

UC Irvine

UC Irvine Previously Published Works

Title

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

Permalink

<https://escholarship.org/uc/item/2dt9k6gg>

Journal

IEEE Transactions on Communications, 63(5)

ISSN

0090-6778

Authors

Avci, Serhat Nazim

Ayanoglu, Ender

Publication Date

2015

DOI

10.1109/tcomm.2015.2417567

Peer reviewed

Link Failure Recovery over Very Large Arbitrary Networks: The Case of Coding

Serhat Nazim Avci and Ender Ayanoglu

Center for Pervasive Communications and Computing
Department of Electrical Engineering and Computer Science
University of California, Irvine
Irvine, CA 92697-2625

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 arbitrary networks. However, its capacity efficiency is restricted by its systematic coding and high design complexity even though its design complexity is lower than the other coding-based recovery techniques. Alternative techniques mitigate some of these limitations, but they are difficult to implement over arbitrary networks. 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 arbitrary networks. The design framework consists of two parts: a main problem and subproblem. Main problem is realized with Linear Programming (LP) and Integer Linear Programming (ILP), whereas the subproblem can be realized with different methods. The simulation results suggest that both the novel coding structure and the novel design algorithm lead to higher capacity efficiency for near-hitless recovery. The novel design algorithm simplifies the capacity placement problem which enables implementing diversity coding-based techniques on very large arbitrary networks.

I. INTRODUCTION

The information carried by wide area networks is, in general, very important. Yet these networks regularly undergo failures. Detailed statistics about the network failures can be found in [1]. This paper focuses on recovery from single link failures since they consist of 70% of all network failures. To minimize the cost of such failures, various restoration and protection techniques are developed. The two main metrics in the design of these techniques are restoration speed and capacity efficiency. Capacity efficiency is measured by the total required capacity, in terms of fiber miles, and restoration speed is measured by the duration between the occurrence of failure and restoration of failed traffic. The goal is to minimize both of these metrics and every technique offers a different tradeoff.

In some recovery techniques, spare resources are shared among different traffic failure scenarios and different connection demands, whereas in others, spare resources are dedicated to connection demands. Dedicated protection techniques are able to offer near-hitless recovery since they do not require the signaling and rerouting of the failed traffic. 1+1 Automatic Protection Switching (APS) is a dedicated protection technique where two link-disjoint paths for each connection demand are

employed to transmit the same data to the destination node. In the case of a link failure over the primary path, the destination node switches to the protection path and restores the traffic nearly instantaneously. However, 1+1 APS requires more than 100% capacity which makes it capacity inefficient. The fact that 1+1 APS is currently employed in today's networks [2] indicates the need for nearly instantaneous link failure recovery despite its low capacity efficiency.

The capacity efficiency of dedicated protection schemes such as 1+1 APS can be improved if the dedicated paths are shared. This can be achieved by employing coding, in particular, erasure coding [3], [4]. The technique introduced in [3], [4], called diversity coding, has two advantages. First, unlike 1+1 APS, it is capacity efficient. Second, unlike rerouting-based restoration schemes, the recovery is nearly instantaneous. References [3], [4] predate network coding, usually considered to be introduced in [5].

In [6], diversity coding is implemented over arbitrary networks using a heuristic algorithm. In [1], optimal algorithms for the diversity coding technique are developed. Diversity coding performs near-hitless recovery while offering competitive capacity efficiency. In [7], a solution of Shared Path Protection (SPP) [8] is converted to a coding-based solution named Coded Path Protection (CPP). Sharing of the spare resources is replaced with the employment of these resources to code different paths. This conversion increases the restoration speed and the transmission integrity, and decreases error signaling complexity. The bidirectional nature of CPP allows encoding and decoding inside the network for unicast demands.

In [9] and [10], network coding-based protection schemes called 1+N protection are proposed in which coding operations are carried out over trees and trails, respectively. The idea is similar to that of diversity coding except the protection is bidirectional. In [11], the cost efficiencies of a network coding-based recovery technique and a simpler version of diversity coding technique are evaluated.

All of the above mentioned techniques implement systematic coding where primary paths are exempt from coding operations. Also, in these techniques, coding operations are bound to specific topologies. In addition, they require strict link-disjointness between each primary path and the protection paths. Even though these assumptions make those techniques easier to implement, they have restricted capacity efficiencies.

In [2], an argument that 1+N coding requires high nodal degree, which reduces its efficiency on sparse topologies was made.

In [12], the primary paths are incorporated into coding operations using a heuristic algorithm for static provisioning. The decodability of the coding structures is preserved by randomly adding the connection demands to the existing coding groups one by one. A coding group is a set of connection demands that are coded and protected together. Coding primary paths increases capacity efficiency over conventional diversity coding, as in [12]. Nonsystematic coding is implemented in wireless mesh networks for single link failure recovery in [13]. In [14], 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 is restricted to specific topologies. In addition, it can protect at most two connection demands simultaneously. In [15], the proposed technique lifts the restriction over the number of protected connection demands for bidirected networks. In general, the coding-based recovery techniques in the literature, such as [9], [14], [15], offer promises in terms of capacity efficiency and restoration time. However, they cannot be optimally implemented on real networks due to their high design complexity limitations. The test networks and traffic matrices in those papers are much smaller than the real networks, such as the long-distance networks of the U.S. and France, to be discussed in sequel. In [16], a novel two step approach is presented to cope with high design complexity on 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. It manages to overcome 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 offers two novel contributions to the field of diversity coding-based (or network coding-based) link failure recovery. First, we introduce an optimal, simple, and modular design algorithm that provisions the static traffic in very large arbitrary networks. The design algorithm uses 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 coherent coding structure 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 to form coding groups with higher flexibility and bigger size. The decodability is preserved while the high nodal degree requirement is mitigated. Coherent diversity coding incorporates also nonsystematic coding.

The performance of the new proposed coding technique and the column generation-based design algorithm are investi-

gated compared to conventional (systematic or nonsystematic) diversity coding and p -cycle protection [17]. The simplicity of the new design algorithm is also tested based on a set of simulations over relatively large the long-distance networks of the U.S. and France.

II. 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. 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 problem, column generation technique results in significant time and memory savings, and therefore it enables the optimal implementation of efficient network coding-based techniques over very large realistic networks.

Column generation has been used for different LP problems, including the well-known cutting stock problem [18]. The problem is to satisfy different widths of paper demand 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 can also be applied to the diversity coding implementation.

The column generation technique is also applied to the p -cycle protection [19] and SPP [20] techniques 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 set of basic coding group. The main problem in this step is an LP formulation that finds the optimal coding group combinations to meet the traffic demands. After the first run, it passes the dual variables of the solution to the subproblem. The subproblem, which is also called the *Coding Group Generation Problem*, attempts to find a new useful coding group. A new

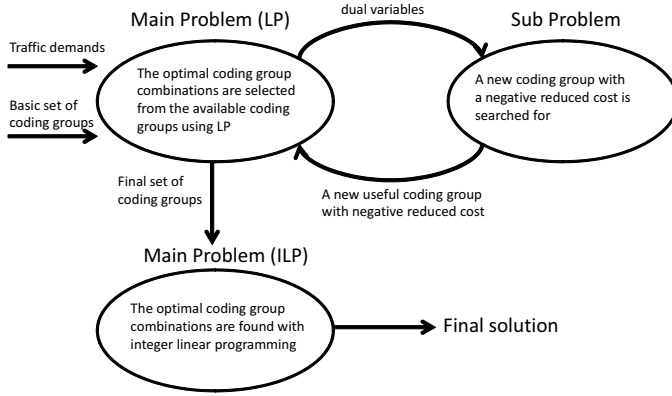


Fig. 1. Steps in the column generation method.

useful coding group has a negative reduced cost 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 VI.

A. Example 1

As an example, assume there are 2 connection demands from S_1 and S_2 to D . Each has a unit traffic demand. The cost of coding groups that protect only $S_1 - D$ and only $S_2 - D$ are 10 and 7, respectively. The coding groups combination problem employs one of each coding group to satisfy the traffic demands at a total cost of 17. The dual variables of this solution are input to the subproblem. The subproblem returns a new coding group consisting of both $S_1 - D$ and $S_2 - D$ at a total cost of 15. The main problem is run one more time and decreases the total cost from 17 to 15 since it employs only the new coding group created by the subproblem. The optimal result is achieved since the subproblem cannot create any more coding groups with negative reduced costs.

III. ILP FORMULATIONS

In this section, we present the algorithms that realize the main problem and the subproblem. 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. 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 replaced 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. In this section, we present three different coding group generation algorithms using mixed integer programming (MIP) or ILP formulations.

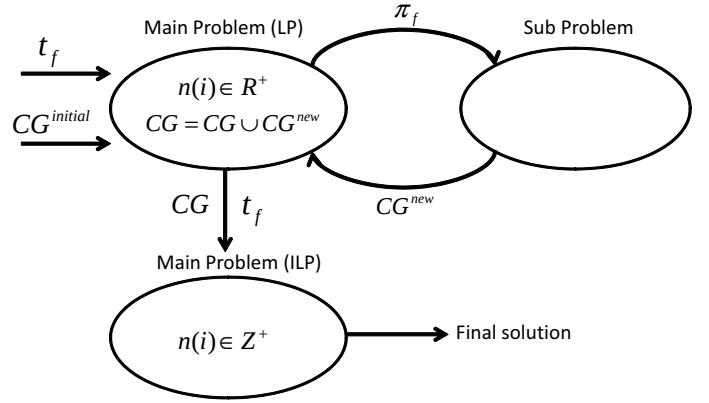


Fig. 2. Steps in the column generation method in terms of LP and ILP variables.

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 a coding group with minimum total cost while meeting the traffic demands. The input parameters of the LP are

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

The variables related to the coding groups placement problem can be either continuous or integer

- $n(i)$: Keeps the number instances of coding group i placed on the network, normally continuous.

The variables $n(i)$ are converted to integer variables at the final step of column generation.

The objective function is

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

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

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

The diagram of the column generation method in terms of the parameters and variables of the LP formulations is shown in Fig. 2, where π_f are the dual variables of each constraint. The traffic demand parameters t_f and an initial basic coding group set $CG^{initial}$ 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.

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 at least one. We present three different coding group generation algorithms, each implementing a different version of diversity coding. These versions have a 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 [21]. 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 achieve the destination node. An example is taken from [21] and is shown in Fig. 3(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

- $G(V, E)$: Network graph,
- 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 ,
- $\Gamma_i(f)$: The set of incoming links of each node f ,
- $\Gamma_o(f)$: The set of outgoing links of each node f ,
- 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, $\beta \geq 2 \times \max(|V|, \max_i(ND_i))$,
- π_f : The values of the dual variables of the main problem.

The set of variables of this MIP formulation are

- CG_f^{new} : Integer variable, equals to the number of connections originating from node f in the new coding group,
- $d_e \in \{0-1\}$: Integer variable, equals 1 iff the primary tree of the new coding group passes through link e ,
- $c_e \in \{0-1\}$: Integer variable, equals 1 iff the protection tree of the new coding group passes through link e ,
- p_f : A continuous variable between 0 and 1. It keeps the “voltage” value of node f 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

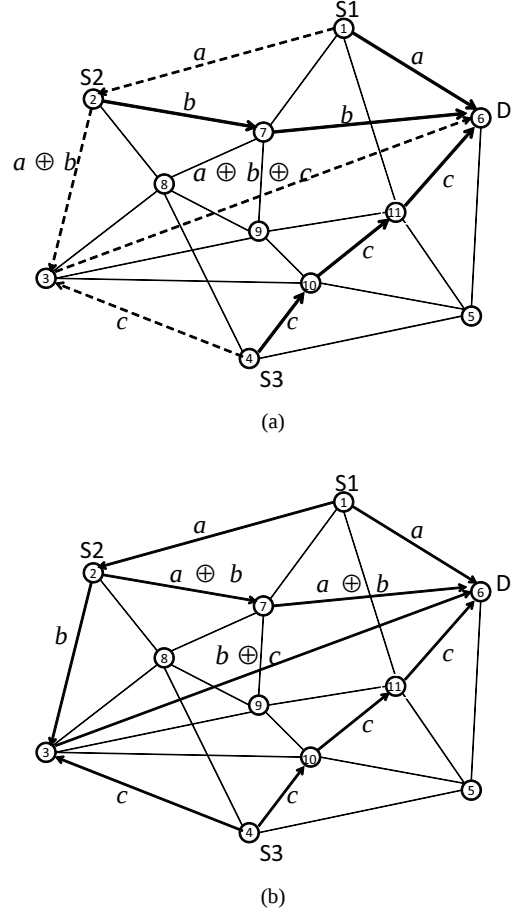


Fig. 3. (a) An example of the systematic diversity coding tree structure. There are three link-disjoint primary paths spanned by the primary tree and there is a link-disjoint protection tree, (b) An example of nonsystematic diversity coding structure for the same set of connections.

coding group

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

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

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

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

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

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

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

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

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

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

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

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 equation (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 equation (7) are responsible for the termination of the protection trees. 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. We refer to *Lemma 1* from [13] while building valid nonsystematic diversity coding. This coding structure increases the capacity efficiency of systematic diversity coding with extra design complexity. An example is shown in Fig. 3(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 [21] with small changes. Reference [21] presents how to optimally build nonsystematic diversity coding structures. The algorithm in [21] looks for every possible coding scenario by eliminating the invalid cases that can be identified as *coding cycles*. 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 ,
- $n(i, s)$: 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 ,
- $\sigma_{f,i}$: Equals 1 iff node f is the source node of demand i .

The objective function is

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

$$\sum_{f \in V} \sigma_{f,i} \leq 1, \quad 1 \leq i \leq ND - 1 : \text{mod}(i, 2) = 0, \quad (14)$$

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

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

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

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

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

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

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

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

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

such that $j^* = j - 1$ if $\text{mod}(j, 2) = 0$ and $j^* = j + 1$ otherwise.

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

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

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 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 either be indirectly related or one 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 optimal extent. It is called *Coherent Diversity Coding*. This coding structure is optimal under the conditions listed as

- 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 typical diversity coding. Typical diversity coding, systematic or nonsystematic, requires two paths to be either coded or to be link-disjoint. For example, that prevents applying typical diversity coding at the destination nodes with a nodal degree of 2, even though the rest of the network is highly connected. There is a space for improvement in the capacity efficiency of coding groups by relaxing the link-disjointness criterion between different paths. Fig. 4 is taken from [14] and shows how the strict link-disjointness criterion for two connections can be relaxed in order to save capacity. The connection demands are from node s to node t , carrying signals p_1 and p_2 , respectively. There is no available nontrivial solution for diversity coding on this topology since there are only N , which is 2 in this case, number of link-disjoint paths, less than the required $N + 1$ ($N + 1 = 3$), from source to destination. Therefore, in Fig. 4(a), the solution of diversity coding is identical to that of 1+1 APS. The low nodal degree of the source node is a bottleneck for diversity coding. On the other hand, the network-coding based technique proposed by [14] shows that these two data signals can be coded to save capacity in Fig. 4(b). However, the technique in [14] is nontractable for more than two connection demands and lacks an efficient capacity placement algorithm.

Therefore, we developed the optimal link-disjointness criteria between paths in the same coding group that can mitigate the effects of low nodal degree in the network. The 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 optimal and feasible. Under the optimality 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. The terms of coherent and noncoherent paths are coined to keep the track of link-disjointness relationship between paths. If two paths are coherent to each other, then they can fail simultaneously,

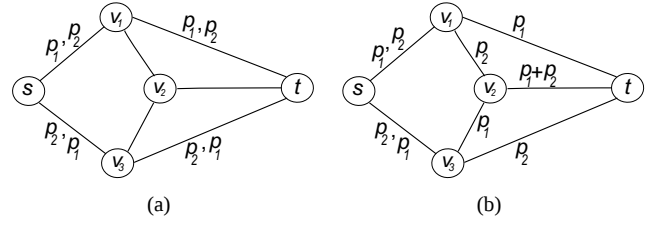


Fig. 4. Effect of low nodal degree on coding (a) Diversity coding solution (identical to 1+1 APS), (b) Network coding-based solution [14].

therefore they can share the same links. Otherwise, their simultaneous failure will impair the decodability as will be shown with an example. The proposed technique is nearly as simple to implement as diversity coding.

The received vector of systematic diversity coding for two connection demands looks like

$$\begin{bmatrix} p_1 \\ p_2 \\ p_1 + p_2 \end{bmatrix} \quad (26)$$

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 will look like

$$\begin{bmatrix} 0 \\ p_2 \\ p_1 + 0 \end{bmatrix}. \quad (27)$$

The destination node will still be able to decode symbols p_1 and p_2 . It is clearly seen that 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 be link-joint. After those relaxations, the solution in Fig. 4(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 has to be 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 complimentary paths of its coherent paths,
4. A path is coherent with the paths that are coded with its noncoherent paths.

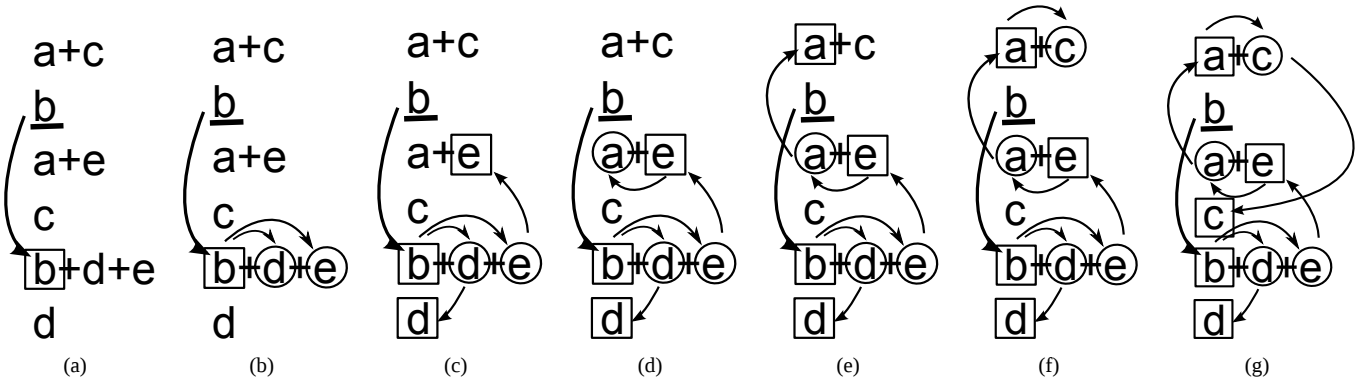


Fig. 5. The process of finding the coherent and noncoherent paths to the underlined path in the second subgroup. Coherent paths are put in a circle and noncoherent paths are put in a square.

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 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

$$\begin{bmatrix} a+c \\ b \\ a+e \\ c \\ b+d+e \\ d \end{bmatrix} \quad (28)$$

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. 5. In the first step, the complementary path of underlined b is set as a noncoherent path in Fig. 5(a) following *rule 1*. The coherent paths are put in a circle, whereas noncoherent paths are put in a square. In Fig. 5(b), paths that are combined with a noncoherent path are set as coherent paths following *rule 2*. In Fig. 5(c) and in Fig. 5(d), the third and fourth rules of the general strategy are applied, respectively. The process is carried out by following *rule 3* and *rule 4* interchangeably until those rules are no longer applicable. 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.

For example, in Fig. 5, 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

$$\begin{bmatrix} c \\ a+e \\ c \\ b+d+e \\ d \end{bmatrix}. \quad (29)$$

This vector clearly violates one of the conditions of decodability because the set of four signals $\{a, e, b, d\}$ are 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 from the path of interest.

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

- $\phi(i, j) \in \{0, 1\}$: Binary variable, equals 1 iff the path i and path j are coherent, in other words, they can fail simultaneously.

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

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

The additional constraints are

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

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

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

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

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).

IV. COMPLEXITY ANALYSIS

The column generation method is a very effective technique when used for the implementation of diversity coding since it optimally decomposes the problem into two iterative steps. The alternative coding-based methods, [9], [11], and [21] usually formulate the coding groups placement problem in a

single block. Among those, the diversity coding tree algorithm in [21] has significantly fewer variables than the others. On the other hand, in [16], it is shown that implementation of diversity coding with a 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 depend on

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

where ND is the nodal degree and $|V|$ is the number of nodes in the network. The number of available coding groups gets exponentially higher as the network size and connectivity increases. 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 optimal result without explicitly enumerating all of the possible coding groups. 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 [16]. 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.

TSA is the abbreviation of the two step approach defined in [16] and CGM is the proposed column generation method. C.G. corresponds to coding groups. In that table, it is 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. In addition, $ND \ll |N|$ and $O(|V|) \ll J$. Moreover, in CGM, the complexity of the subproblem and the complexity of the main problem are only linearly added together and multiplied with the average number of useful coding groups. 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 $|N|$ and ND . In addition, TSA works on much more coding groups than CGM does. As a result, the proposed CGM is much simpler and scalable than competitive techniques. Therefore, it can implement diversity coding over very large arbitrary networks. The simplicity of the CGM is also reflected in the next section in terms of runtime of different algorithms. In [21], it is explained that adopting single destination diversity coding enables near-hitless recovery and simplifies design complexity significantly. Therefore, in this paper, single des-

ination diversity coding is adopted.

V. THEORETICAL LOWER BOUND

In this section, we look into the theoretical limits of the capacity requirements of single destination coding-based recovery techniques. The derived lower bounds will be helpful to understand the space of improvement over our proposed techniques, which already can be implemented on very large real networks. Advancement in coding techniques usually results in much higher design complexity which prevents to apply them on real networks, such as [14]. Therefore, we need to find out the extent of incentive to developed more advanced coding techniques in terms of capacity efficiency.

A coding-based recovery technique can recover from single link failures instantaneously as long as it can support the traffic demand even if any of the edges are removed from the network [14]. In [14], it is mathematically stated using max-flow min-cut theorem [17] as

$$\min_C \left[\sum_{e \in C(E)} c(e) - \max_{e \in C(E)} c(e) \right] \geq h, \quad (36)$$

where C is a cut that disconnects the source node from the destination, $C(E)$ is the set of links in cut C , $c(e)$ is the capacity put on link e , and h is the total demand. In [14], there is a single source and destination node. However, we assume multiple source nodes and a single destination node. Therefore, we need to take the partial cuts, which disconnects a subset of source nodes from the destination, into account. In addition, we assume an edge includes two opposite directional links which fail simultaneously. The inequality (36) is modified as

$$\min_C \left[\sum_{e \in C(E)} c(e) - \max_{e \in C(E)} c(e) \right] \geq \sum_{s_i \in V(C)} t(s_i), \quad (37)$$

where $V(C)$ is the set of source nodes disconnected due to cut C and $t(s_i)$ is the demand of source node s_i .

We have benefited from an ILP formulation to find the lower bound of capacity requirements over arbitrary networks. This formulation finds the minimum capacity placements that satisfy the cut capacity criterions stated by inequality (37). Inequality (37) includes nonlinear operations therefore, we need to apply a set of conversions to make it linear, such as

$$\sum_{e \in C(E)} c(e) - \max_{e \in C(E)} c(e) \geq \sum_{s_i \in V(C)} t(s_i), \quad \forall C. \quad (38)$$

After one more conversion, inequality (38) becomes

$$\sum_{e \in C(E)} c(e) - c(f) \geq \sum_{s_i \in V(C)} t(s_i), \quad \forall f \in C(E), \forall C. \quad (39)$$

In the final step, the only set of constraints of the ILP formulation is

$$\sum_{e \in C(E)} (c(m_1) + c(m_2)) - c(l_1) - c(l_2) \geq \sum_{s_i \in V(C)} t(s_i), \quad \forall l_1, l_2 \in f, \forall f \in C(E), \forall C, \quad (40)$$

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 [21]	$ N E + N ^2/2$	$3 N E /2 + N V + 7 N /2$	-	-	1
ILP in [9]	$ N ^2/2(E + 1) + 3 N E $	$ N ^4/8 + \dots$	-	-	1
ILP in [11]	$ N E (V + 2) + N (N + 2 V) + \dots$	$ N V (3 E + N + V) + \dots$	-	-	1
TSA	J	$ V $	$ ND E + ND ^2/2$	$3 ND E /2 + ND V + \dots$	J
CGM	$O(V)$	$ V $	$(ND) E + (ND)^2/2$	$3(ND) E /2 + (ND) V + \dots$	$O(V)$

where $c(i)$ is an integer variable that keeps the required capacity over link i . The symbols m_1 and m_2 are opposite directional links over edge e . The objective function of the ILP formulation is

$$\min \sum_{l \in E} c(l) \times a_l, \quad (41)$$

where a_l is the length of link l .

The scalability of this operation is questionable since the number of all possible cuts is proportional to

$$\binom{|E|}{1} + \binom{|E|}{2} + \dots + \binom{|E|}{|E|}. \quad (42)$$

Even for a small network with 20 edges, the total number of cuts exceeds millions. Moreover, for each cut with a size $|C(E)|$, we require $|C(E)|$ inequalities. Therefore, it is very difficult to derive the tightest lower bound by investigating all possible cuts. Therefore, we opt to look into cuts including limited number of edges that gives an approximate lower bound.

VI. SIMULATION RESULTS

In this section, we present various 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. 6. The number 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 [22]. 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 [16], p -cycle protection [19], diversity coding tree from [21] 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 capacity (TC) and the runtime. The first technique in this table is the diversity coding tree algorithm. 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 coding, nonsystematic diversity coding, and coherent diversity coding. It is noted that 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 [19], which is also based on column generation.

Table II presents various trade-offs between protection techniques. First of all, the coding-based techniques are able to offer near-hitless recovery. Their restoration speed is at least two orders of magnitude higher than that of p -cycle protection [21]. 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 method. 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 shown compared to the systematic diversity coding and the theoretical lower bound with a breakdown over the nodes. It must be noted that the lower bound is not the tightest bound due to the exponential complexity of calculating it. The full and partial cuts with at most 5 edges are taken into account. For the tightest bound, all possible cuts should be investigated. 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. As expected,

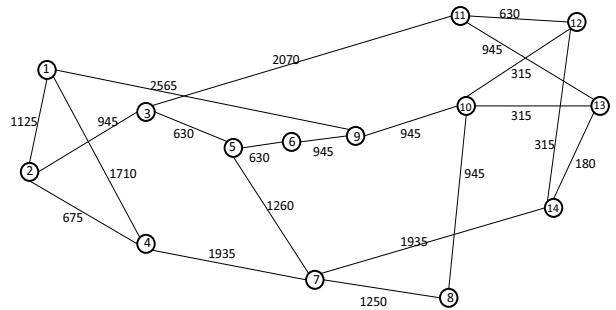


Fig. 6. NSFNET network.

TABLE II
COST AND RUNTIME COMPARISON BETWEEN DIFFERENT TECHNIQUES

Protection Technique	Total Cost	Runtime
Diversity Coding Tree	16880400	≈ 6 hours
TSA-SDC	15788730	≈ 6 minutes
TSA-NSDC	15746690	≈ 9 minutes
CGM-SDC	15793170	≈ 10 seconds
CGM-NSDC	15746690	≈ 5 minutes
CGM-CDC	15678770	≈ 1 hour
<i>P</i> -cycle algorithm	14814350	≈ 3 minutes

nonsystematic diversity coding performs better than systematic diversity coding, whereas coherent diversity coding performs best of all. As mentioned before, the improvement due to introduction of advanced coding techniques is limited over all different destination node scenarios. The difference between total capacity required by coherent diversity coding and the theoretical lower bound varies between 0-11% depending on the destination node. On average, coherent diversity coding requires 7.1% extra capacity than the theoretical lower bound, which is expected to get smaller if a tighter lower bound can be achieved bearing more complex calculations.

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 doing the same operation again. It is seen from the results, 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 [23], which is shown in Fig. 7. The traffic matrix is created using a gravity-based model [22]. In total, there are 23,204 static unit connection demands. This setup is chosen in order to 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 and *p*-cycle algorithm from [19] in terms of spare capacity percentage (SCaP) defined in [6]. The other coding-based recovery design algorithms are too complex to implement in this setup. The results are presented in Table V.

As seen from the results, the proposed design algorithm can achieve 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

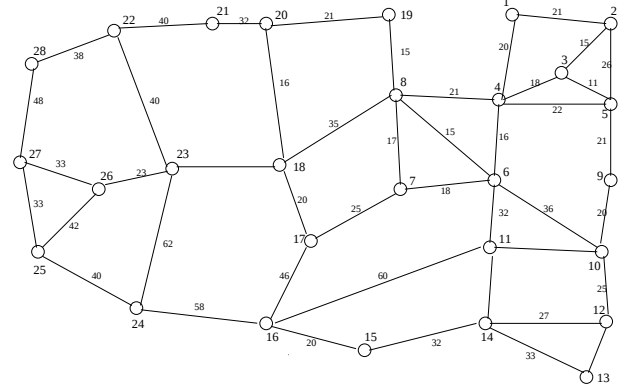


Fig. 7. U.S. long-distance network.

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 paths in TSA increases exponentially with the nodal degree and the number of nodes, whereas the number of candidate paths in CGM increases linearly with the number of nodes. The SCaP result of the new technique is better than that of the column generation based *p*-cycle algorithm. It should be noted that, *p*-cycle algorithm carries out Spare Capacity Placement (SCP) [17] due to high complexity, whereas the proposed algorithm carries out Joint Capacity Placement (JCP) [17]. Even with that adjustment, the proposed CGM is simpler than the *p*-cycle algorithm.

The third network is the long-distance network of France with 43 nodes and 142 unidirectional links taken from [24]. It is depicted in Fig. 8. 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 results are presented in Table VI. The runtime of 1+1 APS is equal to 1 minute.

CGM can achieve the optimal result in such a large network with over four million unit demands. The capacity efficiency

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	753450
Node 2	1301320	1301320	1301320	1215250
Node 3	211500	211500	211500	199350
Node 4	3913950	3913950	3913950	3709300
Node 5	455100	455100	455100	415600
Node 6	128150	128150	128150	128150
Node 7	1119200	1072720	1072720	1010350
Node 8	602000	602000	595700	536300
Node 9	1595700	1595700	1595700	1484850
Node 10	1177230	1172970	1172970	1054550
Node 11	573650	573650	573650	546650
Node 12	2293720	2293720	2264700	2074550
Node 13	1150350	1150350	1130350	1036700
Node 14	508750	508750	496150	473650
Total	15793170	15746690	15678770	14638700

TABLE IV
THE EFFECT OF GRANULARITY ON NETWORK OPTIMIZATION

Protection Technique	No. of connection demands	Total Cost	LP Bound	Optimality Gap
CGM-SDC	300	1602780	1578407	$\approx 1.47\%$
CGM-SDC	3000	1579355	1578407	$\approx 0.06\%$
CGM-SDC	30000	1579317	1578407	$\approx 0.06\%$

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%	≈ 3 hours	31464
CGM-SDC	105.6%	≈ 2 minutes	61
CGM-NSDC	105.5%	≈ 2 hours	72
CGM-CDC	93.4%	≈ 9 hours	79
<i>P</i> -cycle algorithm	107.0%	≈ 2.5 hours	32 (<i>p</i> -cycles)

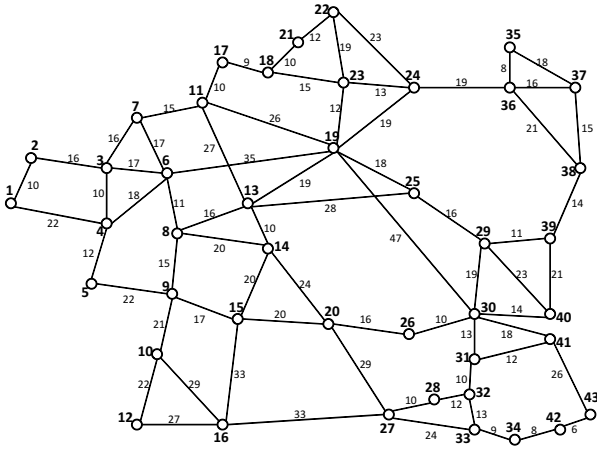


Fig. 8. Long-distance network of France.

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 the 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, diversity coding acts the same as 1+1 APS as we mentioned before.

VII. CONCLUSION

In this paper, we introduced an advanced version of diversity coding and an optimal and simple design algorithm 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 capacity.

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%	$\approx$ 2 minutes	12 nodes
3 links	125.5%	149.4%	$\approx$ 16 minutes	13 nodes
4 links	106.7%	140.6%	$\approx$ 39 minutes	14 nodes
5 links	146.4%	184.5%	$\approx$ 26 minutes	2 nodes
6 links	89.5%	126.5%	$\approx$ 85 minutes	1 node
7 links	86.6%	136.6	$\approx$ 53 minutes	1 node
Total	105.7%	141.0%	$\approx$ 85 minutes	43 nodes

The design algorithm consists of two parts, namely a main problem and 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 column-generation method and consist of two parts, a main problem where the traffic demands are met with the available coding groups and the 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 different for different coding techniques based on either ILP or MIP. There is a complexity versus capacity efficiency trade off 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 optimal 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. The coherent diversity coding has a higher capacity efficiency then both the nonsystematic and systematic diversity coding. 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 optimal solutions in a much shorter time then the competitive techniques. The complexity of the new technique is also more scalable then 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, "Optimal algorithms for near-hitless network restoration via diversity coding," in *Proc. IEEE GLOBECOM*, December 2012, pp. 1–7.

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

[3] 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 '90*, vol. 1, June 1990, pp. 95–104.

[4] —, "Diversity coding for transparent self-healing and fault-tolerant communication networks," *IEEE Trans. Commun.*, vol. 41, pp. 1677–1686, November 1993.

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

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

[7] S. N. Avci and E. Ayanoglu, "Coded path protection: Efficient conversion of sharing to coding," in *Proc. IEEE ICC*, June 2012.

[8] S. Ramamurthy, L. Sahasrabudde, and B. Mukherjee, "Survivable WDM mesh networks," *J. Lightwave Technol.*, vol. 21, no. 4, pp. 870–883, April 2003.

[9] A. E. Kamal and O. M. Al-Kofahi, "Efficient and agile 1+N protection," *IEEE Trans. Comm.*, vol. 59, no. 1, pp. 169–180, January 2011.

[10] 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.

[11] H. Ørby, G. Bóicz, P. Barbarcz, and J. Tapolcai, "Cost comparison of 1+1 path protection schemes: A case for coding," in *Proc. IEEE ICC*, June 2012.

[12] S. N. Avci and E. Ayanoglu, "Extended diversity coding: Coding protection and primary paths for network restoration," in *Proc. of the International Symposium on Network Coding*, June 2012, pp. 1–6.

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

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

[15] A. Sprinston, S. E. Rouayheb, and C. Georgiades, "Robust network coding for bidirected networks," in *Proc. of the Information Theory and Applications Workshop*, Jan.-Feb. 2007, pp. 378–383.

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

[17] W. D. Grover, *Mesh-Based Survivable Networks: Options and Strategies for Optical, MPLS, SONET, and ATM Networking*. Prentice-Hall PTR, 2004.

[18] G. Desaulniers, J. Desrosiers, and M. Solomon, *Column Generation*. Springer, 2005.

[19] T. Stidsen and T. Thomadsen, "Joint routing and protection using p-cycles," Informatics and Mathematical Modelling, Technical University of Denmark, DTU, Kgs. Lyngby, Denmark, Tech. Rep., May 2005.

[20] C. Rocha and B. Jaumard, "Revisiting p-cycles / FIPP p-cycles vs. shared link / path protection," in *Proc. 17th International Conference of Computer, Communications and Networks (ICCCN 2008)*, Virgin Island, USA, August 2008.

[21] S. N. Avci and E. Ayanoglu, "Optimal algorithms for near-hitless network restoration via diversity coding," *IEEE Trans. Comm.* (to be published), 2013.

[22] 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*, June 2003.

- [23] Y. Xiong and L. G. Mason, "Restoration strategies and spare capacity requirements in self-healing ATM networks," *IEEE/ACM Trans. Netw.*, vol. 7, pp. 98–110, February 1999.
- [24] 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 2003*, Banff, Canada, October 2003, pp. 212–220.