Comparison of Embedding Objectives for Next Generation Networks

1st Stelios Prekas
Dept. of Informatics & Computer Engineering
University of West Attika
Egaleo, Greece
Email: sprekas@uniwa.gr

2nd Panagiotis Karkazis
Dept. of Informatics & Computer Engineering
University of West Attika
Egaleo, Greece
Email: p.karkazis@uniwa.gr

3rd Panagiotis Trakadas
General Department
National & Kapodistrian University of Athens
Evia, Greece
Email: ptrakadas@uoa.gr

***Abstract*—Virtualization is considered a key factor for the commercial success of the future Internet and next generation networking in general. But, the plethora of modern day specialized applications migrating to an outsourced hosting environment, introduces new challenges into the process of optimizing the mapping of virtual resources over physical ones. In this paper, we categorize the variety of the current distinctive mapping objectives and performance metrics and we thoroughly compare the most common proposed embedding solutions with different optimization strategies under the same simulation circumstances, evaluating their suitability for serving network applications judging by their experimental results.**

***Keywords*—Virtualization; Next generation networks; Virtual network embedding; Embedding objectives; Performance metrics.**

I. Introduction

Network virtualization stands out as the catalyst technology for the future Internet and a major milestone to its architectural transition [1]–[3]. By its nature, network virtualization enables the co-existence of multiple intangible networks over a shared physical infrastructure. However, network virtualization regardless of having been established as the cornerstone of the modern Internet architecture, it faces a crucial decision of whether transitioning to new and innovative technological concepts or remaining limited by legacy architectures. Nevertheless, the majority of business and mainstream commercial electronic applications already face a transition of their own, dealing with the migration to outsourced virtual hosting infrastructures. Such online cloud based services are known as *Infrastructure as a Service* (IaaS).

IaaS is the delivery of hardware, either servers, storages or networks, and associated software, like operating systems or file systems, as a service [4]. It is the evolution of traditional hosting that does not require any long term commitment regarding in-house investments in data center equipment and other infrastructures and allows corporate administrators or even end users, to provision computational and network resources transparently and on-demand. The architectural philosophy of IaaS also led to the redesign of that market's business model. The role of the Internet Service Provider (ISP) is divided into two new ones; the Infrastructure Provider (InP), who deploys and maintains the physical equipment and the Service Provider (SP) [4], who has the responsibility to provide end-to-end services and applications to the end users. Thus, the appropriate conditions were created for new concepts to emerge like the Next Generation Networks (NGNs), that remodel the known and established architectures by separating and decoupling the underlying technologies and infrastructures from the services that run on top.

An NGN is a packet-based networking environment able to provide digital services including telecommunications, across multiple broadband, Quality of Service (QoS) enabled transport technologies [5]. Inside an NGN, all service-related functions are independent from underlying transport-related technologies. They offer unrestricted access to different SPs and support generalized mobility, which allows consistent and ubiquitous provision of services to end users [6].

The aforementioned massive virtual convergence of specialized applications has also added new levels of complexity to the already complex virtualization concept of Virtual Network Embedding (VNE). VNE, also being independent from the underlying networking technologies, is a fully compatible concept with NGNs, such as the 5G ecosystem and in particular its management and orchestration (MANO) platform. The introduction of these formulations will further allow to combine such technologies with the Software-defined Networking (SDN) and the Network Functions Virtualization (NFV) concepts [6], [7]. Studies regarding VNE, research novel strategies whose main goal is to solve the problem of mapping Virtual Networks (VNs) over substrate resources in an optimal way. This is in fact the gist of the modern virtualization philosophy. Through dynamic allocation of virtual resources onto existing hardware, the benefits gained by all the counterparts from the physical underlay, can be maximized. Even so, optimality is the key to the success of VNE. Arbitrary satisfaction to the demands of incoming service requests, may only lead to poor management of substrate networks and computational assets, thus wasting resources, capital, energy and time.

A substantial number of VNE algorithms have already been researched and developed [8], providing targeted solutions for each arising particular need. Nevertheless, the increasing demand in specialized NGN oriented applications, constantly springs new problem formulations resulting in new embedding objectives [7], [9]. A multitude of customized guaranteed services to the end user can only be made feasible by solutions

Copyright (c) IARIA, 2020. ISBN: 978-1-61208-793-1

that incorporate combinatorial network features, like self-configuration and organization, with regard to different objectives, ranging from guaranteeing the QoS, ensuring economical profit, providing survivability, controlling energy efficiency or establishing network security [10]–[14].

In this paper, we document the variety of distinguished VNE objectives and performance metrics, we thoroughly compare the most common proposed VNE solutions with different optimization strategies under the same simulation circumstances and we investigate their suitability for serving NG applications judging by their experimental results. The remainder of this paper is organized as follows. In Section II, we present the state of the art and the current distinctive VNE objectives. Performance evaluation of the most common VNE algorithms under the same simulation environment are presented in Section III. Finally, Section IV concludes the paper and presents future work.

II. State of the Art

The study of VNE deals with the process of optimally binding demanded resources of the incoming Virtual Network Requests (VNRs) with nodes and links of the Substrate Network (SN). The VNE problem can be formulated mathematically. What is more, by reduction from other computer science optimization problems, like the multicommodity flow or k-multiway separator problems, VNE can be classified as an NP-hard decision problem itself [15].

A. VNE Problem formulation

Several papers have provided similar formulations for VNE [8], [12], [15]–[17]. Taking all into account, the VNE lifecycle can be broken down into the following main phases.

1) ***SN and VNR modeling:*** The common formulation directives typically suggest that the substrate network is denoted by an undirected graph $SN = (N_s, L_s)$ where N_s represents the set of substrate nodes and L_s the set of substrate links. Let $VN^i = (N^i, L^i)$ be a set of $i = 1, ..., n$ VNRs, where N^i and L^i represent the sets of virtual nodes and virtual links of the VN^i, respectively.

2) ***Setting capacities of element parameters:*** In addition, let $\widehat{R} = \prod_{j=1}^{m} R_j$, where $\widehat{R}$ is a vector space of node and link resource vectors, over resource sets $R1, ..., Rm$.
Let $rc : N_s \cup L_s \rightarrow \widehat{R}$ be a function that sets the capacity thresholds of the elements, either nodes or links, of the substrate network. Let $rd : N^i \cup L^i \rightarrow \widehat{R}$ be a function that sets the demand thresholds of the elements, either nodes or links, of all the VNs of the VNRs.

3) ***Establishing networking constraints:*** Depending on the specifications of each applicable scenario, there are some restrictions to be considered before the launch of the mapping process. The most obvious case is that the candidate substrate resources have to be adequate in order to support the demand requirements of the virtual resources being mapped. Exceptional cases may emerge, where redundancy is required, thus even more substrate resources may have to be reserved. Moreover, there are algorithms that are developed with embedded support for the distance constraint. The authors in [16] suggest that the function M_N :

$$\begin{matrix} M_N : N^i \rightarrow N_S \\ \forall (n^V, m^V) \in N^i \end{matrix} \ni \begin{cases} M_N(n^V) \epsilon N_S \\ M_N(m^V) = M_N(n^V) \iff m^V = n^V \end{cases} \quad (1)$$

which implements the mappings of virtual nodes upon substrate ones, should comply with the following constraint:

$$dis\left(loc(n^V), loc(M_N(n^V))\right) \leq D^V \quad (2)$$

where $dis(j, k)$ measures the distance between the location of two nodes j and k and D^V is the non-negative value expressing how far virtual node n^V of location $loc(n^V)$, can be mapped regarding the location $loc(M_N(n^V))$ of the candidate substrate node calculated by $M_N(n^V)$.
Additionally, the concept of the Hidden Hop (HH) property, first introduced in [18] and [19], establishes the portional consumption of the computational resources of those substrate nodes, which forward network packets within an embedded virtual path that aggregates more than one physical link. The importance of this realistic feature is highlighted in our study (see Section III-C *Evaluation results*).

4) ***Launching the desirable mapping strategies:*** A substrate resource is partitioned to host several virtual resources. A single substrate node can host several virtual nodes. In some cases, substrate resources can also be combined to create new virtual resources. This is the case for a virtual link which spans several physical ones, to form a network path in the SN. In this case, a virtual link between two virtual nodes n^V and m^V is mapped to a path in the SN that connects the substrate hosts of n^V and m^V. Each substrate link may then be part of several virtual links. As such, the mapping of virtual links to substrate paths describes a $N : M$ relationship [8].

Furthermore, there is the case where specialized algorithms can be utilized in order to map a single virtual link over several substrate paths, with flexible splitting ratios depending on the solution's constraints and objectives. This is called the path-splitting (PS) technique [12] and provides resource redundancy and demand distribution.

From the studied literature we can derive that the VNE problem can be divided in two interdependent sub-problems. The mapping of the demands of the virtual nodes over the available resources of the physical nodes, and the embedding of the demands of the virtual links interconnecting those virtual nodes, onto the residual network resources of the physical paths.

Taking the above into account, we can adopt the following summarized functions, suggested in [8], as the ones comprising the embedding process.

$$f_i : N^i \rightarrow N_S \quad (3)$$

$$g_i : L^i \rightarrow SP \subseteq SN \quad (4)$$

The function from (3) is used for the mapping of the virtual nodes belonging to N^i, over the physical ones belonging to N_s. And the function from (4) is used for the mapping of the virtual links belonging to L^i, onto a substrate path SP,

Copyright (c) IARIA, 2020.

consisted of physical links belonging to L_s. Together, they form an embedding for each VN^i.

In any case, substrate resources should be spent economically, therefore the mapping procedure has to be optimized. Throughout the VNE oriented literature, every study adopts different objective strategies regarding the individual perspective of the research. Many pursuit the maximization of the revenue metric. The revenue of a VNR can be defined as the weighted sum of its demands. The total revenue is the sum of the revenue of all successfully mapped VNRs. For instance, the authors in [16] and [12] define the revenue of a VNR as follows:

$$\underset{G^V \epsilon VN^i}{\mathbb{R}(G^V)} = \sum_{e^V \epsilon L^i} b(e^V) + \sum_{n^V \epsilon N^i} c(n^V) \quad (5)$$

where $b(e^V)$ and $c(n^V)$ are the bandwidth and CPU requirements for virtual link e^V and virtual node n^V, respectively.

Whereas, in [17] it is stated that the main aim is to decrease the number of congested substrate links, since the authors argue that the substrate nodes do not affect the rejection probability of a VNR. The aforementioned study sets two main objectives; at first, the authors denote the set of α-congested substrate links by defining ζ as follows:

$$\zeta = \{e^s \epsilon L^s : B_{e^s} \leq (1-\alpha) B_{e^s}^{max}\} \quad (6)$$

where $B_{e^s}^{max}$ is the bandwidth capacity of physical link e^s and $0 \leq \alpha \leq 1$. The objective is to reconfigure the accepted VNRs (VN_{AR}^i) within the SN, in the aim of minimising the number of α-congested links.

$$minimize_{VN_{AR}^i}(|\zeta|) \quad (7)$$

Secondly, they formulate the minimization of the reconfiguration cost,

$$minimize_{AR}(\alpha\phi_n + b\phi_e) \quad (8)$$

where $0 \leq \alpha \leq 1$ and $0 \leq b \leq 1$ are the weights of the migrated virtual nodes (ϕ_n) and virtual links (ϕ_e) respectively.

5) ***Outcome evaluation:*** Depending on each individual embedding objective, the mapping strategies can be evaluated by examining the values of their performance metrics (see Section III-B *Performance Metrics*) and comparing them with the results of other VNE solutions, bearing identical or similar mapping objectives. Their evaluation can be achieved by the means of dedicated software tools, that can simulate the networking environment and stage the desired circumstances. In Table I, we present the most distinctive embedding objectives, distinguished among the related literature, as well as the performance metrics that the solutions proposed and tried to optimize. The final column suggests a possible matching between the documented solutions popular network application categories.

B. Variations of embedding objectives

While some approaches generically seek to optimize the common performance objectives, such as acceptance ratio, rejection rate, revenue or embedding cost, others are specially focused on specific aspects of the VNE problem. These distinctive objectives that have been researched throughout the existing literature, are summarized as follows.

1) ***Limit energy consumption:*** Energy consumption has always been a key profit factor for the InPs. Moreover, the rapid pacing of business digitization has led to an immense increase of the energy footprint of the IaaS industry. Although the energy expenses alone are a major concern of the InPs, excess IaaS energy consumption has an environmental impact as well. Nevertheless, the energy consumed by the mapped VNs and their respective applications can be managed and limited to an accepted minimum by specialized VNE algorithms, which can grant the mapping process energy awareness by tidying up the mapped virtual resources and enabling the hibernation of the substrate idle ones [10].

2) ***Ensure security robustness:*** Data integrity and transactional privacy are the two most important ingredients of communication trustworthiness. Additionally, regarding virtualization environments, where multiple VNs are layered over the same physical infrastructure. Throughout related literature,

TABLE I. DISTINCTIVE OBJECTIVES AND METRICS.

References	Algorithms	Performance Objectives	Distinctive Metrics	Candidate Applications
I. Fajjari, N. Aitsaadi, G. Pujolle and H. Zimmermann (2011) [17]	Greedy VN Reconfiguration	-Minimize rejection rate -Reduce cost of reconfiguration	-Rejection rate -Reconfiguration cost	Ad hoc and sensor networks
N. M. K. Chowdhury, M. R. Rahman and R. Boutaba (2009) [16]	(MIP), D-ViNE, R-ViNE	-Increase AR & revenue -Decrease cost	-AR, Revenue & Cost -Node/link utilization	Generic network applications
M. Yu, Y. Yi, J. Rexford and M. Chiang (2008) [12]	Modularized algorithm ("greedy" node mapping and Unsplittable Flow)	-Optimize link utilization -Minimize cost -Maximize revenue	- Cost & Revenue -Splitting Ratio	Fault-sensitive applications
Y. Zhu and M. H. Ammar (2009) [20]	VNA-I & VNA-II	-Minimize both the maximum node stress and maximum link stress	-Node/link stress	Generic network applications
J. Lischka and H. Karl (2009) [21]	VNM algorithm (based on SID)	-Efficient use of the underlying resources	-Revenue-to-Cost Ratio	Generic network applications
L. Gong, Y. Wen, Z. Zhu and T. Lee (2014) [14]	GRC-VNE (heuristic)	-Maximize revenue of the InP -Minimize the cost of the InP	-Revenue -Globall resource capacity	IaaS business software
Botero et al (2012) [10]	VNE-EA (MIP)	-Minimize the inactive substrate links and nodes after a mapping	-Inactive links and nodes -Incurred cost in the SN -The percentage of accepted VNRs	Energy consuming Cloud Data Centers
S. Liu, Z. Cai, H. Xu and M. Xu (2014) [11]	Heuristic algorithm	-Maximize Acceptance ratio -Maximize the long-term Revenue to Cost Ratio	-Acceptance ratio -Long-term revenue -Long-term Revenue to Cost Ratio	Online banking
X. Zhang, C. Phillips and X. Chen (2011) [13]	Integer Linear Program	-Find effective backup paths during the mapping process	-Average mapping delay -Cost of resilience	Multi-layer networks

early attempts have been made to introduce security aware VNE algorithms which involve both resource and security restrictions, by taking into account the specific security requirements of each incoming VNR [11].

*3) **Achieve VN survivability:*** The survivability of a network application can be achieved by means of increasing the resilience of the underlying network, either partially or fully. This means that fallback resources should be calculated during the VNR analysis. These backup resources (nodes, links or both) are bound to the embedded VN until its lifecycle is fulfilled. Fault-sensitive applications require total transparency whenever transitions between the primary and the backup resources take place and vice versa, thus granting a recovery from failure seamless to the application users [12], [13].

*4) **QoS driven embeddings:*** There are situations where VNRs are submitted with QoS-specific requirements. This fact can effectively differentiate the approach a SP should adopt in order to process each VNR. For instance, the given constraints and metrics regarding a VN that provides IP telephony services are weighted and evaluated quite differently in contrast to the ones related to a VN that provides peer-to-peer services [12], [13].

*5) **Maximize economical profit:*** We should not forget that VNE is a real life problem that incorporates the request of a commercial service that involves the leasing of networking and computational resources of a specific or several Infrastructure Providers. One way or another, all the entities involved throughout the commercial chain, spend money for the realization of the mappings of VNs over substrate resources. So, a general approach is to pursuit an objective of maximizing the metric of revenue [14]. This objective is directly proportional to the maximization of the acceptance ratio and, at the same time, the minimization of the embedding cost.

III. Performance evaluation

In this section, we first describe the simulation environment and the methods implemented for comparison reasons. Then, we present the adopted evaluation metrics and the noteworthy evaluation results.

A. Simulation environment

For the realization of our experiments we used the ALEVIN simulation tool [22]. ALEVIN is a Java based framework that focuses on modularity and efficient handling of arbitrary parameters of substrate resources and VNR demand, as well as on supporting the integration of new and existing algorithms and evaluation metrics [23]. A set of algorithms with high impact due to the popularity of their publication among the existing literature, has already been implemented [18]. Each computational experiment has been performed on a x86 architecture computer with an Intel quad core 2,67GHz processor and 8GB of RAM. Although the scale of the undermentioned simulated experiments did not produce the same volume of results as in a real life environment, large scale simulations requiring powerful testbeds, are a goal of future work.

1) Substrate network: By using ALEVIN's scenario generator, we created a physical network, modeled as an undirected graph. The size and scale of the substrate network and the capacities of its resources remain the same for all the simulation scenarios. A meshed topology of 50 nodes has been automatically generated, with a CPU resource capacity of 100 processing units (CPU cycles) for each physical node, and a bandwidth resource capacity of 100 link utilization units (BW units) for every substrate link.

2) Virtual network requests: The current architecture of the ALEVIN framework provides the ability to simulate VNRs in an off-line manner exclusively. Thus, two main simulation approaches have been adopted, regarding the VNRs to be mapped. The first deals with a large number of small scaled VNs and the second with a small number of VNs with greater scale, both with a progressive increase in the demanded CPU and Bandwidth resources. The exact specifications of all the simulation scenarios are presented in Table II.

TABLE II. VALUES OF PHYSICAL AND VIRTUAL PARAMETERS.

	Physical Network			Virtual Network			
Scenario ID	Number of nodes	CPU cycles	BW units	Number of VNRs	nodes per VNR	CPU cycles	BW units
sid1	50	100	100	50	3	15	15
sid2	50	100	100	50	3	30	30
sid3	50	100	100	50	3	30	50
sid4	50	100	100	20	10	15	15
sid5	50	100	100	20	10	30	30
sid6	50	100	100	20	10	50	50

3) Comparison method: In our evaluation, we compared four embedding solutions referenced in [18] and [23], that combine different node mapping and link mapping algorithms. The notations that we used to refer to these different strategies are presented in Table III.

TABLE III. COMPARED EMBEDDING STRATEGIES.

Notation	Embedding Solution Description [Reference]
GreedykSP	Greedy Available and K Shortest Paths [20]
GreedySplit	Greedy Available and Path Splitting [12]
CNLMsplit	Coordinated Node and Link Mapping with Path Splitting [16]
CNLMkSP	Coordinated Node and Link Mapping with k Shortest Paths [16]

B. Performance Metrics

In our experiments we used several performance metrics for evaluation purposes. We measured the average acceptance ratio (AR), the cost-revenue relationship (COSTREV) and the running time (RT), for VNRs of different proportions and with different progressive resource demands, as described in the previous sub-section. Here follows a brief description of the significance of the aforementioned metrics.

*1) **AR:*** describes the number of VNRs that could be completely embedded by the embedding algorithm, divided by the total number of VNRs. A simple conclusion could be that the higher the AR, the more efficient the embedding strategy. But, this can be proven as a rather unidimensional approach.

*2) **COSTREV:*** measures the proportion of cost spent in the substrate network, taking into account the revenue that has been mapped. However, when a set of VNRs has been included in the mapping process, probably the algorithms may not be able to map all of them. In this case, when the percentage of mapped revenue is not too high, the COSTREV metric is not a good indicator of the network mapping quality. That's why this metric can be modified, multiplying it by the percentage of mapped revenue.

*3) **RT:*** the time spent by each algorithm to complete its trials of mapping the entirety of all the incoming VNRs.

C. Evaluation results

In Figures 1 to 4, we present the most important of the produced results of our experiments. In the following paragraphs we analyze and discuss distinctive conclusions of the examined solutions, based on these results.

*1) **Coordinated node and link mapping (CNLM):*** All the studied cases that use the CNLM strategy performed comparatively well regarding the overall number of successfully mapped VNs. Moreover, as shown in Figure 2, the CNLM's embedding efficiency deteriorates in a slower rate than the other mapping strategies, as the VNRs' size and numbers become greater.

On the other hand, the solutions that were using CNLM performed these successful mappings with disproportionally higher cost in RT, as is profound in Figure 1, ranking them the worst for applications where promptly embeddings are crucial. This strategy seems best fitted for applications with a long term life-cycle, expecting high quality results no matter the RT cost and having no need for VN reconfiguration or mapping migration.

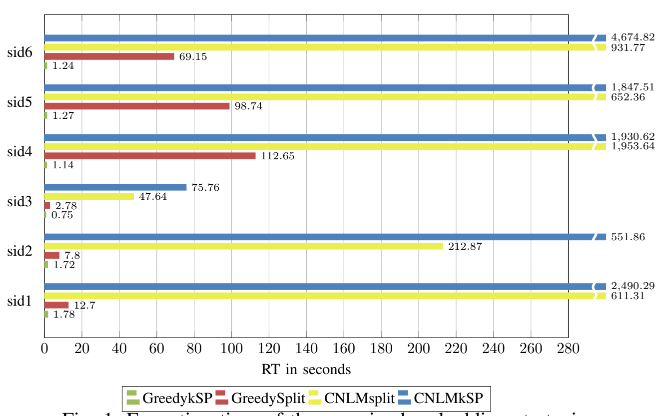

Fig. 1. Execution time of the examined embedding strategies.

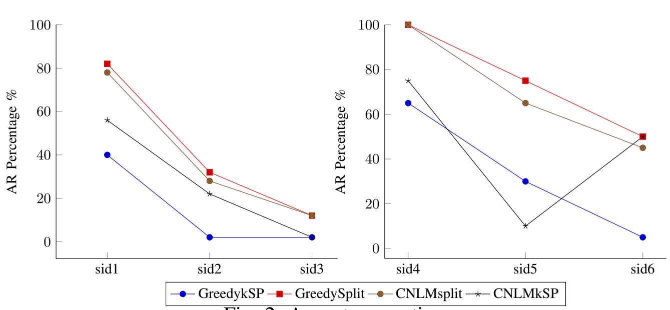

Fig. 2. Acceptance ratio.

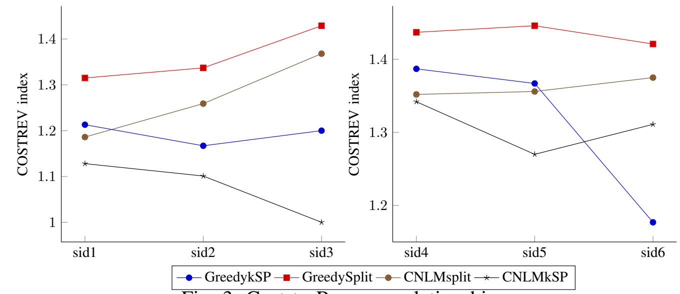

Fig. 3. Cost to Revenue relationship.

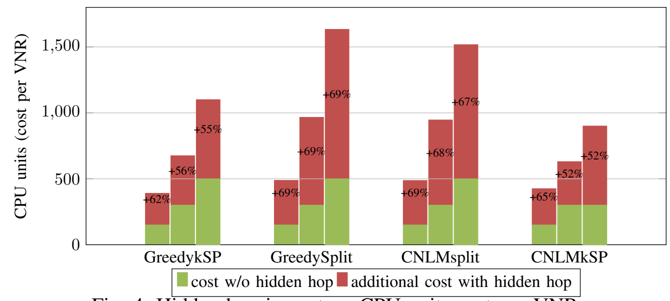

Fig. 4. Hidden hop impact on CPU units cost per VNR.

*2) **Link mapping with path splitting:*** Algorithms using the PS mechanism outperformed all the other studied solution in the field of most mapped VNRs, with a balanced RT. This resulted in fewer rejected VNRs and the algorithms obtained satisfactory AR percentages, as shown in Figure 2. However, due to the use of the PS technique, the cost of the embeddings in every scenario is slightly higher compared to the other solutions with similar ARs, as depicted by the COSTREV index in Figure 3. Moreover, the embedding cost increases even more when the HH property is activated. As depicted in the stacked bar chart of Figure 4, it is easily discerned that while the HH property is taken into account, the solutions that adopt the PS strategy are burdened with a higher cost in substrate CPU units per embedded VNR, almost by 10%, in contrast to the other studied algorithms.

Additionally, the fact that a number of virtual links are mapped over multiple physical paths, enforces the utilization of extra physical computational and networking resources, thus making the technique suitable of applications demanding redundancy and robustness. On the other hand, such utilization of physical resources results in spending increased amounts of operational energy, classifying the solution potentially as the least green among the examined strategies.

*3) **K shortest path link mapping:*** While the RT of the algorithm utilizing the GreedykSP is fast in every scenario as depicted in Figure 1, algorithms adopting the k shortest path link mapping technique perform poorly overall (Figure 2), especially when a large number of VNRs are involved, regardless of being small in size. But in spite of this weakness, the algorithms involving the K shortest path link mapping procedure, produces cheaper embeddings resource-wise, as Figure 3 indicates. Consequently, such approaches should be avoided for fault-sensitive applications and seem more qualified for stateless, express request serving grids like ad-hoc or sensor networks.

IV. Conclusion and future work

Our research studied the most common proposed VNE solutions under progressively increasing demands and within different optimization objective scopes. The goal was to compare novel strategies to the VNE problem, each one distinctive for its optimization approach, and derive the suitability of each solution for any of the key aspects of the VNE problem.

By analyzing the simulation outcomes, it was comprehended that each strategy could yield per case satisfactory results provided that the specifications of the scenario examined tend to be compatible with its optimization objective. However, by further analyzing the produced results of the trials presented in the previous section, it is discerned that although the values of the performance metrics can be reviewed and contemplated about, different network applications will always have varied requirements. All algorithms produce results and values for the same set of metrics and under the conditions of the same parameters at any given scenario, regarding the physical and virtual topologies as well as their respective resources. Some algorithms may excel in certain domains, where others may seem to produce poorer results. Yet still, depending on the prism of another embedding objective, the failure can become the prominent choice. Nevertheless, each proposed VNE solution is developed with predefined objective functions, thus obeying to certain preset constraints and rules. In order to conduct more comprehensive and accurate simulations, all examined solutions should be capable of functioning with other objective functions and not adhere to a certain set of constraints.

Mechanisms and techniques presented by their authors as capable of providing an optimal solution to the VNE problem, lack the architectural structure and ingenuity that could produce optimal solutions, tailored to address a variety of modern day scenarios. This need becomes more timely now that the NGN ecosystem may rely its success entirely on adaptive virtualization architectures. This novel per use case scalability feature, namely a VNE solution matrix, obsoletes the established practice of repeatedly modifying each algorithm entirely in order to meet the optimization objective specifications of a particular network application. Instead, it can provide a variety of metric specific optimal solutions, by even combining a set of metrics into a modular class of service VNE algorithmic toolbox, thus achieving the coveted goal of making available customized end-to-end services to the end users, in an truly optimal manner. This innovative concept will play a pivotal role in our future work.

V. Acknowledgment

This work is partially supported by University of West Attika.

References

[1] K. Tutschku, T. Zinner, A. Nakao, and P. Tran-Gia, "Network virtualization: Implementation steps towards the future internet," Electronic Communications of the EASST, vol. 17, 2009, p. 14. [Online]. Available: http://dx.doi.org/10.14279/tuj.eceasst.17.216

[2] P. Barham et al., "Xen and the art of virtualization," ACM SIGOPS operating systems review, vol. 37, no. 5, 2003, pp. 164–177.

[3] F. Alvarez et al., "An edge-to-cloud virtualized multimedia service platform for 5g networks," IEEE Transactions on Broadcasting, vol. 65, no. 2, 2019, pp. 369–380.

[4] S. Bhardwaj, L. Jain, and S. Jain, "Cloud computing: A study of infrastructure as a service (iaas)," International Journal of engineering and information Technology, vol. 2, no. 1, 2010, pp. 60–63.

[5] I. T. Union, Next Generation Networks – Enhancements to NGN. ITU-T, Sep. 2016, chapter 1, pp. 2, language = english,.

[6] P. Trakadas et al., "Comparison of management and orchestration solutions for the 5g era," Journal of Sensor and Actuator Networks, vol. 9, no. 1, 2020, p. 4.

[7] A. Pol et al., "Advanced nfv features applied to multimedia real-time communications use case," in 2019 IEEE 2nd 5G World Forum (5GWF). IEEE, 2019, pp. 323–328.

[8] A. Fischer, J. F. Botero, M. T. Beck, H. De Meer, and X. Hesselbach, "Virtual network embedding: A survey," IEEE Communications Surveys & Tutorials, vol. 15, no. 4, 2013, pp. 1888–1906.

[9] Y. Shekhawat et al., "Orchestrating live immersive media services over cloud native edge infrastructure," in 2019 IEEE 2nd 5G World Forum (5GWF). IEEE, 2019, pp. 316–322.

[10] J. F. Botero et al., "Energy efficient virtual network embedding," IEEE Communications Letters, vol. 16, no. 5, 2012, pp. 756–759.

[11] S. Liu, Z. Cai, H. Xu, and M. Xu, "Security-aware virtual network embedding," in 2014 IEEE International Conference on Communications (ICC). IEEE, 2014, pp. 834–840.

[12] M. Yu, Y. Yi, J. Rexford, and M. Chiang, "Rethinking virtual network embedding: substrate support for path splitting and migration," ACM SIGCOMM Computer Communication Review, vol. 38, no. 2, 2008, pp. 17–29.

[13] X. Zhang, C. Phillips, and X. Chen, "An overlay mapping model for achieving enhanced qos and resilience performance," in 2011 3rd International Congress on Ultra Modern Telecommunications and Control Systems and Workshops (ICUMT). IEEE, 2011, pp. 1–7.

[14] L. Gong, Y. Wen, Z. Zhu, and T. Lee, "Toward profit-seeking virtual network embedding algorithm via global resource capacity," in IEEE INFOCOM 2014-IEEE Conference on Computer Communications. IEEE, 2014, pp. 1–9.

[15] E. Amaldi, S. Coniglio, A. M. Koster, and M. Tieves, "On the computational complexity of the virtual network embedding problem," Electronic Notes in Discrete Mathematics, vol. 52, 2016, pp. 213–220.

[16] N. M. K. Chowdhury, M. R. Rahman, and R. Boutaba, "Virtual network embedding with coordinated node and link mapping," in IEEE INFOCOM 2009. IEEE, 2009, pp. 783–791.

[17] I. Fajjari, N. Aitsaadi, G. Pujolle, and H. Zimmermann, "Vnr algorithm: A greedy approach for virtual networks reconfigurations," in 2011 IEEE Global Telecommunications Conference-GLOBECOM 2011. IEEE, 2011, pp. 1–6.

[18] J. F. Botero et al., "Flexible vne algorithms analysis using alevin," in Euroview 2011, 2011, in Proc. of the 11th Würzburg Workshop on IP: Joint ITG, ITC, and EuroNF Workshop "Visions of Future Generation Networks".

[19] J. F. Botero, X. Hesselbach, A. Fischer, and H. De Meer, "Optimal mapping of virtual networks with hidden hops," Telecommunication Systems, vol. 51, no. 4, 2012, pp. 273–282.

[20] Y. Zhu and M. H. Ammar, "Algorithms for assigning substrate network resources to virtual network components." in INFOCOM, vol. 1200, no. 2006. Citeseer, 2006, pp. 1–12.

[21] J. Lischka and H. Karl, "A virtual network mapping algorithm based on subgraph isomorphism detection," in Proceedings of the 1st ACM workshop on Virtualized infrastructure systems and architectures, 2009, pp. 81–88.

[22] Alevin2 - tool for the evaluation of algorithms for embedding virtual networks. [Online]. Available: https://sourceforge.net/p/alevin/wiki/home/ [retrieved: May, 2020]

[23] A. Fischer et al., "Alevin-a framework to develop, compare, and analyze virtual network embedding algorithms," in Open-Access-Journal Electronic Communications of the EASST, 2011, pp. 1–12.