os objetivos alcançados na proposta.

**Palavras-chave:** Redes de sensores sem fio. Roteamento. ICA. *Fuzzy*.

*Submetido em: 10/03/2014 - Aceito em: 09/04/2014

[1]Bacharel em Ciência da Computação pelo Centro de Ensino Unificado de Teresina CEUT, Brasil, e-mail: diego-macedo_cp@hotmail.com.

[2]Doutorando em Informática Aplicada pelo Programa de Pós-graduação em Informática Aplicada da Universidade de Fortaleza - UNIFOR, Brasil, e-mail: harilton@unifor.edu.br.

[3]Doutor em Ciência pelo Programa de Pós-graduação em Engenharia Elétrica da Escola de Engenharia de São Carlos – Universidade de São Paulo. Professor Adjunto da Universidade Estadual do Piauí – UESPI, Brasil, e-mail: aldirss@uespi.br.

**Abstract**

Wireless sensor networks (WSN) consist of small sensors capable of capturing phenomena information of the environment they are inserted and to transmit this information. Sensor nodes are very limited in their processing power and capacity and energy storage. When there is a fairly large number of nodes in an environment, energy consumption is greater. This paper presents a proposal for density control in wireless sensor networks using computational intelligence. Simulations indicate that the proposal is effective with respect to reduction of energy consumption, increase in the number of messages sent and received, increase in the residual energy of the network, increasing the lifetime of the network and reducing the cost of receiving the message.

**Keywords:** Wireless sensor networks. Routing algorithms. Inter cluster routing algorithm. Fuzzy inference system.

## 1 INTRODUÇÃO

Uma rede de sensores sem fio (RSSF) pode ser definida como um conjunto de nós sensores que têm a função de captar as ações do meio em que estão inseridos e, por meio de comunicação sem fio, transmitir informações para outros nós da rede. Estas redes diferem das redes de computadores tradicionais em muitos aspectos: são compostas de um grande número de nós sensores distribuídos, possuem restrições de energia e processamento e devem ter mecanismos para autoconfiguração em caso de perda de comunicação e falhas nos nós sensores. Apesar de suas limitações, quando executam tarefas colaborativamente são muito eficientes (LOUREIRO et al., 2003).

Este trabalho consiste em aperfeiçoar o algoritmo de roteamento *Inter Cluster Routing* (ICA) para lidar com o roteamento visando melhorar o desempenho da RSSF. Neste trabalho, propõe-se a aplicação de um sistema de inferência *fuzzy* no algoritmo de roteamento ICA no intuito de se monitorar um evento em um espaço geográfico, objetivando otimizar as rotas para maximizar o desempenho da rede. Com a abordagem proposta neste artigo, o sistema *fuzzy* possibilita que a rede se autoconfigure no intuito de atender regiões geográficas em que ainda haja energia suficiente e que esteja próxima a algum fenômeno.

Este artigo está organizado da seguinte forma. Na seção seguinte apresentamos uma breve revisão teórica sobre as redes de sensores sem fio. Em seguida, apresentamos, respectivamente, o algoritmo ICA e utilização do sistema de inferência *fuzzy*. Em seguida, mostramos as características da rede onde foi utilizada esta proposta. Finalmente, apresentamos os resultados experimentais e conclusões.

## 2 A METODOLOGIA PROPOSTA

Uma rede de Sensores sem Fio (RSSF), em aplicações práticas, visa monitorar, instrumentar, e possivelmente, controlar o meio físico. Estas redes são constituídas de um número elevado de dispositivos sem fios (nós sensores). Esses são distribuídos aleatoriamente por toda uma região de interesse.

A RSSF pode ser formada por centenas ou até milhares de sensores posicionados dentro do fenômeno a ser observado ou próximo a ele. Os nós sensores são dispositivos compostos de transceptor, fonte de energia, unidade de sensoriamento, processador e memória. Como as redes de sensores sem fio são capazes de se auto-organizar, torna-se dispensável um planejamento minucioso de posicionamento dos sensores.

Cada um dos sensores é alimentado por uma fonte de energia escassa, ou seja, com pouca autonomia. Portanto, o consumo de energia influencia a durabilidade da RSSF. Sabendo disso,

faz-se mister a aplicação de algoritmos de roteamento eficientes, que minimizem a transmissão de informações desnecessárias e/ou redundantes. Este trabalho propõe uma nova abordagem no intuito de minimizar o envio e recepção de informações ao aplicar mecanismos de inteligência computacional para aperfeiçoar um algoritmo de roteamento clássico da literatura.

Nas próximas seções destacam-se o algoritmo de roteamento utilizado neste trabalho e o sistema de inferência *fuzzy* aplicado para seu aperfeiçoamento.

## 2.1 Algoritmo de roteamento ICA

O *Inter Cluster Routing Algorithm* (ICA) é um algoritmo de roteamento em RSSF baseado no algoritmo de roteamento LEACH (HABIB; CÂMARA; LOUREIRO, 2004). O ICA foi desenvolvido para redes de sensores que coletam informações periodicamente durante o tempo de vida da rede. Para estes ambientes a vida útil da rede é normalmente um fator muito importante. No ICA, quando há o início do funcionamento da rede, a estação base envia mensagem (*broadcast*) para os nós da rede informando sua posição geográfica do ambiente. Após isso, os nodos sabem a posição geográfica do *sink*, suas devidas posições e a distância e a rota até o *sink*. O ICA possibilita ao nó se tornar líder (*cluster head*) de uma determinada região. Depois disso, os nós próximos se conectam ao *cluster head* e os nós que possuírem informações a serem enviadas para o *sink* as mandarão para o *cluster head*, para depois este repassá-las para a estação base. Depois de um nó se eleger líder, esse manda um *broadcast* para os nós próximos avisando-os de que ele se tornou um *cluster head*. Assim, estes nós terão a posição geográfica deste nó líder e terão uma rota mais curta até ele. Com isso, os nós sabem a localização do nodo líder mais próximo, com o qual devem se comunicar. Portanto, no ICA, os nós sensores não enviam as mensagens diretamente para o *sink*, e sim para outro *cluster head* mais próximo e que esteja na direção da estação base.

O algoritmo ICA possibilita uma degradação suave da energia da rede como um todo e não necessariamente nodo por nodo.

## 2.2 Utilização do *fuzzy*

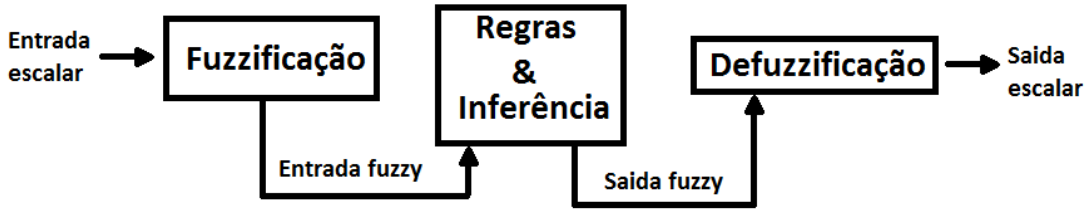
O sistema de inferência *fuzzy* foi utilizado no ICA de forma a selecionar os nós líderes que possuam a distância mais próxima a um local que se está acontecendo algum tipo de fenômeno no ambiente e que possua energia suficiente. Portanto, o sistema exerce a função de identificar aqueles nós que podem se tornarem líderes. Por exemplo, quando acontece um evento em algum local, a rede se autoconfigura de forma que os nós determinem suas distâncias ao evento mais próximo; após isso entra o funcionamento do *fuzzy* que com o dado da distância

e a energia do nó, elegerá os nós permitidos na rede. Na próxima seção será apresentada a característica do sistema *fuzzy*

## 2.3 Sistema de inferência *fuzzy*

Um sistema de inferência *fuzzy* foi aplicado e tem fundamental importância para o bom desempenho desta proposta. O sistema de inferência *fuzzy* é capaz de lidar com processos altamente complexos, que são representados por informação qualitativa. Normalmente, sistemas de inferência *fuzzy* são baseados em regras linguísticas do tipo **se** <condição> **então** <ação> (SOBRAL et al., 2013). O sistema de inferência *fuzzy* aplicado neste trabalho é composto por uma interface de *fuzzyficação*, uma base de regras o procedimento de inferência e a interface de *defuzzificação*, conforme a Figura 1.

**Figura 1 – Estrutura básica de um controlador *fuzzy***




**Fonte: Elaborada pelos autores**

- *Interface de fuzzyficação*: é o processo de mapear valores escalares para graus de pertinência aos termos linguísticos dos conjuntos *fuzzy*.
- *Procedimento de inferência*: consiste em mapear valores de entradas para saídas *fuzzy* de acordo com as regras *fuzzy*. Portanto diversas regras *fuzzy* são combinadas para produzir a saída do sistema de inferência.
- *Interface de defuzzyficação*: consiste em converter as saídas *fuzzy* em valores escalares.

### *2.3.1 Funções de pertinência e regra de linguística*

O sistema *fuzzy* desenvolvido neste artigo baseia-se em duas variáveis de entrada e uma de saída. As variáveis *fuzzy* de entrada são *distância do nó e energia contida no nó*. A variável de saída, aqui chamada de *chance de se tornar líder*, consiste em saber se o nó tem chance ou não para se tornar líder no ICA.

Os conjuntos *fuzzy* atribuídos à variável *distância do nó* foram: Longe (L), Distância Média (DM) e Perto (P). O universo de discurso destes conjuntos está no intervalo discreto

entre 0 a 5. Para a *energia contida no nó*, foram atribuídos os seguintes conjuntos: Muito Baixa (MB), Baixa (B), Média (M), Alta (A) e Muito Alta (MA). O universo de discurso destes conjuntos está no intervalo contínuo entre 0 e 5. A *chance de se tornar líder* é composta pelos seguintes conjuntos *fuzzy*: Pouca Chance (PC), Chance Média (CM) e Muita Chance (MC). O universo de discurso destes conjuntos está no intervalo contínuo entre 0 e 30.

Todas as variáveis de entrada foram modeladas como função de pertinência trapezoidal. A Equação (1) define a função trapezoidal para os parâmetros $a, b, c, d$.

(1)

A seguir, listam-se os parâmetros TRAPE[a b c d] para cada um dos conjuntos *fuzzy* utilizados neste trabalho:

- L: TRAPE[3 4 5 5]
- DM: TRAPE[ 1 2 3 4]
- P: TRAPE[0 0 1 2]
- MB: TRAPE[ 0 0 0.5 2]
- B: TRAPE[0.5 1 1.5 2]
- M: TRAPE[1.5 2 2.5 3]
- A: TRAPE[2.5 3 3.5 4]
- MA: TRAPE[3.5 4 5 5]

Na Tabela 1, resumem-se as regras linguísticas do *fuzzy* utilizada neste artigo. Os resultados serão PC (Pouca Chance), CM (Chance Média) e MC (Muita Chance):

**Tabela 1 – Exemplo de sequências em uma base de dados empírica**

| | | Energia | | | | |
|---|---|---|---|---|---|---|
| | | MB | B | M | A | MA |
| Distância | L | PC | PC | PC | PC | PC |
| | DM | PC | PC | CM | CM | MC |
| | P | PC | PC | CM | MC | MC |

**Fonte: Dados da pesquisa**

### 2.4 Algoritmos de eleição do *Cluster Head*

**Algorithm 1:** Principal método para eleger o *Cluster Head*

```
1:  Public void preStep(){
2:  ...
3:      if (Tools.getGlobalTime() % 10 == 0){
4:          Simplenode.rotaPrincipalFeita = true;
5:          if rotaPrincipalFeita && Tools.getGlobalTime() % 10 == 0) {
6:              Double t = p / (1 – p  ((Tools.getGlobalTime() / 10) % umSobreP));
7:              Double random = Math.random();
8:              if (random < t) {
9:                  if (g >= umSobreP ) {
10:                     Double valor = fuzzy.resultadofuzzy(this.energia,
                        this.distanciaAoEventoMaisProximo);
11:                     if (Double.compare(valor,10.0) > 0 && Double.compare(valor,30.0) < 0){
12:                         clusterStart();
13:                     }
14:                 }
15:         } else if (clusterHead)
16:             clusterStop();
17:             g++;
18:         }
19:     }
20: }
```

**Fonte: Baseado em Dijkstra**

Este é o método que é chamado a cada *round* no simulador, mas somente são eleitos os novos *cluster heads* na rede quando o tempo em *round* atinge um valor múltiplo de dez. Os valores dos *rounds* são obtidos pelo método *Tools.getGlobalTime()*. O valor de p é o percentual desejado de nós agregadores, o *umSobreP* refere-se a 1/*p* e *g* ao arredondamento de *p*. A classe *Fuzzy* é responsável por determinar se o nó será líder ou não pelas regras determinada no *fuzzy*; para isso ela chama o método *resultadoFuzzy()* passando por parâmetro a energia e a *distância* do nó até um evento mais próximo. Será retornado um valor *Double* se o valor estiver entre dez e trinta, que é a interseção do *fuzzy* trapezoidal que indicará que este nó tem chance de se tornar o *cluster head* da rede.

## 3 TESTES COMPUTACIONAIS

Com objetivo de avaliar o desempenho da abordagem proposta neste trabalho simulações foram realizadas com o simulador SINALGO (2008). Este simulador foi escolhido por ser largamente utilizado na literatura e de fácil configuração.

O pacote java *jFuzzyLogic* foi utilizado para aplicar o sistema de inferência *fuzzy* (FUZZY-

LOGIC, 2013).

### 3.1 Ambiente dos testes

As simulações foram realizadas em um Computador *HP Pavilion dv4 – 2040br*, com sistema operacional Windows 7, processador *Intel Core i3 CPU M 330 @ 2,13GHz, 2128MHz*. Memória RAM de 4 GB DDR3 e memória cache L2 256 KB.

### 3.2 Características da rede

As simulações foram realizadas em redes simuladas compostas de 200 nós sensores e um *sink node*, dispostos de forma aleatória em um ambiente de duas dimensões sendo de 1500x1500. O tempo em *rounds* determinado para cada simulação foi de 15.000 *rounds* e as posições geográficas dos nós foram salvas para que todas as simulações fossem realizadas com a mesma configuração, dando segurança às comparações realizadas. Foi configurado para que cada nó sensor possua uma bateria com capacidade inicial de 5 J (*joule*).

Nas simulações realizadas foi utilizado o modelo de dissipação de energia proposto por Heinzelman, Chandrakasan e Balakrishnan (2000), por ser o amplamente aceito e utilizado na literatura. O modelo pode ser sintetizado conforme segue.

**Energia gasta na transmissão:**

$E_{Tx}(k,d) \ = \ E_{elec} * k + E_{amp} * k * d^2$

**Energia gasta na recepção:**

$E_{Rx}(K) \ = \ E_{elec} * k$

**Especificações:**

$k =$ número de bits da mensagem

$d =$ distância

$E_{Tx} =$ Energia gasta na transmissão de mensagem

$E_{Rx} =$ Energia gasta na recepção da mensagem

$E_{amp} =$ Energia do amplificador de transmissão

Neste trabalho, $E_{elec} = 50nJ/bit$ e $E_{amp} = 100pJ/bit/m2$. Nas simulações realizadas, o tamanho das mensagens que trafegam na rede foi configurado para 500 bytes, que é igual a 4000 bits.

Na unidade de processamento, foi definido que a energia dissipada nas execuções dos protocolos de roteamento e no processamento.

No modelo de comunicação, a energia gasta no processamento equivale à energia dissipada na transmissão de um bit a 100 metros dividido por 3.000, conforme a Equação (2).

$$E_{Trans} = E_{Cir} * 1 + E_{amp} * 1 * 100^2 \tag{2}$$

Logo:

$$Energia\ de\ \textbf{\textit{Instrução}} = ((E_{Cir} + 1000E_{Amp}))/3000 \tag{3}$$

onde

ETrans = Energia de transmissão

Eelec = ECir = Energia gasta nos circuitos do rádio

EAmp = Energia do amplificador de transmissão.

Neste trabalho, foram realizadas 10 simulações para cada cenário para se fazer comparação destas informações.

### 3.3 Resultados

Os resultados exibem a comparação de dois cenários: no primeiro foi utilizado o algoritmo de roteamento ICA clássico, sem o sistema de inferência *fuzzy* aqui proposto. No segundo cenário, foi utilizado o algoritmo de roteamento ICA com a utilização do sistema de inferência *fuzzy* no intuito de obter o controle de densidade dos nós na rede.

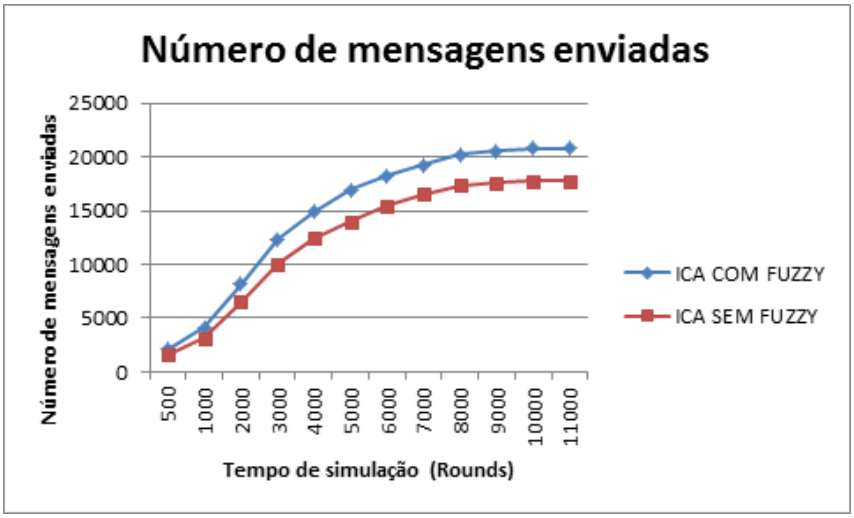
As métricas utilizadas como forma de comparação dos dois cenários foram: a quantidade de mensagens enviadas, a quantidade de mensagens recebidas, a energia restante da rede, o custo de recebimento de mensagens e o tempo de vida da rede. Abaixo, definem-se as métricas utilizadas.

- A quantidade de mensagens enviadas se refere a quantas mensagens a rede dos sensores tentou mandar para estação base;
- A quantidade de mensagens recebida se refere à quantidade de mensagens que o *sink* conseguiu receber dos nós.
- A métrica da energia restante da rede se refere ao somatório das energias restantes em cada nó da rede nó fim da simulação.

- A métrica do custo do recebimento de mensagens leva em consideração a quantidade de mensagens recebidas pela energia residual total da rede, obtendo assim, o valor do custo do recebimento da mensagem.
- A métrica do tempo de vida da rede se refere a até quando a estação base conseguiu receber mensagens. Assim que a estação base deixar de receber mensagens a rede será considerado como morta.

A Figura 2 mostra os resultados das simulações tendo como métrica a quantidade de mensagem enviada ao *sink*.

**Figura 2 – Quantidade de mensagens enviadas x *rounds* (tempo)**




**Fonte: Elaborada pelos autores**

O cenário do ICA com a utilização do *fuzzy* obteve resultados significativamente melhores do que o cenário do ICA clássico. O ICA com *fuzzy* começa a se destacar no número de envios de mensagens pelos nós sensores ao *sink* a partir de 1.000 *rounds* (Figura 2).

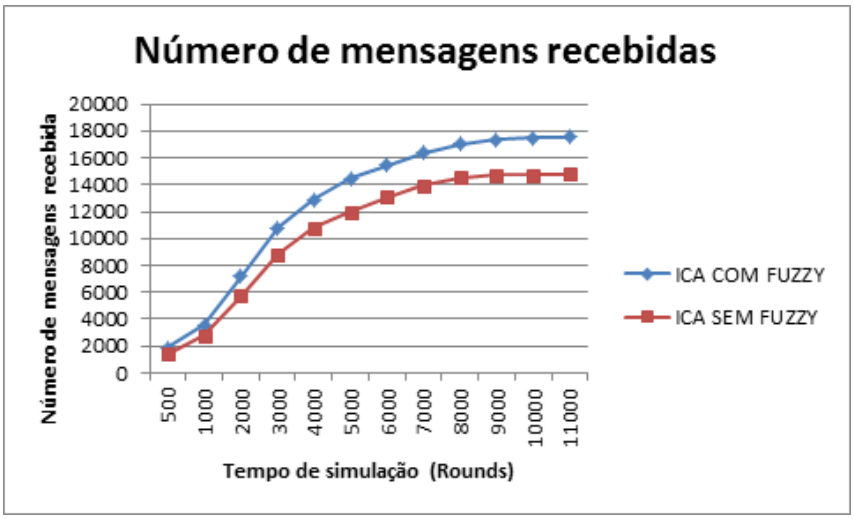
A Figura 3 mostra os resultados das simulações tendo como métrica a quantidade de mensagem recebida no *sink*.

O cenário do ICA com a utilização do *fuzzy* teve melhor desempenho também para a métrica quantidade de mensagens recebidas por rounds. O ICA com *fuzzy* começa a se destacar no número de envios de mensagens pelos nós sensores ao *sink* a partir dos 1000 rounds (Figura 3).

A Figura 4 mostra os resultados das simulações tendo como métrica a energia residual da rede. Como se pode observar na referida figura, o ICA com *fuzzy* obteve um melhor consumo de energia na rede, fazendo com que a rede não desperdiçasse energia e obtendo assim uma maior economia de energia.

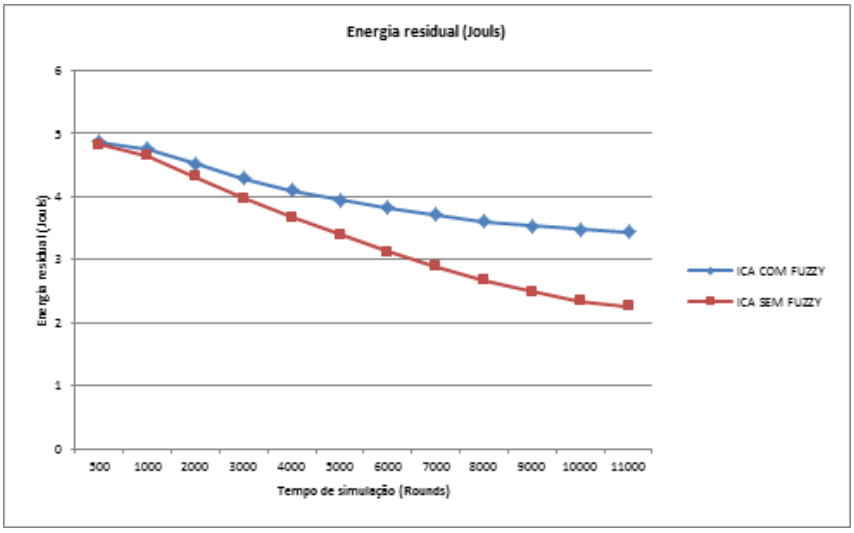
A Figura 5 mostra os resultados das simulações tendo como métrica de comparação o custo de recebimento de mensagens da rede, que é a razão da quantidade de mensagens recebida pela energia residual total da rede.

**Figura 3 – Quantidade de mensagens recebidas x *rounds* (tempo)**



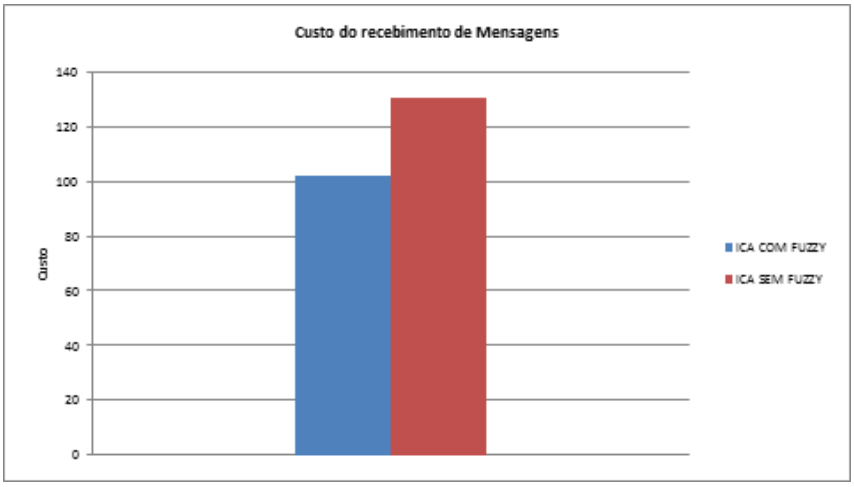

**Fonte: Elaborada pelos autores**

**Figura 4 – Energia residual da rede x *rounds* (tempo)**




**Fonte: Elaborada pelos autores**

**Figura 5 – Custo do recebimento de mensagens**

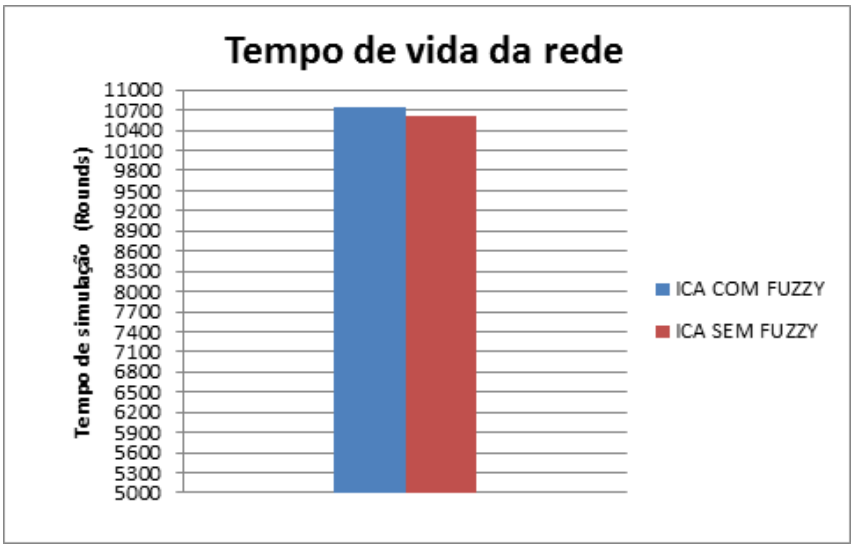



**Fonte: Elaborada pelos autores**

O ICA clássico possui um valor bastante elevado no custo de recebimento de mensagens, sendo, portanto, o ICA com *fuzzy* muito mais viável do que o ICA clássico (Figura 5).

A Figura 6 mostra os resultados das simulações tendo como referência o tempo de vida útil da rede, que é o tempo necessário para que a rede pare de funcionar.

**Figura 6 – Tempo de vida da rede**




**Fonte: Elaborada pelos autores**

Em relação ao tempo de vida da rede, o ICA com *fuzzy* se destacou melhor que o ICA clássico, considerando o fim da rede quando o *sink* parou de receber mensagem dos nós.

## 4 Conclusão

Este artigo apresenta um novo algoritmo de roteamento utilizando inteligência computacional de inferência *fuzzy* aplicado no algoritmo de roteamento *Inter Cluster Routing Algorithm* (ICA) baseado no LEACH (HABIB; CÂMARA; LOUREIRO, 2004). A utilização da inteligência computacional otimizou o algoritmo ICA clássico possibilitando que a rede se autoconfigurasse no intuito de melhorar o controle de densidade de nós em uma RSSF. Com essa abordagem, uma RSSF tem a possibilidade de se autoconfigurar mediante a ocorrência de um evento futuro em outra região geográfica da rede se ela ainda possuir energia residual. O algoritmo ICA clássico opera dessa forma, já que os líderes são eleitos aleatoriamente e nem sempre em locais apropriados. Esse fato incorre em maior consumo de energia.

Com base nos resultados obtidos, foram realizadas análises comparativas considerando o algoritmo ICA com o sistema de inferência *fuzzy* e o algoritmo ICA clássico. As seguintes métricas forma avaliadas: número de mensagens enviadas, mensagens recebidas, energia residual, custo de recebimento de mensagens e o tempo de vida da rede. Em todas as simulações realizadas, foi possível observar que um ganho significativo em desempenho da rede para todas as métricas analisadas.

## REFERÊNCIAS

FUZZYLOGIC. **Open Source Fuzzy Logic library and FCL language implementation**. [S.l.], 2013. Disponível em: <http://jfuzzylogic.sourceforge.net/html/index.html>. Acesso em: 16 maio 2013.

HABIB, E.; CÂMARA, D.; LOUREIRO, A. A. Ica: Um novo algoritmo de roteamento para redes de sensores. In: SIMPÓSIO BRASILEIRO DE REDES DE COMPUTADORES, 22, 2004. **Anais...** Gramado: SBRC, 2004. p. 147–160.

HEINZELMAN; CHANDRAKASAN, A.; BALAKRISHNAN, H. Energy-efficient communication protocol for wireless microsensor networks. In: ANNUAL HAWAII INTERNATIONAL CONFERENCE, 33, 2000. **Proceedings...** Hawaii: System Sciences (IEEE), 2000. p. 10.

LOUREIRO, Antônio A. F. et al. Redes de sensores sem fio. In: SIMPÓSIO BRASILEIRO DE REDES DE COMPUTADORES, 2003. **Anais...** Belo Horizonte: SBRC, 2003. p. 179–226. Disponível em: <http://homepages.dcc.ufmg.br/l̃oureiro/cm/docs/sbrc03.pdf>. Acesso em: 16 maio 2013.

RIBEIRO, Anna Izabel João Tostes. **Representações neural e *fuzzy* de controle de admissão de chamadas para redes E-UMTS**. 2010. Dissertação (Mestrado) — Pontifícia Universidade Católica de Minas Gerais, Programa de Pós-graduação em Informática, Belo Horizonte.

SINALGO. **Simulator for Network Algorithms**. [S.l.], 2008. Disponível em: <http://www.disco.ethz.ch/projects/sinalgo/>. Acesso em: 16 maio 2013.

SOBRAL, J. et al. A fuzzy inference system for increasing of survivability and efficiency in wireless sensor networks. In: INTERNATIONAL CONFERENCE ON NETWORKS, 20, 2013, Seville, Spain. **Proceedings...** [S.l.]: ICN, 2013. p. 34–41.

SOUSA, A. S.; ASADA, E. N. Combined heuristic with fuzzy system to transmission system ex-pansion planning. **Electric Power Systems Re-search**, Elsevier, v. 81, n. 1, 2011.