Приднестровский государственный университет им. Т.Г. Шевченко

Кафедра прикладной математики и информатики

**Доклад**

на тему:

«Задача о назначениях»

Выполнил: студент 502 группы

Соловенко Вадим Григорьевич

Тирасполь, 2016 год

**Постановка задачи.**

При решении некоторых задач менеджмента приходится назначать исполнителей (людей, механизмы и т.п.) для выполнения некоторых однотипных работ. При этом дополнительно известны значения – доходы (затраты) выполнения -м исполнителем -ой работы. Требуется распределить исполнителей по работам таким образом, чтобы максимизировать (минимизировать) суммарный доход (затраты) выполнения всех работ.

Данная задача носит название «задача о назначениях» и является частным случаем более общей транспортной задачи. Если число исполнителей равно числу выполняемых работ, то такая задача является сбалансированной, в противном случае – не сбалансированной. В случае сбалансированной задачи о назначениях выполняются два условия: каждый исполнитель выполняет только одну работу, и каждая работа выполняется только одним исполнителем.

Выполним математическую постановку задачи.

1. Определим неизвестные и их количество. Рассмотрим переменные , которые равны 1, если -й исполнитель назначен на выполнение -ой работы и 0, если он не назначен . Таким образом, значения образуют матрицу назначений , состоящую из нулей и единиц.
2. Запишем критерий оптимизации (целевую функцию) – суммарный доход (затраты) от выполнения всех работ. Целевая функция задачи о назначениях зависит от неизвестных пока переменных и известных значений – дохода (затрат) от выполнения -м исполнителем -ой работы и запишется в следующем виде.

|  |  |
| --- | --- |
|  | (1) |

1. Сформулируем ограничения рассматриваемой задачи.

* каждый исполнитель выполняет только одну работу. Выполнение данного условия означает, что каждая строка матрицы назначений содержит только одно значение равное единицы, а все остальные равны нулю. Т.е. сумма элементов каждой строки матрицы назначений равна 1, и это ограничение можно записать в виде системы n уравнений

|  |  |
| --- | --- |
|  | (2) |

* каждая работа выполняется только одним исполнителем. Выполнение данного условия означает, что каждый столбец матрицы назначений содержит только одно значение равное единицы, а все остальные равны нулю. Т.е. сумма элементов каждого столбца матрицы равна 1, и это ограничение можно записать в виде системы n уравнений

|  |  |
| --- | --- |
|  | (3) |

* двоичность переменных . Так как областью допустимых изменений каждой переменной является не множество целых неотрицательных чисел, а конечное множество

|  |  |
| --- | --- |
|  | (4) |

Целевая функция (1) и ограничения (2−4) составляют математическую модель задачи о назначениях. В случае, когда задача называется сбалансированной, при – несбалансированной.

**Методы решения**

Из математической модели задачи видно, что задача о назначениях является частным случаем транспортной задачи, когда объемы производства всех поставщиков и потребности всех потребителей равны единице. Транспортная задача в свою очередь является частным случаем задачи линейного программирования.

Поэтому, задачу о назначениях можно решать, как методами линейного программирования (например, симплекс-методом), так и методами решения транспортной задачи (например, метод итерационного улучшения плана перевозок). Однако, для решения задачи о назначениях были созданы специализированные методы, самым известным из которых является венгерский алгоритм.

**Венгерский алгоритм** — алгоритм оптимизации, решающий сбалансированную задачу о назначениях за полиномиальное время. Он был разработан и опубликован Харолдом Куном (англ.) в 1955 году. Временная сложность оригинального алгоритма была , однако Эдмондс и Карп показали, что его можно модифицировать так, чтобы достичь времени выполнения .

Алгоритм основан на двух идеях:

* если из всех элементов некой строки или столбца вычесть одно и то же число , общая стоимость уменьшится на , а оптимальное решение не изменится;
* если есть решение нулевой стоимости, оно оптимально.

Алгоритм ищет значения, которые надо вычесть из всех элементов каждой строки и каждого столбца (разные для разных строк и столбцов), такие, что все элементы матрицы останутся неотрицательными, но появится нулевое решение.

Последовательность действий:

**Шаг 1.** Алгоритм предназначен для нахождения назначения при котором реализуется минимум целевой функции. Поэтому, если у нас задача на максимум, то ее нужно преобразовать в задачу на минимум. Для этого, из исходной матрицы дохода , получают матрицу затрат . Матрицу можно получить несколькими способами. Самый простой из них это:

То есть, из задачи на нахождение максимума целевой функции получается задача на нахождение минимального отклонения от максимума.

Однако, более эффективным способом является переход к матрице отклонений от максимума для каждого исполнителя в отдельности:

При таком подходе в каждой строке новой матрицы будет как минимум один нулевой элемент, что будет полезно на следующий шагах алгоритма.

В случае задачи на минимум сразу переходим к шагу 2.

**Шаг 2.** Алгоритм предназначен для решения сбалансированной задачи. В случае, когда , т.е. число работ не равно числу исполнителей поступают следующим образом:

* Если , то добавляют фиктивных исполнителей с затратами равными нулю: при ;
* Если , то добавляют фиктивных работ с затратами равными нулю: при ;

**Шаг 3.** Уменьшаем элементы построчно. Для каждой строки находим минимальный элемент и вычитаем его из всех элементов данной строки. Получаем новую матрицу:

Теперь в каждой строке матрицы есть хотя бы один ноль.

**Шаг 4.** Уменьшаем элементы по столбцам. Для каждого столбца находим минимальный элемент и вычитаем его из всех элементов данного столбца. Получаем новую матрицу:

Теперь в каждом столбце матрицы есть хотя бы один ноль.

**Шаг 5.** Мы получили матрицу , в которой в каждой строки и каждом столбце есть ноль. Нашей целью является назначение каждому исполнителю работы с минимальной стоимостью, то есть, со стоимостью равной нулю. Таким образом необходимо последовательно отмечать нули в матрице , руководствуясь правилом, что в одной строке (столбце) может быть отмечен только один ноль. Например:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  | 0 |  |  |  |
|  |  | 0 |  |  |  |  |
|  |  |  |  |  |  | 0 |
|  |  |  |  |  |  |  |
|  |  |  |  | 0 |  |  |

Если в каждой строке и в каждом столбе есть отмеченный элемент, то для каждой работы мы смогли назначить исполнителя с нулевой себестоимостью, а значит, задача о назначениях решена. Из матрицы можно составить матрицу назначений во следующим правилам:

* Если элемент равен нулю и отмечен, то
* Иначе .

Для приведенного выше примера матрица назначений примет вид:

**Шаг 6.** Если на шаге 5 решение не найдено. То отмечаем все строки без назначения и все столбцы с нулями в этих строках. Теперь проводим линии через все отмеченные столбцы и неотмеченные строки. Например:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  |  | \* | \* |  |  |  |
|  |  |  |  |  |
|  |  |  | 0 |  |  |  |
|  |  | 0 |  |  |  |  |
|  |  |  |  |  |  | 0 |
|  |  |  |  |  |  |  |
| \* | |  | 0 |  |  |  |
|  |  |  |  |  |  |  |
| \* | | 0 |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  | 0 |  |  |

Все эти действия преследовали лишь одну цель: провести наименьшее количество линий (вертикалей и горизонталей), чтобы покрыть все нули. Можно было воспользоваться любым другим методом вместо описанного.

**Шаг 7.** Из непокрытых линиями элементов матрицы (в данном случае это и где ) найти наименьший . Вычесть его из всех не отмеченных строк и прибавить ко всем пересечениям отмеченных строк и столбцов.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  |  | \* | \* |  |  |  |
|  |  |  |  |  |
|  |  |  | 0 |  |  |  |
|  |  | 0 |  |  |  |  |
|  |  |  |  |  |  | 0 |
|  |  |  |  |  |  |  |
| \* | |  | 0 |  |  |  |
|  |  |  |  |  |  |  |
| \* | | 0 |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  | 0 |  |  |

Повторять процедуру (шаги 3-7) до тех пор, пока назначение не станет возможным.

**Пример**

В Галактике бушует гражданская война между повстанцами и жестокой Галактической Империей. Силы не равны и повстанцы держатся только потому, что Империя не знает расположения их баз. Повстанцам необходимо совершить налеты на объекты империи таким образом, чтобы суммарное расстояние которые преодолевают ударные звенья космолетов до своих целей было максимальным.  
Информация о базах повстанцев:

|  |  |  |
| --- | --- | --- |
| **База** | **Координаты относительно галактического центра (в парсеках)** | **Количество ударных звеньев на базе** |
| **База 1 (Б1)** | (-1000, 5000) | 1 |
| **База 2 (Б2)** | (-3500, -7000) | 1 |
| **База 3(Б3)** | (5000, 500) | 1 |
| **База 4 (Б4)** | (8000, 2000) | 2 |

Информация о целях:

|  |  |
| --- | --- |
| **Цель** | **Координаты относительно галактического центра (в парсеках)** |
| **Цель 1 (Ц1)** | (-7000 2000) |
| **Цель 2 (Ц2)** | (-6500, 4500) |
| **Цель 3 (Ц3)** | (2000, 1500) |
| **Цель 4(Ц4)** | (1000, 6000) |
| **Цель 5 (Ц5)** | (2000, -5000) |
| **Цель 6 (Ц6)** | (7000, -6500) |
| **Цель 7 (Ц7)** | (-8000, -4000) |

Карта Галактики:

**Решение**

Так как координаты звеньев и целей нам известны , то расстояние, которое необходимо пролететь -му звену до -ой цели можно найти по формуле:

Наша цель назначить пять звеньев на поражение целей так, чтобы пройденное суммарное расстояние, пройденное звеньями было максимальным. Видно, что данная задача является задачей о назначениях с целевой функцией:

Где – матрица назначений, – количество звеньев, – количество целей – матрица эффективности.

Подставляя координаты баз и целей в выражение для нахождения получим

Для того, чтобы применить венгерский алгоритм нужно перейти от матрицы дохода к матрице затрат . Для этого воспользуемся соотношением:

Получим

Целей больше чем звеньев. Задача несбалансированная, поэтому введем два фиктивных звена.

**Шаг1.** Получим:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | Ц1 | Ц2 | Ц3 | Ц4 | Ц5 | Ц6 | Ц7 |
| Зв 1 | 7301 | 8486 | 9399 | 11773 | 3569 | 0 | 2607 |
| Зв 2 | 4100 | 1872 | 3633 | 0 | 7905 | 3245 | 8349 |
| Зв 3 | 1664 | 1581 | 10595 | 6956 | 7492 | 6477 | 0 |
| Зв 4 | 2088 | 2374 | 11067 | 9026 | 7868 | 8529 | 0 |
| Зв 5 | 2088 | 2374 | 11067 | 9026 | 7868 | 8529 | 0 |
| Зв 6 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Зв 7 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

В каждой строке и в каждом столбце есть по нулю. Отмечаем нули так, чтобы было только по одному нулю в каждой строке и столбце. Например, так:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | Ц1 | Ц2 | Ц3 | Ц4 | Ц5 | Ц6 | Ц7 |
| Зв 1 | 7301 | 8486 | 9399 | 11773 | 3569 | 0 | 2607 |
| Зв 2 | 4100 | 1872 | 3633 | 0 | 7905 | 3245 | 8349 |
| Зв 3 | 1664 | 1581 | 10595 | 6956 | 7492 | 6477 | 0 |
| Зв 4 | 2088 | 2374 | 11067 | 9026 | 7868 | 8529 | 0 |
| Зв 5 | 2088 | 2374 | 11067 | 9026 | 7868 | 8529 | 0 |
| Зв 6 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Зв 7 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Не во всех строках и столбцах удалось отметить ноль. Отмечаем все строки без назначения и все столбцы с нулями в этих строках. Проводим линии через все отмеченные столбцы и неотмеченные строки.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | **Ц1** | **Ц2** | **Ц3** | **Ц4** | **Ц5** | **Ц6** | **Ц7 +** |
| **Зв 1** | 7301 | 8486 | 9399 | 11773 | 3569 | 0 | 2607 |
| **Зв 2** | 4100 | 1872 | 3633 | 0 | 7905 | 3245 | 8349 |
| **Зв 3** | 1664 | 1581 | 10595 | 6956 | 7492 | 6477 | 0 |
| **Зв 4 +** | 2088 | 2374 | 11067 | 9026 | 7868 | 8529 | 0 |
| **Зв 5 +** | 2088 | 2374 | 11067 | 9026 | 7868 | 8529 | 0 |
| **Зв 6** | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| **Зв 7** | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Из непокрытых линиями элементов матрицы выбираем наименьший . Вычитаем его из всех не отмеченных строк и прибавляем ко всем пересечениям отмеченных строк и столбцов.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | **Ц1** | **Ц2** | **Ц3** | **Ц4** | **Ц5** | **Ц6** | **Ц7 +** |
| **Зв 1** | 7301 | 8486 | 9399 | 11773 | 3569 | 0 | 4695 |
| **Зв 2** | 4100 | 1872 | 3633 | 0 | 7905 | 3245 | 10437 |
| **Зв 3** | 1664 | 1581 | 10595 | 6956 | 7492 | 6477 | 2088 |
| **Зв 4 +** | 0 | 286 | 8979 | 6938 | 5780 | 6441 | 0 |
| **Зв 5 +** | 0 | 286 | 8979 | 6938 | 5780 | 6441 | 0 |
| **Зв 6** | 0 | 0 | 0 | 0 | 0 | 0 | 2088 |
| **Зв 7** | 0 | 0 | 0 | 0 | 0 | 0 | 2088 |

**Шаг2.** Вычитая минимальные элементы, получаем нули в каждой строке и столбце:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | **Ц1** | **Ц2** | **Ц3** | **Ц4** | **Ц5** | **Ц6** | **Ц7 +** |
| **Зв 1** | 7301 | 8486 | 9399 | 11773 | 3569 | 0 | 4695 |
| **Зв 2** | 4100 | 1872 | 3633 | 0 | 7905 | 3245 | 10437 |
| **Зв 3** | 83 | 0 | 9014 | 5375 | 5911 | 4896 | 507 |
| **Зв 4 +** | 0 | 286 | 8979 | 6938 | 5780 | 6441 | 0 |
| **Зв 5 +** | 0 | 286 | 8979 | 6938 | 5780 | 6441 | 0 |
| **Зв 6** | 0 | 0 | 0 | 0 | 0 | 0 | 2088 |
| **Зв 7** | 0 | 0 | 0 | 0 | 0 | 0 | 2088 |

Отмечаем нули так, чтобы в каждой строке и в каждом столбце был только один отмеченный ноль.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | **Ц1** | **Ц2** | **Ц3** | **Ц4** | **Ц5** | **Ц6** | **Ц7 +** |
| **Зв 1** | 7301 | 8486 | 9399 | 11773 | 3569 | 0 | 4695 |
| **Зв 2** | 4100 | 1872 | 3633 | 0 | 7905 | 3245 | 10437 |
| **Зв 3** | 83 | 0 | 9014 | 5375 | 5911 | 4896 | 507 |
| **Зв 4 +** | 0 | 286 | 8979 | 6938 | 5780 | 6441 | 0 |
| **Зв 5 +** | 0 | 286 | 8979 | 6938 | 5780 | 6441 | 0 |
| **Зв 6** | 0 | 0 | 0 | 0 | 0 | 0 | 2088 |
| **Зв 7** | 0 | 0 | 0 | 0 | 0 | 0 | 2088 |

Так как в каждой строке и столбе нам удалось отметить ноль, то назначение удалось и матрицу назначений можно записать в виде:

Значение целевой функции при таком назначении:

Ответ: Первое звено должно поразить шестую цель, второе звено – четвертую, третье – вторую, четвертое – первую, пятое – седьмую. Третья и пятая цели останутся не пораженными, так как у повстанцев не хватает ударных звеньев. Суммарное расстояние, которое пролетят звенья до своих целей равно 72030 парсек.

Зв1

Зв3

Зв4

**Литература**