QATAR UNIVERSITY

COLLEGE OF ENGINEERING

DESIGN OF EFFICIENT HEURISTIC ALGORITHM FOR DENSE HETEROGENEOUS NETWORKS WITH MULTI-HOP BACKHAULING

BY

ABDELRAHMAN MAZEN ABU SAFA

A Thesis Submitted to

the College of Engineering

in Partial Fulfillment of the Requirements for the Degree of

Masters of Science in Engineering Management

January 2021

COMMITTEE PAGE

The members of the Committee approve the Thesis of
Abdelrahman Mazen Abu safa defended on 02/12/2020.

Mohamed Haouari
Thesis/Dissertation Supervisor

Mazen Hasna
Committee Member

Tamer Khattab
Committee Member

Muhammad Shaker
Committee Member

Wael Alhajyaseen
Committee Member

Approved:

Khalid Kamal Naji, Dean, College of Engineering

ABSTRACT

ABUSAFA, ABDELRAHMAN, M., Masters : January : 2021, Masters of Science in Engineering Management

Title: Design of an Efficient Heuristic Algorithm for Dense Heterogeneous Networks with Multi-Hop Backhauling

Supervisor of Thesis: Mohamed Haouari.

5G networking is considered a promising technology which offers a variety of advantages over the previous networks generations such as higher peak data rate, higher density of connections (ultra-dense networks) and latency. Using of ultra-dense networks as an advantage of 5G networks helps improving the capacity of the network. However, at the same time, it implies challenges on the backhauling between small cells (SC) of the network. Using terrestrial wired hubs is not practical any more as the communication between large number of SCs requires the availability of many locations within line of sight (LOS). Furthermore, this method is costly. An alternative way for achieving high level of backhauling between SCs is using Unmanned aerial vehicle (UAV) for establishing wireless backhauling between SCs. One model for maximizing the throughput of a backhauling network is studied. The model was formulated by (Almohamad et al., 2019) and it has been identified to be NP-hard. As the formulated model is NP-hard, it takes longer periods to solve dense networks with large number of small cells. In this thesis, novel heuristic approaches are designed for providing fast and accurate solutions for dense networks with large number of small cells.

DEDICATION

I would like to dedicate this work to my beloved family for providing me the most suitable environment to build this thesis. Also, I would like to dedicate this thesis to Prof. Mohamed Al-Haouari for his appreciated efforts.

ACKNOWLEDGMENTS

I would like to thank my supervisor Prof. Mohamed Haouari for the huge effort and time he spent with me as well as his continuous support. Also, I would like to thank my family for their support and providing me a good environment to build this thesis. Also, I would like to thank my manager Eng. Jagtar Singh who was always understanding my situation as master student and trying to help me.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Chapter 1: Introduction And Background

The demand for Mobile data traffic has dramatically increased over the 21st century. This explosive growth is driven by data-hungry devices like mobile phones, tablets and broadband wireless applications like video games, multimedia, …Etc (Bogale & Le, 2016). The number of data connections has increased from 7.6 billion in 2015 to 8.0 billion in 2016. This huge & rapid increment of approximately half billion in 2016 is 6 times the yearly average growth of world's population, which is 83 million people per year (Aboagye, 2018). This trend will continue and it is forecasted that the number of data-connections is going to reach 11.6 billion by the year 2021 (Aboagye, 2018). Figure 1 represents a forecast for the global wireless data traffic from 2015 till 2021.

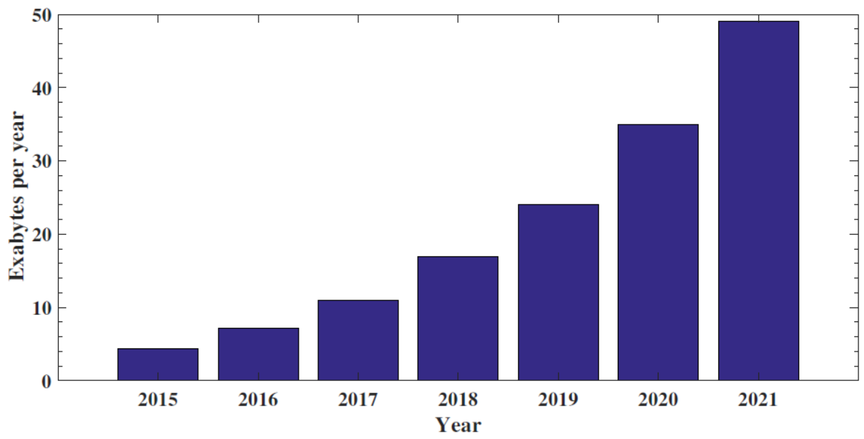

Figure 1. Global wireless data traffic from 2015 till 2021 (Aboagye, 2018).

Unfortunately, there is a significant gap between the explosive growth of wireless data demand and the capabilities of the current networking technologies (Hu & Yi Qian, 2014). While the data traffic becomes one of the most potential challenges for the current cellular networks, including the most advanced fourth generation networks (4G), the technology of 5G networks is considered to be a promising candidate to improve the efficiency of the wireless communications and overcome the problem of explosive data demand traffic. 5G networks are expected to deliver a capacity of 1000 times more than the current 4G networks and to improve the quality of service (QoS) (Bogale & Le, 2016). 5G networks are expected to have the following features and criteria (Bogale & Le, 2016): -

- Coverage and Data Rate: the minimum data rate experienced by user anywhere and anytime shall be 1 Gb/s.
- Latency: the latency is measured by how fast the data is delivered to the destination point. The end-to-end latency requirement of 5G networks will be at the range of 1-5 milliseconds.
- Quantity of connected devices: 5G networks have the capability to support 100 times the maximum quantity of devices that could be supported by the current (4G) network.
- Energy and cost efficiency: 5G networks are designed to have a higher cost efficiency (US$/bit). Also, the energy efficiency (bits/J) of 5G networks shall be improved by a factor of 1000 compared with that achieved by 4G networks.

Specifically, 5G networks are developed to serve a larger amount of data traffic and achieve higher reliability, security, and quality of service (QoS) (Bogale & Le, 2016). Dense Heterogeneous networks (HetNets) with wireless backhauling and millimetre waves (mmWaves) are considered to be critical key enablers of 5G technology (Almohamad et al., 2018). Both key enablers are discussed in the following two sections. It is expected that the optimization of the huge available bandwidth in mmWaves and the deployment of HetNets will help to address the data rate requirements of 5G networks (Niknam et al., 2016).

Millimeter Waves: -

Currently, most of the communications use the frequencies of “sweet spot” region. “sweet spot” region refers to the frequencies from 300 MHz to 3 GHz. The reason behind selecting this band is because of its unique propagation characteristics for wireless communication. Recently, "sweet spot” became almost fully occupied (Aboagye, 2018). This enforced the researchers from the different academic and industrial sectors to investigate the utilization of mmWaves in 5G incoming networks. mmWaves are those waves which are located in the band from 30-300 GHz (Almohamad et al., 2018). MmWaves have the following advantages over the other bands (Gao et al., 2015): -

- The large bandwidth which provides the potential GHz rates for wireless communications.
- As the wavelengths of mmwaves are small, we can deploy large number of antennas at both the sending and receiving nodes. This improves the directivity of the signal and hence reduces the co-channel interference. The spectrum of mmwaves is represented by Figure 2.

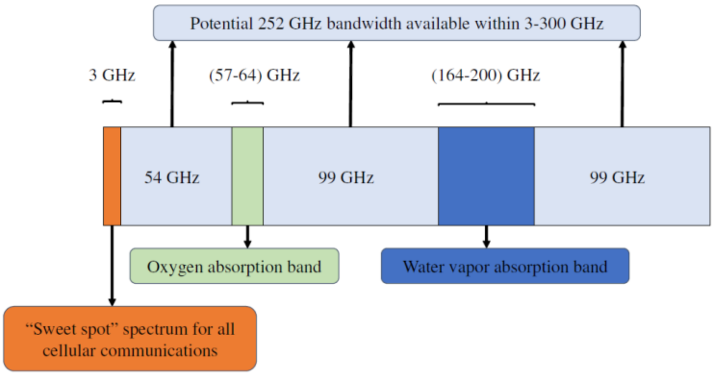

Figure 2. The spectrum of mmwaves (Aboagye, 2018).

It is shown in Figure 2 that the different frequencies in mmwaves' band have different characteristics. For example, it is known that oxygen molecules absorb the energy of 60 GHz wave. This phenomena weakens the waves over the distance and makes them unsuitable for long-distance applications. Instead, 60 GHz signals can be used in the secure indoor communications (Bogale & Le, 2016). On the other hand, the other frequencies in the region 30-160 GHz are less affected by oxygen molecules which makes them more suitable for long-distance applications than 60 GHz signal (Bogale & Le, 2016). Accordingly, the properties of the different frequencies of mmwaves shall be deeply studied with respect to the application, blockages, and absorptions.

Heterogeneous Network (HetNet): -

In the recent years, heterogenous networks (HetNet) have been considered as promising candidate to improve coverage/capacity and optimize the energy consumption of 5G networks (Hu & Yi Qian, 2014). Heterogenous network can be defined as a wireless network which includes nodes with different coverage sizes and transmission powers (Hu & Yi Qian, 2014). Figure 3 represents an example of a HetNet.

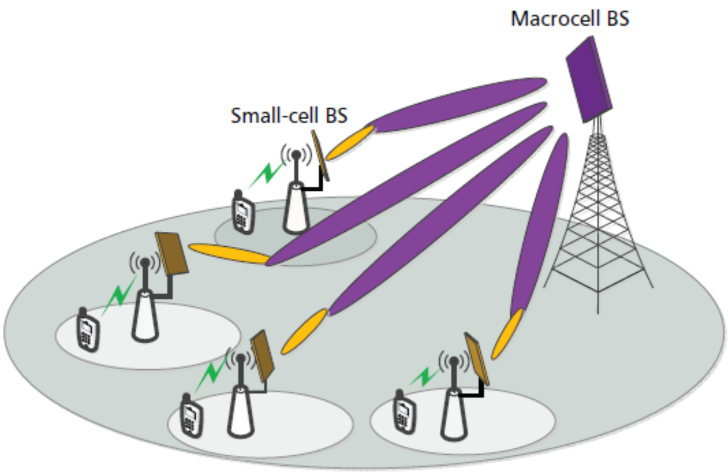

Figure 3. Heterogeneous Network (Bogale & Le, 2016).

There are 3 main elements in any HetNet which are: - the microcell or the macro base station (MBS), small cells (SCs) and end users (EUs). As shown in Figure 3, EUs are connected to the SCs and SCs are connected to the macro base station which itself is connected to the core network. Hence, the data are transferred from MBS to EU and vice versa through the SC. The process of the communication between the MBS and SC is referred to as the Wireless Backhauling process (Almohamad et al., 2018). Sometimes bulk EUs are connected directly to the MBS. In HetNet, MBS is referred to as high power node which has large coverage area. This type of nodes is deployed in a planned pattern to distribute the data from the core network to the different areas. SCs are referred to as low power nodes, which have small coverage areas and used to extend the coverage of high-power nodes (Hu & Yi Qian, 2014).

As mentioned previously, the wireless backhauling is done between the different HetNet's elements. As per (Gao et al., 2015), in order to maximize the effectiveness and efficiency of HetNet, Gigahertz backhauling shall be maintained between MBS and SCs. Mmwave is a suitable candidate for this purpose as it provides transmission rates of 30-300 GHz.

Using of UAV As Macro Base Stations

The conventional types of terrestrial MBS in Figure 3, which uses Fibre Optics for the connection to the core network, is not an efficient solution in urban areas due to the following reasons (Almohamad et al., 2018): -

- They are very costly in terms of implementation. Implementing a terrestrial MBS requires much infrastructural work and modifications for the purpose of laying the Fibre Optics cables which rises the costs of implementing MBSs.
- Shortage of the locations with line of sight as most of locations in Urban areas are occupied by buildings and streets.
- It is not flexible solution. It cannot be reused, shifted or transferred to different locations.

Due to the previous reasons, Terrestrial MBS are not practical solution anymore. An alternative is using Unmanned aerial vehicles (UAV) technology as MBS (Almohamad et al., 2018). The network with UAVs implemented as MBSs is represented in Figure 4.

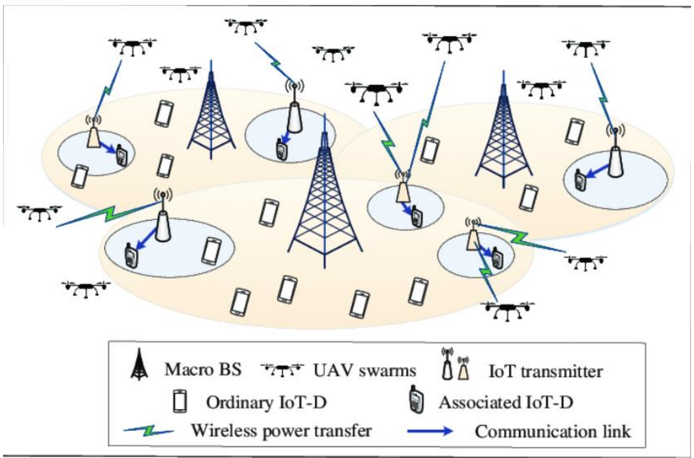

Figure 4. Using of UAV in HetNets (Lagunas et al. (2017))

As shown in Figure 4, the SCs in the ground are backhauled to the UAV in the air by mmwaves. As UAVs are flying platforms, designers do not require to free a location in the ground (with line of sight) (Almohamad et al., 2018). Moreover, UAVs are connected to the core network by wireless connection which eliminates the need for Fibre Optics usage (Almohamad et al., 2018). Hence, much cost will be saved by considering using the UAV as MBS. In addition to the low implementation cost, UAVs are easy to be reused or shifted to different locations.

The studied problem of this thesis will be a UAV-based network

Path Losses Of Mmwaves: -

Despite the valuable advantages of mmwaves, they suffer from high propagation losses over the distance (Almohamad et al., 2018). This creates major challenges for network operator to use mmwaves in long distance applications. In HetNets, there are two types of pathlosses which are discussed as following (Almohamad et al., 2018): -

- Air to Ground losses (Almohamad et al., 2018): -

This type of losses occurs within the connections between the air and ground, in our case, these losses occur within the connections from UAV to the SCs and vice versa. Air to ground path loses can be calculated using the following formula: -

$$PL_{ATG} = 20\ log_{10}\left(\frac{4\pi f_c^u d_s^u}{c}\right) + P(LOS)n_{LOS} + P(NLOS)n_{NLOS}$$

Where f_c^u is carrier frequency of UAV-SC link, d_s^u is the distance between UAV and SC, and c is the speed of light. n_{LOS} and n_{NLOS} refer to the additional losses of line and non-line of sights propagation, respectively. $P(LOS)$ refers to the probability of line of sight and it is calculated using the formula: -

$$P(LOS) = \frac{1}{1 + \alpha\, EXP(-\beta\left(\frac{180}{\pi}\theta - \alpha\right))}$$

Where α and β are parameters related to the environment. θ is the elevation angle between UAV and SC as shown in Figure 4. $P(NLOS) = 1 - P(LOS)$.

- Ground to Ground losses (Almohamad et al., 2018): -

This type of losses occurs between the SCs in the ground and it is calculated using the formula:

$$PL_{ATG} = 20\ log_{10}\left(\frac{4\pi f_c^s}{c}\right) + 10\ \alpha\ log_{10}(d_s^s)$$

Where f_c^s in the carrier frequency in SC-SC link, c is the speed of light, and d_s^s is the distance between the two SCs.

It can be noticed that for both types of losses, if carrier frequency and distance between nodes increase, the losses increase too. This makes mmwaves impractical for long-distance and therefore, a practical solution is required.

HetNet With Multi-hop Backhauling (Almohamad et al., 2018): -

The only solution to minimize the losses discussed earlier is to minimize the distance travelled by mmwaves. This can be done by allowing SC to transfer its demand to the nearby SCs first. From nearby SCs, demand will be forwarded to the UAV. In other words, the demand will be passed through different intermediate stations (SCs) before reaching the UAV. By implementing this technique, the full distance between the transmitting SC and UAV is decomposed into smaller distances which reduces the overall losses. This technique is called multi-hop backhauling where a hop indicates a single link. As shown in Figure 4, this idea can be implemented by allowing only a limited number of SCs to be connected to UAV. The SCs which are connected directly to UAV are called Gateways. The remaining SCs will backhaul their demands to the

UAV through the gateways.

The Challenging Aspects Of HetNet: -

Despite the role of HetNets in supporting the capacity/Coverage of 5G networks, the implementation of HetNets is associated with many major challenges in terms of optimizating the design of the network. As 57% of energy consumption of HetNet is consumed by the radio access nodes (MBS & SCs), deploying large number of nodes in a network may improve the quality of communication but it will increase the total energy consumption of that network (Hu & Yi Qian, 2014). At the same time, if a smaller number of nodes is used, the capacity/coverage will be affected negatively (Hu & Yi Qian, 2014). Hence, there is trade-off between the cost and quality. Moreover, NetNets have many constrains in terms of the connected links, received flows, link's capacity, … Etc.

The main challenge is to optimize the available limited resources (including SCs and UAVs) of HetNet in such a way that quality of the network is maximized with the consideration of the network's constrains. The different optimization problems with different objectives and models are discussed in Literature review. Some examples of the objectives of the networks optimization models are maximizing the total data flow (throughput) of networks, minimizing the energy consumption, …. Etc.

The Contribution Of This Thesis: -

In this thesis, we will investigate the optimization problem which was raised by (Almohamad et al., 2019). The problem is deeply discussed in the literature review. The main objective of the investigated paper was to maximize the total throughput of a Heterogeneous network with multi-hop backhauling subjected to a set of constrains. The solution of the model (decision variables) provides the optimized planning of the flows of the network such that the total throughput of the network is maximized. As what we will explain later in chapter 2, two NP-hard models were formulated: one model for full demand problem and the other for partial demand problem. The contribution of this thesis is to perform the following: -

1- To design very fast and efficient heuristic algorithms for maximizing the total throughput of dense networks with large number of SCs. The algorithm shall provide near-optimal solutions within a very small period of time (part of a second). As the investigated models by (Almohamad et al., 2019) are NP-hard, they could solve only small-sized instances (< 35 nodes) efficiently. For larger instances, investigated models took longer periods. As the real practice networks contain large number of SCs (> 100), there is a need to develop efficient heuristic algorithm to solve these large-scaled instances efficiently and effectively.

2- To apply the developed algorithms on large number of instances and assess the accuracy of developed heuristics by comparing the results to the optimal results of the model.

3- To assess the sensitivities of the developed heuristics with respects to the different network's parameters. In other words, the different parameters of the network will be varied and the effect of variation on the accuracy of heuristics will be analysed. These parameters include: -

    - Maximum number of links per SC
    - Maximum number of flows per SC
    - The maximum allowed hops
    - The number of the edges in the network

Chapter 2: Literature Review

Introduction

As discussed earlier, many challenges are associated with applying the principle of HetNet with multi-hop backhauling. As there are a limited number of nodes in a network, the proper optimization of the available resources shall be performed in order to enhance the capacity, quality and efficiency of the network with consideration of the constrains of the network. Over the last 20 years, the field of HetNets has attracted many researchers from both academic and industrial sectors. Many problems have been modelled for different types of HetNets with different constrains. Some examples of these problems are (Bogale & Le, 2016):

- Network planning and traffic management: by optimizing the locations of nodes, the number of the required SCs and MBSs, and the paths of the demand's flow of each SC. The optimization can be performed for different objectives such as maximizing the total throughput, minimizing the consumption energy, minimizing the implementation cost, … Etc. The studied problem of this thesis belongs to this category where the model determines the optimized routes by which the demand of each SC should flow so the total throughput of the network is maximized (Bogale & Le, 2016).
- Cell association: to determine which node (SC or MBS) will serve each end user EU in the network. In some networks, one UE can have several cell associations during its active periods. In some other networks, bulk consumers are connected directly to the MBS. Hence, proper cell association shall be accomplished to optimize the resources of the network and deliver an acceptable quality of service (Bogale & Le, 2016).

In this chapter, we present some valuable contributions toward the optimization of

HetNet. Chapter includes 2 main parts: the first part represents the main problem of this thesis for which the heuristic algorithms are designed. The second part includes a variety of contributions in the field of HetNets optimization. The contributions were made for different types of HetNet with different features and constrains.

The Concerned Problem Of The Thesis:

Almohamad et al. (2019) studied the problem of total flow (throughput) maximization of dense HetNet with multi-hop backhauling and UAVs. Figure 4 in the previous chapter represents the studied network. There are two main types of nodes in the network which are the small cells (SCs) and the Unmanned aerial vehicles (UAV). Each SC in the network serve a group of end users (EUs) and therefore has a demand in bits per second (bps) which needs to be transferred to the UAV node (which itself is connected to the core network) without violating some constrains which will be discussed in the following part. As shown in Figure 4, some SCs are connected directly to UAV and these SCs are called gateways. The other SCs are connected to the UAV through the gateways. In other words, each SC forward its demand to the neighboring SCs till the demand reaches a gateway SC. From the gateway, the demand is forwarded to the UAV. This process is called the multi-hop backhauling such that a hop represents a link between two nodes. When the demand passed through many links before reaching the UAV, then we can say that it was backhauled through multi-hops or multi-links.

Problem Formulation

Consider the graph $G = (E, V)$ where V represents the set of n nodes such that $V = \{0,1,2,3, \ldots (n-1)\}$. Node 0 is the UAV and the nodes 1 to (n-1) are SCs. The set E represents the set of links such that $E \in \{(i,j) : i,j \in V\}$. It can be noticed that only one UAV has been considered for this problem which is node 0. Each link $(i,j) \in E$ has a capacity b_{ij}. The capacity of a link can be calculated using the formula:

$$b_{ij} = B_{ij} log_2(1 + SNR_{ij})$$

where B_{ij} represents the bandwidth of the channel between nodes i and j. Each SC node $k = 1, \dots, (n-1)$ has a demand d^k to be transferred to node 0 (UAV). d_{min} represents the minimum portion of SC's demand that needs to be fulfilled (transferred to UAV). Every node in the network has a relaying capacity F_j which can be defined as the maximum number of flows that can pass through it. Also, each node has a maximum number of connected links L_j which represents the maximum number of nodes that may connect to the node such that $\forall j \in V$. The maximum number of hops (links) between a SC node and UAV is constrained by H where H represents the maximum number of the permitted hops.

Using the information above, (Almohammad, 2019), have formulated the 2 models as the following: -

Partial Demand Model:

In partial demand problem, it is not necessary to transfer the whole demand d^k of a SC node to UAV. Instead a minimum portion d_{min} of the demand d^k shall be fulfilled. In other words, we have the flexibility to satisfy any amount of demand in the range ($d_{min} \times d^k$) to d^k. The following are the decision variables of the model:

- Binary decision variables: -

1- $Y_{ij}^k = \begin{cases} 1, if\ the\ demand\ of\ k^{th}\ SC\ pass\ through\ the\ link\ (i,j) \in E \\ 0, Otherwise \end{cases}$

2- $Z^k = \begin{cases} 1, if\ the\ demand\ of\ k^{th}\ SC\ isfulfilled\ in\ the\ solution \\ 0, Otherwise \end{cases}$

3- $D_{ij} = \begin{cases} 1, if\ the\ link\ (i,j)\ is\ existing\ in\ the\ solution\ of\ the\ model \\ 0, Otherwise \end{cases}$

- Continuous decision variables: -

1- $F_{ij}^{k} = \begin{cases} \textit{Flow variable which represents the amount} \\ \textit{of the } k^{th} \textit{ SC's demand transferred through link } (i,j) \end{cases}$

2- $\vartheta^{k} = \begin{cases} \textit{represents the amount of the fulfilled demand} \\ \textit{of the } k^{th} \textit{ SC} \end{cases}$

The optimization model was formulated as the following: -

$$\text{(PF)} \quad \max \sum_{k \in K} \vartheta^{k} \tag{1}$$

subject to

$$\sum_{i:(i,j)\in E} Y_{ij}^{k} - \sum_{i:(j,i)\in E} Y_{ji}^{k} = \left\{ \begin{array}{ll} 0, & \notin \{0,k\} \\ -Z^{k}, & j = k \\ Z^{k}, & j = 0 \end{array} \right\}, k \in K, \tag{2}$$

$$\sum_{i \in V} \sum_{j \in V} Y_{ij}^{k} \leq H, \quad k \in K, \tag{3}$$

$$\sum_{k \in K} \sum_{i \in V} Y_{ij}^{k} \leq F_{j}, \quad j \in K, \tag{4}$$

$$\sum_{i \in V} (D_{ij} + D_{ji}) \leq L_{j}, \quad j \in K, \tag{5}$$

$$Y_{ij}^{k} \leq D_{ij}, \quad (i,j) \in E, k \in K, \tag{6}$$

$$Y_{ij}^{k} \leq Z^{k}, \quad (i,j) \in E, k \in K, \tag{7}$$

$$\sum_{k \in V} (f_{ij}^{k} + f_{ji}^{k}) \leq b_{ij}, \quad (i,j) \in E, \tag{8}$$

$$0 \leq f_{ij}^{k} \leq d^{k}.Y_{ij}^{k}, \quad (i,j) \in E, k \in K, \tag{9}$$

$$d_{min}.d^{k}.Z^{k} \leq \vartheta^{k} \leq d^{k}.Z^{k}, \quad k \in K, \tag{10}$$

$$\sum_{j \in V} (f_{kj}^{k}) = \vartheta^{k}, \quad k \in K, \tag{11}$$

$$\sum_{j \in V} (f_{j0}^{k}) = \vartheta^{k}, \quad k \in K, \tag{12}$$

$$\sum_{(i,j)\in E} (f_{ij}^{k}) - \sum_{(i,j)\in E} (f_{ji}^{k}) = 0, \quad k \in K, j \in V, j \neq 0, k \tag{13}$$

$$Y_{ij}^{k} \in \{0,1\}, \quad (i,j) \in E, k \in K, \tag{14}$$

$$D_{ij} \in \{0,1\}, \quad (i,j) \in E, \tag{15}$$

$$Z^{k} \in \{0,1\}, \quad k \in K. \tag{16}$$

Figure 5. Optimization model

It is clear that the aim of objective function (1) is to maximize the summation of fulfilled demands (throughput) ϑ^{k} which is equivalent to maximize the total flows to UAV node. The constrain (2) ensures that the demand d^{k} will pass from k^{th} SC to

UAV through a single path without any splitting. Constrains (3), (4) and (5) guarantee that the routing design obeys constrains of maximum hops H, relaying capacity F_j and maximum affordable links L_j respectively. The constrain (6) is to ensure that the demand of k^{th} SC will not pass through the link (i, j) unless and until the link exists so It is a linking constrain. The constrain (7) is also a linking constrain to ensure that no link is used to carry the flow of k^{th} SC unless the path between k^{th} SC and UAV exists. Constrain (8) is a capacity constrain to ensure that the total flow pass through a link does not exceed the capacity of that link. The constrain (9) is to ensure that the maximum flow through a link f_{ij}^k is d^k only and only if Y_{ij}^k is equal to 1, while the constrain (10) ensures that a SC node is considered to be fulfilled only if at least $(d_{min} \times d^k)$ of its demand is satisfied. Constrains (11) to (13) are formulated to guarantee the flow conservation among the nodes. The remaining set of constrains are to define the variables as real and binaries.

Full Demand Model

In the previous problem, a SC is considered fulfilled if at least a portion of d_{min} of the full demand is satisfied. However, in full demand problem, SC is considered to be fulfilled only and only if it's full demand d^k is satisfied. In other words, either full or none of k^{th} SC's demand can be transmitted to UAV. This condition eliminates the need for the variables f_{ij}^k and ϑ^k so they can be replaced as the following: -

$$f_{ij}^k = d^k Y_{ij}^k$$

$$\vartheta^k = d^k Z^k$$

Accordingly, constrains (9) and (10) can be omitted and Constrain (8) can be replaced by the following constrain: -

$$\sum_{k \in K} \left(Y_{ij}^k + Y_{ji}^k\right) . \ d^k \leq b_{ij} \qquad \forall (i, j) \in E \qquad (14)$$

Flow conservation constrains (11) to (13) can be eliminated as they are redundant to (2). The objective function (1) will be edited as the following: -

$$maximize \sum_{k=1}^{n} d^k Z^k$$

Subject to (2) to (7) and (14).

The decision variables of the full demand model include only the binary decision variables of partial demand problem. The continuous decision variables of partial demand problem have not been considered as they are unneeded in full demand model. In both full and partial demand models, the values of the decision variables control the design of the network. The model determines the optimal values of the decision variables which maximize the total throughput of SCs toward UAV node. Therefore, it can be noticed that the models perform the following main tasks: -

- Select the SCs which will be considered for demand fulfilling.
- Design the route from UAV to the selected SC or determine the path through which the node will transfer its demand.

Both models have been identified to be NP-hard. This creates the need for efficient heuristic approaches to solve the same models. The design of the heuristic approaches to solve both partial and full demand models is the concern of this thesis and it will be discussed in the following chapter.

Other Contributions To The Optimization Of Multi-hop Backhauling Networks: -

Zhu et al. (2016) studied the scenario of heterogeneous cellular network HCN where small cells are densely deployed under homogeneous macro-cells. The word “flow” means a single-hop link between 2 small cells. Zhu et al. (2016) formulated the problem of optimal scheduling of the HCN in order to maximize both the number of scheduled flows per each link and the total network throughput. The main constrain of the model was to guarantee the quality of service (QoS) of the scheduled flows. The requirement of QoS of a flow is satisfied if throughput of a flow (T_i) is equal or bigger than minimum throughput requirement (q_i). The problem does not include the routing of the network. Instead, it schedules the flows of a ready-routed network. The model was identified to be NP-hard. Accordingly, a heuristic algorithm called Maximum QoS-aware Independent Set (MQIS) was developed. The proposed algorithm was tested in 73 GHz band and showed a superior performance in terms of the number of successful scheduled flows and the total system throughput compared with other existing schemes.

Gupta & Kalyanasundaram (2017) investigated the problem of resource allocation of networks with self-backhauled half-duplex small cells. Resource allocation means how much of data rate is assigned to each link connected between two small cells. Gupta & Kalyanasundaram (2017) developed a model based on the maximization of sum-utility, where the algorithm of end users (EUs) throughput is considered as the utility function. Using Karush-Kuhn-Tucker (KKT) conditions, a conditional closed-form solution has been derived to determine the optimal fraction of resources which has to be distributed to each link in the network. Furthermore, using the obtained optimal fractions, per transmission time interval (TTI) scheduling and resource allocation policy was derived, which tracks as closely as possible the desired resource allocation fraction. The simulation results show significant gains with the proposed scheme when compared to other schemes which are unaware of the number of users served by each small cell.

Mcmenamy et al. (2020) Studied the ultra-dense mmWave backhaul network which uses the Madrid-grid layout. The studied network composed of Backhaul nodes (BN) with wireless connectivity & Backhaul gateways (BGW) with wired connectivity.

The optimization model was developed for the purpose of maximizing the flow of the network taking into account the minimum flow demand per node. The proposed model has 2 stages: first stage is to select which nodes are going to be considered as BGWs using minimum hitting set approach. The second stage is to maximize the overall demand of the network while meeting the minimum demand for each node and constrain of the maximum number of RF chains per each node. The results showed that the optimal flow rate of a network can be achieved by increasing the number of RF chains at the nodes. Furthermore, the results show that decreasing the allowed number of hops leads to significant increase in the number of BGW and in some cases it leads to negligible increase in the total throughput.

Arribas et al. (2020) formulated the optimization problem of mmWave Backhaul Scheduling (MMWBS). The problem is formulated as MILP. The aim of the model was to search for minimum time T (the time of routing data from sources to destinations) such that the MILP has a feasible solution. Hence, the model does not contain utility function. The problem has been proven to be NP-hard and can be approximated only and only if the interference between links is negligible. Based on linear programming, an algorithm was developed to provide a schedule that achieves a constant approximation of the optimal makespan when the interference is negligible. Furthermore, two heuristic algorithms were developed to solve the model. On average, both algorithms achieved near-optimal results when tested on different topologies.

Islam et al. (2014) investigated the use of multiple frequency bands in the wireless backhaul networks. The studied bands are: - sub-6 GHz band, (6-42 GHz) microwave band & millimetres wave bands (e.g. 60, 70 and 80 GHz). The features of each band were discussed. Also, the integration of aggregator node to the backhaul network was studied. The aggregator is considered as a relay between small cells and macro-cell. Two optimization models have been formulated as mixed integer nonlinear problems. The first model assumes that the communication between small cells and aggregator nodes is done using microwave band while the second model assumes that communication is done using sub-6 GHz band. The aim of both models was to perform joint cost optimal allocation of aggregator node in order to optimize the wireless backhauling network. Models were solved by linear relaxation & branch-and-bound algorithm. Finally, both models were applied to the network of downtown Manhattan. Rezaabad et al. (2018) investigated communication networks with both Fiber and wireless backhauling. These types of networks contain both wired base stations (W-BS) and un-wired base stations (U-BS). The authors formulated a multi-objective optimization problem for joint cell and fiber backhaul planning with the following objectives: to minimize the number of users with unsatisfied demands, to minimize the capital cost of implementing the network and to minimize the cost of fiber optics installation by minimizing the number of W-BS. Two heuristic algorithms were developed. The first algorithm is to calculate the minimum number of BSs which are required to provide full coverage. The second algorithm is non-dominated sorting genetic algorithm II (NSGA-II) and it was used to solve the formulated model. The results showed that (NSGA-II) is more efficient than many other existing optimization algorithms.

Zhang et al. (2018) Studied the problem of the energy-efficient allocation of bandwidth and power allocation in backhauling networks. They built a scheme to maximize energy efficiency by the optimal allocation of power and bandwidth. The problem was formulated as non-convex nonlinear programming problem and then decomposed into 2 separate convex problems: the first problem is for optimal power allocation, and the second one is for optimal bandwidth allocation. An iterative resource allocation algorithm was developed to solve the resource allocation problem. In order to decrease the complexity of the iterative algorithm, another low-complexity algorithm was designed. By comparing the results of both algorithms to the results of the existing schemes, authors found that the developed algorithms are more efficient and accurate.

Lagunas et al. (2017) investigated the networks with in-band full-duplex self-backhauling architecture. In which small cells (SCs) operate on the same spectrum used by the end-users to access the network for the purpose of backhauling. The authors have formulated a power allocation model which considers both access and backhaul links of the network. The novelty of the model is that it optimizes the transmit power of SCs together with the transmit power of backhaul station so that the total flow is maximized. An iterative algorithm has been developed to solve the mentioned model which successfully accommodates the powers in such a way that total flow of the network is maximized.

(Li & Xu, 2018) investigated the use of UAV as SCs in HetNets. Figure 6 represents a UAV-enabled network.

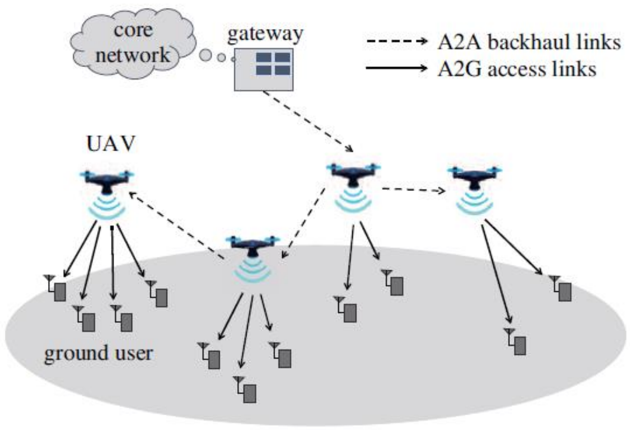

Figure 6. UAV enabled as small cells(Li & Xu, 2018)

As shown in Figure 6, UAVs are connected to the core network via multi-hop wireless backhauls. The aim of the study was to maximize the total throughput of all the end users of the network by optimizing UAVs' deployment locations, the transmit power allocation and the bandwidth allocation. However, the optimization model was identified to be non-convex. Using the techniques of alternating optimization and successive convex programming, authors obtained an efficient algorithm which provides a locally optimal solution. The numerical results showed that the novel algorithm improved the common throughput among the end users as compared to the other existing approaches.

Aboagye et al. (2020) formulated a model for energy efficiency optimization of networks with multi-hop backhauling. Two optimization frameworks were developed to maximize energy efficiency of backhauling networks. The first framework is joint energy, power, and flow control (JEEPF). JEEPF enforces a strict throughput requirement on end users (EU) and maximizes the energy efficiency by the proper optimization of power and backhaul flows. The second framework is joint energy efficiency, power, flow and throughput (JEEPFT) which allows an acceptable range of throughput for each EU and maximizes the energy efficiency by the proper optimization of power, backhaul flows and the total throughput of EUs. The JEEPF was considered to be convex problem which could be solved by the dual decomposition. JEEPFT was considered to be non-convex problem and it was solved using two techniques based on Dinkelbach method and bisection method. It has been observed through the simulation results that JEEPFT achieves better energy efficiency, larger throughput and less power consumption.

Big portion of backhauling networks uses mmWave. Due to vulnerability of mmwave to obstacles, the flows between two stations are easy to be blocked. To overcome the blockage of mmwave in backhauling networks, (Niu et al., 2019) proposed relay-assisted and QoS aware (RAQS) scheduling scheme. RAQS consisted of two parts which are the relay selection and the transmission scheduling algorithms. Relay selection algorithm selects the non-repeating relays with high link rates to be used for the blocked flows so the requirements of QoS of the flows can be achieved as soon as possible. Then, the heuristic transmission scheduling algorithm is applied to exploit the concurrent transmissions to meet the QoS requirements of flows as much as possible. The simulation results proved the effectiveness of RAQS in overcoming the blockage problem of mmwave backhauling networks.

Ghimire & Rosenberg (2015) studied the effect of limiting the capacity of backhaul links on the process of end users scheduling in backhauling networks. Two types of backhaul links were considered which are backhaul links from small cells to the macrocell and the backhaul link from the macrocell to the core network. The authors formulated a global user scheduling problem and analyzed it under different 3 different cases of backhaul limitations. The first case is when both types of backhaul links (between small cells & macrocell and between macrocell & core network) are not limited. In this case, authors derived closed-form scheduling solutions to the main global user scheduling problem. The second case is when the backhaul links between small cells and macrocell are limited while the backhaul link between the macrocell and core network is not. In this case, the global problem can be decomposed into a set of local user scheduling problems (one per small cell). A simple heuristic scheduling algorithm was developed and it was observed to perform well under the previous case of backhaul limitation. In the last case, where the backhaul link between the macrocell and core network is limited, the optimal scheduling becomes more complex. A simple realization-agnostic heuristic algorithm was developed to solve the scheduling problem of the third case of backhaul limitation.

Zhang et al. (2018) investigated the impact of deploying NOMA in a backhauling network. NOMA enables many end users to use the same frequency band by implementing the technique of successive interference cancellation. The authors developed a two-step user scheduling and power optimization scheme in order to maximize energy efficiency of NOMA networks. As a higher throughput rate of a network requires more power, the aim of the scheme was to balance both the power consumption and the improvement of sum rate of the system. The simulation results proved that the developed scheme succeeded to improve the energy efficiency of the network.

Patil & Bhavikatti (2019) developed RPRA scheme for backhaul network with one macro-cell and several femto-cells (small cells). The scheme included 2 robust approaches: the robust power controller approach and the robust channel allocation approach. RPRA could improve the spectral efficiency at areas with low coverage by eliminating week coverage zones. PRPA was compared with some existing techniques and it showed improvement in total throughput and reduction in consumed power and time delay.

Chapter 3: Novel Algorithms

Problem Definition: -

The novelty of this thesis is to design heuristic algorithms to solve the problem of total throughput maximization of HetNets with multihop backhauling which has been investigated and modelled by (Almohamad et al., 2019). The problem has been explained in details in chapter 2. Just as a brief recap, the problem is to optimize the flows of the SCs' demands to the UAV. This can be accomplished using the following:

- Select the SCs which will be considered for demand fulfilling.
- Design the route from UAV to the selected SC or determine the path through which the node will transfer its demand.

As discussed in chapter 2, this kind of networks has a set of standard constrains which needs to be considered. These constrains are: -

- L_j: It represents the maximum number of links/nodes which are allowed to be connected to node j. where $j \in V$.
- F_j: It is defined as the relaying capacity or the maximum number of demands (flows) which can be received by node $j \in V$ at a moment. Note that one link can carry more than one flow. For example, suppose that a link is established from node 3 to 2. The mentioned link is capable to transmit the demand of 3 in addition to the other demands which are backhauled through node 3 as long as the other constrains are met.
- H: It is the maximum number of hops or transitions of demand between a node and UAV.
- b_{ij}: it is defined as link capacity.

Moreover, the problem has the following decision variables: -

- Y_{ij}^{k}: It determines whether the demand of node $k \in C$ is passed through the link (i, j) or not.
- Z_k: - This variable determines whether the node $k \in C$ has been chosen for the demand transfer to UAV. $Z_k = 1$ indicates that the node k is chosen.
- F_{ij}^{k}: - Determines how much of SC's demand is transferred through link (i, j).
- Φ(k): - it represents how much of node k's demand is satisfied.
- Total flow: - it is the summation of Φ(k) for all SCs such that $\sum_{k=0}^{k=(n-2)} \Phi(\mathrm{k})$

Important note: for our heuristic algorithms, we consider node (n-1) as the UAV and nodes 0 to (n-2) as SCs nodes. This different from previous chapter where authors considered node 0 as UAV.

Full Demand Vs Partial Demand Problem

The problem has two forms in term of the demand. The first form is the full demand problem where either the full or none of SC's demand can be fulfilled. It is not allowable to consider only a portion of the demand for transfer. If it is not possible to consider the full demand, then the SC shall not be considered in the solution and Z_k will be set to 0.

On the other hand, there is a possibility to consider a portion of the demand in partial demand problem which is the second form of the main problem. In this case, a constant $dmin$ is used to specify the minimum potion of demand that can be transferred such that $dmin \leq 1$. Assume that $dmin$ is equal to 0.75, then we can transfer 75-100% of the demand. Otherwise, the SC will not be considered in the solution of the network and Z_k is set to 0.

Motivation

Both full and partial demand models have been identified to be NP-hard Problem. It is known that the optimal solutions of NP-Problems are time consuming which makes them impractical for networking problems. Because of the continuous and rapid change of demands among the network, the solution shall be provided in a part of second. This creates the need for efficient heuristic algorithms which are capable to provide an accurate and fast solution.

The Novelty Of The Thesis

The novelty of this thesis is to design very fast and efficient heuristic algorithms to maximize the total throughput of dense networks with large number of SCs (>100). The formulated models by (Almohamad et al., 2019) could solve only small-sized instances (< 35 nodes). For larger instances, investigated models were not efficient as CPU time increased excessively. As the real practice networks contain large number of SCs (> 100), there is a need to develop efficient heuristic algorithm to solve these large-scaled instances efficiently and effectively. The designed algorithms are novel and to the best of our knowledge, there is no researcher, till the date of writing this thesis, has developed similar algorithms to solve the mentioned problem.

Two heuristic algorithms have been developed to solve full demand problem while 1 only has been designed for partial demand problem. The first algorithm for full demand problem is based on Depth first search DFS technique while the second algorithm is based one Dijkstra shortest path algorithm. The third algorithm which used for partial demand problem is derived from the first algorithm. The 3 algorithms have generated efficient solutions in terms of time and accuracy.

The Novel Heuristic Algorithms:

Algorithm 1 For Full Demand Problem:

Algorithm 1 sample algorithm

Input: n total number of nodes, Set of nodes including uav V, set of small cells nodes only without uav C, b_{ij} links capacities, F_j maximum flows, L_j maximum links, d_c demands set, H maximum hops $\forall i, j \in V and c \in C$
Output: $Totalflow, Y_{ij}^k, D_{ij}, Z_k, Flow_{ij}$ $\forall i, j \in V, K \in C$
Initialization: Iprev$= [0, 0, 0, 0, 0]$, Jprev$= [0, 0, 0, 0, 0]$
$m = 0$
$totalflows = 0$

1: Determine the level of each node $j \in V$ such that $level_j$ represents level of node j in network
2: Arrange nodes of V in ascending order according to $level_j$.
3: $ARRANGEDNODES \leftarrow$ ascending order of V according to level
4: **if** C is not empty **then**
5:     Select node $k \in C$ which has the largest demand d_k among set C
6:     $C \leftarrow C - (k)$
7: **else**
8:     end the algorithm
9: **end if** {Iprev, Jprev are memory vectors to store values of i and j}
10: $i \leftarrow k$ {i is sending node}
11: $j = \emptyset$ {j is receiving node}
12: $Iprev = [0, 0, 0, 0, 0]$, $Jprev = [0, 0, 0, 0, 0]$
13: $m = 0$
14: **for** $l = 0$ to $l = (n - 1)$ **do**
15:     $r \leftarrow arrangednodes_l$
16:     **if** $((b_{i,r} \geq d_k)$ and $(D_{i,r} = 1 || (L_i \geq 1$ and $L_r \geq 1))$ and $(H - M \geq level_r)$ and $(F_r \geq 1))$ **then**
17:         $j \leftarrow r$
18:         BREAK FOR LOOP
19:     **end if**
20: **end for**
21: **if** $j \neq \emptyset$ and $j \neq (n - 1)$ **then**
22:     $Y_{i,j}^k \leftarrow 1$ {receiving end is found but still it is not the UAV}
23:     **if** $D_{i,j} \neq 1$ **then**
24:         $L_i \leftarrow L_i - 1$
25:         $L_j \leftarrow L_j - 1$
26:     **end if**
27:     $F_j \leftarrow F_j - 1$
28:     $b_{i,j} \leftarrow b_{i,j} - d_k$
29:     $b_{j,i} \leftarrow b_{j,i} - d_k$
30:     Iprev[m] $\leftarrow i$
31:     Jprev[m] $\leftarrow j$
32:     $i \leftarrow j$
33:     $m \leftarrow m + 1$
34:     $j = \emptyset$
35:     Go to line 10
36: **else if** $j \neq \emptyset$ and $j = (n - 1)$ **then**
37:     Do the same steps in line 18-25 {receiving end is found and it is the UAV}
38:     $Z_k = 1$
39:     $Totalflow = Totalflow + d_k$
40:     Assign $D_{i,j} = 1 : i, j \in selectedroute$
41:     Assign $Flow_{i,j} = Flow_{i,j} + d_k : i, j \in selectedroute$
42:     Go to line 4

else if $j = \emptyset$ and $i \neq k$ **then**
    $m = m - 1$ {receiving end is not found. route is blocked in middle so go one step back}
    $i \leftarrow$ Iprev[m]
    $j \leftarrow$ Jprev[m]
    $Y^k_{i,j} \leftarrow 0$
    if $D_{i,j} \neq 1$ **then**
        $L_i \leftarrow L_i + 1$
        $L_j \leftarrow L_j + 1$

1

    end if
    $F_j \leftarrow F_j + 1$
    $b_{i,j} \leftarrow b_{i,j} + d_k$
    $b_{j,i} \leftarrow b_{j,i} + d_k$
    $j = \emptyset$
    Go to line 10
else if $j = \emptyset$ and $i = k$ **then**
    $Z_k = 0$ {receiving end is not found so it is not possible to consider node k for demand transfer}
    Go to line 4
end if
if $C \neq \emptyset$ **then**
    GO to line 4
end if

Figure 7. Algorithm 1 for full demand

The Inputs And Outputs

Figure 7 represents Pseudo code of the developed algorithm 1. The inputs to the algorithm are the same inputs to the original model. The inputs are: -

- V set: It is the set of all nodes available in the network. The set includes both SCs & UAV nodes. The nodes are numbered from 0 to (n-1) such that $V = \{0,1,2, \dots . (n-1)\}$ where (n-1) is the UAV.
- F_j: The maximum flows (demands) which can be received by node j where $j \in V$
- L_j: The maximum number of links that can be connected to node j where $j \in V$.
- B: - it is N*N matrix composed of set of b_{ij} where b_{ij} represents the capacity of the link $(i,j) \in E$ in MB/s.
- H: - the maximum number of the allowed hops between the SC and the UAV.
- D: - a set of d_k such that d_k is the demand of node k where $k \in V \backslash \{n-1\}$.

The outputs (decision variables) of the algorithm are: -

- Y_{ij}^{k}: It determines whether the demand of node $k \in C$ is passed through the link (i, j) or not. Y_{ij}^{k} is equal to 1 if the link (i, j) is considered for demand transfer of node k. Otherwise Y_{ij}^{k} is equal to 0.
- Z_k: This variable determines whether the demand of node k will be transferred to UAV. Zk=1 indicates that the demand will be transferred.
- F_{ij}^{k}: Determines how much of kth SC's demand is transferred through link (i, j).
- Total flow: - it is the summation of all demands which are transferred to UAV. In other words, it is the summation of Φ(k) for all SCs such that $\sum_{k=0}^{k=(n-2)} \Phi(\mathrm{k})$
- D_{ij}: Indicates whether a link is existing between nodes i & j.

Depth-First Search Algorithms

DFS method is used in algorithm 1 to find a route between the SC and UAV. It is defined as an algorithm to find a route between two nodes by exploring each and every possible path as far as possible before backtracking. Although the way of implementing DFS will be self-explained through the steps of algorithm 1, you can refer to (Stanford, 2020) for more information about DFS.

The Time Complexity Of The Algorithm

As mentioned in the pseudo code, there are 2 main for loops in this algorithm. One loop to select the node with largest demand and the second loop to build the root to the UAV. The first loop can run for a maximum K times where K is equal to the number of SCs in the network. The other for loop can run for a maximum N times where N is equal to the total nodes number including UAV. So time complexity can be found using the formula $O(N * K)$ where N and K are total nodes and SCs numbers respectively.

The Steps Of The Algorithm:

1- The first step of the algorithm is to classify the nodes of the network into levels such that $level_j$ is the level of node $j \in V$. The level of a node is determined according to how far the node is located from UAV in terms of links. The more the links available between SC & UAV, the higher the level of SC. In all cases, the UAV is considered at level 0. The SCs which are directly connected to UAV (only one link is available between UAV and SC) are classified to level 1. The SCs which are connected to the nodes at level 1 (2 links are available between UAV & SC) are considered at level 2 … Etc.

2- The second step is to sort the nodes in ascending order according to their levels (from level 0 to the maximum level available) and assign the sorted order to the vector ARRANGEDNODES so $arrangednodes_0$ is the UAV node (level 0) and $arrangednodes_{n-1}$ is a node located at the highest available level.

3- Pick up a node k such that $k = V\backslash\{n-1\}$ has the largest demand among the other SCs in the network. Assign the node k to the variable i. Variable i indicates the node from which the demand is transferred (sending node). Hence, the priority for transferring the demand is for SCs with larger demands.

4- Using FOR loop, check the vector ARRANGEDNODE in ascending order from $arrangednodes_0$ to $arrangednodes_{n-1}$for node j. j will be the node that receives the demand sent by i (receiving node). Node j shall meet the following constrains: -

- $L_j \geq 1$ or $D_{ij} = 1$, at least one free vacancy for new link shall be available at node j or a link between nodes i & j is already existing)
- $F_j \geq 1$, at least one free vacancy at node j shall be available for new flow
- $b_{ij} \geq d_k$, the capacity of link between nodes i & j shall be equal to or bigger than demand of node k.
- $(H - m) \geq level_j$ where m is the hop number. This condition is created to ensure that the maximum hops H between the sending node and UAV is not exceeded.

Note that at step 4, the algorithm checks the nodes according to the level, starting from nodes with the lowest level to the nodes with highest level which means that the priority is to transfer the demand to the nodes which are closer to UAV. This procedure helps to reduce number of links (hops) between sending node and UAV which accordingly increases the reliability of the system.

5- After step 4, there are only 4 possible cases which are discussed as the following: -

- Case 1: At step 4, the algorithm succeeded to find node j that satisfies all the constrains. However, it is not the UAV node: -

In this case, the demand is transferred from node i (sending node) to node j (receiving node) as j has satisfied all the mentioned constrains. However, still the demand is not transferred to the desired node which is UAV (node (n-1)). Hence, after transferring the demand to j, it shall be transferred further to other nodes till it finally reaches the UAV. The current j node acts as a relay through which the demand of another SC is passed. The following decision variables shall be updated accordingly: -

- $Y_{ij}^{k} = 1$: As the demand of node k is passed through link (i, j), Y_{ij}^{k} shall be set to 1.
- $F_j = F_j - 1$: Node j is going to receive additional demand (flow) so one free vacancy for additional flows shall be detected.
- $b_{ij} = b_{ij} - d_k$: The capacity of the link (i, j) shall be decreased by the same amount of the demand that passes through it.
- $b_{ji} = b_{ji} - d_k$: b_{ij} & b_{ji} represent the capacity of the same link so the capacity of b_{ji} shall be decreased too.
- $Iprevm = i, Jprev_m = j$: Iprev and Jprev vectors are used to store the values of i and j at the different steps through the route as we may need these values in case of the blocked route (case 3)
- If $D_{ij} \neq 1$, $L_j = L_j - 1$ & $L_i = L_i - 1$: If there is no existing link between nodes i & j, then one vacancy for new link shall be consumed from both i & j.

As the node j will be the sending node in the following steps, node j shall be assigned as the new i node such that $i \leftarrow j$. Then, the algorithm moves back to step 4 where a new receiving node j is selected.

- Case 2: if there is no node j has been found at step 4 and still the demand did not leave node k (node with largest demand) such that $i = k$:

  In this case there is no possibility to transfer the demand of node k due to the fact that there is no receiving node j was found to meet all the constrains. The following steps are accomplished: -

- The node k is not considered for the demand transfer and Z_k is set to 0.

- The algorithm moves back to step 3 to select new node k.

- Case 3: If there is no node j has been found at step 4 and the demand has already transferred from node k to another SC such that $i \neq k$:

  This condition indicates that the algorithm could not find a node j that satisfies all constrains. This case occurs in the middle of the route after the demand has been transferred from the main node k to another SC. This means that the selected route is blocked in the middle and the only solution is to move back to the previous i node and select another node j instead of the current one. This step can be accomplished using the vectors Iprev & Jprev. These two vectors store all the previous values of i and j throughout the route. The algorithm performs the following steps in order to backtrack: the first step is to delete the link (i, j) between previous nodes i & j by performing the following steps:

  - $i = Iprev_m,\ j = Jprev_m$

  Assign the previous i&j nodes to the current i&j variables

  - $Y_{ij}^k = 0$: As the demand of node k is removed from link(i, j), Y_{ij}^k shall be set to 0.

  - $F_j = F_j + 1$
  - $b_{ij} = b_{ij} + d_k$
  - $b_{ji} = b_{ji} + d_k$
  - If $D_{ij} \neq 1$, $L_j = L_j + 1$ & $L_i = L_i + 1$

These 4 steps are to retrieve the resources which were consumed to establish the link between the previous i&j nodes.

After that, the algorithm go back to step 4 and find new j node in order to try different route.

For example, suppose that the algorithm built a route from 3 (main node k) to node 2 (case 1). After that it has been realized that the route from node 2 is blocked. In this case, the algorithm will cancel the link from 3 to 2 by implementing the previous mentioned steps. Then, algorithm will move back to step 4 to find another node j given that $i = 3$.

- Case 4: When $j = (n - 1)$

  This is the desired condition where the demand reaches UAV node (n-1). After the demand is finally going to reach UAV, the route will end & the final solution will be considered. The same updates of case 1 are accomplished here in addition to the following: -

- $Z_k = 1$: As the demand of node k is transferred to UAV, the decision variable Zk is set to 1.
- $totalflow = totalflow + d_k$: The demand of node k is added to the total throughput of the network.
- Set $D_{ij} = 1$ for all links (i, j) in the selected route

How Algorithm 1 Is Related To DFS Algorithm?

As we saw in algorithm 1, we select from a node k from the graph and at each step we move deeper through the path to another node with higher level till we reach finally to UAV. This represents the same principle of implementing DFS which is go as deep as possible before backtracking.

Algorithm 2 For Full Demand Problem

Algorithm 1 algorithm 2 for full demand

Input: n total number of nodes, set of arcs E, Set of nodes including uav V, set of small cells nodes only without uav C, b_{ij} links capacity, F_j maximum flows, L_j maximum links, d_c demands set, H maximum hops $\forall i, j \in V and c \in C$
Output: $Total flow, Y_{ij}^{k}, D_{ij}, Z_k \; \forall i, j \in V, K \in C$
Initialization: $total flows = 0$

1: Determine the level of each node $j \in V$ such that $level_j$ represents level of node j in network
2: **if** C is not empty **then**
3:    Select node $k \in C$ which has the largest demand d_k among set C
4:    $C \leftarrow C - (k)$
5: **else**
6:    end the algorithm
7: **end if**
8: **if** $L_k < 1$ **then**
9:    go to line 2 {checking the constrain of links of node k}
10: **end if**
11: **for all** $i \in (V - k)$ **do**
12:    **if** $((L_i < 2$ and $(F_i < 1))$ **then**
13:       $V \leftarrow V - (i)$ {checking the constrain of links and flows of other nodes}
14:    **end if**
15: **end for**
16: **for all** $(i, j) \in E$ **do**
17:    **if** $(b_{i,j} < d_k)$ **then**
18:       $E \leftarrow E - (i, j)$ {drop links with low capacities}
19:    **end if**
20: **end for**
21: **for all** $(i, j) \in E$ **do**
22:    **if** $(level_i = level_j)$ **then**
23:       $E \leftarrow E - (i, j)$ {drop links between nodes at same levels}
24:    **end if**
25: **end for**
26: Implement Dijkstra Algorithm to find shortest path using sets E and V as inputs
27: Dijkstra Algorithm generates set S, such that S contains links (i, j) of shortest path
28: **if** Number of links in S $\leq H$ **then**
29:    **for all** $(i, j) \leftarrow S$ **do**
30:       $Y_{i,j}^{k} \leftarrow 1$ {Updating decision variables and constrains}
31:       **if** $D_{i,j} \neq 1$ **then**
32:          $L_i \leftarrow L_i - 1$
33:          $L_j \leftarrow L_j - 1$
34:       **end if**
35:       $F_j \leftarrow F_j - 1$
36:       $b_{i,j} \leftarrow b_{i,j} - d_k$
37:       $b_{j,i} \leftarrow b_{j,i} - d_k$
38:       $Z_k = 1$
39:       $Total flow = Total flow + d_k$
40:       $D_{i,j} = 1$
41:    **end for**
42: **else**
43:    go to line 2
44: **end if**
45: GO to line 2

Figure 8. Algorithm 2 for full demand

The main idea of algorithm 2 is to use the technique of the finding the shortest path between the nodes and UAV. The unit of measuring the distance in our problem is link so the shortest possible route contains the minimum number of links. As mentioned before, the reliability of the system is dependent on the number of links between UAV and nodes.

Dijkstra Algorithm

Dijkstra algorithm is one of the most powerful algorithms to determine the shortest path between 2 nodes in a network. Dijkstra Algorithm will be used in Algorithm 2 as a tool to find the shortest path between node k & UAV. As we use Dijkstra Algorithm as a tool only and it is not the goal of the thesis, the steps of the algorithm will not be explained. Instead, the thesis is going to deal with the inputs and outputs of the algorithm only. For more information about Dijkstra algorithm, kindly refer to (Ahuja et al., 1993).

The Time Complexity Of Dijkstra Algorithm

As per (Ahuja et al., 1993), the time complexity of the algorithm can be found using the following formula: -

$$(|V| + |E|)\log(V)$$

Where V is the number of nodes and E is the number of links. It is clear that Dijkstra algorithm is faster than many other shortest path algorithms.

The Inputs And Outputs Of Algorithm

The algorithm has exactly same inputs and outputs of algorithm 1 as it is developed to solve the same model.

The Steps Of The Algorithm

1. Same as algorithm 1, the first step of the algorithm is to classify the nodes of the network into levels such that $level_j$ is the level of node $j \in V$. It is the same principle which has been used in algorithm 1.
2. Pick up a node k such that $k = V\backslash\{n-1\}$ has the largest demand among the other SCs in the network. Same as algorithm 1, the priority for demand transfer is for the nodes with larger demands.
3. Check that node k obeys the constrain of links which is: -

$$L_k \geq 1$$

Before transferring the demand of k, there should be a free vacancy for the new link. Otherwise go back to step 2.

4. Check all the other nodes in the set $V\backslash\{k\}$ for the following constrains to ensure that they are capable to build a route with node k: -

- $L_j \geq 2$ At least two free vacancies for new links shall be available at node j
- $F_j \geq 1$ At least one free vacancy for flows shall be available at node j.

In case that any of nodes in $V\backslash\{k\}$ does not obey any of the above constrains, the node will be excluded from set $V\backslash\{k\}$ so it will not be fed into Dijkstra Algorithm.

5. Remove all the links $(i,j) \in E$ which have capacity b_{ij} less than d_k : -

The set E contains the links (i,j) of the network which will be fed into Dijkstra Algorithm in order to find the shortest route. Hence, the capacities of all links shall be bigger than or equal to the demand of node k:

$$b_{ij} \geq d_k : (i,j) \in E \ \& \ k \in V\backslash\{n-1\}$$

6. Remove the all $(i,j) \in E$ which connect 2 nodes from the same level such that $level_i = level_j$

7. Implement Dijkstra algorithm to find the shortest route from node k to UAV node: -

   The inputs of the algorithm are: - set of nodes $V \setminus \{k\}$ (which contains the nodes that satisfied all the constrains except node k), node k & set of links E. The output of the algorithm will be the set S which contains the links of the shortest path between k and UAV such that $S \in E$.

8. Check that the shortest path between nodes k & UAV meets the condition of H, The total number of hops in the route shall not exceed maximum allowed hops H. If the number of links is more than H, move back to step 2.

9. Update the decision variables & constrains of the problem considering the shortest path as the final solution to transfer the demand of k. The following updates shall be performed: -

   - $Y_{ij}^k = 1 \qquad \forall i,j: (i,j) \in S$
   - $F_j = F_j - 1 \qquad \forall j: (i,j) \in S$
   - $b_{ij} = b_{ij} - d_k \qquad \forall i,j: (i,j) \in S$
   - $b_{ji} = b_{ji} - d_k \qquad \forall i,j: (i,j) \in S$
   - If $D_{ij} \neq 1$, $L_j = L_j - 1$ & $L_i = L_i - 1 \qquad \forall i,j: (i,j) \in S$
   - $Z_k = 1$
   - $totalflow = totalflow + d_k$
   - $D_{ij} = 1 \qquad \forall i,j: (i,j) \in S$

Remark

As we can notice from the example, Djikstra algorithm does not end up with one shortest route. Instead, it ends up with many alternative routes which have same length (number of links). This is due to the fact that the distances of links are equal (unity) among the network thus all the routes that pass through same levels are equal in length. Hence any route of them can be chosen randomly as the shortest path.

Comparison Of Algorithms 1 And 2

Table 1. Comparison between algorithms 1 & 2

Algorithm	Advantages	Disadvantages
1	The algorithm consider each and every possible root as it is similar to trial and error method.	Has higher time complexity than the second algorithm as it determines the single root in many steps and there is a possibility for backtracking.
2	Has lower time complexity (fast) which makes the algorithm suitable for applications which require fast continuous computation.	The algorithm does not consider the links between the nodes with same level which wastes many opportunities for throughput maximization.

Algorithm 3 For Partial Demand

Algorithm 3 is used to solve partial demand problem where 75-100% of the demand can be sent to UAV. Algorithm 3 has been derived from algorithm 1 and the steps of both algorithms are same except for step 3 where the largest node is selected.

Algorithm 1 algorithm 3 for partial demand

Input: n total number of nodes, Set of nodes including uav V, set of small cells nodes only without uav C, b_{ij} links capacities, F_j maximum flows, L_j maximum links, d_c demands set, H maximum hops $\forall i, j \in V$ and $c \in C$
Output: $Totalflow, Y_{ij}^k, D_{ij}, Z_k, Flow_{ij}$ $\forall i, j \in V, K \in C$
Initialization: Iprev= $[0,0,0,0,0]$, Jprev= $[0,0,0,0,0]$
$m = 0$
$totalflows = 0$

1: Determine the level of each node $j \in V$ such that $level_j$ represents level of node j in network
2: Arrange nodes of V in ascending order according to $level_j$.
3: $ARRANGEDNODES \leftarrow$ ascending order of V according to level
4: For each demand d_k consider the set of d_k^N where $N \in 1, 0.99, 0.98, \ldots.., 0.75$ such that $d_k^N = Nd_k$.
5: **if** C is not empty **then**
6:     Select node $k \in C$ which has the largest demand d_k^N among set C
7:     $C \leftarrow C - (k)$
8: **else**
9:     end the algorithm
10: **end if** {Iprev, Jprev are memory vectors to store values of i and j}
11: $i \leftarrow k$ {i is sending node}
12: $j = \emptyset$ {j is receiving node}
13: $Iprev = [0,0,0,0,0]$, $Jprev = [0,0,0,0,0]$
14: $m = 0$
15: **for** $l = 0$ to $l = (n-1)$ **do**
16:     $r \leftarrow arrangednodes_l$
17:     **if** $((b_{i,r} \geq d_k)$ and $(D_{i,r} = 1||(L_i \geq 1$ and $L_r \geq 1))$ and $(H - M \geq level_r)$ and $(F_r \geq 1))$ **then**
18:         $j \leftarrow r$
19:         BREAK FOR LOOP
20:     **end if**
21: **end for**

22: **if** $j \neq \emptyset$ and $j \neq (n-1)$ **then**
23:     $Y_{i,j}^k \leftarrow 1$ {receiving end is found but still it is not the UAV}
24:     **if** $D_{i,j} \neq 1$ **then**
25:         $L_i \leftarrow L_i - 1$
26:         $L_j \leftarrow L_j - 1$
27:     **end if**
28:     $F_j \leftarrow F_j - 1$
29:     $b_{i,j} \leftarrow b_{i,j} - d_k^N$
30:     $b_{j,i} \leftarrow b_{j,i} - d_k^N$
31:     Iprev[m] $\leftarrow i$
32:     Jprev[m] $\leftarrow j$
33:     $i \leftarrow j$
34:     $m \leftarrow m + 1$
35:     $j = \emptyset$
36:     Go to line 15
37: **else if** $j \neq \emptyset$ and $j = (n-1)$ **then**
38:     Do the same steps in line 23-30 {receiving end is found and it is the UAV}
39:     $Z_k = 1$
40:     $Totalflow = Totalflow + d_k^N$
41:     Assign $D_{i,j} = 1 : i, j \in selectedroute$
42:     Assign $Flow_{i,j} = Flow_{i,j} + d_k^N : i, j \in selectedroute$
43:     Go to line 5
44: **else if** $j = \emptyset$ and $i \neq k$ **then**
45:     $m = m - 1$ {receiving end is not found. route is blocked in middle so go one step back}
46:     $i \leftarrow$ Iprev[m]
47:     $j \leftarrow$ Jprev[m]
48:     $Y_{i,j}^k \leftarrow 0$

1

49:     **if** $D_{i,j} \neq 1$ **then**
50:         $L_i \leftarrow L_i + 1$
51:         $L_j \leftarrow L_j + 1$
52:     **end if**
53:     $F_j \leftarrow F_j + 1$
54:     $b_{i,j} \leftarrow b_{i,j} + d_k^N$
55:     $b_{j,i} \leftarrow b_{j,i} + d_k^N$
56:     $j = \emptyset$
57:     Go to line 15
58: **else if** $j = \emptyset$ and $i = k$ **then**
59:     $Z_k = 0$ {receiving end is not found so it is not possible to consider node k for demand transfer}
60:     Go to line 5
61: **end if**

Figure 9. Algorithm 3 for partial demand problem

It can be noticed that the code is almost identical to the code of algorithm 1 except for selecting node with largest demand. In partial demand problem, we have the option to reduce the satisfied demand into 75% of full demand.

In order to make algorithm 1 compatible with this feature, the following update is added: -

For each demand d_k consider the set of d_k^N where $N \in \{1, 0.99, 0.98, \ldots\ldots, 0.75\}$ such that $d_k^N = N \times d_k$.

By this way, for each node k, we create ((1-0.75)/0.01) = 25 different demands which are varying from 75-100% of the main d_k. Each demand is considered to be independent so it can be selected as the largest demand. Hence, instead of selecting the largest demand from {V-1} demands, now it is selected from 25*{V-1} demands. The remaining steps are identical to algorithm 1 except that we consider the variable d_k^N instead of d_k for updating totalflow, F_{ij}^k, and b_{ij}.

Chapter 4: Computational Study

Introduction And Chapter Content

In this section, computational analysis will be carried out to assess the performance of the proposed heuristics. First, we will consider the full demand model, and then we will consider the partial demand model. Also, the performance of the proposed algorithms will be assessed with respect to the maximum number of flows, links and hops.

All the studied topologies in this section are generated by a MATLAB code which was developed by (Almohammad, 2019).

Computational Analysis Of Heuristic Algorithms 1 And 2

Testing The Heuristic Algorithms On Small-sized Instances (less Than 35 Nodes)

As discussed in Chapter 3, the heuristic algorithms 1 and 2 are designed to maximize flow rate of full demand problem in which either the full demand of small cell is transferred to UAV or the demand of the small cell is not considered at all. Both approaches were tested on 85 small instances with number of nodes ranges from 5 to 27. Each instance is solved by both heuristic approaches and the objective values and the run times are reported. The optimal objective value of instance is obtained using CPLEX. The maximum numbers of links, flows and hops for all instances are 3, 3, and 3 respectively. Table 1 summarizes the results of the 85 instances.

Table 2. Performance of Heuristics 1 and 2 on small-sized instances

instance	NO of nodes	Optimal value (MB/s)	Solution provided by heuristic 1 (MB/s)	solution provided by heuristic 2 (MB/s)	Gap 1 (%)	Gap 2 (%)	Time 1 (s)	Time 2 (s)
1	5	220	220	220	0.0	0.0	< 0.5	< 0.5
2	7	270	270	270	0.0	0.0	< 0.5	< 0.5
3	7	220	220	220	0.0	0.0	< 0.5	< 0.5
4	8	340	340	340	0.0	0.0	< 0.5	< 0.5
5	8	260	230	230	11.5	11.5	< 0.5	< 0.5
6	8	300	300	300	0.0	0.0	< 0.5	< 0.5
7	9	250	250	250	0.0	0.0	< 0.5	< 0.5
8	9	200	200	200	0.0	0.0	< 0.5	< 0.5
9	10	380	380	350	0.0	7.9	< 0.5	< 0.5
10	10	390	390	390	0.0	0.0	< 0.5	< 0.5
11	10	490	490	490	0.0	0.0	< 0.5	< 0.5
12	10	510	510	430	0.0	15.7	< 0.5	< 0.5
13	10	410	410	410	0.0	0.0	< 0.5	< 0.5
14	10	380	380	320	0.0	15.8	< 0.5	< 0.5
15	10	390	330	380	15.4	2.6	< 0.5	< 0.5
16	11	270	270	270	0.0	0.0	< 0.5	< 0.5
17	11	280	260	260	7.1	7.1	< 0.5	< 0.5
18	11	490	490	490	0.0	0.0	< 0.5	< 0.5
19	11	70	70	70	0.0	0.0	< 0.5	< 0.5
20	12	660	660	580	0.0	12.1	< 0.5	< 0.5
21	12	400	370	350	7.5	12.5	< 0.5	< 0.5
22	12	520	520	520	0.0	0.0	< 0.5	< 0.5
23	12	340	300	340	11.8	0.0	< 0.5	< 0.5
24	12	380	330	380	13.2	0.0	< 0.5	< 0.5
25	12	580	580	580	0.0	0.0	< 0.5	< 0.5
26	12	300	300	300	0.0	0.0	< 0.5	< 0.5
27	13	590	590	590	0.0	0.0	< 0.5	< 0.5
28	13	680	680	680	0.0	0.0	< 0.5	< 0.5
29	13	700	700	700	0.0	0.0	< 0.5	< 0.5
30	13	470	470	470	0.0	0.0	< 0.5	< 0.5
31	14	340	320	300	5.9	11.8	< 0.5	< 0.5
32	14	610	580	580	4.9	4.9	< 0.5	< 0.5
33	15	620	620	620	0.0	0.0	< 0.5	< 0.5
34	15	310	310	280	0.0	9.7	< 0.5	< 0.5
35	15	530	500	530	5.7	0.0	< 0.5	< 0.5
36	16	520	490	490	5.8	5.8	< 0.5	< 0.5
38	16	470	470	470	0.0	0.0	< 0.5	< 0.5
39	16	580	580	580	0.0	0.0	< 0.5	< 0.5
40	16	210	190	190	9.5	9.5	< 0.5	< 0.5
41	16	200	200	200	0.0	0.0	< 0.5	< 0.5
42	16	270	270	270	0.0	0.0	< 0.5	< 0.5

instance	NO of nodes	Optimal value (MB/s)	Solution provided by heuristic 1 (MB/s)	solution provided by heuristic 2 (MB/s)	Gap 1 (%)	Gap 2 (%)	Time 1 (s)	Time 2 (s)
43	16	510	510	510	0.0	0.0	< 0.5	< 0.5
44	17	450	450	450	0.0	0.0	< 0.5	< 0.5
45	17	610	610	610	0.0	0.0	< 0.5	< 0.5
46	17	230	190	150	17.4	34.8	< 0.5	< 0.5
47	17	260	260	260	0.0	0.0	< 0.5	< 0.5
48	18	570	570	570	0.0	0.0	< 0.5	< 0.5
49	18	800	680	800	15.0	0.0	< 0.5	< 0.5
50	18	680	660	680	2.9	0.0	< 0.5	< 0.5
51	19	490	460	410	6.1	16.3	< 0.5	< 0.5
52	19	510	510	450	0.0	11.8	< 0.5	< 0.5
53	19	590	550	590	6.8	0.0	< 0.5	< 0.5
54	19	270	240	240	11.1	11.1	< 0.5	< 0.5
55	20	270	260	260	3.7	3.7	< 0.5	< 0.5
56	20	510	480	450	5.9	11.8	< 0.5	< 0.5
57	20	500	500	500	0.0	0.0	< 0.5	< 0.5
58	20	280	260	260	7.1	7.1	< 0.5	< 0.5
59	20	660	660	660	0.0	0.0	< 0.5	< 0.5
60	20	1040	1010	960	2.9	7.7	< 0.5	< 0.5
61	20	670	620	670	7.5	0.0	< 0.5	< 0.5
62	20	630	630	630	0.0	0.0	< 0.5	< 0.5
63	20	680	680	620	0.0	8.8	< 0.5	< 0.5
64	20	420	390	420	7.1	0.0	< 0.5	< 0.5
65	21	550	500	530	9.1	3.6	< 0.5	< 0.5
66	21	360	330	360	8.3	0.0	< 0.5	< 0.5
67	21	590	590	590	0.0	0.0	< 0.5	< 0.5
68	22	690	660	660	4.3	4.3	< 0.5	< 0.5
69	22	610	610	610	0.0	0.0	< 0.5	< 0.5
70	22	930	870	930	6.5	0.0	< 0.5	< 0.5
71	22	1020	920	990	9.8	2.9	< 0.5	< 0.5
72	22	1010	890	930	11.9	7.9	< 0.5	< 0.5
73	22	590	590	540	0.0	8.5	< 0.5	< 0.5
74	22	420	420	420	0.0	0.0	< 0.5	< 0.5
75	22	680	560	640	17.6	5.9	< 0.5	< 0.5
76	23	640	530	600	17.2	6.3	< 0.5	< 0.5
77	23	380	380	380	0.0	0.0	< 0.5	< 0.5
78	23	800	710	800	11.3	0.0	< 0.5	< 0.5
79	23	760	760	760	0.0	0.0	< 0.5	< 0.5
80	24	710	640	610	9.9	14.1	< 0.5	< 0.5
81	25	980	890	980	9.2	0.0	< 0.5	< 0.5
82	25	1210	1010	1210	16.5	0.0	< 0.5	< 0.5
83	25	830	760	830	8.4	0.0	< 0.5	< 0.5
84	26	740	700	740	5.4	0.0	< 0.5	< 0.5
85	27	640	610	590	4.7	7.8	< 0.5	< 0.5
				Average	4.0	3.5		

The gap of a heuristic approach is calculated using the formula: -

$$\frac{optimal\ value - objective\ value\ of\ heuristic\ approach}{optimal\ value} \times 100\ \%$$

For heuristic algorithm 1, it is observed that for 73 out of 85 instances, the heuristic achieved small gap of <= 10%. For the remaining 12 instances, it achieved a gap of <= 20%. The average gab for the 85 instances is 4%. The run time of the heuristic was < 0.5 for all instances. The exact timing cannot be determined due to the limitation of the clock function of C++ program. Shifting to heuristic algorithms 2, the algorithm achieved a gap of <=10% for 73 instances and a gap of <= 80% for 11 samples. For one sample only, the gap between the objective value of the heuristic and the optimal value was 34.8% which considered to be one of rare exceptional cases where the considered approach achieves non-accepted result. The run time of all instances did not exceed 0.5 s. Finally, the average gap was 3.5 which is slightly less than heuristic 1. The run time of CPLEX to obtain the optimal solution (solving the model) varied from 4 seconds to 3 minutes based on the size of the instant. This proves the efficiency of both heuristic algorithms as they both took less than 0.5 seconds to provide the solution regardless of the size of the instant.

Sensitivity Analysis Of The Performance Of The Heuristics

In this section, the accuracy of both heuristic algorithms 1 & 2 will be analysed with respect to the different parameters. As discussed in chapter 3, the available constraints in multi-hop backhauling network are: -

- Maximum number of links (L): - represents the maximum number of nodes that could be connected to a small cell node.
- Maximum number of flows (F): - represents the maximum number of flows which can be received by a small cell at an instant.

- Maximum number of hops (H): - represents the maximum allowed transitions between a small cell and NFP node.

The three mentioned parameters will be varied from 1 to 3 and the effect of variation will be analysed for both the objective value and run time. The aim is to observe the general effect of limiting the available resources of the network.

Eight instances with different number of nodes (16-33 nodes) are analysed for this purpose. This section consists of three parts. Each part studys the effect of a different parameter.

The Effect Of Varying The Number Of Maximum Links Per Small Cell:

Table 3. Effect of L on the accuracy of heuristic algorithm

Ins.	Sam.	L	Optimal value (MB/s)	Solution provided by heuristic 1 (MB/s)	Time 1 (s)	Gap 1 (%)	Solution provided by heuristic 2 (MB/s)	Time 2 (s)	Gap 2 (%)
1	1	3	240.00	240.00	<0.5	0.00	200.00	<0.5	16.67
2	1	2	190.00	190.00	<0.5	0.00	160.00	<0.5	15.79
3	1	1	70.00	70.00	<0.5	0.00	70.00	<0.5	0.00
4	2	3	170.00	170.00	<0.5	0.00	170.00	<0.5	0.00
5	2	2	170.00	170.00	<0.5	0.00	130.00	<0.5	23.53
6	2	1	100.00	100.00	<0.5	0.00	100.00	<0.5	0.00
7	3	3	410.00	370.00	<0.5	9.76	370.00	<0.5	9.76
8	3	2	390.00	370.00	<0.5	5.13	370.00	<0.5	5.13
9	3	1	350.00	350.00	<0.5	0.00	350.00	<0.5	0.00

Ins tan ce	Sam .	L	Optimal value (MB/s)	Solution provided by heuristic 1 (MB/s)	Time 1 (s)	Gap 1 (%)	Solution provided by heuristic 2 (MB/s)	Time 2 (s)	Gap 2 (%)
10	4	3	300.00	300.00	<0.5	0.00	300.00	<0.5	0.00
11	4	2	260.00	190.00	<0.5	26.92	220.00	<0.5	15.38
12	4	1	200.00	200.00	<0.5	0.00	200.00	<0.5	0.00
13	5	3	310.00	290.00	<0.5	6.45	300.00	<0.5	3.23
14	5	2	290.00	260.00	<0.5	10.34	290.00	<0.5	0.00
15	5	1	200.00	200.00	<0.5	0.00	200.00	<0.5	0.00
16	6	3	280.00	250.00	<0.5	10.71	280.00	<0.5	0.00
17	6	2	270.00	250.00	<0.5	7.41	250.00	<0.5	7.41
18	6	1	150.00	150.00	<0.5	0.00	150.00	<0.5	0.00
19	7	3	340.00	320.00	<0.5	5.88	320.00	<0.5	5.88
20	7	2	320.00	320.00	<0.5	0.00	310.00	<0.5	3.13
21	7	1	220.00	220.00	<0.5	0.00	220.00	<0.5	0.00
22	8	3	380.00	380.00	<0.5	0.00	340.00	<0.5	10.53
23	8	2	370.00	300.00	<0.5	18.92	310.00	<0.5	16.22
24	8	1	190.00	190.00	<0.5	0.00	190.00	<0.5	0.00

Table 2 represents the effect of varying L constrain on the accuracy and run time of both heuristic algorithms. It is very clear from the table that there is no effect on the run time. The run time is < 0.5 for all cases. As discussed earlier, the exact timing cannot be observed due to the limitation of clock function of C++ program. Shifting to the accuracy, Figure 11& 12 summarizes the effect of varying number of L on both heuristics 1 & 2 respectively.

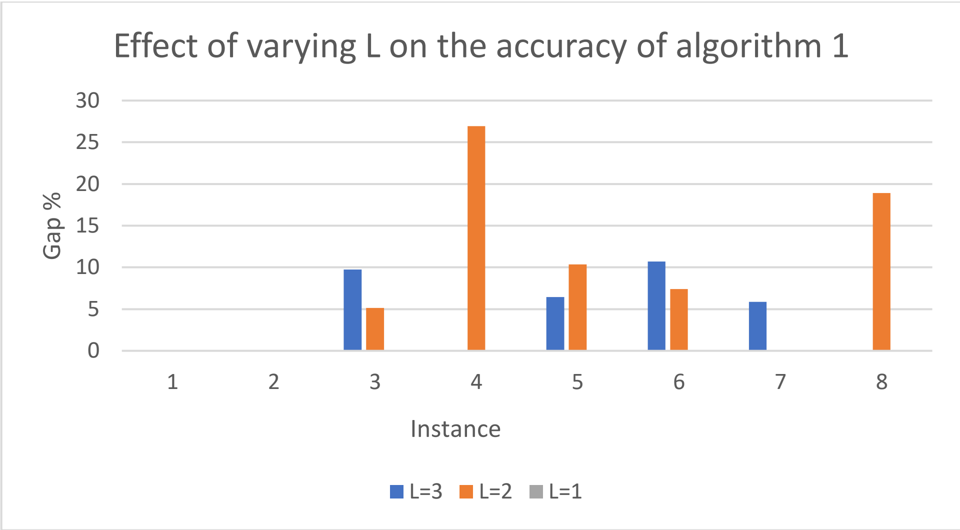

Figure 10. Effect of varying L on the accuracy of algorithm 1

It is clear from Figure 11 that there is no particular unified effect for increasing or decreasing the number of maximum links on the accuracy. For example, in sample 4, the gap between optimal value and heuristic objective value increased from 0% to almost 27% when L is reduced from 3 to 2. Then, gap decreased again to 0% when L is reduced further to 1.

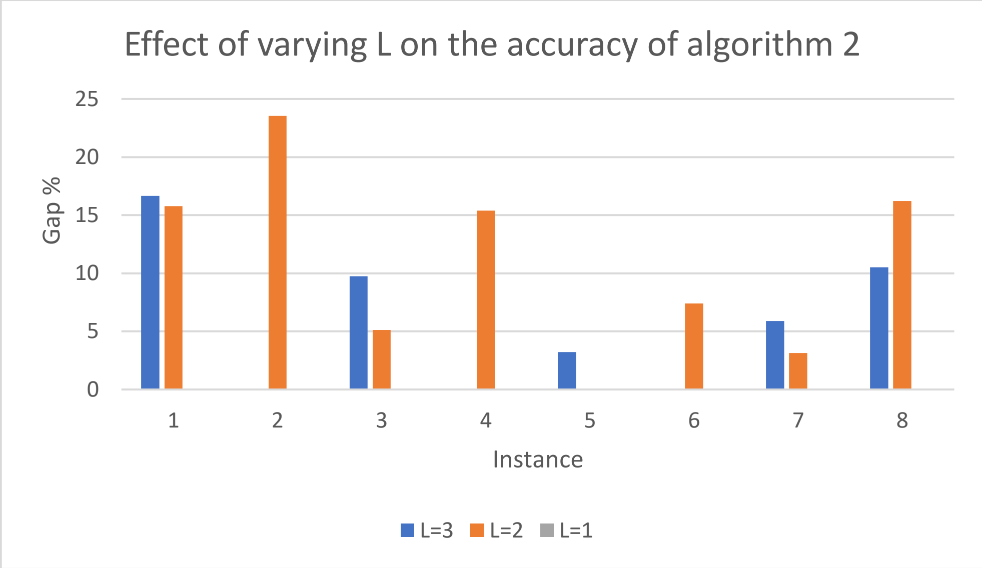

Figure 11. Effect of varying L on the accuracy of heuristic algorithm 2

The same situation is observed with heuristic 2. Figure 12 does not show any unified effect with increasing or decreasing L. For example, in instant 4, the gap increased from 0% to 15.4% when L is decreased from 3 to 2. However, the gap decreased again to 0% when L is decreased further to 1.

The Effect Of Varying The Number Maximum Flows Which Can Be Received By a Small Cell At An Instant

Table 4. Effect of varying F on the accuracy of heuristic algorithms

Inst.	Sam.	F	Optimal value (MB/s)	Solution provided by heuristic 1 (MB/s)	Time 1 (s)	Gap 1 (%)	Solution provided by heuristic 2 (MB/s)	Time 2 (s)	Gap 2 (%)
1	1	3	240.00	240.00	<0.5	0.00	200.00	<0.5	16.67
2	1	2	240.00	240.00	<0.5	0.00	200.00	<0.5	16.67
3	1	1	220.00	200.00	<0.5	9.09	200.00	<0.5	9.09
4	2	3	170.00	170.00	<0.5	0.00	170.00	<0.5	0.00

Instance	Sam.	F	Optimal value (MB/s)	Solution provided by heuristic 1 (MB/s)	Time 1 (s)	Gap 1 (%)	Solution provided by heuristic 2 (MB/s)	Time 2 (s)	Gap 2 (%)
5	2	2	170.00	170.00	<0.5	0.00	170.00	<0.5	0.00
6	2	1	170.00	170.00	<0.5	0.00	170.00	<0.5	0.00
7	3	3	410.00	370.00	<0.5	9.76	370.00	<0.5	9.76
8	3	2	410.00	370.00	<0.5	9.76	370.00	<0.5	9.76
9	3	1	390.00	370.00	<0.5	5.13	370.00	<0.5	5.13
10	4	3	300.00	300.00	<0.5	0.00	300.00	<0.5	0.00
11	4	2	300.00	300.00	<0.5	0.00	300.00	<0.5	0.00
12	4	1	300.00	300.00	<0.5	0.00	300.00	<0.5	0.00
13	5	3	310.00	290.00	<0.5	6.45	300.00	<0.5	3.23
14	5	2	310.00	290.00	<0.5	6.45	300.00	<0.5	3.23
15	5	1	300.00	290.00	<0.5	3.33	300.00	<0.5	0.00
16	6	3	280.00	250.00	<0.5	10.71	280.00	<0.5	0.00
17	6	2	280.00	250.00	<0.5	10.71	280.00	<0.5	0.00
18	6	1	280.00	250.00	<0.5	10.71	280.00	<0.5	0.00
19	7	3	340.00	320.00	<0.5	5.88	320.00	<0.5	5.88
20	7	2	340.00	320.00	<0.5	5.88	320.00	<0.5	5.88
21	7	1	320.00	320.00	<0.5	0.00	320.00	<0.5	0.00
22	8	3	380.00	380.00	<0.5	0.00	340.00	<0.5	10.53
23	8	2	380.00	380.00	<0.5	0.00	340.00	<0.5	10.53
24	8	1	380.00	380.00	<0.5	0.00	340.00	<0.5	10.53

Table 3 represents the effect of variation of F constrain on the accuracy and run times of both heuristic algorithms. It is very clear from table that there is no effect on the run time. Figures 13 & 14 summarizes the effect of varying number of F on both heuristics 1 & 2 respectively.

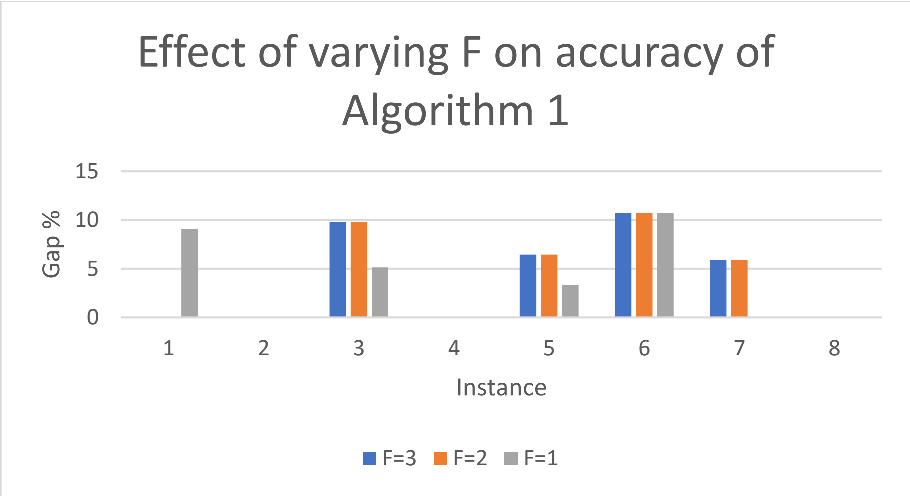

Figure 12. Effect of varying F on the accuracy of heuristic algorithm 1

Figures 13 & 14 represent the effect of varying F on the accuracy of heuristic approaches 1 & 2 respectively. It can be noticed that the gap and F are directly proportional in instances 3, 5 & 7. However, this pattern is violated by instance 1 in Figure 13 which shows inverse relation between gap and F when F is decreased from 2 to 1. Accordingly, no specific pattern can be concluded.

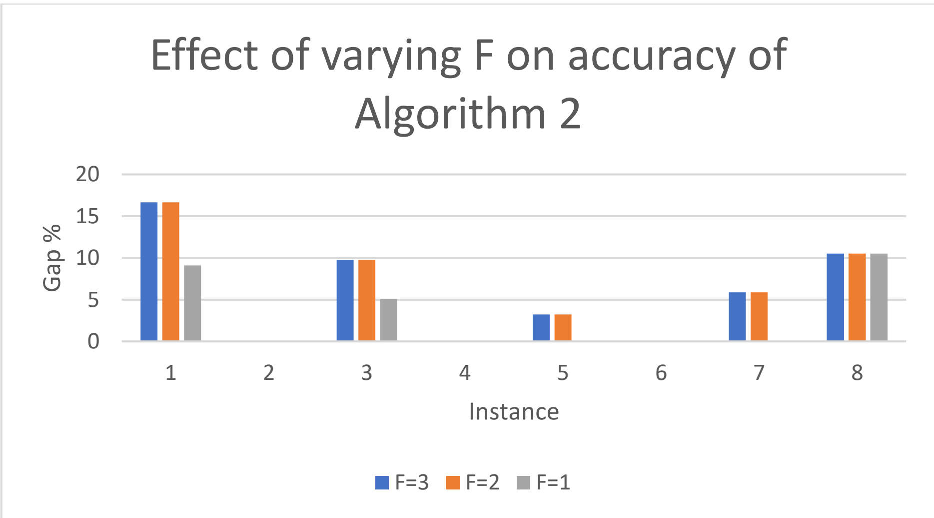

Figure 13. Effect of varying F on the accuracy of heuristic approach 2

__The Effect Of Varying The Number Of Maximum Hops Per SC:__

Table 5. Effect of varying H on the accuracy of heuristic approaches 1 &2

Ins.	Sam.	H	Optimal value (MB/s)	Solution provided by heuristic 1 (MB/s)	Time 1 (s)	Gap 1 (%)	Solution provided by heuristic 2 (MB/s)	Time 2 (s)	Gap 2 (%)
1	1	3	240	240	<0.5	0	200	<0.5	16.67
2	1	2	220	220	<0.5	0	220	<0.5	0.00
3	1	1	70	70	<0.5	0	70	<0.5	0.00
4	2	3	170	170	<0.5	0	170	<0.5	0
5	2	2	170	170	<0.5	0	170	<0.5	0
6	2	1	100	100	<0.5	0	100	<0.5	0
7	3	3	410	370	<0.5	9.76	370	<0.5	9.76
8	3	2	370	370	<0.5	0.00	370	<0.5	0.00
9	3	1	350	350	<0.5	0.00	350	<0.5	0.00
10	4	3	300	300	<0.5	0	300	<0.5	0
11	4	2	290	290	<0.5	0	290	<0.5	0
12	4	1	200	200	<0.5	0	200	<0.5	0
13	5	3	310	290	<0.5	6.45	300	<0.5	3.23
14	5	2	270	270	<0.5	0.00	270	<0.5	0.00
15	5	1	200	200	<0.5	0.00	200	<0.5	0.00
16	6	3	280	250	<0.5	10.71	280	<0.5	0.00
17	6	2	250	210	<0.5	16.00	240	<0.5	4.00
18	6	1	150	150	<0.5	0.00	150	<0.5	0.00
19	7	3	340	320	<0.5	5.88	320	<0.5	5.88
20	7	2	330	320	<0.5	3.03	320	<0.5	3.03
21	7	1	220	220	<0.5	0.00	220	<0.5	0.00
22	8	3	380	380.00	<0.5	0.00	340.00	<0.5	10.53
23	8	2	340	340.00	<0.5	0.00	340.00	<0.5	0.00
24	8	1	190	190.00	<0.5	0.00	190.00	<0.5	0.00

Table 4 represents the effect of varying H constrain on the accuracy and run times of both heuristic algorithms. Same as previous cases, the run time is fixed among all instances (<0.5 s). Figures 15 & 16 summarizes the effect of varying H on the accuracy of both heuristics 1 & 2 respectively.

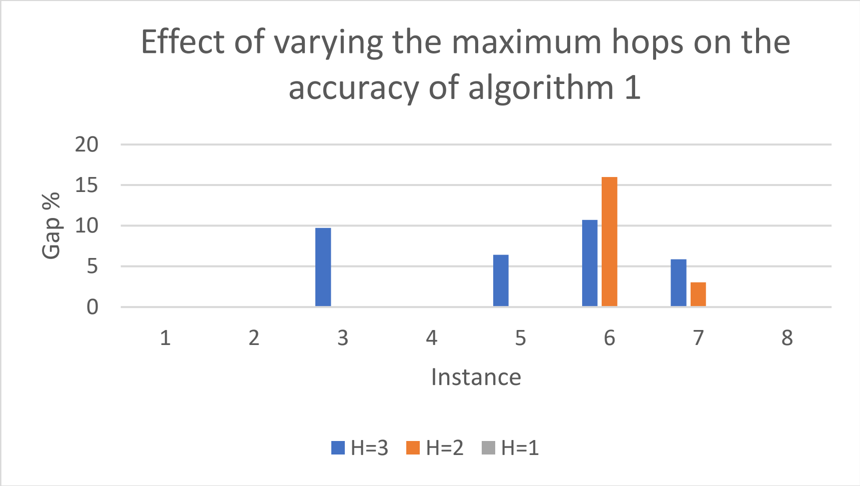

Figure 14. Effect of varying H on heuristic approach 1

Figures 15 & 16 represent the effect of varying H on the accuracy of heuristic approaches 1 & 2 respectively. It can be noticed that there is direct relation between accuracy gap and H in instances 3, 5 & 7. However, this pattern is violated by sample 6 which shows inverse relation between accuracy gap and H when H is decreased from 3 to 2.

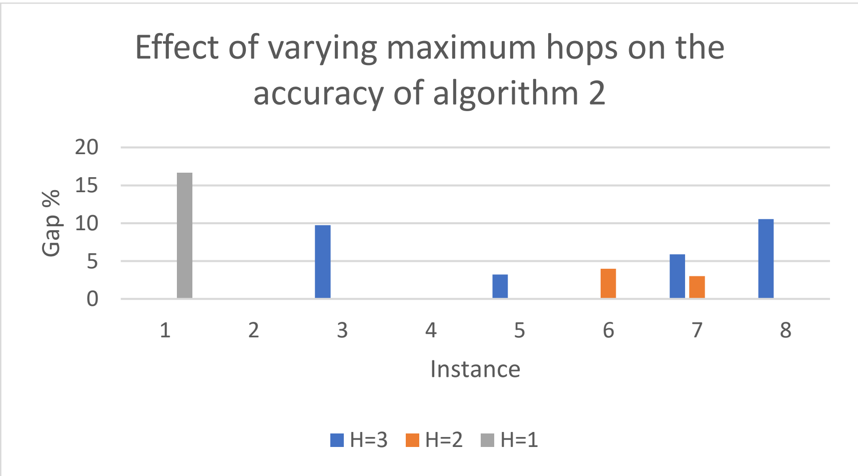

Figure 15. Effect of varying H on heuristic approach 2

The Effect Of Varying The Number Of Edges On The Accuracy Of Heuristic Approaches

In this part, we will study the effect of varying the number of edges between nodes on the accuracy of heuristic approaches. The number of nodes will be kept constant while number of edges will be varied. Note that changing the number of edges between nodes will affect the architecture of network and hence generate a new network which is totally different from the previous one. Accordingly, the aim is to perform the following: -

- Prove that both heuristic approaches still provide good accuracies even if the number of edges increases (the network becomes denser).
- Study the effect of increasing No. of edges on the run time.

In this part, 9 groups will be analysed and each of them consists of 3-4 instances. All the instances at each group have the same number of nodes, but with different number of edges. Both heuristic approaches will be applied on all samples and the results will be compared to the optimum values obtained by using CPLEX. Table 5 summarizes the results of this section.

Table 6. Varying number of edges among the samples

Instance	Edges	Optimal value (MB/s)	Solution provided by heuristic 1 (MB/s)	Time 1 (s)	Gap 1 (%)	Solution provided by heuristic 2 (MB/s)	Time 2 (s)	Gap 2 (%)
1	24.00	120.00	120.00	<0.5	0.00	120.00	<0.5	0.00
2	30.00	210.00	210.00	<0.5	0.00	150.00	<0.5	28.57
3	44.00	160.00	160.00	<0.5	0.00	160.00	<0.5	0.00
4	60.00	220.00	220.00	<0.5	0.00	220.00	<0.5	0.00

Instance	Edges	Optimal value (MB/s)	Solution provided by heuristic 1 (MB/s)	Time 1 (s)	Gap 1 (%)	Solution provided by heuristic 2 (MB/s)	Time 2 (s)	Gap 2 (%)
1	40.00	0.00	0.00	<0.5	0.00	0.00	<0.5	0.00
2	54.00	240.00	240.00	<0.5	0.00	200.00	<0.5	16.67
3	74.00	350.00	300.00	<0.5	14.29	310.00	<0.5	11.43
4	88.00	280.00	280.00	<0.5	0.00	280.00	<0.5	0.00
5	48.00	170.00	170.00	<0.5	0.00	170.00	<0.5	0.00
6	56.00	160.00	160.00	<0.5	0.00	160.00	<0.5	0.00
7	64.00	60.00	60.00	<0.5	0.00	60.00	<0.5	0.00
8	72.0	130.00	130.00	<0.5	0.00	130.00	<0.5	0.00
9	76.00	160.00	120.00	<0.5	25.00	120.00	<0.5	25.00
10	100.00	410.00	370.00	<0.5	9.76	370.00	<0.5	9.76
11	110.00	320.00	320.00	<0.5	0.00	320.00	<0.5	0.00
12	142.00	480.00	480.00	<0.5	0.00	400.00	<0.5	16.67
13	88.00	80.00	80.00	<0.5	0.00	80.00	<0.5	0.00
14	110.00	300.00	300.00	<0.5	0.00	300.00	<0.5	0.00
15	146.00	260.00	260.00	<0.5	0.00	210.00	<0.5	19.23
16	180.00	600.00	600.00	<0.5	0.00	600.00	<0.5	0.00
17	88.00	110.00	110.00	<0.5	0.00	110.00	<0.5	0.00
18	106.00	310.00	290.00	<0.5	6.45	300.00	<0.5	3.23
19	130.00	240.00	240.00	<0.5	0.00	240.00	<0.5	0.00
20	172.00	630.00	580.00	<0.5	7.94	530.00	<0.5	15.87
21	142.00	200.00	200.00	<0.5	0.00	200.00	<0.5	0.00
22	160.00	130.00	100.00	<0.5	23.08	100.00	<0.5	23.08

Instance	Edges	Optimal value (MB/s)	Solution provided by heuristic 1 (MB/s)	Time 1 (s)	Gap 1 (%)	Solution provided by heuristic 2 (MB/s)	Time 2 (s)	Gap 2 (%)
23	190.00	280.00	250.00	<0.5	10.71	280.00	<0.5	0.00
24	234.00	390.00	360.00	<0.5	7.69	390.00	<0.5	0.00
25	200.00	260.00	260.00	<0.5	0.00	260.00	<0.5	0.00
26	230.00	340.00	320.00	<0.5	5.88	320.00	<0.5	5.88
27	264.00	400.00	370.00	<0.5	7.50	350.00	<0.5	12.50
28	294.00	780.00	750.00	<0.5	3.85	750.00	<0.5	3.85
29	190.00	240.00	240.00	<0.5	0.00	240.00	<0.5	0.00
30	234.00	190.00	190.00	<0.5	0.00	190.00	<0.5	0.00
31	250.00	380.00	380.00	<0.5	0.00	340.00	<0.5	10.53
32	328.00	820.00	740.00	<0.5	9.76	780.00	<0.5	4.88

As shown in table 5**,** the run time of both heuristic approaches did not exceed 0.5 seconds for all instances among all groups. This proves the efficiency of both algorithms regardless of how dense the networks are. Furthermore, both algorithms generate near-optimal solutions for most of the instances with gap values vary from 0% to 20%. Algorithm 1 provided 2 objective values with low accuracy which are instance 2 of group 7 (23.08% gap) & instance 1 of group 4 (25.00% gap). Shifting to algorithm 2, it shows 3 cases with poor accuracy which are sample 2 of group 7 (23.08% gap), sample 1 of group 4 (25.00% gap) & sample 2 of group 1 (28.57% gap). Table 6 & Figure 17 represent the average accuracy gaps of the 9 groups.

Table 7. Results of heuristic approaches 1 & 2

Group	average Gap of heuristic algorithm 1 (%)	average Gap of heuristic algorithm 2 (%)
1	0.00	7.14
2	3.57	7.02
3	0.00	0.00
4	8.69	12.86
5	0.00	4.81
6	3.60	4.77
7	10.37	5.77
8	4.31	5.56
9	2.44	3.85

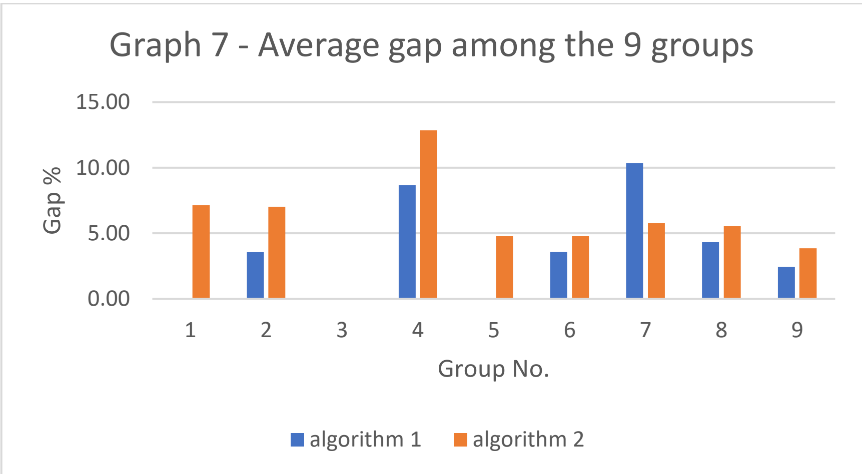

Figure 16. Summary of the results of heuristic algorithms 1 & 2

Testing Heuristic Approaches 1 & 2 On Large-sized Instances (> 70 node)

As mentioned earlier, It is difficult to obtain the optimal values of instances with more than 35 nodes as they take extremely longer time to be solved by CPLEX. Instead, the solutions provided by heuristics 1 & 2 will be compared to the upper bound values derived by column generation algorithm which developed by Msakni (2020). Run times of both algorithms will be checked in order to ensure the efficiency.

In this section, 28 large instances with different number of nodes (70-168) will be analysed.

Table 8. Testing heuristic algorithms on large instances

Testing heuristics algorithms on large samples (70-168 nodes). Given that H=3, L=3, F=3								
Instance	Number of nodes	Upper bound by CG	solution provided by heuristic 1 (MB/s)	solution provided by heuristic 2 (MB/s)	Gap 1 %	Gap 2 %	T1 (s)	T2 (s)
1	70	741.74	490	610	33.94	17.76	< 0.5	< 0.5
2	70	1155.2	1020	1030	11.71	10.84	< 0.5	< 0.5
3	70	932.44	810	830	13.13	10.99	< 0.5	< 0.5
4	70	1157.5	940	920	18.80	20.52	< 0.5	< 0.5
5	72	1209.9	1040	1040	14.05	14.05	< 0.5	< 0.5
6	72	1155.1	1000	1010	13.43	12.57	< 0.5	< 0.5
7	72	1314.7	1100	1090	16.33	17.09	< 0.5	< 0.5
8	72	1131.9	900	930	20.49	17.84	< 0.5	< 0.5
9	77	1152.8	980	980	14.99	14.99	< 0.5	< 0.5
10	77	1232.5	1070	1040	13.18	15.62	< 0.5	< 0.5
11	77	773.15	660	670	14.63	13.34	< 0.5	< 0.5
12	77	1082.0	980	970	9.43	10.35	< 0.5	< 0.5
13	90	1319.9	1100	1130	16.66	14.39	0.5	< 0.5

Instance	Number of nodes	Upper bound by CG	solution provided by heuristic 1 (MB/s)	solution provided by heuristic 2 (MB/s)	Gap 1 %	Gap 2 %	T1 (s)	T2 (s)
14	90	854.60	770	780	9.90	8.73	< 0.5	< 0.5
15	90	1032.1	910	910	11.83	11.83	< 0.5	< 0.5
16	90	881.91	740	740	16.09	16.09	< 0.5	< 0.5
17	92	999.30	890	890	10.94	10.94	< 0.5	< 0.5
18	92	860.22	710	750	17.46	12.81	< 0.5	< 0.5
19	92	1042.4	920	940	11.75	9.83	< 0.5	< 0.5
20	92	1125.6	1000	1000	11.16	11.16	< 0.5	< 0.5
21	102	1107.5	1010	980	8.80	11.51	< 0.5	< 0.5
22	102	962.23	860	860	10.62	10.62	< 0.5	< 0.5
23	102	599.71	500	500	16.63	16.63	< 0.5	< 0.5
24	102	1281.1	1080	1080	15.70	15.70	< 0.5	< 0.5
25	103	1440.2	1250	1170	13.21	18.76	< 0.5	< 0.5
27	103	1194.5	1050	1050	12.10	12.10	< 0.5	< 0.5
28	103	786.02	710	710	9.67	9.67	< 0.5	< 0.5
				Average gap	14.32	13.58		

Table **7** shows that the run times of both heuristic algorithms did not exceed 0.5 seconds for all instances, and this proves the efficiency of both algorithms. Also, It is noticed that for most of instances, that the gap value did not exceed 20% given that the optimal throughput value is located somewhere below the upper bound. It is very remarkable that both algorithms took only a portion of a second to solve the instances which proves the efficiency of algorithms.

Computational Analysis Of Heuristic Algorithm 3

In full demand problem, it was compulsory to transfer the whole demand of a small cell in case that the same cell has been selected for maximization. In partial demand problem, a minimum percentage of 70% of the demand can be transferred in case that the SC has been selected for maximization. This implements some sort of flexibility to the problem and accordingly the total throughput increases. The problem has the same constrains of full demand problem which are L, F & H. partial demand problem is solved using heuristic algorithm 3.

Testing H3 On Small Samples Which Contain Up To 16 Nodes

In this section, a total of 74 instances will be solved by heuristic algorithm 3. As explained in chapter 2**,** partial demand problem has more constrains than full demand problem which makes it more difficult be solved for the optimal values. In other words, CPLEX program takes longer time in calculating the optimized throughput of a network. Unfortunately, the optimal values could be computed for the samples with a maximum of 16 nodes only. For the samples with more than 16 nodes, CPLEX program took extremely longer time to solve. Table 8 represents the results of testing algorithm 3 on small instants. The values of L, F & H were set to 3.

Table 9. testing heuristic algorithm 3 on small samples

Instance	Number of nodes	Optimal value (B/s)	solution provided by heuristic 3 (MB/s)	Gap (%)	Time (s)
1	5	250000000	250000000	0.00	< 0.5
2	5	220000000	220000000	0.00	< 0.5

Instance	Number of nodes	Optimal value (B/s)	solution provided by heuristic 3 (MB/s)	Gap (%)	Time (s)
3	6	140000000	140000000	0.00	< 0.5
4	6	120000000	120000000	0.00	< 0.5
5	7	200000000	200000000	0.00	< 0.5
6	7	330000000	330000000	0.00	< 0.5
7	7	77354857	49150000	36.46	< 0.5
8	7	49110767	49080100	0.06	< 0.5
9	7	270000000	270000000	0.00	< 0.5
10	7	220000000	220000000	0.00	< 0.5
11	8	85196911	80500100	5.51	< 0.5
12	8	0	0	0.00	< 0.5
13	8	129484839	79750000	38.41	< 0.5
14	8	106791325	104000000	2.61	< 0.5
15	8	279277661	230000000	17.64	< 0.5
16	8	300000000	300000000	0.00	< 0.5
17	9	350000000	350000000	0.00	< 0.5
18	9	229108075	225080000	1.76	< 0.5
19	9	70000000	70000000	0.00	< 0.5
20	9	296978913	290550000	2.16	< 0.5
21	9	0	0	0.00	< 0.5
22	9	334262216	270880000	18.96	< 0.5
23	10	200000000	200000000	0.00	< 0.5
24	10	266699693	254900000	4.42	< 0.5
25	10	254008996	254000000	0.00	< 0.5
26	10	109480119	80000000	26.93	< 0.5
27	10	429161878	429150000	0.00	< 0.5
28	10	390000000	390000000	0.00	< 0.5
29	10	490000000	490000000	0.00	< 0.5
30	10	458087779	418050000	8.74	< 0.5
31	10	458272654	452960000	1.16	< 0.5
32	10	400004497	330000000	17.50	< 0.5
33	11	134388302	134350000	0.03	< 0.5
34	11	467528589	382480000	18.19	< 0.5
35	11	161502901	130000000	19.51	< 0.5
36	11	270000000	270000000	0.00	< 0.5
37	11	336958786	294560000	12.58	< 0.5
38	11	418379328	372050000	11.07	< 0.5
39	11	204179064	180000000	11.84	< 0.5
40	11	350421297	350400000	0.01	< 0.5
41	11	340000000	310000000	8.82	< 0.5
42	11	318819179	318780000	0.01	< 0.5
43	11	294070939	280000000	4.78	< 0.5
44	11	125495765	115200000	8.20	< 0.5

Instance	Number of nodes	Optimal value (B/s)	solution provided by heuristic 3 (MB/s)	Gap (%)	Time (s)
45	12	263728081	263700000	0.01	< 0.5
46	12	175804136	130000000	26.05	< 0.5
47	12	58687162	50000000	14.80	< 0.5
48	12	272069349	266680000	1.98	< 0.5
49	12	239750697	239720000	0.01	< 0.5
50	12	120000000	120000000	0.00	< 0.5
51	12	520000000	520000000	0.00	< 0.5
52	12	359333660	345900000	3.74	< 0.5
53	13	130510426	119600000	8.36	< 0.5
54	13	137366917	119850000	12.75	< 0.5
56	13	356213806	349560000	1.87	< 0.5
57	13	60000000	60000000	0.00	< 0.5
58	13	590000000	590000000	0.00	< 0.5
59	13	680000000	680000000	0.00	< 0.5
60	13	700000000	700000000	0.00	< 0.5
61	13	470000000	470000000	0.00	< 0.5
62	14	53336936	50000000	6.26	< 0.5
63	14	310000000	310000000	0.00	< 0.5
64	14	435131280	435100000	0.01	< 0.5
65	14	319064732	269040000	15.68	< 0.5
66	15	116648693	114850000	1.54	< 0.5
67	15	318677327	314300000	1.37	< 0.5
68	15	140727729	130000000	7.62	< 0.5
69	15	196226780	170000000	13.37	< 0.5
70	15	136365909	136340000	0.02	< 0.5
71	15	620000000	620000000	0.00	< 0.5
72	15	314863737	310000000	1.54	< 0.5
73	15	530000000	500000000	5.66	< 0.5
74	16	80643693	78820100	2.26	< 0.5
			Average accuracy	5.51	

Table 8 shows a good average gap of 5.51%. The algorithm achieved a gap of less than or equal 10% for 58 samples. The algorithm represented a poor accuracy for instance 46 (26.05% gap) & instance 26 (26.93% gap). The run time of all trials did not exceed 0.5 seconds. The run time of CPLEX to obtain the optimal solution (solving the model) varied from couple of seconds to couple of minutes based on the size of the instant. This proves the efficiency of both heuristic algorithms as they both took less than 0.5 seconds to provide the solution regardless of the size of the instant.

Sensitivity Analysis Of The Performance Of The Heuristic Algorithm

In this section, the three mentioned constrains will be varied from 3 to 1 and the effect of variation will be analysed with respect to objective value and run time. The aim is to observe the effect of limiting the available resources on the accuracy and run time of the algorithm.

6 samples with different number of nodes (8-15 nodes) are analysed for this purpose. This section includes three parts. Each part studies the effect of variation of a single constrain from (3-1).

The Effect Of Variation The Number Of Maximum Links Per Small Cell:

Table 10. Effect of varying L on accuracy of heuristic 3

instance	Sam.	L	Optimal value (B/s)	Solution provided by heuristic 3 (MB/s)	Time (s)	Gap (%)
1	1	3	85196911.00	80500100.00	<0.5	5.51
2	1	2	80502995.00	80500100.00	<0.5	0.00
3	1	1	80502995.00	80500100.00	<0.5	0.00
4	2	3	266699693.00	254900000.0	<0.5	4.42
5	2	2	223959572.00	212200000.0	<0.5	5.25
6	2	1	132216878.00	132200000.0	<0.5	0.01
7	3	3	134388302.00	134350000.0	<0.5	0.03
8	3	2	134388302.00	134350000.0	<0.5	0.03
9	3	1	90000000.00	90000000.0	<0.5	0.00

instance	Sam.	L	Optimal value (B/s)	Solution provided by heuristic 3 (MB/s)	Time (s)	Gap (%)
10	4	3	263728081.00	263700000.0	<0.5	0.01
11	4	2	203728081.00	203700000.0	<0.5	0.01
12	4	1	123728081.00	123728000.0	<0.5	0.00
13	5	3	130510426.00	119600000.0	<0.5	8.36
14	5	2	130510426.00	89600000.0	<0.5	31.35
15	5	1	79609611.00	79600000.0	<0.5	0.01
16	6	3	136365909.00	136340000.00	<0.5	0.02
17	6	2	136365909.00	136340000.00	<0.5	0.02
18	6	1	80000000.00	80000000.00	<0.5	0.00

Table 9 represent the effect of variation of L constrain on the accuracy and run times of the algorithm. It can be noticed that run time is fixed among all trials (<0.5 seconds). Figure 18 summarizes the effect of varying the number of L on the accuracy of the algorithm.

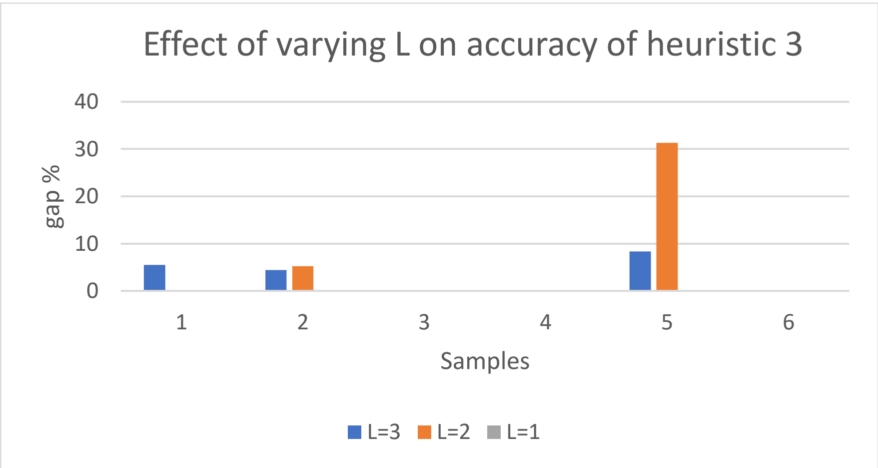

Figure 17. Effect of varying L on accuracy of heuristic approach 3

It is clear from Figure 18 that there is no particular unified effect for increasing or decreasing the number of maximum links. Inverse relation can be noted in instances 5 & 2. However, for instance 1, the accuracy gap is directly proportional with L

The Effect Of Varying The Number Maximum Flows

Table 11. effect of varying F on accuracy of heuristic approach 3

instance	Sam.	F	Optimal value (B/s)	Solution provided by heuristic 3 (MB/s)	Time (s)	Gap (%)
1	1	3	85196911.00	80500100.00	<0.5	5.51
2	1	2	85196911.00	80500100.00	<0.5	5.51
3	1	1	85196911.00	80500100.00	<0.5	5.51
4	2	3	266699693.00	254900000.00	<0.5	4.42
5	2	2	266699693.00	254900000.00	<0.5	4.42
6	2	1	254956999.00	254900000.00	<0.5	0.02
7	3	3	134388302.00	134350000.00	<0.5	0.03
8	3	2	134388302.00	134350000.00	<0.5	0.03
9	3	1	134388302.00	134350000.00	<0.5	0.03
10	4	3	263728081.00	263700000.00	<0.5	0.01
11	4	2	263728081.00	263700000.00	<0.5	0.01
12	4	1	203728081.00	203700000.00	<0.5	0.01
13	5	3	130510426.00	119600000.00	<0.5	8.36
14	5	2	130510426.00	119600000.00	<0.5	8.36
15	5	1	130510426.00	119600000.00	<0.5	8.36
16	6	3	136365909.00	136340000.00	<0.5	0.02
17	6	2	136365909.00	136340000.00	<0.5	0.02
18	6	1	136365909.00	136340000.00	<0.5	0.02

Table **10** represent the effect of varying F constrain on the accuracy and run times of the algorithm. It is very clear from table that there is no effect on the run time. Figure 19 summarizes the effect of varying F on heuristic 3.

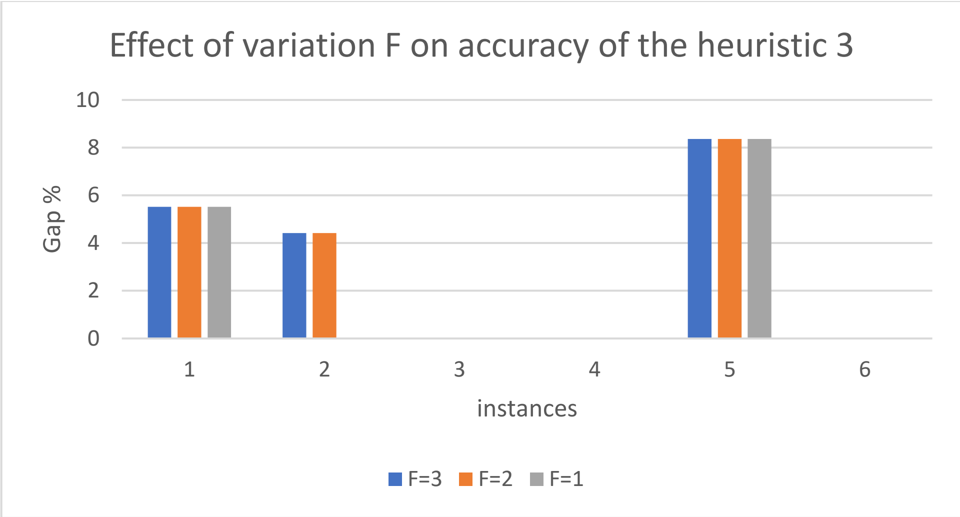

Figure 18. Effect of varying F on accuracy of heuristic approach 3

It can be noticed that changing F has very minor effect on the accuracy gap. The optimized value has very slight change when F is varied. Accordingly, no relationship between the accuracy and F can be concluded.

The Effect Of Varying Number Of Maximum Hops Per Small Cell:

Table 12. effect of varying H on accuracy of heuristic approach 3

instance	Sam.	H	Optimal value (B/s)	Solution provided by heuristic 3 (MB/s)	Time (s)	Gap (%)
1	1	3	85196911.00	80500100.00	<0.5	5.51
2	1	2	80502995.00	80500100.00	<0.5	0.00
3	1	1	80502995.00	80500100.00	<0.5	0.00
4	2	3	266699693.00	254900000.00	<0.5	4.42
5	2	2	266699693.00	254900000.00	<0.5	4.42
6	2	1	132216868.00	132200000.00	<0.5	0.01
7	3	3	134388302.00	134350000.00	<0.5	0.03

instance	Sam	H	Optimal value (B/s)	Solution provided by heuristic 3 (MB/s)	Time (s)	Gap (%)
8	3	2	134388302.00	134350000.00	<0.5	0.03
9	3	1	90000000.00	90000000.00	<0.5	0.00
10	4	3	263728081.00	263700000.00	<0.5	0.01
11	4	2	263728081.00	263700000.00	<0.5	0.01
12	4	1	123728081.00	123700000.00	<0.5	0.02
13	5	3	130510426.00	119600000.00	<0.5	8.36
14	5	2	116164617.00	116120000.00	<0.5	0.04
15	5	1	79609611.00	79600000.00	<0.5	0.01
16	6	3	136365909.00	136340000.00	<0.5	0.02
17	6	2	130000000.00	130000000.00	<0.5	0.00
18	6	1	80000000.00	80000000.00	<0.5	0.00

Table 11 represents the effect of varying H constrain on the accuracy and run time of the heuristic algorithm. Same as the previous case, the run time is fixed among all the trials (<0.5 s). Figure 20 summarizes the effect of varying H on the accuracy gap of algorithm 3. It can be noticed that there is no conclusive relation between H variation and the accuracy of the algorithm. In most cases, the accuracy is stable regardless of the value of H.

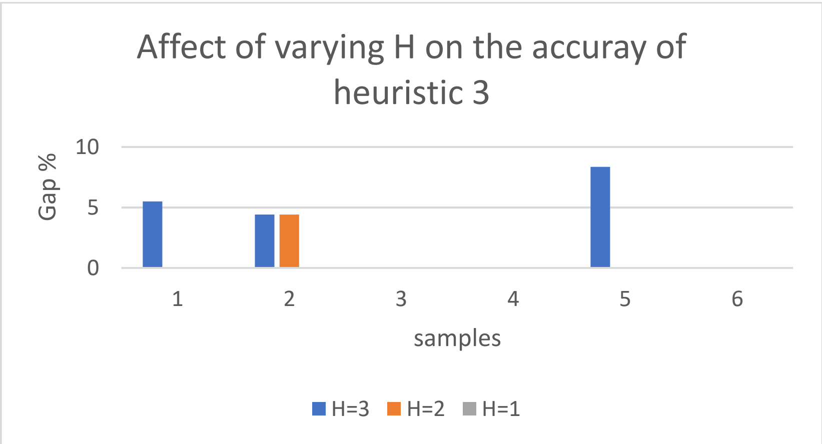

Figure 19. Effect of varying H on accuracy of heuristic approach

The Effect Of Varying The Number Of Edges On The Run Time Of Heuristic 3

In this part, we will study the effect of varying the number of edges between nodes on the run time of the heuristic approach. The number of nodes will be kept constant while number of edges will be varied. The aim of this section is to ensure that increasing the density of the network (by increasing the number of edges) does not affect the run time of the algorithm negatively.

In this part, 9 groups were analysed, and each group consisted of 3-4 instances. All instances within a group have the same number of nodes with different number of edges. Table 12 summarizes the results of this section.

Table 13. varying number of edges

Instance	No. of edges	heuristic approach 3 solution (B/s)	run time (s)
1	40.00	46650000.00	<0.5
2	54.00	240000000.00	<0.5
3	74.00	284240000.00	<0.5
4	88.00	280000000.00	<0.5
5	48.00	222740000.00	<0.5
6	56.00	160000000.00	<0.5
7	64.00	60000000.00	<0.5
8	72.00	192660000.00	<0.5
9	76.00	163790000.00	<0.5
10	100.00	375320000.00	<0.5
11	110.00	349520000.00	<0.5
12	142.00	534830000.00	<0.5
13	88.00	80000000.00	<0.5
14	110.00	303940000.00	<0.5
15	146.00	263070000.00	<0.5
16	180.00	666080000.00	<0.5
17	88.00	158550000.00	<0.5
18	106.00	323160000.00	<0.5
19	130.00	279640000.00	<0.5
20	172.00	609660000.00	<0.5

Instance	No. of edges	heuristic approach 3 solution (B/s)	run time (s)
21	142.00	224900000.00	<0.5
22	160.00	180040000.00	<0.5
23	190.00	262180000.00	<0.5
24	234.00	440151000.00	<0.5
25	200.00	278910000.00	<0.5
26	230.00	345280000.00	<0.5
27	264.00	459341000.00	<0.5
28	294.00	750340000.00	<0.5
29	190.00	254150000.00	<0.5
30	234.00	212100000.00	<0.5
31	250.00	403630000.00	<0.5
32	328.00	832150000.00	<0.5

As shown in table 12**,** the run time of the algorithm did not exceed 0.5 seconds for all instances among the 8 groups. This proves the efficiency of the algorithm regardless of the number of edges.

Chapter 5: Conclusion And Future Work

In conclusion, novel heuristic algorithms have been developed to maximize the total throughput of dense Heterogeneous networks with multi-hop backhauling. 2 algorithms have been developed to solve full demand model and 1 algorithm has been developed for partial demand model. The three algorithms were tested on small instances and provided accurate results with gaps of 4%, 3.5% and 5.5% for algorithms 1,2 and 3 respectively. The run time times of the three algorithms did not exceed 0.5 seconds which considered to be perfect run time. The sensitivity of the three algorithms with respect to L, F and H has been analyzed and no specific uniform pattern has been noticed with any of the variables. However, we ensured that the varying L, F and H does not affect the accuracy of the algorithms. Algorithms 1 and 2 have been tested on large-scaled networks with large number of SCs (up to 103 nodes) and provided accuracy gaps of 14.3 and 13.5, respectively. it is very remarkable that the run times for solving large-scaled networks (up to 103 nodes) of both algorithms did not exceed 0.5 seconds which considered to be perfect compared to the original model formulated by Almohamad et al. (2019) which takes an infinite time to solve a network with same capacity.

For the future, we recommend the following: -

- applying the algorithm 3 on large-scaled networks with > 100 nodes as the same has not been analyzed in this thesis.
- As the developed algorithms solve the networks with single UAV, it will be a good idea to improve the algorithms so we can solve larger networks with multiple UAVs
- Implementing the field of machine learning to the developed algorithms.
- Average performance: instead of considering the nodes with largest demands

only, we can try many combinations of different demands and consider the average throguhput among them.

REFERENCES

Aboagye, S. (2018). *Energy Efficiency Optimization in Millimeter Wave Backhaul Heterogeneous Networks* (Master). Memorial University of Newfoundland.

Aboagye, S., Ibrahim, A., & Ngatched, T. (2020). Frameworks for Energy Efficiency Maximization in HetNets With Millimeter Wave Backhaul Links. *IEEE Transactions On Green Communications And Networking*, *4*(1), 83-94. https://doi.org/10.1109/tgcn.2019.2949288

Ahuja, R., Magnanti, T., & Orlin, J. (1993). *Network flows*. Prentice Hall.

Almohamad, A., Hasna, M., Khattab, T., & Haouari, M. (2019). On Network Flow Maximization via Multihop Backhauling and UAVs: An Integer Programming Approach. *2019 IEEE 89Th Vehicular Technology Conference (VTC2019-Spring)*. https://doi.org/10.1109/vtcspring.2019.8746478

Almohamad, A., Hasna, M., Khattab, T., & Haouari, M. (2018). Maximizing Dense Network Flow through Wireless Multihop Backhauling using UAVs. *2018 International Conference On Information And Communication Technology Convergence (ICTC)*. https://doi.org/10.1109/ictc.2018.8539573

Arribas, E., Fernandez Anta, A., Kowalski, D., Mancuso, V., Mosteiro, M., Widmer, J., & Wong, P. (2020). Optimizing mmWave Wireless Backhaul Scheduling. *IEEE Transactions On Mobile Computing*, *19*(10), 2409-2428. https://doi.org/10.1109/tmc.2019.2924884

Bogale, T., & Le, L. (2016). Massive MIMO and mmWave for 5G Wireless HetNet: Potential Benefits and Challenges. *IEEE Vehicular Technology Magazine*, *11*(1), 64-75. https://doi.org/10.1109/mvt.2015.2496240

Gao, Z., Dai, L., Mi, D., Wang, Z., Imran, M., & Shakir, M. (2015). MmWave massive-MIMO-based wireless backhaul for the 5G ultra-dense network. *IEEE Wireless Communications*, *22*(5), 13-21. https://doi.org/10.1109/mwc.2015.7306533

Ghimire, J., & Rosenberg, C. (2015). Revisiting Scheduling in Heterogeneous Networks When the Backhaul Is Limited. *IEEE Journal On Selected Areas In Communications*, *33*(10), 2039-2051. https://doi.org/10.1109/jsac.2015.2435291

Gupta, R., & Kalyanasundaram, S. (2017). Resource allocation for self-backhauled networks with half-duplex small cells. *2017 IEEE International Conference On Communications Workshops (ICC Workshops)*. https://doi.org/10.1109/iccw.2017.7962657

Hu, R., & Yi Qian. (2014). An energy efficient and spectrum efficient wireless heterogeneous network framework for 5G systems. *IEEE Communications Magazine*, *52*(5), 94-101. https://doi.org/10.1109/mcom.2014.6815898

Islam, M., Sampath, A., Maharshi, A., Koymen, O., & Mandayam, N. (2014). Wireless backhaul node placement for small cell networks. *2014 48Th Annual Conference On Information Sciences And Systems (CISS)*. https://doi.org/10.1109/ciss.2014.6814156

Lagunas, E., Lei, L., Maleki, S., Chatzinotas, S., & Ottersten, B. (2017). Power allocation for in-band full-duplex self-backhauling. *2017 40Th International Conference On Telecommunications And Signal Processing (TSP)*. https://doi.org/10.1109/tsp.2017.8075953

Li, P., & Xu, J. (2018). Placement Optimization for UAV-Enabled Wireless Networks with Multi-Hop Backhauls. *Journal Of Communications And Information Networks*, *3*(4), 64-73. https://doi.org/10.1007/s41650-018-0040-3

Mcmenamy, J., Narbudowicz, A., Niotaki, K., & Macaluso, I. (2020). Hop-Constrained mmWave Backhaul: Maximising the Network Flow. *IEEE Wireless Communications Letters*, *9*(5), 596-600. https://doi.org/10.1109/lwc.2019.2961879

Niknam, S., Nasir, A., Mehrpouyan, H., & Natarajan, B. (2016). A Multiband OFDMA Heterogeneous Network for Millimeter Wave 5G Wireless Applications. *IEEE Access*, *4*, 5640-5648. https://doi.org/10.1109/access.2016.2604364

Niu, Y., Ding, W., Wu, H., Li, Y., Chen, X., Ai, B., & Zhong, Z. (2019). Relay-Assisted and QoS Aware Scheduling to Overcome Blockage in mmWave Backhaul Networks. *IEEE Transactions On Vehicular Technology*, *68*(2), 1733-1744. https://doi.org/10.1109/tvt.2018.2890308

Niu, Y., Li, Y., Jin, D., Su, L., & Vasilakos, A. (2015). A survey of millimeter wave communications (mmWave) for 5G: opportunities and challenges. *Wireless Networks*, *21*(8), 2657-2676. https://doi.org/10.1007/s11276-015-0942-z

Patil, S., & Bhavikatti, A. (2019). Heterogeneous network optimization using robust power-and-resource based algorithm. *International Journal Of Electrical And Computer Engineering (IJECE)*, *9*(5), 4226. https://doi.org/10.11591/ijece.v9i5.pp4226-4237

Rezaabad, A., Beyranvand, H., Salehi, J., & Maier, M. (2018). Ultra-Dense 5G Small Cell Deployment for Fiber and Wireless Backhaul-Aware Infrastructures. *IEEE Transactions On Vehicular Technology*, *67*(12), 12231-12243. https://doi.org/10.1109/tvt.2018.2875114

Web.stanford.edu. (2020). Retrieved 11 October 2020, from https://web.stanford.edu/class/archive/cs/cs106x/cs106x.1192/lectures/Lecture22/Lecture22.pdf.

Zhang, H., Fang, F., Cheng, J., Long, K., Wang, W., & Leung, V. (2018). Energy-Efficient Resource Allocation in NOMA Heterogeneous Networks. *IEEE Wireless Communications*, *25*(2), 48-53. https://doi.org/10.1109/mwc.2018.1700074

Zhang, H., Liu, H., Cheng, J., & Leung, V. (2018). Downlink Energy Efficiency of Power Allocation and Wireless Backhaul Bandwidth Allocation in Heterogeneous Small Cell Networks. *IEEE Transactions On Communications*, *66*(4), 1705-1716. https://doi.org/10.1109/tcomm.2017.2763623

Zhu, Y., Niu, Y., Li, J., Wu, D., Li, Y., & Jin, D. (2016). QoS-aware scheduling for small cell millimeter wave mesh backhaul. *2016 IEEE International Conference On Communications (ICC)*. https://doi.org/10.1109/icc.2016.7511065