Branch-and-Price for Energy Optimization in Multi-Hop Wireless Networks

Liding Xu, Sonia Haddad-Vanier

Branch-and-Price for Energy Optimization in Multi-Hop Wireless Networks

Liding Xu[1] | **Sonia Haddad Vanier**[1]

[1]LIX CNRS, École Polytechnique, Institut Polytechnique de Paris, France

Correspondence
Sonia Vanier, LIX, Batiment Alan Turing, Campus de l'École Polytechnique, 1 Rue Honoré d'Estienne d'Orves, 91128 Palaiseau, France
Email: sonia.vanier@polytechnique.edu

Funding information

Wireless networks are widely deployed to support transmissions of the huge amount of data emitted by heterogeneous connected devices. The quality of service (QoS) and energy requirements in wireless networks are dramatically increasing and becoming challenging issues. Integrating network coding, in the data transmission process, leads to reduced energy consumption and improved QoS. In this work, we study the problem of energy efficiency in wireless multi-hop networks, namely wireless unsplittable multicommodity flow with network coding (wUMCFC). We propose mixed-integer programming formulations of the wUMCFC problem and an exact approach based on the branch-and-price algorithm. To the best of our knowledge, this work is the first study of this problem by mathematical programming tools. We perform a computational study on realistic instances with an analysis of the performance of the proposed methods. The obtained results show the efficiency of the network coding and provide important information for strategic decisions on routing policies and network management.

KEYWORDS
Network Optimization, Mixed Integer Programming (MIP), Branch and Price, Unsplittable Multi-Commodity Flow.

1 | INTRODUCTION

Multi-hop wireless networks support many applications requiring wireless communication on various platforms. We can mention unmanned aerials vehicles (UAVs) [41, 44], flying taxis (Vertical Take-Off and Landing: VTOL) [45], Internet-of-Things (IoT) sensor devices for monitoring [11, 52], connected healthcare [27, 53] and various emerging smart city [38, 56], whose components are connected via wireless networks. In all these cases, energy efficiency and traffic optimization are key challenges. In the upcoming IoT landscape [2, 47], smart systems are expected to be widely deployed everywhere over the world [22]. Current statistics indicate that billions of IoT devices are already deployed and connected in 2020, and this is expected to grow substantially in the future [3]. The research community faces solving optimization problems that will shape the connectivity of billions of devices with significant energy issues. Wireless networks are having significant and growing effects on energy consumption and environmental issues.

The main research in wireless networks focuses on single-hop communications. However, multi-hop extensions attract increasing interest for various wireless and IoT-related technologies, both in cellular [48] and non-cellular [26] multi-hop approaches.

In this work, we study the problem of energy efficiency of unsplittable routing with network coding in wireless multi-hop networks.

In wireless networks, each traffic demand is routed unsplittably through a single path [17]. The routing path of a demand connects its source node to its target node; data packets of the same demand follow the same path. Hence, the intermediate nodes in the path are not in charge of computing for each packet the next hop-node. Unsplittable routing speeds up packet handling processes, minimizes loss rates, and enables better quality of service (QoS) [17]. In contrast, splittable routing is very complex to apply in a wireless context. This would require sophisticated protocols and more intelligence in the network components. For more details on routing protocols in wireless networks, we refer to [18].

Network coding [1, 13, 16, 32, 37] can be defined as a mechanism allowing the intermediate nodes of a network to combine (encode) their incoming data and then transmit by broadcasting. Broadcasting is the term used to describe communication where the data packets are sent from one node to all other nodes. There is just one sender, but the information is sent to all connected receivers. Applying ideas from network coding and broadcasting realizes significant benefits in terms of resource and energy efficiency. Energy consumption is closely linked to the number of transmissions necessary for a unit of data to reach other nodes of the network. With network coding, we can get benefits in terms of throughout, scalability, and security [21].

In wireless networks, interference [54] is the dominant factor affecting the performance in terms of capacity, quality of service and energy consumption.

In [43], the authors proposed a linear program to compute routes that minimize the number of data transmissions, but that solution did not take the interference into account. In [35], an MILP model was proposed with network coding but it did not consider the effect of energy saved by network coding.

Many network optimization problems are naturally modelled as multi-commodity flow (MCF) problems [4, 10, 12, 24, 36, 51]. The classical MCF problems can be solved efficiently by mathematical programming methods. However, each network optimization problem is faced with new specific network features with new technical constraints. These yield complex MCF problems. The unsplittable multi-commodity flow (UMCF) problem [6] is a particularly well-studied variant of MCF problems, which has invoked the applications [30] in wireless networks. UMCF is also one of the well known $\mathcal{NP}$-hard problems in combinatorial optimization [20, 33, 34]. The column generation approaches [39] have proven efficient in solving MCF-related problems [9, 25, 30, 46, 50].

The problem addressed in this work is more complex since it generalizes UMCF with additional coding and inter-

ference constraints. Formulating telecommunication problems and adapting mathematical programming approaches to develop efficient algorithms require extensive research.

1.1 | Contributions

The first contribution of this work is our quantitative analysis and modeling of interference, network coding and energy consumption in the context of wireless networks. We propose a new problem, namely wireless UMCF with network coding (wUMCFC) and new models to integrate interference and coding.

Given the network topology, source-target traffic demands, transmissions capacities, and channels' interference, the wUMCFC problem seeks to minimize the energy cost of data transmission in multi-hop wireless networks and find an optimal unsplittable traffic routing. The wUMCFC incorporates network coding to reduce data transmission, save energy consumption and improve quality of service (QoS). We show that wUMCFC is $\mathcal{NP}$–hard.

Concerning the interference, we generalize the conventional concept of edge capacity constraints to capacity constraints over a set of edges, namely clique capacity constraints on a conflict graph. In addition, we show that network coding improves power consumption and network management.

The second contribution involves our formulations of the wUMCFC problem. The first class of formulations are compact edge-based formulations, which consist of a mixed-Boolean quadratic programming (MBQP) formulation and two mixed-integer linear programming (MILP) formulations. The two MILP formulations are the edge balance formulation and the edge linearization formulation. We study the strength of these two formulations. The second class of formulations is a Dantzig-Wolfe reformulation of the edge balance formulation, namely a path-based formulation.

To solve the path-based formulation efficiently, we develop a column generation approach and a branch-and-price (B&P) algorithm [5, 9, 25]. This yields our third contribution. In our B&P algorithm, the pricing problem is reduced to a shortest path problem in an extended graph. Although the edges' weights of the extended graph can be negative, we prove that the cycles of the extended graph have positive costs. Therefore, the shortest path in the extended graph can be calculated in polynomial time. We show that, under our reduction, the path generated in the original graph is always a simple path.

We perform a computational study on realistic problem instances with an analysis of the performance of the B&P algorithm and the effect of the network coding.

1.2 | Outline

This work is organized as follows: In Section 2, we introduce the classical UMCF problem formulation and notations. Then we introduce energy consumption, clique capacity constraints, and network coding, which form the wUMCFC problem. We then analyze the complexity of the wUMCFC problem. In Section 3, we present the compact edge-based formulations and the path-based formulation and compare these formulations. In Section 4, we present our algorithm to solve the LP relaxation of the path-based formulation. We describe the column generation and the solution approach for the pricing problem. In Section 5, we present branching rules to enforce the integrality of path variables. In Section 6, we perform two experiments. The first experiment shows that the B&P algorithm for the path-based formulation outperforms the MILP solver CPLEX for the edge balance formulation. The second experiment demonstrates that the network coding mechanism can decrease the energy cost significantly. Conclusions are drawn in Section 7 along with the prospect of future research.

2 | MODELS AND NOTATION

The network topology is represented by a bi-directed graph $G = (V, E)$, where V denotes the set of nodes corresponding to wireless transmission devices and E denotes the set of transmission links that can be used to route the traffic flows.

Unsplittable traffic demands are denoted by a set D of source-target node pairs (s, t) (st, for short).

Let's define the following notation:

C_{ij}: the transmission capacity of the edge (i, j), it measures the number of data packets that can be sent through the channel (i, j) per unit of time.

β_{ij}: the energy cost parameter for flow transmission along the edge (i, j). It measures the energy cost to transmit a unit data packet per unit of time.

d^{st}: the traffic demand from the source node s to the target node t for $st \in D$.

Decision variables:

x_{ij}^{st} : binary variable indicating whether demand st is routed on the edge (i, j), for $st \in D$ and $(i, j) \in E$.

The occupancy time ratio (OTR) is an important concept in wireless communication. For each edge $(i, j) \in E$, its OTR is defined as the total flow per unit of time divided by its capacity, i.e.,

$$\frac{\sum_{st \in D} d^{st} x_{ij}^{st}}{C_{ij}}. \tag{1}$$

The OTR measures the ratio that the channel along (i, j) is transmitting data per unit of time. Hence, the forwarding node i consumes energies during the occupancy time.

The UMCF problem aims to find a unique routing path for each demand that minimizes the total energy cost under capacity and demand constraints.

Before introducing network interference and network coding, let us recall the ILP formulation of the classical minimum cost (UMCF) problem:

$$\text{UMCF} \begin{cases} \min z = \displaystyle\sum_{st \in D} \sum_{(i,j) \in E} \beta_{ij} d^{st} x_{ij}^{st} & & (2.0) \\ \displaystyle\sum_{j:(i,j) \in E} x_{ij}^{st} - \sum_{j:(j,i) \in E} x_{ji}^{st} = 0, & \forall i \in V - \{s, t\}, \ \forall st \in D, & (2.1) \\ \displaystyle\sum_{(s,i) \in E} x_{si}^{st} - \sum_{(i,s) \in E} x_{is}^{st} = 1, & \forall st \in D, & (2.2) \\ \displaystyle\sum_{(i,t) \in E} x_{it}^{st} - \sum_{(t,i) \in E} x_{ti}^{st} = 1, & \forall st \in D, & (2.3) \\ \displaystyle\sum_{st \in D} \frac{d^{st}}{C_{ij}} x_{ij}^{st} \le 1, & \forall (i, j) \in E, & (2.4) \\ x_{ij}^{st} \in \{0, 1\}, & \forall (i, j) \in E, \ \forall st \in D. & \end{cases} \tag{2}$$

Objective function z (2.0): energy consumption of data transmission per unit of time.

Flow conservation constraints (2.1) to (2.3): flow conservation constraints at each node.

Edge capacity constraint (2.4): the total flow on an edge should be equal at most to the available transmission capacity. This constraint stipulates that the OTR of an edge must be at most 1.

The mathematical formulation of our wUMCFC problem requires the addition of new constraints and new variables to the classical UMFC. This to integrate interference and coding mechanisms.

In the following subsections, we present energy consumption, clique capacity constraints, and network coding. We also analyze the complexity of the wUMCFC problem.

2.1 | Energy consumption

This work considers the energy consumption induced by data transmissions via active devices (nodes) in the network.

A node $i \in V$ is active when it is transmitting data. We introduce an energy cost parameter β_i that measures the cost of energy consumed by an active node per unit of time. This parameter depends on the characteristics of the node's communication device.

Let f_{ij} be the flow on the edge $(i,j) \in E$, i.e., the number of data packets to transmit from node i to node j per unit of time. Recall that the OTR $\frac{f_{ij}}{C_{ij}}$ is the time ratio that node i is active in transmitting data along the channel. The energy cost per unit of time by the channel (i,j) is $\frac{f_{ij}}{C_{ij}}\beta_i$. During the other $1 - \frac{f_{ij}}{C_{ij}}$ portion of a unit time, no energy is consumed because the node does not transmit data through this channel. β_{ij}, the energy cost parameter of (i,j), follows as:

$$\beta_{ij} = \frac{\beta_i}{C_{ij}} \tag{3}$$

2.2 | Clique capacity constraint

The difference between wireless and wired networks lies mainly in the use of communication channels and transmission technologies [55]. In wired networks, the capacity of one channel is not affected by the data transmissions of any other channels. In wireless networks, channels inherently share the same communication space, and the interference between channels in a neighborhood would decrease their capacities. When a channel transmits a data packet, it consumes capacity at its neighbors. Network interference reduces transmissions capacities significantly [7].

We model interference using capacity constraints, which are defined over the clique set of an undirected conflict graph $G_c = (N, L)$ where $N = E$.

The nodes of the conflict graph G_c are the edges of the network graph G. The links of G_c represent interference between the edges of G which cannot transmit data at the same time.

If two edges in G are in interference, then a link between their corresponding nodes is in G_c.

The graph G_c is constructed incrementally with an initial empty link set L, as follows:

1. $N = E$ and $L = \emptyset$.
2. If $(i,j) \in N$ and $(k,l) \in N$ are under interference then
add the link $\{(i,j),(k,l)\}$ to L: $L = L \cup \{(i,j),(k,l)\}$.

Hence, two edges of G share capacity if they are adjacent in G_c.

Interference can be modeled in various ways [28]. We introduce the $n-$dist interference model to construct the conflict graphs in our experiments. The $n-$dist model extends single-hop and two-hop models proposed in [15].

Let $p = (v_1, \ldots, v_h)$ be a simple path in the graph G, where v_t ($t \in \{1, \ldots, h\}$) is the node in the path p. We define the length of p as the number of nodes h in the path. For $i, k \in V$, we define $\mathrm{dist}(i,k)$ as the length of the shortest path from i to k or from k to i, i.e., the path with the minimum number of hops. Let (i,j) and $(k,l) \in E$, the distance $\mathrm{dist}\,((i,j),(k,l))$ is defined as the length of a shortest path between any pair of $\{i,k\}$, $\{i,l\}$, $\{j,k\}$, and $\{j,l\}$,

i.e., $\mathrm{dist}\,((i,j),(k,l)) = \min_{t_1\in\{i,j\},t_2\in\{k,l\}}\{\mathrm{dist}(t_1,t_2),\mathrm{dist}(t_2,t_1)\}$. Hence, $\mathrm{dist}\,((i,j),(k,l))$ measures the minimum distance between the tail and end nodes of edges (i,j) and (k,l) .

If $\mathrm{dist}\,((i,j),(k,l)) \leq n$, then under the $n-$dist model, (i,j) and (k,l) are in interference.

For example, Figure (1a) shows a network of five nodes and Figure (1b) shows its conflict graph constructed via the 2−dist interference model. The 2−dist interference model corresponds to the single-hop model in [15]. This model considers that a node interferes with its neighbors.

The distance between (1,2) and (4,5) is 3, so they are not connected via any link in G_c.

The distance between (1,2) and (3,4) is 2, so they are connected via one link in G_c. One can check whether two edges share capacity by the adjacency of their corresponding nodes in G_c.

(a) Network with 5 hops (b) Conflict graph

FIGURE 1 Interference model

We adopt the capacity-sharing model in wireless communication context from [29]. Let f_{ij} be the flow on the edge (i,j); recall that $\frac{f_{ij}}{C_{ij}}$ is the OTR of this edge. Two nodes in interference can not transmit at the same time otherwise these transmissions will fail.

Let m be a subset of edges of Gf t. Ihe corresponding nodes in G_c are a clique, then every edge of m shares OTR with other edges in m.

The sum of OTRs of edges in m should be at most 1. The clique capacity constraint follows as:

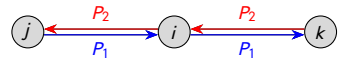
$$\sum_{(i,j)\in m} \frac{f_{ij}}{C_{ij}} \leq 1. \tag{4}$$

For any two clique sets m_1 and m_2, such that $m_1 \subset m_2$, it follows that the clique capacity constraint over m_1 is dominated by the clique capacity constraint over m_2:

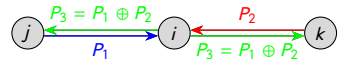
$$\sum_{(i,j)\in m_1} \frac{f_{ij}}{C_{ij}} \leq \sum_{(i,j)\in m_2} \frac{f_{ij}}{C_{ij}}. \tag{5}$$

Therefore, non-dominated constraints are defined over maximal cliques. The dominance relation over clique capacity constraints is equivalent to set inclusion over the corresponding cliques. Then, it suffices to consider only non-dominated constraints induced by maximal cliques in the model.

We denote the set of maximal cliques of G_c by M. For example in Figure (1b), $\{(1,2),(2,3)\}$ is a clique set, but $\{(1,2),(2,3),(3,4)\}$ is the maximal clique set including it, so the model solely needs to stipulate the constraint: $\sum_{(i,j)\in\{(1,2),(2,3),(3,4)\}} \frac{f_{ij}}{C_{ij}} \leq 1$. There are two maximal cliques of G_c in Figure (1b). Then, in this case, $M = \{\{(1,2),(2,3),(3,4)\},\{(2$



(a) Classical forwarding



(b) Forwarding with network coding

FIGURE 2 Network coding

2.3 | Network coding

Wireless communications cause the interference, but they allow data to be transmitted simultaneously between nearby components. Therefore, when a node $i \in V$ forwards a data packet to a node $j \in V$, other connected nodes will also receive this packet. This communication case is called broadcasting. Broadcasting is the term used to describe communication where the data packets are sent from one node to all other connected nodes.

Network coding [42] is a mechanism that allows intermediate nodes to combine data packets before forwarding them. It saves data transmissions by broadcasting.

Figure (2) illustrates the principle of network coding by a triple of nodes; node $j \in V$ and node $k \in V$ send packets P_1 and P_2 to each other through the intermediate node $i \in V$.

The classical forwarding scheme in Figure (2a) uses 4 transmissions to send P_1 and P_2. Node j (resp. k) sends its packet P_1 (resp. P_2) to node i, and node i sends P_1 to k and P_2 to j, separately.

With network coding in Figure (2b), only 3 transmissions are needed, and hence the energy cost by the device at i is saved. After node j and node k record and send their packets to node i, node i encodes data of these two data packets by XOR operation to obtain an encoded packet $P_3 \coloneqq P_1 \oplus P_2$, where $\oplus$ is the bit-wise Boolean addition.

Node i broadcasts the encoded packet P_3 to node j and node k simultaneously. Node j and node k decode data of P_3 by XOR-ing with recorded packets, $P_1 = P_3 \oplus P_2$ and $P_2 = P_3 \oplus P_1$. As a result, i only needs one transmission.

In principle, if $(j,i),(i,k),(k,i)$, and (i,j) are in E, then there is a coding opportunity, we call it a three-hop pattern.

We define the opportunity set as follows:

$$\Lambda^i_{\{k,j\}} \coloneqq \{(j,i),(i,k),(k,i),(i,j)\}. \tag{6}$$

For $\Lambda^i_{\{k,j\}} \subset E$, let f_{jik} be the flow along $j \to i \to k$, and let f_{kij} be the flow along $k \to i \to j$. If f_{jik} and f_{kij} are non-zero, then network coding can be applied, and the node i could encode parts of the opposite flows f_{jik} and f_{kij}.

We define $u^i_{\{k,j\}}$ as the flow encoded by node i it measures the number of f_{jik} and f_{kij} data packets that node i could code per unit of time.

Since the maximum encoded data cannot exceed data arriving at a node i, $u^i_{\{k,j\}}$ has to satisfy the following inequality:

$$u^i_{\{k,j\}} \leq \min(f_{jik}, f_{kij}). \tag{7}$$

The next Sections explain how the network coding increases the capacity of the network and decreases the energy cost by fewer data transmissions.

2.3.1 | Effects on clique capacity constraints

Network coding reduces the occupancy time rate (OTR) and hence decreases the left-hand side of the capacity constraints.

Without network coding, the sum of OTRs of transmitting f_{jik} and f_{kij} is

$$\frac{f_{jik}}{C_{ji}} + \frac{f_{jik}}{C_{ik}} + \frac{f_{kij}}{C_{ki}} + \frac{f_{kij}}{C_{ij}}. \tag{8}$$

The amount of encoded flow $u^i_{\{k,j\}}$ could be broadcast from i to its targets k and j. Let the OTR of broadcast $u^i_{\{k,j\}}$ by network coding be $T^i_{\{k,j\}} u^i_{\{k,j\}}$, where $T^i_{\{k,j\}}$ is a transmission parameter according to the network property.

It must be that

$$T^i_{\{k,j\}} \geq \max\{\frac{1}{C_{ik}}, \frac{1}{C_{ij}}\}, \tag{9}$$

because the time of broadcasting should be at least the time of any of two separated forwardings.

The remaining parts of $f_{jik} - u^i_{\{k,j\}}$ and $f_{kij} - u^i_{\{k,j\}}$ are transmitted by the classical forwarding scheme.

The OTR with network coding follows as:

$$\begin{aligned} &\frac{f_{jik}}{C_{ji}} + \frac{f_{jik} - u^i_{\{k,j\}}}{C_{ik}} + \frac{f_{kij}}{C_{ki}} + \frac{f_{kij} - u^i_{\{k,j\}}}{C_{ij}} + T^i_{\{k,j\}} u^i_{\{k,j\}} \\ =&\frac{f_{jik}}{C_{ji}} + \frac{f_{jik}}{C_{ik}} + \frac{f_{kij}}{C_{ki}} + \frac{f_{kij}}{C_{ij}} - (\frac{1}{C_{ij}} + \frac{1}{C_{ik}} - T^i_{\{k,j\}}) u^i_{\{k,j\}}. \end{aligned} \tag{10}$$

Let $C^i_{\{k,j\}}$ be the increased capacity with network coding within the three-hop pattern $\Lambda^i_{\{k,j\}}$. $C^i_{\{k,j\}}$ measures the number of data packets that will no longer be transmitted on $\Lambda^i_{\{k,j\}}$ due to network coding, and

$$\frac{1}{C^i_{\{k,j\}}} = \frac{1}{C_{ij}} + \frac{1}{C_{ik}} - T^i_{\{k,j\}}. \tag{11}$$

We obtain the following bound on $\frac{1}{C^i_{\{k,j\}}}$ by the bound on $T^i_{\{k,j\}}$:

$$\frac{1}{C^i_{\{k,j\}}} \leq \min\left\{\frac{1}{C_{ij}}, \frac{1}{C_{ik}}\right\}. \tag{12}$$

Equivalently, $C^i_{\{k,j\}} \geq \max\{C_{ij}, C_{ik}\}$.

As a result, the OTR can be rewritten as:

$$\frac{f_{jik}}{C_{ji}} + \frac{f_{jik}}{C_{ik}} + \frac{f_{kij}}{C_{ki}} + \frac{f_{kij}}{C_{ij}} - \frac{u^i_{\{k,j\}}}{C^i_{\{k,j\}}}. \tag{13}$$

We observe that the decrease of OTR within $\Lambda^i_{\{k,j\}}$ is $\frac{u^i_{\{k,j\}}}{C^i_{\{k,j\}}}$. Let $m \in M$ be a maximal clique of G_c. By summing up the OTRs decreased by network coding for all opportunity sets in m, the resulting clique capacity constraint over m follows as:

$$\sum_{(i,j)\in m} \frac{f_{ij}}{C_{ij}} - \sum_{\Lambda^i_{\{k,j\}} \subset m} \frac{u^i_{\{k,j\}}}{C^i_{\{k,j\}}} \leq 1. \tag{14}$$

Inequalities (14) and (7) define the clique capacity constraint with network coding. In our experiments, we set $T^i_{\{k,j\}}$ to its lower bound $\max\{\frac{1}{C_{ki}}, \frac{1}{C_{ij}}\}$, and correspondingly set $C^i_{\{k,j\}}$ to $1/(\frac{1}{C_{ij}} + \frac{1}{C_{ik}} - T^i_{\{k,j\}}) = \min\{C_{ki}, C_{ij}\}$. In practice, the OTR of broadcast $u^i_{\{k,j\}}$ may be greater than the lower bound $\max\left\{\frac{u^i_{\{k,j\}}}{C_{ki}}, \frac{u^i_{\{k,j\}}}{C_{ij}}\right\}$; a higher $C^i_{\{k,j\}}$ can be set accordingly.

2.3.2 | Effects on energy consumption

Network coding can also reduce the energy cost and yield a negative term in the objective function.

Recall, without network coding, the energy consumption to send $u^i_{\{k,j\}}$ separately from node i to node j and node k is $u^i_{\{k,j\}}(\beta_{ij} + \beta_{ik})$. Let $\beta^i_{\{k,j\}} u^i_{\{k,j\}}$ be the cost of broadcasting $u^i_{\{k,j\}}$ by network coding, where the energy cost parameter $\beta^i_{\{k,j\}}$ measures the energy consumption to send a unit packet of $u^i_{\{k,j\}}$ per unit time by broadcasting. Because i is responsible for transmitting the data in $u^i_{\{k,j\}}$, the cost $\beta^i_{\{k,j\}} u^i_{\{k,j\}}$ equals the OTR of broadcasting $u^i_{\{k,j\}}$ at i times β_i, i.e., $\beta^i_{\{k,j\}} u^i_{\{k,j\}} = u^i_{\{k,j\}} T^i_{\{k,j\}} \beta_i$. Dividing by $u^i_{\{k,j\}}$, it follows that $\beta^i_{\{k,j\}} = \beta_i T^i_{\{k,j\}}$.

Therefore, the energy cost of sending flow f_{jik} and f_{kij} is reduced to

$$\begin{aligned} &\beta_{ji} f_{jik} + \beta_{ik}(f_{jik} - u^i_{\{k,j\}}) + \beta_{ki} f_{kij} + \beta_{ij}(f_{kij} - u^i_{\{k,j\}}) + \beta^i_{\{k,j\}} u^i_{\{k,j\}} \\ =&\beta_{ji} f_{jik} + \beta_{ik} f_{jik} + \beta_{ki} f_{kij} + \beta_{ij} f_{kij} - u^i_{\{k,j\}}(\beta_{ik} + \beta_{ij} - \beta^i_{\{k,j\}}). \end{aligned} \tag{15}$$

Denote $\tau^i_{\{k,j\}} = \beta_{ij} + \beta_{ik} - \beta^i_{\{k,j\}}$, so the energy cost saved by network coding is $\tau^i_{\{k,j\}} u^i_{\{k,j\}}$. The energy saving and the increased capacity are coupled according to Section (2.1):

$$\tau^i_{\{k,j\}} = \beta_i \left(\frac{1}{C_{ij}} + \frac{1}{C_{ik}} - T^i_{\{k,j\}}\right) = \beta_i \frac{1}{C^i_{\{k,j\}}}. \tag{16}$$

It follows from the relationship in (2.1) and the inequality (12) that:

$$0 \leq \tau^i_{\{k,j\}} \leq \beta_i \min\left\{\frac{1}{C_{ij}}, \frac{1}{C_{ik}}\right\} \leq \min\left\{\beta_{ij}, \beta_{ik}\right\}. \tag{17}$$

2.4 | Complexity analysis

We model the wUMCFC problem as a UMCF problem with additional network coding and clique capacity constraints. This problem is a generalization of the single-source unsplittable flow problem [20, 33, 34, 40].

wUMCFC problem can be reduced to a single-source unsplittable flow problem by considering a single source demand, where clique sets are singletons, and the coding variables are fixed to zero,

Single-source unsplittable flow problem is $\mathcal{NP}$–hard, hence the wUMCFC problem is also $\mathcal{NP}$–hard.

3 | MATHEMATICAL FORMULATIONS

This Section is dedicated to presenting mathematical programming formulations of the wUMCFC problem: the compact edge-based formulations and the path-based formulation. We first clarify our model notations.

We denote by wUMCF a model derived from the UMCF (2) problem, in which the edge capacity constraints are replaced by clique capacity constraints, and the objective function is the same as the objective function z of UMCF (2). The wUMCF problem is a generalization of the wUMCF problem in the wireless context, but it is not a completed model.

We denote by wUMCFC an extended model of wUMCF that includes network coding related variables and constraints. Moreover, its objective function $z\_$ is obtained by subtracting $z_c \coloneqq \sum_{\Lambda^i_{\{k,j\}} \subset E} \tau^i_{\{k,j\}} u^i_{\{k,j\}}$ from z, where z_c is the energy cost saved by network coding (CSC).

An abbreviation for edge (resp. path) formulation, i.e.„ E# (resp. P), is appended at the end of model notation, where # will be revealed subsequently. For example, wUMCFC-P denotes the path-based formulation of the wUMCFC model.

3.1 | Compact edge-based formulations

In this subsection, we propose and compare two edge-based MILP formulations, the edge balance formulation and the edge linearization formulation. These two edge-based formulations differ by the way they represent flows on incident edges.

3.1.1 | MBQP formulation

We denote the variables of our first MBQP formulation as follows:

Decision variables:

x^{st}_{ij} : binary variable indicating whether demand $st \in D$ is routed on the edge $(i,j) \in E$.

q^{st}_{jik} : fractional variable denoting the quantity of flow of the demand $st \in D$ routed along the incident edges $j \to i \to k$. q^{st}_{jik} should satisfy the quadratic constraint $q^{st}_{jik} = d^{st} x^{st}_{ji} x^{st}_{ik}$.

$u^i_{\{k,j\}}$: fractional coding variable denoting the quantity of two opposite flows along the incident edges $k \to i \to j$ and $j \to i \to k$, which would be encoded at i.

Recall that M is the set of the maximum cliques of the conflict graph G_c, and $\tau^i_{\{k,j\}} u^i_{\{k,j\}}$ ($\Lambda^i_{\{k,j\}} u^i_{\{k,j\}} \subset E$) is the energy cost saved by network coding.

The MBQP formulation, denoted by wUMCFC-EQ, follows as:

$$
\text{wUMCFC-EQ}\begin{cases}
\min z_{-} = \sum\limits_{st\in D}\sum\limits_{(i,j)\in E}\beta_{ij}d^{st}x_{ij}^{st} - \sum\limits_{\Lambda^{i}_{\{k,j\}}\subset E}\tau^{i}_{\{k,j\}}u^{i}_{\{k,j\}} & & (18.0)\\
\sum\limits_{(i,j)\in E}x_{ij}^{st} - \sum\limits_{(j,i)\in E}x_{ji}^{st} = 0, & \forall i\in V-\{s,t\},\ \forall st\in D, & (18.1)\\
\sum\limits_{(s,i)\in E}x_{si}^{st} - \sum\limits_{(i,s)\in E}x_{is}^{st} = 1, & \forall st\in D, & (18.2)\\
\sum\limits_{(i,t)\in E}x_{it}^{st} - \sum\limits_{(t,i)\in E}x_{ti}^{st} = 1, & \forall st\in D, & (18.3)\\
q_{jik}^{st} - d^{st}x_{ji}^{st}x_{ik}^{st} = 0, & \forall (j,i),(i,k)\in E,\ \forall st\in D, & (18.4)\\
u^{i}_{\{k,j\}} - \sum\limits_{st\in D}q_{jik}^{st} \le 0, & \forall \Lambda^{i}_{\{k,j\}}\subset E, & (18.5)\\
u^{i}_{\{k,j\}} - \sum\limits_{st\in D}q_{kij}^{st} \le 0, & \forall \Lambda^{i}_{\{k,j\}}\subset E, & (18.6)\\
\sum\limits_{st\in D}\sum\limits_{(i,j)\in m}\dfrac{d^{st}}{C_{ij}}x_{ij}^{st} - \sum\limits_{\Lambda^{i}_{\{k,j\}}\subset m}\dfrac{u^{i}_{\{k,j\}}}{C^{i}_{\{k,j\}}} \le 1, & \forall m\in M, & (18.7)\\
x_{ij}^{st}\in\{0,1\}, & \forall (i,j)\in E,\ \forall st\in D, & \\
q_{jik}^{st}\in\mathbb{R}_{+}, & \forall (j,i),(i,k)\in E,\ \forall st\in D, & \\
u^{i}_{\{k,j\}}\in\mathbb{R}_{+}, & \forall \Lambda^{i}_{\{k,j\}}\subset E. &
\end{cases}
\tag{18}
$$

Objective function z_{-} (18.0): minimize the energy consumption of data transmissions per unit of time, after removing the energy cost saved by network coding from z in (2.0).

Flow conservation constraints (18.1) to (18.3): flows entering and leaving at each node are balanced.

Incident edge flow constraints (18.4): the flow q_{jik}^{st} on incident edges (j,i) and (i,k) equals d^{st} if the demand st takes its unsplittable path through (j,i) and (i,k), otherwise $q_{jik}^{st}=0$.

Coding opportunity constraints (18.5) and (18.6): the coding variable $u^{i}_{\{k,j\}}$ (cf. inequality (7)) is less or equal to the minimum of the two aggregated opposite flows along $j\to i\to k$ and $k\to i\to j$.

Clique capacity constraint (18.7) with network coding: OTR within a clique set m should be at most 1 (cf. (14)).

The wUMCFC-EQ contains the classical constraints of the UMCF, and additionally the clique capacity constraints and the network coding constraints and variables. The wUMCFC-EQ problem is a non-convex mixed-Boolean quadratic program.

3.1.2 | Edge linearization formulation

In this Section we propose an MILP reformulation of the nonlinear MBQP formulation.

Recall that $q_{jik}^{st} = d^{st}x_{ji}^{st}x_{ik}^{st}$ is a quadratic constraint, and x_{ji}^{st} and x_{ik}^{st} are binaries. $q_{jik}^{st} = d^{st}x_{ji}^{st}x_{ik}^{st}$ admits the standard exact linearization as follows:

$$
\begin{aligned}
q_{jik}^{st} &\ge d^{st}(1 - x_{ji}^{st} - x_{ik}^{st}),\\
q_{jik}^{st} &\le d^{st}x_{ji}^{st},\\
q_{jik}^{st} &\le d^{st}x_{ik}^{st},\\
q_{jik}^{st} &\ge 0.
\end{aligned}
\tag{19}
$$

Edge linearization formulation, denoted by wUMCFC-EL, replaces incident edge flow constraints (18.4) of wUMCFC-EQ with the following linearization constraints:

$$q^{st}_{jik} \geq d^{st}(1 - x^{st}_{ji} - x^{st}_{ik}), \qquad \forall (j,i) \in E,\ \ \forall (i,k) \in E,\ \ \forall st \in D, \tag{20}$$

$$q^{st}_{jik} \leq d^{st} x^{st}_{ji}, \qquad \forall (j,i) \in E,\ \ \forall (i,k) \in E,\ \ \forall st \in D, \tag{21}$$

$$q^{st}_{jik} \leq d^{st} x^{st}_{ik}, \qquad \forall (j,i) \in E,\ \ \forall (i,k) \in E,\ \ \forall st \in D, \tag{22}$$

$$q^{st}_{jik} \geq 0, \qquad \forall (j,i) \in E,\ \ \forall (i,k) \in E,\ \ \forall st \in D. \tag{23}$$

Indeed, the wUMCFC-EL is a standard linearization of wUMCFC-EQ, which is currently employed by the commercial solvers, e.g. CPLEX [14].

3.1.3 | Edge balance formulation

The edge balance formulation uses the balanced property of flows on edges to represent the flows on incident edges, which is a similar property to the balanced property of flows entering and leaving nodes. We denote the edge balance formulation by wUMCFC-EB. More precisely, wUMCFC-EB replaces incident edge flow constraints (18.4) of wUMCFC-EQ with the following edge balance constraints:

$$\begin{aligned} \sum_{(i,k)\in E} q^{st}_{jik} - d^{st} x^{st}_{ji} = 0, \ \ \forall (j,i) \in E, \ \ \forall st \in D, \\ \sum_{(j,i)\in E} q^{st}_{jik} - d^{st} x^{st}_{ik} = 0, \ \ \forall (i,k) \in E, \ \ \forall st \in D. \end{aligned} \tag{24}$$

wUMCFC-EB is an MILP.

We illustrate the usage of subscript/superscript notations for the subsequent part of this work. When we omit subscripts and/or superscripts of some variables, we denote the subset of variables restricted to the rest of subscripts and/or superscripts. For example, x denotes the set of entire flow variables over all superscripts (demands) and subscripts (edges), and x^{st} denotes the set of $st-$flow variables over all subscripts (edges). wUMCFC-EB is a reformulation of wUMCFC-EQ by the following theorem.

Theorem 1 *Let x take binary values satisfying flow conservation constraints (18.1), (18.2) and (18.3) on nodes. Then q satisfy (18.4) if and only if q satisfies* (24)*.*

Proof Assume x take binary values satisfying flow conservation constraints (18.1), (18.2) and (18.3) on nodes, then x is already an UMCF. For any (j,i) and $(i,k) \in E$ and $st \in D$, $q^{st}_{jik} = d^{st} x^{st}_{ji} x^{st}_{ik}$ if and only if q^{st}_{jik} is the amount of st-flow over incident edges (j,i) and (i,k). For each $st \in D$, since x^{st} is an unsplittable flow, the $st-$flow takes a unique path from s to t. Then the latter condition is equivalent to that for any edge $(i,j) \in E$, the $st-$ flow entering (i,j) from its incident edges equals the $st-$ flow leaving (i,j) to its incident edges, which is exactly edge balance constraint (24).

According to the following theorem, the linear relaxation of wUMCFC-EL is not stronger than the linear relaxation of wUMCFC-EB.

Theorem 2 *The optimum value of wUMCFC-EL's linear relaxation is less or equal to that wUMCFC-EB's linear relaxation.*

Proof Let $(\bar{x}, \bar{q}, \bar{u})$ be a feasible solution of the linear relaxation of wUMCFC-EB. Since the $\bar{x}$ of the relaxation represents a fractional MCF, for each $st \in D$, we can partition $\bar{x}^{st}$ and d^{st} to a finite number of unsplittable $st-$flows and corresponding flow values. Let the set of paths for those unsplittable $st-$flows with non-zero flow values be χ^{st}. For $p \in \chi^{st}$, let $I_{ij}^{p} \in \{0,1\}$ ($(i,j) \in E$) be the Bool indicating whether the path p contains the edge (i,j), and let $d^{st,p}$ be the value of the $st-$flow routed by path p. It follows that

$$\begin{aligned} d^{st}\bar{x}_{ij}^{st} &= \sum_{p \in \chi^{st}} d^{st,p} I_{ij}^{p}, \\ d^{st} &= \sum_{p \in \chi^{st}} d^{st,p} \end{aligned} \tag{25}$$

According to the edge balance constraints (24), $\bar{q}_{jik}^{st}$ ((j,i) and $(i,k) \in E$) could be decomposed by unsplittable $st-$flows in χ^{st}. Let $\bar{q}_{jik}^{st,p}$ be the $st-$flow of path p containing edge (j,i) and (i,k), if p does not contain edge (j,i) and (i,k), $\bar{q}_{jik}^{st,p}$ is defined as zero. It follows that

$$\begin{aligned} \bar{q}_{jik}^{st} &= \sum_{p \in \chi^{st}} \bar{q}_{jik}^{st,p}, \\ \frac{\bar{q}_{jik}^{st,p}}{d^{st,p}} &\in \{0,1\}, \\ \frac{\bar{q}_{jik}^{st,p}}{d^{st,p}} &= I_{ji}^{p} I_{ik}^{p}. \end{aligned} \tag{26}$$

As $\frac{\bar{q}_{jik}^{st,p}}{d^{st,p}} = I_{ji}^{p} I_{ik}^{p}$ admits exact the linearization as follows:

$$\begin{aligned} \bar{q}_{jik}^{st,p} &\geq d^{st,p}(1 - I_{ji}^{p} - I_{ik}^{p}), \\ \bar{q}_{jik}^{st,p} &\leq d^{st,p} I_{ji}^{p}, \\ \bar{q}_{jik}^{st,p} &\leq d^{st,p} I_{ik}^{p}, \\ \bar{q}_{jik}^{st,p} &\geq 0. \end{aligned} \tag{27}$$

By summing up four inequalities over $p \in \chi^{st}$, and substitute equations (25), we obtain:

$$\begin{aligned} \bar{q}_{jik}^{st} &\geq d^{st}(1 - \bar{x}_{ji}^{st} - \bar{x}_{ik}^{st}), \\ \bar{q}_{jik}^{st} &\leq d^{st}\bar{x}_{ji}^{st}, \\ q_{jik}^{st} &\leq d^{st}\bar{x}_{ik}^{st}, \\ q_{jik}^{st} &\geq 0. \end{aligned} \tag{28}$$

which are exactly constraints (19) of wUMCFC-EL formulation. Therefore, $(\bar{x}, \bar{q}, \bar{u})$ is also a feasible solution of the linear relaxation of wUMCFC-EL formulation. The feasible set of the linear relaxation of wUMCFC-EL includes that of wUMCFC-EB, so the result follows.

wUMCFC-EL formulation is not a stronger formulation, and the linearization (20) introduces more constraints compared to (24). wUMCFC-EB model is more suitable from the computational view, so we only solve wUMCFC-EB

(18) in our experiments (see Section 6).

3.2 | Path-based formulation

The edge-based formulation cannot be applied to large-scale problems because its number of columns and constraints increases significantly with the network size and the number of demands. We propose a new formulation, using path variables, that facilitates the representation of unsplittable routing and coding constraints. The path-based formulation is a Dantzig-Wolfe decomposition of wUMCFC-EB, where path variables are extreme points of the convex hull of edge variables of the edge balance formulation. Their linear relaxations have the same value, but the sizes of LPs are not the same, and hence times to solve corresponding MILPs are different (see Section 6). The path-based formulation contains an exponential number of path variables w.r.t. the size of the graph, but it can be solved efficiently by the column generation method.

Denote by φ^{st} the set of all simple paths from the source s to the target t. We define the following decision variables used by the path-based formulation:

Decision variables:

y_p^{st} : binary variable indicating whether demand $st \in D$ is routed along the path $p \in \varphi^{st}$.

$u^i_{\{k,j\}}$: fractional coding variable denoting the amounts of two opposite flows along $k \to i \to j$ and $j \to i \to k$, which can be encoded at i.

The path-based MILP formulation, denoted by wUMCFC-P, follows as:

$$
\text{wUMCFC-P}\begin{cases}
\min z_- = \displaystyle\sum_{st\in D}\sum_{p\in\varphi^{st}}\sum_{(i,j)\in p}\beta_{ij}d^{st}y_p^{st} - \sum_{\Lambda^i_{\{k,j\}}\subset E}\tau^i_{\{k,j\}}u^i_{\{k,j\}} & & (29.0)\\
u^i_{\{k,j\}} - \displaystyle\sum_{st\in D}\sum_{p\in\varphi^{st}}\sum_{(k,i),(i,j)\in p} d^{st}y_p^{st} \le 0, & & \\
u^i_{\{k,j\}} - \displaystyle\sum_{st\in D}\sum_{p\in\varphi^{st}}\sum_{(j,i),(i,k)\in p} d^{st}y_p^{st} \le 0, & \forall \Lambda^i_{\{k,j\}}\subset E, & (29.1)\\
\displaystyle\sum_{st\in D}\sum_{p\in\varphi^{st}}\sum_{(i,j)\in p\cap m}\frac{d^{st}}{C_{ij}}y_p^{st} - \sum_{\Lambda^i_{\{k,j\}}\subset m}\frac{u^i_{\{k,j\}}}{C^i_{\{k,j\}}} \le 1, & \forall m\in M, & (29.2)\\
\displaystyle\sum_{p\in\varphi^{st}} y_p^{st} = 1, & \forall st\in D, & (29.3)\\
y_p^{st}\in\{0,1\}, & \forall st\in D,\ \forall p\in\varphi^{st}, & \\
u^i_{\{k,j\}}\in\mathbb{R}_+, & \forall \Lambda^i_{\{k,j\}}\subset E. &
\end{cases}
\tag{29}
$$

Objective function z_- (29.0): energy consumption of data transmissions in the network per unit of time.

Coding opportunity constraints (29.1): every three-hop coding opportunity set induces a pair of constraints, associated with two opposite flows, such that the coding variable $u^i_{\{k,j\}}$ (cf. inequality (7)) is less or equal to the aggregated flows along $k \to i \to j$ and $j \to i \to k$.

Clique capacity constraints (29.2) with network coding: OTR within a clique set m should be at most 1 (cf. (14)).

Unsplittable constraints (29.3): each demand has to be routed by a single path.

The path-based formulation wUMCFC-P has an exponential number of path variables, but it contains less other variables and fewer constraints than edge-based formulations. The numerical experiments in Section 6 show that it can be solved efficiently by the column generation approach.

TABLE 2 Comparison of constraints

	#Flow cons	#Coding opportunity cons	#Clique capacity cons
Path-based formulation	$O(\|D\|)$	$O(\|E\|^2)$	$O(\|M\|)$
Edge-based formulation	$O(\|D\|\|V\|)$	$O(\|E\|\|D\|) + O(\|E\|^2)$	$O(\|M\|)$

Tables (1) and (2) compare sizes of edge and path-based formulations in terms of number of variables and number of constraints.

TABLE 1 Comparison of variables

	Routing vars	Coding vars	Auxiliary vars
Path-based formulation	Exponential	$O(\|E\|^2)$	0
Edge-based formulation	$O(\|E\|\|D\|)$	$O(\|E\|^2)$	$O(\|E\|^2\|D\|)$

4 | COLUMN GENERATION

Column generation [19, 25, 39, 49] proved efficiency in solving large-scale linear programming (LP) problems. Column generation approaches start by solving the initial LP restricted to a small subset of variables and generate new columns dynamically.

We use the column generation approach to solve the LP relaxation of the path-based formulation wUMCFC-P (29). The LP relaxation with full columns is called the master problem. At each iteration, the column generation method solves a restricted master problem (RMP) limited to a subset of active columns. The primal and dual information of the RMP is used to add new columns with a negative reduced cost. The pricing algorithm is developed here to generate columns into the basis of RMP.

Our pricing algorithm consists of two parts. The first part is called reduced cost pricing, and it adds paths with the minimum negative reduced cost to improve the optimality. The second part is called Farkas pricing which identifies and repairs the infeasibility. We construct an extended graph to reduce the pricing problem to the shortest path problem.

4.1 | Reduced cost pricing

When the RMP is solved to optimality and no variables have negative reduced costs, we can conclude the RMP indeed converges to be optimal for the master problem. The optimality condition for the RMP primal problem is derived from the RMP dual problem.

We introduce the dual variables $\gamma_{jik}, \gamma_{kij}, \zeta_m$ and η_{st} associated to the constraints ((29).1), (29.2) and (29.3), respectively.

The dual constraint (reduced cost) on a path $p \in \varphi^{st}$, denoted by $\mathrm{RC}^{st}(p)$, follows as:

$$\mathrm{RC}^{st}(p) \coloneqq \sum_{(k,i),(i,j)\in p} -d^{st}\gamma_{kij} + \sum_{(i,j)\in p} d^{st}\beta_{ij} + \sum_{m\in M}\sum_{(i,j)\in p\cap m} d^{st}\frac{\zeta_m}{C_{ij}} + \eta_{st}. \tag{30}$$

The dual of the LP relaxation of wUMCFC-P follows as:

$$\text{Dual-wUMCFC-P}\begin{cases} \max -\sum_{m\in M}\zeta_m - \sum_{st\in D}\eta_{st} & & (31.0)\\ \gamma_{kij}+\gamma_{jik}-\tau^{i}_{\{k,j\}} - \sum_{m\in M:\Lambda^{i}_{\{k,j\}}\subset m}\frac{\zeta_m}{C^{i}_{\{k,j\}}} = 0, & \forall \Lambda^{i}_{\{k,j\}}\subset E, & (31.1)\\ \mathrm{RC}^{st}(p)\geq 0, & \forall st\in D, \forall p\in\varphi^{st}, & (31.2)\\ \zeta_m\in\mathbb{R}_+, & \forall m\in M, & (31.3)\\ \gamma_{kij},\gamma_{jik}\in\mathbb{R}_+, & \forall \Lambda^{i}_{\{k,j\}}\subset E, & (31.4)\\ \eta_{st}\in\mathbb{R}, & \forall st\in D. & (31.5) \end{cases} \tag{31}$$

Note that the domain of the dual variables γ is originally defined over the opportunity sets $\Lambda \subset E\times E$. We extend this domain to $E\times E$, so that the subsequent analysis is simpler: for $(k,i),(i,j),(j,i),(i,k) \in E$, if $\Lambda^{i}_{\{k,j\}} \not\subset E$, we define $\gamma_{kij}=0$ and $\gamma_{jik}=0$.

Let $\sigma^{st} \subset \varphi^{st}$ ($st \in D$) be the set of active path variables (columns) added into RMP, only constraints (31.2) indexed by σ^{st} are included in the dual RMP.

The pricing problem is decomposed into $|D|$ sub-problems. For each $st \in D$, the corresponding sub-problem checks whether there exists a path p from s to t such that $\mathrm{RC}^{st}(p) < 0$ (a violated dual constraint).

We define the path cost function PC, such that for a path p in the graph G we have:

$$\mathrm{PC}(p) \coloneqq \sum_{(k,i),(i,j)\in p} -\gamma_{kij} + \sum_{(i,j)\in p}\beta_{ij} + \sum_{m\in M}\sum_{(i,j)\in p\cap m}\frac{\zeta_m}{C_{ij}}. \tag{32}$$

Let $p^* = \arg\min_{p\in\varphi^{st}} \mathrm{PC}(p)$, it follows that $\mathrm{RC}^{st}(p^*) = d^{st}\mathrm{PC}(p^*) + \eta_{st} = \min_{p\in\varphi^{st}} \mathrm{RC}^{st}(p)$.

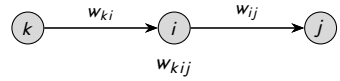
If $\mathrm{RC}^{st}(p^*) < 0$, then we add the column associated to p^* to the RMP.

The pricing problem is reduced to find the path p with the minimum path cost $\mathrm{PC}(p)$. Notice that $\mathrm{PC}(p)$ includes positive and negative costs on incident edges, so the shortest path algorithm in the original graph G is not applicable. However, we have the following polynomial reduction of the pricing problem to the shortest path problem in an extended graph.

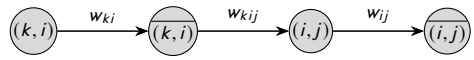
4.1.1 | Extended Graph

Denote by $EG = (H, A)$ the weighted directed extended graph to construct, and let w be the weight function over edges of A. Let $\overline{E} = \left\{\overline{(i,j)} \mid (i,j)\in E\right\}$ be auxiliary edges that are copies of original edges. We build EG as the follows:

1. Let $H = E \cup \overline{E}$.



(a) Edges in original graph G



(b) Corresponding nodes in extended graph EG

FIGURE 3 G and EG

2. If $(k,i) \in E$ and $(i,j) \in E$, then $\left(\overline{(k,i)}, (i,j)\right) \in A$;
If $(i,j) \in E$, then $\left((i,j), \overline{(i,j)}\right) \in A$.
Denote

$$\begin{aligned} A^{\leq} &= \left\{ \left((\overline{(k,i)}, (i,j) \right) \mid (k,i), (i,j) \in E \right\}, \\ A^{\geq} &= \left\{ \left((i,j), \overline{(i,j)} \right) \mid (i,j) \in E \right\}. \end{aligned} \tag{33}$$

Consequently, $A = A^{\leq} \cup A^{\geq}$ is partitioned into two subsets.

3. For $\left(\overline{(k,i)}, (i,j)\right) \in A^{\leq}$, its weight is

$$w\left(\left(\overline{(k,i)}, (i,j)\right)\right) = -\gamma_{kij}, \tag{34}$$

and we abbreviate it by w_{kij}, which is non-positive. We call $A^{\leq}$ the set of non-positive edges.
For $\left((i,j), \overline{(i,j)}\right) \in A^{\geq}$, its weight is

$$w\left(\left((i,j), \overline{(i,j)}\right)\right) = \beta_{ij} + \sum_{m \in M:(i,j) \in m} \frac{\zeta_m}{C_{ij}}, \tag{35}$$

and we abbreviate it by w_{ij}, which is non-negative. We call $A^{\geq}$ the set of non-negative edges.
By construction, non-positive edges are only incident to non-negative edges, and vice versa.

Notice that the size of EG is polynomial in the size of G, its node size and edge size are $O(|E|)$ and $O(|E|^2)$, respectively.

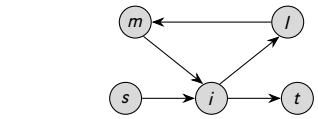
Figure (3) is an example of building an extended graph.

We define the weight $w(p')$ on a path p' in EG as the sum of weights of its edges.

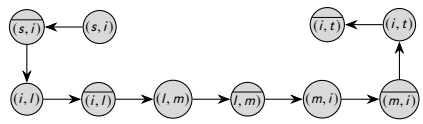
We adopt two representations of the paths. The edge representation is an ordered sequence of edges enclosed by $()$, and the node representation is an ordered sequence of nodes enclosed by $[]$. We call the first (resp. last) node of a path as its source (resp. target) node. We define the following mapping:

Definition Let a path p in G be $(e_1, \ldots, e_n)$ where $e_i \in E$ for $i = 1, \ldots, n$, the path mapping π maps p to a path $\pi(p)$ in EG s.t. $\pi(p) = [e_1, \overline{e_1}, \ldots, e_n, \overline{e_n}]$.

Figure (3) shows that π maps a path $((k,i),(i,j))$ in (3a) to a path $\left[(k,i), \overline{(k,i)}, (i,j), \overline{(i,j)}\right]$ in (3b).



(a) A cyclic path p in G



(b) A simple path $p' = \pi(p)$ in EG

FIGURE 4 π^{-1} does not preserve the acyclicity from EG to G

Therefore, the source and target of p are s and t if and only if the source and target of $\pi(p)$ are (s, i) and $\overline{(j, t)}$ for some $i \in V$ and some $j \in V$.

Recall that φ^{st} is the set of simple paths in G from node $s \in V$ to node $t \in V$.

We define ψ^{st} as the set of simple paths in EG of which source nodes are $(s, i) \in E$ for some $i \in V$ and target nodes are $\overline{(j, t)}$ for some $j \in V$.

Lemma 3 *The following properties are satisfied by the mapping π:*

1. *π is an injection.*
2. *For $p \in \varphi^{st}$, $PC(p) = w(\pi(p))$, where w is the sum of weights of edges of $\pi(p)$.*
3. *$\pi(\varphi^{st}) \subsetneq \psi^{st}$.*

Proof By checking the construction of EG and π, π is injective and $\mathrm{PC}(p) = w(\pi(p))$. By checking the construction of EG, $\pi(\varphi^{st}) \subset \psi^{st}$. To prove the strictness of the inclusion, we give an example in Figure (4).

$$p = ((s, i), (i, l), (l, m), (m, i), (i, t))$$

is cyclic,

$$p' = \pi(p) = \left[(s, i), \overline{(s, i)}, (i, l), \overline{(i, l)}, (l, m), \overline{(l, m)}, (m, i), \overline{(m, i)}, (i, t), \overline{(i, t)}\right]$$

is simple, thus $\pi(\varphi^{st}) \neq \psi^{st}$.

One can check that every path p' in EG with source node in E and target node in $\overline{E}$ is the map of a unique path p in G (see Figure (4)), and the inverse π^{-1} is well defined on those p'. Since π preserves the path cost and is an injection, the pricing problem is reduced to find $\arg\min_{p' \in \pi(\varphi^{st})} w(p')$, and recover the inverse path.

There are two issues to solve:

- Enumerating all paths in $\pi(\varphi^{st})$ is not efficient. Note that ψ^{st} is a set of paths in EG with known sources in E and targets in $\overline{E}$, but it strictly includes $\pi(\varphi^{st})$.

Let $p^* = \arg\min_{p' \in \psi^{st}} w(p')$ be the shortest path in ψ^{st}, it follows that

$$w(p^*) \leq \min_{p' \in \pi(\varphi^{st})} w(p') = \min_{p \in \varphi^{st}} \mathrm{PC}(p). \tag{36}$$

We will prove that $p^* \in \pi(\varphi^{st})$ (equivalently, $\pi^{-1}(p^*)$ is simple), and hence $w(p^*) = \min_{p' \in \pi(\varphi^{st})} w(p')$.

- p^* is the shortest simple path in EG between a known set of sources and targets indicated by ψ^{st}. Notice there are edges with negative weights in EG, we will prove that there are no negative cycles in EG, and hence the shortest simple path problem is solvable in polynomial time.

We first deal with negative weights.

Lemma 4 *For $\Lambda^i_{\{k,j\}} \subset E$, $w_{ik} + w_{kij} + w_{jik} \geq 0$ and $w_{ij} + w_{kij} + w_{jik} \geq 0$.*

Proof From the constraint (31.1) and $\frac{1}{C^i_{\{k,j\}}} \leq \min\left\{\frac{1}{C_{ij}}, \frac{1}{C_{ik}}\right\}$ (cf. (12)), we have

$$\gamma_{kij} + \gamma_{jik} - \tau^i_{\{k,j\}} = \sum_{m \in M: \Lambda^i_{\{k,j\}} \subset m} \frac{\zeta_m}{C^i_{\{k,j\}}} \leq \sum_{m \in M: \Lambda^i_{\{k,j\}} \subset m} \frac{\zeta_m}{C_{ij}} \leq \sum_{m \in M:(i,j) \in m} \frac{\zeta_m}{C_{ij}}. \tag{37}$$

It follows from the definition of w_{kij}, w_{jik} and w_{ij} that

$$\begin{aligned} & -w_{kij} - w_{jik} \\ =& \gamma_{kij} + \gamma_{jik} \\ \leq& \tau^i_{\{k,j\}} + \sum_{m \in M:(i,j) \in m} \frac{\zeta_m}{C_{ij}} \\ \leq& \beta_{ij} + \sum_{m \in M:(i,j) \in m} \frac{\zeta_m}{C_{ij}} \\ =& w_{ij}. \end{aligned} \tag{38}$$

The last inequality follows from (17).

Hence, $w_{ij} + w_{kij} + w_{jik} \geq 0$ holds. The proof for $w_{ik} + w_{kij} + w_{jik} \geq 0$ is similar.

Theorem 5 *Let e_1 and e_2 be two incident edges of EG, and let e_1 be non-positive, and let e_2 be non-negative. Then, $w(e_1) + w(e_2) \geq 0$*

Proof Let $e_1 = \left(\overline{(k,i)}, (i,j)\right)$ be non-positive and $e_2 = \left((i,j), \overline{(i,j)}\right)$ be non-negative. If $\Lambda^i_{\{k,j\}} \subset E$, it follows from Lemma (4) that

$$\begin{aligned} & w(e_1) + w(e_2) \\ =& w_{kij} + w_{ij} \\ \geq& -w_{jik} \\ \geq& 0. \end{aligned} \tag{39}$$

Otherwise, according to the extension of γ in subsection 4.1 and definition of weights, $w(e_1) = w_{kij} = 0$ and $w(e_2) = w_{ij} \geq 0$, so $w(e_1) + w(e_2) \geq 0$.

Corollary 6 *Let r be an arbitrary simple cycle of EG, then $w(r) \geq 0$.*

Proof Denote $r = (e_1, ..., e_t)$, and let its length be t. Because of alternative appearances of non-positive and non-negative edges in the cycle, t is even s.t $t = 2h$ for some integer h. W.l.o.g., we assume e_1 is non-positive.

By Lemma (5), then $w(r) = \sum_{h \in \{1,...,\frac{t}{2}\}} (w(e_{2h-1}) + w(e_{2h})) \geq 0$.

We use the Bellman-Ford algorithm to compute the shortest simple path p^* in the graph EG. The time complexity of the Bellman-Ford algorithm in this graph is $O(|E|^3)$, because the number of nodes and edges of the extended graph are $O(|E|))$ and $O(|E|^2)$, respectively.

The complexity of the pricing algorithm is $O(|D||E|^3)$. Bellman-Ford algorithm can output the shortest length path when there are multiple shortest paths of the same weight, so the pricing algorithm works even if the weight of a cycle is zero. Lemma 7 demonstrates that the removal of cycles of flows would decrease the objective value.

Now, we prove that $\pi^{-1}(p^*)$ is simple based on the perturbation analysis and the optimality of p^*.

Let (y, u) be a feasible solution of LP relaxation of (31), denote by $z_-(y, u)$ the objective value of solution (y, u). If there exists a path p in G such that $y_p > 0$, and p contains a cycle r, we call this solution is cyclic.

Here, we take the edge representation of a path/cycle, and $p \setminus r$ denotes the usual set minus operation. Since the left-hand sides of clique capacity constraints and the objective function are proportional to OTRs, deleting cycles of a path would reduce OTRs on the edges of cycles. As a result, deleting cycles could both decrease the objective and increase the feasibility.

Lemma 7 *Let $(\bar{y}, \bar{u})$ be a cyclic solution such that there exists a path $\dot{p}$ with $\bar{y}_{\dot{p}} > 0$, and $\dot{p}$ contains a cycle $\dot{r}$, denote by $\ddot{p} = \dot{p} \setminus \dot{r}$ the path without the cycle $\dot{r}$. There exists another feasible solution $(\widehat{y}, \widehat{u})$, such that $\widehat{y}_{\dot{p}} = 0$, $\widehat{y}_{\ddot{p}} = \bar{y}_{\dot{p}} + \bar{y}_{\ddot{p}}$, $\widehat{y}_p = \bar{y}_p$, for all $p \neq \dot{p}$ and $\ddot{p}$, and $z_-(\widehat{y}, \widehat{u}) \leq z_-(\bar{y}, \bar{u})$.*

Proof We can assume that $\widehat{u}$ (resp. $\bar{u}$) equals its upper-bound defined by the pair of constraints (31.1) for fixed $\widehat{y}$ (resp. $\bar{y}$). This is because increasing $\widehat{u}$ does not change the feasibility of the solution and decreases the objective value.

Assume $\bar{y}$ follows value assignment of the lemma, and the value of $\bar{u}$ to the upper-bound defined by the pair of constraints (31.1) with $\bar{y}$ fixed.

In contrast to the path $\dot{p}$, the path $\ddot{p}$ does not contain the cycle $\dot{r}$. Denote by $\dot{r}^+ = \dot{r} \cup \{(j, i) \mid (i, j) \in \dot{r}\}$ the union of the cycle $\dot{r}$ and its reverse cycle. For $\Lambda^i_{\{k,j\}} \subset \dot{r}^+ \cap E$, flow on cycle $\dot{r}$ affects coding variable $\widehat{u}^i_{\{k,j\}}$. By deleting the cycle $\dot{r}$, the right hand side of one of the constraint pair (31.1) decreases by at most $\bar{y}_{\dot{p}}$.

Coding variables affected by cycle $\dot{r}$ satisfy the following conditions

$$\bar{u}^i_{\{k,j\}} - \widehat{u}^i_{\{k,j\}} \leq d^{st} \bar{y}_{\dot{p}}, \qquad \forall \Lambda^i_{\{k,j\}} \subset \dot{r}^+ \cap E, \tag{40}$$

$$\bar{u}^i_{\{k,j\}} - \widehat{u}^i_{\{k,j\}} = 0, \qquad \forall \Lambda^i_{\{k,j\}} \not\subset \dot{r}^+ \cap E. \tag{41}$$

We first check that the clique capacity constraint (31.2) is still feasible for each clique set m.

Assume that $\dot{p} \in \varphi^{st}$. If $r \cap m = \varnothing$, then $\dot{p} \cap m = \varnothing$, thus $\widehat{y}_p = \bar{y}_p$ for p s.t. $p \cap m \neq \varnothing$, and $\bar{u}^i_{\{k,j\}} = \widehat{u}^i_{\{k,j\}}$ for all $\Lambda^i_{\{k,j\}} \subset m$, so the left hand side of the clique capacity constraints on m remains the same (feasible). Else if $r \cap m \neq \varnothing$, then for $\Lambda^i_{\{k,j\}} \subset m \cap \dot{r}^+$, associated coding variables decrease from $\widehat{u}^i_{\{k,j\}}$ to $\bar{u}^i_{\{k,j\}}$.

It follows that

$$\begin{aligned}
&\sum_{\Lambda^i_{\{k,j\}}\subset m}\frac{1}{C^i_{\{k,j\}}}\bar{u}^i_{\{k,j\}}-\sum_{\Lambda^i_{\{k,j\}}\subset m}\frac{1}{C^i_{\{k,j\}}}\widehat{u}^i_{\{k,j\}}\\
=&\sum_{\Lambda^i_{\{k,j\}}\subset m\cap\dot{r}^+}\frac{1}{C^i_{\{k,j\}}}\bar{u}^i_{\{k,j\}}-\sum_{\Lambda^i_{\{k,j\}}\subset m\cap\dot{r}^+}\frac{1}{C^i_{\{k,j\}}}\widehat{u}^i_{\{k,j\}}\\
\leq&\sum_{\Lambda^i_{\{k,j\}}\subset m\cap\dot{r}^+}\frac{d^{st}}{C^i_{\{k,j\}}}\bar{y}_{\dot{p}}\\
\leq&\sum_{\Lambda^i_{\{k,j\}}\subset m\cap\dot{r}^+}\min\left\{\frac{d^{st}}{C_{ij}},\frac{d^{st}}{C_{ik}}\right\}\bar{y}_{\dot{p}}\\
\leq&\sum_{(i,j)\subset m\cap r}\frac{d^{st}}{C_{ij}}\bar{y}_{\dot{p}},
\end{aligned}\tag{42}$$

the second inequality follows from (12) and the third inequality follows from an enlarged index set.

We check the left hand side of clique capacity constraint for each m:

$$\begin{aligned}
&\sum_{st\in D}\sum_{p\in\varphi^{st}}\sum_{(i,j)\in p\cap m}\frac{d^{st}}{C_{ij}}\widehat{y}_p-\sum_{\Lambda^i_{\{k,j\}}\subset m}\frac{1}{C^i_{\{k,j\}}}\widehat{u}^i_{\{k,j\}}\\
=&\sum_{st\in D}\sum_{p\in\varphi^{st}}\sum_{(i,j)\in p\cap m}\frac{d^{st}}{C_{ij}}\bar{y}_p-\sum_{(i,j)\in m\cap r}\frac{d^{st}}{C_{ij}}\bar{y}_{\dot{p}}-\sum_{\Lambda^i_{\{k,j\}}\subset m}\frac{1}{C^i_{\{k,j\}}}\widehat{u}^i_{\{k,j\}}\\
\leq&\sum_{st\in D}\sum_{p\in\varphi^{st}}\sum_{(i,j)\in p\cap m}\frac{d^{st}}{C_{ij}}\bar{y}_p-\sum_{\Lambda^i_{\{k,j\}}\subset m}\frac{1}{C^i_{\{k,j\}}}\bar{u}^i_{\{k,j\}}\\
\leq&1,
\end{aligned}\tag{43}$$

The first equation follows from the difference between $\bar{y}$ and $\widehat{y}$ on the cycle $\dot{r}$, and the first inequality follows from (42), and the last inequality follows from the fact that $\bar{y}$ is feasible. Indeed, the left hand side of the clique capacity constraint under the new solution $(\widehat{y},\widehat{u})$ is decreased.

Since the $st-$flow on cycle $\dot{r}$ decreases by $\bar{y}_{\dot{p}}$, and coding variables decrease accordingly, it follows that

$$\begin{aligned}
&z_-(\widehat{y},\widehat{u})-z_-(\bar{y},\bar{u})\\
=&-\sum_{(i,j)\in\dot{r}}\beta_{ij}d^{st}\bar{y}_{\dot{p}}-\sum_{\Lambda^i_{\{k,j\}}\subset\dot{r}^+\cap E}\tau^i_{\{k,j\}}\left(\widehat{u}^i_{\{k,j\}}-\bar{u}^i_{\{k,j\}}\right)\\
\leq&-\sum_{(i,j)\in\dot{r}}\beta_{ij}d^{st}\bar{y}_{\dot{p}}+\sum_{\Lambda^i_{\{k,j\}}\subset\dot{r}^+\cap E}\tau^i_{\{k,j\}}d^{st}\bar{y}_{\dot{p}}\\
=&-\sum_{(i,j)\in\dot{r}:\nexists k,\Lambda^i_{\{k,j\}}\subset\dot{r}^+\cap E}\beta_{ij}d^{st}\bar{y}_{\dot{p}}+\sum_{(i,j)\in\dot{r}:\exists k,\Lambda^i_{\{k,j\}}\subset\dot{r}^+\cap E}(-\beta_{ij}+\tau^i_{\{k,j\}})d^{st}\bar{y}_{\dot{p}}\\
\leq&-\sum_{(i,j)\in\dot{r}:\nexists k,\Lambda^i_{\{k,j\}}\subset\dot{r}^+\cap E}\beta_{ij}d^{st}\bar{y}_{\dot{p}}\\
\leq&0,
\end{aligned}\tag{44}$$

The first equation follows from the difference of objective values on cycle $\dot{r}$, the first inequality follows from (40), and the second equation follows from partition of edges in $\dot{r}$, and the second inequality follows from (17).

Therefore, $z_-(\widehat{y}, \widehat{u}) \leq z_-(\bar{y}, \bar{u})$.

The reduced cost $\mathrm{RC}^{st}(\pi^{-1}(p^*))$ is less or equal to the minimum reduced cost among all paths in φ^{st} according to (36). We conclude that

Theorem 8 *$\pi^{-1}(p^*)$ is a simple path in G.*

Proof By the definition of the minimum reduced cost, among all path variables which admit zero values in RMP, assume a positive perturbation the path variable associated to $\pi^{-1}(p^*)$, such perturbation decreases the objective function.

If $\pi^{-1}(p^*)$ is not simple, by Lemma (7), there are two cases:

- if deleting p^*'s cycles decreases the objective value, it would contradict with the minimum reduced cost of $\pi^{-1}(p^*)$;
- otherwise, deleting cycles produces a solution of the same objective value, i.e.,, the reduced cost of the new simple path is the same as that of $\pi^{-1}(p^*)$. But according to the Bellman-Ford algorithm, among paths of the same shortest cost, only the shortest length one, i.e.,, the new acyclic path would be the output of the algorithm, which contradicts our assumption.

Therefore, $\pi^{-1}(p^*)$ must be simple.

Therefore, there is no need to check whether an inverse optimal path is simple. To summarize, pricing algorithm constructs extended graph EG, for each $st \in D$, finds the shortest path in φ^{st} (Bellman-Ford Algorithm), and adds the inverse path in σ^{st}. If no path is found, the master problem is optimal.

4.2 | Farkas pricing

The Farkas pricing improves the feasibility when the LP relaxation of the RMP problem is infeasible.

(31.2) is the sole constraint that differentiates the dual RMP problem from the dual master problem (Dual- wUMCFC-P (31)), for which the quantifier φ^{st} is replaced by its active subset σ^{st}.

According to Farkas' lemma, the RMP problem is infeasible if and only if the dual RMP problem is unbounded.

The dual RMP problem is unbounded if and only if there exists an improving ray $(\Delta(\eta), \Delta(\gamma), \Delta(\zeta))$ (Farkas certificate) of the dual RMP problem along which the objective function can be improved infinitely.

The dual simplex algorithm could detect this improving ray. $(\Delta(\eta), \Delta(\gamma), \Delta(\zeta))$ is an improving ray for the dual

RMP problem if and only if it satisfies the following conditions:

$$
\begin{cases}
-\sum_{m\in M}\Delta(\zeta)_m - \sum_{st\in D}\Delta(\eta)_{st} > 0, & & (45.0)\\
\Delta(\gamma)_{kij} + \Delta(\gamma)_{jik} - \sum_{m\in M:\Lambda^i_{\{k,j\}}\subset m} \frac{\Delta(\zeta)_m}{C^i_{\{k,j\}}} = 0, & \forall \Lambda^i_{\{k,j\}} \subset E, & (45.1)\\
\mathrm{FC}^{st}(p) \geq 0, & \forall st \in D, \forall p \in \sigma^{st}, & (45.2)\\
\Delta(\zeta)_m \in \mathbb{R}_+, & \forall m \in M, & (45.3)\\
\Delta(\gamma)_{kij}, \Delta(\gamma)_{jik} \in \mathbb{R}_+, & \forall \Lambda^i_{\{k,j\}} \subset E, & (45.4)\\
\Delta(\eta)_{st} \in \mathbb{R}, & \forall st \in D. & (45.5)
\end{cases}
\tag{45}
$$

where the Farkas coefficient is defined as:

$$
\mathrm{FC}^{st}(p) \coloneqq \sum_{(k,i),(i,j)\in p} -d^{st}\Delta(\gamma)_{kij} + \sum_{m\in M}\sum_{(i,j)\in p\cap m} d^{st}\frac{\Delta(\zeta)_m}{C_{ij}} + \Delta(\eta)_{st}. \tag{46}
$$

If $(\Delta(\eta), \Delta(\gamma), \Delta(\zeta))$ is an improving ray for the dual RMP, it might violate the constraints of (45.2) for some $st \in D, p \in \varphi^{st} \setminus \sigma^{st}$.

The pricing algorithm finds, for each demand $st \in D$, a path with the minimum Farkas coefficient.

Each path with negative Farkas coefficient is added to σ^{st}. Adding the corresponding cut repairs unboundedness of the RMP dual problem, and equivalently, adding the corresponding column in the RMP problem repairs its infeasibility.

If the Farkas pricing does not find any path, then $(\Delta(\eta), \Delta(\gamma), \Delta(\zeta))$ certifies that the dual master problem is readily unbounded, so we can conclude that the primal master problem is infeasible.

The left-hand side value of the clique capacity constraint, of a solution with a non-zero cyclic path, is always greater than that value of another solution with cycles deleted. Therefore, the simple path improves feasibility. Thus, like the reduced cost pricing, the inverse of the optimal path is also simple.

In summary, we have the following corollary for the correctness of the pricing algorithm.

Corollary 9 *The pricing algorithm finds a simple path to improve the RMP in polynomial time.*

Since the pricing problem is solved for several iterations, at each node of the branch-and-bound tree, the above corollary explains the efficiency of our approach.

5 | BRANCHING RULE

B&P is a branch-and-bound method in which at each node of the search tree, columns may be added to the LP relaxation. The search tree (B&P tree) implicitly enumerates the candidates and explores the nodes whose LP relaxations are at most the incumbent value.

The path-based formulation wUMCFC-P (29) contains binary path variables. Hence, at every node of the B&P tree, if the column generation converges to a fractional LP solution, then the branching rules enforce the integrality of path variables.

Two branching strategies can be applied: branching on edges or branching on paths.

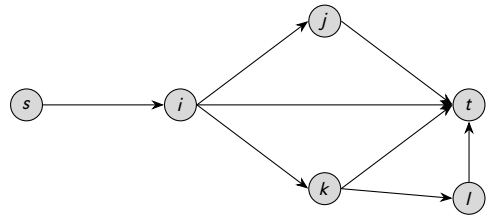


FIGURE 5 Paths diverge at i and k

Branching on paths is a natural way to set binary path variables to 0 or 1. If the variable is fixed to 1, there is no issue. If the variable is fixed to 0, the pricing algorithm might regenerate it. In the worst case, the solver generates a column, fixes it to zero and regenerates it again, and therefore never terminates.

The edge branching method proposed in [8] forbids flows to use certain edges.

For $st \in D$, we say an edge $e \in E$ is $st-$forbidden at a sub-B&P tree if e is forbidden to transmit any $st-$flow in any nodes of the sub-search tree. To forbid an edge e, one can set to zero the existing columns of the paths containing e at the branching node, and forbid to generate any $st-$paths containing e in its sub-search tree. Every node of B&P tree records a set F^{st} of forbidden edges for each $st-$flow.

We choose the branching rule by edges. To be compatible with the proposed pricing algorithm, we can simply delete edges in F^{st} from the original G, and generate the extended graph EG from this subgraph of G. In this way, no $st-$path through a forbidden edge would be generated.

Note that all the properties of the pricing algorithm still hold for the subgraph of G, and all the paths generated are simple in the subgraph and G as well.

Let $\chi^{st} = \{p \mid y_p > 0,\ p \in \varphi^{st}\}$ be the set of $st-$path variables with non-zero values in the LP relaxation. Let T be the sub-graph of G supporting χ^{st}. In T, the in-degree of s is zero, the out-degree of t is zero, and the in-degree and the out-degree of the rest nodes in T are identical. If an st flow variable is not binary, i.e.,, $|\chi^{st}| > 1$, let the first divergence node of T be i.

For example, in Figure (5) there are 4 $st-$paths, 2 divergence nodes i and k, and i is the first divergence node.

In our implementation (1), we choose the flow $\overline{st} = \arg\max_{st \in D:|\chi^{st}|>1} d^{st}$. In this way, the branching rule has a strong impact on the dual bound, and branching at the first divergence node fixes more active paths than branching at other divergence nodes.

Denote by $N_i^{\overline{st}} = \{(i,j) \mid (i,j) \in T\}$ the set of out edges at node i in T, by $O_i^{\overline{st}} = \{(i,j) \mid (i,j) \in E\} \setminus (F^{st} \cup N_i^{\overline{st}})$ the set of non-forbidden out edges of $E \setminus T$ at node i, notice that $N_i^{\overline{st}} \cap O_i^{\overline{st}} = \emptyset$. Since the flow st should be binary, it must take at most one edge of $N_i^{\overline{st}} \cup O_i^{\overline{st}}$ in its unique path, this forms a disjunction for binary branch.

We partition $N_i^{\overline{st}}$ and $O_i^{\overline{st}}$ equally into $F_1^{\overline{st}}$ and $F_2^{\overline{st}}$ which are $\overline{st}-$forbidden edge subsets for two child nodes.

Firstly, $N_i^{\overline{st}}$ is divided into two parts according to $\overline{st}-$flow values on its edges, such that the current $\overline{st}-$flow is partitioned into the two child nodes in a balanced way. Secondly, the partition of $O_i^{\overline{st}}$ is just a random choice.

Algorithm 1: Branching rule

Data:

G: the original graph representing the network, $G = (V, E)$

χ: the set of path variables with non-zero values for each demand.

F: the set of forbidden edge sets at the current B&P node.

Result: Updated forbidden edge sets for two child nodes

$\overline{st} \longleftarrow \arg\max_{st \in D: |\chi^{st}|>1} d^{st}$

Construct the sub-graph T supporting $\chi^{\overline{st}}$

Find the first divergence node i of T

$N_i^{\overline{st}} \longleftarrow \{(i,j) \mid (i,j) \in T\}$

$O_i^{\overline{st}} \longleftarrow \{(i,j) \mid (i,j) \in E\} \setminus (F^{\overline{st}} \cup N_i^{\overline{st}})$

Compute values of $\overline{st}-$flow on the edges of $N_i^{\overline{st}}$ according to $\chi^{\overline{st}}$

Sort $N_i^{\overline{st}}$ in decreasing order by flow values on the edges

$F_1^{\overline{st}}, F_2^{\overline{st}} \longleftarrow \varnothing$

for $(i,j) \in N_i^{\overline{st}}$ **do**

  if $|F_1^{\overline{st}}| < |F_2^{\overline{st}}|$ **then**

    $F_1^{\overline{st}} \longleftarrow F_1^{\overline{st}} \cup \{(i,j)\}$

  else

    $F_2^{\overline{st}} \longleftarrow F_2^{\overline{st}} \cup \{(i,j)\}$

for $(i,j) \in O_i^{\overline{st}}$ **do**

  if $|F_1^{\overline{st}}| < |F_2^{\overline{st}}|$ **then**

    $F_1^{\overline{st}} \longleftarrow F_1^{\overline{st}} \cup \{(i,j)\}$

  else

    $F_2^{\overline{st}} \longleftarrow F_2^{\overline{st}} \cup \{(i,j)\}$

return $F_1^{\overline{st}}, F_2^{\overline{st}}$

6 | COMPUTATIONAL RESULTS

In this Section, we describe our test instances and evaluate the performance of our algorithms. The first experiments compare the performance of the B&P algorithm, for the path-based formulation wUMCFC-P (29), to CPLEX solver for the compact edge balance formulation wUMCFC-EB.

Recall that wUMCF and wUMCFC differ whether the model includes network coding-related variables and constraints. The wUMCF model is obtained by setting $u = 0$ and deleting coding opportunity constraints from wUMCFC. The second experiment solves wUMCF and wUMCFC problems by the B&P algorithm to observe the impact of network coding.

We generate 40 instances and divide them into two classes, low-demand, and high-demand testbeds. Each instance describes the graph G of a wireless network, the costs β of its edges, the capacities C of its edges, the demands D, and the clique set M.

For each testbed, there are 10 sub-classes, and each sub-class contains 2 instances with the same number of nodes and demands. The number of nodes $|V|$ of a wireless network ranges from 30 to 120 by a step of 10, and each testbed contains instances of these 10 distinct sizes. The number of demands, $|D|$, equals $0.4|V|$ and $0.8|V|$ for

low-demand and high-demand testbeds, respectively.

Given the number of nodes and demands, the generation procedure follows as:

1. Generate a random bi-directed geometric graph $G = (V, E)$ of a given number of nodes in the unit square, where the Euclidean radius for linking two nodes is proportional to $\sqrt{\frac{1}{n}}$.
2. For each $i \in V$, sample the node cost β_i from the truncated standard normal distribution s.t $0.8 \leq \beta_i \leq 1.2$.
3. For each $(i, j) \in E$, sample the capacity C_{ij} uniformly from $\{1, 1.1, 1.2, 1.3, 1.4, 1.5, 1.6\}$, and set the cost $\beta_{ij} = \frac{\beta_i}{C_{ij}}$.
4. For each demand $st \in D$, choose a pair of nodes as the source s and the target t randomly, and sample the demand d^{st} from a normal distribution with a mean proportional to $\frac{1}{|D|}$.
5. Construct the conflict graph G_c of G according to 2-dist interference model in Section 2.2.
6. Find the set M of maximal cliques in G_c using *Networkx's* recursive backtracking algorithm [31].

Note that finding all maximal cliques of a graph is an $\mathcal{NP}-$ complete problem. But since the generated geometric graphs are sparse, the number of maximal cliques is not very large and the enumeration of *Networkx's* recursive backtracking algorithm is efficient.

Therefore, using the M clique set computed by the *Networkx's* algorithm, we generate all the clique capacity constraints of our models.

For each coding opportunity set $\Lambda^i_{\{k,j\}} \subset E$, the parameters $\tau^i_{\{k,j\}}$ (for energy saving) and $C^i_{\{k,j\}}$ (for capacity increasing) are computed based on Section 2.3.

We only generate feasible and non-trivial instances. An instance is feasible if it has a solution, and an instance is trivial if its root node LP relaxation is already mixed-integer feasible. For each input, we repeat the above procedure, by adjusting factors of the proportions of steps 1 and 4, until the instance is feasible and non-trivial.

The generated test instances are described in the first column of Table (3) and the first column of Table (4). The first letter indicates whether it belongs to the low or high-demand testbed (L or H), and the following **V.E.D** indicates the numbers of nodes, edges, and demands, respectively.

The B&P algorithm is implemented using SCIP (version 7.0.2) [23] with CPLEX (version 12.10.0) as LP solver. We use CPLEX in the single-thread mode (without parallelism) to solve the compact edge balance formulation. The computing environment has an Intel Core i7-6700K CPU at 4.00GHz and 16GB of RAM under Ubuntu 20.04 system. The time limit is set to 3600 CPU seconds. The relative duality gap tolerance is set to 1e-4.

6.1 | Numerical comparisons of formulations

In this part, we perform the numerical comparison between the path-based formulation wUMCFC-P and the compact edge balance formulation wUMCFC-EB. We evaluate performance metrics of the B&P algorithm, for the path-based formulation, and CPLEX for the compact edge balance formulation.

We implement the pricing algorithm and branching rules for the B&P by SCIP, initial columns are the shortest cost path for each demand.

Tables (3) and (4) report statistics for low and high-demand data sets respectively.

We report the primal bound $\overline{z^*}$, the relative duality gap in percentage, the run time t in CPU seconds, the number of variables, the number of constraints, and the number of nodes. Besides, we also record the pricing time t_P in CPU seconds, the number of pricing calls, and the number of generated paths for the B&P.

Among all 40 instances, the B&P algorithm solves 13 instances to optimality, and finds primal feasible solutions

for all 40 instances; CPLEX solves 10 instances to optimality and finds primal feasible solutions for 23 instances. This result can be explained by the fact that the size of the edge-based formulation is significantly larger than the size of the path-based formulation, especially for large networks and high-demand instances. In the edge-based formulation, the number of constraints grows linearly to the number of demands and the size of the network. While the number of major constraints of the path-based formulation mainly depends on the size of the network. Both the edge-based formulation and the path-based formulation need auxiliary continuous variables to model the coding opportunities, but the edge-based formulation has additional variables q on incident edges. The number of variables of the edge-based formulation is $O(|E|^2|D|)$, and the number of variables of the path-based formulation is $O(|E|^2)$ in addition to the path variables.

Even with the generated columns (paths), the size of the path-based formulation is still smaller than the size of the edge-based formulation. Moreover, for each demand, a small part of the path variables are non-zero and active. Columns generated, at one node in the search tree, are not necessarily included in the LP relaxation of a different node. Even at the same node or at a descendant node, SCIP can remove columns of non-active paths from the LP if they are 0 in the LP relaxation. Therefore, column generation selects columns of active path variables in a smaller size than the generated paths set. Experiments show that the size of the path-based formulation is much smaller than the size of the edge-based formulation.

To compare the speeds of B&P to CPLEX, we compute the shifted geometric means (by 10s) of run times for low-demand, high-demand testbeds, and all instances respectively. The rum time of unsolved instances is taken as the time limit (3600 seconds). As for the B&P algorithm, the mean times for low-demand, high-demand testbeds and all instances are 588.3 seconds, 1365.2 seconds, and 897.1 seconds, respectively. As for CPLEX, the mean times for low-demand, high-demand testbeds and all instances are 1005.9 seconds, 1798.3 seconds, and 1345.4 seconds, respectively. Therefore, B&P is 0.71, 0.31, and 0.5 times faster than CPLEX on low-demand, high-demand testbeds and all instances respectively.

This conclusion is conservative because for the CPLEX side there are many instances with 100% duality gap. We also compare the average duality gaps for instances of which feasible solutions are found by CPLEX. The average duality gap for B&P is 0.62%, and the average duality gap for CPLEX is 1.04%. In summary, B&P outperforms CPLEX in the speed and the quality of solutions.

The two tables show that the pricing time of B&P occupies a significant part of the total running time. Let $\frac{t_p}{t}$ be the ratio of the pricing time over total run time. On average, the ratio is 26%, and it increases with the size of the network and the number of demands. Therefore, a fast shortest path algorithm could speed up B&P.

TABLE 3 low-demand experiments

Instance	SCIP's B&P for wUMCFC-P									CPLEX for wUMCFC-EB					
	$\overline{z^*}$	Gap(%)	t	t_p	#calls	#paths	#vars	#cons	#nodes	$\overline{z^*}$	Gap(%)	t	#vars	#cons	#nodes
L30.152.12	2.72	0.0	1.1	0.1	499	1156	1532	755	95	2.72	0.01	8.08	10909	4562	41
L30.166.12	2.18	0.0	4.8	1.1	1242	2673	3139	940	226	2.18	0.01	2.28	13444	5078	5
L40.240.16	2.26	0.0	3.5	0.5	486	1839	2563	1487	29	2.26	0.01	16.27	28197	9719	0
L40.260.16	2.37	0.0	43.5	9.6	1724	6307	7234	1924	150	2.37	0.01	156.58	34576	10739	46
L50.308.20	3.06	0.01	25.9	6.2	1766	5422	6391	1975	131	3.06	0.0	246.16	47400	15461	40
L50.334.20	2.67	0.0	30.9	7.7	1480	5802	7071	2595	64	2.67	0.01	76.34	59215	16857	6
L60.394.24	4.39	4.85	3600.1	1931.0	23524	70034	71426	2841	1735	4.59	9.92	3600.03	80568	23468	89
L60.342.24	3.87	1.0	3600.2	2296.7	33058	55133	56101	2015	7586	3.84	0.27	3600.02	58671	20109	1119
L70.446.28	4.09	0.57	3600.0	2214.9	30522	74835	76320	3002	2648	4.09	0.57	3600.03	102021	30482	298
L70.380.28	3.96	0.01	478.6	195.9	9299	24732	25769	2136	1610	-	-	3600.02	74036	26123	155
L80.482.32	3.77	0.01	72.8	19.0	2587	8362	9903	3145	187	3.78	0.32	3600.08	123478	37761	90
L80.672.32	2.51	0.52	3600.2	1127.5	5111	31152	34772	7781	131	2.57	3.59	3600.2	263385	54060	0
L90.544.36	6.44	1.83	3600.1	1311.4	25327	43417	45040	3376	5685	6.4	0.58	3600.07	149607	47719	8
L90.552.36	5.97	0.69	3600.1	1682.6	27540	60768	62509	3549	3476	5.97	0.82	3600.05	156861	48240	10
L100.716.40	4.16	1.68	3600.2	1171.6	15791	68773	71573	5681	1065	-	-	3600.08	270651	68860	0
L100.568.40	6.18	1.46	3600.2	1332.3	19217	52610	54258	3463	3327	6.21	1.57	3600.15	167922	54855	0
L110.710.44	5.98	2.02	3600.0	714.6	24847	39220	41461	4623	10531	-	-	3600.18	248479	74770	0
L110.710.44	6.11	1.6	3600.0	942.9	17980	60739	63001	4625	2486	6.26	4.07	3600.12	251989	74860	0
L120.708.48	4.95	0.34	3600.0	946.6	41914	45198	47283	4221	19147	-	-	3600.07	256114	81248	35
L120.698.48	6.28	0.45	3600.1	1144.6	35174	49818	51782	3994	12351	6.27	0.22	3600.08	245289	80517	3

TABLE 4 high-demand experiments

Instance	SCIP's B&P for wUMCFC-P									CPLEX for wUMCFC-EB					
	$\overline{z^*}$	Gap(%)	t	t_p	#calls	#paths	#vars	#cons	#nodes	$\overline{z^*}$	Gap(%)	t	#vars	#cons	#nodes
H30.146.24	2.56	0.0	0.8	0.1	301	880	1222	688	56	2.56	0.01	29.78	20150	8371	16
H30.132.24	3.12	0.01	0.3	0.1	195	580	859	560	45	3.12	0.01	8.22	16700	7564	7
H40.220.32	3.56	0.01	123.6	50.2	5330	12931	13545	1261	962	3.56	0.01	599.12	47152	16505	193
H40.226.32	2.69	0.01	195.3	42.7	8675	14541	15195	1306	3453	2.69	0.01	370.43	49522	16680	284
H50.224.40	4.81	0.13	3600.0	1040.3	126298	25862	26444	1165	110427	4.81	0.21	3600.01	58325	23213	2302
H50.282.40	4.15	0.0	821.7	316.9	15530	37497	38313	1642	3607	-	-	3600.02	78667	26351	146
H60.394.48	4.61	0.69	3600.0	1708.4	27550	52328	53725	2815	6622	-	-	3600.04	158063	43869	23
H60.362.48	3.45	1.25	3600.0	1574.0	16715	52420	53635	2525	2411	-	-	3600.05	137815	40112	3
H70.452.56	4.91	7.53	3600.0	839.7	7596	54355	56049	3543	432	-	-	3600.08	223788	58485	0
H70.464.56	4.02	0.47	3600.0	1593.2	15744	55929	57557	3354	2450	4.02	0.69	3600.15	217070	60690	0
H80.552.64	4.42	2.11	3600.1	931.8	19115	56330	58030	3458	4560	-	-	3600.16	264689	77730	0
H80.484.64	4.54	5.04	3600.0	692.5	21358	39771	41294	3127	7455	-	-	3600.12	236580	71154	0
H90.552.72	5.52	0.86	3600.0	710.7	26380	40918	42670	3523	12196	-	-	3600.12	308703	92601	0
H90.504.72	4.91	0.74	3600.0	777.9	31117	39608	41038	2894	15619	4.92	1.1	3600.12	257387	84783	7
H100.594.80	4.71	1.26	3600.1	1090.4	22752	55539	57394	3747	6329	-	-	3600.1	366192	111618	0
H100.598.80	4.59	1.67	3600.2	1069.0	18056	47139	48945	3741	4337	-	-	3600.14	359387	112133	0
H110.772.88	5.61	3.51	3600.0	815.8	8290	53669	56688	6143	470	-	-	3600.26	630648	156050	0
H110.630.88	7.01	2.57	3600.0	885.1	22104	38323	40105	3619	5209	-	-	3600.13	397273	130245	0
H120.720.96	7.18	2.13	3600.0	586.0	21033	40337	42621	4601	7588	-	-	3600.21	542513	161613	0
H120.740.96	6.25	1.56	3600.0	429.2	13977	38935	41304	4821	4206	-	-	3600.48	562586	165990	0

6.2 | Effects of network coding

According to Section 2.3, network coding has two effects: capacity increasing and energy saving.

The first effect enables more feasible routes. It depends on networks' bottlenecks, capacities and demand values. Thus, it is more useful for large instances where demands are large and capacities are limited.

The second effect yields fewer data transmission in the network. Recall that the saved energy equals the CSC z_c. This effect reduces the energy cost of every feasible route, which depends on the costs of edges.

Recall that wUMCFC is the UMCF model integrating clique capacity constraints and network coding, and wUMCF is the UMCF only with clique capacity constraints.

Since in the previous experiments, B&P algorithm for the path-based formulation outperforms CPLEX for the edge balance formulation, we compare wUMCF and wUMCFC by solving their path-based formulations. The path-based formulation of wUMCF, i.e.,, wUMCF-P, is obtained by removing coding constraints and coding variables from wUMCFC-P (29). In our starting experiment, following instances have solutions under wUMCFC model but have no solutions under wUMCF model:

- low-demand: L30.166.12, L50.308.20, L110.710.44, L120.708.48 and L120.698.48.
- high-demand: H30.132.24, H40.226.32, H50.224.40, H50.282.40, H90.552.72, H90.504.72 and H110.630.88.

Demands of these instances exceed the capacity of the networks, but network coding could increase the capacity. As a result, more feasible routes are possible, and these instances have solutions under the wUMCFC model. These in-

stances demonstrate the first effect of network coding, i.e.,„ the increased capacities, which improves QoS of wireless networks.

Evaluating the energy cost on feasible instances is needed. A binary search method is used to find feasible demands for each of these infeasible instances. Revised instances have a suffix 'R' appended to their labels, and we replace original instances with revised instances in new testbeds. We still set the time limit to 3600 seconds. Although B&P might produce primal solutions with non-zero duality gaps at the end of the time limit, the subsequent experiment shows that network coding reduces the energy cost significantly even with the error induced by the unclosed duality gap.

Recall that the objective function of wUMCF model (denoted by z) is the first term of the objective function of wUMCFC model (denoted by z_-), and their difference $z_c = \sum_{\Lambda^i_{\{k,j\}} \subset E} \tau^i_{\{k,j\}} u^i_{\{k,j\}}$ is the energy cost explicitly saved by network coding (CSC). For each instance, we report the incumbent value $\overline{z^*}$ of wUMCF model (resp. $\overline{z^*_-}$ of wUMCFC), the relative duality gap in percentage, the run time t in CPU seconds, the number of variables and the number of constraints. For wUMCFC, we also record CSC $\overline{z^*_c}$ of the incumbent solution. The results are reported in Tables (5) and (6) for low and high-demand testbeds respectively.

Let z^* and z^*_- be the optimal values of wUMCF and wUMCFC respectively, B&P could compute them with unlimited time and memory. Define by $f = \frac{z^* - z^*_-}{z^*} = 1 - \frac{z^*_-}{z^*}$ the relative energy saved by network coding, and $\hat{f} = 1 - \frac{\overline{z^*_-}}{z^*}$ is the estimator of f.

Given the relative duality gap and the primal bound (incumbent value), we can compute the dual lower bound of the optimal objective value.

Let $\underline{z^*}$ be the lower bound of UMCFC, the error of the estimator follows that

$$f - \hat{f} = \frac{z^* \overline{z^*_-} - \overline{z^*}\, z^*_-}{z^* \overline{z^*}} \geq \frac{\underline{z^*}\, \overline{z^*_-} - \overline{z^*}\, \overline{z^*_-}}{\overline{z^*}\, \underline{z^*}} := \hat{e}, \tag{47}$$

for which $\hat{e}$ gives a lower bound of the error.

The average of relative energy saving is 18.35%, and the average of error lower bound is −0.04%. Therefore, the non-closed duality gap just induces a negligible estimation error of relative energy saving. The network coding saves energy cost significantly.

Capacity increasing could lead to a larger feasible route set, hence the optimal objective value is decreased. Indeed, the CSC contributes to the main part of the energy saving. For each instance, we compute the first term of the objective function of wUMCFC, i.e., $\widetilde{z^*_-} = \overline{z^*_-} + \overline{z^*_c}$, which is energy cost without saving by network coding of the given incumbent routing. Among 62.5% instances, $\widetilde{z^*_-} > \overline{z^*}$, so only can the second effect (CSC) contribute to energy saving in these instances. Among other 37.5% instances, $\overline{z^*} \geq \widetilde{z^*_-} > 0.973\overline{z^*}$, the energy cost without saving ($\widetilde{z^*_-}$) is close to the energy cost of wUMCFC ($\overline{z^*}$), so the first effect (capacity increasing) still has a minor contribution to energy saving. $\frac{\overline{z^*_c}}{\overline{z^*_-}}$ is the ratio of CSC over unreduced energy cost, its average value among all instances is 19.16%. Therefore, we can conclude that CSC has a major contribution to energy saving.

However, there is no free lunch for using network coding, as the numbers of variables and constraints of wUMCFC are much larger than those of wUMCF. As a result, both the solving time and the duality gap of wUMCFC increase. To achieve better performance, wUMCFC requires a faster algorithm, so a tailored B&P algorithm is developed in this work.

TABLE 5 low-demand testbed

Instance	wUMCF					wUMCFC					
	$\overline{z^*}$	t	Gap (%)	#vars	#cons	$\overline{z_-^*}$	$\overline{z_c^*}$	t	Gap (%)	#vars	#cons
L30.152.12	3.0	0.0	0.0	33	27	2.72	0.37	1.1	0.0	1487	755
L30.166.12.R	2.42	0.0	0.01	32	32	1.76	0.34	0.1	0.0	620	940
L40.240.16	2.37	0.0	0.0	55	71	2.26	0.2	3.3	0.0	2445	1487
L40.260.16	2.64	0.2	0.0	175	102	2.37	0.36	42.9	0.01	7101	1924
L50.308.20.R	2.86	0.2	0.0	155	77	2.47	0.3	39.5	0.0	7214	1975
L50.334.20	3.0	0.1	0.0	77	97	2.67	0.41	40.1	0.0	7818	2595
L60.394.24	5.33	51.1	0.01	1856	105	4.39	1.35	3600.0	4.92	66727	2841
L60.342.24	4.69	0.9	0.01	347	127	3.87	0.77	3600.2	1.02	52664	2015
L70.446.28	4.59	0.2	0.01	121	88	4.09	0.6	3600.0	0.58	71244	3002
L70.380.28	4.62	2.9	0.01	481	118	3.96	0.77	502.7	0.01	25769	2136
L80.482.32	4.08	10.2	0.01	869	127	3.77	0.3	74.7	0.01	9903	3145
L80.672.32	2.83	0.0	0.0	32	605	2.51	0.54	3600.5	0.52	34712	7781
L90.544.36	7.57	2.4	0.01	282	202	6.44	1.6	3600.0	1.89	36228	3376
L90.552.36	7.77	3600.0	0.46	5773	139	5.97	1.9	3600.0	0.78	49755	3549
L100.716.40	5.07	1.2	0.0	229	161	4.17	0.98	3600.1	1.99	68123	5681
L100.568.40	7.02	0.8	0.0	186	247	6.18	1.12	3600.0	1.47	51570	3463
L110.710.44.R	8.15	3600.0	0.91	10055	229	5.51	1.98	3600.1	1.59	40143	4623
L110.710.44	7.56	88.9	0.01	718	189	6.11	1.73	3600.0	1.6	61856	4625
L120.708.48.R	5.18	0.6	0.01	220	147	4.13	1.11	3600.0	0.13	31142	4221
L120.698.48.R	6.47	3600.1	0.06	484	162	5.17	1.23	3600.0	0.32	39410	3994

TABLE 6 high-demand testbed

Instance	wUMCF					wUMCFC					
	$\overline{z^*}$	t	Gap (%)	#vars	#cons	$\overline{z^*_-}$	$\overline{z^*_c}$	t	Gap (%)	#vars	#cons
H30.146.24	3.36	0.0	0.0	97	52	2.56	0.79	0.7	0.0	1258	688
H30.132.24.R	2.84	0.6	0.0	223	50	2.51	0.32	0.3	0.01	745	560
H40.220.32	4.01	0.1	0.01	118	97	3.56	0.48	132.8	0.01	14000	1261
H40.226.32.R	2.45	0.2	0.01	101	62	2.1	0.34	34.8	0.01	6671	1306
H50.224.40.R	5.37	102.3	0.01	1332	81	4.51	0.92	1563.0	0.01	15351	1165
H50.282.40.R	5.44	3600.0	0.4	3866	90	3.97	1.01	233.7	0.01	19707	1642
H60.394.48	5.32	33.0	0.01	1511	117	4.61	0.7	3600.0	0.64	59957	2815
H60.362.48	4.04	2.0	0.01	442	191	3.45	0.85	3600.0	1.3	51053	2525
H70.452.56	5.94	3600.0	0.07	1979	267	4.91	1.35	3600.0	7.53	54468	3543
H70.464.56	4.66	0.5	0.01	192	210	4.02	0.71	3600.0	0.48	56404	3354
H80.552.64	5.72	0.6	0.01	271	186	4.42	1.54	3600.2	2.12	56716	3458
H80.484.64	6.24	3600.0	0.31	1933	209	4.54	1.63	3600.0	5.08	36467	3127
H90.552.72.R	6.07	3600.1	0.03	929	163	4.5	1.47	3600.0	0.38	27090	3523
H90.504.72.R	6.68	3600.1	0.17	674	178	4.66	1.94	3600.0	0.84	35466	2894
H100.594.80	5.93	1.1	0.01	235	197	4.71	1.29	3600.0	1.26	56155	3747
H100.598.80	6.09	3.5	0.01	539	289	4.59	1.64	3600.1	1.67	48364	3741
H110.772.88	7.4	2.0	0.01	289	281	5.61	1.78	3600.0	3.51	56031	6143
H110.630.88.R	7.47	1.3	0.01	329	231	5.33	2.36	3600.0	2.08	40768	3619
H120.720.96	9.95	996.2	0.01	1935	225	7.18	2.88	3600.0	2.13	42412	4601
H120.740.96	8.25	1.9	0.01	290	275	6.25	2.26	3600.0	1.56	41935	4821

7 | CONCLUSION

In this article, we propose a mathematical programming approach to address the problem of optimizing the energy cost of data transmission in wireless multi-hop networks. We propose new formulations of the problem that take into account specific technical constraints of wireless communication, such as unsplittable routing, interference and network coding.

A column generation approach is developed as well as a *branch&price* framework to solve this $\mathcal{NP}$-hard and challenging problem. The numerical experiments show that the proposed *branch&price* algorithm outperforms the CPLEX solver both in terms of running time and duality gap. We show also that, for all the instances, network coding reduces energy cost significantly. For hard instances, it enables more feasible routes with lower objective function values. The computational efficiency of the *branch&price* algorithm is improved by adapted branching rules and solution tracking mechanisms.

In the future, we plan to integrate more complex coding schemes and optimize the number of devices turned on in the network see [50]. This integration may increase the complexity of the problem and introduces more binary decision variables in the models. We aim to develop a polyhedral study of this problem, generate new valid inequalities and develop *branch&price&cut* algorithms to solve it.

We also plan to extend this work, from general wireless networks to the special cases of Internet-Of-Things (IoT) deployments. Connecting IoT devices through wireless networks require specific technologies. These technologies induce more technical constraints on routing, resource management and network coding protocols. We are interested in extending our models to integrate the specific constraints of the IoT deployment on multi-hop wireless networks.

The further study of such optimization problems and the development of corresponding decision-support tools will help to offer new services with increased quality of service requirements in future wireless networks deployments.

ACKNOWLEDGMENTS

The authors would like to thank Juan-Antonio Cordero-Fuertes, Claudia D'Ambrosio, and Leo Liberti from Laboratoire d'Informatique de l'Ecole Polytechnique for their highly appreciated help in understanding the optimization problem and their fruitful advice on problem modeling and solving.

References

[1] S. Abdul-Nabi, A. Khalil, P. Mary, and J.F. Hélard, *Efficient network coding solutions for limiting the effect of packet loss*, Eurasip J. Wireless Commun. Networking **2017** (2017), 35.

[2] G.A. Akpakwu, B.J. Silva, G.P. Hancke, and A.M. Abu-Mahfouz, *A Survey on 5G Networks for the Internet of Things: Communication Technologies and Challenges*, IEEE Access **6** (2017), 3619–3647.

[3] N.C. Almeida, R.P. Rolle, E.P. Godoy, P. Ferrari, and E. Sisinni, *Proposal of a Hybrid LoRa Mesh / LoRaWAN Network*, 2020 IEEE International Workshop on Metrology for Industry 4.0 and IoT, MetroInd 4.0 and IoT 2020 - Proceedings, 2020, pp. 702–707.

[4] E. Álvarez-Miranda, V. Cacchiani, T. Dorneth, M. Jünger, F. Liers, A. Lodi, T. Parriani, and D.R. Schmidt, *Models and algorithms for robust network design with several traffic scenarios*, Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), Vol. 7422 LNCS, 2012, pp. 261–272.

[5] F. Alvelos and J.M.V. de Carvalho, *Comparing Branch-and-price Algorithms for the Unsplittable Multicommodity Flow Problem*, International Network Optimization Conference, 2003, pp. 7–12.

[6] A. Atamtürk and D. Rajan, *On splittable and unsplittable flow capacitated network design arc-set polyhedra*, Math. Program., Ser. B **92** (2002), 315–333.

[7] S.A. Baharudin, M.F. Zuhairi, and H. Dao, *Impact of interference on wireless mesh network performance*, ARPN J. Eng. Appl. Sci. **14** (2019), 1389–1395.

[8] C. Barnhart, C.A. Hane, and P.H. Vance, *Using branch-and-price-and-cut to solve origin-destination integer multicommodity flow problems*, Oper. Res. **48** (2000), 318–326.

[9] C. Barnhart, E.L. Johnson, G.L. Nemhauser, M.W. Savelsbergh, and P.H. Vance, *Branch-and-price: Column generation for solving huge integer programs*, Oper. Res. **46** (1998), 316–329.

[10] P.O. Bauguion, C. D'Ambrosio, and L. Liberti, *Maximum concurrent flow with incomplete data*, Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), Vol. 10856 LNCS, 2018, pp. 77–88.

[11] D. B.D. and F. Al-Turjman, *A hybrid secure routing and monitoring mechanism in IoT-based wireless sensor networks*, Ad Hoc Networks **97** (2020), 102022.

[12] W. Ben-Ameur and J. Neto, *Acceleration of cutting-plane and column generation algorithms: Applications to network design*, Networks **49** (2007), 3–17.

[13] O. BenRhaiem and L. Chaari, *Information transmission based on network coding over wireless networks: a survey*, Telecommunication Syst. **65** (2017), 551–565.

[14] C. Bliek1ú, P. Bonami, and A. Lodi, *Solving mixed-integer quadratic programming problems with ibm-cplex: a progress report*, Proceedings of the twenty-sixth RAMP symposium, 2014, pp. 16–17.

[15] P. Chaporkar, K. Kar, and S. Sarkar, *Throughput guarantees through maximal scheduling in wireless networks*, 43rd Ann. Allerton Conference Commun., Control Comput. 2005 **3** (2005), 1557–1567.

[16] X. Cheng, Q. Wang, Q. Wang, and D. Wang, *A high-reliability relay algorithm based on network coding in multi-hop wireless networks*, Wireless Networks **25** (2019), 1557–1566.

[17] J. Cordero, J. Yi, T. Clausen, and E. Baccelli, *Enabling Multihop Communication in Spontaneous Wireless Networks*, ACM SIGCOMM eBook on" Recent Advances Networking **1** (2013), 413–457.

[18] J.A. Cordero, Link-State Routing Optimization for Compound Autonomous Systems in the Internet, Ph.D. thesis, École Polytechnique X, 2011.

[19] G. Desaulniers, J. Desrosiers, and M.M. Solomon, *Column generation* Vol. 5, Springer Science & Business Media, 2005.

[20] Y. Dinitz, N. Garg, and M.X. Goemans, *On the single-source unsplittable flow problem*, Ann. Symp. Foundations Comput. Sci. - Proc. **19** (1998), 290–299.

[21] C. Fragouli, J. Widmer, and J.Y. Le Boudec, *Efficient broadcasting using network coding*, IEEE/ACM Trans. Networking **16** (2008), 450–463.

[22] K. Fysarakis, I. Askoxylakis, O. Soultatos, I. Papaefstathiou, C. Manifavas, and V. Katos, *Which IoT protocol? Comparing standardized approaches over a common M2M application*, 2016 IEEE Global Communications Conference, GLOBECOM 2016 - Proceedings, 2016, pp. 1–7.

[23] G. Gamrath, D. Anderson, K. Bestuzheva, W.K. Chen, L. Eifler, M. Gasse, P. Gemander, A. Gleixner, L. Gottwald, K. Halbig, G. Hendel, C. Hojny, T. Koch, P. Le Bodic, S.J. Maher, F. Matter, M. Miltenberger, E. Mühmer, B. Müller, M.E. Pfetsch, F. Schlösser, F. Serrano, Y. Shinano, C. Tawfik, S. Vigerske, F. Wegscheider, D. Weninger, and J. Witzig, *The SCIP Optimization Suite 7.0*, Technical report 05, Optimization Online, 2020.

[24] B. Gendron, *Revisiting Lagrangian relaxation for network design*, Discr. Appl. Math. **261** (2019), 203–218.

[25] B. Gendron and M. Larose, *Branch-and-price-and-cut for large-scale multicommodity capacitated fixed-charge network design*, EURO J. Comput. Optim. **2** (2014), 55–75.

[26] C. Gomez, P. Salvatella, O. Alonso, and J. Paradells, *Adapting AODV for IEEE 802.15.4 mesh sensor networks: Theoretical discussion and performance evaluation in a real environment*, Proceedings - WoWMoM 2006: 2006 International Symposium on a World of Wireless, Mobile and Multimedia Networks, Vol. 2006, 2006, pp. 159–167.

[27] P. Gope and T. Hwang, *BSN-Care: A Secure IoT-Based Modern Healthcare System Using Body Sensor Network*, IEEE Sensors J. **16** (2016), 1368–1376.

[28] P. Gupta and P.R. Kumar, *The capacity of wireless networks*, IEEE Trans. Informat. Theory **46** (2000), 388–404.

[29] R. Gupta, J. Musacchio, and J. Walrand, *Sufficient rate constraints for QoS flows in ad-hoc networks*, Ad Hoc Networks **5** (2007), 429–443.

[30] S. Haddad-Vanier, C. Gicquel, L. Boukhatem, K. Lazri, and P. Chaignon, *Virtual network functions placement for defense against distributed denial of service attacks*, ICORES 2019 - Proceedings of the 8th International Conference on Operations Research and Enterprise Systems, 2019, pp. 142–150.

[31] A.A. Hagberg, D.A. Schult, and P.J. Swart, *Exploring network structure, dynamics, and function using NetworkX*, Tech. report, Los Alamos National Lab.(LANL), Los Alamos, NM (United States), 2008.

[32] P. Jacquet, P. Mühlethaler, T. Clausen, A. Laouiti, A. Qayyum, and L. Viennot, *Optimized link state routing protocol for ad hoc networks*, Proceedings - IEEE International Multi Topic Conference 2001: Technology for the 21st Century, IEEE INMIC 2001, 2001, pp. 62–68.

[33] J.M. Kleinberg, *Single-source unsplittable flow*, Annual Symposium on Foundations of Computer Science - Proceedings, 1996, pp. 68–77.

[34] J.M. Kleinberg, *Decision algorithms for unsplittable flow and the half-disjoint paths problem*, Conference Proceedings of the Annual ACM Symposium on Theory of Computing, 1998, pp. 530–539.

[35] A. Laube, S. Martin, D. Quadri, and K. Alagha, *Optimal flow aggregation for global energy savings in multi-hop wireless networks*, Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), Vol. 9724, 2016, pp. 124–137.

[36] M. Leitner, I. Ljubić, M. Riedler, and M. Ruthmair, *Exact approaches for network design problems with relays*, INFORMS J. Comput. **31** (2019), 171–192.

[37] S.Y.R. Li, R.W. Yeung, and N. Cai, *Linear network coding*, IEEE Trans. Informat. Theory **49** (2003), 371–381.

[38] W. Lu, Y. Gong, X. Liu, J. Wu, and H. Peng, *Collaborative Energy and Information Transfer in Green Wireless Sensor Networks for Smart Cities*, IEEE Trans. Indus. Informatics **14** (2018), 1585–1593.

[39] M.E. Lübbecke and J. Desrosiers, *Selected topics in column generation*, Oper. Res. **53** (2005), 1007–1023.

[40] M. Martens, F. Salazar, and M. Skutella, *Convex combinations of single source unsplittable flows*, Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), Vol. 4698 LNCS, 2007, pp. 395–406.

[41] M. Mozaffari, W. Saad, M. Bennis, Y.H. Nam, and M. Debbah, *A Tutorial on UAVs for Wireless Networks: Applications, Challenges, and Open Problems*, IEEE Commun. Surveys Tutorials **21** (2019), 2334–2360.

[42] D. Nguyen, T. Tran, T. Nguyen, and B. Bose, *Wireless broadcast using network coding*, IEEE Trans. Vehicular Technology **58** (2009), 914–925.

[43] B. Ni, N. Santhapuri, Z. Zhong, and S. Nelakuditi, *Routing with opportunistically coded exchanges in wireless mesh networks*, 2006 2nd IEEE Workshop on Wireless Mesh Networks, WiMESH 2006, 2006, pp. 157–159.

[44] O.S. Oubbati, M. Atiquzzaman, P. Lorenz, M.H. Tareque, and M.S. Hossain, *Routing in flying Ad Hoc networks: Survey, constraints, and future challenge perspectives*, IEEE Access **7** (2019), 81057–81105.

[45] M. Pelegrín and C. D'Ambrosio, *Aircraft deconfliction via Mathematical Programming: Review and insights*, working paper Mar. 2021.

[46] A. Pessoa, R. Sadykov, E. Uchoa, and F. Vanderbeck, *Automation and combination of linear-programming based stabilization techniques in column generation*, INFORMS J. Comput. **30** (2018), 339–360.

[47] H. Rahimi, A. Zibaeenejad, and A.A. Safavi, *A Novel IoT Architecture based on 5G-IoT and Next Generation Technologies*, 2018 IEEE 9th Annual Information Technology, Electronics and Mobile Communication Conference, IEMCON 2018, 2019, pp. 81–88.

[48] F.S. Shaikh and R. Wismüller, *Routing in multi-hop cellular device-to-device (D2D) networks: A survey*, IEEE Commun. Surveys Tutorials **20** (2018), 2622–2657.

[49] F. Vanderbeck, "*Implementing mixed integer column generation*," *Column Generation*, Springer US, Boston, MA, 2005, pp. 331–358.

[50] S. Vanier, *Column generation for the energy-efficient in multi-hop wireless networks problem*, 16th Cologne-Twente Workshop on Graphs and Combinatorial Optimization, CTW 2018 - Proceedings of the Workshop, 2019, pp. 173–174.

[51] S. Vanier and K. Arnaud, *New partition inequalities for the unsplittable flow problem*, ISCO International Symposium on Combinatorial Optimization, 2018.

[52] F. Wu, T. Wu, and M.R. Yuce, *An internet-of-things (IoT) network system for connected safety and health monitoring applications*, Sensors (Switzerland) **19** (2019), 21.

[53] T. Wu, F. Wu, J.M. Redoute, and M.R. Yuce, *An Autonomous Wireless Body Area Network Implementation Towards IoT Connected Healthcare Applications*, IEEE Access **5** (2017), 11413–11422.

[54] Y. Xin, X. Wang, G. Leus, G. Yue, and J. Jiang, 2010. Interference management in wireless communication systems: Theory and applications.

[55] Y. Yang and R. Kravets, *Contention-aware admission control for ad hoc networks*, IEEE Trans. Mobile Comput. **4** (2005), 363–377.

[56] I. Yaqoob, I.A.T. Hashem, Y. Mehmood, A. Gani, S. Mokhtar, and S. Guizani, *Enabling communication technologies for smart cities*, IEEE Commun. Magazine **55** (2017), 112–120.