

Resource Scheduling for Mobility Scenarios with Time Constraints

Marcelo Petri
Programa de Pós-Graduação
em Computação Aplicada
Universidade do Estado
de Santa Catarina - UDESC
Joinville, Brasil
Email: marcelo.petri@gmail.com

Janine Kniess
Programa de Pós-Graduação
em Computação Aplicada
Universidade do Estado
de Santa Catarina - UDESC
Joinville, Brasil
Email: jakniess@gmail.com

Rafael Stubs Parpinelli
Programa de Pós-Graduação
em Computação Aplicada
Universidade do Estado
de Santa Catarina - UDESC
Joinville, Brasil
Email: rafael.parpinelli@udesc.br

Abstract—This paper presents an approach for resource scheduling in service discovery to MANETs operating in scenarios with time constraints, as emergency scenarios. The shared resources can be ambulances or rescue cars. With an efficient model of resource scaling, it is intended to provide the largest number of victims in the shortest time. In this work, we present an approach for resource scheduling modeled based on two concepts: Genetic Algorithm and A-Star Algorithm. The results obtained from the *Network Simulator (NS3)* show that the resource scheduling mechanism is efficient in response time and scalable in relation to different numbers of victims.

Index Terms—Protocol, Discovery Services, Scheduling, Resources, MANET, Genetic, A-Star, Multi-Hop, Ad Hoc.

I. INTRODUÇÃO

Em cenários pós-desastre a busca pelas equipes de resgate pelas vítimas (p.ex. carros de apoio) deve ocorrer rapidamente. Um gerenciamento eficiente das equipes de resgate possibilita o atendimento a um maior número de vítimas. Neste cenário, protocolos para descoberta de serviços são essenciais para que as vítimas localizem os recursos necessários para o atendimento.

Contudo, em situações de emergência, a infraestrutura de comunicação pode ser danificada não sendo possível fornecer serviços adequados para a comunicação entre vítimas e socorristas.

As Redes Móveis Ad Hoc (*Mobile Ad Hoc Networks MANETs*), são atrativas para cenários onde exista a necessidade de instalar rapidamente uma rede não infraestruturada e podem ser usadas como tecnologia de comunicação entre as vítimas e socorristas. Com as informações obtidas através do protocolo de descoberta de serviços, os socorristas podem tomar decisões sobre os atendimentos a serem realizados [1].

O número de vítimas em uma situação de emergência pode aumentar consideravelmente de acordo com a gravidade da calamidade. Unidades do corpo de bombeiros e ambulâncias nem sempre estão disponíveis em número suficiente em muitas cidades. A complexidade para gerenciar os recursos em relação ao número de atendimentos requer o uso de algoritmos adequados de escalonamento de recursos.

Um algoritmo de escalonamento neste contexto, deve atender ao requisito de restrição de tempo para o sucesso do

atendimento. Dessa forma, identificou-se que a problemática em questão visa encontrar um caminho de custo mínimo, no qual os socorristas devem percorrer para atender um conjunto de requisitantes dentro de uma janela de tempo.

Neste artigo apresenta-se um algoritmo de escalonamento de recursos para cenários de emergência com restrição de tempo no atendimento as vítimas. O algoritmo de escalonamento estende o protocolo de descoberta de serviço *Location Aware Discovery Protocol (LADP)* [1].

O protocolo LADP foi desenvolvido no contexto de MANETs de múltiplos saltos, formadas por grupos de colaboração temporários (equipes de resgate) e especificado com a finalidade de oferecer suporte ao desenvolvimento de aplicações para descoberta, seleção e invocação de serviço, possibilitando aos dispositivos sem fio, utilizar os recursos disponíveis na rede [1] e [2]. Entretanto, questões de escalonamento de recursos não foram consideradas no protocolo LADP.

O algoritmo de escalonamento proposto foi modelado com base em Algoritmos Genéticos (AG) [3], e para fins de comparação, foi modelado com base na abordagem de busca em grafos A-Estrela (A^*) [4]. O algoritmo de escalonamento leva em consideração a localização geográfica das vítimas e provedores de recursos (e.g., ambulância), a velocidade de deslocamento dos nós provedores e o tempo máximo para o atendimento para determinar o escalonamento dos provedores em relação aos atendimentos.

O artigo está organizado como segue: na Seção II apresenta-se a motivação e os trabalhos relacionados. Na Seção III, a abordagem proposta para o escalonamento de recursos. Os resultados computacionais são discutidos na Seção IV. Na Seção V discorre-se sobre as conclusões e trabalhos futuros.

II. MOTIVAÇÃO E REVISÃO DE LITERATURA

Nas regiões pós-desastres naturais, equipes especializadas em resgate são distribuídas na área afetada. Nesse contexto, conjectura-se que equipes de resgate (formadas por humanos e veículos) interconectadas por uma rede sem fio móvel a exemplo do WiFi 802.11, poderia contribuir na comunicação e no salvamento de sobreviventes [1]. Neste cenário, as vítimas e socorristas transportam um dispositivo para a comunicação

e conhecem sua posição geográfica através de um sistema de localização. Os nós que proveem os serviços são entidades físicas chamadas de provedores e os nós que solicitam o serviço são entidades físicas chamadas de requisitantes. Os provedores de serviço são móveis, porém o requisitante permanece fixo a espera de atendimento.

No ambiente descrito, assume-se que partindo de algoritmos especializados, o dispositivo móvel do requisitante deduz que é necessário um provedor capaz de atuar em poucos minutos naquela área. Após a identificação do tipo de recurso, o protocolo LADP [1] (entidade lógica no cenário), estabelece que o dispositivo móvel envie para a rede uma mensagem de descoberta de serviço buscando o provedor de serviço apropriado para o atendimento.

A. Protocolos para Descoberta de Serviços em Redes Móveis

De forma geral, os protocolos de descoberta de serviços viabilizam a detecção automática de dispositivos e de serviços oferecidos em uma rede de computadores.

Não obstante, muitas pesquisas sobre protocolos de descoberta de serviços para MANETs de múltiplos saltos tenham sido realizadas como por exemplo, RESCUE [5], LADP [1], Gadallah e Serhani [6], e BASHD [7], não identificou-se na literatura protocolos para descoberta de serviços que apresentem soluções para o escalonamento de provedores móveis em cenários pós-desastres com restrição de tempo para o atendimento e que levem em consideração os requisitos do cenário proposto: localização geográfica dos provedores e vítimas, velocidade de deslocamento do provedor de serviço e tempo máximo para o atendimento.

BeeAdHocServiceDiscovery (BAHSD) [7] é um protocolo de descoberta de serviço baseado no conceito de *Bee Colony Optization* (BCO) [8]. O algoritmo BAHSD utiliza uma abordagem *cross-layer* que integra o algoritmo proposto com a camada de rede. Especificamente, o algoritmo de descoberta de serviço opera com o algoritmo de roteamento *BeeAdHoc* [9]. Os clientes ao receberem as informações sobre os provedores aptos enviam uma invocação de serviço para estes provedores. Os autores não apresentam estratégias para o escalonamento de recursos. Além disso, o BAHSD não atende aos requisitos relacionados a descoberta de serviço em situações de emergência.

Gadallah and Serhani [6] apresentam um protocolo de descoberta de serviços para MANETs em cenários de desastre. Os provedores de serviços enviam anúncios para uma central que realiza a seleção dos provedores para os clientes. O protocolo permite localizar os serviços com base nos requisitos especificados pelo cliente. O protocolo faz a divisão da rede em diferentes zonas geográficas. Um solicitante que necessita de tratamento é delimitado a uma determinada zona. O protocolo disponibiliza os mecanismos de descoberta de serviço e invocação de serviço. O escalonamento dos recursos não foi abordado neste trabalho.

O RESCUE [5], é um *middleware* de código aberto para comunicação orientada a serviços em ambientes pós-desastres, projetado para MANETs. O RESCUE oferece mecanismos

para a descoberta e invocação de serviços. Os nós anunciam seus serviços e escutam os anúncios de outros nós. Informações sobre o serviços locais ou remotos são mantidas em um banco de dados local que contém metadados publicados (descrições) sobre todos os serviços. Os metadados consistem em registros obrigatórios que são necessários para identificar e invocar os serviços. Diferente da abordagem proposta neste artigo, o escalonamento dos recursos não foi investigado.

O protocolo de descoberta de serviços *Location Aware Discovery Protocol* (LADP) [1] foi desenvolvido para atender aos requisitos de sistemas para ambientes pós-desastres: localização geográfica dos provedores e vítimas, velocidade de deslocamento do provedor de serviço e tempo máximo para o atendimento. Com base nas características do LADP, este protocolo foi selecionado como o protocolo de descoberta de serviço neste trabalho.

No protocolo LADP [1] e [10], a mensagem de descoberta será transmitida para a rede, salto a salto, através de um mecanismo de difusão (*flooding*). A mensagem irá trafegar dentro de um raio R_i delimitado pelo mecanismo de descoberta de serviço. Como cada provedor mantém armazenadas as informações sobre o recurso que disponibiliza, logo qualquer nó fora do raio de alcance, que receba a mensagem de descoberta, irá descartá-la. Na mensagem de descoberta, o nó requisitante (vítima) envia as seguintes informações: uma identificação do nó e da requisição, o número de sequência da requisição, suas coordenadas geográficas, o tempo máximo para atendimento da requisição Δ_{tmax} , o serviço procurado e o número de provedores desejado. Após, um mecanismo de seleção de serviços, executa um filtro seletivo nos nós intermediários durante o encaminhamento da mensagem de resposta na rede e envia as respostas dos provedores mais aptos na direção do nó requisitante. Na fase de invocação de serviços, o requisitante envia uma mensagem de invocação ao (s) provedor (es) selecionado (s). Um provedor ao receber a primeira mensagem de invocação de um requisitante, envia confirmação para o mesmo e inicia o deslocamento na direção deste requisitante. Nenhuma estratégia de escalonamento dos recursos foi proposta. Quando o provedor inicia o deslocamento para realizar o atendimento, ele estará alocado para este requisitante.

A discussão apresentada sobre os protocolos de descoberta de serviços reforça a necessidade de uma estratégia de escalonamento de recursos que contemple as peculiaridades do cenário alvo, na qual os provedores devem se deslocar para o local do evento dentro de um tempo limite visando atender ao maior número de requisitantes. Neste contexto, na Seção III apresenta-se uma solução para o escalonamento de recursos modelada com base em duas abordagens: Algoritmo Genético e Algoritmo A-Estrela.

III. ABORDAGEM PARA ESCALONAMENTO DE RECURSOS

O algoritmo de escalonamento de recursos estende o protocolo LADP [1] e atua na fase de invocação de serviço do protocolo. O desenvolvimento dos algoritmos de escalonamento

de recursos com base nos algoritmos, AG e A* são ilustrados nas Seções, III-B e III-C respectivamente.

A. Algoritmo de Escalonamento de Recursos

No escalonamento de recursos proposto neste trabalho, a entidade responsável pelo agendamento é o nó provedor. Após o provedor ser selecionado conforme estabelece o mecanismo de seleção do protocolo LADP, o requisitante envia uma mensagem de invocação ao provedor que ao recebê-la inicia o temporizador $\Delta_{schedule}$ com o propósito de aguardar mensagens de invocação de outros requisitantes.

Quando o tempo definido em $\Delta_{schedule}$ expirar, o algoritmo irá gerar uma matriz de adjacências (um exemplo pode ser visualizado na Tabela I) contendo informações relacionadas ao tempo de deslocamento do provedor ($serviceTimeProvider$) para cada um dos requisitantes e seus pares adjacentes. Outra informação gerada consiste em um vetor contendo o tempo estipulado pelo requisitante (na mensagem de descoberta) para o atendimento Δ_{tmax} denominado, janelamento.

Tabela I
MATRIZ DE ADJACÊNCIA (TEMPO EM SEGUNDOS)

		Para							
		P	R7	R2	R3	R1	R6	R4	R5
De	P	0	307	257	268	197	154	279	62
	R7		0	63	39	201	230	114	248
	R2			0	41	138	167	139	196
	R3				0	176	199	100	210
	R1					0	47	254	142
	R6						0	264	105
	R4							0	233
	R5								0

Os valores na matriz de tempo de deslocamento correspondem a distância euclidiana entre dois pontos [11], sendo cada ponto uma representação do nó requisitante e seus nós adjacentes. Utilizou-se um sistema de coordenadas cartesianas, considerando que a posição de um nó no plano é especificada através do conjunto de coordenadas X, Y.

Analisando a Figura 1, suponha que os requisitantes R1, R2, R3, R4, R5, R6 e R7 enviem uma mensagem de invocação de serviço para um provedor P solicitando um atendimento, e o tempo máximo Δ_{tmax} é igual a 360s. Após o provedor P receber a primeira mensagem de invocação (p.ex. requisitante R7), o provedor inicia o temporizador $\Delta_{schedule}$. Como resposta, o nó provedor enviará aos requisitantes uma mensagem de confirmação da invocação em *unicast* e aguarda a mensagem de reconhecimento da confirmação. As mensagens de invocação recebidas durante o tempo definido pelo temporizador $\Delta_{schedule}$ são enfileiradas no provedor P.

Cada nó é considerado um possível caminho. Para a formulação da função foram consideradas algumas restrições como por exemplo, a viabilidade do provedor atender ao próximo requisitante, o tempo limite Δ_{tmax} , ou se o provedor continua para o próximo atendimento. Considera-se uma solução válida, o resultado que atenda a todos os requisitantes em Δ_{tmax} . Para os casos que não resulte em uma solução que atenda a

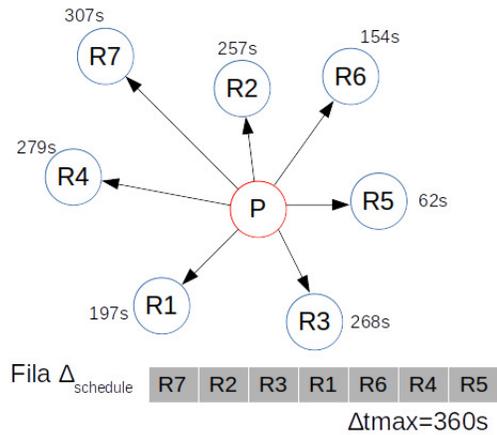


Figura 1. Geração da Fila no Provedor ($\Delta_{schedule}$)

todos os requisitantes, escolhe-se aquele que oferecer o melhor resultado.

A Figura 2 representa uma possível solução de acordo com a matriz de tempo de deslocamento gerada. Na matriz demonstra-se uma trajetória contendo 7 (sete) requisitantes e todos com $\Delta_{tmax} = 360s$. Levando em consideração o tempo de deslocamento do provedor até os requisitantes e o tempo de atendimento estipulado pelos mesmos, o algoritmo de escalonamento encontrou uma solução possível, definindo um trajeto que atenda a 4 (quatro) requisitantes. O provedor P levou 338s para o deslocamento até os requisitantes definidos.

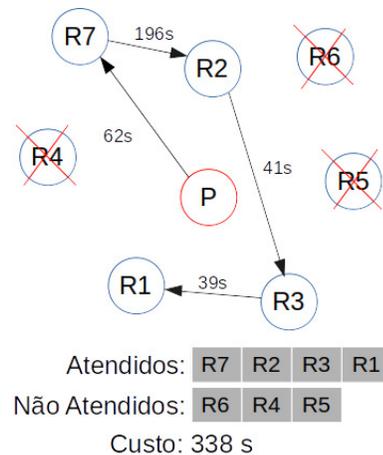


Figura 2. Exemplo de Solução Possível

Em outras palavras, foi possível atender no máximo quatro requisitantes dentro do tempo solicitado sem infringir o limite Δ_{tmax} .

Após definir a ordem de atendimento, o provedor envia uma mensagem de aviso de atendimento (*msgAtendimento*) aos requisitantes e inicia o deslocamento na direção do primeiro da fila. O provedor também envia uma mensagem aos requisitantes que não puderam ser atendidos. Dessa maneira, esses requisitantes reiniciarão o processo de descoberta, a fim de localizar um novo provedor apto para o atendimento.

No protocolo LADP [1] os possíveis casos de falhas durante a execução do protocolo em uma MANET são tratados. Com a adição do algoritmo de escalonamento de recursos, falhas não previstas na versão original do protocolo podem ocorrer.

Por exemplo, suponha que um requisitante envie uma mensagem de reconhecimento da confirmação de serviço (*msgRecConfirmação*) para o provedor. Porém, a mensagem perdeu-se na rede e o nó provedor não a recebeu durante o período do temporizador $\Delta_{schedule}$. Neste caso, o provedor enviará uma mensagem (*NACK* reconhecimento) ao requisitante comunicando o não recebimento da mensagem de reconhecimento da confirmação. Antes do temporizador $\Delta_{schedule}$ expirar, o provedor fará mais duas tentativas de comunicação com o requisitante. Caso receba o reconhecimento da confirmação (*msgRecConfirmação*), o provedor o incluirá novamente no cálculo da fila para redefinir a sequência de atendimento. Ressalta-se que a ordem de grandeza do temporizador $\Delta_{schedule}$ é de segundos, enquanto do temporizador para a espera do reconhecimento da confirmação é de milissegundos.

Na ocorrência da falha mencionada acima, e considerando-se que o provedor receba a mensagem de reconhecimento da confirmação (*msgRecConfirmação*) do requisitante quando já estiver em movimento, definiu-se no $AG_{schedule}$ (Seção III-B) e no $AS_{schedule}$ (Seção III-C) que o provedor continuará o atendimento na direção do primeiro requisitante. No decorrer do primeiro atendimento, o dispositivo móvel no provedor analisará as novas mensagens de invocação (*msgInvocação*) recebidas. Caso o nó provedor verifique a viabilidade de incluir algum novo atendimento no percurso, sem prejuízo no janelamento previamente definido, o provedor enviará uma mensagem de alerta ao(s) novo(s) requisitante(s) e aos já selecionado(s) previamente, anunciando o novo tempo para o atendimento.

B. Escalonamento com Algoritmo Genético (AG)

A crescente complexidade dos problemas do mundo real tem motivado os cientistas da computação a buscar métodos eficientes de resolução de problemas de permutação [12]. Os Algoritmos Genéticos (AGs) consistem em uma técnica de Inteligência Computacional (IC) fundamentada em teorias e conceitos da genética e evolução das espécies. Os AGs se propõem a resolver problemas complexos do mundo real no qual métodos tradicionais de otimização possuem dificuldades em encontrar soluções viáveis [3].

O Algoritmo Genético, denominado neste ponto de $AG_{schedule}$, a partir da lista de adjacências de tempo de deslocamento gera uma fila de atendimento, tendo como prioridade o tempo de deslocamento do provedor, e o tempo máximo para o atendimento.

A codificação do indivíduo é dada pela permutação de um conjunto finito de seus elementos, neste caso os requisitantes. O indivíduo que representa a rota de atendimento do provedor é composto de uma ordenação entre requisitantes. A codificação define um caminho completo da origem ao destino em que será mensurado o tempo de deslocamento para o atendimento. Uma vez que as soluções iniciais foram

geradas aleatoriamente é necessário calcular a aptidão de cada indivíduo, representando seu *fitness*. Neste caso, o custo é associado ao caminho, de modo que, quanto menor o custo melhor a solução encontrada.

A formalização matemática para o problema de escalonamento de recursos com o uso de algoritmos genéticos requer a definição de um conjunto de parâmetros fundamentais: $N = \{1, 2, \dots, n\}$ que representa o conjunto com n nós do plano. O custo associado ao tempo de deslocamento entre os nós $i, j \subset N$ é dado por d_{ij} , e por fim, x_{ij} representa o número de vezes que é usada a ligação entre o nó i e o nó j (designadas como variáveis de decisão).

O modelo matemático do cálculo da função objetivo é apresentado na Equação 1, que consiste em ajustar o tempo de deslocamento para que o provedor de serviço percorra os distintos pontos uma única vez. A variável pn consiste no número de penalidades que ocorreram para cada possível solução na população, na qual é verificada sua factibilidade, ou seja, se é válida. A variável m representa um valor de penalidade. As Equações de 2 a 6 são as restrições da função objetivo.

$$\min \sum_{i=1}^n \sum_{j=1}^n d_{ij} x_{ij} + pn.m \quad (1)$$

Sujeito às restrições:

$$\sum_{i=1}^n x_{ij} = 1 \quad \forall i \in N \quad (2)$$

$$\sum_{j=1}^n x_{ij} = 1 \quad \forall i \in N \quad (3)$$

$$\sum_{i,j \in N} x_{ij} = 0 \quad \forall i \in N \quad (4)$$

$$x_{ij} \in \{0, 1\} \quad \forall i, j \in N, i \neq j \quad (5)$$

$$\sum_{i,j \in N} d_i + d_j \leq t_j \quad \forall i, j \in N \quad (6)$$

As Equações 2 e 3 estabelecem as restrições para que o provedor passe uma única vez em cada nó requisitante, devendo partir para o próximo ponto até que todo o percurso seja realizado. A Equação 4 determina que não é possível ir de um nó para ele mesmo. A Equação 5 representa confirmação de deslocamento x_{ij} , tal que $x_{ij} = 1$, se o provedor irá se deslocar do nó i para o nó j , ou $x_{ij} = 0$, caso contrário.

Na Equação 6 ajusta-se uma janela de tempo com o horário para o atendimento para cada ponto do percurso, $N = (i, j)$. Com isso, para cada nó i define-se uma janela de tempo Δ_{tmax} . Dessa maneira, um provedor não pode chegar ao local da ocorrência após Δ_{tmax} de um nó i . O principal objetivo a ser alcançado é a redução do custo de deslocamento visando minimizar o tempo total de percurso e aumentar o número de requisitantes atendidos.

Na função objetivo apresentada na Equação 1 se define os pesos e as penalizações de acordo com os requisitos para o atendimento. A variável pn na função objetivo consiste no número de penalidades e a variável m o valor do peso dessa penalidade. Caso não exista uma solução válida que compõe o caminho, o mesmo é julgado inválido. Como resultado, este caminho receberá uma nota alta como penalização, diminuindo suas chances de ser selecionado na próxima geração. Na modelagem do problema adotou-se a seguinte premissa: quanto maior a nota pior o caminho.

Com o algoritmo $AG_{schedule}$ é possível obter o tempo de deslocamento de um provedor até os requisitantes respeitando o tempo de atendimento Δ_{tmax} . Dessa forma, o $AG_{schedule}$ deverá evoluir suas soluções candidatas até chegar no seu critério de parada, que consiste no número de gerações. Na implementação do algoritmo optou-se pela escolha das seguintes técnicas: método de torneio para a seleção dos indivíduos; permutação de 2 (dois) pontos para o cruzamento entre os indivíduos para que não ocorra repetição dos segmentos (*Partially Mapped Crossover (PMX)*) e; para a mutação nos indivíduos utilizou-se a técnica de mutação aleatória, responsável por mudar a ordem dos genes no indivíduo [13].

Após o temporizador $\Delta_{schedule}$ terminar o algoritmo $AG_{schedule}$ calcula a fila de atendimento para que o provedor possa iniciar o deslocamento na direção do primeiro requisitante.

C. Escalonamento com Algoritmo A* (A-Estrela)

Como contribuição desse trabalho, também investigamos o emprego do algoritmo A-estrela (A*) [4], considerado um algoritmo de busca em grafos que encontra um caminho entre um nó de origem e um nó de destino. O algoritmo A-Estrela é reconhecido na literatura por sua precisão ao encontrar o melhor caminho para problemas de otimização.

O algoritmo A-Estrela (denominado aqui, $AS_{schedule}$) é baseado em uma heurística que encontra o caminho (mínimo *spanning tree*). A heurística opera através da função apresentada na expressão 7. Na função, $g(n)$ representa o custo atual entre um nó (i) e um nó (j), e $h(n)$ uma estimativa do custo restante para os nós adjacentes ao nó (j). Somente o nó mais promissor, aquele com o menor $f(n)$ será expandido, desde que respeitando a janela de tempo (t_i) dos requisitantes (ver Função 8) no momento da expansão.

$$f(n)=g(n)+h(n) \quad (7)$$

$$g(n) + h(n) \leq t_i \quad (8)$$

O algoritmo $AS_{schedule}$, a partir da lista de adjacências de tempo de deslocamento (ver Figura I) gera uma fila de atendimento, tendo como prioridade o tempo de deslocamento do provedor, e o tempo máximo para o atendimento.

Em relação ao A-Estrela na sua versão original, no $AS_{schedule}$ foi necessário desenvolver um mecanismo que possibilitasse ao provedor percorrer todos os nós com um menor custo. Além disso, foi adicionado ao $AS_{schedule}$ uma janela de tempo com o horário para o atendimento t_i em

cada ponto do percurso, calculado conforme a expressão 8. A expressão 8 estabelece que um provedor não pode chegar ao local da ocorrência após Δ_{tmax} de um nó i .

IV. RESULTADOS COMPUTACIONAIS

Nesta seção são apresentados os resultados computacionais obtidos da comparação entre as abordagens para escalonamento de recursos: $AG_{schedule}$ e $AS_{schedule}$. Também foi realizada a comparação com o algoritmo de invocação de serviços do protocolo LADP [1] em sua versão original quanto ao número de requisitantes atendidos. O desempenho dos algoritmos foi avaliado no simulador de redes *Network Simulator (NS3)* [14].

A. Ambiente de Simulação

Um número de 350 nós (composto por requisitantes e intermediários) foram distribuídos em uma área de 2000m x 2000m. O número de requisitantes varia entre 50 e 300 nós. Um provedor (ambulância) foi distribuído na área para prestar o serviço de atendimento. O tamanho do pacote da mensagem de invocação é de 128 *bytes* contendo informações como, localização geográfica do requisitante e o tempo para o atendimento do pedido. A taxa de transmissão na rede é de 2Mbps, sendo o raio de alcance da antena dos dispositivos de 250m. O tempo total de simulação é de 7200s. A energia inicial dos dispositivos é de 1000 Joules, que foram configurados com potência de transmissão igual a 0,055A, potência de recepção igual a 0,042A, idle de 0,24A e sleep de 0,016A.

O tempo máximo para atendimento da requisição Δ_{tmax} estipulado pelos requisitantes é de 15 minutos, esse tempo foi definido conforme as normas de tempo de resposta recomendado pelo Serviço de Atendimento Móvel de Urgência (SAMU) quando o socorro é feito por ambulâncias é entre 10 a 12 minutos. O tempo de espera pelo nó provedor para o recebimento de pedidos $\Delta_{schedule}$ foi ajustado em 10s.

Os nós são classificados em: nós requisitantes, nós intermediários e nó provedor. Os nós móveis são distribuídos conforme o modelo de mobilidade *Gauss-Markov* [15]. Os nós requisitantes são estáticos, enquanto o nó provedor move-se com a velocidade de 1,8 m/s e os intermediários com 0,5 m/s. Nas simulações optou-se pelo protocolo de roteamento *Optimized Link State Routing Protocol (OLSR)*. O OLSR é um protocolo de roteamento para MANETs de múltiplos saltos. Cada nó usa sua informação local para rotear os pacotes, buscando reduzir o número de mensagens de controle para realizar o roteamento [16].

Alguns fatores como, tamanho da área, número de nós, velocidade de deslocamento do provedor e dos nós intermediários e número de requisitantes podem afetar o desempenho dos algoritmos. Para analisar o desempenho dos algoritmos para escalonamento foram definidas as métricas: Tempo de processamento e Número de atendimentos.

B. Resultados e Análises

Nos cenários analisados, foram avaliados os algoritmos: $AG_{schedule}$ e $AS_{schedule}$. Nos experimentos determinou-se o

número de 10 execuções no simulador. No $AG_{schedule}$ o valor da população é 100, o número de gerações é 1000, o valor da penalidade m é igual a 1000, o método para mutação da ordem dos genes é o *SwapMutator* com valor de 1% e o *PMX* é de 90%. Para realizar a análise do tempo de processamento dos algoritmos e quantidade de atendimentos, os requisitantes foram distribuídos na área de uma forma que o provedor esteja dentro do seu raio de alcance possibilitando o recebimento da mensagem de descoberta.

1) *Análise de Processamento dos Algoritmos de Escalonamento de Recursos*: O experimento conforme sistematizado na Tabela II demonstra o tempo médio de processamento dos algoritmos de escalonamento para a geração da fila de atendimento. No cenário com 50 requisitantes o algoritmo $AS_{schedule}$ apresentou um tempo médio de processamento menor quando comparado ao $AG_{schedule}$. Por exemplo, o $AG_{schedule}$ alcançou um tempo de 1,49s enquanto, o $AS_{schedule}$ levou 0,28s. Este comportamento justifica-se pelo fato do $AS_{schedule}$ ter organizado a fila dentro de um espaço de busca reduzido. Neste aspecto o $AS_{schedule}$ superou o $AG_{schedule}$ que precisou evoluir suas soluções candidatas até alcançar o número de gerações definidas pelo algoritmo.

Tabela II
TEMPO DE PROCESSAMENTO(S) E DESVIO PADRÃO

Nós	$AG_{schedule}$		$AS_{schedule}$		LADP	
	Tempo(s)	DP	Tempo(s)	DP	Tempo(s)	DP
50	1.49	0.09	0.28	0.03	0.001	0.00
100	2.56	0.25	2.87	0.13	0.001	0.00
150	3.91	0.85	16.35	1.14	0.001	0.00
200	6.06	1.61	63.33	3.85	0.001	0.00
250	10.00	1.80	182.32	10.76	0.001	0.00
300	12.04	4.13	438.17	41.73	0.001	0.00

Contudo, percebe-se que a medida que aumenta a quantidade de requisitantes o tempo de processamento do $AS_{schedule}$ também aumenta, isso ocorre porque o espaço de busca causada é exponencial. Tendo como exemplo, no cenário com 300 requisitantes o $AS_{schedule}$ estimou um tempo médio de 438,17s e o $AG_{schedule}$ um tempo médio de 12,04s. O protocolo LADP ficou com um tempo bem abaixo dos demais algoritmos, próximo dos 0,001s. O LADP por ser determinístico e possuir um mecanismos de escalonamento simples, apesar de ser mais rápido, a quantidade de requisitantes atendidos foi menor. Como exemplo, com 300 nós o protocolo LADP conseguiu atender no máximo 4 requisitantes. No gráfico da Figura 3, ilustra os resultados obtidos na Tabela II e devido a grande gama de valores gerados utilizou-se uma escala logarítmica para uma melhor visualização.

No gráfico reproduzido na Figura 4 apresenta-se o número de requisitantes atendidos pelo provedor. Ambos algoritmos organizaram a fila de atendimento em relação ao número de invocações (*msgInvocação*) recebidas pelo provedor. Analisando o resultado do algoritmo $AS_{schedule}$ percebe-se que a quantidade de atendimentos tendo como número de requisitantes entre 50 e 250 foi superior aos algoritmos $AS_{schedule}$ e LADP, porém, a medida que aumenta a quantidade de requisitantes, o tempo de processamento é maior, diminuindo dessa forma

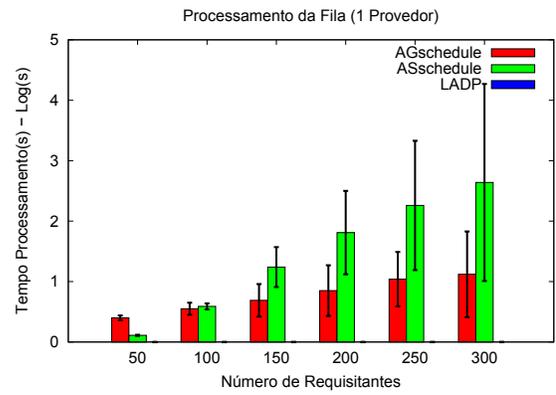


Figura 3. Avaliação do tempo de processamento para a geração da fila, em função do número de requisitantes.

o tempo disponibilizado ao provedor para atendimento aos requisitantes. Por exemplo, no cenário com 300 nós requisitantes o número de atendidos foi 3,37%, isso porque o tempo necessário para o algoritmo $AS_{schedule}$ organizar a fila de atendimento foi de 468,17s (acima de 7 minutos). O algoritmo $AG_{schedule}$ manteve seu tempo de processamento com média de 12,04s (abaixo de 1 minuto).

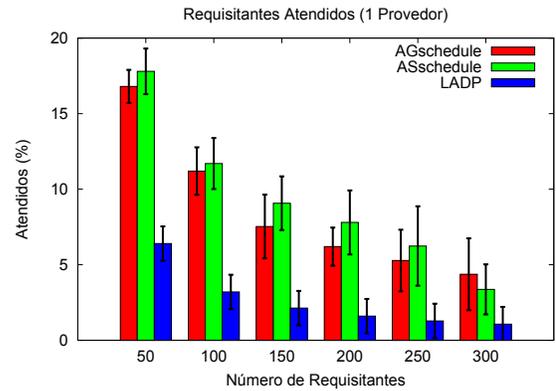
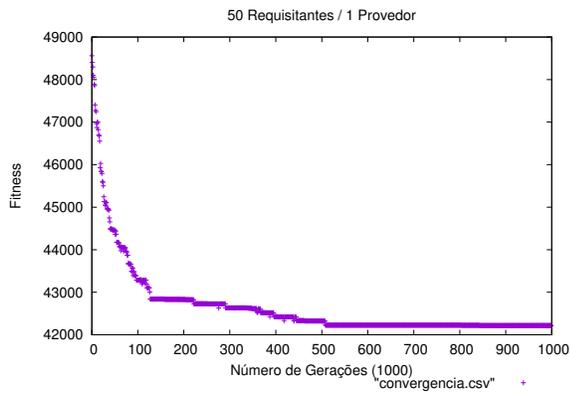


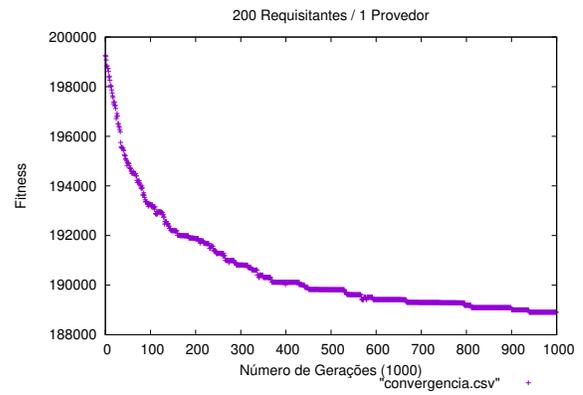
Figura 4. Avaliação dos requisitantes atendidos, em função do número de requisitantes.

Buscando reduzir o tempo de atendimento e aumentar a quantidade de clientes atendidos, objeto do algoritmo genético para escalonamento de recursos proposto, avaliou-se a capacidade de convergência do algoritmo $AG_{schedule}$. A atuação do $AG_{schedule}$ teve um comportamento variado de acordo com a quantidade de requisitantes, um exemplo pode ser visualizado no gráfico de convergência da Figura 5(a) quando se comparado ao gráfico de convergência da Figura 6(c). O eixo x representa a *fitness* e o eixo y o número de gerações. A média obtida em cada geração é ilustrada em relação ao custo de tempo de deslocamento até os requisitantes.

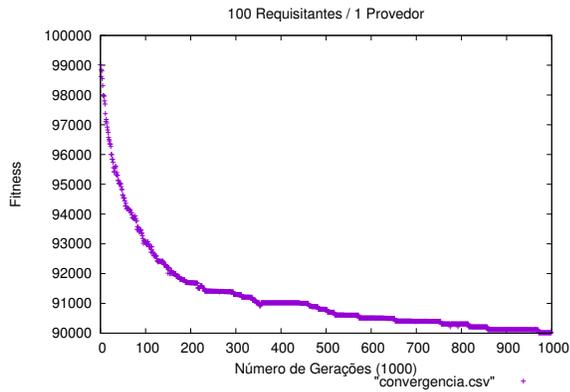
Analisando os gráficos de convergência da métrica *fitness*, percebe-se que a convergência do algoritmo $AG_{schedule}$ não ocorreu ao atingir o máximo de gerações disponíveis. Este resultado indica que uma melhor solução pode ser encontrada



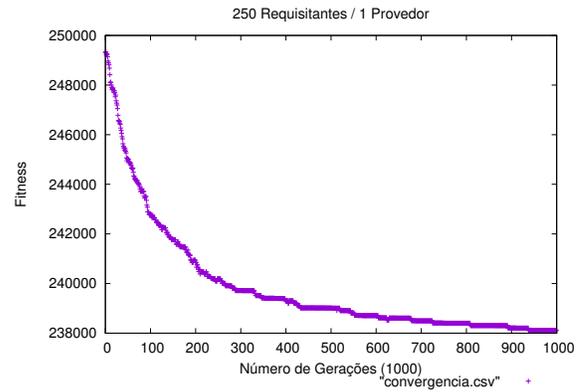
(a) Gráfico convergência para 50 requisitantes



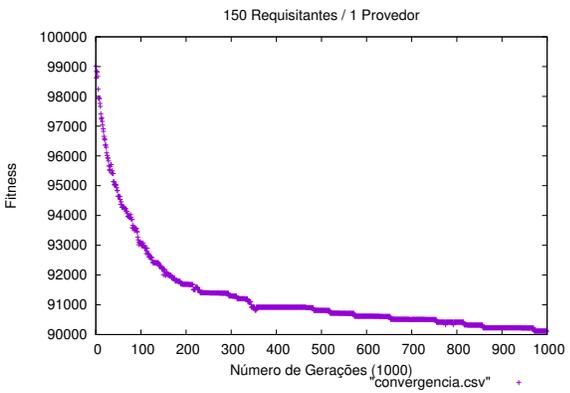
(a) Gráfico convergência para 200 requisitantes



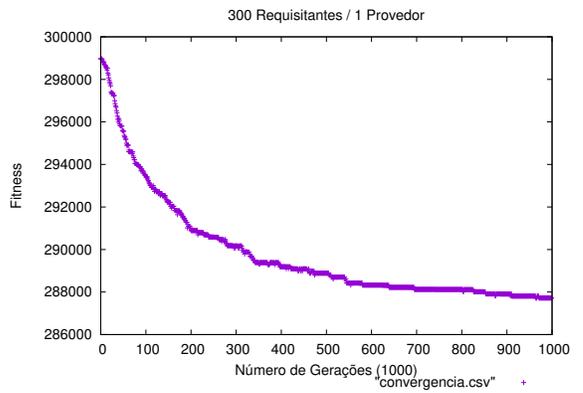
(b) Gráfico convergência para 100 requisitantes



(b) Gráfico convergência para 250 requisitantes



(c) Gráfico convergência para 150 requisitantes



(c) Gráfico convergência para 300 requisitantes

Figura 5. Avaliação do gráfico de convergência (*fitness*), em função do número de gerações (50 a 150 requisitantes).

Figura 6. Avaliação do gráfico de convergência (*fitness*), em função do número de gerações (200 a 300 requisitantes).

aumentando o número de total de gerações. Porém isso resultaria em um aumento no tempo de processamento do algoritmo, algo que impacta diretamente com os objetivos propostos neste trabalho, que é a redução do tempo de atendimento.

V. CONCLUSÕES

O mecanismo de escalonamento de recursos foi uma contribuição desenvolvida no contexto de protocolos de descoberta de serviços para MANETs de múltiplos saltos. Este mecanismo estende o protocolo *Location Aware Discovery protocol*

(LADP) [1] que oferece as fases de descoberta, seleção e uma simples invocação. A versão original do protocolo não contempla o escalonamento dos provedores, bem como os protocolos de descoberta de serviço para o mesmo contexto conforme verificado na literatura.

Neste artigo apresentou-se o desenvolvimento e a análise de desempenho dos algoritmos de escalonamento de recursos, $AG_{schedule}$ e $AS_{schedule}$. Ambos os algoritmos encontraram soluções diante das métricas avaliadas possibilitando que um

provedor consiga atender a um número de requisitantes e dentro do tempo hábil.

A discussão apresentada sobre os protocolos de descoberta de serviços reflete a necessidade de um mecanismo de escalonamento de recursos que permita que os provedores possam se deslocar para o local do evento dentro de um tempo limite visando atender ao maior número de requisitantes. No mecanismo, pode-se incluir algum novo atendimento no percurso, e analisar o prejuízo no janelamento definido. O mecanismo também permite a inclusão de um tempo de parada em cada requisitante e obstáculos no percurso.

Existem na literatura abordagens para escalonamento de recursos, contudo, a utilização do Algoritmo Genético e do A-Estrela deve-se ao fato de possibilitarem a organização da fila de atendimento e com baixa transmissão de mensagens. A economia de energia é essencial para prolongar o tempo de vida da rede, considerando que os dispositivos não podem ser recarregados facilmente.

Com os testes realizados, foi observado que $AS_{schedule}$ obteve melhores resultados quando o seu espaço de busca para solução eram mais curtos, mas a medida que aumenta a quantidade de requisitantes o algoritmo requer maior tempo de processamento, o que faz com que implique diretamente no tempo de deslocamento do provedor para atendimento aos requisitantes.

No $AG_{schedule}$ o tempo de execução manteve-se baixo a medida que aumentou o número de nós, por outro lado a quantidade de requisitantes atendidos foram inferiores ao $AS_{schedule}$. Uma solução é aumentar o número de gerações sem afetar o tempo de execução do algoritmo. Um estudo aprofundado será realizado com relação ao emprego de paralelização em AGs [13], visto que tem como princípio básico evoluir uma grande população de indivíduos. Dentro de um curto prazo, novos experimentos serão realizados, aumentando número de gerações do $AG_{schedule}$, visando verificar a evolução do algoritmo.

Cenários com uma área maior e com obstáculos serão avaliados. Outro aspecto que será analisado diz respeito as situações em que o provedor receba mensagens de invocação de outros requisitantes quando o mesmo já estiver em movimento.

REFERÊNCIAS

- [1] J. Kniess, O. Loques, and C. V. N. de Albuquerque, "Descoberta de serviço em redes ad hoc móveis," Ph.D. dissertation, Universidade Federal Fluminense (UFF), 7 2011.
- [2] J. Kniess, L. Arantes, P. Sens, and C. V. N. de Albuquerque, "Saving resources in discovery protocol on delay-sensitive rescue mobile networks," in *31st IEEE International Conference on Advanced Information Networking and Applications, AINA 2017, Taipei, Taiwan, March 27-29, 2017*, 2017, pp. 538–545. [Online]. Available: <https://doi.org/10.1109/AINA.2017.81>
- [3] A. Shukla, H. M. Pandey, and D. Mehrotra, "Comparative review of selection techniques in genetic algorithm," in *Futuristic Trends on Computational Analysis and Knowledge Management (ABLAZE), 2015 International Conference on*. IEEE, 2015, pp. 515–519.
- [4] P. E. Hart, N. J. Nilsson, and B. Raphael, "A formal basis for the heuristic determination of minimum cost paths," *IEEE transactions on Systems Science and Cybernetics*, vol. 4, no. 2, pp. 100–107, 1968.
- [5] L. Juszczyk and S. Dustdar, "A middleware for service-oriented communication in mobile disaster response environments," in *Proceedings of the 6th international workshop on Middleware for pervasive and ad-hoc computing*. ACM, 2008, pp. 37–42.
- [6] M. A. Serhani and Y. Gadallah, "A service discovery protocol for emergency response operations using mobile ad hoc networks," in *Telecommunications (AICT), 2010 Sixth Advanced International Conference on*. IEEE, 2010, pp. 280–285.
- [7] G. Arenella, F. de Santis, and D. Malandrino, "Beead hoc servicediscovery: A MANET service discovery algorithm based on bee colonies," in *Informatics in Control, Automation and Robotics (ICINCO), 2014 11th International Conference on*, vol. 1. IEEE, 2014, pp. 244–251.
- [8] S. Chalotra, S. K. Sehra, and S. S. Sehra, "A systematic review of applications of bee colony optimization," in *Innovation and Challenges in Cyber Security (ICICCS-INBUSH), 2016 International Conference on*. IEEE, 2016, pp. 257–260.
- [9] H. N. Saha, A. Chattopadhyay, and D. Sarkar, "Review on intelligent routing in MANET," in *Computing and Communication (IEMCON), 2015 International Conference and Workshop on*. IEEE, 2015, pp. 1–6.
- [10] J. Kniess, O. Loques, and C. V. N. de Albuquerque, "Service discovery with time constraints in mobile ad hoc networks," *Earth Science Informatics*, vol. 8, no. 2, pp. 439–452, 2015. [Online]. Available: <http://dx.doi.org/10.1007/s12145-014-0164-4>
- [11] M. M. Deza and E. Deza, "Encyclopedia of distances," in *Encyclopedia of Distances*. Springer, 2009, pp. 1–583.
- [12] R. S. Parpinelli and H. S. Lopes, "New inspirations in swarm intelligence: a survey," *International Journal of Bio-Inspired Computation*, vol. 3, no. 1, pp. 1–16, 2011.
- [13] R. Linden, *Algoritmos genéticos (2a edição)*. Brasport, 2008.
- [14] N.-. D. Team, "Ns-3 network simulator," Acessado em 25-04-2017 a url <https://www.nsnam.org/>, 2008.
- [15] D. Chenghao, "An improved routing protocol based on gauss-markov model in ad hoc networks utilizing prediction of link quality," in *Control Conference (CCC), 2015 34th Chinese*. IEEE, 2015, pp. 6507–6511.
- [16] T. Clausen and P. Jacquet, "Optimized link state routing protocol (olsr)," 2003.