**BÁO CÁO NHÓM 7**

ĐỀ TÀI : ROSTERING PROBLEM

Bản báo cáo này trình bày một cách chi tiết hơn các giải thuật mà nhóm đã sử dụng trong bài tập lớn, bao gồm các giải thuật : Nhánh cận, Tham lam và Heuristic. Về phần sử dụng công cụ Ortools nhóm không trình bày ở đây.

1. **Giải thuật nhánh cận**

Ta lần lượt duyệt các khả năng lịch làm việc của N người theo mỗi ngày.

Với ngày thứ i, ta tìm tập Q những người có khả năng làm việc ngày hôm nay (hôm nay không phải ngày nghỉ phép, hôm qua không làm ca đêm). Nếu |Q| < 4a (Tức là không đảm bảo có đủ a nhân viên / ca ) thì ta thực hiện quay lui.

Phân tích : Nếu đã có >= 4a người có thể làm việc hôm nay, ta sẽ duyệt tất cả các trường hợp những người nào sẽ làm ca đêm hôm nay (hay sinh ra tất cả các tập con có a phần tử từ tập Q). Trong số những người còn lại ta có thể chọn vào các ca để thỏa mãn các ràng buộc và không làm ảnh hưởng đến hàm mục tiêu, vì vậy ta không cần quan tâm đến những người còn lại ai làm ca nào, ai nghỉ, .. điều này giúp giảm thiểu không gian tìm kiếm. Sau đó sẽ duyệt lần lượt tất cả những tập con này.

Điều kiện cắt nhánh : Sau khi duyệt xong 1 lời giải thỏa mãn ta sẽ cập nhật giá trị kỷ lục. Sau đó duyệt tiếp, nếu tính đến ngày đang duyệt (ngày thứ i) mà đã xuất hiện 1 người có số ca đêm >= kỷ lục thì ta cũng thực hiện quay lui, bỏ qua nhánh đang xét.

Ngoài ra, còn một điều kiện nữa có thể dùng để cắt nhánh, đó là tìm lower\_bound của hàm mục tiêu, nếu đã tìm ra được 1 lời giải bằng với giá trị lower\_bound này thì ta có thể kết thúc sớm giải thuật hơn.

Cách tính lower\_bound dựa theo nguyên lý Dirichlet : Do có N nhân viên,D ngày, mỗi ngày có một ca đêm, mỗi ca có a nhân viên làm việc do đó tồn tại 1 nhân viên có số ca đêm ít nhất là D\*a / N. Do số ca đêm là nguyên nên :

lower\_bound = D\*a/N nếu D\*a chia hết cho N hoặc

lower\_bound = [D\*a/N] + 1, ngược lại.

1. **Giải thuật tham lam**

Ta cũng duyệt những khả năng lập lịch cho N người ở mỗi ngày. Tuy nhiên khác với quay lui ta chỉ chọn 1 cách lập lịch duy nhất cho mỗi ngày mà không xét đến các trường hợp khác.

Giống như phân tích ở trên ta cần chọn a người mỗi ngày để làm ca đêm và đảm bảo có đủ 4a người có thể làm việc mỗi ngày.

Nhận xét : Do mỗi người sau khi làm ca đêm thì ngày hôm sau buộc phải nghỉ dù cho hôm sau có là ngày nghỉ phép hay không, vì vậy một cách “tham lam” để nhanh chóng có thể tối thiểu số ràng buộc vi phạm đó là nếu ngày mai là ngày phép thì cho họ làm ca đêm hôm nay.

Với ngày thứ i, ta tìm tập Q những người có khả năng làm việc ngày hôm nay (hôm nay không phải ngày nghỉ phép, hôm qua không làm ca đêm). Nếu |Q| < 4a thì giải thuật Tham lam thất bại, không tìm được lời giải.

Ngược lại : Ta sẽ chọn a người làm ca đêm từ tập này bằng cách sắp xếp :

Thứ nhất ưu tiên những người ngày mai nghỉ phép

Tiếp theo ưu tiên những người mà làm ít ca đêm nhất cho đến thời điểm hiện tại.

Ta duyệt lần lượt từng ngày đến khi i = D (Tìm ra lời giải, không đảm bảo tối ưu) hoặc giải thuật bị dừng lại, thuật toán thất bại, không tìm được lời giải.

1. **Giải thuật Heuristic**

Biểu diễn 1 lời giải : Ma trận N\*D, mỗi ô [i,j] trong ma trận nhận 1 trong 3 giá trị : 0,1,4 tương ứng người i vào ngày thứ j : nghỉ(0), làm 1 trong 3 ca sáng, trưa, chiều(1), và làm ca đêm (4).

Sử dụng Genetic Algorithm và Local Search

Sơ đồ thuật toán :

Khởi tạo quần thể 🡪 Local Search 🡪 Lai ghép 🡪 Đột biến 🡪 Chọn lọc

1. Khởi tạo quần thể : Khởi tạo P cá thể (nhóm sử dụng P = 200) trong quần thể, mỗi cá thể là một lời giải biểu diễn bằng 1 ma trận N\*D. Mỗi cá thể được khởi tạo bằng cách xây dựng lịch làm việc cho tất cả N người lần lượt theo mỗi ngày.

* Ở mỗi ngày tìm ra tập K các nhân viên có thể đi làm ngày hôm nay (Không nghỉ phép hôm nay, hôm qua không làm ca đêm), và tập L các nhân viên hôm qua làm ca đêm.
* Nếu |K| >= 4a, chọn ngẫu nhiên 4a người trong tập K người đó và cho họ đi làm hôm nay. Tiếp theo chọn ngẫu nhiên a người trong số 4a người đó và cho họ làm ca đêm hôm nay.
* Nếu tập |K| < 4a, chọn ngẫu nhiên 4a – |K| người từ tập L cho vào tập K để |K| = 4a và làm tương tự như trên.
* Việc khởi tạo như trên đảm bảo 2 ràng buộc : Mỗi ca làm có đúng a người làm, không ai phải đi làm vào ngày đã nghỉ phép. Như vậy chỉ còn duy nhất 1 điều ràng buộc có thể vi phạm đó là ràng buộc ngày hôm nay làm ca đêm nhưng ngày mai vẫn phải đi làm.

1. Local Search

Local Search bằng cách cố gắng đổi ca làm của 2 người trong 1 ngày để sinh ra 1 lời giải mới tốt hơn lời giải ban đầu bằng cách : 1) Giảm số lượng ràng buộc vi phạm, 2) Nếu không còn vi phạm ràng buộc thì tìm cách giảm số ngày làm ca đêm của người làm nhiều ca đêm nhất.

1. Giảm số lượng ràng buộc

Ví dụ 1 lời giải còn ràng buộc vi phạm :

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 4 | 1 | 1 | 0 | 4 | 1 |
| 1 | 0 | 1 | 4 | 1 | 4 |
| 0 | 4 | 0 | 1 | 1 | 0 |
| 1 | 1 | 1 | 1 | 0 | 1 |
| 1 | 0 | 4 | 1 | 1 | 1 |

B1 : Chọn ngẫu nhiên 1 vi phạm trong tất cả các vi phạm ràng buộc

B2 : Đổi ca của ngày hôm nay từ ca đêm thành ca khác hoặc nghỉ, nếu không được thì tìm cách đổi ca ngày mai thành nghỉ.

1. Đổi ca đêm của ngày hôm nay

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 4 | 1 | 1 | 0 | 4 | 1 |
| 1 | 0 | 1 | 1 | 1 | 4 |
| 0 | 4 | 0 | 1 | 1 | 0 |
| 1 | 1 | 1 | 4 | 0 | 1 |
| 1 | 0 | 4 | 1 | 1 | 1 |

1. Đổi ca của ngày mai thành ngày nghỉ

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 4 | 1 | 1 | 0 | 4 | 1 |
| 1 | 0 | 1 | 4 | 0 | 4 |
| 0 | 4 | 0 | 1 | 1 | 0 |
| 1 | 1 | 1 | 1 | 1 | 1 |
| 1 | 0 | 4 | 1 | 1 | 1 |

Trong trường hợp không còn vi phạm ràng buộc, ta chọn người c1 đang làm nhiều ca đêm nhất (t ca), ta chọn một người bất kì trong số những người còn lại (c2) có số ca đêm < t -1, xong đó xét trong số những ca làm đêm của người c1 1 ca bất kỳ mà có thể đổi cho người c2 và không bị vi phạm ràng buộc.

Ví dụ : Người 1 đang làm 2 ca đêm, người 4 chưa phải làm ca đêm nào

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 4 | 0 | 1 | 1 | 4 | 1 |
| 1 | 1 | 1 | 4 | 0 | 4 |
| 0 | 4 | 0 | 1 | 1 | 1 |
| 1 | 1 | 1 | 1 | 1 | 0 |
| 1 | 0 | 4 | 0 | 1 | 1 |

Ta tìm được ngày thứ 5 mà 2 người có thể đổi ca cho nhau

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 4 | 0 | 1 | 1 | 0 | 1 |
| 1 | 1 | 1 | 4 | 0 | 4 |
| 0 | 4 | 0 | 1 | 1 | 1 |
| 1 | 1 | 1 | 1 | 4 | 0 |
| 1 | 0 | 4 | 0 | 1 | 1 |

Ta thu được 1 lời giải có hàm mục tiêu tốt hơn.

1. Lai ghép

Ta có 2 cá thể như sau :

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 1 | 1 | 0 | 1 | 4 | 1 |
| 1 | 0 | 1 | 0 | 1 | 4 |
| 4 | 1 | 1 | 1 | 1 | 0 |
| 0 | 4 | 1 | 1 | 0 | 1 |
| 1 | 1 | 0 | 4 | 1 | 1 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 4 | 1 | 1 | 0 | 1 | 0 |
| 1 | 0 | 1 | 4 | 4 | 1 |
| 0 | 4 | 0 | 1 | 1 | 4 |
| 1 | 1 | 1 | 1 | 1 | 1 |
| 1 | 0 | 4 | 1 | 0 | 1 |

Cách lai ghép : Chọn 1 điểm cắt chia mỗi lời giải thành 2 phần, sau đó ghép chéo với nhau để tạo ra 2 lời giải con mới.

A screenshot of a computer

Description automatically generated with medium confidence

1. Đột biến :

Chọn 2 người và 1 ngày bất kỳ sao cho họ làm khác ca nhau, sau đó đổi ca làm việc của họ trong ngày đó cho nhau, ví dụ :

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 4 | 1 | 1 | 0 | 1 | 0 |
| 1 | 0 | 1 | 4 | 4 | 1 |
| 0 | 4 | 0 | 1 | 1 | 4 |
| 1 | 1 | 1 | 1 | 1 | 1 |
| 1 | 0 | 4 | 1 | 0 | 1 |

Chuyển thành :

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 4 | 1 | 1 | 0 | 1 | 0 |
| 1 | 0 | 1 | 1 | 4 | 1 |
| 0 | 4 | 0 | 1 | 1 | 4 |
| 1 | 1 | 1 | 1 | 1 | 1 |
| 1 | 0 | 4 | 4 | 0 | 1 |

1. Chọn lọc :

Ta đánh giá điểm của mỗi cá thể qua công thức :

100\* (số ràng buộc vi phạm) + số ca đêm lớn nhất của mỗi người.

Ở công thức này ta đã phạt rất nặng những ràng buộc bị vi phạm.

Sau đó ta thực hiện chọn lọc tinh hoa, chọn trong các cá thể cha mẹ và cá thể con vừa sinh ra những cá thể có điểm số thấp nhất duy trì trong quần thể.

Sau g vòng lặp (thế hệ) (nhóm chọn g = 150) thì ta kết thúc giải thuật, trả về cá thể có điểm số thấp nhất.