

---

## Remise 2

POLYTECHNIQUE  
MONTRÉAL

LE GÉNIE  
EN PREMIÈRE CLASSE



---

# Projet de session sur la modélisation du placement des UAV dans les "Flying Networks"

MTH8408 - Méthodes d'optimisation et contrôle optimal

Présenté à :

**MIGOT, Tangi**

Hiver 2023

Département de Mathématiques et Génie Industriel  
Polytechnique Montréal

Dernière mise à jour: 1<sup>er</sup> mai 2023

---

Olivier BÉLANGER  
Bouh ABDILLAHI

1954451  
1940646

---

## Table des matières

<b>1</b>	<b>Description de la problématique</b>	<b>1</b>
<b>2</b>	<b>Modèle ADNLP</b>	<b>2</b>
2.1	Description du modèle . . . . .	2
2.2	Résultats . . . . .	3
<b>3</b>	<b>Vue d'ensemble et vision générale du but du projet</b>	<b>4</b>
<b>4</b>	<b>Implémentation de l'algorithme GPQM de placement de FGW</b>	<b>6</b>
4.1	Description de l'algorithme . . . . .	6
4.2	Résultats . . . . .	6
<b>5</b>	<b>Perspective personnelle sur le projet</b>	<b>7</b>
<b>6</b>	<b>Lien vers les scripts de programmation</b>	<b>7</b>
	<b>Références</b>	<b>9</b>

## Liste des tableaux

# 1 Description de la problématique

L'article étudié (Coelho, Campos, & Ricardo, 2022) se penche sur la conception des réseaux aériens composés de drones servant à acheminer une connexion Internet à des usagers locaux, par exemple des équipes de pompiers dans le cas d'un feu de forêt. La principale problématique étudiée par cet article est la conception du réseau de *backhaul* afin d'assurer une qualité de service aux usagers. Plus précisément, les auteurs cherchent à trouver le placement optimal du *gateway* (le drone 0, noté UAV<sub>0</sub>, en plus d'optimiser la taille des files d'attente des drones pour maximiser le débit global.

Sur le plan de la conception, les auteurs cherchent précisément à minimiser la capacité globale utilisée afin de minimiser les ressources de télécommunications dans le réseau (et donc le coût total). Cette optimisation est effectuée par rapport à trois variables, soit la position du UAV<sub>0</sub>,  $(x_0, y_0, z_0)$ , la puissance de chacun des points d'accès,  $P_T$ , et la taille de chacune des files d'attente,  $Q_i$ . Le tout se fait pour chaque intervalle de temps  $t_k$ . La liste des constantes utilisées est présentée en Annexe. Formellement :

$$\min_{(x_0, y_0, z_0), P_T, Q_i} C(t_k) = \sum_{i=1}^{N-1} C_{0,i}(t_k) \quad (1)$$

$$\text{s.t. } 0 \leq P_T \leq P_T^{MAX} \quad (2)$$

$$C(t_k) \leq C^{MAX} \quad (3)$$

$$0 < T_i(t_k) \leq C_{0,i}(t_k), \quad i \in \{1, \dots, N-1\} \quad (4)$$

$$Q_i(t_k) \geq 0, \quad i \in \{1, \dots, N-1\} \quad (5)$$

$$D_i(t_k) < H, \quad i \in \{1, \dots, N-1\} \quad (6)$$

$$0 \leq x_i \leq x^{MAX}, \quad i \in \{1, \dots, N-1\} \quad (7)$$

$$0 \leq y_i \leq y^{MAX}, \quad i \in \{1, \dots, N-1\} \quad (8)$$

$$0 \leq z_i \leq z^{MAX}, \quad i \in \{1, \dots, N-1\} \quad (9)$$

$$(0, i), (i, 0) \in E(t_k), \quad i \in \{1, \dots, N-1\} \quad (10)$$

$$(x_0 - x_i)^2 + (y_0 - y_i)^2 + (z_0 - z_i)^2 > M^2, \quad i \in \{1, \dots, N-1\} \quad (11)$$

L'équation 1 est la fonction objectif. Les contraintes sont les suivantes :

- 2 : La puissance sur chaque lien est positive et inférieure à une valeur maximale.
- 3 : La capacité globale est inférieure à la valeur maximale allouée.
- 4 : La capacité allouée sur un lien  $i$  entre le drone  $i$  et le *gateway* est supérieure ou égale au flot sur ce lien.
- 5 : La taille de la file d'attente d'un drone  $i$  est non négative.
- 6 : Le délai moyen généré par le drone  $i$  est sous un seuil  $H$ .
- 7, 8, 9 : La position du drone  $i$  est dans l'espace 3D autorisé.
- 10 : Il est toujours possible d'établir une connexion entre le *gateway* et le drone  $i$ .
- 11 : La position du *gateway* est toujours au maximum à  $M$  mètres de distance du drone  $i$ .

## 2 Modèle ADNLP

### 2.1 Description du modèle

Dans cette section, nous présentons le modèle ADNLP implémenté. Le code UAV\_projet\_ADNLP.ipynb est un notebook Jupyter dans lequel le modèle ADNLP a été implémenté. Ce modèle nécessite d'abord la définition de trois fonctions. D'abord, l'objectif étant de minimiser la capacité globale, il est nécessaire de la définir. Selon l'article étudié, la capacité se calcule à l'aide du théorème de Shannon-Hartley. La capacité du lien entre le UAV 0 et le UAV  $i$  est donnée par 12.

$$C_{0,i} = B \log_2(1 + SNR_i) \quad (12)$$

On y remarque la nécessité de définir le rapport signal sur bruit. Celui-ci s'obtient à l'aide du modèle de propagation de Friis, comme présenté à l'équation 13.

$$SNR_i = P_T - 20 \log_{10}(d_i) - 20 \log_{10}(f) - 20 \log_{10}\left(\frac{4\pi}{c}\right) - P_N \quad (13)$$

$f$  et  $P_N$  sont donnés,  $c$  est la vitesse de la lumière dans le vide et

$$d_i = \sqrt{(x_i - x_0)^2 + (y_i - y_0)^2 + (z_i - z_0)^2}.$$

Il est important de noter que nous planifions initialement définir un modèle JuMP, mais que ce dernier ne supportait pas les définitions des fonctions mentionnées ci-dessus. Nous avons donc réussi à définir le modèle ADNLP. Pour ce faire, nous avons d'abord défini des vecteurs  $x0\_arr$ ,  $lvar$  et  $uvar$  définissant respectivement le point de départ, les bornes minimales et les bornes maximales pour les variables. Celles-ci sont dans l'ordre suivant :  $P_T$ ,  $Q[1]$ ,  $Q[2]$ ,  $Q[3]$ ,  $x_0$ ,  $y_0$ ,  $z_0$ . Ensuite, nous avons défini les bornes minimale et maximale pour la contrainte de capacité.

Nous avons ensuite défini une fonction *objective\_function* qui est la fonction objectif. Celle-ci a comme objectif de minimiser la capacité globale, et retourne ladite capacité en utilisant le SNR. La fonction de contrainte, *constr*, sert quant à elle à s'assurer que la capacité globale est située entre 0 et  $C_{MAX}$ . Nous créons finalement un modèle ADNLP avec les éléments mentionnés dans cette section et le résolvons à l'aide de *ipop*.

Afin de tester notre modèle et d'éventuellement obtenir les mêmes résultats que dans l'article, voici les constantes utilisées :

- $P_N = -85$  dB
- $f = 5250$  MHz
- $B = 160$  MHz
- $c = 3 \cdot 10^8$  m/s

Voici la position des 3 *Flying Access Points*(FAP) :

$$(x_1, y_1, z_1) = (50, 75, 10)$$

$$(x_2, y_2, z_2) = (75, 25, 10)$$

$$(x_3, y_3, z_3) = (25, 25, 10)$$

## 2.2 Résultats

Selon l'article, la position attendue du UAV<sub>0</sub> est  $(x_0, y_0, z_0) = (46.4, 12.3, 10)$ , la puissance transmise  $P_T$  est de 0dB et la taille des files d'attente est, dans l'ordre de 1, 111 et 133.

En résolvant notre modèle avec ipopt, nous obtenons bel et bien une puissance de transmission de 0dB comme attendu. Toutefois, nous obtenons une position du UAV<sub>0</sub> de  $(x_0, y_0, z_0) = (50, 0, 20)$  et des tailles de file d'attente de 100 000 pour les trois.

Nous pensons que cela est dû au manque d'information dans l'article quant à certaines contraintes (ex : sur le délai), ou bien sur la valeur des bornes pour d'autres contraintes. Nous sommes d'avis qu'une définition exhaustive des contraintes, ainsi qu'une spécification des bornes, pourrait nous aider à converger aux mêmes valeurs que les auteurs. Il aurait aussi été souhaitable de connaître le point de départ qu'ils ont utilisé, car étant donné qu'on ne peut garantir un minimum global, c'est possible que plusieurs minimums locaux existent, dont celui que nous avons trouvé. Notons finalement que la capacité globale, qui est la valeur de la fonction objectif, est de tout près de  $10^9$ , qui est la valeur que nous avons sélectionné pour  $C_{MAX}$ .

### 3 Vue d'ensemble et vision générale du but du projet

Les auteurs ont fourni 2 manières de calculer la position optimale des UAVs (position minimisant la capacité du lien de communication entre UAV et FGW) : la première méthode se base sur le calcul d'un SNR avec la formule exacte du SNR dérivée du modèle de propagation de Friis, la seconde se basant sur le calcul du SNR en se basant sur l'index MCS des UAVs, lui-même fonction du trafic demandé par l'UAV. Cette seconde manière de calculer le SNR avec les index MCS est spécifiée en détail par la norme 802.11ac, les auteurs de l'article fournissant cet exemple :

Table 2: Minimum SNR required to select the IEEE 802.11ac MCS indexes 2, 5, and 7, and corresponding fair share for 3 FAPs.

SNR (dB)	MCS index	Data Rate (Mbit/s)	Fair share (Mbit/s)
15	2	175.5	$0.80 \times \frac{175.5}{3 \text{ FAPs}} = 50$
27	5	468	$0.80 \times \frac{468}{3 \text{ FAPs}} = 133$
35	7	585	$0.80 \times \frac{585}{3 \text{ FAPs}} = 166$

Figure 1 – **Tableau** : Spécifications IEEE 802.11ac lien SNR et trafic demandé par une UAV

Avec le SNR obtenu d'une manière pratique, nous pouvons appliquer l'algorithme GPQM suivant :

**Algorithm 1:** GPQM algorithm

---

**Input:** Traffic demand  $T_i$ , in bit/s, of each FAP<sub>*i*</sub>,  
packet size  $S$  in bytes, and position  $P_i$ , in  
3D Cartesian coordinates, of each FAP<sub>*i*</sub>

**Output:** Transmission (TX) power  $P_T$ , in dBm,  
for all UAVs, FGW position  $P_0$  in 3D  
Cartesian coordinates, and queue size  $Q_i$ ,  
in number of packets, for each FAP<sub>*i*</sub>

---

```

/* 0 dBm TX power */
1  $P_T \leftarrow 0$ 
/* FAPi packet inter-arrival time */
2  $\lambda_i \leftarrow \frac{T_i}{S \times 8 \text{ bits/packet}}$ 
/* Iterate up to maximum allowed TX power -> 0(n) */
3 while  $P_T \leq P_T^{MAX}$  do
    /* Compute the FGW position, considering
        $i \in \{1, \dots, N-1\}$ . SNRi is determined taking
       into account  $T_i$  by means of a lookup table
       similar to Table 2 */
    4 computeFgwPos( $[x_i], [y_i], [z_i], [\text{SNR}_i], P_T$ )
    /* i.e.,  $(x_0, y_0, z_0) \in S_G$  */
    5 if  $(x_0, y_0, z_0) \neq \emptyset$  then
        /* FAPi packet service time */
        6  $\mu_i \leftarrow \frac{C_{0,i}}{S \times 8 \text{ bits/packet}}$ 
        /* FAPi traffic load */
        7  $\rho_i = \frac{\lambda_i}{\mu_i}$ 
        /* FAPi queue size */
        8  $Q_i = \frac{1}{2} \times \frac{\rho_i^2}{1 - \rho_i}$ 
        /* FAPi delay */
        9  $D_i = \frac{2 - \rho_i}{2 \times \mu_i \times (1 - \rho_i)}$ 
        /* FAPi delay lower than max. delay  $H$  */
        10 if  $D_i < H$  then
            /* TX power for all UAVs, FGW position,
               and queue size for each FAPi */
            11 return  $P_T, (x_0, y_0, z_0), Q_i$ 
        12 else
            /* Increase TX power by 1 dBm */
            13  $P_T \leftarrow P_T + 1$ 
    /* Compute and return  $x_0, y_0$ , and  $z_0 \rightarrow 0(1)$  */
14 Function computeFgwPos( $[x_i], [y_i], [z_i], [\text{SNR}_i], P_T$ ):
    /* System of equations in the form of (6) */
    15  $(x_0 - x_1)^2 + (y_0 - y_1)^2 + (z_0 - z_1)^2 \leq$ 
         $\left(10^{\frac{K + P_T - \text{SNR}_1}{20}}\right)^2$ 
    16 ...
    17  $(x_0 - x_{N-1})^2 + (y_0 - y_{N-1})^2 + (z_0 - z_{N-1})^2 \leq$ 
         $\left(10^{\frac{K + P_T - \text{SNR}_{N-1}}{20}}\right)^2$ 
    18  $K = -20 \log_{10}(f_i) - 20 \log_{10}\left(\frac{4 \times \pi}{c}\right) - (P_N)$ 
    19 return  $(x_0, y_0, z_0)$ 

```

---

Figure 2 – **Script** : Algorithme GPQM permettant de trouver la position idéale minimisant la capacité sur le lien

Cet algorithme calcule le délai du lien en fonction de la capacité du lien, et s'arrête lorsque le délai est en dessous d'une valeur  $H$  prédéfinie. Cela répond au problème de la fonction objectif étant convexe. On se demandait comment savoir que nous avons atteint un minimum local où global de capacité : à toute fin pratique, nous définissons un seuil qui arrête l'optimisation lorsqu'atteint.

## 4 Implémentation de l'algorithme GPQM de placement de FGW

### 4.1 Description de l'algorithme

Cet algorithme s'implémente de manière similaire à ce qui a été fourni par le pseudocode des auteurs.

L'algorithme se divise en deux fonction, la fonction "*GPQM\_algorithm*", elle-même appelant la fonction *computeFgwPos*. La fonction *GPQM\_algorithm* prend en entrée les paramètres  $T_i$  (temps d'arrivée inter-paquets),  $S$  (taux de transmission des données sur le lien sans-fils UAV-FGW),  $P_i$  (position des UAVs en m),  $P_{MAXT}$  (seuil de puissance lors de la transmission sur le lien en dBm),  $C_{0i}$  (capacité offerte sur le lien en Mbps),  $H$  (seuil de délai maximal en ms),  $SNR_i$  (liste des ratio signal-sur-bruit en dB),  $f_i$  (fréquence de la porteuse en GHz), et  $PN$  (seuil de puissance du bruit sur le lien de communication sans-fils).

La fonction *computeFgwPos* prend quant à elle en paramètre d'entrée les positions  $x, y, z$  d'une seule UAV (extrait sous la forme  $x[i], y[i], z[i]$  en m), ainsi que les paramètres  $SNR_i$ ,  $PT$ ,  $f_i$  et  $PN$  décrits plus haut.

La fonction *computeFgwPos* trouve une solution du système (6) décrit dans l'article à l'aide d'une méthode de résolution de racine et d'un estimé initial de la solution recherchée, et retourne la position  $x_0$ ,  $y_0$  et  $z_0$  du gateway (FGW) correspondant, la fonction "*GPQM\_algorithm*" calcule ensuite le délai fournie avec la position calculée et augmente la puissance de transmission  $PT$  sur le lien si le délai est jugé inadéquat.

Cependant, lorsque lancé, un problème survient lors de la résolution du système d'équations de la fonction "*computeFgwPos()*", celle-ci ne converge pas vers une solution.

Plusieurs méthodes de résolutions de racines ont été essayées, notamment la méthode de Newton et la fonction *fzero* du package Julia "Roots". L'estimé initial de la solution ne se trouve jamais être une valeur permettant à l'algorithme d'aboutir, ce qui fait que celui-ci ne retourne pas de valeur pertinente. L'utilisation de nombres aléatoires pour trouver l'estimé initial de la solution fut aussi tenté sans succès.

### 4.2 Résultats

L'algorithme, si utilisé avec un estimé initial  $pos_0$  lors de la résolution du système (6) de l'article, devrait fonctionner pour les valeurs de SNR (suggéré en 1) et de position de UAVs fournies par les auteurs de l'article afin de nous retourner des valeurs similaires à celles de ceux de l'article et ainsi valider la démarche décrite dans ce dernier.

Les auteurs ne fournissent pas leurs paramètres initiaux pour l'algorithme ni la méthode de résolution utilisée pour le système décrit à l'équation (6) de l'article. Cela mène à un problème lors de la reproduction de l'algorithme, où nous nous retrouvons dans l'incapacité de valider leurs résultats.

Par ailleurs, les fonctions décrites dans la section 5 de l'article (Performance evaluation) sont des fonctions propres au simulateur d'évolution de réseau dans le temps Network Simulator ns-3 non disponible sous Julia. Il est à noter qu'une version du simulateur écrite en C++ et Python existe cependant. Une solution serait donc de passer sur le simulateur utilisé par les auteurs.



## 5 Perspective personnelle sur le projet

Nous sommes ravis d'avoir développé le modèle ADNLP du problème, ainsi que l'algorithme GPQM développé par les auteurs; c'est là l'essence de la démarche scientifique. Malgré le fait que nous n'ayons pas pu arriver aux mêmes résultats que ces derniers, nous sommes confiants que plus d'information nous permettrait d'atteindre ces résultats.

Ce projet nous a permis de renforcer nos compétences en programmation avec Julia, notamment pour la définition du modèle ADNLP, qui a été plus ardue qu'anticipé. De plus, il nous a permis de comprendre l'importance de bien définir toutes les valeurs utilisées et d'être le plus clair possible lorsqu'on écrit un article scientifique. En effet, si jamais des gens comme nous veulent reproduire les résultats, il est important que la méthodologie soit claire et que tout code utilisé soit proprement référencé avec les traces adéquate (idéalement en utilisant git).

## 6 Lien vers les scripts de programmation

Les informations concernant les algorithmes implémentés pour réaliser ce projet sont disponibles sur [Github](#).

## Annexe

### Notation

Voici la notation utilisée pour le projet.

Notation	Définition
$B$	Largeur de bande constante d'un canal, en Hz
$c$	Vitesse de la lumière, en m/s
$C_{j,i}(t_k)$	Capacité maximale, en bit/s, du lien entre les drones $i$ et $j$ , à l'instant $t_k$
$C^{MAX}$	Capacité maximale globale, en bit/s
$D_i(t_k)$	Délai moyen, en secondes, des paquets appartenant au flot $F_{0,i}$ à l'instant $t_k$
$d_{max_i}$	Distance maximale, en mètres, entre le drone $i$ et le <i>gateway</i> , pour assurer la sélection du $MCS_i$
$E_T$	Ensemble des liens sans fil entre le drone $i$ et le <i>gateway</i>
$E(t_k)$	Ensemble des liens directionnels sans fil entre les drones $i$ et $j$ à l'instant $t_k$
$f$	Fréquence de la porteuse, en Hz
$F_{0,i}$	Flot transmis du drone $i$ au <i>gateway</i>
$G(t_k)$	Graphe orienté représentant le réseau volant à l'instant $t_k$
$H$	Délai moyen maximal alloué, en secondes
$L$	Distribution de référence équitable, en bit/s, de la capacité d'un canal sans fil
$M$	Distance maximale, en mètres, entre un drone $i$ et le <i>gateway</i>
$MCS_i$	Indice de modulation et de codage du drone $i$
$P_i$	Position du drone $i$
$P_N$	Puissance du bruit, en dBm
$P_T$	Puissance transmise totale, en dBm
$Q_i$	Taille de la file d'attente, en paquets, du drone $i$
$R_i(t,k)$	Débit binaire, en bit/s, du flot $F_{0,i}$ mesuré au <i>gateway</i> à l'instant $t_k$
$S$	Taille d'un paquet, en bytes
$S_G$	Sous-espace pour le placement du <i>gateway</i>
$SNR_i$	Rapport signal-sur-bruit minimal, en dB, pour sélectionner le $MCS_i$
$T_i(t_k)$	Débit binaire, en bit/s, du flot transmis du drone $i$ au <i>gateway</i> , à l'instant $t_k$
$T(V, E_T)$	Topologie active du réseau volant
$UAV_i$	Drone $i$
$V$	Ensemble des drones formant le réseau volant
$(x_i, y_i, z_i)$	Coordonnées cartésiennes 3D du drone $i$

## Références

Coelho, A., Campos, R., & Ricardo, M. (2022). *Traffic-aware Gateway Placement and Queue Management in Flying Networks*. Consulté sur <https://arxiv.org/pdf/2202.02911.pdf>