

Report

Balanced Staff Routing for Maintenance

I. Problem description

There exist N customers where customer i resides in location i ($i = 1, 2, 3, \dots, N$) that demand Internet maintenance service. The maintenance service of the customer i lasts for $d[i]$ (time unit). There are K employees departing from one single depot (denoted by 0). The traveling time between two locations i and j ($i, j = 0, 1, \dots, N$) is $t[i, j]$. The objective is to schedule a plan in which the maximum working time (equals to the traveling time plus the maintenance time) among the employees is minimized.

II. Modelling

- Denote:
 - $\mathbf{D} = [d[1], d[2], \dots, d[N]]$ is a vector containing the maintenance time of every customer
 - $\mathbf{T} = [T[i, j]]$ for i, j in $\{0, 1, \dots, N\}$ is a 2D-array that represent the traveling time between location i and location j .
- Decision variable:
 - $\mathbf{x}[i, k] = 1$, nếu khách hàng i có trong lịch trình bảo trì của nhân viên k ; $\mathbf{x}[i, k] = 0$, nếu ngược lại.
(với $1 \leq i \leq N$; $1 \leq k \leq K$)
 - $\mathbf{y}[i, j, k] = 1$, nếu quãng đường di chuyển từ i đến j có trong lịch trình bảo trì của nhân viên k ; $\mathbf{y}[i, j, k] = 0$, nếu ngược lại.
(với $0 \leq i, j \leq N$; $1 \leq k \leq K$)
 - $\mathbf{c}[k]$: Thời gian làm việc của nhân viên k
(với $1 \leq k \leq K$)
- Constraints
 - Ràng buộc giữa x, y, c :

$$c[k] = \sum_{i=1}^N x[i, k]d[i] + \sum_{i=0}^N \sum_{j=0}^N y[i, j, k]t[i, j]$$

- K nhân viên đều phải xuất phát từ trụ sở 0:

$$\sum_{j=1}^N y[0, j, k] = K \quad (1 \leq k \leq K)$$

- K nhân viên đều phải quay về trụ sở 0:

$$\sum_{j=1}^N y[j, 0, k] = K \quad (1 \leq k \leq K)$$

- Đảm bảo có đúng 1 cạnh vào nốt i và 1 cạnh ra khỏi nốt i trong một lịch trình:

$$\sum_{j=1}^N y[i, j, k] = \sum_{j=1}^N y[j, i, k] = x[i, k] \quad (1 \leq i \leq N; 1 \leq k \leq K)$$

- Mỗi khách hàng chỉ được bảo trì bởi đúng 1 nhân viên:

$$\sum_{k=1}^K x[i, k] = 1 \quad (1 \leq i \leq N)$$

- Loại bỏ các hành trình con không hợp lệ:

$$\sum_{(i,j) \in S} y[i, j, k] \leq |S| - 1, \quad (1 \leq k \leq K; \forall S \subseteq \{1, 2, \dots, N\}; S \neq \emptyset)$$

• Objective function:

$$\max\{c[k]\} \rightarrow \min \quad (1 \leq k \leq K)$$

III. Proposed Algorithms

Algorithm 1: Integer Linear Programming - Branch and Cut (Quyên)

Algorithm 2: Greedy (Quý)

Algorithm 3: Local Search (Minh)

Algorithm 4: CP-SAT (Quý)

IV. Experiments

V. Conclusion

Overall, all the aforementioned algorithms could give rather acceptable or even optimal solutions. But the performance is different between these algorithms.

- + The first algorithm gives feasible solutions in the shortest time and of quite good quality. The bias of this algorithm is that the convergence rate of this algorithm depends much on the random seed_customer, and the failure chance gets higher when the ratio of N / K gets smaller.
- + The second algorithm
- + The third algorithm
- + The fourth algorithm