An Introduction of Real-time Embedded Optimisation Programming for UAV Systems under Disaster Communication

Long D. Nguyen[1, *], Ayse Kortun[1] and Trung Q. Duong[1]

[1]Queen's University Belfast, Belfast BT7 1NN, UK

Abstract

For disaster communications, it is very challenging for the contemporary wireless technology and infrastructure to meet the demands for connectivity. Modern wireless networks should be developed to satisfy the increasing demand of quality-of-service (QoS) in mission-critical communications for disaster management which are currently faced with the challenges of limited spectrum, expensive resources, reliable and green communication. There is a tremendous need for optimisation techniques in the study and design of the key functionalities of wireless systems. Until now, almost current optimisation scenarios are often carried out on large timescales (e.g., minutes or hours) without strict time constraints for solving the problems. With the improvement of computational speed, efficient algorithms and advanced coding approaches, a framwork of real-time optimisation programming, which plays a major role in the trend of modern engineering such as mission-critical communications, is introduced for the first time in this paper in the context of natural disaster. In particular, we propose an introduction of embedded convex optimisation programming for unmanned ariel vehicles (UAVs) communications in disaster networks with the strict supervision of executive time in real-time scenarios.

1. Introduction

Major disasters usually result in damage to communication and power supply networks which are vital services in those early hours of rescue, thus making prompt restoration and network sustainability a formidable task [4, 18]. In addition, it may take long time to grasp the destructive situation when a disaster actually occurs. Thus, it is a huge challenge to accurately evaluate the situation of the entire regions in real time, which can consequently cause the loss of thousands of lives in the surrounding communities [13, 17].

To overcome this serious problem, emerging wireless technologies [6, 16, 27, 28, 31] based on Internet-of-thing (IoT) platform consisting of device-to-device (D2D) communications, wireless sensor networks (WSN) and the unmanned aerial vehicles (UAVs) enabled communication [12, 15] are used to communicate in disaster environments controlled by Ground Control Stations (GCSs) for supporting disaster management. With flexible configuration and the mobility nature, UAVs will be very efficient and inexpensive for deployment that is critical in the disaster-prone areas [14, 22]. UAVs can easily gather information, manipulate physical objects, or engage some equipment in remote or dangerous places as in the disaster events [9, 20, 32]. In order to provide adequate network coverage for particular areas affected by the disaster, there are many optimisation problems which should be considered and solved in real-time scenarios for exploiting and controlling the UAV systems. This will ensure the ultra-low network latencies transmission traffic so this enables the real-time applications in the disaster scenarios [13, 22]. These optimisation problems have often been associated with many efficient (linear and nonlinear) methods and algorithms such as filter design, resource allocation, and transmission strategies. These methods are chosen such that the network performance is optimised corresponding to the demand of network services via optimisation problems.

In many scenarios of communication system with the constantly changing environment, optimisation application is used to optimise the problems which need to

---

*Long D. Nguyen, Ayse Kortun and Trung Q. Duong are with Queen's University Belfast, Belfast BT7 1NN, UK. Email: long.nguyen@qub.ac.uk, akortun@gmail.com and trung.q.duong@qub.ac.uk

be solved in online applications, i.e., channel estimation, dynamic resource allocation, and video streaming. The strict real-time deadline is the most important requirement for online applications. In all current optimised scenarios, the processing time solving for the convex optimisation problem is achieved in minutes or hours even longer than that. In fact, the traditional convex optimisation methods have been considered computationally expensive which require plenty of time for evaluation. On the other hand, the modern optimisation problems are potentially formed in expensive and complex problems such as the combination of continuous and integer decision variables, multiple decision making for multi-objective, large-scale problems and unknown objectives information.

Fortunately, the rapidly increasing of powerful computers, efficient algorithms, and novel approaches have promised an enormous degradation of calculation time in optimisation algorithms for the embedded convex optimisation programming, which deliver for solving modern optimisation problems on the strict time deadlines of microsecond or millisecond timescales [19, 22]. In fact, some modest-sized optimisation problems can be handled by the on-line algorithms in the fast speed processing and a sufficient amount of computer memory size [19, 22]. Moreover, in the near future, efficient methods using parallel computing are applicable to real-time applications for solving large-scale problems. Another potential approach for real-time optimisation is to interplay the frontier of modern optimisation and machine learning techniques such as low-rank minimization, sparsity methods, decomposition techniques or statistically learning models.

2. Practical convex optimisation in real–time scenarios

Over many past years, convex optimisation theory has demonstrated its important role in signal processing and wireless communication systems. The variety of classes of convex optimisation programming has allowed the convex optimisation applications in a larger number of wireless communication problems. Together with the explosion of wireless communication, convex optimisation has become the most potential trend for the design, analysis and deployment of wireless infrastructures. Convex optimisation methods are used to provide a good heuristic suboptimal solution or bound to non-convex problems. Furthermore, there are many available optimisation software which can efficiently solve the convex optimisation problems. Meanwhile, there is a lack of efficient methods and software for solving the non-convex problem since there would be no guarantee that the optimal solution can be explicitly found in the non-convex problems.

Figure 1. How to choose a good (best) decision for practical optimisation application?

In contrast, convex optimisation methods often bring the insight of the optimal solution models which are relevant to the original problems of wireless communications.

Each of optimisation applications is the balance and trade-off of analysing and designing strategies by developers. However, there is a huge gap between the optimisation theory and optimisation implementation in their applications. In addition, it is still a lack of research to link the theory and implementation of optimisation. As shown Fig. 1, the decision of optimisation application in reality is a complicated decision making of the environment models, characteristics of the problem (variables, objectives), appropriate methods and the trade-offs for an expected solution.

Real-time convex optimisation: The role of processing time, which depends on how fast the optimisation problems are solved, is very crucial in signal processing of wireless communication. However, traditional convex optimisation methods are currently still expensive and implementing them can be extremely time-consuming. This calls for the development of novel strategies, which could be particularly beneficial when applied at times of mission-critical communications such as natural disasters. A real-time optimisation process has to respond to these questions:

- how to deal with the executive time of optimisation problems in real-time scenarios?

- how to face with the large-scale and/or high complexity of problems in on-line applications?

- how to embed advanced algorithms of real-time optimisation in the real systems?

In this context, the practical optimisation problems can be transformed into the mathematical optimisation models. Then these problems can be programmed in embedded systems with online solutions. However, the practical optimisation problems in wireless networks are frequently non-linear programming, large-scale problems with massive data, high-complex problems with multi-objective, high-dimensional stochastic problems, and hybrid resource allocation problems. An online solution should satisfy the demand of the strict deadline in the implementation. Fast optimisation decisions are used to calibrate the weights to improve the transmission strategy, optimise network performance or other objectives in wireless communications. Nevertheless, in the sense that "faster is better" [19], there are many challenges which will also generate opportunities for the deployment of real-time optimisation.

3. UAV-enabled networks under disaster communications

UAV has been emerging as a major trend in the next generation wireless networks [1, 11, 23]. With flexible configuration and high mobility nature, UAVs can easily gather information, manipulate physical objects, or engage some equipment in remote or dangerous places. With a wide variety of vehicle types and operational environments, UAVs can be easily deployed in ground, space, water, or underground [25]. As a result, there are many types of applications that UAVs can be exploited such as environmental remediation, navigation to gather data, military applications, transportation of goods, performing dangerous tasks, and so on [25]. In addition, UAVs can be utilized for governmental or non-governmental purposes in rescue operations such as public protection, disaster recovery. Some good examples are surveillance and reconnaissance, homeland security, environmental monitoring, agriculture, construction industry, and goods transportations. For a real-time application, the tracking results will be collected at the UAVs and reported to the central system within given strict time deadlines [24].

In natural disasters, keeping communication connectivity provides a lifeline. The lack of communications in remote areas and poor conditions for communications in developing countries can have detrimental effects. One of the most feasible technology, the online algorithms could be integrated within UAVs, which could be of great help in situations where networks are congested, and there is a lack of power supply. Therefore, the real-time optimisation in UAV communications to tackle time constraint will play a crucial role in disaster scenarios. In these cases, UAVs that are flying above the affected area could help first responders to assess the situation and deliver a coverage based cellular network as quickly as possible. To develop practical optimisation

in emergency situations, developers should focus on the ways of reducing the execution time and computational complexity of optimisation programming in the real-time scenarios.

3.1. The challenges of UAV-enabled communications

There are many issues to solve in order to provide effective, stable and reliable UAV networks. The efficient deployment and operation of UAV communications is very important yet challenging to meet the promising high capability and capacity [11]. The control of UAVs might depend heavily on sensors, communication devices, onboard processing power, and appropriate existing models. The modeling of UAVs has led to interest in areas such as nonlinear system and artificial intelligence [25]. The UAVs working on space often move in organized swarms in three dimensions with a rapid change of network trajectory. For example, a change in the relative positions of UAVs may leads to some UAVs losing their trajectory and power, and thus, they need to be brought down for redesigning and recharging. Furthermore, UAVs may malfunction or interrupt network, causing the network links to vanish and changing nodes' postions [5, 26, 29].

On the other hand, UAV devices typically have limited energy storage for flying operations. The energy constraints will be much tighter in the long-term operation of UAV networks. Otherwise, almost signals are transmitted under lower power and thus the outage probability is high because the links might be intermittent. More importantly, these nodes would dynamically work to frequently reorganizing the network. This means their routing protocols need to verify changes over time and use more energy to prolong the network stability. Therefore, the deployment and resource allocation should be considered for efficient utility in UAV-based applications.

A fully autonomous multi-UAV network will require a robust inter-UAV network to cooperate in keeping the network organized. The ability of self-organizing is essential in UAV networks to change the MAC and network layers, be tolerant to delays enable flexible and automatic control as well as employing efficient energy saving schemes. Therefore, the UAV systems must be more dynamic and wise. The machine learning context is a potential trend for UAV networks by lending efficient robust algorithms, low-complexity estimates, supporting self-organizing system. Finally, real-time optimisation for embedded UAV systems is a promising research direction for the modern wireless communications.

3.2. Characteristics of UAV communications

In this section, we consider the general characteristics of UAV-based wireless communication. The general communication model consists of cellular base stations (BSs), UAV-drones, and ground terminals (GTs).

3D location. Within the completion time T and trajectory design, UAVs' location is denoted as as $q(t) = [x(t), y(t), H(t)]^T \in \mathbb{R}^3$ at time t. We assume that the initial and final time of UAVs' location follow by $t^{(0)} = 0$, $t^{(N)} = T$. By introducing the elemental time-slot length δ_t, the horizon time T is divided into M time-slots, e.g., $T = M\delta_t$. The elemental time-slot length is chosen so that the location of UAVs and ground terminals (GTs) are approximately assumed constants within each slot. Otherwise, 3D (three-dimensional) location of the UAV in time-slot m can be rewritten as

$$q[m] = [x[m], y[m], H[m]]^T, \ m = 1, ..., M \qquad (1)$$

Channel model. Due to the higher chance of line-of-sight (LoS) propagation via 3D location, the air-to-ground (ATG) channel differs from the terrestrial channel [7]. In these scenarios, the effect of the environment on LoS occurrence becomes critical. However, the effects of propagation blockage [2], e.g., building blockage, still exist for the complete channel models. As a result, large-scale Rayleigh and free-space fading models are appropriate for the ATG channels.

For an arbitrary elemental time-slot, the distance between the UAV located at (x, y, H) and the kth GT is denoted as $R_k = \sqrt{d_k^2 + H^2}$ where $d_k = \sqrt{(x - x_k)^2 + (y - y_k)^2}$ is the horizontal distance between the UAV and the kth GT and x_k, y_k is the location of the kth GT. The simple distance path loss between the UAV and the kth GT is denoted as [21]

$$L(R_k) = 10 \log(\frac{4\pi f_c R_k}{c})^\alpha \qquad (2)$$

where f_c is carrier frequency (Hz), c is the speed of light (m/s), $\alpha \geq 2$ is the path loss exponent.

On the other hand, the probability of LoS is given by [3, 7]

$$P_{LoS} = \frac{1}{1 + a \exp(-b(\arctan(\frac{H}{d_k}) - a))} \qquad (3)$$

where a, b are constants depending on environment. Thus, one has $P_{NLoS} = 1 - P_{LoS}$.

Then the total path loss expression from UAV to kth GT is as

$$PL(R_k) \quad = \quad L(R_k) + \eta_{LoS} P_{LoS} + \eta_{NLoS} P_{NLoS} \qquad (4)$$

where η_{LoS} and η_{NLoS} are average additional losses for LoS and NLoS, respectively.

Free-space model: We assume LoS-dominated in free-space path loss model ($\alpha = 2$) [30]. For time-slot t, the channel gain in the free-space model is defined as

$$g_k(t) = \beta_0 R_k^{-2}(t) \qquad (5)$$

where β_0 is the channel power gain at reference distance d_0, $R_k(t) = \sqrt{d_k^2(t) + H^2(t)}$ is the distance between the UAV and the kth GT at the time-slot t.

For a constant transmit power (P_U) at the UAV, the received power at the kth GT can be addressed as

$$P_k(t) = P_U g_k(t) \triangleq \frac{\beta_0 P_U}{(x(t) - x_k(t))^2 + (y(t) - y_k(t))^2 + H^2(t)} \qquad (6)$$

Thus, the total received power at the kth GT in the period of time (T) is given as

$$E_k = \int_0^T P_k(t)dt \qquad (7)$$

Routing and trajectory designs. We consider three fundamentals of the trajectory of UAVs including trajectory location $\{q[m]\}_{m=1}^M$, speed $\{v[m]\}_{m=1}^M$ and acceleration $\{a_{cc}[m]\}_{m=1}^M$. For the fixed altitude of UAV at H, the trajectory location of UAVs in the mth time-slot can simply as the horizontal location $q[m] = [x[m], y[m]]^T$, $m = 1, ..., M$ where M is the final time-slot at the end of trajectory.

The trajectory of UAV in a typical timeslot is defined as

$$q[m + 1] = q[m] + v[m]\delta_t + \frac{1}{2} a_{cc}[m]\delta_t^2, \ m = 0, 1, ..., M. \qquad (8)$$

$$v[m + 1] = v[m] + a_{cc}[m]\delta_t, \ m = 0, 1, ..., M. \qquad (9)$$

For a constant of velocity ($a_{cc}[m] = 0$) and allowable maximum velocity of the UAV (V_{max}), the trajectory constraints of the UAV are as

$$\|q[m + 1] - q[m]\| \leq V_{max}\delta_t, \ m = 0, 1, ..., M \qquad (10)$$

where $q[0] = [x[0], y[0]]^T$ is the initial horizontal location of the UAV.

3.3. The practical embedded optimisation programming for UAV systems

In [19], the embedded convex optimisation programming is able to solve many optimisation problems at the level of fast time scales. In fact, in real-time scenarios, the solving of optimisation problem not only has to solve the problem with an optimal feasible point but also mention the solving time of the problem with a strict time limit (deadline time constraint). A diagram
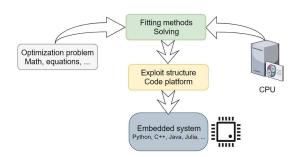
Figure 2. The diagram of embedded optimisation system in UAV–enabled communication.

of embedded convex optimisation system is proposed in Fig. 2. Efficient methods are used for solving the problem which is exploited on a type of computer (CPU, GPU, FPGA) applying the novel approaches for real-time optimisation programming. Finally, a custom code platform is produced by automatic code generation by an embedded software.

For an instance, Fig. 3 presents a general model of UAV communication in where multiple embedded UAV kits operate under a ground station for exchanging the network information via backhaul connections, e.g., channel state information, energy harvesting and auto-code generation of embedded convex optimisation. The ground station can be represented as a central computing for controlling a group of UAV-drones communication networks. The UAV-drone can apply real-time optimisation applications on the testing stage by learning-based optimisation algorithms on training stage from the ground station.
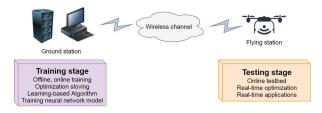

Figure 3. A practical model of UAV communication system with a ground station and a flying station.

3.4. Some applications for embedded UAV systems

To give the reader a better appreciation for the performance of the embedded optimisation, some basic evaluations of the UAV network with the embedded optimisation programming are implemented in this work by using CVXPY 1.0.6 package [8] in Python and CVX 2.1 [10] in Matlab. The computational platform with the Laptop using Intel Core(TM) i7, CPU @2.80GHz and 16GB memory is used for performing.

EXAMPLE 1: (optimal network coverage)
In the first example, the location of UAV-drones is to

optimise for supporting as much as possible number of GTs. An example of the considered model is provided in Fig. 4. The system consists a UAV with the fixed altitude at H and K users (UEs) which are randomly distributed in a given coverage network. Firstly, we assume the 3D
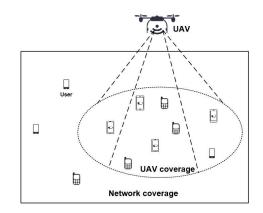

Figure 4. A network coverage of an embedded UAV–drone.

location of the UAV is as (x, y, H). The location of GTs is as (x_k, y_k) for $k = 1, ..., K$. We also define the ratio of altitude and coverage network of the UAV is as

$$\phi = H/r_{cov} \qquad (11)$$

where r_{cov} is the radius of network coverage of UAV. In this work, we choose the value of ϕ at 1 (e.g. unbar scenarios) which means $H = r_{cov}$.

Hence, the kth GT will be served by the UAV if

$$d_k^2 = (x - x_k)^2 + (y - y_k)^2 \leq r_{cov}^2 = H^2$$

Our target is to maximize the number of GTs that the UAV can support at the same time. The considered problem is given by

$$\max_{x,y,u_k} \sum_{k=1}^{K} u_k \qquad (12a)$$

$$\text{s.t. } d_k^2 \leq r_{cov}^2 + \lambda(1 - u_k), \ k = 1, ..., K \qquad (12b)$$

$$x_{min} \leq x \leq x_{max} \qquad (12c)$$

$$y_{min} \leq y \leq y_{max} \qquad (12d)$$

$$u_k \in \{0, 1\}, \ k = 1, ..., K \qquad (12e)$$

where λ is chosen as a specific value such that it is larger than the maximum network area that we consider. (x_{min}, x_{max}) and (y_{min}, y_{max}) are the lower and upper bounds of the horizontal and vertical range of UAV, respectively. Note that the problem in (18) is mixed-integer (binary) quadratic programming which is non-convex program. Fortunately, a Python-embedded optimisation programming (CVXPY) can solve the problem (18) effectively. For a simulation, we assume the coverage is a square area with the length of side of 300m. (x_{min}, x_{max}) and (y_{min}, y_{max})

and H are set as $(-300, 300)$, $(-300, 300)$ and 100 in meter, respectively. There are $K = \{10, 20, 30\}$ GTs in the considered network. For this example, the problem (12) is transformed into a mixed-integer programming with $10, 20, 30$ binary variables and $15, 25, 35$ inequality constraints, respectively. Then, the Python code generated by CVXPY solves this problem in approximately $69, 362, 703$ ms corresponding to $10, 20, 30$ GTs, respectively.

EXAMPLE 2: (tracking and routing plan)
With natural mobility and flexible configuration, UAV is developed for the tracking targets or routing plan. In this work, the trajectory of a UAV is designed in the data collection for wireless sensor network. An example of considered model is provided in Fig. 5 in where a number of sensors is distributed in a network area.
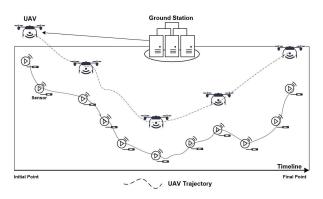

Figure 5. A model of routing UAV for data collection from sensor networks.

We assume the UAV operates with the constant of altitude as H. There are total $K + 1$ of time slot corresponding to K sensor nodes which need to collect their data and an initial time slot. At the initial point ($k = 0$), we assume the horizontal location of the UAV at $q[0] = [0, 0]$ and kth timeslot is as

$$q[k] = [x[k], y[k]]^T \tag{13}$$

The Euclidean distance between the UAV and sensor k at the time slot k is as

$$d_k[k] = \sqrt{(x[k] - x_k)^2 + (y[k] - y_k)^2} \tag{14}$$

where (x_k, y_k) represents the location of kth sensor node.

Each of sensors should forward its data with size (D_k) and transmission rate (C_k) to the UAV for collecting. Thus, the flying time of UAV over sensor node k (T_k) for completing the data collection should be followed the inequality

$$T_k \geq D_k / C_k , \quad k = 1, ..., K. \tag{15}$$

For convenience, the coverage of UAV can be its altitude for the urban scenarios. UAV can only collect

the data from sensor nodes when the sensor node belongs in its coverage

$$d_k[k] \leq r_{cov} = H , \quad k = 1, ..., K. \tag{16}$$

Next, the following constraint of flying time of UAV between two adjacent time slots is as

$$\delta_k \geq \frac{\|q[k+1] - q[k]\|}{V} , \quad k = 0, 1, ..., K - 1 \tag{17}$$

where V is the constant flight speed of the UAV.

Our target is to minimize the total flying time that the UAV need to spent for collecting the data from K sensor nodes. The considered problem is given by

$$\min_{q, \delta} \sum_{n=1}^{K} \delta_k + T_k \tag{18a}$$

$$\text{s.t. } (15), (16), (17) \tag{18b}$$

$$x_{min} \leq x[k] \leq x_{max} , \quad k = 1, ..., K \tag{18c}$$

$$y_{min} \leq y[k] \leq y_{max} , \quad k = 1, ..., K \tag{18d}$$

For measurement simulation, the number of sensor nodes is horizontally distributed in a rectangle network area with the width and height of 5000m and 200m. The altitude of the UAV is set as $H = 50$m. The flight speed of the UAV is constant at all time with $V = 10$m/s. The size data and transmission data rate at all sensor nodes are the same at $D = 400$bits and $C = 200$bits/s. There are $K = [20, 60, 100]$ sensors in the considered network. For 100 sensor nodes, the problem (18) is transformed into a QP programming with 300 variables and 700 inequality constraints. The Python code generated by CVXPY solves the problem (18) in approximately 600 milliseconds while CVX tool solves the problem in 6 seconds as shown in Table 1.

Table 1. The average executive time results in different tools for solving the problem (18).

# of sensors	Average flying time	CVXPY	CVX
20	530 s	138 ms	2.19s
60	625 s	397 ms	4.04s
100	740 s	638 ms	6.2s

EXAMPLE 3: (relay-assisted network)
In this example, a UAV is used as a relay node to serve multiple users who cannot be directly served from the BS by block fading such as building blockage. Due to the block fading, there is no direct link between the BS and GTs. The location of UAV-drones is to optimise for supporting multiple users. An example of considered model is provided in Fig. 6. Utilizing the high chance LOS propagation of the UAV deployment, we consider that the link between the BS and the UAV and the link
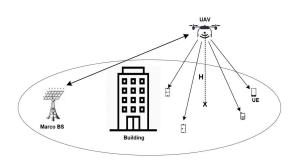
Figure 6. A relay network of an embedded UAV–drone.

between the UAV and the GTs are dominated by LOS links.

For presenting the mathematical formulation of the problem, we assume the locations of the BS and the UAV are (x_{BS}, y_{BS}, H_{BS}) and (x_U, y_U, H_U), respectively. Then the distance between the BS and the UAV is expressed as

$$R_{BU} = \sqrt{d_{BU}^2 + (H_{BS} - H_U)^2} \qquad (19)$$

where $d_{BU} = \sqrt{(x_U - x_{BS})^2 + (y_U - y_{BS})^2}$, H_{BS} and H_U are the antenna altitude of the BS and the UAV. We assume the antenna altitude of the UAV is also the altitude of the UAV. Thus, the free-space path loss channel can be used for the BS and the UAV link as follow: $g_{BU} = \beta_0 R_{BU}^{-2}$.

And the free-space path loss channel for the UAV and kth GT link is $g_{Uk} = \beta_0 R_{Uk}^{-2}$, where $R_{Uk} = \sqrt{d_{Uk}^2 + H_U^2}$ and $d_{Uk} = \sqrt{(x_U - x_k)^2 + (y_U - y_k)^2}$, (x_k, y_k) is the location of the kth GT.

In this work, our target is to optimise the position of the UAV to maximize the total channel gain of the considered system including the BS-UAV link and UAV-GTs links. Hence, the considered problem is as follow:

$$\max_{x_U, y_U, H_U} \sum_{k=1}^{K} g_{Uk} + g_{BU} \qquad (20a)$$

$$\text{s.t. } x_{min} \leq x_U \leq x_{max} \qquad (20b)$$

$$y_{min} \leq y_U \leq y_{max} \qquad (20c)$$

$$H_{min} \leq H_U \leq H_{max} \qquad (20d)$$

However, the problem (20) is non-convex which provides many challenges to solve. By relaxing the objective function of the problem (20) without loss of generality, the problem (20) is rewritten as the convex problem

$$\min_{x_U, y_U, H_U} \sum_{k=1}^{K} R_{Uk}^2 + R_{BU}^2 \qquad (21a)$$

$$\text{s.t. } (20b), (20c), (20d) \qquad (21b)$$

For simulation, we assume a coverage network with circle radius of 500m. The location of MBS is $(-300, 0, 30)$ while the location of GTs are randomly distributed in the coverage network. (x_{min}, x_{max}), (y_{min}, y_{max}) and (H_{min}, H_{max}) are set as $(-500, 500)$, $(-500, 500)$ and $(47, 122)$ in meter, respectively. There are $K = \{10, 20, 30\}$ UEs in the considered network which are located at the opposite of BS. For implementation, the problem (21), which is transformed into a QP programming, is solved by generating Python code in our platform in approximately $\{131, 325, 361\}$ms using CVXPY and $\{1.54, 1.92, 2.35\}$s using CVX, respectively.

4. Conclusions

This paper has introduced the practical optimisation programming related to real-time applications and proposed the basic theory of the UAV systems based on the development of optimisation field for the disaster communications in real-time scenarios. This work opens a novel trend of optimisation context for the modern wireless communication systems. As a result, there are many issues and future works to study and necessarily call the novel solutions in the real-time optimisation for critical mission wireless networks.

Acknowledgement

References

[1] Agogino, A., HolmesParker, C. and Tumer, K. [2012], Evolving large scale UAV communication system, *in* 'Proceedings of the 14th Annual Conf. Genetic and Evolutionary Computation', ACM, pp. 1023–1030.

[2] Al-Hourani, A., Kandeepan, S. and Jamalipour, A. [2014], Modeling air-to-ground path loss for low altitude platforms in urban environments, *in* '2014 IEEE Global Commun. Conf. (GLOBECOM)', pp. 2898–2904.

[3] Al-Hourani, A., Kandeepan, S. and Lardner, S. [2014], 'Optimal LAP altitude for maximum coverage', *IEEE Wireless Commun. Lett.* **3**(6), 569–572.

[4] Ali, K., Nguyen, H. X., Vien, Q.-T. and Shah, P. [2015], Disaster management communication networks: Challenges and architecture design, *in* '2015 IEEE Int. Conf. Pervasive Computing and Commun. Workshops (PerCom Workshops)', IEEE, pp. 537–542.

[5] Betts, J. T. [1998], 'Survey of numerical methods for trajectory optimization', *Journal of guidance, control, and dynamics* **21**(2), 193–207.

[6] Bianco, A., Giraudo, L. and Hay, D. [2010], Optimal resource allocation for disaster recovery, *in* '2010 IEEE Global Telecommunications Conf. 2010', pp. 1–5.

[7] Bor-Yaliniz, R. I., El-Keyi, A. and Yanikomeroglu, H. [2016], Efficient 3-D placement of an aerial base station in next generation cellular networks, *in* '2016 IEEE Int. Conf. Commun. (ICC)', pp. 1–5.

[8] Diamond, S. and Boyd, S. [2016], 'CVXPY: A Python-embedded modeling language for convex optimization', *J. Machine Learning Research* **17**(83), 1–5.

[9] Erdelj, M., Natalizio, E., Chowdhury, K. R. and Akyildiz, I. F. [2017], 'Help from the sky: Leveraging UAVs for disaster management', *IEEE Pervasive Computing* **16**(1), 24–32.

[10] Grant, M. and Boyd, S. [2014], 'CVX: MATLAB software for disciplined convex programming, version 2.1', `http://cvxr.com/cvx`.

[11] Gupta, L., Jain, R. and Vaszkun, G. [2016], 'Survey of important issues in UAV communication networks', *IEEE Commun. Surveys Tuts.* **18**(2), 1123–1152.

[12] Hayajneh, A. M., Zaidi, S. A. R., McLernon, D. C., Renzo, M. D. and Ghogho, M. [2018], 'Performance analysis of UAV enabled disaster recovery networks: A stochastic geometric framework based on cluster processes', *IEEE Access* **6**, 26215–26230.

[13] Lei, S., Wang, J., Chen, C. and Hou, Y. [2018], 'Mobile emergency generator pre-positioning and real-time allocation for resilient response to natural disasters', *IEEE Trans. Smart Grid* **9**(3), 2030–2041.

[14] Liu, X. and Ansari, N. [2018], 'Resource allocation in UAV-assisted M2M communications for disaster rescue', *IEEE Wireless Commun. Lett.* pp. 1–1.

[15] Liu, X., Li, Z., Zhao, N., Meng, W., Gui, G., Chen, Y. and Adachi, F. [2018], 'Transceiver design and multi-hop D2D for UAV IoT coverage in disasters', *IEEE Internet Things J.* pp. 1–1.

[16] Lorincz, K. and et al. [2004], 'Sensor networks for emergency response: challenges and opportunities', *IEEE Pervasive Computing* **3**(4), 16–23.

[17] Manoj, B. S. and Baker, A. H. [2007], 'Communication challenges in emergency response', *Commun. ACM* **50**(3), 51–53.

[18] Mase, K. [2011], 'How to deliver your message from/to a disaster area', *IEEE Commun. Mag.* **49**(1), 52–57.

[19] Mattingley, J. and Boyd, S. [2010], 'Real-time convex optimization in signal processing', *IEEE Signal Process. Mag.* **27**(3), 50–61.

[20] Merwaday, A., Tuncer, A., Kumbhar, A. and Guvenc, I. [2016], 'Improved throughput coverage in natural disasters: Unmanned aerial base stations for public-safety communications', *IEEE Veh. Technol. Mag.* **11**(4), 53–60.

[21] Mozaffari, M., Saad, W., Bennis, M. and Debbah, M. [2016], 'Efficient deployment of multiple unmanned aerial vehicles for optimal wireless coverage', *IEEE Commun. Lett.* **20**(8), 1647–1650.

[22] Nguyen, M., Nguyen, L. D., Duong, T. Q. and Tuan, H. D. [2018], 'Real-time optimal resource allocation for embedded UAV communication systems', *IEEE Wireless Commun. Lett.* pp. 1–1.

[23] Olsson, P., KvarnstrÃűm, J., Doherty, P., Burdakov, O. and Holmberg, K. [2010], Generating uav communication networks for monitoring and surveillance, *in* '2010 11th Int. Conf. Control Auto. Robotics Vision (ICARCV)', pp. 1070–1077.

[24] Romer, K. and Mattern, F. [2004], 'The design space of wireless sensor networks', *IEEE Wireless Commun.* **11**(6), 54–61.

[25] Schoenwald, D. A. [2000], 'AUVs: In space, air, water, and on the ground', *IEEE Control Systems* **20**(6), 15–18.

[26] Shima, T. and Rasmussen, S. [2009], *UAV cooperative decision and control: challenges and practical approaches*, SIAM.

[27] Vo, N., Duong, T. Q. and Guizani, M. [2015], Quality of sustainability optimization design for mobile Ad Hoc networks in disaster areas, *in* '2015 IEEE Global Commun. Conf. (GLOBECOM)', pp. 1–6.

[28] Vo, N.-S., Masaracchia, A., Nguyen, L. D. and Huynh, B.-C. [2018], 'Natural disaster and environmental monitoring system for smart cities: Design and installation insights', *EAI Endorsed Trans. Industrial Networks and Intelligent Systems* **5**(16).

[29] Wu, Q., Zeng, Y. and Zhang, R. [2018], 'Joint trajectory and communication design for multi-UAV enabled wireless networks', *IEEE Trans. Wireless Commun.* .

[30] Xu, J., Zeng, Y. and Zhang, R. [2018], 'UAV-enabled wireless power transfer: Trajectory design and energy optimization', *IEEE Trans. Wireless Commun.* pp. 1–1.

[31] Yuan, W., Wang, J., Qiu, F., Chen, C., Kang, C. and Zeng, B. [2016], 'Robust optimization-based resilient distribution network planning against natural disasters', *IEEE Trans. Smart Grid* **7**(6), 2817–2826.

[32] Zhang, S. and Liu, J. [2018], 'Analysis and optimization of multiple unmanned aerial vehicle-assisted communications in post-disaster areas', *IEEE Trans. Veh. Technol.* pp. 1–1.

8