

**ĐẠI HỌC BÁCH KHOA HÀ NỘI**  
**\_\_TRƯỜNG CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG\_\_**

\*\*\*



**BÁO CÁO PROJECT 3**

**ĐỀ TÀI**

***“Áp dụng giải thuật di truyền GA cho bài toán Throughput Maximization for Backscatter-and Cache-Assisted Wireless Powered UAV Technology”***

Giảng viên hướng dẫn: Trịnh Văn Chiến

Sinh viên thực hiện: Lóp: IT2-04

Họ và tên	MSSV
Đinh Thanh Tùng	20200570
Ngô Công Dũng	20204729

**Hà Nội, 1/2024**

# MỤC LỤC

MỤC LỤC.....	2
DANH MỤC HÌNH ẢNH.....	2
LỜI NÓI ĐẦU .....	3
CHƯƠNG 1: GIỚI THIỆU BÀI TOÁN .....	4
1. Giới thiệu về UAV và các công nghệ hỗ trợ UAV .....	4
2. Bài toán tối ưu sử dụng 2 UAV trong không gian.....	4
CHƯƠNG 2: MÔ HÌNH HÓA BÀI TOÁN.....	7
1. Mô hình UAV .....	7
2. Caching model .....	8
3. Energy constraints.....	8
4. Shannon Capacity.....	9
5. Các ràng buộc và hàm mục tiêu.....	10
CHƯƠNG 3: GIẢI THUẬT ĐỀ XUẤT.....	12
1. Thuật toán GA .....	12
2. Thuật toán quy hoạch lồi – điều kiện KKT (so sánh) .....	14
CHƯƠNG 4: CÀI ĐẶT .....	16
CHƯƠNG 5: KẾT QUẢ.....	20
KẾT LUẬN.....	23
TÀI LIỆU THAM KHẢO .....	24

## DANH MỤC HÌNH ẢNH

Hình 1. Mô hình hệ thống: 1 UAV có hỗ trợ cache thực hiện BackCom và hoạt động .....	5
Hình 2. Hình ảnh mô tả vị trí UAV_1 .....	7
Hình 3. Quỹ đạo ban đầu của 2 UAV.....	18
Hình 4. Quỹ đạo của 2 UAV sau khi tối ưu sử dụng giải thuật GA .....	21
Hình 5: Quỹ đạo của 2 UAV sau khi tối ưu sử dụng thuật toán quy hoạch lồi .....	22

## LỜI NÓI ĐẦU

Trong thời đại ngày nay, sự phát triển không ngừng của công nghệ đã đem lại những cơ hội mới và tiềm năng không giới hạn. Trong lĩnh vực truyền thông, việc sử dụng các thiết bị không người lái (Unmanned Aerial Vehicles – UAVs) hỗ trợ backscatter và cache đang nổi lên như một giải pháp đầy triển vọng.

Project 3 này tập trung vào việc nghiên cứu và phân tích giải thuật di truyền GA cho bài toán kết hợp giữa UAVs, backscatter và cache trong hệ thống truyền thông mạng để tối đa hóa thông lượng thông tin truyền từ nguồn tới đích. Báo cáo này sẽ làm rõ hơn về cách tối đa hóa thông lượng trong truyền thông mạng sử dụng UAV có hỗ trợ backscatter và cache.

Chương đầu tiên sẽ tập trung giới thiệu về khái niệm, và bài toán. Chương 2 mô hình hóa lại bài toán bằng các công thức toán học. Chương 3 thì sẽ đề xuất các giải thuật để giải quyết bài toán. Chương 4 sẽ là phần cài đặt thuật toán và chương cuối cùng sẽ là kết quả của bài toán cũng như vấn đề của Project 3 này.

Chúng em xin cảm ơn thầy Trịnh Văn Chiến đã đóng góp và hỗ trợ bọn em trong quá trình nghiên cứu và viết báo cáo này. Chúng em cũng hi vọng rằng nội dung báo cáo sẽ đem lại những kiến thức hữu ích và mới mẻ trong lĩnh vực truyền thông mạng.

Trân trọng,

Ngô Công Dũng, Đinh Thanh Tùng.

# CHƯƠNG 1: GIỚI THIỆU BÀI TOÁN

## 1. Giới thiệu về UAV và các công nghệ hỗ trợ UAV

Unmanned aerial vehicles (UAV) là thiết bị bay không người lái có khả năng triển khai linh hoạt, chi phí thấp và khả năng cơ động cao. Đặc biệt, nếu được thiết kế và triển khai hợp lý, UAV có thể cung cấp các giải pháp cho mạng truyền thông không dây.

UAV được đề xuất dưới dạng chuyển tiếp để cải thiện khả năng kết nối các mạng. Đặc biệt, trong trường hợp các thiết bị liên kết truyền thông trực tiếp bị thiếu do bị che khuất hoặc không liên lạc được trong giờ cao điểm. UAV được triển khai dưới dạng chuyển tiếp để giúp truyền thông tin từ nguồn tới đích.

Với đề tài này, chúng ta sẽ nghiên cứu mạng truyền thông UAV chạy bằng năng lượng không dây với công nghệ tán xạ ngược (backscatter) và bộ nhớ đệm (cache).

Cụ thể chúng ta giả định rằng UAV tự cung cấp năng lượng có bộ nhớ đệm cache được triển khai dưới dạng thiết bị bay tán xạ ngược (BD), gọi là BD hỗ trợ UAV (UB), để chuyển tiếp thông tin từ nguồn tới đích.

## 2. Bài toán tối ưu sử dụng 2 UAV trong không gian

Với những điều đã nói ở trên và thực tế là năng lượng không dây, bộ nhớ đệm và BackCom là các công nghệ truyền thông tiết kiệm năng lượng cho các mạng truyền thông UAV. Chúng ta sẽ nghiên cứu mạng BackCom hỗ trợ 2 UAV tạo bộ nhớ đệm, trong đó một UAV có thể lưu trữ một phần nội dung phổ biến trong bộ đệm (cache) của nó. Bên cạnh đó, UAV được trang bị mạch thu năng lượng có thể thu tín hiệu RF từ nguồn và sau đó sử dụng năng lượng này cho BackCom và truyền thông tin đến đích.

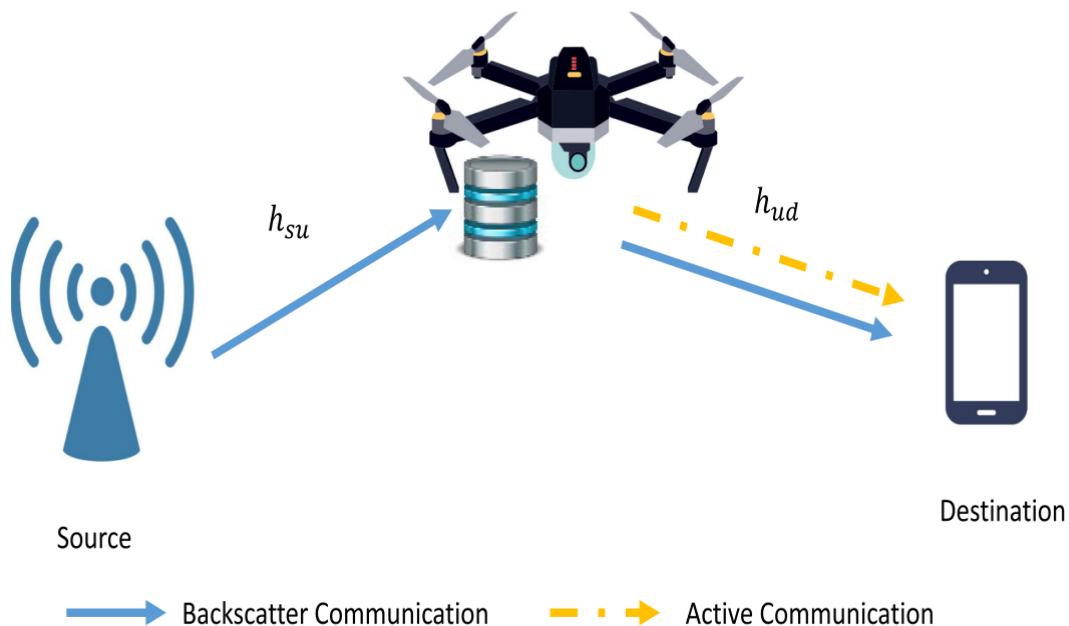
Vì vậy, đề xuất trong báo cáo này là một mạng truyền thông UAV không dây hỗ trợ tán xạ ngược (backscatter) và hỗ trợ bộ nhớ đệm (cache). Do đây là công trình cùng xem xét UAV, cache, và BackCom, điều này đặt ra những

thách thức lớn trong mô hình hóa được hệ thống và giải quyết các vấn đề liên quan.

Bởi vì UB (UAV-enabled BD) di chuyển từ vị trí ban đầu đến vị trí cuối cùng nên nó không thể di chuyển qua nguồn mọi lúc được. Do đó, tỷ lệ phân chia thời gian động (DTS) và quỹ đạo UB phải được thiết kế cẩn thận để tối đa hóa tổng thông lượng của hệ thống trong khi vẫn đáp ứng được hạn chế về năng lượng. Nếu như  $\tau n$  lớn, thời gian được phân bổ nhiều hơn để truyền dữ liệu nhưng ít thời gian để thu năng lượng và ngược lại.

Với những vấn đề trên, chúng ta xây dựng một bài toán tối ưu hóa để tối đa hóa tổng thông lượng truyền đến đích, tùy thuộc vào các ràng buộc thời gian bay giới hạn, tốc độ tối đa của UB, quỹ đạo của UB và tỷ lệ của DTS trong mỗi khoảng thời gian. Đây là bài toán non-convex và là vấn đề cần giải quyết.

Chúng ta phân tách bài toán thành 2 bài toán con, trong đó đầu tiên với mỗi UAV, tối ưu tỷ lệ DTS với 1 quỹ đạo UB nhất định, tiếp theo tối ưu hóa quỹ đạo của UB với tỷ lệ DTS nhất định.



Hình 1. Mô hình hệ thống: 1 UAV có hỗ trợ cache thực hiện BackCom và hoạt động

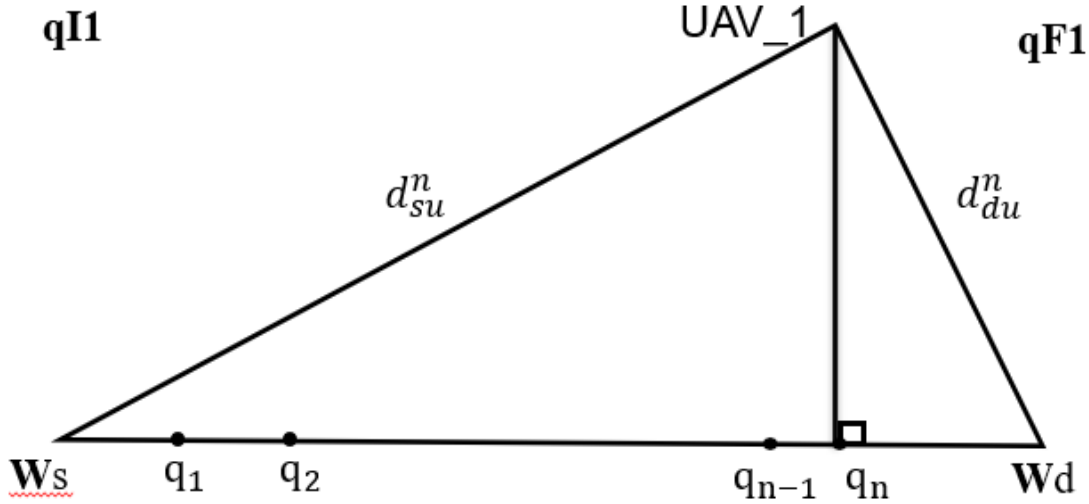
Trong Fig. 1 thì ta có thể thấy được với 1 UAV:

- Nguồn (Source) hoạt động như một bộ sạc không dây hoặc trạm gốc để cung cấp điện và truyền thông tin
- Đích (Destination) là các thiết bị cần kết nối, nhận thông tin truyền đến mà không thể thực hiện kết nối trên mặt đất

## CHƯƠNG 2: MÔ HÌNH HÓA BÀI TOÁN

### 1. Mô hình UAV

UAV bay trong không gian 3D với tọa độ là  $(x, y, z)$  có các thông số (giả sử các UAV đều có chất lượng và giá thành như nhau):



Hình 2. Hình ảnh mô tả vị trí UAV\_1

- Tổng thời gian bay của UAV là  $T(s)$  được chia thành  $N$  khe thời gian.
- Thời gian một khe thời gian:  $\delta_t = \frac{T}{N}$ .
- Vị trí của nguồn (S) và đích (D) tương ứng là  $w_s, w_d$ .
- Vị trí bắt đầu và kết thúc của UAV\_1 tương ứng là  $q_{I1}, q_{F1}$ .
- Vị trí bắt đầu và kết thúc của UAV\_1 tương ứng là  $q_{I2}, q_{F2}$ .

Mỗi UAV phải thỏa mãn:

$$1. \|q_{n+1} - q_n\| \leq \delta_d = V_{max}\delta_t, n = \overline{0, N-1}$$

Trong đó:  $\mathbf{q}_i$  với  $n = \overline{0, N}$  là vị trí của UAV tại khe thời gian  $i$ ;  $V_{max}$  là vận tốc tối đa của UAV.

2.  $\mathbf{q}_0 = \mathbf{q}_I, \mathbf{q}_N = \mathbf{q}_F$  với  $\mathbf{q}_I, \mathbf{q}_F$  là vị trí bắt đầu và kết thúc của mỗi UAV.

3.  $d_{su}^n = \sqrt{\|\mathbf{q}_n - \mathbf{w}_s\|^2}$  ;  $d_{du}^n = \sqrt{\|\mathbf{q}_n - \mathbf{w}_d\|^2}$  với  $n = \overline{0, N}$

Trong đó:  $d_{su}^n, d_{du}^n$  lần lượt là khoảng cách từ UAV đến nguồn và đích.

4. Hệ số kênh truyền trong khe thời gian  $n$ :

$$h_{iu}^n = \sqrt{\psi_{iu}^n} \tilde{h}_{iu}^n, i \in \{d, s\}$$

Với  $\psi_{iu}^n$  và  $\tilde{h}_{iu}^n$  lần lượt là 2 loại suy hao năng lượng tín hiệu Large-scale fading (suy hao năng lượng tầm rộng) và Small-scale fading (suy hao năng lượng tầm hẹp).

Trong đó:  $\psi_{iu}^n = \omega_0 (d_{iu}^n)^{-\alpha}$ ;  $\tilde{h}_{iu}^n = \sqrt{\frac{K}{1+K}} \bar{h}_{iu}^n + \sqrt{\frac{1}{1+K}} \hat{h}_{iu}^n$  với  $\bar{h}_{iu}^n, \hat{h}_{iu}^n$  lần

lượt là hệ số LoS (tín hiệu truyền thẳng) và NLoS (tín hiệu truyền bị phản xạ hoặc khúc xạ).

## 2. Caching model

Ta gọi  $\sigma_1$  và  $\sigma_2$  là hệ số bộ nhớ đệm của 2 UAV. UAV có thể lưu trữ  $0 \leq \sigma_i \leq 1, i \in \{1, 2\}$  phần của mỗi file cần truyền trong bộ nhớ cache của nó. Do đó, mỗi nguồn chỉ cần gửi  $1 - \sigma_i$  phần còn lại của file đến UAV để truyền đến đích.

## 3. Energy constraints



Do khả năng lưu trữ năng lượng hạn chế tại UB, chúng ta sẽ sử dụng giải pháp EH (Energy Harvesting), mỗi UAV thực hiện 2 giai đoạn trong một khe thời gian  $\delta_t$ :

- $\tau_n \cdot \delta_t$  là thời gian truyền dữ liệu
- Và  $(1 - \tau_n)\delta_t$  là thời gian sạc cho UAV.

Do rằng tổng năng lượng tiêu thụ của UAV phải nhỏ hơn hoặc bằng tổng năng lượng thu được cho đến khe thời gian n nên ta có:

$$\sum_{i=1}^n (E_{fly}^n(q) + \tau_n \delta_t (P_b + P_u)) \leq \sum_{i=1}^n E_h^n$$

Trong đó:

- $E_{fly}^n$  là năng lượng cần thiết để UAV bay trong khe thời gian  $\delta_t$ ;
- $E_h^n$  là năng lượng UAV thu được trong khe thời gian  $\delta_t$ ;
- $P_s$  là năng lượng truyền của nguồn S cho việc truyền tin;
- $P_u$  là năng lượng truyền của UAV tạo khe thời gian n cho việc truyền tin;
- $P_b$  là công suất mạch của UB trong khoảng thời gian backscatter (tán xạ ngược).

#### 4. Shannon Capacity

Shannon Capacity (giới hạn Shannon) của một kênh truyền thông là tỷ lệ tối đa trên lý thuyết về lượng thông tin một kênh truyền thông có thể truyền tải, với một độ nhiễu nhất định.

Công thức tổng quát tính tốc độ dữ liệu của 1 UAV là:

$$R = B \log_2 \left( 1 + \frac{\text{Năng lượng tín hiệu}}{\text{Năng lượng nhiễu}} \right)$$

Khi đó ta có tốc độ dữ liệu gần đúng đạt được lần lượt tại mỗi UB và đích trong khe thời gian n là:

$$\bar{R}_u^n = B \log_2 \left( 1 + \frac{e^{-E} w_0 P_s}{(\|q_n - w_s\|^2)^{\frac{\alpha}{2}} \sigma_u^2} \right)$$

$$\bar{R}_d^n = B \delta_t \log_2 \left( 1 + \frac{\theta(\eta_u^n w_0 P_s + \bar{P}_u (d_{su}^n)^\alpha)}{\varrho} \right)$$

Trong đó:

- $\theta = \frac{e^{-E} w_0}{\sigma_u^2}$
- $\bar{P}_u = \lceil \sigma \rceil P_u$
- $\varrho = \|\mathbf{q}_n - \mathbf{w}_s\|^\alpha \|\mathbf{q}_n - \mathbf{w}_d\|^\alpha$

Với  $\sigma_u^2$ : công suất nhiễu tại UAV;

$\alpha$ : số mũ suy hao đường dẫn;

$w_0$ : độ tăng công suất kênh;

$\eta_u^n$ : hệ số tán xạ ngược.

## 5. Các ràng buộc và hàm mục tiêu

Với mục đích tối đa hóa tổng truyền dữ liệu từ UAV đến D, bài toán phải thỏa mãn các ràng buộc sau:

- Dữ liệu từ UAV truyền đi phải lớn hơn dữ liệu đích nhận được:

$$B \sum_{n \in N} \tau_n \delta_t \log_2 \left( 1 + \frac{e^{-E} w_0 P_s}{(H^2 + \|q_n - w_s\|^2)^{\frac{\alpha}{2}} \sigma_u^2} \right) + \sigma S \geq B \sum_{n \in N} \tau_n \delta_t \log_2 \left( 1 + \frac{\theta(\eta_u^n w_0 P_s + \bar{P}_u (d_{su}^n)^\alpha)}{Q} \right)$$

- Tốc độ dữ liệu nhận được tại đích phải hơn hơn S (S là tốc độ dữ liệu yêu cầu tối thiểu tại đích) nhằm đảm bảo yêu cầu kỹ thuật và nhu cầu của người dùng:

$$B \sum_{n \in N} \tau_n \delta_t \log_2 \left( 1 + \frac{\theta(\eta_u^n w_0 P_s + \bar{P}_u (d_{su}^n)^\alpha)}{Q} \right) \geq S$$

- Tổng năng lượng bay, backscatter (tán xạ ngược) và trao đổi thông tin không được vượt quá năng lượng UAV đã nhận được:

$$\sum_{i=1}^n \left( E_{fly}^n(q) + \tau_n \delta_t (P_b + P_u) \right) \leq \sum_{i=1}^n \frac{\mu(1 - \tau_n) \delta_t w_0 P_{WPT}}{(H^2 + \|q_n - w_s\|^2)^{\alpha/2}}$$

- Năng lượng truyền của nguồn và UAV cho việc truyền tin phải nhỏ hơn một giá trị năng lượng tối đa:

$$P_u + P_s \leq P_{MAX}$$

- Các ràng buộc về phần cứng của mô hình của UAV tại mỗi khe thời gian:  
 $\|q_{n+1} - q_n\| \leq \delta_d = V_{max} \delta_t, n = 0, \dots, N - 1$   
 $q_0 = q_I, q_N = q_F$   
 $0 \leq \tau_n \leq 1, n \in N$

**Hàm mục tiêu:**

$$\mathcal{P}_1 : \max_{q, \tau} B \sum_{n \in \mathcal{N}} \tau_n \delta_t \log_2 \left( 1 + \frac{\Theta(\eta_u^n \omega_0 P_s + \bar{P}_u (d_{su}^n)^\alpha)}{\varrho} \right)$$

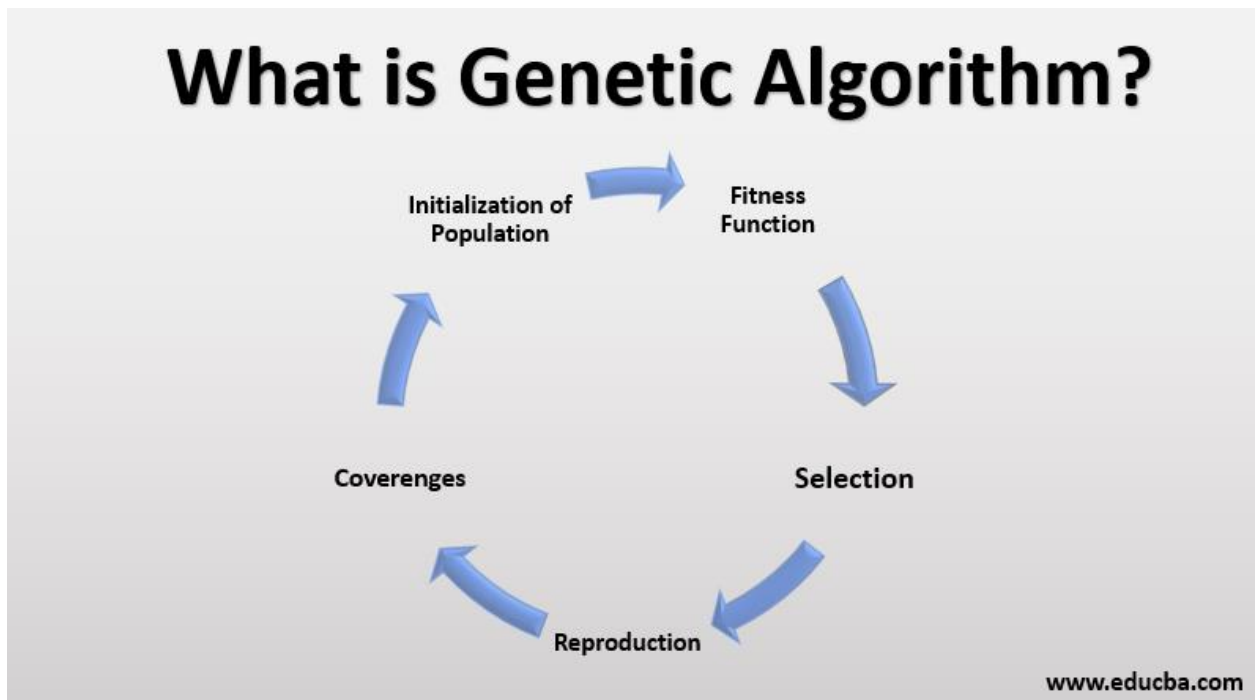
Trong đó:  $q = \{q_n, n \in N\}$ ;  $\tau = \{\tau_n, n \in N\}$ .

## CHƯƠNG 3: GIẢI THUẬT ĐỀ XUẤT

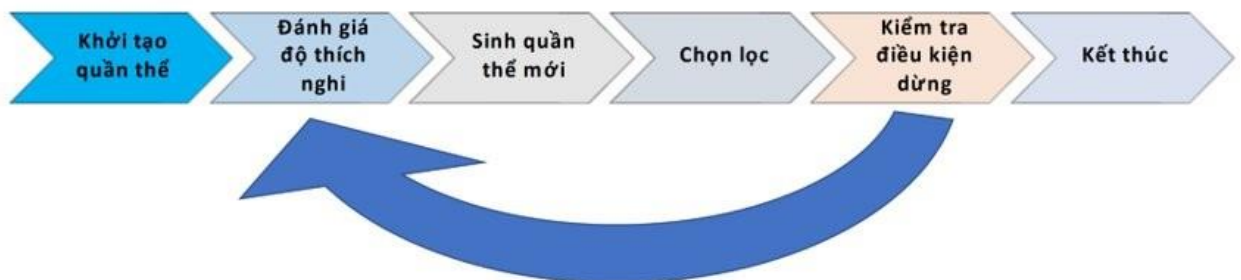
### 1. Thuật toán GA

Thuật toán GA (Genetic Algorithm) là thuật toán tiến hóa dùng để tối ưu giải quyết các vấn đề tối ưu không liên tục. Chúng ta sẽ xét thêm thuật toán này để giải quyết bài toán trên sau đó so sánh 2 kết quả nhận được và đánh giá.

Mô hình thuật toán GA:



Các bước thực hiện thuật toán GA



### 1.1 Mã hóa lời giải

Kết quả cuối cùng của bài toán là tìm ra được lời giải tốt nhất bao gồm tọa độ (x, y, z), tau (tỷ lệ truyền tin và nhận năng lượng), Pu (công suất truyền tin của UAV) và Ps (công suất truyền tin tại nguồn) tại các khe thời gian.

Cách mã hóa lời giải:

- Chuẩn hóa tọa độ x,y,z về [0, 1]
- Chuẩn hóa Ps, Pu về [0, 1]

Tham số đầu vào (input) của bài toán sẽ gồm một mảng số thực trong khoảng (0, 1) theo thứ tự là giá trị của x, y, z, tau, Pu, Ps.

Tham số đầu ra (output) kết quả của bài toán cũng là một mảng số thực trong khoảng (0,1) biểu diễn giá trị x, y, z, tau, Pu, Ps.

### 1.2 Khởi tạo quần thể

Khởi tạo quần thể ban đầu với các giá trị ngẫu nhiên trong khoảng (0, 1) với số các thể là *size*.

Sử dụng hàm đánh giá (fitness function) để đánh giá ban đầu với những ràng buộc cần thiết, lặp lại quá trình khởi tạo đến khi tạo ra được quần thể thỏa mãn tất cả các ràng buộc ban đầu của bài toán.

### 1.3 Lai ghép (crossover)

Sử dụng chéo hóa *Arithmetical crossover* để lai ghép giữa hai cá thể cha và mẹ.

Chọn bất kỳ 2 cá thể P1, P2 trong quần thể hiện tại để lai ghép và chọn một số ngẫu nhiên u trong khoảng (0, 1).

Sau khi lai ghép tạo ra 2 cá thể con:

$$c1 = u * P1 + (1-u) * P2$$

$$c2 = (1-u) * P1 + u * P2$$

2 cá thể con mới được tạo ra được thêm vào quần thể mới. Lặp lại quá trình tạo cá thể con sao cho quần thể mới có số lượng cá thể đúng bằng "*size*" (số lượng cá thể trong quần thể).

### 1.4 Đột biến (Mutation)

Với quần thể mới vừa tạo ra ở phần lai ghép (crossover), chọn một vị trí bất kỳ ngẫu nhiên ở trên cá thể.

Gán lại giá trị ở vị trí này thành một giá trị ngẫu nhiên mới.

Sau đó thêm các cá thể mới ở quần thể mới vào quần thể hiện tại tạo ra quần thể mới có số lượng cá thể gấp đôi ( $2 \cdot size$ ).

### 1.5 Đánh giá độ thích nghi

Hàm đánh giá độ thích nghi là hàm mục tiêu của bài toán:

- Nếu không thỏa mãn ràng buộc khoảng cách bay, năng lượng => Giá trị thích nghi = 0.
- Nếu thỏa mãn ràng buộc khoảng cách, năng lượng nhưng không thỏa mãn về ràng buộc dữ liệu => Giá trị thích nghi = giá trị thích nghi \* 0.5
- Nếu thỏa mãn tất cả các ràng buộc => Giá trị thích nghi = giá trị thích nghi \* 1

### 1.6 Chọn lọc (Choose)

Với quần thể hiện tại có số lượng cá thể gấp đôi kích thước ban đầu của quần thể. Vì vậy, chúng ta cần chọn lọc cá thể theo thứ hạng giữ lại kích thước ban đầu của quần thể.

Chọn lọc phân cấp:

- Sắp xếp các cá thể trong quần thể dựa vào hàm thích nghi
- Chia quần thể thành: 40% tốt, 30% trung bình, 30% kém
- Lựa chọn: 50% tốt, 50% trung bình, 50% kém

## 2. Thuật toán quy hoạch lồi – điều kiện KKT (so sánh)

Để giải quyết vấn đề bài toán  $\mathcal{P}_1$  trên thì ta phân tách  $\mathcal{P}_1$  thành 3 bài toán con:

- Đầu tiên sẽ đi tối ưu hóa tỉ lệ DTS ( $\tau_n$ ) với một quỹ đạo nhất định và giá trị  $P_s, P_u$  ban đầu:

$$\mathcal{P}_1^\tau: \max_{\tau} \sum_{n \in \mathcal{N}} \tau_n \delta_t \bar{R}_d^n$$

- Thứ hai ta sẽ tối ưu quỹ đạo (trajectory) của UAV với tỷ lệ DTS vừa tối ưu ở trên:

$$\mathcal{P}_1^q: \max_q B \sum_{n \in \mathcal{N}} \tau_n \delta_t \log_2 \left( 1 + \frac{\Theta(\eta_u^n \omega_0 P_s + \bar{P}_u (d_{su}^n)^\alpha)}{\varrho} \right)$$

- Thứ ba ta sẽ tối ưu  $P_s, P_u$  với tỷ lệ DTS và quỹ đạo ở trên:

$$\mathcal{P}_1^{P_s, P_u}: \max_{P_s, P_u} B \sum_{n \in \mathcal{N}} \tau_n \delta_t \log_2 \left( 1 + \frac{\Theta(\eta_u^n \omega_0 P_s + \bar{P}_u (d_{su}^n)^\alpha)}{\varrho} \right)$$

Bằng thuật toán quy hoạch lồi – điều kiện KKT: ta luân phiên tối ưu 3 bài toán con đến khi thuật toán hội tụ đến một ngưỡng đã cho.

Do bài toán ban đầu là sử dụng 2 UAV để truyền tin nên chúng ta sẽ giải quyết bài toán  $\mathcal{P}_1$  cho từng UAV riêng biệt, từ đó kết hợp lại để giải quyết bài toán ban đầu.

## CHƯƠNG 4: CÀI ĐẶT

### 1. THUẬT TOÁN TỐI ƯU (GENETIC ALGORITHM)

#### 1. Khởi tạo quần thể ban đầu

Tạo hàm *init-one-child()*:

*@parameter: Init one individual;*

*@return:*

*\*individual: if pass fitness function*

*\*None: if not pass fitness function.*

- Khởi tạo  $x, y, z, e\_fly, e\_h, u$  (Pu),  $s$  (Ps)
- Vòng lặp từ 1 đến  $N-1$ :
  - Lấy vị trí  $x, y$  tại thời điểm  $i-1$
  - Khởi tạo vị trí từ  $(0, 1)$  sang thực tế. Khởi tạo vị trí  $x, y, z, u, s$  ngẫu nhiên tại thời điểm  $i$
  - Tính khoảng cách từ  $Q_{i-1}$  đến  $Q_i$ :  $dis$ . Tính  $e\_fly\_i$  và  $e\_h\_i$  tại thời điểm  $i$ . Đặt **checkpoint** khi không tìm được điểm thỏa mãn ràng buộc
  - Trong khi khoảng cách  $dis > d$  và  $e\_fly > e\_h$ :
    - a. **checkpoint++**
    - b. if **checkpoint** = 100 return None
    - c. Khởi tạo lại vị trí  $x, y, z, u, s$  ngẫu nhiên tại thời điểm  $i$ .
      - i. Tính  $dis, e\_fly\_i$  và  $e\_h\_i$ .
  - Thêm vị trí  $x, y, z, u, s$  tại thời điểm  $i$  vào danh sách
  - Cập nhật  $e\_fly$  và  $e\_h$ .
- Thêm vị trí cuối cùng của  $x, y, z, u, s$  vào danh sách. Tính **fitness** cho cá thể.
- If **fitness** > 0 return cá thể; else return None.

#### 2. Hàm đánh giá độ thích nghi (fitness)

Hàm *fitness* ( $c, log=False$ ):

*@parameter:*

*c: individual: list*

*@return:*

*\*0: if fail constraint: distance,  $E\_fly$ , total rate*

*\*rate/2: if only fail rate*



*\*rate: if pass all constraint*

- Khởi tạo  $x, y, z, u, s$  từ  $c$ . Chuyển đổi từ  $(0, 1)$  sang thực tế.
- Khởi tạo  $e\_fly, e\_h, r\_u$  và  $r\_d$
- Vòng lặp 0 đến  $N-1$ :
  - Tính  $P\_u\_bar, d\_su2, d\_du2, r\_ui$  và cập nhật  $r\_u$ . Tính  $r\_d\_i$  và cập nhật  $r\_d$
- Vòng lặp từ 0 đến  $N-1$ :
  - Tính  $q1, q0, dis$
  - Tính  $e\_fly\_i$  và  $e\_fly, e\_h\_i$  và  $e\_h$
- Vòng lặp từ 0 đến  $N-1$ :
  - if khoảng cách giữa  $l[i]$  và  $l[i+1]$  lớn hơn  $d$  trả về 0
  - if  $e\_fly > e\_h$  return 0
  - if  $r\_u + \sigma * S < r\_d$  return 0
  - if  $r\_d < S$  return  $r\_d / 2$
- return  $r\_d$

### 3. Hàm lai ghép (crossover)

Hàm `linear_cross_over(f, m)`:

*Cross over from parents*

*@parameter:*

*f: father*

*m: mother*

*@return: 2 children from parents*

- Sao chép  $f$  và  $m$  thành  $c1$  và  $c2$
- Khởi tạo  $u$  ngẫu nhiên trong khoảng  $(0,1)$
- Vòng lặp từ 0 đến  $3*N+2$ :
  - Cập nhật  $c1[i]$  và  $c2[i]$  dựa trên  $f[i], m[i]$  và  $u$
- Return  $c1, c2$

### 4. Hàm đột biến

Hàm `random_mutation(c)`:

*@parameter: c: individual*

- Khởi tạo ngẫu nhiên  $p$  từ  $[-1, 1]$

- If  $p < 0$ :  $p = 0$
- Cập nhật bằng giá trị ngẫu nhiên hoặc  $p$ .

5. Chọn lọc

Chọn lọc phân cấp 40% tốt nhất, 30% trung bình, 30% kém.

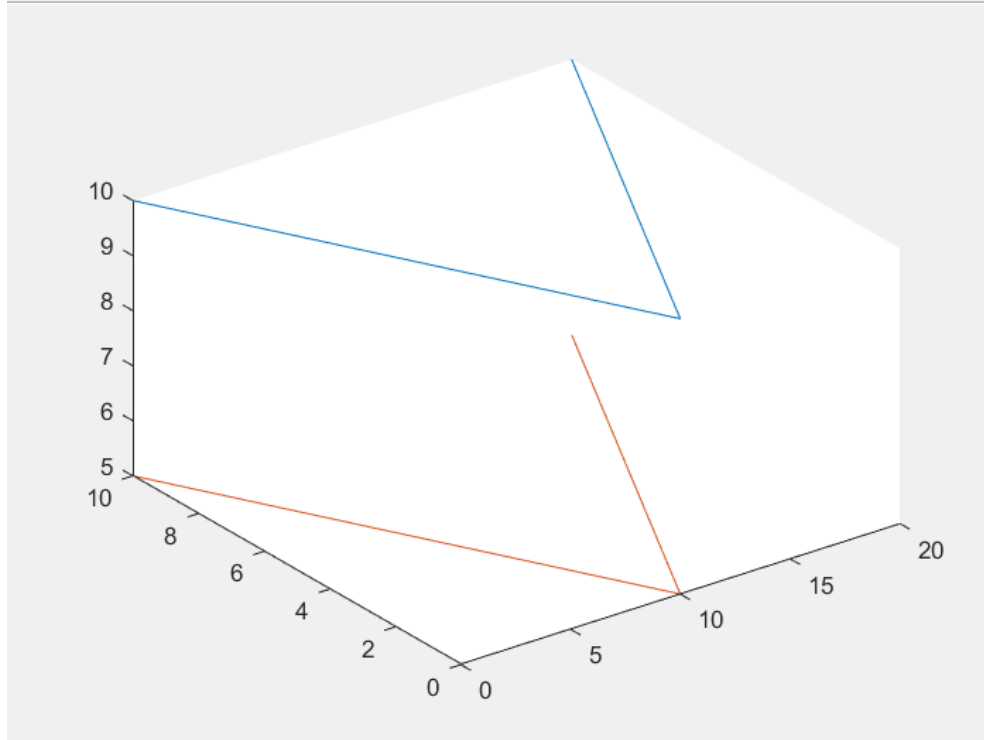
## 2. THUẬT TOÁN QUY HOẠCH LỖI – ĐIỀU KIỆN KKT

Ta thực hiện các bước sau:

1. Khởi tạo quỹ đạo ban đầu với

$$\mathbf{q}_{I1} = (0; 10; 10); \mathbf{q}_{F1} = (20; 10; 10)$$

$$\mathbf{q}_{I2} = (0; 10; 5); \mathbf{q}_{F1} = (20; 10; 5)$$



Hình 3. Quỹ đạo ban đầu của 2 UAV

2. Với mỗi UAV ta tối ưu hóa tỉ lệ DTS ( $\tau_n$ ) với quỹ đạo vừa khởi tạo ở trên với các ràng buộc sau:

- $\max_{\tau} \sum_{n \in \mathcal{N}} \tau_n \delta_t \bar{R}_d^n;$
- $\sum_{n \in \mathcal{N}} \tau_n \delta_t \bar{R}_u^n + \sigma S \geq \sum_{n \in \mathcal{N}} \tau_n \delta_t \bar{R}_d^n;$

- $\sum_{n \in N} \tau_n \delta_t \bar{R}_d^n \geq S;$
- $\sum_{i=1}^n \left( E_{fly}^n(q) + \tau_n \delta_t (P_b + P_u) \right) \leq \sum_{i=1}^n \frac{\mu(1-\tau_n) \delta_t w_0 P_{WPT}}{(\|q_n - w_s\|^2)^{\alpha/2}};$
- $0 \leq \tau_n \leq 1, n \in N$

3. Với tỷ lệ DTS( $\tau_n$ ) vừa tối ưu ở trên ta sẽ tối ưu quỹ đạo (trajectory) của mỗi UAV với các ràng buộc sau:

- $\max_q B \sum_{n \in N} \tau_n \delta_t \log_2 \left( 1 + \frac{\theta(\eta_u^n \omega_0 P_s + \bar{P}_u (d_{su}^n)^\alpha)}{Q} \right);$
- $B \sum_{n \in N} \tau_n \delta_t \log_2 \left( 1 + \frac{e^{-E w_0 P_s}}{(\|q_n - w_s\|^2)^{\frac{\alpha}{2}} \sigma_u^2} \right) +$   
 $\sigma S \geq B \sum_{n \in N} \tau_n \delta_t \log_2 \left( 1 + \frac{\theta(\eta_u^n w_0 P_s + \bar{P}_u (d_{su}^n)^\alpha)}{Q} \right);$
- $B \sum_{n \in N} \tau_n \delta_t \log_2 \left( 1 + \frac{\theta(\eta_u^n w_0 P_s + \bar{P}_u (d_{su}^n)^\alpha)}{Q} \right) \geq S;$
- $\sum_{i=1}^n \left( E_{fly}^n(q) + \tau_n \delta_t (P_b + P_u) \right) \leq \sum_{i=1}^n \frac{\mu(1-\tau_n) \delta_t w_0 P_{WPT}}{(\|q_n - w_s\|^2)^{\alpha/2}};$
- $\|q_{n+1} - q_n\| \leq \delta_d = V_{max} \delta_t, n = 0, \dots, N-1$  (1);  $q_0 = q_I, q_N = q_F$  (2);

4. Sử dụng tỷ lệ DTS( $\tau_n$ ) và quỹ đạo vừa tối ưu ở trên ta đi tối ưu lại quỹ đạo của mỗi UAV và tính hàm mục tiêu với biến tối ưu  $P_s, P_u$  qua các ràng buộc sau:

- $\max_{P_s, P_u} B \sum_{n \in N} \tau_n \delta_t \log_2 \left( 1 + \frac{\theta(\eta_u^n \omega_0 P_s + \bar{P}_u (d_{su}^n)^\alpha)}{Q} \right);$
- $B \sum_{n \in N} \tau_n \delta_t \log_2 \left( 1 + \frac{e^{-E w_0 P_s}}{(\|q_n - w_s\|^2)^{\frac{\alpha}{2}} \sigma_u^2} \right) +$   
 $\sigma S \geq B \sum_{n \in N} \tau_n \delta_t \log_2 \left( 1 + \frac{\theta(\eta_u^n w_0 P_s + \bar{P}_u (d_{su}^n)^\alpha)}{Q} \right);$
- $B \sum_{n \in N} \tau_n \delta_t \log_2 \left( 1 + \frac{\theta(\eta_u^n w_0 P_s + \bar{P}_u (d_{su}^n)^\alpha)}{Q} \right) \geq S;$
- $\sum_{i=1}^n \left( E_{fly}^n(q) + \tau_n \delta_t (P_b + P_u) \right) \leq \sum_{i=1}^n \frac{\mu(1-\tau_n) \delta_t w_0 P_{WPT}}{(\|q_n - w_s\|^2)^{\alpha/2}};$
- $0 \leq P_s + P_u \leq P_{MAX};$

## CHƯƠNG 5: KẾT QUẢ

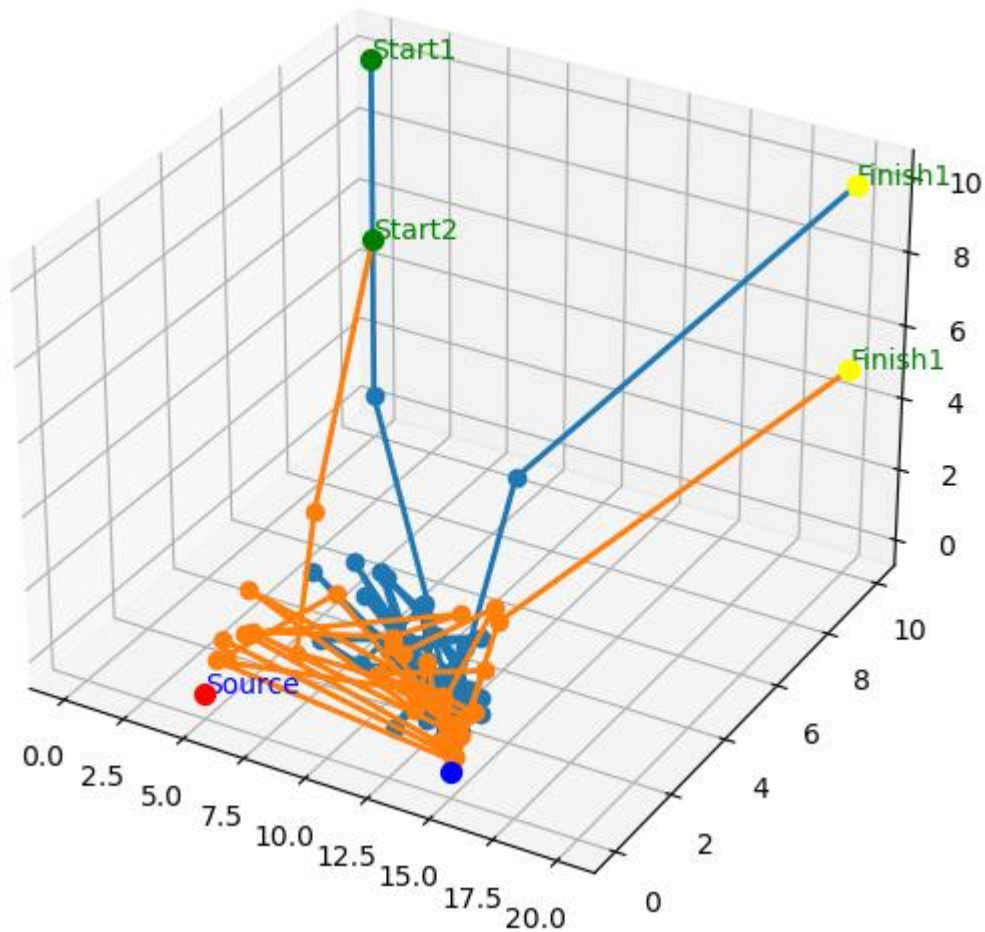
### Tham số bài toán:

Max_iteration	5	Nhiều $\sigma^2$	-90 dB
Tốc độ tối đa, $V_{max}$	20m/s	Hệ số mất mát, $\alpha$	2.2
Vị trí bắt đầu UAV1, $q_{I1}$	[0;10;10]	Năng lượng tán xạ, $P_b$	$10^{-6}$ W
Vị trí kết thúc UAV1, $q_{F1}$	[20;10;10]	Tiêu hao theo khoảng cách, $w_0$	-30dB
Vị trí bắt đầu UAV2, $q_{I2}$	[0;10;5]	Giá trị $P_s$ ban đầu	16 dBm
Vị trí kết thúc UAV2, $q_{F2}$	[20;10;5]	Giá trị $P_u$ ban đầu	10 mW
Vị trí nguồn, $w_S$	[5;0;0]	Tỉ lệ thu năng lượng, $\mu$	0.84
Vị trí đích, $w_D$	[15;0;0]	Năng lượng truyền tin nguồn, $P_s$	16dBm
Thời gian bay, T	20s	Năng lượng xác của nguồn, $P_{WPT}$	70dB
Số khe thời gian, N	40	Dữ liệu tại nguồn, S	50Mbits
Giá trị mỗi khe t/gian, $\delta_t$	0.5s	Hệ số bộ nhớ đệm, $\sigma$	0.5

### Kết quả đạt được:

*Giải thuật di truyền GA:*

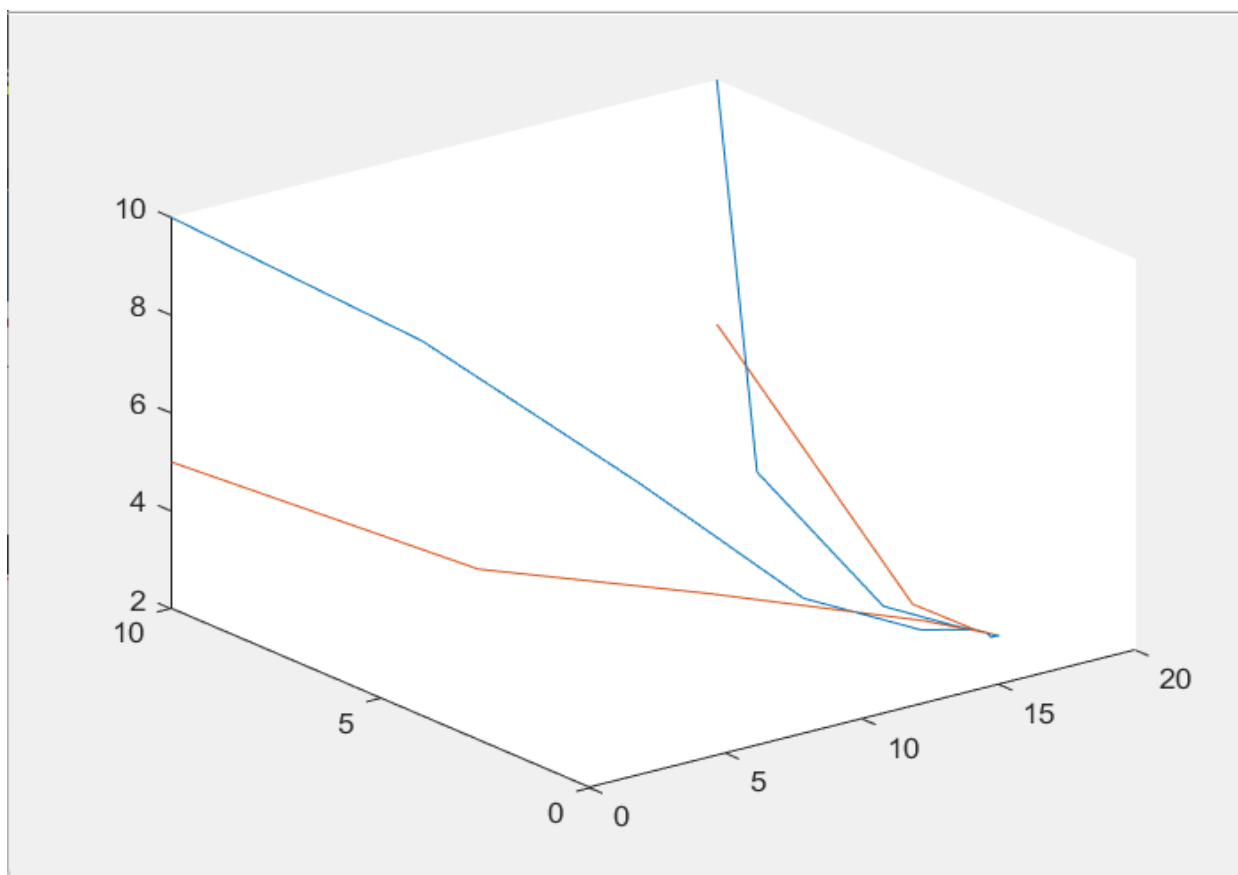
- Hàm mục tiêu(Rate): 159.28965 Mbit/s
- Quỹ đạo của 2 UAV:



Hình 4. Quỹ đạo của 2 UAV sau khi tối ưu sử dụng giải thuật GA

Thuật toán quy hoạch lồi – điều kiện KKT :

- Hàm mục tiêu(Rate): 155.234315 Mbit/s
- Quỹ đạo của 2 UAV:



Hình 5: Quỹ đạo của 2 UAV sau khi tối ưu sử dụng thuật toán quy hoạch lồi

## KẾT LUẬN

Bài báo cáo này trình bày sơ lược về nội dung của Project 3 “Áp dụng giải thuật di truyền GA cho bài toán Throughput Maximization for Backscatter-and Cache-Assisted Wireless Powered UAV Technology”. Các vấn đề đưa ra trong báo cáo này là giới thiệu bài toán, mô hình hóa lại bài toán, từ đó đưa ra các giải thuật đề xuất rồi tiến hành cài đặt. Kết quả cuối cùng sẽ được so sánh với 1 thuật toán lập khác để đánh giá.

Với các phần nội dung lý thuyết, giải thuật đề xuất cũng như cách cài đặt trên thì chúng em đã sử dụng Python để mô phỏng bài toán và cài đặt thuật toán để giải quyết bài toán và thu được kết quả.

Kết quả thu được của bài toán với một bộ tham số nhất định là giá trị thông lượng đạt được tối đa của mạng sử dụng 2 UB trong không gian 3D, và quỹ đạo của 2 UB đây. Mặc dù báo cáo đã nhận được những kết quả khá quan trọng và chính xác, tuy nhiên vẫn còn nhiều hạn chế vì mới dừng ở việc sử dụng 2 UB, và chưa đề xuất được thuật toán nào tốt hơn để so sánh,... Vì vậy mong muốn là trong thời gian tới sẽ xem xét kỹ hơn về đề tài này.

Qua Project 2 này thì chúng em đã đạt được rất nhiều kinh nghiệm cho bản thân và cũng rút ra được nhiều bài học cho mình. Em chân thành cảm ơn thầy hướng dẫn đã giúp đỡ em rất nhiều trong thời gian nghiên cứu và thực hiện đồ án này.

## **TÀI LIỆU THAM KHẢO**

1. “Throughput Maximization for Backscatter- and Cache-Assisted Wireless Powered UAV Technology” Dinh-Hieu Tran , Graduate Student Member, IEEE, Symeon Chatzinotas , Senior Member, IEEE, and Björn Ottersten , Fellow, IEEE