Received November 1, 2019, accepted November 19, 2019, date of publication November 22, 2019,
date of current version December 6, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2955434

SDN-Controller Placement for D2D Communications

YARISLEY PEÑA LLERENA, (Member, IEEE),
AND PAULO R. L. GONDIM, (Senior Member, IEEE)
Department of Electrical Engineering, Brasília University (UnB), Brasília 70910-900, Brazil

Corresponding author: Yarisley Peña Llerena (yarisleyllerena@aluno.unb.br)

This work was supported in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES).

ABSTRACT The increasing demand for data transmission resources to be handled by next generation cellular networks has led to the emergence of new technologies, such as Device-to-Device (D2D) communications and software-defined networks (SDN). D2D expands the use of resources from a location perspective and SDN enables an efficient management and control of the available resources. This article addresses a study of the influence of D2D communications management on a Long Term Evolution (LTE) network with SDN controllers regarding the obtaining of their necessary number and best location within the cellular infrastructure. The controller placement problem was modeled as an optimization problem and solved by the ant colony system with external memory (ACS-EM) algorithm. The proposed algorithm was compared with a particle swarm optimization (PSO)-based algorithm and its effectiveness has been validated.

INDEX TERMS Ant colony system with external memory, controller placement problem, D2D, software defined networking.

I. INTRODUCTION

Cellular networks have evolved according to the population's needs. As such, new forms of evolution in telecommunications have changed the current paradigms.

Fifth-generation communications or 5G Systems have been designed to offer both more advanced and complex configurations aimed at better performance to meet the requirements of society and a new form in the thinking and components that comprise them. Such a technology will provide unlimited access to information, as well as availability for its sharing anywhere, at any time, by any person or thing, for the benefit of society [1].

5G communications are characterized by numerous devices and interconnected networks and an increase in the data traffic in comparison to the current one. Other characteristics include seamless integration of heterogeneous networks, use of femtocells and increased capacity and performance with reduced latency [2], [3]. Moving networks, ultra-dense networks, device to device (D2D) communications, ultra-reliable communications, and mass communication of machines have also been considered.

The associate editor coordinating the review of this manuscript and approving it for publication was Antonino Orsino.

D2D communication refers to the connection among devices through any network infrastructure, controlling radio access resources and direct links for minimizing the resulting interference. It also provides an alternative mode of communication that increases connectivity, utilization of the spectrum, and coverage area [3].

D2D communication is defined as a direct communication between two mobile devices without passing through a cellular base station (BS) or network core [4]. It requires different levels of control by the operator. Based on the business model, it has either full, or partial control over the resource allocation among source, destination, and relay devices, or no control. Below are the different types of D2D communications defined according to the function of the cellular infrastructure (Bastos and Cecílio Da [5]):

1) D2D communication with establishment of an operator-controlled connection: source and destination devices talk and exchange data with each other with no BS; however, they are served by a BS to establish the link.
2) D2D communication with the establishment of a device-controlled connection: source and destination devices communicate directly with each other with no operator control, and must implement methods that

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see http://creativecommons.org/licenses/by/4.0/

guarantee limited interference with other devices on both same layer and macrocell layer.

In this study, we considered D2D communication with establishing/managing connection through the cellular infrastructure. Among the advantages of such an approach are possible offloading between the D2D network and the cellular network, which guarantees the continuity of sessions, and an efficient allocation of the spectrum if inband D2D is used.

Another important design principle for future wireless access is the decoupling of user data and a functional control system. The latter includes the information and necessary procedures for a device to access the system. The use of Software Defined Networking (SDN) can increase efficiency in the management and control of the networks.

SDNs approach the creation of networks whose control is detached from the hardware and given to a software application, called SDN controller. The SDN refers to a network architecture that enables the separation of the control plane and data plane to be more programmable, automatable and flexible. The network is virtualized and independent of the underlying physical infrastructure.

The main advantages of SDN include: (i) control and centralized management of network devices, (ii) an agile and flexible network that can adapt automatically through the use of common software interfaces, (iii) easy implementation of complex network functions through algorithms defined by users and implemented in software in the controller, (iv) rapid deployment of new network features, and (v) manufacturer's independence in network devices. Since the intelligence of the network is centralized in the controller, the network devices, which are only handlers of the data flows, can be obtained from multiple manufacturers.

Our proposal employs one or more SDN controllers in the cellular infrastructure for the management of ongoing D2D communications in the coverage area of each eNB (evolved NodeB). Each controller maintains the registrations and information on the location of users and their possible establishment of D2D communication. Such data enable the controller to determine the beginning of a D2D communication and send messages to the devices involved for beginning/establishing a direct communication among them and allocating network resources.

The controller also monitors the state of the D2D communication. If the link is lost (due to, for example, a change in location), it reconfigures the user equipment (UE) and the network entities to return the communication to the cell infrastructure and guarantees the continuity of the sessions. Therefore, it can implement efficient algorithms of offloading and resources allocation on the application layer (Northbound) and communicate with UE, eNB and SGW (Serving Gateway) for defining the path of the data packs by protocols, such as OpenFlow (Southbound).

On the other hand, systems must be designed with high scalability and reliability for avoiding problems (e.g., single point of failure). Since a single centralized SDN controller can lead to architectures with such problems, the following research questions have emerged:

1) How many controllers are required to manage all D2D communications?
2) Can a certain quality of services be guaranteed regarding the response time of the controller?
3) What is the best location of SDN controllers within the cellular network?
4) How can eNBs be assigned to the controllers?

To answer them, we must determine the minimum number of SDN controllers required, their optimal location and the optimal assignment of eNBs to them. The optimality criterion is based on the satisfaction of a response time lower than or equal to a specific QoS (Quality of Service) time. The problem was first defined in [6] as a Controller Placement Problem (CPP) for minimizing the latency between switches and controllers.

The CPP problem is similar to the facility location problem; therefore, it is considered an NP-hard problem [7]. The design of an algorithm that finds an optimal solution to such a problem is an exhausting and time-consuming task. The CPP search space consists of all k combinations of n potential nodes that produce a large set of feasible solutions. In this scenario, metaheuristic techniques are an alternative for the exploration of a subset of the search space and obtaining of a near-ideal solution. We used a metaheuristic known as Ant Colony System with External Memory Algorithm [8], [9] in the CPP solution.

Below are the main contributions of this study:

- an integrated approach to the CPP problem solution, considering the allocation of radio resources and the maximization of D2D communications allowed in each eNB;
- an SDN controller response time modeling with D2D communications management in LTE cellular networks;
- use of the response time metric considering the response time of the SDN controller and the latency between eNBs and controllers; and
- use of the ant colony system with external memory algorithm to solve the CPP problem considering management of D2D communications and SDN controllers.

To the best of our knowledge, this is the first manuscript on the placement of multiple SDN controllers for the management of D2D communications. We originally evaluated the impact of D2D communications management on the quantity and location of SDN controllers within an LTE cellular network.

The remainder of the article is organized as follows: Section II addresses some related work; Section III focuses on the modeling of the system; Section V provides an overview of the proposed approach to CPP solution; Section IV is devoted to the definition of the optimization problem; Section VI describes the ant colony system with external memory algorithm; Section VII reports on analyses of the results from cellular networks of different complexities; finally, Section VIII provides the conclusions.

II. RELATED WORK

A traditional implementation of SDN depends on a logically centralized controller that executes the control plane. However, in a large-scale WAN deployment, such a centralized approach shows several limitations related to performance, reliability, and scalability. Recent proposals have advocated the use of multiple controllers that work cooperatively to control the network, which has led to problems related to the necessary number of controllers, their best location and assignment of nodes (ex: switch) to them.

Heller *et al.* [6] evaluated such problems and defined them as a Controller Placement Problem (CPP) in SDN networks. The authors sought for the best location of k controllers on a WAN network by minimizing the propagation latency between the nodes and the controller. The authors evaluated the propagation latency with the use of concepts of average-case latency and worst-case latency in the solution of the optimization problem. The number of controllers was set in the interval $[1, 5]$ and the impact of the controller location on the two metrics was evaluated independently. The Internet2 OS3E topology was used and the analysis was expanded according to several topologies defined in Internet Topology Zoo (a controller was usually sufficient). However, important metrics in the evaluation of the response time for satisfying a given QoS, such as processing capacity of the controllers and load balancing, were not considered.

Bari *et al.* [10] presented a solution that dynamically adapts the number of controllers and their location to the conditions of the network. They adjusted the number of active controllers and assigned OpenFlow switches dynamically minimizing flow setup time and communication overhead. Moreover, they considered the statistics collection cost, flow setup cost, synchronization cost and switch reassignment cost, which represent, respectively, number of messages necessary for the collection of network statistics, cost of the path regarding propagation latency or number of hops, average number of flows generated by the switches, number of messages exchanged between the controllers, and cost of reassignment of a switch to a new controller. The authors used a simple modeling of the processing capacity of the controllers, which was defined in a vector representing the maximum number of requests each controller can manage per second.

Hu *et al.* [11] maximized the reliability of the network control to reach a best location for the controllers that minimizes the loss of the control path caused by network failures. A reliability metric represented the percentage of paths broken due to network failures. The problem was defined as NP-hard with k controllers and random placement, greedy search, simulated annealing and brute force were used to solve it. The authors evaluated the impact of the number of controllers on the reliability of the network and the trade-offs between reliability and latency. The results showed latency increases in locations that optimize reliability, however, according to the authors, this increase is still acceptable. Metrics related to response time and QoS assurance were not optimized.

Hock *et al.* [12] modeled CPP as a multi-objective problem and proposed a framework called Pareto-based Optimal COntroller-placement (POCO). They obtained the best location of k controllers considering first all possible combinations of faults of up to $k - 1$ controllers and then evaluating disruptions in the network. The metrics used were maximum node-controller latency and controller-controller latency in scenarios with failures, and load imbalance between controllers. The authors evaluated Topology Zoo topologies and the Internet2 OS3E topology, and showed the best latency value and best resiliency or fault tolerance cannot be obtained simultaneously. Therefore, the metrics involved must be compensated for, depending on their importance in the goal pursued. The authors explored the search space for a small number of controllers exhaustively; however, in large- scale networks, the obtaining of such results in acceptable time and with the existing computing capabilities is a difficult task.

Rath *et al.* [13] proposed a solution that dynamically enables/disables controllers considering the load on the network. According to the optimization mechanism used, each controller calculates a payoff function and compares its own value with that of neighboring controllers. It then makes appropriate decisions, so that new controllers are added or existing ones are deleted, or the download is performed between the controllers dynamically. The authors considered delay and usage constraints, such that the delay associated with the controller must be lower than a predefined threshold value (to support QoS) and the utilization must be within the minimum and maximum limits. Delay combines path and processing latency. The purpose of the scheme is to minimize packet loss, latency, and deployment cost.

Liao and Leung [14] used Multi-Objective Genetic Algorithm (MOGA) in the solution of the CPP problem. The mutation function is based on Particle Swarm Optimization (PSO). The authors presented optimal Pareto solutions by minimizing switch-controller latency, controller-controller latency, and load imbalance between controllers. In the tests performed, they defined the number of controllers and sought for the location and assignment of the switches.

Sahoo *et al.* [7] used metaheuristics based on biologically inspired populations to solve the CPP problem in large-scale WAN networks. They applied particle swarm optimization (PSO) and firefly algorithms (FA) to find the optimum location of the controllers and minimize latency between switches and controller. They compared the two meta-heuristics in analyses with a random localization strategy in three topologies defined on the Topology Zoo website and, according to the results, FA achieved better performance.

Ahmadi and Khorramizadeh [15] adapted multi- objective genetic algorithms to solve CPP and introduced Multi-start hybrid non-dominated sorting genetic algorithm (MHNSGA). They evaluated two approaches to the problem. The first considers the minimization of the following metrics: inter-controller latency, link or switch failures, and load balancing between controllers (using the load imbalance concept), whereas the second includes both load and capacity of the controllers in the evaluation metrics as a constraint of the problem. Solutions are sought for near the Pareto front and the trade-off between the metrics is conducted. The number of controllers at work must be known and several topologies defined in Internet Topology Zoo are used in the evaluation.

The main metrics considered in the resolution of the CPP problem are propagation latency between switches and controllers, inter-controller propagation latency, and load balancing between controllers. However, the capacity of the controllers and the load on the network are also important metrics owing to their direct influence on the response time of the controllers, hence, users' quality experience. Some of the previous work has included such metrics as constraints, using values of load and capacity known a priori. On the other hand, A. Farshin, and S. Sharifian [16] and Mackenzie *et al.* [17] presented solutions that use models based on the queuing theory to define the capacity of controllers.

Farshin and Sharifian [16] also optimized the number of SDN controllers used in a cellular network. They considered the problem of dynamic allocation of controllers and proposed a framework that uses a population-based metaheuristic to solve it. However, the location of the SDN controllers in the cellular infrastructure was not taken into consideration.

Mackenzie *et al.* [17] introduced a framework to solve the CPP problem in SDN cellular networks involving the uncertainty of the geographical distribution of users. Two schemes were proposed, of which one aims to optimize the number of controllers necessary for the management of all eNBs while ensuring the response time of each eNB does not exceed a specific value with a certain probability. The other scheme aims to optimize the eNB-controller assignment considering the variation in the eNB request rate for minimizing the response time for several eNBs.

We investigated the CPP problem taking into consideration the average response time of the SDN controllers and the eNB-controller latency within the response time. As in [16] and [17], we modeled the processing of the controllers applying the queuing theory. The request rate received by the SDN controllers considers the management of D2D communications established in the eNB coverage area. The ant colony system with external memory algorithm was applied for the finding of an approximate solution in acceptable time. The path taken by ants determines the eNB-controller assignment, number of controllers and their location.

According to a literature review on the SDN CPP problem for wireless networks, the treatment of D2D communications has not been addressed and the ant colony system with external memory meta-heuristic has not been applied to the problem considered here. Table 1 shows a summary of the main characteristics of the studies on the CPP problem.

III. SYSTEM MODEL

We considered a cellular network formed by one PGW (Packet Data Network Gateway), a set $S = \{1, 2, \ldots, s\}$ of SGW (Serving Gateway), a set $B = \{1, 2, \ldots, b\}$ of eNBs

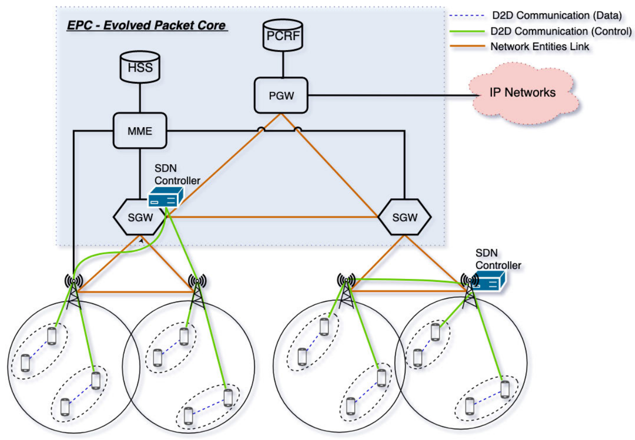


FIGURE 1. SDN Cellular Network for the control of D2D communications.

(evolved NodeB), and a set $C = \{1, 2, \ldots, c\}$ of controllers that can be located in any entity in the cellular network (PGW, SGW or eNB). A simplified representation of our topology is shown in Figure 1 for $s = 2$, $b = 4$, and $c = 2$. SGWs are connected to PGW and can communicate via a wired link. The eNBs are evenly distributed among the SGWs and can communicate with each other through a wired link.

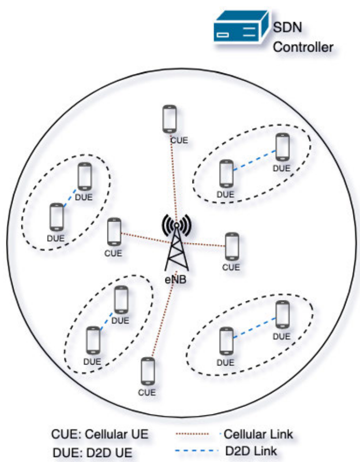


FIGURE 2. D2D and cellular concurrent users inside an SDN controller domain.

The system involves multiple D2D pairs and cellular users, as shown in Figure 2. User equipment (UE) is classified into cellular UE (CUE) and D2D-capable UE (DUE), which are under the control of an SDN controller. The CUEs communicate directly with the BS and D2D pairs exploit the direct link. Therefore, we defined a cellular scenario where cellular and D2D connections coexist in the same cell and transmit over the same bandwidth. The eNB is located in the cell center and the UEs are uniformly distributed in the cell.

D2D communications can operate in the following multiple modes [18], [19]: (i) Dedicated or overlay mode, when the cellular network allocates a fraction of the available resources

TABLE 1. Strategies used in CPP.

Proposal	Details	Metrics	Strategies	Focus on D2D Communications
Heller et al. [6]	best location by minimizing the node -controller propagation latency	average-case latency; worst-case latency	minimum k- median (local search heuristic for k-median); minimum k-center	No
Bari et al. [10]	number of active controllers and assignment of switches to controllers considering network failures	statistics collection cost; flow setup cost; synchronization cost; switch reassignment cost	a greedy approach based on the knap-sack problem; a simulated annealing-based meta-heuristic approach	No
Hu et al. [11]	best location of controllers that maximizes network reliability	reliability	random placement; l-w-greedy; simulated annealing; brute force	No
D. Hock, M. Hartmann, and S. Gebert [12]	better location considering higher resilience and fault tolerance	node-controller latency; controller-controller latency; load imbalance	Pareto-based optimal placement (considering all solution space)	No
Rath et al. [13]	best location of controllers after increasing/decreasing the number of controllers according to the load on the network	latency (processing and path delay); controller utilization	Non-zero-sum-based game theory	No
L. Liao and V. C. M. Leung [14]	best location and assignment of switches minimizing latency and load unbalance between the controllers	node-controller latency; controller-controller latency; load imbalance	Multi-Objective Genetic Algorithm (MOGA) with a PSO-based mutation	No
Sahoo et al. [7]	best location for controllers that optimizes latency	node-controller latency	PSO; firefly algorithm	No
V. Ahmadi, and M. Khorramizadeh [15]	better location and assignment of nodes to controllers, minimizing latency and load unbalance between controllers with load restrictions	node-controller latency; controller-controller latency; load imbalance; capacity of controllers; switch load	Multi-start hybrid non-dominated sorting genetic algorithm (MHNSGA)	No
Mackenzie et al. [17]	minimum number of SDN controllers, their ideal locations and assignment to eNBs, where the optimization criteria are based on the satisfaction of the eNBs delay requirements	transmission and propagation latency; queuing latency controller	Stochastic programming	No
A. Farshin, and S. Sharifian [16]	dynamic calculation of number of SDN controllers allocated considering network traffic	controller response time; controller utilization; Jain's Fairness Index	Chaotic version of Grey Wolf Optimizer (GWO)	No
Our proposal	number of controllers, their location and eNB assignment considering response time	controller response time; eNB-controller latency; Jain's Fairness Index	ant colony system with external memory algorithm	Yes

for the exclusive use of D2D devices, (ii) Reuse or underlay mode, when D2D devices use some of the radio resources together with the UEs of the cellular network, and (iii) Cellular mode, when D2D traffic passes through the eNB, as in traditional cellular communications.

We selected the underlay mode, as in [20]–[24], which assumes the bandwidth is divided into equal size resource blocks (RBs) and the total number of RBs in the uplink (UL) period is equal to the number of CUEs. We consider D2D communications reuse the UL period of the LTE-Advanced network. A CUE transmits uplink date to eNB, while other D2D pairs may reuse the same RB (resource block) preassigned to CUE to complete their transmission.

U CUEs and D D2D pairs are assumed to coexist in the cellular network, where $D > U$. We considered a set $U = \{1, 2, \ldots, u\}$ of the group of CUEs and a set $D = \{1, 2, \ldots, d\}$ of the group of D2D pairs. The corresponding transmitter and receiver in D are denoted by D_d^T and D_d^R, respectively. To prioritize cellular communications, we considered eNBs pre-assigned a set of uplink RBs to U CUEs before sharing the RBs with DUEs; each CUE occupies a separate RB in the UL period.

A. INTERFERENCE MODEL

We considered the interference between D2D communications and cellular communication occurring in the same RB (Figure 3) in the process of radio resource allocation. RBs are allocated orthogonally among CUEs within the cell, and each CUE occupies a separate RB in the UL period. No interference occurs among CUEs. All CUEs have minimum SINR (Signal-to-interference-plus-noise ratio) requirements. Each CUE u can share its RB with a set of D2D pairs if its SINR

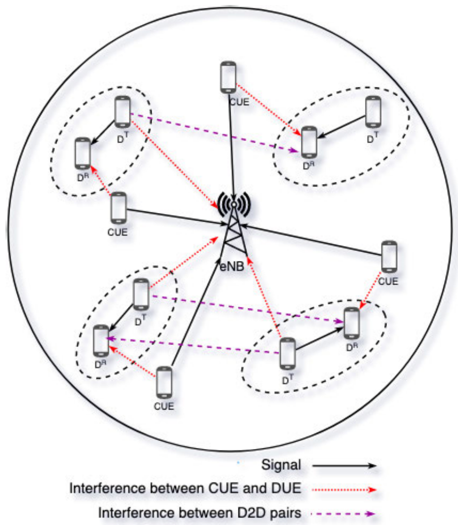


FIGURE 3. Interference when D2D communications reuse the uplink resources of CUEs in a cell.

requirement is satisfied. Therefore, the received SINR_u must be beyond the SINR threshold as:

$$\mathrm{SINR}_u = \frac{P_U g_{u,b}}{I_{\triangle u,b} + \sigma^2} \geq \mathrm{SINR}_u^t \tag{1}$$

where b and u denote eNB_b and CUE u, respectively, P_U is the transmitted power of CUEs, $g_{u,b}$ is the channel gain from transmitter CUE_u to receiver eNB_b, $I_{\triangle u,b}$ is the interference received at eNB_b from D2D pair transmitters using the RB assigned to CUE_u, and σ^2 is the thermal noise power assumed the same for eNB and all the D2D receivers.

Interference ($I_{\triangle u,b}$) received at the eNB_b associated with CUE_u, must be less than or equal to I_u^{max} defined by SINR_u^t and given by:

$$I_{\triangle u,b} = \sum_{d \in \triangle u} P_D g_{d,b} \leq I_u^{max} \tag{2a}$$

$$I_u^{max} = \frac{P_U g_{u,b}}{\mathrm{SINR}_u^t} - \sigma^2 \tag{2b}$$

where $\triangle u$ is the set of D2D pairs that reuse the RBs allocated to CUE_u, $g_{d,b}$ is the channel gain from transmitter D_d^T to receiver eNB_b, P_D is the transmitted power of D2D pairs assumed the same for all D2D pairs.

Similarly, D2D pairs have their minimum SINR requirements. For any D2D pair d, it reuses certain RBs only if receiver SINR at D_d^R exceeds the SINR threshold:

$$\mathrm{SINR}_d = \frac{P_D g_{d^T,d^R}}{I_{u,d^R} + I_{d',d^R} + \sigma^2} \geq \mathrm{SINR}_d^t \tag{3}$$

where g_{d^T,d^R} is the channel gain from transmitter D_d^T to receiver D_d^R; I_{u,d^R} is the interference received at D_d^R from CUE_u, and I_{d',d^R} is the interference received at D_d^R from D2D pair transmitters using the same RBs as D_d. The above interference can be obtained by

$$I_{u,d^R} = P_U g_{u,d^R} \tag{4a}$$

$$I_{d',d^R} = \sum_{d' \in \triangle u - d^T} P_D g_{d',d^R} \leq I_d^{max} \tag{4b}$$

$$I_d^{max} = \frac{P_D g_{d^T,d^R}}{\mathrm{SINR}_d^t} - \sigma^2 \tag{4c}$$

Therefore, if two users are in the LTE Direct coverage area, they can initiate a direct link between them whenever the SINR limits are met.

Table 5 shows a list of the symbols and notations used to provide an easier understanding of the article.

B. M/M/M/K MODEL FOR MULTI-CORE SDN CONTROLLERS

The SDN controller is treated here as a Markovian model M/ M/m/K, with a single queue of limited capacity, where m is the number of cores of each controller. We suppose the SDN controllers, used in the cellular network, are running virtual machines (VMs) with four CPU cores. Their limit is set by the K parameter and follows a FIFO (First In First Out) discipline.

Let us consider:

- λ_c : average request rate received in the SDN controller;
- λ_{eff} : effective request rate in the SDN controller;
- λ_{lost} : lost requests rate in the SDN controller;
- μ : processing rate of each core, which follows an exponential distribution;
- m : number of controller cores ($m = 4$); and
- $m\mu$: processing rate of the SDN controller.

When the system is full, the new requests are discarded, hence, $\lambda_c = 0$ for $n \geq K$, where n is the request number received by the controller, and $\lambda_{eff} = \lambda_c - \lambda_{lost}$ and $\lambda_{lost} = \lambda_c \cdot p_K$, where p_K is the probability of K requests in the system.

The controller utilization rate ρ is given by ([25])

$$\rho = \frac{\lambda_{eff}}{m\mu} \tag{5}$$

The probability (p_0) of no requests in the controller depending on the utilization rate is given by Equation 6, as shown at the bottom of the next page.

The average queue length (L_q) is given by Equation 7, as shown at the bottom of the next page.

The probability (p_K) of K requests in the controller in function of the utilization rate is given by

$$p_K = \frac{(\lambda_c/\mu)^K}{m! \cdot m^{(K-m)}} \cdot p_0 \tag{8}$$

Finally, the average response time of each SDN controller (t_c) is the time spent by the packet in the queue plus the processing time, obtained as

$$t_c = \frac{L_q}{\lambda_{eff}} + \frac{1}{\mu} \tag{9}$$

IV. PROBLEM DEFINITION

This study aims to find the minimum number of required SDN controllers, their optimal location and assignment to eNBs. The optimality criterion is based on the satisfaction of a response time lower than or equal to a specific QoS time, at a lost requests rate in the SDN controller lower than a value $\lambda_{lost}^{threshold}$. t_r was defined as the response time of a request and t_{QoS} was the QoS time considered, so that the constraints of our problem are $t_r \leq t_{QoS}$ and $\lambda_{lost} \leq \lambda_{lost}^{threshold}$.

t_r depends on the average response time of the SDN controller (t_c) and propagation latency in packet delivery, as follows:

$$t_r = t_c + t_{bc} \tag{10}$$

The propagation latency between the eNB$_b$ and the SDN controller c (t_{bc}) is obtained as the sum of the number of hops between controller location and eNB, multiplied by the weight associated with each link on the path (without loss of generality, all links were considered equal).

After the average response time (t_r) has been calculated, the cost function that checks whether the time constraint has been satisfied must be defined and should grow rapidly when the response time starts to be higher than t_{QoS}. Therefore, an exponential function was applied to characterize this behavior; when the time was higher than t_{QoS}, a constant ξ was used and increased the cost function at a high rate. The utility function (Figure 4) can be calculated as

$$T(t_{r_i}) = \begin{cases} e^{(t-t_{QoS})}, & t \leq t_{QoS} \\ e^{\xi(t-t_{QoS})}, & t > t_{QoS} \end{cases} \tag{11}$$

where i is the controller index and ξ denotes constant factor that specify the increase/decrease rate of the cost function, defined as $\xi = 2\ t_{QoS}$. The total average response time (T_r) is defined as the average response time of each controller as:

$$T_r = \frac{1}{c}\sum_{i=1}^{c} T(t_{r_i}); \quad c = \text{controllers number} \tag{12}$$

In the controller location problem, the metric most commonly used is the latency between the switches (eNBs) and the controller [7]. It is defined as the average propagation latency between the location of the controller and the location of the eNBs assigned to it ([6], [7], [12], [14], [15], [26], [27]),

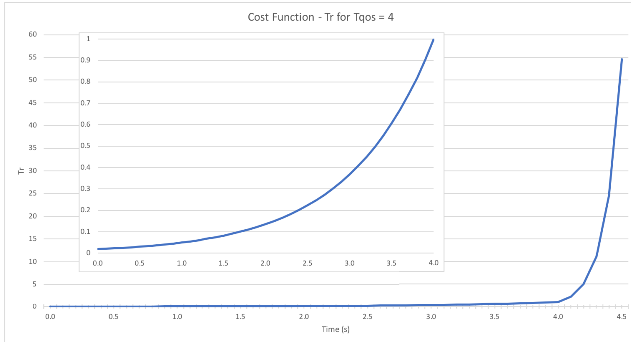


FIGURE 4. Average response time function.

and can be obtained as

$$\pi^{avglat}(L) = \frac{1}{b}\sum_{b\in B} \min_{l\in L} \quad t_{bl} \tag{13}$$

where set $L = PGW \cup \{1, 2, \ldots, s\} \cup \{1, 2, \ldots, b\}$ was defined as the possible locations of the controllers, i.e., the controller can be located in any network entity and t_{bl} is the propagation latency between eNB b and the controller located in l. The objective is to minimize the average propagation latency.

$$\pi_C^{avglat} = \frac{1}{c}\sum_{c\in C}\left(\frac{1}{b}\sum_{b\in B}\left(\min_{l\in L} \quad t_{bl}\right)\right) \tag{14}$$

such that a given controller cannot be placed in multiple locations and each eNB will be managed only by a controller c.

$$\sum_{c\in C} M(b,c) = 1 \quad \forall b \in B \tag{15a}$$

$$\sum_{c\in C} M(c,l) = 1 \quad \forall l \in L \tag{15b}$$

where $M_{b,c} \in 0, 1$ represents the eNB assignment to the controllers and $M_{c,l} \in 0, 1$ denotes the controller location. Time restriction $t_r \leq t_{QoS}$ is maintained.

Another important factor in the assignment of eNBs to SDN controllers is the fair distribution of eNBs between controllers, which results in load balancing between them.

$$p_0 = \begin{cases} \left\{\sum_{n=0}^{m-1}\frac{(\lambda_c/\mu)^n}{n!} + \frac{(\lambda_c/\mu)^m}{m!}\cdot\left[\frac{1-(\lambda_c/m\mu)^{(K-m+1)}}{1-(\lambda_c/m\mu)}\right]\right\}^{-1}, & \text{if } \frac{\lambda_c}{m\mu} \neq 1 \\ \left\{\sum_{n=0}^{m-1}\frac{(\lambda_c/\mu)^n}{n!} + \left[\frac{(\lambda_c/\mu)^m}{m!}\cdot(K-m+1)\right]\right\}^{-1}, & \text{if } \frac{\lambda_c}{m\mu} = 1 \end{cases} \tag{6}$$

$$L_q = \begin{cases} \frac{p_0(\lambda_c/\mu)^m\cdot(\lambda_c/m\mu)}{m!\cdot(1-(\lambda_c/m\mu))^2}\cdot\left[1-(\frac{\lambda_c}{m\mu})^{K-m} - (K-m)\cdot(\frac{\lambda_c}{m\mu})^{K-m}\cdot(1-\frac{\lambda_c}{m\mu})\right], & \text{if } \frac{\lambda_c}{m\mu} \neq 1 \\ \frac{(\lambda_c/\mu)^m(K-m)(K-m+1)}{2m!}\cdot p_0, & \text{if } \frac{\lambda_c}{m\mu} = 1 \end{cases} \tag{7}$$

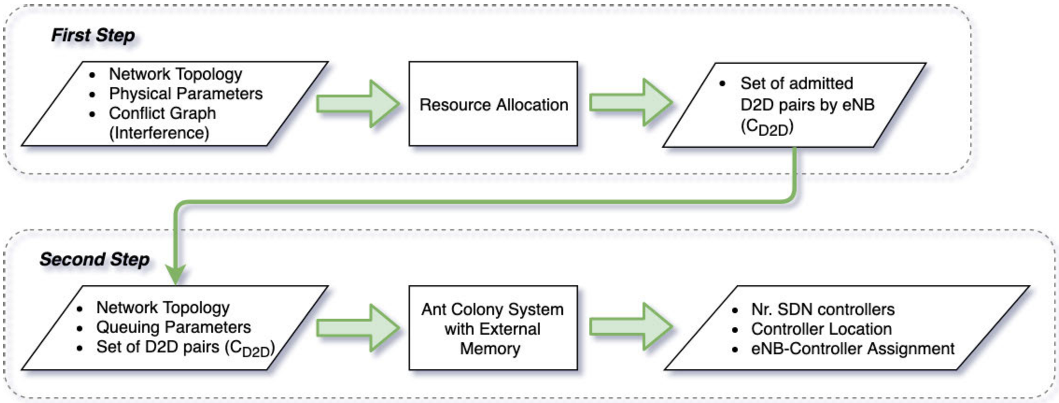


FIGURE 5. Steps involved in CPP solution.

Jain's Fairness Index [28] measured fairness in the allocation of controllers and guaranteed that condition. We assumed the total load of the controllers could be calculated as $\rho = \frac{\lambda_c}{m\mu}$ [16], and the Jain's index could be obtained by

$$f = \frac{\left(\sum_{i=1}^{c} \rho_i\right)^2}{c \cdot \sum_{i=1}^{c} \rho_i^2} \quad (16)$$

Since our interest was to minimize the lack of fairness among SDN controllers, we considered $F = 1 - f$ a factor to be minimized.

A function that provides the number of controllers used is required; it must increase, as the number of controllers increases, and be between [0, 1]. Therefore, we defined $C_c = 1 - 1/c$, where c is the controller number.

Finally, the objective function (Eq. 17) considered the weighted sum of the number of controllers, response time (Eq. 12), average propagation latency between the controllers and the assigned eNBs (Eq. 14), and justice parameter (F), such that

$$OF = \gamma_c \cdot C_c + \gamma_T \cdot T_r + \gamma_\pi \cdot \pi_C^{avglat} + \gamma_F \cdot F \quad (17)$$

where $\gamma_c, \gamma_T, \gamma_\pi$, and γ_F are the weights of each cost function.

In summary, the objective in this step is to minimize the objective function (Eq. 17) considering the following constraints:

(i) the response time of each controller must be lower than or equal to t_{QoS};
(ii) each eNB is assigned to a single controller; and
(iii) two controllers cannot be located in the same network entity.

The ant colony system with external memory algorithm was applied to solve this optimization problem.

V. OVERVIEW OF THE PROPOSED SOLUTION

For the CPP solution we present an approach that integrates radio resource allocation and D2D communications control. Our approach considers two steps as shown in Figure 5.

In the first step, the number C_{D2D} of simultaneous D2D communications that can be established in each cell is calculated. To obtain C_{D2D}, the resource allocation problem is formulated to maximize the number of admitted D2D pairs in the network, subject to the constraints of the maximum tolerable interference level for both eNB and DUE.

When D2D links are added to the network, two main levels of interference can be found. One interference is caused by the cellular network (i.e. from CUEs to D2D receivers) and the other one is caused by D2D network (i.e from D2D transmitter to the eNBs and from D2D transmitter to D2D receivers of other links). Therefore, the constraints can be summarized as follows:

- $\mathrm{SINR}_u \geq \mathrm{SINR}_u^t \quad \forall u \in U$ (Equation 1);
- $\mathrm{SINR}_d \geq \mathrm{SINR}_d^t \quad \forall d \in D$ (Equation 3) and
- each D2D pair can only reuse one CUE's resource block.

The serving eNB is assumed to obtain the channel state information for all links and know the SINR threshold of all user equipment. The purpose is to allow multiple D2D pairs to reuse the same RBs that have been pre-assigned to CUEs, whenever QoS requirements are met. To face this issue, we used a conflict graph [29] to model the network interference.

In a conflict graph, constructed for each set of D2D links that share the same resources, i.e. RBs, each vertex denotes a communication link (a D2D communication link) and each edge represents the unacceptable interference between the vertices that it connects ([[22], [30]]). Given the interference model, it is possible to create edges in the conflict graph that represent mutual interference between D2D links. Thus, in each conflict graph the vertices represent the D2D links that do not have conflicts with the CUE that uses that RB and the edges represent the interference between the D2D links.

The Greedy Resource Allocation Algorithm, proposed in [20], was employed for solving the problem presented in the first part of our approach. In the algorithm, a sequential resource allocation mode was considered, where eNB determines the admitted D2D pairs according to the RB index. Conflict graphs are updated whenever a D2D pair is selected to use a resource block and tolerable interference values are evaluated by both eNB and CUE that have already been admitted. The above procedures are implemented sequentially in each RB and the D2D pairs with the lowest interference rates are evaluated first. The algorithm runs on each eNB and the set of admitted D2D pairs in each eNB is returned.

The obtained C_{D2D} values are used as input parameters in the second step of the proposed solution. They define the average control requests rate received by SDN controllers as below:

$$\lambda_{c_i} = \sum_{j=1}^{b_{c_i}} C_{D2D_j} * \lambda \tag{18}$$

where b_{c_i} is the number of eNBs managed by controller c_i, C_{D2D_j} is the number of D2D communications established in *jth* eNB and λ is the average requests rate generated by each D2D pair. C_{D2D_j} will depend on the geographical location of the users and the radio resources allocation on the LTE-Advanced network.

In the second step, the CPP problem is solved aiming to minimize the response time in the control of D2D communications, while attending the mentioned constraints. This time considers the average response time of each SDN controller obtained using the queuing model described in the previous section (Equation 9). Therefore, the set of pairs D2D admitted in each cell is an input parameter in the second step and included in the response time.

CPP was modeled as an optimization problem and, to solve it, an ant colony algorithm approach based on external memory was used.

VI. ANT COLONY ALGORITHM WITH EXTERNAL MEMORY (ACS-EM)

In optimization algorithms based on ant colonies (ACO - Ant Colony Optimization) [31], artificial ants build a solution to a combinatorial problem by walking through a graph called 'construction graph', $G_C(V, E)$ that consists of a set of vertices V and a set of arcs E. In this study, the construction graph was divided into two parts for troubleshooting the controller placement problem (CPP). The first part assigns eNBs to controllers and vertex subset $V_1 = \{b_1c_1, \cdots, b_1c_n, b_nc_1, \cdots, b_nc_n\}$ represents all possible pairs of eNB - SDN Controller that exist in the scenario considered. In the second part, SDN controllers are located in the cellular infrastructure. Vertex subset $V_2 = \{c_1l_1, \cdots, c_1l_n, c_nl_1, \cdots, c_nl_n\}$ represents all possible controller-location pairs, as shown in Figure 6.

Ants move from one vertex to another along the arc of the graph, incrementally constructing a partial solution, and deposit a certain amount of pheromone $\Delta\tau$ in the components, i.e, in the arches they cross. The amount depends on the quality of the solution found. Ants use pheromone information as a guide to travel through the most promising regions of the search space.

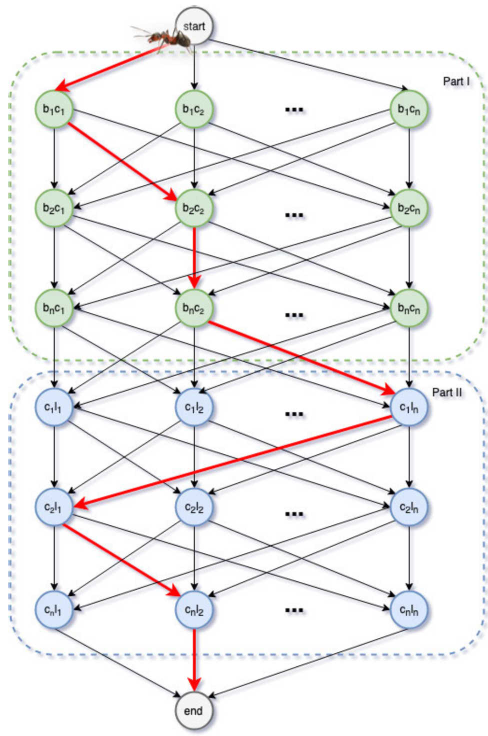


FIGURE 6. Construction graph.

To obtain the number of SDN controllers used, the graph is constructed assuming an SDN controller is necessary for managing the traffic of each eNB ($c = b$). When the ants reach the end of the path, SDN controllers that have not assigned eNB are eliminated from the solution, and the number of controllers actually used is obtained.

ACO algorithms generate candidate solutions for an optimization problem by a construction mechanism, by which the choice of a solution component to be added at each stage The following rule is used in the intensification of solutions: of construction is probabilistically influenced by pheromone traces and heuristic information [31]. This study analyzes the possibility of alternating the way solution components are chosen, introducing an external memory as an auxiliary mechanism for making decisions at each stage of the construction of a solution.

A frequency-based memory [32], which stores components of the solutions most frequently chosen, was used. It "prohibits" an ant from choosing a solution component, because it is often chosen in the solutions. It is sought through the "prohibition" for generating solutions that effectively differ from those already generated, thus expanding the exploration of the search space. On the other hand, this information can be used to "promote" a choice because it is considered attractive. Most ants choose it as part of their solutions and, therefore, it should be considered part of a new solution.

The algorithm has a frequency array of size $[B, C]$, called *memory_frec*, associated. Position *memory_frec*$[b_i, c_j]$ stores the number of times eNB b_i was assigned to controller c_j

during the algorithm execution. It represents the vertices chosen by ants in the construction of the solution and is used in the construction of solutions in two possible ways, i.e., intensification, by the choice of a node that matches as an eNB-controller which was most often selected from the current location, and diversification, by the choice of a node less frequently chosen from the current location. External memory values are updated at the end of each iteration considering the value of the best solution.

During the construction of a solution, ants apply a choice of the action rule similar to that used in the ant colony system (ACS) algorithm [31]. A parameter $q_0 \in [0, 1]$ is introduced and promotes the choice of a solution component that favors intensification or diversification in the search for solutions. When an ant chooses a component from a solution, it first generates a uniformly distributed random number in the [0, 1] range. If the number is lower than the value of the q_0 parameter, a component is chosen to perform a local search. If the generated number is greater than the q_0 parameter, the component is chosen favoring the search for new solutions. Each of such options includes a decision on traces of pheromones or external memory. Therefore, a vertex v_j is chosen from a vertex v_i according to the following rule:

$$v_j = \begin{cases} R_I, & \text{if } q < q_0 \text{ (Intensification)} \\ R_D, & \text{if } q \geq q_0 \text{ (Diversification)} \end{cases} \tag{19}$$

With a fixed probability q_0, the ant chooses the ''best possible element'' according to the acquired knowledge. It can be based on either external memory, or traces of pheromone, whereas with probability $(1 - q_0)$, it performs a controlled exploration of new solutions, where q is a uniformly distributed random variable in the [0, 1] range.

The following rule is used in the intensification of solutions:

$$R_I = \begin{cases} arg\ \text{máx}_{j \in N_i^k} \{memory\_frec[b_j, c_j]\}, & \text{if } r < r_0 \\ arg\ \text{máx}_{j \in N_i^k} \{\tau_{ij} \cdot \eta_{ij}^{\beta}\}, & \text{if } r \geq r_0 \end{cases} \tag{20}$$

where N_i^k is the neighborhood reachable by ant a when it lies in node v_i, τ_{ij} is the amount of pheromone between nodes v_i and v_j, η_{ij} is the heuristic or desirability information between nodes v_i and v_j and β is a parameter that defines the relative importance of heuristic information. A fixed probability $r_0(0 \leq r_0 \leq 1)$ chooses the element most often selected from the current location, whereas with probability $(1 - r_0)$, the most desirable element is chosen according to the pheromone stroke. r is a uniformly distributed random variable in the [0.1] range.

The following rule is used in the exploration of new solutions (diversification):

$$R_D = \begin{cases} arg\ \text{mín}_{j \in N_i^k} \{memory\_frec[b_j, c_j]\}, & \text{if } z < z_0 \\ Z, & \text{if } z \geq z_0 \end{cases} \tag{21}$$

A fixed probability $z_0(0 \leq z_0 \leq 1)$ enables the selection of the least frequently chosen element from the current location, whereas probability $(1 - z_0)$ promotes the choice of element Z according to the basic selection rule, as in the ACS algorithm [31]. z is a uniformly distributed random variable in the [0.1] range, selected by the roulette method, according to the probability distribution p_z similarly to the ant system algorithm [31]:

$$p_z(i,j) = \begin{cases} \dfrac{\tau_{ij}^{\alpha} \cdot \eta_{ij}^{\beta}}{\sum_{j' \in N_i^k} \tau_{ij'}^{\alpha} \cdot \eta_{ij'}^{\beta}}, & \text{if } j \in N_i^k \\ 0, & \text{in another way} \end{cases} \tag{22}$$

where α is the parameter that establishes the relative importance of pheromone tracks.

The pheromone tracks are updated in two steps: (i) each time an ant traverses an arc, called a local update, and (ii) at the end of each iteration, called global update.

The online or local pheromone update rule is applied by all ants whenever they cross an arc (i, j) during the solution construction, calculated as:

$$\tau_{ij} = (1 - \delta) \cdot \tau_{ij} + \delta \cdot \tau_0 \tag{23}$$

where τ_0 is the initial pheromone value and $\delta \in (0, 1)$ is the pheromone evaporation coefficient.

The step-by-step update rule includes both pheromone evaporation and pheromone deposition. Since the pheromone deposited is very small, the application of this rule causes traces of pheromones between the paths traveled by ants to decrease, which leads to an additional exploration technique of the algorithm. The paths traversed by a large number of ants become less attractive to the other ants that cross them in the current iteration. Consequently, they may not follow the same path.

In the global update of pheromone tracks, only the path taken by the ant with the best solution (the *best-so-far* ant) is updated after each iteration by the following equation:

$$\tau_{ij} = \begin{cases} (1 - \delta)\tau_{ij} + \delta \triangle \tau_{ij}^{bs}, & \text{if } \ j \in \Omega \\ (1 - \delta)\tau_{ij}, & \text{if } \ j \notin \Omega \end{cases} \tag{24}$$

where $\Omega = \{1, 2, \ldots, J\}$ is the set of nodes that belong to the path traveled by the *best-so-far* ant and $\triangle \tau_{ij}^{bs}$ is the increment in the pheromone tracks expressed as:

$$\triangle \tau_{ij}^{bs} = \frac{1}{OF^{bs}} \tag{25}$$

where OF^{bs} is the value of the objective function (Equation 17) of the best iteration solution.

The pseudo-code of the ACS-EM algorithm to troubleshoot the CPP is shown in Algorithm 1.

VII. EVALUATION

This section provides the results of the approach proposed in this article to CPP, considering D2D communications that take place in an LTE-Advanced network. In the first phase,

Algorithm 1 ACS-EM - Ant Colony System With External Memory

Input: $Network\_Topology$, C_{D2D}, $Queuing\_Parameters$, τ_0, α, β, q_0, r_0, z_0, δ, $ant\_quantity$, $iteration\_max$

$\tau[\cdot] = InitPheromone()$
$memory[\cdot] = InitMemory()$
for $i = 0$ to iteration_max **do**
  for each $a \in ant\_quantity$ **do**
    $currentPosition \leftarrow RootNode$
    $TourList \leftarrow currentPosition$
    while $currentPosition.Children \neq 0$ **do**
      $\eta[\cdot] \leftarrow CalculateChildrenHeuristic()$
      $q \leftarrow RandomDouble()$
      if $q < q_0$ **then** ▷ R_I - Exploitation
        $r \leftarrow RandomDouble()$
        if $r < r_0$ **then** ▷ External Memory
          $nextPosition \leftarrow arg\_\text{máx}\{memory[child]\}$ ▷ eq. 20
        else ▷ Probabilistic ACS
          $nextPosition \leftarrow arg\_\text{máx}\{\tau[child] * \eta[child]^{\beta}\}$
        end if
      else ▷ R_D - Exploration
        $z \leftarrow RandomDouble()$
        if $z < z_0$ **then** ▷ External Memory
          $nextPosition \leftarrow arg\_\text{mín}\{memory[child]\}$ ▷ eq. 21
        else ▷ Probabilistic ACS
          $p_z[\cdot] \leftarrow CalculateChildrenProbability()$ ▷ eq. 22
          $nextPosition \leftarrow Get\_Z(p_z)$
        end if
      end if
      Add nextPosition to TourList
      $ValidateFeasiblePath(TourList)$ ▷ Validate problem constraints
      $currentPosition \leftarrow nextPosition$
      LocalPheromoneUpdate() ▷ eq. 23
    end while
    if $CalculateAntsTour() < BestTour$ **then** ▷ eq. 17
      $BestTour \leftarrow TourList$
    end if
  end for
  $OfflinePheromoneUpdate(BestTour)$ ▷ eq. 24
  $memory[\cdot]$ = UpdateMemory(BestTour)
  if $BestTour < GlobalBestTour$ **then**
    $GlobalBestTour \leftarrow BestTour$
  end if
end for

Output: GlobalBestTour

a greedy algorithm is used to obtain the number of D2D communications allowed on the network (C_{D2D}). In the second phase, the ACS-EM algorithm is applied to CPP, aiming to find the number and respective location of SDN controllers, necessary for the management of D2D communications occurring in an LTE cellular network.

An LTE network with 1 PGW, 2 SGW and 8 eNB was considered for the evaluation of the influence of D2D communications on the location and number of controllers. Table 2 shows the parameters of the System Model.

Adjustments were also made in the simulation environment for the definition of the values of the ACS-EM algorithm parameters. The best values were: $ant\_quantity = 100$; $q_0 = 0.2$; $r_0 = 0.2$; $z_0 = 0.2$; $\alpha = 0.1$; $\beta = 0.9$; $\tau_0 = 0.05$ and $\delta = 0.1$. The algorithm was applied 32 times and 450 iterations were considered.

Different values of λ were used and enabled the evaluation of the effect of an increase/decrease of control traffic related to D2D communications. The following metrics were considered: number of controllers, response time, use of average

TABLE 2. Simulation parameters.

Parameter	Value
Carrier frequency	2 GHz
System bandwidth	1 MHz per RB
Number of CUEs	10 user per eNB
DUE density	$10^{-3} user/Km^2$
CUE transmitted power	23 dBm
DUE transmitted power	20 dBm
eNB coverage radius	500 m
Distance between D2D pair	Uniformly distributed in [0,20]m
t_{QoS}	4 ms
D2D request rate - λ	0.1 req/ms
core processing rate - μ	50 req/ms
number of controller cores - m	4
SDN controller capacity - K	4μ
$\lambda_{lost}^{threshold}$	5% of $c \cdot \lambda_c$
Path loss model for cellular link	$128.1 + 37.6 log_{10}(d[Km])$
Path loss model for D2D pair	$148 + 40 log_{10}(d[Km])$
Shadow fading standard deviation	10 dB for cellular link, 12 dB for D2D pair
Small-scale fading	i.i.d complex Gaussian distributed with zero mean and unit variance
Antenna gain	14 dBi for eNB 0 dBi for UE
Noise spectral density	-174 dBm/Hz
SINR_u^t	10 dB
SINR_d^t	10 dB
number of iterations	450
γ_c	0.4
γ_π	0.1
γ_T	0.3
γ_F	0.2

controllers, assignment of eNB-controllers, location of controllers in the cellular network, and lost request rate in the SDN controller (Table 3).

As shown in Figure 7, the increase in the rate of requests generated by D2D communications increases the number of required SDN controllers. According to Table 3, the number of controllers required increases only when the constraints are not satisfied. As an illustrative example, four controllers are used for $\lambda = 0.5$; however, the controller load is $\simeq$100%, and the loss rate begins to increase. When λ increases to 0.6, more controllers are used not to exceed the threshold set for lost requests rate and satisfy all constraints of the problem.

The results also show when more than one controller is used, their best location is in the access network, near D2D communications, since eNBs represent the first hop, which reduces both propagation latency and total response time. However, when only one controller is used, it must be placed in the core of the network, in the SGW entity,

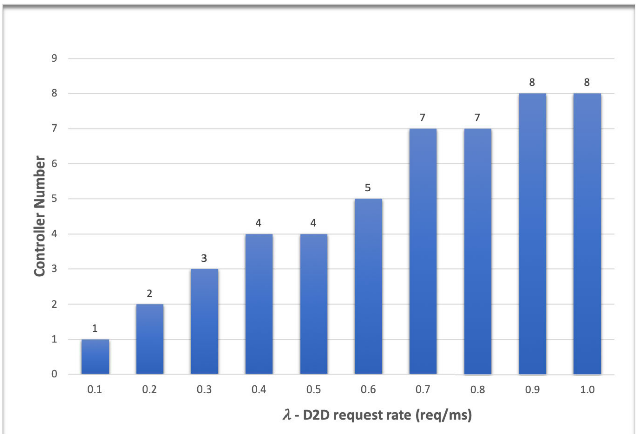


FIGURE 7. Effect of λ on the number of required SDN controllers.

which increases the response time of each eNB for the controller. Regarding the entire network, the average value of the response time provides a better result without exceeding the QoS value considered. The fairness in the allocation of the control for the eNB is also among the objectives. As shown in Table 3, the assignment was balanced between the controllers; therefore, the results from ACS-EM are consistent with the expected solutions.

ACS-EM was compared with the ACS algorithm [31] and the particle swarm-based metaheuristic, PSO algorithm [33], for the evaluation of its effectiveness. ACS was applied with the following parameters: $ant\_quantity = 100$; $q_0 = 0.2$; $\beta = 2$; $\tau_0 = 0.05$ and $\delta = 0.1$.

In PSO [33], the particles cooperate to search for the global optimum in the n-dimensional search space. The *ith* particle maintains both position and velocity. In each iteration, each particle uses its own search experience (self-cognitive) and the whole swarm's search experience (social-influence) to update velocity, and flies to a new position. Therefore, $c1$ and $c2$ are two parameters that weigh the importance of self-cognitive and social-influence, respectively. A scheme that linearly decrements inertia weight w between w_{max} and w_{min} is used in [33] to update the velocity. The discrete PSO modeling algorithm presented in [33] was applied with the following parameters: $particle\_quantity = 100$; $w_{max} = 0.9$; $w_{min} = 0.4$; $c1 = 2$; $c2 = 2$.

Figure 8 shows the average of the objective function value obtained in the 32 observations. The standard deviation of the results for ACS-EM and ACS is small, whereas the results for PSO show a higher dispersion. Since ACS-EM achieved a lower average in most instances, we inferred its performance was probably better than that of ACS and PSO algorithms. However, statistical tests were applied for theoretically supporting the previous statement.

First, the nature of the samples was tested for the checking of their distribution (normal or not). If the results followed a normal distribution, parametric hypothesis tests were applied for finding the one of better performance. Otherwise, non-parametric hypothesis tests were applied. A 95% confidence level was used for all tests.

TABLE 3. Simulation results.

λ (req/ms)	Nro Controller	Tr	Controller Utilization	eNB-Controller	Controller Location	λ_{lost}
0.1	1	1.614	89.50 %	$b_0c_0, b_1c_0, b_2c_0, b_3c_0,$ $b_4c_0, b_5c_0, b_6c_0, b_7c_0$	$c_0 - SGW_1$	0.0002
0.2	2	0.8706	89.50 %	$b_0c_0, b_2c_0, b_4c_0, b_6c_0$ $b_1c_1, b_3c_1, b_5c_1, b_7c_1$	$c_0 - eNB_6$ $c_1 - eNB_5$	0.0012
0.3	3	0.9824	89.48 %	b_3c_0, b_5c_0 b_0c_1, b_2c_1, b_6c_1 b_1c_2, b_4c_2, b_7c_2	$c_0 - eNB_5$ $c_1 - eNB_6$ $c_2 - eNB_1$	0.0155
0.4	4	0.6254	89.50 %	b_2c_0, b_6c_0 b_1c_1, b_3c_1 b_5c_2, b_7c_2 b_0c_3, b_4c_3	$c_0 - eNB_2$ $c_1 - eNB_3$ $c_2 - eNB_5$ $c_3 - eNB_0$	0.0020
0.5	4	1.3857	99.97 %	b_0c_0, b_4c_0 b_2c_1, b_6c_1 b_1c_2, b_3c_2 b_5c_3, b_7c_3	$c_0 - eNB_0$ $c_1 - eNB_2$ $c_2 - eNB_1$ $c_3 - eNB_7$	11.9048
0.6	5	1.1068	93.60 %	b_4c_0 b_3c_1, b_7c_1 b_0c_2, b_2c_2 b_1c_3, b_6c_3 b_5c_4	$c_0 - eNB_4$ $c_1 - eNB_3$ $c_2 - eNB_0$ $c_3 - eNB_6$ $c_4 - eNB_5$	13.8000
0.7	7	0.3634	87.12 %	b_4c_0 b_5c_1 b_3c_2 b_0c_3 b_2c_4 b_1c_5, b_7c_5 b_6c_6	$c_0 - eNB_4$ $c_1 - eNB_5$ $c_2 - eNB_3$ $c_3 - eNB_0$ $c_4 - eNB_2$ $c_5 - eNB_1$ $c_6 - eNB_6$	2.3754
0.8	7	0.642	93.48 %	b_6c_0 b_5c_1 b_4c_2 b_2c_3 b_1c_4, b_7c_4 b_0c_5 b_3c_6	$c_0 - eNB_6$ $c_1 - eNB_5$ $c_2 - eNB_4$ $c_3 - eNB_2$ $c_4 - eNB_7$ $c_5 - eNB_0$ $c_6 - eNB_3$	8.8090
0.9	8	0.5384	90.91 %	b_7c_0 b_6c_1 b_3c_2 b_1c_3 b_2c_4 b_0c_5 b_5c_6 b_4c_7	$c_0 - eNB_7$ $c_1 - eNB_6$ $c_2 - eNB_3$ $c_3 - eNB_1$ $c_4 - eNB_2$ $c_5 - eNB_0$ $c_6 - eNB_5$ $c_7 - eNB_4$	9.7801
1	8	0.6425	94.37 %	b_7c_0 b_6c_1 b_5c_2 b_4c_3 b_3c_4 b_2c_5 b_1c_6 b_0c_7	$c_0 - eNB_7$ $c_1 - eNB_6$ $c_2 - eNB_5$ $c_3 - eNB_4$ $c_4 - eNB_3$ $c_5 - eNB_2$ $c_6 - eNB_1$ $c_7 - eNB_0$	17.5041

The experiments were conducted according to the following methodology:

1) search the results for ACS-EM, ACS and PSO using the system model defined at the beginning of this section; 32 simulations (observations) were performed for each case;
2) generate the three samples with the best results for each case (32 observations);

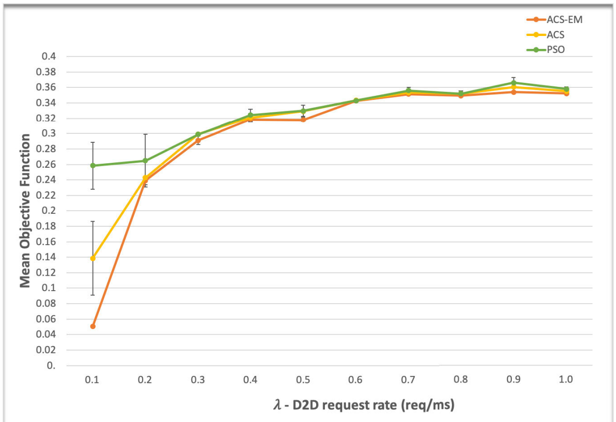


FIGURE 8. Algorithms Comparison - Average Objective Function.

3) verify if the results of each algorithm follow a normal distribution using the Kolmogorov-Smirnov test [34];
4) apply the Kruskal-Wallis test [35] among the algorithms samples to assess whether or not their population medians differ (whether three independent samples had been selected from populations with an identical distribution).
5) apply Wilconxon test [34], [35] among the algorithm of best median and all others to find the best.

As shown in Table 4, the results of the Kolmogorov-Smirnov test for the three algorithms were $h = 1$, which rejects the null hypotheses. Therefore, the samples obtained do not follow a normal distribution, and non-parametric tests must be used.

Kruskal-Wallis test was applied to determine whether at least one algorithm showed a different median and also whether a performance comparison could be made between them. The null hypothesis (Ho) assumes independent samples from two or more groups provide distributions with equal medians. If $h = 1$, the null hypothesis is rejected. Therefore, the algorithm of lowest median should be compared with the others by Wilconox test for proving its best performance in solving the CPP problem.

Wilcoxon Rank Sum was applied to find the algorithm of best performance. The null hypothesis (Ho) assumes the two independent samples derive from distributions with equal medians. If $h = 1$, the null hypothesis is rejected. Therefore, the algorithm of lower median shows better performance. Otherwise, no conclusion can be drawn.

The Wilcoxon test result was $h = 1$ for eight instances of the problem, and ACS-EM showed a lower median, which indicates better performance (see Table 4). No conclusions could be drawn in the last two instances from the comparison of ACS and ACS-EM. However, since ACS-EM performed better than PSO for all instances, we can conclude its performance is higher than that of ACS and PSO in solving the problem of quantity and location of SDN controllers in D2D communications management.

TABLE 4. Hypothesis test results.

λ (req/ms)	Algorithm	Kolmogorov-Smirnov Test (h)	Wilcoxon Rank Sum Test (h)	Median
0.1	ACS-EM	1.0	ACS-EM - ACS => 1.0 ACS-EM - PSO => 1.0	**0.0505**
	ACS	1.0		0.1505
	PSO	1.0		0.2422
0.2	ACS-EM	1.0	ACS-EM - ACS => 1.0 ACS-EM - PSO => 1.0	**0.2421**
	ACS	1.0		0.2431
	PSO	1.0		0.2480
0.3	ACS-EM	1.0	ACS-EM - ACS => 1.0 ACS-EM - PSO => 1.0	**0.2872**
	ACS	1.0		0.2987
	PSO	1.0		0.2987
0.4	ACS-EM	1.0	ACS-EM - ACS => 1.0 ACS-EM - PSO => 1.0	**0.3177**
	ACS	1.0		0.3185
	PSO	1.0		0.3198
0.5	ACS-EM	1.0	ACS-EM - ACS => 1.0 ACS-EM - PSO => 1.0	**0.3171**
	ACS	1.0		0.3341
	PSO	1.0		0.3341
0.6	ACS-EM	1.0	ACS-EM - ACS => 1.0 ACS-EM - PSO => 1.0	0.3427
	ACS	1.0		0.3427
	PSO	1.0		0.3427
0.7	ACS-EM	1.0	ACS-EM - ACS => 1.0 ACS-EM - PSO => 1.0	0.3513
	ACS	1.0		0.3513
	PSO	1.0		0.3596
0.8	ACS-EM	1.0	ACS-EM - ACS => 1.0 ACS-EM - PSO => 1.0	0.3494
	ACS	1.0		0.3494
	PSO	1.0		0.3494
0.9	ACS-EM	1.0	ACS-EM - ACS => 1.0 ACS-EM - PSO => 1.0	**0.3540**
	ACS	1.0		0.3620
	PSO	1.0		0.3720
1.0	ACS-EM	1.0	ACS-EM - ACS => 1.0 ACS-EM - PSO => 1.0	**0.3524**
	ACS	1.0		0.3566
	PSO	1.0		0.3582

The computational complexities of the three algorithms were analyzed, and are expressed as:

$$ACS-EM \;=>\; O(iteration\_max * [ant\_quantity * bl + cl^2]) \tag{26a}$$

$$ACS \;=>\; O(iteration\_max * [ant\_quantity * bl + cl^2]) \tag{26b}$$

$$PSO \;=>\; O(iteration\_max * particle\_quantity * cl) \tag{26c}$$

The analysis revealed the computational complexity of PSO is lower than that of ACS and ACS-EM. However, ACS-EM showed better performance in solving the problem, according to statistical tests, with characteristics that enable a more efficient exploration of the search space, thus avoiding sub-optimal solutions.

Due to the focus of external memory addressed in ACS-EM, in certain cases, ants do not make random decisions, and choose solution components influenced by the values recorded in that memory. Since it stores information specific to the search history from the beginning of the algorithm, it effectively focuses on regions of the search space not visited (not recorded in memory and with values reflecting this), or, instead, on those already visited and that are promising (also recorded in memory and with values that

TABLE 5. The list of symbols and notations used in this paper.

Symbol	Description	Symbol	Description
$B = \{1, 2, \ldots, b\}$	set of eNBs	$S = \{1, 2, \ldots, s\}$	set of SGWs
b_{c_i}	number of eNBs managed by controller c_i	z, z_0	ACS-EM algorithm parameter
$c1$	PSO algorithm parameters	Z	random variable
$c2$	PSO algorithm parameters	SINR_d	received SINR of D_d^R
$C = \{1, 2, \ldots, c\}$	set of controllers	SINR_d^t	SINR threshold of D_d^R
C_c	cost function associated with the number of controllers	SINR_u	received SINR of eNB when CUE_u transmits
C_{D2D}	number of D2D pairs allowed in each eNB	SINR_u^t	SINR threshold when CUE_u transmits
$D = \{1, 2, \ldots, d\}$	set of D2D pairs	t_{bc} or t_{bl}	propagation latency between the eNB and the controller c or the controller located in l
D_d^T	DUE transmitter	t_c	average response time of the SDN controller
D_d^R	DUE receiver	t_{QoS}	QoS time
E	construction graph edges	t_r	response time of a request
f	Jain's index	T_r	cost function associated with the response time of a request
F	cost function associated with Jain's index	$U = \{1, 2, \ldots, u\}$	set of CUEs
$g_{Tx,Rx}$	channel gain from transmitter Tx to receiver Rx	$V = \{1, 2, \ldots, v\}$	set of vectors
$G_C(V, E)$	construction graph	w_{min}, w_{max}	PSO algorithm parameters
$I_{Tx,Rx}$	interference received at Rx from Tx	$\Omega = \{1, 2, \ldots, J\}$	set of nodes that belong to the path traveled by the *best-so-far* ant
K	SDN controller capacity	α	ACS-EM algorithm parameter
$L = \{1, 2, \ldots, l\}$	set of possible locations of the controllers	β	ACS-EM algorithm parameter
L_q	average queue length	$\gamma_c, \gamma_T, \gamma_\pi, \gamma_F$	weights of each cost function
m	number of controller cores	δ	pheromone evaporation coefficient
$M(\cdot, \cdot)$	a matrix	η	heuristic or desirability information
N_i^k	neighborhood reachable by the ant	λ	average request rate received in the SDN controller
OF	Objective Function	λ_c	average request rate received by the controller
p_K	probability of K requests in the system	λ_{eff}	effective request rate in the SDN controller
p_0	probability of no requests in the controller	λ_{lost}	lost requests rate in the SDN controller
p_z	probability distribution in ACS-EM algorithm	$\lambda_{lost}^{threshold}$	threshold of lost requests rate
P_D	transmitted power of DUEs	μ	processing rate of each core
P_U	transmitted power of CUEs	ξ	constant factors that specify the increase/decrease rate of the cost function $T(t_{r_i})$
q, q_0	ACS-EM algorithm parameter	π_C^{avglat}	cost function associated with the controller location
r, r_0	ACS-EM algorithm parameter	ρ	controller utilization rate
R_I	ACS-EM algorithm rule (intensification)	σ^2	thermal noise power
R_D	ACS-EM algorithm rule (diversification)	τ	amount of pheromone
$\triangle\tau_{ij}$	increment in the pheromone	$\triangle u$	set of D2D pairs that reuse RBs allocated for CUE_u

reflect it). Such uses of memory reflect the intensification and diversification mechanisms that avoid premature convergence and achieve higher performance [8] against optimization problems of particular characteristics, such as the one discussed in this article.

Figure 9 displays the computational complexity (number of operations performed in the execution of the algorithms) based on the number of both eNBs and iterations. According to the results, PSO shows the lowest complexity, as discussed above. Regarding the algorithms based on ant colony, ACS-EM showed lower computational complexity than ACS.

Observations revealed the difference between the number of computations for different iterations increases along with the number of eNB, which demonstrates the numbers of eNB and iterations increase computational complexity. The results

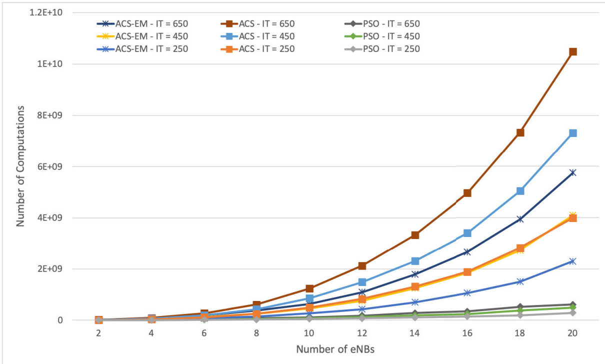


FIGURE 9. Computational complexity curve with number of eNBs for different numbers of iterations.

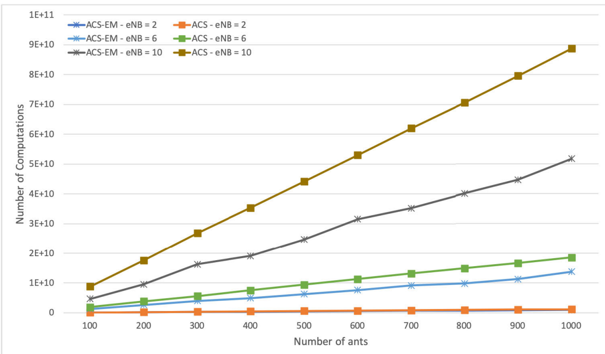


FIGURE 10. Computational complexity versus number of ants plot for 2, 6 and 10 eNBs.

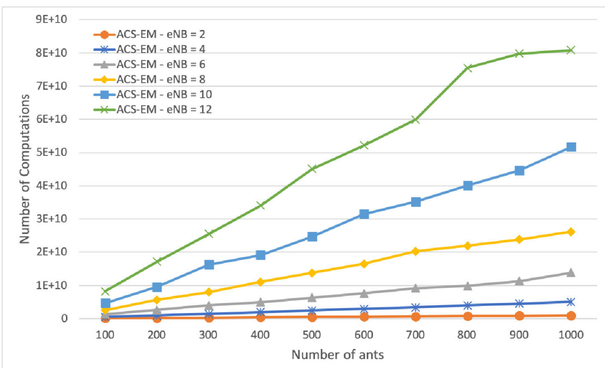


FIGURE 11. Computational complexity for ACS-EM versus number of ants plot for 2, 4, 6, 8, 10 and 12 eNBs.

also show the polynomial behavior of the computational complexity of ACS and ACS-EM.

Figures 10 and 11 display the computational complexity based on the *ant_quantity* parameter and eNBs quantity in the cellular network. Figure 10 shows the computational complexity of ACS is higher than that of ACS-EM. According to the results, the computational complexity highly increases when networks of larger extents (larger number of network features) are considered. Figure 11 shows the variation in the number of ants did not increase significantly for networks with 2, 4 and 6 eNB. The increase is more pronounced for larger networks.

VIII. CONCLUSION

The controller placement problem was analyzed for an LTE-A architecture with SDN-based D2D communications management, and two models were used to evaluate the effect of D2D communications management on the number of SDN controllers used in a cellular network. An interference model was used to obtain the maximum number of D2D communications allowed in the network, considering radio resource allocation. A model based on the queueing theory was used for SDN controllers to obtain their average response time.

The solution of the problem was then divided into two steps. In the first, a greedy algorithm is used to solve the resource allocation problem. In the second, we have proposed an approach for the ant colony system with external memory metaheuristic for the evaluation of CPP.

The objective was to minimize the response time considering the SDN controller average response time and propagation latency among the eNB and the controllers, with some interference constrains. *C#* simulations checked the effect of D2D communications on the location of SDN controllers. The ACS-EM algorithm yielded acceptable and accurate results in the different scenarios evaluated, in comparison to ACS and a discrete version of PSO metaheuristics.

As future work, we intend to solve the CPP considering other metrics, such as reliability and distance between controllers, and use network emulators, such as Mininet, to evaluate the effectiveness of the proposed system. We also intend to evaluate the influence of different radio resource allocation methods on CPP, and aspects related to traffic differentiation for multimedia applications. The effect of users' mobility on D2D communications and its influence on SDN controllers will also be considered.

ACKNOWLEDGMENT

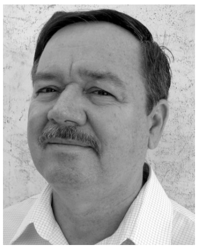
The authors acknowledge the financial support from Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES) to Yarisley Peña Llerena through Programa de Pós-Graduação em Engenharia Elétrica (PPGEE) of the University of Brasília.

REFERENCES

[1] E. Dahlman, G. Mildh, S. Parkvall, J. Peisa, J. Sachs, and Y. Selén, "5G radio access," *Ericsson Rev.*, vol. 6, pp. 2–7, Jun. 2014.
[2] E. Hossain and M. Hasan, "5G cellular: Key enabling technologies and research challenges," *IEEE Instrum. Meas. Mag.*, vol. 18, no. 3, pp. 11–21, Jun. 2015.
[3] W. H. Chin, Z. Fan, and R. Haines, "Emerging technologies and research challenges for 5G wireless networks," *IEEE Wireless Commun.*, vol. 21, no. 2, pp. 106–112, Apr. 2014.
[4] A. Asadi and V. Mancuso, "WiFi direct and LTE D2D in action," in *Proc. IFIP Wireless Days*, Nov. 2013, pp. 1–8.
[5] A. V. Bastos and D. C. Da. (Sep. 2017). *Minicurso-Comunicação D2D Para 5G de Arquiteturas de Redes Celulares: Da Teoria à Prática*. Accessed: Jun. 29, 2018. [Online]. Available: https://www.researchgate.net
[6] B. Heller, R. Sherwood, and N. McKeown, "The controller placement problem," in *Proc. 1st Workshop Hot Topics Softw. Defined Netw. (HotSDN)*, New York, NY, USA, 2012, pp. 7–12.

[7] 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," in *Proc. IEEE Region Conf. (TENCON)*, Nov. 2017, pp. 3123–3128.
[8] G. M. Jaradat, A. Al-Badareen, M. Ayob, M. Al-Smadi, I. Al-Marashdeh, M. Ash-Shuqran, and E. Al-Odat, "Hybrid elitist-ant system for nurse-rostering problem," *J. King Saud Univ.-Comput. Inf. Sci.*, vol. 31, no. 3, pp. 378–384, 2019.
[9] W. von Thienen and T. J. Czaczkes, "Do ants use ant colony optimization?" in *Shortest Path Solvers. From Software to Wetware*. Cham, Switzerland: Springer, 2018, pp. 265–291.
[10] M. F. Bari, A. R. Roy, S. R. Chowdhury, Q. Zhang, M. F. Zhani, R. Ahmed, and R. Boutaba, "Dynamic controller provisioning in software defined networks," in *Proc. 9th Int. Conf. Netw. Service Manage. (CNSM)*, Oct. 2013, pp. 18–25.
[11] Y. Hu, W. Wendong, X. Gong, X. Que, and C. Shiduan, "Reliability-aware controller placement for software-defined networks," in *Proc. IFIP/IEEE Int. Symp. Integr. Netw. Manage. (IM)*, May 2013, pp. 672–675.
[12] D. Hock, M. Hartmann, S. Gebert, M. Jarschel, T. Zinner, and P. Tran-Gia, "Pareto-optimal resilient controller placement in SDN-based core networks," in *Proc. 25th Int. Teletraffic Congr. (ITC)*, Sep. 2013, pp. 1–9.
[13] H. K. Rath, V. Revoori, S. M. Nadaf, and A. Simha, "Optimal controller placement in software defined networks (SDN) using a non-zero-sum game," in *Proc. IEEE Int. Symp. World Wireless, Mobile Multimedia Netw. (WoWMoM)*, Jun. 2014, pp. 1–6.
[14] L. Liao and V. C. M. Leung, "Genetic algorithms with particle swarm optimization based mutation for distributed controller placement in SDNs," in *Proc. IEEE Conf. Netw. Function Virtualization Softw. Defined Netw. (NFV-SDN)*, Nov. 2017, pp. 1–6.
[15] V. Ahmadi and M. Khorramizadeh, "An adaptive heuristic for multi-objective controller placement in software-defined networks," *Comput. Elect. Eng.*, vol. 66, pp. 204–228, Feb. 2018.
[16] 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)," *Comput. Commun.*, vol. 108, pp. 94–109, Aug. 2017.
[17] 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," in *Proc. 26th Int. Conf. Comput. Commun. Netw. (ICCCN)*, Jul./Aug. 2017, pp. 1–9.
[18] J. Dai, J. Liu, Y. Shi, S. Zhang, and J. Ma, "Analytical modeling of resource allocation in D2D overlaying multihop multichannel uplink cellular networks," *IEEE Trans. Veh. Technol.*, vol. 66, no. 8, pp. 6633–6644, Aug. 2017.
[19] A. Abrardo and M. Moretti, "Distributed power allocation for D2D communications underlaying/overlaying OFDMA cellular networks," *IEEE Trans. Wireless Commun.*, vol. 16, no. 3, pp. 1466–1479, Mar. 2017.
[20] H. Sun, M. Sheng, X. Wang, Y. Zhang, J. Liu, and K. Wang, "Resource allocation for maximizing the device-to-device communications underlaying LTE-Advanced networks," in *Proc. IEEE/CIC Int. Conf. Commun. China-Workshops (CIC/ICCC)*, Aug. 2013, pp. 60–64.
[21] S.-A. Ciou, J.-C. Kao, C. Y. Lee, and K.-Y. Chen, "Multi-sharing resource allocation for device-to-device communication underlaying 5G mobile networks," in *Proc. IEEE 26th Annu. Int. Symp. Pers., Indoor, Mobile Radio Commun. (PIMRC)*, Aug./Sep. 2015, pp. 1509–1514.
[22] K.-Y. Chen, J.-C. Kao, S.-A. Ciou, and S.-H. Lin, "Joint spectrum reuse and power control for multi-sharing device-to-device communication," 2017, *arXiv:1702.06780*. [Online]. Available: https://arxiv.org/abs/1702.06780
[23] X.-P. Zhai, G. Lei, and N. Zhang, "Resource allocation and power control for D2D communication underlaying LTE-advanced networks," in *Proc. Int. Conf. Commun., Netw. Artif. Intell. (CNAI)*, Beijing, China, Apr. 2018.
[24] Y.-J. Liang and Y.-S. Lin, "A non-iterative resource allocation strategy for device-to-device communications in underlaying cellular networks," *Wireless Netw.*, vol. 23, no. 8, pp. 2485–2497, Nov. 2017.
[25] J. F. Shortle, J. M. Thompson, D. Gross, and C. M. Harris, *Fundamentals of Queueing Theory* (Wiley Series in Probability and Statistics), 5th ed. Hoboken, NJ, USA: Wiley, 2018, doi: 10.1002/9781119453765.
[26] L. Yao, P. Hong, W. Zhang, J. Li, and D. Ni, "Controller placement and flow based dynamic management problem towards SDN," in *Proc. IEEE Int. Conf. Commun. Workshop (ICCW)*, Jun. 2015, pp. 363–368.
[27] S. Lange, S. Gebert, T. Zinner, P. Tran-Gia, D. Hock, M. Jarschel, and M. Hoffmann, "Heuristic approaches to the controller placement problem in large scale SDN networks," *IEEE Trans. Netw. Service Manag.*, vol. 12, no. 1, pp. 4–17, Mar. 2015.
[28] R. K. Jain, D.-M. W. Chiu, and W. R. Hawe, "A quantitative measure of fairness and discrimination for resource allocation in shared computer system," Eastern Res. Lab., Digit. Equip. Corp., Hudson, MA, USA, Tech. Rep. DEC-TR-301, 1984, pp. 1–38.
[29] K. Jain, J. Padhye, V. N. Padmanabhan, and L. Qiu, "Impact of interference on multi-hop wireless network performance," *Wireless Netw.*, vol. 11, no. 4, pp. 471–487, Jul. 2005.
[30] X. Yi and G. Caire, "Topological interference management with decoded message passing," *IEEE Trans. Inf. Theory*, vol. 64, no. 5, pp. 3842–3864, May 2018.
[31] M. Dorigo and T. Stützle, "Ant colony optimization: Overview and recent advances," in *Handbook of Metaheuristics*. Cham, Switzerland: Springer, 2019, pp. 227–263.
[32] 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," Dept. Informática, Universidad Nacional de San Luis Facultad de Ciencias Físico Matemáticas y Naturales, San Luis, Argentina, Apr. 2010.
[33] W.-N. Chen, J. Zhang, H. S. H. Chung, W.-L. Zhong, W.-G. Wu, and Y.-H. Shi, "A novel set-based particle swarm optimization method for discrete optimization problems," *IEEE Trans. Evol. Comput.*, vol. 14, no. 2, pp. 278–300, Apr. 2010.
[34] K. M. Ramachandran and C. P. Tsokos, *Mathematical Statistics With Applications*. Amsterdam, The Netherlands: Elsevier, 2009.
[35] D. Wackerly, W. Mendenhall, and R. L. Scheaffer, *Mathematical Statistics With Applications*. Boston, MA, USA: Cengage, 2014.

YARISLEY PEÑA LLERENA received the bachelor's degree in telecommunications and electronics engineering from the Polytechnic Institute José Antonio Echeverría (ISPJAE), Havana, Cuba, in 2010, and the master's degree in electrical engineering from the Universidade de Brasília, Brasília, Brazil, in 2016, where she is currently pursuing the Ph.D. degree and works with handoff management and data offloading in heterogeneous networks. She was with the Department of Medical Equipment, Central Institute of Digital Research (ICID), Havana, from 2010 to 2014. Her research interests include wireless and heterogeneous networks.



PAULO R. L. GONDIM received the bachelor's degree in computer engineering, in 1987, the master's degree in systems and computing from the Instituto Militar de Engenharia, Rio de Janeiro, in 1992, and the Ph.D. degree in electrical engineering from the Pontifícia Universidade Católica do Rio de Janeiro, in 1998. Since 2003, he has been with the Universidade de Brasília, Brasília, Brazil. He has authored over 70 articles in international periodicals and events and advised 28 master's degree dissertations and three doctoral theses. Prof. Gondim has been a member of Technical Program Committees of several technical-scientific events and contributed as a reviewer of articles submitted to high-quality periodicals and events. He is also a member of the Editorial Advisory Boards of four international journals and has experience in Computer Science, particularly in wireless networking, video streaming, performance evaluation, and quality of service/quality of experience assessment.