Network-Wide Local Unambiguous Failure Localization (NWL-UFL) via Monitoring Trails

János Tapolcai*, Pin-Han Ho†, Lajos Rónyai‡, and Bin Wu†
* Dept. of Telecommunications and Media Informatics, Budapest University of Technology, tapolcai@tmit.bme.hu
† Dept. of Electrical and Computer Engineering, University of Waterloo, Canada, p4ho@uwaterloo.ca
‡ Computer and Automation Research Institute Hungarian Academy of Sciences (MTA SZTAKI),
2^{nd} Inst. of Mathematics, BME, ronyai@sztaki.hu

Abstract—**Monitoring trail (m-trail) has been proposed as an effective approach for link failure localization in all-optical WDM (Wavelength Division Multiplexing) mesh networks. Previous studies in failure localization rely on alarm dissemination via control plane signaling such that the network controller can collect the flooded alarms to form an alarm code for failure identification. Such cross-layer signaling effort obviously leads to additional control complexity. The paper investigates a novel m-trail failure localization scenario, called network-wide local unambiguous failure localization (NWL-UFL), where each node can perform UFL based on locally available on-off state of traversing m-trails, such that alarm dissemination in the control plane can be completely avoided. The paper firstly defines and formulates the m-trail allocation problem under NWL-UFL, and conducts a series of bound analysis on the cover length required for localizing any single link failure. This is the first study on monitoring trail allocation problem which aims to gain understanding on the consumed cover length via analytical approaches due to the special feature of the NWL-UFL scenario. A novel heuristic algorithm based on random spanning tree assignment (RSTA) and greedy link swapping (GLS) is developed for solving the formulated problem. Extensive simulation on thousands of randomly generated network topologies is conducted to verify the proposed scheme by comparing it with a naive counterpart and with the derived lower bounds. We also demonstrate the impact of topology diversity on the performance of the proposed scheme as well as its scalability regarding network sizes.**

I. Introduction

Unambiguous Failure Localization (UFL) is defined in all-optical mesh WDM networks where any link failure can be precisely and instantly identified via monitoring a set of supervisory lightpaths. Such capability in the network optical layer is highly desired in order to meet the stringent requirement on service continuity and to support various failure-dependent recovery mechanisms [1]–[3].

Several UFL schemes have been proposed [4], including simple and non-simple m-cycle [5]–[8], m-trails [5], [9]–[12], directional m-trail [6], [13], [14], and m-trail with loopback switching [9], [15]. Although they differ from each other by adopting a specific monitoring structure and design objectives, a common feature is that each monitoring structure is all-optically pre-cross-connected as a supervisory lightpath dedicated for the failure localization purpose. The on-off status of a supervisory lightpath, as detected by the monitor equipped at the supervisory lightpath, indicates whether or not a link failure event occurs along the lightpath. The reading of the monitor (either "on" or "off") serves as a single bit of an alarm code, and the "off" status of the lightpath is taken as an indication that one of the links along the lightpath is cut. With a set of well routed supervisory lightpaths, an alarm code can be generated by collecting the alarm bits from all the lightpaths, such that the failed link can be unambiguously identified.

Using m-trails with bi-directional lightpaths in all-optical WDM networks has been proposed [9], [12], [15], [16]; and all these studies set the destination node of each m-trail as the only node that can detect the status of the m-trail. This results in a fact that alarm dissemination is needed upon a failure event mostly via flooding and dedicated failure notification based on network layer signaling, such that the alarm code can be formed at a remote site or any network node in order to localize the failure. Obviously, using electronic signaling in alarm bit collection incurs additional control complexity, operational overhead, and lower robustness of the system. The dependency on the upper layer signaling mechanism makes the previous arts hard to be claimed as purely operated in the optical domain. Such an issue was considered in [5], which aimed to minimize the number of monitoring locations (MLs) in order to reduce the alarm dissemination. Nonetheless, the research in [5] can hardly be applicable to the scenario where all the nodes are required to perform UFL in the optical domain, and may yield solutions with multiple MLs in sparse networks, and in this case the MLs have to exchange their alarm bits via control plane signaling. Most importantly, the approach in [5] does not concern the number of required supervisory lightpaths which nonetheless stands for some important physical resource consumptions such as the number of transmitters and the total cover length.

Motivated by the fact that UFL should be carried out completely in the optical domain without taking any control plane signaling effort, the paper investigates an interesting scenario of all-optical failure localization using bi-directional m-trails. Here we define *Local UFL* (L-UFL) at a node if the node can individually perform UFL based on locally available on-off status of the traversing m-trails; and *Network-Wide*

J.T. was supported by the Magyary Zoltán post-doctoral program. R.L. was supported by the Hungarian Research Fund (OTKA grants K77476, and K77778). The project was also supported by TÁMOP 421B-09/1/KMR-2010-0002 grant. P. -H. Ho was supported by National Science and Engineering Research Council (NSERC), Canada.

Local UFL (NWL-UFL) in the network if every node is L-UFL capable. By assuming that any node along an m-trail can obtain the on-off status of the m-trail via optical signal tapping, all the nodes traversed by the m-trail can share the on-off status of the m-trail. The proposed NWL-UFL m-trail allocation problem is to find a set of m-trails such that every node can perform L-UFL based on the traversing m-trails.

The paper first introduces the NWL-UFL framework along with its possible application scenarios, and officially defines the NWL-UFL m-trail allocation problem. Different from any previous art, the paper targets the minimization of total cover length of m-trail solutions, mainly due to various unique features of the proposed NWL-UFL scenario. To gain sufficiently deep understanding of the problem, we conduct a series of bound analysis by proving optimal solutions in terms of minimal cover length in a number of special graphs, such as lines, stars, and complete graphs, as well as the derivation of lower bounds in general graphs. Inspired by the optimal solution for complete graphs, we develop a novel heuristic algorithm for solving the NWL-UFL m-trail allocation problem based on random spanning tree assignment (RSTA) and greedy link swapping (GLS). Extensive simulation on thousands of randomly generated topologies is conducted to verify the proposed heuristic algorithm and examine the derived lower bounds. The impact of network diversity on the m-trail solutions is also investigated.

The rest of the paper is organized as follows. Section II briefly reviews existing failure localization schemes. Section III presents the NWL-UFL m-trail allocation problem. Bound analysis is provided in Section IV, and our heuristic algorithm is introduced in Section V. Section VI presents our simulation results. Section VII concludes the paper.

II. LITERATURE REVIEW

Studies on link failure localization in all-optical mesh networks via out-of-band monitoring and dedicated supervisory lightpaths have long been reported. Since the supervisory lightpaths are launched dedicatedly for the failure localization purpose, the failure localization system can operate independently from the data plane and is expected to be much more robust than its counterparts which rely on monitoring of working lightpaths.

The studies [5], [7], [8] focused on failure localization using close-loop supervisory lightpaths (or termed m-cycles), and [10]–[12] for open-loop supervisory lightpaths (or termed m-trails). The study in [5] targeted at minimizing the number of MLs, which is determined by analyzing the connectivity among the 2- and 3-connected components in the topology. With each ML determined, a graph transformation is performed such that the MLs are merged into a supernode (denoted by v_s), and cycles are cumulatively added into the transformed graph one by one via Suurballe's algorithm. To distinguish two links e_1 and e_2, a cycle must be disjoint from e_1 while passing v_s and e_2.

M-trail with loopback switching was first investigated in [9], [15], [17] with a goal to minimizing the number of m-trails. In essence, an m-trail is a supervisory lightpath which can pass a node by multiple times and a directed link by once (or once per direction in a bidirectional link). Upon a link failure, the m-trails passing the failed link are interrupted, and the monitors of the m-trails detect Loss of Light (LoL) and issue alarms. The alarms are flooded in the network control plane and collected by the remote routing entities or the network controller to form an alarm code. Here an alarm code contains a series of bits, and each bit stands for the on-off status of the corresponding m-trail. The j-th bit in the alarm code (denoted as $a_{<j>}$) is 1 if the j-th m-trail is subject to LoL, and 0 otherwise. The study [18] the number of required laser diodes by allowing multiple receivers for each m-trail.

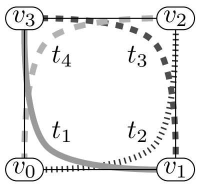
Our previous study [19] investigated the L-UFL scenario with bi-directed m-trails in all-optical mesh networks. A number of MNs are first given as the input of the problem, and m-trails are routed such that each of the MNs can perform L-UFL. The study [19] provided an integer linear program (ILP) for the formulated m-trail allocation problem which was solved using CPLEX. Although [19] demonstrated the relation between cover length and the number of L-UFL MNs with directed m-trails, the study is far from sufficient to achieve in-depth understanding of the problem. Further, it falls short of an intelligent heuristic that can obtain efficient solutions while avoiding very lengthy computation taken by solving the ILP.

III. PROPOSED NWL-UFL FRAMEWORK

A. Introduction of NWL-UFL

All the previously reported monitoring systems using multi-hop supervisory lightpaths extensively rely on signaling efforts in the network control plane in alarm dissemination and collection. The proposed NWL-UFL scenario is a novel m-trail application aiming to completely remove the dependency on such control plane signaling. By assuming that a node can tap the optical signals of a traversing m-trail, it is ensured that a node can detect a link cut along the m-trail. This is achieved because the transmitter of an m-trail will be shut down when LoL is detected, which darkens both channels of each link along the m-trail. A feasible NWL-UFL m-trail solution consists of a set of m-trails such that each node can perform UFL based on the on-off status of a subset of the m-trails that pass through the node. In other words, each node has a specific alarm code table (ACT) based on the traversing m-trails, where any failed link can be identified at the node using the locally collected alarm bits.

Fig. 1(a) shows an example of m-trail allocation in a topology with 4 nodes. Fig. 1(b) is a link-trail incidence table (IT) which keeps the routes of the four m-trails t_1, t_2, t_3, and t_4. In this case, each of the four nodes is an L-UFL capable MN and can localize any single link failure by inspecting the locally available on-off status of the traversing m-trails. For example, v_1 can achieve L-UFL by keeping the ACT as shown in Fig. 1(c), by which the on-off status of the traversing m-trails t_1, t_2, and t_3 can be used to uniquely identify any single link failure event. The failure localization at v_1 can be done via a look-up-table process using the ACT. Each row of the ACT



(a) M-trails

Link	t_1	t_2	t_3	t_4
(v_0, v_1)	1	1	0	0
(v_1, v_2)	0	1	1	0
(v_2, v_3)	0	0	1	1
(v_3, v_0)	1	0	0	1

(b) link-trail IT

Link	t_1	t_2	t_3
(v_0, v_1)	1	1	0
(v_1, v_2)	0	1	1
(v_2, v_3)	0	0	1
(v_3, v_0)	1	0	0

(c) ACT at v_1

Fig. 1. An NWL-UFL m-trail solution

corresponds to a failure state. For example, if v_1 finds that t_1 and t_2 become suddenly off while t_3 is still on, the link (v_0, v_1) is considered down by matching the first row of ACT; if t_1 and t_2 are on while t_3 is of (as shown in the third row of the ACT), the link (v_2, v_3) is considered down. Similarly, v_0, v_2, and v_3 can perform L-UFL by maintaining their respective ACTs, each of which keeps the mapping between all the considered failure states and the on-off status of the traversing m-trails.

In this study, we investigate the scenario where the m-trails are bi-directional, and a link is taken by an m-trail iff the corresponding lightpath passes through both directions of the link. As such, the m-trail allocation problem under NWL-UFL does not need to follow the cycle or Euler constraint as required in [5], [8], [14]; instead, an m-trail in this study could be in any shape, including simple/non-simple path, tree, and simple/non-simple cycle, whereas the transmitter and receiver can be placed at any node.

The physical length limitation of each m-trail could certainly be an issue in some circumstances; for example, lengthy m-trails may require regenerators (e.g., fiber amplifiers) to maintain the optical power level sensed at each intermediate node. The study does not consider the length limitation of each m-trail by assuming proper deployment of the regenerators. Since only the on-off status of each m-trail is of interest, strict requirements on 3R (re-amplification, re-shaping, and re-timing) imposed on regular working lightpaths can be much relaxed.

An important design requirement on the proposed NWL-UFL scenario is that an intermediate node of an m-trail can obtain the on-off status of the m-trail via signal tapping. This is based on the assumption that the optical energy tapped at each intermediate node is close to zero or no more than the noise level if any link is cut along the m-trail. Such an assumption is valid provided with sufficiently good distinction ratios at the traversing regenerators. Note that the state-of-the-art technique for fiber amplifiers such as EDFA relies on an external bumping source with a much different frequency from the carrier, which yields a trivial output energy level at a specific wavelength in case the input signal of the wavelength is absent. The same mechanism was employed in [20], [21] where the alarm messages were collected at transit nodes along a lightpath via optical signal tapping.

It is possible that an m-trail solution contains some existing working lightpaths, so as to reduce consumption of monitoring resources. This is nonetheless at the expense of knowledge leakage between the service and control planes which causes a significant amount of additional management complexity and less robustness. By envisioning that hundreds of wavelengths can be multiplexed onto a single fiber and several fibers are bundled in a conduit in the future Internet backbone, deploying m-trails for out-of-band monitoring could be a viable approach in terms of the reasonably consumed resources and overhead for achieving the distributed and agile fault management systems. Without loss of generality, the paper assumes that working lightpaths are not taken in the monitoring systems.

B. Problem Definition

The input is an undirected graph $G = (V, E)$ with node set V and link set E, where the number of nodes is denoted by $n = |V|$ and the number of links by $m = |E|$. The m-trail problem for single-link NWL-UFL is to establish a set of m-trails with minimum cover length, denoted by $T = \{t_1, \ldots, t_b\}$ where $b = |T|$ is the number of m-trails, such that each m-trail t_i is a connected subgraph of G, and each node $v_j \in V$ can achieve L-UFL according to the on-off status of m-trails in T^j - a subset of T containing the m-trails passing v_j. Let the ith bit of alarm code for link e at v_j be denoted by $a^{e,j}_{<i>}$ for $1 \le i \le b$. We have $a^{e,j}_{<i>} = 1$ if $t_i \in T^j$ and passes through e and 0 otherwise. The total cover length is denoted by $||T||$ and is defined as the total number of wavelengths (WLs) taken by T. Formally, $||T|| = \sum_{i=1}^{b} |t_i|$, where $|t_i|$ is the number of links in t_i.

The set of m-trails T^j for v_j must satisfy the following two requirements:

(R1): Every link e should be passed by unique set of m-trails in T^j, such that every link has a unique alarm code seen by v_j denoted by $[a^{e,j}_{<1>}, ..., a^{e,j}_{<b>}]$.

(R2): t_i for $1 \le i \le b$ must travel through all the links e with $a^{e}_{<i>} = 1$ while disjoint from any link with $a^{e}_{<i>} = 0$.

The following theorem proves the feasibility of the problem in any connected graph. The theorem demonstrates that an m-trail solution for single-link NWL-UFL can always be obtained in a connected graph.

Theorem 1: Given a connected graph and any node in the graph, an m-trail solution for NWL-UFL can always be found.

Proof: The necessary and sufficient condition for NWL-UFL is that every node can achieve L-UFL. For this purpose, we have to show that the failure of any arbitrary link (v_i, v_j) can be unambiguously localized at any arbitrary node v_s using a suitable collection of m-trails. NWL-UFL can be achieved by doing the same task for all the nodes upon all the links.

Let t_1 be a (possibly empty) shortest path in terms of hop count from v_s to v_i while disjoint from the link (v_i, v_j). Also, let $t_2 = t_1 \cup \{(v_i, v_j)\}$. Then the fault on (v_i, v_j) can be

TABLE I
NOTATION LIST

Notations	Description
$G=(V,E)$	the undirected graph representation of the topology with nodes V and edge set E
n	the number of nodes in G
m	the number of links in G
b	the number of m-trails
t_i	the i^{th} m-trail of the solution, which is a connected subgraph of G
$T=\{t_1,\ldots,t_b\}$	the set of m-trails with NWL-UFL
$\|t_i\|$	the length in hops of i^{th} m-trail
$\|\|T\|\|$	the total cover length
a^e	the alarm code of $e\in E$
$a^e_{<i>}$	the i^{th} bit of the alarm code of $e\in E$
$a^{e,j}_{<i>}$	the i^{th} bit of the alarm code of $e\in E$ visible at node v_j

unambiguously identified if t_2 gives an alarm signal while t_1 does not. These events are obviously visible at v_s. This finishes the proof. ■

We would justify the use of total cover length as the optimization target in the problem formulation. First of all, since NWL-UFL enables all nodes to individually perform UFL via optical signal tapping, the resultant m-trail solution potentially requires significantly more resources than that with a single UFL node. The study takes the total cover length as the performance metric and optimization target due to its simplicity and generality, in order to explore the feasibility of the proposed NWL-UFL scenario. Other widely used performance metrics such as the number of transmitters and the total physical length of an m-trail solution, are nonetheless left as our future studies. Note that the performance evaluation and analysis on the number of m-trails has been extensively conducted in the past, while very little understanding was gained regarding the cover length of an m-trail solution. Further, since NWL-UFL does not rely on any upper layer signaling effort for alarm code dissemination, the length of alarm code becomes not as critical as that in the previous research [7-17].

C. Applicability of NWL-UFL

The subsection summarizes the unique features of the proposed NWL-UFL scenario, and positions its applicability and role in the optical network control and management.

One of the interesting features of NWL-UFL that it can be integrated with any conventional failure dependent/independent protection scheme. Such a feature is desired in some networking scenarios in order to achieve an ultra-fast and a completely all-optical recovery process, which is essential to hide the optical layer failure events from upper layer protocols/applications. For example, it is very useful if a fiber cut event can be restored purely in the optical domain while leaving the network layer link state protocol unaware of the event. On the other hand, a transport layer protocol will not be seriously affected in its throughput if the fiber cut event can be restored fast without causing serious packet loss and/or delay. To achieve the above, a recovery process with a few tens milliseconds of recovery time is desired.

In general, a recovery process is composed of a number of tasks that need to be performed on-line after the occurrence of the failure, where *failure localization*, *failure notification*, *failure correlation*, and *device configuration* (i.e., backup lightpath setup) serve as main contributors to the total recovery time in a path recovery scheme [22]. Detailed recovery time analysis can be seen in [23]. With NWL-UFL, an all-optical recovery process can be achieved such that the recovery time due to the above real-time tasks is significantly reduced. Firstly, since the source node of an interrupted working lightpath is aware of the failed SRLG instantly under NWL-UFL, the time of failure localization is short and deterministic, and the time of failure notification can be completely waived. The time of failure correlation at a node, on other other hand, becomes deterministic and solely depends on the length of the longest m-trail passing through the node. Further, since all the intermediate nodes of the backup lightpath corresponding to the failure event can localize the failure event thanks to NWL-UFL, they can start configuring their switch fabrics based on the collected alarm code without waiting for the backup path setup request from any other network entity. Thus the backup lightpath setup latency can be minimized as well. Therefore, the whole recovery process can be performed completely in the optical domain without the aid of any upper layer control signaling protocol, and is expected to fundamentally improve the recovery speed for the working lightpaths with very high service-continuity requirements.

The expense paid for the NWL-UFL scenario is the additional network resources for supporting the m-trails in terms of wavelength channels and transmitters/receivers. As demonstrated by the simulation results in Section VI, NWL-UFL consumes in average 2-4 WLs along each link and no less than $\log_2(|E|+1)$ transmitter-receiver pairs.

IV. Bound Analysis

This section presents our bound analysis for cover length in the proposed NWL-UFL m-trail problem. We will first look into a lower bound on general connected graphs, followed by the optimal solutions for a number of special graph topologies: line, star and complete graphs.

A. Lower Bound for General Graphs

Theorem 2: Let T be a valid m-trail solution for NWL-UFL. We have

$$||T|| \geq \sum_{j=1}^{n}\sum_{\mu=1}^{n}\frac{\mu-1}{\mu}\left(\lceil\log_2\delta_j(\mu-1)\rceil-\lceil\log_2\delta_j(\mu)\rceil\right) \quad (1)$$

where $\delta_j(\mu)$ denotes the number of links whose shortest distance from node v_j is μ.

Proof: Theorem 2 holds for the following intuitive reasons. Each node must be traversed by at least $\lceil\log_2(m+1)\rceil$ m-trails in order to have sufficient bits for L-UFL. Each m-trail t_i can traverse at most $|t_i|+1$ nodes, thus long m-trails lead to a lower bound on cover length almost $(n-1)\lceil\log_2(m+1)\rceil$. We show that a NWL-UFL solution cannot have too many short m-trails, because the m-trails must carry the information from one side of the network to the other. The detailed argument is relegated to the Appendix. ■

Corollary 1: Let T be a valid m-trail solution for NWL-UFL. We have $||T|| \geq \frac{n}{2}\log_2(m+1)$.

Theorem 3 provides a lower bound on $||T||$ as a linear function of m. With a single link failure to be identified, any link must be traversed by at least one m-trail. A link is called *singular* if it is traversed by exactly one m-trail. The following two lemmas are related to singular links.

Lemma 1: An m-trail t_j of a valid m-trail solution could traverse no more than one singular link.

Proof: Let links e and e' be two singular links on t_j in a valid m-trail solution. Since t_j is the only m-trail traversing the two links, the two links cannot be distinguished when failure occurs to either one of the links. Thus the m-trail solution is not valid, which contradicts the assumption. ■

Lemma 2: If t_j traverses a singular link, then t_j must be a spanning subgraph (i.e., connected to all the nodes) with $|t_j| \geq n-1$.

Proof: Let t_j be an m-trail of a valid m-trail solution. If e is a singular link of t_j, and v is a node not in t_j, then the failure of e cannot be detected by v via any m-trail. This contradicts the fact that the m-trail solution is valid. Since t_j is a spanning subgraph, we have $|t_j| \geq n-1$. ■

Theorem 3: Let T be a valid m-trail solution for NWL-UFL on a connected graph with m links. We have

$$||T|| \geq 2m\left(1-\frac{1}{n}\right)$$

Proof: Let the number of singular links be denoted as σ. Clearly, there is at least a number of σ m-trails according to Lemma 1. The cover length of an m-trail solution can be estimated in terms of σ as follows:

$$||T|| \geq 2(m-\sigma)+\sigma = 2m-\sigma. \tag{2}$$

A direct consequence of the Lemma 1 and 2 is:

$$||T|| \geq \sigma(n-1). \tag{3}$$

In case $\sigma \geq \frac{2m}{n}$, according to Eq. (3) we have

$$||T|| \geq (n-1)\cdot\frac{2m}{n} = 2m\left(1-\frac{1}{n}\right).$$

While in case $\sigma < \frac{2m}{n}$, according to Eq. (2) we have

$$||T|| \geq 2m-\sigma \geq 2m\left(1-\frac{1}{n}\right).$$

Thus, for any value of σ the statement holds. ■

B. Line Graphs

The line graph P_n has nodes $v_1,\dots,v_n$, and the links are (v_i,v_{i+1}) for $i=1,\dots,n-1$.

Theorem 4: The optimal cover length for line graph P_n with n nodes and $m=n-1$ links is m^2.

Proof: For P_n the m-trails $t_{1,2},\dots,t_{1,n};t_{2,n},\dots t_{n-1,n}$ form a valid m-trail solution for NWL-UFL, where $t_{i,j}$ is the subpath form v_i to v_j.

If (v_i,v_{i+1}) is the faulty link, then at v_1 we recognize this correctly from the fact that $t_{1,i}$ is on, while the other $t_{1,i+1}$ is off. Every node between v_1 and v_i can tap these m-trails and thus achieve L-UFL. $t_{i+1,n}$ and $t_{i,n}$ can be used similarly at the right side of the line graph, in particular at v_n.

The above m-trail solution is optimal in the sense that if a feasible solution does not contain $t_{1,i}$ for some i (or an $t_{j,n}$ for some j) then we cannot have NWL-UFL for P_n. If $t_{1,i}$ is not in the m-trail solution, then at v_1 we cannot distinguish the failure of (v_{i-1},v_i) from the failure of (v_i,v_{i+1}), if $i<n$. If $t_{1,n}$ is not in the m-trail solution then v_1 cannot tell the difference between the errorless state of P_n and the failure of (v_{n-1},v_n). Same considerations apply to all the paths $t_{j,n}$.

The cost of the collection $t_{1,2},\dots,t_{1,n};t_{2,n},\dots t_{n-1,n}$ is

$$1+2+\dots+n-1+1+2+\dots n-2 = \frac{n(n-1)}{2}+\frac{(n-1)(n-2)}{2} = (n-1)^2 = m^2. \quad ■$$

C. Stars

Let the nodes and links of star S_n be denoted by $v_c,v_1,\dots,v_{n-1}$, and (v_c,v_i) for $i=1,\dots,n-1$, respectively. We note the simple facts that every set of links of S_n spans a connected subgraph; every link is adjacent to the center v_c; and a link set is adjacent to v_i, also referred to as a leaf node, iff it contains the link (v_c,v_i). These observations imply that any set of links is a valid m-trail, and thus the problem is simplified to a coding process for the links without considering whether a set of links can form an m-trail. We will provide a lower and an upper bound on the cover length, which are exactly the same in case $|E|=m=2^{b'}$ where b' is an integer.

Lemma 3: A feasible m-trail solution on S_n has a lower bound on the total cover lengths: $||T|| \geq m\lceil\log_2(m+1)\rceil$.

Proof: For any node in S_n, it needs to be supported by at least $\lceil\log_2(m+1)\rceil$ m-trails (each providing a bit of information in its alarm code table) to uniquely identify $m+1$ possible states in the network. This is according to the binary coding mechanism to perform UFL in a graph with m links. Since each node v_i is adjacent to a single link for $i=1,\dots,n-1$, the number of m-trails traversing through each link is at least $\lceil\log_2(m+1)\rceil$; and the total cover length is $||T|| \geq m\lceil\log_2(m+1)\rceil$ since there are m such links. Thus, lemma is proved. ■

In the following an m-trail construction is developed which exactly yields a solution with $||T|| = m(\lceil\log_2 m\rceil+1)$.

Lemma 4: A m-trail allocation construction can be found to achieve NWL-UFL in S_n with cover length $||T|| \leq m(\lceil\log_2 m\rceil+1)$.

Proof: The proposed m-trail construction is introduced as follows. Let the alarm code a^e be a bit vector of length b assigned to link e $\forall e\in E$ such that the i-th bit is 1 if $e\in t_i$, and 0 if $e\notin t_i$. The construction is to determine how the alarm code for each link should be designed, which can exclusively determine the set of m-trails $t_1,t_2,\dots,t_b$.

Let us define $b' = \lceil \log_2 m \rceil$. The construction has each alarm code with a length $b = b' + 1$. We have the first b' bits uniquely code the m links. Such a link coding process must be feasible since $2^{b'} \leq m$. The next b' bits will be exactly the complements of the first b' bits allocated to e. For example, if a^e has the first bit as 0, then its $(b'+1)$-th bit should be 1, and so on. This results in the fact that the m-trail corresponding to the i-th bit position (denoted as t_i) is the complement graph of m-trail $t_{i+b'}$, $\forall 1 \leq i \leq b'$. Finally, the last bit of a^e is 1 for every e. With the $2b'+1$ m-trails, the construction is complete.

To prove the construction yields a feasible solution, our first step is to argue that every node in the star can perform L-UFL. Clearly, the number of 1's in each alarm code is $b' + 1$; in other words, exactly $b' + 1$ m-trails are terminated at each leaf node. By further considering that each link is uniquely coded in the construction (using the first $b' + 1$ bits), we can conclude that each leaf node can obtain sufficient information to perform UFL based on the terminated $b'+1$ m-trails. Similar consideration can be applied to the proof of L-UFL for the center which is traversed by all the $2b' + 1$ m-trails.

With the construction, the number of 1's in the alarm codes of all the m links is $m(b' + 1) = m(\lceil \log_2 m \rceil + 1)$, which stands for the total cover length of the m-trail solution. ■

Theorem 5: The optimal cover length for star graph with n nodes is $||T|| = m(1 + \lceil \log_2 m \rceil)$ if $m = 2^{b'}$, where b' is an integer.

Proof: Based on lemma 3 and lemma 4, we conclude that the construction is optimal when $m = 2^{b'}$ because the upper and lower bounds are equal $||T|| = m(\lceil \log_2(m + 1) \rceil) = m(b' + 1)$. ■

D. Complete Graphs

Theorem 6: The minimal cover length for complete graph K_n with n nodes is $||T|| = (n - 1)^2$.

Proof: We prove the theorem with a construction introduced as follows. Let $S(i)$ denote the star in K_n centered at v_i. $S(i)$ simply consists of $n - 1$ links adjacent to v_i. Now let us arbitrarily fix a node v_w in K_n. The m-trail solution, T, is composed of a set of stars each centered at $\forall v_i \neq v_w$. Clearly, each star in T is a connected subgraph in K_n and is taken as an m-trail. It is clear that with the construction, T has $n - 1$ stars each covering $n - 1$ links adjacent to the center of the star; thus we have $||T|| = (n - 1)^2$.

Such T is a valid m-trail solution for NWL-UFL in K_n. Clearly with the construction, every link must be traversed by two m-trails except for the links adjacent to v_w, and each star is a spanning graph in K_n such that each node can access the status of all the m-trails. Thus, the failure on any link, say (v_i, v_j), can be uniquely identified by any node in K_n due to either (1) the off status of $S(v_i)$ and $S(v_j)$ if $v_i \neq v_w$ and $v_j \neq v_w$, or (2) the off status of $S(v_i)$ or $S(v_j)$ in case $v_i = v_w$ or $v_i = v_w$, respectively. The validity of the construction is thus proved.

For optimality, by Theorem 3 we have

$$||T|| \geq 2m\left(1 - \frac{1}{n}\right) = n(n-1)\left(1 - \frac{1}{n}\right) =$$
$$= n(n-1) - (n-1) = (n-1)^2.$$

Thus we proved the theorem. ■

V. A Novel Heuristic for NWL-UFL

The section presents a novel heuristic algorithm to solve the NWL-UFL m-trail allocation problem in general 2-connected graphs. Inspired by the optimal solution for complete graphs in Section IV, we have observed the great suitability of using spanning subgraphs as the m-trails with bi-directional lightpaths. Thus, the proposed approach uses spanning trees as basic structures of m-trails. To be specific, the proposed approach is based on two novel mechanisms referred to as *random spanning tree assignment* (RSTA) and *greedy link swapping* (GLS), aiming to take the best advantage of flexible and cost-effective spanning tree structures, so as to overcome the topology diversity and maximize sharing of information in solving the m-trail allocation problem for NWL-UFL.

With RSTA, b randomly generated spanning trees are launched in the network. The set of b randomly generated spanning trees, denoted as $T' = [t_1, t_2, \ldots, t_b]$, is used to determine the initial assignment of alarm code for every link e (denoted as a^e), where the alarm code a^e has the j-th bit as 1 if t_j traverses through e, and 0 otherwise. Clearly, a^e is of length b bits. We claim that T' can be taken as a valid m-trail solution for NWL-UFL if $a^e \neq a^f, \forall e \neq f \in E$. Note that the global code uniqueness, as well as the spanning nature of each t_j, sufficiently guarantees the validity of the solution for L-UFL at each node.

Let us define a *collision* of two codes if they are identical and used by at least two links. Let the bitwise pair at the i-th position of a^e be denoted as $a^e_{<i>}$, which is the code with all identical bits as a^e except for the i-th bit. For example, 011100 is the bitwise pair for the third position of 010100.

The GLS aims to remove all possible collisions by swapping collided codes with unused ones while maintaining the spanning nature of each m-trail. The GLS is performed iteratively upon each bit position $i = 1, \ldots, b$ one by one, where two tasks are defined in each iteration. (1) The GLS checks each bit of a link code a^e under collision, and swaps the link code with $a^e_{<i>}$ if it can resolve the code collision. (2) The code swapping in the first task will turn t_i into a subgraph with either a cycle or two isolated components. Thus GLS swaps another link code a^f on the i-th position with $a^f_{<i>}$ which is currently unused such that t_i is retained as a spanning tree.

A. Algorithm description

Algorithm 1 shows the pseudo code of the proposed heuristic algorithm. With each m-trail as a spanning subgraph, the cover length is at least $||T|| \geq b(n - 1)$; therefore it is easy to see that a smaller b leads to a smaller total cover length, at the expense of smaller opportunities for successful GLS due to a smaller number of unused codes that can be taken for

Algorithm 1: M-Trail Design Problem for L-UFL

```
    Input: $G(V,E)$
    begin
1       Set $b_{ini} := \min\{\lceil\log_2(n-1)\rceil+1, \lceil\log_2(m+1)\rceil\}$
        for $b := b_{ini}$ to $n-1$ do
2           RSTA: Randomly generate $b$ spanning trees
3           Sort the alarm codes in descending order
4           for $i := 1$ to 500 do
                for iterate through the sorted alarm codes do
5                   if link $e$ and $f$ has the same code then
6                       call $GLS(G,e,i)$
                if every link has unique alarm code then
7                   return succeed
```

Algorithm 2: Greedy Link Swapping (GLS)

```
    Input: $G(V,E)$ link $e$ with collided code and iterator $i$
    begin link $e$ has code $a^e = \{a^e_1, \dots, a^e_b\}$
6.1     for $i := 1$ to $b$ do
            if $a^e_{<i>}$ is currently unused and nonzero then
6.2             if $a^e_i = 0$ then
                    After adding $e$ into $t_i$ the m-trail has a
                    cycle. Let $L$ be the path in m-trail $t_i$
                    between the adjacent nodes of $e$
6.3             else $a^e_i = 1$
                    After erasing $e$ in $t_i$ the m-trail falls into
                    two parts. Let $L$ be the set of edges in $G$
                    that can be used to reconnect the parts
6.4             forall the link $f \in L$ do
                    if $a^f_{<i>}$ is unused and nonzero then
6.5                     flip the bits: $a^e_i = \neg a^e_i$ and $a^f_i = \neg a^f_i$
                        return succeed
                if $i > 250 \wedge sparse(G) \wedge a^e_i = 0$ then
6.6                 flip the bits: $a^e_i = \neg a^e_i$; return succeed
        return not succeed
```

replacement. Two lower bounds on b are identified: the first is $b \geq \lceil\log_2(m+1)\rceil$, which comes from the fact that each link must have a unique nonzero alarm code; the second is $2^{b-1} \geq n-1$, which can be argued in the following claim.

Claim 1: $2^{b-1} \geq |t_j| \geq n-1$ holds for $j = 1, \dots, b$.

Proof: Since t_j is a spanning sub-graph, we have $|t_j| \geq n-1$. Further, $|t_j|$ is upper bounded by the number of codes with 1's at the bit position j, which is nonetheless no more than half of the 2^b according to the binary coding mechanism. Thus we have $2^{b-1} \geq |t_j| \geq n-1$. ■

With the two lower bounds, the initial value of b in Step (1) is set as $b = \min\{\lceil\log_2(n-1)\rceil+1, \lceil\log_2(m+1)\rceil\}$. In case the algorithm has never succeeded with b, the algorithm starts over again by increasing b, and this iteration continues until either we find a valid solution in Step (7) or $b = n$. With such initial setting of b, the minimum achievable cover length $||T^*||$ can be expressed as:

$$||T^*|| \geq (n-1)\min\{\lceil\log_2(n-1)\rceil+1, \lceil\log_2(m+1)\rceil\} \quad (4)$$

In Step (2), b random spanning trees are generated, and each link is assigned with a code of length b where the i-th bit is 1 if the link is traversed by t_i. In our implementation the method of Aldous/Broder [24], [25] is adopted for this purpose. The code assignment in Step (2) may cause code collision (i.e., a code assigned to two or multiple links); and some links may not be traversed by any spanning tree (with an all-zero code "$00\dots0$"). From Step (3) to (6), the collided codes are identified by sorting all the codes. For each link pairs with a collided code, the GLS method is called in Step (6) to resolve the collision. The algorithm stops if the loop in Step (4) is executed over 500 times to avoid infinite loops.

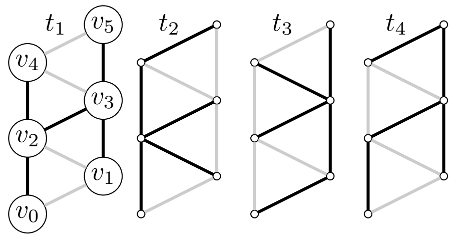
Algorithm 2 is called when a collision is found at link e, and the goal of the algorithm is to find an unused non-zero alarm code that can replace the old one and resolve the collision. In Step (6.1) we inspect each bit position on a^e to see if there is a nonzero and unused bitwise pair. If there is such $a^e_{<i>}$ (i.e., the bitwise pair of a^e at the i-th position), then we check how such swapping of the code pair would impact t_i. In case the swapping is to flip the bit at the i-th position from 0 to 1 (in Step (6.2)), it means the swapping simply adds link e to t_i. Since t_i is a spanning tree, adding e to t_i must create a loop for t_i. Thus, the algorithm tries to remove the loop by inspecting each link along the loop (denoted as L) except for e, to see if removing any link along L is possible, as shown in Step (6.4). If there is any link $f \in L$ for which $a^f_{<i>}$ is unused and nonzero, the pair of codes are swapped to remove the link from t_i such that t_i is still a spanning tree in Step (6.5).

In case the swapping is to flip the bit at the i-th bit position from 1 to 0 (in Step (6.3)), the swapping will remove e from t_i. Since t_i is a spanning tree where each node pair is at most connected by a single link, removing e from t_i must break t_i into two isolated components. Then, the GLS needs to reconnect the two components. The links that can be used to reconnect the two isolated components except for e (denoted as L), are inspected one after the other, until any link f in L with an unused non-zero $a^f_{<i>}$ is found. If such link f is found successfully, the code pair is swapped in order to retain t_i as a spanning tree in Step (6.5).

In our implementation, in case the problem cannot be solved in 250 iterations (which happens when the networks are very sparse), cycles are also allowed. We call a network sparse, i.e. $sparse(G) := true$, iff the lower bound by Theorem 2 is sharper the bound of Theorem 3. As shown in Step (6.6), a^e can be swapped with $a^e_{<i>}$ if a^e is nonzero and unused with 0 at the i-th position.

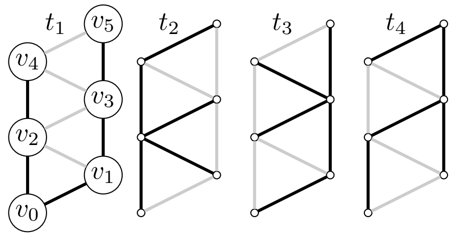
Note that in case the algorithm fails to find a way to resolve a code collision under a specific code length b, it breaks the loop and adds one more bit in the alarm code. It implies that the number of unused nonzero codes is doubled, which significantly helps the algorithm to resolve any possible code collision. We will show in the simulation that the proposed heuristic returned a valid NWL-UFL solution in all the ran-

link	code
(v_3, v_4)	0010
(v_0, v_1)	0010
(v_1, v_2)	0100
(v_5, v_4)	0101
(v_5, v_3)	1011
(v_3, v_1)	1011
(v_2, v_4)	1100
(v_0, v_2)	1101
(v_3, v_2)	1111



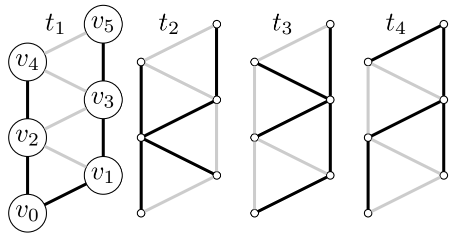
(a) Four random spanning trees

link	code
(v_3, v_4)	0010
(v_1, v_2)	0100
(v_5, v_4)	0101
(v_3, v_2)	0111
(v_0, v_1)	1010
(v_5, v_3)	1011
(v_3, v_1)	1011
(v_2, v_4)	1100
(v_0, v_2)	1101



(b) Swapping $a^{(v_0,v_1)}$ and $a^{(v_3,v_2)}$ with their bitwise pairs at the first position.

link	code
(v_5, v_4)	0001
(v_3, v_4)	0010
(v_1, v_2)	0100
(v_3, v_2)	0111
(v_0, v_1)	1010
(v_3, v_1)	1011
(v_2, v_4)	1100
(v_0, v_2)	1101
(v_5, v_3)	1111



(c) A feasible solution by swapping $a^{(v_5,v_3)}$ and $a^{(v_5,v_4)}$ with their bitwise pairs at the second position.

Fig. 2. Illustrative example of the proposed algorithm.

domly generated topologies (over 2000).

B. An Illustrative Example

First let us illustrate the algorithm through an example using the graph in Fig. 2. Four random spanning trees are generated first (i.e., $b = 4$), each corresponding to an m-trail as shown in Fig. 2(a). Clearly, these m-trails explicitly define an alarm code for each link as shown on Fig. 2(a). The four m-trails do not form a valid m-trail solution due to the collisions $a^{(v_3,v_4)} = a^{(v_0,v_1)} = 0010$ and $a^{(v_5,v_3)} = a^{(v_3,v_1)} = 1011$.

The GLS first tries to swap $a^{(v_0,v_1)}$ with its bitwise pair of the first bit position, i.e., 1010, to remove the collision between $a^{(v_3,v_4)}$ and $a^{(v_0,v_1)}$. It is feasible since 1010 is currently unused. But with the swapping t_1 will no longer be a spanning tree due to the cycle $L_1 = (v_0, v_1, v_3, v_2)$. Thus, a link from L_1 should be removed, and the GLS does this by inspecting each link along L_1 except for the link (v_0, v_1). In our case, the first link inspected is (v_0, v_2) with an alarm code 1101. Since $a^{(v_0,v_2)}_{<1>} = 0101$ has already been taken by link (v_5, v_4), we proceed to the next link (v_3, v_2) in L_1. Luckily, the bitwise pair $a^{(v_3,v_2)}_{<1>} = 0111$ is currently unused by any link. Since both of the tasks in the iteration are feasible, the GLS then finalizes the swapping attempts for $a^{(v_0,v_1)}$ and $a^{(v_3,v_2)}$ with their unused bitwise code pairs: $a^{(v_0,v_1)}_{<1>} = 1010$ and $a^{(v_3,v_2)}_{<1>} = 0111$, respectively. As a result, t_1 is reshaped as shown in Fig. 2(b).

For the collision between $a^{(v_5,v_3)}$ and $a^{(v_3,v_1)}$, the GLS first finds the bitwise pair of $a^{(v_5,v_3)}$ at the first bit position as 0011 that is currently unused. But by swapping $a^{(v_5,v_3)}$ as 0011, t_1 breaks into two components: $\{v_0, v_1, v_2, v_3, v_4\}$ and $\{v_5\}$, and the only way to reconnect them is to swap $a^{(v_5,v_4)} = 0101$ by its bitwise pair at the first bit position: 1101, which, unfortunately, has already been used. Therefore, there is no simple solution at the first bit, and the algorithm turns to find solution from t_2. The bitwise pair of $a^{(v_5,v_3)} = 1011$ at the second bit position is 1111, which is unused. Further, due to the swapping, a loop $L_2 = (v_2, v_3, v_5, v_4)$ is formed on t_2 and needs to be removed. The GLS inspects each link on L_2 except (v_5, v_3), where the bitwise pair of $a^{(v_5,v_4)} = 0101$ at the second bit position, 0001 is currently unused. Thus, the GLS concludes the feasibility of the swapping for $a^{(v_5,v_3)}$ and $a^{(v_5,v_4)}$ with their unused bitwise pairs $a^{(v_5,v_3)}_{<2>} = 0011$ and $a^{(v_5,v_4)}_{<2>} = 0001$, respectively.

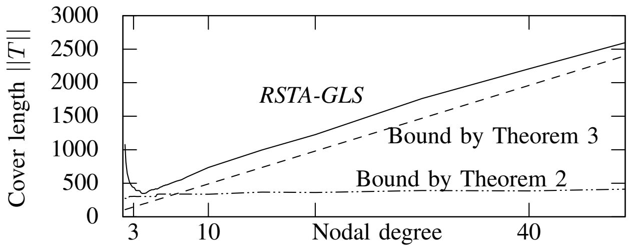
Finally, the GLS completely solved the code collisions, and the final result is shown in Fig. 2(c), which is a valid m-trail solution for NWL-UFL.

VI. Simulation results

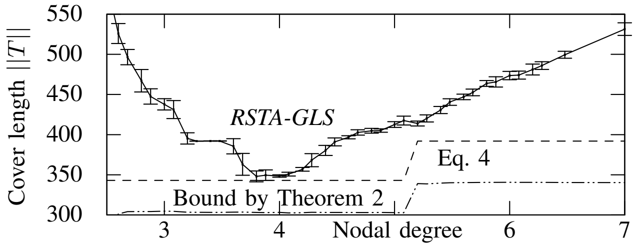
Simulations on thousands of randomly generated planar 2-connected backbone network topologies via LEMON [26] were conducted. With LEMON random graph generator, nodes are firstly allocated into an area of unit square with a uniform distribution, and a link with small physical length are added to keep the graph planar if possible, and to keep the facets of the planar graph of equal size (see examples on Fig. 4). In such a way thousands of random networks were generated and classified according to their size in number of edges or nodes and nodal degrees. Finally a series of networks are selected and stored respectively, according to their (1) number of nodes, (2) number of links, and (3) nodal degrees. The performance metrics of interest are the total cover length of the solution and the running time. In addition to the evaluation of the proposed heuristic with the derived upper bounds and optimal solutions, the impact of topology diversity to the NWL-UFL m-trail problem will be investigated. All the three metrics are examined with respect to different network sizes (i.e., the number of nodes or links) and topology densities (i.e., nodal degree), which will be presented in the following three subsections.

A. Performance Comparison

We first examine the proposed scheme by comparing it with the derived bounds as shown in Fig. 3(a) and Fig. 3(b) in term of total cover length (i.e., $||T||$). Fig. 3(a) shows the cover length when the average nodal degree of the topologies with 50 nodes is increased from 2.5 to 49, where three curves by the proposed *RSTA+GLS*, Theorem 2, and Theorem 3, are plotted respectively. It is shown that *RSTA+GLS* yields an average gap of less than 22.7% to the best bound of the two, which is given by Theorem 2 when the topology is sparse while by Theorem

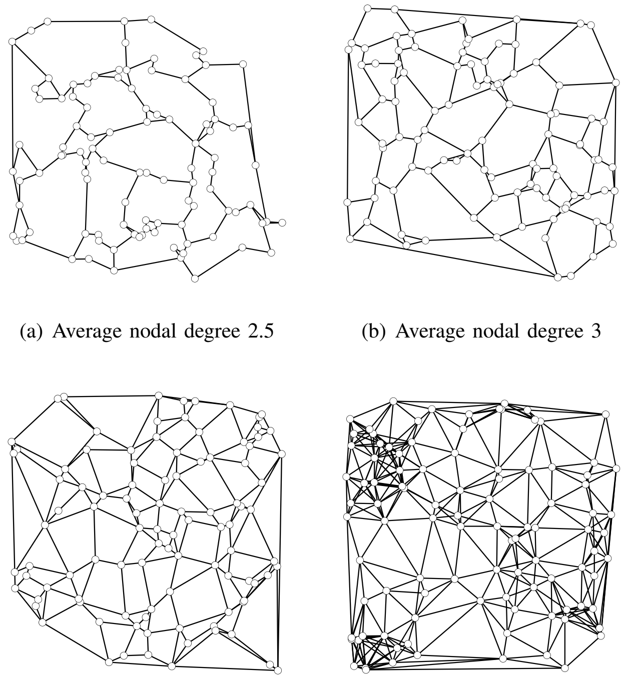


(a) Networks with 50 nodes and different number of links.



(b) Plot with nodal degree from 2.5 to 7.

Fig. 3. The cover length versus average nodal degree (900 randomly generated topologies with 50 nodes).



(a) Average nodal degree 2.5 (b) Average nodal degree 3

(c) Average nodal degree 4 (d) Average nodal degree 8

Fig. 4. Examples of random 100 node backbone networks with different number of links.

3 in denser ones. Fig. 3(b) provides a closer look at the range of nodal degree $[2.2, 6]$ in Fig. 3(a), where the bounds given by Theorem 2 and Eq. (12) are also plotted. Note that Eq. (12) is a bound that intrinsically exists due to the initial assignment of b. To summarize, the proposed *RSTA+GLS* can yield solutions very close to the lower bounds that we derived, and the two bounds in Theorem 2 and Theorem 3 can well account for sparse and dense topologies, respectively. Further, since the bounds derived in Section IV for general graphs do not consider sufficient topological features, both bounds are not effective when the network nodal degrees are quite low (e.g., less than 3). Nonetheless, the minimal gap is observed when the nodal degree ≈ 4.

Three schemes were implemented and compared in the simulation. *RSTA+GLS* (denoted by $\times$ on the charts) refers to the proposed heuristic. *CA*$_S$ (denoted by $+$ on the charts) corresponds to the method that each m-trail is allocated one after the other to distinguish each pair of links using any Dijkstra's algorithm based scheme, such that L-UFL can be achieved at an arbitrarily chosen monitoring node (MN). Such a method is generic and has been considered in a number of previously reported studies [5], [28], [29]. Rather than just for L-UFL at a single MN with *CA*$_S$, *CA*$_N$ (denoted by $\triangle$ on the charts) achieves NWL-UFL by performing *CA*$_S$ sequentially at each node. In our implementation loopback switching is allowed, thus *CA*$_S$ and *CA*$_N$ algorithms are modified such that an L-UFL solution for any node can always be found in a connected graph according to Lemma 1.

The comparison results are shown in Fig. 5. We normalize the cover length over the number of links and both the numbers of links and nodes, denoted as $||T||_E$ and $||T||_{EN}$, respectively. $||T||_E$ is a measure on the average number of WLs per edge while $||T||_{EN}$ is on the average number of WLs per edge per MN, respectively. Clearly we have $||T||_E = \frac{||T||}{m}$ and $||T||_{EN} = \frac{||T||}{n \cdot m}$.

In Figs. 5(a) and (b), we observed that the proposed *RSTA+GLS* consumes much smaller $||T||_E$ and $||T||_{EN}$, respectively, than that by *CA*$_N$ when the number of network nodes is 100 and the average nodal degree is increased from 2.5 to 6. It is clearly demonstrated that *RSTA+GLS*, which enables all the 100 nodes to individually serve as L-UFL MNs, can achieve similar cover lengths to that by *CA*$_S$ where only a single L-UFL MN is supported. This demonstrates the effect of on-off status sharing among the on-trail nodes of a common m-trail, such that the increase of the number of MNs would lead to little increase in the total cover length. They also show that $||T||_E$ and $||T||_{EN}$ decrease when the nodal degree is increased (or as the number of links is increased) due to the better connectivity which yields larger design space for allocating the m-trails. Fig. 5(c) and (d) plot $||T||_E$ and $||T||_{EN}$ against the number of nodes given the nodal degree ≈ 4, which yield similar comparison results as in Figs. 5(a) and (b). Fig. 5(e) plots the solutions by *RSTA+GLS* and the best lower bound derived in Section IV-A. One can verify that the normalized cover length has a logarithmic relation with the increasing number of nodes (from 100 to 1,000) when the average nodal degree is kept as a constant (i.e., ≈ 4).

Finally, the computation efficiency of the proposed *RSTA+GLS* is examined, which is shown in Fig. 6. We can clearly see that our scheme yields significant better computation efficiency than *CA*$_N$. Note that *CA*$_S$ takes shorter

Network [27]	Graph n	m	diam.	Thm. 2	Thm. 3	$\|\|T\|\|$ *RSTA+GLS*	*CA$_N$*	$\|\|T\|\|_E$ *RSTA+GLS*	*CA$_N$*	#m-trails *RSTA+GLS*	*CA$_N$*	Time [s] *RSTA+GLS*	*CA$_N$*
Pan-European	16	22	6	79	82	107	241	2.43	5.47	7	24	0.39	0.42
German	17	26	6	84	97	128	368	2.46	7.07	8	33	0.51	0.65
ARPA	21	25	7	108	95	140	354	2.80	7.08	6	33	0.36	0.64
European	22	45	5	127	171	231	617	2.56	6.85	11	53	2.10	4.59
USA	26	42	8	155	161	229	667	2.72	7.94	9	53	2.16	3.99
Nobel EU	28	41	8	168	158	248	689	3.02	8.40	7	50	0.87	3.38
Italian	33	56	9	200	217	329	1113	2.93	9.93	10	66	4.83	11.07
Cost 266	37	57	8	225	221	343	918	3.00	8.05	8	55	2.04	14.45
North Amer.	39	61	10	241	237	378	1020	3.09	8.36	8	50	2.31	15.48
NSFNET	79	108	16	579	426	760	2634	3.51	12.19	9	100	6.05	227.34

TABLE II
RESULTS BY THE PROPOSED *RSTA+GLS* AND *CA$_N$* ON SOME WELL-KNOWN NETWORKS.

computation time since only a single MN is considered, compared with the case where *RSTA+GLS* enables all the nodes as MNs each being able to perform L-UFL. With *RSTA+GLS*, the NWL-UFL m-trail problem on a 1000-node network can be solved within 3 minutes.

Table II provides results of *RSTA+GLS* and *CA$_N$* on some well-known network topologies taken from [27]. The number of nodes, links and the diameter in hops of every topology graph is also shown in the first three columns of the table. The results are similar to that on randomly generated graphs as in Fig. 3 regarding the number of required m-trails (i.e., $||T||$), average WLs per link (i.e., $||T||_E$), and running time (in seconds) under various network sizes and topology densities.

To summarize the comparison results above, the proposed *RSTA+GLS* is witnessed to achieve desired computation efficiency in handling large networks, and its feasibility in the operation of future all-optical backbone is proved in terms of resource consumption for achieving NWL-UFL using bi-directional m-trails. For example, an optical network with 288 links and 100 nodes takes averagely 3.4 WLs along each link, and each MN only consumes approximately 0.034 WLs per link for achieving NWL-UFL. Thus, in the case that each fiber has 100 wavelength channels and 10 fibers bundled in a single conduit, the system needs to spend only 0.68% of the total capacity for achieving NWL-UFL of any single conduit.

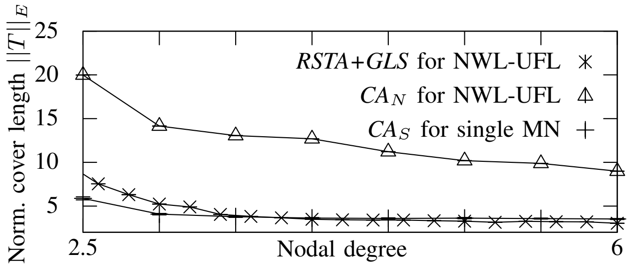
B. The impact of Topology Diversity

Fig. 7 shows the performance of the proposed *RSTA+GLS* on topologies with $n = 25, 50, 100, 200$ as the increase of the average nodal degree of the topologies. Instead of plotting $||T||$ for each case, we normalize the results on topologies of n nodes by $(n-1)^2$ (which is denoted as $||T||_{(n-1)^2}$). Note that a line graph P_n and a complete graph K_n represent the least and most densely meshed topology with n nodes, respectively. Both topologies have an optimal solution $||T|| = (n-1)^2$ as derived in Section IV, which is supposed to be the largest possible $||T||$ for any topology with n nodes. We expect that all other topologies should yield solutions less than $(n-1)^2$. This is attested in the figure, where the curves of $||T||_{(n-1)^2}$ is observed as a "V" shape for each specific n value. It also shows that $||T||_{(n-1)^2}$ is minimal when the average nodal degree of the topologies is in the range of $[3.5, 4.5]$ of all the values of n investigated. This demonstrates that the proposed scheme can be the most suitable to work in optical networks with moderate connectivity.

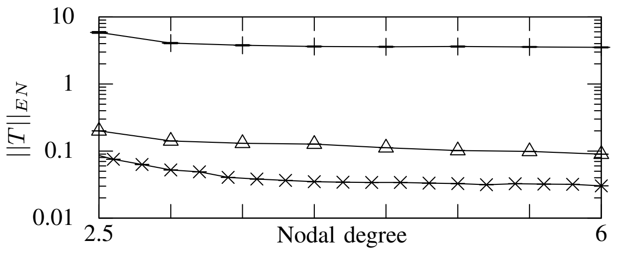
VII. CONCLUSIONS

The paper investigated the problem of failure localization of any single link in all-optical mesh WDM networks via a generic out-of-band monitoring scheme - monitoring trails (m-trails). The paper firstly defined the Network-Wide Local Unambiguously Failure Localization (NWL-UFL) scenario for the m-trail allocation problem, where each node can perform UFL based on locally available on-off status of the m-trails traversing the node (also referred to as L-UFL). This is a novel failure localization framework in all-optical mesh networks which can be operated completely in the optical domain without any aid from the upper-layer control signaling.

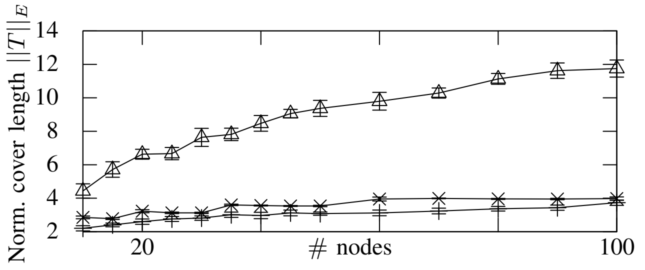
To gain deep understanding of the formulated problem, bound analysis was conducted on the total cover length of the problem, where optimal solutions on a number of special graphs were developed and proved (i.e., lines, stars, and complete graphs), and decent lower bounds for general graphs were derived. To specifically solve the m-trail problem in general graphs, a novel heuristic algorithm was introduced. Inspired by the design of our optimal construction for complete graphs, the proposed heuristic algorithm employs spanning tree as the structure for m-trails. The randomly generated spanning trees are deployed into the network at the beginning. To ensure global code uniqueness of each link, an intelligent yet very fast code swapping mechanism, called GLS (greedy link swapping), was implemented so as to resolve possible collisions among preliminarily assigned link codes. Extensive simulation on thousands of randomly generated topologies was conducted to verify the proposed heuristic algorithm, which showed its performance and scalability in comparison with a generic approach based on Dijkstra's algorithm (i.e., *CA$_N$*), mainly due to the on-off status sharing among on-trail nodes. We have further compared the proposed scheme with the derived upper bounds and the optimal solutions, which demonstrated its practical significance as a design guidance for related problems. Finally, we have investigated the impact of topology diversity, and found that the best performance (i.e.,



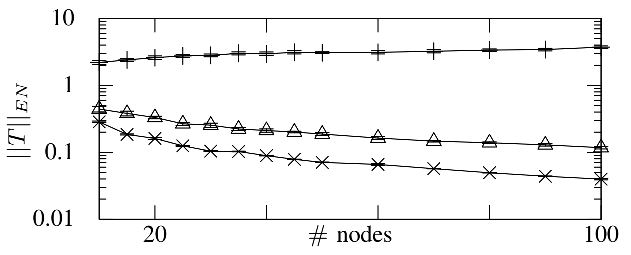
(a) 1560 randomly generated topologies with 100 nodes.



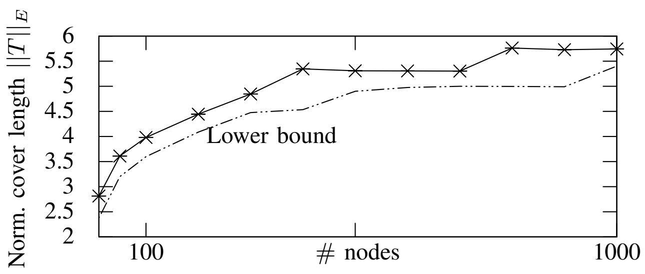
(b) 1560 randomly generated topologies with 100 nodes.



(c) 2400 randomly generated topologies with nodal degree ≈ 4.



(d) 2400 randomly generated topologies with nodal degree ≈ 4.



(e) 240 randomly generated topologies with nodal degree ≈ 4.

Fig. 5. The normalized cover length by *RSTA+GLS* ($\times$) and CA_S ($+$) and CA_N ($\triangle$).

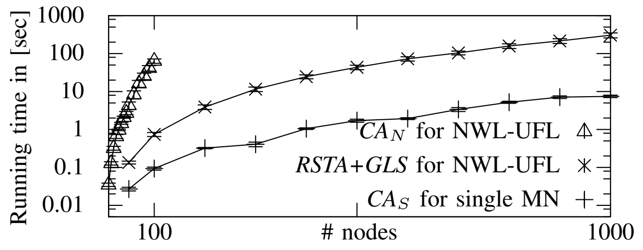


Fig. 6. The running time of 1640 randomly generated topologies with nodal degree ≈ 4.

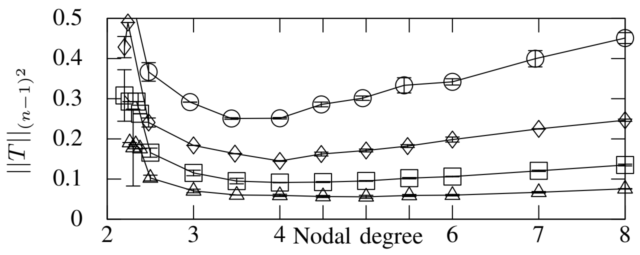


Fig. 7. Plot of $||T||_{(n-1)^2}$ with $n = 25(\circ), 50(\diamond), 100(\Box), 200(\triangle)$ (1280 randomly generated topologies.

the minimum cover length) occurs when the average nodal degree of topologies ≈ 4.

REFERENCES

[1] D. Zhou and S. Subramaniam, "Survivability in optical networks," *IEEE Network*, vol. 14, no. 6, pp. 16–23, 2000.

[2] K. Lee and E. Modiano, "Cross-Layer Survivability in WDM-Based Networks," in *IEEE INFOCOM*, 2009, pp. 1017–1025.

[3] M. Médard, R. Barry, S. Finn, W. He, and S. Lumetta, "Generalized loop-back recovery in optical mesh networks," *IEEE/ACM Transactions on Networking (TON)*, vol. 10, no. 1, p. 164, 2002.

[4] B. Wu, P.-H. Ho, K. Yeung, J. Tapolcai, and H. Mouftah, "Optical Layer Monitoring Schemes for Fast Link Failure Localization in All-Optical Networks," *IEEE Communications Surveys & Tutorials*, vol. 13, no. 1, pp. 114 –125, quarter 2011.

[5] S. Ahuja, S. Ramasubramanian, and M. Krunz, "Single link failure detection in all-optical networks using monitoring cycles and paths," *IEEE/ACM Transactions on Networking (TON)*, vol. 17, no. 4, pp. 1080–1093, 2009.

[6] ——, "SRLG Failure Localization in All-Optical Networks Using Monitoring Cycles and Paths," in *IEEE INFOCOM*, 2008, pp. 181–185.

[7] B. Wu, K. Yeung, B. Hu, and P. H. Ho, "M^2-CYCLE: an Optical Layer Algorithm for Fast Link Failure Detection in All-Optical Mesh Networks," *Elsservier Computer Networks*, vol. 55, no. 3, pp. 748 – 758, 2011.

[8] B. Wu, K. Yeung, and P.-H. Ho, "Monitoring Cycle Design for Fast Link Failure Localization in All-Optical Networks," *IEEE/OSA Journal of Lightwave Technology*, vol. 27, no. 10, pp. 1392–1401, 2009.

[9] J. Tapolcai, L. Rónyai, and P.-H. Ho, "Optimal solutions for single fault localization in two dimensional lattice networks," in *IEEE INFOCOM Mini-Symposium*, San Diego, CA, USA, 2010.

[10] A. Haddad, E. Doumith, and M. Gagnaire, "A meta-heuristic approach for monitoring trail assignment in WDM optical networks," in *Int. Workshop on Reliable Networks Design and Modeling (RNDM)*. IEEE, 2010, pp. 601–607.

[11] Y. Zhao, S. Xu, X. Wang, and S. Wang, "A new heuristic for monitoring trail allocation in all-optical wdm networks," in *IEEE GLOBECOM*, dec. 2010, pp. 1 –5.

[12] N. Ogino and H. Nakamura, "All-optical monitoring path computation based on lower bounds of required number of paths," in *IEEE ICC*, 2011.

[13] B. Wu, P.-H. Ho, and K. Yeung, "Monitoring trail: On fast link failure localization in all-optical wdm mesh networks," *IEEE/OSA Journal of Lightwave Technology*, vol. 27, no. 18, pp. 4175–4185, 2009.
[14] J. Tapolcai, B. Wu, and P.-H. Ho, "On monitoring and failure localization in mesh all-optical networks," in *IEEE INFOCOM*, Rio de Janero, Brasil, 2009, pp. 1008–1016.
[15] N. Harvey, M. Patrascu, Y. Wen, S. Yekhanin, and V. Chan, "Non-Adaptive Fault Diagnosis for All-Optical Networks via Combinatorial Group Testing on Graphs," in *IEEE INFOCOM*, 2007, pp. 697–705.
[16] M. Mao and K. Yeung, "Super Monitor Design for Fast Link Failure Localization in All-Optical Networks," in *IEEE ICC*, 2010.
[17] J. Tapolcai, P.-H. Ho, L. Rónyai, P. Babarczi, and B. Wu, "Failure localization for shared risk link groups in all-optical mesh networks using monitoring trails," *IEEE/OSA Journal of Lightwave Technology*, vol. 29, no. 10, pp. 1597 –1606, may.
[18] E. A. Doumith, S. A. Zahr, and M. Gagnaire, "Monitoring-tree: An innovative technique for failure localization in WDM translucent networks," in *IEEE GLOBECOM*, 2010, pp. 1–6.
[19] B. Wu, P.-H. Ho, J. Tapolcai, and X. Jiang, "A novel framework of fast and unambiguous link failure localization via monitoring trails," in *IEEE INFOCOM WIP*, San Diego, 2010.
[20] C. Mas and P. Thiran, "An efficient algorithm for locating soft and hard failures in wdm networks," *IEEE Journal on Selected Areas in Communications*, vol. 18, no. 10, pp. 1900–1911, 2000.
[21] S. Stanic, S. Subramaniam, G. Sahin, H. Choi, and H. A. Choi, "Active monitoring and alarm management for fault localization in transparent all-optical networks," *IEEE Transactions on Network and Service Management*, vol. 7, no. 2, pp. 118–131, 2010.
[22] D. Papadimitriou and E. Mannie, "Analysis of generalized multi-protocol label switching (gmpls)-based recovery mechanisms (including protection and restoration)," RFC 4428, March, Tech. Rep., 2006.
[23] P. Chołda, J. Tapolcai, T. Cinkler, K. Wajda, and A. Jajszczyk, "Quality of resilience QoR as a network reliability characterization tool," *IEEE Network Magazine*, vol. 23, no. 2, pp. 11–19, March/April 2009.
[24] D. Aldous, "The random walk construction of uniform spanning trees and uniform labelled trees," *SIAM Journal on Discrete Mathematics*, vol. 3, no. 4, pp. 450–465, 1990.
[25] A. Broder, "Generating random spanning trees," in *Annual Symposium on Foundations of Computer Science*. IEEE Computer Society, 1989, pp. 442–447.
[26] "LEMON: A C++ Library for Efficient Modeling and Optimization in Networks," http://lemon.cs.elte.hu.
[27] S. Orlowski, M. Pióro, A. Tomaszewski, and R. Wessäly, "SNDlib 1.0–Survivable Network Design Library," in *Proc. Int. Network Optimization Conference (INOC)*, April 2007, http://sndlib.zib.de.
[28] H. Zeng, C. Huang, and A. Vukovic, "A novel fault detection and localization scheme for mesh all-optical networks based on monitoring-cycles," *Photonic Communication Networking*, pp. 277–286, 2006.
[29] C. Assi, Y. Ye, A. Shami, S. Dixit, and M. Ali, "A hybrid distributed fault-management protocol for combating single-fiber failures in mesh based DWDM optical networks," in *IEEE GLOBECOM*, 2002, pp. 2676–2680.

APPENDIX

Proof of Theorem 2: Let the *reputation* of t_i be denoted by $r(t_i)$, which represents the number of nodes along m-trail t_i that are aware of the on-off status of t_i. A trivial upper bound on $r(t_i)$ is $|t_i|+1$ if t_i is loopless; or formally

$$r(t_i) \leq |t_i| + 1. \tag{5}$$

Let us define a matrix Ω with n columns and b rows, where

$$\omega_{i,j} = \begin{cases} \frac{|t_i|}{r(t_i)} & \text{the } i\text{-th m-trail traverses node } v_j, \\ 0 & \text{otherwise.} \end{cases} \tag{6}$$

The length of t_i can be expressed as

$$\sum_{j=1}^{n} \omega_{i,j} = \sum_{j=1}^{n} \frac{|t_i|}{r(t_i)} = |t_i|. \tag{7}$$

Thus we have

$$\sum_{i=1}^{b}\sum_{j=1}^{n} \omega_{i,j} = \sum_{i=1}^{b} |t_i| = ||T|| \tag{8}$$

As a lower bound on $\omega_{i,j}$ according to Eq. 5, we have

$$\omega_{i,j} = \frac{|t_i|}{r(t_i)} \geq \frac{|t_i|}{|t_i|+1} \geq \frac{1}{2} \tag{9}$$

Let us denote the set of m-trails passing through node v_j by T^j. Next, we give a lower bound on $\sum_{i=1}^{b} \omega_{i,j}$ for each node v_j, which is denoted by ω_j, formally

$$\omega_j = \sum_{i=1}^{b} \omega_{i,j} = \sum_{t_i \in T^j} \frac{|t_i|}{r(t_i)} \geq \sum_{t_i \in T^j} \frac{|t_i|}{|t_i|+1} \tag{10}$$

where $t_i \in T^j$ means that m-trail t_i passes through node v_j. Note that the cover length is the sum of ω_j for all nodes, formally

$$||T|| = \sum_{j=1}^{n} \omega_j, \tag{11}$$

Let us denote the distance between node v_j and link $e \in E$ by $\delta(v_j, e)$, which equals to the length of the shortest path between node v_j and the closest adjacent node of e plus 1. Without loss of generality, the distance can be simply measured in hops. Let $\delta_j(k)$ denote the number of links whose distance from node v_j is at least k. For example, $\delta_j(1) = m$ as every link is at least 0 hop from the node, and $\delta_j(2) = m - \Delta_j$, where Δ_j is the nodal degree of j. Let n_j denote the maximum δ distance from node v_j, thus $\delta_j(n_j+1) = 0$ and $\delta_j(n_j) > 0$. Note that such n_j always exist because there are no links with distance $n-1$. Note that $\delta_j(\mu)$ is a monotonically non-increasing function of μ (i.e. $\delta_j(1) \geq \ldots \geq \delta_j(n_j) \geq 1$). To localize any single failure at node v_j among links whose distance from v_j is at least $\mu - 1$, there must be $\lceil \log_2 \delta_j(\mu) \rceil$ m-trails with at least a distance of μ.

From T^j we select $\lceil \log_2 \delta_j(n_j) \rceil$ m-trails whose length is at least n_j. By Eq. (10) they contribute to ω_j at least

$$\frac{n_j - 1}{n_j} \lceil \log_2 \delta_j(n_j - 1) \rceil = \frac{n_j - 1}{n_j} (\lceil \log_2 \delta_j(n_j - 1) \rceil - \lceil \log_2 \delta_j(n_j) \rceil).$$

In a similar manner, from the remaining m-trails in T^j we may select as many as $\lceil \log_2 \delta_j(n_j - 2) \rceil - \lceil \log_2 \delta_j(n_j - 1) \rceil$ m-trails whose lengths are at least $n_j - 2$. Their contribution to ω_j is at least

$$\frac{n_j - 2}{n_j - 1} (\lceil \log_2 \delta_j(n_j - 2) \rceil - \lceil \log_2 \delta_j(n_j - 1) \rceil),$$

by Eq. (10) again. Continuing in this way for $\mu = n_j, \ldots, 1$ we obtain a collection of m-trails in T^j such that the following inequality follows

$$\omega_j \geq \sum_{\mu=1}^{n_j} \frac{\mu - 1}{\mu} \left(\lceil \log_2 \delta_j(\mu - 1) \rceil - \lceil \log_2 \delta_j(\mu) \rceil \right) \tag{12}$$

By summing up ω_j for every node according to Eq. (11) we get a lower bound on cover length in Eq. (1). ■

János Tapolcai received his M.Sc. ('00 in Technical Informatics), and Ph.D. ('05 in Computer Science) degrees in Technical Informatics from Budapest University of Technology and Economics (BME), Budapest, Hungary. Currently he is an Associate Professor at the High-Speed Networks Laboratory at the Department of Telecommunications and Media Informatics at BME. His research interests include applied mathematics, combinatorial optimization, mathematical programming, optical and IP level routing and survivability, availability analysis and distributed computing. He has been involved in several related European and Canadian projects. He is an author of over 80 scientific publications, and is the recipient of the Best Paper Award in ICC'06 and in DRCN'11.

Pin-Han Ho received his B.Sc. and M.Sc. degrees from the Electrical and Computer Engineering, Department of National Taiwan University in 1993 and 1995, respectively. He started his Ph.D. studies in 2000 at Queen's University, Kingston, Ontario, Canada, focusing on optical communications systems, survivable networking, and QoS routing problems. He finished his Ph.D. in 2002, and joined the Electrical and Computer Engineering Department at the University of Waterloo as an assistant professor in the same year. He is the author/co-author of more than 100 refereed technical papers and book chapters, and the co-author of a book on optical networking and survivability. He is the recipient of the Distinguished Research Excellence Award in the ECE Department at the University of Waterloo, the Early Researcher Award in 2005, the Best Paper Award at SPECTS '02 and the ICC '05 Optical Networking Symposium, and the Outstanding Paper Award in HPSR '02.

Lajos Rónyai is a research professor with the Informatics Laboratory of the Computer and Automation Institute of the Hungarian Academy of Sciences. He leads a research group there which focuses on theoretical computer science and discrete mathematics. He is also a full professor at the Mathematics Institute of the Budapest University of Technology and Economics. He received his PhD in 1987 from the Eötvös Loránd University, Budapest. His research interests include efficient algorithms, complexity of computation, algebra, and discrete mathematics.

Bin Wu received the B.Eng. degree from Zhe Jiang University, Hangzhou, China, in 1993, M.Eng. degree from University of Electronic Science and Technology of China, Chengdu, China, in 1996, and PhD degree from the University of Hong Kong, Hong Kong, in 2007. During 1997-2001, he served as the department manager of TI-Huawei DSP co-lab in Huawei Tech. Co. Ltd, Shenzhen, China. Currently he is a postdoctoral research fellow at the University of Waterloo, Waterloo, Canada.