

# Programação Automática de Redes de Sensores sem Fio Utilizando Programação Genética

Renato R. R. de Oliveira<sup>1</sup>, Tales Heimfarth<sup>1</sup>, Ariel F. F. Marques<sup>1</sup>  
Raphael W. de Bettio<sup>1</sup>, Jesimar S. Arantes<sup>1</sup>

<sup>1</sup>Departamento de Ciência da Computação  
Universidade Federal de Lavras (UFLA)

Câmpus Universitário, C.P. 3037 – 37200-000 – Lavras – MG – Brasil  
renatorro@comp.ufla.br, {tales, raphaelwb}@dcc.ufla.br,  
{arielffmarques, jesimar.arantes}@gmail.com

**Resumo.** A programação de Redes de Sensores Sem Fio (RSSF) é uma tarefa complexa devido à programação em linguagens de baixo nível e à necessidade de uma aplicação distinta para cada nó sensor. Além disso, sensores sem fios possuem grandes limitações de hardware, como baixo poder de processamento, pouca memória e limitação energética. Dessa forma a programação automática de RSSF é desejável, uma vez que pode-se contemplar essas dificuldades automaticamente, além de economizar em custos, pois elimina a necessidade de alocar um desenvolvedor para programar a RSSF. O presente artigo apresenta um estudo inicial acerca de um framework baseado em programação genética que é capaz de gerar automaticamente aplicações para RSSF. O framework é composto por um middleware que provê uma linguagem de programação de alto nível para nós sensores, um simulador de redes sem fio e um algoritmo genético responsável por gerar códigos-fonte de aplicações para RSSF. Experimentos computacionais mostram que a abordagem é muito promissora, uma vez que é capaz de solucionar instâncias de um problema de detecção de eventos de forma ótima. Entre as instâncias solucionadas, existem instâncias grandes com mais de 500 nós sensores.

## 1. Introdução

O presente artigo propõe um *framework* baseado em Programação Genética (PG) para realizar a programação de Redes de Sensores sem Fios (RSSF) de forma automática. O usuário do *framework* necessita apenas definir o objetivo geral da RSSF e o *framework* é capaz de gerar aplicações para essa RSSF de forma autônoma.

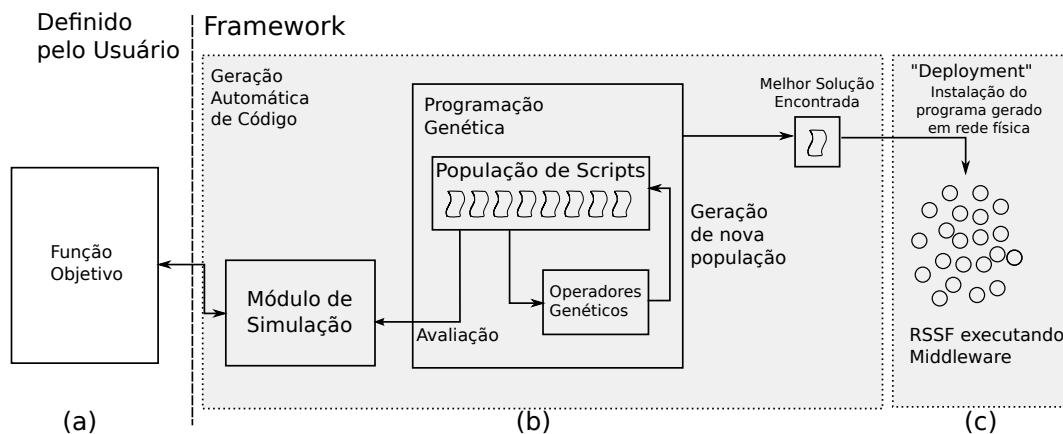
Foram encontrados poucos trabalhos similares na literatura, um deles propõe uma abordagem similar focada apenas em problemas de rastreamento de alvos através de redes reguladoras de genes [Markham and Trigoni 2011]. Já [Weise 2006] realiza a programação automática de RSSF utilizando programação genética, entretanto os problemas tratados são muito simplificados e a comunicação entre os nós sensores não é levada em consideração, todas as mensagens são enviadas em *broadcast*.

## 2. Visão Geral do *Framework*

O principal componente do sistema é um *middleware* que é responsável pela execução dos programas escritos em uma linguagem *script* em cada nó sensor. Os nós sensores são

distribuídos em uma determinada área a ser monitorada e realizam suas tarefas que são codificadas em um *script* executado por uma máquina virtual presente no *middleware*. Cada nó sensor da rede executa uma cópia local do *middleware*. O código *script* contém todas as instruções necessárias para a realização da tarefa especificada pelo usuário. Esse código é inserido na rede, replicado para todos os nós e executados pela máquina virtual.

Pode-se encontrar na literatura várias abordagens de *middleware* orientados a agentes móveis para a (re)programação de RSSF [Heimfarth et al. 2010]. A inovação da presente proposta apresenta-se no método de geração do código *script*: ao invés da convencional geração manual do código utilizado em cada nó sensor, uma abordagem de geração automática baseada em programação genética (PG) será utilizada. A Figura 1 apresenta o processo. Em um sistema convencional, o usuário descreve as ações que devem ser realizadas pelos nós sensores através de alguma linguagem imperativa. Na presente abordagem, o usuário tem somente a tarefa de descrever uma função objetivo, que será utilizada para avaliar os *scripts* candidatos gerados pela PG. Isso é mostrado em 1 (a).



**Figura 1. Visão Geral do Framework Proposto**

A geração automática de código através da PG é mostrada na 1 (b). O método de programação genética, baseia-se em uma população com um conjunto de indivíduos. Cada indivíduo é um *script* que pode ser executado pela RSSF. A ideia da PG é a evolução de *scripts* aleatórios iniciais em direção a um programa que solucione de maneira eficiente a tarefa desejada pelo usuário.

Em cada geração, cada *script* da população é avaliado utilizando a função objetivo definida pelo usuário. Como uma avaliação real utilizando nós sensores físicos de cada solução candidata não é possível, um módulo de simulação é utilizado nessa etapa. Cada *script* candidato da população é copiado para todos os nós sensores dentro do simulador e é executado em uma versão simulada do *middleware*. Durante a execução no simulador, uma avaliação do desempenho funcional e não funcional do *script* é feita utilizando-se para isso a função objetivo definida pelo usuário. Após essa fase, toda a população de *scripts* terá sido avaliada, e o próximo passo é a geração de uma nova população utilizando-se para isso operadores genéticos que combinam os indivíduos da população atual (baseados na avaliação dos indivíduos).

Assim, uma nova geração de indivíduos é gerada. O processo é novamente iniciado com a avaliação desses novos indivíduos. Após a execução desse ciclo por um

período determinado de tempo, a PG encerra e a melhor solução gerada será utilizada no ambiente real. A Figura 1 (c) ilustra a instalação do *script* final gerado nos nós sensores reais. Cada nó executa uma versão local do *middleware* que é responsável pela execução do código do *script*.

### 3. Resultados e Discussão

Com a finalidade de avaliar a capacidade do *framework* proposto, quatro instâncias de teste foram criadas para o problema de detecção de eventos utilizando RSSF. Cada instância é uma representação de uma RSSF com topologia de rede em grade. As instâncias possuem respectivamente 25, 49, 225 e 625 nós sensores. Para cada uma das quatro instâncias a PG foi executada 10 vezes e o resultado médio foi considerado nas análises. A utilização da média de várias execuções é importante uma vez que a PG desenvolvida é um método estocástico.

O Algoritmo Genético foi configurado para utilizar uma população de 20 indivíduos. A taxa de *crossover* foi ajustada para 0,6 e a taxa de mutação para 0,85. O AG utiliza elitismo, ou seja, o melhor indivíduo de cada geração é copiado sem alterações para a geração seguinte. A seleção de indivíduos foi realizada através de torneio com tamanho 3. Os parâmetros foram ajustados através de testes empíricos.

A Tabela 1 apresenta os resultados médios obtidos para as 10 execuções em cada uma das instâncias utilizadas. Todas as execuções foram realizadas por 2.000 gerações do algoritmo genético. A Tabela 1 mostra o valor de *fitness* ótimo para cada instância. Esse valor de *fitness* ótimo é obtido analisando cada instância e verificando o comportamento ótimo da rede, ou seja, o número mínimo de mensagens necessárias para comunicar o nó *sink* da ocorrência de cada um dos eventos ocorridos.

Todas as execuções para todas as instâncias foram capazes de alcançar a solução ótima. Assim a tabela apresenta o número médio de gerações necessárias para encontrar a solução ótima em cada instância. A tabela apresenta também o tempo médio (em segundos) necessário para obter a solução ótima. Por último, é apresentado o tempo médio total gasto para executar as 2.000 gerações para cada instância.

**Tabela 1. Resultados obtidos**

Número de Nós	Fitness da Sol. Ótima	Gerações até a Sol. Ótima	Tempo até a Sol. Ótima (s)	Tempo Total Gasto (s)
25 Nós	25	122	5,5	65,3
49 Nós	41	314,5	25,6	124,1
225 Nós	105	596,7	244,7	667
625 Nós	185	592,9	811	2.196,6

Com o aumento do número de nós sensores da rede, o número médio de gerações e o tempo médio para obter a solução ótima aumentam. Um dos fatores que causa esse comportamento é o fato de que se a RSSF possui mais nós sensores, o tempo de simulação de cada *script* é maior, com isso o tempo para avaliar cada indivíduo aumenta. Além disso, como cada instância é uma RSSF diferente, as componentes da função objetivo têm comportamentos diferentes para cada uma. Esse comportamento da função objetivo afeta diretamente a evolução do AG, assim as instâncias com maior número de nós precisam de mais gerações (em média) para obter a solução ótima.

Dentre as 4 instâncias abordadas (25, 49, 225 e 625 nós), após um determinado número de gerações todas as execuções foram capazes de obter a solução ótima. Para a instância com 25 nós, todas as execuções foram capazes de obter a solução ótima com menos de 300 gerações. Para a instância com 49 nós, o ótimo foi obtido por todas as execuções em menos de 800 gerações. Para as instâncias com 225 e 625 nós, foram necessárias, respectivamente, menos de 1200 e de 2000 gerações para obter a solução ótima.

#### 4. Conclusão e Trabalhos Futuros

Este artigo apresenta um *framework* inovador capaz de gerar aplicações automaticamente para RSSFs. Para isso, o projetista da rede precisa somente definir uma função objetivo. Através de programação genética, uma população de programas inicial, gerada aleatoriamente, evolui guiada por essa função. Ao final do processo, o melhor programa encontrado cumpre o objetivo almejado pelo projetista. Este programa é definido em uma linguagem *script* que é interpretada por uma máquina virtual presente no *middleware* proposto.

Experimentos foram realizados para o problema de detecção de eventos com diferente número de nós na rede. Nesse problema, a PG evoluiu um algoritmo que foi capaz de perceber a existência de um evento em um nó arbitrário da rede e após isso, guiar (rotear) o evento até o nó concentrador. Em todos os cenários realizados, a técnica desenvolvida foi capaz de encontrar a solução ótima. Isso mostra que a abordagem desenvolvida é bastante promissora em automatizar a geração de aplicações para RSSFs através de uma função objetivo, retirando do usuário a necessidade de programação dos nós sensores em uma linguagem imperativa.

Outros cenários com diferentes topologias devem ser testados em trabalhos futuros. Além disso, o *framework* será estendido para realizar a melhoria do programa nos nós sensores durante a execução da rede. O objetivo futuro é desenvolver uma ferramenta que seja capaz de tornar uma RSSF quase totalmente autônoma. A ideia é que a RSSF seja auto-adaptativa, sendo capaz de tolerar falhas e contornar problemas com perda de nós na rede.

#### 5. Agradecimentos

Agradecemos à Fundação de Amparo à Pesquisa de Minas Gerais (FAPEMIG) e ao Centro Nacional de Desenvolvimento Científico e Tecnológico (CNPq) pelo apoio e o pelo recurso financeiro para o desenvolvimento do presente trabalho.

#### Referências

- Heimfarth, T., Freitas, E. P., Wagner, F. R., and Larsson, T. (2010). *Handbook of Research on Developments and Trends in Wireless Sensor Networks: From Principle to Practice*, chapter Middleware Support for Wireless Sensor Networks: a Survey. Hershey, Information Science Reference (IGI Global).
- Markham, A. and Trigoni, N. (2011). The automatic evolution of distributed controllers to configure sensor networks operation. *The Computer Journal*, 54(3).
- Weise, T. (2006). Genetic programming for sensor networks. Technical report, University of Kassel, University of Kassel.