

Article



Comprehensive Comparison of VNE Solutions Based on Different Coordination Approaches

Stelios Prekas^{1,*}, Panagiotis Karkazis¹, Vasileios Nikolakakis², and Panagiotis Trakadas²

- ¹ Department of Informatics and Computer Engineering, School of Engineering, University of West Attica, Agiou Spyridonos Street, Egaleo, 12243 Athens, Greece; p.karkazis@uniwa.gr
- ² Department of Ports Management and Shipping, National and Kapodistrian University of Athens, 34400 Psachna, Greece; nikolakakis.vasileios@gmail.com (V.N.); ptrakadas@uoa.gr (P.T.)
- * Correspondence: sprekas@uniwa.gr

Abstract: Virtualization is commonly accepted as the catalyst that would grant the internet the ability to shed the shackles of legacy technologies and evolve to its future self. In order to make this transformation feasible, unavoidable concurrent radical remodeling of all the involved counterparts is required, particularly in the context of automatically and optimally serving application requests by taking advantage of new virtualized environments and their latent capabilities. A considerable number of embedding strategies that efficiently map virtual computational and networking demands over physical resources have already been proposed in the literature, following different strategies. Among these suggested strategies, different coordination proposals have been adopted, in order to solve the node and link mapping functions. In this paper, we study the capabilities of the most popular embedding strategies, based on their coordination categorization. Our study confirms that by invoking a coordination strategy, especially the single stage one, added benefits on multiple performance layers can be achieved. Moreover, we propose a new mixed coordination algorithm, essentially creating a new category in the field of VNE coordination strategies.

Keywords: coordination; embedding objectives; mapping algorithms; network virtualization; performance metrics; virtual network embedding

1. Introduction

Network Virtualization (NV) is predicted to tip the balance of the structural technologies that constitute the current internet working ecosystem toward more intangible architectures [1–3], particularly in the context of 5G networking technologies [4–9]. By its definition, NV enables the abstraction of network and computational resources and, depending on the applied scenario, it either implements a combination of multiple physical networks to a single virtual software-based entity or partitions one physical network into separate heterogeneous virtual ones. However, in order to materialize any of the aforementioned concepts in actual real life scales, the commercial hosting infrastructures transcend legacy limitations and transform into outsourced virtual environments, enabling the provisioning of hosting services called Infrastructure as a Service (IaaS).

IaaS is classified as a cloud commodity that performs transparent and on-demand delivery of hardware and software resources horizontally to corporations and domestic users alike [10,11]. By investing in the IaaS market, which embraces a pay as you use billing model, corporations avoid any long term commitment in the context of creating and sustaining infrastructures with internal means. IaaS also facilitates the administration of any instant provisioned computing set-up, by offering a unified and flexible management over the internet. The business architecture of the virtualization and infrastructure market was shaped accordingly. The infrastructure Provider (InP) and the Service Provider (SP) are the two roles that have been decoupled from the single traditional entity of the Internet Service Provider (ISP) [12].



Citation: Prekas, S.; Karkazis, P.; Nikolakakis, V.; Trakadas, P. Comprehensive Comparison of VNE Solutions Based on Different Coordination Approaches. *Telecom* **2021**, *2*, 390–412. https://doi.org/ 10.3390/telecom2040023

Academic Editor: Alessandro Pozzebon

Received: 15 September 2021 Accepted: 19 October 2021 Published: 25 October 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https:// creativecommons.org/licenses/by/ 4.0/). Two more roles can spring from the SP, thus, establishing a quad layered NV business architecture, as shown in Figure 1. In this version of the model, all processes are performed by the InP, the virtual network provider (VNPs), the virtual network operator (VNO), and the SP. While the group of the SPs are in charge of providing services to the end-users, the VNPs focus on instantiating virtual networks responsible for satisfying requests stemmed from the interacting SPs. On the other hand, the responsibility of the VNOs is to manage the successfully deployed virtual networks. The VNP and VNO roles alleviate the burden of task orchestration from the SPs, presenting the opportunity to focus only on service provisioning [13].



Figure 1. Revised Service Provider business model.

However, the aforementioned radical innovations have also induced chain renovations to other established networking concepts and simultaneously created the necessary space for new ones to emerge. A virtualization mechanism that may prove to become the focal point of internet modernization is Virtual Network Embedding (VNE). The VNE study focuses on innovative solutions that incorporate optimal mapping of Virtual Networks (VNs) and their demands onto physical resources by adhering to specific optimization objectives. Here follows a summary of those distinctive objectives, researched throughout the existing literature.

Limiting energy consumption—Energy consumption has always been a catalyst for an increase in the profit margins of the InPs. Moreover, the race for the fourth industrial revolution and the emergence of immersive ideas, such as functionality and resource disaggregation, encountered in environments, like 5G and beyond [12,14,15], have spawned large scale data-centers with a constantly increasing energy footprint. Consequently, the IaaS industry has to deal with unceasingly increasing energy expenses as well as the obligation to comply with collections of legislation that aim to limit the environmental impact by the InPs' operating cycle. In the context of VNE, the energy consumption of the SN can be managed and restricted to an accepted minimum by mapping the VNs in specific ways. Specialized VNE algorithms can grant the mapping operations energy awareness by rearranging the mapped virtual resources onto the underlying SN and, in the process, idling physical ones [16].

Ensuring security robustness—Data integrity and transactional privacy are two of the most important ingredients of communication trustworthiness. This notion applies to the fullest regarding virtualization environments, where a multitude of VNs can be layered over the same physical medium. Early attempts have been made by researchers in order to introduce security aware VNE algorithms, which would involve both resource and security restrictions, by incorporating specific sets of security rules [17].

Achieving VN survivability—A network application can acquire characteristics of added resilience provided that the underlying physical network adopts survivability

mechanisms, applied either to the entirety or to specific parts of the SN. These mechanisms would take effect during the Virtual Network Request (VNR) analysis and are meant to calculate the appropriate fallback resources in order to achieve the desired resilience. These backup resources are materialized as additional substrate nodes, substrate links or a mixture of both and are assigned to a specific VNR until the end of its lifecycle. In the same sense, fault-sensitive applications also require nonnegotiable transparency while transitioning from the primary to the backup resources and vice versa, thus, rendering the recovery from any kind of failure seamless to the application users [18,19].

Guaranteeing Quality of Service (QoS)—In the case of QoS-specific VNRs, a SP may choose an entirely different approach for handling the submitted request depending on the situated scenario. In this context, a telecommunications service, which is likely susceptible to quality discrepancies, may impose configuration variations and weighted indices regarding the respective constraints and metrics [18,19].

Maximizing economical profit—VNE is a problem that needs to be solved by taking into account its real life proportions as well. A VNR will almost always represent a request for a commercial service, the satisfaction of which would most certainly involve the lease of expensive and finite networking and computational resources for a certain period of time, from a single or several InPs. Under such circumstances, all the links of this commercial chain may end up spending a substantial share of capital in order to materialize the VN mappings required by the VNR. As a result, a popular approach and mainstream VNE goal is the maximization of revenue [20]. This objective is directly proportional to the minimization of the embedding cost.

Arbitrary satisfaction of service requests may only lead to poor management of networking and computational assets, thus, wasting financial resources and temporal and energy capital. The customized dynamic allocation of virtual resources over existing bare metal environments can maximize the benefits gained from the physical underlay by all the counterparts, from the InP to the end user. Nevertheless, the achievement of the desired optimality may be a rather complex endeavor and, at the same time, the key to the success of VNE.

VNE deals with the assignment of the demanded resources of virtual nodes and virtual links that constitute a VN. The Virtual Node Mapping (VNM) and Virtual Link Mapping (VLM) subproblems deal with the allocation of the computational demands over substrate nodes and the synthesis of virtual links into physical paths, respectively. Both are considered mathematically complex [21], and the algorithms used to calculate their solutions can deal with them separately or adopt a form of coordination (see Section 3). Depending on the coordination strategy, the embedding results may vary considerably, and the satisfaction of the optimization objective can be jeopardized.

In this paper, we study the most prominent embedding solutions among the VNE literature, covering the entire range of coordination choices, and we submit them to thorough and targeted simulated experimentation by tweaking all the available functional parameters and capabilities that their developers have endowed them with. The volume of the acquired data provided us with the chance to observe the limits of the studied strategies and to reach to useful conclusions about their actual performance regarding VNE. Furthermore, we propose a new mixed coordination algorithm—essentially creating a new category in the field of VNE coordination strategies.

The rest of the paper is organized in the following manner. In Sections 2 and 3, we present the commonly accepted VNE formulations and the mixture of coordination choices among prominent embedding algorithms respectively. Section 4 illustrates the evaluation of the performance of well-known virtual mapping methodologies under different coordination approaches. Section 5 describes the proposed mixed coordination strategy and the adopted methodology, while Section 6 provides the conclusions reached by this research and sets future milestones.

2. VNE Problem Decomposition and Modeling

2.1. Problem Description

The study of VNE focuses on methodologies that can be proven capable of providing not only successful but also optimal bindings between the computational and networking virtual demands of an incoming VNR and the corresponding substrate resources of the physical underlay. This process is portrayed in Figure 2.



Figure 2. Visualization of VNR mappings.

The computational and networking demands of the incoming VNR#1 and VNR#2 are allocated over the residual resources of the depicted SN. Specifically, virtual nodes "b" of VNR#2 and "a" of VNR#1 share the same host, physical node "N1". Similar is the case for the virtual links connecting virtual nodes "a" and "b" of VNR#1 and VNR#2 respectively. Both are bound to the physical link "L1-2". However, it is feasible to combine different substrate resources to accommodate individual virtual demands. Such is the implementation that we encounter by the formation of the network path, synthesized by the substrate links "L2-3" and "L3-4", in order to connect the already embedded but distant virtual nodes "a" and "c" of VNR#2.

The most commonly established notions from the related literature suggest a uniform model regarding the formulation guidelines of VNE [18,22,23]. Table 1 provides the semantic terminology used in the analysis of the following sections. Consequently, a VNE strategy can be decomposed into the main phases presented below.

| Tabl | le | 1. ` | VNE | dormu | lation | termino | logy. |
|------|----|------|-----|-------|--------|---------|-------|
|------|----|------|-----|-------|--------|---------|-------|

| Formula | Description |
|--|---|
| $SN = (N_s, L_s)$ | SN is a substrate network, consisting of sets of nodes N_s and links L_s . |
| $G^V = (VN^i)$ | G^V represents a VNR as a set of <i>i</i> Virtual Networks. |
| $VN^i = (N^i, L^i)$ | VN ^{<i>i</i>} denotes the <i>i</i> th Virtual Network of an incoming VNR, consisting of nodes N^{i} and links L^{i} |
| $\widehat{R} = \prod_{j=1}^{m} R_j$ | \widehat{R} is the vector space of all the available resources, R_1, \ldots, R_m . |
| $rc: N_s \cup L_s \to \widehat{R}$ | <i>rc</i> is the function that assigns a capacity to an element of the substrate network. |
| $rd^i: N^i \cup L^i \to \widehat{R}$ | rd^i is the function that assigns a demand to an element of VN ^{<i>i</i>} . |
| $f_i: N^i \to N_s$ | f_i is the function that maps a virtual node of VN ^{<i>i</i>} to a substrate node. |
| $g_i: L^i \to \mathrm{SP} \subseteq \mathrm{SN}$ | g_i is the function that maps a virtual link of VN ^{<i>i</i>} to a path (SP) in the substrate network |
| $M(G^V) = (f_i, g_i)$ | <i>M</i> is the generalized function, that maps a VNR denoted by G^V onto the substrate network SN. |

2.1.1. Modeling of the SN and VNR

As described by the majority of the VNE research community, the SN and the VN of a VNR can be modeled either as directed or undirected graphs. The physical network $SN = (N_s, L_s)$ consists of the sets of substrate nodes and links and denoted by N_s and L_s , respectively. In the same sense, the set $VN^i = (N^i, L^i)$ contains i = 1, ..., n VNRs. L^i and N^i are the sets of virtual links and virtual nodes belonging to VN^i .

2.1.2. Configuring the Element Parameters

The existing resources of the computational and networking SN elements can be summarized by a vector space $\hat{R} = \prod_{j=1}^{m} R_j$, which is constituted by the resource sets $R1, \ldots, Rm$. Their capacity thresholds can be defined by the function $rc : N_s \cup L_s \to \hat{R}$. In the same sense, function $rd : N^i \cup L^i \to \hat{R}$ represents the thresholds of the demanded resources of a VN included in a VNR. The most commonly used elements defining the thresholds of node and link resources are CPU cycles and bandwidth (BW) units, respectively.

2.1.3. Constraints Definition

The establishment of a number of scenario-specific restrictions should precede the launch of the mapping process. One universal undisputed constraint is to ensure that the residual substrate resources are going to suffice for the mapping of the computational and networking requirements of the candidate VNR as implemented in [18]. Prior to the deployment of the link mapping phase, the respected algorithm attempts to allocate all the possible substrate paths from a source node to a destination node that is fit for the embedding of a virtual link $l^i \in L^i$, based on the following constraint:

$$\forall l_S \in L_S, \sum_{i=1}^r BW(l^i)_{l_S} \le BW(l_S) \tag{1}$$

As (1) explicitly suggests, the total BW units spent in order to accommodate the demands of *r* virtual links, or parts of them, cannot surpass the total BW capacity of the specified $l_S \in L_S$.

In situations where tailored solutions are expected, algorithms may incorporate additional constraints. Such is the case of the distance constraint that thresholds the maximum permitted hops between a node of the processed VNR and the respective physical one, introduced in [22]. The authors suggest that the function implementing the mappings of virtual nodes upon substrate ones, should comply with the constraint as expressed by the following inequality:

$$dis(loc(m), loc(n)) \le D^V \tag{2}$$

...

Let $m \in N^t$ be a virtual node of the processed VNR, and $n \in N_S$ its candidate physical one. After using dis() to measure the distance between their location, which is calculated by loc(), the authors then delimit its value by the non-negative constraint D^V .

A more combinatorial approach can be adopted, as in [18]. Here, the authors take into consideration not only the residual available computational resources of a physical node $n \in N_S$ but also the remaining BW capacity of its neighboring substrate links, creating the parameter H(n), defined in (3), thus, avoiding falling into the trap of classifying the mapping candidate nodes unidimensionally.

$$H(n) = \operatorname{CPU}(n) \sum_{l \in L_S(n)} BW(l)$$
(3)

While taking into account elements that could attribute a suggested solution with more real life characteristics, the hidden hop (HH) property should also be considered. It establishes the proportional consumption of CPU cycles of an intermediate substrate node

due to its involvement in the forwarding of network packets by being part of an embedded path that aggregates more than one physical link [24,25].

2.1.4. Adopting an Embedding Objective

In real life environments, the physical resources of a substrate infrastructure are finite, and therefore their use is bound to a number of restrictions, such as economical profit thresholds. Thus, the mapping procedure has to be feasible and optimal at the same time. There is no real common ground throughout the VNE-oriented literature, regarding the goals set by the authors in each study. The commonly pursued embedding objectives vary from the maximization of revenue to the avoidance of substrate link congestion. Nevertheless, while some approaches seek to optimize the common performance metrics, other researchers are specifically focused on more intricate aspects of the VNE problem.

2.1.5. Execution of the Embedding Functions

As aforementioned, VNE can be decomposed into two discrete, yet interdependent, subproblems. The computational demands of the virtual nodes should be mapped on the available resources of the physical nodes (Virtual Node Mapping—VNM), whereas the networking demands of the virtual links interconnecting those virtual nodes should be embedded onto the residual resources of the physical paths (Virtual Link Mapping—VLM).

Under specific VNE requirements, such as QoS assurance or VN survivability, a physical resource can be accordingly divided for hosting a number of virtual resources. Furthermore, specifically modified algorithms have been documented with the functionality to split a virtual link and map its fractions over several substrate paths (SPs). This technique is called path-splitting (PS) and is utilized with the purpose to achieve resource redundancy and demand distribution [18,22].

The VNM and VLM embedding subproblems can be summarized in the following two functions as suggested in [26].

$$f_i: N^i \to N_S$$
 (4)

$$g_i: L^i \to \mathrm{SP} \subseteq \mathrm{SN} \tag{5}$$

The function f_i , defined in (4), represents the virtual node embedding, whereas the function g_i , defined in (5), represents the virtual link mapping. The combined solution of both functions form a complete embedding of the *i*th VN $\in G^V$, notated by $M(G^V) = (f_i, g_i)$.

2.1.6. Importance of Coordination

An initial approach is to solve the VNM (Equation (4)) and VLM (Equation (5)) functions separately and isolated from each other [18]. The absence of coordination suggests that the VNM is solved during the first phase, providing its outcome as an input for the solution of the VLM function, which is calculated during the second phase. Contrariwise, the computational steps of VNM can be executed in coordination with the VLM calculations with the goal to acquire an optimized output.

From the experimental results of the studied literature, we can recognize that the adoption of any form of coordination between the node and link mapping stages, enriches the solution space and increases the overall effectiveness of the embedding process, in contrast to the uncoordinated use cases [22]. An alternative form of coordination can be implemented by simultaneously solving the VNM and VLM subproblems during a single stage, as performed in [27].

2.1.7. Outcome Evaluation

A universal method of evaluating the performance of an embedding strategy is to collect and analyze the values of specified performance metrics after the completion of the mapping stages. By doing so, the results of the mapping strategies can be evaluated and the effectiveness of the VNE solutions can be contrasted on the basis of their mapping objectives as performed in [28]. Furthermore, other comparison approaches can be utilized, as in [29], where the successfulness of algorithms that incorporate traversal techniques of the same nature is challenged.

The required evaluation data can be collected by means of either hands-on experimentation in productive or laboratory environments or by utilizing specialized software tools, which can simulate the impact of VNE scenarios applied on networking domains of controlled structure, properties, and scale.

2.1.8. VNE Problem Complexity

There is a multitude of computational problems of various categories. However, the computational complexity theory focuses mainly on decision problems where the answer is rather simple, such as the binary outcome of affirmative or negative responses [30]. An algorithm may be categorized as efficient when it utilizes a polynomial amount of computational resources. If *P* is the class of decision problems that can be solved efficiently, then *NP* could be identified as the class of problems for which there is an algorithm of polynomial time, able to verify whether a given solution is correct or not.

As was previously stated, the VNE problem can be formulated mathematically. Under the prism of this fact, VNE can be reduced to computer science optimization and decision problems known for their complexity, such as the multi-commodity flow (MCF) or kmultiway separator. Therefore, the NP-hardness of VNE is also established [21]. To avoid resulting in unsustainable execution times, VNE implementations usually resort to solving relaxed variants of the original formulations.

3. Related Work-Coordination Analysis

Previous works have contemplated with how coordination plays a pivotal role regarding the success rates of VNE strategies. This section elaborates on the coordination methodologies applied by existing VNE approaches. As stated in Section 2.1.5 the VNE problem is solved if and only if both of its subproblems, VNM and VLM (mathematically represented by the f_i and g_i functions, respectively) are solved. Their respective results can be calculated separately, or alternatively a more coordinated approach could be adopted.

3.1. Uncoordinated VNE

One approach for the VNM/VLM coordination is to solve each sub-problem in an isolated and independent manner. In this case, VNM should be dealt with primarily, in the sense that it provides the input to the VLM's function. This lack of coordination between the VNE subproblems is called uncoordinated VNE, and it was primarily proposed in [18], where the authors' main goal was to maximize the long-term revenue. A greedy algorithm is used to solve VNM by choosing a set of eligible substrate nodes for each virtual node and then assign the computational demands to the prominent one in every case based on its available resources.

The gist of that methodology is to constantly utilize the resources of the most capable substrate node for the satisfaction of the most demanding virtual one. Depending on the functional capabilities of the underlying physical network, the VLM solution can be implemented as either a single path mapping using the k-shortest path algorithm (Greedy-kSP) [31] for increasing values of k or a multiple path mapping, by reduction from the MCF problem [32] that splits the BW demand of a virtual link among several physical paths in the SN (Greedy-PS).

However, the absence of coordination between node and link mapping may result in disproportionally distancing the neighboring virtual nodes onto the substrate topology, thus, increasing the overall cost of the embedded paths and, consequently, resulting in low acceptance ratios, high network latencies and low long-term revenue.

3.2. Coordinated VNE

Other VNE approaches have improved the performance of their solution by applying coordination mechanisms between the two embedding phases, in contrast to their uncoordinated counterparts. Coordinated VNE can be solved either in two separate, but coordinated, stages or in only one.

3.2.1. Two Stages Coordination

An early proposal to illustrate the two stages coordination was by Chowdhury et al. in [22]. The objective of their study was to achieve a lower embedding cost by introducing new constraints, such as the non-negative distance D^V .

After receiving each incoming VNR and before the node mapping stage is commenced, the substrate nodes are clustered into groups of candidate nodes, obeying the location and capacity constraints for every virtual node. Afterward, a set of meta-nodes, one per virtual node, are placed on top of the SN. By connecting each one to their respected neighboring clustered nodes with edges of infinite resources, a temporary augmented substrate graph is constructed.

Then, an algorithm of a Mixed Integer Programming (MIP) formulation is introduced and applied over the aforementioned augmented graph with its main goal being the minimization of the embedding cost. To circumvent the obstacle of the NP completeness of the MIP, its linear programming relaxation is solved by rounding the obtained solution either deterministically (DR) or randomly (RR). After a path between two meta-nodes is chosen for the realization of a successful mapping, the computational demands of the actual virtual nodes represented by those meta-nodes are embedded onto each of the respected neighboring physical nodes of that path. Finally, the VLM function is performed by implementing either the single path or the multi-path technique proposed in [18] and documented above.

Coordination between both stages may prove to be robust, since the MIP formulation used for the virtual node mapping ultimately takes into account the embedding of virtual links existing onto the path between the source and destination meta-nodes. Therefore, the goal of low embedding cost is likely to be achieved.

3.2.2. One Stage Coordination

For the realization of a single stage VNE solution, the simultaneous mapping of virtual links and virtual nodes is presupposed. One good example for this variant is the approach that incorporates the the Sub-graph Isomorphism Detection (SID) algorithm as suggested by J. Lischka and H. Karl in [27]. An SID-based VNE solution was developed, incorporating node and link mapping during the same phase. The authors' main goal was to optimize the revenue to cost ratio by efficiently managing the underlying substrate resources.

The devised algorithm searches for isomorphic subgraphs of VNs in the managed SN that, at the same time, satisfy the demands of the processed VNR. Furthermore, the authors proposed a parameter called the Search Length Limit (SLL), which thresholds the algorithm's search space and, consequently, the length of the formed SPs. The presence of the SLL helps in preventing execution times of unacceptable proportions; however, depending on its restrictiveness, the number of rejected VNRs may be increased substantially. Two variations of the proposed algorithm are derived from their study. The SimpleVNMFlib, whose SLL is statically set to a value of 10 hops, and the AdvancedVNMFlib with an SLL value ranging between 1 and 10 hops.

It is advocated that these algorithms achieve better and faster mappings than the two stage heuristic approach they compare it with and was developed by the authors of [18]. Their results are allegedly more compelling especially regarding large SNs with high resource consumption, which may be harder to map.

3.3. Inter-InP Coordination

For completeness purposes, it should be noted that the entirety of the coordination variants mentioned above consider the VNE as a concept delimited in a single-InP environment (Intra-InP VNE). However, in real life conditions, the mapping of a VNR could become a subject of collaboration among multiple InPs. A level of transparency is expected in order to enable each InP to embed parts of the VN and connect the border nodes of their constituent mappings by using the external links that interconnect those InPs. Therefore, the VNR is divided by the SP into a number of sub-requests, which are eventually mapped across several InPs.

Each partial embedding is performed by standard VNE algorithms and adheres to the specific constraints and policies an individual InP imposes. Such an Inter-InP coordination mechanism is intended to minimize the embedding cost and, under certain circumstances, allocate the economical profit among the InPs accordingly and fairly [33]. Inter-InP coordination is out of the scope of this study.

3.4. Related Work Challenges

The authors of [18], despite utilizing the k shortest path technique during the link mapping phase of their uncoordinated Greedy-kSP solution, did not present any results regarding the performance variations of the aforementioned algorithm while selecting different k values. Moreover, the study was entirely focused on portraying the algorithm's efficiency in the context of the revenue and cost metrics, forsaking the other available performance indices which could have offered a more complete overview of the capabilities of the involved solution.

The work presented in [22] represents a robust attempt to introduce a VNE solution supporting two stages coordination by utilizing a range of performance metrics depicting the algorithms' capabilities over time. Nonetheless, the variations in the incoming VNRs' density and intensity were not addressed in a managed way that could produce usable data leading to useful conclusions, instead adopting a random distribution of VN scale.

Although, the researchers of the single stage coordinated SID solution presented in [27] used an adequate variety of VN sizes between 20 and 40 nodes, the validity of their experimental results relied on only two indices of intensity of the demanded resources. By doing so, they narrowed the value space of received data, stripping their research from possible important conclusions. Furthermore, despite the fact that the evaluation of their results was achieved by utilizing the most prominent of the existing performance metrics, an index that could represent the substrate resource economy, implied by the mapped node distribution methodology, was not considered.

One common factor of overview incompleteness among the contemplated solutions is the lack of a comparison between all the different coordination methodologies. Our research aims to experimentally explore the potential of those solutions, on the basis of their coordination taxonomy, and observe the factual limits of their capabilities.

4. Evaluation of Examined Solutions

In the beginning of this section, we present the optimization objective functions with which the simulations are carried out. Afterward, we describe the setup of the simulated experiments. We continue by presenting the selected evaluation metrics and conclude with the distinctive experimental results.

4.1. Adopted Objective Functions

The targeted objectives of the solutions participating in our simulations are, hereafter, listed and explained, being a key and critical element of the VNE problem. Specifically, the

main interest of the authors of [18] is to maximize the long-term average revenue as given by the following notation:

$$\lim_{T \to \infty} \frac{\sum_{t=0}^{T} R(G^V(t))}{T}$$
(6)

To achieve this goal, it is crucial to embed incoming VNRs efficiently, such that the substrate resources are minimally occupied, hence, the maximization of revenue. For a VNR denoted as G^V , they define its revenue $R(G^V(t))$ at any particular chronological instance t, as:

$$R(G^{V}(t)) = \sum_{l^{i} \in L^{i}} BW(l^{i}) + a \sum_{n^{i} \in N^{i}} CPU(n^{i})$$
(7)

where l^i is a virtual link that belongs to the set L^i , and n^i is a virtual node that belongs to the set N^i . They also introduce the tunable weight α , which allows the InP to maintain a balanced outcome regarding the imminent costs of the given resource parameters, $BW(l^i)$ and $CPU(n^i)$.

The authors of [22] follow a similar approach regarding the principle of their objective. They advocate that the lower the residual capacity of a particular substrate node or link, the higher the value of the objective function. This notion is notated as follows:

$$\begin{array}{l} \text{minimize} \sum_{uv \in L_S} \frac{\alpha_{uv}}{R_{L_S}(u,v)} \sum_{l \in L^i} BW(l) \\ + \sum_{n \in N_S} \frac{\beta_n}{R_{N_S}(n)} \sum_{m \in N^i} CPU(m) \end{array}$$

$$(8)$$

This simplified form of their objective function interprets that, for the given substrate node *n* and physical path *uv*, the respective resources R_{N_s} and R_{L_s} with more residual capacities are preferred over the resources with less residual capacities, thus, achieving higher values of revenue. The parameters α_{uv} and β_n are used as weights in order to control the balancing of the load while embedding a VNR.

However, in [27], the prime goal was to increase the value of the revenue divided by the total mapping cost as a means to express the quality of a VNR mapping.

$$maximize \ R(G^{V})/cost(M(G^{V}))$$
(9)

In order to achieve such a measurement for every embedded VNR, the developed algorithm calculates the costs of the node and link mappings, defined in (10) and (11), respectively. The authors associate the computational and bandwidth demands of the processed virtual networks with the constraints of CPU and BW correspondingly. The calculated costs are, therefore, combined in the constraint cost function defined in (12).

$$cost_1(M(G^V)) = \sum_{n^i \in N^i} CPU(n^i)$$
(10)

$$cost_2(M(G^V)) = \sum_{l^i \in L^i} BW(l^i) \cdot length(g_i)$$
(11)

$$cost(M(G^V)) = \sum_{i=1}^{2} cost_i M(G^V)$$
(12)

The overall revenue of the mapping is denoted as $R(G^V)$ and expressed in (13)

$$R(G^V) = \sum_{n^i \in N^i} \operatorname{CPU}(n^i) + \sum_{l^i \in L^i} BW(l^i)$$
(13)

The ratio in (9) receives values between 1 and 0, where the value of 1 represents an optimal VN embedding and 0 an infeasible one.

4.2. Comparison Methodology

In our evaluation, we compare embedding solutions that represent the most distinguished coordination strategies. Another criterion for our selection was the common ground that renders these solutions comparable—the direct relation among the optimization objectives of their respective algorithms, as explained in Section 4.1. A concise overview of the operation of these algorithms is presented in Sections 3.1 and 3.2. The exact mapping mechanisms and the variations that were involved in our simulated experiments are displayed in Table 2.

Table 2. Examined mapping strategies.

| Notations | Embedding Solutions | Coordination |
|-----------------|---|--------------------------|
| Greedy-kSP | Greedy Node Mapping and k shortest path Link Mapping | Uncoordinated |
| Greedy-PS | Greedy Node Mapping and Path Splitting Link Mapping | Uncoordinated |
| D-ViNE-SP | Coordinated Node and Link Mapping (kSP VLM w. DR) | Coordinated two-stage |
| D-ViNE-PS | Coordinated Node and Link Mapping (PS VLM w. DR) | Coordinated two-stage |
| R-ViNE-SP | Coordinated Node and Link Mapping (kSP VLM w. RR) | Coordinated two-stage |
| R-ViNE-PS | Coordinated Node and Link Mapping (PS VLM w. RR) | Coordinated two-stage |
| SimpleVNMFlib | Subgraph Isomorphism Detection (SID) w. SLL = 10 | Coordinated single-stage |
| AdvancedVNMFlib | Subgraph Isomorphism Detection (SID) w. SLL = {1 to 10} | Coordinated single-stage |

Before orchestrating and commencing the simulation trials, the setup of the supported parameters of the selected algorithmic methodologies has to be agreed. By studying the related literature, it was made clear that, despite the fact each compared solution provided a variety of available tunable parameters, the range of their allowed thresholds was partially explored, depriving the authors of substantial amount of produced data. This experimental data incompleteness may have led to insufficient observations regarding the behavior of each examined algorithm, thus, introducing the possibility of reaching to ambiguous conclusions regarding the efficiency of the studied solutions. This eventuality fueled our conviction that a more thorough comparison venture was required and particularly under the prism of the applicable adopted coordination strategies.

4.3. Simulation Environment

To perform our experimentation trials, we used the ALEVIN simulation tool [34]. The most prominent characteristic of the ALEVIN framework is its extensive modular capabilities. Depending on the examined scenario, a wide range of parameters, either physical or virtual, is available. Moreover, because of its modular nature, new algorithms and metrics can be regularly integrated. Our experimental scenarios were deployed on a baseline mid-range computer platform with an Intel quad core architecture and 8 GB of memory.

One can easily comprehend that the scale of data produced cannot approximate the volume of experimental data a productive, real-life environment could provide. Therefore, the implementation of simulations of greater magnitude that would require testbeds of significant processing power, constitutes a milestone in our forthcoming work. The exact specifications of all the simulation scenarios are presented in Tables 3–5.

4.3.1. Configuring the SN

The scale and size of the SN as well as the capacity of its elements remained the same throughout all the executed scenarios, thus, creating a comparison baseline for our experimental results, as portrayed in Table 3. We automatically created a topology of 50 substrate nodes via ALEVIN's topology generator. The link utilization and processing capacity thresholds were configured to 100 BW and CPU units, respectively.

| | Substrate Network | | | | | |
|-----------------------------------|--------------------|---------------|-------------|--|--|--|
| Scenario ID | Number of Nodes | CPU Cycles | BW Units | | | |
| s - $id_n: n = 1, 2, \dots, 16$ | 50 | 100 | 100 | | | |

Table 3. Values of the physical parameters.

4.3.2. Configuring the VNRs

Two fundamental simulation approaches have been adopted throughout the known literature regarding handling the arrival of VNRs both online and offline. On the one hand, the requests arrive dynamically and may remain in the network for an arbitrary amount of time. By invoking the latter approach, the number and specifications of all candidate VNRs are known in advance, and the algorithm process them at once. Although the online handling of the VNRs is the equivalent of real-world environments, the offline variant is utilized for the purposes of our research, in order to maintain the highest level of control possible over the simulation parameters, thus, increasing the comparability of the studied solutions.

Consequently, two simulation setups were designed, which differ on VNR density and intensity: the first introduces 50 incoming VNRs, with three nodes per VNR (many–small orientation) and the second adopts a smaller number of 20 VNRs, but with a higher density of 10 nodes per VNR (few–large orientation). Both simulation approaches incorporate a progressive increase in CPU and BW demands throughout the experimental iterations (s-id), as observed in Tables 4 and 5.

| | Virtual Network | | | | | | | |
|-------------|-------------------|------------------|---------------|-------------|--|--|--|--|
| Scenario ID | Number of VNRs | Nodes per VNR | CPU Cycles | BW Units | | | | |
| s-id1 | 50 | 3 | 15 | 15 | | | | |
| s-id2 | 50 | 3 | 15 | 30 | | | | |
| s-id3 | 50 | 3 | 15 | 50 | | | | |
| s-id4 | 50 | 3 | 15 | 70 | | | | |
| s-id5 | 50 | 3 | 30 | 15 | | | | |
| s-id6 | 50 | 3 | 30 | 30 | | | | |
| s-id7 | 50 | 3 | 30 | 50 | | | | |
| s-id8 | 50 | 3 | 30 | 70 | | | | |
| s-id9 | 50 | 3 | 50 | 15 | | | | |
| s-id10 | 50 | 3 | 50 | 30 | | | | |
| s-id11 | 50 | 3 | 50 | 50 | | | | |
| s-id12 | 50 | 3 | 50 | 70 | | | | |
| s-id13 | 50 | 3 | 70 | 15 | | | | |
| s-id14 | 50 | 3 | 70 | 30 | | | | |
| s-id15 | 50 | 3 | 70 | 50 | | | | |
| s-id16 | 50 | 3 | 70 | 70 | | | | |

 Table 4. The many–small parameter configuration.

| | Virtual Network | | | | | | | |
|-------------|-------------------|------------------|---------------|-------------|--|--|--|--|
| Scenario ID | Number of VNRs | Nodes per VNR | CPU Cycles | BW Units | | | | |
| s-id1 | 20 | 10 | 15 | 15 | | | | |
| s-id2 | 20 | 10 | 15 | 30 | | | | |
| s-id3 | 20 | 10 | 15 | 50 | | | | |
| s-id4 | 20 | 10 | 15 | 70 | | | | |
| s-id5 | 20 | 10 | 30 | 15 | | | | |
| s-id6 | 20 | 10 | 30 | 30 | | | | |
| s-id7 | 20 | 10 | 30 | 50 | | | | |
| s-id8 | 20 | 10 | 30 | 70 | | | | |
| s-id9 | 20 | 10 | 50 | 15 | | | | |
| s-id10 | 20 | 10 | 50 | 30 | | | | |
| s-id11 | 20 | 10 | 50 | 50 | | | | |
| s-id12 | 20 | 10 | 50 | 70 | | | | |
| s-id13 | 20 | 10 | 70 | 15 | | | | |
| s-id14 | 20 | 10 | 70 | 30 | | | | |
| s-id15 | 20 | 10 | 70 | 50 | | | | |
| s-id16 | 20 | 10 | 70 | 70 | | | | |

 Table 5. The few-large parameter configuration.

4.4. Performance Metrics

For the evaluation of our experimental results, we used the following performance metrics. The acceptance ratio (AR) calculates the successfulness of an embedding solution by dividing the number of completely mapped VNRs by the total number of processed VNRs. To avoid any overly simplifications, we should mention that a high AR value by itself does not necessarily translate to an absolutely efficient embedding strategy. The cost to revenue measurement (COSTREV) represents the relationship between the cost of SN resources spent and the revenue grossed after the completion of the mappings. However, the higher the number of the rejected VNRs the lower the eventual cost may seem to be, thus, determining a respectively low COSTREV value, which would fail to depict the real successfulness of the embedding algorithm.

In this context, the COSTREV metric would prove to be a lesser quality indicator of the executed mappings. A modified version of COSTREV could be the CR*MR metric [24], which multiplies the former by the mapped revenue ratio. By dividing the number of the mapped nodes of the successfully embedded VNRs to the total number of the residual nodes of a SN that actively serve those VNRs, we can calculate the active node stress metric (ANSTRESS). The running time (RT) is the time spent by each embedding process in order to achieve a complete and successful VNR mapping.

4.5. Evaluation Results

Our simulation data and experimental results were evaluated and classified based on specific directions as follows.

4.5.1. Variations on CPU and BW Demands

Acceptance Ratio (AR)—While the examined scenarios shift between progressively increased CPU and BW demands, the studied solutions exhibit different variations in

success rates. More specifically, the single stage coordination algorithms exceed all the rest in successfully accepted incoming VNRs, as Figure 3a illustrates. They achieve such a performance by also maintaining a stable and relatively reasonable decrease in AR percentages as the intensity of the demand in CPU and BW resources steadily increase. On the contrary, the two stages coordination and the uncoordinated techniques provide less stable acceptance results as the demands intensify, especially those variants that employ the k-shortest path link mapping algorithm. The two stages coordination, however, outperforms the latter in every examined scenario.



Figure 3. Success rates of the examined embedding strategies.

Equivalent performance behavior is also observed, regarding the few–large experimental orientation with prospectively lower AR percentages, as depicted in Figure 3b. Nevertheless, the success rates of the uncoordinated solutions collapse dramatically faster in contrast to their coordinated counterparts, proving that denser VNRs pose as a heavier task for that category of VNE algorithms.

Running Time (RT)—The solutions that incorporate the SID method obtain successful results in undeniably lower execution times in comparison to the other strategies. By the RT graphs, depicted in Figure 4, it is made clear that the SID-based solutions are faster in either the many–small and the few–large orientations. Even so, the AdvancedVNMFlib demands greater RT values regarding the more demanding VNRs due to the variable nature of the SLL parameter in contrast to the SimpleVNMFlib (see Section 3.2.2). The backtracking process of the AdvancedVNMFlib is a time consuming mechanism and, for denser demands, provides no better results than the faster SimpleVNMFlib.

The greatest toll on execution time was taken by those strategies utilizing the two stages coordination node and link mapping technique. As the RT graphs of Figure 4 illustrate, the time needed for the two-stage solutions to perform is many times greater than the other opposed ones, exhibiting high complexity in the VNM and VLM calculation steps of this coordination category.

The uncoordinated solutions appear to achieve equivalent execution times to their single stage counterparts. However, they do so by lacking in AR percentages as was aforementioned.



Figure 4. Execution times of the examined embedding strategies.

Cost/Revenue multiplied by Mapped Revenue (CR*MR)—The SimpleVNMFlib implementation that incorporates a 10-hop static SLL value and invokes inflexibly and constantly longer network paths in comparison to the AdvancedVNMFlib alternative, which supports values from 1 to 10. Furthermore, while the two stages coordination and uncoordinated strategies appear to perform more efficiently over SimpleVNMFlib regarding the CR*MR index, AdvancedVNMFlib achieves embeddings that are noticeably of lower cost during both scenario orientations.

In addition, as depicted in Figure 5, the cost of the embeddings that D-ViNE-PS and Greedy-PS achieve, is lower in comparison to D-ViNE-SP and Greedy-kSP, respectively, in almost every scenario due to the use the multipath approach and the flexibility it provides.

Active Node Stress (ANSTRESS)—The two stages coordination and the uncoordinated orientations tend to provide an even distribution of mapped resources, because of their adopted methodology. As a result, they present lighter physical node stresses overall (many–small scenarios as shown in Figure 6).



Figure 5. Provisioning costs of the examined strategies.



Figure 6. ANS of the examined embedding strategies.

Conversely, the single stage solutions, which are more revenue driven, tend to utilize the more suitable rather than the more capable among the candidate substrate nodes. This differentiation tends to fade rapidly for the few–large orientation scenarios, where the capability of maintaining the number of Active Nodes (AN) to a minimum diminishes due to the increased density of the incoming VNRs, as is profound in Figure 7.



Figure 7. Active nodes of the examined embedding strategies.

4.5.2. Variations on the "k" Parameter

As described in Section 4.2 any possible latent potential the experimentation on the variation of other parameters could reserve was not thoroughly explored. Such is the case of the "k" variable regarding the utilization of the k shortest path technique in [18].

Acceptance Ratio (AR)—Figure 8 clarifies that, by relaxing the demand of exclusively using the calculated shortest path for the interconnection of every mapped pair of virtual nodes on the SN, one can achieve better AR percentages. Nevertheless, the range of the successfulness of the preference in longer network paths is dependent on the scale and density

of the incoming VNRs. As far as the many–small scenarios are concerned, the improvement on the AR percentages is limited to the initial "k" values, after which no positive impact is observed whatsoever. On the other hand, the increment in the success rates in the case of the few–large simulations is more distinctive and ongoing as the "k" values advance.

Running Time (RT)—Although greater "k" values provide more successful embeddings, they also inflict significant delays in execution time, more severely so in the case of the denser VNRs. This impact on RT is depicted in Figure 9.



Figure 8. Success rate of "k" variations on Greedy-kSP.



Figure 9. RT of the "k" variations on Greedy-kSP.

The two stages solutions were developed with the functionality of choosing between a deterministic and a random way of rounding the solution of the linear program to approximate the values of the binary variables in the original MIP (see Section 3.2.1). However, the authors did not provide any experimental results regarding the interchange between those methods.

By observing the graphs of Figure 10, it can easily be discerned that almost all randomized trials completed faster compared to the deterministic ones, especially for the less demanding scenarios, meaning that less computational resources are needed to calculate the rounded solution of the MIP's linear programming relaxation stochastically rather than deterministically.



Figure 10. RT of the rounding techniques of R-ViNE-PS.

Both approaches achieved equivalent embedding success with more or less the same cost indices, proving both rounded solutions as equally efficient (Figure 11). Moreover, in most of the less demanding scenario cases (Figure 12a) of the two stages coordination algorithms that utilize the PS technique (see Section 2.1.5) the deterministic approach occupied less substrate nodes under the same or equivalent acceptance circumstances (Figure 11a). Under the light of these discoveries, one can assume that the deterministic approach is able to achieve a greener footprint in contrast to its random counterpart.



Figure 11. Success of the rounding techniques of R-ViNE-PS.



Figure 12. ANs of the rounding techniques of R-ViNE-PS.

5. Mixed Coordination Algorithm Proposal

The above results gave us the stimulus to propose a new and innovative algorithm that has not been proposed before in the known literature to our knowledge. The algorithm essentially creates a new category in the field of VNE coordination strategies. This mixed stage coordination solution combines the benefits of both the single and two stage candidates. The choice among the examined strategies was made according to their performance during the presented experimental procedure. The R-ViNE-PS was chosen as the two-stage representative because it performed equivalently to its deterministic counterpart regarding the AR and ANS metrics as Figures 11 and 12 illustrate, respectively, but with more competitive RTs (Figure 10).

As far as the single stage solution is concerned, the AdvancedVNMFlib Algorithm 1 was chosen, due to the fact that it clearly outperformed the SimpleVNMFlib variant during the CR*MR index comparison (Figure 5), achieving embeddings of lower cost with relation to the respected mapped revenue ratio.

Algorithm 1 Mixed-stage algorithm concept.

```
\begin{array}{l} \mathrm{SN} \leftarrow (N_s, L_s) \\ \mathrm{VNR}[j] \leftarrow G_j^V(\mathrm{VN}^i), \mathrm{VN}^i = (N^i, L^i) \\ \mathrm{VNR}'[j] \leftarrow \mathrm{Greedy\_sort}(\mathrm{VNR}[j]) \\ \mathrm{SN}' \leftarrow \mathrm{SN} \\ \mathbf{while} \ IS\_EMPTY(\mathrm{VNR}'[]) \neq TRUE \ \mathbf{do} \\ \mathrm{VNR\_proc} \leftarrow \mathrm{VNR}'[\mathrm{x}] \\ \mathbf{if} \ \mathrm{RIL\_calc}(\mathrm{VNR\_proc}) < 0.04 \ \mathbf{then} \\ \mathrm{SN}' \leftarrow R \text{-} ViNE \text{-} PS(\mathrm{SN}', \mathrm{VNR\_proc}) \\ \mathbf{else} \\ \mathrm{SN}' \leftarrow \mathrm{Advanced} \mathrm{VNMFlib}(\mathrm{SN}', \mathrm{VNR\_proc}) \\ \mathbf{end} \ \mathbf{if} \\ \mathbf{x} \leftarrow \mathbf{x} + 1 \\ \mathbf{end} \ \mathbf{while} \\ \mathbf{return} \ \mathrm{SN}' \end{array}
```

The algorithm begins by initializing the substrate network SN with the sets of physical nodes N_s and links L_s . By populating the VNR object, it is established that the number and specifications of the incoming VNRs, denoted as $G_j^V(VN^i)$, are predetermined and already known, creating an offline VNE environment. In this context, index *j* represents the number of the incoming VNRs, while index *i* indicates the scale of the VNs that constitute those

VNRs. Afterward, each VNR is stored in the VNR_proc object and processed according to the following strategy. The initial VNRs are mapped by the embedding procedures of R-ViNE-PS, until the Relative Intensity Index (RII), calculated by the RII_calc() function, surpasses the value of 0.04. The RII is calculated for every scenario by Equations (14)–(17):

$$RII_i = \frac{AAI_i}{\sum_{i=1}^n AAI_i}$$
(14)

$$AAI_i = DI_i \times TD_i \tag{15}$$

$$DI_i = \frac{\text{Nodes per VNR}}{\text{Number of VNRs}}$$
(16)

 $TD_i = [(\text{Number of nodes}) \times \text{CPU}] + [(\text{Number of links}) \times BW)]$ (17)

where *AAI* is the Absolute Intensity Index of each scenario *i*, *DI* is the Density Index of the respected trial, and *TD* represents the Total Demand of the incoming VNRs during an experimental iteration.

According to our experimental results and as Tables 6 and 7 portray, the performance of the two-stage algorithm deteriorates after the RII passes the aforementioned threshold of 0.04. As the embedding process progresses, our algorithm begins to utilize the procedures of AdvancedVNMFlib for the realization of the embeddings. Finally, when all the VNRs have been successfully allocated or the SN resources have proven to be insufficient, the algorithm stops and returns the set of the successful embeddings SN'.

This novel solution produced promising results and performed satisfactorily under any circumstance of incoming VNRs, regardless of density and intensity. While the values of the RII index allow it, the algorithm incorporates the additional distance constraint (see Equation (2), Section 2.1.3 which thresholds the maximum permitted hops between virtual nodes and the respective physical mapping candidates. By doing so, shorter substrate paths are utilized, and less networking and computational resources are occupied; thus, lower costs are achieved. However, when the critical threshold is reached, the algorithm shifts to a more agile mapping strategy, adopting longer paths across the SN and maximizing the mapped revenue.

Table 6. RII of R-ViNE-PS in the many-small parameter configuration.

| | Virtual Network | | | | | | | | |
|-------------|-------------------|------------------|---------------|-------------|--------------------|-----------------|------------------|-----------------------------|-----------------------------|
| Scenario ID | Number of VNRs | Nodes per VNR | CPU Cycles | BW Units | Number of Links | Total Demand | Density Index | Absolute Intensity Index | Relative Intensity Index |
| s-id1 | 50 | 3 | 15 | 15 | 166 | 4740 | 0.06 | 284.4 | 0.0227 |
| s-id2 | 50 | 3 | 15 | 30 | 166 | 7230 | 0.06 | 433.8 | 0.0347 |
| s-id3 | 50 | 3 | 15 | 50 | 166 | 10,550 | 0.06 | 633 | 0.0506 |
| s-id4 | 50 | 3 | 15 | 70 | 166 | 13,870 | 0.06 | 832.2 | 0.0665 |
| s-id5 | 50 | 3 | 30 | 15 | 166 | 6990 | 0.06 | 419.4 | 0.0335 |
| s-id6 | 50 | 3 | 30 | 30 | 166 | 9480 | 0.06 | 568.8 | 0.0455 |
| s-id7 | 50 | 3 | 30 | 50 | 166 | 12,800 | 0.06 | 768 | 0.0614 |
| s-id8 | 50 | 3 | 30 | 70 | 166 | 16,120 | 0.06 | 967.2 | 0.0773 |
| s-id9 | 50 | 3 | 50 | 15 | 166 | 9990 | 0.06 | 599.4 | 0.0479 |
| s-id10 | 50 | 3 | 50 | 30 | 166 | 12,480 | 0.06 | 748.8 | 0.0598 |
| s-id11 | 50 | 3 | 50 | 50 | 166 | 15,800 | 0.06 | 948 | 0.0758 |
| s-id12 | 50 | 3 | 50 | 70 | 166 | 19,120 | 0.06 | 1147.2 | 0.0917 |
| s-id13 | 50 | 3 | 70 | 15 | 166 | 12,990 | 0.06 | 779.4 | 0.0623 |
| s-id14 | 50 | 3 | 70 | 30 | 166 | 15,480 | 0.06 | 928.8 | 0.0742 |
| s-id15 | 50 | 3 | 70 | 50 | 166 | 18,800 | 0.06 | 1128 | 0.0901 |
| s-id16 | 50 | 3 | 70 | 70 | 166 | 22,120 | 0.06 | 1327.2 | 0.1061 |

| | Virtual Network | | | | | | | | |
|-------------|-------------------|------------------|---------------|-------------|--------------------|-----------------|------------------|-----------------------------|-----------------------------|
| Scenario ID | Number of VNRs | Nodes per VNR | CPU Cycles | BW Units | Number of Links | Total Demand | Density Index | Absolute Intensity Index | Relative Intensity Index |
| s-id1 | 20 | 10 | 15 | 15 | 166 | 5490 | 0.5 | 2745 | 0.0227 |
| s-id2 | 20 | 10 | 15 | 30 | 166 | 7980 | 0.5 | 3990 | 0.0330 |
| s-id3 | 20 | 10 | 15 | 50 | 166 | 11,300 | 0.5 | 5650 | 0.0468 |
| s-id4 | 20 | 10 | 15 | 70 | 166 | 14,620 | 0.5 | 7310 | 0.0605 |
| s-id5 | 20 | 10 | 30 | 15 | 166 | 8490 | 0.5 | 4245 | 0.0351 |
| s-id6 | 20 | 10 | 30 | 30 | 166 | 10,980 | 0.5 | 5490 | 0.0455 |
| s-id7 | 20 | 10 | 30 | 50 | 166 | 14,300 | 0.5 | 7150 | 0.0592 |
| s-id8 | 20 | 10 | 30 | 70 | 166 | 17,620 | 0.5 | 8810 | 0.0729 |
| s-id9 | 20 | 10 | 50 | 15 | 166 | 12,490 | 0.5 | 6245 | 0.0517 |
| s-id10 | 20 | 10 | 50 | 30 | 166 | 14,980 | 0.5 | 7490 | 0.0620 |
| s-id11 | 20 | 10 | 50 | 50 | 166 | 18,300 | 0.5 | 9150 | 0.0758 |
| s-id12 | 20 | 10 | 50 | 70 | 166 | 21,620 | 0.5 | 10,810 | 0.0895 |
| s-id13 | 20 | 10 | 70 | 15 | 166 | 16,490 | 0.5 | 8245 | 0.0683 |
| s-id14 | 20 | 10 | 70 | 30 | 166 | 18,980 | 0.5 | 9490 | 0.0786 |
| s-id15 | 20 | 10 | 70 | 50 | 166 | 22,300 | 0.5 | 11,150 | 0.0923 |
| s-id16 | 20 | 10 | 70 | 70 | 166 | 25,620 | 0.5 | 12,810 | 0.1061 |

Table 7. RII of R-ViNE-PS in the few-large parameter configuration.

6. Conclusions and Future Work

Our research studied the most common proposed VNE solutions with different coordination proposals for solving the VNM and VLM functions and experimentally tested their capabilities. The goal was to exploit their potential in the context of their coordination classification by undergoing exhaustive and thorough managed simulation trials and, at the same time, invoke the variety of their available tunable parameters. By analyzing the abundant acquired simulation data and by utilizing the majority of the available performance indices, we reached the following conclusions.

The single stage coordination methods provided more robust outcomes overall, contrary to the two stages and uncoordinated approaches, classifying them as the most suitable for fault-sensitive applications. The trend of the SID-based algorithms (AdvancedVNM-Flib and SimpleVNMFlib) gave them a minor embedding success precedence over the two stage contestants. Despite that fact, the former achieved such successfulness in both outright short running times and within stable and consistent RT thresholds without being affected by the demand intensity of the incoming VNRs. Whereas, the uncoordinated counterparts were unable to handle the progressive increment of the demanded resources despite their competitive running times, posing as more qualified for stateless, express serving configurations, such as ad-hoc or sensor networks.

The evaluation of the ANSTRESS measurements, which can literally be translated into an average provisioning cost index, revealed that the two stage coordination representatives produced more balanced results in contrast to their single stage competitors, which are restricted by the SLL limitations in order to perform competitively. The sparser the node distribution was, the more robust the embedded VNs were and the less prone to general loss of service.

An alternative interpretation could suggest that the uncoordinated and two stages coordination solutions occupy an excessive number of physical nodes regardless of the variation step and the gradual increase in the resource demands, whereas the single stage solutions restrict the reservation of substrate computational entities, when it is feasible. Thus, while the single stage proposals lack a resource distribution strategy according to the previous analysis, they could achieve a greener embedding footprint and lower operating costs for the InPs, especially for those VNRs of lesser demands in processing power.

Needless to say, judging by the abundance of our simulation results and the validity of our concluding points, our efforts appear more than warranted. This study attributed added value to the most prominent VNE strategies based on their coordination classification, an approach not encountered in the VNE-oriented literature to the best of our knowledge. It produced new measurable results for unexplored territory with respect to the studied established VNE strategies and regarding the key points addressed in Section 3.4. Moreover, our proposed algorithm offers another baseline of comparison between VNE solutions as well as the beginning of a new class of mixed coordination strategies that reap the benefits of more than one coordination category.

Nevertheless, this research should be furthered, and the next milestones are still to be set. The development of a contestant to the two stages coordination solution based on its unique non-negative distance attribute (see Section 3.2.1) is a goal for future work. Furthermore, a research for a novel VLM driven solution, integrated with the routing algebra functionality and enriched with isotonicity and monotonicity is the key to extending the scope of our study.

Author Contributions: Conceptualization, S.P., P.K. and P.T.; methodology, P.K. and P.T.; software, S.P., V.N. and P.T.; validation, S.P., P.K. and P.T.; investigation, S.P.; resources, P.K. and P.T.; data curation, S.P. and V.N.; writing—original draft preparation, S.P.; writing—review and editing, S.P., P.K. and P.T.; supervision, P.K. and P.T. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Informed consent was obtained from all subjects involved in the study.

Data Availability Statement: Comprehensive Comparison of VNE Solutions Based on Different Coordination Approaches. IEEE Dataport. https://dx.doi.org/10.21227/ft0t-tk36.

Acknowledgments: This work is partially supported by University of West Attika.

Conflicts of Interest: The authors declare no conflict of interest.

References

- Tutschku, K.; Zinner, T.; Nakao, A.; Tran-Gia, P. Network virtualization: Implementation steps towards the future internet. Electron. Commun. EASST 2009, 17, 1–14. [CrossRef]
- 2. Barham, P.; Dragovic, B.; Fraser, K.; Hand, S.; Harris, T.; Ho, A.; Neugebauer, R.; Pratt, I.; Warfield, A. Xen and the Art of Virtualization. *SIGOPS Oper. Syst. Rev.* **2003**, *37*, 164–177. [CrossRef]
- 3. Alvarez, F.; Breitgand, D.; Griffin, D.; Andriani, P.; Rizou, S.; Zioulis, N.; Moscatelli, F.; Serrano, J.; Keltsch, M.; Trakadas, P.; et al. An Edge-to-Cloud Virtualized Multimedia Service Platform for 5G Networks. *IEEE Trans. Broadcast.* **2019**, *65*, 369–380. [CrossRef]
- 4. Trakadas, P.; Karkazis, P.; Leligou, H.C.; Zahariadis, T.; Vicens, F.; Zurita, A.; Alemany, P.; Soenen, T.; Parada, C.; Bonnet, J. Comparison of management and orchestration solutions for the 5G era. *J. Sens. Actuator Netw.* **2020**, *9*, 4. [CrossRef]
- Peuster, M.; Schneider, S.; Zhao, M.; Xilouris, G.; Trakadas, P.; Vicens, F.; Tavernier, W.; Soenen, T.; Vilalta, R.; Andreou, G.; et al. Introducing Automated Verification and Validation for Virtualized Network Functions and Services. *IEEE Commun. Mag.* 2019, 57, 96–102. [CrossRef]
- Soenen, T.; Van Rossem, S.; Tavernier, W.; Vicens, F.; Valocchi, D.; Trakadas, P.; Karkazis, P.; Xilouris, G.; Eardley, P.; Kolometsos, S. Insights from SONATA: Implementing and integrating a microservice-based NFV service platform with a DevOps methodology. In Proceedings of the NOMS 2018-2018 IEEE/IFIP Network Operations and Management Symposium, Taipei, Taiwan, 23–27 April 2018; pp. 1–6. [CrossRef]
- Zafeiropoulos, A.; Fotopoulou, E.; Peuster, M.; Schneider, S.; Gouvas, P.; Behnke, D.; Müller, M.; Bök, P.B.; Trakadas, P.; Karkazis, P. Benchmarking and Profiling 5G Verticals' Applications: An Industrial IoT Use Case. In Proceedings of the 2020 6th IEEE Conference on Network Softwarization (NetSoft), Ghent, Belgium, 29 June–3 July 2020; pp. 310–318. [CrossRef]
- Rizou, S.; Athanasoulis, P.; Andriani, P.; Iadanza, F.; Trakadas, P.; Griffin, D.; Kheirkhah, M.; Breitgand, D.; Weit, A.; Ustok, R.F. Programmable Edge-to-Cloud Virtualization for 5G Media Industry: The 5G-MEDIA Approach. In Proceedings of the IFIP International Conference on Artificial Intelligence Applications and Innovations, Neos Marmaras, Greece, June 5–7, 2020; pp. 95–104. [CrossRef]
- Trakadas, P.; Nomikos, N.; Michailidis, E.T.; Zahariadis, T.; Facca, F.M.; Breitgand, D.; Rizou, S.; Masip, X.; Gkonis, P. Hybrid Clouds for Data-Intensive, 5G-Enabled IoT Applications: An Overview, Key Issues and Relevant Architecture. *Sensors* 2019, 19, 3591, doi:10.3390/s19163591. [CrossRef] [PubMed]
- 10. Bhardwaj, S.; Jain, L.; Jain, S. Cloud computing: A study of infrastructure as a service (IAAS). *Int. J. Eng. Inf. Technol.* **2010**, 2, 60–63.

- 11. Manvi, S.S.; Krishna Shyam, G. Resource management for Infrastructure as a Service (IaaS) in cloud computing: A survey. J. Netw. Comput. Appl. 2014, 41, 424–440. [CrossRef]
- 12. Caruso, G.; Nucci, F.; Gordo, O.P.; Rizou, S.; Magen, J.; Agapiou, G.; Trakadas, P. Embedding 5G solutions enabling new business scenarios in Media and Entertainment Industry. In Proceedings of the 2019 IEEE 2nd 5G World Forum (5GWF), Dresden, Germany, 30 September–2 October 2019; pp. 460–464. [CrossRef]
- Schaffrath, G.; Werle, C.; Papadimitriou, P.; Feldmann, A.; Bless, R.; Greenhalgh, A.; Wundsam, A.; Kind, M.; Maennel, O.; Mathy, L. Network Virtualization Architecture: Proposal and Initial Prototype. In Proceedings of the 1st ACM Workshop on Virtualized Infrastructure Systems and Architectures, VISA '09, Barcelona, Spain, 17 August 2009; pp. 63–72. [CrossRef]
- Pol, A.; Roman, A.; Trakadas, P.; Karkazis, P.; Kapassa, E.; Touloupou, M.; Kyriazis, D.; Juan, L.; Alemany, P.; Vilalta, R. Advanced NFV features applied to multimedia real-time communications use case. In Proceedings of the 2019 IEEE 2nd 5G World Forum (5GWF), Dresden, Germany, 30 September–2 October 2019; pp. 323–328. [CrossRef]
- Shekhawat, Y.; Piesk, J.; Sprengel, H.; Gómez, I.D.; Vicens, F.; Carrillo, S.C.; Trakadas, P.; Karkazis, P.; Zahariadis, T.; Touloupou, M. Orchestrating Live Immersive Media Services Over Cloud Native Edge Infrastructure. In Proceedings of the 2019 IEEE 2nd 5G World Forum (5GWF), Dresden, Germany, 30 September–2 October 2019; pp. 316–322. [CrossRef]
- 16. Botero, J.F.; Hesselbach, X.; Duelli, M.; Schlosser, D.; Fischer, A.; De Meer, H. Energy efficient virtual network embedding. *IEEE Commun. Lett.* 2012, *16*, 756–759. [CrossRef]
- 17. Liu, S.; Cai, Z.; Xu, H.; Xu, M. Security-aware virtual network embedding. In Proceedings of the 2014 IEEE International Conference on Communications (ICC), Sydney, NSW, Australia, 10–14 June 2014; pp. 834–840. [CrossRef]
- 18. Yu, M.; Yi, Y.; Rexford, J.; Chiang, M. Rethinking virtual network embedding: Substrate support for path splitting and migration. *SIGCOMM Comput. Commun. Rev.* **2008**, *38*, 17–29. [CrossRef]
- Zhang, X.; Phillips, C.; Chen, X. An overlay mapping model for achieving enhanced QoS and resilience performance. In Proceedings of the 2011 3rd International Congress on Ultra Modern Telecommunications and Control Systems and Workshops (ICUMT), Budapest, Hungary, 5–7 October 2011; pp. 1–7.
- Gong, L.; Wen, Y.; Zhu, Z.; Lee, T. Toward profit-seeking virtual network embedding algorithm via global resource capacity. In Proceedings of the IEEE INFOCOM 2014-IEEE Conference on Computer Communications, Toronto, ON, Canada, 27 April–2 May 2014; pp. 1–9. [CrossRef]
- 21. Amaldi, E.; Coniglio, S.; Koster, A.M.; Tieves, M. On the computational complexity of the virtual network embedding problem. *Electron. Notes Discret. Math.* **2016**, *52*, 213–220. [CrossRef]
- 22. Chowdhury, N.M.K.; Rahman, M.R.; Boutaba, R. Virtual network embedding with coordinated node and link mapping. In Proceedings of the IEEE INFOCOM 2009, Rio de Janeiro, Brazil, 19-25 April 2009; pp. 783–791. [CrossRef]
- Fajjari, I.; Aitsaadi, N.; Pujolle, G.; Zimmermann, H. Vnr algorithm: A greedy approach for virtual networks reconfigurations. In Proceedings of the 2011 IEEE Global Telecommunications Conference-GLOBECOM 2011, Houston, TX, USA, 5–9 December 2011; pp. 1–6. [CrossRef]
- Botero, J.F.; Hesselbach, X.; Duelli, M.; Schlosser, D.; Fischer, A.; De Meer, H. Flexible VNE algorithms analysis using ALEVIN. In Proceedings of the 11th Würzburg Workshop on IP: Joint ITG, ITC, and EuroNF Workshop "Visions of Future Generation Networks" (Euroview 2011), Würzburg, Germany, 1–2 August 2011; [CrossRef]
- 25. Botero, J.F.; Hesselbach, X.; Fischer, A.; De Meer, H. Optimal mapping of virtual networks with hidden hops. *Telecommun. Syst.* **2012**, *51*, 273–282. [CrossRef]
- 26. Fischer, A.; Botero, J.F.; Beck, M.T.; De Meer, H.; Hesselbach, X. Virtual Network Embedding: A Survey. *IEEE Commun. Surv. Tutor.* **2013**, *15*, 1888–1906. [CrossRef]
- Lischka, J.; Karl, H. A Virtual Network Mapping Algorithm Based on Subgraph Isomorphism Detection. In Proceedings of the 1st ACM Workshop on Virtualized Infrastructure Systems and Architectures, VISA '09, Barcelona, Spain, 17 August 2009; pp. 81–88. [CrossRef]
- Prekas, S.; Karkazis, P.; Trakadas, P. Comparison of Embedding Objectives for Next Generation Networks. In Proceedings of the Fifteenth International Multi-Conference on Computing in the Global Information Technology, Porto, Portugal, 18–22 October 2020; IARIA XPS Press: Wilmington, DE, USA, 2020; pp. 7–12.
- 29. Prekas, S.; Karkazis, P.; Trakadas, P.; Nikolakakis, V. Comparison of VNE heuristic solutions with similar objective functions. In Proceedings of the 24th Pan-Hellenic Conference on Informatics, Athens, Greece; 20–22 November 2020. [CrossRef]
- 30. Du, D.Z.; Ko, K.I. Theory of Computational Complexity, 2nd ed.; John Wiley & Sons Inc.: Hoboken, NJ, USA, 2014. [CrossRef]
- 31. Eppstein, D. Finding the k shortest paths. SIAM J. Comput. 1998, 28, 652–673. [CrossRef]
- 32. Pióro, M.; Medhi, D. *Routing, Flow, and Capacity Design in Communication and Computer Networks,* 1st ed.; Elsevier: Morgan Kaufmann, 500 Sansome Street, Suite 400, San Francisco, CA, USA, 2004. [CrossRef]
- Chowdhury, M.; Samuel, F.; Boutaba, R. Polyvine: Policy-based virtual network embedding across multiple domains. In Proceedings of the Second ACM SIGCOMM Workshop on Virtualized Infrastructure Systems and Architectures, New Delhi, India, 3 September 2010; pp. 49–56. [CrossRef]
- 34. Duelli, M.; Fischer, A.; Botero, J.F.; Diaz, L.; Schlosser, D.; Singeorzan, V.; Hesselbach, X. Alevin2-Tool for the evaluation of ALgorithms for Embedding VIrtual Networks. 2021. Available online: https://sourceforge.net/p/alevin/wiki/home/ (accessed on 10 September 2021).