Energy-Aware Optimal Service Function Chain Embedding in Emulated Multi-Access Edge Computing for Internet of Things

Theviyanthan Krishnamohan, Paul Harvey
{theviyanthan.krishnamohan,paul.harvey}@glasgow.ac.uk
University of Glasgow
Glasgow, United Kingdom

Abstract

Rising data processing demands from ever-increasing numbers of IoT devices have led to increased use of edge compute deployments, where energy consumption has become a critical challenge. By switching from dedicated hardware functions to chains of virtualised network functions, called Service Function Chains (SFCs), energy consumption can be reduced when processing IoT network data. However, optimally embedding these functions in edge deployments to achieve this is $\mathcal{NP}$-hard, with existing solutions optimising only a subset of the problem space. In this work, we explore a Genetic Algorithm-based approach that optimises all three sub-problems scalably to minimise energy consumption in SFCs. We test our proposed solution across two Multi-Access Edge Computing scenarios and show that our solution efficiently converges on an optimal solution in terms of the number of embedded SFCs and energy consumed.

CCS Concepts: • **Computing methodologies** → *Heuristic function construction*; • **Networks** → **Network resources allocation**; **Network experimentation**; *Network performance modeling*; *Programmable networks*; *Cloud computing*; *Mobile networks.*

Keywords: Multi-Access Edge Computing, Internet of Things, Energy Optimisation, Service Function Chaining, Evolutionary Networks

ACM Reference Format:
Theviyanthan Krishnamohan, Paul Harvey. 2026. Energy-Aware Optimal Service Function Chain Embedding in Emulated Multi-Access Edge Computing for Internet of Things. In *4th International Workshop on Testing Distributed Internet of Things Systems (TDIS '26), April 27–30, 2026, Edinburgh, Scotland Uk.* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3802513.3803485

TDIS '26, Edinburgh, Scotland Uk

ACM ISBN /2026/04
https://doi.org/10.1145/3802513.3803485

1 Introduction

Processing the large volumes of network data generated by devices in the Internet of Things (IoT) [4] requires increased deployments of compute resources, which in turn increases energy consumption. To reduce the traffic load on the network and the delay in processing IoT data, Multi-access Edge Computing (MEC) has been proposed [4]. MEC brings compute resources to the edge of the network, reducing bandwidth and time consumption. However, processing data at the edge can require more energy than in the cloud [27]. In parallel, significant increases in the number of servers deployed at the edge and the complexity of MEC networks make energy consumption a significant problem [16, 33].

Service Function Chaining (SFC) is one way of facilitating the reduction of the energy consumed by MEC. To ensure safe and reliable transmission of data from IoT devices, MEC employ network functions, such as firewalls and Intrusion Detection Systems. Traditionally, these network functions are deployed as individual middlebox hardware [10] that consume significant energy [9], driving up overall MEC energy consumption. SFC uses Network Function Virtualisation (NFV) to separate the network functions from their hardware so that they can be virtually embedded on servers, and Software Defined Networking (SDN) to virtually link the Virtualised Network Functions (VNFs), forming a chain of VNFs [9]. The virtualisation achieved by SFC enables *programmatic* embedding of VNFs and Virtual Links (VLs), enabling network operators to optimally embed them on the physical MEC network by optimising objectives, such as maximising throughput and minimising energy consumption.

Optimally composing and embedding SFCs, which we call the SFC-Optimal Composition and Embedding (SFC-OCE) problem, is an $\mathcal{NP}$-hard optimisation problem [9] consisting of three sub-problems: optimally ordering the VNFs in an SFC named VNF Chain Composition (VNF-CC), optimally embedding the VNFs on servers named VNF Embedding (VNF-EM), and optimally embedding the virtual links across physical links named VL Embedding (VL-EM). All these three sub-problems have to be optimised to produce an optimal SFC embedding [9]. We found 17 studies in the literature that optimised the energy consumed by SFC embeddings [5, 12, 11, 34, 7, 28, 26, 19, 2, 23, 1, 29, 22, 3, 32, 24, 21]; only one [24] optimised all three sub-problems, however, it used an exact algorithm that does not scale well as the complexity of networks grows. Genetic Algorithms (GAs) are meta-heuristic algorithms that are well-known for producing scalable solutions to $\mathcal{NP}$-hard optimisation problems [18] and have been used in the literature to optimise SFC-OCE with different objectives, making it a promising solution space. Of the studies that used GA to optimise SFC-OCE, only one study minimised energy consumption [12], but it did not optimise all three sub-problems.

In this paper, we propose an approach based on GA to optimally embed SFCs in MEC for IoT workloads, such that their energy consumption is minimised and the acceptance ratio, which is the ratio between the number of SFCs that can be embedded and the number of SFC Requests (SFCRs), is maximised. We use the GENESIS framework [15] that we developed to encode and decode candidate solutions. To evaluate the fitness of the candidate solutions, we use a hybrid approach consisting of a faster but less accurate offline mechanism and a slower but more accurate online mechanism. The offline mechanism uses Deep Neural Network (DNN) models trained on VNF profiling data to approximate the CPU usage of VNFs and a linear CPU power model to compute the energy usage based on the CPU usage. The online mechanism uses the SFC emulation environment OpenRASE [13] we developed. To the best of our knowledge, this is the first paper to 1. optimise all three SFC-OCE sub-problems to minimise energy consumption in a scalable manner, and 2. use an emulator to evaluate an energy-aware approach in the context of MEC.

We evaluate our energy-aware GA approach across a random MEC network topology and a real MEC network topology in a dynamic network environment, where the number of SFCRs to embed, network traffic and topology vary with time. The results show that the GA approach converges on an optimal solution, taking a median of 3.84 minutes and minimising the energy consumption to a median of 730.8 kWh, while maintaining an acceptance ratio of 1.

2 Related Work

We surveyed the literature from 2014 to 2025 and identified 17 energy-aware SFC-OCE embedding approaches. 6 of these studies optimised only the VNF-EM sub-problem [5, 34, 26, 19, 23, 3]. 10 studies optimised both VNF-EM and VL-EM [12, 11, 7, 28, 2, 1, 29, 22, 32, 21]. Only one study optimised all three SFC-OCE sub-problems [24], but the approach used an exact algorithm, which would not scale well for complex networks. Only 2 studies considered a MEC context [19, 3].

To evaluate their approaches, 13 studies performed numerical evaluations [24, 21, 32, 22, 29, 28, 7, 11, 3, 23, 19, 26, 5], 3 used simulation [2, 1, 12], and one used a testbed [34].

Table 1. A summary of energy-aware SFC-OCE approaches.

Study	VNF-CC	VNF-EM	VL-EM	Context	Scalable	Evaluation
[5]	X	✓	X	Generic	✓	Numerical
[34]	X	✓	X	Generic	✓	Testbed
[26]	X	✓	X	Generic	✓	Numerical
[19]	X	✓	X	MEC	✓	Numerical
[23]	X	✓	X	Data Centre	✓	Numerical
[3]	X	✓	X	MEC	✓	Numerical
[12]	X	✓	✓	Generic	✓	Simulation
[11]	X	✓	✓	Data Centre	✓	Numerical
[7]	X	✓	✓	Data Centre	X	Numerical
[28]	X	✓	✓	Generic	✓	Numerical
[2]	X	✓	✓	Data Centre	✓	Simulation
[1]	X	✓	✓	Data Centre	✓	Simulation
[29]	X	✓	✓	Generic	✓	Numerical
[22]	X	✓	✓	Generic	✓	Numerical
[32]	X	✓	✓	B5G	✓	Numerical
[21]	X	✓	✓	Generic	✓	Numerical
[24]	✓	✓	✓	5G	X	Numerical

A summary of the energy-aware SFC-OCE approaches is given in Table 1. As shown, no study optimised all three SFC-OCE sub-problems in a scalable manner, leading to suboptimal energy minimisation. Additionally, all MEC and mobile network approaches (e.g. 5G or B5G) used numerical evaluation, limiting their accuracy [6].

3 GA-Based Optimal SFC Embedding

To address the $\mathcal{NP}$-hard challenge of energy-aware embedding of SFCs with a high acceptance ratio in MEC networks, we propose a GA-based approach that addresses all three SFC-OCE sub-problems. We first describe the problem formulation, followed by the genetic encoding-decoding scheme of our approach, the fitness evaluation mechanism, the fitness function, and the hyperparameters we used.

3.1 Problem Formulation

We consider a dynamic MEC networking environment for IoT, where SFCRs arrive at regular intervals. The traffic pattern and the topology also vary with time. Our goal is to automatically explore embedding SFCRs on the MEC network, such that the SFCR acceptance ratio is maximised, and the energy consumption of the MEC network is minimised. The energy consumption of a MEC is a function of the amount of resources used by MEC servers and the number of servers that are powered on [17, 20]. Consequently, increasing the SFCs embedded on the MEC network increases the energy consumption, making this a Pareto optimisation problem. Further, embedding many VNFs on one server increases its CPU usage, thereby increasing its energy consumption. Embedding VNFs across many servers reduces the energy consumed by an individual server; however, when many servers are powered on, the idle power consumed by servers increases the total energy consumption of the MEC network. An optimal SFC-OCE approach must balance between the number of SFCs to embed, the number of VNFs to embed on each server, and the number of servers that should be used.

3.2 Genetic Encoding and Decoding

We adapt our GENESIS[1] [15] framework to encode and decode candidate solutions. The GENESIS framework uses a one-dimensional floating-point array with 6 elements as the genetic encoding and uses three solvers to decode the solution. It uses a solver for each of the three SFC-OCE sub-problems. A solver consists of a predictor and a generator. A predictor is a Deep Neural Network (DNN) with one hidden layer with two neurons and takes the network topology, traffic, and the SFCRs as inputs. The weight values of the hidden layer are given by the genetic encoding. A generator takes the output of a predictor to generate a solution to the concerned SFC-OCE sub-problem. In effect, GENESIS uses GA to evolve three DNNs that are then used to produce an optimal SFC embedding. GENESIS is, according to the best of our knowledge, the only genetic encoding-decoding scheme enabling optimisation of all three SFC-OCE sub-problems.

3.3 Fitness Evaluation

We now describe the online-offline evaluation mechanism and the fitness function to score individuals.

3.3.1 Online Evaluation. The fitness of an individual is evaluated by deploying them on an actual network and running experiments to evaluate their performance. This provides an accurate fitness evaluation of an individual, but takes a long time compared to numerical approaches. For instance, the OpenRASE emulator used in this paper requires 10 minutes to evaluate a candidate solution [13]. Despite the accuracy provided, the longer evaluation time limits the size of the population and the number of generations, putting the evolution at risk of getting stuck at local optima.

3.3.2 Offline Evaluation. In offline evaluation, the fitness of an individual is evaluated numerically, via simulation or predictive models. Offline evaluation is faster than online evaluation, but less accurate. This enables GA to efficiently evolve several thousand individuals in a population over several hundred generations. However, the inaccuracy of this mechanism risks converging on a sub-optimal solution.

3.3.3 Hybrid Evaluation. Hybrid evaluation combines online and offline evaluation. GA evolves using offline evaluation until a threshold is met, and then uses online evaluation

[1] https://github.com/Project-Kelvin/OpenRASE/tree/main/src/algorithms/hybrid

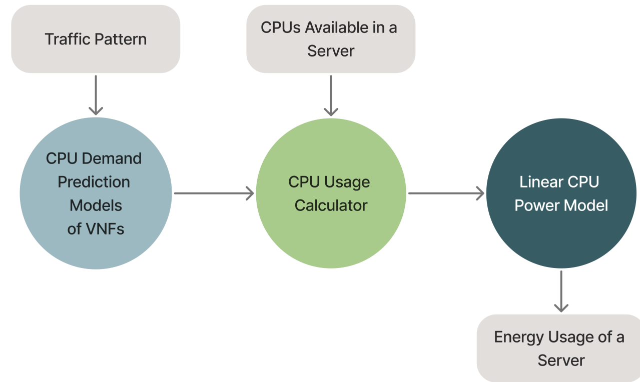

Figure 1. We find the CPU demand of VNFs embedded on a server using DNN models, sum them, divide them by the number of CPUs in the server, and use the linear CPU power model to estimate the energy usage.

to verify the fitness of the best individuals and correct course if the fitness approximated by offline evaluation is not accurate enough. This allows GA to evolve faster with many individuals over several generations, minimising the risk of getting stuck at local optima, while ensuring the converged solution is optimal. Hybrid evaluation combines the speed of offline evaluation with the accuracy of online evaluation.

In this study, we use hybrid evaluation to evaluate the fitness of individuals. We use the OpenRASE emulator to evaluate the fitness of individuals online via experimentation. For offline evaluation, we use the DNN models from our BENNS model [14] to predict the CPU demand of VNFs for a given number of traffic requests. We sum the CPU demand of each VNF embedded on a server and divide it by the total number of CPUs available in that server to find the CPU usage of the server, as shown in Fig. 1.

3.3.4 Defining Fitness. We define the fitness of an individual in terms of the acceptance ratio and the energy consumed by the MEC network. We maximise the acceptance ratio as shown in Equation 1, where s is the number of SFCs that can be embedded and t is the total number of SFCRs received.

$$\max \frac{s}{t} \tag{1}$$

For energy consumption, a study on Google's data centre showed that energy usage can be accurately estimated using only the CPU usage [20]. Based on this study, we consider only the CPU usage of the servers in the MEC network to create a simplified energy model. The energy consumed by memory, storage, and networking equipment will be considered in future work. We use the linear CPU power model, which has been shown to offer comparatively higher accuracy [17], to estimate the energy consumption based on the predicted CPU usage of VNFs. The CPU power model is given in Equation 2, where $u \in [0, 1]$ is the CPU utilisation,

p_{min} is the power usage when the CPU is idle, and p_{max} is the power consumed at maximum CPU usage.

$$P(u) = p_{min} + (p_{max} - p_{min})u \tag{2}$$

We used Intel's PowerTOP[2] tool to estimate the power consumed by one CPU core of the test machine (Section 4) in the idle state and when it is stressed to its maximum. We used the stress-ng[3] tool to stress the server. Accordingly, p_{min} was found to be 8 W while p_{max} was 11 W. We minimise the the power consumed by the MEC network, as shown in Equation 3, where H is the set of servers in the MEC network, u_h is the CPU usage of server h, and $P(u_h)$ is the power consumed by server h as given by Equation 2.

$$\min \sum_{h \in H} P(u_h) \tag{3}$$

We use the CPU power model and the DNN models to approximate the energy consumption of the MEC network until the threshold is met. We then evaluate the best individual on OpenRASE by embedding the individual on the emulated MEC network, measuring the CPU usage of the MEC servers, and using the CPU power model from Equation 2 to compute the energy consumption. As the acceptance ratio is mathematically computed, experimental evaluation is not required.

3.4 Hyperparameters

The initial population was generated as 100 random individuals from a uniform distribution. We used the NSGA-II algorithm for parent selection since maximising acceptance ratio and minimising energy consumption is a Pareto optimisation problem (Section 3.1). We used blend crossover to perform crossover, as our genetic encoding uses a floating-point array. We set its alpha value to 0.5 as it is the typical value. Based on the initial set of experiments, we set the individual mutation probability and gene mutation probability to 0.5 as these parameters produced the best GA performance. Mutation was carried out by randomly sampling a gene value from a Gaussian distribution.

4 Evaluation

We evaluated our energy-aware GA approach on a virtual machine running Ubuntu 20.04.6 on a QEMU hypervisor with 64 cores of Intel Xeon Gold 6240R CPUs having a clock speed of 2.4 GHz, and 128 GB of RAM. We designed 4 distinct SFCRs based on VNFs used in IoT and MEC networks from the literature [31, 8]: 1. Load Balancer → Web Application Firewall 2. HTTP Accelerator → Load Balancer → Web Application Firewall 3. HTTP Accelerator → Traffic Monitor → Load Balancer → Web Application Firewall 4. Load Balancer → Traffic Monitor → Web Application Firewall. In OpenRASE, we implemented deployable instances of these VNFs to evaluate the performance of our approach. To ensure more realistic evaluation, we used a randomly generated MEC network topology, called 25N50E (25 nodes, 50 edges), and a real MEC network topology from the Milan city centre (30 nodes, 35 edges) [30]. We scaled down the resources available in the topology by 100 to be able to emulate an entire MEC network in our test machine. We set the number of CPUs available in each MEC server to 1, the memory available to 5 GB, and the link bandwidth to 10 Mbps. We generated traffic requests to SFCs from a real traffic trace from an IoT testbed [25].

We scaled down an hour in the 24-hour traffic trace to a minute and ran two different experiments using the 25N50E topology and the Milan City Centre topology in a dynamic environment. In the dynamic environment, every 1 minute, a set of 2 SFCRs arrived, and the traffic pattern changed. Our GA approach completes its current evolution before adapting to a change in the environment. We call each evolution following a change a segment. After the 18th segment, every 1 minute, we crashed a server, altering the network topology.

We set a *minimum acceptance ratio of 1* and a *maximum energy consumption of 1000 kWh* as the fitness threshold.

5 Results

Time to Solve: Our energy-aware GA approach converged on an optimal solution following every change in the dynamic environment, taking a median time of 3.84 minutes (3.01-12.43) in the Milan City Centre experiment, and 3.95 minutes (2.98-7.59) in the 25N50E experiment. Fig. 2 shows the time taken by each experiment to converge on an optimal solution during each segment. The time taken gradually increased as the number of SFCRs increased, before rising sharply as we started crashing servers one by one (to the right of the red line), increasing the complexity of the SFC-OCE problem.

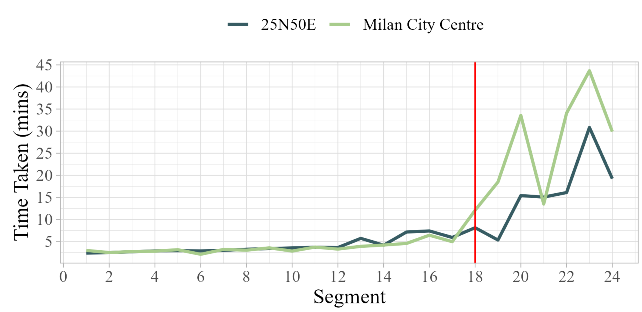

Figure 2. The time taken to converge in each segment. The vertical red line indicates the point from which we started crashing servers.

Energy Minimisation: The median energy consumption of the MEC network was 779.4 kWh (535.5-873.0) during the Milan City Centre experiment, and 720.0 kWh (532.8-756.9) during the 25N50E experiment. Fig. 3 and 4 show how the

[2] https://github.com/fenrus75/powertop
[3] https://github.com/ColinIanKing/stress-ng

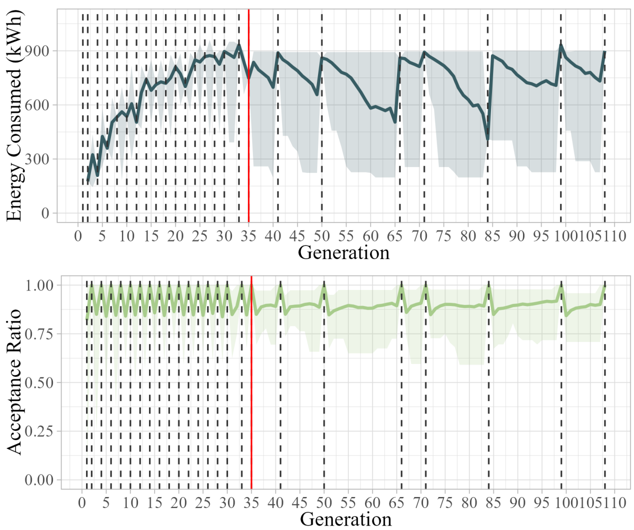

Figure 3. The acceptance ratio and energy consumption during the Milan City Centre experiment.

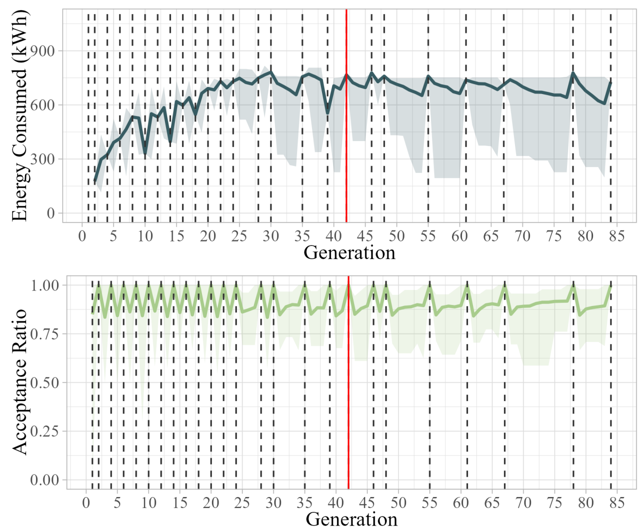

Figure 4. The acceptance ratio and energy consumption during the 25N50E experiment.

energy consumption and acceptance ratio varied as the GA evolved to adapt to the changes in the environment. The dashed vertical lines show the segments. The vertical red line shows the point from which we started crashing servers. The solid curve shows the average fitness, while the shaded region shows the range between maximum and minimum fitness. The average curve and the shaded range meet at the dashed lines because, at these points, the evolution switched from offline evaluation to online evaluation, selecting only the best-fit individual, reducing the population to just one individual. The number of generations taken to converge increased once we started crashing servers, as it increased the complexity of the SFC-OCE problem.

The results show that our energy-aware GA approach can minimise energy consumption without sacrificing on the acceptance ratio within bounded execution times in dynamic network environments, despite the complexity of the problem increasing as the SFCRs continue to arrive and servers start to crash.

6 Conclusion

SFCs can reduce energy consumption in MEC networks by enabling network providers to optimally embed virtualised network functions on MEC networks. Optimally embedding SFCs is an $\mathcal{NP}$-hard optimisation problem with three sub-problems. This paper proposes an energy-aware GA approach to optimally embed SFCs on MEC networks, maximising the number of embedded SFCs, while minimising their energy consumption. We evaluate our approach on real MEC network topologies, using SFCRs based on VNFs used in MEC and IoT networks, and a real traffic trace obtained from an IoT testbed to perform dynamic network experiments. The experiments show that our energy-aware GA approach converges within a median of 3.84 minutes and minimises the median energy consumption of the network to 730.8 kWh.

References

[1] Marwa A. Abdelaal, Gamal A. Ebrahim, and Wagdy R. Anis. 2021. Efficient Placement of Service Function Chains in Cloud Computing Environments. *Electronics*, (2021).

[2] Marwa A. Abdelaal, Gamal A. Ebrahim, and Wagdy R. Anis. 2021. High Availability Deployment of Virtual Network Function Forwarding Graph in Cloud Computing Environments. *IEEE Access*.

[3] Anselmo Luiz Éden Battisti, Evandro Luiz Cardoso Macedo, Marina Ivanov Pereira Josué, Hugo Barbalho, Flávia C. Delicato, Débora Christina Muchaluat-Saade, Paulo F. Pires, Douglas Paulo de Mattos, and Ana Cristina Bernardo de Oliveira. 2022. A Novel Strategy for VNF Placement in Edge Computing Environments. *Future Internet*, (2022).

[4] Davide Borsatti, Gianluca Davoli, Walter Cerroni, and Carla Raffaelli. 2021. Enabling Industrial IoT as a Service with Multi-Access Edge Computing. *IEEE Communications Magazine*.

[5] Roberto Bruschi, Alessandro Carrega, and Franco Davoli. 2016. A Game for Energy-Aware Allocation of Virtualized Network Functions. *Journal of Electrical and Computer Engineering*.

[6] Hossam Mahmoud Ahmad Fahmy. 2023. Simulators and Emulators for WSNs. In *Concepts, Applications, Experimentation and Analysis of Wireless Sensor Networks*. Springer Nature Switzerland, Cham.

[7] Behrooz Farkiani, Bahador Bakhshi, and S. A. Mirhassani. 2019. A Fast Near-Optimal Approach for Energy-Aware SFC Deployment. *IEEE Transactions on Network and Service Management*, (2019).

[8] Xiaoyuan Fu, F Richard Yu, Jingyu Wang, Qi Qi, and Jianxin Liao. 2020. Dynamic Service Function Chain Embedding for NFV-Enabled IoT: A Deep Reinforcement Learning Approach. *IEEE Transactions on Wireless Communications*.

[9] Juliver Gil Herrera and Juan Felipe Botero. 2016. Resource Allocation in NFV: A Comprehensive Survey. *IEEE Transactions on Network and Service Management*, (2016).

[10] Shuang Guo, Yarong Du, and Liang Liu. 2023. A Meta Reinforcement Learning Approach for SFC Placement in Dynamic IoT-MEC Networks. *Applied Sciences*.

[11] Insun Jang, Dongeun Suh, Sangheon Pack, and György Dán. 2017. Joint optimization of service function placement and flow distribution for service function chaining. *IEEE Journal on Selected Areas in Communications*, (2017).

[12] Siri Kim, Yunjung Han, and Sungyong Park. 2016. An Energy-Aware Service Function Chaining and Reconfiguration Algorithm in NFV. In *2016 IEEE 1st International Workshops on Foundations and Applications of Self* Systems (FAS*W)*.

[13] Theviyanthan Krishnamohan and Paul Harvey. 2025. OpenRASE: Service Function Chain Emulation. In *International Conference on Software, Telecommunications and Computer Networks (SoftCOM)*.

[14] Theviyanthan Krishnamohan, Lauritz Thamsen, and Paul Harvey. 2025. BENNS: A Surrogate Model for Hybrid Online-Offline Evolution of SFC Embedding. *arXiv e-prints*, (2025).

[15] Theviyanthan Krishnamohan, Lauritz Thamsen, and Paul Harvey. 2025. Simultaneous Genetic Evolution of Neural Networks for Optimal SFC Embedding. *arXiv e-prints*, (2025).

[16] Chunlin Li, Yong Zhang, Qinqin Sun, and Youlong Luo. 2021. Collaborative caching strategy based on optimization of latency and energy consumption in MEC. *Knowledge-Based Systems*.

[17] Antonios T Makaratzis, Christos K Filelis-Papadopoulos, Konstantinos M Giannoutakis, George A Gravvanis, and Dimitrios Tzovaras. 2017. A comparative study of CPU power consumption models for cloud simulation frameworks. In *Proceedings of the 21st Pan-Hellenic Conference on Informatics* (PCI '17). Association for Computing Machinery, New York, NY, USA.

[18] Melanie Mitchell. 1996. *An Introduction to Genetic Algorithms*. The MIT Press, (1996).

[19] Phuong Duy Nguyen and Long Bao Le. 2020. Joint Computation Offloading, SFC Placement, and Resource Allocation for Multi-Site MEC Systems. In *2020 IEEE Wireless Communications and Networking Conference (WCNC)*. Institute of Electrical and Electronics Engineers Inc., (2020).

[20] Ana Radovanović et al. 2023. Carbon-Aware Computing for Datacenters. *IEEE Transactions on Power Systems*.

[21] Rania Sahraoui, Omar Houidi, and Fetia Bannour. 2024. Energy-Aware VNF-FG Placement with Transformer-based Deep Reinforcement Learning. In *NOMS 2024-2024 IEEE Network Operations and Management Symposium*.

[22] Guto Leoni Santos, Patricia Takako Endo, Theo Lynn, Djamel Sadok, and Judith Kelner. 2022. A reinforcement learning-based approach for availability-aware service function chain placement in large-scale networks. *Future Generation Computer Systems*, (2022).

[23] Guto Leoni Santos, Theo Lynn, Judith Kelner, and Patricia Takako Endo. 2021. Availability-aware and energy-aware dynamic SFC placement using reinforcement learning. *Journal of Supercomputing*, (2021).

[24] Ehsan Sargolzaei, Mehdi Rasti, and Siavash Khorsandi. 2025. Topology and Energy Aware Approximate Algorithm for QoS-based Resource Slicing in 5G Core Networks. *IEEE Access*.

[25] Arunan Sivanathan, Hassan Habibi Gharakheili, Franco Loi, Adam Radford, Chamith Wijenayake, Arun Vishwanath, and Vijay Sivaraman. 2019. Classifying IoT Devices in Smart Environments Using Network Traffic Characteristics. *IEEE Transactions on Mobile Computing*.

[26] Ruben Solozabal, Josu Ceberio, Aitor Sanchoyerto, Luis Zabala, Bego Blanco, and Fidel Liberal. 2020. Virtual Network Function Placement Optimization with Deep Reinforcement Learning. *IEEE Journal on Selected Areas in Communications*, (2020).

[27] Xiang Sun and Nirwan Ansari. 2016. EdgeIoT: Mobile Edge Computing for the Internet of Things. *IEEE Communications Magazine*.

[28] Mohammad M. Tajiki, Stefano Salsano, Luca Chiaraviglio, Mohammad Shojafar, and Behzad Akbari. 2019. Joint Energy Efficient and QoS-Aware Path Allocation and VNF Placement for Service Function Chaining. *IEEE Transactions on Network and Service Management*, (2019).

[29] Amir Varasteh, Basavaraj Madiwalar, Amaury Van Bemten, Wolfgang Kellerer, and Carmen Mas-Machuca. 2021. Holu: Power-Aware and Delay-Constrained VNF Placement and Chaining. *IEEE Transactions on Network and Service Management*, (2021).

[30] Bin Xiang, Jocelyne Elias, Fabio Martignon, and Elisabetta Di Nitto. 2021. A dataset for mobile edge computing network topologies. *Data in Brief*.

[31] Lingyi Xu, Wenbin Liu, Zhiwei Wang, Jianxiao Luo, Jinjiang Wang, and Zhi Ma. 2024. Mobile-Aware Service Function Chain Intelligent Seamless Migration in Multi-access Edge Computing. *Journal of Network and Systems Management*.

[32] Zexi Xu, Lei Zhuang, Weihua Zhuang, Yuxiang Hu, Wenshuai Mo, and Zihao Wang. 2023. Meta Relational Learning-Based Service-Tailored VNF Deployment for B5G Network Slice. *IEEE Internet of Things Journal*.

[33] Jinming Yang, Awais Aziz Shah, and Dimitrios Pezaros. 2023. A Survey of Energy Optimization Approaches for Computational Task Offloading and Resource Allocation in MEC Networks. *Electronics*.

[34] Bangchao Yu, Wei Zheng, Xiangming Wen, Zhaoming Lu, Luhan Wang, and Lu Ma. 2018. Dynamic Resource Orchestration of Service Function Chaining in Network Function Virtualizations. In *5G for Future Wireless Networks*. Springer International Publishing, Cham.