Link Failure Recovery Over Large Arbitrary Networks: The Case of Coding

Serhat Nazim Avci and Ender Ayanoglu, *Fellow, IEEE*

***Abstract*—Network coding-based link failure recovery techniques provide near-hitless recovery and offer high capacity efficiency. Diversity coding is the first technique to incorporate coding in this field and is easy to implement over small networks. However, the capacity efficiency of this implementation is restricted by its systematic coding and high design complexity despite having lower complexity than the other coding-based recovery techniques. In this paper, we propose a simple column generation-based design algorithm and a novel advanced diversity coding technique to achieve near-hitless recovery over large networks. The traffic matrix, which consists of unicast connection demands, is decomposed into traffic vectors for each destination node. Further, the connection demands in each traffic vector are partitioned into coding groups. The design framework consists of two parts: a main problem and a subproblem. The main problem is solved with Linear Programming (LP) and Integer Linear Programming (ILP), whereas the subproblem can be solved with different methods. Simulation results suggest that the novel design algorithm simplifies the capacity placement problem, which enables implementing diversity coding-based recovery including the novel coding structure on large networks with arbitrary topology. It achieves near-hitless recovery with an almost optimal capacity efficiency for any single destination-based recovery.**

***Index Terms*—Computer network reliability, fault tolerance, linear programming, integer linear programming, network coding.**

I. Introduction

THE protection of the data in wide area networks is very important since the network failures, which happen regularly, pose social, economical, and security threats. A breakdown of the network failure statistics can be found in [1]. In this paper, we focus on single link failure recovery, which makes up to 70% of all network failures [2]. Various protection and recovery techniques are developed to minimize the costs of such failures, each offering a tradeoff in terms of different recovery metrics. The two main recovery metrics are restoration speed and capacity efficiency. Capacity efficiency is calculated by the total required capacity, in fiber miles, to route and protect the data streams. Restoration speed is measured by the total outage duration between the instant of failure and the restoration of failed traffic. Capacity efficiency has as a pre-failure cost, whereas restoration speed has a post-failure cost. The goal is to minimize both of these costs.

Recovery techniques differ from each other depending on if they dedicate the spare resources to single demands or share them among different failure scenarios and traffic demands. Hitless recovery is considered as the ideal case of link failure recovery where the end nodes do not experience the failure at all. The biggest advantage of dedicated recovery techniques is near-hitless recovery, which has very minimal error detection and switching, without signaling and rerouting operations. $1+1$ Automatic Protection Switching (APS) has two link-disjoint dedicated paths for each connection demand and those paths are employed to transmit the same data to the destination node [3]. The destination node switches to the protection path and restores the traffic nearly instantaneously in the case of a link failure over the primary path. However, $1+1$ APS is capacity-inefficient since it requires more than 100% redundant capacity. In [4], it is cited that $1+1$ APS is currently employed in today's networks despite its low capacity efficiency, which is an indication of the need for nearly instantaneous link failure recovery.

Coding-based recovery techniques emerged to improve the capacity efficiency of dedicated protection techniques. In a coding-based recovery technique, the dedicated protection paths share the spare resources by coding operations, in particular, erasure coding [5]. The first coding-based recovery technique is called diversity coding [5]. This technique has two advantages. First, like $1+1$ APS, it offers nearly instantaneous recovery. Second, like rerouting-based restoration schemes, it is capacity-efficient. The first work of diversity coding [5] predates network coding, usually considered to have been introduced in [6].

In [7], a heuristic algorithm is developed to implement diversity coding over arbitrary networks. In [8], optimal algorithms for the diversity coding technique are developed. It is shown that diversity coding can offer competitive capacity efficiency while providing near-hitless recovery. In [1], a coding-based solution named Coded Path Protection (CPP) is developed by converting a solution of Shared Path Protection [9]. In CPP, sharing of the spare resources is replaced with the employment of these resources to code different paths, which results in higher restoration speed, higher transmission integrity, and lower error signaling complexity. The symmetric nature of CPP in bidirectional networks allows encoding and decoding inside the network for unicast demands.

In [10] and [11], network coding-based protection schemes, similar to diversity coding, are proposed in which coding operations are carried out over trees and trails, respectively. These schemes are called $1+N$ protection and differ from diversity

Manuscript received October 8, 2013; revised April 2, 2014, July 28, 2014, and January 5, 2015; accepted March 16, 2015. Date of publication March 27, 2015; date of current version May 14, 2015. This work was partially supported by NSF under Grant No. 0917176. The associate editor coordinating the review of this paper and approving it for publication was J. Widmer.

The authors are with the Center for Pervasive Communications and Computing, Department of Electrical Engineering and Computer Science, University of California, Irvine, CA 92697-2625 USA (e-mail: serhat.nazim.avci@gmail.com; ayanoglu@uci.edu).

Digital Object Identifier 10.1109/TCOMM.2015.2417567

coding due to their symmetric bidirectional nature. Diversity coding is also implemented on bidirectional networks only after converting them into directed networks, which enables asymmetric solutions and hence higher restoration speed and higher capacity efficiency. In [12], the cost efficiencies of a network coding-based recovery technique and a simpler version of diversity coding technique are evaluated.

The coding-based techniques mentioned above have certain assumptions to make them easier to implement. First, in systematic coding, primary paths are exempt from coding operations. Second, in these techniques, coding operations are bound to specific topologies. Third, these protection schemes require strict link-disjointness between each primary path and the protection paths. However, those assumptions restrict their capacity efficiencies. In [4], an argument that $1 + N$ coding requires high nodal degree, which reduces its efficiency on sparse topologies, was made. However, Generalized $1 + N$ protection partitions the connections into groups, and protects each group separately which helps to mitigate the high nodal degree requirement.

Nonsystematic coding, where both primary and protection paths are incorporated into coding operations, is implemented in wireless mesh networks for single link failure recovery in [13]. Reference [13] is based on finding bipartite network equivalents for networks with arbitrary topology, which is not required in our work published in [7], [8], [14], or this paper. In [14], nonsystematic diversity coding is implemented using a heuristic algorithm for static provisioning. In that paper, a coding group is defined as a set of connection demands, with a common destination node, that are coded and protected together. The connection demands are added to the existing coding groups one by one ensuring the decodability of the coding structures. In [14], it is shown that nonsystematic diversity coding has more coding flexibility than conventional diversity coding resulting in higher capacity efficiency. In [15], a general network-coding based approach is presented which employs nonsystematic coding and does not explicitly require link-disjointness between primary paths and protection paths. However, this approach can protect at most two connection demands in restricted specific topologies. In [16], the restriction over the number of protected connection demands is removed for bidirectional networks. However, it produces symmetric solutions in bidirectional networks which induce higher restoration time. In [17], network coding is used to recover from shared risk link group (SRLG) failures. Reference [17] uses splitting of flows into two, or bifurcation, to achieve polynomial complexity during the design process, thereby increasing the required capacity. In [17], undirected networks are converted into directed networks in a similar way to we do in this paper.

Due to high design complexity limitations of optimal solutions, the coding-based recovery techniques in the literature, such as [10], [15], [16], fail to offer solutions in large realistic networks even though they have potential in terms of capacity efficiency and restoration time. These techniques are tested on relatively small networks and with relatively few traffic demands compared to the long-distance networks of the U.S. and France, to be discussed in the sequel. In [18], a heuristic approach which runs in polynomial time is introduced. In [19], a novel two-step approach is presented to cope with high design complexity in realistic networks. The first step of this algorithm is the pre-processing phase in which all candidate coding groups are calculated and enumerated. In the second step, some of those candidate coding groups are selected and placed on the networks to meet the traffic demand. This approach overcomes the complexity incurred by the size of the traffic matrix. However, the number of coding groups is exponentially dependent on the network size and the nodal degree of the destination node. This paper extends [19].

This paper contributes to the field of diversity coding-based (or network coding-based) link failure recovery in three novel ways. First, we introduce a simple and modular design algorithm that achieves LP optimal solutions of coding-based recovery techniques in large arbitrary networks. The static traffic matrix which consists of multiple unicast connection demands from each node to every other node is partitioned into traffic vectors in terms of their destination node. Those traffic vectors are input to the algorithm independently and each simulation scenario inputs demands with multiple source nodes and a single destination node. The design algorithm uses the column generation technique which does not require explicit enumeration of the coding groups. It starts the problem with a small set of coding groups and creates new coding groups when they are needed. The underlying coding structure of this algorithm is arbitrary as long as the destination nodes of the connections are the same, which offers a solution for different techniques under the same framework. Second, we improve the coding structure of simple diversity coding by offering a technique we call coherent diversity coding. This coding structure is implemented using an Integer Linear Programming (ILP) formulation. In a coherent diversity coding structure, we implement a more relaxed link-disjointness criterion between the paths in a coding group. This enables one to form coding groups with higher flexibility and bigger size. The decodability is preserved while the high nodal degree requirement is mitigated. Moreover, coherent diversity coding incorporates nonsystematic coding. Third, we show that the proposed coding technique almost achieves the theoretical lower bound for any single destination-based recovery technique. In this paper, the performance of the new proposed coding technique and the column generation-based design algorithm are investigated compared to conventional (systematic or nonsystematic) diversity coding and p-cycle protection [3]. The simplicity of the new design algorithm is also tested based on a set of simulations over the relatively large long-distance networks of the U.S. and France.

II. Problem Description

Our research problem is survivable uncapacitated network design against single link failures using network coding. We assume a single link failure disconnects traffic on that link in both directions. Given a communication network modeled by an undirected graph $G(V, S)$, a set of weights a_e representing the length of the node of link modeled by the edge $e \in S$, and an arbitrary static unicast traffic matrix of size $|V| \times |V|$, the objective is to minimize the total transmission cost. The

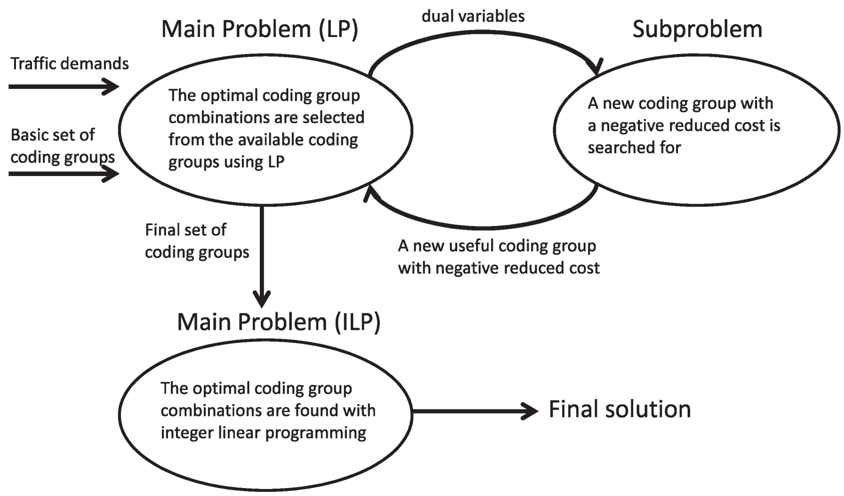


Fig. 1. Steps in the column generation method.

transmission cost on each link is calculated by multiplying the weight and the utilized capacity of that link. The summation of the costs of each link makes up the total cost. Before solving this problem, the undirected network graph $G(V, S)$ is converted into a directed one $G(V, E)$ by converting the undirected links into directed arcs in opposite directions with the same weight and infinite available capacity, as done in Section 4.3 of [20].

In our model, each source node sends two copies of the same data over link-disjoint paths to provide near-hitless recovery against any single link failure. Different data signals are encoded and decoded in the network nodes to minimize total transmission cost. We developed different network coding mechanisms for that purpose as explained in Section IV.

More specifically, our goal is to improve upon our work published in [8], in terms of the speed and complexity of the design. To that end, we will employ a technique from the field of operations research, known as column generation. This technique is described in the next section.

III. Column Generation Method

The column generation method is an effective technique to solve relatively large linear programming (LP) formulations without explicitly enumerating all possible variables [21]. In some problems, only a small subset of the variables are nonzero in the final solution. In those problems, column generation starts with a small set of variables and creates new and useful variables (columns) which will be likely employed in the final solution. In general, column generation dramatically decreases the time and space complexity depending on the nature of the problem. In the network-coding based link failure recovery problem, we have observed that column generation technique results in significant time and memory savings, and therefore it enables the near-optimal implementation of efficient network coding-based techniques over large realistic networks. As it will be shown in Section VII, column generation achieves optimal solutions for LP but there is a small optimality gap for ILP solutions.

Column generation has been used for different LP problems, including the well-known cutting stock problem [21]. The cutting stock problem is to satisfy paper demand of different widths by cutting fixed width rolls in different patterns. The goal is to use a minimum number of rolls. The problem starts with a small set of basic cutting patterns. The useful cutting patterns are generated one-by-one. We observed that the diversity coding-based link failure recovery problem is very similar to the cutting stock problem. Diversity coding over arbitrary networks can be implemented like the cutting stock problem as long as the cutting patterns are replaced by coding groups and the demands for different widths of paper are replaced with the traffic demands of a single destination node. The only difference is the fact that coding groups can have different costs, whereas in the cutting stock problem, each cutting pattern is cut from rolls with the same total width. Other advanced methods developed for the cutting stock problem, such as extended Dantzig-Wolfe decomposition [21], can also be applied to the implementation of diversity coding.

The column generation technique is also applied to the p-cycle protection [22] and SPP [23] problems resulting in significant time and memory savings. It is a better fit to diversity coding technique than p-cycle protection and SPP since, in diversity coding, there is a single subproblem that generates coding groups. However, in p-cycle protection, there is a subproblem for both generating p-cycles and generating candidate paths for each connection demand. Likewise, in SPP, there is a different subproblem for generating candidate path pairs for each connection demand.

The column generation method for diversity coding is visualized in Fig. 1. There are two main components of this method. The main problem, which is also called the *Coding Groups Placement Problem*, inputs the traffic demands and a subset of the basic coding groups. This set includes coding groups consisting of a single connection demand originating from each source node to the destination node. The main problem in this step is an LP formulation that finds the optimal coding group combinations to meet the traffic demands. Each coding group has a cost which is the total length of all paths

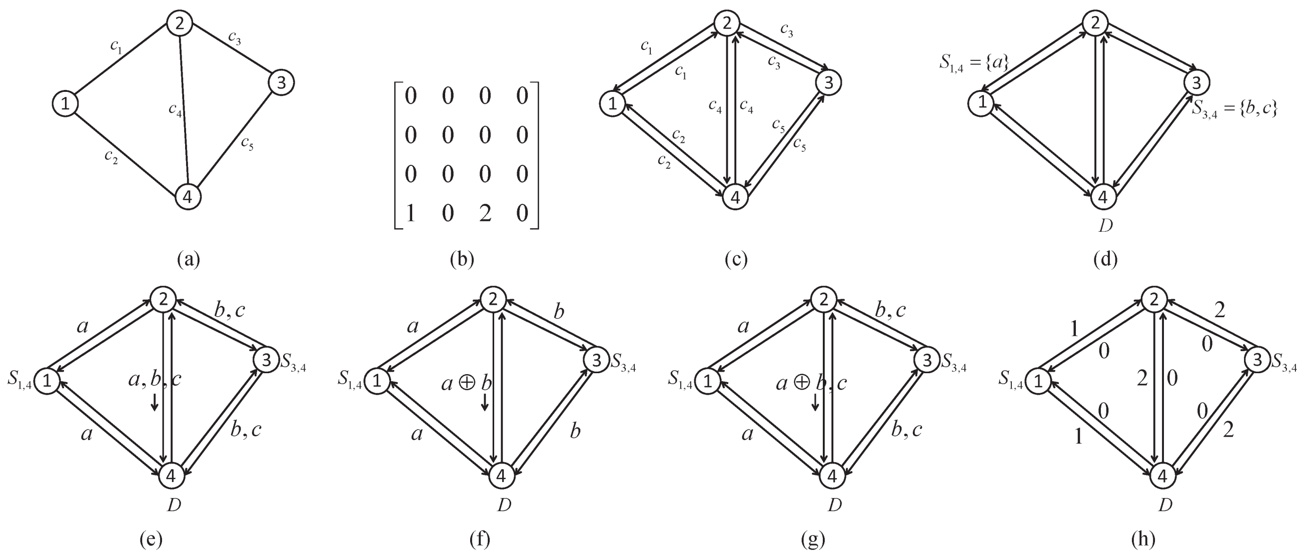


Fig. 2. Coding group generation. (a) The undirected network topology. (b) Traffic matrix. (c) The undirected network is converted to directed network. (d) $S_{1,4}$ sends a and $S_{3,4}$ sends b and c to D. (e) $1+1$ APS solution. (f) Subproblem creates a new coding group. (g) Network coding-based solution. (h) Utilized capacity on each link.

in that coding group. After the first run, it passes the dual variables of the solution to the subproblem. The dual variables of this solution are obtained using the native function of the optimization tool, where the syntax depends on the program CPLEX is called from. The subproblem, which is also called the *Coding Group Generation Problem*, attempts to find a new useful coding group. The subproblem generates a reduced cost. A new useful coding group has a negative reduced cost, specified in Section IV-B, given the dual variables of the main problem. The new useful coding group is input to the main problem iteratively. In the next round, optimal coding group combinations are found given the expanded coding group set. The dual variables of this run are input to the subproblem as before. This iterative operation is carried out until the subproblem cannot find any new coding group with a negative reduced cost. The main problem is then solved one last time as an ILP. The gap between ILP and LP solutions of the main problem is generally very small, as will be discussed in Section VII.

A. Example 1

As an example, an undirected network and a traffic matrix are given as input in Fig. 2(a) and (b). In Fig. 2(a), the cost (length) of each span is given as c_i for $1 \leq i \leq 5$. The traffic matrix is asymmetric but it can be symmetric. As the first step, in Fig. 2(c), the undirected network is converted into a directed network by converting undirected spans into directed edges in opposite directions with the same cost. In Fig. 2(d), the destination node D and the source nodes are shown. There is one unit traffic between $1 \rightarrow 4$ symbolized by data a whereas traffic $3 \rightarrow 4$ has two unit traffic demands given as b and c. The initial solution of the master problem is $1+1$ APS in which all traffic demands are submitted over link disjoint primary and protection paths as shown in Fig. 2(e). The total cost of the solution is equal to $c_1 + c_2 + 2 \times c_3 + 3 \times c_4 + 2 \times c_5$. The subproblem tries to find a new coding group that will potentially improve the solution of the main problem. It finds the coding group shown in Fig. 2(f) and sends it to the main problem. The main problem runs one more time after incorporating the new coding group. The new solution of the main problem is shown Fig. 2(g). Instead of separately sending signals a and b over link $2 \rightarrow 4$, the new solution encodes them into $a \oplus b$ by using the new coding group and reduces the utilized capacity on that link, where $\oplus$ is logical EX-OR operation on binary variables, a and b. The subproblem runs again but cannot find a new coding group indicating the optimal result is achieved. In Fig. 2(h), the utilized capacity on each link is calculated resulting at a total cost of $c_1 + c_2 + 2 \times c_3 + 2 \times c_4 + 2 \times c_5$.

IV. ILP Formulations

This section presents the solution models of our research problem in detail. As stated before, each data is sent over two link-disjoint paths and the goal is to minimize total cost in terms of fiber miles used traversed by the data. First, we decompose the traffic matrix into vectors based on the destination nodes of the connections. Second, we use coding schemes to encode multiple copies of different data from the same traffic vector to minimize the total cost. Third, we repeat the second step for each destination node. In these coding schemes, different data signals are grouped under coding groups. In the subsections, we first present the general optimization framework titled as the main problem, which minimizes the total cost. Then, we present different coding schemes that work under this modular optimization framework as subproblems.

The main problem finds the optimal combination of coding groups out of a given set and places them on the network to meet the traffic demands of a single destination node. Throughout the iterative process, the main problem is realized with an LP formulation, whereas in the last step, the formulation is converted to an ILP since in the final solution coding groups must be represented in integer numbers. On the other hand, the realization of the subproblem is not unique. The coding group generation algorithm depends on the adopted coding structure. In addition, the way new coding groups are generated can be realized by heuristic techniques, which does not violate the LP optimality of the whole algorithm. These algorithms are run for each destination node independently. In Section IV-B, we present three different coding group generation algorithms using mixed integer programming (MIP) or ILP formulations.

A. *Main Problem (Coding Groups Placement Problem)*

An LP formulation is developed to implement the coding groups placement algorithm, which serves the main problem of the column generation method. The goal is to place the coding groups with minimum total cost while meeting the traffic demands. The input parameters of the LP are

- d: The common destination node,
- CG^d: The set of coding groups for destination node d, this set is expanded at each iteration,
- V: The set of nodes,
- t_f^d: The traffic demand from source node f to destination node d,
- $cost_i$: The cost of coding group i of set CG^d,
- $CG_{i,f}^d$: The number of connections originating from node f in coding group i destined for node d.

The variables related to the coding groups placement problem are

- $n(i)$: Keeps the number of instances of coding group i of set CG^d placed on the network, normally a continuous variable.

The variables $n(i)$ are continuous when the main problem is LP. They are converted to integer variables at the final ILP step of column generation.

The objective function is

$$\min \sum_{i \in CG^d} cost_i \times n(i). \tag{1}$$

The following inequalities ensure a sufficient number of coding groups are placed to protect all of the traffic demands

$$\sum_{i \in CG^d} CG_{i,f}^d \times n(i) \geq t_f^d \qquad \forall f \in V,\ f \neq d. \tag{2}$$

A flow diagram of the column generation method in terms of the parameters and variables of the LP formulations is shown in Fig. 3, where π_f are the dual variables of the constraints in (2). The traffic demand parameters t_f^d and an initial basic coding group set $CG_{initial}^d$ are input to the main problem. After the first run, the main problem inputs the resulting dual values of the constraints to the subproblem. The subproblem returns a new coding group with negative reduced cost, if available. The iterative process terminates when the subproblem cannot produce any more new coding groups with reduced cost. Then the variables $n(i)$ are converted to integer variables and ILP is run at the last step to get the final solution.

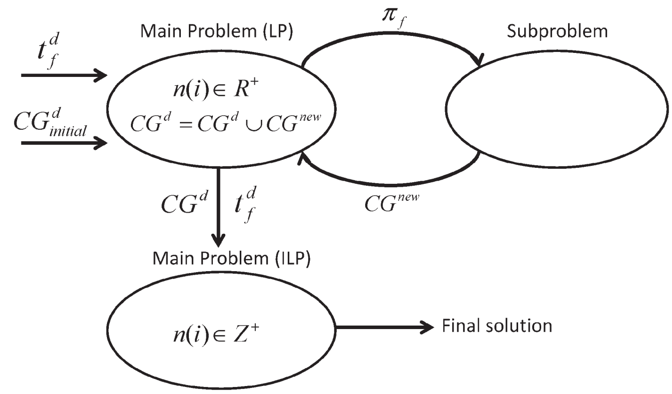


Fig. 3. Steps in the column generation method in terms of LP and ILP variables.

B. *Subproblem (Coding Group Generation Problem)*

The objective of the subproblem is to find a new coding group in each iteration that will be useful in the main problem. The subproblem inputs the dual variables of the main problem and returns a new coding group. A new coding group can be selected among many which have negative reduced costs. In this paper, we opt to search for a new coding group with the minimum negative reduced cost until there is none. We present three different coding group generation algorithms, each implementing a different version of diversity coding. These versions have the tradeoff of simplicity versus capacity efficiency. In the following subsections, they are presented in increasing order of capacity efficiency and design complexity.

1) Systematic Diversity Coding: In this algorithm, we adopt systematic diversity coding where only protection paths are encoded. The core algorithm is adopted from the diversity coding tree algorithm in [8]. In a coding group, there is a primary tree serving as the union of the primary paths of the protected connections. There is also a link-disjoint protection tree whose branches originate from the source nodes of the protected connections. Those branches merge when they come together until they reach at the destination node. We also define a metric called "voltage" value for each node. Those "voltage" values are used to prevent cyclic structures inside primary and protection trees similar to cycle exclusion technique in [24]. They are used in a sense similar to Kirchoff's voltage law. An example is taken from [8] and is shown in Fig. 4(a). There are three connection demands originating from S_1, S_2, and S_3 going to node D. The solid black lines represent the primary tree whereas dashed lines represent the protection tree.

The input parameters required in the MIP formulation of the coding group generation algorithm based on systematic diversity coding are

- $G(V, S)$: Network graph with undirected spans,
- S: The set of spans in the network, a span consists of two links in the opposite directions,
- a_e: Cost associated with link e,
- $G(V, E)$: Network graph by converting undirected spans into directed arcs with the same weight,
- $\Gamma_i(f)$: The set of incoming links of each node f,
- $\Gamma_o(f)$: The set of outgoing links of each node f,

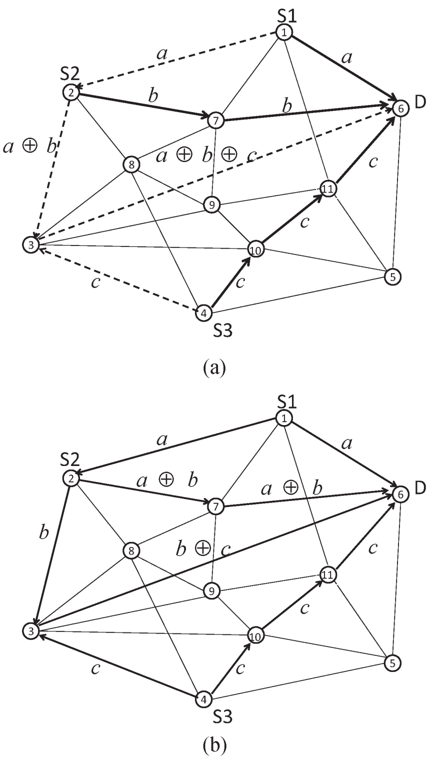


Fig. 4. (a) An example of the systematic diversity coding tree structure. There are three link-disjoint primary paths spanned by the primary tree and there is a link-disjoint protection tree. (b) An example of nonsystematic diversity coding structure for the same set of connections.

- d: The common destination node,
- ND: The nodal degree of the destination node d,
- α: A constant employed in the algorithm where $\frac{1}{|V|} \geq \alpha \geq 0$,
- β: A constant employed in the algorithm, $\beta \geq 2 \times \max(|V|, ND)$,
- π_f: The values of the dual variables of the main problem for destination node d.

The set of variables of this MIP formulation are

- CG_f^{new}: Integer variable, equals the number of connections originating from node f to node d in the new coding group,
- $y_e \in \{0, 1\}$: Integer variable, equals 1 iff the primary tree of the new coding group passes through link e and goes to node d,
- $c_e \in \{0, 1\}$: Integer variable, equals 1 iff the protection tree of the new coding group passes through link e and goes to node d,
- p_f: A continuous variable between 0 and 1. It keeps the "voltage" value of node f [8] in the protection tree of the new coding group,
- g_f: Same description as p_f except it is used for the primary tree of the new coding group.

The objective function minimizes the reduced cost of a new coding group

$$\min \sum_{e \in E} (y_e + c_e) \times a_e - \sum_{f \in V} CG_f^{new} \times \pi_f. \qquad (3)$$

If the value of the objective function comes out to be negative, then a new coding group is found and input to the main problem.

$$\sum_{f \in V} CG_f^{new} \leq ND - 1, \qquad (4)$$

$$\sum_{e \in \Gamma_o(f)} y_e = CG_f^{new} + \sum_{e \in \Gamma_i(f)} y_e \quad \forall f \in V, f \neq d, \qquad (5)$$

$$\sum_{e \in \Gamma_i(d)} y_e = \sum_{f \in V} CG_f^{new}, \qquad (6)$$

$$\sum_{e \in \Gamma_o(d)} y_e + c_e = 0, \qquad (7)$$

$$\sum_{e \in \Gamma_o(f)} c_e \geq \frac{CG_f^{new}}{\beta} + \frac{\sum_{e \in \Gamma_i(f)} c_e}{\beta} \quad \forall f \in V, \ f \neq d. \qquad (8)$$

$$\sum_{e \in \Gamma_i(d)} c_e \geq \frac{\sum_{f \in V} CG_f^{new}}{\beta}, \qquad (9)$$

$$y_{e1} + y_{e2} + c_{e1} + c_{e2} \leq 1 \quad \forall e1, e2 \in g, \forall g \in S, \qquad (10)$$

$$g_f - g_f \geq \alpha \cdot y_e - (1 - y_e) \quad \forall e \in E, \qquad (11)$$

$$p_f - p_f \geq \alpha \cdot c_e - (1 - c_e) \quad \forall e \in E. \qquad (12)$$

Inequality (4) ensures that the size of the new coding group does not exceed $ND - 1$. Equation (5) carries out the origination and continuation of the primary tree, whereas equation (6) and (7) carry out the termination of the primary tree. Inequality (8) is responsible for the origination and continuation of the protection tree, whereas inequality (9) and (7) are responsible for the termination of the protection tree. Inequality (10) makes sure that primary and protection trees are link-disjoint. Inequalities (11) and (12) assign voltage values to nodes to prevent getting cyclic structures in primary and protection trees, respectively.

2) Nonsystematic Diversity Coding: In this section, the coding groups are generated based on a more generic coding structure where both primary and protection paths can be encoded. Nonsystematic diversity coding is based on *Lemma 1* from [13]. We developed our own design algorithm to implement this technique over large arbitrary networks. This coding structure increases the capacity efficiency of systematic diversity coding with extra design complexity. An example is shown in Fig. 4(b). Different from systematic diversity coding, the primary paths of $S_1 - D$ and $S_2 - D$ are encoded. The core algorithm to generate new coding groups in the column generation method is an ILP formulation taken from [8] with small changes. Reference [8] presents how to optimally build nonsystematic diversity coding structures. The algorithm in [8] looks for every possible coding scenario by eliminating the invalid cases that can be identified as *coding cycles*. In a coding group, if path i is not coded with any path carrying demand f, they become indirectly related to each other. Otherwise, they become directly related and considered in the same subgroup. Paths and demands in separate coding groups are unrelated. The ILP formulation of the nonsystematic diversity coding group generation algorithm has a set of binary integer variables taking values from the set {0, 1}

- $x_e(i)$: Equals 1 iff the path i passes through link e and goes to node d,
- $n(i,s)$: Equals 1 iff path i is in subgroup s,
- $m(i,j)$: Equals 1 iff path i and path j are in the same subgroup so are coded together,
- $r(i,f)$: Equals 1 iff path i and connection demand f are indirectly related,
- $t_e(s)$: Equals 1 iff one of the paths in subgroup s traverses over link e and goes to node d,
- $\sigma_{f,i}$: Equals 1 iff node f is the source node of demand i.

The objective function is

$$\min \sum_{e\in E}\sum_{s=1}^{2N} t_e(s)\times a_e - \sum_{f\in V} CG_f^{new}\times \pi_f. \tag{13}$$

The constraints are

$$\sum_{f\in V}\sigma_{f,i}\leq 1, \quad 1\leq i\leq ND-1, \tag{14}$$

$$CG_f^{new} = \sum_{i=1}^{ND-1}\sigma_{f,i} \quad \forall f\in V,\ f\neq d, \tag{15}$$

$$\sum_{f\in V}\sum_{i=1}^{ND-1}\sigma_{f,i}\leq ND-1 \tag{16}$$

$$\sum_{e\in\Gamma_i(f)} x_e(j) - \sum_{e\in\Gamma_o(f)} x_e(j) = \begin{cases}-\sigma_{f,i} & \text{if } f\neq d,\\ \sum\sigma_{f,i} & \text{if } f=d,\end{cases}$$
$$j=2i,\ j=2i-1. \tag{17}$$

$$\sum_{s=1}^{2(ND-1)} n(i,s)=1, \quad 1\leq i\leq 2(ND-1), \tag{18}$$

$$n(i,s)+n(i-1,s)\leq 1,$$
$$1\leq i,s\leq 2(ND-1): \quad \bmod (i,2)=0, \tag{19}$$

$$t_e(s)\geq x_e(i)+n(i,s)-1 \quad \forall e,i,s \tag{20}$$

$$t_e(s_1)+t_e(s_2)+t_k(s_1)+t_k(s_2)\leq 1$$
$$\forall e,k\in g,\forall g\in S,\forall s_1,s_2 \tag{21}$$

$$m(i,j)\geq n(i,s)+n(j,s)-1 \quad \forall i\neq j,s. \tag{22}$$

Assuming that $j^*=j-1$ if $\bmod (j,2)=0$ and $j^*=j+1$ otherwise

$$r(i,f)\geq m(i,j)+m(j^*,2f)+m(j^*,2f-1)$$
$$-m(i,2f)-m(i,2f-1)-1 \quad \forall i,j,f,: i\neq j \tag{23}$$

$$r(i,f)\geq r(i,g)+m(2g,2f)+m(2g,2f-1)$$
$$+m(2g-1,2f)+m(2g-1,2f-1)-1 \quad \forall i,f\neq g:$$
$$i\neq 2f, i\neq 2f-1, i\neq 2g, i\neq 2g-1. \tag{24}$$

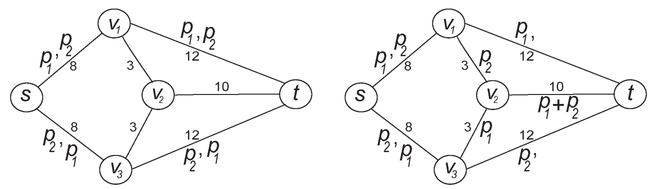
$$r(2f,g)+r(2f-1,g)+m(2f,2g)+m(2f-1,2g)$$
$$+m(2f,2g-1)+m(2f-1,2g-1)\leq 1 \quad \forall g,f: g\neq f, \tag{25}$$

Inequality (14) ensures that each demand has at most one source node. Some connection demands may be empty. Equation (15) calculates the number of connection demands originating from each node at the new coding group. Inequality (16) bounds the total number of connection demands in the new coding group by the nodal degree of the destination node minus 1. Equation (17) carries out the origination, continuation, and termination of the paths of each connection demand. Each connection demand has two paths in a coding group. Equation (18) ensures that each connection demand is a part of a coding subgroup. Inequality (19) ensures that paths belonging to the same connection cannot be a part of the same subgroup. Inequality (20) compiles the topologies of the subgroups by combining the paths of the demands in that subgroup. Inequality (21) satisfies the link-disjointness criterion between the topologies of different subgroups. Inequality (22) says that if two paths are in the same subgroup then they are assumed to be coded together. In inequality (23), path i becomes indirectly related to demand f if there exists a path j that is coded with both path i and one of the paths carrying demand f. Moreover, path i must not be coded with either paths of demand f. In inequality (24), path i becomes indirectly related to demand f if there exists a demand g that is indirectly related to path i, and one of the paths of demand g must be coded together with one of the paths of demand f. Inequality (25) ensures that two different connection demands can be indirectly related only if no pair of their paths are encoded together. Otherwise, a coding cycle occurs which is a violation of the validity of the coding structure.

3) Coherent Diversity Coding: In this section, we introduce a novel coding structure that can mitigate the limiting link-disjointness criterion to the full extent. It is called *Coherent Diversity Coding*. It deconstructs the link-disjointness relationships between multiple paths of different data signals in a coding group. It removes the link-disjointness criteria between paths up to any point where coding mechanism is still feasible. It is most effective when

- There is a single destination node,
- There are two link-disjoint paths for each connection demand,
- The coding operations are within $GF(2)$.

It enables one to achieve more capacity-efficient results than conventional diversity coding. Conventional diversity coding, systematic or nonsystematic, requires two paths to be either coded or to be link-disjoint. This prevents applying conventional diversity coding at the destination nodes with a nodal degree of 2, even though the rest of the network is highly connected. There is room for improvement in the capacity efficiency of coding groups by relaxing the link-disjointness criterion between different paths. Fig. 5 is taken from [15] and shows how the strict link-disjointness criterion for two connections can be relaxed to save capacity. The connection demands are from node s to node t, carrying signals p_1 and p_2, respectively. The numbers next to links are the length (unit cost) of those links. There is no available nontrivial solution for conventional diversity coding on this topology since there are only N, which is 2 in this case, number of link-disjoint paths,



(a) (b)

Fig. 5. Effect of low nodal degree on coding. (a) Diversity coding solution (identical to $1+1$ APS). (b) Network coding-based solution [15].

less than the required $N+1$ $(N+1=3)$, from source to destination. Therefore, in Fig. 5(a), the solution of conventional diversity coding is identical to that of $1+1$ APS with a total cost of 80. The low nodal degree of the source node is a bottleneck for conventional diversity coding. On the other hand, the network-coding based technique proposed by [15] shows that these two data signals can be coded to save cost in Fig. 5(b) by reducing the total cost down to 72. However, the technique in [15] is intractable for more than two connection demands and lacks an efficient capacity placement algorithm. Note that summation is modulo-2 sum in this context. Therefore, we developed an efficient link-disjointness criteria between paths in the same coding group that can mitigate the effects of low nodal degree in the network. Coherent diversity coding enables paths sharing the same link, even if they are not coded together, up to the extent that decodability is preserved. Therefore, it is both flexible and feasible. Under the effectiveness conditions stated above, the necessary and sufficient conditions of decodability are to ensure that at least one copy of each signal is alive and any subset of k signals resides in at least k subgroups after any single link failure. The resulting coding structure will be decodable according to *Lemma 1* in [13]. Therefore, we build the coding structure of coherent diversity coding such that after any single link failure, there will be at least one copy of each signal and any subset of k signals reside in at least k subgroups. We define two paths coherent to each other if the receiver can recover from their simultaneous failure. Therefore, coherent paths can share the same link. Otherwise, the simultaneous failure of noncoherent paths will impair the decodability as will be shown with an example. The terms of coherent and noncoherent paths are coined to keep the track of link-disjointness relationship between paths. The proposed technique is nearly as simple to implement as diversity coding.

The received vector of systematic diversity coding for two connection demands is

$$\begin{bmatrix} p_1 \\ p_2 \\ p_1+p_2 \end{bmatrix} \tag{26}$$

where p_1 and p_2 are the data signals of two different connection demands. Each symbol on the received vector represents a single path and each data signal is carried with two different paths. The paths carrying the same signal are complementary of each other. If two paths have to be link-disjoint, then they are defined as noncoherent to each other. Assume the path carrying p_1 in the first subgroup and the path carrying p_2 in the third subgroup fail simultaneously, then the received vector is

$$\begin{bmatrix} 0 \\ p_2 \\ p_1+0 \end{bmatrix}. \tag{27}$$

The destination node will still be able to decode symbols p_1 and p_2. It is clearly seen that conventional diversity coding can tolerate failure of symbols in more than one subgroup. Therefore, the path carrying p_1 in the first subgroup and path carrying p_2 in the third subgroup can share some of the links. Therefore, they are coherent to each other. Similarly, the path carrying p_2 in the second subgroup and the path carrying p_1 in the third subgroup can share links. After those relaxations, the solution in Fig. 5(b) is achieved with a modified diversity coding approach. This approach is simpler to keep track of since there are at most $2\times N$ paths for N connection demands. Intuitively, in systematic diversity coding, a path can be link-joint with the paths that are combined with its complementary path. However, to implement those relaxations over nonsystematic codes with an arbitrary number of data signals, a general strategy is needed. The set of rules that define the general strategy are

1. A path is link-disjoint (noncoherent) with its complementary path,
2. A path is coherent with the path that is coded with its complementary path,
3. A path is noncoherent with the complementary paths of its coherent paths,
4. A path is coherent with the paths that are coded with its noncoherent paths.

The logic behind these rules is to make sure that at least one path carrying each data signal survives and any subset of k signals are found within at least k subgroups under any single link failure scenario. It is also important to keep the number of nonzero subgroups greater than or equal to N under any failure scenario. The following example visualizes how coherent and noncoherent relationships between paths are found. A valid nonsystematic code is

$$\begin{bmatrix} a+c \\ b \\ a+e \\ c \\ b+d+e \\ d \end{bmatrix} \tag{28}$$

with five connection demands. The procedure to find the set of coherent and noncoherent paths of the path carrying b in the second subgroup is shown in Fig. 6. The decodability of the coherent diversity coding is guaranteed recursively. In the first step, the complementary path of underlined b is set as a noncoherent path in Fig. 6(a) following *Rule 1*. The coherent paths are placed in a circle, whereas noncoherent paths are placed in a square. *Rule 1* ensures that at least one copy of each data signal, signal b in this case, survives a link failure. In Fig. 6(b), paths that are combined with its complementary

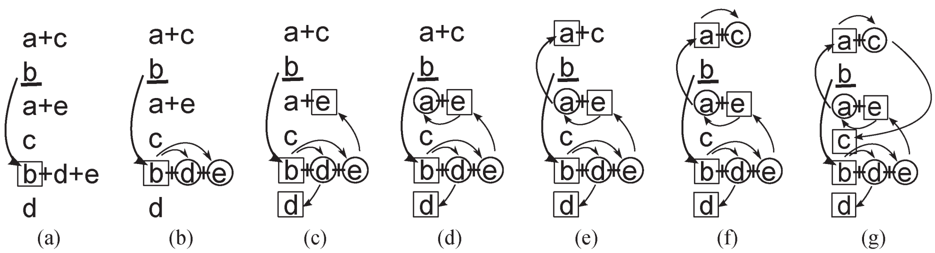


Fig. 6. The process of finding the coherent and noncoherent paths to the underlined path in the second subgroup. Coherent paths are depicted in a circle and noncoherent paths in a square.

path are set as coherent paths following *Rule 2*. If all paths that are coded with the complementary path of signal b fails jointly with the path carrying signal b then signal b on line 5 becomes the only signal on this subgroup and is easily extractable. In the next step, if signal b is removed from the coding structure, the rest of the coding structure includes $N-1$ signals in $N-1$ subgroups, which is valid for decoding according to *Lemma 1* from [13]. In Fig. 6(c) and in Fig. 6(d), the third and fourth rules of the general strategy are applied, respectively. After assuming paths carrying signals d and e failed, *Rule 3* ensures that the other copies of signals d and e survive. Therefore, the third rule is the recursive reformulation of the first rule after removing signal b from the coding group. Following the same logic, *Rule 4* becomes the recursive reformulation of the second rule for the $N-1$ subset of data signals after removing signal b. After this step, signal d and signal e are removed from the coding group. The rest of the coding group consists of $N-3$ signals in $N-3$ subgroups. Recursively, *Rule 3* and *Rule 4* are implemented until the termination scenario is reached. The termination scenario is when there is a single path by itself, which is sufficient for extracting. In this example, the termination scenario solely includes the path on line 4 carrying signal c. At the end, if there is any nonvisited path in the coding group, it is assumed to be coherent. In that case, the rest of the paths are set as coherent paths to the path of interest.

A following example is shown in Fig. 6. Assume that the underlined path carrying signal b fails simultaneously with the path carrying signal a in the first subgroup which is noncoherent to itself. If so, the received vector at the destination node becomes

$$\begin{bmatrix} c \\ 0 \\ a+e \\ c \\ b+d+e \\ d \end{bmatrix}. \tag{29}$$

This vector clearly violates one of the conditions of decodability because the set of four signals $\{a, e, b, d\}$ is bounded within only three subgroups $\{\{a+e\}, \{b+d+e\}, \{d\}\}$. Therefore, the resulting decoding vector is not decodable. The other scenarios can also be checked to confirm that simultaneous failures of noncoherent paths impair the survivability, unlike the simultaneous failures of the coherent paths. If more than two paths are supposed to share the same link then each pair of paths must be coherent to each other. To find the coherent and noncoherent set of paths of each path, this process is repeated starting with the path of interest.

We developed an ILP formulation to generate new coding groups based on the principles of coherent diversity coding. The ILP formulation of this coding structure inherits all of the variables, parameters, objective function, and constraints of Section IV-B2. The extra variables needed for this ILP formulation are

- $\theta(i,j) \in \{0,1\}$: Binary variable, equals 1 iff the path i and path j are noncoherent, in other words, they cannot fail simultaneously.

The objective function to find a new coding group with the most negative reduced cost is

$$\min \sum_{e \in E} \sum_{s=1}^{2N} t_e(s) \times a_e - \sum_{f \in V} CG_f^{new} \times \pi_f. \tag{30}$$

The additional constraints are

$$\theta(i, i-1) = \theta(i-1, i) = 1 \quad \forall i : \mathrm{mod}(i,2) = 0, \tag{31}$$

$$\theta(i,j) \geq m(i^*, j^*) \quad \forall i, j, \tag{32}$$

$$\theta(i,k) \geq \theta(i,j) + m(j,k^*) - 1 \quad \forall i \neq j \neq k. \tag{33}$$

$$x_e(i) + x_e(j) + x_{e^*}(i) + x_{e^*}(j) \leq 2 - \theta(i,j) \quad \forall i, j, e \tag{34}$$

such that link e and link e^* are links of the same span in the opposite directions. Equation (31) makes sure that complementary paths have to be link-disjoint with each other according to *Rule 1*. Inequality (32) ensures both *Rule 2* and *Rule 3* are satisfied. In addition, inequality (33) ensures that both *Rule 3* and *Rule 4* are satisfied. Two paths cannot share a link if they are noncoherent, which is guaranteed by inequality (34).

We note that coherent diversity coding can be interpreted as a form of network coding for protection while protecting connections on networks with Shared Risk Link Groups (SRLGs). Furthermore, this technique can be extended to multiple link failure protection if the set of coherent paths are transformed into SRLGs. For example, in Fig. 5, the link $S - V_1$ can be represented as two parallel links which are in the same SRLG.

V. Complexity Analysis

The column generation method is a very effective technique when used for the implementation of diversity coding since it

TABLE I
COMPLEXITY COMPARISONS OF THE LP FORMULATIONS OF DIFFERENT TECHNIQUES

Technique	Main Problem		Subproblem		No. of C.G.
	No. of integer var.	No. of constraints	No. of integer var.	No. of constraints	
MIP in [8]	$\|N\|\|E\| + \|N\|^2/2$	$3\|N\|\|E\|/2 + \|N\|\|V\| + 7\|N\|/2$	-	-	1
ILP in [10]	$\|N\|^2/2(\|E\|+1) + 3\|N\|\|E\|$	$\|N\|^4/8 + ...$	-	-	1
ILP in [12]	$\|N\|\|E\|(\|V\|+2) + \|N\|(\|N\|+2\|V\|) + ...$	$\|N\|\|V\|(3\|E\| + \|N\| + \|V\|) + ...$	-	-	1
TSA in [19]	J	$\|V\|$	$(ND)\|E\| + (ND)^2/2$	$3(ND)\|E\|/2 + (ND)\|V\| + ...$	J
CGM	$Q << J$	$\|V\|$	$(ND)\|E\| + (ND)^2/2$	$3(ND)\|E\|/2 + (ND)\|V\| + ...$	$Q << J$

decomposes the problem into two iterative steps without loss of LP optimality. The alternative coding-based methods [8], [10], and [12] usually formulate the coding groups placement problem in a single block. Among those, the diversity coding tree algorithm in [8] has significantly fewer variables than the others. On the other hand, in [19], it is shown that implementation of diversity coding with a sequential two-step approach is much simpler than the single-step approaches. The power of the two-step approach is to decompose the bigger main problem into many smaller problems that can be solved in a much shorter time. The complexity of the two-step approach also does not depend on the traffic demand matrix. However, the two-step approach requires a pre-processing phase where every possible coding group is calculated and enumerated before starting the coding groups placement problem. The number of available coding groups depends on

$$J = \binom{|V|-1}{ND-1} + \binom{|V|-1}{ND-2} + \cdots + \binom{|V|-1}{1}, \quad (35)$$

where ND is the nodal degree and $|V|$ is the number of nodes in the network. The number of available coding groups gets exponentially larger as the network size and connectivity increase. It is also costly in terms of memory since it needs to store every possible coding group before starting the coding groups placement problem.

The novelty of the column generation method is to achieve the LP optimal result without explicitly enumerating all of the possible coding groups. After achieving the LP optimal result, the integer solution is found with a negligible optimality gap. It generates the useful coding groups when they are needed. In the coding groups placement problem, only a very small fraction of all the possible coding groups are placed in the final solution. Therefore, the column generation method needs to generate dramatically fewer coding groups than the two-step approach of [19]. Table I highlights the complexity comparison between different optimal techniques based on LP in terms of the number of integer variables and the number of constraints. The number of continuous variables are negligible compared to the number of integer variables in terms of complexity analysis.

TSA is the abbreviation of the two-step approach defined in [19] and CGM is the proposed column generation method. C.G. corresponds to coding groups. Q is the number of iterations column generation takes to find the LP optimal result and $Q << J$. It is not easy to state Q in a closed form in terms of other variables but simulation results in three different networks show that it is much smaller than J and the number of integer variables of other techniques. It also increases with a much smaller ratio. Looking at simulation results, the maximum values of Q are only 19, 79, and 205 in test networks with $|V|$ equal to 14, 28, and 43, respectively. For the same networks, the respective values of J are 377, 20853, and 6220767. It can be seen that the proposed CGM has dramatically fewer variables and constraints in the main problem. When we assume systematic diversity coding is employed, the subproblem of CGM has fewer variables than competitive techniques taking $ND << |N|$ into account. Moreover, in CGM, the complexity of the subproblem and the complexity of the main problem are independent of each other. In the first three techniques, the complexity is exponentially dependent on the number of unit traffic demands and the network size. In TSA, the complexity is dependent on the number of candidate coding groups, which exponentially increases with $|V|$ and ND. As a result, the proposed CGM is much simpler and scalable than competitive techniques. Therefore, it can implement diversity coding over large arbitrary networks. The simplicity of the CGM is also reflected in Section VII in terms of the runtime of different algorithms. In [8], it is explained that adopting single destination diversity coding enables near-hitless recovery and simplifies design complexity significantly. Therefore, in this paper, single destination diversity coding is adopted.

VI. Theoretical Lower Bound

In this section, we explore the theoretical limits of the cost requirements of single destination coding-based recovery techniques. The derived lower bounds will be helpful to understand the room of improvement over our proposed techniques, which already can be implemented on large real networks. Advancements in coding techniques usually result in very high design complexity which prevents them to be optimally applicable on large sized networks, such as [15]. Therefore, we need to find out the extent of incentive to develop more advanced coding techniques in terms of capacity efficiency.

In undirected networks, the well-known Li and Li [25] conjecture states that network coding does not increase the throughput compared to routing for unicast traffic demands. In other words, network information flow problem is equivalent to max-flow multicommodity flow problem in undirected networks. This conjecture with different versions is proven in [26]. By applying LP duality in the form of the Japanese theorem, [26] shows that network coding cannot help reduce the transmission cost in undirected networks for unicast demands. Therefore, the coding-based lower bound we aim to compare is actually the solution of the min-cost multicommodity flow problem for the unicast traffic demands with the same destination node.

The last step to find the lower bound comparable with our solutions is to incorporate the single link failure cases in the min-cost multicommodity flow problems. A mathematical model for survivable multicommodity flow problems for general failure scenarios is developed in [27]. A similar model can be built by making some adjustments. First, we assume only single link (span) failure cases where edges sharing the same link (span) in opposite directions fail simultaneously. Second, we assume the available capacity is infinite and therefore, the objective function is based on optimizing the utilized capacity not based on constructed available capacity. In the next subsection, we present the mathematical LP model used to calculate the lower bound comparable to our proposed techniques.

A. The Mathematical Model for the Lower Bound

The extra set of parameters defined in this model are

- F: The set of failure scenarios,
- r: The failure scenario index,
- G_r: The network graph in failure scenario r,
- $\overline{G}_r$: The set of arcs failed in failure scenario r,
- $\Gamma_{i,G_r}(v)$: The set of incoming links of each node v in graph G_r,
- $\Gamma_{o,G_r}(v)$: The set of outgoing links of each node v in graph G_r.

Since we assume single link failure cases, $\overline{G}_r = \{e_1, e_2| \ e_1 \in r, e_2 \in r, \forall r \in S\}$ and $G_r = G - \overline{G}_r$ where—is the "set minus" operator [27].

The variables needed for the model are

- λ_e: The aggregate utilized capacity on link e for all failure scenarios,
- $\lambda_e(r)$: The utilized capacity on link e under failure scenario r,
- $\lambda_e^{d,f}(r)$: The utilized capacity on link e by demand from source f to common destination d under failure scenario r.

The objective function of the model is

$$\min \sum_{e \in E} \lambda_e \times a_e. \tag{36}$$

The total cost is the sum of the costs of each link which is calculated as a multiplication of the aggregate capacity on that link with the cost (weight) of that link.

The additional constraints are

$$\sum_{e \in \Gamma_{i,G_r}(v)} \lambda_e^{d,f}(r) - \sum_{e \in \Gamma_{o,G_r}(v)} \lambda_e^{d,f}(r) = \begin{cases} -t_f^d & \text{if } v = f, \\ t_f^d & \text{if } v = d, \\ 0 & \text{otherwise,} \end{cases}$$

$$\forall v, \forall f, \forall r, \tag{37}$$

$$\lambda_e(r) = \sum_{f \neq d} \lambda_e^{d,f}(r) \quad \forall r, e, \tag{38}$$

$$\lambda_e \geq \lambda_e(r) \quad \forall r, e. \tag{39}$$

The set of equations (37) are flow conservation equations for all flows under each link failure scenario r. Equation (38) calculates the total utilized capacity on link e in failure scenario r for all flows f, d. In inequality (39), the aggregate utilized capacity of each link is equal to the maximum of the utilized capacity in all failure scenarios. After decomposing traffic based on each destination node, the results of this LP model are given in Section VII as the theoretical lower bound of the single destination recovery techniques and compared with the solutions of the proposed techniques.

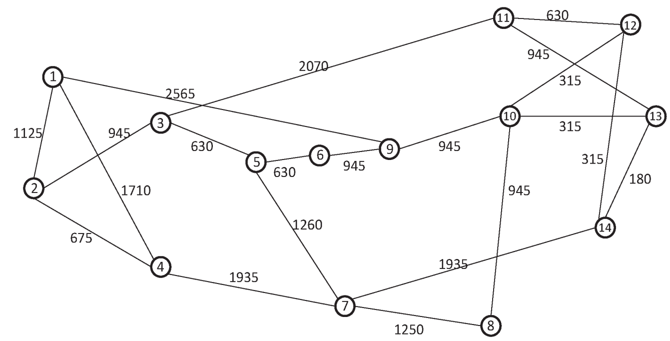


Fig. 7. NSFNET network.

TABLE II
COST AND RUNTIME COMPARISON BETWEEN DIFFERENT TECHNIQUES

Protection Technique	TC	Runtime
Diversity Coding Tree	16410400	$\approx$ 6 hours
TSA-SDC	15788730	$\approx$ 6 minutes
TSA-NSDC	15742000	$\approx$ 9 minutes
CGM-SDC	15793170	$\approx$ 10 seconds
CGM-NSDC	15742000	$\approx$ 5 minutes
CGM-CDC	15674520	$\approx$ 1 hour
P-cycle algorithm	14814350	$\approx$ 3 minutes

VII. Simulation Results

In this section, we present simulation results to investigate the performance of the novel design algorithm and the new coding structure differentially. The first test network is the NSFNET network, which is depicted in Fig. 7. The numbers next to the nodes are the index of those nodes and the numbers next to the edges are the length of those edges. The traffic matrix of the NSFNET network consists of 3000 random unit-sized demands, which are chosen using a realistic gravity-based model [28]: Each node in the NSFNET network represents a U.S. metropolitan area and their population is proportional to the weight of each node in the connection demand selection process. In this network, we simulated TSA from [19], p-cycle protection [22], diversity coding tree from [8], and the proposed CGM. CPLEX 12.2 is used for the simulations. We also adopted different coding structures for TSA and CGM. There are three different tables that present the simulation results of this network. In Table II, the performance metrics are the total cost (capacity) (TC) and the runtime. The first technique in this table is the diversity coding tree algorithm [8]. TSA-SDC refers to the two-step approach implementing systematic diversity coding, whereas TSA-NSDC means TSA for nonsystematic diversity coding. CGM-SDC, CGM-NSDC, and CGM-CDC correspond to the CGM implementing systematic diversity

TABLE III
NSFNET, TC RESULTS FOR EACH DESTINATION NODE

Destination Node	CGM-SDC	CGM-NSDC	CGM-CDC	Theoretical Lower Bound
Node 1	762550	762550	762550	762550
Node 2	1301320	1301320	1301320	1301320
Node 3	211500	211500	211500	211500
Node 4	3913950	3913950	3913950	3913950
Node 5	455100	455100	455100	455100
Node 6	128150	128150	128150	128150
Node 7	1119200	1072725	1072725	1072725
Node 8	602000	602000	595700	595700
Node 9	1595700	1595700	1595700	1585200
Node 10	1177230	1172975	1172970	1172970
Node 11	573650	573650	573650	573650
Node 12	2293720	2293720	2264700	2237150
Node 13	1150350	1150350	1130350	1122700
Node 14	508750	508750	496150	489850
Total	15793170	15742000	15674520	15622515

TABLE IV
THE EFFECT OF GRANULARITY ON NETWORK OPTIMIZATION

Protection Technique	No. of connection demands	Total Cost	LP Bound	Optimality Gap
CGM-SDC	300	1603325	1578407	$\approx$ 1.57%
CGM-SDC	3000	1579331	1578407	$\approx$ 0.058%
CGM-SDC	30000	1579279	1578407	$\approx$ 0.055%

coding, nonsystematic diversity coding, and coherent diversity coding. In our implementation, to reuse previous results to save time, these three algorithms are implemented sequentially. The coding groups (columns) generated by CGM-SDC are inherited by CGM-NSDC. Likewise, CGM-CDC inherits the coding groups generated by CGM-NSDC. The p-cycle algorithm is taken from [22], which is also based on column generation.

Table II presents various trade-offs between protection techniques. First of all, coding-based techniques offer near-hitless recovery. Their restoration speed is at least two orders of magnitude higher than that of p-cycle protection [8]. On the other hand, p-cycle protection has higher capacity efficiency than the tested coding-based methods. As it is seen, the diversity coding tree algorithm has the highest complexity which keeps it from achieving optimal results even though it implements the same systematic diversity coding like TSA-SDC and CGM-SDC do. The proposed CGM is more scalable than the diversity coding tree algorithm and TSA, as seen from the runtime column. In both TSA and CGM, nonsystematic diversity coding is more capacity-efficient than systematic diversity coding. In addition, proposed coherent diversity coding is the most capacity-efficient among coding-based methods. However, the increase in capacity efficiency is negligible compared to the savings in runtime. Network designers can opt to carry out the implementations of CGM-NSDC and CGM-CDC after the implementation of CGM-SDC. We believe that CGM-SDC is the most efficient coding-based technique in terms of restoration speed, capacity efficiency, and design complexity.

In Table III, the performance of the nonsystematic and coherent diversity coding is presented compared to the systematic diversity coding and the theoretical lower bound with a breakdown over the nodes. The performance metric is the TC to route and protect the connection demands. The goal is to measure the decrease in TC due to the introduction of nonsystematic and coherent diversity coding and see how much room is left for improvement. As expected, nonsystematic diversity coding performs better than systematic diversity coding, whereas coherent diversity coding performs best of all. As mentioned before, the improvement due to the introduction of advanced coding techniques is limited over all different destination node scenarios. It is because coherent diversity coding almost achieves the theoretical lower bound for single destination-based recovery techniques. The difference between total capacity required by coherent diversity coding and the theoretical lower bound varies between 0–1.28% depending on the destination node. On the average, coherent diversity coding requires only 0.33% extra capacity than the theoretical lower bound, which means our solution is almost optimal for any single destination-based solution.

In Table IV, the effect of the traffic granularity is investigated over the total cost, the LP lower bound of the integer solution and the optimality gap between the ILP solution and the LP lower bound. We input three different traffic scenarios. In the first scenario, there are 300 unit connection demands created by the gravity-based model. In the second scenario, each traffic demand is divided into 10 smaller unit connection demands creating 3000 connection demands. In the final scenario, 30000 connection demands are created by performing the same operation again. It is seen from the results that the optimality gap decreases as the granularity of the connection demands decreases. Optimality gap converges to zero fast. The fact that column generation is implemented over LP increases the simulation speed significantly but does not deteriorate the performance.

The second test network is the U.S. long-distance network, taken from [29], which is depicted in Fig. 8. The traffic matrix is created using a gravity-based model [28]. In total, there are 23,204 static unit connection demands. This setup is chosen to

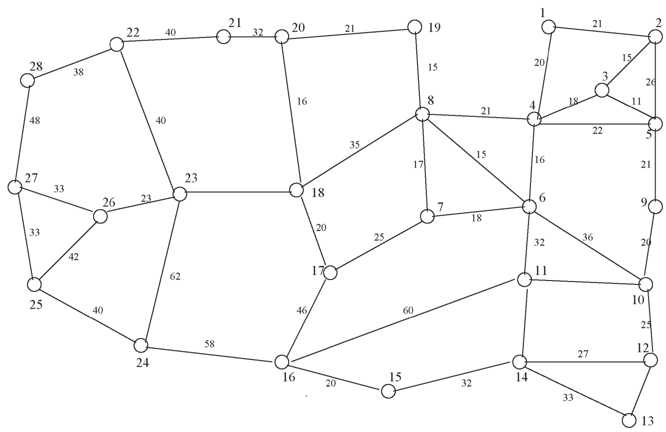


Fig. 8. U.S. long-distance network.

TABLE V
COMPARATIVE PERFORMANCE OF THE NEW ALGORITHMS IN U.S. LONG-DISTANCE NETWORK

Protection Technique	SCaP	Runtime	No. of Coding Groups
TSA-SDC	105.6%	$\approx$ 3 hours	31464
CGM-SDC	105.6%	$\approx$ 2 minutes	61
CGM-NSDC	105.5%	$\approx$ 2 hours	72
CGM-CDC	102.4%	$\approx$ 9 hours	79
P-cycle algorithm	107.0%	$\approx$ 2.5 hours	32 (*p*-cycles)
1+1 APS	134.6%	$\approx$ 50 seconds	-

observe the performance of the new design algorithm in a large realistic network with a dense traffic scenario. We compare the performance of CGM with TSA, the p-cycle algorithm from [22], and $1 + 1$ APS in terms of spare capacity percentage (SCaP). SCaP is defined in [7] as

$$SCaP = \frac{Total\ Capacity - Shortest\ Working\ Capacity}{Shortest\ Working\ Capacity} \times 100\%.$$

Shortest Working Capacity is the total cost when there is no protection and the traffic is routed over the shortest paths. The other coding-based recovery design algorithms are too complex to implement in this setup. The results are presented in Table V. The results cannot be elaborated in terms of the common destination nodes since the number of destination nodes are too large for this network.

As seen from the results, the proposed design algorithm can achieve near-optimal results with different versions of diversity coding even in a large realistic network with a dense traffic scenario. Proposed coherent diversity coding technique performs best compared to other coding-based recovery techniques at the expense of higher complexity. The increase in capacity efficiency due to the advanced coding technique is more significant than it is in the NSFNET network. The implementation of systematic diversity coding with the proposed CGM is highly scalable since its runtime does not increase as much as others when the network size gets bigger. The TSA approach is not as scalable as CGM since the number of candidate coding groups in TSA increases exponentially with the nodal degree and the number of nodes, whereas the number of candidate groups in CGM increases with a much smaller ratio of the number of nodes. The SCaP result of the new technique is better than that of the column generation based p-cycle algorithm and $1 + 1$ APS algorithm. It should be noted that, p-cycle algorithm carries out Spare Capacity Placement (SCP) [3] due to its high complexity, whereas the proposed algorithm carries out Joint Capacity Placement (JCP) [3]. Even with that adjustment, the proposed CGM is simpler than the p-cycle algorithm.

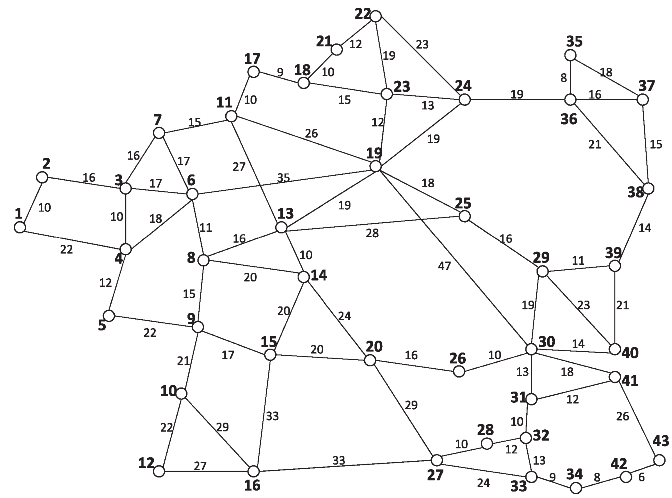


Fig. 9. Long-distance network of France.

The third network is the long-distance network of France with 43 nodes and 142 unidirectional links taken from [30]. It is depicted in Fig. 9. There are a total number of 4518318 unit connection demands. The traffic scenario is created following the same gravity-based model. The reason to select this network is to test the performance of CGM in very large realistic networks. Therefore, we only simulate CGM-SDC to investigate the runtime performance of the column generation method without extra complexity due to the advanced coding structure. It is compared to $1 + 1$ APS. We also break down the results in terms of the nodal degree of the nodes to see the effect of the nodal degree on both capacity efficiency and runtime. The number of destination nodes is too large to present the results for each destination node individually. The results are presented in Table VI. The runtime of $1 + 1$ APS is equal to 1 minute.

CGM can achieve the near-optimal result in such a large network with over four million unit demands. The capacity efficiency of CGM-SDC improves as the nodal degree increases with the exception of nodal degree being equal to 5. It may be seen as an exception due to the small sample size. According to Table VI, there is a trade-off between the runtime and the capacity improvement over $1 + 1$ APS. When the nodal degree increases, the SCaP improvement of CGM-SDC over $1 + 1$ APS increases at the expense of increased runtime of CGM-SDC with some exceptions due to the small sample size. When the nodal degree is equal to 2, systematic diversity coding acts the same as $1 + 1$ APS as we mentioned before.

VIII. CONCLUSION

In this paper, we introduced an advanced version of diversity coding and a near-optimal and simple design algorithm

TABLE VI
SCaP PERFORMANCE OF THE NEW ALGORITHM WITH RESPECT TO THE NODAL DEGREE

Nodal Degree	CGM-SDC (SCaP)	1+1 APS (SCaP)	Runtime of CGM-SDC	Sample Size
2 links	155.3%	155.3%	≈ 2 minutes	12 nodes
3 links	125.5%	149.4%	≈ 16 minutes	13 nodes
4 links	106.7%	140.6%	≈ 39 minutes	14 nodes
5 links	146.4%	184.5%	≈ 26 minutes	2 nodes
6 links	89.5%	126.5%	≈ 85 minutes	1 node
7 links	86.6%	136.6	≈ 53 minutes	1 node
Total	105.7%	141.0%	≈ 85 minutes	43 nodes

to achieve near instantaneous recovery with higher capacity efficiency. The proposed coherent diversity coding method employs nonsystematic coding, which enables all paths to be encoded, and relaxes the link-disjointness criterion between paths to cope with the low nodal degree in the network. The code is developed with the objective of minimum utilized capacity. The design algorithm consists of two parts, namely a main problem and a subproblem. These two advanced techniques combined achieve results with higher capacity efficiency in a much shorter amount of time in relatively large networks. The advantages of both techniques are shown with examples and simulation results.

The new design framework is based on the column-generation method and consists of two parts, a main problem where the traffic demands are met with the available coding groups and a subproblem where new useful coding groups are generated at each iteration. The main problem starts with a set of dummy coding groups and inputs new coding groups at each iteration. The subproblem creates a new coding group depending on the information coming from the main problem. The iterations are terminated when a new useful coding group cannot be found. The main problem is formulated as LP throughout the iteration process. At the end, the main problem is solved via ILP which creates a very small optimality gap. We have formulated the subproblem differently for different coding techniques based on either ILP or MIP. There is a complexity versus capacity efficiency tradeoff in formulating the subproblem. The main problem consists of only $|V| - 1$ constraints. It finds and places the optimal coding group combinations to match the traffic demands, which takes sub-ms to run. The new algorithm can be implemented over networks with arbitrary topology and it can achieve near-optimal integer results in very large arbitrary networks for arbitrary traffic scenarios.

We ran various sets of simulations to investigate the performance of the new coding structure and the new design algorithm differentially. Coherent diversity coding has a higher capacity efficiency than both nonsystematic and systematic diversity coding. It produces results within 0.33% of any single destination-based solution in NSFNET network. The improvement is very small in some networks but is more significant in other networks. The most important observation of the paper is how the new column generation-based design method simplifies implementation of coding-based recovery techniques in very large arbitrary networks. The new technique can find solutions with negligible optimality gap in a much shorter time than the competitive techniques. The complexity of the new technique is more scalable than the competitive techniques depending on the network size, the size of the traffic demands, and the nodal degree of the nodes in the network.

REFERENCES

[1] S. N. Avci and E. Ayanoglu, "Coded path protection: Efficient conversion of sharing to coding," *IEEE Trans. Commun.*, vol. 61, no. 10, pp. 4294–4309, Oct. 2013.

[2] M. Menth, M. Duelli, and J. Milbrandt, "Resilience analysis of packet-switched communication networks," *IEEE/ACM Trans. Netw.*, vol. 17, no. 6, pp. 1950–1963, Dec. 2009.

[3] W. D. Grover, *Mesh-Based Survivable Networks: Options and Strategies for Optical, MPLS, SONET, and ATM Networking*. Upper Saddle River, NJ, USA: Prentice-Hall, 2004.

[4] I. B. Barla, F. Rambach, D. A. Schupke, and G. Carle, "Efficient protection in single-domain networks using network coding," in *Proc. IEEE GLOBECOM*, Dec. 2010, pp. 1–5.

[5] E. Ayanoglu, C.-L. I, R. D. Gitlin, and J. E. Mazo, "Diversity coding: Using error control for self-healing in communication networks," in *Proc. IEEE INFOCOM*, Jun. 1990, vol. 1, pp. 95–104.

[6] R. Ahlswede, N. Cai, S.-Y. R. Li, and R. W. Yeung, "Network information flow," *IEEE Trans. Inf. Theory*, vol. 46, pp. 1204–1216, Jul. 2000.

[7] S. N. Avci, X. Hu, and E. Ayanoglu, "Recovery from link failures in networks with arbitrary topology via diversity coding," in *Proc. IEEE GLOBECOM*, Dec. 2011, pp. 1–6.

[8] S. N. Avci and E. Ayanoglu, "Optimal algorithms for near-hitless network restoration via diversity coding," *IEEE Trans. Commun.*, vol. 61, no. 9, pp. 3878–3893, Sep. 2013.

[9] S. Ramamurthy, L. Sahasrabuddhe, and B. Mukherjee, "Survivable WDM mesh networks," *J. Lightw. Technol.*, vol. 21, no. 4, pp. 870–883, Apr. 2003.

[10] A. E. Kamal and O. M. Al-Kofahi, "Efficient and agile $1 + \mathrm{N}$ protection," *IEEE Trans. Commun.*, vol. 59, no. 1, pp. 169–180, Jan. 2011.

[11] A. E. Kamal, A. Ramamoorthy, L. Long, and S. Li, "Overlay protection against link failures using network coding," *IEEE/ACM Trans. Netw.*, vol. 19, no. 4, pp. 1071–1084, Aug. 2011.

[12] H. Øverby, G. Bóizck, P. Barbarczi, and J. Tapolcai, "Cost comparison of $1 + 1$ path protection schemes: A case for coding," in *Proc. IEEE ICC*, Jun. 2012, pp. 3067–3072.

[13] O. M. Al-Kofahi and A. E. Kamal, "Network coding-based protection of many-to-one wireless flows," *IEEE J. Sel. Areas Commun.*, vol. 27, no. 5, pp. 787–813, Jun. 2011.

[14] S. N. Avci and E. Ayanoglu, "Extended diversity coding: Coding protection and primary paths for network restoration," in *Proc. Int. Symp. Netw. Coding*, Jun. 2012, pp. 1–6.

[15] S. E. Rouayheb, A. Sprinston, and C. Georghiades, "Robust network codes for unicast connections: A case study," *IEEE/ACM Trans. Netw.*, vol. 19, no. 3, pp. 644–656, Jun. 2011.

[16] A. Sprinston, S. E. Rouayheb, and C. Georghiades, "Robust network coding for bidirected networks," in *Proc. Inf. Theory Appl. Workshop*, Jan./Feb. 2007, pp. 378–383.

[17] P. Barbarczi, J. Tapolcai, P.-H. Ho, and M. Médard, "Optimal dedicated protection approach to shared risk link group failures using network coding," in *Proc. IEEE ICC*, Jun. 2012, pp. 1–5.

[18] M. Mohandespour and A. E. Kamal, "$1 + N$ protection in polynomial time: A heuristic approach," in *Proc. IEEE GLOBECOM*, Dec. 2010, pp. 1–5.

[19] S. N. Avci and E. Ayanoglu, "Network coding-based link failure recovery over large arbitrary networks," in *Proc. IEEE GLOBECOM*, Dec. 2013, pp. 1–7.

[20] A. R. Ravindran, *Operations Research Methodologies*. Boca Raton, FL, USA: CRC Press, 2008.

[21] G. Desaulniers, J. Desrosiers, and M. Solomon, *Column Generation*. Berlin, Germany: Springer-Verlag, 2005.

[22] T. Stidsen and T. Thomadsen, "Joint routing and protection using p-Cycles," Inf. Math. Model., Tech. Univ. Denmark, Lyngby, Denmark, Tech. Rep., May 2005.

[23] C. Rocha and B. Jaumard, "Revisiting p-cycles/FIPP p-cycles vs. shared link/path protection," in *Proc. 17th ICCCN*, St. Thomas, US Virgin Island, Aug. 2008, pp. 1–6.

[24] B. Wu, K. L. Yeung, and P.-H. Ho, "ILP formulations for p-cycle design without candidate cycle enumeration," *IEEE/ACM Trans. Netw.*, vol. 18, no. 1, pp. 284–295, Feb. 2010.

[25] Z. Li and B. Li, "Network coding: The case of multiple unicast sessions," in *Proc. Allerton Conf. Commun.*, Urbana, IL, USA, 2004, pp. 1–9.

[26] T. Xiahou, Z. Li, C. Wu, and J. Huang, "A geometric perspective to multiple-unicast network coding," *IEEE Trans. Inf. Theory*, vol. 60, no. 5, pp. 2884–2895, May 2014.

[27] M. Garg and J. C. Smith, "Models and algorithms for the design of survivable multicommodity flow networks with general failure scenarios," *Omega*, vol. 36, no. 6, pp. 1057–1071, Dec. 2008.

[28] Y. Zhang, M. Roughan, N. Duffield, and A. Greenberg, "Fast accurate computation of large-scale IP traffic matrices from link loads," in *Proc. ACM SIGMETRICS*, Jun. 2003, pp. 206–217.

[29] Y. Xiong and L. G. Mason, "Restoration strategies and spare capacity requirements in self-healing ATM networks," *IEEE/ACM Trans. Netw.*, vol. 7, no. 1, pp. 98–110, Feb. 1999.

[30] J. Doucette, D. He, W. D. Grover, and O. Yang, "Algorithmic approaches for efficient enumeration of candidate p-cycles and capacitated p-cycle network design," in *Proc. DRCN*, Banff, AB, Canada, Oct. 2003, pp. 212–220.

Serhat Nazim Avci received the B.S. degree in electrical and electronics engineering from Bilkent University, Ankara, Turkey, in June 2009, and the M.S. and Ph.D. degrees from the University of California, Irvine in December 2011 and December 2013, respectively. In 2009, he was accepted into the Department of Electrical Engineering and Computer Science of the University of California, Irvine with a departmental fellowship. In March 2014, he started to work as a Research Intern in Futurewei Technologies, Santa Clara, CA, where he has been since August 2014, as a Research Engineer. His research interests are in development of new data center networking technologies, content-based traffic engineering, applications of coding theory into problems in networking, and variations of linear programming, especially the generation of fast algorithms for linear programming applications.

Ender Ayanoglu (S'82–M'85–SM'90–F'98) received the M.S. and Ph.D. degrees from Stanford University, Stanford, CA in 1982 and 1986, respectively, in electrical engineering. He was with the Communications Systems Research Laboratory, part of AT&T Bell Laboratories, Holmdel, NJ until 1996, and Bell Labs, Lucent Technologies until 1999. From 1999 until 2002, he was a Systems Architect at Cisco Systems, Inc., San Jose, CA. Since 2002, he has been a Professor in the Department of Electrical Engineering and Computer Science, University of California, Irvine, Irvine, CA, where he served as the Director of the Center for Pervasive Communications and Computing and held the Conexant-Broadcom Endowed Chair during 2002–2010.

His past accomplishments include invention of the 56K modems, characterization of wavelength conversion gain in wavelength division multiplexed (WDM) systems, and diversity coding, a technique for link failure recovery in communication networks employing erasure coding in 1990, prior to the publication of the first papers on network coding. During 2000–2001, he served as the founding chair of the IEEE-ISTO Broadband Wireless Internet Forum (BWIF), an industry standards organization which developed and built a broadband wireless system employing Orthogonal Frequency Division Multiplexing (OFDM) and a Medium Access Control (MAC) algorithm that provides Quality-of-Service (QoS) guarantees. This system is the precursor of today's Fourth Generation (4G) cellular wireless systems such as WiMAX, LTE, and LTE-Advanced.

From 1993 until 2014, Dr. Ayanoglu was an Editor, and since January 2014 a Senior Editor, of the IEEE TRANSACTIONS ON COMMUNICATIONS. He served as the Editor-in-Chief of the IEEE TRANSACTIONS ON COMMUNICATIONS from 2004 to 2008. Currently, he is serving as the Editor-in-Chief of the IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS—SERIES ON GREEN COMMUNICATIONS AND NETWORKING. From 1990 to 2002, he served on the Executive Committee of the IEEE Communications Society Communication Theory Committee, and from 1999 to 2001, was its Chair. Dr. Ayanoglu is the recipient of the IEEE Communications Society Stephen O. Rice Prize Paper Award in 1995 and the IEEE Communications Society Best Tutorial Paper Award in 1997.