On the Distance Spectrum Assignment in Elastic Optical Networks

Haitao Wu, Fen Zhou, *Senior Member, IEEE*, Zuqing Zhu, *Senior Member, IEEE*, Yaojun Chen

***Abstract*—In elastic optical networks (EONs), two lightpaths sharing common fiber links might have to be isolated in the spectrum domain with a proper guard-band to prevent crosstalk and/or reduce physical-layer security threats. Meanwhile, the actual requirements on guard-band sizes can vary for different lightpath pairs because of various reasons. Therefore, in this work, we consider the situation in which the actual guard-band requirements for different lightpath pairs are different, and formulate the distance spectrum assignment (DSA) problem to investigate how to assign the spectrum resources efficiently in such a situation. We first define the DSA problem formally and prove its $\mathcal{NP}$-hardness and inapproximability. Then, we analyze and provide the upper and lower bounds for the optimal solution of DSA, and prove that they are tight. In order to solve the DSA problem time-efficiently, we develop a two-phase algorithm. In its first phase, we obtain an initial solution and then the second phase improves the quality of the initial solution with random optimization. We prove that the proposed two-phase algorithm can get the optimal solution in bipartite DSA conflict graphs and can ensure an approximate ratio of $\mathcal{O}(\log(|V|))$ in complete DSA conflict graphs, where $|V|$ is the number of vertices in the conflict graph, *i.e.*, the number of lightpaths to be considered. Numerical results demonstrate our proposed algorithm can find near-optimal solutions for DSA in various conflict graphs.**

***Index Terms*—Elastic Optical Networks (EONs), Distance Spectrum Assignment (DSA), Physical-Layer Security.**

I. Introduction

RECENTLY, with the rapid growth of traffic demands in backbone networks, how to utilize the spectral resources in optical fibers efficiently and intelligently has become a key challenge for all-optical networks. To address this challenge, flexible-grid elastic optical networks (EONs) have been proposed to enhance the agility of bandwidth allocation in the optical layer [1, 2]. Specifically, in EONs, the bandwidth-variable transponders (BV-Ts) and wavelength-selective switches (BV-WSS') establish lightpaths with several narrow-band (*i.e.*, 12.5 GHz) and spectrally-contiguous frequency slots (FS) and realize data transmissions over them [3]. Therefore, EONs can offer just-enough bandwidth to traffic demands from upper-layer networks, with the fine bandwidth allocation granularity of an FS [4, 5]. For instance, in Fig. 1, there are three lightpath requests in an EON, *i.e.*, R_1, R_2 and R_3, and their bandwidth requirements are 2, 4 and 3 FS, respectively. The spectrum assignments of these lightpaths are illustrated at the bottom of Fig. 1 with blocks in different colors.

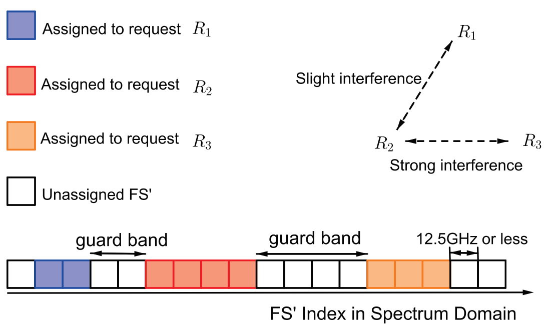


Fig. 1. Spectrum assignments with guard-bands in EONs.

Note that, in order to minimize the potential physical-layer security threat due to inter-channel crosstalk [6], the spectrum assignments of two lightpaths should be separated by a sufficient guard-band when their routing paths share one or more fiber links [7, 8]. These guard-bands, as shown in Fig. 1, can have different sizes, which are not trivial since they determine the impact of inter-channel crosstalk between the lightpaths. In general, the stronger the crosstalk level is or the higher the security requirement is, a larger sized guard-band should be applied. Since the crosstalk level can be affected by many factors such as the required bandwidth, the number of common fiber links and the lightpaths' modulation-levels [9] while the security requirement would depend on the defense of various physical-layer attacks, *e.g.*, eavesdropping and jamming attacks [10], the actual guard-band requirements in EONs would change for different lightpath pairs. Nevertheless, the guard-bands' sizes and the way in which we deploy them would generate spectrum fragmentation and hence significantly influence the spectrum utilization in EONs [11, 12].

Therefore, the service provisioning scheme that uses guard-bands with constant sizes [13] might not be suitable to handle the situation in which the crosstalk levels and/or the security requirements of lightpath pairs are diverse. For instance, a fix-sized guard-band might be insufficient to mitigate a strong crosstalk level while result in spectrum waste for satisfying a relatively low security requirement. Hence, it would be relevant to study how to realize spectrum assignments with various guard-band sizes efficiently.

H. Wu is with the Department of Mathematics, Nanjing University, Nanjing 210093, China, and he is also with with CERI-LIA (Computer Science Laboratory) at the University of Avignon, France. (email: haitao.wu@alumni.univ-avignon.fr).

F. Zhou is with the CERI-LIA (Computer Science Laboratory) at the University of Avignon, France. (email: fen.zhou@univ-avignon.fr).

Z. Zhu is with the School of Information Science and Technology, University of Science and Technology of China, Hefei, Anhui 230027, China (e-mail: zqzhu@ieee.org).

Y. Chen is with Department of Mathematics, Nanjing University, Nanjing 210093, China. (email: yaojunc@nju.edu.cn).

In this paper, we put forward a new spectrum assignment model, which uses guard-bands with different sizes to adapt to the crosstalk level or the security requirement of each lightpath pair in an EON. Our model is named as distance spectrum assignment (DSA). We consider the network planning problem in which all the lightpath requests and their routing paths are known, the spectral resources in the EON are sufficient to serve all the requests, and the mutual crosstalk levels or security requirements of the lightpath pairs are also known. With all the aforementioned information, DSA tries to achieve efficient spectrum assignment that can not only use guard-bands with various sizes to adapt to the mutual crosstalk levels or security requirements of the lightpaths, but also minimize the maximum FS index used in the EON. Note that, to the best of our knowledge, the problem described by DSA has never been studied theoretically in the literature. Moreover, as we will explain in the paper, it is an extremely challenging problem. Hence, we explore the characteristics of the DSA problem and provide some interesting and insightful theoretical results to support future studies in this direction. The contributions of this work can be summarized as follows.

- To the best of our knowledge, this is the first work to formally study the DSA problem. We prove the $\mathcal{NP}$-hardness of the problem and analyze its inapproximability, and also formulate an integer linear programming (ILP) model to solve it exactly.
- We formally provide the upper and lower bounds of the optimal solution of DSA and prove that they are tight.
- We propose a two-phase algorithm to solve the DSA problem time-efficiently, and study its performance in various DSA situations, which are represented by different conflict graphs. Specifically, in a conflict graph, each vertex represents a lightpath while an edge signifies the guard-band requirement between two lightpaths. In its first phase, the algorithm generates an initial solution, which is proven to be the optimal solution in bipartite conflict graphs and can guarantee an approximate ratio of $\mathcal{O}(\log|V|)$ in complete conflict graphs. The second phase improves the initial solution with a random optimization procedure, whose convergence performance are also analyzed mathematically.

The rest of this paper is organized as follows. Section II presents our motivation and the related work. In Section III, we model the DSA problem and analyze its hardness. The upper and lower bounds of the optimal solution of DSA are analyzed in Section IV. In Section V, we transform DSA into a permutation-based optimization problem (POP), and with this transformation, the two-phase algorithm is developed in Section VI. The performance of the two-phase algorithm is theoretically analyzed in Section VII, and the numerical results for performance evaluation are presented in Section VIII. Finally, Section IX summarizes the paper.

II. Motivation and Related Work

With the recent advances in optical devices and transmission techniques, the concept of EON has been proposed to make the resource management in the optical layer more flexible and hence attracted intensive research interests [1–4, 7, 11–15]. The authors of [1] systematically discussed the enabling technologies and building blocks of EONs, *e.g.*, bandwidth-variable wavelength cross-connects (BV-WXCs), and laid out the network architecture of EON, which enables flexible bandwidth allocation with a fine granularity to provide just-enough bandwidth to customer traffic demands. Hence, compared with the traditional fixed-grid wavelength-division multiplexing networks, EON can significantly improve the utilization efficiency of spectrum resources. Note that, since the channel spacing in EONs becomes much narrower than that in WDM networks, the usage of guard-bands, *i.e.*, the unused FS in between the spectrum assignments of two spectrally adjacent lightpaths, becomes more tricky. Specifically, if the guard-bands are not properly chosen, the physical impairments in fibers would induce crosstalk between the lightpaths and thus their quality-of-transmission (QoT) would be deteriorated. Moreover, the crosstalk between two spectrally adjacent lightpaths can be easily utilized to realize physical-layer attacks such as eavesdropping and power jamming [6, 7, 10, 16], and mixed modulation attacks can also degrade the quality of high-bit-rate phase-modulated lightpaths with cross-phase modulation [17]. Therefore, we have to carefully choose the guard-bands to reduce the risk of physical-layer attacks, the degradation of QoT and the nonlinear penalty in EONs [6, 10, 17].

In order to realize spectrally efficient lightpath provisioning, the routing and spectrum assignment (RSA) problem has already been intensively studied. In [4], RSA has been formally defined along with the discussion on its complexity, and an ILP model and two time-efficient heuristics have been designed to solve the RSA problem. The authors of [13, 18] have considered to provision multicast requests in EONs with the multicast-capable routing, modulation-level, and spectrum assignment (RMSA). Moreover, the RSA/RMSA algorithms for more sophisticated service provisioning schemes, such as advance reservation [19], spectrum defragmentation [20], and virtual network embedding [21], have also been investigated before. However, most of the previous studies on RSA assumed that the guard-bands use a constant size for all the lightpath pairs. Note that, the work in [9] had already revealed that the filtering characteristics of optical components can make the selection of guard-band sizes extremely sophisticated. Therefore, using a fixed guard-band size does not coincide with the practice and thus the problem of DSA, *i.e.*, the spectrum assignment with various guard-band sizes should be investigated in a timely manner.

In general, the wavelength assignment (WA) problem in WDM networks (each request in WA problem is assigned with a fixed-wavelength frequency) and the spectrum assignment (SA) problem in EONs (each request in SA problem is assigned with a number of FS, the details of distinction between the WA and SA are shown in Table I) can both be studied by leveraging the graph coloring method [22] in conflict graphs that are constructed based on the routing results of lightpaths. Specifically, WA can be solved by finding the chromatic number of the conflict graph [23, 24] while SA can be solved with the interval chromatic number [4, 25]. Nevertheless, DSA differs from the classical graph coloring

TABLE I
COMPARISON OF RELATED COLORING PROBLEMS

	Classical Coloring [22] (*e.g.*, WA [23])	Fractional Coloring [26]	Traditional SA [4]	DSA (this work)
Vertex Color	One color	A set of colors	A set of colors	A set of colors
Color Contiguity	N/A	No need	Required	Required
Color Distance of Adjacent Vertices	Disjoint	Disjoint	Identical positive integer	Various positive integers

problem [22] in two aspects: 1) each vertex in the conflict graph, which represents a lightpath, is assigned with a set of contiguous colors (*i.e.*, FS) according to the bandwidth demand rather than only one color; and 2) the distance of the color sets of two adjacent vertices is no longer one but a positive integer, representing the guard-band requirement, which is not identical for all the vertex pairs. More specifically, DSA is similar to the fractional coloring problem [26], with two differences: 1) contiguous colors should be assigned to each vertex in DSA, while this is not the case for fractional coloring; and 2) various distances between adjacent color sets should be kept in DSA while color sets only need to be disjoint in the latter one. For clarity, Table I provides the comparison of the four coloring related problems that have been discussed above, *i.e.*, the classical coloring, the fractional coloring, the traditional SA, and the DSA problem. We can see that DSA is apparently a new combinatorial optimization problem, which has not yet been studied before. In the next section, we will formally define the DSA problem.

III. Distance Spectrum Assignment (DSA) Problem

In an EON, a set of FS is available in each optical fiber to carry lightpaths. Hence, efficient spectrum assignment algorithms are needed to optimize the spectrum usages of lightpaths under the spectrum contiguous and non-overlapping constraints [2]. Meanwhile, in DSA, to address the crosstalk level and/or security requirement of each lightpath pair, we need to choose a proper guard-band to insert.

A. Problem Description

For DSA, we consider the network planning problem in which all the lightpath requests and their routing paths are known, the spectral resources in the EON are sufficient to serve all the requests, and the mutual crosstalk levels or security requirements of the lightpath pairs are also known (*i.e.*, the required guard-band sizes are given for all the spectrally adjacent lightpath pairs). Then, DSA tries to achieve efficient spectrum assignment that can not only accommodate all the lightpath requests to satisfy all the constraints, but also minimize the maximum used FS index (MUFI) in the EON.

To solve DSA, we construct a conflict graph based on the known information regarding the lightpaths. Specifically, we first use a vertex to represent each lightpath and assign a weight to it for its bandwidth demand in FS, and then we connect two vertices with an edge if there would be crosstalk between their lightpaths or a guard-band has to be inserted in between the lightpaths' spectrum assignments due to certain customer-specified security reasons. Note that, a weight is also assigned to each edge in the conflict graph to represent the actual required guard-band size. Figs. 2 and 3 and Table II show an illustrative example on how to construct the conflict graph. There are four lightpaths with the information in Table II and their routing paths in a 4-node ring topology is illustrated in Fig. 2. Then, we assume that the guard-band requirements for the lightpaths are shown in Fig. 3(a), where for simplicity, we use the number of common links in two lightpaths' routing paths as their guard-band requirement. Note that, previous experimental investigation has suggested that the crosstalk level between two spectrally adjacent lightpaths is positively correlated with the number of common links in their routing paths [9]. For instance, since the routing path of R_1, *i.e.*, B-A-D, shares two common links with that of R_4 (C-B-A-D), the required guard-band size between them would be at least 2 FS. In the conflict graph in Fig. 3(a), the number inside a cycle is the bandwidth demand in FS while the number on an edge indicates the required guard-band size. Based on the conflict graph in Fig. 3(a), we can figure out that optimal solution of DSA is that in Fig. 3(b), where the assigned FS to each lightpath are marked with red braces.

TABLE II
INFORMATION ON LIGHTPATHS

	Bandwidth Demand	**Route**
Request R_1	3 FS	B-A-D
Request R_2	2 FS	C-B-A
Request R_3	3 FS	A-D-C-B
Request R_4	1 FS	C-B-A-D

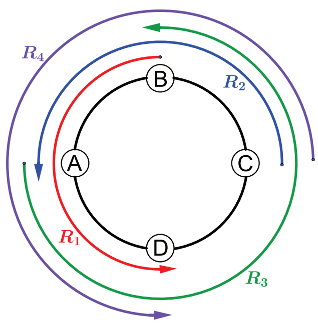
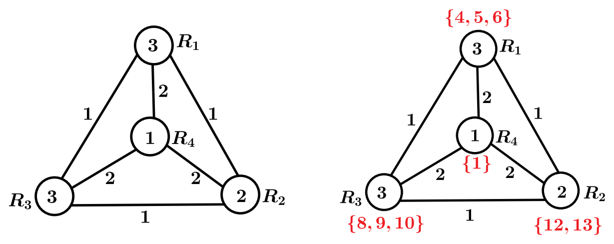


Fig. 2. Lightpaths in Table II in a 4-node ring topology.

B. DSA model and Integer Linear Program

Note that, since we only consider the spectrum assignment problem in DSA, which is already a relatively complex problem as we will explain below, we assume that the routing and guard-band information on the lightpaths are known and thus for each instance of DSA, the conflict graph has already been constructed. Therefore, from now on, we concentrate on how to obtain the optimal spectrum assignments for the lightpaths (*i.e.*, the optimal solution of DSA) based on a known conflict



(a) Conflict graph (b) Optimal solution

Fig. 3. Conflict graph for lightpaths in Fig. 2 and optimal DSA solution.

graph, and consider various types of conflict graphs in our analysis. We first introduce the following notations for DSA.

Necessary Notations:

- $G(V, E)$: The DSA conflict graph, where V is the set of vertices, and E is the set of the conflict edges.
- $\mathbb{N}^+$: The set of natural numbers for representing the FS indices in the spectrum domain, which starts from 1.
- v_i: $v_i \in V$ represents the i-th lightpath request.
- v_i^w: The integer weight signifies bandwidth demand of lightpath v_i, in the number of contiguous FS.
- w_{v_i}: The set of contiguous FS assigned to v_i.
- v_i^b: $v_i^b \in \mathbb{N}^+$ is the start-index of w_{v_i}.
- v_i^a: $v_i^a \in \mathbb{N}^+$ is the end-index of w_{v_i}.
- e or v_iv_j: The edge $e \in E$ connecting v_i and v_j, which represents that the lightpaths of v_i and v_j share common link(s). For convenience, we also use v_iv_j to represent an edge e.
- d_e $(d_{v_iv_j})$: The positive integer weight that represents the least guard-band size between lightpaths v_i and v_j.
- B: $B \in \mathbb{N}^+$ is a reasonably large integer.

For ease of discussion, we also use $G(V, E, \{v_i^w\}, \{d_{v_iv_j}\})$ to represent a DSA graph, *i.e.*, making the weights of vertices and edges explicit. Our objective is to minimize the MUFI in the EON. Note that, it is also possible that the conflict graph G is not a fully connected one. In that case, we can partition G in to a few connected components, solve the DSA problem in them, and then get the MUFI in all the components as the final solution. Hence, we will ignore the cases of non-connected conflict graph in our discussions. The DSA problem can then be defined as

$$\text{Minimize} \quad \max_{s \in \left(\underset{v_i \in V}{\cup} w_{v_i}\right)} (s) \qquad (\textbf{DSA}), \tag{1}$$

where $s \in \mathbb{N}^+$ is the index of a used FS. Meanwhile, DSA should be subject to the following constraints:

- **Bandwidth Requirement Constraint:** Each lightpath should be assigned with enough FS to satisfy its bandwidth demand. In other words, the cardinality of FS set assigned to a vertex $v_i \in V$ should be equal to its weight:
$$|w_{v_i}| = v_i^w, \ \forall v_i \in V, \tag{2}$$
- **Spectrum Continuity Constraint:** The FS assigned to a lightpath should be the same on each fiber link in its routing path. Basically, since each lightpath is pre-routed and represented by a vertex in the conflict graph, this constraint will always be satisfied automatically.
- **Spectrum Contiguity Constraint:** The FS assigned to a vertex should be contiguous in $\mathbb{N}^+$, *i.e.*, w_{v_i} can be expressed as $\{v_i^b, v_i^b + 1, ..., v_i^a - 1, v_i^a\}$, where $v_i^b, v_i^a \in \mathbb{N}^+$.
- **Spectrum Set Distance Constraint:** To satisfy the guard-band requirements, the distance between the FS sets assigned to two spectrally adjacent lightpaths should be large enough. Specifically, for each edge $v_iv_j \in E$, the distance between w_{v_i} and w_{v_j} in $\mathbb{N}^+$ should not be smaller than the edge's weight:
$$\text{distance}(w_{v_i}, w_{v_j}) \ge d_{v_iv_j}, \ \forall v_iv_j \in E, \tag{3}$$
where,
$$\text{distance}(w_{v_i}, w_{v_j}) = \min_{s \in w_{v_i}, t \in w_{v_j}} (|s - t| - 1).$$

The DSA problem is $\mathcal{NP}$-hard, which will be proven formally in the next subsection. To solve DSA exactly, we formulate an ILP model to obtain the optimal spectrum assignment scheme.

Decision Variables:

- x_i^b: Integer variable to represent the value of v_i^b.
- x_i^a: Integer variable to represent the value of v_i^a.
- y: Integer variable to represent the upper bound of x_i^a.
- $o_{v_iv_j}$: Boolean variable for each edge v_iv_j to represent the order of x_i^b and x_j^b, *i.e.*, if $x_i^b > x_j^b$, we have $o_{v_iv_j} = 1$, and $o_{v_iv_j} = 0$ otherwise.

Objective Function:

$$\begin{aligned} &\text{Minimize} \quad y \quad \text{(ILP-DSA)}, \\ &s.t. \quad \text{Eqs. (5)-(10).} \end{aligned} \tag{4}$$

$$x_i^a - x_i^b + 1 = v_i^w, \qquad \forall v_i \in V \tag{5}$$
$$o_{v_iv_j} + o_{v_jv_i} = 1, \qquad \forall v_iv_j \in E \tag{6}$$
$$x_i^a - x_j^b + d_{v_iv_j} + 1 \le B \times o_{v_iv_j}, \qquad \forall v_iv_j \in E \tag{7}$$
$$y \ge x_i^a, \qquad \forall v_i \in V \tag{8}$$
$$x_i^a \in \mathbb{N}^+, x_i^b \in \mathbb{N}^+, \qquad \forall v_i \in V \tag{9}$$
$$o_{v_iv_j} \in \{0, 1\} \qquad \forall v_iv_j \in E \tag{10}$$

C. Hardness and Inapproximability Analysis

To analyze the hardness of the DSA problem, we introduce the Minimum Hamilton Path problem (MHP) [27], whose objective is to find a minimum Hamilton path in a weighted complete graph. MHP is strongly $\mathcal{NP}$-hard [27].

If the conflict graph of a DSA instance is complete, which means every two vertices v_i and v_j are directly connected. Hence, the FS sets assigned to the lightpaths should be pairwise disjoint. If the complete graph satisfies the triangle inequality, *i.e.*, $d_{v_iv_k} + d_{v_kv_j} \ge d_{v_iv_j}$, $\forall v_i, v_j, v_k \in V$, owing to this inequality, any Hamilton path satisfies the spectrum set distance constraint of DSA. Therefore, in this case, the DSA problem is equivalent to the MHP problem. If the triangle inequality cannot be satisfied in the complete graph, then the solution of DSA might be longer than a Hamilton path. This is because the spectrum set distance constraint might not be satisfied by a Hamilton path. Precisely speaking, the distance between two vertices $v_i, v_j \in V$ in a Hamilton path may be smaller than the required spectrum distance $d_{v_iv_j}$. Theorem 1 indicates the hardness of DSA.

Theorem 1: MHP $\leq_T^{P1}$ DSA

Proof: To prove the $\mathcal{NP}$-hardness of DSA, we just need to prove: 1) any instance $\mathcal{I}$ of MHP can be polynomial-time reduced to an instance $\mathcal{I}'$ of DSA, and 2) the solution of $\mathcal{I}'$ can be converted to that of $\mathcal{I}$ in polynomial time.

We get $\mathcal{I}'$ from $\mathcal{I}$ by giving the biggest edge weight b to each vertex of $\mathcal{I}$ as its weight and keeping the edges' weights unchanged. Then, we have $d_{v_iv_k} + b + d_{v_kv_j} \geq d_{v_iv_j}$, $\forall v_i, v_j, v_k$. With this reduction, each vertex pair in a Hamilton path in $\mathcal{I}'$ should satisfy the spectrum set distance constraint. Hence, the solution of $\mathcal{I}$ equals that of $\mathcal{I}'$ minus $|V| \cdot b$, where $|V|$ is the number of vertices. For example, in Fig. 4, if we set the weights of the four vertices (*i.e.*, v_1, v_2, v_3 and v_4) to 3, which is the biggest edge weight, the MHP instance $\mathcal{I}$ becomes a DSA instance $\mathcal{I}'$. The minimum Hamilton path can be obtained by solving the DSA instance, which is shown in Fig. 4 with red color. The total weight of the minimum Hamilton path is 5, which is obtained by subtracting 12 from the solution of $\mathcal{I}'$. Therefore, we prove that the DSA problem is also $\mathcal{NP}$-hard. ∎

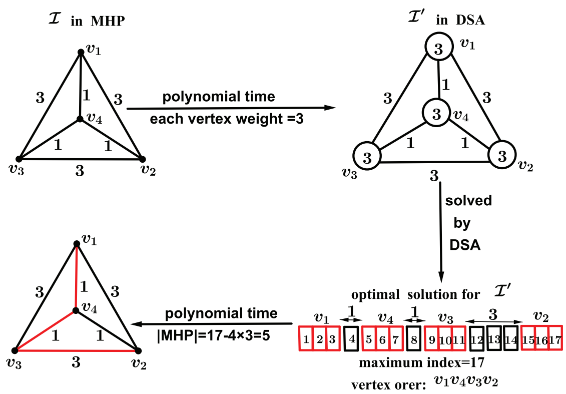


Fig. 4. Example on reducing MHP to DSA in polynomial time.

To analyze the inapproximability of DSA, we first introduce the inapproximability result on the chromatic number $\chi(G)$ [2] of a graph $G(V, E)$. Given a polynomial-time algorithm A to compute the chromatic number of a graph G, we use $A(G)$ to denote the chromatic number obtained by A. The inapproximability of $\chi(G)$ is given by the following statement:

Unless $\mathcal{NP} \subset \mathcal{ZPP}$, no polynomial-time algorithm A that computes the chromatic number of G can guarantee $\frac{A(G)}{\chi(G)}$ within $\mathcal{O}(|V|^{1-\epsilon})$ for an arbitrary graph G, where $|V|$ is the number of vertices in G and $\epsilon > 0$ [28].

[1] $\leq_T^P$ means a polynomial-time Turing reduction.

[2] The chromatic number is the minimum number of colors needed to color G such that adjacent vertices do not share the same color [22].

Then, we have Theorem 2.

Theorem 2: Unless $\mathcal{NP} \subset \mathcal{ZPP}$, no polynomial-time heuristic algorithm APX for DSA can guarantee $\frac{APX(\mathcal{I})}{OPT(\mathcal{I})}$ within $\mathcal{O}(|V|^{1-\epsilon})$ for all the instances $\mathcal{I}$ of DSA, where $APX(\mathcal{I})$ denotes the MUFI obtained by APX, $OPT(\mathcal{I})$ denotes the optimal result, $|V|$ is the number of vertices in $\mathcal{I}$ and $\epsilon > 0$.

Proof: For an arbitrary graph $G(V, E)$, we can reduce G to a DSA instance $\mathcal{I}$ in polynomial time by setting the weights of all the vertices and edges as one. Here, we denote this reduction as R, *i.e.*, $R(G) = \mathcal{I}$. According to Theorem 4, which will be given in the next section, $OPT(R(G)) < 2\chi(G)$. Therefore, if we assume that APX can ensure $\frac{APX(\mathcal{I})}{OPT(\mathcal{I})} < \mathcal{O}(|V|^{1-\epsilon})$ for an arbitrary DSA instance $\mathcal{I}$, $\frac{APX(R(G))}{OPT(R(G))} < \mathcal{O}(|V|^{1-\epsilon})$ would be valid for an arbitrary graph G. Hence, $\frac{APX(R(G))}{2\chi(G)} < \frac{APX(R(G))}{OPT(R(G))} < \mathcal{O}(|V|^{1-\epsilon})$ would be valid for an arbitrary graph G, which means that $APX(R(G))$ can guarantee a ratio within $\mathcal{O}(|V|^{1-\epsilon})$ for $\chi(G)$. This, however, contradicts with the inapproximability of $\chi(G)$. Thus, we get the proof. ∎

IV. Upper and Lower Bounds of DSA's Optimal Solution

In this section, we analyze the upper and lower bounds of DSA's optimal solution. For ease of discussion, we first introduce some terminologies and definitions.

- $C(G)$: The condensation graph of a DSA conflict graph $G(V, E, \{v_i^w\}, \{d_{v_iv_j}\})$. For the conflict graph, the vertex set V can be partitioned into $\chi(G)$ independent sets. We merge the vertices in the same set as a single super-vertex and assign the maximum weight of the vertices in the set as the weight of the super-vertex. Then, each super-vertex pair in the new graph might have multiple edges. Among these edges, we only keep the one with the biggest weight and remove the others. Finally, we obtain the condensation graph of G.
- $V_{C(G)}$: Set of the vertices in $C(G)$ and $|V_{C(G)}| = \chi(C)$.
- $E_{C(G)}$: Set of the edges in $C(G)$.
- v_i' and $v_i'^w$: $v_i' \in V_{C(G)}$ is the vertex with the i-th biggest weight and $v_i'^w$ is its weight.
- $w_{v_i'}$, $v_i'^b$ and $v_i'^a$: Their definitions are similar as those of w_{v_i}, v_i^b and v_i^a.
- e_i' and $d_{e_i'}$: $e_i' \in E_{C(G)}$ is the edge with the i-th biggest weight and $d_{e_i'}$ is its weight.
- Maximal clique ψ and Maximal clique set Ψ: Given a graph $G(V, E)$, we call a subgraph $\psi(V_\psi, E_\psi) \subseteq G(V, E)$ a clique when $\psi(V_\psi, E_\psi)$ is a complete graph. A clique $\psi(V_\psi, E_\psi)$ is a maximal clique if and only if there is no clique $\psi' \subseteq G$ and $\psi \subsetneq \psi'$. We use $\Psi(G)$ to denote the set of maximal cliques in G. In the example in Fig. 5, there are three maximal cliques ψ_1, ψ_2 and ψ_3, and thus $\Psi(G) = \{\psi_1, \psi_2, \psi_3\}$ as indicated in the figure.

To study the feature of DSA's optimal solution, we start with the bipartite graphs whose chromatic number $\chi(G)$ is 2.

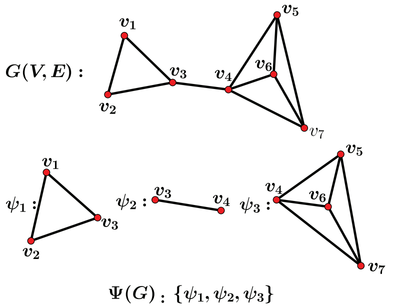


Fig. 5. Maximal clique ψ and maximal clique set $\Psi(G)$.

Then, we continue to investigate the connection between the optimal solution and $\chi(G)$ of a DSA conflict graph. Firstly, we give an obvious fact, which is needed in later proofs.

Fact 1: Given a DSA conflict graph $G(V,E)$, if there is a solution whose MUFI equals $\max\limits_{v_i v_j \in E}(d_{v_i v_j} + v_i^w + v_j^w)$, then it is an optimal one.

Proof: Due to the spectrum set distance constraint, the MUFI cannot be less than $d_{v_i v_j} + v_i^w + v_j^w$ for any $v_i v_j \in E$. Hence, if a solution reaches this lower bound, it is optimal. ■

We use $opt(G)$ to represent an optimal solution of a DSA conflict graph G, and $|opt(G)|$ to denote the numerical value of the optimal solution, *i.e.*, the optimal MUFI. A *proper spectrum assignment* means that we assign FS sets to the vertices under the four constraints of the DSA problem.

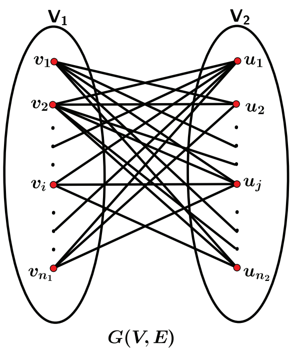


Fig. 6. Bipartite DSA graph $G(V,E)$, $V = (V_1, V_2)$.

Theorem 3: If a DSA conflict graph $G(V,E)$ is a bipartite graph (as shown in Fig. 6), where $V = (V_1, V_2)$ with $v_i \in V_1$, $1 \le i \le |V_1|$ and $u_j \in V_2$, $1 \le j \le |V_2|$, we have $|opt(G)| = \max\limits_{v_i u_j \in E}\{d_{v_i u_j} + v_i^w + u_j^w\}$.

Proof: Based on Fact 1, we just need to find a proper spectrum assignment for all the vertices in $G(V,E)$ whose MUFI equals $d_{v_i u_j} + v_i^w + u_j^w$ for certain edge $v_i u_j \in E$. For each $v_i \in V_1$, we assign the FS set w_{v_i} with $v_i^b = 1, v_i^a = v_i^w$. For vertex $u_j \in V_2$, we assign the FS set w_{u_j} with $u_j^b = \max\limits_{v_i u_j \in E}\{v_i^a + d_{v_i u_j} + 1\}$ and $u_j^a = u_j^b + u_j^w - 1$. Because vertices in the same side of the bipartite graph are not adjacent, we can simply verify that the aforementioned spectrum assignment is proper. Hence, we can see that the final MUFI equals $\max\limits_{v_i u_j \in E}\{d_{v_i u_j} + v_i^w + u_j^w\}$. Hence, we finish the proof. ■

Corollary 1: If a DSA conflict graph $G(V,E)$ is of such topological structure as tree, even ring or grid, we have $|opt(G)| = \max\limits_{v_i v_j \in E}\{d_{v_i v_j} + v_i^w + v_j^w\}$.

Proof: Since tree, even ring and grid are bipartite graphs, the proof is trivial based on Theorem 3. ■

Based on the analysis above, we can see that the optimal solution for a DSA conflict graph G with $\chi(G) = 2$ can be obtained easily. Next, the question is how about the conflict graph with $\chi(G) \ge 3$. Apparently, the analysis becomes more difficult for a larger chromatic number. But fortunately, we can get the upper and lower bounds for $|opt(G)|$ by leveraging $\chi(G)$ and the maximal clique ψ. We use MHP(ψ) to represent a minimum Hamilton path in a clique ψ, $|\text{MHP}(\psi)|$ to represent its length, and ψ^w to denote the total weight of the vertices in ψ, *i.e.*, $\psi^w = \sum\limits_{v_i \in V_\psi} v_i^w$.

Theorem 4: Given an arbitrary conflict graph G, the inequality in Eq. (11) holds for the optimal solution of the DSA problem.

$$\max_{\psi \in \Psi(G)}\{|\text{MHP}(\psi)| + \psi^w\} \le |opt(G)| \le \sum_{i=1}^{\chi(G)-1} d_{e_i'} + \sum_{i=1}^{\chi(G)} v_i'^w. \quad (11)$$

Proof: Firstly, we prove $|opt(G)| \le \sum_{i=1}^{\chi(G)-1} d_{e_i'} + \sum_{i=1}^{\chi(G)} v_i'^w$. To achieve this inequality, we just need to find proper spectrum assignments for all the vertices in $G(V,E)$ and the MUFI would not be bigger than $\sum\limits_{i=1}^{\chi(G)-1} d_{e_i'} + \sum\limits_{i=1}^{\chi(G)} v_i'^w$. Hence, we can first treat $C(G)$ as a conflict graph and find a proper spectrum assignment P^* for $C(G)$. Since each super-vertex $v_i' \in V_{C(G)}$ represents an independent set of G (*i.e.*, its weight is the maximum weight of the vertices in the independent set of G and edge $v_i'v_j' \in E_{C(G)}$ is the largest-weighted edge between the independent sets represented by v_i' and v_j'), we can utilize P^* to find a proper spectrum assignment for G by packing the vertices in v_i' into the FS set $w_{v_i'}$ (as shown in the example in Fig. 7). Therefore, if the MUFI of P^* does not exceed $\sum\limits_{i=1}^{\chi(G)-1} d_{e_i'} + \sum\limits_{i=1}^{\chi(G)} v_i'^w$, we prove the inequality. Here, the solution P^* can be obtained with *Algorithm* 1.

In *Algorithm* 1, we start from an arbitrary vertex in $C(G)$, *e.g.*, v_{i_1}', and get the FS set $w_{v_{i_1}'}$ by setting $v_{i_1}'^b = 1$, $v_{i_1}'^a = v_{i_1}'^w$. We select the largest-weighted incident edge of v_{i_1}', *e.g.*, $e_1'' = v_{i_1}'v_{i_2}'$, and the corresponding adjacent vertex v_{i_2}' is chosen as the next vertex. Then, we assign $w_{v_{i_2}'}$ by setting $v_{i_2}'^b = v_{i_1}'^a + d_{e_1''} + 1$, $v_{i_2}'^a = v_{i_2}'^b + v_{i_2}'^w - 1$. After that, we select the largest-weighted incident edge of v_{i_2}' to a vertex whose FS set has not been assigned. The same procedure is repeated until all the vertices in $C(G)$ are assigned with FS sets, and it terminates in $\chi(G) - 1$ steps.

The assignment P^* satisfies all the constraints in DSA, since we select the largest-weighted incident edge in each step.

Algorithm 1: Process Get P^*

Input : $C(G)$
Output: A proper spectrum assignment P^* for $C(G)$

$P^* \leftarrow \emptyset$;
$V'_C \leftarrow$ Random Select v'_{i_1}; % Let v'_{i_1} be Current Vertex
$V'^w_C \leftarrow v'^w_{i_1}$;
$V'^b_C \leftarrow 1$;
$V'^a_C \leftarrow V'^w_C$;
$P^* \leftarrow P^* \cup [V'^b_C, V'^a_C]$;
mark V'_C visited;
while $C(G)$ *still has unvisited vertices* **do**
  search the vertex v' which is the farthest neighbour of V'_C among the unvisited vertices in $C(G)$;
  set e'' as the edge linking v' and V'_C;
  $V'_N \leftarrow v'$; % Let v' be Next Vertex
  $V'^w_N \leftarrow v'^w$;
  $V'^b_N \leftarrow V'^a_C + d_{e''} + 1$;
  $V'^a_N \leftarrow V'^b_C + V'^w_N - 1$;
  $P^* \leftarrow P^* \cup [V'^b_N, V'^a_N]$;
  $V'_C \leftarrow V'_N$;
  mark V'_C visited;
return P^*

Hence, P^* is a proper spectrum assignment, and the MUFI of P^* equals $\sum_{i=1}^{\chi(G)-1} d_{e''_i} + \sum_{i=1}^{\chi(G)} v'^w_i \leq \sum_{i=1}^{\chi(G)-1} d_{e'_i} + \sum_{i=1}^{\chi(G)} v'^w_i$. By now, the inequality of the right side is proven.

Next, we prove the left side. As a maximal clique ψ is a subgraph of G, we have $|opt(\psi)| \leq |opt(G)|$. Hence, we just need to prove $|\text{MHP}(\psi)| + \psi^w \leq |opt(\psi)|$ for any ψ. If we assume that $P(\psi)$ is an optimal proper spectrum assignment for ψ, the FS sets assigned to all the vertices would be mutually disjoint since ψ is a complete subgraph. The distance between any two FS sets in $P(\psi)$ would not be smaller than the weight of the edge connecting the two vertices. Hence, the value of solution $P(\psi)$ would not be smaller than the length of the minimum Hamilton path plus the total weight of all the vertices, *i.e.*, $|\text{MHP}(\psi)| + \psi^w \leq |P(\psi)| = |opt(\psi)|$. Consequently, we finish the proof. ■

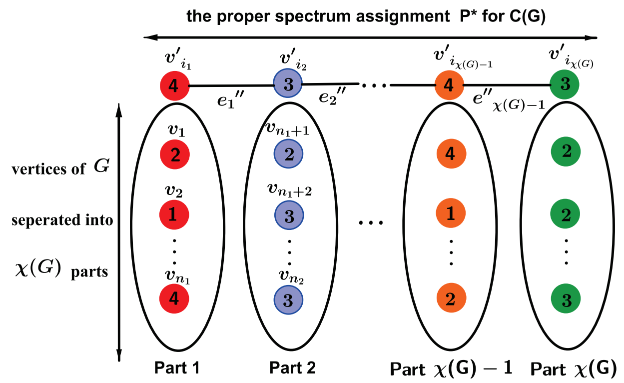


Fig. 7. A proper spectrum assignment for $C(G)$.

In general, it is known that calculating the chromatic number of a graph is extremely difficult. Hence, we provide a more practical method to calculate the bounds. For a graph G, we have $\chi(G) \leq \Delta(G) + 1$ according to the Brook's theorem [29], where $\Delta(G)$ is the maximum degree of G. With a DSA conflict graph $G(V, E, \{v_i^w\}, \{d_{v_i v_j}\})$, we sort the edges and vertices in G in the descending order of their weights, respectively. To avoid confusion, we rename the sorted edges and vertices by denoting the i-th largest-weighted edge as e_i^s and the vertex with the i-th biggest weight as v_i^s, *i.e.*, $d_{e_i^s} \geq d_{e_j^s}, v_i^{s^w} \geq v_j^{s^w}, \forall i < j, e_i^s, e_j^s \in E, v_i^s, v_j^s \in V$. Then, we have the following corollary.

Corollary 2: If $G(V, E, \{v_i^{s^w}\}, \{d_{e_i^s}\})$ is a DSA conflict graph, we have $|opt(G)| \leq \sum_{i=1}^{\Delta(G)} d_{e_i^s} + \sum_{i=1}^{\Delta(G)+1} v_i^{s^w}$.

Proof: Based on the construction procedure of $C(G)$ and the Brook's Theorem, we have $\sum_{i=1}^{\chi(G)-1} d_{e'_i} + \sum_{i=1}^{\chi(G)} v'^w_i \leq \sum_{i=1}^{\chi(G)-1} d_{e_i^s} + \sum_{i=1}^{\chi(G)} v_i^{s^w} \leq \sum_{i=1}^{\Delta(G)} d_{e_i^s} + \sum_{i=1}^{\Delta(G)+1} v_i^{s^w}$. Then, with Theorem 4, we can verify the proof. ■

In order to make a fast estimation for the bounds of the DSA's optimal solution, we can say that the MUFI would not exceed the total weight of $\Delta(G)$ largest-weighted edges plus the total weight of $\Delta(G)+1$ largest-weighted vertices in G.

Corollary 3: The two bounds obtained in Theorem 4 are tight.

Proof: $\chi(G)$ and ψ are vital for the two bounds. If $G(V, E)$ is a perfect graph [3], then the two bounds can converge under certain conditions. For instance, bipartite graphs are perfect graphs. For a bipartite graph $G(V, E)$, each $\psi(G)$ just contains one edge. As a result, the lower bound $\max_{\psi \in \Psi(G)} \{|\text{MHP}(\psi)| + \psi^w\}$ becomes $\max_{v_i u_j \in E} \{d_{v_i u_j} + v_i^w + u_j^w\}$. In this case, $|opt(G)|$ reaches this lower bound according to Theorem 3. Moreover, when $\chi(G) = 2$, the upper bound equals $\max_{e \in E}(d_e) + \max_{\forall v_i, \forall u_j} \{v_i^w + u_j^w\}$. When v_i^w and u_j^w are the weights of two adjacent largest-weighted vertices and $d_{v_i u_j}$ is also the maximum weight of the edges, the upper bound equals the lower bound. Hence, the two bounds are tight. ■

V. Ordered Distance Spectrum Assignment (ODSA)

In order to solve DSA efficiently, we simplify it to an ordered DSA (ODSA) problem, which we will prove that can be solved optimally in polynomial time. Basically, ODSA bears the same objective and constraints of DSA, and besides, it imposes a new **vertex order constraint** as follows. The vertices should be ordered such that the start-FS indices of vertices are in the ascending order, *i.e.*,

$$O_i = (v_{i_1}, v_{i_2}, ..., v_{i_n}) : v_{i_j}^b \geq v_{i_k}^b, \forall j > k \tag{12}$$

With the ordered vertices, ODSA becomes a much easier problem than DSA. We formulate the ILP model for ODSA:

$$\begin{aligned} &\text{Minimize } y \quad \text{(ILP-ODSA)}, \\ &s.t. \quad \text{Eqs. (14)-(18)}. \end{aligned} \tag{13}$$

[3]A graph G is perfect if $\chi(G) = \max_{\psi \in \Psi} |\psi(G)|$ [22].

$$x_{i_j}^a - x_{i_j}^b + 1 = v_{i_j}^w, \qquad \forall 1 \leq j \leq |V| \tag{14}$$

$$x_{i_j}^b - x_{i_k}^a \geq d_{v_{i_j} v_{i_k}} + 1, \qquad \forall j > k, v_{i_j} v_{i_k} \in E \tag{15}$$

$$x_{i_j}^b \geq x_{i_k}^b, \qquad \forall j > k \tag{16}$$

$$y \geq x_{i_j}^a, \qquad \forall 1 \leq j \leq |V| \tag{17}$$

$$x_{i_j}^a \in \mathbb{N}^+, x_{i_j}^b \in \mathbb{N}^+, \qquad \forall 1 \leq j \leq |V| \tag{18}$$

Then, we design a polynomial-time algorithm (O-L) to solve ODSA optimally, and its procedure is shown in *Algorithm* 2.

Algorithm 2: Procedure of O-L

Input : A DSA graph $G(V, E, \{v_i^w\}, \{d_{v_i v_j}\})$, and a vertex order $O_i = (v_{i_1}, v_{i_2}, ..., v_{i_n})$

Output: An assignment strategy for the star-index Seq $= \{v_{i_j}^b : 1 \leq j \leq |V|\}$ and the MUFI

$v_{i_1}^b \leftarrow 1$;

$v_{i_1}^a \leftarrow v_{i_1}^w$;

Seq $\leftarrow v_{i_1}^b$;

$j \leftarrow 2$;

while $j \leq |V|$ **do**

  $s_1 \leftarrow \max_{\forall k<j, v_{i_k} v_{i_j} \in E} \{v_{i_k}^a + d_{v_{i_k} v_{i_j}} + 1\}$;

  $s_2 \leftarrow v_{i_{j-1}}^b$;

  $v_{i_j}^b \leftarrow \max\{s_1, s_2\}$;

  $v_{i_j}^a \leftarrow v_{i_j}^b + v_{i_j}^w - 1$;

  Seq $\leftarrow$ Seq $\cup \{v_{i_j}^b\}$;

  $j \leftarrow j + 1$;

return Seq and $\max_{1 \leq j \leq |V|} (v_{i_j}^a)$

The main idea of *Algorithm* 2 is to assign the FS sets to vertices in sequence according to the pre-defined order such that $v_{i_j}^b$ takes the smallest possible value to satisfy all the constraints of ODSA. We begin with vertex v_{i_1}, and set its start-FS index $v_{i_1}^b$ as 1 and its end-FS index according to its bandwidth demand, *i.e.*, $v_{i_1}^a = v_{i_1}^w$. Then, $v_{i_1}^b$ is added to the spectrum assignment scheme Seq. For each vertex v_{i_j}, we use s_1 to denote the smallest index permitting to keep enough guard-bands from the adjacent vertices of v_{i_j} that have already been assigned FS sets, and use $s_2 = v_{i_{j-1}}^b$ to satisfy the order constraint. Then, the start-index of v_{i_j} is $\max\{s_1, s_2\}$, and the end-index equals $\max\{s_1, s_2\} + v_{i_j}^w - 1$. The procedure terminates when all the vertices have been assigned FS sets. The time complexity of *Algorithm* 2 is $\mathcal{O}(|E|)$.

Theorem 5: Algorithm 2 obtains an optimal FS assignment scheme for ODSA.

Proof: Firstly, it is easy to verify that the spectrum assignment scheme in Seq is a feasible solution for ODSA, since all the constraints are satisfied. Then, we prove that Seq indicates an optimal ODSA solution by contradiction. If Seq is not optimal, $opt = \{v_{i_j}^{b^{opt}} : 1 \leq j \leq |V|\}$ would be the optimal start-index arrangement for ODSA and $opt \neq$ Seq. Let j be the index of the first vertex such that $v_{i_j}^{b^{opt}} \neq v_{i_j}^b$ ($\because v_{i_1}^{b^{opt}} = v_{i_1}^b = 1 \Rightarrow j \geq 2$). Then, with the greedy strategy, we have $v_{i_j}^{b^{opt}} > v_{i_j}^b$. For $j+1$, let $\mathcal{F}_{j+1}^{opt}$ and $\mathcal{F}_{j+1}$ be the feasible region for $v_{i_{j+1}}^{b^{opt}}$ and $v_{i_{j+1}}^b$ in ILP-ODSA (*cf.*, Eq. (13)), respectively. The other constraints are the same for $v_{i_{j+1}}^{b^{opt}}$ and $v_{i_{j+1}}^b$ except for Eq. (16), *i.e.*, $v_{i_{j+1}}^{b^{opt}} \geq v_{i_j}^{b^{opt}}$ and $v_{i_{j+1}}^b \geq v_{i_j}^b$. As $v_{i_j}^{b^{opt}} > v_{i_j}^b$, the lower bound of $\mathcal{F}_{j+1}$ denoted as ζ_{j+1} would not be larger than that of $\mathcal{F}_{j+1}^{opt}$ denoted as ζ_{j+1}^{opt}. Since we can get $v_{i_{j+1}}^b = \zeta_{j+1}$ with the greedy strategy of *Algorithm* 2, we have $v_{i_{j+1}}^{b^{opt}} \geq \zeta_{j+1}^{opt} \geq \zeta_{j+1} = v_{i_{j+1}}^b$. By induction, we have $v_{i_k}^{b^{opt}} \geq v_{i_k}^b$, where k is within $[j, |V|]$. Therefore, we prove that the MUFI of ODSA under opt arrangement would not be smaller than that is provided by Seq, which causes the contradiction. Then, we finish the proof. ■

Corollary 4: A DSA problem can be solved optimally with *Algorithm* 2 under certain vertex order.

Proof: If opt is an optimal solution for a DSA problem, there has to be an order O_{opt} among the start-FS indices of opt. Therefore, the optimal solution for ODSA with vertex order O_{opt} equals that of the DSA problem. With Theorem 5, we prove that *Algorithm* 2 can get the optimal solution for the DSA problem under the order O_{opt}. ■

Now, we can see that it would be vital to determine the optimal vertex order. Note that, in the analysis above, we actually have already transformed the DSA problem into the permutation-based optimization problem (POP). POP is a classical combinational optimization [30]: Let S be a set of n elements, Σ be the permutation space that consists of $n!$ permutations over S, and $f(\cdot)$ be an estimation function for any $\sigma \in \Sigma$. The objective of POP is to optimize $f(\cdot)$ over Σ.

$$\sigma^* = arg \min_{\sigma \in \Sigma} f(\sigma). \tag{19}$$

For the DSA problem, S is vertex set V, Σ is the whole $|V|!$ vertex orders and we can utilize *Algorithm* 2 as our estimation function. In the next section, we will get an initial vertex order with a heuristic algorithm and then improve the vertex order with the nested partitions method (NPM) [31].

VI. Time-Efficient Approximation Algorithm for DSA

For any DSA problem, if the vertex order (*i.e.*, in the ascending order of the start-FS index) in the optimal solution is known beforehand, then it can be transformed into an ODSA problem and solved optimally by *Algorithm* 2 in polynomial time. Inspired by this, we develop a two-phase algorithm to solve DSA. Specifically, in the first phase, we use a greedy strategy to generate an initial vertex order, and then the second phase utilizes NPM to improve the initial order.

A. First Phase Greedy Algorithm (FPGA)

For a DSA conflict graph $G(V, E, \{v_i^w\}, \{d_{v_i v_j}\})$, we can get the initial vertex order with the following procedure. Firstly, we start from any vertex $v_i \in V$, and find the FS set for v_i with a greedy strategy, *i.e.*, $v_i^b = 1$ and $v_i^a = v_i^w$. Meanwhile, we set a variable O_i to record the order of vertices according to the assigned FS sets. Hence, O_i takes v_i as the first element. Then, we find the vertex v_j from the vertices that are not yet in O_i to ensure that v_j^b is the minimum to satisfy the constraints of DSA for all the vertices that are in O_i. We insert this v_j into O_i and assign the corresponding FS set to it. The same procedure is repeated until all the vertices have been included into O_i. After $|V|$ while-loops, $|V|$ vertex orders $\{O_1, O_2, ..., O_{|V|}\}$ have been generated and we choose the one that results in the minimum MUFI as our initial order. *Algorithm* 3 gives the procedure of the proposed First Phase Greedy Algorithm (FPGA). In *Lines* 1-3, starting from $j = 1$, we initialize O_j as $\emptyset$ and use s_j to record the current MUFI used in O_j, whose initial value is 0. Then, in *Lines* 4-20, with the $|V|$ while loops, we generate $|V|$ vertex orders. As mentioned above, *Lines* 5-8 let v_j be added into O_j, assign the FS set to it, and update s_j as $s_j = v_j^w$. In the for-loop covering *Lines* 9-20, we organize the remaining vertices for O_j one by one using the aforementioned greedy strategy. Finally, we select the vertex order that results in the minimum MUFI. We can see that there are three cascading loops in *Algorithm* 3, and thus its time complex is $\mathcal{O}(|V|^3 \cdot \Delta)$, where $|V|$ is the number of vertices and Δ is the maximum degree of G.

Algorithm 3: Procedure of FPGA

Input : $G(V, E, \{v_i^w\}, \{d_{v_i v_j}\})$
Output: An initial vertex order and an initial MUFI

1 $j \leftarrow 1$;
2 $O_j \leftarrow \emptyset$; % initialize vertex order O_1
3 $s_j \leftarrow 0$; % record the MUFI of O_1
4 **while** $j \leq |V|$ **do**
5     $O_j \leftarrow O_j \cup \{v_j\}$;
6     $v_j^b \leftarrow 1$;
7     $v_j^a \leftarrow v_j^w$;
8     $s_j \leftarrow v_j^w$;
9   **for** $i = 2 : |V|$ **do**
10         $v \leftarrow \emptyset$; % v is the next vertex entering O_j
11         $v^b \leftarrow B$; % B is large enough
12         $v^w \leftarrow 0$;
13   **for** $k = 1 : |V|$ **do**
14   **if** $v_k \notin O_j$ **then**
15                 $v_k^b \leftarrow \max\limits_{\forall v_l \in O_j, v_k v_l \in E} \{v_l^a + d_{v_k v_l} + 1\}$;
16   **if** $v_k^b < v^b$ **then**
17                     $v \leftarrow v_k$; $v^b \leftarrow v_k^b$; $v^w \leftarrow v_k^w$;
18         $O_j \leftarrow O_j \cup \{v\}$;
19         $v^a \leftarrow v^b + v^w - 1$;
20         $s_j \leftarrow \max\{s_j, v^a\}$;
21     $j \leftarrow j + 1$; $O_j \leftarrow \emptyset$; $s_j \leftarrow 0$;
22 $O^* = \underset{1 \leq j \leq |V|}{\underset{O_j}{argmin}} \; s_j$; $s^* = \underset{1 \leq j \leq |V|}{\underset{s_j}{argmin}} \; s_j$;

After getting the initial vertex order O^* and initial MUFI value s^*, we utilize NPM to improve the initial solution. In the next subsection, we will provide the details of NPM and our two-phase algorithm.

B. Two-phase Algorithm

The NPM method was proposed in [31] to leverage a general random method to solve global optimization problems, which includes POP. Specifically, we consider the following problem

$$\theta^* = arg \min_{\theta \in \Theta} f(\theta), \tag{20}$$

where Θ is the entire solution space and $f(\cdot) : \Theta \rightarrow \mathbb{R}$ is the objective function. Firstly, NPM gives a *partitioning scheme* to partition Θ systematically, and then it uses a iterative approach to optimize $f(\cdot)$. In each iteration, NPM operates on a solution space η, which is a subset of Θ from the partitioning scheme and is named as *the most promising* region. Then, according to the partitioning scheme, we divide the most promising region η into $M(\eta)$ disjoint subregions, and we call $\Theta \backslash \eta$ *surrounding region*. Note that, if the partition scheme obtains a region, then we say the region is *valid*, and if a valid region σ is formed by partitioning a valid region η, then σ is a *subregion* of η and η is called the *superregion* of σ. Therefore, η is divided into $M(\eta)$ disjoint subregions. Next, each of the $M(\eta)$ subregions and the surrounding region are sampled by a *random sampling scheme* and we use the objective function to evaluate the samples and calculate the *promising index* for each subregion. If the promising index of a subregion among the $M(\eta)$ subregions of η turns to be the best one, we set this subregion as the most promising region in the next iteration. If the surrounding region is proven to be the best, the method will *backtrack* to another region to be the next most promising region (*e.g.*, a region that contains the previous most promising region or a subregion of Θ that contains the best sample). The most promising region is then partitioned and sampled with the procedure discussed above.

For DSA, the entire solution space Θ is the $|V|!$ vertex orders and the objective function is *Algorithm* 2. In the second phase of the time-efficient approximation algorithm for DSA, the partitioning scheme is as follows: we first divide Θ into n disjoint subregions by choosing $v_1, v_2, ..., v_{|V|}$ as the first vertex in the ordered vertices, and then each of the $|V|$ subregions is divided into $|V| - 1$ subregions by selecting the second vertex and so on so forth. Fig. 8 provides an illustrative example on the partitioning scheme for DSA. The random sampling scheme samples the surrounding region and each subregion uniformly and the most promising region will backtrack to the least supperregion if the promising index in the surrounding region is the best. The vertex order O^* obtained by *Algorithm* 3 is the original most promising region.

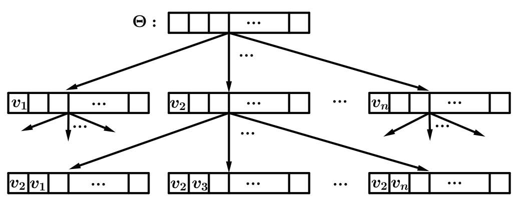


Fig. 8. Example on the partitioning scheme for DSA.

VII. Algorithm Analysis

In this section, we analyze the performance of the two-phase algorithm, which is composited by *Algorithm* 3 (*i.e.*, an approximation algorithm) and NPM (*i.e.*, a random optimization algorithm). For *Algorithm* 3, as DSA is intractable according to Theorem 2, we focus the analysis on some specific graph types, *e.g.*, complete graphs and bipartite graphs. For NPM, we provide two of its key properties, *i.e.*, the convergence performance and the number of expected iterations.

A. Approximate Ratio of FPGA in Special Graphs

1) Complete Graph with Triangle Inequality: If a DSA conflict graph $G(V, E)$ is a complete graph, the FS sets assigned to the vertices must be pairwise disjoint. Hence, to satisfy the bandwidth requirement and spectrum contiguity constraints, the union of the FS sets assigned to the vertices has a fixed cardinality, denoted as $V^w = \sum_{i=1}^{n} v_i^w$. Consequently, the optimization objective in this case is equivalent to minimize the sizes of the guard-band between any two spectrally adjacent FS sets under the spectrum set distance constraint.

An algorithm, called *Nearest Neighbor* (NN) to solve MHP, can guarantee an approximate ratio for complete conflict graphs that satisfy the triangle inequality. *Algorithm* 4 shows the procedure of NN.

Algorithm 4: Procedure of NN

Input : $G(V, E), v_i, \{d_{v_i v_j}\}$
Output: A Hamilton path P
set CurrentVertex $\leftarrow v_i$;
mark v_i visited;
while *G still has unvisited vertices* **do**
    find vertex v which is the nearest neighbor to CurrentVertex among all the unvisited vertices in G;
    CurrentVertex $\leftarrow v$;
    mark CurrentVertex visited;
    $P \leftarrow P \cup \{v\}$;
return P

Let $|NN(G)|$ denote the length of the Hamilton path produced by *Algorithm* 4 and $|MHP(G)|$ denote the length of the minimum Hamilton path. Then according to [27, 32], we have the approximate ratio $\dfrac{|NN(G)|}{|MHP(G)|} \leq \dfrac{1}{2}(\lceil \log_2(|V|) \rceil + 1)$. For a complete DSA conflict graph $G(V, E, \{v_i^w\}, \{d_{v_i v_j}\})$ that satisfies the triangle inequality, we apply *Algorithm* 3 to G. Note that, the while-loop in *Algorithm* 3 obtains a vertex order O_j in the j-th iteration. There is a proper spectrum assignment induced by O_j, which in fact represents a Hamilton path in G. Then, we have Lemma 1.

Lemma 1: If the conflict graph G is a complete graph that satisfies the triangular inequality, the Hamilton path induced by the order O_j from *Algorithm* 3 is equivalent to the result from *Algorithm* 4 with input v_j.

Proof: We assume that the order O_j obtained in the j-th while-loop of *Algorithm* 3 is $(v_{j_1}, v_{j_2}, ..., v_{j_n})$, where $v_{j_1} = v_j$. At first we have $|O_j| = 1$, which means that only v_{j_1} is included in order O_j. Then, with the greedy strategy of *Algorithm* 3, v_{j_2} is the nearest neighbor to v_{j_1} in G. Supposing this inference is true when $|O_j| = k$, where $k < |V|$, we assert $v_{j_{k+1}}$ is the nearest neighbor of v_{j_k} among those vertices that are not yet in O_j. After we have included the first k vertices in O_j, the innermost for-loop of *Algorithm* 3 searches the $(k+1)$-th vertex *i.e.*, $v_{j_{k+1}}$, whose FS start-index is the smallest among those unordered vertices. We use v_l to denote the nearest neighbor of v_{j_k} among all the unordered vertices. Since the triangle inequality is held, the spectrum set distance constraint for $v_{j_{k+1}}$ only comes from v_{j_k}, *i.e.*, $v_{j_{k+1}}^b = v_{j_k}^a + d_{v_{j_k} v_{j_{k+1}}} + 1$. As $d_{v_{j_k} v_l}$ is the smallest guard-band size, $v_{j_{k+1}}^b = v_{j_k}^a + d_{v_{j_k} v_l} + 1$ reaches the minimum. Therefore, using the greedy strategy, we can get $v_{j_{k+1}} = v_l$ and the proof is verified. ■

In fact, we select the minimum one from $\{O_1, O_2, ..., O_n\}$ after $|V|$ while-loops in *Algorithm* 3. Let $|FPGA(G)|$ be the finial output value and $|opt(G)|$ be the optimal value for a DSA conflict graph G. Then, according to Lemma 1 and the analysis above, $|FPGA(G)| - V^w$ and $|opt(G)| - V^w$ are the length of the Hamilton path produced by *Algorithm* 4 and $|MHP(G)|$ respectively. Then, we get the following theorem.

Theorem 6: If $G(V, E, \{v_i^w\}, \{d_{v_i v_j}\})$ is a complete DSA conflict graph that satisfies the triangle inequality, the approximate ratio of *Algorithm* 3 would not be larger than $\dfrac{1}{2}(\lceil \log_2(|V|) \rceil + 1)$.[4]

Proof: According to the analysis above, we have $\dfrac{|FPGA(G)| - V^w}{|opt(G)| - V^w} \leq \dfrac{1}{2}(\lceil \log_2(|V|) \rceil + 1)$. As $|FPGA(G)| \geq |opt(G)|$, we have $\dfrac{|FPGA(G)|}{|opt(G)|} \leq \dfrac{|FPGA(G)| - V^w}{|opt(G)| - V^w} \leq \dfrac{1}{2}(\lceil \log_2(|V|) \rceil + 1)$. ■

2) Bipartite Graphs: Then, we consider the case in which the DSA conflict graph is a bipartite graph. Before the analysis, we introduce the following definition.

Definition 1: For a bipartite graph $G(V_1, V_2)$, V_1 and V_2 are the two parts of the vertices in G. We call **its vertex labeling is good** if the vertices are labeled in the way that the vertices in V_1 are labeled as the first $|V_1|$ ones, *i.e.*, $\{v_1, v_2, ..., v_{|V_1|}\} = V_1$, and apparently, the remaining vertices are all in V_2 and labeled as $\{v_{|V_1|+1}, v_{|V_1|+2}, ..., v_{|V|}\} = V_2$. For a bipartite graph $G(V_1, V_2)$, the time needed to get a good vertex labeling is $\mathcal{O}(|E|)$.

Theorem 7: If a DSA conflict graph $G(V, E, \{v_i^w\}, \{d_{v_i v_j}\})$ is a bipartite graph and we label its vertices in a good way, *Algorithm* 3 can get the optimal solution for DSA.

Proof: Let V_1 and V_2 be the two parts of a bipartite V. According to *Algorithm* 3 and Theorem 3, we just need to prove the MUFI obtained with order O_1 in *Algorithm* 3 equals $\max\limits_{v_i v_j \in E}\{d_{v_i v_j} + v_i^w + v_j^w\}$. After v_1 has entered O_1, since V_1 is an independent set, *Algorithm* 3 includes vertices $v_2, ..., v_{|V_1|}$ in O_1 in sequence and $v_i^b = 1, 1 \leq i \leq |V_1|$. Also, because V_2 is an independent set, $v_i^b = \max\limits_{\forall v_j \in V_1, v_i v_j \in E}\{v_j^a + d_{v_i v_j} + 1\}, |V_1| + 1 \leq i \leq |V|$. Therefore, considering

[4]Actually, for this special case, *Double Minimum Spanning Tree* algorithm [33] of MHP can be utilized for DSA, which can guarantee a 2-approximation ratio with the similar proof of Theorem 6.

the four constraints of DSA, we get the MUFI of O_1 as $\max_{v_i v_j \in E}\{d_{v_i v_j} + v_i^w + v_j^w\}$. ■

B. Convergence Performance and Expected Number of Iterations of Two-phase Algorithm

1) Convergence Performance: For *Algorithm* 3, we assume that the partitioning scheme has been defined and let Σ denote the set of all the valid regions, where $\sigma(0)$ is the initial region state, *i.e.*, the initial vertex order that is obtained, and $\sigma(k) \in \Sigma$ is the region state of the k-th iteration. Then, $\{\sigma(k)\}_{k=0}^{\infty}$ is the iteration sequence and the region state $\sigma(k+1)$ depends on the estimated values of the promising index in the state $\sigma(k)$, which is related with the sampling points. Therefore, $\{\sigma(k)\}_{k=0}^{\infty}$ is a Markov chain with state space Σ, and we have Theorem 8 according to [31].

Theorem 8: $\eta \in \Sigma$ is an absorbing state of the Markov chain $\{\sigma(k)\}_{k=0}^{\infty}$, **if and only if** η is the optimal vertex order for DSA.

Proof: Firstly, we prove the "if" part and use Algorithm 2 as the object function $f(\cdot)$ to evaluate the promising index of a region. If we assume that η is the optimal vertex order for DSA, then the transition probability of staying in η is: $P_{\eta\eta} = P[f(\eta) \leq f(\Theta\backslash\eta)] = 1$. Hence, η is an absorbing state. Next, we prove the reverse. Supposing ξ is an absorbing state and ξ does not represent the optimal order for DSA, the transition probability of not staying in ξ is: $P_{\xi\Theta\backslash\xi}$ = $P[f(\xi) > f(\Theta\backslash\xi)]$ $\geq P$[randomly select a point θ in $\Theta\backslash\xi$ and $f(\theta) < f(\xi)] > 0$. This inequality reveals that ξ is a transient state, which leads to a contradiction. Therefore, we finish the proof. ■

According to Theorem 8, the Markov chain will eventually converge to an optimal vertex order and stay there forever. Since the transient states are finite, we can see that the Markov chain would reach an optimal vertex order within finite time.

2) Expected Number of Iterations: The expected number of iterations to reach the optimal vertex order directly impacts the time-efficiency of our two-phase algorithm. To evaluate the expected number of iterations, we need to introduce several random variables and symbols [34]. We use Σ to represent the state space, σ_{opt} to represent the optimal solution regions, *i.e.*, the optimal vertex order. We define Σ_1 = $\{\eta \in \Sigma\backslash\{\sigma_{opt}\} | \sigma_{opt} \in \eta\}$, *i.e.*, the valid regions that include σ_{opt} and Σ_2 = $\{\eta \in \Sigma\backslash\{\sigma_{opt}\} | \sigma_{opt} \notin \eta\}$, *i.e.*, the valid regions that do not include σ_{opt}. Then, we have $\Sigma = \{\sigma_{opt}\} \cup \Sigma_1 \cup \Sigma_2$. We use Y_η to denote the number of visits of a state $\eta \in \Sigma$ and use T_η to represent its hitting time (the first time of visiting this state). Besides, we denote the probability of an event under constraint that the chain starts in a state $\eta \in \Sigma$ as P_η[event].

According to [34], the number of iterations for the Markov chain to reach an absorbing state Y equals the number of iterations to visit all the transient states plus one (*i.e.*, the transition to the absorbing state), which is $Y = 1 + \sum_{\eta\in\Sigma_1} Y_\eta + \sum_{\eta\in\Sigma_2} Y_\eta$. As Σ is finite, we get the expected number of iterations as

$$\mathbf{E}[Y] = 1 + \sum_{\eta\in\Sigma_1} \mathbf{E}[Y_\eta] + \sum_{\eta\in\Sigma_2} \mathbf{E}[Y_\eta]. \tag{21}$$

Theorem 9: Let $\sigma(0)$ be the initial vertex order provided by *Algorithm* 3. The expected number of iterations for our two-phase algorithm to get the optimal solution for DSA is

$$\begin{aligned}\mathbf{E}[Y] = 1 &+ \sum_{\eta\in\Sigma_1} \frac{1}{P_\eta[T_{\sigma_{opt}} < T_\eta]} \\ &+ \sum_{\eta\in\Sigma_1} \frac{P_{\sigma(0)}[T_\eta < \min\{T_{\sigma(0)}, T_{\sigma_{opt}}\}]}{P_\eta[T_{\sigma(0)} < T_\eta] \cdot P_{\sigma(0)}[T_{\sigma_{opt}} < \min\{T_{\sigma(0)}, T_\eta\}]}.\end{aligned} \tag{22}$$

Proof: As given in [31], the expected number of visits to the transient states is

$$\mathbf{E}[Y_\eta] = \begin{cases} \dfrac{1}{P_\eta[T_{\sigma_{opt}} < T_\eta]}, \eta \in \Sigma_1, \\ \dfrac{P_{\sigma(0)}[T_\eta < \min\{T_{\sigma(0)}, T_{\sigma_{opt}}\}]}{P_\eta[T_{\sigma(0)} < T_\eta] \cdot P_{\sigma(0)}[T_{\sigma_{opt}} < \min\{T_{\sigma(0)}, T_\eta\}]}, \eta \in \Sigma_2. \end{cases} \tag{23}$$

By substituting Eq. (23) in Eq. (21), we finish the proof. ■

In each iteration, we at most take n sampling points in the n valid regions. Each sampling and calculating of the promising index will use the Procedure O-L whose time complexity is $\mathcal{O}(|E|)$. Therefore, the expected time complexity for the second phase is $\mathcal{O}(|V| \cdot |E| \cdot \mathbf{E}(Y))$.

Although we have Theorem 9, calculating the expected number is still tough. Hence, we leverage the approximation stochastic model in [34]. Specifically, in each iteration, if the promising index of the surrounding region is the best, we backtrack to the entire solution space Θ. Let P_0 be the the probability of the two-phase algorithm moving towards the correct direction, *i.e.*, backtracking if the optimal solution is not in the current most promising region and selecting the correct subregion otherwise. Then, we have Theorem 10.

Theorem 10: Assuming the above approximation stochastic model is held, the expected number of iterations for two-phase algorithm to find the optimal solution for DSA is

$$\begin{aligned}\mathbf{E}(|Y|) &= \frac{1}{P_0^n}(1 - \frac{(1-P_0)^n}{n!}) \\ &- (\sum_{d=0}^{n-2} \frac{(n-d)!}{n!} \cdot \frac{(1-P_0)^d}{P_0^{n-1}}) + (\frac{1}{P_0^{n-1}} \cdot \frac{P_0 - P_0^n}{1-P_0}),\end{aligned} \tag{24}$$

where $n = |V|$ is the number of vertices in G.

Proof: Theorem 10 can be proved using the similar procedure that proves Theorem 2 in [34]. ■

With the approximate expected number, we can set the stopping criteria to terminate the two-phase algorithm under certain probability significance. We utilize the expected number in Eq. (24) and apply the Markov inequality: $P(|Y| \geq \varepsilon) \leq \frac{1}{\varepsilon^\alpha}\mathbf{E}(|Y|)^\alpha$ to get the upper bound of the number of iterations for finding the optimal solution for DSA.

VIII. Numerical Results

In this section, we evaluate the performance of our proposed two-phase algorithm. As DSA is a new spectrum assignment model, there is no existing heuristic algorithm for comparison. Hence, we applied Pure Random Algorithm (PRA) as the benchmark algorithm, in which we randomly selected a vertex order at each iteration and calculate the optimal solution for this vertex order by using *Algorithm* 2. The ILP model for DSA was solved by MATLAB2015a with the CPLEX toolbox and the approximate solutions from our two-phase algorithm and PRA were both obtained with MATLAB2015a under the same number of iterations. We run 30 independent simulations on each conflict graph and average the results to ensure sufficient statistical accuracy. We set the probability of moving in the correct direction as $P_0 = 0.5$ in Eq. (24) and the significance probability as 90%. All the simulations run on a computer with 3.2 GHz Intel(R) Core(TM) i5-4690S CPU and 8 GBytes RAM.

A. Simulation Setup

We perform simulations in different scenarios:

- **Random graphs**: We use the NetworkX package [35] to generate random graphs, in which each vertex pair is directly connected with a probability of 0.5, as DSA conflict graphs. The weights of vertices and edges are randomly chosen within $[1, |V|]$. Specifically, Fig. 9 shows the six random graphs that are used in the simulations. They have $|V| \in [14, 19]$. Hence, we assessed the performance of *Algorithm* 3 and our two-phase algorithm under the pure random conditions.
- **Complete graphs with random weights**: To reveal the effectiveness of the two-phase algorithm, we also use complete conflict graphs with $|V| \in [14, 19]$, whose vertex and edge weights were also randomly chosen within $[1, |V|]$, as the DSA conflict graphs.
- **Edge number**: By intuition, the more edges or the larger the biggest guard-band size that a conflict graph has, the bigger its MUFI is. Therefore, we apply the two-phase algorithm on six random conflict graphs, each of which has 14 vertices and the number of edges ranges within {15, 30, 45, 60, 75, 90} as shown in Fig. 10. The vertex and edge weights are still chosen randomly as above.
- **14-node NSFNET and 28-node US Backbone**: To mimic the realistic situations, we run simulations on two practical EON topologies, i.e., the 14-node NSFNET and the 28-node US Backbone [13]. Here, each lightpath request is randomly generated and we use the shortest path to route it. The guard-band requirement between two lightpaths is computed as the number of common links on their routing paths. Following these principles, DSA conflict graphs are constructed and we applied the two-phase algorithm to solve the DSA problems.

B. Simulation Results

1) Random Graphs: Table III presents the average MUFI computed by PRA, FPGA, two-phase and ILP-DSA, respectively for the six random topologies in Fig. 9. The relative gaps (errors-optimal ratios) with a 95% confidence interval are shown in Fig. 11. In Table III, both the initial solutions from FPGA and the improved solutions from the two-phase algorithm are better than those from PRA under the same number of iterations. We also observe that the solutions are truly improved in the second phase, since the MUFI from the two-phase algorithm are closer to the optimal one obtained from FPGA, as shown in Fig. 11. Another notable fact is that the results of Fig. 9(b) are better than those in Fig. 9(a). We observe that there is a vertex with degree one in the topology of Fig. 9(b), which is different from Fig. 9(a). This fact implies that the topology does have impact on the final MUFI.

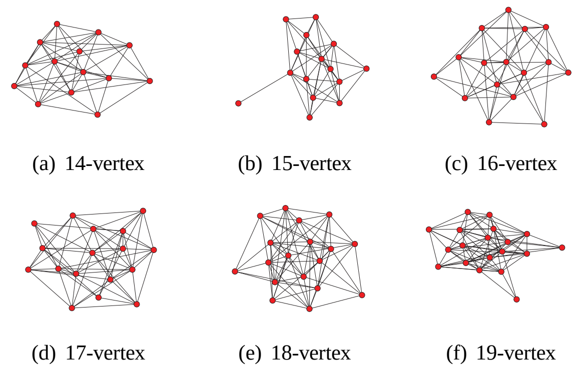


Fig. 9. Six random graphs with 14-19 vertices.

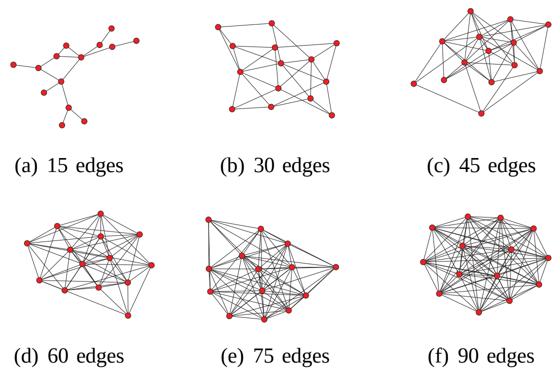


Fig. 10. Six random graphs with 14 vertices and 15-90 edges.

TABLE III
NUMERICAL RESULTS FOR FIG. 9

Fig. 9	(a)	(b)	(c)	(d)	(e)	(f)
PRA	95.2	92.8	104.	128.	147.	160.
FPGA	75.4	75.3	77.4	96.5	101.	111.
Two-phase	72.7	72.5	75.5	91.0	99.4	110.
ILP-DSA	71.6	70.1	73.6	87.5	94.5	105.

2) Random Complete Graphs: Table IV presents the average MUFI obtained in the six random complete graphs. The relative gaps with a 95% confidence interval are shown in Fig. 12. We can observe the similar trends as discussed above for random conflict graphs. Moreover, we can see that both the relative gaps and the confidence intervals in complete graphs are smaller than those in random graphs for two-phase, FPGA and PRA. This can be interpreted as follows. In complete graphs, the FS set assigned to each vertex should be mutually disjoint, which makes the optimal MUFI (computed by the ILP) bigger. While in random graphs, the FS sets assigned to certain vertices could be overlapped, and hence the optimal value of MUFI would be smaller. However, the overlapped FS sets make it more difficult for the three algorithms to optimize the spectrum assignment, which leads to smaller relative gaps

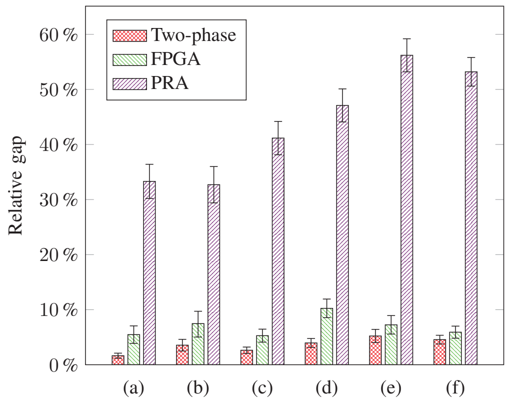


Fig. 11. Relative gaps of Table III by Two-phase, FPGA and PRA.

and confidence intervals in complete graphs.

TABLE IV
NUMERICAL RESULTS FOR RANDOM COMPLETE GRAPHS

# vertices	14	15	16	17	18	19
PRA	169.0	194.1	238.7	259.1	283.3	297.3
FPGA	145.0	164.7	197.4	216.5	234.4	246.1
Two-phase	143.6	163.6	196.6	213.3	231.3	241.5
ILP-DSA	142.4	160.5	191.1	207.6	223.7	231.8

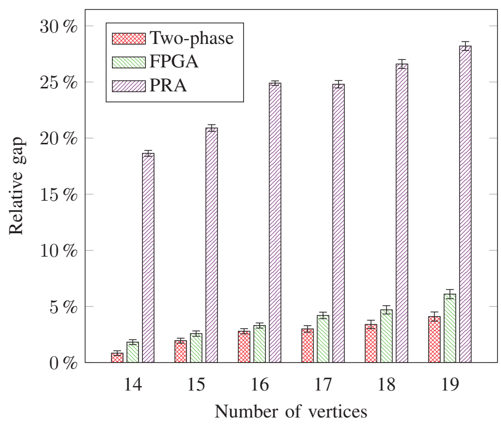


Fig. 12. Relative gaps of Table IV by Two-phase, FPGA and PRA.

3) Edge number: Fig. 13 plots the simulation results on six random graphs in Fig. 10. The results on MUFI from two-phase algorithm and ILP-DSA are marked as purple and blue bars respectively, and the approximate ratio is plotted in red line. It can be seen that the approximate ratio of the two-phase algorithm increases with the number of edges in the conflict graph.

These results coincided well with the intuitive observation that the more edges or the bigger edge weights that a graph has, the more spectrum resources that DSA would consume. The feature also inspires us that a good routing algorithm should be used to reduce the common links and thus further improve the quality of the results for DSA in EONs.

4) 14-vertex NSFNET and the 28-vertex US Backbone: We evaluate the performance of two-phase algorithm with two

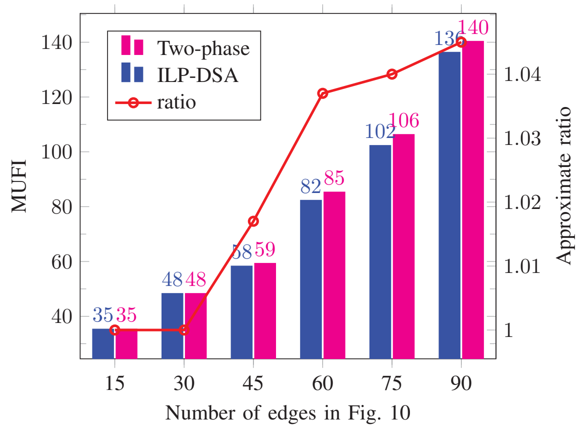


Fig. 13. Numerical results for Edge number scenario.

practical EON topologies. In Table V, we can see that ILP-DSA can only get the optimal solution when the number of lightpaths is within 50. Meanwhile, our two-phase algorithm can obtain almost the same solutions as ILP-DSA.

Based on all these observations, we can conclude that our proposed two-phase algorithm can approximate the optimal solution for DSA well.

IX. CONCLUSIONS

In this paper, we studied the DSA problem in EONs. By reducing MHP and graph coloring to DSA, we have proven that DSA is $\mathcal{NP}$-hard and inapproximable. Then, we analyzed and provided the upper and lower bounds for the optimal solutions of DSA, and proved that they are tight. Next, by leveraging a vertex order and developing a polynomial-time algorithm (*i.e.*, *Algorithm* 2), we transformed DSA into POP. Then, we developed a two-phase algorithm to solve DSA time-efficiently. For the first phase (*i.e.*, *Algorithm* 3) in the algorithm, we theoretically proved that its time complexity is $\mathcal{O}(|V|^3 \cdot \Delta)$, and it can get the optimal solution for bipartite conflict graphs and guarantee an approximate ratio of $\mathcal{O}(\log(|V|))$ for complete conflict graphs with triangle inequality. The second phase utilized a random optimization algorithm, and we applied theoretical analysis to obtain the expected number of iterations for getting the optimal solution. The numerical simulation results demonstrated that our two-phase algorithm can find the near-optimal solutions for DSA in various conflict graphs.

ACKNOWLEDGMENTS

The preliminary version of this paper has been published as a post-deadline paper [36] in the proceedings of the 21st European Conference on Networks and Optical Communications (NOC 2016).

REFERENCES

[1] M. Jinno *et al.*, "Spectrum-efficient and scalable elastic optical path network: architecture, benefits, and enabling technologies," *IEEE Commun. Mag.*, vol. 47, pp. 66–73, Nov. 2009.

TABLE V
SIMULATION RESULTS FOR EONS WITH NSFNET AND US BACKBONE TOPOLOGIES

	# requests	**ILP-DSA**	**Two-phase**	**PRA**		# requests	**ILP-DSA**	**Two-phase**	**PRA**
NSFNET	10	29	29	29	US Backbone	10	33	33	33
	20	72	72	76		30	186	189	197
	30	153	153	177		50	351	363	462
	40	200	201	252		100	—	1339	1898
	50	420	423	500		150	—	2843	4666
	60	—	469	602		200	—	3784	7743
	70	—	598	805		250	—	6347	13020
	80	—	890	1155		300	—	8140	17303

[2] Z. Zhu *et al.*, "Dynamic service provisioning in elastic optical networks with hybrid single-/multi-path routing," *J. Lightw. Technol.*, vol. 31, pp. 15–22, Jan. 2013.

[3] P. Lu *et al.*, "Highly efficient data migration and backup for Big Data applications in elastic optical inter-data-center networks," *IEEE Netw.*, vol. 29, pp. 36–42, Sept./Oct. 2015.

[4] Y. Wang *et al.*, "Towards elastic and fine-granular bandwidth allocation in spectrum-sliced optical networks," *J. Opt. Commun. Netw.*, vol. 4, pp. 906–917, Nov. 2012.

[5] F. Ji *et al.*, "Dynamic p-cycle protection in spectrum-sliced elastic optical networks," *J. Lightw. Technol.*, vol. 32, pp. 1190–1199, Mar. 2014.

[6] M. Médard *et al.*, "Security issues in all-optical networks," *IEEE Netw.*, vol. 11, pp. 42–48, Jun. 1997.

[7] J. Zhu *et al.*, "Attack-aware service provisioning to enhance physical-layer security in multi-domain EONs," *J. Lightw. Technol.*, vol. 34, pp. 2645–2655, Jun. 2016.

[8] J. Zhu, B. Zhao, and Z. Zhu, "Leveraging game theory to achieve efficient attack-aware service provisioning in EONs," *J. Lightw. Technol., in Press*, 2017.

[9] B. Kozicki *et al.*, "Filtering characteristics of highly-spectrum efficient spectrum-sliced elastic optical path (SLICE) network," in *Proc. of OFC 2009*, pp. 1–3, Mar. 2009.

[10] M. Fok *et al.*, "Optical layer security in fiber-optic networks," *IEEE Trans. Inf. Forensics Security*, vol. 6, pp. 725–736, Sept. 2011.

[11] W. Shi *et al.*, "On the effect of bandwidth fragmentation on blocking probability in elastic optical networks," *IEEE Trans. Commun.*, vol. 61, pp. 2970–2978, Jul. 2013.

[12] M. Zhang, C. You, and Z. Zhu, "On the parallelization of spectrum defragmentation reconfigurations in elastic optical networks," *IEEE/ACM Trans. Netw.*, vol. 24, pp. 2819–2833, Oct. 2016.

[13] L. Gong *et al.*, "Efficient resource allocation for all-optical multicasting over spectrum-sliced elastic optical networks," *J. Opt. Commun. Netw.*, vol. 5, pp. 836–847, Aug. 2013.

[14] X. Zhou *et al.*, "Dynamic RMSA in elastic optical networks with an adaptive genetic algorithm," in *Proc. of GLOBECOM 2012*, pp. 2912–2917, Dec. 2012.

[15] S. Ma *et al.*, "Demonstration of online spectrum defragmentation enabled by OpenFlow in software-defined elastic optical networks," in *Proc. of OFC 2014*, pp. 1–3, Mar. 2014.

[16] N. Skorin-Kapov, J. Chen, and L. Wosinska, "A new approach to optical networks security: Attack-aware routing and wavelength assignment," *IEEE/ACM Trans. Netw.*, vol. 18, pp. 750–760, Jun. 2010.

[17] N. Skorin-Kapov *et al.*, "Physical-layer security in evolving optical networks," *IEEE Commun. Mag.*, vol. 54, pp. 110–117, Aug. 2016.

[18] X. Liu, L. Gong, and Z. Zhu, "On the spectrum-efficient overlay multicast in elastic optical networks built with multicast-incapable switches," *IEEE Commun. Lett.*, vol. 17, pp. 1860–1863, Sept. 2013.

[19] W. Lu and Z. Zhu, "Dynamic service provisioning of advance reservation requests in elastic optical networks," *J. Lightw. Technol.*, vol. 31, pp. 1621–1627, May 2013.

[20] M. Zhang *et al.*, "Dynamic and adaptive bandwidth defragmentation in spectrum-sliced elastic optical networks with time-varying traffic," *J. Lightw. Technol.*, vol. 32, pp. 1014–1023, Mar. 2014.

[21] L. Gong and Z. Zhu, "Virtual optical network embedding (VONE) over elastic optical networks," *J. Lightw. Technol.*, vol. 32, pp. 450–460, Feb. 2014.

[22] A. Bondy and U. Murty, *Graph Theory*. Springer-Verlag London, 2008.

[23] D. Banerjee and B. Mukherjee, "A practical approach for routing and wavelength assignment in large wavelength-routed optical networks," *IEEE J. Sel. Areas Commun.*, vol. 14, pp. 903–908, Jun. 1996.

[24] G. Wilfong and P. Winkler, "Ring routing and wavelength translation," in *Proc. ACM-SIAM SODA 1988*, pp. 333–341, 1988.

[25] S. Shirazipourazad *et al.*, "On routing and spectrum allocation in spectrum-sliced optical networks," in *Proc. INFOCOM 2013*, pp. 385–389, Apr. 2013.

[26] E. Scheinerman and D. Ullman, *Fractional Graph Theory: A Rational Approach to the Theory of Graphs*. Dover Publications, 2013.

[27] T. Boffey, "A note on minimal length hamilton path and circuit algorithms," *Opera. Res.*, vol. 24, pp. 437–439, 1973.

[28] U. Feige and J. Kilian, "Zero knowledge and the chromatic number," *J. Comput. Syst. Sci.*, vol. 57, pp. 187 – 199, 1998.

[29] R. Brooks, "On colouring the nodes of a network," *Proc. Cambridge Philos. Soc.*, pp. 194–197, 1941.

[30] S. Turrini, "Optimization in permutation spaces," *West. Res. Lab. Rep.*, 1996.

[31] L. Shi and S. Ólafsson, "Nested partitions method for global optimization," *Opera. Res.*, vol. 48, pp. 390–407, 2000.

[32] D. Johnson and L. McGeoch, *The Traveling Salesman Problem: A Case Study in Local Optimization*. John Wiley and Sons, Ltd, 1997.

[33] D. Rosenkrantz, R. Stearns, and P. Lewis, "Approximate algorithms for the traveling salesperson problem," in *Proc. SWAT 1974*, pp. 33–42, Oct. 1974.

[34] L. Shi *et al.*, "New parallel randomized algorithms for the traveling salesman problem," *Comput Opera. Res.*, vol. 26, pp. 371–394, 1999.

[35] "Networkx." [Online]. Available: http://networkx.github.io/

[36] H. Wu *et al.*, "Security-and-interference-aware distance spectrum assignment in elastic optical networks," in *Proc. of NOC 2016*, pp. 1–6, Jun. 2016.