Trường Đại học Bách Khoa Hà Nội

Trường Công nghệ thông tin và Truyền thông



**Báo cáo bài tập lớn**

Môn: Tối ưu lập kế hoạch

Đề tài: Vận chuyển hành khách bằng xe bus

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

**Sinh viên thực hiện: Nhóm 4**

|  |  |
| --- | --- |
| Nguyễn Thành Đạt | 20190041 |
| Trương Quang Phú | 20194138 |
| Nguyễn Trí Trung Nguyên | 20194133 |
| Vũ Quang Huy | 20183558 |

Hà Nội, 8-2022

Mục lục

[LỜI CẢM ƠN 4](#_Toc110546787)

[I. GIỚI THIỆU BÀI TOÁN: 5](#_Toc110546788)

[II. Mô hình: 5](#_Toc110546789)

[1. Mô hình MIP: 5](#_Toc110546790)

[a. Biến: 5](#_Toc110546791)

[b. Ràng buộc: 5](#_Toc110546792)

[c. Hàm mục tiêu: 6](#_Toc110546793)

[2. Mô hình CP 6](#_Toc110546794)

[a. Các thành phần quan trọng 6](#_Toc110546795)

[b. Các biến quyết định: 6](#_Toc110546796)

[c. Các ràng buộc: 6](#_Toc110546797)

[d. Hàm mục tiêu 7](#_Toc110546798)

[3. Thuật toán Heuristics 7](#_Toc110546799)

[a. Thuật Toán tham lam 1 7](#_Toc110546800)

[b. Thuật Toán tham lam 2 9](#_Toc110546801)

[4. Back Tracking 10](#_Toc110546802)

[a. Ý tưởng thuật toán 10](#_Toc110546803)

[b. Thuật toán back tracking 11](#_Toc110546804)

[III. Dữ liệu 11](#_Toc110546805)

[1. Mô tả dữ liệu: 11](#_Toc110546806)

[2. Sinh dữ liệu: 12](#_Toc110546807)

[IV. Kết quả và đánh giá 12](#_Toc110546808)

# LỜI CẢM ƠN

Nhóm em xin được dành những lời cảm ơn chân thành tới TS. Bùi Quốc Trung - giảng viên Bộ môn Khoa học Máy tính, Trường Công nghệ Thông tin và Truyền thông, Đại học Bách khoa Hà Nội. Nhờ sự tận tình chỉ dạy, hướng dẫn của thầy, nhóm em đã hoàn thành tốt mini project môn học Tối ưu và Lập kế hoạch này.

Tuy nhiên vì kiến thức bản thân còn hạn chế, trong quá trình học tập và hoàn thiện mini project này, nhóm em không tránh khỏi những sai sót. Kính mong nhận được những ý kiến đóng góp từ thầy để giúp nhóm em hoàn thiện hơn. Nhóm em xin chân thành cảm ơn.

# I. GIỚI THIỆU BÀI TOÁN:

Có N hành khách {1, 2, 3,..., N} khách hàng thứ i muốn đi từ bến i đến bến i+N. Có K xe bus {1, 2, 3,..., K}. Mỗi xe xuất phát từ 0 chở khách và quay lại điểm 0. Xe k thì chỉ có thể vận chuyển cùng lúc tối đa khách.

là khoảng cách giữa 2 điểm i và j (i,j {1,2,3,.. 2N}).

Lập phương án vận chuyển sao cho tổng quãng đường của k xe là nhỏ nhất.

Đây là một bài toán thuộc lớp bài toán Vehicle Routing: Pickup and delivery, nhóm 4 chúng em đưa ra những giải thuật sau: MIP, CP, Heuristic (Greedy Algorithm) và Back Tracking.

# II. MÔ HÌNH:

## 1. Mô hình MIP:

### a. Biến:

x[i][j][k]= {0, 1} bằng 1 nếu xe k có đi từ i đến j, 0 thì ngược lại

y[i][k] = t: Chi phí quãng đường đi từ 0 đến i của xe k

sl[i][k] = h: số lượng khách trên xe k khi đến i là h

### b. Ràng buộc:

- M là một số nguyên cực lớn

- Mối quan hệ giữa x và y:

Nếu xe k có đi từ i đến j thì y[j][k] = y[i][k] + d[i][j]

y[i][k]+d[i][j]-y[j][k]+M(1-x[i][j][k])≥0

y[i][k]+d[i][j]-y[j][k]-M(1-x[i][j][k])≤0

- Liên hệ giữa s và x:

+ Với j<=n: Nếu xe k có đi từ i đến j mà j<=n thì sl[j][k]=sl[i][k]+1

sl[i][k]+1-sl[j][k]+M(1-x[i][j][k])≥0

sl[i][k]+1-sl[j][k]-M(1-x[i][j][k])≤0

+ Với j>n: Nếu xe k có đi từ i đến j mà j<=n thì sl[j][k]=sl[i][k]-1

sl[i][k]-1-sl[j][k]+M(1-x[i][j][k])≥0

sl[i][k]-1-sl[j][k]+M(1-x[i][j][k])≤0

- Mỗi xe phải xuất phát từ 0 và trở về 0:

x[0][j][k]=x[i][0][k]=1 ∀  i≠j, i,j∈(0,2n)  k∈(0,K)

- Mỗi điểm chỉ được đi qua bởi 1 xe và chỉ có một đầu vào và đầu ra tại i:

- Điểm i và i+n phải được đi bởi 1 xe:

- Đi đến i trước i+n:

### c. Hàm mục tiêu:

## 2. Mô hình CP

### a. Các thành phần quan trọng

Với ta có tập các cặp điểm đi - điểm đến:

Các cặp điểm đón – 0:

Các cặp 0 – điểm trả:

Các cặp điểm trả - điểm đón:

Các cặp điểm đi – điểm đến trùng nhau:

=> Các cặp điểm hợp lệ:

Ta có:

### b. Các biến quyết định:

x[k][i][ j]: bằng 1 nếu xe bus k đi từ i đến j và ngược lại bằng 0

y[i]: sức chứa của xe bus qua i sau khi đến i

z[i]: chỉ số của xe bus đến i

b[i][j] : bằng 1 nếu điểm i được đến trước j và ngược lại bằng 0

### c. Các ràng buộc:

- Xe đi vào điểm i cũng là xe đi ra:

- Điểm nào cũng chỉ được đến bởi 1 xe:

- Mỗi xe phải xuất phát từ 0 và trở về 0:

- Nếu xe đi từ i đến j: lượng người trên xe +1 nếu j là điểm đón và -1 nếu là điểm trả:

- Nếu xe k đi từ i đến j thì chỉ số xe của điểm i và j là k:

- Xe đi qua điểm i và i+N đều là 1 xe:

- Nếu xe k đi từ i đến j thì lượng trên xe tại điểm i và j không được quá sức chứa của xe:

- Xe phải đến điểm i trước i+N:

- Nếu xe đi từ i đến j thì tất cả các điểm u khác i, j đều được đến trước i và j hoặc sau i và j:

### d. Hàm mục tiêu

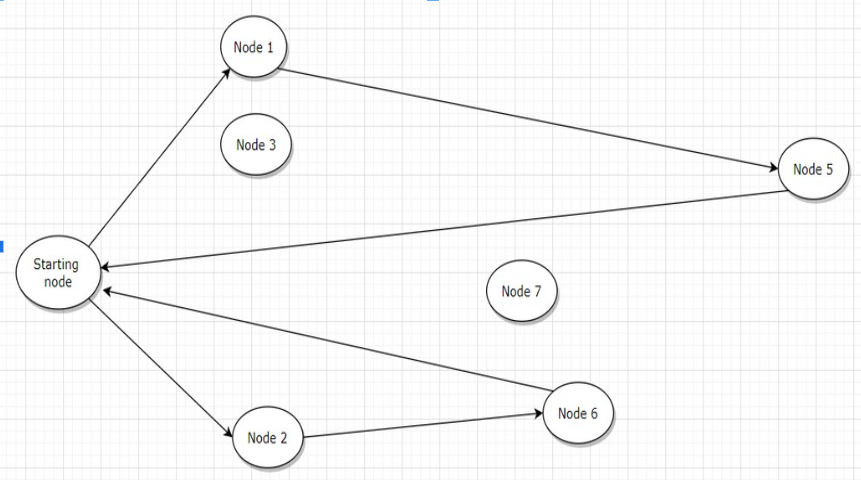
:

## 3. Thuật toán Heuristics

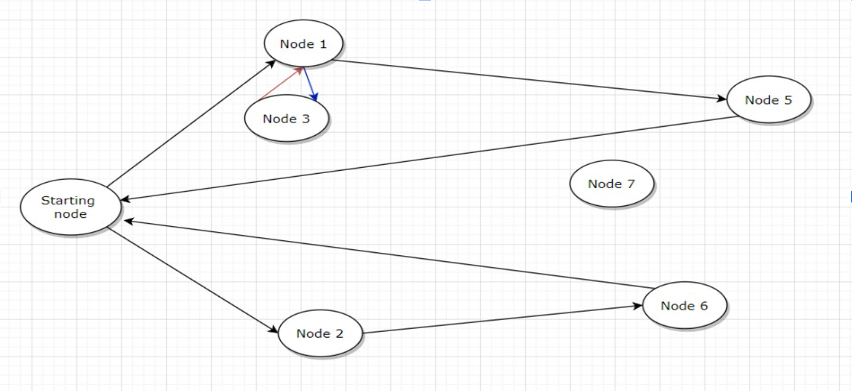
Sử dụng thuật toán tham lam để giải quyết bài toán, tối thiếu hóa tổng quãng đường và tối đa hóa điểm xe buýt đón và trả khách.

### a. Thuật Toán tham lam 1

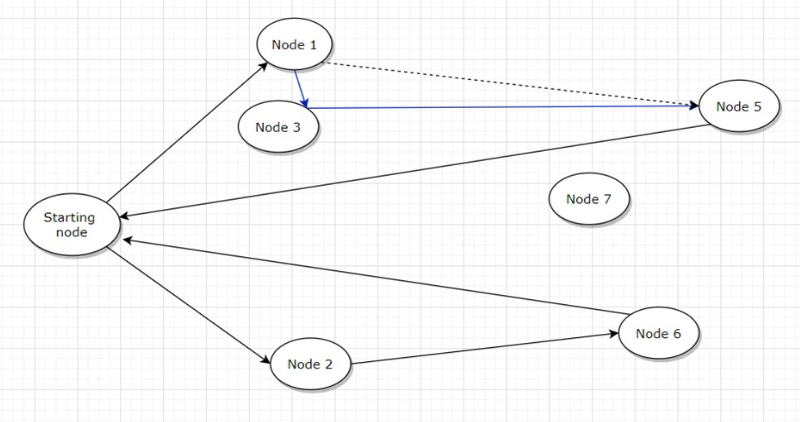
Bước 1: Ta xét tất cả các tuyến (0 à i à(i+N) à0) với 1iN. Sau đó, chọn K tuyến đường có độ dài nhỏ nhất để làm tuyến đường ban đầu.



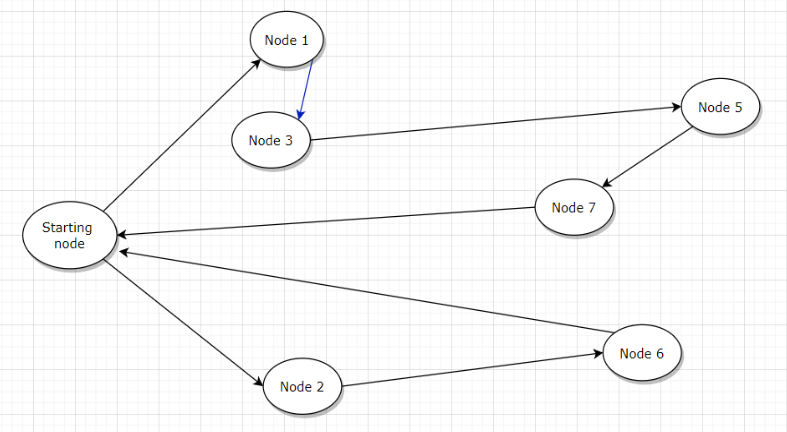
Bước 2: Ta tính tổng khoảng cách từ điểm đón khách khác đến từng điểm K đón khách ta đã chọn ở trên. Tổng đó càng nhỏ thì sức chứa của xe bus mà ta sử dụng cho tuyến đường đó càng cao.



Bước 3: Xét điểm i từ điểm 1 🡪 N nếu i chưa ở trong tuyến đường nào. Ta tìm thấy điểm C đã có trong lộ trình là điểm gần nhất với điểm đang xét và thử đặt điểm i ngay sau hoặc ngay trước điểm C trong lộ trình của nó. Nếu sau khi đặt nó vào vị trí đó mà điều kiện công suất vẫn thỏa mãn thì ta chọn vị trí đó cho i. Nếu không thỏa mãn, chúng ta tiếp tục xem xét điểm tiếp theo gần nhất với điểm i.



Bước 4: Sau khi kết thúc bước 3, ta đã biết điểm i thuộc lộ trình nào. Bây giờ, ta sẽ tìm vị trí điểm (i+N) trong lộ trình đó. Việc chèn một điểm trả khách vào tuyến đường sẽ không vi phạm điều kiện sức chứa, miễn là xe đã qua điểm i. Vì vậy, ta sẽ chỉ tìm điểm gần i+N nhất (được đi qua sau điểm i) trong tuyến đường đó và đặt i + N xung quanh nó.



Bước 5: Quay lại bước 3 làm tương tự.

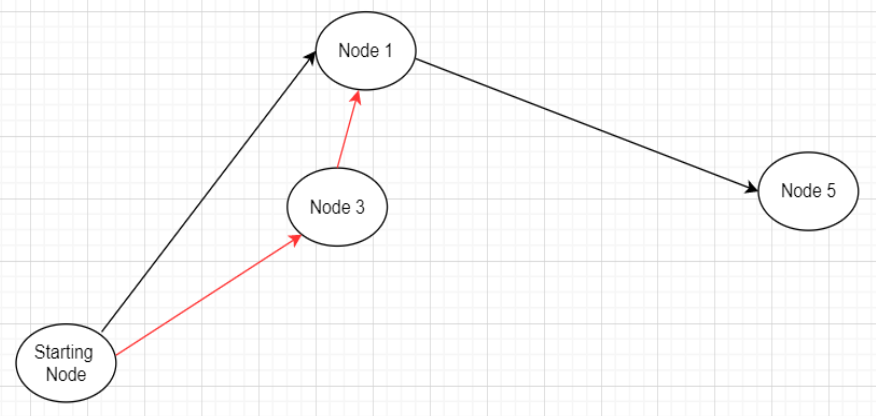
### b. Thuật Toán tham lam 2

Hai bước đầu chúng ta làm tương tự như 2 bước đầu của Thuật toán Tham Lam 1.

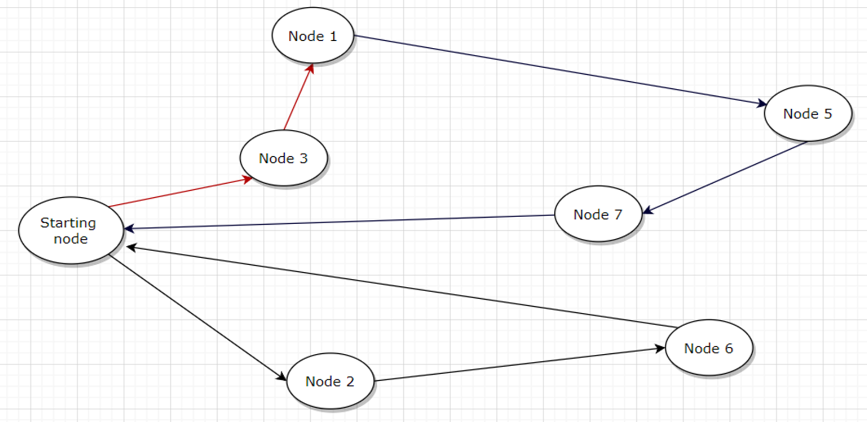
Bước 1: Ta xét tất cả các tuyến (0 à i à(i+N) à0) với 1 i N. Sau đó, chọn K tuyến đường có độ dài nhỏ nhất để làm tuyến đường ban đầu.

Bước 2: Ta tính tổng khoảng cách từ điểm đón khách khác đến từng điểm K đón khách ta đã chọn ở trên. Tổng đó càng nhỏ thì sức chứa của xe bus mà ta sử dụng cho tuyến đường đó càng cao.

Bước 3: Cho điểm t từ 1 tới 2N. Nếu t vẫn chưa thuộc tuyến nào thì ta xét tất cả các cặp điểm LIÊN TỤC (điểm m và n) sao cho d(m,t) + d(t,n) là nhỏ nhất và vẫn đồng thời thỏa mãn điều kiện dung lượng khi thêm điểm t vào vị trí đó.



Bước 4: Sau khi kết thúc bước 3, ta đưa ra điểm t đã nằm trong một lộ trình. Tiếp theo, ta sẽ xét tương tự đối với vị trí điểm (t+N) trong lộ trình đó, nhưng chỉ xét các cặp điểm LIÊN TỤC sau điểm t trong lộ trình đó.



Bước 5: Quay lại bước 3 làm tươg tự.

## 4. Back Tracking

### a. Ý tưởng thuật toán

B1: Chia N người vào K xe

B2: Với mỗi xe sẽ thực hiện thuật toán như bài toán TSP

### b. Thuật toán back tracking

TRY\_Y(a): chia N người vào K xe

Y = []: số khách mà xe thứ k cần phục vụ

sum = 0: tổng số người đã được chia vào xe

TRY\_Y(a):

for i in (1, N-K):

Y[a] = i

sum += i

if a =k and sum = N:

TRY\_X(1)

else:

TRY\_Y(a+1)

sum -= i

X = []: các điểm đã đi qua

d\_min=99999999: tổng quãng đường nhỏ nhất

distance = 0: tổng quãng đường đã đi

P= []: số lượng hành khách ở trên xe i

T= []: số lượng hành khách mà xe i đã đón

Z= []: số lượng hành khách mà xe i đã trả

V=[false, .., false]: mảng đánh dấu các điểm đã đi qua

c = 1: xe hiện tại đang xét

TRY\_X(a):

for i in (0, 2N):

if check(a, i) = true:

X[a] = i

cập nhật distance, P, Z, T, V

if a == 2N+K:

Ghi nhận cấu hình, cập nhật d\_min

else:

TRY\_X(a+1)

khôi phục distance, P, Z, T, V

main():

TRY\_Y(1)

Nhánh và cận:

c\_min: khoảng cách nhỏ nhất giữa hai điểm bất kỳ

Tại a = j trong TRY\_X:

cận dưới: distance + (2N+K-j)\*c\_min < d\_min

# III. Dữ liệu

## 1. Mô tả dữ liệu:

Dòng 1: N và K

Dòng 2: , ,…,

Dòng thứ 3+i (i= 0,..., 2N): ghi dòng thứ i của ma trận khoảng cách

## 2. Sinh dữ liệu:

Sinh ngẫu nhiên các điểm trong mặt phẳng từ (-100, 100), sau đó tính khoảng các giữa 2 cặp điểm.

# IV. Kết quả và đánh giá

1. Với N 8. Không giới hạn thời gian chạy

Table

Description automatically generated

Nhận xét

- Với MIP, CP, BackTranking: cho kết quả tối ưu, với N nhỏ, back tracking làm việc rất tốt

- Với Heuricstic: cho kết quả tối ưu địa phương

2. Với N 8, giới hạn thời gian chạy 30 phút

Table

Description automatically generated

Nhận xét:

- Với MIP, CP, Back Tracking: bị limit thời gian

- Với Heurictic: thuật toán chạy rất nhanh, với những bộ dữ liệu lớn, Heuristic cho ra kết quả tốt hơn MIP, CP, Back tracking sau 30 phút.