UNIVERSIDADE ESTADUAL DE FEIRA DE SANTANA
Autorizada pelo Decreto Federal nº 77.496 de 27/04/76
Recredenciamento pelo Decreto nº17.228 de 25/11/2016
PRÓ-REITORIA DE PESQUISA E PÓS-GRADUAÇÃO
COORDENAÇÃO DE INICIAÇÃO CIENTÍFICA

XXIV SEMINÁRIO DE INICIAÇÃO CIENTÍFICA DA UEFS
SEMANA NACIONAL DE CIÊNCIA E TECNOLOGIA - 2020

UM SISTEMA FUZZY GENÉTICO PARA O GERENCIAMENTO DE ENERGIA DE UMA REDE DE SENSORES SEM FIO

Jaevillen F. Oliveira[1]; Matheus G. Pires[2]

1. Bolsista PIBIC/CNPq, Graduanda em Engenharia de Computação, Universidade Estadual de Feira de Santana, e-mail: jaevillenks@gmail.com

2. Orientador, DEXA, Universidade Estadual de Feira de Santana, e-mail: mgpires@ecomp.uefs.br

PALAVRAS-CHAVE: sistema fuzzy genético; redes de sensores sem fio; consumo de energia.

INTRODUÇÃO

Na sociedade revolucionada pelos meios de comunicação, tem se popularizado o uso de Redes de Sensores Sem Fio (RSSF) em diversas aplicações, dentre as quais, destaca-se os sistemas de monitoramento. Essa popularidade, impulsionada pela diminuição do preço desta tecnologia, também apresenta um aumento na demanda por mais eficiência, principalmente a energética, que é um aspecto importante a ser considerado, pois afeta diretamente a qualidade na transmissão dos dados [1].

Atualmente, abordagens baseadas em Inteligência Artificial, mais especificamente os Sistemas Baseados em Regras *Fuzzy* (SBRF), têm sido usados para para controlar o comportamento das RSSF [1,2]. Um fator importante a ser considerado na aplicação de um SBRF é a modelagem dos conjuntos *fuzzy* das variáveis e a construção da Base de Regras, pois estes dois componentes constituem a Base de Conhecimento de um SBRF, a qual possui relação direta no desempenho final do sistema. Sendo assim, formas automáticas no aprendizado da Base de Conhecimento de um SBRF vem sendo amplamente utilizadas, com destaque para o uso de *Algoritmos Genéticos* (AG) [3], os quais realizam um processo de otimização ou busca em um espaço de soluções em potencial [4] .

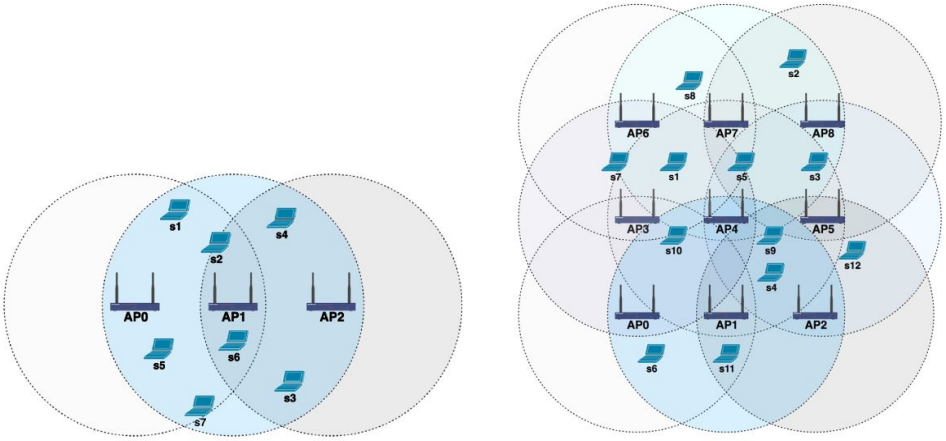
Neste contexto, este trabalho faz uso de um SBRF para controlar uma rede de sensores sem fio, visando a eficiência e economia de energia da rede. Além disso, um AG é utilizado para a otimização dos conjuntos *fuzzy* das variáveis do SBRF, com o objetivo de aumentar a eficiência da rede.

METODOLOGIA

Primeiramente, foi necessário a criação de um ambiente para simulação da RSSF. Para isto, foi utilizada a ferramenta *Omnet++* [5], que é um simulador de eventos modular

orientado a objetos. Além deste simulador, também foi usado o *framework INET* [6], que é um modelo *Omnet++* para redes com e sem fio e aplicações móveis.

Em relação aos experimentos, baseado no trabalho de Collotta et al. (2018), foram testadas duas redes com diferentes topologias. No primeiro cenário, a rede possui três pontos de acesso e sete sensores (ver Figura 1a). No segundo cenário, a rede possui nove pontos de acesso e doze sensores (ver Figura 1b).



(a) (b)

Figura 1. Configuração das RSSF. Fonte: Collotta et al (2018).

Cada teste possui um tempo de simulação de 240 segundos, onde os sensores enviam pacotes de 1500 *bytes* através de seus respectivos pontos de acesso, em intervalos de tempo determinados através da distribuição de Poisson [7]. Para cada cenário são realizados 10 testes, com diferentes padrões de transmissão.

O SBRF, responsável por controlar o funcionamento dos pontos de acesso (ligado/desligado), foi implementado no *MATLAB* [8]. As variáveis de entrada do SBRF descrevem informações dos pontos de acesso, que são, *Received Signal Strength Indication* (RSSI); número de pontos de acesso vizinhos; número de sensores conectados ao ponto de acesso e *throughput*. A variável de saída controla o ponto de acesso da rede, ou seja, se o ponto de acesso permanecerá ligado ou não.

A base de regras foi construída a partir da observação dos valores das variáveis de entrada durante algumas simulações, resultando em uma base com 27 regras.

Por fim, o AG foi implementado em *JAVA* utilizando o *Framework JMetal* [9], versão 5.6. Como o AG é responsável por otimizar os conjuntos *fuzzy*, cada indivíduo (cromossomo) representa uma codificação dos parâmetros dos conjuntos das variáveis de entrada e saída do SBRF. A população inicial foi gerada aleatoriamente, com exceção de um indivíduo modelo, que é a codificação dos conjuntos com distribuição uniforme. O operador de seleção utilizado foi o Torneio e taxa de Elitismo de 30%. Os operadores de cruzamento e mutação utilizados foram *BLX Alpha* e Mutação Simples, com taxas de 70% e 20%, respectivamente. A função de *fitness* é a soma do consumo de energia da RSSF para todos os 20 testes (10 testes para o cenário 1 e 10 testes para o cenário 2.

RESULTADOS E DISCUSSÃO

A modelagem dos conjuntos *fuzzy* encontrada pelo AG está ilustrada nas Figuras de 2 a 6.

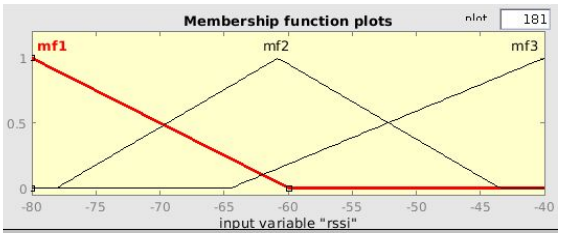


Figura 2. Variável *RSSI*.

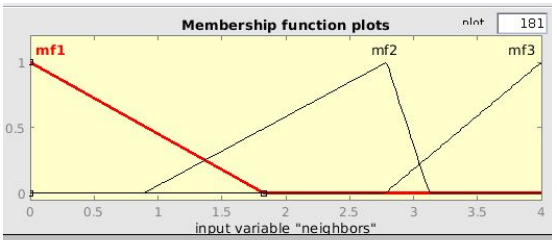


Figura 3. Variável *Número de vizinhos*

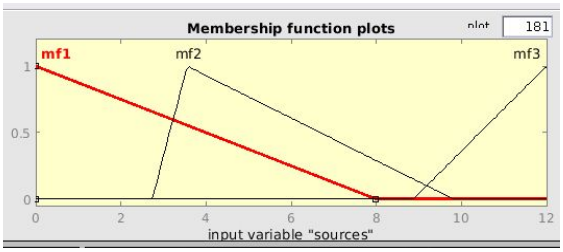


Figura 4. Variável *Número de Sensores.*

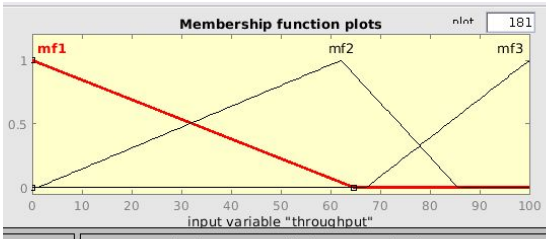


Figura 5. Variável *Throughput*.

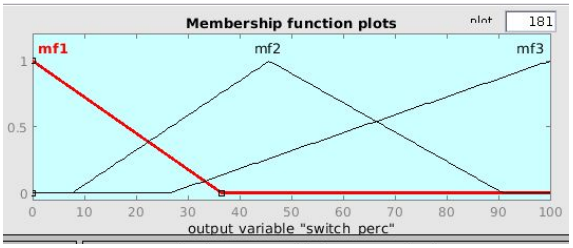


Figura 6. Variável de saída.

Nas Figuras 7 e 8 são ilustrados o consumo energético das redes para cada teste, nos cenários 1 e 2, utilizando as abordagens A*lways On* (os pontos de acesso permanecem ligados durante toda a simulação), *Random Off* (os pontos de acesso são desligados aleatoriamente), *Fuzzy Controlled* (os pontos de acesso são gerenciados pelo SBRF), e *GA Fuzzy Controlled (*os pontos de acesso são gerenciados pelo SBRF otimizado pelo algoritmo genético).

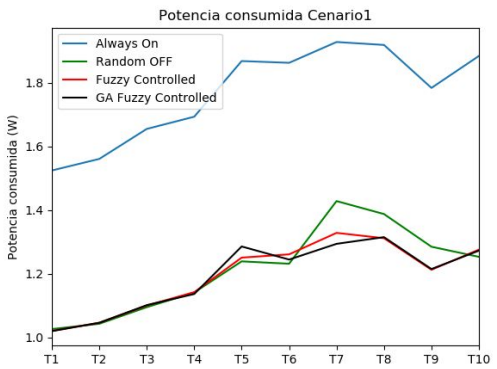


Figura 7. Potência consumida cenário 1.

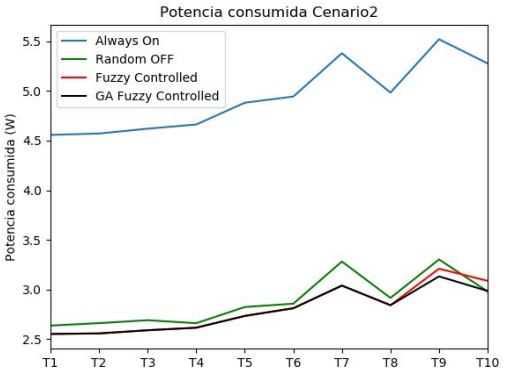


Figura 8. Potência consumida cenário 2.

Comparando o consumo energético das redes usando as abordagens *Fuzzy Controlled* e *GA Fuzzy Controlled,* a primeira obteve um consumo de energia total de 39,98 *Watts*, e a segunda obteve um consumo de 39,78 *Watts*. Portanto, uma redução de apenas 0,2 *Watts* (0,5% do consumo inicial). Esta diferença foi pequena e perceptível em apenas alguns dos testes.

Em relação às medidas de qualidade, tanto a abordagem *Fuzzy Controlled* quanto a *GA Fuzzy Controlled* obtiveram melhores resultados em perda de pacotes, *jitter* e latência, quando comparadas com as demais. Entre as abordagens *Fuzzy Controlled* e *GA Fuzzy Controlled*, os resultados são muito próximos.

CONSIDERAÇÕES FINAIS

Os resultados obtidos nos experimentos mostraram que ambas as abordagens *Fuzzy Controlled* e *GA Fuzzy Controlled* obtiveram melhores resultados quando comparados com as demais abordagens. No entanto, a abordagem *GA Fuzzy Controlled* obteve uma redução no consumo de energia de apenas 0.2 Joules em comparação com a abordagem *Fuzzy Controlled*. Assim, esta pequena diferença não é suficiente para afirmar que o AG conseguiu gerar uma modelagem *fuzzy* melhor que a modelagem uniforme.

Por outro lado, vale destacar que não foi possível realizar muitos experimentos com o AG, ou seja, variar os valores dos seus parâmetros. Essa variação é importante quando se utiliza algoritmos genéticos, pois estes parâmetros afetam diretamente o resultado final do algoritmo. Portanto, pretende-se como trabalhos futuros, realizar mais experimentos variando, além dos valores, os tipos de operadores genéticos, buscando uma otimização genética que resulte em um sistema *fuzzy* mais robusto.

REFERÊNCIAS

[1] COLLOTTA, Mario; PAU, Giovanni; COSTA, Daniel G. A fuzzy-based approach for energy-efficient Wi-Fi communications in dense wireless multimedia sensor networks. **Computer Networks**, v. 134, p. 127-139, 2018.

[2] COSTA, Daniel et al. A fuzzy-based approach for sensing, coding and transmission configuration of visual sensors in smart city applications. **Sensors**, v. 17, n. 1, p. 93, 2017.

[3] Mitchell, M. **An Introduction to Genetic Algorithms**, The MIT Press, 1996.

[4] Herrera, F. Genetic fuzzy systems: taxonomy, current research trends and prospects. **Evolutionary Intelligence**, vol.1, n.1, pp.27-46, 2008

[5] **OMNeT++ Discrete Event Simulator**. Disponível em: <https://omnetpp.org/>. Acesso em: 28 de jul. de 2020.

[6] **INET Framework**. Disponível em: <https://inet.omnetpp.org/>. Acesso em: 28 de jul. de 2020.

[7] C. Daskalakis, I. Diakonikolas, R.A. Servedio, Learning poisson binomial distributions, **Algorithmica**, vol.72, pp.316–357, 2015.

[8] **MATLAB**.Disponível em: <https://www.mathworks.com/products/matlab.html>. Acesso em: 28 de jul. de 2020.

[9] **jMetal 5**. Disponível em: <https://jmetal.github.io/jMetal/>. Acesso em: 28 de jul.de 2020.