The Multi-Commodity Flow Problem with Disjoint Signaling Paths: A Branch-and-Benders-Cut Algorithm

Jiachen Zhang[1], Youcef Magnouche[2], Sebastien Martin[2], Antoine Fressancourt[2], J. Christopher Beck[1]

Abstract— **Data routing in networks is required to be efficient and reliable. Fast detection and recovery from link or path failures play crucial roles in the reliability guarantee. In this work, we investigate a variant of the multi-commodity flow problem to address one formalization of reliability, where for each demand, a primary path transmits the demand without exceeding a jitter limit and an arc-disjoint secondary path signals the possible failure of the primary path. We first present a compact mixed-integer linear programming model and then we devise a Branch-and-Benders-Cut algorithm to solve this combinatorial optimization problem. On a diverse set of instances, we evaluate the algorithm's performance and discuss several numerical results.**

I. INTRODUCTION

In telecommunications, routing refers to the process of discovering and selecting a path to send information to a destination. While in legacy communication networks this involved building an end-to-end circuit between two communicating parties, the Internet has popularized a radically new approach in which data is partitioned and sent in the form of packets from a source toward a destination in a best-effort fashion. The adoption of the packet relaying paradigm in communication networks has led to the redesign of most communication services such as landline telephony, video distribution, and cellular real-time communication. In this redesign, signalling operations allow two parties to set up a session with separated media exchange; and networks have been re-architected to route packets that carry media or control data following different paths optimized to meet different objectives: on the media path, jitter is bounded to avoid quality loss associated with buffering while on the control path, the delay is minimized to ensure fast call setups.

From the inception, packet-switched networks have been designed to be probabilistically resilient: in IP networks, failures are bound to occur, and the network should route traffic around them. This design approach is fundamentally different from legacy circuit-switched networks in which resilience is supposed to be achieved by preventing equipment to fail at all costs. With the decommission of circuit-switched communication networks and their replacement by packet-switched alternatives, mission-critical communication services such as emergency calls or infrastructure monitoring now have to be ensured on top of less deterministic networks.

Reliable network routing aims at solving this issue. It consists of protocols and practices that are designed to be resistant to failures and disruptions within a network, such as a link or a router going down, in order to improve the *quality of service* (QoS). Reliable routing is important for the reliable operation of networks, particularly in mission-critical communication services.

Fast detection and recovery play a vital role in overcoming unreliability, where recovery is usually realized by rerouting traffic around the failed path. In real-time communication services, failures affecting the media transfer path should be advertised on the signalling path so communicating parties can take quick actions to avoid communication disruptions. This can be accomplished using various protocols and algorithms that are designed to monitor the status of links and paths in the network, and to take corrective actions when a failure is detected.

In addition to fast detection and recovery, using disjoint paths (paths that do not share any links or routers with the primary path) to carry the signalling and media components of a communication process can provide an additional level of redundancy and reliability to the network, hence help to prevent disruptions to mission-critical communication services [1]. In particular, enforcing the disjointness of the signalling path and of the media transfer path can ensure the availability of mission-critical communication services. This approach can be used in conjunction with fast detection and recovery protocols, to reinforce that traffic is quickly rerouted around the failure.

With the recent trend of switching from the traditional best-effort traffic scheme to traffic engineering methodology, where routes between nodes in the network are explicitly specified [2], network routing problems are frequently abstracted as *multi-commodity flow* (MCF) problems [3]. While the splittable MCF can be solved in polynomial time or as a linear program, the unsplittable MCF is proved to be NP-hard [4] and is often solved as a *mixed-integer linear program* (MILP) with advanced optimization techniques. MCF is prevalent in various application contexts, including communication, transportation, logistics, and circuit design. In this work, the reliable routing problem is modelled as an MCF with arc-disjoint primary and secondary paths, while path jitter and arc capacity are constrained by specific upper bounds. This MCF variant is then solved exactly by a refined *Branch-and-Benders-Cut* (BBC) approach [5].

Our contribution is three-fold: First, to address reliable network routing, we investigate a multi-commodity flow problem with arc-disjoint signaling paths and propose a

[1]Jiachen Zhang and J. Christoper Beck are with the Department of Mechanical and Industrial Engineering, University of Toronto, 5 King's College Rd, Toronto, Ontario M5S 3G8 Canada {jasonzjc,jcb}@mie.utoronto.ca

[2]Youcef Magnouche, Sebastien Martin, and Antoine Fressancourt are with the Huawei Technologies, France Research Center, Boulogne-Billancourt, France {firstname.lastname}@huawei.com

MILP formulation and show that the problem is NP-hard. Second, we derive a Branch-and-Benders-Cut algorithm for solving the problem that outperforms the compact MILP model in speed and scalability. Third, we apply the proposed algorithms on various MCF instances and achieve cost minimization with guaranteed low-delay failure detection via arc-disjoint backup paths.

The paper is organized as follows. Related research is reviewed in Section II. In Section III, the problem is defined and its complexity is proved. In Section 4, the compact MILP model is presented. The Benders reformulation and decomposition are detailed in Section 5, followed by experimental evaluation in Section 6. Finally, in Section 7, discussions and conclusions are presented.

II. Related Works

A. Communication resilience to failures

In packed-switched networks, there are several approaches to achieve network resilience, such as using redundant routes as a backup for failure protection [6], implementing protocols that can quickly detect failures and trigger rerouting [7], and distributing traffic across multiple paths [8]. On top of those mechanisms, the operation of mission-critical media communications is often monitored using the signalling path so information about failure on the media path can be exchanged quickly and actions can be taken to route the media on another path or even link medium technology.

B. MCF with Primary and Secondary Paths

Secondary paths optimization problems related to network reliability mainly focus on finding the optimal secondary paths with route reliability, resource minimization objective, or various constraints [9]. Only few works about the multi-commodity flow problem with primary and secondary paths exist in the literature. Xia and Simonis [10] studied this MCF variant with arc-disjoint primary and secondary paths, both under capacity constraints. However, they overly complicated the capacity constraints with nonlinearity and did not consider delay or jitter. Our approach, by contrast, guarantees that the primary paths meet a jitter limit requirement and does not need to formulate any nonlinear constraints or objectives. Other works addressing primary and backup paths are investigated in *multi-protocol label switching* (MPLS) fast reroute application [11] and hybrid *interior gateway protocol* (IGP) and MPLS networks [12].

C. Arc-Disjointness

Arc-disjointness is required for backup paths to be isolated from the fault propagation in primary paths. From the theoretical perspective, many works have been done for 2-edge (k-edge) connected graphs [13], where 2 (k) arc-disjoint paths exist for each pair of nodes. In a more practical view, problems with disjoint primary and secondary paths have been studied in the context of network design with a focus on survivable flow [14] and have recently been generalized to multiple disjoint paths [15], where scalability is challenging. While a number of good heuristics and approximation algorithms have already been proposed for arc-disjoint paths [16], little has been done on exact methods to optimally solve the problem at a larger scale.

III. Problem Definition

The goals of the reliable network routing problem are (i) to route all demands with constrained jitter and limited arc capacity; (ii) to compute a secondary path, arc-disjoint from the primary, for each demand for fault signalling; and (iii) to minimize the costs of primary and the delays of the secondary paths considering demand bandwidth on the former. An illustration of reliable network routing is shown in Fig. 1.

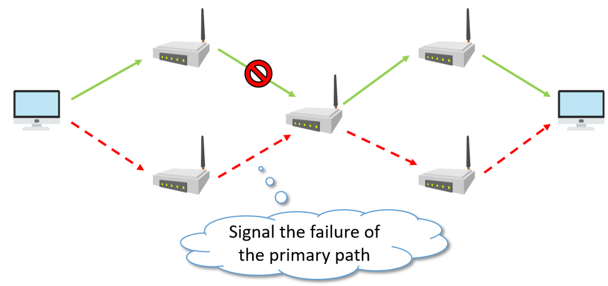

Fig. 1. Fast detection of path failure in reliable network routing.

In order to address the reliable network routing problem, we investigate a complex MCF variant where each demand needs to determine a primary and a secondary path, called *Bi-Path MCF* (BP-MCF). The two paths need to be arc-disjoint, i.e., no arc is used by both paths. Also, due to the accumulated jitter along arcs, the jitter of primary paths is required to be no greater than a given maximum value. Since the secondary paths are used only for signalling path failures, their delays are to be minimized. Moreover, as no demand is routed along secondary paths, the arc capacity constraints only take primary paths into account. The objective is to minimize the total cost of the primary paths and the total delay of the secondary paths.

The BP-MCF variant is defined on a graph $G = (V, A)$, where $V = \{v_1, ..., v_{|V|}\}$ represents the set of vertices and $A = \{a_1, ..., a_{|A|}\}$ represents the set of arcs. For each vertex v, $\delta^{\text{out}}(v)$ and $\delta^{\text{in}}(v)$ represent the sets of outgoing and incoming arcs of v, respectively. For each arc a, use a^- and a^+ to denote the source and destination nodes of the arc, respectively. There is a set of commodities (demands) $K = \{k_1, ..., k_{|K|}\}$ that need to be routed in the network. Each demand $k \in K$ has an origin $s_k \in V$, a destination $t_k \in V$, and a bandwidth $b_k \in \mathbb{R}^+$. Each arc $a \in A$ has a bandwidth capacity $c_a \in \mathbb{R}^+$, a delay $d_a \in \mathbb{R}^+$, a jitter $j_a \in \mathbb{R}^+$ and a cost $r_a \in \mathbb{R}^+$. The primary path jitter of demand k is required to be less than or equal to $J_k \in \mathbb{R}^+$. If the primary path of demand k contains a, the induced cost on the arc is $r_a b_k$. The objective function of the problem is the sum of all induced arc costs of primary paths and arc delays of secondary paths. Without loss of generality, we assume r_a and d_a to be positive. We use the following proposition to demonstrate the complexity of BP-MCF.

Proposition 1: The BP-MCF is NP-hard.

Proof: We prove the NP-hardness by a reduction from the *unsplittable MCF*, which is known to be NP-hard [4]. Specifically, for each demand, we add a dummy arc with zero delay and infinite jitter from the source to the destination. Then, with all the secondary paths assigned to the additional arcs (primary paths cannot use them due to the jitter upper bounds), the problem becomes an unsplittable MCF with the primary paths. Thus, BP-MCF is NP-hard. ■

To solve BP-MCF, we develop a compact MILP model and further a Benders algorithm based on decomposition.

IV. COMPACT MILP FORMULATION

We first present a compact mixed-integer linear programming (MILP) model with arc variables for demands. There are only binary decision variables $\mathbf{x} = \{x_a^k\}$ and $\mathbf{y} = \{y_a^k\}$, where $x_a^k = 1$ if the primary path of demand k uses arc a, and $= 0$ otherwise; and $y_a^k = 1$ if the secondary path of demand k uses arc a, and $= 0$ otherwise. The BP-MCF problem is equivalent to the following compact MILP model.

$$\min \sum_{a\in A} r_a \sum_{k\in K} b_k x_a^k + \sum_{a\in A} d_a \sum_{k\in K} y_a^k \tag{1a}$$

$$\text{s.t.} \sum_{a\in\delta^{\text{out}}(v)} x_a^k - \sum_{a\in\delta^{\text{in}}(v)} x_a^k = \begin{cases} 1 & \text{if } v = s_k, \\ -1 & \text{if } v = t_k, \\ 0 & \text{otherwise}, \end{cases} \quad \forall v\in V, \forall k\in K, \tag{1b}$$

$$\sum_{a\in\delta^{\text{out}}(v)} y_a^k - \sum_{a\in\delta^{\text{in}}(v)} y_a^k = \begin{cases} 1 & \text{if } v = s_k, \\ -1 & \text{if } v = t_k, \\ 0 & \text{otherwise}, \end{cases} \quad \forall v\in V, \forall k\in K, \tag{1c}$$

$$\sum_{k\in K} b_k x_a^k \le c_a \qquad \forall a\in A, \tag{1d}$$

$$\sum_{a\in A} j_a x_a^k \le J_k \qquad \forall k\in K, \tag{1e}$$

$$x_a^k + y_a^k \le 1 \qquad \forall a\in A, \forall k\in K, \tag{1f}$$

$$x_a^k \in \{0,1\} \qquad \forall a\in A, \forall k\in K, \tag{1g}$$

$$y_a^k \in \{0,1\} \qquad \forall a\in A, \forall k\in K. \tag{1h}$$

Term (1a) represents the total weighted costs of arc usage by the primary path and the accumulated delays of arc usage by the secondary path of all demands. Constraints (1b) and (1c) state the path flow constraints of primary and secondary paths. Constraints (1d) are the arc capacity constraints for the primary paths. Constraints (1e) ensure that the jitter of the primary paths respects the upper bound. Constraints (1f) guarantee the two paths to be arc-disjoint. Constraints (1g) and (1h) are the domain constraints of binary decision variables x and y.

Although the MILP model is well-formulated, and it can be directly solved with commercial MILP solvers such as Cplex [17], the scalability of the model is not promising. The reason lies in the fact that there are a large number of integer decision variables, which would be overwhelming for the MILP solvers. Thus, we investigate the Branch-and-Benders-Cut, an algorithm based on Benders decomposition to apportion the computational burden to separate problems containing fewer variables, with lazily generated constraints constructing the links between different problems.

V. BRANCH-AND-BENDERS-CUT ALGORITHM

In this section, we introduce the Branch-and-Benders-Cut (BBC) algorithm adapted to solving BP-MCF. We first give a brief review of the classical Benders decomposition (BD) algorithm for mixed-integer programs.

A. Benders Decomposition

Benders decomposition (BD) [18] is a well-studied exact approach for large-scale combinatorial optimization problems, especially for those with both integer and continuous decision variables. The target problem is usually decomposed into a master problem containing only the integer variables and some subproblems containing continuous variables. In addition, lower bounds of the optimal subproblem values are incorporated into the objective of the master problem.

When a subproblem is feasible, an optimality cut is added to improve the lower bound of the subproblem in the master objective. When a subproblem is infeasible, a feasibility cut is added to prune the infeasible solution from the master problem. The procedure is repeated until the optimal subproblem values are equal to the lower bounds obtained in the master problem. When sub-problems are linear programs, the approach is guaranteed to converge to an optimal solution. Benders approaches have been exploited in survivable [19], multi-layer [20], and hop-constrained [21] network design problems, as well as other telecommunication problems [22].

B. BBC Algorithm for BP-MCF

The Branch-and-Benders-Cut algorithm does not wait for the master problem to find the optimal solution $\hat{\mathbf{x}}$. Instead, for any intermediate integer feasible solution of the master problem found during the Branch-and-Cut [23] procedure, subproblems are solved to add Benders cuts lazily to the master problem. This minor algorithm modification leads to the BBC algorithm.

For BP-MCF, the original problem is decomposed into one master problem and a number of subproblems. The master problem is presented as follows.

$$\min \sum_{a\in A} r_a \sum_{k\in K} b_k x_a^k + \sum_{k\in K} \phi^k \tag{2a}$$

$$\text{s.t.} \sum_{a\in\delta^{\text{out}}(v)} x_a^k - \sum_{a\in\delta^{\text{in}}(v)} x_a^k = \begin{cases} 1 & \text{if } v = s_k, \\ -1 & \text{if } v = t_k, \\ 0 & \text{otherwise}, \end{cases} \quad \forall v\in V, \forall k\in K, \tag{2b}$$

$$\sum_{k\in K} b_k x_a^k \le c_a \qquad \forall a\in A, \tag{2c}$$

$$\sum_{a\in A} j_a x_a^k \le J_k \qquad \forall k\in K, \tag{2d}$$

$$x_a^k \in \{0,1\} \qquad \forall a\in A, \forall k\in K. \tag{2e}$$

In this MILP model, ϕ^k is the under-estimator of the delay of the secondary path for demand k. The goal of the master problem is to find primary paths for all demands with a minimum total cost, subject to jitter and capacity constraints. Once an integer feasible solution to the master problem is found, the primary paths of *all demands* are fixed. The subproblem consists, for every demand, in finding an arc-disjoint secondary path with a minimum delay. This is equivalent to finding a shortest-path, for each demand, in a new graph obtained by removing arcs of the primary path. The subproblem hence becomes polynomial-time solvable with well-known shortest path algorithms such as Dijkstra's algorithm [24]. However, in order to effectively add Benders cuts, we formulate the shortest secondary path problem for demand k as a *linear program* (LP) as follows.

$$\min \sum_{a\in A} d_a \sum_{k\in K} y_a^k \tag{3a}$$

$$\text{s.t.} \sum_{a\in\delta^{\text{out}}(v)} y_a^k - \sum_{a\in\delta^{\text{in}}(v)} y_a^k = \begin{cases} 1 & \text{if } v = s_k, \\ -1 & \text{if } v = t_k, \\ 0 & \text{otherwise,} \end{cases} \quad \forall v \in V, \tag{3b}$$

$$\hat{x}_a^k + y_a^k \le 1 \qquad \forall a \in A, \tag{3c}$$

$$y_a^k \ge 0 \qquad \forall a \in A. \tag{3d}$$

where $\hat{x}_a^k$ is obtained from the master problem.

Of course, in general, an LP formulation may result in fractional solutions. However, with $\hat{x}$ in constraints (3c) being integer, the model becomes a flow problem which is *totally unimodular*, implying that every extreme point (i.e., every optimal solution) is integer [25].

Let w^k be the dual variables of the path constraints (3b) and u^k be the dual variables of the arc-disjoint constraints (3c), the dual problem associated with the subproblem is as follows.

$$\max \sum_{a\in A} (\hat{x}_a^k - 1)u_a^k + w_{s_k}^k + w_{t_k}^k \tag{4a}$$

$$\text{s.t.} \quad e_a^k w_{a^-}^k + f_a^k w_{a^+}^k - u_a^k \le d_a \qquad \forall a \in A, \tag{4b}$$

$$u_a^k \ge 0 \qquad \forall a \in A. \tag{4c}$$

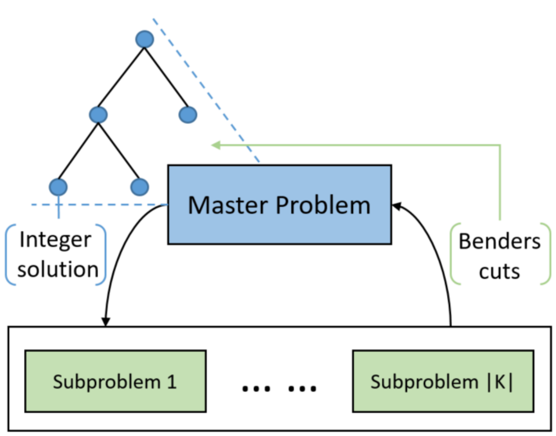

Fig. 2. Branch-and-Benders-Cut approach.

where $e_a^k = -1$ if $a^- = t_k$ and $e_a^k = 1$ otherwise; and $f_a^k = 1$ if $a^+ = t_k$ and $f_a^k = -1$ otherwise.

In practice, we solve the dual subproblem for each demand k separately. If the dual subproblem is feasible and is solved to optimality with optimal dual solution $\hat{u}^k$ and $\hat{w}^k$, the optimality cut that will be added to the master problem is:

$$\phi^k \ge (x_a^k - 1)\hat{u}_a^k + \hat{w}_{s_k}^k + \hat{w}_{t_k}^k. \tag{5}$$

On the contrary, if the dual subproblem is unbounded with dual rays $\hat{u}^k$ and $\hat{w}^k$, the feasibility cut that will be added to the master problem is:

$$0 \ge (x_a^k - 1)\hat{u}_a^k + \hat{w}_{s_k}^k + \hat{w}_{t_k}^k. \tag{6}$$

There are also applications where the delays (costs) of secondary paths are set to zero. In such cases, only the feasibility of the subproblem and the associated feasibility cuts are considered. The BBC approach is illustrated in Fig. 2.

C. Acceleration by Cut Selection

As there exists a huge volume of works on the acceleration of the Benders algorithm, we refine our Benders algorithm by doing cut selection based on identifying a minimal infeasible subsystem [26]. This cut selection technique is simple and elegant and can be applied to the dual subproblem directly. The refined dual subproblem is as follows.

$$\max \sum_{a\in A} (\hat{x}_a^k - 1)u_a^k + v_{s_k}^k + v_{t_k}^k - \hat{\phi}^k \pi_0^k \tag{7a}$$

$$\text{s.t.} \quad e_a^k w_{a^-}^k + f_a^k w_{a^+}^k - u_a^k \le d_a \pi_0^k \qquad \forall a \in A, \tag{7b}$$

$$\pi_0^k + \sum_{a\in A} u_a^k = 1 \tag{7c}$$

$$u_a^k \ge 0 \qquad \forall a \in A, \tag{7d}$$

$$w_v^k \ge 0 \qquad \forall v \in V, \tag{7e}$$

$$\pi_0^k \ge 0. \tag{7f}$$

where π_0^k is an additional continuous variable for normalization. The normalized dual subproblem is always bounded when containing π_0^k. As a result, we only need to consider the optimality cut as follows.

$$\hat{\pi}_0^k \phi^k \ge (x_a^k - 1)\hat{u}_a^k + \hat{w}_{s_k}^k + \hat{w}_{t_k}^k. \tag{8}$$

When $\hat{\pi}_0^k = 0$, this optimality cut becomes a feasibility cut, demonstrating the correctness and compactness of the cut selection method.

In practice, the simple cut selection modification brings a significant speedup to our BBC algorithm. In addition to this method, there is a large volume of literature on improving Benders cuts to accelerate the convergence and overcome the scalability issue of Benders approaches. However, since they are not directly relevant to our work, we refer interested readers to a review for a broader and deeper summary [27].

VI. EXPERIMENTAL EVALUATION

In this section, we present the numerical experiments on generated instances to provide an evaluation of the proposed Branch-and-Benders-Cut algorithm by comparing it to the compact MILP model.

A. Instance Generation

The problem instances are generated from a well-known library: *SNDlib*, a collection of datasets for telecommunication network design (`http://sndlib.zib.de/home.action`). SNDlib contains network topologies with multiple arc capacities, arc costs, and demands with bandwidth. We select 15 large topologies to avoid trivially small instances. The first capacity candidate and the cost of each arc provided in SNDlib are directly used as the capacity and arc cost in BP-MCF. The arc jitter is generated and is inversely proportional to the arc capacity, where the purpose is to allocate traffic on arcs with relatively large capacity during routing. The arc delay is also generated and is proportional to the distance between the source and destination of the arc according to the coordinates of the two nodes provided by SNDlib.

To make sure the number of feasible demands is not trivially small, a reverse link is created for each link if it does not directly exist in the network to implicitly enable multiple elementary paths without modifying the topology too much. The capacity, jitter, delay, and cost of the reverse link are the same as the original link. One demand is generated in advance for each pair of nodes in the network, with the bandwidth randomly selected from a constructed pool of bandwidths. Since SNDlib provides demands with bandwidth, the pool is the union of the demand bandwidths in the problem instance. For example, if $b_1 = 50, b_2 = 50, b_3 = 100$, the pool is $\{50, 100\}$.

For the maximum jitter value of each demand, Dijkstra's algorithm [24] is used to find the jitter-shortest-path. Using J_k^* to denote the smallest path jitter, then $\{2J_k^*, 3J_k^*, 4J_k^*\}$ are used as the jitter upper bounds of primary paths. However, the capacitated network may not be able to route all the generated demands. By using a simple greedy heuristic that routes demands one by one, the total number of feasible demands K_f for each jitter configuration is determined. More importantly, for instances obtained from the same topology and different values of maximum jitters, their demands are exactly the same. This is for the convenience of comparing total objective values between the three configurations of maximum jitter.

The 15 selected topologies from SNDlib with their main parameters are presented in Table I. Note that the number of arcs $|A|$ reflects the number after adding reverse links.

TABLE I
TOPOLOGY INFORMATION

ID	Topology	$\|V\|$	$\|A\|$	$\|K\|$	ID	Topology	$\|V\|$	$\|A\|$	$\|K\|$
1	cost266	37	114	467	9	nobel-eu	28	82	74
2	france	25	90	174	10	norway	27	102	702
3	geant	22	72	382	11	pioro40	40	178	82
4	germany50	50	176	224	12	sun	27	102	235
5	giul39	39	172	977	13	ta1	24	106	423
6	india35	35	160	1190	14	ta2	65	216	1170
7	janos-us-ca	39	122	155	15	zib54	54	161	1151
8	janos-us	26	84	38					

B. Experiment Setting

The algorithms have been implemented in C++ using Cplex 12.6 [17] as MILP-solver on a machine with Intel(R) Xeon(R) CPU E5-4627 v2of 3.30GHz with 504GB RAM, running under Linux 64 bits OS. One thread is used. A time limit is set to 1 hour and a memory limit for the search tree is set to 5Gb.

C. Numerical Results

The aggregated result metrics of the three delay configurations and two exact algorithms are shown in Table II. In the table, *opt gap* refers to the mean optimality gaps of instances with feasible solutions found by at least one method in a 1-hour time limit. In case of the absent upper bound for some instances, we use the objective value found by the greedy heuristic in the instance generation process as a default value to avoid an infinite gap. Also, *time(s)* refers to the mean algorithm runtime of instances with optimality proved by at least one method in the time limit.

From Table II we can observe that for all three configurations, the Branch-and-Benders-Cut (BBC) algorithm consistently outperforms the compact model, though they find feasible or optimal solutions for almost the same number of instances. In terms of speed, for all the instances solved to optimality, the BBC algorithm is faster than the compact model by approximated 2 times. Within the 1-hour time limit, BBC can reach smaller optimality gaps for instances not solved to optimality than the compact model.

With increasing upper bounds of jitters, each demand potentially contains more elementary paths and it is possible that the total routing cost and signalling delay decrease. We evaluate this aspect by detailing the relative objective value of each instance in Fig. 3. In this context, we only consider the 9 instances that can be proved to be optimal by the BBC algorithm. The relative objective for topology k and jitter configuration j is calculated according to the following formula:

$$\zeta_{j,k} = z_{j,k}/z_k^{max} \tag{9}$$

where z_k^{max} is the largest objective value among the three jitter configurations found by the BBC algorithm and $z_{j,k}$ is the optimal objective value of configuration j. From Fig. 3 we can see that the optimal objective values of some

TABLE II
RESULTS OF THE BBC ALGORITHM AND THE COMPACT MODEL

J_k	algo.	opt gap	time(s)	#nodes	#cuts	#feas	#opt
$2J_k^*$	BBC	3.35%	207	104496	7644	11/15	8/15
	compact	4.43%	542	45752	0	9/15	7/15
$3J_k^*$	BBC	3.53%	260	116246	6039	11/15	8/15
	compact	3.98%	365	54927	0	10/15	8/15
$4J_k^*$	BBC	3.10%	218	118526	6039	11/15	8/15
	compact	3.73%	582	50748	0	9/15	7/15
avg.	BBC	3.33%	229	113005	6573	11/15	8/15
	compact	4.05%	497	50476	0	9/15	7/15

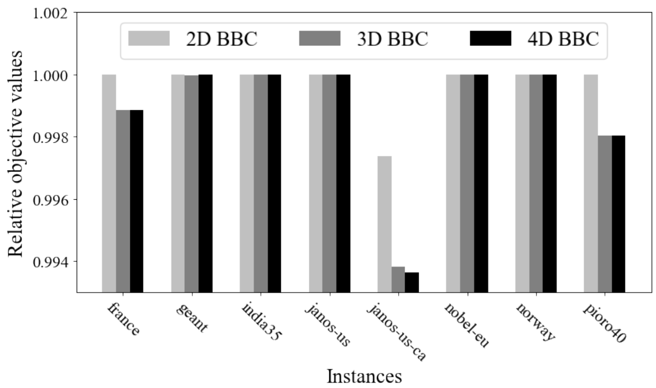

Fig. 3. Relative objective values of the optimal solutions among jitter configurations.

instances decrease slightly with the increase of jitter upper bound, which is consistent with our expectation. For the instances without significant changes, one of the tighter upper bounds is sufficiently large for these instances to end up with the solution with the minimum total objective values without violating jitter upper bounds.

According to the results shown above, we can see that the BBC algorithm outperforms the compact MILP model in terms of speed and scalability.

VII. Conclusion

In this paper, the reliable network routing problem has been modelled as a bi-path multi-commodity flow optimization problem, where a primary path delivers each demand and a secondary path signals the potential failure of the primary path. We have proved the NP-hardness of this MCF variant, provided a compact MILP formulation, and developed a Branch-and-Benders-Cut algorithm, which decouples continuous and integer variables for solving the problem exactly and efficiently. This algorithm also permits to reduce the complexity of subproblems significantly in advance. The experiment results indicate that the Branch-and-Benders-Cut algorithm for the bi-path MCF reduces significantly the CPU time compared to the compact model and thus represents an important contribution to assess the relative performance of bi-path routers existing in the literature.

Acknowledgment

This work is supported by Huawei Technologies for research in network routing.

References

[1] Y. H. Robinson, E. G. Julie, K. Saravanan, R. Kumar, M. Abdel-Basset, P. H. Thong, *et al.*, "Link-disjoint multipath routing for network traffic overload handling in mobile ad-hoc networks," *IEEE Access*, vol. 7, pp. 143 312–143 323, 2019.

[2] V. Foteinos, K. Tsagkaris, P. Peloso, L. Ciavaglia, and P. Demestichas, "Operator-friendly traffic engineering in IP/MPLS core networks," *IEEE Transactions on Network and Service Management*, vol. 11, no. 3, pp. 333–349, 2014.

[3] A. A. Assad, "Multicommodity network flows—a survey," *Networks*, vol. 8, no. 1, pp. 37–91, 1978.

[4] R. M. Karp, "On the computational complexity of combinatorial problems," *Networks*, vol. 5, no. 1, pp. 45–68, 1975.

[5] A. Moreno, P. Munari, and D. Alem, "A branch-and-benders-cut algorithm for the crew scheduling and routing problem in road restoration," *European Journal of Operational Research*, vol. 275, no. 1, pp. 16–34, 2019.

[6] M. Johnston, H.-W. Lee, and E. Modiano, "A robust optimization approach to backup network design with random failures," *IEEE/ACM Transactions on Networking*, vol. 23, no. 4, pp. 1216–1228, 2014.

[7] W. K. Lai, Z. C. Zheng, and C.-D. Tsai, "Fast reroute with pre-established bypass tunnel in MPLS," *Computer Communications*, vol. 31, no. 9, pp. 1660–1671, 2008.

[8] M. Hamdan, E. Hassan, A. Abdelaziz, A. Elhigazi, B. Mohammed, S. Khan, A. V. Vasilakos, and M. N. Marsono, "A comprehensive survey of load balancing techniques in software-defined network," *Journal of Network and Computer Applications*, vol. 174, p. 102856, 2021.

[9] Y. Zhang, Z. Fang, and Z. Xu, "An optimal design of multi-protocol label switching networks achieving reliability requirements," *Reliability Engineering & System Safety*, vol. 182, pp. 133–141, 2019.

[10] Q. Xia and H. Simonis, "Primary/secondary path generation problem: Reformulation, solutions and comparisons," in *International Conference on Networking*. Springer, 2005, pp. 611–619.

[11] D. Hock, M. Hartmann, M. Menth, M. Pióro, A. Tomaszewski, and C. Żukowski, "Comparison of IP-based and explicit paths for one-to-one fast reroute in MPLS networks," *Telecommunication Systems*, vol. 52, no. 2, pp. 947–958, 2013.

[12] A. Mereu, D. Cherubini, A. Fanni, and A. Frangioni, "Primary and backup paths optimal design for traffic engineering in hybrid IGP/MPLS networks," in *7th International Workshop on Design of Reliable Communication Networks*. IEEE, 2009, pp. 273–280.

[13] G.-R. Cai and Y.-G. Sun, "The minimum augmentation of any graph to a k-edge-connected graph," *Networks*, vol. 19, no. 1, pp. 151–172, 1989.

[14] A. Todimala and B. Ramamurthy, "Approximation algorithms for survivable multicommodity flow problems with applications to network design," in *Proceedings 25th IEEE International Conference on Computer Communications, IEEE INFOCOM 2006*. IEEE, 2006.

[15] D. Lopez-Pajares, E. Rojas, J. A. Carral, I. Martinez-Yelmo, and J. Alvarez-Horcajo, "The disjoint multipath challenge: Multiple disjoint paths guaranteeing scalability," *IEEE Access*, vol. 9, pp. 74 422–74 436, 2021.

[16] A. Srinivasan, "Improved approximations for edge-disjoint paths, unsplittable flow, and related routing problems," in *Proceedings 38th Annual Symposium on Foundations of Computer Science*. IEEE, 1997, pp. 416–425.

[17] IBM, "IBM ILOG CPLEX Optimizer," https://www.ibm.com/de-de/analytics/cplex-optimizer.

[18] J. Benders, "Partitioning procedures for solving mixed-variables programming problems," *Numerische Mathematik*, vol. 4, no. 1, pp. 238–252, 1962.

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

[20] B. Fortz and M. Poss, "An improved Benders decomposition applied to a multi-layer network design problem," *Operations Research Letters*, vol. 37, no. 5, pp. 359–364, 2009.

[21] Q. Botton, B. Fortz, L. Gouveia, and M. Poss, "Benders decomposition for the hop-constrained survivable network design problem," *INFORMS Journal on Computing*, vol. 25, no. 1, pp. 13–26, 2013.

[22] F. Gzara and E. Erkut, "Telecommunications network design with multiple technologies," *Telecommunication Systems*, vol. 46, no. 2, pp. 149–161, 2011.

[23] J. E. Mitchell, "Branch-and-cut algorithms for combinatorial optimization problems," *Handbook of Applied Optimization*, vol. 1, no. 1, pp. 65–77, 2002.

[24] E. Dijkstra, "A note on two problems in connexion with graphs," *Numerische Mathematik*, vol. 269, p. 271, 1959.

[25] M. Yannakakis, "On a class of totally unimodular matrices," *Mathematics of Operations Research*, vol. 10, no. 2, pp. 280–304, 1985.

[26] M. Fischetti, D. Salvagnin, and A. Zanette, "A note on the selection of Benders' cuts," *Mathematical Programming*, vol. 124, no. 1, pp. 175–182, 2010.

[27] R. Rahmaniani, T. G. Crainic, M. Gendreau, and W. Rei, "The benders decomposition algorithm: A literature review," *European Journal of Operational Research*, vol. 259, no. 3, pp. 801–817, 2017.