Computers and Electrical Engineering 000 (2017) 1–25

Contents lists available at ScienceDirect

Computers and Electrical Engineering

journal homepage: www.elsevier.com/locate/compeleceng

An adaptive heuristic for multi-objective controller placement in software-defined networks☆

Vahid Ahmadi[a,*], Mostafa Khorramizadeh[a]

[a] Shiraz University of Technology, Department of Mathematics, Shiraz, Iran

ARTICLE INFO

Article history:
Received 2 April 2017
Revised 22 December 2017
Accepted 22 December 2017
Available online xxx

Keywords:
Software-defined network
Controller placement
Multi-objective combinatorial optimization
Heuristic algorithms
Pareto front
Multi-start hybrid non-dominated sorting genetic algorithm

ABSTRACT

Software-defined networking paradigm faces many challenges, including reliability, resiliency, scalability, and availability. These challenges can be tackled by carefully selecting placements within the network. However, the evaluation of all placements is only practical for small networks. In this paper, a fast and efficient adaptation of evolutionary algorithms is presented to solve large-scale multi-objective controller placement problems. The presented algorithm requires reasonable memory resource and enjoys a greedy heuristic to generate a high-quality initial population, smart mechanisms to encourage the diversification and intensification, and a new fast Pareto finder. Moreover, a new variant of the problem is developed in which the capacities of controllers and loads of switches are added as constraints. A new constraint handling technique is applied to adapt our algorithm to solve the new problem. Finally, the results on several topologies from Internet Topology Zoo revealed that our presented algorithms outperformed some other efficient algorithms from the literature.

1. Introduction

The main feature of Software-Defined Network (SDN) architectures is the separation of network controllers and switches. The centralized controller is in charge of managing switches in networks. This is done by generating some rules that dictate their packet forwarding behavior. Indeed, in these networks, controllers are responsible for creating forwarding rules. Although the controller is logically centralized in the network domain, it should be physically distributed to consider some aspects like performance, scalability, resilience, and fault tolerance restrictions [1]. To address issues such as scalability and resilience, concepts like HyperFlow permit the division of SDN architecture into multiple domains such that each one is managed by an individual controller. To address issues such as scalability and resilience, concepts like HyperFlow divide SDN architecture into multiple domains then each domain is managed by an individual controller. A logically centralized controller which operates the control plane is the pivot of a traditional SDN implementation. However, in the deployment of a Wide Area Network (WAN), this rudimentary centralized method suffers from several limits which are relevant to performance and scalability. In order to cope with the mentioned issues, one way is to implement multiple controllers which cooperate to manage a network.

A very fundamental task in SDN architectures is locating a limited number of controllers within a network so that several requirements are fulfilled [2]. This problem is called the controller placement and is known to be NP-hard [3].

☆ Reviews processed and recommended for publication to the Editor-in-Chief by Associate Editor Dr. Z. Arnavut

* Corresponding author
E-mail addresses: v.ahmadi@sutech.ac.ir, vahid@cs.aau.dk (V. Ahmadi), m.khorrami@sutech.ac.ir (M. Khorramizadeh).

https://doi.org/10.1016/j.compeleceng.2017.12.043
0045-7906/

The earlier studies only dealt with the propagation latency between controllers and switches [1,3]. However, they disregarded either the inter-controller latency, i.e. the latency between each pair of controllers, or the capacities of controllers. Both of these factors are crucial in the context of real networks.

The propagation latency between switches and their associated controllers is a key factor for the performance of SDNs. Another type of latency called the latency between controllers should also be considered to maintain a synchronized state. When multiple controllers are deployed, the latency between individual controllers plays an important role. This importance is due to the fact that to achieve the global consistency in a network, the interactions among controllers are crucial. If there is more than a single controller in a network, the synchronization is necessary to maintain a consistent global state. Depending on the frequency of inter-controller synchronization, the latency between individual controllers is vital, hence should be taken into account during the controller placement [4]. In addition, the load of controllers is a critical factor in real networks. When the number of nodes which are controlled by a controller increases, the load on the controller also rises. Moreover, when node-to-controller requests increase, because of processing on the controller system, the probability of extra delays also increases. Therefore, it is inevitable that the node-to-controller assignment is well-balanced in scenarios where nodes communicate with their controllers.

The objectives that could be taken into account include the latency between each switch and its assigned controller, inter-controller latency, load balancing, and failure in nodes, links, and controllers. Therefore, controller placement problem can be formulated as a multi-objective combinatorial optimization problem. Since the objectives are supposed to be pairwise conflicting, applying multi-objective methods allows for a more obvious display of the trade-offs between these objectives [5,6].

In the controller placement problem, the goal is to find the location of a given number of controllers within a network. However, in the presence of some constraints which must be satisfied, finding a feasible solution for the controller placement problem is very important. By simply choosing a random placement of an arbitrary number of controllers, an acceptable performance may not be achieved [3]. Therefore, to design an efficient and reliable SDN, appropriate answers should be found for the following questions: How many controllers are required? Where should they be located? Which nodes should be managed by which controller? Rather than applying estimations in this regard, Heller et al. [3] discussed that an exhaustive evaluation of the whole solution space, i.e. evaluating all probable placements of the controllers, is possible for real networks. Hence, finding optimal placements with respect to the corresponding metrics is guaranteed.

Therefore, obtaining a controller placement which represents a good trade-off between corresponding objectives is critical for an effective operation. Moreover, to tackle large networks and dynamically changing environments, i.e. network conditions, such as traffic patterns or bandwidth demands [7], an efficient approach is required. In other words, it should be computationally fast enough to solve the aforementioned placement problem.

When it comes to large-scale WANs, administrators encounter various challenges such as dynamically changing network conditions. These changes influence the overall performance of a network and should be handled immediately. The influence of these changes emerges on a regular and predictable basis as it occurs in the case of day and night cycles and at shorter time scales, e.g. hourly. To adjust to these changes and ensure a smooth operation, the placements which consider the current network state are required to be calculated. The switching rate to another placement depends highly on the rate of these calculations. As illustrated in [8], an exhaustive evaluation of placements can be performed within a few seconds using the Pareto-Optimal Controller Placement (*POCO*) framework for medium and small-sized instances. Nevertheless, this evaluation is not sufficient to handle the aforementioned dynamic changes in large-scale networks since it needs plenty of running time.

Lange et al. [2] investigated heuristic approaches to efficiently solve the multi-objective controller placement problem. Although it does not guarantee to obtain optimal solutions, such approaches are capable to achieve the results considerably faster than their exhaustive rival.

Regarding features such as scalability, resilience, control plane communication delays, and the constraints that are time-independent, some other aspects should be considered. For dynamically changing network conditions such as traffic patterns or bandwidth demands, a regular and quick recalculation of placements is quite essential to adjust to the current state in a timely manner. This is carried out mainly to deal with such dynamic environments. Despite the fact that performing an exhaustive evaluation of all possible placements may take a reasonable time for small and medium networks, it is impractical for large instances which require drastically more time and memory budget. Hence, alternative approaches need to be investigated for large instances.

This work presents a heuristic algorithm called Multi-Start Hybrid Non-dominated Sorting Genetic Algorithm (or *MHNSGA*) to solve the multi-objective controller placement problem effectively. In the context of multi-objective optimization, in most cases, a single solution that optimizes all the considered objectives may not exist. Hence, achieving the set of optimal solutions- which is called Pareto front in the objective space and Pareto set in the decision space- constitutes the main target of Pareto-based multi-objective optimization methods. However, this set may not be obtained completely by using heuristic methods on a problem instance. This is mainly because these methods explore a small part of the whole search space and have stochastic behavior. Therefore, a set of non-dominated solutions has been investigated as an approximation of the exact Pareto front [2]. The results of several evaluations showed that our proposed algorithm is capable to explore a large portion of the search space and obtain an estimation of the Pareto optimal front with a high degree of accuracy. In other words, it is efficient to generate an approximation set close to the Pareto front and well-distributed over it. In

Please cite this article as: V. Ahmadi, M. Khorramizadeh, An adaptive heuristic for multi-objective controller placement in software-defined networks, Computers and Electrical Engineering (2017), https://doi.org/10.1016/j.compeleceng.2017.12.043