RAIRO-Oper. Res. 49 (2015) 297–312
DOI: 10.1051/ro/2014038 **RAIRO Operations Research**
www.rairo-ro.org

A NETWORK DESIGN PROBLEM WITH TWO-EDGE MATCHING FAILURES

Firdovsi Sharifov[1] and Hakan Kutucu[2]

Abstract. In this paper, we introduce a network design problem with two-edge matching failures. Given a graph, any two edges non-incident to the same node form a two-edge matching. The problem consists in finding a minimum-cost subgraph such that, when deleting any two-edge matching of the graph, every pair of terminal nodes remains connected. We give mixed integer linear programming formulations for the problem and propose a heuristic algorithm based on the Branch-and-Bound method to solve it. We also present computational results.

Keywords. Network design problem, linear programming, Branch-and-Bound method, matching.

Mathematics Subject Classification. 68M10, 90C05, 90C27, 90B10.

1. Introduction

In this paper, we consider a network design problem that consists in finding a minimum-cost network which must contain at least one path connecting any pair of terminal nodes when two-edge matching failures occur. A two-edge matching can be defined as a matching that has only two edges. In order to formulate the problem, we use some terminology from graph theory that will be presented in what follows.

Let $G = (V, E)$ be an undirected simple graph without parallel edges and loops, where V is the node set and E is the edge set. In V is given a subset of particular

Received July 22, 2013. Accepted July 17, 2014.

[1] V.M. Glushkov Institute of Cybernetics, National Academy of Sciences of Ukraine, 40 Prospekt Akademika Glushkova, 03187 Kiev, Ukraine. hakankutucu@iyte.edu.tr

[2] Department of Computer Engineering, Karabuk University, 78050 Karabuk, Turkey

Article published by EDP Sciences

nodes called *terminals*. For any pair of distinct vertices s and t in V, a path between these nodes is the sequence of nodes and edges $s = v_0, e_1, v_1, \ldots, v_{l-1}, e_l, v_l = t$, where each e_i is incident to nodes v_{i-1} and v_i, $(i = 1, \ldots, l)$ and such that no node or edge appears more than once in this sequence. If $s = t$, then the path is called a cycle in graph G. A subgraph $G' = (V', E')$ of G is a connected graph with the node set $V' \subseteq V$ and the edge set $E' \subseteq E$. In order to distinguish between directed and undirected edges, we use *edge* to refer to an undirected edge and *arc* to refer to a directed edge. We also use parentheses to denote an edge between nodes i and j, *i.e.*, (i, j), and arrow to denote an arc from node i to node j, *i.e.*, $(i \rightarrow j)$.

For each edge e in E, we denote a given cost on edge e by $c_e \geq 0$. The cost of any subgraph $G' = (V', E')$ of graph G is the sum of all costs of the edges in the set E'.

The network design problem with two-edges matching failures (NDP2EM) consists in finding a minimum cost subgraph G_* under the condition that in G_*, there exists at least one path between every distinct pair of terminal nodes in N after deleting any two non-incident edges in G.

This problem has a strong interest in the field of network connectivity and failure detection [11,12,18]. In [11], the authors consider the packet recovery problem from dual-links failures in Internet Protocol (IP) networks. They perform simulation experiments with a variety of network topologies to assess the effectiveness of three-edge connected networks for two arbitrary link failures. In [18], the authors present an integer linear program for the minimum monitoring cost problem for fast two link failures localization in a given optical network. They use the Euler cycle (non-simple loop-back cycles) technique to locate these type of failures and propose a heuristic solution to the problem. The effectiveness of using non-simple cycles instead of using edge-disjoint simple cycles is shown for graph topologies with triangular faces. Many real-world networks, U.S. National Network [12], for example, have same topologies. In addition to the communication network, this type of topologies arises in lattices of different crystal structures (see [15,17], for example). Crystal lattice nodes represent atoms and each atom is connected to its neighbor atoms by an edge. In [17], it is shown that the instability of the local crystalline lattice around a vacancy occurs only when at least two neighboring atoms are active enough, that is, they have some big energies in the result of line dislocation. The line dislocation of two atoms can be considered as two non-parallel edges deletion in the lattice. In order to characterize instability of crystal structures, the problem of determining whether a crystal lattice contains a connected sublattice after deleting any two non-parallel edges can be formulated as NDP2EM with unit edge costs.

Obviously, NDP2EM is an NP-hard problem in general since it includes some well-known NP-hard problems such as two-edge connected (when one edge fails) and (1,2)-survivable problems. These and other related network design problems such as the Steiner problem can be solved as minimum cost $s-r$ flow problems in the case $N = \{s, r\}$ $(|N| = 2)$. However, in this case, it is not clear whether a solution to NDP2EM can be found using well-known polynomial time flow algorithms.

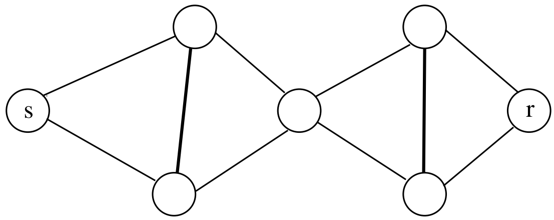

FIGURE 1. An optimal graph for NDP2EM.

To illustrate this, first we give the following characterizations for a feasible solution to NDP2EM.

Property 1. *Any planar graph with inner triangular faces is a feasible solution to NDP2EM.*

Indeed, since just one edge can be deleted in any of the triangular faces, then there are paths between any pair of the nodes of these faces after deleting one of their edges. This means that a planar graph with inner triangular faces remains connected after deleting any two-edge matching.

Property 2. *A feasible solution to NDP2EM is characterized by a graph which is between two-edge and three-edge connected graphs.*

From Property 1, it follows that any connected planar graph with edge-disjoint triangular faces is a feasible solution to NDP2EM. This type of graph is a minimal two-edge connected graph such that after deleting any edge, it becomes one-edge connected. On the other hand, any three-edge connected graph is also a feasible solution to NDP2EM since after deleting any two-edge matching, if the graph does not contain a path between any pair of terminal nodes, then it contradicts that the graph is three-edge connected. However, in many cases of a network, an optimal solution to NDP2EM is characterized by a graph which is between two-edge and three-edge connected graphs.

To show that an optimal solution to NDP2EM is characterized by a graph which is between two-edge and three-edge connected graphs, let us consider NDP2EM for the graph in Figure 1. According to Properties 1 and 2, it is a feasible solution to NDP2EM. It can be easily verified that after deleting any two-edge matching from the graph, there is a path between nodes s and r and NDP2EM has no solution on any subgraph with an edge set $E' \subset E$. This shows that the graph in Figure 1 is the optimal graph for NDP2EM. Clearly, this graph remains two-edge connected even after deleting both bold edges. On the other hand, the graph in Figure 1 is not a three-edge connected graph. Thus, the optimal graph to NDP2EM is between two-edge and three-edge connected graphs. These observations show that NDP2EM is harder than the above-mentioned problems. The graph in Figure 1 is a planar graph with triangular inner faces. It seems that using planar graphs with triangular inner faces is a basic tool in finding a solution to NDP2EM. In [16], it was shown that if a graph $G = (V, E)$ contains a planar subgraph G_0 where all inner faces are triangular and $N \subseteq V(G_0)$, then after deleting the edges of any k-edge matching in graph G, it contains at least one path between any pair of nodes in N. Based on this fact, we can attempt first to find some minimum cost connected planar graph with a node set N and convert this planar graph to a solution to NDP2EM using local three-edge connected graph heuristics. However, the problem of defining a minimum cost planar subgraph with a node set N is NP-hard as well as NDP2EM since the well-known Steiner problem is a special case of the former.

There are a lot of examples illustrating non-triangular of several facets of optimal graphs. This means that a graph with triangular inner faces is not an optimal one for many cases of NDP2EM. The following property is an accurate characterization more than Properties 1 and 2 for a feasible solution to NDP2EM.

Property 3. *If the edges of any cut do not induce a two-edge matching in a graph, then it is a feasible solution to NDP2EM.*

The proof of this property follows from the following fact. Consider the edges of a cut and assume that they do not induce two-edge matching. If any two edges of the cut do not have common nodes then the cut contains more than two edges. In this case, some nodes may be connected by more than three-edge disjoint paths. That is, some subgraph of the graph is at least three-edge connected. When some two edges of the cut have common nodes, it means that one of them cannot be deleted in the graph. Then the graph remains connected after deleting any two-edge matching.

Property 3 says that feasible graphs have fewer edges than a planar subgraph with triangular inner faces. Based on this property of an optimal subgraph, the techniques in [3, 7] can be used to solve NDP2EM.

There is a rich literature in polyhedral combinatorics on the facet manipulation technique for solving network design problems [1, 2, 4–6, 8–10, 13]. Success of the polyhedral approach is provided by generating valid inequalities for describing more precisely the convex hull of feasible solutions to the network design problems. Valid inequalities are usually derived from structural analysis of edge cut sets destroying the connectivity of feasible graphs [1, 2, 5, 6, 8, 9, 13]. The above example shows that deriving valid inequalities for NDP2EM from this type analysis is a hard task.

In order to solve NDP2EM, if we were to apply some trivial valid inequalities, $\delta(v) \geq 2$ for each $v \in N$ ($\delta(v)$ is the set of the edges with one end node v), for example, it could be regarded as local improving executed randomly. In the paper, we are going to use local and dual heuristics in [6, 14, 16] to get some feasible subgraph of G in which the objective value is an upper bound for NDP2EM. To compute a lower bound for the optimal value of NDP2EM, we use the linear relaxation of its model in Section 3. Then, in order to solve NDP2EM, we describe

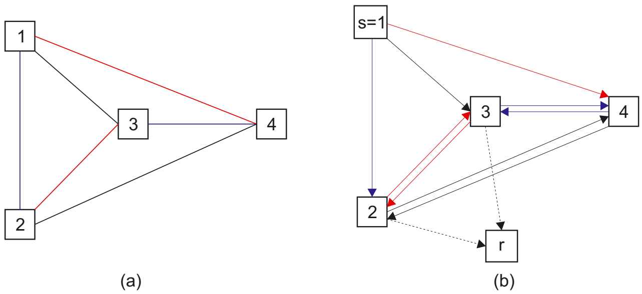

FIGURE 2. The original network (a) and the modified network (b).

a Branch-and-Bound algorithm with the upper and lower bounds procedures and report computational results for the algorithm.

2. Flow-Based formulation for NDP2EM

To formulate NDP2EM for an undirected network G with given terminal nodes in N, without loss of generality, we fix an arbitrary node s in N as the source and add a virtual node r to the network G as the sink. We also add virtual edges with unit capacities between every terminal t and the sink r. The terminal nodes t in $N_0 = N \setminus \{s\}$ are connected by the arcs $(t \to r)$ with unit capacities. Each edge $(s, v) \in E$ is then directed from s to v and weighted with a capacity $b = |N_0|$. Moreover, each edge $(v, w) \in E$, where $v \neq s, r$, is replaced by two arcs $(v \to w)$ and $(v \leftarrow w)$ with opposite directions with capacity b, too. Figure 2b shows such a modification for a complete graph with 4 nodes, which is shown in Figure 2a. After deleting a two-edge matching M from graph G, we denote an amount of the flow on an arc $(i \to j)$ by $x_{ij}(M)$ for each arc $(i \to j)$, when the edge $(i, j) \notin M$. Let Π be the set of all two-edge matchings in G. Let x_{ij} be an integer variable that represents the edge (i, j) in the network.

The flow model for NDP2EM can be formulated as the following integer linear program:

$$\min \sum_{(i,j)\in E} c_{ij} x_{ij} \tag{2.1}$$

subject to

$$\sum_{j\in\delta_M^-(i)} x_{ji}(M) - \sum_{j\in\delta_M^+(i)} x_{ij}(M) = \begin{cases} -b, & \text{if } i = s, \\ b, & \text{if } i = r, \ \forall\, i \in V, M \in \Pi, \\ 0, & \text{otherwise}, \end{cases} \tag{2.2}$$

$$0 \le x_{ij}(M) + x_{ji}(M) \le b \cdot x_{ij}, \quad \forall\, (i,j) \in E, M \in \Pi : (i,j) \notin M, \tag{2.3}$$

$$0 \le x_{tr}(M) \le 1, \quad \forall\, t \in N_0, M \in \mathit{\Pi}, \tag{2.4}$$

$$0 \le x_{ij} \le 1, \quad \forall\, (i,j) \in E, \tag{2.5}$$

$$x_{ij} = 0 \vee 1, \quad \forall\, (i,j) \in E, \tag{2.6}$$

where $\delta_M^+(i)$ is the set of arcs directed out of node i and $\delta_M^-(i)$ is the set of arcs directed into node i in the network $G_r = (V \cup \{r\}, E \cup \{(t,r), t \in N \setminus \{s\}\})$, after deleting a two-edge matching M from the network G.

The above model for NDP2EM contains flow conservation conditions for each two-edge matching fail. Clearly, the number of different two-edge matchings may be estimated as $O(m^2)$ or $O(n^4)$. We show that we can actually use only $O(m)$ two-edge matchings in the linear relaxation of NDP2EM. As an immediate consequence of this fact, the number of constraints in the primal problem or the number of dual variables is also reduced, and that makes computations of lower bound faster.

Note that $\mathit{\Pi} = \emptyset$ only when G is a star (a tree on n nodes having $n-1$ leaves) or a triangular (a cycle containing three edges). For these cases of G, it needs to find a minimum cost tree connecting terminal nodes since any edge cannot be deleted in G. Thus, we assume that G is neither a star nor a triangular. Now, we want to show some interesting properties of the dual problem of (2.1)–(2.5) (without the constraints (2.6)) to reduce the number of dual variables or the number of constraints of the primal problem (2.1)–(2.5). The dual problem can be formulated as follows:

$$\max \quad b \cdot \sum_{M \in \mathit{\Pi}} (u_r(M) - u_s(M)) - \sum_{t \in N_0} \sum_{M \in \mathit{\Pi}} y_{tr}(M) - \sum_{(i,j) \in E} z_{ij} \tag{2.7}$$

subject to

$$u_j(M) - u_i(M) \le w_{ij}(M), \quad \forall\, (i,j) \in E, M \in \mathit{\Pi} : (i,j) \notin M, \tag{2.8}$$

$$u_i(M) - u_j(M) \le w_{ij}(M), \quad \forall\, (i,j) \in E, M \in \mathit{\Pi} : (i,j) \notin M, i \ne s, \tag{2.9}$$

$$b \cdot \sum_{M \in \mathit{\Pi} : (i,j) \notin M} w_{ij}(M) \le c_{ij} + z_{ij}, \quad \forall\, (i,j) \in E, \tag{2.10}$$

$$y_{tr}(M) \ge 0,\ w_{ij}(M) \ge 0,\ z_{ij} \ge 0, \quad \forall\, (i,j) \in E, \tag{2.11}$$

$$u_r(M) - u_t(M) \le y_{tr}(M), \quad \forall\, t \in N_0, M \in \mathit{\Pi}. \tag{2.12}$$

In this dual linear program, $u_i(M)$ is the dual variable for the flow balance constraint (2.2) at node i after deleting a two-edge matching M, and $w_{ij}(M)$, $y_{tr}(M)$, and z_{ij} are the dual variables for constraints (2.3), (2.4) and (2.5), respectively.

Proposition 2.1. *There exists an optimal solution to* (2.7)–(2.12) *for which*

$$u_r(M) - u_t(M) = y_{tr}(M),$$

for all $t \in N_0$ *and* $M \in \Pi$.

Proof. An easy proof follows from the complementary slackness optimality conditions for the linear programs (2.1)–(2.5) and (2.7)–(2.12). Since the capacity of the cut separating the node r and the other nodes is equal to b, then $x_{tr}(M) = 1$ for all nodes $t \in N_0$ and for an optimal solution to (2.1)–(2.5). Then, by the complementary slackness conditions, we have

$$(u_r(M) - u_t(M) - y_{tr}(M))x_{tr}(M) = 0.$$

It follows that $u_r(M) - u_t(M) = y_{tr}(M)$ for all nodes $t \in N_0$. □

By Proposition 2.1, there exists an optimal solution such that $u_r(M) = u_t(M) + y_{tr}(M)$ for all $t \in N_0$ and $M \in \Pi$. Taking into account that $b = |N_0|$, the objective function (2.7) can be rewritten as follows:

$$\max \sum_{M\in\Pi} \left(\sum_{t\in N_0} u_t(M) - u_s(M) \right) - \sum_{(i,j)\in E} z_{ij} \tag{2.13}$$

and the constraints (2.12) can be deleted in the model.

There are usually a lot of two-edge matchings not covering some edges (i, j). This means that the dual problem includes the constraints (2.8)–(2.11) for each of these matchings. The following theorem states that in the dual program for each edge, we can consider only one matching among all these matchings not covering the edge.

Theorem 2.2. *There exists an optimal solution to the dual problem* (2.8)–(2.11) *and* (2.13) *such that* $u_j(M) - u_i(M) \neq 0$ *for at most one two-edge matching* M *among matchings not covering an edge* $(i, j) \in E$.

Proof. Consider any edge (i, j) and let $\sigma_{ij}(M) = u_i(M) - u_j(M)$. Suppose that $\sigma_{ij}(M_1) \neq 0$ and $\sigma_{ij}(M_2) \neq 0$ for any two-edge matchings M_1 and M_2 which do not cover the edge (i, j), where $u_i(M_1)$, $u_i(M_2)$ and $u_j(M_1)$, $u_i(M_2)$ are dual variables in an optimal solution to the dual problem (2.8)–(2.11), (2.13). Let $\sigma_{ij}(M) \neq 0$. Then either $\sigma_{ij}(M) > 0$ or $\sigma_{ji}(M) > 0$ for $M = M_1, M_2 \in \Pi$. Consider the case when $\sigma_{ij}(M_1) > 0$ and $\sigma_{ji}(M_2) > 0$. Now let us redefine

$$u_j(M_1) - u_i(M_1) = u_i(M_1) - u_j(M_1) + u_j(M_2) - u_i(M_2),$$

$\sigma_{ij}(M_2) = \sigma_{ji}(M_2) = 0$ and $w_{ij}(M_1) = w_{ij}(M_1) + w_{ij}(M_2)$, and $w_{ij}(M_2) = 0$. It is easy to see that these redefined values of these dual variables together with the optimal values of the remainder dual variables satisfy the constraints (2.8)–(2.10).

Thus, we have defined a feasible solution to the dual problem such that the constraints that held as equalities for the dual optimal solution hold as equalities for this dual feasible solution. Linear programming theory states that there exists a primal optimal solution corresponding to the dual optimal solution such that they together satisfy the complementary slackness optimality conditions. Therefore, this dual solution and the primal problem optimal solution together satisfy the complementary slackness optimality conditions, too. That is, the dual feasible solution is optimal. The proof of Theorem 2.2 is the same for the other possible cases of $\sigma_{ij}(M_1)$ and $\sigma_{ji}(M_2)$. Continuing this process for the edge (i,j), we obtain $w_{ij}(M_1) \geq 0$ and $\sigma_{ij}(M) = 0, w_{ij}(M) = 0$ for $M \neq M_1$.

By the above redefinition, we can set $u_k^*(M_1) = \sum_{M\in\Pi} u_k(M_1) - u_k(M_1)$ for $k = i, j$. Hence,

$$\sum_{M\in\Pi}\left(\sum_{t\in N_0} u_t(M) - bu_s(M)\right) = \sum_{t\in N_0}\left(\sum_{M\in\Pi} u_t(M) - u_s(M)\right) = \sum_{t\in N_0}(u_t^* - u_s^*),$$

where $u_t^* = u_t^*(M_t)$ and $u_s^* = u_s^*(M_s)$ for M_t and M_s in Π. □

Theorem 2.2 states that for each edge $(i,j) \in E$, we can add into Π only one two-edge matching M such that $(i,j) \notin M$. On the other hand, NDP2EM requires the existence of at least one path connecting any pair of terminal nodes, or the flow conservation conditions for any two-edge matching fail. Therefore, the set Π must contain one two-edge matching for which one of the edges is (i,j). Hence, we first add into Π all possible disjoint two-edge matchings, and if some edges are not covered by them, we add into Π a minimum number of distinct two-edge matchings covering these edges. Yet it is possible that no two-edge matching covers some edge. To clarify this edge, we claim the following lemma.

Lemma 2.3. *In a simple graph G, it does not exist a two-edge matching which covers an edge (i,j) if and only if $deg(i) + deg(j) = m + 1$, where $deg(v)$ denotes the degree of any $v \in V$.*

Proof. Suppose that M is a two-edge matching that contains the edges (i,j) and (k,l). After deleting the edges (i,j) and (k,l), we obtain a graph with $m-2$ edges. In the resulting graph, for node degrees $deg_2(i)$ and $deg_2(j)$, we have $deg_2(i) + deg_2(j) = m + 1 - 2 = m - 1$ since the edge (k,l) is not incident to the nodes i and j and the nodes i and j are not adjacent in this graph. The inequality $m - 1 > m - 2$ contradicts the fact that in a simple graph, the number of edges is less than the number of edges incident to two non-adjacent nodes. This is because the edge (i,j) cannot be covered by any two-edge matching.

Now, let $deg(i) + deg(j) = m + 1$ for the edge (i,j). Again after deleting the edge (i,j) from the graph G, the obtained graph has $m-1$ edges and the equality $deg_1(i) + deg_1(j) = m + 1 - 2 = m - 1$ holds for $deg_1(i)$ and $deg_1(j)$ of the nodes i and j in this graph. Since the nodes i and j are not adjacent in this graph, this equality says that any edge is incident to either the node i or the node j. This means that in G, there is not a two-edge matching covering the edge (i,j). □

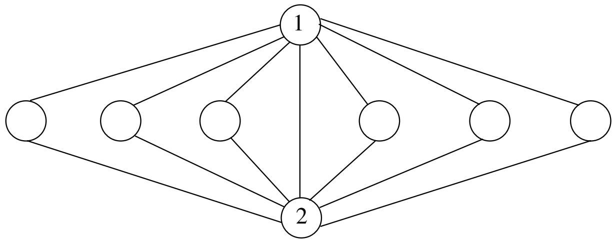

FIGURE 3. The edge $(1, 2)$ is not covered by any two-edge matching.

Lemma 2.3 has the interesting corollary that if $deg(i) + deg(j) = m + 1$ for the end nodes of some edge (i, j), then there is no matching (containing more than one edge) covering the edge (i, j) in G. This fact may be used in the well-known perfect and maximum weight matching algorithms for detecting this type of edges in advance since they can be deleted in G.

Let us consider the graph in Figure 3. It can be easily checked that none of the two-edge matching $M \in \Pi$ covers the edge $(1, 2)$.

The statement of Theorem 2.2 is true for the edge (i, j) since the existence of a two-edge matching covering any edge is not used in the proof. The number of two-edge distinct matchings can be estimated as $O(m^2)$. However, by Theorem 2.2, it suffices to add into Π no more than $O(m/2)$ two-edge matchings to solve the linear relaxation of NDP2EM. The probability of covering any edge by some two-edge matching is high in huge graphs. For these graphs, the number of two-edge matchings to be considered is reduced remarkably. For example, when G is a complete graph with 100 nodes, there are 11763675 two-edge distinct matchings. Yet to solve the dual problem (2.8)–(2.11) and (2.13) in this graph, we can only use 2475 distinct two-edge matchings that cover all edges.

3. Strengthened model for NDP2EM

Our experimental results show that there exists a big gap (relative error) with respect to the lower bound which is computed by solving (2.8)–(2.11) and (2.13) that is the dual problem of the linear relaxation (2.1)–(2.5). Though the model (2.1)–(2.5) includes fewer variables than the one that we are going to present in this section, the existence of the big gaps do not allow us to get a good solution to some test problems in Table 2. Constraints (2.3) are an aggregation of the flow conservation conditions when the amount of the flow from the source to any sink is restricted by unit value. In order to reformulate NDP2EM, we assign some terminal node s as a source and the remainder terminal nodes as sinks. Moreover, we use $x_{ij}^t(M)$ to denote the unit amount of flow on each arc $(i \to j)$ from the source s to a sink t after deleting a two-edge matching M in G, such that

$(i,j) \notin M$. In terms of variables $x_{ij}^t(M)$, the model for NDP2EM can be rewritten as follows:

$$\min \sum_{(i,j)\in E} c_{ij} x_{ij} \tag{3.1}$$

subject to

$$\sum_{j\in\delta_M^-(i)} x_{ji}^t(M) - \sum_{j\in\delta_M^+(i)} x_{ij}^t(M) = \begin{cases} -1, & \text{if } i = s, \\ 1, & \text{if } i = r, \ \forall\, i \in V, M \in \Pi, t \in N_0, \\ 0, & \text{otherwise}, \end{cases} \tag{3.2}$$

$$0 \le x_{ij}^t(M) + x_{ji}^t(M) \le x_{ij}, \quad \forall\,(i,j) \in E, M \in \Pi : (i,j) \notin M, t \in N_0, \tag{3.3}$$

$$0 \le x_{ij}^t(M), \quad \forall\,(i,j) \in E, M \in \Pi, t \in N_0, \tag{3.4}$$

$$0 \le x_{ij} \le 1 \quad \forall\,(i,j) \in E, \tag{3.5}$$

$$x_{ij} = 0 \vee 1, \quad \forall\,(i,j) \in E. \tag{3.6}$$

The dual program to the linear relaxation of (3.1)–(3.5) can be written as follows:

$$\max \sum_{M\in\Pi} \sum_{t\in N_0} (u_t^t(M) - u_s^t(M)) - \sum_{(i,j)\in E} z_{ij} \tag{3.7}$$

subject to

$$u_j^t(M) - u_i^t(M) \le w_{ij}^t(M), \quad \forall\, t \in N_0, (i,j) \in E, M \in \Pi : (i,j) \notin M, \tag{3.8}$$

$$u_i^t(M) - u_j^t(M) \le w_{ij}^t(M), \quad \forall\, t \in N_0, (i,j) \in E, M \in \Pi : (i,j) \notin M, \tag{3.9}$$

$$\sum_{t\in N_0} \sum_{M\in\Pi:(i,j)\notin M} w_{ij}^t(M) \le c_{ij} + z_{ij}, \quad \forall\,(i,j) \in E, \tag{3.10}$$

$$w_{ij}^t(M) \ge 0, \ \ z_{ij} \ge 0, \quad \forall\,(i,j) \in E. \tag{3.11}$$

It is easy to show that Theorem 2.2 holds for this dual problem, too. This allows to add into the set Π no more than $O(m)$ two-edge matchings such that each edge $(i,j) \in E$ is in one of them. This will permit to compute the lower bound by solving

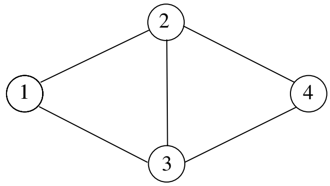

FIGURE 4. A feasible graph for NDP2EM.

the dual problem (3.7)–(3.11). For example, consider (3.1)–(3.6) for the graph in Figure 4, where the terminal nodes are 1 and 4. Let $c_{ij} = 1$ for all edges (i, j) in this graph. We set $s = 1$ and $t = 4$ and ignore the superscript t in all variables. Two-edge matchings are just $M_1 = \{(1, 3), (2, 4)\}$ and $M_2 = \{(1, 2), (3, 4)\}$, and no matching covers the edge $(2, 3)$. The objective value is 5 in the solution

$$x_{12}(M_1) = 1, x_{23}(M_1) = 1, x_{34}(M_1) = 1,$$

$$x_{13}(M_2) = 1, x_{32}(M_2) = 1, x_{24}(M_2) = 1,$$

of the problem (3.1)–(3.5). Now, consider a feasible solution to the dual program (3.7)–(3.11) ($b = 1$):

$$u_1(M_1) = 0, u_2(M_1) = 1, u_3(M_1) = 2, u_4(M_1) = 3,$$

$$u_1(M_2) = 0, u_3(M_2) = 1, u_2(M_1) = 2, u_4(M_2) = 3$$

and $z_{23} = 1$, $z_{ij} = 0$ for all edges $(i, j) \neq (2, 3)$. Since the dual objective value is again 5 these primal and dual solutions are optimal. Since $u_3(M_1) - u_2(M_1) = 1$ and $u_2(M_1) - u_3(M_1) = 1$ for the edge $(2, 3)$, we can redefine $u_2(M_1) - u_3(M_1) = 2$ and $u_3(M_2) - u_2(M_2) = 0$, $w_{23}(M_1) = 2$ and $w_{23}(M_2) = 0$ without changing the dual optimal objective value although no matching covers the edge $(2, 3)$.

4. BRANCH-AND-BOUND ALGORITHM

We use the results obtained after solving the dual problem (3.7)–(3.11) to get some feasible graph for NDP2EM. That is, the constraints (3.2)–(3.6) hold for this graph. The value of (3.1) corresponding to this graph is an upper bound $UB(root)$ for the optimal objective value of NDP2EM. Let E_0 be the set of edges for which the dual constraints (3.10)

$$\sum w_{ij}^t = c_{ij} + z_{ij}, \quad (i, j) \in E_0$$

hold as equalities for the optimal solution to the dual problem (3.7)–(3.11). From the theory of the linear programming problems, we have $x_{ij} \geq 0$ just for the edges in E_0 where x_{ij} are components of the optimal solution to the primal problem (3.1)–(3.5). Therefore, there exists a subgraph G_* with the edge set $E_* \subseteq E_0$

that is feasible for (3.1)−(3.6). To define a graph with the edge set E_*, the algorithm below uses some heuristics by solving NDP1EM test problems in [16], where the results showed that the improving of the objective value $UB(root)$ (the initial upper bound) occurs in few iterations of the Branch-and-Bound algorithm. In order to define G_*, we also use some local heuristics in [6,14] that improve locally.

Algorithm 1: Upper bound algorithm

Data: The set of terminal nodes N, the costs c_{ij} on the edges, and the subgraph $G_0 = (V_0, E_0)$ of $G = (V, E)$ defined by the set $E_0 \subseteq E$ of edges (i, j) for which the constraints

$$\sum w_{ij}^t \leq c_{ij} + z_{ij}, \quad (i, j) \in E$$

hold as equalities in the dual model (3.7)−(3.11).

Result: A feasible solution $G_* = (V_*, E_*)$ for NDP2EM and an upper bound.

1 **begin**
2 $\quad V_* \longleftarrow \emptyset$;
3 $\quad E_* \longleftarrow \emptyset$;
4 $\quad$ Find two edge-disjoint paths between source node s and each pair of other terminal nodes in the network G_0 ;
5 $\quad$ Add nodes and edges on the paths into V_* and E_*, respectively to create the subgraph $G_* = (V_*, E_*)$;
6 $\quad$ Set the capacities of the edges in G_* to 1;
7 $\quad$ Find the minimum cut in G_* ;
8 $\quad$ **if** *the size of the minimum cut* ≥ 3 **then**
9 $\qquad$ go to 16;
10 $\quad$ **end**
11 $\quad$ **if** *the size of the minimum cut* $= 2$ *and the edges of this cut form a matching,* **then**
12 $\qquad$ find the shortest path between the cut separating connected components after deleting the edges in the cut;
13 $\quad$ **end**
14 $\quad$ Add nodes and edges on the shortest path into V_* and E_*, respectively;
15 $\quad$ go to 6;
16 $\quad$ Subgraph $G_* = (V_*, E_*)$ is a feasible solution and the sum of the cost of the edges in G_* is an upper bound to NDP2EM;
17 **end**

As it will be presented in the next section, this algorithm can provide a small gap between the lower and upper bounds $LB(root)$ and $UB(root)$ that are the objective optimal value of the dual problem (3.7)−(3.11) and the objective value of (3.1) on the graph $G_*(root)$, which are to be computed at the root node $(root)$ of a Branch-and-Bound tree, where $G_*(root)$ is a feasible subgraph defined by the above Upper bound algorithm. When the gap is big, we process further to get the best feasible graph G_* by the Branch-and-Bound method. We use special techniques to make a decision how to branch and bound. Let $G_*(cur)$ denote a feasible graph defined by the above algorithm at the current node of the Branch-and-Bound tree. Let $LB(cur)$ and $UB(cur)$ denote the lower and the upper bounds at the current node cur of the Branch-and-Bound tree, respectively. At the beginning of branching, $cur = root$. To process the branching from the node cur, we choose an edge (i, j) for which constraints (3.10) hold as equalities and such that (i, j) is an edge of maximum number of paths (in $G_*(cur)$) connecting distinct pairs of terminal nodes. Then we work with either NDP2EM($c_{ij} = 0$) obtained by setting $c_{ij} = 0$ in (3.1)−(3.6) or NDP2EM($c_{ij} = \infty$) obtained by setting $c_{ij} = \infty$ in (3.1)−(3.6).

The optimal objective value of the dual problem (3.7)−(3.11) in which $c_{ij} = \infty$ is $LB(cur)$ for NDP2EM($c_{ij} = \infty$). Since $c_{ij} = \infty$, constraints (3.10) do not hold as equalities for the edge (i, j). Hence, $(i, j) \notin E(G_*(cur))$. The objective value of (3.1) in graph $G_*(cur)$ is $UB(cur)$ for NDP2EM($c_{ij} = \infty$). Since $(i, j) \notin E(G_*(cur))$, $LB(cur)$ and $UB(cur)$ are the lower and the upper bounds for the subproblem of (3.1)−(3.6) created in the branching process by setting $x_{ij} = 0$.

For NDP2EM($c_{ij} = 0$), the lower bound $LB(cur)$ is the sum of the initial edge cost c_{ij} and the optimal objective value of the dual problem (3.7)−(3.11). In this case, clearly constraints (3.10) hold as equalities for the edge (i, j), and from that (i, j) is an edge of maximum number of paths connecting distinct pairs of terminal nodes. This implies that the probability of edge (i, j) in $E(G_*(cur))$ is higher than the probability of the same event for the other edges in G. The objective value of (3.1) in the graph $G_*(cur)$ is $UB(cur)$ for NDP2EM($c_{ij} = 0$). From $(i, j) \in E(G_*(cur))$ it also follows that $LB(cur)$ and $UB(cur)$ are the lower and the upper bounds for the subproblem of (3.1)−(3.6) created in the branching process, respectively, by setting $x_{ij} = 1$. If $(i, j) \notin E(G_*)$, then fixing the variable x_{ij} in the branching process will not be successful. We use depth first search in the Branch-and-Bound tree to continue the branching process. The method stops as soon as the gap is under 2%.

5. Computational results

In this section, we use the methodology described in the previous sections to solve NDP2EM test problems. We compute a lower bound by solving the dual problem (3.7)−(3.11) by CPLEX 11.0 as the linear programming solver. The upper bound algorithm in Section 4 and branching depth first search procedures are coded in C. The algorithm is tested on a workstation with a 2.4 Ghz processor with 8 cores and 8GB RAM. We fix the maximum CPU time to 10 hours.

Our random problem generator creates NDP2EM test problems. It uses the following three input parameters: the number of nodes $|V|$, the number of edges $|E|$, and the number of terminals $|N|$. Then it randomly selects $|V|$ nodes on a 100×100 grid and arbitrarily creates $|E|$ edges between these nodes. Some graphs in the test problems are complete and some of them are not.

To ensure that the problems are feasible, it also creates a cycle that contains all the nodes. Random problems with 4 to 40 nodes are generated, and we test five

Table 1. List of abbreviations.

$\|V\|$	number of nodes in the graph
$\|E\|$	number of edges in the graph
$\|N\|$	number of terminals
LB(root)	lower bound obtained at the root of the Branch-and-Bound tree
LB(final)	global lower bound obtained at the end of the optimization
UB(root)	upper bound obtained in the beginning
UB(final)	best upper bound found
#UB	number of changes of upper bound
#B&B	number of Branch-and-Bound examined nodes (including the root node)
#Vars	number of variables in the dual model (3.7)−(3.11)
#Cons	number of constraints in the dual model (3.7)−(3.11)
Gap	gap between the best upper and final lower bounds: $\text{gap}=\dfrac{UB(final)-LB(final)}{LB(final)}\times 100$
CPU time	overall time (GAMS(The General Algebraic Modeling System) + C code)

instances of each size. Table 2 reports the average results obtained for randomly generated problems. Abbreviations used in Table 2 are summarized in Table 1.

Remark that a value of 0 for Gap indicates that UB (Final) is the optimal solution of the problem. Table 2 shows that for the test problems used in these experiments, 4 instances over 16 has been solved to optimality. Ten instances have been solved with a gap less than 2% within 10 hours. Two instances have not been solved with a gap less than 2% within the time limit. Optimal graphs obtained in the test problems are not 3-edge connected. For the instances marked by *, the initial graphs are almost complete and the resulting graphs are almost planar graphs with triangular inner faces. Solution graphs are between 2-edge connected and 3-edge connected graphs for the other test problems.

6. Conclusion

In this article, we have introduced a new network design problem using matching failures called NDP2EM. We have given a mathematical model (2.1)−(2.6) for the proposed problem. Then we have developed a strong model (3.7)−(3.11) to find tight lower bounds. Numerical results were obtained with a heuristic algorithm based on the Branch-and-Bound method using the improved dual model. From these results, we can conclude that the proposed algorithm is able to solve to optimality instances of small size (up to 40 nodes).

We also note that (3.1)−(3.6) can also be solved in the following way that is similar to the polyhedral approach. Let us consider the dual problem (3.7)−(3.11) when the set Π contains one two-edge matching for each edge. Then, we can find a solution of the primal problem (3.1)−(3.6) to the obtained dual problem. If all the variables are integer valued in the solution, then it is an optimal solution to NDP2EM. Otherwise, we can add a new two-edge matching to set Π that covers

TABLE 2. Computational results.

$\|V\|$	$\|E\|$	$\|N\|$	LB (root)	LB (final)	UB (root)	UB (final)	#UB	#B&B	#Vars	#Cons	Gap	CPU time (hh:mm:ss)
4	6	4	120.2	141.4	165	141.4	1.2	2.4	78	60	0	00:00:08
5	10	5	122.2	134	183.2	134	2.4	8.2	270	266	0	00:00:12
6	15	6	150.5	245	298.8	245	3.6	13	775	880	0	00:00:18
8	28	8	184.4	338.2	410.4	338.2	5.2	18.2	3360	5124	0	00:01:04
*12	66	12	301.6	377.6	661.2	378.6	6.6	25.4	27 655	42 659	0.26	00:08:03
20	100	10	292.2	351	527.4	352	5.8	27.2	53200	83 008	0.28	00:13:01
20	100	20	361	425.8	633	427.8	7.2	32.6	112 200	175 128	0.47	00:24:02
*20	190	10	231.4	311.2	475.4	312.2	7.6	35	178 031	305 596	0.32	00:31:00
*20	190	20	281.6	598.2	922.8	601.2	11	42.2	375 630	644 936	0.5	01:05:04
32	200	10	222	384	647	387	6	47	207200	344 126	0.78	01:00:05
32	200	32	620	950.8	1665	962.2	23	71.8	713 200	1 184 834	1.2	02:01:03
*32	496	32	510.8	917.2	1484.2	931.2	18.2	132.4	4 044 384	7 358 873	1.52	03:19:00
40	400	10	213.6	346.7	718.8	353.4	15	101.2	788800	1397380	1.93	03:22:10
40	400	20	387.2	490.4	1611.2	499.4	25.6	183	1664800	2949580	1.83	06:05:08
40	400	30	576	770.2	2007	787.2	41.2	304	2 540 800	4 501 780	2.2	10:00:00
*40	780	40	626.2	993.6	2411.6	1018.6	47.4	467.6	12 442 560	11 833 380	2.51	10:00:00

edges whose corresponding variables x_{ij} are not integer valued and consider the dual problem (3.7)−(3.11) with respect to the set Π. Again, if all the variables are integer valued in the solution of the primal problem (3.1)−(3.6) to the latter dual one, then this is a solution to NDP2EM. Otherwise, we add a new two-edge matching to Π as above to continue this process. After repeating this process for some time, we can use the above scheme to compute some lower and upper bounds which are used in the Branch-and-Bound method. Testing the efficiency of this polyhedral approach is the subject of future investigations.

Acknowledgements. We would like to thank the anonymous referee for his/her helpful comments that led to several improvements in the paper.

References

[1] M. Baïou, F. Barahona and A.R. Mahjoub, Separation of partition inequalities, *Math. Oper. Res.* **25** (2000) 243–254.

[2] F. Barahona, Network design using cut inequalities. *SIAM. J. Optim.* **6** (1996) 823–837.

[3] F.R.K. Chung and D. Mumford, Chordal completions of planar graphs. *J. Comb. Theor.* **31** (1994) 96–106.

[4] B. Fortz and M. Labbé, Polyhedral results for two-connected networks with bounded rings. *Math. Program., Ser. A* **93** (2002) 27–54.

[5] M. Grötschel, C.L. Monma and M. Stoer, Facets for polyhedra arising in the design of communication with low-connectivity constraints. *SIAM J. Optim.* **2** (1992) 474–504.

[6] M. Grötschel, C.L. Monma and M. Stoer, Polyhedral and computational investigations for designing communication networks with high survivability requirements. *Oper. Res.* **43** (1995) 1012–1024.

[7] P. Haggernes, Minimal triangulation of graphs. A survey. *Discrete Math.* **306** (2008) 297–317.

[8] H. Kerivin and A.R. Mahjoub, Separation of partition inequalities for the (1,2) survivable network design problem. *Oper. Res. Lett.* **30** (2002) 265–268.

[9] H. Kerivin and A.R. Mahjoub, Design of Survivable Networks: A survey. *Networks* **46** (2005) 1–21.

[10] H. Kerivin and A.R. Mahjoub, On Survivable Network Polyhedra. *Discrete Math.* **290** (2005) 183–210.

[11] S. Kini, S. Ramasubramanian and A. Kvalbein, A. Hensen, Fast recovery from dual -link failures in IP networks using tunneling. *IEEE/ACM Trans. Netw.* **18:6** (2010) 1988–1999.

[12] S.S. Lumetta and M. Médard, Classification of two-link failures for all-optical networks, In *Proceedings of the Optical Fiber Communication Conference, California* (2001).

[13] A.R. Mahjoub, Two-edge connected spanning subgraphs and polyhedra. *Math. Program.* **64** (1994) 199–208.

[14] C.L. Monma and D.F. Shallcross, Methods for designing communications networks with certain two-connected survivability constraints. *Oper. Res.* **3** (1989) 531–541.

[15] J.W. Morris Jr., *Survey of materials science, I. Structure*, Department of Materials Science and Engineering, University of California, Berkeley (2007) 138.

[16] F.A. Sharifov, Reliable network design problem. *Cybern. Syst. Anal.* **4** (2000) 145–156.

[17] L.W. Wang, L. Zhang and K. Lu, Vacancy-decomposition-induced lattice instability and its correlation with the kinetic stability limit of crystal, *Philosophical Magazine Lett.* **85** (2005) 213–219.

[18] B. Wu, K.l. Yeung and P.H. Ho, Monitoring cycle design for fast link failure location in all-optical networks. *J. Lightwave Technol.* **27** (2009) 1392–1401.