Задача ранжирования на примере реальных данных

Шахматный турнир

В данной задаче рассмотрим шахматный турнир, который проводится по круговой системе в два круга (аналог турнира претендентов).
Всего участников $n = 8$.
Количество туров - 14 (каждый участник играет по две партии с каждым другим участником турнира).

На вход подаем матрицу смежности $A = a_{ij}$ размера $n \times n$, которая строится следующим образом:

$$a_{ij} = \begin{cases} 1, \text{если игрок } i \text{ победил у игрока } j; \\ 1/2, \text{если игрок } i \text{ и игрок } j \text{ сыграли вничью;} \\ 0, \text{иначе.} \end{cases}$$

Сумма элементов i-й строки $s_i^1 = \sum_{j=1}^{n} a_{ij}$ соответствует количеству побед, одержанных i-ым игроком в ходе турнира и является мерой силы i-го игрока. Вектор $\mathbf{S^1}$, составленный из элементов s_i^1 называется вектором сил первого порядка.
В тривиальном случае все элементы вектора $\mathbf{S^1}$ различны, и ранжирование игроков не представляет сложности. Однако если по итогам турнира несколько игроков одержали одинаковое количество побед (набрали одинаковое количество очков), однозначное ранжирование игроков может быть затруднительным.
В реальных шахматных турнирах используется коэффициент Бергера, который высчитывается следующим образом: коэффициент Бергера определённого участника складывается из суммы всех очков противников, у которых данный участник выиграл, плюс половина суммы очков противников, с которыми данный участник сыграл вничью. Практика показывает, что этого коэффициента достаточно для определения, среди нескольких участников с одинаковыми очками, более сильного игрока (если, все таки, коэффициент совпадает, то существуют другие коэффициента). Но мы будем использовать Алгоритм Кендалла-Уэя.

Алгоритм Кендалла-Уэя

Алгоритм Кендалла-Уэя предполагает вычисление итерированных сил k-го поорядка игроков по одной из следующих формул:

$$\mathbf{S^k} = \mathbf{A} \cdot \mathbf{S^{k-1}};$$

$$\mathbf{S^k} = \mathbf{A^k} \cdot \mathbf{S^1},$$

где $k \leq 2$ - номер итерации; A - матрица смежности; $\mathbf{S^j}$ - вектор итерированных сил j-го порядка.
Далее, наибольший интерес представляют относительные силы σ_i, которые вычисляются следующим образом:

$$\sigma_i = \lim_{k\to\infty} \frac{s_i^k}{\sum_{j=1}^n s_j^k}.$$

Вектор σ, составленный из σ_i, сходится к нормализированному собственному вектору матрицы A, который соответствует максимальному собственному (в абсолюте) значению. Таким образом, суть алгоритма Кендалла-Уэя такая, что мы ранжируем участников в соответствии с компонентами векторами σ, т.е. у кого значение компоненты больше, тот стоит выше.

Так как, турнир проходит по круговой системе в два круга, то предлагаю подавать на вход две матрицы смежности. Далее, для обоих матриц запускать алгоритм и результирующие вектора σ сложить. Таким образом, ранжировать игроков уже по в соответствии с компонентами нового вектора.

Что реализовано в программе?

1. Шахматы

В программе Шахматы (Chess):
1. Считываем матрицу из файла (данные были взяты из первого круга Турнира претендентов 2020-2021 года).
2. Добавляем столбец с количеством побед каждого игрока и сортируем.
3. Так как несколько участников имеют одинаковое количество очков, добавляем коэффициент Бергера.
4. Так как несколько участников имеют одинаковый коэффициент Бергера, то добавляем усеченный коэффициент Бергера (Berg 2), который соответствует коэффициенту Бергера, из которого вычитается самый низкий результаты соперников.
4.1. Если и усеченные коэффициенты Бергера совпадают, то можно продолжать процедуру усечения.
5. Для сравнения подготовил Алгоритм Кендалла-Уэя. При дележке мест есть несовпадение игрока 3 и 7. Алгоритм К-У считает, что Игрок 7 сильнее.

1. Футбол

В программе Футбол (FB) матрица строится следующим образом:

$$a_{ij} = \begin{cases} k, \text{если команда } i \text{ забила команде } j\ k \text{ голов;} \\ 0, \text{если } i = j. \end{cases}$$

На вход подается матрица ($n \times 2n$). Где все n команд играют между собой 2 раза (поэтому количество столбцов - $2n$). Главный показатель "силы" команды - количество набранных очков (если команда i победила команду j, то команде i присуждается 3 очка и проигравшей 0 очков, если команды сыграли вничью, то обеим командам присуждается по 1 очку). Если после всех матчей у нескольких команд одинаковое количество очков, то следующий показатель - разница между забитыми и пропущенными голами.
Таким образом, в программе реализовано следующее:

1. Считываем матрицу из файла (данные были взяты рандомно).
2. Добавляем столбец с количеством очков каждой команды и сортируем.
2.1 Сразу добавил столбец с разницей мячей и отсортировал.