Faculty of Computer Science and Engineering
Ho Chi Minh City University of Technology

NHẬP MÔN AI

GVHD: Vương Bá Thịnh
Nguyễn Phương Vương – 1614186

1. Định nghĩa trạng thái

- Sử dụng ma trận vuông M có kích thước n x n để biểu diễn các trận đấu của các đối thủ sao cho tất cả các đối thủ đều đấu đủ k trận. Nếu 2 bên là đối thủ của nhau thì đánh dấu 1, ngược lại đánh dấu 0 trên ma trận.
- Gọi i, j lần lượt là chỉ số hàng và chỉ số cột với (0 <= i, j <= n-1).
- Ma trận phải thoả các điều kiện sau:

+ Sum(M[i]) = k; Sum(M[j]) = k , với i,j = 0..n-1 tức tổng mỗi hàng bằng tổng mỗi cột và bằng k.
+ M[i][j] = 0 với i = j thể hiện 1 vận động viên không đấu với chính mình.
- Ví dụ, với n = 4, k = 2:

0	1	0	0
1	0	1	0
0	1	0	1
1	0	1	0

2. Tạo trạng thái khởi đầu

- Trạng thái khởi đầu được sinh cố định và đảm bảo mỗi vận động viên đầu đấu với k trận.
- Cách hiện thực:

+ Sinh ma trận M có kích thước n x n và đánh tất cả các chỉ số = 0.
+ Truy cập từng hàng trong ma trận (i=1..n), điền đủ k chỉ số 1 vào mỗi cột của hàng đó bằng cách duyệt cột (j=1..n). Chỉ số tại cột j đang bằng 0 ,tức M[i][j] = 0, xét hàng tại vị trí đối xứng trong ma trận (hàng j), nếu tổng chỉ số của hàng j nhỏ nhất so với tổng các chỉ số tất cả các hàng còn lại thì cập nhật M[i][j] = 1 và M[j][i] = 1.

Tức là Sum(M[j]) < Sum(M[k]) , với k = 1 .. n và k khác j
Lưu ý: ta không xét chỉ số ở dường chéo.
Ví dụ: Ma trận 4x4 , k = 2

0	0	0	0
0	0	0	0
0	0	0	0
0	0	0	0

+ Với i = 0, sum(M[0]) < 2) . Xét j = i+1 đến n-1:

- j = 1: M[0][1] = 0, (sum(M[1]) = 0) <= sum(M[k]) với (0<=k<=3 và k !=1) đạt Min, cập nhật M[0][1] = 1, M[1][0] =1

0	1	0	0
1	0	0	0
0	0	0	0
0	0	0	0

- j = 2: M[0][2] = 0, sum(M[2]) = 0) <= sum(M[k]) với (0<= k <=3 và k != 2) đạt Min, cập nhật M[0][2] = 1, M[2][0] =1

  Khi đó sum(M[1]) = 2, đủ số trận => thoát vòng lặp j, chuyển sang hàng kế tiếp

0	1	1	0
1	0	0	0
1	0	0	0
0	0	0	0

+ Với i = 1, sum(M[1) = 1 < 2). Xét j = i+1 đến n-1:
j = 2: M[1][2] = 0, sum(M[2] = 1 > sum(M[3] = 0), chuyển đến j = 3.
j = 3: M[1][3] = 0, sum(M[3] = 0 đạt min), cập nhật M[1][3] = 1, M[3][1] = 1.
Khi đó, sum(M[1] = 2), đủ số trận, chuyển sang hàng kế tiếp.

0	1	1	0
1	0	0	1
1	0	0	0
0	1	0	0

+ Tương tự với i = 2, ta có kết quả cuối cùng:

0	1	1	0
1	0	0	1
1	0	0	1
0	1	1	0

3. Cách chuyển sang trạng thái kế tiếp

3.1. Các bước thực hiện

- Từ trạng thái ban đầu ta chuyển trạng thái tốt hơn tức đảm bảo hàm lượng giá là tốt nhất.
- Ta chuyển trạng thái bằng cách:

Bước 1: Ta duyệt từng hàng của ma trận với điều kiện các vị trí ta xét đến ở đường chéo chính đều được bỏ qua.
Bước 2: Khi gặp vị trí 0 hoặc 1 tại M[i][j] ta chạy tiếp cột để tìm vị trí thoả M[i][j] + M[i][cotTiep] = 1.
Bước 3: Sau khi tìm được ta duyện đi xuống từ vị trí cột j và cột cotTiep đó với dòng từ i + 1 -> n – 1.
Bước 4: Ta tiếp tục tìm được 2 vị trí mới M[dongMoi][j] và M[dongMoi][cotTiep] thoả: M[dongMoi][i] + M[i][j] = 1 và M[i][cotTiep] = M[dongMoi][cotTiep]
Bước 5: Ta hoán đổi giá trị của vị trí (M[i][j], M[i][cotTiep]) , (M[dongMoi][j], M[dongMoi][cotTiep]).
Bước 6: Ta cập nhật lại ma trận đối xứng với những thay đổi ở trên.
Bước 7: Tính hàm lượng giá cho mỗi ma trận thoả mãn.
Bước 8: Ta tiếp tục duyệt ma trận có hàm lượng giá nhỏ nhất rồi lặp lại bước 1 để kiểm tra các ma trận con thoả mãn điều kiện được sinh ra từ ma trận đó. Việc lặp được dừng lại đến khi ta không tìm được ma trận con nào có hàm lượng giá nhỏ hơn ma trận cha tức tất cả các ma trận con sinh ra từ cha đều có hàm lượng giá lớn hơn ma trận cha.
Bước 9: Lấy ma trận cha thoả mãn ra và có được lời giải tối ưu cho bài toán tạo xếp lịch thi đấu.

3.2. Ví dụ

- Với 1 ma trận ở trạng thái khởi đầu như sau:

Số hàng / số cột	0	1	2	3
0	0	1	1	0
1	1	0	0	1
2	1	0	0	1
3	0	1	1	0

- Theo thứ tự duyệt từ trên xuống, ta bỏ qua M[0][0] do nó nằm đường chéo chính.
- Ta tìm được M[0][1] + M[0][3] = 1 thoả điều kiện.
- Sau đó, ta duyệt xuống dưới với đúng 2 cột đó. Ta được M[2][1] + M[0][1] = 1 và M[2][3] + M[0][3] = 1 thoả điều kiện.
- Thực hiện hoán đổi vị trí của M[0][1] và M[0][3], M[2][1] và M[2][3].

Số hàng / số cột	0	1	2	3
0	0	0	1	1
1	1	0	0	1
2	1	1	0	0
3	0	1	1	0

- Cập nhật ma trận đối xứng với từng thay đổi ta được ma trận mới thoả mãn.

Số hàng / số cột	0	1	2	3
0	0	0	1	1
1	0	0	1	1
2	1	1	0	0
3	1	1	0	0

- Thực hiện kiểm tra nó với hàm lượng giá, nếu nó đạt nhỏ nhất thì dừng lại. Nếu không thì thực hiện sinh các ma trận con từ nó để xét tiếp.

4. Hàm lượng giá

4.1. Các bước thực hiện

Ta chọn hàm lượng giá như vậy vì ứng với mỗi ma trận con sinh ra từ ma trận cha, ta phải xét ma trận con nào là tốt nhất tức có hàm lượng giá nhỏ nhất để tìm ra lời giải bài toán tối ưu nhất.

Bước 1: Với mỗi ma trận thoả mãn ta có được đối thủ của từng vận động viên. Từ đó ta tìm được tổng số điểm của đối thủ mà vận động viên phải đấu.

Bước 2: Sau khi có tổng số điểm của đối thủ mà vận động viên phải đấu ta tính số điểm trung bình của tổng điểm đó.

Bước 3: Tính độ chênh lệch trung bình của từng tổng số điểm trung bình đối thủ của mỗi vận động viên với số điểm trung bình của tổng điểm vừa tìm được.

4.2. Ví dụ

- Số điểm của mỗi vận động viên:

score = [2,3,4,5], k = 2 (mỗi vận động viên đấu k trận).

- Tổng số điểm của đối thủ mỗi vận động viên:

sum = [8, 6, 8, 6] → sumAverage = 7

- Tính độ chênh lệch trung bình của từng tổng điểm đó:

standardDeviation = (|8 - 7|+|6 - 7|+|8 - 7|+|6 - 7|) / 4 = 1.

Vậy ta phải tìm ma trận có kết quả hàm lượng giá nhỏ nhất để tìm ra lời giải tối ưu của bài toán.

0	1	0	1
1	0	1	0
0	1	0	1
1	0	1	0

5. Đánh giá thuật toán

- Thuật toán thực hiện bằng cách chạy các ma trận con thoả mãn (tức đảm bảo số hàng bằng số cột bằng k) được sinh ra từ ma trận cha. Nếu ma trận con được sinh ra với hàm lượng giá nhỏ hơn thì tiếp tục được duyệt để đánh giá. Nếu tất cả các ma trận con đều có hàm lượng gía không tốt hơn ma trận cha thì ma trận cha được xem là tối ưu nhất và giải thuật dừng.
- Thuật toán còn khá chậm. Chạy test 60 VĐV và đấu 2 trận trong 208s.