**1.2.1. Cơ sở lý thuyết của thuật toán tham lam**

**1.2.1.1 Lý thuyết chung về giải thuật tham lam**

**- Thuật toán tham lam** (Greedy Algorithm) là một phương pháp giải bài toán trong đó các quyết định được đưa ra dựa trên việc chọn lựa tối ưu trong từng bước, mà không xem xét đến các quyết định trong tương lai. Ý tưởng chính của thuật toán tham lam là xây dựng một giải pháp theo từng bước, mỗi bước được chọn là tốt nhất trong số các lựa chọn có sẵn tại thời điểm đó.

* **Giải thuật tham lam có năm thành phần:**
* Một tập hợp các ứng viên (candidate), để từ đó tạo ra lời giải
* Một hàm lựa chọn, để theo đó lựa chọn ứng viên tốt nhất để bổ sung vào lời giải
* Một hàm khả thi (feasibility), dùng để quyết định nếu một ứng viên có thể được dùng để xây dựng lời giải
* Một hàm mục tiêu, ấn định giá trị của lời giải hoặc một lời giải chưa hoàn chỉnh
* Một hàm đánh giá, chỉ ra khi nào ta tìm ra một lời giải hoàn chỉnh.

**1.2.1.2 Thuật toán tham lam trong bài toán phân công công việc (Job Assignment Problem)**

* Thuật toán tham lam có thể được áp dụng trong bài toán phân công công việc (Job Assignment Problem) khi các điều kiện sau được thoả mãn:

***+ Tính Chất Tối Ưu Cục Bộ***

* Thuật toán tham lam hoạt động tốt khi việc lựa chọn tối ưu cục bộ tại mỗi bước dẫn đến một giải pháp tối ưu toàn cục. Điều này có nghĩa là việc chọn người thực hiện với chi phí thấp nhất cho từng công việc sẽ tạo ra phân công công việc tối ưu.

***+ Ma Trận Chi Phí Cố Định***

* Ma trận chi phí phải được định nghĩa trước, với các giá trị dương (hoặc không âm). Điều này giúp đảm bảo rằng không có chi phí âm có thể dẫn đến sự bất ổn trong việc tính toán tổng chi phí.

***+ Mỗi Công Việc Chỉ Được Phân Công Một Lần***

* Trong trường hợp mỗi công việc chỉ có thể được thực hiện bởi một người và mỗi người chỉ có thể thực hiện một công việc, thuật toán tham lam sẽ hoạt động hiệu quả. Nếu có thể phân công nhiều công việc cho cùng một người, hoặc một công việc có thể được thực hiện bởi nhiều người, thuật toán cần được điều chỉnh hoặc không thể áp dụng.

***+ Không Có Ràng Buộc Phức Tạp***

* Bài toán không có các ràng buộc phức tạp khác, chẳng hạn như giới hạn số lượng công việc cho mỗi người, độ ưu tiên của công việc, hoặc các yêu cầu cụ thể về năng lực của người thực hiện. Các ràng buộc như vậy có thể làm cho bài toán trở nên phức tạp hơn và cần sử dụng các phương pháp khác (như lập trình động hay quay lui).

**1.2.1.3 Nguyên lý hoạt dộng của thuật toán tham lam**

Giải thuật tham lam xây dựng các giải pháp bằng cách lựa chọn hành động tốt nhất tại mỗi bước, mà không xem xét tác động của hành động đó đến các bước sau. Ý tưởng này dựa trên quan sát rằng, nếu mỗi lần chọn hành động tốt nhất tại thời điểm hiện tại, ta có thể đạt được kết quả tối ưu toàn cục.

**1.2.1.4 Các bước giải thuật**

**Bước 1: Đầu vào**

* Có danh sách các công việc và danh sách các người thực hiện công việc.
* Cần có thông tin về thời gian thực hiện từng công việc đối với mỗi người hoặc các tiêu chí khác (chi phí, lợi ích).

**Bước 2: Sắp xếp**

* Sắp xếp các công việc dựa trên một tiêu chí nhất định (ví dụ: thời gian hoàn thành ngắn nhất trước, hoặc chi phí thấp nhất trước).
* Nếu không có tiêu chí rõ ràng, có thể chọn ngẫu nhiên.

**Bước 3: Phân công công việc**

* Bắt đầu từ công việc có ưu tiên cao nhất (theo sắp xếp ở bước 2) và gán công việc đó cho người phù hợp nhất.
* Đối với mỗi công việc, chọn người có khả năng thực hiện nhanh nhất hoặc có chi phí thấp nhất cho công việc đó.

**Bước 4: Cập nhật trạng thái**

* Sau khi phân công công việc cho người nào đó, cập nhật trạng thái của họ (ví dụ: thời gian rảnh tiếp theo của họ hoặc tổng chi phí đã phát sinh).
* Loại bỏ công việc đã được phân công ra khỏi danh sách công việc cần xử lý.

**Bước 5: Lặp lại**

* Lặp lại quá trình phân công cho các công việc còn lại, tiếp tục chọn công việc với tiêu chí tối ưu cục bộ và phân công cho người tốt nhất.

**Bước 6: Hoàn thành**

* Khi tất cả công việc đã được phân công, thuật toán dừng lại.