International Journal for Innovative Engineering and Management Research
A Peer Revieved Open Access International Journal
www.ijiemr.org

IJIEMR Transactions, online available on 1st Jan 2021. Link :http://www.ijiemr.org/downloads.php?vol=Volume-09&issue=ISSUE-12

DOI: 10.48047/IJIEMR/V09/I12/141

Title: **TOWARD INTELLIGENT NETWORK OPTIMIZATION IN WIRELESS NETWORKING AN AUTO-LEARNING FRAMEWORK**

Volume 09, Issue 12, Pages: 829-834

Paper Authors

CH. AKANKSHA, M. PRANAYA, S. SWETHA, GANDLA SWATHI, G.VENKAT RAMANA

International Journal for Innovative Engineering and Management Research
A Peer Revieved Open Access International Journal
www.ijiemr.org

TOWARD INTELLIGENT NETWORK OPTIMIZATION IN WIRELESS NETWORKING AN AUTO-LEARNING FRAMEWORK

CH. AKANKSHA[1], M. PRANAYA[2], S. SWETHA[3], GANDLA SWATHI[4], G.VENKAT RAMANA[5]

[1,2,3,4,] B TECH Students, Department of CSE, Princeton Institute of Engineering & Technology For Women, Hyderabad, Telangana, India.

[5] Assistant Professor, Department of CSE, Princeton Institute of Engineering & Technology For Women, Hyderabad, Telangana, India.

ABSTRACT: In remote correspondence frameworks (WCSs), network enhancement issues (NOPs) assume a significant job in augmenting framework execution by setting fitting organization setups. When managing NOPs by utilizing ordinary enhancement techniques, there exist the accompanying three issues: human mediation, model deficiency, and high calculation unpredictability. In that capacity, in this article, we propose an auto-learning system to accomplish shrewd and programmed network advancement by utilizing AI (ML) strategies. We survey the essential ideas of ML, and propose their simple business models in WCSs, including programmed model development, experience replay, proficient experimentation, RL-driven gaming, unpredictability decrease, and arrangement suggestion. We trust these recommendations can give new bits of knowledge and inspiration in future exploration for managing NOPs in WCSs by utilizing ML methods.

I. INTRODUCTION

In wireless communication systems (WCSs), network optimization problems (NOPs) have been extensively studied to maximize system performance by setting appropriate network configuration settings [1]. NOP contains a broad range of research aspects in wireless networking; typical applications include resource allocation and management, system parameter provision, task scheduling, and user quality of service (QoS) optimization. Figure 1 shows the basic process of solving a NOP in WCSs, which includes the following four steps. Data Collection: the collection of essential information of the system and the surrounding environment. The collected data can be channel state information (CSI), interference, noise, user location, spectrum and time slot occupations, and so on. Some QoS information, such as delay and energy consumption rates and mobility state, can also be the input data to support the following optimization process. Model Construction: in which the expert constructs an optimization model that contains an objective function and several constraints. The objective of the optimization model can be throughput, spectrum utilization, user-perceived delay, energy consumption/gain, facility deploy (cment cost, and so on. Typically, model construction is conducted by using a mathematical formulation process, and experts are required to master the domain knowledge and theories involved in the model. Optimization: The most commonly used methodologies for solving optimization problems are mathematical derivation-based methods (DBMs) and heuristic algorithms. The former adopt a mathematical derivation process to find the solution, such as the Lagrangian duality, Karush-Kuhn-Tucker (KKT) conditions,

International Journal for Innovative Engineering and Management Research
A Peer Revieved Open Access International Journal
www.ijiemr.org

and gradient descendent methodologies. The latter adopt a heuristic neighborhood searching process to approach the optimal solution, including genetic algorithm, simulated annealing, particle swarm optimization, firefly algorithms, and so on. In general, DBMs are quite suitable for solving problems with explicit and convex objective functions, while heuristic algorithms do not require the derivatives of the objective functions, and are generally able to produce high-quality solutions for complex optimization problems if the optimization complexity is suitably high [2]. Besides the above two optimization methods, game theoretical techniques, including non-cooperative games, cooperative games, and Bayesian games, also have been successfully applied to solve the optimization problem by learning automatic configuration strategies from interactions with other functional nodes [3]. Configuration: With the optimization results, the system then reconfigures the settings of the system to improve the performance. Possible reconfigurations may include transmission power allocation, energy harvesting scheduling, routing decision, and spectrum resource allocation, to name a few. After configuration, the system then repeats the optimization process to keep the system in suitable working conditions. Although NOPs have been extensively studied in WCSs, existing optimization methodologies still face the following three dilemmas. Human intervention: The optimization models in NOPs are always constructed by experts with domain knowledge, and this knowledge-driven process is expensive and inefficient in practical implementations. If we can conduct the optimization operations automatically, network optimization will be easier to conduct in real world applications. However, how to reduce human intervention in solving NOPs is still an unexplored field in WCSs. When the training data are not sufficient, the system may need to conduct a re-sampling process to collect more data. A data filtering process needs to be done since the quality of used data has critical influences on the performance of the obtained black-box model. The outliers, incomplete data, and repeating data will be abandoned or refined in data filtering process.

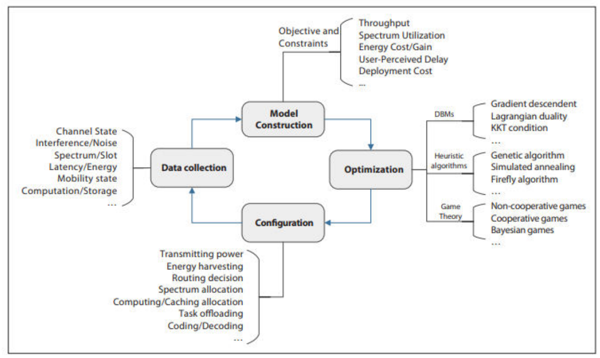

Figure 1: Workflow of network management in wireless communication systems.

Auto-Learning Framework As shown in Fig. 2, we propose ALF to achieve intelligent and automatic network optimization in WCSs. The basic workflow of ALF includes the following three steps. Data Collection. Collecting the experience data is the prerequisite for conducting ML-based models [6] and must be properly addressed. Besides the system and environment state information, in ALF the output solution data of an optimization process is also collected as historical experience. When the training data are not sufficient, the system may need to conduct a resampling process to collect more data. A data filtering process needs to be done since the quality of used data has a critical influence on the performance of the obtained blackbox model. The outliers,

International Journal for Innovative Engineering and Management Research
A Peer Revieved Open Access International Journal
www.ijiemr.org

incomplete data, and repeating data are abandoned or refined in the data filtering process. Model Training. The model training process is conducted in an ML engine, in which different ML techniques are provided, including supervised learning, RL, and unsupervised learning. Their detailed application models are introduced in the following section. After training, a cross validation process needs to be conducted to test the performance of the obtained model. More specifically, when the learning problem is a regression problem, that is, outputs are continuous; the performance metric is the mean square error (MSE) between the predicted results and real outputs. When the outputs are discrete decisions, collecting a large number of data samples in a short time may be impractical for some systems with very high reconfiguration cost, such as the reconfiguration of virtualized network function resources in software defined WCSs.

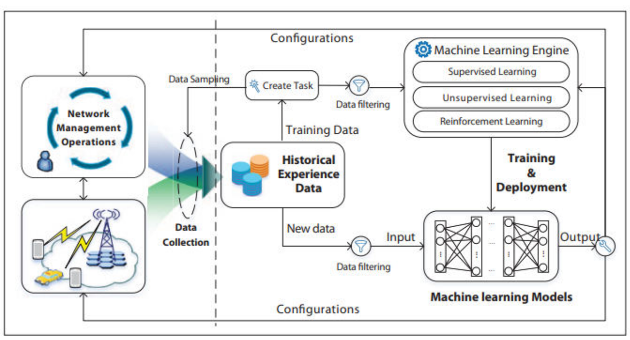

Figure 2: Auto-learning framework for dealing with NOPs in WCSs

Therefore, how to reduce training data samples is critical in automatic model construction-based NOPs the problem can be regarded as a classification problem, and the performance metric can be classification accuracy.

Supervised Learning-Automatic Model Construction and Experience Replay: With sufficient training data, a complex nonlinear mapping function from input data space to the output data space can be obtained by training a supervised learning model. Benefitting from this learning ability, supervised learning has been successfully applied in point-to-point learning tasks in communications systems, such as delay prediction, channel estimation, and signal detection. According to the amount of training samples, supervised learning can be divided into the following two categories: small-sample learning (SSL) and deep learning (DL). Possible choices for SSL include shallow neural networks, kernel-based methods, and ensemble learning methods. For DL, possible choices include deep belief networks, deep Boltzmann machines, and deep convolutional neural networks.

Automatic Model Construction Model: Supervised-learning-based black-box regression provides an effective way to solve the expensive human intervention and model invalidity problems. In situations when the explicit functions between the input and output are not available, but we have sufficient data samples that contain the inputs and outputs of the system, the mapping function can be trained by using a supervised regression technique. Given new input data, the target performance objective can be accurately predicted by using the previously obtained model. We propose to use supervised learning techniques to automatically conduct the model construction process in NOPs. As illustrated in Fig. 3a, in conventional NOPs, the mathematical optimization model is constructed by experts with domain knowledge. In ALF, we propose to use black-box modeling to automatically construct the optimization model, as shown in Fig. 3b. In the automatic model construction process, we can directly regress the objective function and constraints by using regression models. In the same way, the constraints can also be constructed. With the obtained model, a following heuristic algorithm can be used to solve the optimization model, since it just needs to know the objective response in each searching iteration. When the target function contains several independent parts, we can first train the independent mapping functions of these parts and then combine them into a unified one. For example, in mobile edge computing, the user-perceived delay mainly includes three parts: data transmission time, queuing time, and task execution time. In this scenario, we can build the optimization model by combining the three black-box delay time prediction models. Challenges: The successful implementation of a supervised learning method requires a dataset with sufficient and reliable data samples to train the mapping model. In some tasks like network delay and energy consumption rate prediction, the data samples can be collected easily. However, collecting a large number of data samples in a short time may be impractical for some systems with very high reconfiguration cost, such as the reconfiguration of virtualized network function resources in software defined WCSs. Therefore, how to reduce training data samples is critical in automatic model construction-based NOPs.

Unsupervised Learning: Complexity Reduction and Solution Recommendation A clustering algorithm is one typical unsupervised learning method that aims to partition the data into several clusters with similar regional distribution properties. The k-means algorithm is an efficient and effective clustering algorithm, and it can be used to solve most clustering problems [14]. Also, the similarity learning process used in k-nearest neighbor (k-NN) search can be used in finding recommended solutions. Complexity Reduction Model: It is recognized that increasing variable dimensions will greatly increase the complexity of the optimization process. We therefore discuss the potential of using clustering algorithms to reduce the complexity of NOPs with high-dimensional variables. As shown in Fig. 5, we can modify the original NOP into a hierarchical NOP problem to reduce the complexity by dividing the target high-dimensional variables into several clusters. First, a cluster-level optimization process is conducted; then variable-level optimization is executed within each cluster. In this way, since the cluster number and variable dimension of each cluster is much smaller than the original variable vector, the complexity of the optimization process can be greatly reduced. Applications: In applications like resource management with large numbers of variables, the optimization process can be an expensive task with high-dimensional target variables. In this situation, the model complexity can be relaxed by using a clustering process. The variable vector can be divided into several sub-vectors according to factors like throughput demand, channel states, computation demands, and data transmission amount. Some other factors, such as user priority, geographical position, and residual energy, also can be used as the features for clustering. In this way, optimization can be conducted at the cluster level and task level separately, and the complexity can be significantly reduced. Challenges: The drawback of clustering-based hierarchical optimization is that the obtained results may suffer from a performance loss since the hierarchical optimization process is not the same as the original one, and cluster-optimal results are not equivalent to variable-optimal results. Therefore, how to reduce the performance loss in hierarchical optimization is a challenge for future work. Solution Recommendation Model: One can use a similarity measurement to find similar historical tasks, then directly combine the solution of this similar task as the solution of the new task. To realize similarity-based solution recommendation (SSR) in ALF, first we define the feature vector that is able to distinguish the differences of the tasks, and subsequently a k-NN searching process can be used to find the tasks with similar features. The k-NN algorithm is a well-known lazy learning method that searches the nearest instances according to similarity measurements, and it can be efficiently realized by using a kd-tree algorithm. We assume that the environment stays stable in a period of time. Given a new task, when the historical tasks with similar features are known, we can combine the solutions of these similar tasks and directly use the average result as the solution. Applications: Large-scale power allocation is usually a computation-intensive task due to the high dimension of the solution. If we have sufficient historical feature data, the SSR can be used to solve the real-time optimization problem. The feature data can be defined as a vector containing user geographic location and user terminal type. When the locations are close to each other, the corresponding CSI will be similar. In addition, when the user terminal type is the same, their antenna capacities will also be the same. In this way, the power assignments will also be similar. Challenges: First, collecting user feature data may impose privacy concerns since the manager may want to collect sensitive information, such as geographic locations, user behaviors, and user preferences. Second, since SSR assumes that the environment stays static in a period of time, it is not able to deal with problems with dynamic or stochastic conditions. Third, the recommended solution is just an approximate version of the real one, and the corresponding performance will also not be optimal. Forth, the distribution of the collected data may not be evenly distributed. For some new tasks without a sufficiently close neighbor, SSR will fail to find reliable results.

CONCLUSIONS

This article recalls the models of network optimization in WCSs and proposes an ALF that employs the advantages of powerful ML techniques to deal with the human intervention, model invalidity, and high complexity problems in conventional optimization models. We review the basic concepts of supervised learning, reinforcement learning, and unsupervised learning, and then propose several potential models to deal with NOPs, including automatic model construction, experience replay, efficient trial and error, RL-driven gaming, complexity reduction, and solution recommendation. We encourage readers to test and modify these proposals, and further design more new ML-based methods for dealing with NOPs in WCSs.

REFERENCES

[1] David Tse and Pramod Viswanath, Fundamentals of Wireless Communication, Cambridge Univ. Press, 2005.

International Journal for Innovative Engineering and Management Research
A Peer Revieved Open Access International Journal
www.ijiemr.org

[2] E. K. P. Chong and S. H. Zak, An Introduction to Optimization, 3rd ed., 2011.

[3] W. Saad et al., "Coalitional Game Theory for Communication Networks: A Tutorial," IEEE Signal Processing Mag., vol. 26, no. 5, 2009, pp. 77–97.

[4] M. Wang et al., "Machine Learning for Networking: Workflow, Advances and Opportunities," IEEE Network, vol. 31, no. 2, Mar./Apr. 2017.

[5] M. Chen et al., "Cognitive-LPWAN: Towards Intelligent Wireless Services in Hybrid Low Power Wide Area Networks", IEEE Trans. Green Commun. and Networking, vol. 3, no. 2, June 2019, pp. 407–17.

[6] M. Chen and V. Leung, "From Cloud-Based Communications to Cognition-Based Communications: A Computing Perspective," Comp. Commun., vol. 18, 2018, pp. 74–79.

[7] M. Chen et al., "Labelless Learning for Traffic Control in an Edge Network," IEEE Network, vol. 32, no. 6, Nov./Dec. 2018, pp. 8–14.

[8] K. Hwang et al., Big Data Analytics for Cloud/IoT and Cognitive Computing, Wiley. ISBN: 9781119247029, 2017.

[9] E Bjornson, M Kountouris, and M Debbah, "Massive MIMO and Small Cells: Improving Energy Efficiency by Optimal Soft-Cell Coordination," Int'l. Conf. Telecommun., 2013, pp. 1–5.

[10] G. B. Huang, "Extreme Learning Machines: A Survey," Int'l. J. Machine Learning & Cybernetics, vol. 2, no. 2, 2011, pp. 107–22.

[11] Y. He et al., "Deep Reinforcement Learning-Based Optimization for Cache-Enabled Opportunistic Interference Alignment Wireless Networks," IEEE Trans. Vehic. Tech., vol. 66, no. 11, 2018, pp. 10,433–45.

[12] J. Snoek, H. Larochelle, and R. P. Adams, "Practical Bayesian Optimization of Machine Learning Algorithms," Int'l. Conf. Neural Info. Processing Systems, 2012, pp 2951–59.