Solution Nhóm 7 - Bài toán Công ty

Hướng tiếp cận & Thuật toán

Hướng tiếp cận

- Chỉ số phức tạp công việc của mỗi nhân viên u, với $deg(u)$ là số cấp dưới trực tiếp của u:

$$C_u = w_u \ + \ deg(u)$$

- Nếu ta sa thải x với cha là p thì, chỉ có duy nhất chỉ số phức tạp của p bị thay đổi, C_p tăng lên một lượng bằng:

$$\Delta_p(x) = (w_x + deg(x) - 1)$$

Vì vậy , nếu ta đang đứng ở một nút u, khi sa thải một cấp dưới v của u thì u phải gánh chịu một chi phí là: $cost(v) = w_v + deg(v) - 1$

Gọi khả năng gánh chịu chỉ số phức tạp của một nhân viên u là $S_u = m - (w_u + deg(u))$. Ta chỉ được sa thải một tập con các con trực tiếp hiện tại của u sao cho tổng $cost \leq S_u$.

Chi tiết thuật toán

Ta duyệt DFS theo thứ tự PostOrder. Với mỗi nút u:

- Gọi đệ quy xử lý xong mọi cây con của u.
- Tạo một tập ứng viên là toàn bộ con v trực tiếp hiện tại của u và chi phí mà u phải gánh chịu khi mà sa thải v, ta dùng CTDL priority_queue ưu tiên theo chi phí tăng dần (min-heap).
- Trong khi min-heap không rỗng thì ta xem xét nút v có chi phí sa thải nhỏ nhất, nếu $cost_v \leq S_u$ thì ta có thể sa thải nút v này và cập nhật S_u và w_u.
- Khi phần tử có chi phí sa thải thấp nhất còn lại trong heap mà u không thể sa thải được do $cost > S_u$ thì ta dừng việc sa thải nhân viên dưới quyền u.

Lưu ý: Ở một nút u, khi ta sa thải một nhân viên trực tiếp v thì u không thể sa thải bất kì một nhân viên nào khác ở dưới quyền của v bởi vì theo tính chất tham lam của thuật toán, ta ưu tiên sa thải hết những nhân viên theo thứ tự tăng dần về chi phí cho đến khi không thể sa thải được nữa. Gọi k là nhân viên có chi phí sa thải thấp nhất của v, nhưng khi v không thể sa thải được k thì tức là: $cost_k + w_v > m$. Suy ra, khi u sa thải v và sa thải luôn k thì không được vì $cost_k + cost_v + w_u > m$. Do đó, u không cần phải quản lý các nhân viên của v nữa.

Phân tích độ phức tạp thuật toán

- **Độ phức tạp thời gian của thuật toán** là $O(nlog(n))$, vì dùng DFS để xử lý mỗi nhân viên đúng một lần nên thao tác DFS tốn O(n) và mỗi cạnh cha-con chỉ được đưa vào/lấy ra khỏi priority_queue đúng một lần (mỗi thao tác tốn $O(log(n))$.
- **Độ phức tạp không gian của thuật toán** là $O(n)$ để lưu trữ cấu trúc cây, các hàng đợi ưu tiên và thông tin của mỗi nhân viên.

Chứng minh tính đúng đắn của thuật toán tham lam

Với mỗi nhân viên u, ta có:

$$C_u = w_u + deg(u), \quad S_u = m - C_u$$

Khi u sa thải một con trực tiếp v thì chỉ có C_u bị thay đổi:

$$C_u \leftarrow C_u + (C_v - 1)$$

Gọi $cost(v) = C_v - 1$ là chi phí mà u phải gánh chịu khi sa thải v.

1. Mô hình cục bộ

- Xét một nút u. Sau khi mọi cây con của từng con trực tiếp v của u đã được xử lý tối ưu trong phạm vi các cây con ấy, thì việc quyết định ở u là chọn một tập con $\mathcal{T} \subseteq \mathrm{Child}(u)$ sao cho:

$$\sum_{v \in \mathcal{T}} c_v \leq S_u$$

để tối đa hóa phần tử $|\mathcal{T}|$.

- Các chi phí $cost(v) = C_v - 1$ là cố định tại thời điểm u ra quyết định vì ta đã duyệt xong trong cây con gốc v.
- Việc sa thải một con v không làm thay đổi chi phí sa thải của các nhân viên trực tiếp v' khác của u.

2. Chiến lược tham lam tối ưu cục bộ

Cho một tập số thực $\{C_v\}_{v \in \mathcal{C}}$ và một ngưỡng $S \geq 0$.
Bài toán:

$$\max\{\, |\mathcal{T}| : \mathcal{T} \subseteq \mathcal{C}, \sum_{v \in \mathcal{T}} c_v \leq S \,\}$$

đạt nghiệm tối ưu bằng cách sắp tăng dần các C_v và lấy tiền tố dài nhất có tổng $\leq S$.

Sắp $C_{(1)} \leq C_{(2)} \leq \cdots \leq C_{(k)} \leq \ldots$. Gọi $\mathcal{T}^*$ là một nghiệm tối ưu với $|\mathcal{T}^*| = K$ lớn nhất. Nếu $\mathcal{T}^*$ không phải là tập gồm K phần tử rẻ nhất, tồn tại một phần tử $x \in \mathcal{T}^*$ và một phần tử $y \notin \mathcal{T}^*$ với $c_y \leq c_x$. Thay x bằng y ta được tập mới $\widehat{\mathcal{T}}$ có cùng kích thước, tổng chi phí vẫn $\leq S$, nên vẫn khả thi. Lặp trao đổi đến khi thu được đúng K phần tử rẻ nhất. Cuối cùng, để K là lớn nhất, hiển nhiên phải là tiền tố dài nhất có tổng $\leq S$.

3. Quy nạp toàn cục

- Khi chọn theo thứ tự tăng dần C_v và dừng lại tại tiền tố có tổng $\leq S_u$ thì mọi bước thực hiện ở phía sau đều đảm bảo $C_u \leq m$. Vì sau khi chọn một tiền tố có tổng $\sum$, giá trị C_u trở thành $C_u^{'}$ = C_u + $\sum$. Và $\sum \leq S_u = m - C_u$ nên $C_u^{'} \leq m$.

Quy nạp:

- Cơ sở: Với các node lá, không có con $\rightarrow$ Tối ưu
- Bước quy nạp: Giả sử mọi cây con của u đã tối ưu $\rightarrow$ các C_v cố định. Khi đó, theo mục 2, cách chọn tại u là tối ưu cho toàn bộ cây con gốc u. Áp dung quy nạp đến gốc 1 $\rightarrow$ Nghiệm toàn cục tối ưu.