Near-Optimal Robust Virtual Controller Placement in 5G Software Defined Networks

Ehsan Tohidi, Saeedeh Parsaeefard, Mohammad Ali Maddah-Ali, *Senior Member, IEEE*, Babak Hossein Khalaj, and Alberto Leon-Garcia, *Life Fellow, IEEE*

***Abstract*—Fifth generation (5G) wireless networksare characterized by applying the software defined networking (SDN) and network function virtualization (NFV) concepts to provide higher reliability and scalability, and lower latency for advent data-hungry services. In the initial version of SDN based 5G, a centralized architecture for SDN controller has been proposed, which inherently cannot fulfill those requirements simultaneously. This observation has led us toward distributed architectures, where multiple SDN controllers across the network constitute the control layers. Furthermore, to fully achieve the network flexibility and agility offered by SDN, and also improve resource utilization and consequently to reduce network cost, we incorporate virtualized SDN (vSDN) controllers, enabled through the NFV technology. In vSDN with distributed architecture, controller placement is one of the main challenges, which is an NP-hard combinatorial optimization problem with conflicting objectives: 1) maximizing the reliability of network with low latency; 2) minimizing the total cost of implementation, which can be interpreted as reducing the number of vSDN controllers. In order to achieve such a compromise, in this work, we formulate the problem of vSDN controller placement and introduce some performance metrics to model the latency of communication between controller-switch pairs and also robustness against vSDN controller failures in a network. The introduced formulation is some sorts of submodular optimization, which is exploited to develop several algorithms. Simulations results demonstrate that, with a negligible performance loss, a significant reduction in the number of vSDN controllers is achievable. Finally, latency, cost, and robustness trade-offs are investigated.**

***Index Terms*—5G, robustness, submodularity, VSDN controller placement.**

Manuscript received March 18, 2020; revised November 23, 2020; accepted January 7, 2021. Date of publication March 25, 2021; date of current version July 7, 2021. B. Hossein Khalaj work was supported in part by a grant from the Institute for Research in Fundamental Sciences (IPM). Recommended for acceptance by Prof. Falko Dressler. *(Corresponding author: Ehsan Tohidi.)*

Ehsan Tohidi is with the Department of Telecommunication Systems, Technical University of Berlin, 10623 Berlin, Germany (e-mail: tohidi@tu-berlin.de).

Saeedeh Parsaeefard and Alberto Leon-Garcia are with the Department of Electrical, and Computer Engineering, University of Toronto, Toronto, ON M5S 3G4, Canada (e-mail: saeede.parsaeefard@gmail.com; alberto.leongarcia@utoronto.ca).

Mohammad Ali Maddah-Ali is with the Department of Electrical Engineering, Sharif University of Technology, Tehran 11365-11155, Iran (e-mail: maddah_ali@sharif.edu).

Babak Hossein Khalaj is with the Electrical Engineering Department, Sharif University of Technology, Tehran 11365-11155, Iran, also with the School of Computer Science, Institute for Research in Fundamental Sciences, Tehran 19395-5746, Iran (e-mail: khalaj@sharif.edu).

Digital Object Identifier 10.1109/TNSE.2021.3068975

I. Introduction

A. Motivation and Background

MOVING towards fifth generation (5G) wireless networks have become realistic through the integration of the two technology enablers: network function virtualization (NFV) and software defined networking (SDN) [1]. NFV reduces both capital and operational costs of the network by introducing virtualized network functions (VNFs), implemented on general-purpose servers or high volume servers (HVS) [2]. On the other hand, through decoupling control and data planes in SDN, flexibility, agility, and automatic configuration of networks have been addressed [3]. Taking these technologies into account, 5G is facilitated via a programmable and flexible structure, where based on network states and end-users' traffic, each physical, virtual, networking, and computing resources of network, e.g., (HVS), and even each required entities of network, e.g., SDN controllers, can be dynamically managed to satisfy the requests of services and to utilize the network resources in the most efficient manner [4], [5]. There exist a lot of recent works aiming to address different aspects of this context from both academia and industrial bodies, e.g., [4]–[7]. In this paper, our focus is on the problem of SDN controller placement, which plays an important role in the architecture of 5G [8].

SDN was originally proposed with a centralized controller architecture that provides optimal routing and control for the network [9]. However, this central architecture suffers from a major problem, which is lack of scalability [10]. In fact, as the size of the network grows, the distance between the controller and the switches under control grows, which leads to larger delays in delivery of control messages and may fail the system to satisfy the requirements for quality of service (QoS) represented as a maximum tolerable latency [10]. Furthermore, central architecture is not robust against failure of the SDN controller (i.e., the problem of single point of failure). Therefore, distributed SDN controllers are becoming a compulsory architecture for future networks. Although a distributed scheme promises performance improvements in terms of latency and robustness, its gain fully depends on how the SDN controllers are placed within the network [11]. The placement of the SDN controllers will also affect the configuration of the switches. Following the SDN controller placement, switches are partitioned into several groups, where each switch belongs to one specific SDN control zone [11]. Therefore, the problems of SDN controller placement in the network are usually represented as optimization problems, which aim to maximize speed in delivering the controlling messages of network between SDN controllers and SDN switches, e.g., to close SDN controller-switch pairs, or to minimize the network costs, e.g., placement cost of SDN controllers, subject to the network limitations, e.g., candidate locations for placement, and QoS constraints of services, e.g., latency. Of course, inter-controller communication can also influence the end to end latency of the network [12]. However, in this paper, we do not consider controller-to-controller communication for three main reasons. First, the controller-switch delay is the dominant factor of total communication latency as it occurs quite often (i.e., every time that the routing rule does not exist in the lookup table available at the switch repository) and also, the number of switches is much more than the number of controllers (several switches are assigned to each controller) which in turn, intensifies the effect of controllers-switches communication, while the controller-to-controller communications only take place when a message is required to be passed to switches controlled by the others. The latter is essentially rare and plays a less important role in forming the total latency. Second, if one takes the inter-controller communication delay into account, then, a properly weighted summation of controller-switch and controller-to-controller communication latencies should be employed. This in turn requires further studies to investigate optimal weighting. Existing works in this literature apply a heuristic weighting or simply weigh both equally which are obviously not optimal, particularly considering the first point. Third, in this paper, we shed light on some other aspects of the controller placement problem including robustness, non-uniform placement cost, and the trade-off between robustness and latency, while analytical performance guarantee with low computational complexity algorithm is provided. Thus, in order to make the illustration easier and provide intuitions, such minor simplifications are chosen.

On the other hand, in a distributed architecture for SDN controllers, the network should be resilient against the SDN controller failure. This calls for a robust controller placement, where for each switch, a sequence of SDN controllers in a specific order is assigned and when each SDN controller fails, the next one on the sequence manages the switch. In this context, the sequence of SDN controllers should be determined based on the probability of SDN controllers failure in the procedure of SDN controller placement. Clearly, increasing the number of SDN controllers and deploying the backup SDN controllers will decrease the latency and improve the robustness. However, these improvements come with the extra cost of implementation from a network provider's perspective.

One approach to tackle this issue is to deploy SDN controller based on NFV concept in 5G, where each SDN controller is implemented as one VNF running on a specific server, called virtualized SDN (vSDN) controller [1]. Depending on the tasks of each SDN controller, its related VNF requires a specific set of resources such as storage and processing capacity. In this context, providing these resources imposes severe costs of implementation in 5G networks and should be considered in the optimization problem of vSDN placement, where these costs are functions of diverse parameters such as the location of servers and traffic demand of network services. Consequently, vSDN controller placement in a realistic 5G scenario faces a key trade-off among network performance metrics such as robustness and cost of implementation. In this paper, our main goal is to address this problem for the robust vSDN controller placement.

B. Related Works

As mentioned, problems of controller placement in SDN based networks are aimed to assign the switches to the SDN controllers to maximize objective functions of the network, subject to the network limitations and requested QoS. SDN controller placement has been studied in the literature from different perspectives [9]–[11], [13]–[54].

The most important objective for SDN controller placement is to reduce network latency [14]–[21]. For instance, in [14], the impact of SDN controller placement on both average and maximum latency is studied where the objectives are to minimize the maximum and average latency, and the minimum k-center and k-median algorithms are employed to find the placement, respectively. In [15], a comprehensive mathematical model for the optimal placement of the controllers to reduce network latency in SDN is proposed. The model looks at the optimal planning for the SDN, which simultaneously determines the optimal number, the location, and the type of controllers, and an exhaustive search is used to find the solution.

Resilience and robustness against link and node failures in networks have also been taken into account in a number of works [10], [20], [22]–[30]. A recent heuristic controller placement algorithm called Varna-based optimization (VBO) is presented in [20]. Although SDN controllers' failure is studied in this work, it is not considered during the designing procedure which makes it less resilient. In [22], joint switch-controller latency and SDN controllers failure are considered where to provide resilience, multiple SDN controllers are assigned to each switch. One problem with that approach is the reservation of multiple resources, which increases the cost of network implementation. In [22], two placement algorithms are proposed: The first algorithm is based on an exhaustive search with a non-polynomial computational complexity and the second one is a heuristic algorithm with a complexity of $O(N^{2Q+3})$ where Q is the number of SDN controllers assigned to each switch. Even the heuristic algorithm becomes infeasible when the number of nodes in the network exceeds a few thousand. Moreover, no bounds on the performance are given. Another recent model is presented in [30], where in addition to latency, SDN controllers failure is also taken into account, where several SDN controllers are assigned to each switch in descending order (in terms of latency). If the first SDN controller fails, which is usually the best SDN controller derived from the SDN controller placement problem, the second SDN controller is assigned and so on. In [30], simulated annealing is proposed to solve the optimization problem, however, there is no guarantee on the optimality of the solution. Table I presents a summary of some of the related works.

While the past works studied the SDN controller placement problem from diverse prescriptive, still due to the NP-hard nature of this type of problems, proposing an efficient algorithm to solve the problem is of importance where its performance bound e.g., the gap between its solution to the global

TABLE I
COMPARISON OF LITERATURE OF CONTROLLER PLACEMENT

Ref.	Objective	Constraint	Robust	Solution
[14]	latency	num. of controllers	no	exhaustive search
[15]	placement cost	latency & capacity	no	exhaustive search
[17]	outage prob. & latency	throughput	yes	K-Median
[18]	latency	num. of controllers	no	exhaustive search
[19]	latency	num. of controllers & capacity	yes	K-Means
[20]	latency	num. of controllers & capacity	yes	heuristic
[22]	num. of controllers	latency & capacity	yes	heuristic
[24]	num. of controllers	latency	yes	exhaustive search
[25]	latency	num. of controllers	yes	exhaustive search
[26]	latency	num. of controllers & capacity	yes	exhaustive search
[27]	num. of controllers	load balance	yes	exhaustive search
[29]	reliability	num. of controllers	yes	simulated annealing
[30]	latency	num. of controllers	yes	simulated annealing
[31]	num. of controllers	capacity	no	heuristic
[32]	delay & load balance	switch-controller assignment	no	Genetic algorithm

optimum can be quantified. In this paper, we will aim to cover this point for our proposed problem.

C. Contributions and Novelties

In this paper, we present a robust scheme for vSDN controller placement, where failure of vSDN controllers is taken into account, and our objective is to reduce controller-switch communication latency. Due to the virtualized nature of vSDN, there is a higher chance to have more unavailable vSDN controllers considering scale up/down or migration of vSDN controllers. Consequently, we define a new definition for the robustness of this type of networks as follows: a network is robust if the number of unavailable vSDN controllers does not exceed a predefined threshold. We consider the problem of vSDN controller placement with a constraint on the number of vSDN controllers. Note that the number of vSDN controllers in the network directly impacts the cost of network implementation.

We present the problem for two cases: 1) uniform placement cost where the placement costs of all vSDN controllers are similar; 2) non-uniform placement cost where the placement costs of vSDN controllers are different and depend on the location and servers. The proposed model includes a parameter, which determines the number of tolerable failures in vSDN controllers and provides a robust vSDN controller placement solution. We introduce a surrogate function for latency, which measures the closeness between controller-switch pairs. Therefore, we aim to maximize the closeness function. We show that the vSDN placement problem is a combinatorial optimization and NP-hard. Henceforth, we need a near-optimal solution with low computational complexity. We prove that closeness is a submodular function and thus a greedy algorithm achieves approximately optimum solutions within $1-1/e$ factor [55]. Using this approach, we can characterize the latency-cost trade-off for the proposed set-up. Moreover, we characterize how robustness is achieved at the price of higher latency or cost. This result explains a tripartite trade-off among latency, cost, and robustness. The proposed algorithms are with low computational complexity and realizable for large-scale networks. The contributions of this paper can be summarized as follows:

- Introducing controllers-switches closeness as a surrogate function of network latency.
- Formulating the problem of vSDN controller placement with robustness against controllers failures for both uniform and non-uniform placement costs.
- Proving that closeness is a submodular function.
- Establishing NP-hardness of the optimization problem and proposing robust vSDN controller placement solutions with near-optimal guarantees.
- Illustrating fundamental trade-offs among latency, cost, and robustness.
- Explaining the diminishing return of adding vSDN controllers that enables us to reduce the network cost significantly, at a negligible performance reduction.

D. Outline and Notations

The rest of the paper is organized as follows. The system model is introduced in Section II. In Section III, the optimization problem is formulated. Near-optimal vSDN controller placement algorithms are proposed in Section IV. Section V is dedicated to the numerical results, and finally, Section VI concludes the paper.

We adopt the notation of using boldface lower (upper) case for vectors $\mathbf{a}$ (matrices $\mathbf{A}$) and calligraphic upper case for sets $\mathcal{A}$. In addition, $\{0,1\}^{N\times M}$ is the set of $N\times M$ binary matrices. In addition, unit vector $\mathbf{e}^i$ is defined as $\mathbf{e}^i = [0,\ldots,0,1,0,\ldots,0]^T$ with a 1 in the ith coordinate. Finally, $\setminus$ is used as the operator of set subtraction.

II. System Model

We consider a network with a set of N nodes and a set of weighted links that connect a number of these nodes. Network nodes are assumed to be NFV nodes and capable of hosting

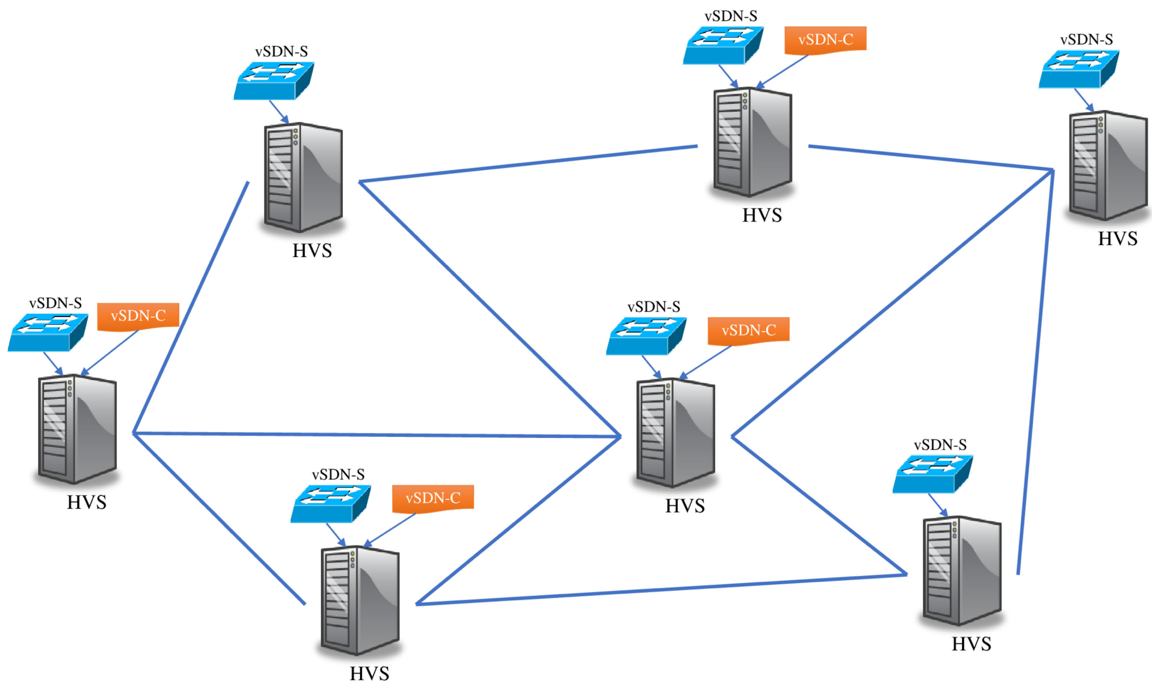

Fig. 1. A sample network configuration.

several VNFs (i.e., controllers, switches, and so on) in general. Set of nodes is denoted by $\mathcal{N} = \{1, \ldots, N\}$, where nodes are numbered from 1 to N. A link between two nodes is weighted proportionally to the distance (or communication delay) between those nodes. We denote the link's weight between the ith and jth nodes by $E_{i,j}$, where $\mathbf{E} \in \mathbb{R}^{N\times N}$ is the link weight matrix. For the sake of simplicity, un-directed links are assumed, and therefore we have $E_{i,j} = E_{j,i}$, i.e., $\mathbf{E}$ is symmetric .[1] If there is no link between two nodes, ∞ is considered for its weight.

A set $\mathcal{S}$ of N switches distributed across the network is given where a switch $s \in \mathcal{S}$ is placed at each node. Switches are assumed to be implemented as VNFs in host servers and all are active. In addition, $\mathcal{C} = \{v_1, \ldots, v_C\} \subseteq \mathcal{N}$ is the set of candidate vSDN controllers, where $v_c \in \mathcal{N}$ is the host node of the cth candidate vSDN controller. vSDN controllers are also assumed to be implemented as VNFs in the host server and consequently consume a certain amount of resources (i.e., processing and/or storage) and energy from the host server. Our aim is to select a subset of active vSDN controllers out of the vSDN controller pool $\mathcal{C}$ in order to control the network switches and perform routing for network flows. In Fig. 1, a sample network including 7 nodes is presented, where the high volume server (HVS) at each node can host a vSDN switch, while a subset of them are capable of hosting a vSDN controller (i.e., candidate vSDN controllers).

Besides the vSDN controller placement, we need to assign a vSDN controller to each switch. Since the communication delay between a controller-switch pair is proportional to their distance, which is the main driving factor of latency, for each switch, the closest active vSDN controller is assigned.

One of the important features in a 5G network is reliability and availability. Therefore, vSDN controllers, similar to other network components should have a robust placement against failures. We model the robustness by tolerating up to Q vSDN controller failures for some constant predefined Q. In other words, if a set of vSDN controllers with a size of less than or equal to Q vSDN controllers break down, the network can still continue to manage vSDN switches with almost the same efficiency. We consider this new definition of robustness in 5G due to the virtualized nature of vSDN controllers in the networks where due to the scale up/down or even immigration of VNFs from one HVS to another HVS, the chance of having a large number of unavailable vSDN controllers is high.

When the vSDN controller of one zone breaks down, all the switches in this zone migrate to the zone of the closest available vSDN controller, i.e., the second closest active vSDN controller. When the second closest vSDN controller fails, switches migrate to the zone of the third closest vSDN controller and so on. Hence, in a robust strategy of vSDN controller placement, the $Q+1$ closest vSDN controllers are considered. Therefore, during the procedure of vSDN controller placement, on top of the cost of the vSDN controller placement, the controller-switch closeness is also taken into account in the model.

[1] It is straightforward to extend the system model and adapt the proposed algorithms to networks with directed links. As explained in Sections III and IV, the problem formulation and the proposed algorithms depend on the shortest path between nodes. Thus, in a network with directed links, we only need to consider directed paths, taking into account that the distance from node i to node j is not necessarily equal to the distance from node j to node i.

III. Problem Formulation

The problem is comprised of two tasks: *vSDN controller placement* and *controller-switch assignment*. The vSDN controller placement problem is to select a set of nodes out of the candidates pool $\mathcal{C}$ and activate the vSDN controllers in servers of the selected nodes. We model this problem with a selection vector $\mathbf{w} \in \{0,1\}^C$ where w_c is 1(0) if the cth vSDN controller is active(inactive). Taking cost of vSDN controller placement into account, this task implies a total cost of $\sum_{c=1}^{C} w_c p_c$. The second task is the controller-switch assignment. For each switch, among the active vSDN controllers, we assign the active vSDN controllers with the shortest distances. In order to make the system robust against failure of up to Q vSDN controllers, the $Q+1$ closest vSDN controllers are assigned to each switch where Q of them are reserved to be employed in an ascending order (based on distance) in case of failure. Therefore, to determine the closest vSDN controllers, we run the shortest path algorithm [56] in the network graph to find the distance, i.e., closest path between vSDN controllers and switches, of all nodes pair where $D(i,j)$ is the shortest path between the ith and jth nodes. We denote the closeness of the qth closest vSDN controller in the set of active vSDN controllers to the switch s by $d_q(s|\mathbf{w})$, which is determined as follows

$$d_q(s|\mathbf{w}) = \frac{1}{D(v_{c_{s,q}}, s) + \epsilon}, \tag{1}$$

where $c_{s,q}$ is the index of the qth closest active vSDN controller (i.e., $w_{c_{s,q}} = 1$) to the switch s and ϵ is introduced to make the model consistent with zero distances, i.e., when both the switch and the active vSDN controller are placed in the same server in one physical location, and also the measure becomes less sensitive to small distances. One way to determine ϵ is based on network policies, e.g., it can be set equal to the acceptable delay of the network. If the number of active vSDN controllers is less than q, $d_q(s|\mathbf{w})$ is 0, i.e., $D(v_{c_{s,q}}, s) = \infty$.

Since the $Q+1$ closest vSDN controllers are assigned to each switch, the problem of controllers-switches assignment is dependent on the problem of vSDN controller placement, and our approach is to propose a joint solution. Hence, the objective function is to maximize the sum of the closeness of active vSDN controllers to the switches, subject to the given total placement budget B, such that the total cost of placement does not exceed B. Therefore, the objective function is

$$f(\mathbf{w}) = \sum_{q=1}^{Q+1} f_q(\mathbf{w}), \tag{2}$$

where $f_q(\mathbf{w}) = \sum_{s\in\mathcal{S}} \alpha_q d_q(s|\mathbf{w})$ is the closeness of the qth closest vSDN controllers in the network, and $\alpha_q, q = 1,\ldots,Q+1$, is the weight assigned to the qth closest vSDN controller. Since we expect that the failure probability of q vSDN controllers be higher than the probability of $q+1$ vSDN controllers failure, we set these weights in a descending order (i.e., $\alpha_1 \geqslant \cdots \geqslant \alpha_{Q+1}$) to give a higher priority to $d_q(s|\mathbf{w})$ with lower qs. Hence, the placement problem aiming at the maximization of controller-switch closeness can be formulated as

$$\max_{\mathbf{w}} \quad f(\mathbf{w}) \text{s.t.} \quad \text{C1: } \mathbf{w} \in \{0,1\}^{C\times 1}, \text{C2: } \sum_{c=1}^{C} w_c p_c \leqslant B, \tag{3}$$

where C1 enforces the selection to be Boolean and C2 is the constraint on the placement budget. In the following, we prove that the optimization problem in (3) is NP-hard.

Theorem 1: The optimization problem in (3) is NP-hard.

Proof: We prove that the placement problem aiming at the maximization of controller-switch closeness is a Knapsack problem, which is known to be NP-hard [57]. In order to demonstrate this equivalence, we reformulate (3) as follows

$$\begin{aligned} \max_{\mathbf{w}} \quad & \sum_{c=1}^{C}\sum_{s=1}^{N}\sum_{q=1}^{Q+1} \frac{1}{D(v_c, s)+\epsilon}\alpha_q x_{c,s,q} \\ \text{s.t.} \quad & \text{C1: } \mathbf{w} \in \{0,1\}^{C\times 1}, \\ & \text{C2: } \mathbf{x} \in \{0,1\}^{C\times N\times (Q+1)}, \\ & \text{C3: } \sum_{c=1}^{C} w_c p_c \leqslant B, \\ & \text{C4: } \sum_{q=1}^{Q+1} x_{c,s,q} \leqslant w_c, \quad \forall c, s, \\ & \text{C5: } \sum_{c=1}^{C} x_{c,s,q} \leqslant 1, \quad \forall s, q, \end{aligned} \tag{4}$$

where $x_{c,s,q}$ is a binary variable that determines whether the cth vSDN controller is considered as the qth closest vSDN controller of switch s. In addition, C4 guarantees that a vSDN controller is assigned to a switch for at most one q and such assignment is only possible when the vSDN controller is selected (i.e., $w_c = 1$). Moreover, C5 states that not more than one vSDN controller could be considered as the qth closest vSDN controller to switch s. It can be observed that (4) is a Knapsack problem with a linear objective, linear constraints, and binary variables, which proves that the problem is NP-hard [57]. ■

Theorem 1 proves that (3) is NP-hard. Here we solve this problem approximately using submodular optimization, which achieves the optimum solution within $1 - 1/e$ factor via greedy algorithms [58].

Based on the placement cost of the vSDN controllers, the vSDN controller placement problem in (3) is categorized into uniform and non-uniform placement cost scenarios. For the uniform scenario, the cost of placement of all vSDN controllers are the same, w.l.o.g., unit cost, and henceforth, the budget B in C2 is transformed into the cardinality constraint K on the maximum number of active vSDN controllers. In contrast, in the non-uniform cost scenario, we consider different placement cost for vSDN controllers where placement cost of vSDN controller c is p_c, and the sum of placement cost of selected vSDN controllers cannot exceed the total budget B. Clearly, the former is a special case of the latter class, i.e., for all $c = 1,\ldots,C$ set $p_c = 1$ and $B = K$. However, since the uniform cost class is a simpler optimization problem and more efficient algorithms exist for this class, we first solve the uniform cost problem and then proceed to the general case with the non-uniform placement cost of vSDN controllers.

In the next sections, we prove submodularity and propose the placement algorithms with the near-optimality guarantees for the optimization problem in (3).

IV. Near-Optimal Placement Algorithms for vSDN Controllers

In the beginning, we prove that the objective function $f(\mathbf{w})$ defined in (2) is a: 1) normalized function, i.e., $f(0)=0$, 2) monotonic function, i.e., if $\mathbf{w}^a=\mathbf{w}+\mathbf{e}^a$, then $f(\mathbf{w})\leqslant f(\mathbf{w}^a)$, and 3) submodular function, i.e., if $\mathbf{w}^a=\mathbf{w}+\mathbf{e}^a$, $\mathbf{w}^b=\mathbf{w}+\mathbf{e}^b$, and $\mathbf{w}^{ab}=\mathbf{w}+\mathbf{e}^a+\mathbf{e}^b$, then $f(\mathbf{w}^a)-f(\mathbf{w})\geqslant f(\mathbf{w}^{ab})-f(\mathbf{w}^b)$.

Theorem 2: $f(\mathbf{w})$ is a normalized and monotonic function.

Proof: Assume $\mathcal{A}$ be an arbitrary set of vSDN controllers, $a\in\mathcal{C}\setminus\mathcal{A}$, and consider $\mathbf{w}$ and $\mathbf{w}^a$ be the selection vectors corresponding to the vSDN controller sets $\mathcal{A}$ and $\mathcal{A}\cup\{a\}$, respectively, i.e., $\mathbf{w}^a=\mathbf{w}+\mathbf{e}^a$. Since for each $s\in\mathcal{S}$ and $q\in\{1,\ldots,Q+1\}$, we have $d_q(s|\mathbf{w}^a)\geqslant d_q(s|\mathbf{w})$, the following inequality holds

$$f(\mathbf{w}^a)-f(\mathbf{w})=\sum_{s\in\mathcal{S}}\sum_{q=1}^{Q+1}\alpha_q\big(d_q(s|\mathbf{w}^a)-d_q(s|\mathbf{w})\big)\geqslant 0, \quad (5)$$

which means that $f(\mathbf{w}^a)$ is a monotone function. For the case that the set of active vSDN controllers is empty, for each $s\in\mathcal{S}$ and $q\in\{1,\ldots,Q+1\}$, we have $d_q(s|\mathbf{w})=0$, and $f(0)=0$. Therefore, $f(.)$ is also a normalized function. ∎

Theorem 3: $f(\mathbf{w})$ is a submodular function.

Proof: See Appendix VIII. ∎

Based on Theorems 2 and 3, $f(\mathbf{w})$ is a normalized, monotone, submodular function. Therefore, by employing a greedy algorithm for (3), the solution can be derived where its near-optimality can be guaranteed [58]. In the sequel, we first propose a near-optimal greedy algorithm for the uniform cost problem and then proceed to the general case of vSDN controller placement with non-uniform costs.

A. Algorithm for Uniform Placement Cost of vSDN Controllers

In this case, for a given K, we need to select at most K vSDN controllers such that the maximum value of $f(\mathbf{w})$ is obtained. The algorithm starts with an empty set of vSDN controllers $\mathcal{A}=\emptyset$ and an all-zero selection vector $\mathbf{w}=\underline{0}$. At each iteration, among the un-selected set of vSDN controllers, we choose the vSDN controller providing the maximal marginal gain, i.e., $\underset{c\in\mathcal{C}\setminus\mathcal{A}}{\operatorname{argmax}}\ f(\mathbf{w}+\mathbf{e}^c)-f(\mathbf{w})$. The iterations are continued until K vSDN controllers are selected. The proposed algorithm is summarized in **Algorithm 1**.

Based on the Theorems 2 and 3, $f(\mathbf{w})$ is a normalized, monotone, submodular function. Therefore, **Algorithm 1** maximizes the objective function with a $1-1/e$ optimality guarantee [58].

B. Algorithm for Non-Uniform Placement Cost of vSDN Controllers

In this case, we consider vSDN controllers with non-uniform placement cost. The algorithm starts with an empty set of vSDN controllers $\mathcal{A}=\emptyset$ and a set of feasible unselected vSDN controllers $\mathcal{B}$, i.e., the vSDN controllers that do not exceed the budget constraint. At each iteration, among the feasible un-selected vSDN controllers, we select the vSDN controller with the maximum marginal gain and continue the iterations until no feasible vSDN controller exists. The proposed uniform cost algorithm for vSDN controllers placement with non-uniform costs is summarized in **Algorithm 2**.

It is shown that **Algorithm 2** can perform arbitrarily bad, as it ignores differences in placement cost [58]. In fact, this algorithm might make a mistake by adding an expensive vSDN controller with a marginal gain improvement with respect to the other vSDN controllers. In such examples, it is possible to select more vSDN controllers with the same total cost and a greater total gain. To overcome this weakness, we propose a gain-cost greedy algorithm that jointly considers placement cost and gain of vSDN controllers. The gain-cost greedy algorithm is similar to **Algorithm 2** except that for adding a new vSDN controller, we select the vSDN controller with the maximum gain-cost ratio as follows

---

Algorithm 1. The proposed uniform cost greedy algorithm for vSDN controller placement

Require: K, D, ϵ.
Ensure: $\mathcal{A}$, $\mathbf{w}$.
Initialization:
$\mathcal{A}=\emptyset$;

Greedy algorithm:

while $|\mathcal{A}|<K$ **do**
  $c^*=\underset{c\in\mathcal{C}}{\operatorname{argmax}}\ f(\mathbf{w}+\mathbf{e}^c)-f(\mathbf{w})$
  $\mathcal{A}=\mathcal{A}\cup\{c^*\}$;
  $\mathbf{w}=\mathbf{w}+\mathbf{e}^{c^*}$;
end while
$\mathcal{A}$ and $\mathbf{w}$ are the set of selected vSDN controllers and the selection vector, respectively.

Algorithm 2. The uniform cost greedy algorithm for vSDN controller placement of non-uniform cost

Require: $\mathbf{q}$, B, D, ϵ.
Ensure: $\mathcal{A}$, $\mathbf{w}$.
Initialization:
$\mathcal{A}=\emptyset$;
$\mathcal{B}=\{c|c\in\mathcal{C},p_c\leq B\}$;

Greedy algorithm:

while $|\mathcal{B}|>0$ **do**
  $c^*=\underset{c\in\mathcal{B}}{\operatorname{argmax}}\ f(\mathbf{w}+\mathbf{e}^c)-f(\mathbf{w})$
  $\mathcal{A}=\mathcal{A}\cup\{c^*\}$;
  $\mathbf{w}=\mathbf{w}+\mathbf{e}^{c^*}$;
  $B=B-p_{c^*}$;
  $\mathcal{B}=\{c|c\in\mathcal{C}\setminus\mathcal{A},p_c\leq B\}$;
end while
$\mathcal{A}$ and $\mathbf{w}$ are the set of selected vSDN controllers and the selection vector, respectively.

$$c^* = \underset{c \in \mathcal{B}}{\operatorname{argmax}} \frac{f(\mathbf{w} + \mathbf{e}^c) - f(\mathbf{w})}{p_c}. \quad (6)$$

Unfortunately, even the gain-cost algorithm can perform arbitrarily bad and there is no performance guarantee for this algorithm. However, running both the uniform cost and gain-cost algorithms and selecting the best of them, provides a $\frac{1-1/e}{2}$ optimality guarantee [58]. Since both algorithms are of linear complexity, running both the algorithms and selecting the best solution is also of linear complexity.

It should be noted that there is a more computationally complex algorithm that achieves a better optimality guarantee. Such an algorithm enumerates all the feasible vSDN controller subsets with the cardinality of 3 and separately augments each of them with a gain-cost algorithm. It is shown that the best of these sets and all the feasible subsets with cardinality 1 and 2, achieve the $1 - 1/e$ optimality guarantee [58]. We call this algorithm, 3-combinations gain-cost greedy algorithm. The proposed 3-combinations gain-cost greedy algorithm is summarized in **Algorithm 3**.

In the following, we calculate the computational complexity of the proposed algorithms. To select a new controller in **Algorithm 1**, we search among C candidate nodes and evaluate their effects on the other nodes which is accomplished with a complexity of $O(NQ)$. Therefore, the computational complexity of **Algorithm 1** is $O(KCNQ)$. For **Algorithm 2**, it is not clear that how many controllers will be selected as it depends on the budget B and cost of vSDN controllers. However, if we assume that **Algorithm 2** selects K controllers in total, it will also lead to the computational complexity of $O(KCNQ)$. Finally, **Algorithm 3** performs a similar procedure, however for $O(C^3)$ times. Thus, the computational complexity of **Algorithm 3** is $O(KC^4NQ)$.

Algorithm 3. The 3-combinations gain-cost greedy algorithm for vSDN controller placement

Require: $\mathbf{q}$, B, D, ϵ.
Ensure: $\mathcal{A}$, $\mathbf{w}$.
Initialization:
$f_{12} = f_3 = 0$;
for all $\mathcal{A}_{temp} \subseteq \mathcal{C}$ **do**
  if $|\mathcal{A}_{temp}| = 1$ or $|\mathcal{A}_{temp}| = 2$ **then**
    $\mathbf{w}_{temp} = \underline{0}$
    $B_{temp} = 0$
    for all $a \in \mathcal{A}_{temp}$ **do**
      $\mathbf{w}_{temp} = \mathbf{w}_{temp} + \mathbf{e}^a$
      $B_{temp} = B_{temp} + p_a$
    end for
    if $B_{temp} \leqslant B$ and $f(\mathbf{w}_{temp}) > f_{12}$ **then**
      $f_{12} = f(\mathbf{w}_{temp})$
      $\mathcal{A}_{12} = \mathcal{A}_{temp}$
      $\mathbf{w}_{12} = \mathbf{w}_{temp}$
    end if
  else if $|\mathcal{A}_{temp}| = 3$ **then**
    $\mathbf{w}_{temp} = \underline{0}$
    $B_{temp} = 0$
    for all $a \in \mathcal{A}_{temp}$
      $\mathbf{w}_{temp} = \mathbf{w}_{temp} + \mathbf{e}^a$
      $B_{temp} = B_{temp} + p_a$
    end for
    if $B_{temp} \leqslant B$
      $\mathcal{B} = \{c | c \in \mathcal{C} \setminus \mathcal{A}_{temp}, p_c \le B - B_{temp}\}$
      while $|\mathcal{B}| > 0$ **do**
        $c^* = \underset{c \in \mathcal{B}}{\operatorname{argmax}} \frac{f(\mathbf{w}_{temp} + \mathbf{e}^c) - f(\mathbf{w}_{temp})}{p_c}$
        $\mathcal{A}_{temp} = \mathcal{A}_{temp} \cup \{c^*\}$;
        $\mathbf{w}_{temp} = \mathbf{w}_{temp} + \mathbf{e}^{c^*}$;
        $B_{temp} = B_{temp} + p_{c^*}$;
        $\mathcal{B} = \{c | c \in \mathcal{C} \setminus \mathcal{A}_{temp}, p_c \le B - B_{temp}\}$;
      end while
      if $f(\mathbf{w}_{temp}) > f_3$ **then**
        $f_3 = f(\mathbf{w}_{temp})$
        $\mathcal{A}_3 = \mathcal{A}_{temp}$
        $\mathbf{w}_3 = \mathbf{w}_{temp}$
      end if
    end if
  end if
end for
if $f_{12} > f_3$ **then**
  $\mathcal{A} = \mathcal{A}_{12}$
  $\mathbf{w} = \mathbf{w}_{12}$
else
  $\mathcal{A} = \mathcal{A}_3$
  $\mathbf{w} = \mathbf{w}_3$
end if

V. Simulation Results

In this section, we evaluate the proposed algorithms and illustrate the fundamental trade-offs among different parameters of the problem. In particular, the resulting robustness-closeness trade-off is depicted. In the following subsections, we model the network with an Erdős-Rény graph [59] with the edge probability of 0.2. Unless specifically mentioned, we set the following parameters for the simulations: We use Monte Carlo simulations with 10 runs where the results are obtained by averaging over the 10 realizations. The number of nodes is assumed to be $N = 1000$ and nodes are randomly spanned (with a uniform distribution) in a rectangular area with a dimension of 1000 m by 1000 m. We also consider $\epsilon = 50$m and $\alpha_q = 1/q$ for $q = 1, \ldots, Q+1$.

A. Uniform Cost

In this section, we evaluate the proposed algorithm for the uniform case (i.e., **Algorithm 1**), where the placement cost of all the vSDN controllers is equal. Without loss of generality, we assume $p_c = 1, c = 1, \ldots, C$. First, we present a toy example of the controller placement problem to visually demonstrate the performance of the proposed algorithm. In Fig. 2, a sample network composed of $C = 10$ nodes and 11 connections is presented where the numbers on the edges are the distance between the two corresponding nodes. The red circles are the candidates pool and we aim to select two vSDN controllers ($K = 2$) where the maximum number of controller failures is assumed to be $Q = 1$. The green encircled nodes are the controller placement solution obtained by an exhaustive search, while the

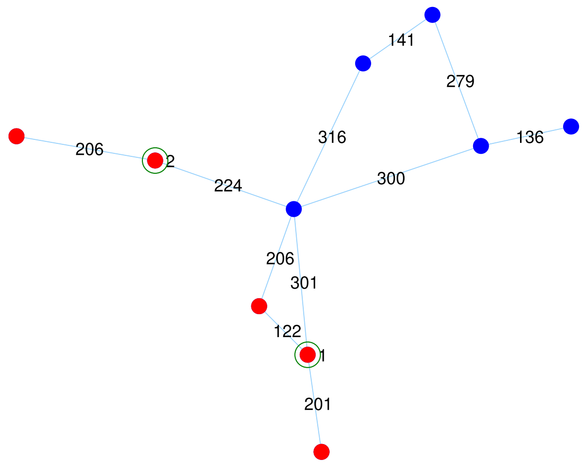

Fig. 2. A toy example of controller placement problem. The graph is composed of 10 vertices and 11 edges which represent NFV nodes ($\mathcal{N}$) and the existing connections among them, respectively. The red circles denote the candidates pool ($\mathcal{C}$) and the numbers on the edges are the distance among two nodes. In this example, we aim to place two controllers. The green encircled nodes are the controller placement solution obtained by an exhaustive search (i.e., the optimal solution), while the numbers 1 and 2 beside two of the nodes present the first and second locations selected by the proposed algorithm, respectively.

numbers 1 and 2 beside two of the nodes present the first and second controllers selected by the proposed algorithm, respectively. As shown in Fig. 2, for this toy example, the proposed algorithm has selected the optimal set of controllers.

Fig. 3 depicts the controllers-switches closeness f versus number of placed vSDN controllers K for two networks with $C = 15$ and $C = 30$. As expected, by increasing K, f increases. In Fig. 3(a), $C = 15$ is considered and a comparison between the result of the proposed greedy method and the exhaustive search is depicted. It is observable that the greedy algorithm has obtained the optimal solution for each value of K. In Fig. 3(b), another network with $C = 30$ is considered. Since exhaustive search is computationally expensive for this case, only the result of the greedy algorithm is plotted. From Fig. 3, it turns out that, increasing K in the beginning, when K is a small number, e.g., $0 \leqslant K \leqslant 5$, results in a huge increase in closeness, however, the rate of improvement reduces for large K. This phenomenon is known as diminishing return [58].

B. Non-Uniform Cost

In this section, we evaluate the proposed algorithms for the case of non-uniform cost where placement of each vSDN controller implies a different cost. Without loss of generality, we assume that the placement cost of each vSDN controller is a random variable with a uniform distribution between 0 and 1. Fig. 4 depicts the controllers-switches closeness f versus budget B. In Fig. 4(a), for a network with $C = 15$, the result of the exhaustive search is compared with the results of the uniform cost greedy algorithm for placement with non-uniform cost (i.e., **Algorithm 2**), the gain-cost greedy algorithm (i.e., **Algorithm 2** where the selection metric is substituted by (6)), the

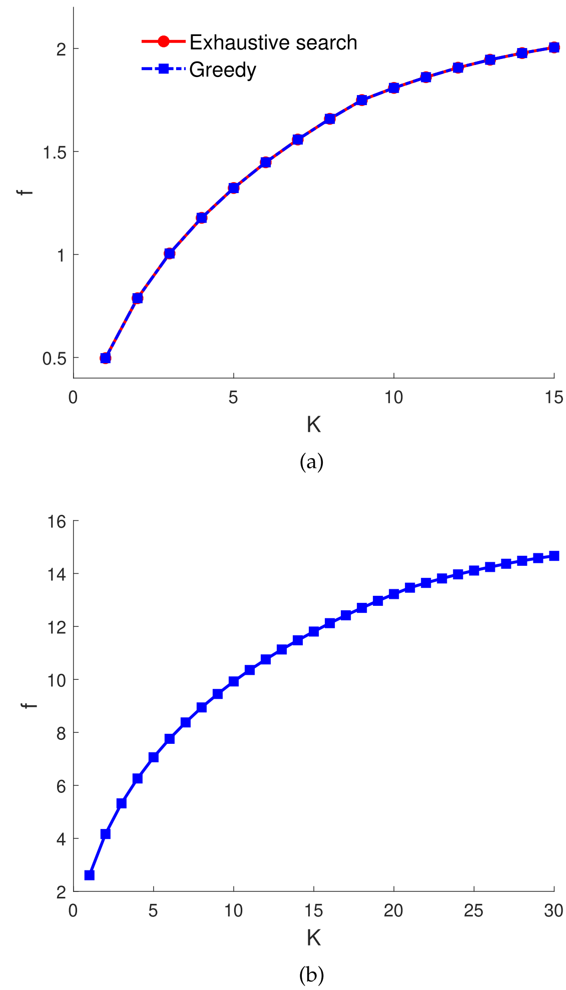

Fig. 3. Controllers-switches closeness f versus number of placed vSDN controllers K for a network with (a) $C = 15$, (b) $C = 30$.

maximum of these two (i.e., max greedy), and the 3-combinations gain-cost greedy algorithm (**Algorithm 3**). Although all the proposed algorithms are performing close to the optimum solution, the 3-combinations greedy algorithm outperforms the other algorithms, however at the cost of more computational complexity. Fig. 4(b) depicts a similar graph with $C = 30$ where the exhaustive search is not plotted due to the high computational complexity. As expected, increasing the budget leads to a higher closeness. It is evident that closeness exhibits a saturation behavior for B higher than 15.

C. Robustness-Closeness Trade-Off

In this section, we present a fundamental trade-off between robustness and closeness, where we demonstrate that considering backup vSDN controllers, i.e., the 2nd up to the $Q + 1$ closest vSDN controllers, reduce the closeness of the closest vSDN controller. In other words, if no failures occur in the network, a non-robust placement algorithm (i.e., $Q = 0$) outperforms a robust one. In the following, we consider $C = 30$ and perform 100 runs for the Monte Carlo simulations. To achieve the trade-off, we consider two simulations. For the

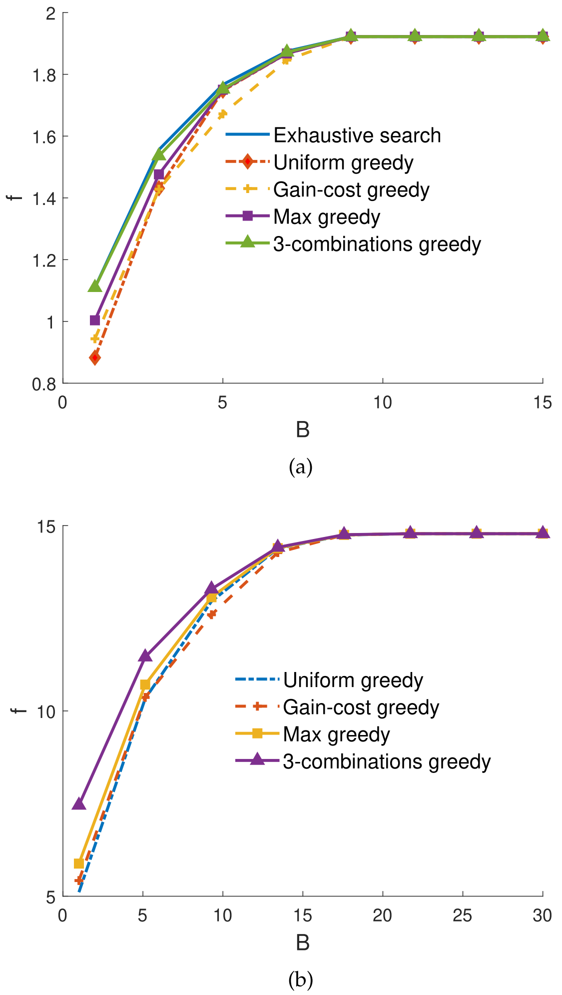

Fig. 4. Controllers-switches closeness f versus budget B for a (a) small scenario with $C = 15$, (b) large scenario with $C = 30$.

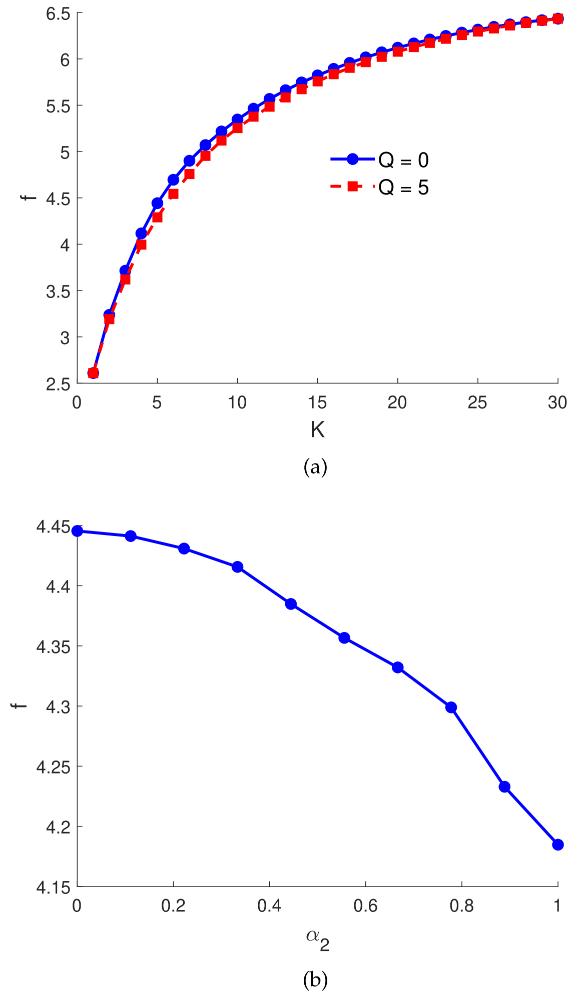

Fig. 5. Controllers-switches closeness of the closest switch (i.e., f_1) versus: (a) K and comparing the result for cases with $Q = 0$ and $Q = 5$, (b) α_2.

first scenario, we perform the proposed greedy algorithm for two cases with $Q = 0$ and $Q = 5$. However, in both cases, we only plot closeness of the closest vSDN controller f_1 calculated based on the set of selected vSDN controllers. Although for the case with $Q = 5$, the 6 closest vSDN controllers affect the objective function, and subsequently the set of selected vSDN controllers, we only plot the obtained f_1. Fig. 5(a) depicts the closeness versus K. As expected, we obtain a higher closeness for the $Q = 0$, since the objective function of both optimization problem and plot are the same, while for $Q = 5$, the objective function of the optimization problem is slightly different from what is plotted. This plot illustrates that making the selection robust (e.g., $Q = 5$) is obtained at the price of a reduction in optimality (i.e., lower closeness). For the second scenario, we perform the selection by setting $Q = 1$, $\alpha_1 = 1$, and sweeping α_2 in the range of [0,1]. In Fig. 5(b), similar to Fig. 5(a), only the closest vSDN controller (f_1) is considered for the plot. As expected, by increasing α_2, a larger weight is assigned to the second closest vSDN controller and therefore reduces f_1, illustrating the robustness-closeness trade-off from another perspective.

The results of simulations can be summarized as follows:

- Increasing the number of selected vSDN controllers increases the closeness function, which illustrates the trade-off between cost and latency.
- The closeness function follows the law of diminishing return meaning that a significant reduction in the number of vSDN controllers is achievable, at the cost of a negligible loss in closeness.
- For small scenarios, the proposed algorithms approach the optimal solution.
- There exists a fundamental trade-off between robustness and closeness.

VI. Conclusion

In this paper, we study the vSDN controller placement problem in 5G, which faces two conflicting objectives: 1) introducing more vSDN controllers to attain a more robust network with lower latency, 2) holding the deployment costs of vSDN controller in the reasonable limit. In this paper, we introduce a proper model for closeness and robustness for vSDN controllers and vSDN switches, and formulate the related problem. Proving the submodularity of the objective function of this paper called closeness function, we proposed near-optimal algorithms with low computational complexity to solve the placement problem for two scenarios: uniform and non-uniform cost of vSDN controller placement. We illustrated cost-latency and robustness-closeness trade-offs through the simulation results. Notably, simulation results demonstrate that at the price of negligible robustness loss, a significant reduction in network implementation cost for vSDN controllers is achievable.

VII. APPENDIX

VIII. PROOF OF THEOREM 3

Proof: Assume the vSDN controllers set $\mathcal{A} \subset \mathcal{C}$ and vSDN controllers $a, b \in \mathcal{C} \setminus \mathcal{A}$. Let us introduce the selection vectors $\mathbf{w}$, $\mathbf{w}^a$, $\mathbf{w}^b$, and $\mathbf{w}^{ab}$ corresponding to the active vSDN controllers set $\mathcal{A}$, $\mathcal{A} \cup \{a\}$, $\mathcal{A} \cup \{b\}$, and $\mathcal{A} \cup \{a, b\}$, respectively. Therefore, the submodularity condition is given by

$$f(\mathbf{w}^a) - f(\mathbf{w}) \geq f(\mathbf{w}^{ab}) - f(\mathbf{w}^b), \tag{7}$$

where $f(.)$ is the objective function in (2). By substituting the summation with $f(.)$, (7) recasts to

$$\sum_{s\in\mathcal{S}} \sum_{q=1}^{Q+1} \alpha_q \big(d_q(s|\mathbf{w}^a) - d_q(s|\mathbf{w})\big) \geqslant \sum_{s\in\mathcal{S}} \sum_{q=1}^{Q+1} \alpha_q \big(d_q(s|\mathbf{w}^{ab}) - d_q(s|\mathbf{w}^b)\big). \tag{8}$$

We prove (8) separately for each $s \in \mathcal{S}$, which subsequently provides the proof for the summation. Consider a specific arbitrary switch $s \in \mathcal{S}$. With some rearrangements, (8) reduces to

$$\sum_{q=1}^{Q+1} \alpha_q \big(d_q(s|\mathbf{w}^a) + d_q(s|\mathbf{w}^b)\big) \geqslant \sum_{q=1}^{Q+1} \alpha_q \big(d_q(s|\mathbf{w}^{ab}) + d_q(s|\mathbf{w})\big). \tag{9}$$

To make the proof more readable, we use the following notations. We define $c_q, q = 1, \ldots, Q+1$, as the qth closest vSDN controller in the active vSDN controllers set $\mathcal{A}$ for the switch s. In addition, we denote the closeness metric (i.e., defined in (1)) between s and an arbitrary vSDN controller c by d_c. Therefore, $d_q(s|\mathbf{w}), q = 1, \ldots, Q+1$ is represented by d_{c_q}, and consequently we have $d_{c_1} \geqslant \cdots \geqslant d_{c_{Q+1}}$. Note that if $|\mathcal{A}| < q$, then c_q does not exist and $d_{c_q} = 0$, which is consistent with the model. Since (9) is symmetric with respect to a and b, without loss of generality, we assume that a is closer to the switch s in comparison with the vSDN controller b. Moreover, in terms of closeness to switch s, we consider positions q_1 and q_2 for vSDN controllers a and b, respectively. In other words we assume

$$d_{c_1} \geqslant \cdots \geqslant d_{c_{q_1-1}} \geqslant d_a \geqslant d_{c_{q_1}} \geqslant \cdots \geqslant d_{c_{q_2-1}} \geqslant d_b \geqslant d_{c_{q_2}} \geqslant \cdots \geqslant d_{c_{Q+1}}. \tag{10}$$

Employing (10) in (9) leads to the following inequality

$$\sum_{q=1}^{q_1-1} \alpha_q d_{c_q} + \alpha_{q_1} d_a + \sum_{q=q_1}^{Q} \alpha_{q+1} d_{c_q} + \sum_{q=1}^{q_2-1} \alpha_q d_{c_q} + \alpha_{q_2} d_b + \sum_{q=q_2}^{Q} \alpha_{q+1} d_{c_q} \geq \sum_{q=1}^{q_1-1} \alpha_q d_{c_q} + \alpha_{q_1} d_a + \sum_{q=q_1}^{q_2-1} \alpha_{q+1} d_{c_q} + \alpha_{q_2+1} d_b + \sum_{q=q_2}^{Q-1} \alpha_{q+2} d_{c_q} + \sum_{q=1}^{Q+1} \alpha_q d_{c_q}, \tag{11}$$

which simplifies to

$$\alpha_{q_2} d_b + 2 \sum_{q=q_2}^{Q} \alpha_{q+1} d_{c_q} \geq \alpha_{q_2+1} d_b + \sum_{q=q_2}^{Q-1} \alpha_{q+2} d_{c_q} + \sum_{q=q_2}^{Q+1} \alpha_q d_{c_q}. \tag{12}$$

Regrouping the terms in both sides of (12), we have

$$\Big(\alpha_{q_2} d_b + \alpha_{q_2+1} d_{c_{q_2}}\Big) + \Big(\alpha_{q_2+1} d_{c_{q_2}} + \alpha_{q_2+2} d_{c_{q_2+1}}\Big) + \cdots + \Big(\alpha_Q d_{c_{Q-1}} + \alpha_{Q+1} d_{c_Q}\Big) + \Big(\alpha_{Q+1} d_{c_Q}\Big) \geqslant \Big(\alpha_{q_2} d_{c_{q_2}} + \alpha_{q_2+1} d_b\Big) + \Big(\alpha_{q_2+1} d_{c_{q_2+1}} + \alpha_{q_2+2} d_{c_{q_2}}\Big) + \cdots + \Big(\alpha_Q d_{c_Q} + \alpha_{Q+1} d_{c_{Q-1}}\Big) + \Big(\alpha_{Q+1} d_{c_{Q+1}}\Big), \tag{13}$$

where using Lemma 1, each group of terms of the left side of (13) is greater than or equal to the corresponding terms of the right side, and thus the proof is complete. ■

Lemma 1: If $a \geqslant b \geqslant 0$ and $c \geqslant d \geqslant 0$, then $ac + bd \geqslant ad + bc$.

Proof:

$$ac + bd \geqslant ad + bc \rightarrow a(c-d) \geqslant b(c-d), \tag{14}$$

which is correct based on the assumptions. ■

REFERENCES

[1] F. Z. Yousaf, M. Bredel, S. Schaller, and F. Schneider, "NFV and SDN-key technology enablers for 5G networks," *IEEE J. Sel. Areas Commun.*, vol. 35, no. 11, pp. 2468–2478, Nov. 2017.
[2] J. G. Herrera and J. F. Botero, "Resource allocation in NFV: A comprehensive survey," *IEEE Trans. Netw. Service Manag.*, vol. 13, no. 3, pp. 518–532, Sep. 2016.
[3] W. Xia, Y. Wen, C. H. Foh, D. Niyato, and H. Xie, "A survey on software-defined networking," *IEEE Commun. Surv. Tut.*, vol. 17, no. 1, pp. 27–51, Jan.–Mar. 2015.
[4] M. Derakhshani, S. Parsaeefard, T. Le-Ngoc, and A. Leon-Garcia, "Leveraging synergy of SDWN and multi-layer resource management for 5G networks," *IET Netw.*, vol. 7, no. 5, pp. 336–345, Aug. 2018.
[5] I. Afolabi, T. Taleb, K. Samdanis, A. Ksentini, and H. Flinck, "Network slicing and softwarization: A survey on principles, enabling technologies, and solutions," *IEEE Commun. Surv. Tut.*, vol. 20, no. 3, pp. 2429–2453, Jul.–Sep.2018.
[6] M. Otokura, K. Leibnitz, Y. Koizumi, D. Kominami, T. Shimokawa, and M. Murata, "Evolvable virtual network function placement method: Mechanism and performance evaluation," *IEEE Trans. Netw. Service Manag.*, vol. 16, no. 1, pp. 27–40, Mar. 2019.
[7] J. G. Herrera and J. F. Botero, "Resource allocation in NFV: A comprehensive survey," *IEEE Trans. Netw. Service Manag.*, vol. 13, no. 3, pp. 518–532, Sep. 2016.
[8] Z. Zaidi, V. Friderikos, Z. Yousaf, S. Fletcher, M. Dohler, and H. Aghvami, "Will SDN be part of 5G?," *IEEE Commun. Surv. Tut.*, vol. 20, no. 4, pp. 3220–3258, Oct.–Dec. 2018.

[9] Y. Chang, A. Rezaei, B. Vamanan, J. Hasan, S. Rao, and T. Vijaykumar, "Exploring functional slicing in the design of distributed SDN controllers," in *Proc. Int. Conf. Commun. Syst. Netw*. Berlin, Germany: Springer, 2017, pp. 177–199.

[10] A. K. Singh and S. Srivastava, "A survey and classification of controller placement problem in SDN," *Int. J. Netw. Manage.*, vol. 28, no. 3, 2018, Art no. e 2018.

[11] G. Wang, Y. Zhao, J. Huang, and W. Wang, "The controller placement problem in software defined networking: A survey," *IEEE Netw.*, vol. 31, no. 5, pp. 21–27, 2017.

[12] B. Isong, R. R. S. Molose, A. M. Abu-Mahfouz, and N. Dladlu, "Comprehensive review of SDN controller placement strategies," *IEEE Access*, vol. 8, pp. 170 070–170 092, 2020.

[13] D. Simeonidou, "Joint fronthaul optimization and SDN controller placement in dynamic 5G networks," in *Proc. Opt. Netw. Des. Model.: 23rd IFIP WG 6. 10 Int. Conf., ONDM 2019*, Athens, Greece, May 13-16, 2019, vol. 11616. Springer Nature, 2020, pp. 181–192.

[14] B. Heller, R. Sherwood, and N. McKeown, "The controller placement problem," in *Proc. 1st Workshop Hot Top. Softw. Defined Netw.*, 2012, pp. 7–12.

[15] A. Sallahi and M. St-Hilaire, "Optimal model for the controller placement problem in software defined networks," *IEEE Commun. Lett.*, vol. 19, no. 1, pp. 30–33, Jan. 2015.

[16] J. Liu, Z. Jiang, N. Kato, O. Akashi, and A. Takahara, "Reliability evaluation for NFV deployment of future mobile broadband networks," *IEEE Wireless Commun.*, vol. 23, no. 3, pp. 90–96, Jun. 2016.

[17] A. Dvir, Y. Haddad, and A. Zilberman, "Wireless controller placement problem," in *Proc. 15th IEEE Annu. Consum. Commun. Netw. Conf.*, Jan. 2018, pp. 1–4.

[18] B. P. R. Killi and S. V. Rao, "On placement of hypervisors and controllers in virtualized software defined network," *IEEE Trans. Netw. Service Manag.*, vol. 15, no. 2, pp. 840–853, Jun. 2018.

[19] G. Wang, Y. Zhao, J. Huang, and Y. Wu, "An effective approach to controller placement in software defined wide area networks," *IEEE Trans. Netw. Service Manag.*, vol. 15, no. 1, pp. 344–355, Mar. 2018.

[20] A. K. Singh, S. Maurya, N. Kumar, and S. Srivastava, "Heuristic approaches for the reliable sdn controller placement problem," *Trans. Emerg. Telecommun. Technol.*, vol. 31, no. 2, 2020, Art no. e 3761.

[21] Y. Li, S. Guan, C. Zhang, and W. Sun, "Parameter optimization model of heuristic algorithms for controller placement problem in large-scale SDN," *IEEE Access*, vol. 8, pp. 151 668–151 680, 2020.

[22] M. Tanha, D. Sajjadi, R. Ruby, and J. Pan, "Capacity-aware and delay-guaranteed resilient controller placement for software-defined WANs," *IEEE Trans. Netw. Service Manag.*, vol. 15, no. 3, pp. 991–1005, Sep. 2018.

[23] G. Nencioni, B. E. Helvik, A. J. Gonzalez, P. E. Heegaard, and A. Kamisinski, "Impact of SDN controllers deployment on network availability," 2017, *arXiv:1703.05595*.

[24] 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.*, Jul. 2017, pp. 1–9.

[25] B. P. R. Killi and S. V. Rao, "Controller placement with planning for failures in software defined networks," in *Proc. IEEE Int. Conf. Adv. Netw. Telecommun. Syst.*, Nov. 2016, pp. 1–6.

[26] B. P. R. Killi and S. V. Rao, "Optimal model for failure foresight capacitated controller placement in software-defined networks," *IEEE Commun. Lett.*, vol. 20, no. 6, pp. 1108–1111, Jun. 2016.

[27] N. Perrot and T. Reynaud, "Optimal placement of controllers in a resilient SDN architecture," in *Proc. 12th Int. Conf. Des. Reliable Commun. Netw.*, Mar. 2016, pp. 145–151.

[28] S. Yang, L. Cui, Z. Chen, and W. Xiao, "An efficient approach to robust SDN controller placement for security," *IEEE Trans. Netw. Service Manag.*, vol. 17, no. 3, pp. 1669–1682, Sep. 2020.

[29] 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*, May 2013, pp. 672–675.

[30] B. P. R. Killi and S. V. Rao, "Capacitated next controller placement in software defined networks," *IEEE Trans. Netw. Service Manag.*, vol. 14, no. 3, pp. 514–527, Sep. 2017.

[31] Y. Chen, Y. Yang, X. Zou, Q. Li, and Y. Jiang, "Adaptive distributed software defined networking," *Comput. Commun.*, vol. 102, pp. 120–129, Apr. 2017.

[32] 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. Funct. Virtual. Softw. Defined Netw.*, Nov. 2017, pp. 1–6.

[33] M. T. I. ul Huque, W. Si, G. Jourjon, and V. Gramoli, "Large-scale dynamic controller placement," *IEEE Trans. Netw. Service Manag.*, vol. 14, no. 1, pp. 63–76, Mar. 2017.

[34] M. Cello, Y. Xu, A. Walid, G. Wilfong, H. J. Chao, and M. Marchese, "BalCon: A distributed elastic SDN control via efficient switch migration," in *Proc. IEEE Int. Conf. Cloud Eng.*, Apr. 2017, pp. 40–50.

[35] Y. E. Oktian, S. Lee, H. Lee, and J. Lam, "Distributed SDN controller system: A survey on design choice," *Comput. Netw.*, vol. 121, pp. 100–111, 2017.

[36] A. Sallahi and M. St-Hilaire, "Expansion model for the controller placement problem in software defined networks," *IEEE Commun. Lett.*, vol. 21, no. 2, pp. 274–277, Feb. 2017.

[37] M. He, A. Basta, A. Blenk, and W. Kellerer, "Modeling flow setup time for controller placement in SDN: Evaluation for dynamic flows," in *Proc. IEEE Int. Conf. Commun.*, May 2017, pp. 1–7.

[38] M. J. Abdel-Rahman, E. A. Mazied, A. MacKenzie, S. Midkiff, M. R. Rizk, and M. El-Nainay, "On stochastic controller placement in software-defined wireless networks," in *Proc. IEEE Wireless Commun. Netw. Conf.*, Mar. 2017, pp. 1–6.

[39] Q. Zhong, Y. Wang, W. Li, and X. Qiu, "A min-cover based controller placement approach to build reliable control network in SDN," in *Proc. NOMS IEEE/IFIP Netw. Oper. Manage. Symp.*, Apr. 2016, pp. 481–487.

[40] A. Ksentini, M. Bagaa, T. Taleb, and I. Balasingham, "On using bargaining game for optimal placement of SDN controllers," in *Proc. IEEE Int. Conf. Commun.*, May 2016, pp. 1–6.

[41] A. Ksentini, M. Bagaa, and T. Taleb, "On using SDN in 5G: The controller placement problem," in *Proc. IEEE Glob. Commun. Conf.*, Dec. 2016, pp. 1–6.

[42] Y. Hu, T. Luo, N. C. Beaulieu, and C. Deng, "The energy-aware controller placement problem in software defined networks," *IEEE Commun. Lett.*, vol. 21, no. 4, pp. 741–744, Apr. 2017.

[43] A. Ruiz-Rivera, K. W. Chin, and S. Soh, "GreCo: An energy aware controller association algorithm for software defined networks," *IEEE Commun. Lett.*, vol. 19, no. 4, pp. 541–544, Apr. 2015.

[44] S. Lange *et al.*, "Heuristic approaches to the controller placement problem in large scale SDN networks," *IEEE Trans. Netw. Serv. Manag.*, vol. 12, no. 1, pp. 4–17, Mar. 2015.

[45] A. Sallahi and M. St-Hilaire, "Optimal model for the controller placement problem in software defined networks," *IEEE Commun. Lett.*, vol. 19, no. 1, pp. 30–33, Jan. 2015.

[46] Y. Hu, W. Wang, X. Gong, X. Que, and S. Cheng, "On reliability-optimized controller placement for software-defined networks," *China Commun.*, vol. 11, no. 2, pp. 38–54, Feb. 2014.

[47] Y. Jiménez, C. Cervelló-Pastor, and A. J. García, "On the controller placement for designing a distributed SDN control layer," in *Proc. IFIP Netw. Conf.*, Jun. 2014, pp. 1–9.

[48] D. Hock, S. Gebert, M. Hartmann, T. Zinner, and P. Tran-Gia, "POCO-framework for pareto-optimal resilient controller placement in SDN-based core networks," in *Proc. IEEE Netw. Oper. Manage. Symp.*, May 2014, pp. 1–2.

[49] L. F. Müller, R. R. Oliveira, M. C. Luizelli, L. P. Gaspary, and M. P. Barcellos, "Survivor: An enhanced controller placement strategy for improving sdn survivability," in *Proc. IEEE Glob. Commun. Conf.*, Dec. 2014, pp. 1909–1915.

[50] M. Guo and P. Bhattacharya, "Controller placement for improving resilience of software-defined networks," in *Proc. 4th Int. Conf. Netw. Distrib. Comput.*, Dec. 2013, pp. 23–27.

[51] M. F. Bari *et al.*, "Dynamic controller provisioning in software defined networks," in *Proc. 9th Int. Conf. Netw. Serv. Manage.*, Oct. 2013, pp. 18–25.

[52] 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.*, Sep. 2013, pp. 1–9.

[53] E. Tohidi, S. Parsaeefard, M. A. Maddah-Ali, B. H. Khalaj, and A. Leon-Garcia, "Controller-switch assignment in 5G networks," in *Proc. IEEE 2nd 5G World Forum*, Sep. 2019, pp. 40–45.

[54] G. Ramya and R. Manoharan, "Enhanced optimal placements of multi-controllers in SDN," *J. Ambient Intell. Human. Comput.*, pp. 1–18, 2020.

[55] E. Tohidi, R. Amiri, M. Coutino, D. Gesbert, G. Leus, and A. Karbasi, "Submodularity in action: From machine learning to signal processing applications," *IEEE Signal Process. Mag.*, vol. 37, no. 5, pp. 120–133, Sep. 2020.

[56] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to Algorithms*. Cambridge, MA, USA: MIT Press, 2009.

[57] A. Schrijver, *Theory of Linear and Integer Programming*. Hoboken, NJ, USA: Wiley, 1998.

[58] A. Krause and D. Golovin, "Submodular function maximization," in *Tractability: Practical Approaches to Hard Problems*. Cambridge, U.K.: Cambridge Univ. Press, pp. 69–104, 2014.

[59] B. Bollobás, *Random Graphs*. Cambridge U.K.: Cambridge Univ. Press 2001.