



ALGORITMO GENÉTICO PARA ESCALONAMENTO DE RECURSOS EM DESCOBERTA DE SERVIÇOS PARA MANETS

Marcelo Petri

Programa de Pós-Graduação em Computação Aplicada-PPGCA
UDESC
Joinville, Brasil
marcelo.petri@gmail.com

Janine Kniess

Programa de Pós-Graduação em Computação Aplicada-PPGCA
UDESC
Joinville, Brasil
janine.kniess@udesc.br

Rafael Stubs Parpinelli

Programa de Pós-Graduação em Computação Aplicada-PPGCA
UDESC
Joinville, Brasil
rafael.parpinelli@udesc.br

RESUMO

Este artigo apresenta um algoritmo genético para escalonamento de recursos em descoberta de serviços em MANETs operando em cenários de emergência. Os recursos a serem compartilhados são por exemplo, ambulâncias ou carros de apoio. Com um modelo eficiente de escalonamento de recursos pretende-se maximizar o número de atendimento as vítimas na área afetada. Os resultados obtidos da avaliação de desempenho realizada no *Network Simulator NS-3* comprovam que o algoritmo é eficiente em relação ao tempo de atendimento e escalável para diferentes tamanhos de área.

PALAVRAS CHAVE. Algoritmos Genéticos, Descoberta de Serviços, MANETS.

MH - Metaheurísticas

ABSTRACT

This paper presents a genetic algorithm for resource scheduling in service discovery to MANETs operating in emergency scenarios. The shared resources are for example, ambulances or support cars. Through an appropriate model for scheduling resources, we aim to attend the greatest number of victims in the affected area. Performance evaluation results on the *Network Simulator NS-3* confirm the effectiveness of genetic algorithm for resource scheduling.

KEYWORDS. Genetic Algorithms. Service Discovery. MANETS.

MH - Metaheurísticas



1. Introdução

Um desastre natural ou não natural é uma situação de emergência que requer o gerenciamento eficiente das equipes de resgate para atender ao maior número de vítimas. A busca por provedores de serviço (e.g., ambulâncias ou carros do corpo de bombeiros) pelas vítimas deve ocorrer rapidamente após o desastre. Neste contexto, protocolos para descoberta e reserva de recursos são essenciais para possibilitar que as vítimas localizem e aloquem os recursos (provedores de serviços) para o atendimento. Contudo, em situações de emergência, a infraestrutura de comunicação pode ser danificada não sendo possível fornecer serviços de comunicação adequados para as equipes de resgate e vítimas (clientes).

As Redes Móveis Ad Hoc (*Mobile Ad Hoc Networks MANETs*), são particularmente atrativas para cenários onde exista a necessidade de instalar rapidamente uma rede de comunicação. As MANETs podem ser usadas em cenários de emergência para troca de informações entre clientes e socorristas (provedores de serviços). Com essas informações, os provedores podem tomar decisões sobre os atendimentos a serem realizados [Kniess et al., 2011].

Corpo de bombeiros e ambulâncias são serviços indispensáveis e nem sempre disponíveis a contento em muitas cidades, com propósito de atendimento eficiente em situações de emergência ou pós-desastres. A rapidez com que as equipes de resgate possam responder ao chamado e prestar o atendimento necessário no local da ocorrência, são essenciais em situações de emergência. A complexidade para gerenciar os recursos em relação ao número de atendimentos requer o uso de algoritmos adequados de escalonamento de recursos.

A crescente complexidade dos problemas do mundo real tem motivado os cientistas da computação a buscar métodos eficientes de resolução de problemas. Computação evolutiva e de inteligência de enxames são exemplos de que a natureza tem sido uma fonte inesgotável de inspiração [Parpinelli e Lopes, 2011]. Os Algoritmos Genéticos (AGs) consistem em uma técnica de Inteligência Computacional (IC) fundamentada em teorias e conceitos da genética e evolução das espécies de acordo com a seleção natural e descendência com hereditariedade proposta por Darwin [Varghese e Raj, 2016]. Os AGs se propõem a resolver problemas complexos do mundo real onde métodos tradicionais de otimização possuem dificuldades em encontrar soluções viáveis. O princípio do funcionamento consiste em explorar o espaço de busca utilizando uma população de soluções candidatas (cromossomos), evoluindo a cada geração por meio de operadores genéticos de crossover e mutação. O AG requer a definição do problema codificando soluções possíveis para a questão na forma de cromossomo (indivíduo) e assim avaliar cada solução de acordo com a função de avaliação ou aptidão (*fitness*). Entre os componentes básicos dos algoritmos genéticos destacam-se: cromossomos (indivíduo ou soluções possíveis), operador de crossover, operador de mutação, operador de seleção e função aptidão [Shukla et al., 2015].

O número de vítimas em uma situação de emergência pode aumentar consideravelmente de acordo com a gravidade da calamidade. Portanto, caracterizou-se o escalonamento de recursos no cenário alvo como um problema de permutação. Além do número de vítimas, um algoritmo de escalonamento neste contexto, deve atender ao requisito de tempo máximo de atendimento especificado pelas vítimas. Dessa forma, identificou-se na literatura que a problemática em questão é uma variação do clássico problema do caixeiro viajante, onde se aplicam janelas de tempo (*Travelling Salesman Problem with Time Windows - TSPTW*) [Xiong e Qidi, 2002], que visa encontrar um caminho de custo mínimo, visitando um conjunto de cidades uma única vez e dentro de uma janela de tempo específica.

Neste artigo apresenta-se como principal contribuição um AG para escalonamento de recursos direcionado para protocolos de descoberta de serviços em MANETs de múltiplos saltos, denominado de Algoritmo Genético para Escalonamento (*AGE*). O algoritmo *AGE* leva em consideração a localização geográfica dos clientes, o tempo de deslocamento dos provedores ao local do evento e o tempo máximo de atendimento determinado pelo cliente para realizar o escalonamento dos recursos. Com o uso do *AGE* pretende-se reduzir o tempo de deslocamento do



provedor até as vítimas e portanto aumentar o número de atendimentos. Este trabalho estende o protocolo de descoberta de serviço proposto em [Kniess et al., 2011] em que questões de escalonamento de recursos entre os clientes não foram considerados, sendo esta uma das contribuições deste trabalho.

Este trabalho está organizado como segue: na Seção 2 apresenta-se a motivação e os trabalhos relacionados. Na Seção 3 apresenta-se o algoritmo genético para escalonamento. Os resultados computacionais são discutidos na Seção 4. Na Seção 5 discorre-se sobre as conclusões e trabalhos futuros.

2. Motivação e Trabalhos Relacionados

O cenário de motivação deste trabalho compreende regiões afetadas por desastres naturais ou não naturais. Nesse cenário, tanto o acesso às áreas isoladas quanto a identificação de sobreviventes e de eventos (focos de incêndio, vazamento de gás etc.), é bastante difícil. Equipes especializadas em salvamento e resgate encontram-se distribuídas na área afetada. Nesse contexto, conjectura-se que equipes de resgate (formadas por humanos, veículos ou mesmo robôs) interconectadas por uma rede móvel *ad hoc* utilizando uma tecnologia de comunicação a exemplo do wifi (802.11), poderia contribuir de forma significativa na identificação de eventos, e no resgate de sobreviventes [Kniess et al., 2011].

No cenário alvo, os elementos móveis transportam um dispositivo para a comunicação sem fio e conhecem sua posição geográfica através de um sistema de localização. Os nós que proveem os serviços são entidades físicas chamados de provedores e os que requisitam o serviço são entidades físicas chamados de clientes. Os provedores de serviço são móveis, porém, o cliente permanece fixo à espera de resposta(s) para uma requisição específica.

Admitindo-se que no local da calamidade exista um dispositivo móvel coletando informações do ambiente, os dados obtidos a partir da identificação de um fenômeno, por exemplo, o vazamento de algum tipo de gás ou um incêndio, serão transformados em uma descrição do evento observado [Kniess et al., 2011]. Em seguida, partindo de algoritmos especializados, o dispositivo móvel deduz que é necessário um provedor capaz de atuar em poucos minutos naquela área. Após a identificação do provedor de serviço necessário, o protocolo proposto por [Kniess et al., 2011] (entidade lógica no cenário), estabelece que o dispositivo móvel envie para a rede a mensagem de requisição buscando o provedor de serviço apropriado.

Visando reduzir o tempo de deslocamento do provedor para o atendimento no cenário alvo, e aumentar o número de vítimas atendidas, propõe-se neste artigo um algoritmo de escalonamento de recursos para descoberta de serviços em MANETs com base em algoritmos genéticos.

2.1. Protocolos para Descoberta de Serviços

O conceito de descoberta de serviços surgiu no contexto de sistemas de informações auto-organizáveis. De uma forma geral, os protocolos de descoberta de serviços viabilizam a detecção automática de dispositivos e de serviços oferecidos por esses dispositivos em uma rede de computadores [Mian et al., 2009].

Não obstante, muitas pesquisas sobre protocolos de descoberta de serviços para MANETs de múltiplos saltos tenham sido realizadas, GSD [Chakraborty et al., 2006], Konark [Helal et al., 2003], Lanes [Klein et al., 2003], FTA [Lenders et al., 2005], Allia (*Alliance*) [Ratsimor et al., 2004], e de Varshavsky *et al.* [2005] [Varshavsky et al., 2005], não foi encontrado na literatura protocolos que apresentam soluções para o escalonamento de provedores de serviços móveis em cenários de desastre [Kniess et al., 2015].

O protocolo de descoberta de serviços *Location Aware Discovery Protocol (LADP)* para MANETs proposto por [Kniess et al., 2011] foi desenvolvido para atender aos requisitos de sistemas para ambientes de desastres naturais e não-naturais. A arquitetura do protocolo LADP foi especificada com a finalidade de oferecer suporte ao desenvolvimento de aplicações para descoberta, seleção e invocação de serviço em MANETs de múltiplos saltos, possibilitando aos dispositivos



sem fio, utilizar os recursos disponíveis na rede, mesmo na presença de obstáculos no caminho e na de falhas dos nós [Kniess et al., 2015]. Características e detalhes do protocolo LADP serão discutidos na Seção 3.

O algoritmo de descoberta e seleção de serviço *BeeAdHocServiceDiscovery* (BAHSD) proposto por [Arenella et al., 2014] é baseado no conceito de (*Bee Colony Optimization BCO*) [Chalotra et al., 2016]. BCO é uma meta-heurística derivada do comportamento de uma colônia de abelhas para resolver problemas combinatórios. 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 juntamente com o algoritmo de roteamento *BeeAdHoc* [Saha et al., 2015]. O *BeeAdHoc* coleta informações de roteamento, tais como quebras de caminho e atualizações, a fim de minimizar o número de mensagens de controle e o consumo de energia.

Devido ao fato de a função de descoberta e seleção de serviço estar integrada ao protocolo de roteamento *BeeAdHoc* [Saha et al., 2015], as características da transmissão das mensagens do protocolo BAHSD [Arenella et al., 2014] ficam associadas ao protocolo de roteamento, diminuindo a interoperabilidade do protocolo. Os clientes ao receberem as informações sobre os provedores aptos enviam uma solicitação de serviço para estes provedores. Mecanismos para o escalonamento de recursos e aspectos relacionados a descoberta de serviço em situações de emergência não foram considerados neste trabalho.

Gadallah and Serhani [Serhani e Gadallah, 2010] apresentam um protocolo de descoberta de serviços para MANETs distribuídas em áreas de desastre. Os provedores de serviço enviam anúncios para uma central que realiza a seleção dos provedores para os clientes. O protocolo permite localizar serviços com base nos requisitos especificados pelo cliente a saber, a proximidade do prestador de serviços e a gravidade da emergência.

No artigo proposto por [Park et al., 2013], é apresentada uma arquitetura distribuída de descoberta e compartilhamento de serviços em MANETs. A proposta consiste em integrar o protocolo de roteamento pró-ativo *Destination-Sequenced Distance Vector (DSDV)* com o protocolo de descoberta de serviços. O protocolo DSDV mantém informações utilizadas pelo usuário na tabela de roteamento. Essas informações servem para descoberta e localização de serviços de acordo com o tipo desejado. São utilizados de cada nó, informações como, posição geográfica, topologia da rede e o tipo de serviço. A arquitetura do protocolo compreende: a descoberta do serviço, o registro do serviço e a invocação do serviço.

De modo geral, os protocolos de descoberta de serviço implementam a fase de invocação de serviços, como ocorre com os protocolos LADP [Kniess et al., 2015], BAHSD proposto por [Arenella et al., 2014], [Serhani e Gadallah, 2010] e [Park et al., 2013]. Todavia, protocolos para descoberta de serviços existentes na literatura foram avaliados no trabalho de [Kniess, 2011] e [Kniess et al., 2015], onde o autor verificou que os protocolos não disponibilizam mecanismos para otimizar o uso dos provedores em relação as vítimas (pessoas) na área afetada. Para lidar com esta limitação dos protocolos de descoberta de serviço para o cenário alvo desenvolveu-se o algoritmo proposto neste artigo.

A discussão apresentada sobre os protocolos [Kniess et al., 2015], [Kniess et al., 2011], [Arenella et al., 2014], [Serhani e Gadallah, 2010] e [Park et al., 2013], reforça a necessidade de um novo mecanismo de escalonamento de recursos que contemple as peculiaridades do cenário alvo deste trabalho. No cenário alvo, os provedores devem se deslocar para o local onde o pedido é necessário dentro de um tempo limite e atender ao maior número de vítimas possível.

Em face das limitações dos protocolos de descoberta de serviços existentes para atender ao escalonamento de recursos no cenário alvo proposto, este artigo oferece uma solução na forma de um algoritmo genético observando, a localização geográfica dos clientes, o tempo de deslocamento dos provedores ao local do evento e o tempo máximo de atendimento determinado pelo cliente, para assim diminuir o tempo de deslocamento do provedor e aumentar o número de atendimentos.



3. Algoritmo Genético para Escalonamento - AGE

Nesta seção apresenta-se o algoritmo genético para escalonamento de recursos operando em cenários de emergência, apresentando-se o detalhamento do problema (Seção 3.1) e detalhes do AG desenvolvido (Seção 3.2).

3.1. Detalhamento do Problema

No protocolo LADP, as vítimas iniciam o processo de descoberta de serviço através de um dispositivo móvel interconectado à MANET. A mensagem de descoberta será transmitida para a rede, salto a salto, através de um mecanismo de difusão do tipo *broadcast* sendo que todos os nós podem atuar como *relay* repassando as informações. O protocolo de descoberta de serviço, previamente configurado nos dispositivos das vítimas e provedores, seleciona os provedores (socorristas) aptos para realizar o atendimento. A mensagem irá trafegar dentro de um raio R_i delimitado pelo mecanismo de descoberta de serviço. O mecanismo de seleção de serviços, executa um filtro seletivo nos nós intermediários e envia as repostas dos provedores mais aptos na direção da vítima. Como cada provedor mantém armazenadas as informações sobre o recurso que disponibiliza, logo qualquer nó fora do raio, que receba a mensagem de descoberta, irá descartá-la. Na mensagem de descoberta, o nó cliente envia as seguintes informações: uma identificação do nó, uma identificação da requisição, o número de sequência da requisição, suas coordenadas geográficas, o tempo máximo para atendimento da requisição (Δ_{max}), o serviço procurado, e o número de provedores desejado [Kniess et al., 2011] e [Kniess et al., 2015].

Após selecionar os provedores, o cliente envia uma mensagem de invocação de serviço para o(s) provedor(es) selecionado(s). Este(s), por sua vez, devem enviar uma mensagem de confirmação de serviço para o cliente. Por fim, o cliente envia uma mensagem de reconhecimento da confirmação ao provedor. Cada mensagem de reconhecimento da confirmação recebida tem por função notificar ao cliente que o provedor irá se deslocar fisicamente até o local onde o recurso é necessário. O algoritmo *AGE* atua na fase de invocação de serviço do protocolo LADP.

A entidade responsável pelo agendamento é o nó provedor. A solicitação do serviço é realizada pelo cliente através da mensagem de descoberta. Após o provedor ser selecionado conforme estabelece o mecanismo de seleção do protocolo LADP, o cliente envia uma mensagem de invocação ao provedor que ao recebê-la inicia um temporizador para aguardar mensagens de invocação de outros clientes.

A Figura 1 ilustra o funcionamento do algoritmo de escalonamento de recursos *AGE* a partir da mensagem de invocação enviada pelos clientes (1, 2, 3 e 4) solicitando uma ambulância (provedor P) para o atendimento. Após o provedor (P) receber a primeira mensagem de invocação de um cliente (e.g., cliente A), o provedor inicia o temporizador Δ_{AGE} . Este temporizador representa o tempo de espera no nó provedor por mensagem(s) de invocação. Para as mensagens de invocação recebidas, o nó provedor enviará aos clientes uma mensagem de confirmação da invocação em *unicast* e aguarda a mensagem de reconhecimento da confirmação.

Quando o temporizador Δ_{AGE} termina, é gerado uma matriz de adjacências contendo informações relacionadas ao tempo de deslocamento do provedor para cada um dos clientes solicitantes e seus pares, e um vetor contendo o tempo estipulado pelo cliente para o atendimento (janelamento). A Seção 3.2 traz uma ilustração conceitual das adjacências e do janelamento. Em seguida, é ilustrado como o algoritmo *AGE* organiza a sequência de atendimento visando maximizar o número de clientes atendidos dentro da janela de tempo definida. Diante da resposta do *AGE*, o provedor envia uma mensagem de aviso de atendimento aos clientes e inicia o deslocamento na direção do primeiro cliente.

A formalização matemática para o problema de escalonamento de recursos com base no TSPTW 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 $i, j \in N$ 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 por variáveis de decisão).

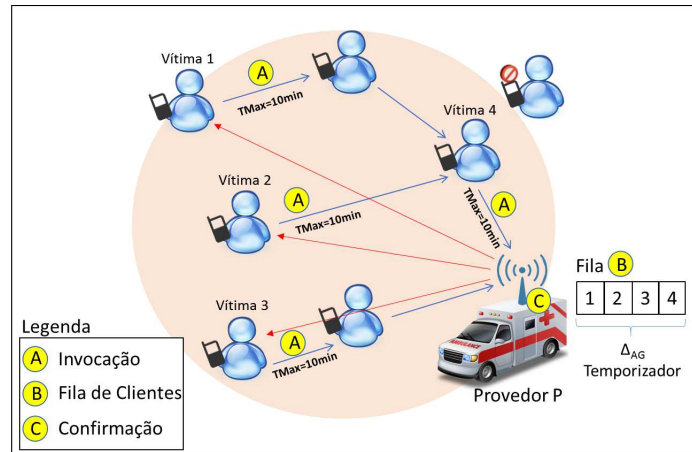


Figura 1: Ilustração do Problema

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 p 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.

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

Sujeito às restrições:

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

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

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

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

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

As Equações (2) e (3) estabelecem as restrições para que o provedor passe uma única vez em cada nó solicitante, 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 cliente i define-se uma janela de tempo $[tmax_i]$, sendo $tmax_i$ um vetor contendo o tempo de atendimento para cada cliente. Dessa maneira, um provedor não pode chegar ao local da ocorrência após $tmax_i$. O principal objetivo a ser alcançado é a redução do custo de deslocamento, minimizando o tempo total de percurso e aumentar o número de clientes atendidos.



3.2. Projeto do AGE

O problema a ser resolvido consiste em fazer que o AG, a partir de uma lista de adjacências de tempo de deslocamento como demonstrado na Figura 2(a), gere uma fila de atendimento, tendo como prioridade o tempo de deslocamento e o tempo de atendimento (Δ_{max}) especificado na mensagem de descoberta de serviço enviada pelo cliente.

A codificação do indivíduo é dada pela permutação de um conjunto finito de seus elementos, neste caso os clientes. 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*. No caso desse exemplo, o custo é associado ao caminho, de modo que, quanto menor o custo melhor a solução encontrada. A Equação (1) gera a nota da avaliação a partir da matriz de tempo de deslocamento gerada pelo simulador *Network Simulator 3 (NS3)* [Ikeda et al., 2011].

Cada indivíduo é considerado um possível caminho. Para a formulação da função foram consideradas algumas restrições, como por exemplo, verifica-se a viabilidade do provedor atender ao próximo cliente ou não. Se o tempo limite excedeu (Δ_{max}), e se o provedor continua para o próximo atendimento ou não. Considera-se uma solução válida, o resultado que possibilite atender a todos os clientes dentro do tempo máximo para atendimento. Para os casos em que não resulte em uma solução que atenda a todos os clientes, escolhe-se aquele que oferecer a melhor solução.

Na função objetivo (Equação 1) são atribuídos pesos e penalizações de acordo com os requisitos dos atendimentos. A variável p na função objetivo consiste no número de penalidades. Caso não exista uma solução válida que compõe o caminho, o mesmo é julgado inválido. Como resultado, receberá uma nota alta para que seja penalizado, diminuindo suas chances de ser selecionado na próxima geração, adotando a seguinte premissa: quanto maior a nota pior o caminho.

O indivíduo, que representa a rota do provedor para o atendimento aos clientes, é composto de uma ordenação entre os clientes solicitantes. A Figura 2(b) apresenta o exemplo de uma possível solução de acordo com a matriz de tempo de deslocamento (Figura 2(a)). Na matriz demonstra-se uma trajetória contendo 4 (quatro) clientes solicitantes e um janelamento de 10 minutos. Para atender a cada cliente, o provedor (P) levou no total 8.15 minutos, definindo assim o *fitness*. Em outras palavras, foi possível atender a todos os clientes dentro do tempo solicitado. A Figura 2(b) representa uma solução válida dado que todos foram atendidos dentro do tempo especificado (Δ_{max}) pelo cliente. Ou seja, não houve penalização.

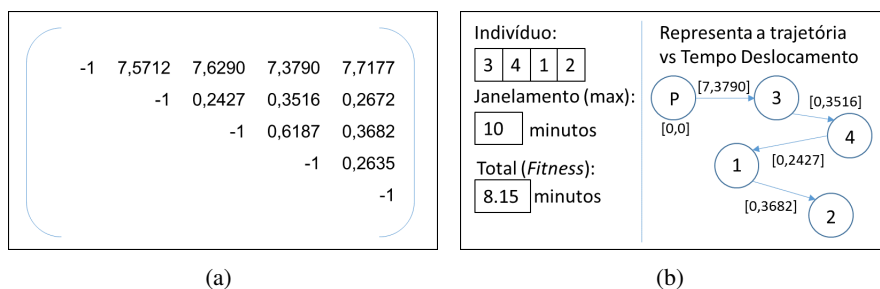


Figura 2: Exemplo da Matriz de Tempo de Deslocamento (a) e Codificação do Indivíduo (b)

O algoritmo AGE analisa o tempo de deslocamento de um provedor até os clientes respeitando o tempo de atendimento (janelamento) de cada cliente. Dessa forma, o AGE deverá evoluir suas soluções candidatas até chegar no seu critério de parada com base no número de gerações. Na implementação do algoritmo optou-se pela escolha das seguintes técnicas: método de torneio para a seleção dos indivíduos; para o cruzamento entre os indivíduos utilizou-se a técnica de permutação de 2 (dois) pontos (*Partially Mapped Crossover (PMX)*). A PMX cria um *crossover* parcialmente mapeado para que não ocorra repetição dos segmentos e; para a mutação nos indivíduos utilizou-se a técnica de mutação aleatória, responsável por mudar a ordem dos genes no indivíduos.



Um provedor, enquanto estiver atendendo as requisições dos clientes (1, 2, 3 e 4), mantém armazenada uma tabela com informações sobre essas requisições. As informações são: identificação da requisição, identificação do nó cliente, o número de sequência da requisição e o valor de *fitness*. Sendo que, o valor atribuído ao *fitness* corresponde ao período no qual a informação será mantida na tabela.

O protocolo LADP descreve e trata os possíveis casos de falhas durante a execução do protocolo em uma MANET. Com a adição do algoritmo de escalonamento de recursos, falhas não previstas na versão original do protocolo podem ocorrer. Considere na Figura 1 que um cliente enviou uma mensagem de reconhecimento da confirmação de serviço para o provedor. Porém, suponha que a mensagem perdeu-se na rede e o nó provedor não a recebe durante o período do temporizador Δ_{AGE} . Neste caso, o provedor enviará uma mensagem (NACK reconhecimento) ao cliente, comunicando o não recebimento da mensagem de reconhecimento da confirmação. Antes de iniciar o deslocamento na direção do primeiro cliente, o provedor fará mais duas tentativas de comunicação com o cliente. Caso receba o reconhecimento da confirmação desse cliente antes de iniciar o deslocamento, o provedor incluirá novamente o cliente no cálculo do *AGE* para definir a sequência de atendimento. Vale destacar, que a execução do *AGE* inicia após o temporizador Δ_{AGE} expirar.

Na ocorrência da falha mencionada acima, e considerando-se que o provedor receba a mensagem de reconhecimento da confirmação do cliente quando já estiver em movimento, definiu-se no *AGE* que o provedor continuará o atendimento na direção do primeiro cliente. No decorrer do primeiro atendimento, o dispositivo móvel no provedor analisará as novas mensagens de invocação 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 aos clientes já selecionados e ao (s) novo (s) clientes anunciando o atendimento.

4. Análise Experimental

Nesta seção são apresentados os resultados experimentais, divididos em três subseções: modelagem interna do algoritmo escalonador de recursos, com o simulador *Network Simulator 3 (NS3)*; definição dos estudos de caso a serem tratados; e resultados computacionais obtidos e análises.

4.1. Configuração e Métricas

Para avaliar o algoritmo de escalonamento de recursos foi utilizado o simulador NS3. Um número de 50 nós foram distribuídos em uma área de 1.000m x 1.000m. O tamanho do pacote da mensagem de invocação é de 64 bytes contendo informações como, localização geográfica do cliente e tempo para o atendimento do pedido. O raio de alcance da antena dos dispositivos é de 250 metros.

Os nós são classificados em: nós clientes, nós intermediários que utilizam o padrão de mobilidade *Gauss-Markov* para simular o movimento de pessoas [Chenghao, 2015] e o nó provedor. Os nós intermediários são móveis e distribuídos conforme o modelo de mobilidade com a velocidade de deslocamento de 1.0 m/s. Os nós clientes são estáticos, enquanto o nó provedor move-se com a velocidade de 1.5 m/s e segue o padrão de mobilidade de *Gauss-Markov*. O tempo total de simulação é de 300s. O temporizador Δ_{AGE} foi ajustado em 20s. O protocolo de roteamento na camada de rede é o *Optimized Link State Routing OLSR* [Kumari et al., 2016].

4.2. Estudos de Caso

O tempo de atendimento estipulado pelos dispositivos móveis dos clientes varia entre 10 e 20 min, e o número de clientes entre 10 e 30. Para avaliar o comportamento dos algoritmos foram realizados ao todo três estudos de caso. Para o primeiro estudo de caso foram realizadas 20 execuções e o tempo de atendimento estabelecidos pelos clientes é de Δ_{max} igual a 20 min. No segundo estudo de caso, o parâmetro Δ_{max} foi ajustado para 10, 15 e 20 minutos. E por fim no terceiro estudo de caso, o valor de Δ_{max} é de no máximo 20 min, no entanto, atribuiu-se randomicamente valores entre 10 e 20 min aos clientes.



As métricas avaliadas compreendem: o número de clientes atendidos dentro do tempo limite e o *fitness* (em relação ao algoritmo genético). O *fitness* leva em consideração o tempo de deslocamento do provedor até o cliente. Foram consideradas 1.000 gerações por não se conhecer o comportamento de um AG neste tipo de aplicação. Para fins de comparação, os resultados obtidos com o algoritmo *AGE* foram comparados com os resultados decorrentes de um método determinístico. Especificamente, o algoritmo do Elevador na versão (*C-LOOK*) [Javed e Khan, 2000]. O *C-LOOK* é um algoritmo de escalonamento de disco e determina a forma com que as cabeças de leitura são posicionadas em função dos pedidos de leitura e escrita realizados.

Alguns parâmetros não são modificados durante os experimentos. Estes são: (i) a Matriz de Tempo de Deslocamento, que é o tempo gasto para chegar a cada cliente a partir do provedor e seus pares adjacentes; (ii) o Tempo de Atendimento (janela), que representa o tempo máximo de atendimento. Este atua no problema como um limitador do espaço de busca das soluções, assegurando que o roteiro válido não ultrapasse o tempo fornecido pelo cliente; (iii) o Algoritmo Genético, no qual o número de execuções = 20, a população = 100, a quantidade de gerações = 1.000, o valor da penalidade $m = 100$, o método para mutação da ordem dos genes = 1% (*SwapMutator*) e o *Partially Mapped Crossover PMX = 90%*.

4.3. Resultados Computacionais

Nos estudos de caso foram obtidos resultados referentes a média e desvio padrão da quantidade de clientes atendidos (Tabela 1) e do tempo de atendimento obtidos em minutos (Tabela 2).

No Estudo de Caso 1, o algoritmo *AGE* obteve a melhor média de atendimento quando comparado ao algoritmo *C-LOOK*. Quando 10 clientes solicitaram o recurso, o *AGE* alcançou o tempo médio de deslocamento de 16.30 min para atender todos os clientes. Neste mesmo cenário, o *C-LOOK* atendeu somente 7 clientes e gastou 19.70 min para deslocamento. A medida que o número de clientes aumenta, o algoritmo *AGE* apontou um melhor desempenho. Por exemplo, com 30 clientes o *AGE* atendeu em média 13 clientes dentro do tempo de 19.08 min. O *C-LOOK* atendeu no máximo 7 clientes em um tempo de 17.03 min.

Tabela 1: Clientes Atendidos

Clientes	Caso 1		Caso 2		Caso 3	
	AGE	C-LOOK	AGE	C-LOOK	AGE	C-LOOK
10	10 ± 0.00	7 ± 0	8 ± 0.00	2 ± 0	8.7 ± 0.67	4 ± 0
20	12.80 ± 0.42	7 ± 0	11.90 ± 0.57	1 ± 0	10.10 ± 0.74	5 ± 0
30	13 ± 0.47	7 ± 0	12.20 ± 0.63	2 ± 0	10.40 ± 0.52	5 ± 0

Tabela 2: Tempo de Atendimento (min)

Clientes	Caso 1		Caso 2		Caso 3	
	AGE	C-LOOK	AGE	C-LOOK	AGE	C-LOOK
10	16.30 ± 0.30	19.70 ± 0	14.88 ± 0.72	7.21 ± 0	15.41 ± 1.54	12.90 ± 0
20	19.25 ± 0.37	19.13 ± 0	18.45 ± 1.07	5.63 ± 0	17.25 ± 0.85	14.52 ± 0
30	19.06 ± 1.98	17.03 ± 0	18.90 ± 0.56	9.08 ± 0	17.51 ± 0.81	14.80 ± 0

Os resultados apresentados no Estudo de Caso 2 mostram o efeito do *AGE* diante de tempos de atendimento diferentes. Os resultados obtidos confirmam a eficácia do algoritmo *AGE* na otimização do tempo de deslocamento e no aumento do número de atendimentos. Em um cenário com 10 clientes, o *AGE* atendeu em média 8 clientes em 14.88 min. Por outro lado, o *C-LOOK* atendeu somente 2 com um tempo médio de 7.21 min. Para o cenário com 20 clientes, observou-se que o *AGE* atendeu em média 11.9 clientes em 18.45 min e o *C-LOOK* atendeu somente 1 com o



tempo de 5.63 min. No cenário com 30 clientes, o *AGE* atendeu uma média de 12.2 clientes em 18.90 min, o *C-LOOK* atendeu 2 clientes em 9.08 min. Atribui-se esse resultado ao fato do provedor ser disposto de acordo com o padrão de mobilidade, resultando em poucos provedores distribuídos na área monitorada.

No Estudo de Caso 3 ilustra-se o tempo de atendimento do provedor em relação as requisições dos clientes. Nesse experimento observou-se que o algoritmo *AGE* atendeu ao maior número de vítimas em relação ao *C-LOOK*. O *AGE* apesar de apresentar desempenho inferior ao mostrado no Estudo de Caso 1, respondeu a uma média de 8.7 clientes em 15.41 min no cenário com 10 clientes. O mesmo comportamento foi observado no experimento com 20 clientes, em que o *AGE* respondeu em média a 10.1 clientes com um tempo de 17.25 min, enquanto que o *C-LOOK* atendeu a 5 clientes em 14.52 min. Outro aspecto significativo desse experimento, consiste no fato de que o número de atendimentos não sofreu redução, mesmo no cenário com 30 clientes, perfazendo uma média de 10.4 clientes no tempo de 17.51 min.

Tomando por base os resultados obtidos, observou-se que o método probabilístico do algoritmo *AGE* demonstrou melhor desempenho quando comparado com o método determinístico utilizado pelo *C-LOOK*. Desse experimento, constatou-se que a medida que o tempo de deslocamento entre os pontos aumenta, o *AGE* atendeu mais clientes em relação algoritmo *C-LOOK*.

Para este experimento foram processadas requisições de 10, 20 e 30 clientes. O tempo de processamento do *AGE* foi em média de 2,19 segundos. Em contrapartida, o *C-LOOK* apresentou uma média de 0.009s. Contudo, o *AGE* atendeu um maior número de clientes e dentro do tempo limite.

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 *AGE*. A atuação do *AGE* teve um comportamento equivalente em todos os casos de estudo e um exemplo pode ser visualizada no gráfico de convergência da Figura (3). O eixo *x* representa o *fitness* e o eixo *y* o número de gerações. A média obtida em cada geração é ilustrada em relação ao custo do tempo de deslocamento até os clientes. O valor da penalidade é aplicado de acordo com a função objetivo representado na Equação 1.

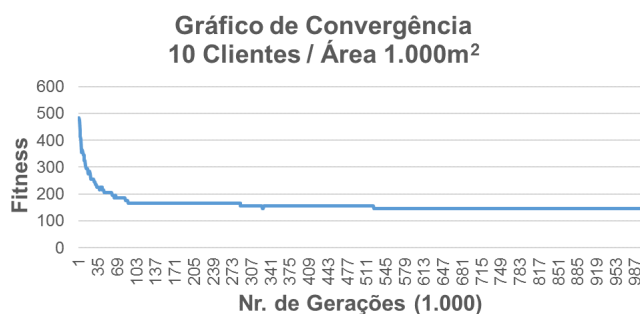


Figura 3: Gráfico de Convergência do Fitness

5. Conclusões

A utilização de um algoritmo genético para resolver o problema de escalonamento de recursos foi motivada pela eficiência dos AGs na solução de problemas combinatórios. Os resultados obtidos através do simulador NS3 demonstraram que os métodos heurísticos são mais adequados para o escalonamento de recursos em cenários de emergência.

No artigo caracterizou-se o escalonamento de recursos no cenário alvo como um problema de permutação, dessa forma identificou-se na literatura que a meta-heurística TSPTW é adequada para uma possível solução do problema. Especificamente, o algoritmo *AGE* foi eficaz diante de



todas as métricas avaliadas, possibilitando que um provedor consiga atender a um maior número de vítimas e dentro do tempo hábil.

A abordagem de comparação é simples, porém viável para a solução do problema. O objetivo é justamente comparar um método heurístico com um determinístico em cenários considerando a quantidade de nós na rede.

Cenários maiores não foram testados pois ao introduzirmos uma janela de tempo no caixa viajante, restringe consideravelmente encontrar uma solução ótima de acordo com a quantidade de nós. Para lidar com essa limitação iremos considerar em um trabalho futuro diferentes medidas de área que possibilite uma análise mais precisa diante da movimentação dos nós e a presença de obstáculos. Outro aspecto que será analisado diz respeito ao caso de falha em que o provedor recebe a mensagem de reconhecimento da confirmação do cliente quando já está em movimento.

Entre os principais pontos a serem trabalhados, a avaliação da sobrecarga e consumo de energia do AGE é uma meta a curto prazo. A médio prazo, pensa-se em avaliar outros métodos probabilísticos e compará-los com o algoritmo genético. Também sugere-se a modelagem do AGE em uma abordagem multi-objetivo considerando a minimização do tempo e a maximização do atendimento.

Referências

- Arenella, G., de Santis, F., e Malandrino, D. (2014). Beeadhoc servicediscovery: A MANET service discovery algorithm based on bee colonies. In *Informatics in Control, Automation and Robotics (ICINCO), 2014 11th International Conference on*, volume 1, p. 244–251. IEEE.
- Chakraborty, D., Joshi, A., Yesha, Y., e Finin, T. (2006). Toward distributed service discovery in pervasive computing environments. *IEEE Transactions on Mobile Computing*, 5:97–112. ISSN 1536-1233.
- Chalotra, S., Sehra, S. K., e Sehra, S. S. (2016). A systematic review of applications of bee colony optimization. In *Innovation and Challenges in Cyber Security (ICICCS-INBUSH), 2016 International Conference on*, p. 257–260. IEEE.
- Chenghao, D. (2015). 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*, p. 6507–6511. IEEE.
- Helal, S., Desai, N., Verma, V., e Lee, C. (2003). Konark - a service discovery and delivery protocol for ad-hoc networks. *Wireless Communications and Networking, WCNC 2003. IEEE*, p. 2107–2113 vol.3.
- Ikeda, M., Kulla, E., Barolli, L., Takizawa, M., e Miho, R. (2011). Performance evaluation of wireless mobile ad-hoc network via NS-3 simulator. In *Network-Based Information Systems (NBIS), 2011 14th International Conference on*, p. 135–141. IEEE.
- Javed, M. Y. e Khan, I. (2000). Simulation and performance comparison of four disk scheduling algorithms. In *TENCON 2000. Proceedings*, volume 2, p. 10–15. IEEE.
- Klein, M., Konig-Ries, B., e Obreiter, P. (2003). Lanes – a lightweight overlay for service discovery in mobile ad hoc networks. Technical Report 2003/6, Universit at Karlsruhe, Faculty of Informatics.
- Kniess, J., Loques, O., e de Albuquerque, C. V. N. (2011). *Descoberta de Serviço em Redes Ad Hoc Móveis*. PhD thesis, Universidade Federal Fluminense (UFF).
- Kniess, J., Loques, O., e de Albuquerque, C. V. N. (2015). Service discovery with time constraints in mobile ad hoc networks. *Earth Science Informatics*, 8(2):439–452. URL <http://dx.doi.org/10.1007/s12145-014-0164-4>.



- Kumari, N., Gupta, S. K., Choudhary, R., e Agrwal, S. L. (2016). New performance analysis of AODV, DSDV and OLSR routing protocol for MANET. In *Computing for Sustainable Global Development (INDIACom), 2016 3rd International Conference on*, p. 33–35. IEEE.
- Lenders, V., May, M., e Plattner, B. (2005). Service discovery in mobile ad hoc networks: A field theoretic approach. *Pervasive Mob. Comput.*, 1:343–370. ISSN 1574-1192. URL <http://dx.doi.org/10.1016/j.pmcj.2005.06.001>.
- Mian, A. N., Baldoni, R., e Beraldi, R. (2009). A survey of service discovery protocols in multihop mobile ad hoc networks. *IEEE Pervasive Computing*, 8(1):66–74.
- Park, J. C., Choi, M. S., Lee, B. J., Kim, K. T., e Youn, H. Y. (2013). Distributed semantic service discovery for MANET. In *Ubiquitous Intelligence and Computing, 2013 IEEE 10th International Conference on and 10th International Conference on Autonomic and Trusted Computing (UIC/ATC)*, p. 515–520. IEEE.
- Parpinelli, R. S. e Lopes, H. S. (2011). New inspirations in swarm intelligence: a survey. *International Journal of Bio-Inspired Computation*, 3(1):1–16.
- Ratsimor, O., Chakraborty, D., Joshi, A., Finin, T., e Yesha, Y. (2004). Service discovery in agent-based pervasive computing environments. *Mobile Networks and Applications*, 9:679–692. ISSN 1383-469X. URL <http://dx.doi.org/10.1145/1035715.1035728>.
- Saha, H. N., Chattopadhyay, A., e Sarkar, D. (2015). Review on intelligent routing in MANET. In *Computing and Communication (IEMCON), 2015 International Conference and Workshop on*, p. 1–6. IEEE.
- Serhani, M. A. e Gadallah, Y. (2010). A service discovery protocol for emergency response operations using mobile ad hoc networks. In *Telecommunications (AICT), 2010 Sixth Advanced International Conference on*, p. 280–285. IEEE.
- Shukla, A., Pandey, H. M., e Mehrotra, D. (2015). Comparative review of selection techniques in genetic algorithm. In *Futuristic Trends on Computational Analysis and Knowledge Management (ABLAZE), 2015 International Conference on*, p. 515–519. IEEE.
- Varghese, B. M. e Raj, R. J. S. (2016). A survey on variants of genetic algorithm for scheduling workflow of tasks. In *Science Technology Engineering and Management (ICONSTEM), Second International Conference on*, p. 489–492. IEEE.
- Varshavsky, A., Reid, B., e de Lara, E. (2005). A cross-layer approach to service discovery and selection in MANETs. *IEEE International Conference on Mobile Adhoc and Sensor Systems Conference*, 0:466.
- Xiong, C. e Qidi, W. (2002). Formulating the steel scheduling problem as a TSPTW. In *Intelligent Control and Automation, 2002. Proceedings of the 4th World Congress on*, volume 3, p. 1744–1748. IEEE.