

В данной работе я решал задачу модификации алгоритма подсчёта PageRank на примере датасета web-Stanford. Модифицированный алгоритм сравнивался по скорости сходимости с алгоритмом PowerIteration.

Файл web-Stanford.txt имеет вид:

```
# Directed graph (each unordered pair of nodes is saved once): web-Stanford.txt
# Stanford web graph from 2002
# Nodes: 281903 Edges: 2312497
# FromNodeId    ToNodeId
1                6548
1                15409
6548            57031
...
```

Первая строчка означает, что из вершины 1 есть ребро в вершину 6548. Из-за длительности вычисления PageRanka на всём наборе, эксперименты проводились на выборке первых 10000 и 100000 строк из файла.

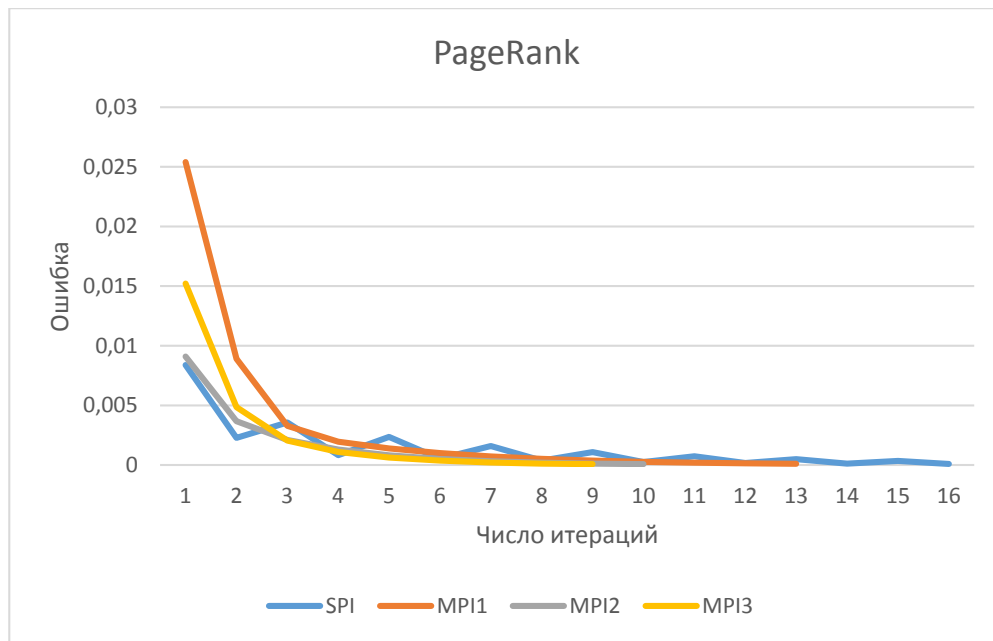
В методе PowerIteration начальный ранг у всех вершин одинаковый. Ранг вершин вычисляется итеративно по формуле:

$$x_{k+1} = dPx_k + (1 - d)Z, \quad (1)$$

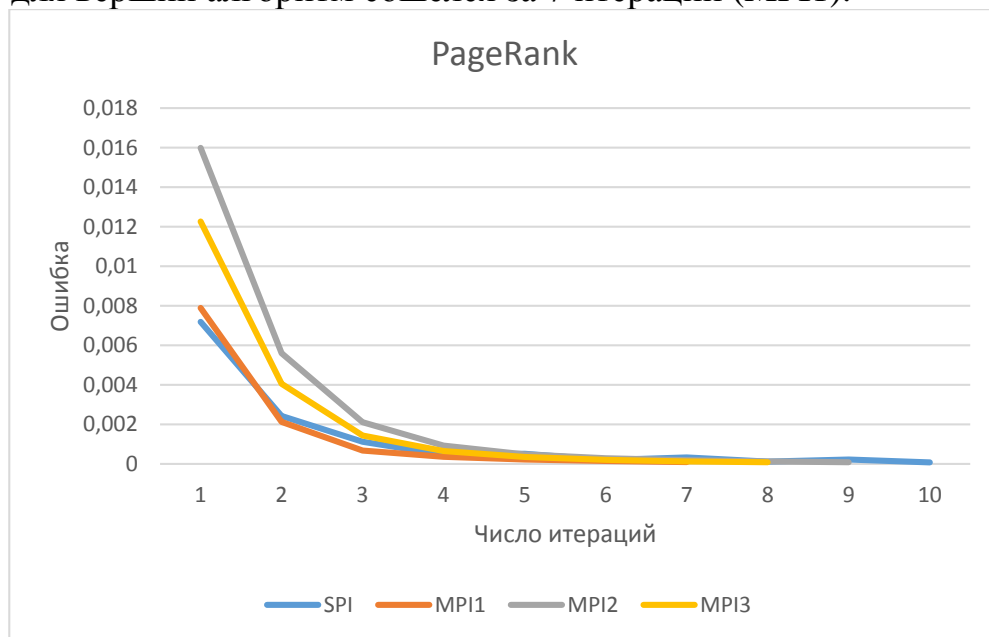
где d равен 0.85; в матрице переходов P вероятность перехода из любой вершины во все вершины, на которые эта вершина ссылается, одинакова; Z — вектор телепортации, в котором переход во все вершины равновероятен. Из всех узлов переход во все остальные также равновероятен.

Основная модификация алгоритма состояла в том, что при подсчёте PageRanka учитывалась история. В формуле (1) в векторе x_k используются уже найденные компоненты вектора x_{k+1} . Также пробовалось брать начальный ранг вершин не одинаковым, как в простом PowerIteration, а в зависимости от числа ссылок: ранг вершины тем больше, чем больше других вершин на неё ссылаются. По такому же принципу пробовалось брать вектор телепортации Z .

При подсчёте PageRanka на выборке из 10000 строк из файла простой PowerIteration сошёлся за 16 итераций (SPI на графике). С учётом истории, равновероятной телепортацией и начальной инициализацией по числу входящих вершин алгоритм сошёлся за 13 итераций (MPI1). При одинаковом начальном ранге для вершин алгоритм сошёлся за 10 итераций (MPI2). Авторитетная телепортация, которая учитывает число ссылок на вершины, позволила сократить число итераций до 9 (MPI3).



На выборке из 100000 строк PowerIteration сошёлся за 10 итераций. С учётом истории, равновероятной телепортацией и начальной инициализацией по числу входящих вершин алгоритм сошёлся за 9 итераций (MPI2). Авторитетная телепортация, которая учитывает число ссылок на вершины, позволила сократить число итераций до 8 (MPI3). При одинаковом начальном ранге для вершин алгоритм сошёлся за 7 итераций (MPI1).



Из эксперимента было выявлено, что использование уже вычисленных компонент вектора рангов для нахождения ещё не известных рангов вершин улучшило скорость сходимости на обоих наборах. Одинаковая начальная инициализация рангов вершин оказалась лучше, чем по числу ссылок на вершины. Авторитетная телепортация на одном наборе улучшила скорость сходимости, на другом ухудшила.