

# Project 5

## Taxi vận chuyển người kết hợp hàng hóa

Trần Huy Hùng   Đỗ Ngọc Sơn

Đại học Bách Khoa Hà Nội

Ngày 4 tháng 6 năm 2020

# Nội dung

- 1 Giới thiệu bài toán
- 2 Mô hình bài toán
- 3 Giải thuật chính xác
- 4 Giải thuật heuristics
- 5 Thực nghiệm và đánh giá
- 6 Kết luận

# Nội dung

- 1 Giới thiệu bài toán
- 2 Mô hình bài toán
- 3 Giải thuật chính xác
- 4 Giải thuật heuristics
- 5 Thực nghiệm và đánh giá
- 6 Kết luận

# Giới thiệu bài toán

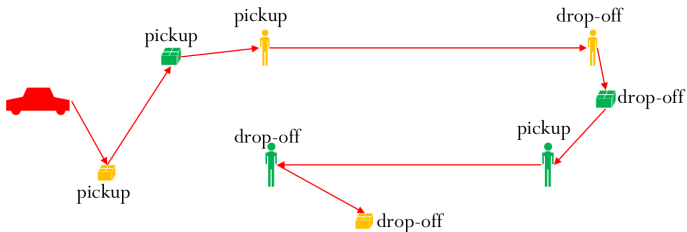
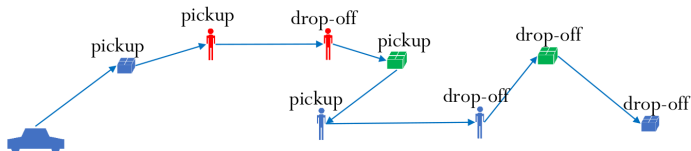
Bài toán taxi vận chuyển người kết hợp hàng hóa:

- Có  $N$  hành khách  $(1, 2, \dots, N)$  và  $M$  gói hàng  $(N + 1, N + 2, \dots, N + M)$ :
  - Hành khách (hoặc gói hàng)  $i$  có điểm đón  $i$  và điểm trả  $i + N + M$  ( $u = 1, 2, \dots, N + M$ ).
  - Gói hàng  $i$  có khối lượng  $q_i$  ( $i = N + 1, \dots, N + M$ ).
- Có  $K$  xe taxi  $(1, 2, \dots, K)$ :
  - Các taxi cùng xuất phát từ điểm 0, thực hiện các yêu cầu chở khách và hàng, rồi quay về điểm 0.
  - Xe taxi  $k$  có thể vận chuyển cùng lúc 1 hành khách và tối đa  $Q_k$  khối lượng hàng ( $k = 1, 2, \dots, K$ ).
  - Taxi đã đón hành khách thì phải đi đến điểm trả khách đó ngay lập tức.
- $d_{ij}$  là khoảng cách từ điểm  $i$  đến điểm  $j$  ( $i, j = 0, \dots, 2N + 2M$ ).

Yêu cầu: Tính toán phương án vận chuyển sao cho quãng đường di chuyển **dài nhất** của các xe là **ngắn nhất**.

# Giới thiệu bài toán

Hình: Lộ trình đón trả người kết hợp hàng hóa



# Nội dung

- 1 Giới thiệu bài toán
- 2 Mô hình bài toán**
- 3 Giải thuật chính xác
- 4 Giải thuật heuristics
- 5 Thực nghiệm và đánh giá
- 6 Kết luận

## Tham số

- Tập  $2N + 2M + 2K$  điểm:
  - Hành khách  $i$ : điểm đón  $i$  và điểm trả  $i + N + M$  ( $i = 1, 2, \dots, N$ )
  - Gói hàng  $j$ : điểm lấy hàng  $j$  và điểm trả  $j + N + M$  ( $j = N + 1, N + 2, \dots, N + M$ )
  - $2K$  điểm logic  $2N + 2M + 1, 2N + 2M + 2, \dots, 2N + 2M + 2K$  tham chiếu tới điểm xuất phát vật lý 0. Điểm  $2N + 2M + k$  tương ứng là điểm bắt đầu,  $2N + 2M + K + k$  là điểm kết thúc lộ trình xe thứ  $k$  ( $k = 1, 2, \dots, K$ )
- $d_{ij}$ : Khoảng cách từ điểm  $i$  tới điểm  $j$  ( $i, j \in \{1, 2, \dots, 2N + 2M + 2K\}$ )
- $w_i$ : Sự thay đổi khối lượng hàng khi đi tới điểm  $i$  ( $i = 1, 2, \dots, 2N + 2M + 2K$ )

$$w_i = \begin{cases} q_i & \text{nếu } N + 1 \leq i \leq N + M \\ -q_{i-(N+M)} & \text{nếu } 2N + M + 1 \leq i \leq 2N + 2M \\ 0 & \text{ngược lại} \end{cases}$$

- $Q_k$ : Khối lượng hàng tối đa xe thứ  $k$  có thể chở ( $k = 1, 2, \dots, K$ )

## Biến quyết định

- $x_{ij}$ : Biến nhị phân, xác định cung đi từ điểm  $i$  đến điểm  $j$  có xuất hiện trong lộ trình của 1 trong  $k$  xe không ( $i, j \in [1..(2N + 2M + 2K)]$ )

$$x_{ij} = \begin{cases} 1 & \text{nếu cung } (i, j) \text{ có trong lộ trình của 1 xe} \\ 0 & \text{ngược lại} \end{cases} \quad (1)$$

- Tại mỗi điểm  $i$  ( $i \in [1..(2N + 2M + 2K)]$ ):
  - $r_i$ : chỉ số của xe đi qua điểm  $i$  trong lộ trình

$$1 \leq r_i \leq K \quad (2)$$

- $l_i$ : khoảng cách tích lũy của xe đi từ điểm 0 đến điểm  $i$  trong lộ trình

$$0 \leq l_i < \infty \quad (3)$$

- $c_i$ : khối lượng hàng xe  $k$  (đi qua điểm  $i$ ) còn chịu được khi đi tới điểm  $i$

$$0 \leq c_i \leq \max_{1 \leq k \leq K} \{Q_k\} \quad (4)$$



## Ràng buộc (ký hiệu $P = 2N + 2M$ )

- Ràng buộc cân bằng luồng vào ra:

$$\sum_{j=1}^{2N+2M+2K} x_{ij} = 1, \quad \forall i \in [1..(P+K)] \quad (5)$$

$$\sum_{i=1}^{2N+2M+2K} x_{ij} = 1, \quad \forall j \in [1..P] \cup [(P+K+1)..(P+2K)] \quad (6)$$

- Xác định  $r_i$ :

$$r_{2N+2M+k} = k, \quad \forall k = 1, 2, \dots, K \quad (7)$$

$$\begin{cases} r_j - r_i \leq \mu \times (1 - x_{ij}), & \forall j \in [1..P] \cup [(P+K+1)..(P+2K)], \\ r_j - r_i \geq -\mu \times (1 - x_{ij}) & i \in [1..(P+2K)], i \neq j \end{cases} \quad (8)$$

- Xác định  $l_i$ :

$$l_{2N+2M+k} = 0, \quad k \in [1..K] \quad (9)$$

$$\begin{cases} l_j - l_i - d_{ij} \leq \mu \times (1 - x_{ij}), & \forall j \in [1..P] \cup [(P + K + 1)..(P + 2K)], \\ l_j - l_i - d_{ij} \geq -\mu \times (1 - x_{ij}) & i \in [1..(P + 2K)], i \neq j \end{cases} \quad (10)$$

- Xác định  $c_i$ :

$$c_{2N+2M+k} = Q_k, \quad k \in [1..K] \quad (11)$$

$$\begin{cases} c_j - c_i - w_j \leq \mu \times (1 - x_{ij}), & \forall j \in [1..P] \cup [(P + K + 1)..(P + 2K)] \\ c_j - c_i - w_j \geq -\mu \times (1 - x_{ij}) & i \in [1..(P + 2K)], i \neq j \end{cases} \quad (12)$$

- Điểm đón và trả của hành khách  $i$  phải thuộc lộ trình của cùng một xe, tương tự với các gói hàng:

$$r_i = r_{i+N+M}, \quad \forall i \in [1..(N+M)] \quad (13)$$

- Điểm đón khách phải liền trước điểm trả khách:

$$x_{i,(i+N+M)} = 1, \quad \forall i \in [1..N] \quad (14)$$

- Điểm lấy hàng phải ở trước điểm giao hàng:

$$l_i \leq l_{i+N+M}, \quad \forall i \in [(N+1)..(N+M)] \quad (15)$$

- Khối lượng còn lại của xe tại mọi thời điểm không âm (đã thỏa mãn).

- $L$ : Độ dài của lộ trình xe dài nhất trong  $K$  lộ trình
- Ràng buộc xác định lộ trình dài nhất:

$$l_i \leq L, \quad \forall i \in [(P + K + 1)..(P + 2K)] \quad (16)$$

## Hàm mục tiêu

$$L \leftarrow \min \quad (17)$$

# Nội dung

- 1 Giới thiệu bài toán
- 2 Mô hình bài toán
- 3 Giải thuật chính xác**
  - Mô hình MIP
  - Thuật toán nhánh cận
- 4 Giải thuật heuristics
- 5 Thực nghiệm và đánh giá
- 6 Kết luận

- Mô hình MIP đã trình bày ở trên.
- Sử dụng thư viện OR-Tools để triển khai.

## Phương pháp:

- Khởi tạo: Lời giải rỗng, lộ trình mỗi xe chỉ có điểm đầu 0 và điểm cuối 0.
- Trạng thái nút: Đang xét xe thứ  $k$ , đã xây xong lộ trình các xe trước.
- Rẽ nhánh: Thực hiện một trong các thao tác:
  - Chọn 1 khách chưa được phục vụ, thêm liên tiếp điểm đón và điểm trả khách vào cuối lộ trình xe  $k$
  - Chọn 1 gói hàng chưa có xe lấy, thêm điểm lấy hàng vào cuối lộ trình
  - Chọn 1 gói hàng mà xe  $k$  đã lấy nhưng chưa giao, thêm điểm giao hàng vào cuối lộ trình
  - Kết thúc xây lộ trình xe  $k$ , chuyển sang xe  $k + 1$
- Tỉa nhánh: Các nút có lộ trình xe  $k$  dài hơn lời giải tốt nhất hiện tại.

# Nội dung

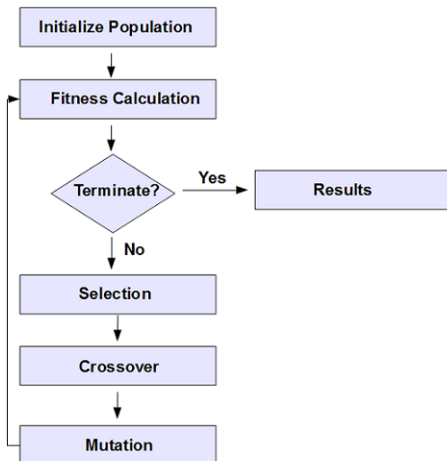
- 1 Giới thiệu bài toán
- 2 Mô hình bài toán
- 3 Giải thuật chính xác
- 4 Giải thuật heuristics**
  - Giải thuật di truyền
  - Thuật toán tham lam
  - Tìm kiếm cục bộ
- 5 Thực nghiệm và đánh giá
- 6 Kết luận



# Giải thuật di truyền

- Sử dụng giải thuật di truyền cho các bộ dữ liệu kích thước lớn
- Tìm ra lời giải tối ưu cục bộ
- Các toán tử lai ghép và đột biến được sử dụng để tạo ra lời giải mới bằng các thay đổi thứ tự của người và hàng hóa trong cùng hành trình của một xe hoặc trên nhiều xe

Hình: Sơ đồ giải thuật di truyền



**Toán tử lai ghép:** Phép lai ghép làm đổi lại thứ tự của các hành khách trong hành trình

- Chọn ra tập cha mẹ bằng phương pháp tournament
- Đối với từng cặp lời giải, xác suất xảy ra lai ghép là  $P_c$
- Xét cặp lời giải bất kì trong tập cha mẹ gồm lời giải  $S_p$  và  $S_q$ :
  - Gọi cặp hành trình  $r_i$  và  $r_j$  ( $r_i \in S_p$  và  $r_j \in S_q$ ) là cặp hành trình tương đồng nếu có số hành khách trùng nhau là lớn nhất
  - Sắp xếp lại thứ tự đón các hành khách chung của  $r_i$  và  $r_j$  trong  $r_i$  theo  $r_j$  ta được lời giải mới

## Toán tử đột biến:

- Di chuyển một món hàng hóa từ route có độ dài lớn nhất sang route có độ dài nhỏ nhất
- Di chuyển Hoán đổi khách hàng giữa 2 route bất kì

## Các bước thực hiện:

- Bước 0: Khởi tạo lời giải rỗng.
- Bước 1: Thêm các yêu cầu chuyển hàng:
  - Duyệt qua các gói hàng chưa xử lý.
  - Duyệt tất cả các cặp vị trí (thuộc cùng 1 xe) có thể chèn điểm lấy và điểm giao hàng (mà không vi phạm ràng buộc).
  - Chọn một trong các cặp vị trí làm lộ trình được chèn có độ dài mới là nhỏ nhất.
- Bước 2: Thêm các yêu cầu chuyển người:
  - Duyệt qua các yêu cầu chờ khách chưa xử lý.
  - Duyệt tất cả các vị trí để chèn cặp điểm đón và điểm trả người (mà không vi phạm ràng buộc).
  - Chọn một trong các vị trí làm lộ trình được chèn có độ dài mới là nhỏ nhất.

**Độ phức tạp:**  $O(\max\{M \times (2M + 2N + K)^3, N \times (2M + 2N + K)^2\})$

## Các bước thực hiện:

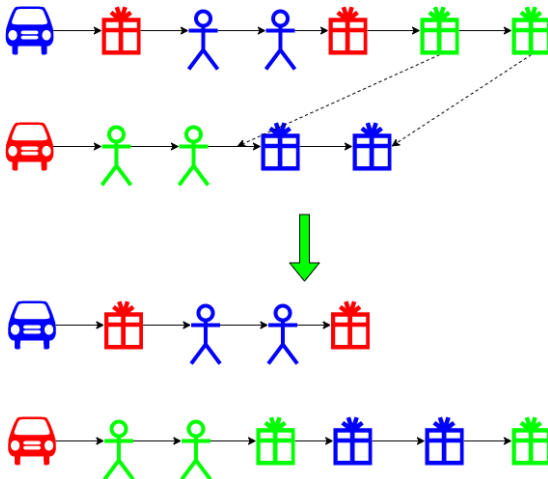
- Khởi tạo: Sử dụng thuật toán tham lam đã trình bày.
- Quá trình tìm kiếm:
  - Tìm một lời giải hàng xóm tốt hơn lời giải hiện tại.
  - Nếu không, tìm lời giải tốt nhất trong các hàng xóm, tạm thay lời giải hiện tại.
  - Nếu đã qua một số vòng lặp mà lời giải hiện tại không cải thiện, khởi tạo lại lời giải.
  - Cập nhật lời giải tốt nhất (nếu có thể).
- Lặp lại quá trình trên tới khi chạm số vòng lặp hoặc thời gian chạy tối đa.

## Xây dựng tập lời giải hàng xóm:

- Move 1: Dịch vị trí điểm lấy và giao hàng:
  - $x_1, x_2$  là điểm lấy và giao cùng một gói hàng
  - Chọn 2 điểm  $y_1, y_2$  thuộc cùng lộ trình
  - Bỏ  $x_1, x_2$  khỏi lộ trình hiện tại, chèn  $x_1$  sau  $y_1$ ,  $x_2$  sau  $y_2$
- Move 2: Dịch vị trí cặp điểm đón trả khách:
  - $x_1, x_2$  là điểm đón và trả cùng một gói hàng
  - Chọn điểm  $y$  bất kỳ trong các lộ trình
  - Bỏ  $x_1, x_2$  khỏi lộ trình hiện tại, chèn  $x_1, x_2$  liên tiếp sau  $y$

# Tìm kiếm cục bộ

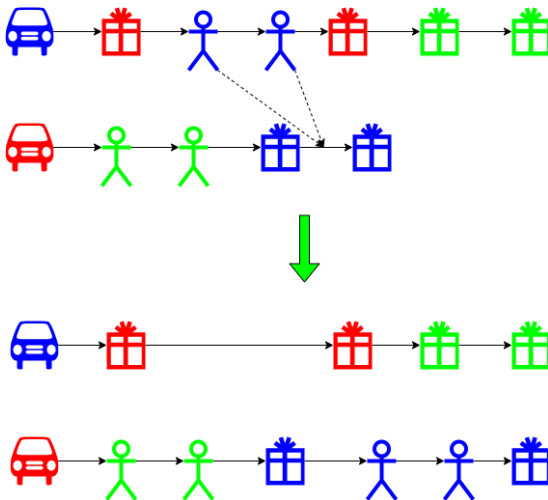
Hình: Move 1: Dịch vị trí gói hàng





# Tìm kiếm cục bộ

Hình: Move 2: Dịch vị trí hành khách



# Nội dung

- 1 Giới thiệu bài toán
- 2 Mô hình bài toán
- 3 Giải thuật chính xác
- 4 Giải thuật heuristics
- 5 Thực nghiệm và đánh giá
  - Tham số thuật toán
  - Kết quả thực nghiệm
- 6 Kết luận

- Số lần chạy heuristics / bộ dữ liệu: 10
- Tìm kiếm cục bộ:
  - Số vòng lặp: 100
  - Số bước cho phép lời giải tồi: 10
- Giải thuật di truyền:
  - Kích thước quần thể khởi tạo: 200
  - Số thế hệ: 500
  - Xác suất lai ghép: 0.5
  - Xác suất đột biến: 0.1

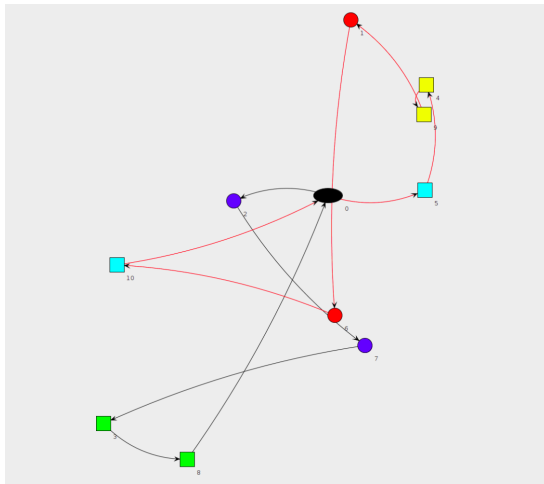
# Kết quả thực nghiệm

## Hình: Thực nghiệm giải thuật chính xác

Bộ dữ liệu	Kích thước		Giải thuật chính xác			
			MIP		Back-tracking	
			f	t(ms)	f	t(ms)
exac_232.txt	N	2	2256.65	760362	2256.65	14
	M	3				
	K	2				
exac_233.txt	N	2	2564.64	1067281	2564.64	7
	M	3				
	K	3				
exac_322.txt	N	3	974.56	141182	974.56	8
	M	2				
	K	2				
exac_433.txt	N	4	TLE	TLE	2208.61	74
	M	3				
	K	3				

# Kết quả thực nghiệm

Hình: Minh họa kết quả giải thuật chính xác



# Kết quả thực nghiệm

Hình: Thực nghiệm giải thuật di truyền

Bộ dữ liệu	Kích thước		GA				
			f_min	f_max	f_avg	std_dev	t_avg (ms)
appr_20305.txt	N	20	8358.53	9808.88	9022.37	434.34	2319.9
	M	30					
	K	5					
appr_30407.txt	N	30	9032.87	10520.07	9973.63	430.23	3044.4
	M	40					
	K	7					
appr_608010.txt	N	60	2589.53	2704.65	2635.07	36.96	5135.4
	M	80					
	K	10					
appr_10012010.txt	N	100	21956.22	22468.28	22265.037	162.71	12246.6
	M	120					
	K	10					
appr_20015020.txt	N	200	18735.04	19475.05	19128.43	283.99	22031
	M	150					
	K	20					
appr_50020050.txt	N	500	3359.31	3755.33	3629.44	101.78	25432.4
	M	200					
	K	50					

# Kết quả thực nghiệm

Hình: Thực nghiệm giải thuật tham lam

Bộ dữ liệu	Kích thước		Greedy				
			f_min	f_max	f_avg	std_dev	t_avg (s)
appr_20305.txt	N	20	5864.24	6041.44	5923.91	49.96	40.4
	M	30					
	K	5					
appr_30407.txt	N	30	6277.56	6875.4	6580.55	183.37	84
	M	40					
	K	7					
appr_608010.txt	N	60	1543.87	1615.34	1577.43	21.94	669.7
	M	80					
	K	10					
appr_10012010.txt	N	100	10469.24	10781.03	10611.85	96.41	5502.1
	M	120					
	K	10					
appr_20015020.txt	N	200	2043.31	2093.13	2073.58	16.82	23443
	M	150					
	K	20					
appr_50020050.txt	N	500	12534.24	12863.59	12687.36	127.06	1234.4
	M	200					
	K	50					

# Kết quả thực nghiệm

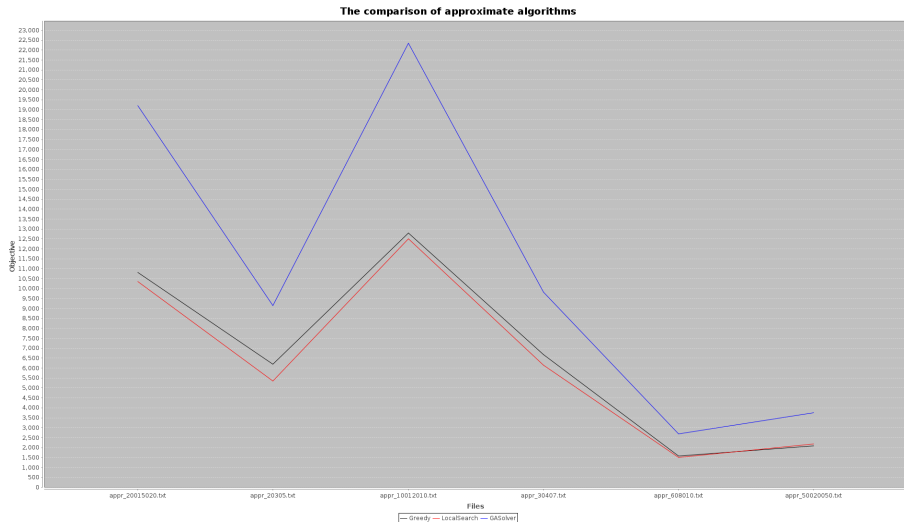
Hình: Thực nghiệm giải thuật tìm kiếm cục bộ

Bộ dữ liệu	Kích thước		Local search				
			f_min	f_max	f_avg	std_dev	t_avg (s)
appr_20305.txt	N	20	5434.74	5744.79	5586.46	107.51	17280.6
	M	30					
	K	5					
appr_30407.txt	N	30	5646.01	6140.23	5901.66	167.04	56135.8
	M	40					
	K	7					
appr_608010.txt	N	60	1463.55	1550.56	1518.01	29.23	253286.6
	M	80					
	K	10					
appr_10012010.txt	N	100	12069.64	12488	12256.41	126.7	315169.1
	M	120					
	K	10					
appr_20015020.txt	N	200	9856.73	10289.45	10125.2	132.77	328774.1
	M	150					
	K	20					
appr_50020050.txt	N	500	2093.84	2192.67	2135.75	31.91	385086.5
	M	200					
	K	50					



# Kết quả thực nghiệm

Hình: So sánh các giải thuật xấp xỉ



# Nội dung

- 1 Giới thiệu bài toán
- 2 Mô hình bài toán
- 3 Giải thuật chính xác
- 4 Giải thuật heuristics
- 5 Thực nghiệm và đánh giá
- 6 Kết luận**

- Mô hình MIP đã đề xuất cho kết quả chính xác, tuy nhiên thời gian chạy lớn hơn rất nhiều so với giải thuật nhánh cận.
- Đối với các bộ dữ liệu kích thước lớn, thuật toán tìm kiếm cục bộ cho hiệu quả tốt nhất