

Optimization of Controller Placement and Information Flow in Softwarized Networks

Stanislav Lange

University of Würzburg, Institute of Computer Science, Chair of Communication Networks, Würzburg, Germany
stanislav.lange@informatik.uni-wuerzburg.de

Abstract—The Software Defined Networking (SDN) paradigm offers network operators numerous improvements in terms of flexibility, scalability, as well as cost efficiency and vendor independence. However, in order to maximize the benefit from these features, several new challenges in areas such as management and orchestration need to be addressed. The dissertation that is summarized in this paper makes contributions towards three key topics from these areas.

Firstly, we design, implement, and evaluate two multi-objective heuristics for the SDN controller placement problem. Secondly, we develop and apply mechanisms for automated decision making based on the Pareto frontiers that are returned by the multi-objective optimizers. Finally, we investigate and quantify the performance benefits for the SDN control plane that can be achieved by integrating information from external entities such as Network Management Systems (NMSs) into the control loop. Our evaluation results demonstrate the impact of optimizing various parameters of softwarized networks at different levels and are used to derive guidelines for an efficient operation.

Index Terms—SDN, Placement, Multi-Objective Optimization, Pareto Frontier, NMS.

I. INTRODUCTION

Nowadays, the software defined networking (SDN) paradigm receives an increasing amount of attention due to its flexible configuration, programmability, and cost efficiency. These are achieved by moving control plane functions from individual network devices to a dedicated, logically centralized controller software that runs on commodity hardware. Communication between this centralized control plane and the data plane is then performed via the southbound API [1] which is implemented by protocols like OpenFlow [2]. However, in order to fully reap these benefits, novel challenges regarding the management and orchestration of such networks need to be tackled. These challenges include the SDN controller placement problem which also entails research questions regarding automated decision making as well as the interaction between the SDN control plane and existing Network Management Systems (NMSs). In the dissertation, approaches for solving these three central problems are proposed and evaluated with respect to their impact on different aspects of network performance. An overview of the contributions is given in Figure 1.

SDN Controller Placement. A particularly important task in SDN architectures is that of controller placement [3], i.e., identifying controller locations that simultaneously optimize multiple objectives such as the number of controller instances, network delays, and the load distribution among instances. Due

to its complexity, we approach this problem with heuristics that trade-off accuracy against run time [4], [5]. The left part of Figure 1 illustrates this step. Given an input topology, multi-objective optimization algorithms calculate a Pareto frontier, i.e., a set of possible solutions that represent different trade-offs between the competing objectives and are incomparable among each other. In the second graphic, each blue square corresponds to one particular Pareto optimal placement.

Automated Decision Making. The multi-objective nature of the placement problem results in sets of Pareto optimal solutions rather than distinct optima. Therefore, scenarios with dynamically changing network conditions require mechanisms for automated decision making based on such Pareto frontiers in order to function without manual interaction. Hence, we investigate techniques from the domain of multi-attribute decision making that aggregate the performance of placements into a single numeric score and compare the resulting rankings. Evaluations featuring over 50 real world topologies from the Internet Topology Zoo [6] demonstrate the feasibility of the proposed mechanisms and their agreement regarding top-ranked solutions [7], [8].

Interaction between SDN and NMS. Finally, an integration of SDN components into existing ecosystems is required for a smooth transition from legacy to SDN-based networks. One of the key aspects for this integration consists of interactions between the SDN controller and other centralized entities such as Network Management Systems (NMSs). In particular, we focus on improved SDN control plane decisions based on monitoring data that is regularly provided by an NMS. To this end, we design, implement, and compare two versions of the ONOS controller [9]. Alongside the default implementation, these represent different trade-offs regarding the complexity of the resulting system and its performance. In addition to evaluations that show a significant performance improvement when using the optimized controllers, a parameter study investigates the performance impact of network characteristics [10].

The remainder of this work is structured as follows. Sections II to IV cover the SDN controller placement problem, automated decision making mechanisms, and the improved NMS-aware SDN controller, respectively. In each of these sections, a brief discussion of related work, our proposed approach, as well as its performance evaluation methodology and results are provided. Finally, Section V concludes the paper with an overview of lessons learned.

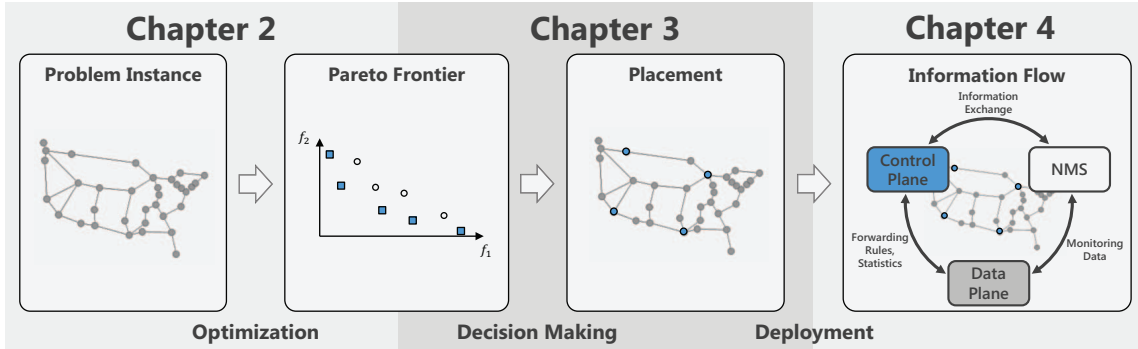


Figure 1: Overview of the topics that are covered in the dissertation.

II. MULTI-OBJECTIVE HEURISTICS FOR THE SDN CONTROLLER PLACEMENT PROBLEM

In order to enhance the scalability and resilience of an SDN infrastructure, the logically centralized SDN control plane is often deployed in a physically distributed fashion. For this, specialized approaches like HyperFlow [11] and Kandoo [12] have been developed. As a consequence, recent releases of modern SDN controllers such as ONOS [9] and OpenDaylight [13] also include support for distributed deployments. Additionally, the position of these distributed instances can have a significant performance impact. Therefore, multiple and possibly competing objectives regarding aspects like load balance between controller instances and communication delays need to be optimized simultaneously.

Furthermore, dynamically changing network conditions like traffic patterns or bandwidth demands need to be considered as well [14]. Consequently, such dynamic environments call for a regular and fast recalculation of placements in order to adapt to the current situation in a timely manner. An exhaustive evaluation of all possible solutions can be performed within a practically feasible time frame for small and medium-sized networks [15]. However, significantly higher time and memory requirements make such an approach out of scope for large problem instances, e.g., in the context of realistic WAN topologies. Therefore, we design and evaluate multi-objective heuristics for solving the SDN controller placement problem that feature a significantly lower computation time while maintaining an acceptable margin of error [4], [5].

A. Problem Statement and Optimization Objectives

Formally speaking, the controller placement problem is a multiobjective combinatorial optimization problem. The network is represented by an undirected graph $G = (V, E)$ with node set V containing n nodes which are connected by edges from the set E . In this context, nodes denote the locations of switches in the network as well as possible locations for controllers. Furthermore, latencies are represented by means of a distance matrix that provides values for each pair of nodes. Given the desired number of controllers, k , there is a finite set of $\binom{n}{k}$ possible placements, hence the term *combinatorial optimization*. Therefore, the size of the search space grows

rapidly with the size of the network and the number of controllers that are placed.

In our work, we optimize a total of five objectives that represent different performance aspects of an SDN-based network. In addition to the latency of the control channel between switches and their assigned controller, we also take into account the latency between each pair of controllers since they need to synchronize state information and minimize the window of inconsistency between each other. For both latency-based objectives, average as well as worst-case values are considered. Finally, the load distribution among controller instances is represented by the difference between the number of switches that are assigned to the controller with the highest and lowest amount of switches, respectively. This measure is important since controller benchmarks have demonstrated that the number of assigned switches can significantly impact controller performance [16].

Furthermore, even in the case of optimizing only the latency between switches and controllers, the placement problem corresponds to the well-known *facility location problem* [17], [18] which is NP-hard. Hence, we propose heuristic approaches for solving this multi-objective version of the problem.

B. Placement Heuristics

The controller placement problem is approached with two different heuristics that have been designed during the course of the dissertation.

Firstly, we adapt and employ the generic *Pareto Simulated Annealing (PSA)* [19] algorithm, a multi-objective method that is based on simulated annealing [20]. This family of optimization algorithms has been successfully used for single-objective placement optimization [14] and is characterized by following an *annealing schedule*. On the one hand, this allows the algorithms to efficiently explore the solution space and escape local optima by accepting worse solutions. On the other hand, convergence and improvement of found solutions is achieved by lowering this acceptance probability towards the end of the optimization. In contrast to the single-objective version, PSA returns a set of Pareto optimal placements that are incomparable with each other and represent trade-offs between the objectives.

While PSA is a generic procedure that can easily be extended with additional objectives, specialized heuristics may improve the optimization performance by leveraging characteristics of certain objectives and results from fields like graph theory. Hence, we additionally design *Pareto Capacitated k-Medoids (PCKM)* that focuses on the two objectives of switch-to-controller latency and controller load distribution. This algorithm is based on k-medoids [21], a clustering algorithm that is similar to k-means [22]. However, in contrast to the latter, the cluster centers that are returned are nodes of the input topology and therefore directly correspond to the desired controller locations. By varying a constraint regarding the cluster size, PCKM is capable of exploring trade-offs between latency and controller load distribution, and finally returns the corresponding Pareto frontier.

C. Performance Evaluation and Comparison

We evaluate both optimization strategies in the context of more than 50 real world topologies from the Internet Topology Zoo [6] and compare their performance based on results from exhaustive evaluations that serve as ground truth. Our results show that the heuristics provide a significant speedup of a factor of up to 20 for large scale instances. For example, the proposed PSA heuristic can reduce the run time for optimizing the placement of 7 controllers in a network with 50 nodes from nearly half an hour to less than 30 seconds if an error of up to 2% is acceptable.

Additionally, we compare the performance of the two heuristics with each other. To this end, we calculate the distance between the Pareto frontiers that are returned by our algorithms and the Pareto frontiers that are obtained by means of exhaustively evaluating the corresponding problem instances. Figure 2 displays cumulative distributions of the distances that are achieved by the specialized PCKM approach, the PSA approach when optimizing only two objectives (PSA2D), and an algorithm that generates solutions by evaluating randomly chosen placements (RND). All three algorithms are given a run time limit of four seconds to ensure a fair comparison.

On the one hand, the specialized heuristic consistently outperforms the generic PSA-based approach, confirming that specialized heuristics can further improve the performance within their narrower scope of applicability. On the other hand, the significant gap between the two proposed heuristics and RND highlights their systematic and efficient exploration of the search space.

III. AUTOMATED DECISION MAKING BASED ON PARETO FRONTIERS

In the context of multi-objective optimization tasks like the placement of SDN controllers, operators are confronted with a multitude of solutions that are incomparable among each other. However, dynamic scenarios in particular require adaptation to changing environments and choosing *one* distinct solution that needs to be implemented in a timely manner. Furthermore, even Pareto frontiers of medium sized problem instances can

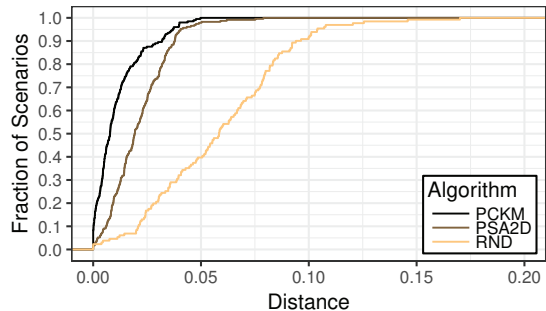


Figure 2: Performance comparison for two-objective optimization and a run time constraint of 4 seconds.

contain thousands of distinct solutions. Combined with solution spaces that have more than two dimensions, these issues make manual decision making infeasible in practice. Hence, the goal of the second part of the dissertation is to design, evaluate, and compare mechanisms that enable automated decision making between multi-dimensional solutions. To this end, we employ ideas from the domain of multi-attribute decision making and devise the following three-step approach.

Firstly, weights are determined for each objective. These weights represent the relative importance of the individual objective in the context of the particular problem instance. In contrast to defining these weights a priori, they are adapted to the specific problem instance at hand. Secondly, the weights are used to calculate a score for each individual solution, so that an order can be imposed on the elements of the Pareto frontier. Finally, we characterize the rankings that result from different combinations of weighting and aggregation techniques and investigate their agreement with each other. We propose four mechanisms for each of the two steps which results in a total of 16 combinations for determining a ranking of Pareto optima [7], [8].

A. Weighting and Ranking Methods

In the following, the four mechanisms for weighting and the four mechanisms for ranking individual solutions are outlined. In order to obtain comparable results, we normalize the recorded objective values beforehand and ensure that the sum of weights across all objectives equals 1.

As a baseline naïve approach, we use a weighting mechanism that does not take into account any observed data and assigns equal weights to every objective. In this case, the weight of the j -th out of a total of m objectives is calculated as $w_j^{uni} = \frac{1}{m}$. The remaining three techniques utilize the entropy [23], [24], coefficient of variation, and standard deviation of observed values in each dimension, respectively. The key idea behind using these weighting methods consists of assigning higher weights to objective dimensions that carry more information, i.e., those that have a higher number of distinct values, low individual occurrence probabilities, and cover a wide range of values.

Given the abovementioned weights, we use four methods that are proposed in the literature to aggregate a vector of attained objective values into a single score. The Simple Additive Weighting (SAW) [25] uses a simple weighted sum approach, whereas Multiplicative Exponential Weighting (MEW) [26] uses the weights as exponent. More sophisticated mechanisms include the Technique for Order Preference by Similarity to Ideal Solution (TOPSIS) [25] which first constructs an ideal solution based on the extreme values of observations. Then, solutions are ranked based on their distance to this ideal reference. Similarly, the VIKOR [27] approach uses a combination of the average distance to the ideal solution and a worst case perspective that focuses on the dimension that results in the largest distance from the reference.

B. Applicability to the SDN Controller Placement Problem

In order to evaluate the proposed approaches in terms of their applicability and agreement among each other, we employ them in the context of SDN controller placement in two ways. Firstly, we conduct a case study on a small problem instance featuring the Internet2 OS3E topology which has 34 nodes. In this topology, 4 controllers are placed with respect to three objectives, resulting in more than 46,000 distinct placements of which 10 are Pareto optimal. Secondly, we perform a broad analysis with more than 50 topologies from the Internet Topology Zoo and 5 objectives in order to analyze agreement and consistency across problem instances.

Figure 3 displays the results of the case study. Each of the 16 combinations of four weighting and four scoring techniques assigns a rank to each of the 10 Pareto optimal placements. These placements are arranged on the x-axis according to the median rank they achieve. Based on the approach in [28], the y-axis of the boxplot shows the distribution of ranks that are achieved by each placement. The median rank of a placement is highlighted by the bold horizontal line in each box. Additionally, outliers whose distance to the median exceeds 1.5 times the inter-quartile range are displayed individually.

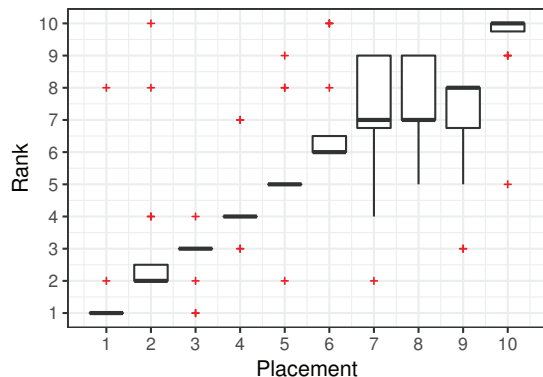


Figure 3: Pareto optimal placements and their ranks according to the 16 combinations of weighting and scoring mechanisms.

It can be observed that the boxes for the first five placements are very narrow, indicating a large agreement among the dif-

ferent weighting-ranking combinations. Only for the last five placements, there is some disagreement among the rankings. Despite this, several outliers can be observed for the entire range of placements. However, a detailed analysis of the data shows that most of these outlier ratings stem from the uniform weighting mechanism. Hence, this weighting method can be used to generate a complementary view of the placements.

As mentioned above, we extend the analysis to a wide range of problem instances featuring real world network topologies. We confirm our findings regarding the agreement of the mechanisms by means of several probabilistic models and agreement measures. These include the models proposed by Luce [29] and Mallows [30], [31] as well as Gordon's α [32]. In summary, the facts that decision making requires choosing exactly one solution and that there is a high agreement regarding top-ranked placements demonstrate the feasibility of the proposed approaches for this task.

IV. INTEGRATION OF NETWORK MANAGEMENT INFORMATION INTO THE SDN CONTROL PLANE

In addition to the challenges regarding the management and orchestration of SDN-based networks, operators need to take into account that modern networks are comprised of a multitude of heterogeneous devices which also include non-SDN legacy devices as well as network management systems (NMSs) that are used for monitoring and configuration. Although both, the SDN controller and the NMS, have a centralized view of the network, they operate at different time scales and deal with information at different levels of granularity [33], [34]. In the third part of the dissertation, we investigate the impact on the network performance when an NMS regularly provides information to the SDN controller.

To achieve this goal, we design, implement, and evaluate two enhanced versions of the popular ONOS controller [9]. Both optimize the load distribution among network links. However, one performs hash-based randomized load balancing without additional information whereas the NMS-aware controller leverages ONOS's intent and annotation frameworks to take into account external information. Hence, these controller versions represent different trade-offs regarding the complexity of the resulting architecture and the overall performance. In addition to evaluations that show a significant performance improvement over the default controller when using these optimized controllers, we present a parameter study that provides insights into the performance impact of network characteristics like the flow interarrival time, the flow duration, and the number of active flows [10].

In contrast to previous works that focus on SDN-based QoS control for different types of applications [35]–[37], we do not put the additional burden of monitoring on the controller and also maintain a separation of concerns among entities. Furthermore, we build upon the widely used ONOS controller rather than developing an entirely new one as proposed in [38]. In a similar fashion, we utilize and extend existing REST interfaces that can be used by any external information source rather than defining a novel framework-specific protocol [39].

A. Testbed Setup

To demonstrate the feasibility of the proposed NMS-aware controller, we perform a case study in a network topology that is emulated in *Mininet*¹. In this network, the links are bandwidth-limited via the Linux traffic control tool *tc* and traffic is generated with *iperf*². Furthermore, we use *tcpdump* at different points in the network to obtain ground truth data regarding the actual bandwidth that goes in and out of the network as well as the flow distribution among different paths. The individual components, i.e., the controller, the network, and the NMS are placed in different virtual machines (VMs) on a Linux PC. Figure 4 shows this setup. The controller is deployed on the first VM. This allows to conveniently switch between controller implementations and log their respective resource utilization. The emulated network and the traffic generating hosts are deployed on the second VM. In the context of the NMS-aware controller scenario, the NMS is also deployed on the second VM.

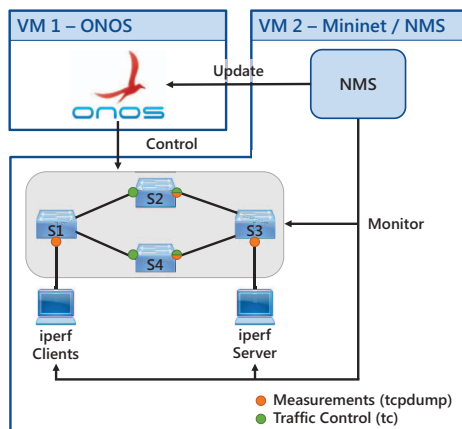


Figure 4: Testbed setup.

In our evaluation, the role of the NMS is fulfilled by a Python tool that adheres to the aforementioned REST interface and regularly provides bandwidth utilization of individual links as well bandwidth requirements of flows. Due to the generic REST interface, the Python tool can easily be replaced with any real world NMS. Finally, the NMS-aware controller leverages the provided information to allocate new flows to the paths with the least utilized links and therefore maximize the throughput and load distribution among links.

B. Performance Benefits of the NMS-aware Controller

In order to quantify and compare the performance of the controllers, we use several metrics that cover different performance aspects. These include the throughput as well as the fairness regarding link utilization and fulfillment of flows' bandwidth requirements. All metrics are normalized and a value of 1 corresponds to the maximum possible

¹<http://mininet.org/>

²<https://iperf.fr/>

throughput and a perfectly fair load distribution, respectively. Additionally, we measure the CPU load at the controller to quantify the cost for processing additional information.

Figure 5 displays results that are obtained from numerous combinations of traffic characteristics like the flow interarrival time, the flow duration, and the number of active flows. For each set of parameters, we perform 10 experiments that each last 10 minutes. While the x-axis represents the controller version, the bars' height denotes the mean value of each performance metric, and whiskers show 95% confidence intervals.

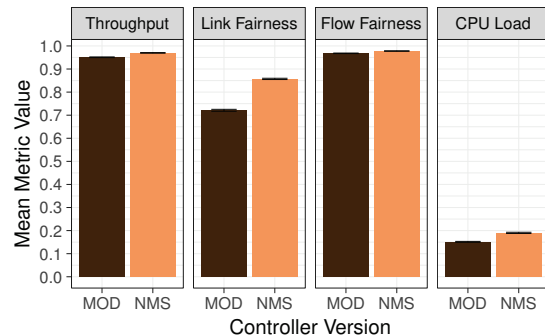


Figure 5: Comparison of performance indicators for modified and NMS-aware ONOS versions across all experiments.

For all performance measures, the NMS-aware controller outperforms the controller that has no access to external information. However, the performance benefits come at the price of an increased CPU load for processing this information.

Further evaluations are performed to investigate the impact of the information exchange frequency. These show that even receiving updates only every two minutes can significantly boost the performance. Finally, we publish the source code of the controller as well as the evaluation framework via [github](https://github.com)³ so that other researchers can reproduce the results or use them as foundation for further extensions.

V. CONCLUSION

In order to fully reap the benefits that are offered by network softwarezation, novel challenges in the domain of management and orchestration need to be tackled. In the dissertation that is outlined in this manuscript, we propose optimization-based approaches at different levels and stages of the deployment process. These include SDN controller placement, automated decision making in the face of Pareto frontiers, and the integration of external information into the SDN control loop.

By employing multi-objective heuristics, we can significantly speed up the optimization of SDN controller locations while maintaining a high level of accuracy. For example, we can improve the run time for placing 7 controllers in a network with 50 nodes from nearly half an hour to less than 30 seconds if an error of up to 2% is acceptable. Additionally, trade-offs between competing objectives can be identified and

³<https://github.com/lsinfo3/{nms-aware-onos, nms-onos-meas}>.

taken into account with the multi-objective approach. With specialized heuristics such as PCKM, the accuracy can be improved even further. Additionally, we propose mechanisms for automated decision making in order to address the need for fast adaptation to dynamically changing conditions. We demonstrate the applicability of these mechanisms in the context of the controller placement problem and find a high degree of agreement when identifying top-ranked placements.

Finally, we investigate the performance gains that can be achieved by including external information from sources like Network Management Systems (NMSs) in SDN controller decisions. In addition to designing and implementing such an NMS-aware controller, we analyze and compare its performance in numerous network conditions. On the one hand, our evaluations demonstrate that significant improvements in terms of throughput and fair load distribution can be achieved by means of this NMS-awareness. On the other hand, we highlight that these benefits come at the price of an increased CPU load that needs to be taken into account prior to deployment.

In summary, the mechanisms proposed in the thesis improve the performance of softwarized networks for various stakeholders, ranging from network operators to end users. Furthermore, evaluations in the context of realistic problem instances demonstrate their practical relevance and applicability.

REFERENCES

- [1] M. Jarschel, T. Zinner, T. Hoßfeld, P. Tran-Gia, and W. Kellerer, "Interfaces, Attributes, and Use Cases: A Compass for SDN," *IEEE Communications Magazine*, 2014.
- [2] N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, J. Rexford, S. Shenker, and J. Turner, "OpenFlow: Enabling Innovation in Campus Networks," *ACM SIGCOMM Computer Communication Review (CCR)*, 2008.
- [3] B. Heller, R. Sherwood, and N. McKeown, "The Controller Placement Problem," in *ACM SIGCOMM Workshop on Hot Topics in Software Defined Networking (HotSDN)*, 2012.
- [4] 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 Transactions on Network and Service Management - Special Issue on Efficient Management of SDN and NFV-based Systems*, 2015.
- [5] S. Lange, S. Gebert, J. Spoerhase, P. Rygielski, T. Zinner, S. Kounev, and P. Tran-Gia, "Specialized Heuristics for the Controller Placement Problem in Large Scale SDN Networks," in *International Teletraffic Congress (ITC 27)*, 2015.
- [6] S. Knight, H. X. Nguyen, N. Falkner, R. Bowden, and M. Roughan, "The Internet Topology Zoo," *IEEE Journal on Selected Areas in Communications (JSAC)*, 2011.
- [7] M. Seufert, S. Lange, and M. Meixner, "Automated Decision Making Methods for the Multi-objective Optimization Task of Cloud Service Placement," in *International Workshop on Programmability for Cloud Networks and Applications (PROCON)*, 2016.
- [8] —, "Automated Decision Making based on Pareto Frontiers in the Context of Service Placement in Networks," in *International Teletraffic Congress (ITC)*, 2017.
- [9] P. Berde, M. Gerola, J. Hart, Y. Higuchi, M. Kobayashi, T. Koide, B. Lantz, B. O'Connor, P. Radoslavov, W. Snow *et al.*, "ONOS: towards an open, distributed SDN OS," in *ACM SIGCOMM Workshop on Hot Topics in Software Defined Networking (HotSDN)*, 2014.
- [10] S. Lange, L. Reinhart, T. Zinner, D. Hock, N. Gray, and P. Tran-Gia, "Integrating network management information into the sdn control plane," in *IEEE/IFIP Network Operations and Management Symposium (NOMS)*, 2018.
- [11] A. Tootoonchian and Y. Ganjali, "HyperFlow: a Distributed Control Plane for OpenFlow," in *Internet Network Management Conference on Research on Enterprise Networking*, 2010.
- [12] S. Hassas Yeganeh and Y. Ganjali, "Kandoo: a framework for efficient and scalable offloading of control applications," in *ACM SIGCOMM Workshop on Hot Topics in Software Defined Networking (HotSDN)*, 2012.
- [13] J. Medved, R. Varga, A. Tkacik, and K. Gray, "OpenDaylight: Towards a model-driven sdn controller architecture," in *IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM)*, 2014.
- [14] 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 *International Conference on Network and Services Management (CNSM)*, 2013.
- [15] 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 *International Teletraffic Congress (ITC)*, 2013.
- [16] M. Jarschel, F. Lehrieder, Z. Magyari, and R. Pries, "A Flexible OpenFlow-Controller Benchmark," in *European Workshop on Software Defined Networks (EWSDN)*, 2012.
- [17] Z. Drezner, *Facility Location: A Survey of Applications and Methods*. Springer, 1995.
- [18] S. H. Owen and M. S. Daskin, "Strategic Facility Location: A Review," *European Journal of Operational Research*, 1998.
- [19] P. Czyżżak and A. Jaskiewicz, "Pareto simulated annealing - a meta-heuristic technique for multiple-objective combinatorial optimization," *Journal of Multi-Criteria Decision Analysis*, 1998.
- [20] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, "Optimization by Simulated Annealing," *Science*, 1983.
- [21] L. Kaufman and P. J. Rousseeuw, *Finding groups in data: an introduction to cluster analysis*. John Wiley & Sons, 2009.
- [22] S. Lloyd, "Least squares quantization in PCM," *IEEE Transactions on Information Theory*, 1982.
- [23] C. E. Shannon, "A Mathematical Theory of Communication," *The Bell System Technical Journal*, 1948.
- [24] J. Huang, "Combining Entropy Weight and TOPSIS Method for Information System Selection," in *IEEE Conference on Cybernetics and Intelligent Systems*, 2008.
- [25] C.-L. Hwang and K. Yoon, *Multiple Attribute Decision Making - Methods and Applications: A State-of-the-art Survey*. Springer, 1981.
- [26] S. H. Zanakis, A. Solomon, N. Wishart, and S. Dublish, "Multi-Attribute Decision Making: A Simulation Comparison of Select Methods," *European Journal of Operational Research*, 1998.
- [27] S. Opricovic and G.-H. Tzeng, "Compromise Solution by MCDM Methods: A Comparative Analysis of VIKOR and TOPSIS," *European Journal of Operational Research*, 2004.
- [28] A. Cohen, "Analysis of Large Sets of Ranking Data," *Communications in Statistics-Theory and Methods*, 1982.
- [29] R. D. Luce, *Individual Choice Behavior*. Courier Corporation, 1959.
- [30] C. L. Mallows, "Non-Null Ranking Models. I," *Biometrika*, 1957.
- [31] P. D. Feigin and A. Cohen, "On a Model for Concordance between Judges," *Journal of the Royal Statistical Society. Series B (Methodological)*, 1978.
- [32] A. Gordon, "A Measure of the Agreement between Rankings," *Biometrika*, 1979.
- [33] A. Clemm, *Network management fundamentals*, 2006.
- [34] M. Subramanian, *Network management: principles and practice*, 2010.
- [35] H. E. Egilmez, S. T. Dane, K. T. Bagci, and A. M. Tekalp, "OpenQoS: An OpenFlow controller design for multimedia delivery with end-to-end Quality of Service over Software-Defined Networks," in *Asia-Pacific Signal & Information Processing Association Annual Summit and Conference (APSIPA ASC)*, 2012.
- [36] H. E. Egilmez, S. Civanlar, and A. M. Tekalp, "An optimization framework for QoS-enabled adaptive video streaming over OpenFlow networks," *IEEE Transactions on Multimedia*, 2013.
- [37] S. Tomovic, N. Prasad, and I. Radusinovic, "SDN control framework for QoS provisioning," in *22nd Telecommunications Forum Telfor*, 2014.
- [38] P. Sharma, S. Banerjee, S. Tandel, R. Aguiar, R. Amorim, and D. Pinheiro, "Enhancing network management frameworks with SDN-like control," in *Integrated Network Management (IM 2013), 2013 IFIP/IEEE International Symposium on*, 2013.
- [39] D. Valocchi, D. Tuncer, M. Charalambides, M. Femminella, G. Reali, and G. Pavlou, "SigMA: Signaling Framework for Decentralized Network Management Applications," *IEEE Transactions on Network and Service Management*, 2017.