RAIRO-Oper. Res. 58 (2024) 185–205
https://doi.org/10.1051/ro/2023167

RAIRO Operations Research
www.rairo-ro.org

EXACT AND HEURISTIC APPROACHES FOR MAXIMIZING FLOWS IN UAV-ENABLED WIRELESS CELLULAR NETWORKS WITH MULTI-HOP BACKHAULS

Mariem Mhiri[1], Mohamed Kais Msakni[2], Mazen O. Hasna[3], Tamer Khattab[3] and Mohamed Haouari[1,*]

Abstract. This paper investigates the problem of data routing in backhaul networks using Unmanned Aerial Vehicles (UAVs) to relay data from Small Cells (SCs) to the core network. The objective is to maximize the total fulfilled demand of data to be routed, while ensuring technical requirements such as hop constraints and edge capacity. The problem is formulated using a compact mixed-integer programming model, which can solve small- and medium-sized topologies. In addition, a fast constructive heuristic based on a maximal tree is developed to solve large-scale topologies, resulting in a significant reduction in CPU time. The quality of the heuristic is evaluated by using column generation for solving the linear programming relaxation of an exponential formulation. The computational study shows the effectiveness and value of the proposed compact model and constructive heuristic for various topology sizes. Furthermore, experiments demonstrate that by keeping the network setup constant and updating the demand vector only, the computational time of the compact model can be drastically reduced for all topology sizes.

Mathematics Subject Classification. 90B18, 90C27.

Received January 30, 2023. Accepted October 11, 2023.

1. Introduction

The next generation of wireless communications systems, namely 5G+, are expected to provide enhanced quality of service, increased privacy and security, higher data rates, broader coverage, ultra-reliability, and lower latency [9, 38]. These advances are not limited to terrestrial networks but also encompass space, air, ground, and underwater communications, to enable continuous and ubiquitous connections across multiple wireless networks. [2, 15]. To achieve these goals, higher frequency bands and lower power consumption are required, which necessitates the integration of more intelligence and automation through various technologies.

Keywords. Bakchaul network, unmanned Aerial Vehicles (UAVs), data routing, mixed-integer programming, heuristic, column generation.

[1] Department of Mechanical and Industrial Engineering, College of Engineering, Qatar University, Doha, Qatar.

[2] Department of Industrial Economics and Technology Management, Norwegian University of Science and Technology, Trondheim, Norway.

[3] Electrical Engineering Department, College of Engineering, Qatar University, Doha, Qatar.

*Corresponding author: mohamed.haouari@qu.edu.qa; mh6368@yahoo.com

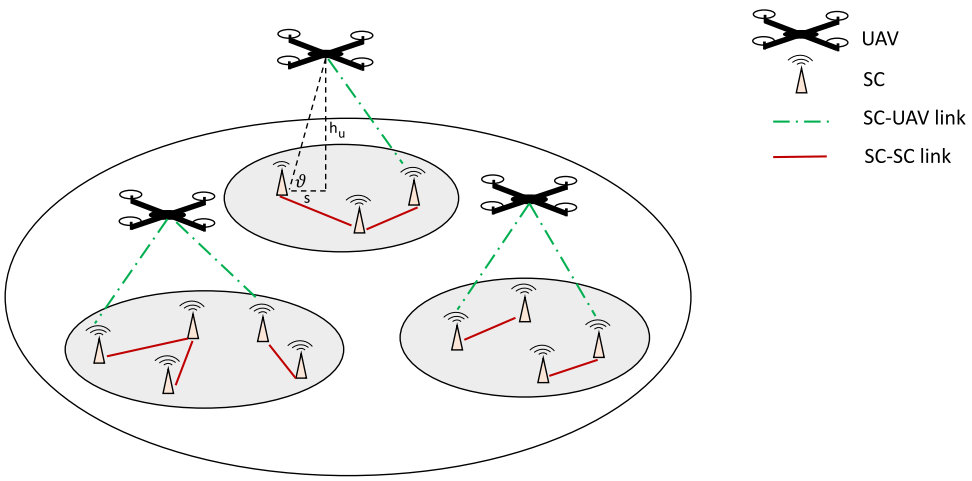

FIGURE 1. Backhaul wireless network: clustered SCs and flying UAVs.

Unmanned Aerial Vehicles (UAVs), also known as drones or remotely operated aircrafts, are one of the technologies expected to play a significant role in the next generation systems due to their low cost and flexible deployment [15]. Indeed, UAVs are no longer limited to military use and have recently been deployed in a wide range of fields, including weather monitoring, forest fire detection, traffic control, cargo transport, emergency search and rescue, and communication relaying [35].

The Third Generation Partnership Project (3GPP) recommends using UAVs in wireless networks as a promising solution to address several challenges associated with the 5G+ systems [24]. Specifically, when used in millimeter-wave (mmWave) frequencies, UAVs can improve capacity and coverage for 5G+ systems by providing on-demand radio access and dramatically enhancing data rates [32]. Contrary to terrestrial communications platforms, UAVs are characterized by their mobility, cost-efficiency, flexibility, and the ability to establish short-distance line-of-sight (LoS) communication links [24]. They can serve as aerial base stations with low latency in specific areas or events where infrastructure is either absent or heavily loaded, such as regions affected by natural disasters, crowded events, emergency situations, rural areas, and network densification for highly populated areas [20, 38].

Given these advantages, there is a great deal of research focused on the use of UAVs in the 5G+ wireless networks. Research efforts are devoted at finding solutions to various related problems, including resource management and energy efficiency, deployment in the presence of terrestrial network, path planning and trajectory optimization, and cellular network connectivity planning (backhauling) [24, 25].

In this paper, we propose a solution to improve data coverage in wireless networks that consist of dense Small Cells (SCs) by leveraging UAVs. Our proposed solution involves a backhauling architecture that utilizes mmWaves, where the SCs transmit data flows to a UAV, which in turn relays the traffic to the core network. The system model assumes the existence of multiple clusters, as illustrated in Figure 1, where each cluster comprises a single UAV that monitors a set of densely distributed SCs.

The problem studied in this paper is the routing of data demand from SCs to the UAV. The objective of the data flow routing is to maximize the sum of the fulfilled demand while satisfying technical constraints, including data flows, and number of links and hops. To meet the demand of next-generation wireless communication systems, this problem, which is NP-hard (as we will show later in Section 4), must be solved efficiently in a short computational time. Similar to previous work in [5], we propose solving the static version of the problem, where the data demand is assumed to be constant. However, we also address the dynamic version of the problem, which requires repeatedly solving the problem as the data demand changes over time.

TABLE 1. Acronyms and meanings.

Acronym	Meaning
3GPP	Third Generation Partnership Project
AtG	Air-to-Ground
BFRP	Backhauling Flows Routing Problem
CG	Column Generation
FSO	Free Space Optics
GtG	Ground-to-Ground
LoS	Line-of-Sight
LP	Linear Programming
MBS	Macro Base Station
MILP	Mixed-integer Linear Programming
mmWave	millimeter-wave
MT	Maximal Tree
NFP	Networked Flying Platforms
NLoS	Non-Line-of-Sight
OR	Operations Research
RD	Routed Demands
RMP	Restricted Master Problem
SC	Small Cell
SNR	Signal-to-Noise-Ratio
SSP	Subset-Sum Problem
TSBS	Terrestrial Small-cell Base Station
UAV	Unmanned Aerial Vehicle
WSN	Wireless Sensor Network

The present paper provides the following contributions:

- First, we propose a novel mixed-integer programming model for the Backhauling Flows Routing Problem (BFRP) that significantly outperforms the model described in [5]. The new model requires less CPU time to provide optimal solutions for small- and medium-sized topologies. Moreover, the computational study shows that a significant portion of the CPU time is spent on constructing the model, while solving it requires much less time. This is relevant to the problem studied because, in practice, once the UAV and SCs are set up and their positions are fixed, only the data demand is constantly changing. Therefore, the constructed mathematical model can be repeatedly re-optimized with the updated demand vector, which can be achieved in a very short computational time (typically in less than 10 seconds) for large-scale instances including several dozens of SCs.
- Second, we develop an iterative heuristic algorithm that efficiently routes the demand, even for large topologies (that cannot be solved using exact model) while requiring a very short CPU time.
- Third, we present a column generation algorithm to obtain a valid upper bound on the optimal objective. For this purpose, we propose a second mathematical model that uses an exponential number of decision variables. The column generation algorithm is then implemented to efficiently solve the linear relaxation of this model. The obtained upper bounds are used to evaluate the heuristic performance for dense networks, *i.e.*, large-sized topologies, when optimal solutions are unknown.
- Fourth, we provide the results of extensive computational experiments that have been conducted on a set of instances of different sizes and perform a sensitivity analysis on the problem parameters.

For the sake of readability, we summarize the acronyms in Table 1.

The remainder of this paper is organized as follows. In Section 2, we cite related works dealing with the data routing problem and highlight the research gaps. In Section 3, we describe the backhaul system. In Section 4, we develop the new optimization formulation. In Section 5, we present the heuristic solution algorithm and describe the column generation algorithm. In Section 6, we analyze the results of the computational study. Finally, we draw our concluding remarks in Section 7.

2. Related works

Data routing in wireless networks is a fundamental and crucial task that plays a critical role in the overall performance of the network. It consists in finding the optimal route/path and corresponding flow between nodes while satisfying various technical constraints. This task is challenging because of the different characteristics and factors that need to be taken into account in the network [1]. In this regard, many authors have studied various data routing problems in wireless sensor networks (WSNs) and have proposed to solve them using different Operations Research (OR) techniques. For instance, Türkoğulları [33] develop a mixed-integer linear programming (MILP) model to determine the optimal locations of sensors and sinks, as well as sensor-to-sink data routes in WSNs, to maximize the network lifetime. The mathematical formulation takes into account certain constraints related to coverage, flow conservation, energy consumption and budget. Due to the complexity of the problem, especially for large-sized instances, an accurate and efficient column generation heuristic is developed. Further works have been conducted in the same vein during the last decade, such as Pinto *et al.* [27] who develop an exact bi-objective approach to minimize the total cost of flows routing and the network bottleneck in a wireless multi-hop network while satisfying link qualities and flow weights. The proposed approach provides a minimal complete set of Pareto-optimal solutions. Flores-Luyo *et al.* [11] study a routing collecting problem of a system of stations that continuously generate information that requires to be delivered to a base station. The problem is then formulated as a vehicle routing problem aiming at maximizing the amount of information collected during a time horizon. The study in [18] investigates four main design issues in wireless sensor networks that affect the distribution of energy and data flow routes. A column generation heuristic for a MILP model that integrates all design issues is proposed and shown to be effective compared to existing works. Kim and Glass [19] consider the problem of coordinating data transmission across a wireless mesh network. A robust solution is provided according to the perfect periodic scheduling. Since the study focuses on routing trees with a chain or a binary tree structure, an optimization algorithm with polynomial time is developed, allowing for energy savings and considerable throughput.

With the captivating technological evolution and, notably, the adoption of drones (UAVs) in 5G+ wireless networks, the OR literature not only focuses on data flow routing problems but has also expanded to finding the optimal location and positioning of these UAVs. Indeed, the use of drones in various applications requires finding their optimal positions. These applications include forest fire risk mitigation [26], delivery and transportation [16], and humanitarian transportation [23, 37]. In addition, UAVs can be deployed in 5G+ wireless networks and operate as backhauling hubs enabling the dense deployment of SCs to achieve higher data rate coverage in wireless networks [8].

In earlier research, particular attention has been paid to using UAVs to maximize the sum data rate. Alzenad *et al.* [8] are among the first to investigate such an objective function using Networked Flying Platforms (NFPs) for traffic transportation between the access and core networks *via* point-to-point Free Space Optics (FSO) links. Their approach considers different weather conditions and proves its high interest, efficiency, and reliability in regions with absent or expensive infrastructure. Given the relevance of this problem, Shah *et al.* [29,30] formulate the association of NFPs and SCs as optimization problems that maximize the sum rate of the overall system, while considering certain practical constraints, including backhaul data rate limit, NFP maximum number of links, bandwidth, and interference. Efficient solutions in terms of total sum rate and execution time are provided using greedy algorithms. Almohamad *et al.* [5–7] design a multihop backhauling network of SCs using UAV hubs. The problem is formulated in [5] as a MILP with the objective of maximizing the total backhaul flow between the SC-UAV and SC-SC connections under links, flows, and hops constraints. Two other MILP

formulations are proposed in [7], and an iterative heuristic algorithm is developed in [6]. Shehzad *et al.* [31] focus on the joint positioning of UAVs and the association of terrestrial small-cell base stations (TSBSs) to maximize the sum rate of the overall network. The optimization approach is based on a genetic algorithm under certain constraints linked to the UAV physical layer, including bandwidth, links, power, interference, altitude, and backhaul data rate. The proposed approach meets the 5G+ requirements and provides efficient results. In another work, Dai *et al.* [10] investigate a multihop UAV-assisted backhaul system by considering two resource allocation schemes, namely the orthogonal frequency scheme and the power control scheme, to maximize the end-to-end throughput. The first scheme only derives the optimal UAV placement, while the second scheme uses an iterative algorithm to optimize the UAV placement and transmit power, with better performance for long distances. In [13], the UAV swarm position is optimized to maximize the MIMO backhaul capacity. Significant sum rate improvements were achieved while requiring only bounded displacements. In [34], the optimization problem aims at maximizing the downlink ergodic sum rate when jointly optimizing UAV location, spectrum resource allocation, and UAV transmit power. The sensitivity study shows significant and accurate results of the approach performance for different parameters of the problem.

Contrary to previous works that deploy UAVs to relay data from SCs, which solve small-size topologies (15 SCs in [7], 25 SCs in [6], 28 SCs in [29], 30 SCs in [30], and 48 SCs in [5]), this paper proposes exact and heuristic approaches that can efficiently solve larger instances with up to 80 SCs per cluster. Interestingly, the proposed exact model can be solved within a very short CPU time once the model is built, making it convenient for responding to dynamic changes in demand when the UAV and SCs are positioned in the network. For large topologies, the proposed heuristic approach affords effective (quality) and efficient (time) solutions. Moreover, the aforementioned research studies assess the quality of their developed methods against either conventional algorithms (*e.g.*, genetic algorithm [31], greedy algorithm [29, 30]) or approximations [34]. This way of comparing is usually justified by the challenging aspect of the problems addressed, *e.g.*, non-convexity of the objective function [34], non-linearity of the constraints [10], or the NP-hardness of the problem [29]. A major drawback of this type of benchmarking is that it does not allow for a rigorous evaluation of the actual performance of the algorithms developed. In contrast, this paper evaluates the performance of the heuristic in terms of effectiveness/efficiency against exact solutions for small- and medium-sized instances and valid upper bounds for large-sized instances.

3. Backhaul system

3.1. Description of the Backhaul model

Due to their mobility and flexibility, UAVs can be utilized in various practical applications, such as for backhauling to assist aerial wireless communications [22]. The use of mmWaves enables UAVs to establish high-data-rate wireless backhaul connections, particularly in congested areas with high data demand [25]. To demonstrate the practical relevance of the backhaul model, we consider a real-world scenario based on an example given in [36]. Toward this end, we assume a large-scale event taking place in a large stadium with a high number of attendees (as for example, a big foot-ball match of the FIFA World Cup). The solution of installing a Macro Base Station (MBS) is not sufficient because the MBS would be overloaded during peak hours when users connect to it directly. An alternative solution to cope with the high demand consists of installing SCs and using UAVs in a static flight over the stadium. Users would be connected to the SCs, and the data flows would be routed according to a multi-hop backhauling scheme. The UAVs would then manage the traffic by relaying the data flows from the SCs to the MBS. This solution has the advantage of being low cost and flexible, for example, if a drone' battery is running low, it can be simply replaced by another drone occupying the same location.

In the sequel, we shall consider a backhaul wireless network architecture similar to the one described in [5]. We assume that the clusters are not adjacent and that each cluster has *preassigned* planned frequencies to avoid interference. As a result, each cluster can be considered independently of the other clusters. The SCs in a given cluster are randomly and densely distributed according to a Matern Cluster Process. The data flow from an

SC can be routed directly to the UAV monitoring the cluster or routed through other SCs until it reaches the UAV. In fact, not all SCs have a direct connection to the UAV due to the phenomenon known as path loss.

3.2. Path loss models

Routing data from the SCs to the UAV involve two distinct types of links: the first type concerns ground-based connections where some SCs are interconnected (SC-SC), and the second type relates to aerial connections between a subset of SCs and the UAV (SC-UAV). These two types of links differ substantially due to the constraints imposed by the network infrastructure. Factors such as network area (urban or rural) and the locations of the transmitter (SC) and the receiver (either another SC or the UAV) have varying effects on each type of link. As shown in the example of Figure 1, the UAV in each cluster is connected to a subset of SCs within the same cluster. These connections represents the SC-UAV links and are illustrated with dashed green lines. The SCs within the same cluster are interconnected through SC-SC links, represented by solid red lines.

In wireless communications, the impact of such environmental factors on signal strength is identified by "Path loss", which is a key factor in the design of wireless communication systems. Path loss refers to the decrease in power density of an electromagnetic wave when it propagates through space, which affects the signal quality and capacity of a system. Path loss is typically modeled as a function of distance and frequency, and is used to calculate the signal-to-noise ratio (SNR) typically measured in decibels (dB) [21]. In the problem at hand, a link between two nodes (a node can be either an SC or a UAV) only exists if the SNR between them meets a certain threshold.

In what follows, we present the path loss models for both SC-SC and SC-UAV links, which will be used to define problem topologies in the computational experiments.

3.2.1. SC-SC links

For an SC-pair separated by a distance Δ, the corresponding link involves the Ground-to-Ground (GtG) path loss whose mean is given by [28]:

$$PL_{GtG} = 20\log_{10}\left(\frac{4\pi f_c^s}{c}\right) + 10\kappa_{\mathrm{GtG}}\log_{10}(\Delta), \tag{1}$$

where f_c^s is the inter-SC carrier frequency, c is the light celerity, and κ_{GtG} is the ground-to-ground (GtG) path loss exponent [7].

The received signal at the SC is then determined by:

$$P_{r,SC}(\mathrm{dB}) = P_t - PL_{GtG}, \tag{2}$$

where P_t is the transmitted power in dB.

Hence, the SNR denoted as Γ and corresponding to the ground-to-ground connection is expressed as follows:

$$\Gamma_{GtG}(\mathrm{dB}) = P_{r,SC} - N_0, \tag{3}$$

where N_0 is the noise in dB.

3.2.2. SC-UAV links

For an SC and a UAV separated by a distance Δ_0, the corresponding link applies the Air-to-Ground (AtG) path loss model. The mean AtG path loss is deduced from the free space path loss and the probabilities of Light of Sight (P(LOS)) and Non Light of Sight (P(NLOS)) multiplied by excessive path loss parameters. It is written as follows [3, 4]:

$$PL_{AtG} = 20\log_{10}\left(\frac{4\pi f_c^u \Delta_0}{c}\right) + P(\mathrm{LOS})\eta_{(\mathrm{LOS})} + P(\mathrm{NLOS})\eta_{(\mathrm{NLOS})}, \tag{4}$$

with:

$$P(\text{LOS}) = \frac{1}{1 + \kappa_{\text{AtG}} \exp(-\nu(\frac{180}{\pi}\vartheta - \kappa_{\text{AtG}}))} \tag{5}$$

$$P(\text{NLOS}) = 1 - P(\text{LOS}) \tag{6}$$

where:

ϑ describes the elevation angle between the SC and the UAV. It is given by $\arctan\left(\frac{h_u}{s}\right)$ where h_u is the UAV elevation and $s = \sqrt{(x_s - x_u)^2 + (y_s - y_u)^2}$ is the horizontal distance between the SC and the UAV ((x_s, y_s) and (x_u, y_u) are the planar coordinates of the SC and the UAV respectively),
f_c^u is the SC-UAV carrier frequency,
Δ_0 is the distance between the SC and the UAV and is equal to $\sqrt{h_u^2 + s^2}$,
$\eta_{(\text{LOS})}$ is the additional loss for the LOS propagation,
$\eta_{(\text{NLOS})}$ is the additional loss for the NLOS propagation.

The received signal at the UAV is then determined by:

$$P_{r,UAV}(\text{dB}) = P_t - PL_{AtG}, \tag{7}$$

Hence, the SNR corresponding to the Air-to-Ground connection is given by:

$$\Gamma_{AtG}(\text{dB}) = P_{r,UAV} - N_0. \tag{8}$$

4. A new compact formulation for the backhauling flows routing problem

The BFRP is defined for a single cluster that is monitored by one UAV. The UAV's role is to relay data from a set of n SCs to the MBS. Within this cluster, both the UAV and SCs constitute the nodes, where the UAV has an index of 0 and an SC is indexed by $1, \ldots, n$. Each SC has a a specific data (demand) that needs to be routed to the UAV. The ultimate objective is to successfully route all demands. However, not all SCs have a direct connection to the UAV, which means that some demand must be routed through intermediary SCs. Additionally, for network efficiency, there are limits on the maximum number of hops (or legs) H between each SC and the UAV, the maximum number of flows F_j incoming to an SC j, and the maximum number of affordable links L_j to an SC and L_0 to the UAV.

These network characteristics are known in advance and collectively constitute the parameters of the BFRP:

n total number of SCs,
d_k demand of the k^{th} SC to be routed to the UAV,
F_j maximum number of flows incoming to the j^{th} SC,
L_j maximum number of affordable links to the j^{th} SC,
L_0 maximum number of affordable links to the UAV,
H maximum number of hops between each SC and the UAV.

The BFRP can be represented using an undirected graph $G = (V, E)$. The set of nodes $V = \{0, 1, \ldots, n\}$ denotes the set of UAV (indexed by 0) and SCs, and the set of edges E represents the possible connections between nodes of V. That is, an edge $e = \{i, j\}$ exists only if the SNR threshold between nodes i and j is met.

From G, we derive a bidirected graph $B = (V, A)$ by replacing an edge $e = \{i, j\} \in G$ with a pair of arcs (i, j) and (j, i), except that arcs going from the UAV, $(0, j)$, are not defined.

The mixed-integer programming model of the BFRP requires the definition of the following sets and decision variables.

Sets:

K	set of SCs, $K = V \setminus \{0\}$,
$\delta_j \subset E$	set of edges in E that are adjacent to $j \in V$,
$\delta_j^+ \subset A$	set of arcs in A that are outgoing from $j \in K$,
$\delta_j^- \subset A$	set of arcs in A that are incoming to $j \in K$.

Decision variables:

x_{ij}^k	binary decision variable that takes the value 1 if arc (i, j) is used to carry the demand of the k^{th} SC, and 0 otherwise,
z^k	binary decision variable that takes the value 1 if the demand of k^{th} SC is carried through the network, and 0 otherwise,
y_e	binary decision variable that takes the value 1 if edge $e = \{i, j\} \in E$, $i < j$, is used, and 0 otherwise.

Then, the model is defined as follows:

$$(P): \text{ Maximize} \sum_{k=1}^{n} d^k z^k \tag{9}$$

subject to:

$$\sum_{i=1}^{n} x_{i0}^k = z^k, \quad k \in K, \tag{10}$$

$$\sum_{i \in \delta_j^-} x_{ij}^k - \sum_{i \in \delta_j^+} x_{ji}^k = 0, \quad k \in K, j \neq 0, k, \tag{11}$$

$$\sum_{j=1}^{n} x_{kj}^k = z^k, \quad k \in K, \tag{12}$$

$$\sum_{(i,j) \in A} x_{ij}^k \leq H, \quad k \in K, \tag{13}$$

$$\sum_{k \in K} \sum_{i \in \delta_j^-} x_{ij}^k \leq F_j, \quad j \in V, \tag{14}$$

$$\sum_{e \in \delta_j} y_e \leq L_j, \quad j \in V, \tag{15}$$

$$\sum_{k \in K} d^k x_{ij}^k + \sum_{k \in K} d^k x_{ji}^k \leq b_e^S y_e, \quad e \in E, (i,j) \in A, \tag{16}$$

$$y_e \in \{0, 1\}, \quad e \in E, \tag{17}$$

$$x_{ij}^k \in \{0, 1\}, \quad (i,j) \in A, k \in K, \tag{18}$$

$$z^k \in \{0, 1\}, \quad k \in K. \tag{19}$$

The objective function (9) maximizes the total fulfilled demand to be routed to the UAV. Constraints (11) specify that the conservation of the flows is satisfied at each cell. If the k^{th} SC is deployed, Constraints (10) enforce to have an incoming flow to the UAV node and Constraints (12) require an outgoing flow from the k^{th} SC. Constraints (13)–(15) limit the enabled number of hops, flows and links in the network, respectively. Constraints (16) ensure that the capacity of an edge e is not exceeded when routing the demand over this edge. Finally, Constraints (17)–(19) identify the binary type of the decision variables.

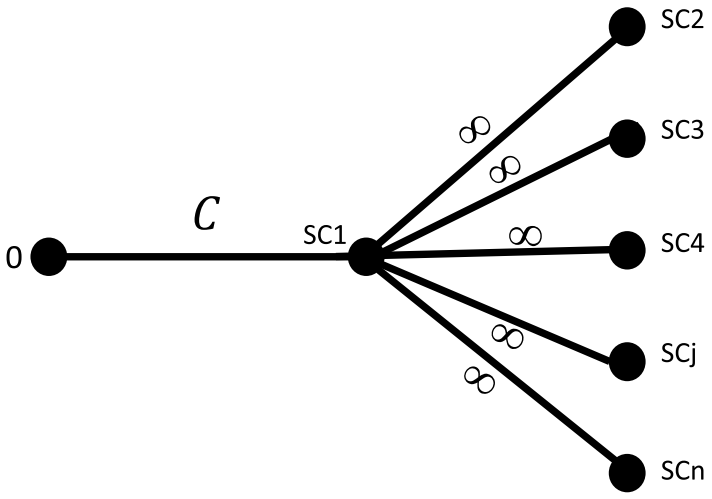

FIGURE 2. Special case of a tree network.

Proposition 4.1. *The BFRP is $\mathcal{NP}$-hard.*

Proof. The BFRP can be reduced to the subset-sum problem (SSP). Consider the special case of a tree network $G = (V, E)$ shown in Figure 2. The root node 0 is adjacent only to node 1. The capacity of this edge is a an integer C. Node 1 is adjacent to nodes $2, \ldots, n$. The capacity of each edge $\{1, j\}$, $(j = 2, \ldots, n)$, is infinite. The demand of each node j is an integer $w_j \leq C$. Clearly, maximizing the flow requires finding the maximum flow through edge $\{0, 1\}$. This problem amounts to solving the following subset-sum problem, where x_j is a binary decision variable that takes value 1 if node j is activated:

$$\text{Maximize} \sum_{j=1}^{n} w_j x_j, \tag{20}$$

subject to:

$$\sum_{j=1}^{n} w_j x_j \leq C, \quad j \in J, \tag{21}$$

$$x_j \in \{0, 1\}, \quad j \in J. \tag{22}$$

Since the SSP problem is $\mathcal{NP}$-hard, the result follows.

□

5. Lower and upper bounds for the BFRP

Our experimental study shows that the proposed mathematical model is effective in solving small- and medium-sized instances, but it becomes computationally intensive for large-scale instances with more than 70 SCs. Given the need for a fast, efficient, and reliable approach to enable demand routing in a wireless multi-hop backhauling network, we propose a heuristic algorithm that aims to provide solutions in a very short CPU time.

5.1. A fast constructive heuristic: a polynomial lower bound

The proposed heuristic is a greedy algorithm that utilizes the edge capacities to compute a maximal tree (MT) with a hop constraint. This results in a sparse connected graph, which is formed by a subset of edges that

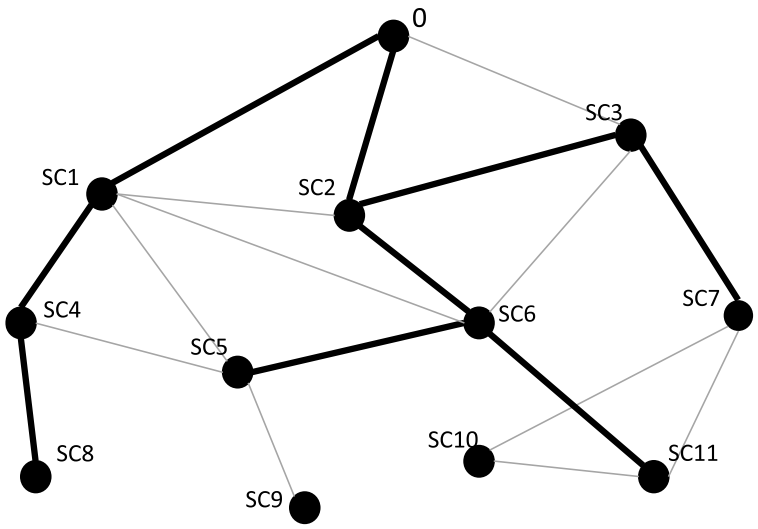

FIGURE 3. An example of a network graph with 12 nodes (including node 0), 3 hops and a feasible tree.

do not contain any cycles or paths with more than H edges. The proposed heuristic starts by connecting node 0 to the L_0 SCs with the largest capacity, forming the initial edges of the MT tree. At each iteration, the tree grows by one edge, ensuring that no cycles are formed and the hop constraint is verified. The edge with the largest capacity is selected and added to the tree if possible, otherwise the next edge with the largest capacity is selected. This process is repeated until no more edge can be added. A pseudo-code for building the maximal tree is provided in Algorithm 1.

Algorithm 1: Maximal Tree with Hop Constraint

Input: Graph $G = (V, E)$ and edges capacities $b_e^S, e \in E$.
Step 0: Initialize the set of the MT nodes S with the UAV ($S = \{0\}$).
Step 1: Connect the UAV (node 0) to the L_0 SCs with the largest capacities.
Update S by adding the L_0 SCs connected to node 0.
Step 2:
while *(S is updated) AND (S ≠ V)* **do**
  Select an edge e that has one endpoint in S and one endpoint in $K \setminus S$ having the largest capacity.
  if *no-cycle AND hop constraint are verified* **then**
    Add edge e to the MT.
    Update S with the new node.
  end
end

Figure 3 shows an example of a network with one UAV node and eleven SC nodes. The bold lines represent the resulting MT, while the thin lines represent the different edges of the network (not selected in the tree). Notably, SCs 9 and 10 cannot be included in the MT because otherwise the number of hops (*i.e.*, number of edges from node 0 to SC 9 (or SC 10)) will exceed the maximum allowed, which is 3.

Once Algorithm 1 builds the MT, the demands can be routed in a nonincreasing order, taking into account the link and flow constraints. For each demand of a node k, the unique route in MT that connects that node to

node 0 is identified and, if feasibility constraints allow, node k's demand is routed along that route. The number of incoming flows Fj and the number of affordable links Lj of any node j in this route are updated along with the edge capacities. If no more demand can be routed due to any of the problem's constraints, there may still be unserved and unmet demands. In this case, a new maximal tree is generated using the residual capacities, and the routing steps are performed again. This process of "building tree and routing" is repeated until all demands are satisfied or no further demand can be routed. The pseudo-code of the heuristic is displayed in Algorithm 2.

Algorithm 2: Heuristic – Demand Routing

Input: Graph $G = (V, E)$, edges capacities $b_e^S, e \in E$ and nodes demands $d_k, k \in K$.
while *(there is a demand that can be routed) AND (the set of non-satisfied demands is not empty)* **do**
  Step 1: Use the edge capacities for computing a maximal tree with hop constraint (Algorithm 1).
  Step 2: Sort the served nodes in non-increasing demand order.
  Step 3:
  for *each k in sorted served nodes* **do**
    if *it is feasible to route the demand of SC k (links and flows constraints are verified)* **then**
      Route its demand along the path that connects it to node 0;
      Update the residual capacities, L_j, and F_j along that path.
    end
  end
  Step 4: With the updated edges capacities, proceed with **Step 1**.
end

The time complexity for the MT is $O(n^2)$ and for the sorting instruction is $O(n \log n)$. These two steps are repeated at most n times. Therefore, the time complexity of the proposed heuristic is $O(n^3)$.

5.2. A column generation-based upper bound

5.2.1. An exponential-size formulation

We present here a second formulation, denoted hereafter by (EF), for the problem under consideration. In contrast to the model of (9)–(19), the number of decision variables involved is exponential. The purpose of proposing such a formulation is to derive a tight and fast linear programming relaxation using a column generation procedure that can be used for deriving an upper bound to evaluate the performance of the proposed heuristic.

First, we define the following sets and decision variables.

Sets:

$\mathcal{P}_k$ set of all possible paths from node k to root node 0, $\mathcal{P}_k = \{P_{1k}, \ldots, P_{|\mathcal{P}_k|,k}\}$,
a_{erk} binary constant that takes value 1 if edge e belongs to path $P_{rk} \in \mathcal{P}_k$, and 0 otherwise,
b_{jrk} binary constant that takes value 1 if node j belongs to path $P_{rk} \in \mathcal{P}_k$, and 0 otherwise.

Decision variables:

w_{rk} binary variable that takes value 1 if path $P_{rk} \in \mathcal{P}_k$ is used for connecting node k to the root node, and 0 otherwise,
y_e binary variable that takes value 1 if edge $e \in E$ is used for carrying a flow to the root node, and 0 otherwise.

Then, (EF) reads as follows.

$$\text{(EF): Maximize} \sum_{k=1}^{n} \sum_{r=1}^{|\mathcal{P}_k|} d_k w_{rk} \tag{23}$$

subject to:

$$\sum_{r=1}^{|\mathcal{P}_k|} w_{rk} \leq 1, \quad k \in K, \qquad (\alpha_k) \qquad (24)$$

$$\sum_{k=1}^{n} \sum_{r=1}^{|\mathcal{P}_k|} b_{jrk} w_{rk} \leq F_j, \quad j \in K, \qquad (\beta_j) \qquad (25)$$

$$\sum_{e \in \delta_j} y_e \leq L_j, \quad j \in V, \qquad (\gamma_j) \qquad (26)$$

$$\sum_{k=1}^{n} \sum_{r=1}^{|\mathcal{P}_k|} d_k a_{erk} w_{rk} \leq b_e^S y_e, \quad e \in E, \qquad (\mu_e) \qquad (27)$$

$$\sum_{r=1}^{|\mathcal{P}_k|} a_{erk} w_{rk} \leq y_e, \quad k \in K, e \in E, \qquad (\theta_{ke}) \qquad (28)$$

$$w_{rk} \in \{0,1\}, \quad k \in K, r = 1, \ldots, |\mathcal{P}_k|, \qquad (29)$$

$$y_e \in \{0,1\}, \quad e \in E. \qquad (30)$$

The objective function (23) maximizes the total flow of the network system and is equivalent to (9). Constraint (24) states that only one path can be selected from each SC node. Constraint (25) limits the flow going through an SC node. Constraint (26) ensures that the number of links of an SC or UAV node is not exceeded. Constraint (27) guarantees that demands transported by each edge is within the capacity. The valid constraint (28) is appended to tighten the model representation. It enforces that for each edge e, at most one path including e can be used for carrying the demand of cell k. Finally, the decision variables are defined in (29)–(30).

5.2.2. The pricing problem

The linear programming (LP) relaxation of the (EF) can be solved using a column generation (CG) algorithm [14]. The master problem of the CG is (EF) with continuous decision variables. First, the model is constructed with an initial limited number of variables to obtain the restricted master problem (RMP). This can be done by randomly selecting a simple route that goes from any SC to the root node. After solving the RMP, the dual vectors α, β, γ, μ, and θ associated to Constraints (24), (25), (26), (27), (28), respectively, are transferred to the pricing problem. The role of the pricing problem is to check the optimally of the RMP. If a non-positive reduced cost is obtained, then the solution of the RMP is optimal. Otherwise, the new column (*i.e.*, route) generated by the pricing problem is introduced to the RMP. This process is repeated until the optimality is proven.

The reduced cost $\tilde{c}_{rk}$ of route r for SC node k is defined in (31).

$$\tilde{c}_{rk} = d_k - \alpha_k - \left(\sum_{j=1}^{n} \beta_j b_{jrk} + \sum_{e \in E} \mu_e d_k a_{erk} + \sum_{e \in E} \theta_{ke} a_{erk} \right). \qquad (31)$$

The pricing problem generates a new variable that maximizes (31), which can be done by finding a variable w_{rk} such that

$$\sum_{j=1}^{n} \beta_j b_{jrk} + \sum_{e \in E} \mu_e d_k a_{erk} + \sum_{e \in E} \theta_{ke} a_{erk}$$

is minimum.

To solve the pricing problem for node k, we use the bi-directed graph B defined in Section 4 and assign to an arc (i, j) the following cost λ_{ij}:

$$\lambda_{ij} = \beta_i + d_k \mu_e + \theta_{ke} a_{erk}.$$

Then, a path with maximum reduced cost can be obtained by solving the following model.

$$Z_k = \min \sum_{(i,j)\in A} \lambda_{ij}\varphi_{ij} \tag{32}$$

subject to:

$$\sum_{i:(i,j)\in A} \varphi_{ij} - \sum_{i:(j,i)\in A} \varphi_{ji} = \begin{cases} 1, i = k \\ 0, i \neq 0, k \\ -1, i = 0 \end{cases}, \quad j \in K \tag{33}$$

$$\sum_{(i,j)\in A} \varphi_{ij} \leq H \tag{34}$$

$$\varphi_{ij} \in \{0,1\}, \quad (i,j) \in A. \tag{35}$$

The problem defined in (32)–(35) is to find the path between node k and root node with minimum cost, while also ensuring that the maximum number of selected arcs does not exceed H. This problem is a resource constrained shortest path problem where the resource consumption along each arc is unitary and the resource capacity is H. It can be efficiently solved using a labeling algorithm [14]. In our computational study, we found that the LP relaxation of Model (EF) can be quickly solved for large-scale instances while the LP relaxation of Model (9)–(19) requires an excessive CPU time.

6. Computational experiments

The aim of this section is to assess the empirical performance of the proposed compact model and heuristic. First, we provide numerical results regarding the performance of Model (P) *versus* the one previously developed in [5]. The comparison focuses on the CPU time required to find an optimal solution. Second, we study the heuristic performance in terms of routed demands compared to the results of Model (P) when it is possible to provide optimal solutions, *i.e.*, for small- and medium-sized instances. For large-scale instances, we provide the gap with respect to the column generation-based upper bound. The CPU time is also taken into account to assess the efficiency of the heuristic.

The small-sized instance set consists of 20, 25, 30, 35, and 40 SCs, each having 20 instances. The medium-sized instances have five instances having 50 and 60 SCs, whereas the large-sized instances have 20 instances with 70 and 80 SCs. The SCs are randomly distributed in a squared region of length $l = 4$ km according to a Matern Cluster Process [12] with SCs density $\bar{c} = 10$ SC/km^2 and cluster density $\lambda_p = 0.3$ Cluster/km^2. The radius of the cluster is $R = 0.5$ km. This process is implemented as in [17], and the UAV location is determined with the k-means clustering method with $k = 1$. We assume that L_0 is set to 12 for $n = 20, .., 40$, to 16 for $n = 50, 60$, and to 20 for $n = 70, 80$. The demands d_k are randomly generated between 30 and 320 Mbps and $(H, F, L) = (5, 10, 7)$. The SNRs thresholds of the GtG and AtG connections are denoted by $\Gamma^{\text{th}}_{\text{GtG}}$ and $\Gamma^{\text{th}}_{\text{AtG}}$ respectively. The capacity of the link between nodes i and j is denoted by $b^S_{\{i,j\}}$ and determined with the Shannon channel capacity equation:

$$b^S_{\{i,j\}} = B_{ij} \log_2(1 + \Gamma_{ij}), \tag{36}$$

where B_{ij} and Γ_{ij} are the bandwidth of the channel and the SNR between SC nodes $(i = 1, \ldots, n)$, and SC or UAV nodes $(j = 0, 1, \ldots, n)$, respectively. The values of the remaining parameters are given in Table 2.

The heuristic was implemented in Python Version 3.8.7. The MILP models were coded in CPLEX Python API and solved using the commercial solver CPLEX 20.10. The tests were carried on a Windows machine with a 11th Gen Intel(R) Core(TM) i7-1165G7 @ 2.80GHz processor and 16GB of RAM.

TABLE 2. Values of the different problem parameters used to generate the BFRP instances.

Parameter	Description	Value
N_0	Noise	10^{-16} W
P_t	Transmitted power	5 W
$\Gamma^{\text{th}}_{\text{GtG}}$	GtG SNR threshold	5 dB
$\Gamma^{\text{th}}_{\text{AtG}}$	AtG SNR threshold	5 dB
κ_{GtG}	GtG PL exponent	2.9
κ_{AtG}	AtG PL exponent	9.61
ν_{AtG}	AtG PL exponent	0.16
η_{LOS}	Excessive path loss	1 dB
η_{NLOS}	Excessive path loss	20 dB
f_c^s	SC-SC carrier frequency	73 GHz
f_c^u	SC-UAV carrier frequency	60 GHz
B_{SS}	SC-SC channel bandwidth	500 MHz
B_{SU}	SC-UAV channel bandwidth	750 MHz
h_u	UAV elevation	100 m

TABLE 3. Execution time of Model (P) *versus* the MILP in [5].

Number of instances	**10**			**5**		**5**	
n	**20**	**25**	**30**	**35**	**40**	**50**	**60**
CPU Time (Model (P)) (sec)	5.79	22.95	65.67	197.67	255.36	1094.51	2641.76
CPU Time (Model of [5]) (sec)	35.32	159.94	455.98	1345.02	2048.41	> 2100	> 4200
Time reduction (%)	**83.61**	**85.65**	**85.60**	**85.30**	**87.53**	-	-

6.1. Performance of Model (P)

In the first experiment, we compare the performance of Model (P) with the one proposed in [5]. Table 3 gives the average CPU time for both models for the small- and medium-sized instances. As it can be seen from Table 3, the proposed model exhibits a superior performance. Indeed, we observe that the CPU time has been drastically reduced by Model (P) by 85.5% on average for small instances. Besides, our proposed model is able to prove the optimality for all medium-sized instances, which is not the case for the model described in [5].

In practical and real-life considerations, once the SCs and the UAV are set up, their positions do not vary and are fixed over time. This means that the SNR calculations used to establish links between the different nodes are already completed, and only demand flows become dynamic and change according to user requests. This property can be exploited to enhance the solving process of Model (P). Once the model is built by the solver for the first instance, it can be reused for other instances as the SCs and UAV locations remain the same and only the demand changes. Hence, it suffices to keep the model built by CPLEX in memory and to call the appropriate API to modify the model with the new demand vector. The second experiment includes this modification and is performed for small and medium topologies, for which the Model (P) requires considerable CPU time to provide exact solutions.

Table 4 shows that a significant amount of time is required to build and solve the mathematical model for the first time, *i.e.*, 1329 seconds for $n = 50$ and 2009 seconds for $n = 60$. However, it is interesting to note that the time required to solve the other instances was drastically reduced for all topology sizes, *i.e.*, less than 1 second

TABLE 4. Efficiency of Model (P) in terms of CPU time (in sec) for small and medium topologies when only the demands are modified.

n	**20**	**25**	**30**	**35**	**40**	**50**	**60**
Instance 1	**6.87**	**15.60**	**26.92**	**85.86**	**326.84**	**1329.44**	**2009.27**
Instance 2	0.16	0.25	0.28	0.74	1.78	5.81	7.58
Instance 3	0.17	0.25	0.27	0.84	1.87	5.29	4.92
Instance 4	0.19	0.26	0.28	0.81	2.18	5.30	4.24
Instance 5	0.18	0.25	0.3	1.15	1.38	4.84	6.59
Instance 6	0.18	0.26	0.27	1.15	1.60	4.17	4.49
Instance 7	0.17	0.25	0.28	1.13	1.71	5.00	5.56
Instance 8	0.18	0.25	0.27	1.26	1.59	4.16	5.73
Instance 9	0.17	0.24	0.29	1.15	1.44	4.94	4.20
Instance 10	0.19	0.23	0.28	1.08	1.90	6.71	4.69
Instance 11	0.16	0.24	0.28	1.28	1.91	4.24	4.41
Instance 12	0.18	0.25	0.29	1.11	2.15	4.70	5.49
Instance 13	0.18	0.24	0.27	0.92	1.70	4.54	4.28
Instance 14	0.16	0.24	0.33	1.20	2.04	4.80	4.20
Instance 15	0.14	0.23	0.26	0.91	1.55	5.76	6.27
Instance 16	0.19	0.25	0.26	1.13	2.08	5.42	4.49
Instance 17	0.17	0.23	0.32	0.90	1.46	4.47	6.08
Instance 18	0.19	0.25	0.37	0.84	1.78	4.27	7.33
Instance 19	0.19	0.24	0.28	0.92	2.05	5.92	5.59
Instance 20	0.18	0.25	0.27	0.97	2.05	4.42	5.71
Instance 21	0.17	0.23	0.28	1.22	1.86	4.86	5.09
Avg. over Instances 2-21	**0.18**	**0.24**	**0.29**	**1.03**	**1.80**	**4.98**	**5.35**
Time reduction (%)	**97.38**	**98.46**	**98.92**	**98.80**	**99.44**	**99.62**	**99.73**

for $n \leq 35$ and 5.4 seconds for $n = 60$. The improvement of CPU time is more than 99% for medium-sized instances.

6.2. Performance of the proposed heuristic

In the third experiment, we conduct a performance study of the developed heuristic in terms of routed demands (RD) as well as CPU time. Unlike the case of small- and medium-sized topologies, the solver failed to provide any solution after running Model (P) for several hours for large-scale instances having more than 70 SCs. Hence, we compare the heuristic to the optimal solution of the BFRP for small- and medium-sized instances using both performance metrics. For large-scale instances, the heuristic is compared to the developed upper bound in terms of routed demands only.

6.2.1. Routed demands

The routed demands obtained by the heuristic are compared to the optimal solutions, when possible. The aim is to measure the solution quality of the proposed heuristic. As shown in Table 3, the optimal values can only be obtained for topologies with small and medium sizes. Therefore, for large instances, we refer to the upper bound solutions obtained by the column generation algorithm (presented in Sect. 5.2). The solution quality of the heuristic is reported in Table 5. The line "$= 0\%$" indicates the percentage of times where the heuristic solution is the same as the optimal solution or the upper bound value. We mention here that if the heuristic solution is the same as the upper bound, it means that this solution is optimal. In the same way, the lines "$\leq 5\%$" and "$\leq 10\%$" refer to the percentage of times where the heuristic gap is less than 5% and 10%, respectively. Finally,

TABLE 5. Gap of the heuristic compared to (P) for small and medium topologies, and upper bound (UB) for large topologies.

Compared to	**Optimal Solution**							**UB**	
Number of instances	**20**					**5**		**20**	
n	**20**	**25**	**30**	**35**	**40**	**50**	**60**	**70**	**80**
=**0**%	75	70	80	40	65	0	0	20	20
≤**5**%	75	90	85	55	95	60	80	65	80
≤**10**%	85	95	85	80	100	80	100	90	95
RD gap (%)	**4.35**	**2.32**	**3.04**	**5.17**	**1.02**	**5.28**	**3.63**	**3.93**	**2.59**

TABLE 6. CPU time (in sec.) of the heuristic *versus* Model (P) for the small and medium topologies.

Number of instances	**20**					**5**	
n	**20**	**25**	**30**	**35**	**40**	**50**	**60**
Model (P)	5.80	27.89	55.74	148.51	288.68	1094.51	2641.76
Heuristic	0.02	0.08	0.14	0.54	0.84	3.74	8.59
Time Reduction (%)	**99.66**	**99.70**	**99.75**	**99.64**	**99.71**	**99.66**	**99.67**

the last line, "*RD gap(%)*" gives the gap (in terms of percentage) of the obtained solution from the optimal value or the upper bound.

The solution quality in terms of routes demands is very high for the heuristic, *i.e.*, more than 90% of the medium and large problems have a gap less than 10%. Table 5 also shows that, in a few cases, the gap is between 5% and 10% and, in the majority of cases, the heuristic gap is less than 5% for all topologies. In addition, the heuristic is able to provide the optimal solutions for several instances, especially those with 20 to 30 SCs. Besides, it is proven that four instances of each large size are optimal since the heuristic solutions are the same as the upper bounds. The high performance of the heuristic is also confirmed by the last line of Table 5, which gives the relative gap to either the optimal solution or upper bound. The gap is less than 5.2% for all topologies and is about 3.5% on average.

6.2.2. CPU time

We consider the computational time as a second criterion for evaluating the heuristic performance. Table 6 reports the average CPU time for the heuristic and compares it to the solution time of Model (P). Since the optimal values cannot be obtained for large problems, the comparison is provided for small and medium sizes. Table 6 shows the significant difference between the two approaches, *i.e.*, the time reduction of the heuristic is more than 99.6% for the shown topology sizes.

We conduct further computational experiments with the heuristic for large instances, and we find that its computation time is relatively significant. Indeed, we observe that instances having $n = 70$ and $n = 80$ SCs require 34.4 and 53.3 seconds, respectively, to be solved. The highest time is recorded for one instance with $n = 80$, which is solved in 188.46 seconds. As we operate in a static framework, the solution time of the heuristic can be further improved by reducing the computational effort in the same way as we did previously with the exact method. For a given topology with fixed locations, the maximal tree is computed only once and remains the same for any new demand. Thus, the "modified heuristic" removes the while loop from Algorithm 2 and operates sequentially to provide a solution for the routed demand.

Table 7 provides the numerical results of the two versions of the heuristic for the largest topology size. Since all instances have a fixed topology, the distinction is between the first time the heuristic is run to determine

TABLE 7. Efficiency of the modified heuristic for large topology sizes, $n = 70$.

	Original Heuristic		**Modified Heuristic**		**RD Gap**	**Time Reduction**
	RD (Mbps)	Time (s)	RD (Mbps)	Time (s)		
First run	9642	19.46	9333	6.66	**2.14%**	**99.84%**
Avg. 20 Instances	9807.65	14.74	9598.35	0.02		

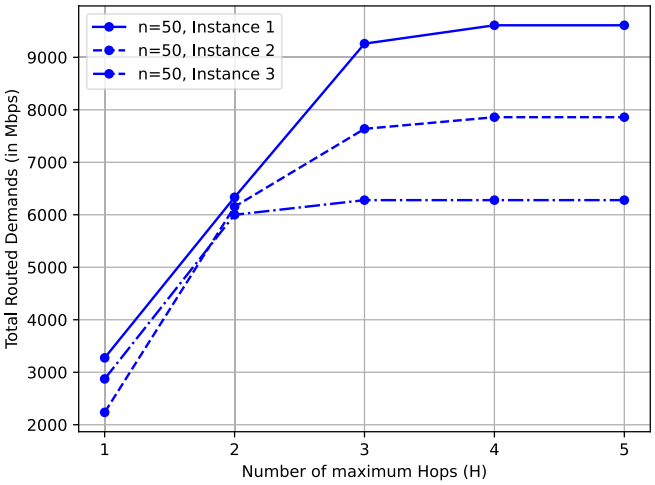

FIGURE 4. Sensitivity to H for $n = 50$.

the maximum tree and the subsequent times it is called to route the demands. As it can be seen from Table 7, the total demand routed by the original version of the heuristic exceeds that of the modified version by 2.14%. However, this minor deterioration in the quality of the solution allows for a significant reduction in computation time. Indeed, the computational time is reduced by 66% for the first run and by 99.8% percent for all subsequent runs. The results indicates that the modified version of the heuristic can be effectively deployed for large instances.

6.2.3. Sensitivity to H and L_0

The last experiment studies the sensitivity of the developed heuristic to some of the problem parameters. The focus is on the maximum number of hops H and the maximum number of links L_0 enabled by the UAV. The hop parameter is related to the connections between the SCs (GtG), while the maximum links involves the UAV reliability, flight duration and power consumption to enable the Air to Ground (AtG) connections.

The heuristic sensitivity to the hop parameter is considered for problems with $n = 50$ and $n = 60$. For a given instance, we keep the same positions of SCs and the UAV, and we keep the demand unchanged. We vary the hop parameter H and then determine the total routed demand. The results of this analysis are depicted in Figures 4 and 5. We can observe that starting from $H = 3$, more demand can be routed. Beyond this value ($H > 3$), no significant improvement can be achieved. This is due to the density of the network and the links capacities of the SCs.

Similarly, we conduct a sensitivity study of the heuristic to the maximum links of the UAV for $n = 50$ and $n = 60$. We keep the same positions of SCs and the UAV, as well as the demand for all instances. We vary the parameter L_0 and then determine the total routed demands. The results of this analysis are presented in Figures 6 and 7. Overall, we observe that the heuristic provides better routing as L_0 increases. In particular,

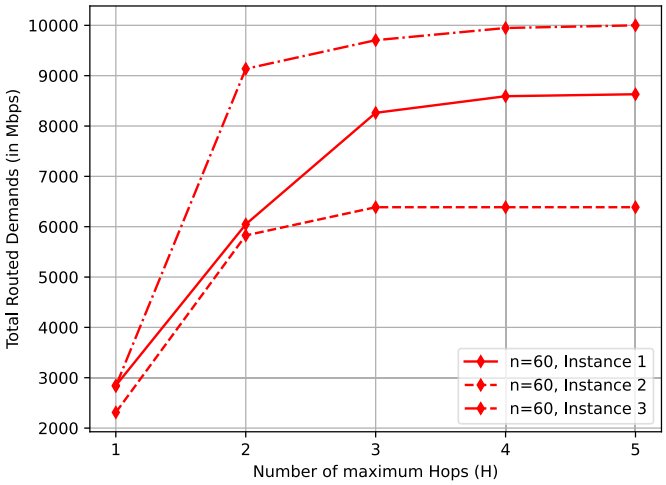

FIGURE 5. Sensitivity to H for $n = 60$.

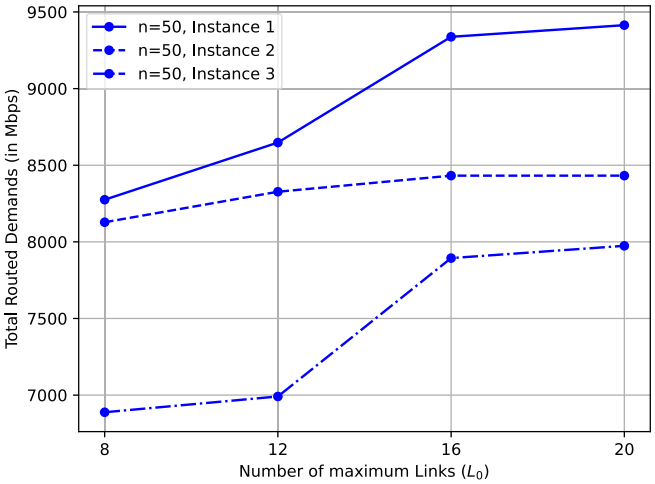

FIGURE 6. Sensitivity to L_0 for $n = 50$.

from $L_0 = 16$, the curve slopes decrease, meaning that the routing improves softly for all instances. Here, we emphasize that maximizing the demands to be routed should not overload the UAV to not impact its reliability.

7. CONCLUSION

In this paper, we proposed a new mathematical model to maximize the total fulfilled routing demands of SCs connected to a drone deployed as a backhaul hub to relay traffic to the core network. We provided empirical evidence that the proposed new model is efficient for small- and medium-sized topologies. We proved the efficiency of the exact approach, especially for the case where the UAV and the SCs of the network are set up and only the demand changes. Specifically, instances with up to 60 SCs can be optimally solved in an average time of less than 6 seconds. Additionally, we proposed a heuristic that generates efficient solutions for large-scale topologies, achieving high-quality solutions with more than 95% of the optimal values found in a

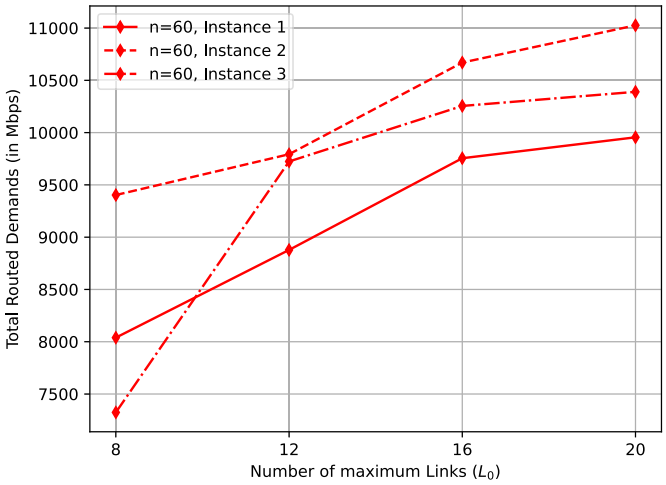

FIGURE 7. Sensitivity to L_0 for $n = 60$.

very short computational time. This is especially the case for the modified version of the heuristic which can solve large instances with 70 or 80 SCs in less than one second. Furthermore, we proposed a column generation algorithm to efficiently solve the LP relaxation of a second MILP model with an exponential number of variables to evaluate the performance of the heuristic on large instances. We also conducted a sensitivity analysis of the proposed heuristic with respect to the number of hops and the number of links between drones. The numerical results demonstrate the performance of the developed heuristic in terms of routed traffic and execution time.

This research is an initial step towards addressing the more complex problem of maximizing the sum rate in a generalized system architecture that includes multiple interfering SC clusters. Future research directions include determining the optimal positions of UAVs to allow for maximum sum rate and studying the adaptation and integration of the flow routing models developed in this paper into a generalized solution framework. This framework would optimize the sum rate in a multi-cluster network while simultaneously maximizing the mobility of multiple UAVs. Furthermore, as a second direction for future research we recommend the development of a branch-and-price algorithm for solving Formulation (EF) and a Benders decomposition algorithm for solving Formulation (P). We expect these approaches to be highly effective in solving the problem exactly, but this needs to be verified experimentally.

Acknowledgements. This research was made possible by NPRPC Grant No. NPRP 13S-0130-200200 from the Qatar National Research Fund (a member of The Qatar Foundation). The statements made herein are solely the responsibility of the authors.

REFERENCES

[1] K. Akkaya and M. Younis, A survey on routing protocols for wireless sensor networks. *Ad Hoc Netw.* **3** (2005) 325–349.

[2] I.F. Akyildiz, A. Kak and S. Nie, 6G and beyond: The future of wireless communications systems. *IEEE Access* **8** (2020) 133995–134030.

[3] A. Al-Hourani, S. Kandeepan and A. Jamalipour, Modeling air-to-ground path loss for low altitude platforms in urban environments. In *2014 IEEE Global Communications Conference* (2014) 2898–2904.

[4] A. Al-Hourani, S. Kandeepan and S. Lardner, Optimal LAP altitude for maximum coverage. *IEEE Wirel. Commun. Lett.* **3** (2014) 569–572.

[5] A. Almohamad, M.O. Hasna, T. Khattab and M. Haouari, Maximizing dense network flow through wireless multihop backhauling using UAVs. In *2018 International Conference on Information and Communication Technology Convergence (ICTC)* (2018) 526–531.

[6] A. Almohamad, M.O. Hasna, T. Khattab and M. Haouari, An efficient algorithm for dense network flow maximization with multihop backhauling and NFPs. In *2019 IEEE 90th Vehicular Technology Conference (VTC2019-Fall)* (2019) 1–5.
[7] A. Almohamad, M.O. Hasna, T. Khattab and M. Haouari, On network flow maximization via multihop backhauling and UAVs: An integer programming approach. In *2019 IEEE 89th Vehicular Technology Conference (VTC2019-Spring)* (2019) 1–6.
[8] M. Alzenad, M.Z. Shakir, H. Yanikomeroglu and M.-S. Alouini, FSO-based vertical Backhaul/Fronthaul framework for 5G+ wireless networks. *IEEE Commun. Mag.* **56** (2018) 218–224.
[9] G. Amponis, T. Lagkas, M. Zevgara, G. Katsikas, T. Xirofotos, I. Moscholios and P. Sarigiannidis, Drones in B5G/6G networks as flying base stations. *Drones* **6** (2022).
[10] Y. Dai, Y. Guo and J. Hao, UAV placement and resource allocation for multi-hop UAV assisted Backhaul system. In *IEEE INFOCOM 2021 - IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)* (2021) 1–6.
[11] L. Flores-Luyo, A. Agra, R. Figueiredo and E. Ocaa, Mixed-integer formulations for a routing problem with information collection in wireless networks. *Eur. J. Oper. Res.* **280** (2020) 621–638.
[12] M. Haenggi, Stochastic Geometry for Wireless Networks. Cambridge University Press (2012).
[13] S. Hanna, E. Krijestorac and D. Cabric. UAV swarm position optimization for high capacity MIMO Backhaul. *IEEE J. Sel. Areas Commun.* **39** (2021) 3006–3021.
[14] S. Irnich and G. Desaulniers, Shortest path problems with resource constraints. In *Column Generation*. Springer (2005) 33–65.
[15] X. Jiang, M. Sheng, N. Zhao, C. Xing, W. Lu and X. Wang, Green UAV communications for 6G: A survey. *Chinese J. Aeronaut.* **35** (2022) 19–34.
[16] E.L.M. Júnior, V.N. Coelho, I.M. Coelho, Y.A. de M. Frota, R.H. Koochaksaraei, L.S. Ochi and B.N. Coelho, UAVs routes optimization on smart cities and regions. *RAIRO:RO* **56** (2022) 853–869.
[17] H.P. Keeler, Simulating a Matérn Cluster Point Process (2018). URL https://hpaulkeeler.com/simulating-a-matern-cluster-point-process/.
[18] M.E. Keskin, A column generation heuristic for optimal wireless sensor network design with mobile sinks. *Eur. J. Oper. Res.* **260** (2017) 291–304.
[19] E.-S. Kim and C.A. Glass, Perfect periodic scheduling for binary tree routing in wireless networks. *Eur. J. Oper. Res.* **247** (2015) 389–400.
[20] H. Koumaras, G. Makropoulos, M. Batistatos, S. Kolometsos, A. Gogos, G. Xilouris, A. Sarlas and M.-A. Kourtis, 5G-enabled UAVs with command and control software component at the edge for supporting energy efficient opportunistic networks. *Energies* **14** (2021) 1480.
[21] S. Kurt and B. Tavli, Path-Loss Modeling for Wireless Sensor Networks: A review of models and comparative evaluations. *IEEE Antennas Propag. Mag.* **59** (2017) 18–37.
[22] B. Li, Z. Fei and Y. Zhang, UAV communications for 5G and beyond: recent advances and future trends. *IEEE Internet Things J.* **6** (2019) 2241–2263.
[23] Y. Lu, C. Yang and J. Yang, A multi-objective humanitarian pickup and delivery vehicle routing problem with drones. *Ann. Oper. Res.* **319** (2022) 291–353.
[24] A. Masaracchia, Y. Li, K.K. Nguyen, C. Yin, S.R. Khosravirad, D.B.D. Costa and T.Q. Duong, UAV-enabled ultra-reliable low-latency communications for 6G: A comprehensive survey. *IEEE Access* **9** (2021) 137338–137352.
[25] 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. Surv. Tutor.* **21** (2019) 2334–2360.
[26] O. Ozkan and S. Kilic, UAV routing by simulation-based optimization approaches for forest fire risk mitigation. *Ann. Oper. Res.* **320** (2023) 937–973.
[27] L.L. Pinto, K.C.C. Fernandes and K.V. Cardoso, Flow routing aiming load balancing and path length in multi-hop networks with different link qualities. *RAIRO:RO* **55** (2021) 2631–2637.
[28] T.S. Rappaport, Wireless Communications: Principles and Practice. Prentice Hall PTR, New Jersey (1996).
[29] S.A.W. Shah, T. Khattab, M.Z. Shakir and M.O. Hasna, A distributed approach for networked flying platform association with small cells in 5G+ networks. In *GLOBECOM 2017–2017 IEEE Global Communications Conference* (2017) 1–7.
[30] S.A.W. Shah, T. Khattab, M.Z. Shakir and M.O. Hasna, Association of networked flying platforms with small cells for network centric 5G+ C-RAN. In *2017 IEEE 28th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC)* (2017) 1–7.
[31] M.K. Shehzad, A. Ahmad, S.A. Hassan and H. Jung, Backhaul-aware intelligent positioning of UAVs and association of terrestrial base stations for fronthaul connectivity. *IEEE Trans. Netw. Sci. Eng.* **8** (2021) 2742–2755.
[32] N. Tafintsev, D. Moltchanov, M. Gerasimenko, M. Gapeyenko, J. Zhu, S.-P. Yeh, N. Himayat, S. Andreev, Y. Koucheryavy and M. Valkama, Aerial access and backhaul in mmWave B5G systems: Performance dynamics and optimization. *IEEE Commun. Mag.* **58** (2020) 93–99.
[33] Y.B. Türkoğulları, N. Aras, I.K. Altınel and C. Ersoy, A column generation based heuristic for sensor placement, activity scheduling and data routing in wireless sensor networks. *Eur. J. Oper. Res.* **207** (2010) 1014–1026.
[34] Y. Xue, B. Xu, W. Xia, J. Zhang and H. Zhu, Backhaul-aware resource allocation and optimum placement for UAV-assisted wireless communication network. *Electronics* **9** (2020).
[35] Y. Zeng, R. Zhang and T.J. Lim, Wireless communications with unmanned aerial vehicles: opportunities and challenges. *IEEE Commun. Mag.* **54** (2016) 36–42.
[36] T. Zhang, Y. Wang, Y. Liu, W. Xu and A. Nallanathan, Cache-enabling UAV communications: Network deployment and resource allocation. *IEEE Trans. Wirel. Commun.* **19** (2020) 7470–7483.

[37] G. Zhang, N. Jia, N. Zhu, Y. Adulyasak and S. Ma, Robust drone selective routing in humanitarian transportation network assessment. *Eur. J. Oper. Res.* **305** (2023) 400–428.
[38] Y. Zhao, W. Zhai, J. Zhao, T. Zhang, S. Sun, D. Niyato and K.-Y. Lam, A Comprehensive Survey of 6G Wireless Communications (2021) 1–34 Preprint arXiv:2101.03889v2.