**ĐẠI HỌC BÁCH KHOA HÀ NỘI**

**TRƯỜNG CÔNG NGHỆ THÔNG TIN & TRUYỀN THÔNG**

**---□**&**□---**



**BÁO CÁO BÀI TẬP LỚN**

**Tối ưu lập kế hoạch**

**Đề tài: Taxi vận chuyển người kết hợp hàng hoá**

***Giảng viên hướng dẫn*:** **TS**. Bùi Quốc Trung

***Sinh viên thực hiện:***

| Đào Nguyễn Tiến Huy | - | 20194077 |
| --- | --- | --- |
| Lê Doãn Biên | - | 20193992 |
| Nguyễn Minh Hiển | - | 20204045 |
| Lê Tiến Dũng | - | 20194027 |

Mục lục

[Giới thiệu bài toán 2](#_heading=h.gjdgxs)

[Giới thiệu 2](#_heading=h.1fob9te)

[Mô tả bài toán 2](#_heading=h.3znysh7)

[Các giải thuật đề xuất 3](#_heading=h.2et92p0)

[Thuật toán chính xác 3](#_heading=h.tyjcwt)

[Giải thuật Localsearch 3](#_heading=h.3dy6vkm)

[Mô hình hoá bài toán 3](#_heading=h.1t3h5sf)

[Cho thuật toán IP và CP 3](#_heading=h.4d34og8)

[Kết quả thực nghiệm và đánh giá 6](#_heading=h.2s8eyo1)

[Các testcase sử dụng 6](#_heading=h.17dp8vu)

[Kết quả 6](#_heading=h.3rdcrjn)

[Kết luận 9](#_heading=h.26in1rg)

[Nhóm thuật toán chính xác 9](#_heading=h.lnxbz9)

[Thuật toán local search 9](#_heading=h.35nkun2)

[Phụ lục 10](#_heading=h.1ksv4uv)

[Source code 10](#_heading=h.44sinio)

[Phân công công việc 10](#_heading=h.2jxsxqh)

[Tài liệu tham khảo 11](#_heading=h.z337ya)

# Giới thiệu bài toán

## Giới thiệu

Bài toán Taxi vận chuyển người kết hợp hàng hoá là một bài toán trong phân lớp CVRP, **Share-a-Ride Problem** (SARP) là một bài toán rất được chú ý hiện nay.

## Mô tả bài toán

* Có N hành khách 1, 2, …, N và M gói hàng N+1,N+ 2, …, N+M. Hành khách (người) (hoặc gói hàng) i có điểm đón là i và điểm trả là i+ N + M (i = 1,2,…,2N+2M). • Mỗi gói hàng i có khối lượng qi (i=N+1,…,N+M)
* Có K taxi 1,…,K xuất phát từ điểm 0. Mỗi xe taxi k có thể vận chuyển cùng 1 lúc 1 hành khách và tối đa Qk khối lượng hàng (xếp vào cốp xe) (k=1,…,K)
* Biết rằng d(i,j) là khoảng cách từ điểm i đến điểm j (i,j=0,…,2N+2M)
* Hãy tính toán phương án vận chuyển sao cho quãng đường di chuyển của xe đi dài nhất là ngắn nhất có thể được (để đảm bảo tính công bằng giữa quãng đường các xe cần phải di chuyển)

# Các giải thuật đề xuất

## Thuật toán chính xác

### Backtracking

Giải thuật quay lui có cắt tỉa nhánh cận

### Interger programing

Thuật toán giải bài toán quy hoạch nguyên tuyến tính thư viện ortools

### Constraint programing

Thuật toán quy hoạch ràng buộc thư viện ortools

## Giải thuật Localsearch

### GREEDY\_DESCENT:

Giải thuật tìm kiếm leo đồi, kết thúc khi tìm được nghiệm cục bộ địa phương

### SIMULATED\_ANNEALING:

Giải thuật tìm kiếm và có phương pháp thoát khỏi nghiệm cục bộ bằng xác suất

### GUIDED\_LOCAL\_SEARCH:

Giải thuật metaheuristic, được đánh giá là một trong những giải thuật hữu hiệu nhất trong các bài toán định tuyến

### TABU\_SEARCH:

Giải thuật metaheuristic sử dụng local search và cơ chế lưu lại những đường đã đi bằng bảng “tabu”

# Mô hình hoá bài toán

## Cho thuật toán IP và CP

* Hằng:

N : Số điểm đón khách

M: Số điểm chở hàng

K: Số xe

: Tổng số đỉnh của đồ thị

: Số lượng khách tại đỉnh

: Khối lượng hàng tại đỉnh

: Tải trọng của xe thứ

: Khoảng cách giữa 2 đỉnh

root: Đỉnh gốc

root = 0

* Biến:

: Tải trọng (hàng hoá) của xe k tại đỉnh i

: Tải trọng (khách) của xe k tại đỉnh i

: Thứ tự thăm của đỉnh I đối với xe k

* Ràng buộc:
* Tổng số cạnh của đồ thị:

=

* Mọi xe đều xuất phát từ root và kết thúc tại root

* Mỗi đỉnh khác root chỉ được thăm 1 lần

=1

=1

* Xe không thể đi từ đỉnh i đến đỉnh i

với

* Xe đi vào đỉnh i thì đi ra từ i
* MTZ constraint

Với ; ; -

Với

* Tổng số hàng của xe k khi thăm điểm i

* Tổng số khách của xe k khi thăm điểm i



y = max()

* Hàm mục tiêu:

min(y)

# Kết quả thực nghiệm và đánh giá

## Các testcase sử dụng

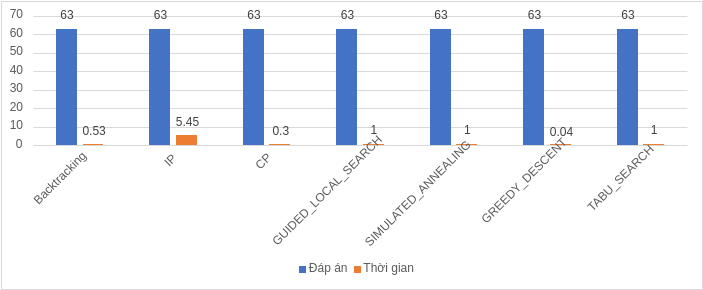
* Testcase 1: 11 đỉnh, 3 xe (5 điểm đón, 5 điểm trả, 1 Điểm gốc)
* Testcase 2: 15 đỉnh, 3 xe (7 điểm đón, 7 điểm trả, 1 Điểm gốc)
* Testcase 3: 21 đỉnh, 3 xe (10 điểm đón, 10 điểm trả, 1 Điểm gốc)
* Testcase 4: 51 đỉnh, 3 xe (10 điểm đón, 10 điểm trả, 1 Điểm gốc)

## Kết quả

### Testcase 1:

|  | test1 | | test2 | | test3 | | test4 | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Thuật toán | solution | time | solution | time | solution | time | solution | time |
| Backtracking | 63 | 0.53 | 67 | 0.59 | 66 | 0.91 | 61 | 0.74 |
| IP | 63 | 5.45 | 67 | 7.75 | 66 | 4.05 | 61 | 7.86 |
| CP | 63 | 0.3 | 67 | 0.44 | 66 | 0.39 | 61 | 0.5 |
| GUIDED\_LOCAL\_SEARCH | 63 | 1 | 67 | 1 | 66 | 1 | 61 | 1 |
| SIMULATED\_ANNEALING | 63 | 1 | 67 | 1 | 66 | 1 | 61 | 1 |
| GREEDY\_DESCENT | 63 | 0.04 | 67 | 0.07 | 67 | 0.07 | 66 | 0.06 |
| TABU\_SEARCH | 63 | 1 | 67 | 1 | 66 | 1 | 61 | 1 |

Kết quả chạy testcase1

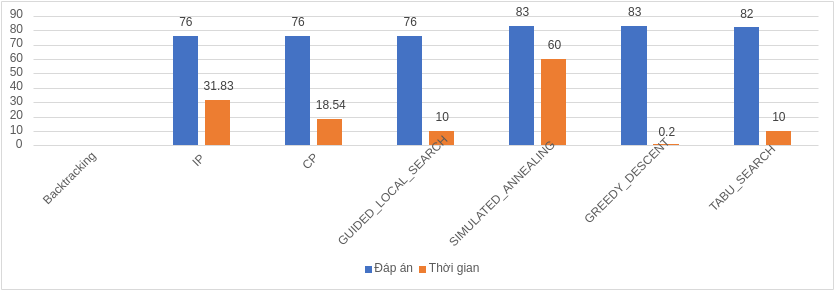


So sánh kết quả các thuật toán ở test 1

### Testcase 2:

|  | test1 | | test2 | | test3 | | test4 | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Thuật toán | solution | time | solution | time | solution | time | solution | time |
| Backtracking |  |  |  |  |  |  |  |  |
| IP | 76 | 31.83 | 77 | 152.49 | 72 | 192.11 | 77 | 504.88 |
| CP | 76 | 18.54 | 77 | 46.93 | 72 | 64.05 | 77 | 57.33 |
| GUIDED\_LOCAL\_SEARCH | 76 | 10 | 77 | 10 | 72 | 10 | 77 | 10 |
| SIMULATED\_ANNEALING | 83 | 60 | 79 | 60 | 72 | 60 | 79 | 60 |
| GREEDY\_DESCENT | 83 | 0.2 | 79 | 0.15 | 72 | 0.17 | 79 | 0.15 |
| TABU\_SEARCH | 82 | 10 | 77 | 10 | 72 | 10 | 77 | 10 |

Kết quả chạy testcase 2



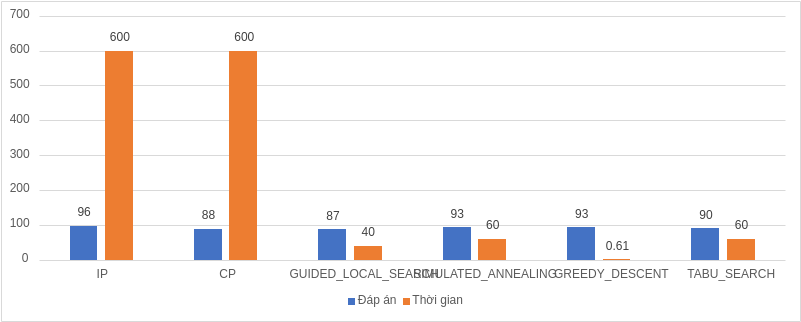
So sánh kết quả chạy test 1

### Testcase 3:

Do số lượng đỉnh là 21 , giới hạn thời gian chạy của thuật toán IP và CP là 10 p

|  | test1 | | test2 | | test3 | |
| --- | --- | --- | --- | --- | --- | --- |
| Thuật toán | solution | time | solution | time | solution | time |
| Backtracking |  |  |  |  |  |  |
| IP | 96 | 600 | 85 | 600 | 93 | 600 |
| CP | 88 | 600 | 85 | 600 | 91 | 600 |
| GUIDED\_LOCAL\_SEARCH | 87 | 40 | 85 | 40 | 91 | 40 |
| SIMULATED\_ANNEALING | 93 | 60 | 86 | 60 | 92 | 60 |
| GREEDY\_DESCENT | 93 | 0.61 | 86 | 0.67 | 92 | 0.7 |
| TABU\_SEARCH | 90 | 60 | 85 | 60 | 92 | 60 |

Kết quả chạy testcase 3

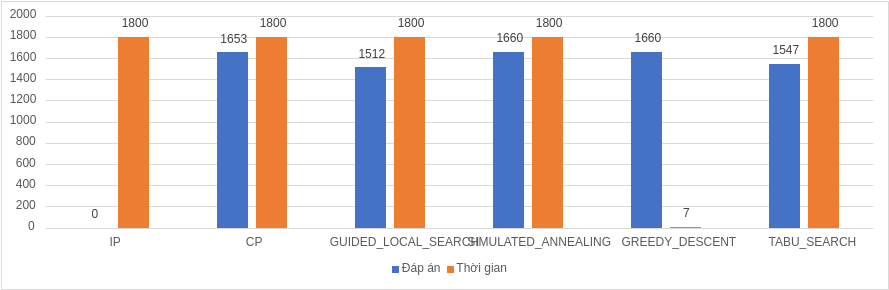


So sánh kết quả chạy test 1

### Testcase 4

|  | test1 | | test2 | | test3 | |
| --- | --- | --- | --- | --- | --- | --- |
| Thuật toán | solution | time | solution | time | solution | time |
| Backtracking |  |  |  |  |  |  |
| IP |  |  |  |  |  |  |
| CP | 1607 | 1800 | 1723 | 301800 | 1632 | 1800 |
| CP | 1653 | 1800 | 1670 | 1800 | 1589 | 2400 |
| GUIDED\_LOCAL\_SEARCH | 1512 | 1800 | 1518 | 1800 | 1484 | 1800 |
| SIMULATED\_ANNEALING | 1660 | 1800 | 1607 | 1800 | 1601 | 1800 |
| GREEDY\_DESCENT | 1660 | 7 | 1611 | 6.3 | 1632 | 8.8 |
| TABU\_SEARCH | 1547 | 1800 | 1535 | 1800 | 1507 | 1800 |

Kết quả chạy testcase 4



So sánh kết quả chạy test 1

# Kết luận

## Nhóm thuật toán chính xác

Thuật toán backtracking ra kết quả nhanh đối với test nhỏ, tuy nhiên không chạy được với các test lớn

Thuật toán CP có thời gian chạy nhanh hơn thuật toán IP

Thuật toán CP có thể đặt giới hạn thời gian và chạy với test lớn (testcase4) cho ra local optimizer còn thuật toán IP không cho ra kết quả nào

## Thuật toán local search

Nhận xét:

Thuật toán GREEDY\_DESCENT nhanh nhưng kết quả không tốt

Thuật toán SIMULATED\_ANNEALING có thời gian chạy lâu hơn nhưng kết quả không chênh lệch nhiều so với GREEDY\_DESCENT

Thuật toán TABU\_SEARCH tối ưu hơn hai thuật toán trên

Thuật toán GUIDED\_LOCAL\_SEARCH chạy cùng thời gian cho đáp án tốt nhất

# Phụ lục

## Source code

[dunga1k58bh/TULKH-20212 (github.com)](https://github.com/dunga1k58bh/TULKH-20212)

## Phân công công việc

| **Họ tên** | **Công việc** | **Đánh giá** |
| --- | --- | --- |
| Đào Nguyễn Tiến Huy | Đọc paper, mô hình hoá, code nhóm thuật toán localsearch. Tham gia thực nghiệm mô hình | Tốt |
| Lê Doãn Biên | Đọc paper, mô hình hoá, code thuật toán CP, Tham gia thực nghiệm mô hình | Tốt |
| Nguyễn Minh Hiển | Đọc paper, mô hình hoá, code thuật toán backtracking, generate testcase, Tham gia thực nghiệm mô hình | Tốt |
| Lê Tiến Dũng | Đọc paper, mô hình hoá, code thuật toán IP, Tham gia thực nghiệm mô hình | Tốt |

# Tài liệu tham khảo

[1] TS. Bùi Quốc Trung, *Slide bài giảng môn Tối ưu lập kế hoạch*

[2] [OR-Tools  |  Google Developers](https://developers.google.com/optimization)