

ИКН НИТУ МИСИС
Комбинаторика и теория графов

Задача построения стабильных бракосочетаний.
Алгоритм Гэйла-Шепли

Исполнитель:
Миронов Е.А. БИВТ-23-8
(https://github.com/Valet-V0ult-de-Fur1e/combinatorics_and_graphs_repo)

Москва 2024 год

Определение задачи стабильных бракосочетаний

Задача о стабильном сопоставлении (она же задача построения стабильных бракосочетаний или алгоритм отложенного согласия) - заключается в нахождении стабильного паросочетания между элементами двух множеств, имеющих свои предпочтения.

Формулировка задачи стабильных бракосочетаний

Есть n мужчин и n женщин. Они обладают следующими особенностями:

1. Каждый человек оценивает лишь людей противоположного пола (все гетеросексуальны);
2. Каждый мужчина может отсортировать женщин от наименее привлекательной к наиболее привлекательной, причем его предпочтения не меняются (у каждого мужчины своя функция оценки);
3. Каждая женщина может отсортировать мужчин от наименее привлекательного к наиболее привлекательному, причем её предпочтения не меняются (у каждой женщины своя функция оценки).

По такому определению строится полный двудольный граф (левая доля — мужчины, правая — женщины)

Устойчивое паросочетание — паросочетание без неустойчивых пар.

Пара $\langle A, b \rangle$ называется неустойчивой, если:

1. В паросочетании есть пары $\langle A, a \rangle$ и $\langle B, b \rangle$ (A женат на a , B женат на b);
2. A предпочитает b элементу a ;
3. b предпочитает A элементу B

Алгоритм Гейла-Шепли

Алгоритм Гейла — Шепли предназначен для определения стабильного паросочетания. Его временная сложность является полиномиальной функцией от размера входных данных. В зависимости от конкретной реализации алгоритм может находить решение, оптимальное либо для одной из групп участников сопоставления, либо для другой.

Данный алгоритм гарантирует честность процедуры с точки зрения участников, то есть предоставление ложных данных одним из участников может привести только к ухудшению его результата.

Алгоритм носит имена своих создателей — Дэвида Гейла и Ллойда Шепли.

Описание алгоритма

1. Мужчины делают предложение наиболее предпочитаемой женщине.
2. Каждая женщина из всех поступивших предложений выбирает наилучшее и отвечает на него «может быть» (помолвка), на все остальные отвечает «нет» (отказ).
3. Мужчины, получившие отказ, обращаются к следующей женщине из своего списка предпочтений, мужчины, получившие ответ «может быть», ничего не делают.

4. Если женщине пришло предложение лучше предыдущего, то онапрежнему претенденту (которому ранее сказала «может быть») говорит «нет», а новому претенденту говорит «может быть».
5. Шаги 11-44 повторяются, пока у всех мужчин не исчерпается список предложений, в этот момент женщины отвечают «да» на те предложения «может быть», которые у них есть в настоящий момент.

Пример работы алгоритма

На представленных ниже рисунках приоритетность каждого кандидата зависит от его места в списке, где приоритет убывает с лева направо.

A(3, 4, 2, 1)	1(A, D, C, B)
B(3, 2, 4, 1)	2(A, D, B, C)
C(1, 3, 4, 2)	3(A, C, D, B)
D(2, 4, 3, 1)	4(B, A, D, C)

Рис1. Список приоритетов двух групп: мужчин (A, B, C, D) и женщин (1, 2, 3, 4)

Данный алгоритм основывается на инициативности только группы «мужчин», но он так же будет работать, если инициативность будут проявлять только группа «женщин».

Шаг 1. Идем по списку мужчин и ищем «свободных», то есть не связанных с какой-либо женщиной, в данном случае первым будет мужчина А. Смотрим на список его приоритетов: на первом месте стоит женщина 3. Смотрим не связана ли женщина 3 с каким-либо мужчиной. Как можем увидеть: она свободна, - следовательно можем их связать. Теперь можно вычеркнуть из списка приоритетов мужчины А женщину 3.

A(3, 4, 2, 1)	1(A, D, C, B)
B(3, 2, 4, 1)	2(A, D, B, C)
C(1, 3, 4, 2)	3(A, C, D, B)
D(2, 4, 3, 1)	4(B, A, D, C)

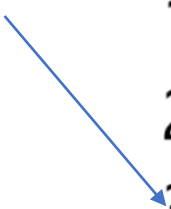


Рис2. Визуализация шага 1

Шаг 2. Идем дальше по списку мужчин и ищем «свободных». Следующим будет мужчина В. Проходимся по списку его приоритетов. На первом месте стоит женщина 3. Проверяем её на связь с каким-либо другим мужчиной. Видим, что она связана с мужчиной А. Смотрим список приоритетов женщины 3: мужчина А имеет больший приоритет, чем мужчина В, следовательно он получает отказ. Вычеркиваем её из списка приоритетов мужчины В и смотрим следующую претендентку: это женщина 2. Смотрим

на наличие у неё каких-либо связей: она свободна. Связываем мужчину 3 и женщину 2. Теперь можно вычеркнуть из списка приоритетов мужчины В женщину 2

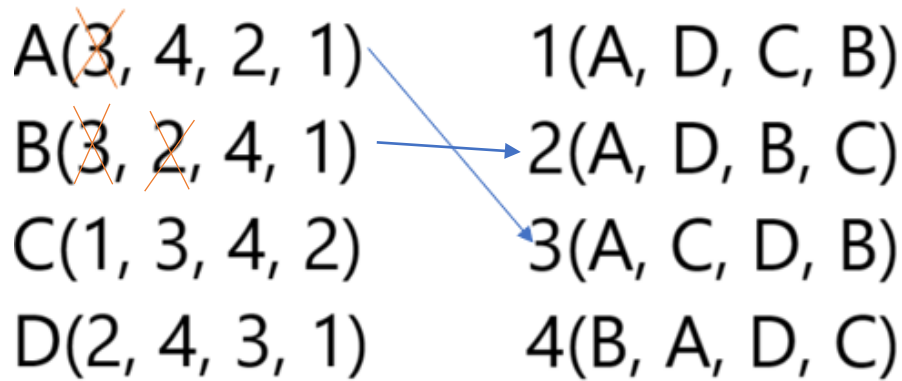


Рис2. Визуализация шага 2

Шаг 3. Идем дальше по списку мужчин и ищем «свободных». Следующим будет мужчина С. Проходимся по списку его приоритетов. На первом месте стоит женщина 1. Проверяем её на связь с каким-либо другим мужчиной. Видим, что она свободна, поэтому можем их связать. Теперь можно её вычеркнуть из списка приоритетов мужчины С

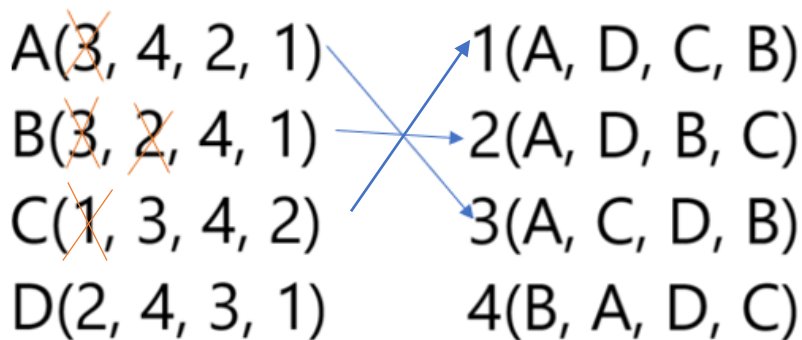


Рис3. Визуализация шага 3

Шаг 4. Идем дальше по списку мужчин и ищем «свободных». Следующим будет мужчина D. Проходимся по списку его приоритетов. На первом месте стоит женщина 2. Проверяем её на связь с каким-либо другим мужчиной. Видим, что она связана с мужчиной В. Смотрим список приоритетов женщины 3: мужчина D имеет больший приоритет, чем мужчина В, следовательно отвязываем мужчину В от женщины 2 (теперь он «свободный») и подвязываем к ней мужчину D, убирая её из списка его приоритетов.

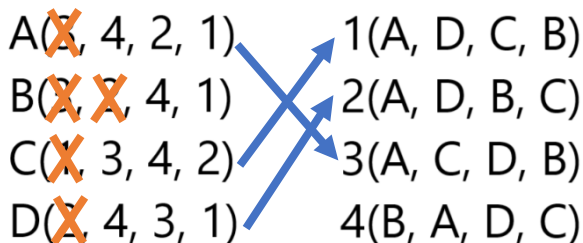


Рис4. Визуализация шага 4

Шаг 5. Дальше идем по списку мужчин и ищем «свободных». Следующим будет мужчина В. Проходимся по списку его приоритетов. На первом месте после женщины 2 стоит женщина 4. Проверяем её на связь с каким-либо другим мужчиной. Видим, что она свободна, поэтому можем их связать. Теперь можно её вычеркнуть из списка приоритетов мужчины В.

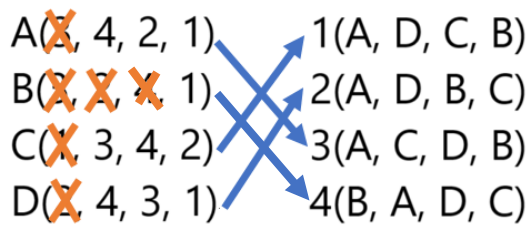


Рис5. Визуализация шага 5

Как можем увидеть свободных мужчин у нас не осталось. Алгоритм завершает своё действие

Реализация алгоритма

Ниже приведены реализации алгоритма в псевдокоде и на языке программирования python версии 3.9

```

пока существуют свободные мужчины:
    м = свободный мужчина
    ж = первая женщина из текущего списка м
    если ж свободна:
        м и ж связаны
    если ж предпочитает м своему текущему партнеру м':
        м и ж связаны
        вычеркиваем ж из списка предпочтений м'
        помечаем м'
    иначе:
        вычеркиваем ж из списка предпочтений м
    
```

Рис.6 Реализация в псевдокоде

```

1  prefer = [[7, 5, 6, 4], [5, 4, 6, 7],
2            [4, 5, 6, 7], [4, 5, 6, 7],
3            [0, 1, 2, 3], [0, 1, 2, 3],
4            [0, 1, 2, 3], [0, 1, 2, 3]]
5  N = len(prefer) // 2
6
7
8  def wPrefersMOverM1(prefer, w, m, m1):
9      for i in range(N):
10         if prefer[w][i] == m:
11             return False
12         if prefer[w][i] == m1:
13             return True
14
15
16 def stableMarriage(prefer):
17     preferData = prefer.copy()
18     wPartner = [-1 for i in range(N)]
19     freeList = [i for i in range(N)]
20     while len(freeList) > 0:
21         m = freeList.pop()
22         w = preferData[m][-1]
23         m1 = wPartner[w - N]
24         if m1 == -1:
25             wPartner[w - N] = m
26         elif wPrefersMOverM1(preferData, w, m, m1):
27             wPartner[w - N] = m
28             preferData[m1].remove(w)
29             freeList.append(m1)
30         else:
31             preferData[m].remove(w)
32             freeList.append(m)
33     print("Woman ", " Man")
34     for i in range(N):
35         print(i + N, "\t", wPartner[i])
36
37 stableMarriage(prefer)

```

Рис.7 Реализация на python 3.9

Временная и пространственная сложность

Пространственная сложность равна $O(n^2)$.

Хранение списка предпочтений имеет пространственную сложность $O(2 * n * n) \sim O(n^2)$, так как включает в себя списки фаворитов всех участников двух групп

Временная сложность: $O(n^2)$

Проход по списку свободных мужчин имеет сложность $O(n)$, проход по списку фаворитов для фаворитки свободного мужчины тоже $O(n)$. Так как один проход вложен в другой, то общая временная сложность будет $O(n^2)$. На каждой итерации мужчина делает предложение очередной женщине. Но всего может быть не более n^2 предложений.

Мужчины делают предложения женщинам в порядке убывания симпатии.

В результате работы алгоритма все мужчины и женщины будут заняты.

Предположим, что некоторый мужчина (А) не женат по завершении алгоритма. Тогда и некоторая женщина (а) незамужняя. а не получала предложений. Но АА сделал предложения всем женщинам, так как он остался не женат. Получаем противоречие. Таким образом, все мужчины заняты.

Аналогичные рассуждения применяем для женщин. Пусть какая-то женщина незамужняя. Значит, есть мужчина, который остался не женат. Но мы доказали, что по завершении алгоритма все мужчины заняты. Снова пришли к противоречию.

После завершения алгоритма не будет неустойчивых пар.

Предположим $\langle A, b \rangle$ (где A, B — мужчины; a, b — женщины; A женат на a , B женат на b) — нестабильная пара в паросочетании, найденном алгоритмом Гейла-Шепли. Возможны два случая:

1. A не делал предложение b . Значит, A находит a более привлекательной, чем b . Но чтобы рассматриваемая пара была неустойчивой, необходимо, чтобы b для A была более привлекательна, чем a . Следовательно, $\langle A, b \rangle$ — устойчивая пара.
2. A делал предложение b . Тогда был такой момент, когда b отказала A , значит, b находит B более привлекательным, чем A . Снова получается, что $\langle A, b \rangle$ — устойчивая пара.

Анализ аналогов

Алгоритм Ирвинга, временная и пространственная сложность которого равна $O(n^2)$. Изначально алгоритм Ирвинга решает похожую задачу, в которой нет разделения на группы, что, как следствие, не гарантирует наличие разбиения на устойчивые паросочетания.

Алгоритм, основанный на Субраманьяне. Он использует подход с фиксированной точкой и, в отличие от Гейла-Шепли, симметричен относительно пола. В конечном итоге достигается фиксированная точка, и можно считать оптимальные для мужчины и женщины соответствия.

Аналог алгоритма Гейла-Шепли с использованием математического ожидания применяют в задаче распределения абитуриентов по вузам. При моделировании различных вариантов приёмной кампании (количество квот, абитуриентов) наиболее близким по результатам к алгоритму Гейла-Шепли во всех случаях является алгоритм поступления по правилу математического ожидания с $\lambda = S - 1$.

Описание алгоритма Ирвинга

1. Участники расставляют других участников по приоритету, а затем делают предложение всем в своём списке, переходя к следующему участнику, если их предложение отклонено.
2. Предложение отклоняется, когда человек получает предложение от того, кому он отдаёт предпочтение. Предложение также может быть отклонено, когда человек, которому мы отдаём предпочтение, делает предложение, и мы отклоняем ранее принятое предложение.
3. Если человек является последним предпочтением для всех остальных и отвергается всеми, стабильного соответствия не существует.
4. Первый этап заканчивается, когда каждый сделал предложение и получил предложение от кого-то другого.
5. Теперь мы исключаем все предпочтения, которые ниже, чем у того, чьё предложение было принято, и симметрично.
6. Далее мы исключаем повороты.
7. Когда у всех участников в списке останется только один человек, мы остановимся. Это стабильное соответствие соседей по комнате.

Модификация алгоритма Гейла-Шепли

Так же процесс составления парасочетаний можно ускорить правильным расставлением «свободных» мужчин в очереди, основанным на рейтинге популярности каждого кандидата среди противоположного пола. Рейтинг рассчитывается из суммы баллов, которые выдаются каждому мужчине за место в списке предпочтений в каждой женщины.

1(A, D, C, B)

2(A, D, B, C)

3(A, C, D, B)

4(B, A, D, C)

Рис8. Список фаворитов для каждой женщины

На основе данных из Рис8 можно составить рейтинг популярности среди мужчин:

1. Мужчина A: 4(от женщины 1) + 4(от женщины 2) + 4(от женщины 3) + 3(от женщины 4) = 15
2. Мужчина B: 1(от женщины 1) + 2(от женщины 2) + 1(от женщины 3) + 4(от женщины 4) = 8
3. Мужчина C: 2(от женщины 1) + 1(от женщины 2) + 3(от женщины 3) + 1(от женщины 4) = 7
4. Мужчина D: 3(от женщины 1) + 3(от женщины 2) + 2(от женщины 3) + 2(от женщины 4) = 10

После сортировки по количеству очков получим следующий список (A, D, B, C), при обходе по которому используя алгоритм Гейла-Шепли, мы составим устойчивые парасочетания за 4 итерации прохода по списку свободных мужчин.

Данная модификация в некоторых случаях может привести к временной сложности $O(n * n \log n)$, когда нет мужчин с одинаковым количеством очков

```
16 def stableMarriage(prefer):
17     preferData = prefer.copy()
18     wPartner = [-1 for i in range(N)]
19     ratings = [[i, 0] for i in range(N)]
20     for place_i in range(N):
21         for w_i in range(N, N * 2):
22             ratings[preferData[w_i][place_i]][1] += place_i
23     freeList = [mi[0] for mi in sorted(ratings, key=lambda x: x[1], reverse=True)]
24     while len(freeList) > 0:
25         m = freeList.pop()
26         w = preferData[m][-1]
27         m1 = wPartner[w - N]
28         if m1 == -1:
29             wPartner[w - N] = m
30         elif wPrefersMOverM1(preferData, w, m, m1):
31             wPartner[w - N] = m
32             preferData[m1].remove(w)
33             freeList.append(m1)
34         else:
35             preferData[m].remove(w)
36             freeList.append(m)
37     print("Woman ", " Man")
38     for i in range(N):
39         print(i + N, "\t", wPartner[i])
```

Рис9. Реализация модификации алгоритма Гейла-Шепли

При этом в общая временная сложность остаётся равной $O(n^2)$

Примеры реального применения

Система поиска донорских органов. Система, которую разработал учёный совместно с Тайфоном Сонмезом и Уткой Унвером Элвин, была призвана помочь тем, кто не может стать донором для своих близких из-за несовместимых типов крови или иных факторов. Согласно системе, производился обмен почками между несовместимыми парами, то есть донор из первой пары отдавал почку реципиенту из второй, а донор из второй — реципиенту из первой. Внедрение данной системы позволило значительно увеличить число проводимых пересадок и существенно сократить время их ожидания для больных.

Координация приёма в высшие учебные заведения во Франции. Форма алгоритма Гейла-Шепли, выполняемая с помощью реального протокола, используется для этого с 2018 года через систему Parcoursup. В рамках этого процесса в течение лета, предшествующего началу учёбы, абитуриенты получают предложения о приёме и должны в каждом раунде процесса выбирать, принимать ли какое-либо новое предложение.

Назначение новых докторов в больницы. В 1980-х годах экономист из Гарварда Эвлин Рот использовал алгоритм Гейла-Шепли для изменения Национальной программы распределения по ординатурам (National Residency Match Program, NRMP) — системы, которая назначает новых докторов в больницы по всей стране.

Поиск наилучших соответствий в портовых операциях. В статье журнала судоходства и торговли от 2023 года говорится, что алгоритм Гейла-Шепли используется для решения задач стабильного сопоставления, в том числе для поиска наиболее подходящего решения искусственного интеллекта для решения каждой задачи.

Ссылка на реализацию

https://github.com/Valet-V0ult-de-Fur1e/combinatorics_and_graphs_repo/tree/main/Задача%20построения%20стабильных%20браков%20сочетаний%20алгоритм%20гейла%20шепли

Список источников

- <https://iq.opengenus.org/stable-roommates-problem/>
- https://neerc.ifmo.ru/wiki/index.php?title=Задача_об_устойчивом_паросочетании#.D0.9E.D1.81.D0.BD.D0.BE.D0.B2.D0.BD.D0.B0.D1.8F_.D0.B7.D0.B0.D0.B4.D0.B0.D1.87.D0.B0
- <https://habr.com/ru/articles/463391/>
- <https://github.com/oedokumaci/gale-shapley>
- https://dspace.spbu.ru/bitstream/11701/10822/1/Julia_Sverlova_-_diploma.pdf
- <https://algorithmica.org/ru/matching>
- <https://axiomaonline.ru/informatics/algorithm/7.html>
- https://kpfu.ru/staff_files/F1094050804/YA_paper.pdf
- https://translated.turbopages.org/proxy_u/en-ru.ru.c3b06f42-673a23f0-ee792ea-74722d776562/cgm.cs.mcgill.ca/~avis/courses/360/2013/ktlaerid/LP-bliss.pdf