

Intelligent Mapping of Virtualized Services on Multi-Domain Networks

Vinicius Fulber-Garcia, Marcelo C. Luizelli, Carlos R. Paula dos Santos, Eduardo J. Spinosa, and Elias P. Duarte Jr.

Abstract One of the challenges of the Network Functions Virtualization (NFV) paradigm is to deploy virtualized network functions and services efficiently. In particular, current solutions for multi-domain service mapping present several restrictions regarding the choice of optimization models and metrics. This lack of flexibility ultimately leads to sub-optimized mappings that do not meet the (often conflicting) requirements of all the parties involved in the deployment process (*e.g.*, network operators, clients, providers). This work proposes GeSeMa (Genetic Service Mapping), a new intelligent mapping solution based on genetic algorithms. GeSeMa allows the specification of arbitrary optimization metrics, constraints, and different evaluation policies. We evaluate GeSeMa through a case study, comparing its results with the results of a state-of-the-art genetic-based mapping solution.

1 Introduction

Network Functions Virtualization (NFV) is driving a paradigm shift in telecommunications. NFV allows network functions that have been traditionally implemented as physical appliances in hardware to be implemented as software that runs on virtual machines [9]. Virtual Network Functions (VNF) [5] can be combined to create virtual network services called Service Function Chains (SFC) [7]. SFCs are compositions of multiple VNFs connected on a service topology. The deployment of virtual services on a network requires that it is efficiently embedded in the infrastructure [6, 16].

Vinicius Fulber-Garcia, Eduardo J. Spinosa, and Elias P. Duarte Jr.
Federal University of Paraná, Curitiba, Brazil e-mail: {vfgarcia, spinosa, elias}@inf.ufpr.br

Marcelo C. Luizelli
Federal University of Pampa, Alegrete, Brazil e-mail: marceloluizelli@unipampa.edu.br

Carlos R. Paula dos Santos
Federal University of Santa Maria, Santa Maria, RS, Brazil e-mail: csantos@inf.ufsm.br

Informally, the problem of mapping a network virtualization service consists of defining where the network functions that make up the service will be instantiated and executed. The problem becomes more challenging if the network consists of multiple administrative domains. Different domains may have restrictions on the number of services they run and the resource requirements of the respective functions. In addition, the policies the domain adopts together with business rules adopted by each domain also have an impact on which alternatives are feasible and their costs. Moreover, there are network functions that are native to specific domains, to which they must necessarily be mapped.

In general, there is a choice of where each function should be executed, which depends on the policies and resources available in the domains. Mapping also depends on the topology of the virtualized service and the multi-domain network topology to which it will be mapped. In this case, the objective is typically to reduce the amount of traffic transferred between domains as flows are forwarded through the network service. Furthermore, other criteria can be defined for each particular mapping process, such as maximizing the number of users and maximizing or minimizing the number of domains used to host the service. It should also be taken into account that the mapping objectives usually change according to the very nature of the service being mapped, the type of environment in which they operate, and also the network technologies involved, such as 5G or earlier cellular networks or even IoT or vehicular networks.

Traditional solutions for mapping VNFs are based on evaluation setups that are often static in terms of the set of optimization metrics they employ, as well as objectives and weights, lacking the flexibility required to customize their execution [9, 10, 15]. Typically, those solutions only allow stakeholders to make simple adjustments of the weights of pre-configured optimization metrics [8]. Thus, the requirements of the multiple stakeholders (*i.e.*, clients, providers, and network operators) are hardly met. A static strategy often leaves stakeholders having to adapt their needs to the restrictions of the mapping solutions they are using. The limitations can be critical in multi-domain environments [18, 16, 11, 14]. To the best of our knowledge, no current virtual service mapping solution allows arbitrary optimization metrics and objectives to be defined.

In this work, we propose a new multi-domain mapping solution called Genetic Service Mapping (GeSeMa). GeSeMa allows the evaluation setup to be customized, providing high flexibility to adapt to the different needs of multiple stakeholders and considering several features. To do that, the stakeholders describe their needs and other service features on a standard request document. GeSeMa then uses a multi-objective optimization metaheuristic based on genetic algorithms to find mapping candidates in a feasible time. We evaluate GeSeMa through a case study, including a comparison with a state-of-the-art genetic-based mapping solution [11].

The rest of this work is organized as follows. Section 2 presents related work. GeSeMa is presented in Section 3. Evaluation results are in Section 4, including a case study comparing GeSeMa with a state-of-the-art mapping solution. Finally, Section 5 concludes the paper and presents future work.

2 Related Work

There are often multiple possible mappings of a given virtualized network service on a multi-domain environment. However, the performance of those distinct mappings varies when different policies, constraints, and optimization metrics are employed [9]. Mapping solutions evaluate the multiple alternatives to guarantee, for instance, the QoS (Quality of Service) and QoE (Quality of Experience) of the final results.

Dietrich *et al.* [3] propose a solution that optimizes the multi-domain mapping by relying on four static metrics: (i) minimization of financial costs; (ii) minimization of the number of different providers and domains; (iii) minimization of resource usage; and (iv) maximization of suitability weights. In [13], a multi-domain mapping solution recovers information about financial costs, transmission delays, and resource usage to evaluate and optimize (with a minimization objective) the candidate mappings. Finally, in [16], a multi-domain mapping strategy is proposed that considers hybrid scenarios where private and public domains provide optical network resources. The objective of that solution is to minimize financial costs and the usage of frequency slots of the optical channels connecting the domains.

The solution proposed in [18] consists of a multi-domain mapping technique based on a vertex-centric algorithm. The solution triggers rounds of message exchanges among providers to find candidate mappings iteratively. The mapping algorithm uses a mechanism to avoid the concentration of the entire service on a single provider. However, it does not optimize any specific metric, only returning for the user a set of candidate mappings that fulfill the allocation and instantiation constraints of the requesting service. With a method similar to [18], DistNSE [1] finds candidate mappings and employs a process based on message exchanges among providers. This solution evaluates two optimization metrics: minimization of financial costs and stabilization of inter-domain load.

In [11], a multi-domain mapping technique based on a mono-objective genetic algorithm is proposed. The objective of that solution is to allocate the network functions of a network service chain on a multi-domain environment based on a single indicator (E). This indicator represents multiple domain metrics, such as link availability, bandwidth, and the number of network functions that each domain can host, among others.

The solution proposed in [14] employs a mono-objective genetic algorithm to map virtualized network services on physical substrate nodes. The solution aims to optimize the consumption of computing and networking resources by the network services. In this way, the authors propose an objective function that minimizes the residual capacity of nodes to host functions and links to handle their communication, given the mapped services.

Despite the fact that most of these solutions evaluate multiple optimization metrics, they do not enable stakeholders to customize the evaluation setup (*i.e.*, it is not possible to define/select neither the metrics employed by the optimization process, nor the objectives/weights). This lack of customization makes it difficult to model and evaluate policies that are closely related to the deployment process (*e.g.*, maximum delay, maximum geographical distance). Furthermore, solutions in [13] and

[16] present limitations in terms of the specification of domain dependencies (*i.e.* they do not allow the specification of which functions should be allocated to which particular domains). Thus, for example, these solutions are not suitable to embed hybrid services (*i.e.*, those in which physical network functions coexist with virtualized network functions along a service topology) in multi-domain environments.

3 Genetic Service Mapping

In this section we present GeSeMa (Genetic Service Mapping), a solution that employs genetic algorithms to map virtualized network services across multiple administrative domains. GeSeMa enables stakeholders to define service and network topologies, function and domain dependencies, and the evaluation setup (optimization metrics, objectives, weights, and constraints). This custom information is specified in a request document written in the YAML Ain't Markup Language (YAML).

3.1 GeSeMa's Request Model

GeSeMa's request model presents three main objects that define (i) the service topology and the network functions (**SERVICE**); (ii) the optimization metrics and objectives (**METRICS**); and (iii) the domains and their characteristics (**DOMAINS**). A string specified according to the rules of the Service ChAin Grammar (SCAG) [6] represents the service topology in the **SERVICE** object. Furthermore, for each network function defined in the service topology, there is a corresponding entry in the **FUNCTIONS** sub-object. This entry, identified by the function ID, specifies the minimum resource requirements, including memory, virtualized processing cores, and virtualized network interfaces, all defined as integer values.

The **METRICS** object defines metrics and objectives used by the genetic algorithms of GeSeMa to search, evaluate, and optimize candidate mappings. Metrics are of two categories: *local* or *transition*. Local metrics are used to evaluate the allocation of network functions to domains, which correspond to the vertices of a graph representing the infrastructure on which the service is to be mapped. Local metrics include, for instance, the financial cost to allocate a function, and the domain load, among others. Transition metrics are related to inter-domain connections – which correspond to the edges of the infrastructure graph. Examples of transition metrics include delay, distance in hops, and geographical distance. The metrics and their categories are defined in the request model using **LOCAL** and **TRANSITION** sub-objects, respectively. Each of these sub-objects can define multiple metrics. A metric must be uniquely identified (by its **ID**), besides having two mandatory attributes: **OBJECTIVE** and **CONSTRAINTS**. The objective attribute shows the evaluation criteria for a particular metric, which can be either **MAXIMIZATION** or **MINIMIZATION**. The last attribute

(CONSTRAINTS) consists of a list of strings, each of which refers to the constraints of an optimization metric. Constraints define acceptance thresholds for the evaluation results of optimization metrics. In order to check results with respect to thresholds, relational operators (“<”, “>”, “<=”, “>=”, “==” and “!=”) are employed to compare numerical values with the corresponding thresholds.

Finally, the DOMAINS object defines the physical and virtual environments available and their transitions (connections). The domains are represented by a directed graph $G = (V, E)$. The set of vertices V corresponds to the set of domains, and the set of edges E represents the logical connections between domains. The model keeps the information about LOCAL metrics of each domain (vertex) and TRANSITION metrics associated with the edges. A particular domain is thus defined with three sub-objects: RESOURCES, LOCAL, and TRANSITION. The RESOURCES sub-object contains information about memory (MEMORY), virtual processing cores (VCPU), and virtual network interfaces (IFACES) made available by the domain. The LOCAL and TRANSITION sub-objects, in turn, define the metrics associated with domains and their connections obtained either with benchmarking or from catalogs; this is used by the optimization process. These sub-objects are also related to the METRICS object, and there must be a correspondence between metric identifiers and benchmark identifiers for both the LOCAL and TRANSITION sub-objects. In special, each entry of the TRANSITION sub-object determines to which domain the transition corresponds (using the domain unique identifier) and then defines the values of the optimization metrics for the transition.

3.2 The Proposed Genetic Multi-domain Mapping Method

GeSeMa executes two well-known genetic algorithms: NSGAI [2] and SPEA2 [19]. Those algorithms have been successfully applied to solve networking problems, including fault diagnosis [4, 12]. Note that the system can be extended to include other algorithms. The stakeholders can choose the genetic algorithm taking into account their characteristics, features of the requested service, and the domains, plus the evaluation setup provided. The genetic algorithms model the virtualized service mapping problem as follows:

Individuals: an individual’s chromosome is modeled as a vector with $N > 1$ genes (*i.e.*, positions), where each gene corresponds to a network function of the service topology (*i.e.*, each function is mapped to a position in the vector). Genes contain alleles, represented by integer values in the range $[0, M-1]$ which correspond to the $M > 0$ domains available to map the network functions. Note that, in GeSeMa, a valid individual is a candidate mapping.

Population: the initial population is created randomly or using a greedy-based strategy. The initial population must not violate any function to domain dependencies, if there is any (*i.e.*, for instance, if a domain *must* host some function, the index corresponding to the specific domain is fixed to the allele of the constrained gene). The population size $P > 0$ is a parameter defined by the stakeholders.

Objectives and constraints: GeSeMa evaluates objectives (with the evaluation setup) and constraints (*e.g.*, policies, network topology, computational resources, and dependencies) for all individuals of each generation. We use a taboo list to keep invalid individuals and avoid re-evaluations in case of new occurrences; If it happens, three actions are possible: (i) discard the individual (a standard action); (ii) replace the individual with a new random individual (in case policies or network topology constraints are violated); or (iii) reduce domain redundancy (in case of computational resources constraints are violated).

Selection: the selection chooses individuals of a generation to crossover. GeSeMa uses a tournament mechanism that randomizes I individuals and returns the one that is the most fitted among them (*i.e.*, the one on the best Pareto frontier). The tournament size $I > 1$ is defined by the stakeholders.

Crossover: GeSeMa provides four crossover operators: Simulated Binary Crossover, Half Uniform Crossover, Partially Mapped Crossover, and Subtour Selection Crossover. The crossover operator and ratio (*i.e.*, operator application probability) are also defined by the stakeholders.

Mutation: the proposed solution employs two mutation operators: replacement and swap. Replacement chooses a random gene and replaces its allele with a new random value. Swap chooses two random genes and exchanges their alleles. Genes with domain constraints are never mutated. Similar to crossover, the stakeholders can define the mutation operator and its ratio.

GeSeMa executes two main procedures: (i) validation and configuration of the genetic algorithm; and (ii) creation and evolution of the population. The first procedure uses the model specified in Subsection 3.1 to validate the provided service request, thus mapping high-level structures to iterable elements (*i.e.*, dictionaries, and lists). Next, the procedure checks previously defined genetic parameters (*i.e.*, population size, tournament size, crossover operator/ratio, mutation operator/ratio, and the number of generations) and, if valid, it configures the genetic algorithm. Finally, the first procedure generates a set of software elements employed for the creation and evolution of individuals by the second procedure.

Figure 1 summarizes the second procedure of GeSeMa. At first, the network service, encoded as a string according to the SCAG grammar, is converted to a format that is processed by the genetic algorithms (Figure 1: A and B). The initial population is generated with valid individuals in terms of the network topology (network domain transitions) and domain dependencies (constrained network functions pinned to their respective domains). Next, the individuals are evaluated (Figure 1: C) considering the availability of computational resources in the chosen domains and other constraints. In this way, each candidate is evaluated iteratively, gene by gene for all metrics. Results of all genes are aggregated to define the overall result for each metric. Finally, GeSeMa executes selections (Figure 1: D) in addition to the crossover and mutation genetic operations (Figure 1: E and F, respectively) to evolve the population. All the stages depicted in Figure 1 C, D, E, and F represent the processing done to create a generation of individuals (Figure 1: G). Finally, after each generation has been created, the genetic algorithm saves the best-fitted results (local Pareto frontier) for

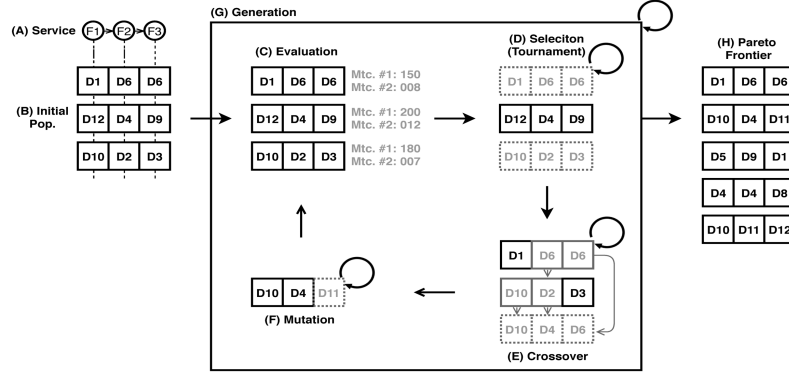


Fig. 1: Summary of the GeSeMa Workflow

reusing in future generations. After a predetermined number of generations, GeSeMa returns the last Pareto frontier found as the final result (Figure 1: H).

In particular, the evaluation stage (Figure 1: C) produces information that is relevant for the next stages. Local optimization metrics are computed with the current gene's allele. Transition optimization metrics, in turn, are processed when a domain transition occurs. The transition metrics use the current gene's allele and the alleles of previously related genes. Besides the allele, for each gene, there is a so-called relation array with indexes of previously related genes (*i.e.*, previous network functions that have a connection with a particular network function in the requested service topology). In this way, linear chromosomes can represent branched service topologies. The set of partial evaluation results (*i.e.*, by gene/allele) are jointly processed, and the individuals are classified in terms of Pareto frontiers.

4 Experimental Evaluation

In this section, we present an empirical evaluation of GeSeMa¹. For the experiments, we employed the topology that corresponds to the Amazon AWS network, consisting of 114 domains [17]. All the experiments were executed 30 times with a confidence level of 95%. Preliminary experiments were run to determine values for the parameters of the genetic algorithms.

GeSeMa is compared with GA+LCB, which is a mapping solution based on a mono-objective genetic algorithm [11]. In addition to the traditional mapping process (mapping the main network functions of a network service), GA+LCB includes a backup mapping mechanism that creates a backup schema for the requested network service. However, as GeSeMa does not create backups, for comparison purposes, GA+LCB is executed to map the main functions, not the

¹ The implementation is available at <https://github.com/ViniGarcia/NFV-FLERAS>

backups. The GA+LCB objective function was configured to maximize the modified domain importance (imp_k from [11]), which consists of the maximization of three metrics – link availability (da_k), bandwidth availability (dc_k), and the availability factor (A_k) – and the minimization of a single metric – inter-domain delay (dd_k). The GA+LCB solution computes this evaluation setup as $E = w1 * nor(da_k) + w2 * nor(dc_k) + w3 * nor(A_k) + w4 * (1 - nor(dd_k))$, where nor indicates a normalization function and w_n the metric weight ($\sum_{n=1}^4 w_n = 1$).

Both GeSeMa and GA+LCB are employed to map a network service with 9 generic network functions. Two restrictions have to be guaranteed by both solutions: the result mapping of network functions should not exceed the computational resource limits of the domains, and no more than two network functions should be mapped to each domain. Furthermore, both solutions were configured to obey both maximum delay and minimum availability constraints. The values for metrics dc_k and A_k are defined randomly in the intervals $[100, 500]$ and $[0.95, 0.99]$, respectively; the value of da_k is 114 for all the domains (the network topology is a complete graph); and the value of dd_k is defined considering the geographical distance between pairs of domains $gd_{k,k+n}$ in the curve $gd_{k,k+n} * (1 - e^{nor(gd_{k,k+n}) * -4}) * 0.05$. As required by GA+LCB, the initial domain and the final domain are specified in the mapping request document.

The genetic parameters of GeSeMa were configured to be as similar as possible to GA+LCB. GA+LCB includes a crossover of half of the population using a personalized algorithm. Thus, we configured GeSeMa with a crossover ratio of 0.5 using the SBX algorithm (SBX has similar behavior to the GA+LCB crossover algorithm). The mutation ratio is set to 0.05, GA+LCB uses a specific, simple mutation algorithm; GeSeMa uses a replacement mutation algorithm. GA+LCB executes a traditional roulette selector; GeSeMa employs a binary tournament selector. GA+LCB creates the initial population based on a k-shortest path algorithm; GeSeMa creates the initial population randomly. GA+LCB uses a self-designed mono-objective genetic algorithm with elitism features; GeSeMa adopts SPEA2. The population size of 50 was the same for both solutions, as well as the execution of 20000 generations. Finally, we removed the parameter weighing of GA+LCB and evaluated the Pareto Frontiers of the returned results for both solutions.

The first experiment compares the quality of the candidates returned by GeSeMa and GA+LCB. We use the mean of the relative Pareto frontiers for the comparison (smaller numbers are better). Figure 2 shows the mean frontiers of candidates returned for two cases: “complete” (frontiers of all candidates from all executions are used to compute the mean value) and “top 10” (frontiers of top ten candidates of all executions are used to compute the mean value). The GA+LCB solution presented a better mean of the relative frontiers in the “complete” case. However, GeSeMa surpasses the GA+LCB results in the “top 10” experiment. This behavior occurs due to the number of candidates returned from GA+LCB at each execution: precisely one. Thus, GA+LCB returns a total of 30 candidates with the best E value achieved in each execution of the solution. GeSeMa, in turn, returns the entire Pareto frontier, which typically contains multiple candidates. In this experiment, GeSeMa provided approximately 49 candidates per execution, from a total of 1463 candidates evalu-

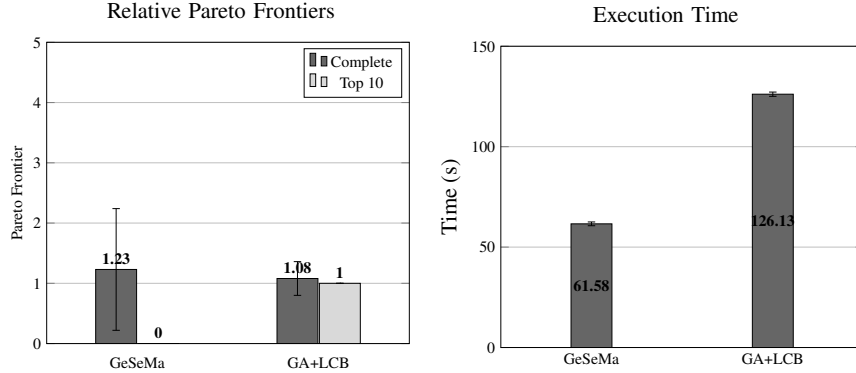


Fig. 2: Frontiers Comparison (Genetic) Fig. 3: Exec. Time Comparison (Genetic)

ated in the “complete” case. Some of these candidates are not better fitted than the ones returned by the GA+LCB, but, as demonstrated by the “top 10” case, the best candidates of GeSeMa are more fitted than the best candidates of GA+LCB.

The second experiment compares the mean execution times of GA+LCB and GeSeMa to map the service in the AWS network topology. Figure 3 shows the results. GeSeMa presented a better mean execution time: 104% faster than GA+LCB. These results can be explained as follows. First, GeSeMa employs a lightweight random initial population strategy, while GA+LCB uses a k-smallest path heuristic to create a possibly more fitted initial population. Thus, the GA+LCB strategy requires the execution of shortest path algorithms that take quite a lengthy amount of time to run in large network topologies. Second, the evaluation of multiple optimization metrics with a mono-objective genetic algorithm requires an aggregated index (in GA+LCB, called E). The creation of this index imposes extra time to process the normalization and weighting required by each generation. Third, GA+LCB does not have any mechanism to avoid the evaluation of candidates which have been already discarded but reappear during the execution of the genetic algorithm. GeSeMa, in turn, uses a taboo list to ignore those candidates.

5 Conclusion

The deployment of virtualized network functions and services depends on proper resource allocation while guaranteeing that restrictions are respected. In this context, multi-domain mapping allows embedding a network service across a distributed environment consisting of multiple administrative domains. Current multi-domain mapping solutions do not enable the stakeholders to customize their evaluation setups. In this paper, we presented Genetic Service Mapping (GeSeMa), an intelligent mapping solution that uses genetic metaheuristics to execute a customizable mapping

of service topologies across multi-domain environments. We evaluated the feasibility and performance of GeSeMa compared with another state-of-the-genetic-based alternative. The results confirm that GeSeMa produced mappings of superior quality with lower execution times.

References

1. Abujoda, A., Papadimitriou, P.: Distnse: Distributed network service embedding across multiple providers. In: International Conference on Communication Systems and Networks, pp. 1–8. IEEE (2016)
2. Deb, K., et al.: A fast and elitist multiobjective genetic algorithm: Nsga-ii. *IEEE transactions on evolutionary computation* **6**(2), 182–197 (2002)
3. Dietrich, D., et al.: Network service embedding across multiple providers with nestor. In: Networking Conference, pp. 1–9. IEEE (2015)
4. Duarte Jr, E.P., Pozo, A.T., Nassu, B.T.: Fault diagnosis of multiprocessor systems based on genetic and estimation of distribution algorithms: a performance evaluation. *International Journal on Artificial Intelligence Tools* **19**(01), 1–18 (2010)
5. Fulber-Garcia, V., et al.: On the design of a flexible architecture for virtualized network function platforms. In: 2019 IEEE Global Communications Conference (GLOBECOM), pp. 1–6. IEEE (2019)
6. Fulber-Garcia, V., et al.: Cusco: A customizable solution for nfv composition. In: International Conference on Advanced Information Networking and Applications, pp. 204–216. Springer (2020)
7. Fulber-Garcia, V., et al.: Network service topology: Formalization, taxonomy and the custom specification model. *Computer Networks* **178**, 107337 (2020)
8. Fulber-Garcia, V., et al.: Customizable deployment of nfv services. *Journal of Network and Systems Management* **29**(3), 1–27 (2021)
9. Herrera, J.G., Botero, J.F.: Resource allocation in nfv: A comprehensive survey. *Transactions on Network and Service Management* **13**(3), 518–532 (2016)
10. Huff, A., et al.: Building multi-domain service function chains based on multiple nfv orchestrators. In: Conference on Network Function Virtualization and Software Defined Networks, pp. 19–24. IEEE (2020)
11. Li, Y., et al.: Cost-and-qos-based nfv service function chain mapping mechanism. In: Network Operations and Management Symposium, pp. 1–9. IEEE (2020)
12. Nassu, B.T., Duarte Jr, E.P., Ramirez Pozo, A.T.: A comparison of evolutionary algorithms for system-level diagnosis. In: Proceedings of the 7th annual conference on Genetic and evolutionary computation, pp. 2053–2060 (2005)
13. Riera, J.F., et al.: Tenor: Steps towards an orchestration platform for multi-pop nfv deployment. In: NetSoft Conference and Workshops, pp. 243–250. IEEE (2016)
14. Rodis, P., Papadimitriou, P.: Intelligent network service embedding using genetic algorithms. In: Symposium on Computers and Communications, pp. 1–7. IEEE (2021)
15. Schardong, F., et al.: Nfv resource allocation: a systematic review and taxonomy of vnf forwarding graph embedding. *Computer Networks* **185**, 107726 (2021)
16. Wang, Y., et al.: Cost-efficient virtual network function graph (vnfg) provisioning in multidomain elastic optical networks. *Journal of Lightwave Technology* **35**(13), 2712–2723 (2017)
17. Wikileaks: Amazon atlas 2015. <https://wikileaks.org/amazon-atlas/> (2018). Accessed in 01 Apr. 2021
18. Zhang, Q., et al.: Vertex-centric computation of service function chains in multi-domain networks. In: NetSoft Conference and Workshops, pp. 211–218. IEEE (2016)
19. Zitzler, E., et al.: Spea2: Improving the strength pareto evolutionary algorithm. *TIK Report* **103** (2001)