Journal of Systems Engineering and Electronics (ISSN NO: 1671-1793) Volume 31 ISSUE 1 2021

Determining the NP-Hardness Boundary for Virtual Network Embedding Considering Constraints on Node Locations

Sandeep Kumar Mishra
College of Engineering Bhubaneswar

***Abstract*— The optimization challenge of determining the least expensive mapping between virtual and substrate networks is known as virtual network embedding, or VNE. While the majority of VNE variants are NP-hard, several are not. If the substrate network allows splittable routing and each virtual node has just one possible substrate node because of location restrictions, then VNE is polynomially solvable. The extent to which this polynomial region can be expanded is uncertain, though. This letter explains the barrier by demonstrating that the polynomial region cannot be expanded unless P = NP. Even in the case when each virtual node has exactly two possible substrate nodes, VNE is NP-hard.**

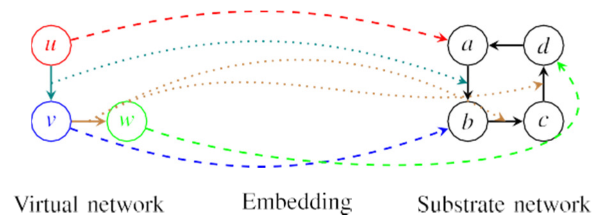

Fig. 1. An example of embedding a virtual network (left) on a substrate network (right). Given location constraints of $l(u) = \{a, b\}$ (i.e., *u* can be mapped to *a* or *b*), $l(v) = \{b\}$, and $l(w) = \{a, d\}$, this example has a feasible embedding (middle); capacity constraints are ignored for simplicity. The dashed lines are node embedding, the dotted lines are link embedding, and the colors indicate the embedding relationship. This instance belongs to *k*-locality of $k \geq 2$, because $|l(v)| \leq 2$, $\forall v \in V_V$ in this example.

I. Introduction

VIRTUAL network embedding (VNE) [1] is an optimization problem, in which a network operator attempts to find a minimum cost mapping between a requested virtual network and its substrate network (Fig. 1). This embedding process is indispensable in network virtualization [2] and is essential in utilizing the substrate infrastructure effectively [3].

Little attention has been paid to the computational complexity [4] aspect of VNE, despite of its long history [1]. This is because proof of the NP-hardness of VNE is trivial since it contains many NP-hard graph problems such as the multi-commodity flow problem [5], the subgraph isomorphism problem [6], and the maximum clique problem [7].

A few existing studies [8]–[10] aim to provide NP-hardness proofs of VNE variants that are believed to be NP-hard. As far as we know, there are only two published papers [9], [10] that discuss and prove the NP-hardness of VNE. These two works formally show that many VNE variants are NP-hard and also difficult to approximate.

Although most variants of VNE are NP-hard as shown by previous studies, some are polynomially solvable. For example, VNE is polynomially solvable if node mapping is fixed and the substrate network supports splittable routing (i.e., a virtual edge can be split and mapped to multiple substrate paths) [11]. Splittable routing is available technology and improves resources utilization [12]. However, it is not clear how large the polynomially-solvable region is.

This letter aims to establish a clear boundary between polynomially solvable VNE and NP-hard VNE with regard to the degree of freedom in node mapping. The existing studies on computational complexity [9], [10] consider the node location constraints, wherein the mapping from a virtual node is limited to only a subset of substrate nodes (the subsets are generally different for each virtual node). These location constraints reflect geographical node placement and functional requirements (FPGA, GPU, SmartNIC, etc.). However, the degree of freedom has yet to be related to computational complexity. To rectify this omission, this letter introduces an upper bound, *k*, on subset size in location constraints and reformulates VNE using parameter *k*. *k*-locality VNE means the subset set size is at most *k* (Fig. 1). For example, 1-locality means that the node location is fixed, while 2-locality means that every virtual node has up to two candidate substrate nodes.

Table I summarizes existing results, trivial results, and our contributions regarding the following three aspects. (i) the degree of freedom of node location constraints, that is, *k*-locality, (ii) splittable routing or unsplittable routing, and (iii) presence of the edge capacity constraint. The existing results [9], [10] correspond to ∞-locality VNE with unsplittable routing and edge capacity (the bottom right corner of the table). Section II explains these existing studies. Trivial results correspond to 1-locality and ∞-locality VNE. All the variants of ∞-locality VNE are NP-hard (the bottom row of the table). 1-locality VNE with unsplittable routing and edge capacity is NP-hard (the top right corner of the table). This result also implies that *k*-locality VNE with unsplittable routing and edge capacity for $k > 1$ is also NP-hard (the rightmost column of the table) because $(k + 1)$-locality

TABLE I
COMPUTATIONAL COMPLEXITY OF VNE PROBLEMS

k-locality	Splittable routing		Unsplittable routing	
	Ignoring edge capacity	With edge capacity	Ignoring edge capacity	With edge capacity
$k = 1$	P [11]		P[a]	NP[b]
$1 < k < \infty$	**NP** (§ IV)			NP[b]
$k = \infty$	NP[c]			NP[b,c] [9], [10]

[a] Clearly P because it can be solved by the shortest path algorithm.
[b] Encompasses the multi-commodity flow problem without splittable routing.
[c] Encompasses subgraph isomorphism.

VNE encompasses k-locality VNE. Other 1-locality VNEs are P (the top row of the table except for the rightmost entry). However, it is unknown to what extent these P regions can be enlarged. Section III explains these trivial results after defining the problem.

Our results, bold in Table I, unveil unknown cases of $1 < k < \infty$ that are NP-hard. Section IV shows that VNE is NP-hard even if each virtual node has up to two candidate substrate nodes, the substrate network supports splittable routing, and all edges have infinite capacity. Our result prevents the waste in trying to construct polynomial-time algorithms of k-locality VNE for $k > 1$ unless P = NP.

II. RELATED WORKS

To the best of our knowledge, only two published works [9], [10] discuss and prove the NP-hardness of VNE. Amaldi *et al.* [9] focus on the VNE with unsplittable routing and multiple virtual networks, in which a network operator simultaneously embeds multiple requested virtual networks onto a physical network. It is shown that, (i) multiple request VNE is NP-hard; (ii) the proof of NP-hardness is also provided for single request VNE with edge capacity, which only corresponds to the case of $k = \infty$ Rost and Schmid [10] deal with single-request VNE variants with unsplittable routing under various combinations of constraints, including location constraints (or the node placement restriction in their terminology). Other notable constraints are routing restriction and latency restriction. They show that many VNE variants are NP-hard and also difficult to approximate. They also consider the location constraint with edge capacity, which merely corresponds to the case of $k = \infty$ In spite of these studies [9], [10] and trivial results, the computational complexity for $1 < k < \infty$ with splittable routing or ignoring edge capacity are still unknown (Table I). Our contribution is to establish a clear boundary for the unknown range of k.

The placement problem of Service Function Chaining (SFC) [13] with Virtual Network Function (VNF) is an extension of VNE [14]. In SFC placement, given the partial orders of VNFs for each flow, the task consists of two parts: virtual network construction and its embedding. The virtual network constructs a network that connects VNFs to satisfy the requested partial orders, and the embedding is the VNE problem. Thus SFC placement can be considered as an extension of VNE. While this letter discusses the condition of P/NP in terms of the node location constraints, [15] studies this problem in terms of the number of virtual nodes; SFC placement can be solved in polynomial time if the upper bound of the number of virtual nodes (VNFs) is three.

III. PROBLEM DEFINITION

We first define a very basic version of VNE. Then, we define the variants of VNE. Finally, we summarize trivial results in terms of the computational complexity.

A. Basic Problem Definition

A substrate network is represented as directed graph (V_S, E_S). Each node and link has available capacity $a : V_S \cup E_S \rightarrow \mathbb{R}_{\geq 0}$ and cost per unit capacity $c : V_S \cup E_S \rightarrow \mathbb{R}_{\geq 0}$. Thus, we write the substrate network as $S = (V_S, E_S, a, c)$.

A virtual network is represented as directed graph (V_V, E_V). Each virtual node and link has a capacity requirement $r : V_V \cup E_V \rightarrow \mathbb{R}_{\geq 0}$. Let $l(v) \subseteq V_S$ be the set of substrate nodes on which the virtual node v can be mapped. We refer to this constraint l as the location constraint. Thus, we write the virtual network as $R = (V_V, E_V, r, l)$. In Section III-B, we will introduce k-locality as a VNE problem in which the size of the location constraint subset $|l(v)|$ is at most k, i.e., $\forall v \in V_V, |l(v)| \leq k$. Since the location constraint is often imposed by the virtual network construction of SFC placement, as noted in Section II, our results can also be applied to the embedding operation.

An embedding consists of node mapping $m_N : V_V \mapsto V_S$ and edge mapping $m_E : E_V \times E_S \rightarrow [0, 1]$. Virtual node $v \in V_V$ is embedded in substrate node $m_N(v) \in V_S$. Edge mapping m_E takes 2 forms: unsplittable and splittable. If the substrate network does not support splittable routing, i.e., edge mapping m_E is unsplittable, then virtual edge $e = (u, v) \in E_V$ is embedded in a single substrate path $P(e) \subseteq E_S$ that connects substrate nodes $m_N(u)$ and $m_N(v)$. Unsplittable edge mapping m_E represents this mapping as follows: $m_E(e, f) = 1 \Leftrightarrow f \in P(e)$. If the substrate network supports splittable routing, then virtual edge $e \in E_V$ is embedded in substrate *paths* and $m_E(e, f)$ is the proportion of virtual link e that is embedded to substrate link $f \in E_S$. Unless otherwise noted, edge mapping follows *splittable* routing.

In VNE, our task is to find the minimum cost embedding (m_N, m_E) that satisfies both capacity and location constraints. The cost of embedding $c(m_N, m_E)$ is the sum of all substrate resource costs

$$c(m_N, m_E) = \sum_{v \in V_V} r(v) c(m_N(v)) + \sum_{e \in E_V, f \in E_S} r(e) c(f) m_E(e, f). \tag{1}$$

The capacity constraints guarantee that substrate resources are sufficient to provision the virtual resource requirements. These constraints are represented as follows:

$$\sum_{v \in V_V : m_N(v) = u} r(v) \leq a(u) \quad \forall u \in V_S, \tag{2}$$

$$\sum_{e \in E_V} r(e) m_E(e, f) \leq a(f) \quad \forall f \in E_S. \tag{3}$$

The location constraints guarantee that node mapping m_N complies with l; i.e., $m_N(v) \in l(v)\ \forall v \in V_V$.

Here we define splittable and unsplittable VNE problems.

Definition 1 (Splittable VNE): Given substrate network (V_S, E_S, a, c) and virtual network (V_V, E_V, r, l). The task is to find the minimum cost embedding (m_N, m_E) that satisfies both capacity and location constraints, following splittable routing.

For the sake of readability, we write splittable VNE as VNE.

Definition 2 (Unsplittable VNE): Given substrate network (V_S, E_S, a, c) and virtual network (V_V, E_V, r, l), the task is to find the minimum cost embedding (m_N, m_E) that satisfies both capacity and location constraints, assuming that edge mapping m_E does not use splittable routing.

B. Variants of VNE

To discuss NP-completeness in Section IV, this subsection defines the decision variants of VNE.

Definition 3 (Decision VNE): Given substrate network (V_S, E_S, a, c), virtual network (V_V, E_V, r, l) and real number $C \in \mathbb{R}$, the task is to determine whether there is an embedding (m_N, m_E) such that its cost $c(m_N, m_E)$ is less than or equal to C and that satisfies both capacity and location constraints.

To discuss whether polynomially solvable VNE variants exist, we introduce the k-locality VNE problem.

Definition 4 (k-Locality VNE): k-locality VNE is the Decision VNE in which the size of the location constraint subset $|l(v)|$ is limited to at most k, i.e., $\forall v \in V_V, |l(v)| \leq k$.

C. Computational Complexity of Variants

Although the computation complexities of 1-locality and ∞-locality VNE are trivial as we will briefly explain, the computational complexity of variants lying between those two extreme points is unknown. Like the well-known SAT problem, where 2SAT is P [16] while 3SAT is NP [7], we aim to clarify the P/NP boundary of VNE, and show that only a part of 1-locality VNE is P while 2-locality VNE is NP.

Here, we explain the complexity of 1-locality VNE variants ($k = 1$ in Table I). Since node mapping m_N is fixed in 1-locality, we discuss the complexity of the remaining edge mapping problem m_E.

- If splittable routing is supported, the edge mapping problem is P, because it can be reduced to the multi-commodity flow problem [11]. The problem is P if flows can be arbitrarily split; otherwise, it is NP [5].
- If splittable routing is unsupported but edge capacity can be ignored, it can be simply solved with the shortest path algorithm, which is P. This is because no mapping of a virtual edge interferes with other mappings.
- If splittable routing is unsupported and edge capacity must be considered, the problem is NP-complete, because it encompasses the multi-commodity flow problem without splittable routing.

Regarding the ∞-locality VNE variants ($k = \infty$ in Table I), all of the variants reduce to NP. This NP-hardness can be deduced from the subgraph isomorphism problem [6]; by using

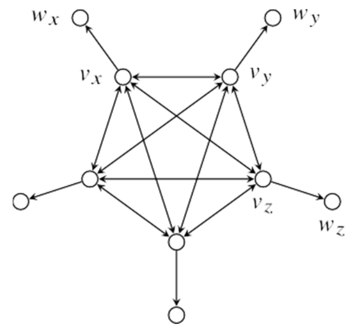

Fig. 2. An example of *virtual* network (V_V, E_V) of V_φ where the number of Boolean variables, n, is 5.

unit cost and unit capacity requirement, i.e., $c \equiv 1$ and $r \equiv 1$, substrate graph (V_S, E_S) contains a subgraph that is isomorphic to request graph (V_V, E_V) if and only if the minimum embedding cost is $|V_V| + |E_V|$.

IV. Complexity Boundary

To prove the NP-completeness of VNE, we use the NP-completeness of MAX-2SAT [17]. We convert an instance of decision MAX-2SAT into an instance of decision 2-locality VNE such that the answer of MAX-2SAT is yes if and only if the answer of VNE is yes. Section IV-A defines MAX-2SAT, while Section IV-B defines a conversion from MAX-2SAT to 2-locality VNE; Section IV-C proves that MAX-2SAT is yes if and only if 2-locality VNE is yes.

A. MAX-2SAT

The decision version of MAX-2SAT is defined as follows.

Definition 5 (Decision MAX-2SAT): Assume the set of clauses $C = \{C_i\}_{i=1}^{m}$ over Boolean variables $X = \{x_1, x_2, \ldots, x_n\}$ and positive integer k. Each clause C_i takes the form of $(l_{i,1} \vee l_{i,2})$ where $l_{i,j}$ is either one of X or the negation of it. The task is to determine whether there is a truth assignment to X that satisfies k or more clauses.

Note that a clause containing only one Boolean variable can be represented by setting $l_{i,1} = l_{i,2}$. We also assume that there is no $(x \vee \neg x)$ since these clauses are always satisfied regardless of the truth assignment.

Decision MAX-2SAT is NP-complete [17].

B. Conversion From 2SAT to VNE

This subsection describes the conversion of an instance of 2SAT into an instance of VNE.

We define an instance of VNE V_φ based on an instance of 2SAT $\varphi = (C, X)$ in the following. Virtual nodes V_V and edges E_V are defined as follows (Fig. 2):

$$V_V = \{v_x | x \in X\} \cup \{w_x | x \in X\}, \tag{4}$$

$$E_V = \{(v_x, v_y) | x, y \in X\} \cup \{(v_x, w_x) | x \in X\}. \tag{5}$$

Substrate nodes V_S and edges E_S are defined as follows (Fig. 3):

$$V_S = \{u_x^0 | x \in X\} \cup \{u_x^1 | x \in X\} \cup \{u^x | x \in X\}, \tag{6}$$

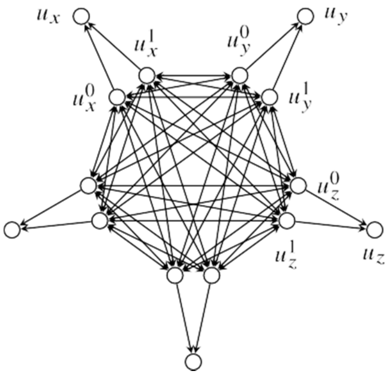

Fig. 3. An example of *substrate* network (V_S, E_S) of V_φ where the number of Boolean variables, *n*, is 5.

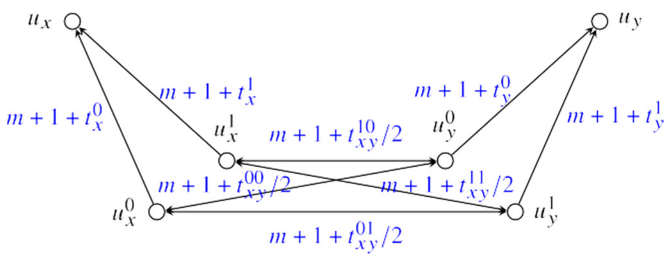

Fig. 4. Enlargement Fig. 3 for nodes related to Boolean variables *x* or *y*. We draw edges only between these nodes and add their cost near the edge in blue. The other parts are omitted for the sake of visibility.

$$E_S = \left\{ \left(u_x^i, u_y^j\right) \mid x, y \in X,\ i, j \in \{0,1\} \right\} \cup \left\{ \left(u_x^i, u_x\right) \mid x \in X,\ i \in \{0,1\} \right\}. \tag{7}$$

To define edge cost *c*, we define t_{xy}^{00} as the number of times clause $(x \vee y)$ occurs for each pair of Boolean variables $x, y \in X$. We also define t_{xy}^{10}, t_{xy}^{01}, t_{xy}^{11} as that of clause $(\neg x \vee y)$, $(x \vee \neg y)$, $(\neg x \vee \neg y)$, respectively. We also define t_x^0, t_x^1 as that of clause $(x \vee x)$, $(\neg x \vee \neg x)$, respectively. We use the following cost, *c*, for substrate edge $e \in E_S$ (Fig. 4):

$$c\left(\left(u_x^i, u_y^j\right)\right) = m + 1 + \frac{t_{xy}^{ij}}{2}, \tag{8}$$

$$c\left(\left(u_x^i, u_x\right)\right) = m + 1 + t_x^i. \tag{9}$$

Node cost is set to 0, that is, we set $c(w) = 0$ for all substrate nodes $w \in V_S$. All virtual nodes and links require one unit resource, that is, $r \equiv 1$. All substrate nodes and links have infinite resource capacity, that is, $\rho \equiv \infty$. We use the following location constraints l : $l(v_x) = \{u_x^0, u_x^1\}$ and $l(w_x) = \{u_x\}$. Thus virtual node w_x is always mapped to the same substrate node, u_x. Note that the locality of VNE instance V_φ is 2.

C. NP-Completeness

To show the NP-completeness of 2-locality VNE, we prove the following theorem.

Theorem 1: Let $\varphi = (C, X)$ be an instance of 2SAT, V_φ be an instance of VNE constructed from φ, *m* be the number of clauses C, and *n* be the number of Boolean variables *X*, that is, $m = |C|$ and $n = |X|$. The maximum number of satisfiable clauses of φ is *k* if and only if the minimum embedding cost of VNE V_φ is $n^2(m+1) + m - k$.

Theorem 1 is sufficient to show the NP-completeness of 2-locality VNE. Because any MAX-2SAT instance (φ, k) produces 2-locality VNE V_φ and if we have a polynomial-time algorithm that can solve 2-locality VNE then we can also solve MAX-2SAT in polynomial-time.

To prove Theorem 1, we transform a truth assignment into an embedding and vice versa. Conversion between a truth assignment and node mapping is trivial since each virtual node of v_x has 2-locality (i.e., $x = i_x \iff m_N(v_x) = u_x^{i_x}$). However, such conversion is not possible for edge mapping since splittable routing generates countless candidates. Here, we use the edge mapping that minimizes the cost. Lemma 1 guarantees the existence and the uniqueness of such embedding, and Lemma 2 evaluates its cost. Lemma 3 describes the transformation from a truth assignment to an embedding, and Lemma 4 describes the reverse. After proving these lemmas, we give the proof of Theorem 1. Note that these four lemmas hold for both splittable and unsplittable VNE.

Lemma 1: If we fix node mapping $m_N : v_x \mapsto u_x^{i_x}$ in V_φ, the following edge mapping, m_E, is the unique edge mapping that minimizes cost, $c(m_N, m_E)$: m_E maps virtual edge (v_x, v_y) to substrate edge $(u_x^{i_x}, u_y^{i_y})$ and virtual edge (v_x, w_x) to substrate edge $(u_x^{i_x}, u_x)$.

Proof: Proof by contradiction. Let us assume that there exists another edge mapping $m_E^l \neq m_E$ that minimizes the embedding cost. Let *a* be the average length of the substrate paths that are mapped to a virtual edge in edge mapping m_E^l.

If average length *a* equals 1 then a virtual edge is mapped to the same substrate edge as m_E. That is, $m_E^l = m_E$. This is because the substrate graph (V_S, E_S) has no parallel edges. However this contradicts the assumption that $m_E^l \neq m_E$.

Hence average length *a* is greater than 1. Thus there exists at least one virtual edge, $e \in E_V$, such that the corresponding substrate path set *P*(*e*) contains at least one substrate path, *Q*, that has *two or more* substrate edges. The cost of this path, *Q*, is at least 2 (*m* + 1) since the cost of any substrate edge is more than *m* + 1. This cost is larger than any substrate edge, $m + 1 + t_{xy}^{ij}$ or $m + 1 + t_x^i$, since $t_{xy}^{ij} \leq m$ and $t_x^i \leq m$. Thus, instead of using path *Q*, using the substrate edge that directly connects the starting node and terminal node of path *Q* gives us edge mapping m_E^* which has a smaller cost than m_E^l. However, this contradicts the assumption that m_E^l minimizes the embedding cost.

Thus, no other edge mapping minimizes embedding cost.

Lemma 2: If we fix node mapping $m_N : v_x \mapsto u_x^{i_x}$ in V_φ, then the minimum embedding cost is

$$n^2(m+1) + \sum_{x,y \in X} \frac{t_{xy}^{i_x i_y}}{2} + \sum_{x \in X} t_x^{i_x}. \tag{10}$$

Proof: Due to Lemma 1, the minimum embedding cost is

$$c(m_N, m_E) = \sum_{e \in E_V} \sum_{f \in E_S} r(e) c(f) m_E(e, f) \tag{11}$$

$$= \sum_{x,y \in X} c\left(\left(u_x^{i_x}, u_y^{i_y}\right)\right) + \sum_{x \in X} c\left(\left(u_x^{i_x}, u_x\right)\right) \tag{12}$$

$$= \sum_{x,y\in X} \left(m + 1 + t_{xy}^{i_x i_y}/2\right) + \sum_{x\in X} \left(m + 1 + t_x^{i_x}\right) \quad (13)$$

$$= n^2(m + 1) + \sum_{x,y\in X} \frac{t_{xy}^{i_x i_y}}{2} + \sum_{x\in X} t_x^{i_x}. \quad (14)$$

Lemma 3: If 2SAT φ is k satisfiable, then VNE V_φ has an embedding whose cost is $n^2(m + 1) + m - k$. ■

Proof: We assume that 2SAT φ is k satisfiable and show that VNE V_φ has an embedding whose cost is $n^2(m+1)+m-k$. Since 2SAT φ is k satisfiable there exists truth assignment A that satisfies exactly k clauses. Based on truth assignment A, we define i_x such that $i_x = 1$ if x is true in A, otherwise $i_x = 0$ for all Boolean variables $x \in X$. We also define node mapping m_N such that $m_N(v_x) = u_x^{i_x}$. Due to Lemma 2, there is edge mapping m_E such that $c(m_N, m_E) = n^2(m + 1) + \sum_{x,y\in X} t_{xy}^{i_x i_y}/2 + \sum_{x\in X} t_x^{i_x}$. By their definition, the latter part of embedding cost $\sum_{x,y\in X} t_{xy}^{i_x i_y}/2 + \sum_{x\in X} t_x^{i_x}$ equals the number of unsatisfied clauses in truth assignment A that is, $m - k$. Thus, embedding cost $c(m_N, m_E)$ equals $n^2(m + 1) + m - k$ which ensures that VNE V_φ has an embedding whose cost is $n^2(m + 1) + m - k$. ■

Lemma 4: If the minimum embedding cost of VNE V_φ is $n^2(m + 1) + m - k$, then 2SAT φ is k satisfiable.

Proof: We assume that the minimum embedding cost of VNE V_φ is $n^2(m + 1) + m - k$ and show that 2SAT φ is k satisfiable. We write the minimum cost embedding as (m_N, m_E). Due to location constraint l, node mapping m_N satisfies the condition of Lemma 2. Thus minimum embedding cost $c(m_N, m_E) = n^2(m + 1) + m - k$ is represented as $c(m_N, m_E) = n^2(m + 1) + \sum_{x,y\in X} t_{xy}^{i_x i_y}/2 + \sum_{x\in X} t_x^{i_x}$ where variable i_x satisfies $m_N(v_x) = u_x^{i_x}$. Based on variable i_x, we define truth assignment A such that x is true if and only if $i_x = 1$. By their definitions, the number of unsatisfied clauses in assignment A is $\sum_{x,y\in X} t_{xy}^{i_x i_y}/2 + \sum_{x\in X} t_x^{i_x} = m - k$. Hence, assignment A satisfies exactly k clauses and we can conclude that 2SAT φ is k satisfiable. ■

Finally, we prove Theorem 1.

Proof of Theorem 1: First, we prove the *only if* part: we assume that the maximum number of satisfiable clauses of 2SAT instance φ is k and prove that the minimum embedding cost of VNE instance V_φ is $n^2(m + 1) + m - k$. Lemma 3 guarantees that there exists an embedding whose cost is $n^2(m + 1) + m - k$. Thus it is sufficient to show that this embedding is the minimum cost embedding. We prove this by contradiction. Suppose that the minimum embedding cost of V_φ is less than $n^2(m + 1) + m - k$. Then, due to Lemma 4, there exists a truth assignment that satisfies more than k clauses. This contradicts the assumption that the maximum number of satisfiable clauses is k.

Next, we prove the *if* part: we assume that the minimum embedding cost of VNE instance V_φ is $n^2(m + 1) + m - k$ and prove that the maximum number of satisfiable clauses of 2SAT instance φ is k. Lemma 4 guarantees that there exists a truth assignment that satisfies k clauses. Thus it is sufficient to show that k is the maximum number of satisfiable clauses of φ. We prove this by contradiction. Suppose that there exists a truth assignment that satisfies more than k clauses. Then, due to Lemma 3, there exists an embedding of V_φ whose cost is less than $n^2(m + 1) + m - k$. However this contradicts the assumption that the minimum embedding cost is $n^2(m + 1) + m - k$. ■

V. Conclusion

Splittable 1-locality VNE is one example of a polynomially solvable version of VNE, despite the fact that the majority of variants are thought to be NP-hard. It was unclear how much more of this polynomial region could be expanded, but this letter dispelled that doubt by demonstrating that k-locality VNE for k > 1 is NP-hard. Because of this, future research into creating polynomial-time k-locality VNE algorithms for k > 1 is no longer necessary, unless P = NP.

References

[1] A. Fischer, J. F. Botero, M. T. Beck, H. de Meer, and X. Hesselbach, "Virtual network embedding: A survey," *IEEE Commun. Surveys Tuts.*, vol. 15, no. 4, pp. 1888–1906, 4th Quart., 2013.

[2] R. Mijumbi, J. Serrat, J.-L. Gorricho, N. Bouten, F. D. Turck, and R. Boutaba, "Network function virtualization: State-of-the-art and research challenges," *IEEE Commun. Surveys Tuts.*, vol. 18, no. 1, pp. 236–262, 1st Quart., 2016.

[3] A. Belbekkouche, M. M. Hasan, and A. Karmouch, "Resource discovery and allocation in network virtualization," *IEEE Commun. Surveys Tuts.*, vol. 14, no. 4, pp. 1114–1128, 4th Quart., 2012.

[4] M. R. Garey and D. S. Johnson, *Computers and Intractability; A Guide to the Theory of NP-Completeness.* New York, NY, USA: W. H. Freeman Co., 1990.

[5] S. Even, A. Itai, and A. Shamir, "On the complexity of time table and multi-commodity flow problems," in *Proc. 16th IEEE Symp. Found. Comput. Sci. (IEEE FOCS)*, Oct. 1975, pp. 184–193.

[6] S. A. Cook, "The complexity of theorem-proving procedures," in *Proc. 3rd ACM Symp. Theory Comput. (ACM STOC)*, May 1971, pp. 151–158.

[7] R. M. Karp, "Reducibility among combinatorial problems," in *Complexity of Computer Computations*. Boston, MA, USA: Springer, 1972, pp. 85–103. [Online]. Available: https://doi.org/10.1007/978-1-4684-2001-2_9

[8] D. G. Anderse, "Theoretical approaches to node assignment," unpublished. [Online]. Available: https://www.cs.cmu.edu/~dga/papers/andersen-assign-abstract.html

[9] E. Amaldi, S. Coniglio, A. M. C. A. Koster, and M. Tieves, "On the computational complexity of the virtual network embedding problem," *Electron. Notes Discrete Math.*, vol. 52, pp. 213–220, Jun. 2016.

[10] M. Rost and S. Schmid, "On the hardness and inapproximability of virtual network embeddings," *IEEE/ACM Trans. Netw.*, vol. 28, no. 2, pp. 791–803, Apr. 2020.

[11] M. Yu, Y. Yi, J. Rexford, and M. Chiang, "Rethinking virtual network embedding: Substrate support for path splitting and migration," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 38, no. 2, pp. 17–29, Apr. 2008.

[12] C.-Y. Hong *et al.*, "B4 and after: Managing hierarchy, partitioning, and asymmetry for availability and scale in Google's software-defined WAN," in *Proc. Conf. ACM Spec. Interest Group Data Commun. (SIGCOMM)*, Aug. 2018, pp. 74–87.

[13] J. Halpern and C. Pignataro, "Service function chaining (SFC) architecture," IETF, RFC 7665, Oct. 2015.

[14] S. Mehraghdam, M. Keller, and H. Karl, "Specifying and placing chains of virtual network functions," in *Proc. 3rd IEEE Int. Conf. Cloud Netw. (IEEE CloudNet)*, Luxembourg City, Luxembourg, Oct. 2014, pp. 7–13.

[15] S. Khebbache, M. Hadji, and D. Zeghlache, "Virtualized network functions chaining and routing algorithms," *Comput. Netw.*, vol. 114, pp. 95–110, Feb. 2017.

[16] M. R. Krom, "The decision problem for a class of first-order formulas in which all disjunctions are binary," *Math. Logic Quart.*, vol. 13, nos. 1–2, pp. 15–20, 1967.

[17] M. R. Garey, D. S. Johnson, and L. Stockmeyer, "Some simplified *NP*-complete graph problems," *Theor. Comput. Sci.*, vol. 1, no. 3, pp. 237–267, Feb. 1976.