VNE-NR: A Node-Ranking Method for Performing Topology-Aware and Resource-Driven Virtual Network Embedding

Aqsa Hashmi[a], C.P Gupta[b]
Computer Science Department, Rajasthan Technical University, Kota, India
Email: [a]aqsajaved260@gmail.com, [b]guptacp2@rediffmail.com

***Abstract*—The technology used to overcome network rigidity is known as Network Virtualization. It is a critical technology which is receiving broad recognition nowadays. Virtual Network Embedding is a vital candidate problem in the implementation of future network architecture. It is defined as provisioning of various heterogeneous virtual networks. Those virtual slices of the network share resources of the physical network. Most traditional heuristic virtual network mapping algorithms presented earlier leads to incompetent utilization of physical network resources in the long duration. Hence, our work is profoundly concerned about increasing the effectiveness of virtual network mapping process. We proposed a topology adaptable Virtual Network Embedding algorithm called VNE-NR. We used a novel node-ranking scheme, inciting from recognized Google PageRank algorithm considering global network resource knowledge. In this paper, we support Node-Ranking formulae which is formed with topological attributes of physical network components. The presented technique works in two stages; in the first stage, a node ranking function has been devised to decide the optimal physical node. Whereas In the second stage, both the hops of the paths and available bandwidth were analyzed, to find the optimal path for link mapping. The simulation results prove that our algorithm significantly outmatches two relevant heuristic algorithms in respect of long term acceptance ratio and average revenue to cost ratio.**

***Index Terms*—Network virtualization, Virtual network embedding, Embedding algorithm, Topological attribute, node-ranking approach, Resource consolidation**

I. INTRODUCTION

Computer network's growing complexity requires a need for deeply flexible control contraption. Network virtualization(NV) assures this flexibility by assembling various independent virtual networks sharing the substrate network resources. Thus breaking the traditional network architecture. NV is Providing users with more flexible network services while significantly improving the efficiency of network resource utilization. It has become the next generation of internet implementation in software defined networking, cloud computing and data center networks. Various arbitrary virtual networks can be assembled on top of a particular substrate network. This allows service and infrastructure providers to modify, network topologies to the distinct requirements of any offered service.

Network virtualization partitions the Internet Service Providers(ISPs) into two entities: Infrastructure provider (InPs) who employs and maintains the network appliances and Service Provider(SPs) who lease the physical resources from InPs to design virtual networks and enabled peer-to-peer deployment of the most advanced networking technologies. To provide separation between InPs and SPs, many business models were proposed already, eg. NetScript, Genesis, X-Bone, AGAVE, etc[1].

With the deepening of NV technology research surfaces many challenges. Such as, virtual resource representation, virtual network instantiation, and virtual network administration [2]. However, the allotment of resources demonstrates as the fundamental challenge in the NV process. The useful embedding of virtual network requests(VNRs) to the substrate network(SN) defined as a virtual network embedding problem(VNEP). Achievement of various objective such as resource limitations of node and link, arbitrary arrival of virtual network requests(VNRs)and provisions for access control make the VNE problem NP-hard to solve [3]. There are several proposed VNE algorithms which come under the classification of heuristic, meta-heuristic, and exact solutions. The exact solutions are used for solving VNE problem when the instances of the problem are large; as it is hard to solve it in a short time. It ensures an optimal embedding solution of VN in the scenarios of small network [4]. Several heuristic algorithms are also intended to achieve a counterbalance within execution time and quality of a solution for breaking computational intractability [5]. Most heuristic solutions handle the mapping of every Virtual Network in two phases, node mapping, and link mapping phase. There are several techniques proposed in past literature classified as node ranking methods, topological information-based methods, linear programming methods and many more to solve VNE.

In our work, we are combining the "greedy" node mapping strategy along with topological features. Greedy node mapping is already used in many previous heuristic techniques[6],[7]. This technique ensures the satisfaction to meet the resource requirement of all the future Virtual Network Requests and provide load balancing of the substrate network. It is already proved that considering topological features while mapping is better for resource utilization. Hence, we combined this strategy with topological network features to have efficient performance.

The contributions of this work for VNE are as follows:

- We presented VNE-NR, a new topology-based heuristic

technique which acknowledges Node-Ranking methodology. According to Google's PageRank algorithm, we rank physical network nodes based on topological features and resources of the network. This technique not only utilizes the global topological features of the network (degree and strength) but also the locally available resources(CPU capacity and bandwidth) for embedding.
- Our algorithm recognises the importance of the nodes in terms of topology and open resources in the node-embedding stage and applies the Bellman-Ford algorithm [8] to find the minimum hop(shortest) path connecting mapped nodes for link mapping stage.
- Simulations proved that our proposed technique significantly improves the long-term Average acceptance ratio and R/C ratio.

The rest of the paper is divided as follows: Section II submits the Previous work in VNE. Section III introduces problem formulation of VNE. The simulated analysis and results are carried out in Section V. At last, the paper work is concluded briefly in Section VI.

II. Related Work

There are several approaches which are based on analyzing topological information. Such as Feng*et al.* [9] presented a topology-aware heuristic VNE solution based upon node degree. Their technique utilized attributes of the physical and virtual network topology. In [10] Feng *et al.* performed node and link embedding by analyzing seven corresponding features such as, degree, closeness centrality, fairness centrality, strength, betweenness centrality, and Katz centrality, eigenvector centrality.

Some approaches which were based on linear programming methods were also proposed. Liu*et al.*[11] introduced the concept of the field theory of physics for a two-stage coordinated heuristic VNE algorithm. A multi-objective ILP model is constructed, which achieved a 10 % high success ratio and rev-to-cost ratio than other VNE solutions.

There are many approaches which are based on greedy mapping of nodes with node-ranking. Cao *et al.*[12] introduced another heuristic algorithm VNE-TAGRD for resolving VNE problem. His Technique used the node-ranking method based on essential attributes such as degree, strength, and distance of the nodes. Before beginning the process of embedding, the ranks are assigned to both physical and virtual nodes. Wang *et al.*[13] introduced a distinct topology-aware node-fusion technique. This VNE method allowed the many to one embedding of several virtual onto substrate nodes. Node embedding phase used the node-ranking technique based on available resources. Results concluded that the algorithm improved the acceptance ratio of the consequent link mapping stage. The proposed algorithm also achieved high R/C ratio.

There exist some techniques which use both linear programming model along with node-ranking methods. Lu in *et al.*[14] proposed a technique, which analyzed multiple topological characteristics to measure ranks for nodes and links. Ranks were used for settling the mapping cost of physical network components. They represented node and link embedding with linear programming. They also introduced a dynamic mechanism for reconfiguration of protected nodes to enhance resource usage.

Existing work as mentioned above, involves dividing the problem into two subproblems(node mapping and link mapping). They are using many attributes from the topology. However, They lack in coordination within node mapping and link mapping. This leads to high bandwidth consumption and embedding energy. Therefore, the motivation is to develop a simple and effective algorithm that performs better in coordination and energy consumption. Hence, we design an effective and efficient node ranking function to achieve considerable energy consumption and embedding cost.

III. Problem Formulation and Performance Metrics

This part of the paper gives a universal description of VNEP, its procedures, along with the performance metrics.

A. Virtual network embedding problem

1) Substrate Network(SN): The representation of physical network is done by an undirected weighted graph, $\mathbf{G}_s = (\mathbf{N}_s, \mathbf{E}_s)$ where $\mathbf{N}_s$ and $\mathbf{E}_s$ represent the physical network node (e.g. routers) and link sets (e.g. coaxial cable) respectively. Physical node basic resource has available CPU capacity represented with $\mathbf{CPU}(\mathbf{N}_s)$, and basic genes of physical links is available bandwidth represented with $\mathbf{BW}(\mathbf{E}_s)$. We are also considering collection of all loop-free paths denoted by $\mathbf{P}^s$ in the physical network. $\mathbf{P}^{xy}$ is the path selected from the set of path between node x and y.

2) Virtual Network(VN): Similar to physical networks, virtual networks use undirected weighted graph $\mathbf{G}_v = (\mathbf{N}_v, \mathbf{E}_v)$. $\mathbf{N}_v$ and $\mathbf{E}_v$ respectively represent virtual network node point and link collection set. $\mathbf{N}_v$ basic properties are CPU resource needs represented by $\mathbf{CPU}(\mathbf{N}_v)$ whereas $\mathbf{E}_v$ basic attribute is virtual link bandwidth resource requirement $\mathbf{BW}(\mathbf{E}_s)$.

3) Virtual network mapping: This problem of VNE is further split into two sub-problems ,i.e. node and link mapping. A feasible solution for virtual network mapping is given in figure 1. The node embedding scheme for virtual network request is $a \to A, b \to C, c \to D$. The link embedding scheme is $(a, b) \to (A, C), (a, c) \to (A, D), (b, c) \to (C, D)$.

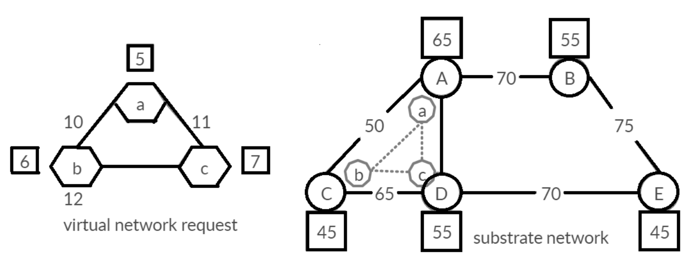

Fig. 1: A feasible illustration of VNE

B. Performance metrics

1) VNR Acceptance Ratio: The long-term average acceptance ratio of arrived VNR(virtual network request) is given by following function:

$$\eta_{suc} = \lim_{T \to \infty} \frac{Num_{Vsuc}}{Num_{Total}} \tag{1}$$

where Num_{Vsuc} represents total number of accepted virtual network request and Num_{Total} corresponds to total number of arrived VNRs in time window *T*.

2) Long-Term Average Revenue: Long-term average revenue obtained in a time frame is given by the following function:

$$\lim_{T \to \infty} \frac{\Sigma_{t=1}^{t=T}(Rev(t)}{T} \tag{2}$$

where $Rev(t)$ represents the revenue of accepting the VNRs at time t which is defined as follows:

$$Rev(G_v, t) = \Sigma_{n_v \in N_v} CPU(n_v) + \Sigma_{e_v \in E_v} BW(e_v) \tag{3}$$

where $CPU(n_v)$ represent CPU capacity of a virtual node n_v, $BW(e_v)$ represent bandwidth of virtual link e_v, N_v represent collection of all virtual nodes and E_v represent collection of virtual links of the VNR G_v respectively.

3) Long-Term Average Revenue over Cost(R/C) Ratio: The long-term average R/C ratio is given by:

$$\xi = \frac{Rev(G_s)}{Cost(G_s)} = \frac{\lim_{T \to \infty} \sum_{t=0}^{T} Rev(G_v, t)}{\lim_{T \to \infty} \sum_{t=0}^{T} Cost(G_v, t)} \tag{4}$$

where $Cost(G_v, t)$ is the accepting cost of VNR_s at the time frame t. $Cost(G_v, t)$ it is defined as:

$$\alpha \sum_{n_v \in N_v} CPU(n_v) + \beta \sum_{e_v \in E_v} \sum_{p \in P_s(e_v)} hops(p) BW(e_v) \tag{5}$$

where parameters α and β are the weighting parameters of the node and link resources respectively, both are set to 1; $P_s(e_v)$ is combined assigned path set for virtual link e_v; $hops(p)$ for the path number p are the number of hops elapsed on the physical network.

IV. VNE-NR: Node-Ranking method for performing Topology-Aware and Resource-Driven VNE

The virtual network embedding problem is considered to be an NP-hard problem. To find an optimal and efficient solution author proposed a new heuristic algorithm which is using node-ranking method.

A. Node importance

1) Node cpu capacity: The node CPU resource can be expressed as:

$$NR(n_i) = CPU(n_i) \tag{6}$$

where node n_i can represent physical nodes in the substrate network.

2) Degree of a node n_i: A degree provides the probability that a node is associated with other nodes. Degree of the node n_i is defined as sum of all adjacent links connected to the node n_i.

$$Deg(n_i) = \sum_{e \in E(n_i)} (e) \tag{7}$$

where E(n_i) is defined as a set of contiguous links for node n_i.

3) Strength of a node n_i: In a weighted graph, links are considered as binary interactions. Strength is defined as collected bandwidths of those nearby links which were considered for the degree. Strength of a node n_i is represented with given formulae:

$$S(n_i) = \sum_{l \in L(n_i)} BW(l) \tag{8}$$

where $L(n_i)$ is set of all adjacent links of n_i

B. Node-Ranking Technique

In this subsection, a node-ranking strategy selected for our algorithm is described.To perform an estimation of the embedding capability for every physical node, we calculated a node-importance index(NIR) in our proposed algorithm. We use topological-attributes such as degree (equation 7) and strength (equation 8) of the node along with locally available resources such as CPU capacity (equation 6) and bandwidth.

Physical node importance index(NIR) is defined as follows:

$$NIR(n_i) = ([NR(n_i) + \frac{S(n_i)}{Link_{tot}}] * Deg(n_i)) \tag{9}$$

where $Link_{tot}$ are the total number of links adjacent to a node n_i. We are considering node importance as the physical node reflects the section from the perspective of the local topology point importance. The greater the node degree the more nodes close to that node for which we are calculating importance index hence, more important the node become. Through this importance index formulae we are ranking the substrate nodes before mapping. The higher the importance index the higher will be the rank and that node is considered first for node mapping with the virtual node of arrived VNR.

C. Node embedding Algorithm

VNE generally done into two separate steps: Virtual node mapping(VNoM) follows a greedy technique to choose a set of available substrate nodes, for each virtual node. Then it selects one of the substrate node based on the maximum number of accessible resources. The goal is to allocate the virtual nodes with more significant requirements to the physical nodes with more significant resources[15]. Then the process moves to the next stage of virtual link mapping, which is defined in the next subsection.

In our proposed work, the embedding of a virtual node to the substrate node described by the node importance index of that node as shown in Algorithm 1 . Importance index of a node gives comprehensive embedding ability to that node. The status of the whole network is backed-up on the arrival of VNR_i (Step 2). Then nodes which are open for mapping are given importance index value(Step 8). If the available resource of node satisfies, the request's demands(CPU capacity), virtual node n_v is greedily embedded to the highest-ranked node n_s (Step 10). If the CPU capacity of selected node cannot meet the required demand of any virtual node of VNR_i, then the request can be rejected (Step 13). If every virtual node of the VNR_i are mapped favorably, the CPU capacity of SN nodes(δ) with label='used' is further updated for the next VNR in the list.

Procedure NODE MAPPING ALGORITHM

Input: Arrived Set of VNRs.
Output: Node Mapping result

```
1:  while There is unmapped virtual node n_v in VNR_i do
2:      Create a copy of SN node list
3:      if label(node)=='close') then
4:          ignore that node
5:      else
6:          Continue with node list δ;
7:      end if
8:      Calculate Node importance index of nodes in list δ by
        using formulae given in equation 9
9:      Take node with maximum index value as n_s
10:     if CPU(n_s) ≥ CPU(n_v) then
11:         Map n_v → n_s
12:     else
13:         Node mapping failed
14:     end if
15:     label(n_s) =' close'
16: end while
```

D. Link mapping algorithm

In the previous algorithms, VLiM has solved in two distinct processes: single path is mapped practicing k-shortest path [16] technique for rising 'k' value when each virtual link necessity is to be mapped just to a single path in the substrate network(SN). Next process is related to multi-path mapping when various paths in the SN can carry each virtual link demand. The solution for multi-path routing of every virtual link is solved by reducing it to Multicommodity flow problem(MCF).

Many algorithms from previous work used an algorithm to find the shortest path. We are also adopting that idea in our approach. The link mapping algorithm Algorithm 2 in our approach maps the virtual links to the substrate links. There are many paths between two mapped nodes, but we are considering the path with minimum hops or with the shortest length called as shortest-path. To find the shortest path within the two pair of mapped nodes in the substrate network, we used the Bellman-Ford algorithm[17](Step 4). Bellman-Ford algorithm is a singular origin shortest path algorithm used to find the shortest path connecting a single source to another node in the network.

Procedure LINK MAPPING ALGORITHM

```
1:  if virtual node mapping of VNR_i is succeded then
2:      for For each unmapped virtual link l in VNR_i do
3:          Obtain two corresponding mapped n_s(start) and
            n_s(end) and take set of all paths connecting these
            two as P^s
4:          Find the physical shortest path (minimum hop) as
            l_s from the set by Bellman-Ford algorithm
5:          if BW(l_s) ≥ BW(l) then
6:              Map the links
7:          else
8:              Reject VNR_i
9:          end if
10:     end for
11: end if
```

TABLE I: Simulation Parameters

Parameter	Value
No. of SN nodes	100
No. of SN links	570
CPU and BW capacity	Uniformly distributed between 50-100
Number of VNR	2000
Arrival rate	15 VNs per time frame with Poisson process.
No. of nodes in each VNR	Randomly generated between 2-20
CPU demand of VNRs	Ranging between 2-25
BW demand of VNRs	Ranging between 5-65 Mbps
Lifetime of a VNR	one time frame or 100 time units
Connection probablity of VNs	0.5

V. PERFORMANCE EVALUATION

This section introduces the evaluation environment settings. We selected two state-of-art heuristic techniques to compare our results. Finally we present our main simulation result.

A. Simulation environment

The experimental set-up of our work is described in table I. We generated substrate network topology which is like a middle sized ISP. We used Python language for simulation and executed are algorithm for 10000 time units for approx 100 time windows. One time window consist of 100 time units.

B. The results have been compared with:

We compare them with the two heuristic algorithms which are latest in this subject and are fit into the simulation of this paper. The description of each algorithm used for comparison is shown in the table II.

C. Evaluation Results

1) Our algorithm performs better with respect to Acceptance Ratio: From figure 2 it can be shown that our algorithm VNE-NR leads to higher acceptance ratio than other two

TABLE II: Compared Algorithms

Strategy	**Description**
EA-FB[18]	A feedback control based algorithm , used energy aware node and link mapping algorithm[19]
MO-NPSO[20]	The adaptive weighted meta-heuristic and multi-objective VNE which is based on particle swarm optimization
VNE-NR	A Node-Ranking Method for Performing Topology-Aware and Resource-Driven Virtual Network Embedding.

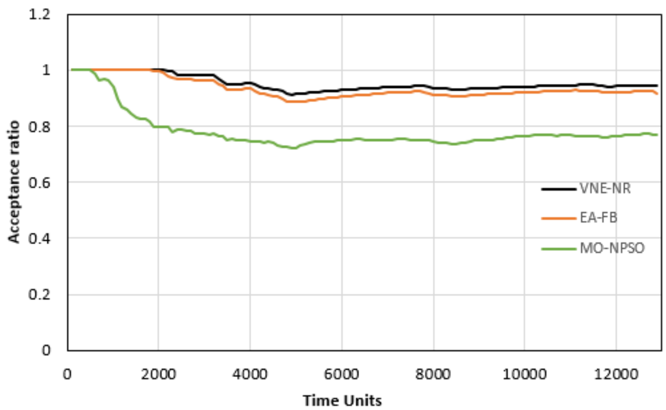

Fig. 2: Acceptance ratio comparison.

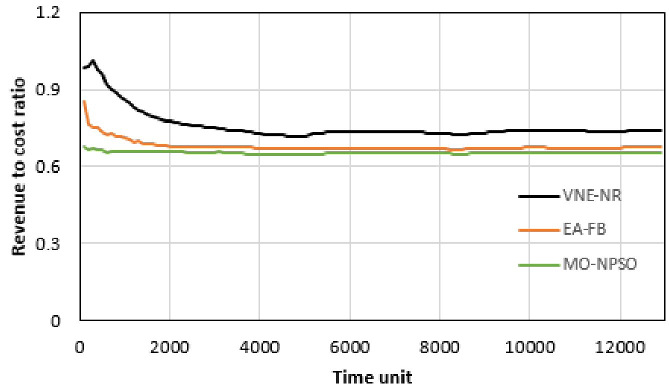

Fig. 3: Comparison of VN R/C ratio.

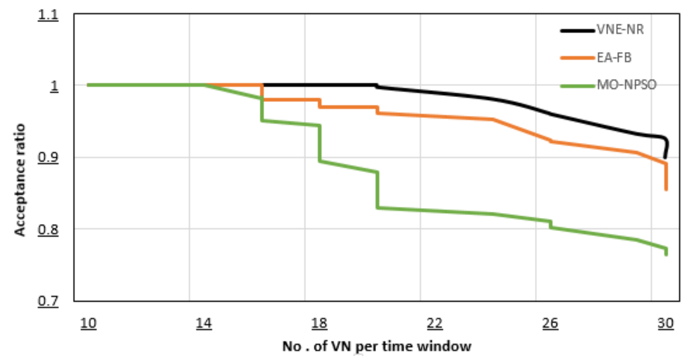

Fig. 4: Acceptance ratio with varying value of λ.

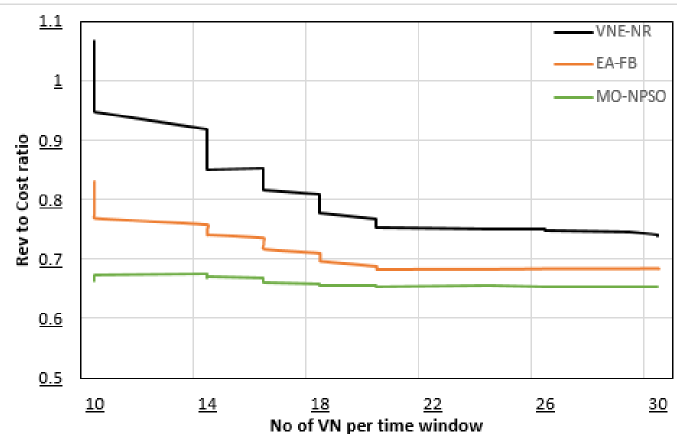

Fig. 5: R/C ratio with varying value of λ.

existing algorithms(EA-FB and MO-NPSO). It runs as expected because VNE-NR takes average of strength of node to calculate importance index. We observed that the link mapping of every VNR is not much suitable in other cases because we used a different algorithm for calculating the shortest path than these two heuristic solutions. Therefore, VNE-NR behave good and performed better in terms of Acceptance ratio.

2) Our algorithm perform better in terms of Rev-to-cost ratio: As we can observe from the figure 3 that our algorithm VNE-NR outperforms other two algorithms with respect to revenue-to-cost ratio. Hence, our approach moreover guarantees the performance of substrate network resource utilization.

3) Observations with increasing rate of VN arrival: The arrival of the VNRs is fashioned by a poisson process. If the particular VNR cannot be accepted instantly it will wait for one time frame. From figure 4 it can be shown that acceptance ratio of all the three algorithms initially remain stable when value of λ is [10-22] after that curve is decreasing at higher λ value.

Whereas from figure 5 it can be shown value of R/C ratio decreases initially with change in rate of arrival of VN request(λ) but remain stable at higher values of λ. We observed efficient stability of our algorithm through this evaluation.

VI. Conclusion and Future Work

This paper introduced a solution for virtual network embedding called VNE-NR. This approach take into account the novel method of node-ranking of physical nodes. The node-ranking function is described based on network resources and topological features. A different approach of calculating average of collected bandwidth is taken into account. Bellmen-Ford shortest path method is adopted for link embedding phase. The consumption of energy and embedding cost is reduced remarkably. As the generation of hibernated links during the process of embedding increases due to path splitting. The calculation of Node ranking is not complex because it is not using many attributes from the substrate network. Hence, the embedding time is taken (cost) by the algorithm is comparatively less. Evaluated results concluded that VNE-NR outperforms two related and novel methods with respect to acceptance ratio and rev-to-cost ratio.

In future work, we will attempt to integrate more topological features like kartz centrality, betweeness centrality to calculate a more better node-ranking function. We will also try to include fault-tolerance in our work to achieve the goal of survivability.

References

[1] Abdallah Jarray and Ahmed Karmouch. Decomposition approaches for virtual network embedding with one-shot node and link mapping. *IEEE/ACM Transactions on Networking*, 23:1012–1025, 2015.

[2] Chenggui Zhao and Behrooz Parhami. Symmetric agency graphs facilitate and improve the quality of virtual network embedding. *Symmetry*, 10(3), 2018.

[3] Edoardo Amaldi, Stefano Coniglio, Arie MCA Koster, and Martin Tieves. On the computational complexity of the virtual network embedding problem. *Electronic Notes in Discrete Mathematics*, 52:213–220, 2016.

[4] H. Cao, L. Yang, Z. Liu, and M. Wu. Exact solutions of vne: A survey. *China Communications*, 13(6):48–62, June 2016.

[5] H. Cao, H. Hu, Z. Qu, and L. Yang. Heuristic solutions of virtual network embedding: A survey. *China Communications*, 15(3):186–219, March 2018.

[6] H.V. Tran and Son Hong Ngo. An enhanced greedy node mapping algorithm for resource allocation in network virtualization. *Journal of Science and Technology*, 12 2012.

[7] Xiang Cheng, Sen Su, Zhongbao Zhang, Kai Shuang, Fangchun Yang, Yan Luo, and Jie Wang. Virtual network embedding through topology awareness and optimization. *Comput. Netw.*, 56(6):1797–1813, April 2012.

[8] D. Cavendish and M. Gerla. *Internet QoS Routing using the Bellman-Ford Algorithm*, pages 627–646. Springer US, Boston, MA, 1998.

[9] Min Feng, Lei Zhang, X. Zhu, J. Wang, Q. Qi, and J. Liao. Topology-aware virtual network embedding through the degree. In *National Doctoral Academic Forum on Information and Communications Technology 2013*, pages 1–6, Aug 2013.

[10] M. Feng, J. Liao, J. Wang, S. Qing, and Q. Qi. Topology-aware virtual network embedding based on multiple characteristics. In *2014 IEEE International Conference on Communications (ICC)*, pages 2956–2962, June 2014.

[11] Xinbo Liu, Buhong Wang, and Zhixian Yang. Virtual network embedding based on topology potential. *Entropy*, 20:941, 12 2018.

[12] H. Cao and L. Yang. Topology attribute and global resource-driven virtual network embedding algorithm via novel node-ranking approach. In *2017 9th International Conference on Wireless Communications and Signal Processing (WCSP)*, pages 1–7, Oct 2017.

[13] Desheng Wang, Weizhe Zhang, Hui He, and Chuanyi Liu. Node-fusion: Topology-aware virtual network embedding algorithm for repeatable virtual network mapping over substrate nodes: Node-fusion: Repeatable virtual network embedding algorithm. *Concurrency and Computation: Practice and Experience*, page e5123, 12 2018.

[14] M. Lu, Y. Lian, Y. Chen, and M. Li. Collaborative dynamic virtual network embedding algorithm based on resource importance measures. *IEEE Access*, 6:55026–55042, 2018.

[15] A. Fischer, J. F. Botero, M. T. Beck, H. de Meer, and X. Hesselbach. Virtual network embedding: A survey. *IEEE Communications Surveys Tutorials*, 15(4):1888–1906, Fourth 2013.

[16] Darryl Veitch. Rethinking virtual network embedding: Substrate support for path splitting and migration. *ACM SIGCOMM Computer Communication Review*, 38:17–17, 04 2008.

[17] Wikipedia contributors. Bellmanford algorithm — Wikipedia, the free encyclopedia, 2019. [Online; accessed 29-July-2019].

[18] Xiaohua Chen, Chunzhi Li, and Yunliang Jiang. A feedback control approach for energy efficient virtual network embedding. *Computer Communications*, 80:16 – 32, 2016.

[19] Sen Su, Zhongbao Zhang, Xiang Cheng, Yiwen Wang, Yan Luo, and Jie Wang. Energy-aware virtual network embedding through consolidation. In *2012 Proceedings IEEE INFOCOM Workshops*, pages 127–132. IEEE, 2012.

[20] Peiying Zhang, Haipeng Yao, Chao Fang, and Yunjie Liu. Multi-objective enhanced particle swarm optimization in virtual network embedding. *EURASIP Journal on Wireless Communications and Networking*, 2016(1):167, Jul 2016.

[21] Bowen Chen, Yongli Zhao, and Jie Zhang. Energy-efficient virtual optical network mapping approaches over converged flexible bandwidth optical networks and data centers. *Optics express*, 23(19):2486024872, September 2015.

[22] P. Zhang, S. Wu, M. Wang, H. Yao, and Y. Liu. Topology based reliable virtual network embedding from a qoe perspective. *China Communications*, 15(10):38–50, Oct 2018.