Minimum-Color Path Problems for Reliability in Mesh Networks

Shengli Yuan*, Saket Varma†, and Jason P. Jue‡

*yuans@uhd.edu

Department of Computer and Mathematical Sciences, University of Houston-Downtown

†svarma@student.utdallas.edu, ‡jjue@utdallas.edu

Department of Computer Science, University of Texas at Dallas

***Abstract*—In this work, we consider the problem of maximizing the reliability of connections in mesh networks against failure scenarios in which multiple links may fail simultaneously. We consider the single-path connection problem as well as multiple-path (protected) connection problems. The problems are formulated as minimum-color path problems, where each link is associated with one or more colors, and each color corresponds to a given failure event. Thus, when a certain color fails, all links which include that color will fail. In a single-path problem, by minimizing the number of colors on the path, the failure probability of the path can be minimized if all colors have the same probability of causing failures. In the case of two paths, where one path is a protection path, if all colors have the same probability of causing failures, the problem becomes that of finding two link-disjoint paths which either have a minimum total number of colors, or which have a minimum number of overlapping colors. By minimizing the total number of colors, the probability that a failure will occur on either of the paths is minimized. On the other hand, by minimizing the number of overlapping colors, the probability that a single failure event will cause both paths to fail simultaneously is minimized. The problems are proved to be NP-complete, and ILP formulations are developed. Heuristic algorithms are proposed for larger instances of the problems, and the heuristics are evaluated through simulation.**

***Index terms*— System design, Simulations, Mathematical programming/optimization, Graph theory, Minimum-Color path, path protection, risk, shared risk link group, link disjoint, node disjoint, risk disjoint, integer linear program (ILP)**

I. INTRODUCTION

In path-routed networks, end users communicate with each other via end-to-end paths such as the label switched paths in MPLS networks [1][2] and lightpaths in all-optical WDM networks [3][4]. In such networks, it is important to provide a high degree of reliability or survivability for each connection.

One method for providing survivability is through path protection schemes, in which a link-disjoint backup path is precomputed for every working path [5][6][7]. Such protection schemes provide 100% reliability against any single-link failure in the network. However, in many cases, a single failure event may result in the failure of multiple links in the network. For example, in an optical WDM network, multiple fiber links may be bundled into the same underground conduit or span. Even though these fiber links are disjoint in the network layer, a cut to the underground conduit can cause all fiber links in the conduit to fail. Such fibers that share a common risk factor are said to belong to the same Shared Risk Link Group (SRLG) [8][9].

One approach to address the problem of SRLG protection is to find a risk-disjoint path for every working path [10]. Two paths that are risk disjoint do not share any links which will fail simultaneously due to a single failure event or risk factor. In this case, the network can provide 100% reliability against any single failure event in the network, even if such an event causes multiple links to fail simultaneously.

A drawback of SRLG protection schemes is that, if there are many risk factors in the network, and if risk factors include many links, then it may be difficult, or even impossible, to find two risk-disjoint paths for every connection [11]. Thus, it may be difficult to provide 100% reliability against certain multiple-link failure events. For cases in which 100% reliability is not possible, the objective should be to find one or more paths for each connection, such that the reliability for each connection is maximized. In this work, we are concerned with maximizing the reliability of connections against failure events in which multiple links may fail simultaneously. Equivalently, the problem is to find one or more paths for each connection such that the failure probability of the connection is minimized.

In this work, we consider the single-path problem as well as multiple-path problems. The problems are formulated as minimum-color path problems, where each link is associated with one or more colors, and each color corresponds to a given failure event. Thus, when a certain color fails, all links which include that color will fail. In a single-path problem, by minimizing the number of colors on the path, the failure probability of the path can be minimized if all risk factors have the same probability of causing failures. In the case of two-paths, where one path is a protection path, if all risk factors have the same probability of causing failures, the problem becomes that of finding two link-disjoint paths which either have a minimum total number of colors, or which have a minimum number of overlapping colors. By minimizing the total number of colors, the probability that a failure will occur on either of the paths is minimized. On the other hand, by minimizing the number of overlapping colors, the probability that a single failure event will cause both paths to fail simultaneously is minimized.

Compared to shortest path problems and their applications [12][13][14], there has not been much research activity on minimum-color path problems, despite their significant practical value. While investigating the blue-red set covering problem, [15] briefly discussed the minimum-color single-path problem and stated that the problem can be proven NP-complete by reduction from the blue-red set covering problem. There was no further discussion on the problem beyond that.

0-7803-8968-9/05/$20.00 (C)2005 IEEE

On the other hand, [16] looked into the problem of establishing a spanning tree using the minimum number of labels (i.e., colors). It proved the problem to be NP-complete, then proposed two heuristic algorithms. The first heuristic is called the Edge Replacement Algorithm. The algorithm first forms an arbitrary spanning tree, then tries to replace each edge with a different edge that can reduce the total number of colors in the tree. The second heuristic is called the Maximum Vertex Covering Algorithm. This algorithm starts with an empty spanning tree, then scans through all the colors and chooses the one that covers the most uncovered vertices. This procedure is repeated until a spanning tree is formed. [17] and [18] investigated the second heuristic algorithm in further detail.

In this paper, we define three minimum-color path problems. The Minimum-Color Single-Path (MCSiP) problem is the problem of finding a single path from a source node s to a destination node d such that it uses the minimum number of colors. We investigate this problem in much greater depth than previous researchers. We then investigate the problem of finding two link-disjoint paths that use the minimum number of total colors, i.e., the Minimum Total Color Disjoint-Paths (MTCDiP) problem. This problem has two variations based on the disjointness of the two paths. One variation is the case where the two paths are to be node-disjoint, while the other is the case where the two paths are to be link-disjoint. We consider both variations in this work. The third problem discussed in this paper is the Minimum Overlapping Color Disjoint-Paths (MOCDiP) problem. The objective of this problem is to find two link-disjoint paths that have the minimum number of common colors. For all three problems, we prove they are NP-complete. We also develop ILP formulations and heuristics for each of the problems.

In the general case of the minimum-color path problems, every network link may be of multiple colors. For the three problems we discuss in this paper, a link of n colors and c cost is equivalent to n concatenated links, each having only one of the n colors and $1/c$ cost. Therefore, we consider only network links of a single color in this paper.

In addition to the discussion of the three problems with a single connection, we also consider the problems with static traffic, in which all connections requests are known. The objective is to find paths with minimum average number of colors for all of the connection requests.

In addition to network reliability, the minimum-color path problems also apply to other areas of network management. For instance, a complex network may consist of links that deploy different transmission mediums, such as WDM, SONET/SDH [19], Frame Relay [20], and X.25 [21]. It is quite likely that the shortest paths between two end nodes traverse links of many different mediums, which are expensive to provision and maintain. Therefore a network carrier may prefer different paths that are suboptimal in length but have the minimum number of transmission mediums. As another example, a nationwide communication network may contain nodes and links that belong to different network carriers. A path is often less expensive to establish and operate if it involves fewer carriers, even if the length of that path is not minimal. If we represent the transmission mediums and the network operators by colors, these two examples can be treated as minimum-color path problems.

To the best of our knowledge, this is the first time the MTCDiP problem and MOCDiP problem are defined and investigated. This is also the first time an ILP formulation and heuristics are proposed for the MCSiP problem.

The rest of the paper is organized as follows. Section II, III, and IV discuss each of the above three problems, respectively. In each section, we present a proof for the NP-completeness of the problem, followed by an ILP formulation and heuristics. We solve the ILPs and conduct computer simulations for the heuristics in Section V. Section VI concludes the paper.

II. The Minimum-color Single-Path Problem

The MCSiP problem is defined as follows. Given network $G = (N, L)$, where N is the set of nodes and L is the set of links, and given the set of colors $C = \{c_1, c_2, c_3, \ldots, c_K\}$ where K is the maximum number of colors in G, and given the color $c_l \in C$ for every link $l \in L$, find one path from source node s to destination node d such that it uses the minimum number of colors.

A. Proof of NP-Completeness

We need to reduce a known NP-complete problem to the MCSiP problem. The known NP-complete problem in this case is the Minimum Set Covering Problem [22]. This problem is stated as follows. Given a finite set $S = \{a_1, a_2, a_3, \ldots, a_n\}$, and a collection $C = \{C_1, C_2, \ldots, C_m\}$ such that each element in C contains a subset of S, is there a minimum subset, $C' \subseteq C$ such that every member of S belong to at least one member of C'?

We construct a graph G for an arbitrary instance of the Minimum Set Covering Problem, such that the graph contains one path from s to d with the minimum number of colors, if and only if C contains a minimum set cover C'. Following are the steps for the graph construction:

Step 1. For every element a_i in S, create a network node a_i.

Step 2. For every subset C_j to which a_i belongs, create a network link $a_{i-1}\, a_i$ of color c_j. For element a_1, the link is sa_1. There is also a single link between a_n and d with color c_0.

An example is given in Fig 1. In this example, we construct graph G for a Minimum Set Covering problem $S = \{a_1, a_2, a_3, a_4\}$, $C = \{C_1, C_2, C_3, C_4, C_5\}$, $C_1 = \{a_1, a_2\}$, $C_2 = \{a_2, a_3\}$, $C_3 = \{a_1, a_3\}$, $C_4 = \{a_3, a_4\}$, $C_5 = \{a_1, a_4\}$.

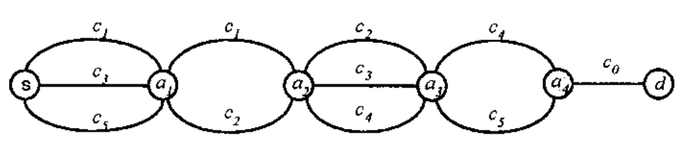

Fig. 1. Reduction of the Minimum Set Covering Problem to the MCSiP problem

It is quite clear that if there is a minimum-color path from s to d, then the colors on that path are mapped directly to a minimum set covering all the elements in S. Conversely, if there is a minimum set covering all the elements, then a minimum-color path can be derived by going through every node and selecting the link with the color representing the set that covers the corresponding element.

Under static traffic, the MCSiP problem is defined as follows. Given network $G = (N, L)$, where N is the set of nodes and L is the set of links, and given the colors $C = \{c_1, c_2, c_3, \dots, c_K\}$ where K is the maximum number of colors in G, and given the color $c_l \in C$ for every link $l \in L$, and given the connection requests $\Delta = \{s_1d_1, s_2d_2, \dots, s_md_m \dots\}$, where source node $s_m \in N$ and destination node $d_m \in N$, find the paths between every source-destination pairs in Δ such that the average number of colors on every path is minimal.

The MCSiP problem under static traffic is NP-hard since it contains the special case of a single connection request.

B. ILP Formulation

We now develop an ILP formulation for the MCSiP problem under static traffic. The problem with a single connection is a special case when there is only one source-destination pair in Δ. The following are given as inputs to the problem.

- N: number of nodes in the network.
- L: number of links in the network.
- $color_c^{ij}$: 1 if link ij is of color c; 0 otherwise.
- $\Delta = \{s_1d_1, s_2d_2, \dots, s_md_m, \dots s_Md_M\}$, $M \geq m \geq 1$: All the source-destination pairs of the connection requests.

The ILP solves for the following variables.

- α_m^{ij}: 1 if link ij is used on the path between source-destination pair s_md_m in Δ; 0 otherwise.
- δ_m^c : 1 if link color c is on the path between source-destination pair s_md_m in Δ; 0 otherwise.

Objective:

$$\text{Minimize } \left(\sum_m \sum_c \delta_m^c\right)/M \quad (1)$$

Constraints:

$$\sum_j \alpha_m^{xj} = 1, \text{ where } x = s_m, \forall m \quad (2)$$

$$\sum_i \alpha_m^{ik} - \sum_j \alpha_m^{kj} = 0, \ \forall k \neq s_m, d_m, \forall m \quad (3)$$

$$\sum_j \alpha_m^{jy} = 1, \text{ where } y = d_m, \forall m \quad (4)$$

$$\delta_m^c \leq \sum_{i,j} (color_c^{ij} \cdot \alpha_m^{ij}), \ \forall m \quad (5)$$

$$L \cdot \delta_m^c \geq \sum_{i,j} (color_c^{ij} \cdot \alpha_m^{ij}), \ \forall m \quad (6)$$

The objective is to minimize the average number of colors used. Eqs. (2)-(4) describe the flow constraints for each source destination pair. Eqs. (5)-(6) specify the color constraints that set the value of δ_m^c to 1 if the path for the source-destination pair contains the color c.

C. Heuristic Algorithms

We give two heuristics to solve the MCSiP problem. The first heuristic is called Single-Path Color Reduction Algorithm. In this algorithm, we first run Dijkstra's algorithm or the Bellman-Ford algorithm to find the shortest path from the source s to the destination d. We then try to eliminate some of the colors while still being able to find a path from s to d. The details are as follows.

Step 1.Run a shortest path algorithm and find a path p. Assume that the collection of all the colors on p is set $C_p = \{c_1, c_2, \dots, c_k\}$.

Step 2. Go through every color in C_p *. Select the color such that, after the links of that color are removed from the network, we run the shortest path algorithm and obtain a shortest path with the minimum number of colors which is also less than* $|C_p|$*. Remove the links of the selected color.*

Step 3. Repeat Step 1 and 2 until the number of colors on the shortest path cannot be further reduced.

The running time is $O(m^2 n \log n)$ where n is the number of nodes and m is the total number of colors in the network.

The next heuristic is called the Single-Path All Color Optimization Algorithm. In this algorithm, we go through all the colors and try to use only a subset of them on paths from s to d. The details are as follows.

Step 1. Run a shortest path algorithm and find a path p. Assume the number of colors on p is $|C_p|$.

Step 2. Set the link cost to zero on the links of one color, and find the shortest path. Repeat for all the colors in the network and select the one that results in a path with the minimum number of colors which is also less than $|C_p|$*. Keep the costs to zero on the links of the selected color.*

Step 3. Repeat Step 1 and 2 until the number of colors on the shortest paths cannot be further reduced.

The running time is $O(m^2 n \log n)$ where n is the number of nodes and m is the total number of colors in the network.

For the case with static traffic, we can run the heuristics sequentially on each of the connection requests. If the network links have limited capacity, we may first sort the connection requests based on the length of the shortest paths between all the source-destination pairs, then apply the heuristics on the requests staring with the ones that have the longest shortest path between the source and the destination. This is because these paths are most likely to be blocked if we route them later.

III. The Minimum Total Color Disjoint-Paths Problem

The MTCDiP problem is defined as follows. Given network $G = (N, L)$, where N is the set of nodes and L is the set of links, and given the set of colors $C = \{c_1, c_2, c_3, ..., c_K\}$ where K is the maximum number of colors in G, and given the color $c_l \in C$ for every link $l \in L$, find two disjoint paths from source node s to destination node d such that the total number of colors on the two paths are minimal.

A. Proof of NP-Completeness

There are two variations of the MTCDiP problem based on the requirement on the path disjointness. In the first variation, the two paths are to be node disjoint. In the second variation, the two paths are to be link disjoint. We reduce the Minimum Set Covering problem to both variations of the problem to prove their NP-completeness. For the Minimum Set Covering problem, assume the given finite set S is $\{a_1, a_2, a_3, ..., a_n\}$ and the collection C is $\{C_1, C_2, ..., C_m\}$.

A.1. MTCDiP Problem with Node-Disjoint Requirement

We construct a graph G for an arbitrary instance of the Minimum Set Covering Problem, such that the graph contains two node-disjoint paths from s to d with the minimum number of colors, if and only if C contains a minimum set cover C'. Following are the steps for the graph construction:

Step 1. For every element a_i in S, create a network node a_i.

Step 2. For every subset C_j to which a_i belongs, create a network link a_{i-2} a_i of color c_j. For elements a_1 and a_2, the links are sa_1 and sa_2 respectively. There are also single links $a_{n-1}d$ and a_nd with color c_0.

An example is given in Fig 2. In this example, we construct graph G for a Minimum Set Covering problem $S = \{a_1, a_2, a_3, a_4\}$, $C = \{C_1, C_2, C_3, C_4, C_5\}$, $C_1 = \{a_1, a_2\}$, $C_2 = \{a_2, a_3\}$, $C_3 = \{a_1, a_3\}$, $C_4 = \{a_3, a_4\}$, $C_5 = \{a_1, a_4\}$.

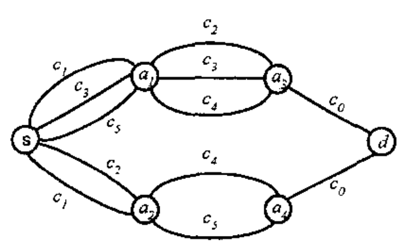

Fig. 2. Reduction of the Minimum Set Covering Problem to the MTCDiP problem with node-disjoint requirement

In the constructed graph G, $a_{n-1}d$ and a_nd are single links. Therefore any two node-disjoint paths from s to d must have one path going along s-a_1-a_3-...-d and the other path going along s-a_2-a_4-...-d. If two node-disjoint paths p_1 and p_2 have the minimum number of colors, since each color except for c_0 is associated with a member in C, the collections of all the colors on p_1 and p_2 map to a minimum subset of C that covers all the elements. Conversely, if there is a minimum subset $C' \subseteq C$ that covers all the elements, then for each node a_i in G, there is at least one member C_j in C' that contains a_i, and we choose a link $a_{i-2}a_i$ (or sa_1, sa_2) of the color c_j. All the links, together with the single links with color c_0, form two node-disjoint paths from s to d that have the minimum number of colors.

A.2. MTCDiP Problem with Link-Disjoint Requirement

We construct a graph G for an arbitrary instance of the Minimum Set Covering Problem, such that the graph contains two link-disjoint paths from s to d with the minimum number of colors, if and only if C contains a minimum set cover C'. Following are the steps for the graph construction:

Step 1. For every element a_i in S, create network nodes a_i and u_i.

Step 2. For every element a_{2i} in S, (except for a_n if n is even, and a_{n-1} if n is odd), create a network node v_i.

Step 3. For every subset C_j to which a_i belongs, create a network link u_ia_i of color c_j.

Step 4. Create a single link su_1, su_2. If n is even, create single link a_1v_1, a_2v_1, v_1u_3, v_1u_4, ..., $a_{2i-1}v_i$, $a_{2i}v_i$, v_iu_{2i+1}, v_iu_{2i+2}, ..., $a_{n-3}v_{n/2-1}$, $a_{n-2}v_{n/2-1}$, $v_{n/2-1}u_{n-1}$, $v_{n/2-1}u_n$, $a_{n-1}d$, a_nd. If n is odd, create single link a_1v_1, a_2v_1, v_1u_3, v_1u_4, ..., $a_{2i-1}v_i$, $a_{2i}v_i$, v_iu_{2i+1}, v_iu_{2i+2}, ..., $a_{n-2}v_{(n-1)/2}$, $a_{n-1}v_{(n-1)/2}$, $v_{(n-1)/2}u_n$, $v_{(n-1)/2}d$, a_nd. All of the links are of color c_0.

An example is given in Fig 3. In this example, we construct graph G for a Minimum Set Covering problem that has an even number of elements in S, *i.e.*, $S = \{a_1, a_2, a_3, a_4\}$, $C = \{C_1, C_2, C_3, C_4, C_5\}$, $C_1 = \{a_1, a_2\}$, $C_2 = \{a_2, a_3\}$, $C_3 = \{a_1, a_3\}$, $C_4 = \{a_3, a_4\}$, $C_5 = \{a_1, a_4\}$.

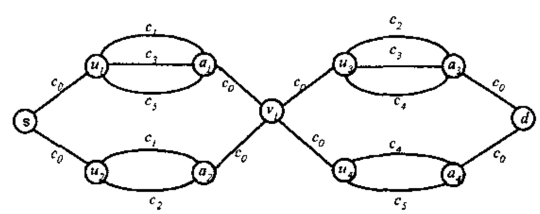

Fig. 3. Reduction of the Minimum Set Covering Problem with an even number of elements to the MTCDiP problem with link-disjoint requirement

Another example is given in Fig 4. In this example, S has an odd number of elements, *i.e.*, $S = \{a_1, a_2, a_3, a_4, a_5\}$, $C = \{C_1, C_2, C_3, C_4, C_5\}$, $C_1 = \{a_1, a_2, a_5\}$, $C_2 = \{a_2, a_3\}$, $C_3 = \{a_1, a_3, a_5\}$, $C_4 = \{a_3, a_4, a_5\}$, $C_5 = \{a_1, a_4\}$.

If there are two link-disjoint paths p_1 and p_2 in the constructed graph G, every network node a_i must be on exactly one of the paths. If the two paths have the minimum number of colors, since each color except for c_0 is associated with a member in C, the collections of all the colors on p_1 and p_2 map to a minimum subset of C that covers all the elements. Conversely, if there is a minimum subset $C' \subseteq C$ that covers all the elements, then for each node a_i in G, there is at least one member C_j in C' that contains a_i, and we choose a link

$u_j a_i$ of the color c_j. All the links, together with the single links with color c_0, form two link-disjoint paths from s to d that have the minimum number of colors.

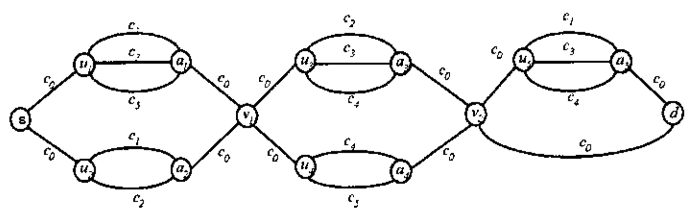

Fig. 4 Reduction of the Minimum Set Covering Problem with an odd number of elements to the MTCDiP problem with link-disjoint requirement

For the purpose of path protection, finding link-disjoint paths is more important than finding node-disjoint paths because modern switching node devices normally have built-in redundancy so that node failure is much less a concern than link failure. Next we develop ILP formulation and heuristics for the MTCDiP problem with the link disjointness requirement.

Under static traffic, the MTCDiP problem is defined as follows. Given network $G = (N, L)$, where N is the set of nodes and L is the set of links, and given the colors $C = \{c_1, c_2, c_3, \ldots, c_K\}$ where K is the maximum number of colors in G, and given the color $c_l \in C$ for every link $l \in L$, and given the connection requests $\Delta = \{s_1d_1, s_2d_2, \ldots, s_md_m \ldots\}$, where source node $s_m \in N$ and destination node $d_m \in N$, find the disjoint-paths between every source-destination pairs in Δ such that the average total number of colors on the disjoint-paths pair is minimal.

The MTCDiP problem under static traffic is NP-hard since it contains the special case of a single connection request.

B. ILP Formulation

We now develop an ILP formulation for the MTCDiP problem with the link disjointness requirement. The problem with single connection is a special case when there is only one source-destination pair in Δ. The following are given as inputs to the problem.

- N: number of nodes in the network.
- L: number of links in the network.
- $color_c^{ij}$: 1 if link ij is of color c; 0 otherwise.
- $\Delta = \{s_1d_1, s_2d_2, \ldots, s_md_m, \ldots s_Md_M\}$, $M \geq m \geq 1$: All the source-destination pairs of the connection requests.

The ILP solves for the following variables.

- α_m^{ij}: 1 if link ij is used on path p_1 between source-destination pair s_md_m in Δ; 0 otherwise.
- β_m^{ij} : 1 if link ij is used on path p_2 between source-destination pair s_md_m in Δ; 0 otherwise.
- δ_{1m}^c : 1 if link color c is on path p_1 between source-destination pair s_md_m in Δ; 0 otherwise.
- δ_{2m}^c : 1 if link color c is on path p_2 between source-destination pair s_md_m in Δ; 0 otherwise.
- $overlap_m^c$: 1 if link color c is on both paths p_1 and p_2 between source-destination pair s_md_m in Δ; 0 otherwise.

Objective:

$$\text{Minimize } \left(\sum_m \sum_c (\delta_{1m}^c + \delta_{2m}^c - overlap_m^c)\right)/M \quad (7)$$

Constraints:

$$\sum_j \alpha_m^{xj} = 1, \text{ where } x = s_m, \forall m \quad (8)$$

$$\sum_i \alpha_m^{ik} - \sum_j \alpha_m^{kj} = 0, \ \forall k \neq s_m, d_m, \forall m \quad (9)$$

$$\sum_j \alpha_m^{jy} = 1, \text{ where } y = d_m, \forall m \quad (10)$$

$$\sum_j \beta_m^{xj} = 1, \text{ where } x = s_m, \forall m \quad (11)$$

$$\sum_i \beta_m^{ik} - \sum_j \beta_m^{kj} = 0, \ \forall k \neq s_m, d_m, \forall m \quad (12)$$

$$\sum_j \beta_m^{jy} = 1, \text{ where } y = d_m, \forall m \quad (13)$$

$$\delta_{1m}^c \leq \sum_{i,j} (color_c^{ij} \cdot \alpha_m^{ij}), \ \forall m \quad (14)$$

$$L \cdot \delta_{1m}^c \geq \sum_{i,j} (color_c^{ij} \cdot \alpha_m^{ij}), \ \forall m \quad (15)$$

$$\delta_{2m}^c \leq \sum_{i,j} (color_c^{ij} \cdot \beta_m^{ij}), \ \forall m \quad (16)$$

$$L \cdot \delta_{2m}^c \geq \sum_{i,j} (color_c^{ij} \cdot \beta_m^{ij}), \ \forall m \quad (17)$$

$$0 \leq 1 - \delta_{1m}^c + 1 - \delta_{2m}^c + 2 \cdot overlap_m^c \leq 2, \ \forall m \quad (18)$$

$$\alpha_m^{ij} + \beta_m^{ij} \leq 1, \quad \forall i,j, \ \forall m \quad (19)$$

The objective is to minimize the average total number of colors used over the two link-disjoint paths for each source-destination pair. Eqs. (8)-(10) describe the flow constraints for one path and Eqs. (11)-(13) describe the same for the second path. Eqs. (14)-(17) describe the color constraints that determine the color used on the two paths. Eq. (18) is the overlapping colors constraint that determines the set of overlapping colors for a source destination pair. Eq. (19) is the link-disjoint constraint for the two paths.

C. Heuristic Algorithms

We give two heuristics to solve the MTCDiP problem with the link disjointness requirement. The first heuristic is called the Disjoint-Paths Color Reduction Algorithm. In this algorithm, we first run Suurballe's algorithm to find two link-disjoint paths with the minimal total cost from the source node s to the destination node d. We then try to eliminate some of the colors while still able to find two link-disjoint paths from s to d. The details are as follows.

Step 1. Run Suurballe's algorithm and find two link-disjoint paths p_1 and p_2. Assume the collection of all the colors on p_1 and p_2 is set $C_p = \{c_1, c_2, \ldots, c_k\}$.

Step 2. Go through every color in C_p. Select the color such that, after the links of that color are removed from the network, we run Suurballe's algorithm again and obtain two link-disjoint paths with the minimum number of total colors which is also less than $|C_p|$. Remove the links of the selected color.

Step 3. Repeat Step 1 and 2 until the number of colors on the link-disjoint paths cannot be further reduced.

The running time is $O(m^2n^2\log n)$ where n is the number of nodes and m is the total number of colors in the network.

The next heuristic is called the Disjoint-Paths All Color Optimization Algorithm. In this algorithm, we go through all colors and try to use only a subset of them on the disjoint paths from s to d. The details are as follows.

Step 1. Run Suurballe's algorithm and find two link-disjoint paths. Assume the total number of colors on the two paths is $|C_p|$.

Step 2. Set the link cost to zero on the links of one color, then run Suurballe's algorithm and find two link-disjoint paths. Repeat for all the colors in the network and select the one that results in two link-disjoint paths with the minimum number of colors which is also less than $|C_p|$. Keep the costs to zero on the links of the selected color.

Step 3. Repeat Step 1 and 2 until the number of colors on the link-disjoint paths cannot be further reduced.

The running time is $O(m^2\ n^2\log n)$ where n is the number of nodes and m is the total number of colors in the network.

For the case with static traffic, we can run the heuristics sequentially on each of the connection requests. If the network links have limited capacity, we may first sort the connection requests based on the length of the shortest paths between all the source-destination pairs, then apply the heuristics on the requests staring with the ones that have the longest shortest path between the source and the destination. This is because these paths are most likely to be blocked if we route them later.

IV. The Minimum Overlapping Color Disjoint-Paths Problem

The MOCDiP problem is defined as follows. Given network $G = (N, L)$, where N is the set of nodes and L is the set of links, and given the colors $C = \{c_1, c_2, c_3, \ldots, c_K\}$ where K is the maximum number of colors in G, and given the color $c_l \in C$ for every link $l \in L$, find two link-disjoint paths from source node s to destination node d such that they share the minimum number of common colors.

A. Proof of NP-Completeness

We reduce a known NP-complete problem to the MOCDiP problem. The known NP-complete problem in this case is the problem of finding two link-disjoint paths from source node s to destination node d that are completely SRLG-disjoint [11][23]. We replace every SRLG with a different color. If we were able to solve the MOCDiP problem and find two link-disjoint paths from s to d with the minimum overlapping colors, then the paths should also be color-disjoint (i.e., SRLG-disjoint) if such color-disjoint paths exist in the network.

Under static traffic, the MOCDiP problem is defined as follows. Given network $G = (N, L)$, where N is the set of nodes and L is the set of links, and given the colors $C = \{c_1, c_2, c_3, \ldots, c_K\}$ where K is the maximum number of colors in G, and given the color $c_l \in C$ for every link $l \in L$, and given the connection requests $\Delta = \{s_1d_1, s_2d_2, \ldots, s_md_m \ldots\}$, where source node $s_m \in N$ and destination node $d_m \in N$, find the disjoint-paths between every source-destination pairs in Δ such that the average number of colors shared by each disjoint-paths pair is minimal.

The MOCDiP problem under static traffic is NP-hard since it contains the special case of a single connection request.

B. ILP Formulation

We now develop an ILP formulation for the MOCDiP problem. The problem with single connection is a special case when there is only one source-destination pair in Δ. The following are given as inputs to the problem.

- N: number of nodes in the network.
- L: number of links in the network.
- $color_c^{ij}$: 1 if link ij is of color c; 0 otherwise.
- $\Delta = \{s_1d_1, s_2d_2, \ldots, s_md_m, \ldots s_Md_M\}$, $M \geq m \geq 1$: All the source-destination pairs of the connection requests.

The ILP solves for the following variables.

- α_m^{ij}: 1 if link ij is used on path p_1 between source-destination pair s_md_m in Δ; 0 otherwise.
- β_m^{ij}: 1 if link ij is used on path p_2 between source-destination pair s_md_m in Δ; 0 otherwise.
- δ_{1m}^c: 1 if link color c is on path p_1 between source-destination pair s_md_m in Δ; 0 otherwise.
- δ_{2m}^c : 1 if link color c is on path p_2 between source-destination pair s_md_m in Δ; 0 otherwise.
- $overlap_m^c$: 1 if link color c is on both p_1 and p_2 between source-destination pair s_md_m in Δ; 0 otherwise.

Objective:

$$\text{Minimize } \left(\sum_m \sum_c overlap_m^c\right)/M \tag{20}$$

Constraints:

$$0 < 1 - \delta^c_{1m} + 1 - \delta^c_{2m} + 2 \cdot overlap^c_m \leq 2, \ \forall m \quad (21)$$

$$\sum_j \alpha^{xj}_m = 1, \text{ where } x = s_m, \forall m \quad (22)$$

$$\sum_i \alpha^{ik}_m - \sum_j \alpha^{kj}_m = 0, \ \forall k \neq s_m, d_m, \forall m \quad (23)$$

$$\sum_j \alpha^{jy}_m = 1, \text{ where } y = d_m, \forall m \quad (24)$$

$$\sum_j \beta^{xj}_m = 1, \text{ where } x = s_m, \forall m \quad (25)$$

$$\sum_i \beta^{ik}_m - \sum_j \beta^{kj}_m = 1, \ \forall k \neq s_m, d_m, \forall m \quad (26)$$

$$\sum_j \beta^{jy}_m = 1, \text{ where } y = d_m, \forall m \quad (27)$$

$$\delta^c_{1m} \leq \sum_{i,j} (color^{ij}_c \cdot \alpha^{ij}_m) \quad (28)$$

$$L \cdot \delta^c_{1m} \geq \sum_{i,j} (color^{ij}_c \cdot \alpha^{ij}_m) \quad (29)$$

$$\delta^c_{2m} \leq \sum_{i,j} (color^{ij}_c \cdot \beta^{ij}_m), \ \forall m \quad (30)$$

$$L \cdot \delta^c_{2m} \geq \sum_{i,j} (color^{ij}_c \cdot \beta^{ij}_m), \ \forall m \quad (31)$$

$$\alpha^{ij}_m + \beta^{ij}_m \leq 1, \ \forall i, j, \forall m \quad (32)$$

The objective is to reduce the average number of overlapping colors on the two disjoint paths. Eq. (21) is the overlapping colors constraint that determines the set of overlapping colors for a source destination pair. Eqs. (22)-(27) are the flow constraints for the two disjoint paths. Eqs. (28)-(31) are the color constraints that determine the set of colors on the two paths. Eq. (32) is the link disjoint constraint.

C. Heuristic Algorithms

We give three heuristic algorithms for solving the MOCDiP problem. The first heuristic is the Simple Two-Step Algorithm. This algorithm is very simple and we use it to establish an upper bound on the number of overlapping colors for the next two heuristics which we discuss in this section. Details of the Simple Two-Step Algorithm are as follows:

Step 1. Run Dijkstra's algorithm and find shortest path p_1.

Step 2. Increase the cost of a link if the color of the link is on p_1. *The additional cost is proportional to the number of links of that color on* p_1.

Step 3. Remove all links on p_1. *Run Dijkstra's algorithm again and find the second shortest path* p_2.

p_1 and p_2 are link-disjoint. Since the costs on the links of those colors on p_1 are increased, it is expected that p_1 and p_2 share fewer colors. The running time of this heuristic is the same as that of Dijkstra's algorithm, which is $O(n\log n)$.

The second heuristic is called Minimum-Color First-Path Algorithm. It differs from the Simple Two-Step Algorithm in that, at the first step, rather than finding the shortest path, this algorithm uses one of the heuristics in Section II.C to find one path from s to d with the minimum number of colors. The details are as follows:

Step 1. Run Single-Path All Color Optimization Algorithm in Section II and find the first path p_1.

Step 2. Increase the cost of a link if the color of the link is on p_1. *The additional cost is proportional to the number of links of that color on* p_1.

Step 3. Remove all links on p_1. *Run Dijkstra's algorithm and find path* p_2.

The running time is $O(m^2 n\log n)$ where n is the number of nodes and m is the total number of colors in the network.

Both the Simple Two-Step Algorithm and the Minimum-Color First-Path Algorithm follow a two-step approach. They are straightforward and work on networks that don't contain the "trap" topology [25]. For networks with the "trap" topology, a two-step approach may fail to find two link-disjoint paths even if such paths exist in the networks. On the other hand, Suurballe's algorithm always finds two link-disjoint paths as long as they exist. The third heuristic for the MOCDiP problem is called the Joint-Search Minimum Overlapping Color Algorithm. It utilizes Suurballe's algorithm. The details are as follows:

Step 1. Run Suurballe's algorithm and find two link-disjoint paths p_1 *and* p_2.

Step 2. Increase the cost of a link if the color of the link is on p_1. *The additional cost is proportional to the number of links of that color on* p_1.

Step 3. Remove all links on p_1. *Run Dijkstra's algorithm and find path* p_1'.

Step 4. Increase the cost of a link if the color of the link is on p_2. *The additional cost is proportional to the number of links of that color on* p_2.

Step 5. Remove all links on p_2. *Run Dijkstra's algorithm and find path* p_2'.

Step 6. Compare the total colors of paths p_1 *and* p_1' *with the total colors of paths* p_2 *and* p_2'. *The paths with fewer total colors are returned as outputs.*

The running time is $O(n^2\log n)$ where n is the number of nodes in the network.

For the case with static traffic, we can run the heuristics sequentially on each of the connection requests. If the network links have limited capacity, we may first sort the connection requests based on the length of the shortest paths between all the source-destination pairs, then apply the heuristics on the requests staring with the ones that have the longest shortest path between the source and the destination. This is because these paths are most likely to be blocked if we route them later.

V. Simulations

We have developed ILP formulations and heuristics for three minimum-color path problems. We now solve the ILPs using CPLEX [26] and compare the results with those of the heuristics on networks that are randomly generated using LEDA [27]. The network size ranges from 10 nodes to 40 nodes. The nodal degree ranges from 2.6 to 3.0. The color intensity ranges from 1 to 20. The network color intensity is defined as the average number of links of the same color. When the color intensity is 1, every network link has a different color. The higher the color intensity, the fewer colors a network has. If the color intensity is equal to the number of links, then all links have the same color. We assume that each network link has the same probability of being any given colors.

A. Minimum-Color Single Path Problem

We have developed an ILP formulation and two heuristics for the MCSiP problem, i.e., the Single-Path Color Reduction Algorithm (SPCRA) and the Single-Path All Color Optimization Algorithm (SPACOA). We run computer simulations for both the individual connection request problem and the static traffic problem.

A.1. Individual Connection Request Problem

For every source-destination pair in the networks, we solve the ILP and use the heuristics to obtain the minimum-color path. We then compare the average numbers of colors on all the paths. To establish an upper bound for the results of the heuristics, we also run Dijkstra's shortest path algorithm on all the source-destination pairs. Two sets of the results are depicted in Fig. 5 and Fig. 6. Results on other network topologies are similar.

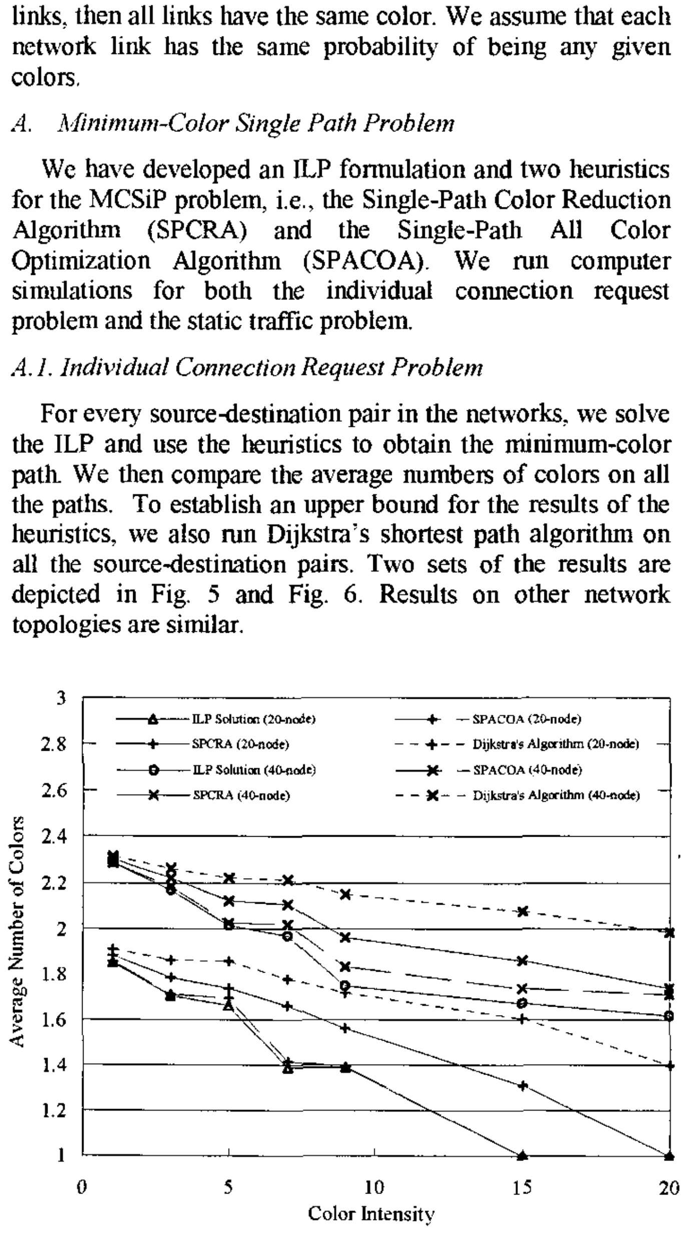

Fig 5. Average number of colors on paths of all source-destination pairs vs. network color intensity. Network nodal degree = 2.6

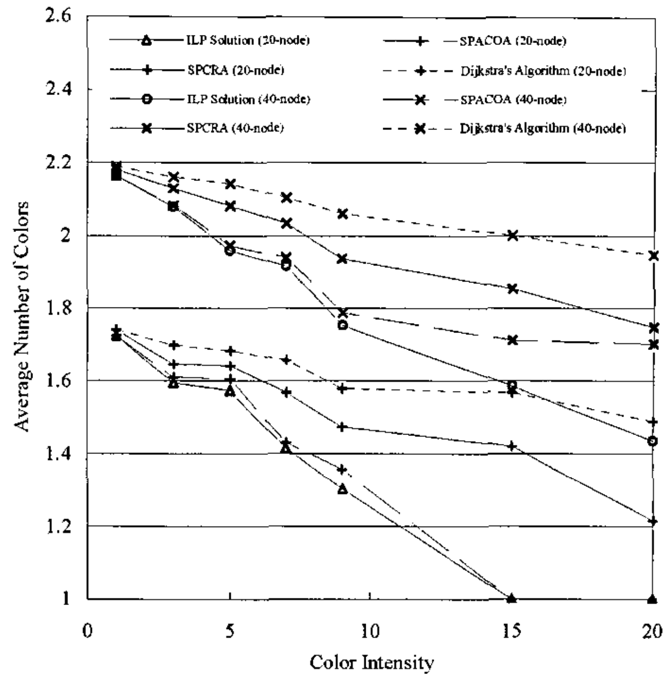

Fig 6. Average number of colors on paths of all source-destination pairs vs. network color intensity. Network nodal degree = 3.0

Based on the simulation results, the paths obtained from the Single-Path All Color Optimization Algorithm are closest to the optimal ILP solutions in the average numbers of colors on the paths. This is because SPACOA selects optimal colors from all colors in the network while SPCRA is restricted to the colors on the initial shortest paths.

We note that, as the nodal degree increases, the number of colors on the path reduces. The reason for this behavior is that an increase in nodal degree results in a wider choice of available routes for each connection request. Furthermore, an increase in nodal degree reduces the average hop distance for each connection, thereby reducing the number of colors. We also note that, color intensity has an impact on the number of colors on the paths as well. When the color intensity is 1, all links in the network have a distinct color; hence, the number of colors for a given path is simply the hop distance of that path, and the average number of colors for each path is simply the average hop distance in the network. As the color intensity increases, the total number of links with the same colors increases. As a result, the number of colors on the path decreases with an increase in color intensity. The network topology and the size of the network also affect the number of colors on the path. Larger networks with more nodes result in a higher average hop count for paths; hence, for the same nodal degree and color intensity, paths in a network with more nodes will have a greater number of colors than paths in a network with fewer nodes.

A.2. Static Traffic Problem

The static connection requests include connections between all the source-destination pairs in the networks. We solve the

ILP for all the requests to obtain the paths with the minimum average number of colors. We then apply the heuristics on the same set of requests. To establish an upper bound for the results of the heuristics, we also run Dijkstra's shortest path algorithm. We use a randomly generated 10-node network of nodal degree 2.6 for the simulations. The capacity of every link is assumed to be infinite. The results are depicted in Fig. 7. The simulation results confirm that the paths obtained from the Single-Path All Color Optimization Algorithm are closest to the optimal ILP solutions

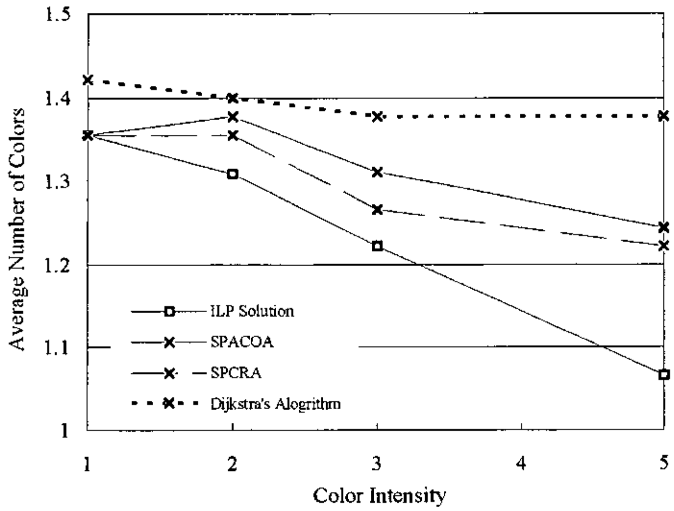

Fig 7. Average number of colors on paths of static connection requests vs. network color intensity

B. Minimum Total-Color Disjoint-Paths Problem

We have developed an ILP formulation and two heuristics for the MTCDiP problem, i.e., the Disjoint-Paths Color Reduction Algorithm (DPCRA) and the Disjoint-Paths All Color Optimization Algorithm (DPACOA). We run computer simulations for both the individual connection request problem and the static traffic problem.

B.1. Individual Connection Request Problem

For every source-destination pair in the networks, we solve the ILP and use the heuristics to obtain link-disjoint paths with minimum total number of colors. We then compare the average numbers of colors on all the paths. To establish an upper bound for the results of the heuristics, we also run Suurballe's algorithm on all the source-destination pairs. Two sets of the results are depicted in Fig. 8 and Fig. 9. Results on other network topologies are similar.

Based on the simulation results, the link-disjoint paths obtained from the Disjoint-Paths All Color Optimization Algorithm are closest to the optimal ILP solutions in the average numbers of total colors on the paths. This is because DPACOA selects optimal colors from all colors in the network while DPCRA is restricted to the colors on the initial shortest paths.

The network nodal degree, the color intensity, and the network size have similar impacts on the number of colors on the disjoint-paths as in the simulations in Section V.A.1. Higher nodal degree, or greater color intensity, or smaller network size all contribute to fewer colors on the disjoint-paths.

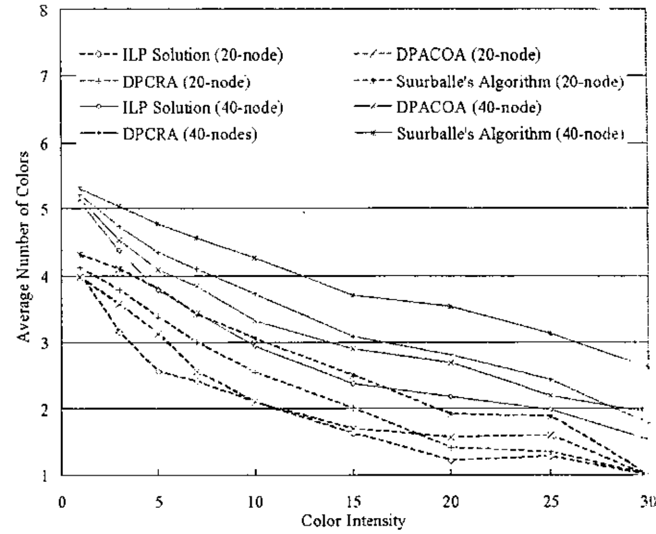

Fig 8. Average number of colors on link-disjoint paths of all source-destination pairs vs. network color intensity. Network nodal degree = 2.6

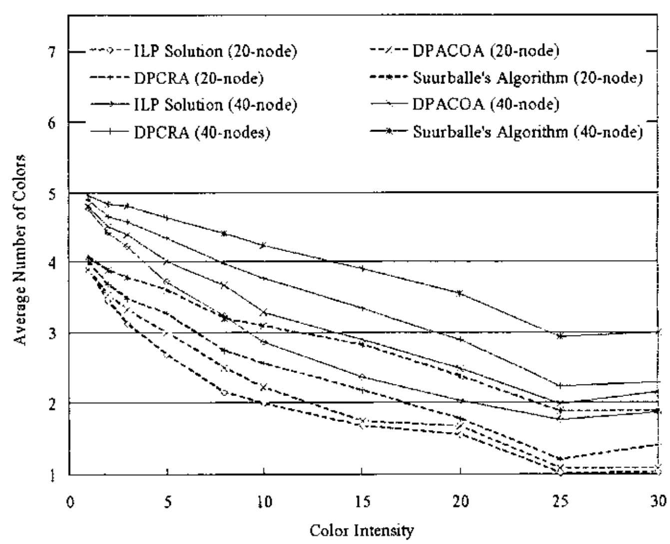

Fig 9. Average number of colors on link-disjoint paths of all source-destination pairs vs. network color intensity. Network nodal degree = 3.0

B.2. Static Traffic Problem

The static connection requests include connections between all the source-destination pairs in the networks. We solve the ILP for all the requests to obtain the disjoint-paths with the minimum average number of total colors. We then apply the heuristics on the same set of requests. To establish an upper bound for the results of the heuristics, we run Suurballe's algorithm. We use a randomly generated 10-node network of nodal degree 2.6 for the simulations. The capacity of every link is assumed to be infinite. The results are depicted in Fig. 10. The simulation results confirm that the paths obtained from the Disjoint-Paths All Color Optimization Algorithm are closest to the optimal ILP solutions.

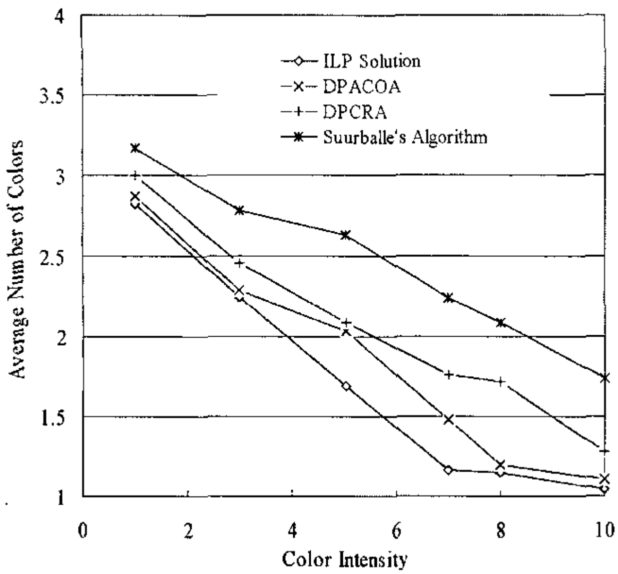

Fig 10. Average number of colors on paths of static connection requests vs. network color intensity

C. *Minimum Overlapping Color Disjoint-Paths Problem*

We have developed an ILP formulation and three heuristics for the MOCDiP problem, i.e., the Simple Two-Step Algorithm (STSA), the Minimum-Color First-Path Algorithm (MCFPA), and the Joint-Search Minimum Overlapping Color Algorithm (JSMOCA). We run computer simulations for both the individual connection request problem and the static traffic problem..

C.1. Individual Connection Request Problem

For every source-destination pair in the networks, we solve the ILP and use the heuristics to obtain link-disjoint paths with minimum number of overlapping colors. We then compare the average numbers of overlapping colors on all the paths. The STSA is used as an upper bound for the other two heuristics. Two sets of the results are depicted in Fig. 11 and Fig. 12. Results on other network topologies are similar.

Based on the simulation results, the link-disjoint paths obtained from the Joint-Search Minimum Overlapping Color Algorithm are closest to the optimal ILP solutions in the average numbers of overlapping colors on the paths. This is because JSMOCA selects optimal colors from all colors in the network while MCFPA is restricted to the colors on the initial shortest paths.

The number of overlapping colors is directly related to how susceptible the two paths are to a single failure event. As the color intensity increases, there are fewer colors in the network, and it is more likely that the two paths will share a greater number of colors. Thus, the paths become more susceptible to common failures.

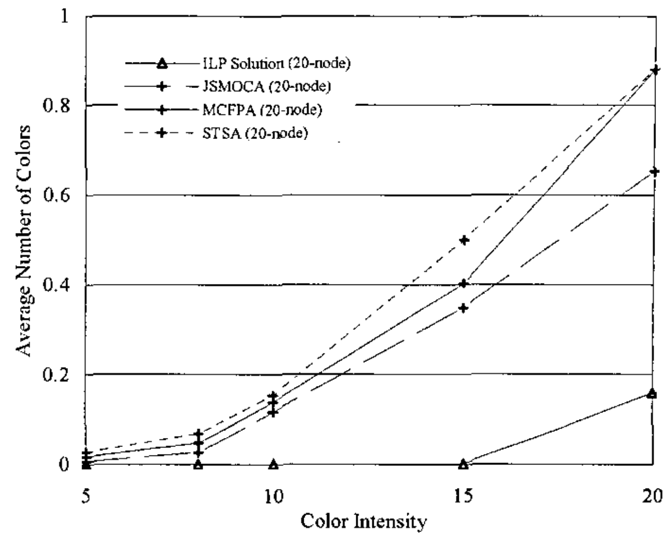

Fig 11. Average number of overlapping colors on link-disjoint paths of all source-destination pairs vs. network color intensity. 20-node network. Nodal degree = 2.6

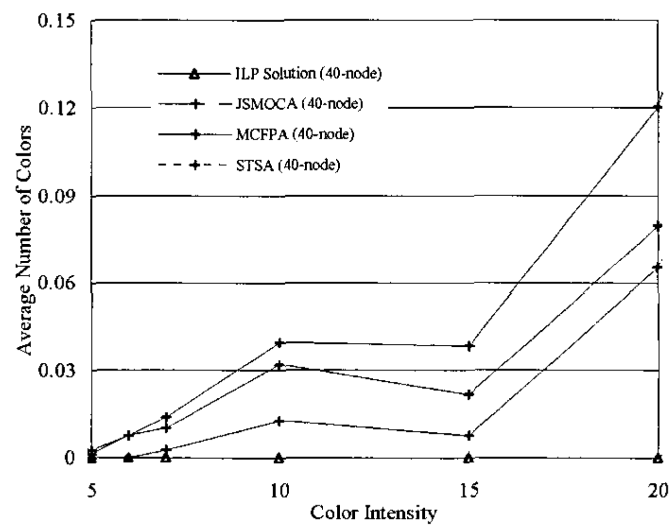

Fig 12. Average number of overlapping colors on link-disjoint paths of all source-destination pairs vs. network color intensity. 40-node network. Nodal degree = 3.0

C.2. Static Traffic Problem

The static connection requests contain all the source-destination pairs in the networks. We solve the ILP for all the requests to obtain the disjoint-paths with the minimum average number of overlapping colors. We then apply the heuristics on the same set of requests. The STSA is used as an upper bound for the other two heuristics. We use a randomly generated 10-node network of nodal degree 2.6 for the simulations. The capacity of every link is assumed to be infinite. The number of overlapping colors is 0 for lower color intensities; hence, we only present results for the higher color intensities. The results are depicted in Fig. 13. The simulation results confirm that the paths obtained from the Joint-Search Minimum Overlapping Color Algorithm are closest to the optimal ILP solutions.

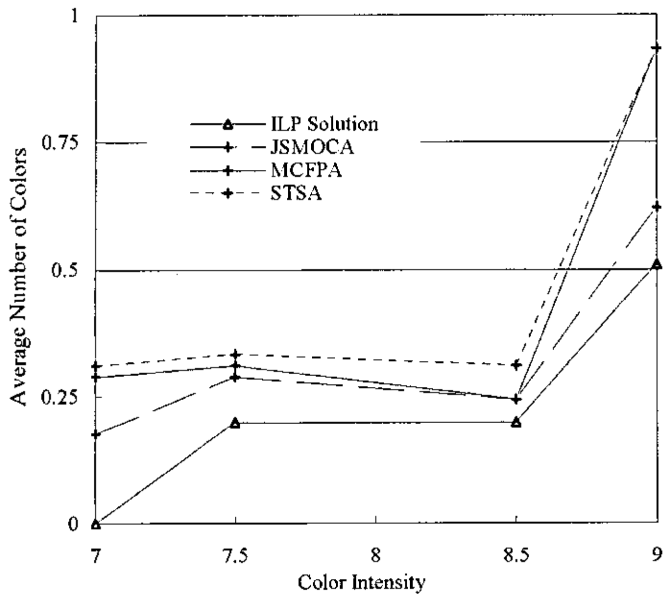

Fig 13. Average number of colors on paths of static connection requests vs. network color intensity

VI. Conclusion

In this paper we discussed a relatively new class of path routing problems referred to as minimum-color path problems. These problems have practical significance in applications that require finding paths satisfying various reliability objectives such as minimizing the probability of failure on a path. We discuss three problems - MCSiP, MTCDiP, and MOCDiP. We prove that these problems are NP-complete and formulate them as ILPs. The solution of the ILPs are intractable for larger networks, and hence we propose various heuristics. The heuristics execute in polynomial times and yield solutions that are very close to the optimal.

Various factors affect the number of colors on the paths, including the nodal degree, the color intensity, and the number of nodes in the network. An increase in the nodal degree helps reduce the number of colors on the paths for the problems that attempt to minimize the total number of colors on the path (MCSiP, MTCDiP). This is due to the fact that there is a greater choice of routes for the connections. SPACOA performs the best while reducing the total number of colors on a single path (MCSiP), whereas DPACOA is the most successful with the two link-disjoint path problem (MTCDiP).

The color intensity also affects the number of colors on the paths. An increase in the color intensity increases the number of overlapping colors among two link-disjoint paths, thereby making both paths more susceptible to the same failure (MOCDiP). JSMOCA reduces the total number overlapping colors and is reasonably close to the ILP's optimal solution.

The generic nature and the wide-ranging applications of these problems make them a perfect candidate for further study. A possible future contribution would be to study the problems with different failure probabilities for each color. In the present environment, the failure probabilities of all colors are assumed to be the same, and hence the number of colors on the path directly relates to the failure probability of the path. Another interesting work would be to consider the multiple-path problems (MTCDiP and MOCDiP) without the link-disjoint constraint. In this case, the primary and backup paths may share a link if the probability of that link failing is low. Another challenge is solving the minimum-color path problems in a dynamic environment. In a dynamic environment, connection requests arrive one at a time, stay for a finite time, and then depart. The typical objective in the dynamic environment is to reduce the blocking of future connections.

References

[1] E. Rosen, A. Viswanathan, R. Callon, "RFC 3031 Multiprotocol Label Switching Architecture". *www.ietf.org*

[2] B. Davie, P. Doolan, Y. Rekhter, *Switching in IP Networks*. Morgan Kaufmann, 1998.

[3] B. Mukherjee, *Optical Communication Networks*. New York: McGraw Hill, 1997.

[4] R. Ramaswami and K. N. Sivarajan, *Optical Networks: A Practical Perspective*. San Francisco, Morgan Kaufmann, 1998.

[5] O. Gerstel, "Opportunities for optical protection and restoration," *Proceedings, IEEE OFC '98*.

[6] T. Wu, *Fiber Network Survivability*, MA: Artech House, 1992.

[7] S. Ramamurthy, B. Mukherjee, "Survivable WDM mesh networks, part I – protection," *Proceedings. IEEE INFOCOM '99*.

[8] J. Strand, "Issues for routing in the optical layers," *IEEE Communication Magazine*, Feb. 2001

[9] P. Sebos, et al. "Auto-discovery of shared risk link groups," *Proceedings, IEEE OFC 2001*.

[10] S. Yuan, J. P. Jue, "Shared protection routing algorithm for optical network," *Optical Network Magazine*, May-June, 2002.

[11] S. Yuan, J. P. Jue, "Dynamic path protection in WDM mesh networks under risk disjoint constraint," *IEEE Globecom* 2004

[12] T. Cormen, C. Leiserson, R. Rivest, *Introduction to Algorithms*, McGraw Hill, 1997.

[13] J. W. Suurballe, R. E. Tarjan, "A quick method for finding shortest pairs of disjoint paths," *Networks*, vol. 14, pp. 325-336, 1984.

[14] Remesh Bhandari, *Survivable Networks – Algorithms for Diverse Routing*, Kluwer Academic Publishers.

[15] R. Carr, S. Doddi, G. Konjevod, M. Marathe, "On the red-blue set cover problem," *Proceedings of the eleventh annual ACM-SIAM symposium on Discrete algorithms*, pp. 345 – 353, 2000

[16] R. Chang, S. Leu, "The minimum labeling spanning trees," *Information Processing Letters*, Vol. 63, pp. 277-282, 1997.

[17] S. Krumke, H. Wirth, "On the minimum label spanning tree problem," *Information Processing Letters*, Vol. 66, pp. 81-85, 1998.

[18] Y. Wan, G. Chen, Y. Xu, "A note on the minimum label spanning tree," *Information Processing Letters*, Vol. 84, pp. 99-101, 2002.

[19] "ITU-T G.803: Architecture of Transport Networks Based on the Synchronous Digital Hierarchy (SDH)," *http://www.itu.int*

[20] "ITU-T I.122: Framework for frame mode bearer services," *http://www.itu.int*

[21] "ITU X.25: Interface between DTE and DCE for terminals operating in packet mode and connected to public data networks by dedicated circuit," *http://www.itu.int*

[22] M. R. Garey and D. S. Johnson, *Computers and intractability: A guide to the theory of NP-Completeness*. pp.222. New York: W. H. Freeman Company, 1979.

[23] G. Ellinas, E. Bouillet, R. Ramamurthy, J. Labourdette, S. Chaudhuri and K. Bala, "Routing and restoration architectures in mesh optical networks," *SPIE Optical. Networks Magazine*, vol. 4, Jan/Feb. 2003.

[24] http://www.ilog.com/products/cplex

[25] G. Z. Li, B. Doverspike, and C. Kalmanek, "Fiber span failure protection in mesh optical networks," *Proc. SPIE OptiComm 2001: Optical Networking and Communications*, vol. SPIE 4599.

[26] http://www.ilog.com/products/cplex/

[27] http://www.algorithmic-solutions.com/enleda.htm