

# Alocação de controladores SDN para gerenciamento de comunicações D2D

Yarisley Peña Llerena e Paulo Roberto L. Gondim

**Resumo**—Para emprego da rede *LTE-Advanced* (LTE-A), a comunicação “device-to-device” é uma tecnologia candidata a permitir um adequado uso de recursos, considerando perspectiva baseada na localização do usuário e em serviços baseados em proximidade. Por outro lado, redes definidas por software favorecem a melhoria da eficiência do gerenciamento e controle dos recursos de rede. Este artigo trata o problema de alocação de controladores SDN visando o adequado gerenciamento de comunicações D2D em uma rede LTE-A. Para solução desse problema, foi adotada abordagem baseada em memória externa do algoritmo de colônia de formigas, com resultados similares aos de busca exaustiva.

**Palavras-Chave**—D2D, SDN, colônia de formigas, alocação de controladores

**Abstract**—For the use of the *LTE-Advanced* network (LTE-A), device-to-device communication is a candidate technology to allow an adequate use of resources, considering user location and proximity-based services perspective. On the other hand, software-defined networks improve the efficiency of the management and control of network resources. This article addresses the problem of allocating SDN controllers for the proper management of D2D communications over an LTE-A network. To solve this problem, the ant colony algorithm based on external memory approach was adopted, with results similar to exhaustive search.

**Keywords**—D2D, SDN, ant colony, controller allocation

## I. INTRODUÇÃO

O tráfego de dados móveis tem aumentado nos últimos anos e, com o advento da Internet das Coisas (IoT, do inglês *Internet of Things*) cada dispositivo será conectado à Internet, produzindo dados oriundos de processos de sensoriamento. O tráfego de dados continuará crescendo a uma taxa exponencial, o que faz com que a próxima geração de redes de telefonia móvel (5G) incorpore necessidades como a disponibilidade de vários caminhos de transferência, a diminuição da latência e alta disponibilidade, entre outros [1]. Para satisfazer os desafios mencionados estão surgindo novas tecnologias como comunicações dispositivo a dispositivo (D2D, do inglês *device-to-device*) e redes definidas por software (SDN, do inglês *software-defined networks*).

A comunicação D2D é definida como uma comunicação direta entre dois dispositivos móveis sem passar pela estação base (BS) ou pelo núcleo da rede [2]. A comunicação D2D ocorre de forma direta (par a par) na transmissão de dados, enquanto o estabelecimento e gerenciamento da conexão pode ser feito pelos dispositivos móveis ou através da infraestrutura celular. Neste trabalho é utilizada uma abordagem onde o

gerenciamento da comunicação é realizado por uma entidade localizada na rede celular e que pode obter informações tanto da rede celular quanto das comunicações D2D, o que permite, por exemplo, executar escoamento de dados (*offloading*) e/ou tratar a mitigação da interferência. Considera-se que uma comunicação D2D será feita utilizando o espectro celular (*Inband*), como ocorre na tecnologia *LTE Direct*.

Por outro lado, conceitos relacionados com SDN têm sido explorados no sentido de aumentar a eficiência no gerenciamento e controle da rede. Em redes SDN o plano de controle e o plano de dados são separados, ficando toda a inteligência da rede em um ponto central chamado controlador SDN. Com isso, se consegue uma maior flexibilidade na administração das entidades de retransmissão (como eNBs - *evolved NodeBs* ou switches), permitindo que exista programabilidade o que propicia a adequação de novos serviços de rede.

Nossa proposta emprega controladores SDN, para gerenciar as comunicações D2D que acontecem na área de cobertura de cada eNB. Assumimos que os controladores são localizados numa infraestrutura celular *LTE-Advanced* (LTE-A).

O emprego de múltiplos controladores SDN na rede LTE-A leva a dois problemas: (i) obter o número de controladores necessários para gerenciar adequadamente todas as comunicações D2D e (ii) definir como realizar a alocação das eNBs aos controladores SDN. Para encontrar uma solução modelamos os problemas anteriores como um problema de otimização, onde pretendemos determinar o número mínimo de controladores SDN e a melhor atribuição balanceada de eNBs a esses controladores, além de satisfazer um tempo de resposta menor ou igual a um QoS específico.

O problema de otimização antes mencionado pode ser entendido como um problema de otimização combinatória, semelhante ao problema de “localização de facilidades - *facility location problem*”, considerado um problema NP-difícil [3], e para o qual gerar uma solução ótima é uma tarefa exaustiva e demorada. Nesse cenário, as metaheurísticas representam uma alternativa para explorar o espaço de pesquisa e fornecer uma solução próxima da ótima.

Neste artigo, utilizamos uma abordagem baseada em memória externa conhecida como *Ant Colony System with external memory* (ACS-EM) [4] para encontrar o número mínimo de controladores SDN, assim como a melhor atribuição de eNB aos controladores. Investigamos o problema de alocação levando em consideração a latência de processamento nos controladores SDN dentro do tempo de resposta. Como em [1] e [5], modelamos o processamento dos controladores aplicando teoria de filas e a taxa de solicitação recebida pelos controladores SDN envolve o tráfego de controle das

comunicações D2D.

No algoritmo ACS-EM, o caminho percorrido pelas formigas permite determinar a atribuição das eNBs e o número mínimo de controladores necessários a um adequado nível de QoS provido pela rede LTE-A.

A principal contribuição deste estudo envolve o uso do algoritmo colônia de formigas com memória externa para resolver o problema de otimização modelado, sujeito à restrição de tempo de resposta do controlador SDN.

O restante deste artigo está assim organizado: a seção II apresenta a modelagem do sistema; a seção III apresenta a definição do problema de otimização; a seção IV descreve o algoritmo ACS-EM; a seção V analisa os resultados obtidos pelo algoritmo, em comparação com busca exaustiva; e finalmente, a Seção VI apresenta as conclusões e trabalhos futuros.

## II. MODELO DO SISTEMA

Consideramos uma rede celular LTE-A formada por um PGW (*Packet Data Network Gateway*), um conjunto  $S = \{1, 2, \dots, s\}$  de SGWs (*Serving Gateways*), um conjunto  $B = \{1, 2, \dots, b\}$  de eNBs e um conjunto  $C = \{1, 2, \dots, c\}$  de controladores SDN. Uma representação simplificada de nossa topologia é mostrada na Figura 1. Assumimos que as entidades SGWs e eNBs se comunicam utilizando enlaces cabeados e que as eNBs são igualmente distribuídas entre as SGWs.

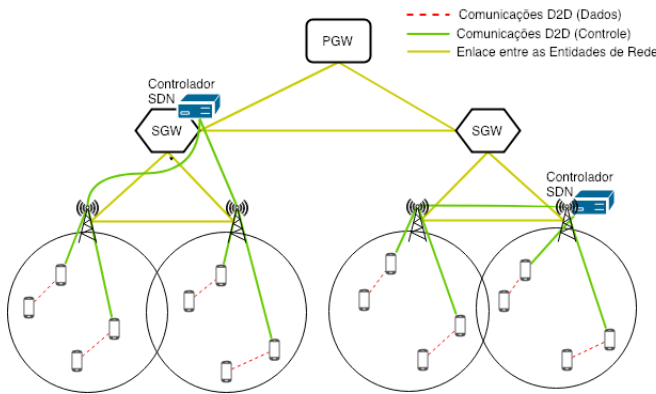


Fig. 1. Rede celular SDN para gerenciar comunicações D2D.

No sistema consideramos a existência de um conjunto  $N = \{1, 2, \dots, n\}$  de UEs, com capacidades D2D, uniformemente distribuídos entre as eNBs. Os UEs são terminais multimodo que podem estabelecer comunicações utilizando tanto a rede celular quanto um enlace direto, utilizando *LTE Direct*. Assim, cada par de UEs, localizados na área de cobertura da mesma eNB, podem estabelecer uma comunicação direta entre eles. Essas comunicações geram tráfego de controle aos controladores SDN com uma taxa de requisições média ( $\lambda$ ), que seguem uma distribuição de Poisson.

A taxa de requisições recebidas por um controlador  $c_i$  pode ser definida como  $\lambda_{c_i} = \sum_{j=1}^{Q_{c_i}} \lambda_{b_j}$ . Onde  $Q_{c_i}$  é a quantidade de eNBs gerenciadas pelo controlador  $c_i$  e  $\lambda_{b_j}$  é a taxa de requisições de controle gerada pela  $j$ th eNB, a qual depende da quantidade de comunicações D2D estabelecidas na sua área de cobertura ( $C_{D2D}$ ) e está dada por  $\lambda_{b_j} = C_{D2D}^j * \lambda$ .

$C_{D2D}$  é calculado durante a simulação após de definir aleatoriamente a localização dos usuários. Para um par comunicante estabelecer um enlace direto entre eles devem estar localizados dentro da área de cobertura *LTE Direct*.

### A. Modelo M/M/m para controladores SDN multi-core

O modelo markoviano multiservidor com fila infinita e disciplina FIFO foi considerado para modelagem dos controladores SDN. Tais controladores, usados na rede celular, são executados em máquinas virtuais (VMs) com quatro núcleos [1], para os quais definimos os seguintes parâmetros:

$\lambda_c$  : taxa média de requisições recebidas pelo controlador SDN;

$\mu$  : taxa média de processamento de cada núcleo, segue uma distribuição exponencial;

$m$  : quantidade de núcleos do controlador ( $m = 4$ ).

O tempo médio de resposta de cada controlador SDN ( $t_c$ ) é o tempo gasto pelo pacote na fila mais o tempo de processamento, que pode ser expresso por [6]:

$$t_c = \frac{\rho}{1 - \rho} A(m, \lambda_c / \mu) + m\rho \quad (1a)$$

$$A(m, \lambda_c / \mu) = \frac{1}{1 + (1 - \rho) \left( \frac{m!}{(m\rho)^m} \right) \sum_{i=0}^{m-1} \frac{(m\rho)^i}{i!}} \quad (1b)$$

$$\rho = \frac{\lambda_c}{m\mu} \quad (1c)$$

## III. DEFINIÇÃO DO PROBLEMA

O problema aqui tratado caracteriza-se pela necessidade de encontrar o número mínimo de controladores SDN necessários para gerenciar as comunicações D2D, realizar a melhor atribuição balanceada das eNBs a esses controladores e satisfazer um tempo de resposta menor ou igual a um tempo de QoS específico. Assim, a função objetivo ( $OF$ , equação 2) considera três termos: (i) o número de controladores SDN; (ii) a função que garante um adequado balanceamento na atribuição e (iii) a função que garante um tempo de resposta dos controladores ( $t_c$ ) menor ou igual que o  $t_{QoS}$  considerado.

$$OF = \gamma_c \cdot C + \gamma_F \cdot F + \gamma_T \cdot T_c \quad (2)$$

Na equação 2,  $\gamma_c$ ,  $\gamma_F$  e  $\gamma_T$  representam pesos que permitem denotar a importância de cada termo da função objetivo. Para minimizar o número de controladores  $c$ , considera-se que  $C = 1 - 1/c$ . A função  $F$  está baseada no índice de Jain's (*Jain's Fairness Index*) [7] para alcançar um adequado nível de justiça no processo de atribuição das eNBs aos controladores e é obtida como:

$$F = 1 - \frac{\left( \sum_{i=1}^c \rho_i \right)^2}{c \cdot \sum_{i=1}^c \rho_i^2} \quad (3)$$

onde  $\rho_i$  representa o grau de utilização do  $i$ -ésimo controlador, calculado como na equação 1c.

Por último, a função  $T_c$  representa o tempo de resposta médio dos controladores SDN, obtido como:

$$T_c = \frac{1}{c} \sum_{i=1}^c T(t_{c_i}) \quad (4)$$

onde  $t_{c_i}$  é o tempo de resposta médio do  $i$ -ésimo controlador (equação 1a) e  $T(t_{c_i})$  é uma função de custo que permite minimizar o tempo de resposta dos controladores considerando um tempo de QoS específico ( $t_{QoS}$ ), definida como:

$$T(t_{c_i}) = \begin{cases} e^{(t-t_{QoS})}, & t \leq t_{QoS} \\ e^{\omega(t-t_{QoS})}, & t > t_{QoS} \end{cases} \quad (5)$$

onde  $\omega$  é um fator constante que permite definir a taxa de crescimento ou decrescimento da função de custo. Após várias simulações definimos  $\omega = 2t_{QoS}$  como o valor que melhor resultado apresentou na resolução do problema tratado neste artigo. Quando  $t \leq t_{QoS}$  a função  $T(t_{c_i}) \in (0, 1]$  com uma taxa de crescimento suave. Porém, quando  $t > t_{QoS}$  o valor da função de custo tem um crescimento abrupto sendo  $T(t_{c_i}) > 1$ , isto provoca que essas soluções não resultem escolhidas pela meta-heurística.

#### IV. ALGORITMO COLÔNIA DE FORMIGAS COM MEMÓRIA EXTERNA (ACS-EM)

Nos algoritmos de otimização baseados em colônias de formigas (ACO, do inglês *Ant Colony Optimization*) [8], as formigas artificiais constroem uma solução para um problema combinatório percorrendo um grafo chamado 'grafo de construção',  $G_C(V, E)$ . O grafo de construção consiste em um conjunto de vértices  $V$  e um conjunto de arcos  $E$ . No problema de atribuição das eNB aos controladores, o conjunto de vértices  $V = \{b_1c_1, \dots, b_1c_n, \dots, b_nc_1, \dots, b_nc_n\}$  representa todos os pares possíveis de eNB - Controlador SDN que existem no cenário considerado.

Para obter o número de controladores SDN utilizados, o grafo é construído considerando que é preciso um controlador SDN para gerenciar o tráfego de cada eNB, sendo  $c = b$ . Depois, quando as formigas chegam ao final do caminho são eliminados da solução os controladores SDN que não tiveram atribuído nenhuma eNB, obtendo assim o número de controladores realmente utilizados.

Neste artigo, examina-se a possibilidade de alternar a maneira pela qual os componentes da solução são escolhidos, introduzindo uma memória externa como um mecanismo auxiliar para tomar as decisões em cada etapa da construção de uma solução. Usamos uma memória baseada na frequência [4]. Esse tipo de memória armazena componentes das soluções que são escolhidas com maior frequência.

O algoritmo tem associado uma matriz de frequência de tamanho  $[B, C]$ , chamada *memory\_freq*, onde a posição *memory\_freq* $[b_i, c_j]$  armazena o número de vezes que a eNB  $b_i$  foi atribuída ao controlador  $c_j$  durante a execução do algoritmo. Isso representa os vértices que são escolhidos pelas formigas na construção da solução.

Durante a construção de uma solução, as formigas aplicarão uma regra de escolha de ação, semelhante à usada no algoritmo *ant colony system* (ACS) [8]. É introduzido um parâmetro  $q_0 \in [0, 1]$ , que permitirá escolher um componente de solução que favoreça a intensificação ou a diversificação na busca de soluções.

Quando uma formiga tem que escolher um componente de uma solução, ela primeiro gera um número aleatório uniformemente distribuído no intervalo  $[0, 1]$ , se esse número for menor que o valor do parâmetro  $q_0$  um componente é escolhido para realizar uma busca local; se o número gerado for maior que o parâmetro  $q_0$ , o componente é escolhido favorecendo a busca de novas soluções. Além disso, cada uma dessas opções, por sua vez, inclui uma decisão sobre rastros de feromônios ou memória externa. Logo, para a escolha de um vértice  $v_j$  a partir de um vértice  $v_i$  é utilizada a regra a seguir:

$$v_j = \begin{cases} E_T, & \text{se } q < q_0 \text{ (Intensificação)} \\ E_R, & \text{se } q \geq q_0 \text{ (Diversificação)} \end{cases} \quad (6)$$

Na intensificação de soluções é utilizada a seguinte regra:

$$E_T = \begin{cases} \arg \max_{l \in N_i^k} \{memory\_freq[b_l, c_l]\}, & \text{se } r < r_0 \\ \arg \max_{l \in N_i^k} \{\tau_{il} \cdot \eta_{il}^\beta\}, & \text{se } r \geq r_0 \end{cases} \quad (7)$$

onde  $N_i^k$  é a vizinhança alcançável pela formiga  $k$  quando se encontra no nó  $v_i$ ,  $\tau_{il}$  é a quantidade de feromona entre os nós  $v_i$  e  $v_l$ ,  $\eta_{il}$  é a informação heurística ou deseabilidade entre os nós  $v_i$  e  $v_l$  e  $\beta$  é um parâmetro que define a importância relativa da informação heurística. Na regra, com uma probabilidade fixa  $r_0 (0 \leq r_0 \leq 1)$  é escolhido aquele elemento que foi mais frequentemente selecionado a partir da localização atual; enquanto com probabilidade  $(1 - r_0)$  o elemento mais desejável é escolhido de acordo com o traço de feromônio. Onde  $r$  é uma variável aleatória uniformemente distribuída no intervalo  $[0, 1]$ . Na exploração de novas soluções (diversificação) é utilizada a seguinte regra:

$$E_R = \begin{cases} \arg \min_{l \in N_i^k} \{memory\_freq[b_l, c_l]\}, & \text{se } p < p_0 \\ S, & \text{se } p \geq p_0 \end{cases} \quad (8)$$

Nesta regra, com uma probabilidade fixa  $p_0 (0 \leq p_0 \leq 1)$  é selecionado o elemento que foi menos vezes escolhido a partir da localização atual; enquanto com probabilidade  $(1 - p_0)$ , o elemento  $S$  é escolhido de acordo com a regra de seleção básica, similar àquela do algoritmo ACS [8]. Onde  $p$  é uma variável aleatória distribuída uniformemente no intervalo  $[0, 1]$ .

$S$  é uma variável aleatória, selecionada utilizando o método da roleta, de acordo com a distribuição de probabilidade  $p_k$  similar à do algoritmo *Ant System* [8]:

$$p_k(i, l) = \begin{cases} \frac{\tau_{il}^\alpha \cdot \eta_{il}^\beta}{\sum_{u \in N_i^k} \tau_{iu}^\alpha \cdot \eta_{iu}^\beta}, & \text{se } l \in N_i^k \\ 0, & \text{de outra forma} \end{cases} \quad (9)$$

onde  $\alpha$  é o parâmetro que estabelece a importância relativa dos rastros de feromona. A heurística  $\eta_{il}$  é obtida a partir da taxa de utilização dos controladores,  $\rho_{c_i}$ , considerando as atribuições de eNB já realizadas nos componentes do caminho percorrido. Assim, os controladores com menor valor de  $\rho$  terão uma maior desejabilidade.

A atualização dos rastros de feromona é realizada em duas etapas, igualmente ao algoritmo ACS: (i) cada vez que uma formiga atravessa um arco, chamada de atualização local e (ii) ao final de cada iteração, chamada atualização global.

A regra de atualização de feromona local (também referida como *online*) é aplicada por todas as formigas cada vez que atravessa um arco  $(i, j)$  durante a construção da solução, calculada como:

$$\tau_{ij} = (1 - \sigma) \cdot \tau_{ij} + \sigma \cdot \tau_0 \quad (10)$$

onde  $\tau_0$  é o valor de feromona inicial e  $\sigma \in (0, 1)$  é o coeficiente de evaporação de feromona.

A regra de atualização passo-a-passo, *online*, inclui tanto a evaporação de feromônio como a deposição de feromônio. Como a quantidade de feromônio depositada é muito pequena, a aplicação dessa regra faz com que traços de feromônios entre os caminhos percorridos pelas formigas diminuam. Assim, isso leva a uma técnica de exploração adicional do algoritmo, uma vez que os caminhos percorridos por um grande número de formigas vão se tornando menos atraentes para o resto das formigas que os atravessam na iteração atual, o que claramente ajuda que nem todas as formigas sigam o mesmo caminho.

Na atualização global dos rastros de feromona só é atualizado o caminho percorrido pela formiga com a melhor solução (a formiga *best-so-far*) após cada iteração. Essa atualização é implementada utilizando a seguinte equação:

$$\tau_{ij} = \begin{cases} (1 - \sigma)\tau_{ij} + \sigma\Delta\tau_{ij}^{bs}, & \text{se } j \in \Omega \\ (1 - \sigma)\tau_{ij}, & \text{se } j \notin \Omega \end{cases} \quad (11)$$

onde  $\Omega = \{1, 2, \dots, I\}$  é o conjunto de nós que pertencem ao caminho percorrido pela formiga *best-so-far* e  $\Delta\tau_{ij}^{bs}$  é o incremento nos rastros de feromona expressado como:

$$\Delta\tau_{ij}^{bs} = \frac{1}{OF^{bs}} \quad (12)$$

onde  $OF^{bs}$  é o valor da função objetivo (equação 2) da melhor solução da iteração.

O pseudocódigo do algoritmo ACS-EM para solucionar o problema de alocação dos controladores SDN é apresentado no Algoritmo 1.

## V. AVALIAÇÃO DO DESEMPENHO

Nesta seção, apresentamos os resultados obtidos usando o algoritmo ACS-EM na alocação dos controladores SDN, envolvendo o número de controladores SDN necessários para gerenciar as comunicações D2D que estão acontecendo em uma rede celular LTE-A, em um cenário com 1 PGW, 2 SGW e 6 eNB. Comparamos o algoritmo ACS-EM com um algoritmo de busca exaustiva que utiliza a técnica *Backtracking*, com o

### Algorithm 1 ACS-EM

---

```

1: Input:  $\tau_0, \alpha, \beta, q_0, r_0, p_0, \sigma, ant\_quantity, iteration\_max$ 
2:  $\tau[\cdot] = \text{InitPheromone}()$ 
3:  $memory[\cdot] = \text{InitMemory}()$ 
4: for  $i = 0$  to  $iteration\_max$  do
5:   for  $a = 0$  to  $ant\_quantity$  do
6:      $currentPos \leftarrow \text{RootNode}$ 
7:      $TourList \leftarrow \text{currentPosition}$ 
8:     while  $filho \neq 0$  do
9:        $\eta[\cdot] = \text{CalculateChildrenHeuristic}()$ 
10:       $p_k[\cdot] = \text{CalculateChildrenProbability}()$   $\triangleright$  eq. 9
11:       $q \leftarrow \text{RandomDouble}()$ 
12:      if  $q < q_0$  then  $\triangleright E_T$  - Exploração
13:         $r \leftarrow \text{RandomDouble}()$ 
14:        if  $r < r_0$  then  $\triangleright$  Memória externa
15:           $nextPos \leftarrow \text{arg\_máx}\{memory[filho]\}$ 
16:        else  $\triangleright$  probabilístico ACS
17:           $nextPos \leftarrow \text{arg\_máx}\{\tau[filho] * \eta[filho]^\beta\}$ 
18:        end if
19:      else  $\triangleright E_R$  - Exploração
20:         $p \leftarrow \text{RandomDouble}()$ 
21:        if  $p < p_0$  then  $\triangleright$  Memória externa
22:           $nextPos \leftarrow \text{arg\_mín}\{memory[filho]\}$ 
23:        else  $\triangleright$  probabilístico ACS
24:           $nextPos \leftarrow \text{Get}_S(p_k)$ 
25:        end if
26:      end if
27:      Adicionar  $nextPosition$  ao  $TourList$ 
28:       $\text{ValidateFeasiblePath}(TourList)$ 
29:       $currentPos \leftarrow nextPos$ 
30:       $\text{LocalPheromoneUpdate}()$   $\triangleright$  eq. 10
31:    end while
32:     $\text{CalculateAntsTourUtility}()$   $\triangleright$  eq. 2
33:     $\text{OfflinePheromoneUpdate}(\text{BestTour})$   $\triangleright$  eq. 11
34:     $memory[\cdot] = \text{UpdateMemory}(\text{BestTour})$ 
35:  end for
36: end for
Output:  $\text{GlobalBestTour}$ 

```

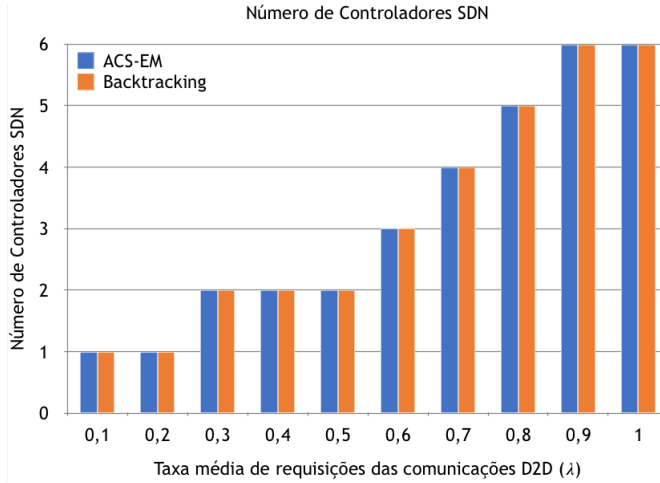
---

objetivo de validar sua eficácia na solução do problema em questão. Os dois algoritmos foram implementados em C#.

Após testes e ajustes, definimos como valores para os parâmetros de simulação do algoritmo ACS-EM:  $ant\_quantity = 15$ ;  $q_0 = 0, 4$ ;  $r_0 = 0, 6$ ;  $p_0 = 0, 6$ ;  $\tau_0 = 0, 1$  e  $\sigma = 0, 1$ . Para obter melhores resultados do algoritmo ACS-EM no início da simulação daremos mais importância à informação heurística e no fim aos rastros de feromônio, então o valor de  $\beta$  começa em 1 e diminui em cada iteração, enquanto o valor de  $\alpha$  começa em 0, 01 e aumenta durante a simulação.

Para executar a simulação, definimos os parâmetros do Modelo do Sistema como: densidade de usuários: 10 *usuários/km<sup>2</sup>*;  $t_{QoS} = 8$  ms; número de iterações: 500; raio de cobertura das eNB: 5000 m; raio de cobertura LTE Direct: 500 m;  $\mu = 175$  req/ms;  $\gamma_c = 0.4$ ;  $\gamma_F = 0.3$  e  $\gamma_T = 0.3$ .

Como resultados da simulação foram obtidos: número de controladores SDN, atribuição das eNBs aos controladores SDN e o valor da função objetivo ( $OF$ ). Na Figura 2 pode-se observar que com o aumento da taxa de requisições de controle das comunicações do D2D, o número de controladores aumenta. O aumento de tráfego de controle nas eNBs aumenta a carga nos controladores SDNs, aumentando o tempo médio de resposta. Portanto, para compensar esse aumento e satisfazer o valor de  $t_{QoS} = 8$  ms, definido nas restrições do problema,

Fig. 2. Impacto de  $\lambda$  no número de controladores SDN.TABELA I  
ATRIBUIÇÃO DE eNBs AOS CONTROLADORES SDN.

$\lambda$ (pkt/s)	Algoritmo	eNB-Controlador	$OF$
0.9	ACS-EM	$C_0 : b_2, C_1 : b_3, C_2 : b_1, C_3 : b_5, C_4 : b_0, C_5 : b_4$	0.3343
	Backtracking	$C_0 : b_0, C_1 : b_1, C_2 : b_2, C_3 : b_3, C_4 : b_4, C_5 : b_5,$	0.3343
0.8	ACS-EM	$C_0 : b_5, C_1 : b_1, C_2 : b_0, C_3 : b_2, C_4 : b_4, b_3,$	0.3341
	Backtracking	$C_0 : b_0, C_1 : b_1, C_2 : b_2, C_3 : b_3, C_4 : b_4, b_5,$	0.3341
0.7	ACS-EM	$C_0 : b_4, b_5, C_1 : b_0, b_3, C_2 : b_1, C_3 : b_2,$	0.2795
	Backtracking	$C_0 : b_0, b_3, C_1 : b_1, b_2, C_2 : b_4, C_3 : b_5,$	0.2795
0.6	ACS-EM	$C_0 : b_0, b_5, C_1 : b_1, b_2, C_2 : b_3, b_4,$	0.2695
	Backtracking	$C_0 : b_0, b_2, C_1 : b_1, b_3, C_2 : b_4, b_5,$	0.2695
0.4	ACS-EM	$C_0 : b_3, b_4, b_5, C_1 : b_0, b_1, b_2,$	0.2028
	Backtracking	$C_0 : b_0, b_1, b_2, C_1 : b_3, b_4, b_5,$	0.2028
0.2	ACS-EM	$C_0 : b_0, b_1, b_2, b_3, b_4, b_5,$	0.0028
	Backtracking	$C_0 : b_0, b_1, b_2, b_3, b_4, b_5,$	0.0028

é necessário utilizar um número maior de controladores SDN.

Na Tabela I são apresentados os resultados da alocação das eNBs aos controladores, para os algoritmos ACS-EM e de busca exaustiva, mostrando uma alocação balanceada entre os controladores SDN e que, com o aumento de  $\lambda$ , um maior número de controladores é necessário. Os valores da função objetivo mostram que a meta-heurística ACS-EM apresentou bons resultados, obtendo os valores ótimos para o cenário considerado, tomando por base a técnica de *Backtracking*.

## VI. CONCLUSÕES

Neste artigo, o problema de alocação de eNBs aos controladores SDN foi analisado para uma arquitetura LTE-A, com gerenciamento de comunicações D2D. Propomos uma abordagem baseada em memória externa para a metaheurística *ant colony system* na resolução do problema de otimização modelado. Os objetivos foram, minimizar o número de controladores SDN, minimizar o tempo de resposta considerando um tempo de QoS específico e realizar uma atribuição balanceada das eNBs aos controladores SDN. O algoritmo ACS-EM apresentou resultados aceitáveis e precisos nas simulações realizadas, em comparação com um algoritmo de busca exaustiva.

Como continuidade deste trabalho, está em andamento uma pesquisa que considera o efeito de redes sociais como Facebook e Twitter no estabelecimento de comunicações D2D. Adicionalmente, pretendemos considerar o problema de localização dos controladores na rede celular, minimizando a latência eNB-controlador e considerando métricas como confiabilidade e distância entre controladores. Por fim, a extensão do trabalho para focar de forma mais específica em alocação de recursos deverá ser realizada, considerando meta-heurísticas (colônia de abelhas artificiais e busca tabu) e técnicas de solução exata.

## AGRADECIMENTOS

Este trabalho foi financiado, parcialmente, pela Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES) por meio da concessão da bolsa de estudos, através do programa de Pós-graduação em Engenharia Elétrica - PPGEE da Universidade de Brasília, à doutoranda Yarisley Peña Llerena.

## REFERÊNCIAS

- [1] A. Farshin and S. Sharifian, "A chaotic grey wolf controller allocator for software defined mobile network (sdmn) for 5th generation of cloud-based cellular systems (5g)," *Computer Communications*, vol. 108, pp. 94 – 109, 2017.
- [2] A. Asadi and V. Mancuso, "WiFi Direct and LTE D2D in action," *2013 IFIP Wireless Days (WD)*, pp. 1–8, Nov 2013.
- [3] K. S. Sahoo, S. Sahoo, A. Sarkar, B. Sahoo, and R. Dash, "On the placement of controllers for designing a wide area software defined networks.," *TENCON 2017 - 2017 IEEE Region 10 Conference*, pp. 3123–3128, 2017.
- [4] F. L. A. Arito and S. Luis-Argentina, "Algoritmos de Optimización basados en Colonias de Hormigas aplicados al Problema de Asignación Cuadrática y otros problemas relacionados," *Universidad Nacional de San Luis Facultad de Ciencias Físico Matemáticas y Naturales Departamento de Informática, San Luis-Argentina Abril de*, 2010.
- [5] M. J. Abdel-Rahman, E. A. Mazied, K. Teague, A. B. MacKenzie, and S. F. Midkiff, "Robust controller placement and assignment in software-defined cellular networks," *2017 26th International Conference on Computer Communication and Networks (ICCCN)*, pp. 1–9, July 2017.
- [6] J. F. Shortle, J. M. Thompson, D. Gross, and C. M. Harris, *Fundamentals of Queueing Theory*. New Jersey, USA: John Wiley & Sons, 5th ed., 2018.
- [7] R. Jain, D.-M. Chiu, and W. R. Hawe, "A quantitative measure of fairness and discrimination for resource allocation in shared computer system," *DEC technical report TR301*, vol. cs.NI/9809, no. DEC-TR-301, pp. 1–38, 1984.
- [8] M. Dorigo and T. Stützle, *Ant Colony Optimization: Overview and Recent Advances*, pp. 311–351. Cham: Springer International Publishing, 2019.