

Balanceamento de Carga em Redes de Sensores Sem Fio baseado em Sistemas Multiagentes: DSA vs. LA-DCOP vs. Swarm-GAP

Paulo R. Ferreira Jr.¹, Alexandre Lemke¹, Marcelo Giesel¹,
Paulo A. Afonso¹ e Lisane B. Brisolara¹

¹Centro de Desenvolvimento Tecnológico
Universidade Federal de Pelotas (UFPEL)
Pelotas – RS – Brasil

{paulo, alemke, mgiesel, paaafonso, lisane}@inf.ufpel.edu.br

Abstract. *Several approaches have been applied in order to extend the useful life time of Wireless Sensor Networks since batteries limit its real application. One of these approaches is based on balancing the load of services ran by the network among its nodes. This paper evaluates the applicability of novel task-assignment algorithms, proposed by Multiagent Systems community, to achieve that load balancing. We compared DSA, LA-DCOP, Swarm-GAP, and the network without load balancing. Our results show that, despite the high battery consumption due to agents' communication overheads, the general efficiency of the network stills the same as not using load balancing. Thus, the use of these algorithms is promising, but for its greater efficiency is mandatory a lower communication cost.*

Resumo. *Várias técnicas têm sido empregadas para buscar a extensão da vida útil de uma Rede de Sensores Sem Fio dado que a alimentação por baterias limita sua aplicabilidade. Uma das formas de conseguir isso é o emprego do balanceamento de carga dos serviços a serem realizados pela rede entre os nodos que a compõe. Este artigo avalia o emprego de algoritmos de alocação de tarefas em Sistemas Multiagentes, apresentados na literatura recente, para obter o balanceamento de carga mencionado. Foram experimentados os algoritmos DSA, LA-DCOP e Swarm-GAP, sempre comparados com a não utilização de balanceamento. Os resultados obtidos mostraram que, apesar de um alto consumo de bateria para a comunicação entre os agentes, a eficiência da rede permaneceu igual a não utilização do balanceamento de carga. Assim, o emprego dos algoritmos experimentados é promissor, mas para sua maior eficiência é preciso que o custo de comunicação seja bastante reduzido.*

1. Introdução

O projeto e aplicação de Redes de Sensores Sem Fio (RSSF) sofrem restrições dada a limitação no consumo de energia dos nodos que compõem tais redes, os quais são alimentados por baterias. Pesquisas tem buscado metodologias para aumentar a vida útil destas redes, diminuindo o consumo de energia dos nodos, sem perder a qualidade do serviço de sensoriamento prestado por eles. Permitir que os nodos comportem-se de maneira adaptativa para coordenarem-se na execução dos serviços de forma autônoma e *on*

line, de acordo com a dinâmica do ambiente, pode contribuir para o aumento da vida útil da rede.

Uma das abordagens estudadas nessa linha é o balanceamento de carga entre os nodos sensores. Nesta abordagem, um serviço é visto como um conjunto de tarefas com relações de dependência entre elas. As tarefas de um serviço devem ser distribuídas entre os nodos da rede de forma a evitar que uma mesma tarefa seja feita por mais de um nodo e a buscar que as tarefas percebidas sejam distribuídas uniformemente entre estes nodos. Estas estratégias objetivam mapear as tarefas de um serviço aos nodos da rede de maneira que os serviços de sensoriamento tenham sua carga balanceada entre os nodos.

As RSSF têm muitas semelhanças com os SMA (Sistemas Multiagentes). Pode-se enxergar um nodo como um agente e o conjunto de agentes organizados em um sistema como uma rede de nodos. Os agentes, tais quais os nodos, precisam fazer escolhas quanto a realização de tarefas na busca dos objetivos do sistema. Vários trabalhos já se valeram desta semelhança para utilizar técnicas empregadas na solução de problemas em SMA na solução de problemas semelhantes em RSSF [Lesser et al. 2003].

Considerando-se tais estudos, que se valem das semelhanças entre SMA e RSSF para o desenvolvimento de metodologias e técnicas aplicáveis a problemas encontrados em RSSF, acredita-se que seja relevante o estudo do emprego das técnicas de alocação de tarefas mais recentemente propostas na área de SMA para o balanceamento de carga em RSSF.

O Extended Generalized Assignment Problem (E-GAP) [Scerri et al. 2005] vem sendo utilizado para modelar muitos problemas de alocação de tarefas em SMA. O objetivo nesse problema é realizar alocações que maximizem o somatório da eficiência dos agentes em cada instante. A eficiência é a medida da competência do agente de realizar a tarefa que ele está realizando no momento.

Vários algoritmos foram propostos recentemente para a solução deste problema e permitir que um SMA, de maneira distribuída, faça a alocação de tarefas de forma a maximizar o uso dos recursos dos agentes. Neste artigo serão analisados os algoritmos DSA [Zhang and Wittenburg 2002], LA-DCOP [Scerri et al. 2005] e Swarm-GAP [Ferreira Jr. et al. 2007].

Neste artigo, um cenário de balanceamento de carga em Redes de Sensores Sem Fio é modelado como um E-GAP. Os algoritmos mencionados no parágrafo anterior para a solução do E-GAP são executadas em um ambiente de simulação de SMA para que se possa comparar o desempenho de cada um deles.

Os resultados obtidos nos experimentos realizados mostram que os algoritmos de SMA, por dependerem de uma grande quantidade de comunicação, demandaram significativamente mais bateria que a não utilização de nenhuma técnica de balanceamento de carga. No entanto, o LA-DCOP e o Swarm-GAP conseguiram, mesmo consumindo cerca de 10 vezes mais energia, sensorear o mesmo número de eventos. O DSA também consumiu cerca de 10 vezes mais energia mas não foi capaz de se igualar ao número de eventos sensoreados.

Este artigo está organizado da seguinte forma: na Seção 2 são discutidos os trabalhos relacionados na área de RSSF, abordando a forma como a área tem tratado o pro-

blema de balanceamento de carga através da alocação de tarefas; na Seção 3 são apresentadas a definição formal do E-GAP e uma breve descrição de como funcionam os algoritmos DSA, LA-DCOP e Swarm-GAP; a Seção 4 apresenta o cenário de RSSF onde os algoritmos foram experimentados; na Seção 5 são discutidos os resultados obtidos nos experimentos; e, finalmente, na Seção 6 são apresentados as conclusões e direções futuras deste trabalho.

2. Trabalhos Relacionados

O mapeamento entre tarefas de um serviço em RSSF e os nodos da rede que vão realizar este serviço pode ser feito de forma estática [Shivle et al. 2006, Braun et al. 2002, Martinovic et al. 2003], em tempo de projeto da rede, ou de forma dinâmica [Pathak and Prasanna 2010, Zeng et al. 2008], em tempo de execução. Este mapeamento define quais dos nodos vão lidar com as tarefas que compõe o serviço percebido pela rede.

Técnicas de mapeamento estáticas usualmente enfrentam o problema de serem incapazes de lidar com ambientes fortemente dinâmicos em que RSSF podem ser aplicadas. Considerando-se redes heterogêneas como no caso dos trabalhos citados a pouco, este mapeamento também precisa lidar com as diferentes capacidades dos nodos em executar diferentes tarefas.

Além disso, a decisão sobre o mapeamento que será estabelecido pode ser tomada de forma centralizada [AbdelSalam and Olariu 2011], por algum nodo específico que se ponha na posição de coordenador [Caliskanelli et al. 2013], ou por todos os nodos, de maneira distribuída [Uney and Cetin 2007].

As diversas técnicas dinâmicas para o balanceamento de carga enfrentam vários problemas. Um dos mais relevantes destes é a grande necessidade de comunicação entre os nodos para que haja a coordenação sobre o a mapeamento que será realizado. Sabe-se que a comunicação consome significativamente a energia de um nodo sensor.

Os algoritmos de alocação de tarefas em SMA mais atuais nunca foram experimentados para tratar esta questão específica de RSSF. Este trabalho pretende comparar o desempenho dos algoritmos mencionados neste artigo entre si para, em trabalhos futuros, comparar os resultados obtidos com o estado-da-arte na área de RSSF discutidas nesta seção.

3. Fundamentação Teórica

3.1. Alocação de Tarefas

O *Generalized Assignment Problem* (GAP) [Martello and Toth 1990] é um problema geral que trata da alocação de tarefas a agentes, respeitando sua capacidade (recursos que estes têm disponíveis), buscando maximizar uma recompensa total associada às competências destes agentes para tal alocação.

Define-se o GAP como segue. Seja \mathcal{J} o conjunto de tarefas e \mathcal{I} o conjunto de agentes. Cada agente $i \in \mathcal{I}$ tem capacidade r_i associada a uma quantidade limitada de recursos. Um único tipo de recurso é considerado. Quando uma tarefa $j \in \mathcal{J}$ é alocada por um agente i , esta consome u_{ij} unidades do recurso deste agente. Cada agente tem uma competência para alocar cada tarefa j dada por k_{ij} ($0 \leq k_{ij} \leq 1$).

A matriz de alocação A , onde a_{ij} é o valor da i -ésima linha e a j -ésima coluna, é dada pela equação 1.

$$a_{ij} = \begin{cases} 1 & \text{se a tarefa } j \text{ está alocada pelo agente } i \\ 0 & \text{caso contrário} \end{cases} \quad (1)$$

O objetivo do GAP é encontrar a matriz A que maximiza a recompensa do sistema dada pela equação 2, respeitando as limitações de recursos e a limitação de ter apenas um agente alocado a cada tarefa.

$$A = \underset{A'}{\operatorname{argmax}} \sum_{i \in \mathcal{I}} \sum_{j \in \mathcal{J}} k_{ij} \times a'_{ij} \quad (2)$$

tal que

$$\forall i \in \mathcal{I}, \sum_{j \in \mathcal{J}} u_{ij} \times a_{ij} \leq r_i \quad (3)$$

e

$$\forall j \in \mathcal{J}, \sum_{i \in \mathcal{I}} a_{ij} \leq 1 \quad (4)$$

Uma extensão para o GAP, para lidar com problemas dinâmicos e com a interrelação entre tarefas, foi proposta e denominada *Extended GAP* (E-GAP) [Scerri et al. 2005]. O E-GAP estende o GAP de duas maneiras diferentes, as quais são:

Interrelacionamentos entre tarefas. As tarefas no E-GAP podem ser interrelacionadas por restrições do tipo AND. Todas as tarefas interrelacionadas por esta restrição devem ser alocadas simultaneamente para serem consideradas no cálculo da recompensa. Esta é uma extensão importante porque o GAP, por definição, restringe a alocação de cada tarefa a um agente apenas. Quando uma tarefa requer mais recursos que um único agente pode oferecer, esta tarefa deve ser decomposta em subtarefas interrelacionadas pela restrição AND. Define-se $\bowtie = \{\alpha_1, \dots, \alpha_k\}$, onde $\alpha_k = \{j_{k_1}, \dots, j_{k_q}\}$ denota o k -ésimo conjunto de tarefas interrelacionadas pela restrição AND. Assim, a recompensa local w_{ij} para a alocação da tarefa j ao agente i é dada pela equação 5.

$$w_{ij} = \begin{cases} k_{ij} \times a_{ij} & \text{if } \forall \alpha_k \in \bowtie, j \notin \alpha_k \\ k_{ij} \times a_{ij} & \text{if } \exists \alpha_k \in \bowtie \text{ with } j \in \alpha_k \wedge \\ & \forall j_{k_q} \in \alpha_k, a_{xj_{k_q}} \neq 0 \text{ with } x \in \mathcal{I} \\ 0 & \text{caso contrário} \end{cases} \quad (5)$$

Recompensa determinada dinamicamente. A recompensa total W é calculada no E-GAP como o somatório das recompensas parciais w_{ij}^t obtidas em cada intervalo de tempo t . Nesse caso, o que está sendo considerado é a recompensa resultante de uma seqüência de alocações, diferentemente do GAP que considera uma única alocação. Com isso, todos os conjuntos e parâmetros da equação 2 são indexadas pelo tempo. Além disso, um custo v_j pode ser usado para punir os agentes

quando a tarefa j não é alocada no tempo t . O objetivo do E-GAP é maximizar a recompensa total W dada pela equação 6.

$$W = \sum_t \sum_{i^t \in \mathcal{I}^t} \sum_{j^t \in \mathcal{J}^t} w_{ij}^t \times a_{ij}^t - \sum_t \sum_{j^t \in \mathcal{J}^t} (1 - \sum_{i^t \in \mathcal{I}^t} a_{ij}^t) \times v_j^t \quad (6)$$

tal que

$$\forall t \forall i^t \in \mathcal{I}^t, \sum_{j^t \in \mathcal{J}^t} u_{ij}^t \times a_{ij}^t \leq r_i^t \quad (7)$$

e

$$\forall t \forall j^t \in \mathcal{J}^t, \sum_{i^t \in \mathcal{I}^t} a_{ij}^t \leq 1 \quad (8)$$

Os agentes devem determinar quais tarefas alocar de maneira que suas competências associadas a estas tarefas sejam as maiores possíveis. A quantidade de tarefas que os agentes podem alocar simultaneamente é restrita pela quantidade de recursos que estes têm disponível em relação a quantidade demandada pela tarefas a serem alocadas. A duração d_j destas tarefas é irrelevante para a tomada de decisão pois os agentes precisam definir quais alocar a cada unidade de tempo.

As competências associadas aos agentes e as tarefas por eles alocadas determinam a recompensa do sistema como um todo. As recompensas parciais são obtidas em cada unidade de tempo e a total é o somatório das recompensas parciais durante um intervalo de tempo.

3.2. Algoritmos para Alocação de Tarefas

Os autores que definiram o E-GAP propuseram um algoritmo, denominado LA-DCOP (*Low-communication Approximation DCOP*) [Scerri et al. 2005], para tratar este problema. O LA-DCOP utiliza um protocolo baseado em *tokens* para aumentar seu desempenho quanto à comunicação. Neste algoritmo os agentes podem perceber tarefas no ambiente ou receber *tokens* de outros agentes.

Cada um dos *tokens* recebidos também contém tarefas, uma cada um. Os agentes decidem ou não alocar uma tarefa baseados em um limiar global, tentando maximizar o uso de seus recursos. Depois de decidir quais tarefas alocar, os agentes enviam *tokens* com tarefas não alocadas para outros agentes.

O limiar mencionado representa a capacidade dos agentes para alocar cada tarefa e pode ser fixo ou dinamicamente calculado. *Tokens* adicionais, chamados “*tokens* potenciais”, são utilizados pelos agentes para estabelecer compromissos com relação à alocação de algumas tarefas para lidar com a interrelação de alocação simultânea (AND) que podem existir entre elas.

Os autores do LA-DCOP argumentam que o seu algoritmo é melhor que abordagens anteriormente propostas, comparando-o diretamente com o DSA [Zhang and Wittenburg 2002].

O DSA utiliza uma estratégia baseada em *hill-climbing* para permitir que os agentes, com base nas informações de outros agentes próximos, aloquem as tarefas de modo a maximizar a solução do problema.

Mais recentemente um novo algoritmo, denominado Swarm-GAP [Ferreira Jr. et al. 2007] foi proposto para lidar com o E-GAP. Este algoritmo foi concebido inspirado na forma como as colônias de insetos sociais (*swarms*) dividem o trabalho [Bonabeau et al. 1999].

O modelo de divisão de trabalho dos insetos sociais é baseado em um modelo que usa o estímulo produzido pelas tarefas e um limiar interno específico de cada indivíduo associado a cada tarefa para determinar a tendência de cada indivíduo de realizar cada tarefa [Theraulaz et al. 1998]. Os agentes utilizando o Swarm-GAP decidem quais tarefas realizar baseado neste mecanismo. O Swarm-GAP também utiliza um coeficiente para modificar a tendência do agente em realizar uma tarefa de acordo com a alocação de tarefas interrelacionadas (AND).

Segundo seus autores, o Swarm-GAP tem vantagens em relação à complexidade de tempo se comparado com o LA-DCOP pois seu processo de tomada de decisão é mais eficiente computacionalmente. Por outro lado, os resultados apresentados [Boffo et al. 2007, Ferreira Jr. et al. 2008, Ferreira et al. 2010] mostram algumas vantagens no emprego do LA-DCOP quanto a maximização da solução obtida para o E-GAP em alguns cenários.

4. Descrição do Cenário

O problema em RSSF que este artigo trata, como comentado na Seção 1, diz respeito ao balanceamento de carga visando aumentar a vida útil da rede, a qual é alimentada por baterias, sem a perda de eficiência quanto ao sensoriamento de eventos. Com isso, o cenário abaixo foi elaborado de maneira que fosse possível realizar experimentos com este problema.

A RSSF é composta de um conjunto N de nodos distribuídos aleatoriamente em um espaço 2D toroidal. Cada nodo tem um raio de sensoriamento R igual. Os nodos prestam serviços à uma aplicação que basicamente são a coleta de informações deste espaço, a partir da percepção de um evento, e a realização de alguns cálculos com essa informação.

Os serviços prestados pelos nodos são divididos em tarefas (organização hierárquica), as quais podem ser de sensoriamento ou processamento, definidas de forma abstrata e consumindo a mesma quantidade de energia. Três diferentes serviços são prestados por esta rede, S_a , S_b e S_c .

Para a realização do serviço S_a , o nodo precisa realizar as tarefas a_1 e a_2 . A tarefa a_1 é dividida em subtarefas $a_{1.1}$ e $a_{1.2}$ que precisam ser realizadas primeiro que sua supertarefa para habilitar a realização desta. A tarefa a_2 é independente. Da mesma forma, o serviço S_b é composto pelas tarefas b_1 , b_2 e b_3 , sem subtarefas e o serviço S_c é composto pelas tarefas c_1 , c_2 e c_3 , tendo a tarefa c_1 subdividida em $c_{1.1}$, $c_{1.2}$, $c_{1.3}$ e $c_{1.4}$. Um nodo pode realizar todo o serviço (todas as subtarefas) ou delegar a realização de todas ou algumas subtarefas para outros nodos.

Um nodo que perceba um evento, o qual demanda a realização de um serviço, é responsável por sua realização. Os eventos que demandam a realização dos serviços ocorrem aleatoriamente com uma probabilidade igual $P(S)$. Um nodo percebe apenas um evento por vez e um evento pode ser percebido por vários nodos.

Os nodos da rede são numerados e interligados em uma rede de comunicação *ad hoc* por proximidade física. Um nodo se liga com todos os nodos que estiverem a um raio D de distância. Apenas o nodo 0 ($id = 0$) se comunica com a aplicação, ou seja, os nodos precisam enviar uma mensagem para o nodo mais perto de si e que esteja mais perto do nodo 0, e assim por diante. O nodo zero não está conectado a baterias e não reage a eventos de sensoriamento. Além disso, este nodo fica no centro do espaço a ser sensorado. Essa topologia de comunicação é fixa e definida na inicialização do sistema. Todos os nodos conhecem esta topologia e sabem para quem enviar uma mensagem para que ela chegue na aplicação.

Cada nodo tem uma quantidade de bateria inicial de I unidades. A realização de cada tarefa j consome uma quantidade de energia Q_j . A comunicação da informação referente a uma tarefa j entre os nodos consome uma quantidade de energia $C_j \leq Q_j$. Os serviços devem ser prestados pelos nodos da rede de forma a minimizar o consumo de energia do sistema com um todo e, com isso, aumentar a vida útil do sistema.

Este cenário de RSSF pode ser traduzido facilmente em um E-GAP. As tarefas que compõe os serviços podem ser vistas como as tarefas no E-GAP. Os nodos, que precisam decidir a alocação destas tarefas, são equivalentes aos agentes no E-GAP. Como os nodos são limitados apenas pela sua bateria para realizar as tarefas, pode-se considerar que não existe limitação de recursos para os agentes. Todas as tarefas que compõe um serviço são interrelacionadas (AND) no E-GAP e só são consideradas para o cálculo da recompensa se todas forem alocadas em cada instante.

O objetivo do balanceamento de carga em RSSF é garantir que todos os nodos atuem de forma equânime, ou seja, manter cada nodo o maior tempo possível com a mesma quantidade de bateria dos demais. Com isso, a competência do agente para alocar a tarefa pode ser dado diretamente pela quantidade de bateria do nodo em cada instante. A maximização da recompensa no E-GAP implicará na alocação das tarefas para os agentes com maior quantidade de bateria em cada instante.

5. Experimentos e Resultados

O cenário apresentado na Seção 4 foi implementado na ferramenta NetLogo e foi baseado no cenário utilizado em um recente trabalho relacionado na área de RSSF [Caliskanelli et al. 2013]. Foram feitas algumas adaptações com relação ao cenário referenciado uma vez que no artigo os autores utilizam um simulador de RSSF, que trata diversos detalhes da rede. Os valores iniciais de bateria e seu consumo a cada iteração, por exemplo, foram escolhidos para manter a proporção destes em relação ao consumo para a realização de tarefas que se pode encontrar no referido trabalho. Em contrapartida, o cenário implementado aqui é bem mais complexo no que tange a topologia da rede.

Foram realizadas 30 execuções com cada cada algoritmo. Cada execução termina quando nenhum nodo consegue mais se comunicar com a aplicação hipotética, ou seja, não possui um caminho de comunicação com o nodo especial que serve de *gateway* da rede toda.

Foram considerados como métrica de qualidade de cada algoritmo a quantidade total de eventos sensorados e a média de bateria dos nodos da rede em cada execução. Foram computadas a média e o desvio padrão das 30 execuções destas duas métricas.

Testes t de student, com confiança de 0.95, foram usados para comparar o desempenho dos algoritmos segundo tais médias.

As simulações foram realizadas com 30 nodos distribuídos aleatoriamente pelo espaço 2d toroidal padrão do NetLogo. Os eventos são objetos que se movem aleatoriamente pelo cenário e são percebidos pelos nodos quando se encontram a uma distância 10 do nodo. Cada evento demanda um e somente um dos serviços descritos na Seção 4 durante toda a simulação e a distribuição inicial de serviços é aleatória.

Os nodos que ficaram distantes menos de 10 unidades, quando estes são distribuídos no espaço, são conectados por um canal de comunicação. Um nodo especial, no centro do espaço e com bateria infinita, é utilizado como *gateway* da rede para a comunicação com uma aplicação hipotética. A comunicação dos nodos com a aplicação se dá utilizando *gateways* determinados de forma *ad-hoc*, com cada nodo utilizando como *gateway* o nodo mais próximo do nodo especial (*gateway* da rede com a aplicação) segundo a distância euclidiana. Todos os nodos se comunicam diretamente com seus vizinhos.

Cada nodo inicialmente possui 54000000 unidades de bateria, consumindo 300 por passo de simulação independentemente do que estiverem fazendo. Quando um nodo processa uma tarefa ele consome 6 unidades de bateria e quando este envia uma mensagem são consumidas 2 unidades de bateria.

O simulador busca evitar que mais de um nodo sensive um evento ao mesmo tempo. Isso é feito fazendo com que cada nodo verifique se algum de seus vizinhos percebeu o evento antes de decidir por sensiveá-lo. O custo desta comunicação não foi considerado.

Além do DSA, LA-DCOP e Swarm-GAP, nos resultados discutidos a seguir, é considerada uma série de execuções sem que nenhuma técnica de balanceamento de carga seja utilizada. Essa abordagem foi denominada “SemBal” nas tabelas que seguem.

Dada uma limitação da implementação atual dos algoritmos no NetLogo, os eventos foram tratados de forma atômica. Todas as tarefas de um evento são tratadas pelo mesmo agente. O balanceamento então é feito através da alocação das tarefas que compõem o serviço entre os nodos vizinhos ao nodo que decidiu sensivear determinado evento que demanda tal serviço.

Várias simulações com os diferentes parâmetros do DSA foram realizadas para determinar empiricamente a melhor combinação destes para o problema em questão. Com isso, o limiar de probabilidade p para aceitação de um novo Δ foi fixado em $p = 0.2$ e foram adotadas 5 rodadas para a parada do algoritmo. A utilidade U foi computada como sendo $U = e^{(\frac{Q_i}{I}) * 10}$, onde Q_i é a quantidade atual de bateria do nodo e I a quantidade inicial de bateria dos nodos.

O LA-DCOP e o Swarm-GAP também foram configurados empiricamente. Em ambos os casos, o limiar interno do agente i para a realização de uma determinada tarefa j é calculado como sendo $\theta_{ij} = 1 - \frac{\sum_{j \in S} Q_j}{Q_i}$, onde Q_j é a quantidade de bateria demandada pelas tarefas que compõem o serviço completo e Q_i é o total de bateria do nodo. No LA-DCOP o limiar foi fixado em 0.6 e no Swarm-GAP o estímulo S_j é dado pela quantidade de bateria que as tarefas consomem Q_j .

Como mostrado na Tabela 1, a não utilização de técnicas de balanceamento de carga necessariamente implica em uma quantidade média maior de bateria disponível todo o tempo de cada execução. A média da quantidade de bateria disponível utilizando os algoritmos de SMA foi cerca de 10 vezes menor em média nos experimentos realizados. Esse comportamento era esperado uma vez que a alocação de tarefas em SMA necessariamente é realizada através de muitas trocas de mensagens.

Tabela 1. Média de bateria dos nodos da rede utilizando os algoritmos experimentados.

	SemBal	DSA	LA-DCOP	Swarm-GAP
Média	783057351,5	65971973,5	73640156,7	77173282,6
Desvio Padrão	23248782,6	173921138,3	233104604,9	323160163,4

No entanto, mesmo com um consumo de bateria bem maior, os algoritmos LA-DCOP e Swarm-GAP foram capazes de sensoriar a mesma quantidade de eventos que a não utilização de técnicas de balanceamento, como mostra a Tabela 2. Apenas o DSA que apresentou diferença significativa em relação aos eventos sensorizados, sendo menos eficiente cerca de 6% menos eficiente nos experimentos realizados.

Tabela 2. Número total de eventos sensorizados utilizando os algoritmos experimentados.

	SemBal	DSA	LA-DCOP	Swarm-GAP
Média	801757,6	754783,1	779909,6	789694,1
Desvio Padrão	29884,9	70130,4	99945,9	33409,60381

Levando em conta os resultados obtidos, é promissora a aplicação de algoritmos de alocação de tarefas em SMA para o balanceamento de carga em RSSF. Contudo, o consumo de bateria para a comunicação precisa ser bem menor se comparado ao consumo de bateria pelo processamento das tarefas para que o emprego de tais algoritmos tenha um efeito positivo real em RSSF. Nos experimentos realizados o consumo de bateria para a comunicação é apenas $\frac{1}{3}$ do consumo de bateria do processamento das tarefas.

6. Conclusões

Este artigo apresentou um estudo sobre a aplicação de algoritmos para a alocação de tarefas em SMA no problema de balanceamento de carga em RSSF. O objetivo foi de verificar se tais algoritmos podem contribuir para a construção de RSSF com maior vida útil, sem que sua eficiência quanto ao sensoriamento seja prejudicada.

Para tanto, um cenário de RSSF foi especificado e implementado em um simulador de SMA. Os algoritmos DSA, LA-DCOP e Swarm-GAP foram implementados no simulador para lidar com tal cenário. As métricas adotadas foram o total de eventos sensorizados pela rede e a média de bateria presente na rede toda durante cada execução.

Os resultados mostraram que, apesar de consumir uma quantidade de bateria significativamente maior dada a necessidade de estabelecer muita comunicação, os algoritmos LA-DCOP e Swarm-GAP foram capazes de sensoriar a mesma quantidade de eventos que a rede sem o uso de balanceamento de carga. Isso mostra que estes algoritmos podem

ser úteis se o consumo de bateria para a comunicação for significativamente mais baixo que o consumo de bateria do processamento dos serviços.

Como trabalhos futuros pretende-se realizar experimentos com um custo de comunicação menor que o utilizado neste artigo, bem como implementar e experimentar o estado-da-arte na área de balanceamento de carga em RSSF para comparar com o desempenho dos algoritmos utilizados neste artigo.

Referências

- AbdelSalam, H. and Olariu, S. (2011). Toward efficient task management in wireless sensor networks. *Computers, IEEE Transactions on*, 60(11):1638–1651.
- Boffo, F., Jr., P. R. F., and Bazzan, A. L. C. (2007). A comparison of algorithms for task allocation in robocup rescue. In *Proceedings of the 5th European Workshop on Multiagent Systems*. To appear.
- Bonabeau, E., Theraulaz, G., and Dorigo, M. (1999). *Swarm Intelligence: From Natural to Artificial Systems*. Oxford University Press, New York, USA.
- Braun, T., Siegel, H., and Maciejewski, A. (2002). Static mapping heuristics for tasks with dependencies, priorities, deadlines, and multiple versions in heterogeneous environments. In *Parallel and Distributed Processing Symposium., Proceedings International, IPDPS 2002, Abstracts and CD-ROM*, pages 8 pp–.
- Caliskanelli, I., Harbin, J., Indrusiak, L. S., Mitchell, P., Polack, F., and Chesmore, D. (2013). Bioinspired load balancing in large-scale wsns using pheromone signalling. *International Journal of Distributed Sensor Networks*, 2013.
- Ferreira, Jr., P. R., dos Santos, F., Bazzan, A. L. C., Epstein, D., and Waskow, S. J. (2010). Robocup rescue as multiagent task allocation among teams: experiments with task interdependencies. 20(3):421–443.
- Ferreira Jr., P. R., Boffo, F., and Bazzan, A. L. C. (2007). A swarm based approximated algorithm to the extended generalized assignment problem (E-GAP). In *Proceedings of the 6th International Joint Conference on Autonomous Agents And Multiagent Systems (AAMAS)*, pages 1231–1233.
- Ferreira Jr., P. R., Boffo, F., and Bazzan, A. L. C. (2008). Using Swarm-GAP for distributed task allocation in complex scenarios. In Jamali, N., Scerri, P., and Sugawara, T., editors, *Massively Multiagent Systems*, Lecture Notes in Artificial Intelligence. Springer Verlag, Berlin. To appear.
- Lesser, V., Ortiz, C. L., and Tambe, M. (2003). *Distributed sensor networks: A multiagent perspective*, volume 9. Springer.
- Martello, S. and Toth, P. (1990). *Knapsack problems: algorithms and computer implementations*. John Wiley & Sons, Inc.
- Martinovic, G., Budin, L., and Hocenski, Z. (2003). Static-dynamic mapping in heterogeneous computing environment. In *Virtual Environments, Human-Computer Interfaces and Measurement Systems, 2003. VECIMS '03. 2003 IEEE International Symposium on*, pages 32–37.

- Nwana, H. S., Lee, L. C., and Jennings, N. R. (1996). Coordination in software agent systems. *The British Telecom Technical Journal*, 14(4):79–88.
- Pathak, A. and Prasanna, V. (2010). Energy-efficient task mapping for data-driven sensor network macroprogramming. *Computers, IEEE Transactions on*, 59(7):955–968.
- Scerri, P., Farinelli, A., Okamoto, S., and Tambe, M. (2005). Allocating tasks in extreme teams. In *Proceedings of the 4th International Joint Conference on Autonomous Agents and Multiagent Systems, AAMAS*, pages 727–734, New York. ACM Press.
- Shivle, S., Siegel, H. J., Maciejewski, A. A., Sugavanam, P., Banka, T., Castain, R., Chindam, K., Dussinger, S., Pichumani, P., Satyasekaran, P., Saylor, W., Sendek, D., Sousa, J., Sridharan, J., and Velazco, J. (2006). Static allocation of resources to communicating subtasks in a heterogeneous ad hoc grid environment. *J. Parallel Distrib. Comput.*, 66(4):600–611.
- Theraulaz, G., Bonabeau, E., and Deneubourg, J. (1998). Response threshold reinforcement and division of labour in insect societies. *Royal Society of London Series B - Biological Sciences*, 265:327–332.
- Uney, M. and Cetin, M. (2007). Graphical model-based approaches to target tracking in sensor networks: An overview of some recent work and challenges. In *Image and Signal Processing and Analysis, 2007. ISPA 2007. 5th International Symposium on*, pages 492–497.
- Zeng, Z., Liu, A., Li, D., and Long, J. (2008). A highly efficient dag task scheduling algorithm for wireless sensor networks. In *Young Computer Scientists, 2008. ICYCS 2008. The 9th International Conference for*, pages 570–575.
- Zhang, W. and Wittenburg, L. (2002). Distributed breakout revisited. In *Proceedings of the 18th National Conference on Artificial Intelligence*, pages 352–357. Menlo Park, CA: American Association for Artificial Intelligence.