Optimal Solutions for Single Fault Localization in Two Dimensional Lattice Networks

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

***Abstract*—Achieving fast, precise, and scalable fault localization has long been a highly desired feature in all-optical mesh networks. Monitoring tree (m-tree) is an interesting method that has been introduced as the most general monitoring structure for achieving unambiguous failure localization (UFL). Ideally, with J m-trees one can monitor up to $2^J - 1$ links when a single failure has to be located. Such a logarithmic behavior has also been observed in numerous case studies of real life network topologies [1], [2]. It is expected that the m-tree framework will lead to a highly scalable link failure monitoring mechanism for not only all-optical mesh networks, but any possible future information system with mesh topologies, such as all-optical mesh networks, touch panels, quantum computing, and VLSI. It is an important task to investigate the extent such an optimal logarithmic behavior may hold, in particular in practically relevant network topologies. As an endeavor toward this goal, the paper investigates the problem by identifying essentially tight logarithmic bounds for two dimensional lattice networks. Experiments are conducted to show the feasibility and performance of the proposed constructions.**

I. Introduction

Failure localization is critical in all-optical mesh networks, and has been extensively studied in the past decade [2]–[7]. Due to the transparency in the data plane, a single failure may trigger a large number of redundant alarms [8], [9], which not only increases the management cost of control plane, but also makes the failure localization difficult. One of the most commonly adopted approaches is to deploy optical monitors responsible for generating alarms when a failure is detected. The alarm signals are then collected via either dedicated channels or any other dissemination approach such that any routing entity can localize the fault event. It is a critical task to minimize the alarm code length and the number of monitors allocated to the topology, while achieving unambiguous failure localization (UFL).

In general, a network topology is modeled as a graph with two directed links between some pairs of nodes in opposite directions. Each monitoring lightpath is composed of a set of links in the network. In the literature, three monitoring structures have been reported and categorized according to the additional restrictions made for each monitoring lightpath. The most general structure is called monitoring-tree (*m-tree*), in which the set of links are interconnected and can be covered by a route [10], [11]. In this case the nodes can loop-back the optical signals coming from the transmission fiber into its reception fiber. In monitoring-trails (*m-trails*) [1], [11], on the other hand, optical loop back switching is not allowed. Both m-tree and m-trail can traverse through a node multiple times. Historically, even more restricted monitoring structures have been investigated in [4], [5], called monitoring-cycles (*m-cycles*). With m-cycles, the transmitter and receiver of a monitoring lightpath must be the same node, and a simple cycle is formed.

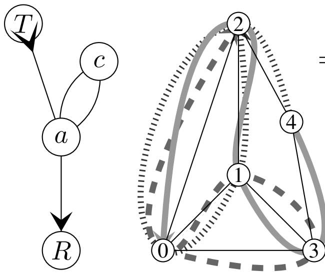
Ideally, J m-trees can localize up to $2^J - 1$ single link failures. In this case, each m-tree corresponds to a single bit in the alarm code, which results in a monitoring system with J-bit alarm codes. In other words, ideally $\lceil \log_2(|E|+1) \rceil$ m-trees would be sufficient for localizing any single link failure in a network with linkset E. In [1] extensive simulations were conducted on thousands of randomly generated topologies investigating the impact of topology diversity on m-trail solutions. Such a logarithmic solution can be achieved on almost all of the tested network topologies without any nodes of degree two and with small diameters, and for many others $\lceil \log(|E|+1) \rceil + 1$ m-trails were enough.

Although the heuristic in [1] managed to solve networks with a few thousands of links, a question is whether such a logarithmic nature is still valid for really large networks with large diameters like the Internet or even in some other applications, such as VLSI design/monitoring, quantum computing grids, and large-scale switches. To the best of our knowledge, the only analytical studies that investigated the upper bound on the number of m-trails for UFL were reported for ring, rectangular lattice, tree and fully connected topologies.

Motivated by the significance of the problem, we intended to develop (nearly) optimal logarithmic solutions in the m-trees scenario for 2-dimensional lattices (i.e. Manhattan grids). The paper provides rigorous proofs on the obtained upper bound $3 + \lceil \log(|E|+1) \rceil$, which are *essentially tight* in the sense that except a small constant it match the lower bound $\lceil \log_2(|E|+1) \rceil$. We expect that the developed analytical constructions can solidly contribute to the industrial solutions of monitoring and fault localization in mesh communication networks and information systems.

The rest of the paper is organized as follows. Section II presents the background and problem formulation for m-tree

Research supported in part by OTKA grants NK 72845, K77476, K77778, and 67651. János Tapolcai was supported by High Speed Network Laboratory.



Link	t_1	t_2	t_3
(0,1)	1	0	1
(0,2)	1	1	1
(0,3)	1	0	0
(1,2)	0	1	1
(1,3)	1	1	0
(2,4)	0	0	1
(3,4)	0	1	0

(a) M-tree (b) An m-tree solution (c) Alarm code table

Fig. 1. Fast link failure localization based on m-trees.

design. In Section III, an essentially optimal solution is given for rectangular grids. Section IV concludes the paper.

II. Background

A. Monitoring Trees (M-trees)

In brief, an m-tree can traverse a common link in different directions via loop-back switching and a node by multiple times. By allocating monitoring lightpaths according to a feasible m-tree solution, a remote routing entity can localize a single failure by collecting the alarm signals issued by the corresponding monitors of m-trees in a timely manner. Let the transmitter and receiver of the lightpath of an m-tree be denoted as T and R, respectively. As shown in Fig. 1(a), the lightpath can be pre-cross-connected along the route $T \to a \to c \to a \to R$.

Generally, an m-tree solution consists of a set of J m-trees $t_1, t_2, \ldots, t_J$. Upon a single link failure, the monitor of any m-tree traversing the failed link will generate an alarm. An alarm code $[a_1, \ldots, a_{J-1}, a_J]$ can be formed by reading the status of each m-tree, where $a_k = 1$ means that the monitor on m-tree t_j alarms and $a_k = 0$ otherwise. Fig. 1(b) shows a solution with three m-trees t_1, t_2, t_3. If link $(0,1)$ fails, the monitors on t_1 and t_3 will alarm to produce the alarm code $[1,0,1]$. Similarly, if link $(0,2)$ fails, the monitors on all the three m-trees will alarm and the resulting alarm code is $[1,1,1]$. The alarm code table (ACT) in Fig. 1(c) is available in each network routing entity, which maintains all the possible alarm codes that could be received at a network entity. Thus, the network entity can unambiguously localize a particular single link failure by matching the alarm code in the ACT.

B. Problem Definition - Deployment of M-Trees/Trails/Cycles

The problem is a structured variant of combinatorial group testing [12], [13]. In [10] the problem was called as combinatorial group testing on graphs.

Let a graph denoted as $G(E,V)$ be given with $|E|$ links and $|V|$ nodes. In order to achieve UFL, each link e must be assigned with a unique binary alarm code $c(e) = A_e = [a_1^e, a_2^e, \ldots\ldots a_J^e]$, where J is the length of alarm code, and a_l^e is a binary digit, which is 1 if the l^{th} m-tree, denoted by t_l, traverses through this link and 0 otherwise. The m-tree t_l has to traverse through all the links e with $a_l^e = 1$ while avoiding to take any link with $a_l^e = 0$. Let L_l denote the l^{th} *linkset* which contains the set of links with $a_l^e = 1$. One may impose three different conditions on L_l, leading to three different failure location models:

m-tree: L_l must be connected;

m-trail: L_l must be an m-tree, and besides; every node along the tree in $G(E,V)$ must have an even nodal degree except the source and destination;

m-cycle: L_l must be an m-trail and the source and destination node must be the same node.

A corresponding solution is referred to as *m-tree/trail/cycle formation*. In all three models, the theoretical lower bound on J is $J \geq \lceil \log_2(|E|+1) \rceil$. We are interested in when the above bound is essentially tight in the m-tree model. An m-tree formation is *essentially optimal* if totally $J = \lceil \log_2(|E|+1) + c \rceil$ m-trees are sufficient for achieving UFL, where c is a small constant. The main contributions of the paper are deterministic polynomial time constructions that achieve essentially optimal m-tree formation on 2-dimension dimensional lattice topologies.

III. Essentially Optimal M-tree Solution in a Rectangular Grid

This section considers general 2-dimensional lattices denoted by $S_{m,n}$, where m and n corresponds to the number of links in the vertical and horizontal direction, respectively. Harvey, *et al.* [10] provided an $2 + 6 \cdot \lceil \log_2(n+1) \rceil$ upper bound on the number of m-trees for a square grid topology with $m = n$. In fact since we have $|E| = 2n^2 + 2n < 2(n+1)^2$, it leads to

$$2 + 6 \cdot \lceil \log_2(n+1) \rceil < 2 + 6 \lceil \log_2 \sqrt{\frac{|E|}{2}} \rceil \approx 3 \lceil \log_2 |E| \rceil$$

In this section, we improve the bound in [10] and generalize it to a 2-D grid graph with arbitrary m and n. The proposed construction provides an essentially optimal solution with $3 + \lceil \log_2(|E|+1) \rceil$ m-trees. Our strategy is to generalize the m-tree solution for "chocolate bar graphs" (i.e., a special case of $S_{m,n}$ with either $n = 1$ or $m = 1$) to arbitrary lattice topologies.

A. M-Tree Solution for Chocolate Bar Graphs

A general chocolate bar graph is denoted as $C_n(E,V)$, which has $|V| = 2n+2$ vertices denoted as $x_{1,0}, \ldots, x_{1,n}$ (the lower points), and $x_{0,1}, \ldots, x_{0,n}$ (the upper points). Fig. 2(a) shows an example of a chocolate bar topology with $n = 5$. For the linkset E, we have lower horizontal links $(x_{1,i}, x_{1,i+1}) \in E$, higher horizontal links $(x_{0,i}, x_{0,i+1}) \in E$ for $i = 0, \ldots, n-1$, and the middle vertical links $(x_{0,i}, x_{1,i}) \in E$ whenever $i = 0, \ldots, n$. Without loss of generality, we simply index each node of C_n in the same way as for rows and columns of a matrix.

Theorem 1: A chocolate bar graph $C_n(E,V)$ takes $J = \lceil \log_2(n+1) \rceil + 2$ m-trails to achieve UFL for $J > 2$, which is at most $\lceil 0.42 + \log_2(|E|+2) \rceil$ m-trails.

Proof: The proof is developed by way of a polynomial time deterministic construction composed of two steps. We

TABLE I

THE ELEMENTS OF $\mathbb{F}_8$ IN MODULO $1 + x + x^3$.

Exponential	Polynomial	Code
α^0	1	$\mathbf{r}^1 = 100$
α^1	x	$\mathbf{r}^2 = 010$
α^2	x^2	$\mathbf{r}^3 = 001$
α^3	$x^3 = 1 + x \mod 1 + x + x^3$	$\mathbf{r}^4 = 110$
α^4	$x + x^2$	$\mathbf{r}^5 = 011$
α^5	$x \cdot (x + x^2) = 1 + x + x^2 \mod 1 + x + x^3$	$\mathbf{r}^6 = 111$
α^6	$x \cdot (1 + x + x^2) = 1 + x^2 \mod 1 + x + x^3$	$\mathbf{r}^7 = 101$

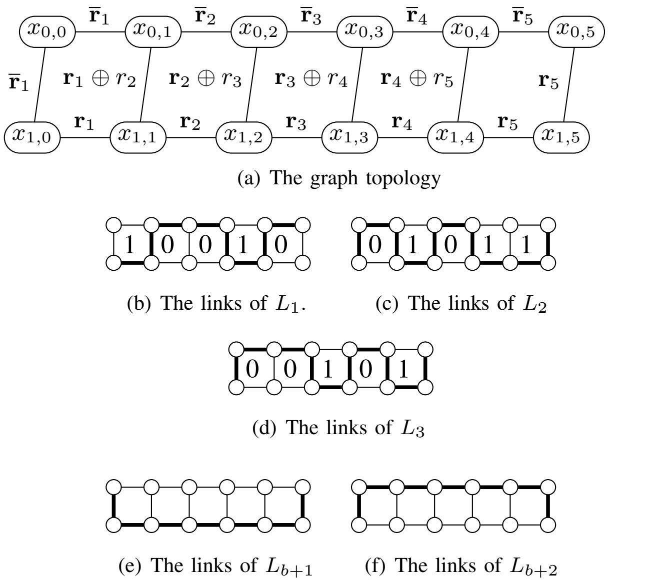


Fig. 2. An example of chocolate bar graph, and the optimal m-tree solution. The bits corresponding each bit position are drawn in each 1×1 rectangular. The $\mathbf{r}^1, \mathbf{r}^2, \dots \mathbf{r}^n$ codes assigned to the links are listed in the Table I

will first introduce the construction, and then explain in details how the construction can achieve the desired bound on the number of m-trees in a chocolate bar topology.

1) Alarm codes for the chocolate bar graph: Let us assign binary alarm codes to the links of C_n in the following way (see also Fig. 2). We first generate n bitvectors $\mathbf{r}^1, \mathbf{r}^2, \dots \mathbf{r}^n$ of length b, where $\mathbf{r}^{i+1}$ is assigned to a lower horizontal link $(x_{1,i}, x_{1,i+1}) \in E$, where $i = 0, \dots, n-1$. The generation of these codes is provided in Lemma 1. On the other hand, the higher horizontal link $(x_{0,i}, x_{0,i+1}) \in E$ we assign to the bitwise complement of $\mathbf{r}^{i+1}$, denoted by $\bar{\mathbf{r}}^{i+1} = \mathbf{r}^{i+1} \oplus \mathbf{1}$ where $\oplus$ stands for the bitwise modulo 2 addition (XOR) and $\mathbf{1}$ is the all 1 vector of length b. Also the middle vertical link $(x_{1,i}, x_{0,i})$ we assign the bitvector $\mathbf{r}^i \oplus \mathbf{r}^{i+1}$ for $i = 1, \dots, n-1$. Finally to the link $(x_{1,0}, x_{0,0})$ bitvector $\mathbf{r}^1 \oplus \mathbf{1}$ is assigned, and to the link $(x_{1,n}, x_{0,n})$ we attach $\mathbf{r}^n$.

In choosing the list of bitvectors $\mathbf{r}^i$, for $i = 1, \dots, n-1$, we make the following three assumptions:

(A1) The vectors $\mathbf{r}^i$ are pairwise different for $i = 1, \dots, n$.

(A2) The vectors $\mathbf{r}^i \oplus \mathbf{r}^{i+1}$ are all nonzero and pairwise different for $i = 1, \dots, n-1$.

(A3) The first coordinates (i.e., bit) of the vectors $\mathbf{r}^1$ and $\mathbf{r}^n$ are the same.

The following statement provides an approach to construct $n \leq 2^b - 1$ bitvectors $\mathbf{r}^i$ which satisfy the requirements (A1), (A2), (A3).

Lemma 1: Let $b := \lceil \log_2(n+1) \rceil$ and $b > 2$. Then a series of $n \leq 2^b - 1$ nonzero codes $\mathbf{r}^1$, $\mathbf{r}^2$, ..., $\mathbf{r}^n$ can be generated in polynomial time to satisfy properties (A1), (A2) and (A3).

Proof: With $b := \lceil \log_2(n+1) \rceil$, $q = 2^b$ is the smallest power of 2 which is greater than n. Following the widely used technique in classical error correcting codes, our code vectors will be vectors from a linear space over the two element field $\mathbb{F}_2$. We shall consider the finite (Galois) field $\mathbb{F}_q$ with q elements.

According to Theorem 2.5 in [14] $\mathbb{F}_q$ always exists and it is a vector space of dimension b over its subfield $\mathbb{F}_2$. This way we can identify $\mathbb{F}_q$ with bit vectors of length b, where the all zero vector corresponds to the 0 element of $\mathbb{F}_q$. In particular, nonzero vectors correspond to the nonzero elements of the field. Also, according to Theorem 2.8 in [14] $\mathbb{F}_q$ contains a primitive element α, which is a nonzero element such that all the powers $\alpha = \alpha^1, \alpha^2, \dots, \alpha^{q-1}$ are pairwise different. See also Table I where the elements and the related codes are listed for $q = 8$ ($b = 3$). Finding the primitive element in q can be done in polynomial time with exhaustive search. Because any non zero element α can be verified for being a primitive element, by raising to a power and checking if the power is equals to 1 on an exponent less than $q - 1$.

We now set $\mathbf{r}^i$ to be the (bit vector of the) element α^i. Condition (A1) is satisfied as $n \leq 2^b - 1$.

Suppose now that (A2) fails. Then there must exist $1 \leq i < j < n$ such that

$$\alpha^i \oplus \alpha^{i+1} = \alpha^j \oplus \alpha^{j+1}$$

holds in $\mathbb{F}_q$. But then we have

$$\alpha^i (1 \oplus \alpha) = \alpha^j (1 \oplus \alpha)$$

which (using that $b > 1$ and hence that $1 \oplus \alpha$ is not 0) would imply that $\alpha^i = \alpha^j$, contradicting to the fact that α is a primitive element.

To establish (A3), we note that (assuming $b > 2$) α and α^n span a subspace of dimension at most 2 of $\mathbb{F}_q$ over $\mathbb{F}_2$, hence we can select the basis of $\mathbb{F}_q$ so as to both element have 0 coordinates with respect to the first basis vector. ■

2) M-trail construction for the chocolate bar graph: In the chocolate bar construction, the linkset L_j corresponding to the j^{th} bit of the vectors of the links is actually a simple path in C_n from $x_{1,0}$ to $x_{0,n}$. In the rest of the paper C_n can be also denoted by $C_{x_{1,0},x_{0,n}}$. As a result b m-trails from $x_{1,0}$ to $x_{0,n}$ are formed in C_n, each corresponding to one bit position of the vectors. An example is given as follows with $n = 5$, where the resultant 5 m-trails by the construction is shown in Fig. 2(b), 2(c), 2(d).

In addition to the above mentioned m-trails, we need to add two more m-trails to the construction. This is exemplified in

Fig. 2(e) and 2(f). Let the two m-trails correspond to linkset L_{b+1} and L_{b+2}, respectively, where L_{b+1} is composed of the links $(x_{1,0}, x_{0,0})$, $(x_{1,n}, x_{0,n})$ and the path consisting of all the links $(x_{1,i}, x_{1,i+1})$ $i = 0, \dots n-1$, while L_{b+2} is composed of the links $(x_{1,0}, x_{0,0})$, $(x_{1,n}, x_{0,n})$ along with the path consisting of all the links $(x_{0,i}, x_{0,i+1})$, $i = 0, \dots n-1$. As a result, L_{b+1} and L_{b+2} can identify whether a failed link was a horizontal or vertical link, and whether the link was $(x_{1,0}, x_{0,0})$ or $(x_{1,n}, x_{0,n})$.

Corollary 1: L_j $j = 1, \dots, b+2$ forms a single m-trail, and the m-trail is a simple path.

The corollary is intuitively holds according to the given construction.

3) Correctness of the constructed m-trail solution: We show here that the paths $L_1, \dots L_{b+2}$ are able to localize any single link failure in the chocolate bar C_n. Obviously, L_1, L_{b+1} and L_{b+2} can unambiguously localize a failure between $(x_{1,0}, x_{0,0})$ and $(x_{1,n}, x_{0,n})$ because the faulty link can be one of $(x_{1,0}, x_{0,0})$ or $(x_{1,n}, x_{0,n})$ if and only if both L_{b+1} and L_{b+2} are faulty. If both L_{b+1} and L_{b+2} alarm (i.e., report failure), the status of L_1 can be used to determine which of the two links $(x_{1,0}, x_{0,0})$ or $(x_{1,n}, x_{0,n})$ is at fault according to (A3).

For the other links, the statuses of L_{b+1} and L_{b+2} can be used to determine whether the faulty link is in the group of lower links, the group of upper links, or the group of middle links. With (A1), the links in the first two groups are pairwise different, while with (A2) it implies that the codes in the group of middle links are pairwise different. Therefore, all the links in each of the 3 groups are distinguishable such that unambiguous failure localization is possible within each group, and hence in C_n.

4) The number of m-trails in the construction: Note that, the chocolate bar graph has $3n+1$ links, thus

$$b = \lceil \log_2(\frac{|E|-1}{3} + 1) \rceil < \lceil -1.58 + \log_2(|E|+2) \rceil.$$

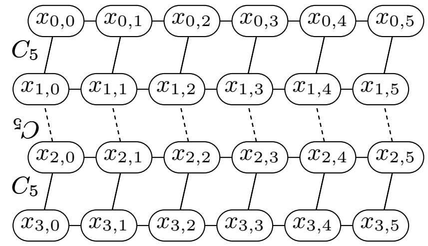
As a result the construction requires at most $J = b + 2 = \lceil 0.42 + \log_2(|E|+2) \rceil$ m-trails. ∎

B. 2-Dimensional Rectangular Grid

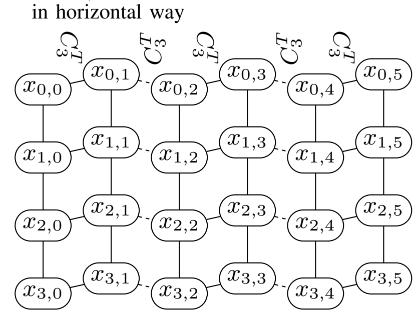
In this section the construction of the chocolate bar graph is generalized for 2-D rectangular grids. The vertices of a $(m+1)$-by-$(n+1)$ grid graph, $S_{m,n}$, are denoted as $x_{i,j}$ for $0 \le i \le m$ and $0 \le j \le n$. The vertical links of $S_{m,n}$ are $(x_{i,j}, x_{i+1,j})$ for $0 \le i < m$ and $0 \le j \le n$. Analogously, the horizontal links of $S_{m,n}$ are $(x_{i,j}, x_{i,j+1})$ for $0 \le i \le m$ and $0 \le j < n$.

Theorem 2: A 2-dimensional rectangular grid graph $S_{m,n}(E,V)$ can be covered with $3 + \lceil \log_2(|E|+1) \rceil$ m-trees to achieve UFL, for $m, n \ge 1$.

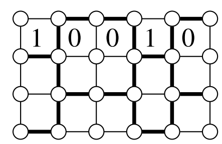
Proof: We shall have two monitoring linksets. The first set has size $b_1 = \lceil \log_2(m+1) \rceil + 2$, while the second has size $b_2 = \lceil \log_2(n+1) \rceil + 2$. Informally speaking, the first of set gives the horizontal position of a failed link, while the other gives the vertical coordinate. This will be sufficient to locate the failed link unambiguously. In total, we shall have $b = b_1 + b_2$ monitoring linksets.



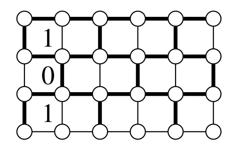
(a) $S_{3,5}$ decomposed into chocolate bars graphs in horizontal way



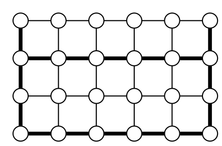
(b) $S_{3,5}$ decomposed into chocolate bars in vertical way



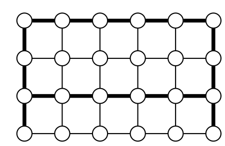
(c) An example of L_i for $i = 1, \dots, b_1 - 2$



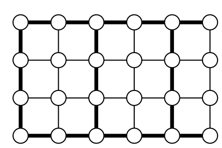
(d) An example of L_i for $i = b_1 + 1, \dots, b - 2$



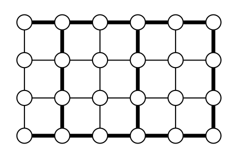
(e) The links of $L_{b_1 - 1}$



(f) The links of L_{b_1}



(g) The links of L_{b-1}



(h) The links of L_b

Fig. 3. An example of 2 dimensional lattice graph of size 3×5.

1) The proposed construction: We are going to extend the monitoring linksets L_i $(i = 1, \dots, b_1)$ from the chocolate bar graph C_n to the whole square grid $S_{m,n}$. We do it step by step: first reflect the path L_i with respect to the line connecting $x_{1,0}$ to $x_{1,n}$. This way the edge set L_i is extended to the chocolate bar defined by the vertices $x_{1,j}$ and $x_{2,j}$, for $j = 0, \dots, n$. From this chocolate bar we extend analogously by reflection to the next one, definied by $x_{2,j}$ and $x_{3,j}$. And so on, by repeating this reflection process we finally extend L_i to the whole $S_{m,n}$. In other words the 2-D rectangular grid is treated as a series of chocolate bars graphs of C_n as it is shown on Fig. 3a, where at every second line the chocolate bar graph is upside down, and the i-th chocolate bar graph C_n consists of vertices $x_{i,0}, \dots, x_{i,n}$ and $x_{i+1,0}, \dots, x_{i+1,n}$, where $i = 0, \dots, m-1$. Suppose that a single (horizontal) link $e = (x_{i,j}, x_{i,j+1})$ is at fault. Then the extended linksets $L_1, \dots, L_{b_1}$ give us the horizontal position j of e. Exactly the same holds for vertically placed links $e = (x_{i,j}, x_{i+1,j})$. The linksets will give us the unique possible j value with the information about the direction (horizontal, vertical) of e.

With the whole situation transposed, exactly the same method can be used to specify the vertical position i of the faulty link e. For the remainig b_2 monitoring linksets of the rectangular grid $S_{m,n}$ we start out with the vertically placed chocolate bar C_m^T at the left end of the grid (see Fig. 3b.) and extend the b_2 monitoring linksets of this C_m^T to the whole grid with the reflection procedure we used before. This time, however, we proceed from left to right to extend the linksets to vertical chocolate bars.

2) Correctness of the constructed m-trail solution: In case of a single failure L_{b_1-1}, L_{b_1}, L_{b-1},and L_b (see also Fig. 3e, 3f, 3g, and 3h) can identify whether a horizontal or a vertical link has failed and if the link is on the frame of the rectangular grid (it is on the first or last row/column). Since the failed link belongs to one of the horizontal chocolate bar graph C_n, the corresponding $b_1 - 2$ linksets can identify the column of the failed link. Similarly, the failed link belongs to one of the vertical chocolate bar graph C_m^T, the corresponding $b_2 - 2$ linksets can identify the row of the failed link. As a result it is known if the link is horizontal or vertical, and its column and row thus it can be localized.

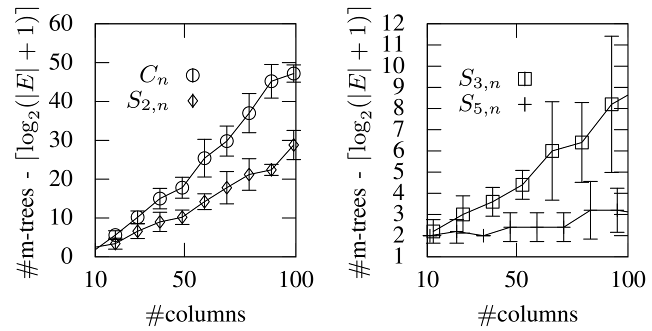
3) The number of m-trails in the construction: Note that linksets L_{b_1-1}, L_{b_1}, L_{b-1},and L_b are only used to decide if the link is horizontal or vertical, and if the link is on the frame of the rectangular grid, which indeed can be done with only two linksets instead of four, namely $L_{b_1-1} \cup L_{b_1}$ and $L_{b-1} \cup L_b$. The graph $S_{m,n}$ has totally $|E| = 2 \cdot m \cdot n + n + m$ links. On the other hand, the number of m-trails is

$$\begin{aligned} J = \lceil \log_2(m+1) \rceil + \lceil \log_2(n+1) \rceil + 2 \leq \\ \lceil 1 + \log_2(m+1) + \log_2(n+1) \rceil + 2 = \\ \lceil \log_2 2 + \log_2(m+1) + \log_2(n+1) \rceil + 2 = \\ \lceil \log_2 2 \cdot (m+1) \cdot (n+1) \rceil + 2 = \\ \lceil \log_2 2mn + 2n + 2m + 2 \rceil + 2 = \\ 2 + \lceil \log_2(2|E| - 2mn + 2) \rceil < \\ 2 + \lceil \log_2(2|E| + 2) \rceil = 3 + \lceil \log_2(|E| + 1) \rceil \end{aligned} \quad (1)$$

for $m, n \geq 1$. Note that the first inequality holds because of the general inequality $\lceil A \rceil + \lceil B \rceil \leq \lceil A + B \rceil + 1$, while the second follows from $m \cdot n > 0$. ■

IV. Conclusions

The paper has conducted a comprehensive study on optimal m-tree allocation in all-optical mesh networks for achieving unambiguous failure localization (UFL) under any single link failure in grid topologies. We have analytical derived essentially tight upper bounds on the number of m-trees via deterministic constructions and rigorous proofs. The new bound obtained is $3 + \lceil \log_2(|E|+1) \rceil$, which is close to the theoretical lower bounds $\lceil \log(|E|+1) \rceil$ as $|E|$ increases. Fig. 4 demonstrates that the derived bound on grid is hard to match by the most state-of-the-art heuristics provided by [1], particularly when a chocolate bar topology is considered.



(a) The performance of GCS [1] with $\gamma = 1000$ on C_n and $S_{2,n}$ (b) The performance of GCS with $\gamma = 1000$ on $S_{3,n}$ and $S_{5,n}$

Fig. 4. Benchmarking experiments on rectangular grid topology with 1,2,3, and 5 number of rows.

References

[1] J. Tapolcai, B. Wu, and P.-H. Ho, "On Monitoring and Failure Localization in Mesh All-Optical Networks," in *IEEE INFOCOM*, 2008.

[2] B. Wu, P. Ho, and K. Yeung, "Monitoring Trail: On Fast Link Failure Localization in WDM Mesh Networks," *IEEE/OSA Journal of Lightwave Technology*, vol. 27, no. 23, Dec. 2009.

[3] C. Mas, I. Tomkos, and O. Tonguz, "Failure Location Algorithm for Transparent Optical Networks," *IEEE Journal on Selected Areas in Communications*, vol. 23, no. 8, pp. 1508–1519, 2005.

[4] H. Zeng, C. Huang, and A. Vukovic, "A Novel Fault Detection and Localization Scheme for Mesh All-optical Networks Based on Monitoring-cycles," *Photonic Network Communications*, vol. 11, no. 3, pp. 277–286, 2006.

[5] B. Wu and K. Yeung, "Monitoring Cycle Design for Fast Link Failure Detection in All-Optical Networks," in *IEEE GLOBECOM '07*, 2007, pp. 2315–2319.

[6] C. Li, R. Ramaswami, I. Center, and Y. Heights, "Automatic fault detection, isolation, and recovery in transparentall-optical networks," *Journal of Lightwave Technology*, vol. 15, no. 10, pp. 1784–1793, 1997.

[7] Y. Wen, V. Chan, and L. Zheng, "Efficient fault-diagnosis algorithms for all-optical WDM networks with probabilistic link failures," *Journal of Lightwave Technology*, vol. 23, pp. 3358–3371, 2005.

[8] B. Wu, P.-H. Ho, and K. Yeung, "Monitoring trail:a new paradigm for fast link failure localization in WDM mesh networks," in *IEEE GLOBECOM '08*, 2008.

[9] M. Maeda, "Management and control of transparent optical networks," *IEEE Journal on Selected Areas in Communications*, vol. 16, no. 7, pp. 1008–1023, 1998.

[10] 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.

[11] S. Ahuja, S. Ramasubramanian, and M. Krunz, "Single link failure detection in all-optical networks using monitoring cycles and paths," *accepted for publication in IEEE/ACM Transactions on Networking*, 2009, http://www.ece.arizona.edu/ srini/Publications.php.

[12] F. K. Hwang and V. T. Sós, "Non-adaptive hypergeometric group testing," *Studia Sci. Math. Hungar*, vol. 22, pp. 257–263, 1987.

[13] D. Du and F. K. Hwang, *Combinatorial Group Testing and Its Applications*. World Scientific, 2000.

[14] R. Lidl and H. Niederreiter, *Introduction to finite fields and their applications*. Cambridge University Press, 1994.