

Основы линейного программирования

Коротков М.

Гаврилов М.

24 апреля 2003 г.

Содержание

1	Линейное программирование.	3
1.1	Задача	3
1.2	Графический метод	6
1.3	Дополнительные переменные	8
1.4	Анализ чувствительности	9
2	Симплекс-метод	12
2.1	Стандартная форма задачи	12
2.2	Базисные решения	14
2.3	Алгоритм симплекс-метода	14
2.4	Искусственные начальные решения	15
2.5	Особые случаи	17
3	Теория симплекс-метода	18
3.1	Матричное представление задачи	18
3.2	Основные теоремы	20
3.3	Обоснование симплекс-метода	23
4	Модификации симплекс-метода	25
4.1	Мультипликативное представление обратной матрицы (модифицированный симплекс-метод)	25
4.2	Метод решения задач с ограниченными переменными	28
5	Марковские процессы с бесконечным числом этапов	32
6	Транспортная задача	35
6.1	Вспомогательные теоремы	35
6.2	Решение транспортной задачи	36
6.3	Метод потенциалов	37

1 Линейное программирование.

1.1 Задача

Дать исчерпывающее определение задачи линейного программирования не так просто. Ниже мы приведем пример задачи данного типа и продемонстрируем разбор этой задачи, то есть решение основных связанных с ней проблем на базе самого простого – графического метода ее решения.

Определение 1 *Задачей линейного программирования называется задача минимизации (максимизации) значения целевой функции (нескольких переменных), с учетом ограничений на значения этих переменных (выраженных явно, или через уравнения связи), где все зависимости являются линейными.*

Это, несколько тяжеловесное определение можно объяснить на примерах конкретных задач.

Замечание 1 (О линейных зависимостях) *Следует помнить, что “полностью линейные” зависимости редко встречаются в практических задачах. Нелепо утверждать, что стоимость перевозки одной книжки составляет 50% стоимости перевозки двух книжек. Также, можно предположить что при покупке 10000 буханок хлеба потраченная сумма не в 10000 раз отличается от стоимости одной буханки. Задача оказывается линейной, если можно выбрать единицы измерения так, чтобы зависимости можно было считать линейными. (Можно предположить, что для перевозки книжек контейнерами стоимость перевозки линейно зависит от количества контейнеров)*

Любая задача линейного программирования включает следующие три элемента.

- (i) **Переменные.**
- (ii) **Целевая функция** (функция, подлежащая минимизации или максимизации)
- (iii) **Ограничения**, которым переменные должны удовлетворять.

Рассмотрим конкретный пример.

ЗАДАЧА 1 Типография “Веселая печать” печатает книги математического и художественного содержания. На 1000 экземпляров математических книг уходит 6 тонн бумаги и 1 бочка типографской краски, а на 1000 экземпляров художественных – 4 тонны бумаги и 2 бочки краски (они тоньше и с картинками :). Причем типография может использовать не более 24 тонн бумаги и 6 бочек краски в месяц, дабы избежать протестов со стороны Greenpeace.

Ежемесячный тираж художественных книг был ограничен 2 тысячами штук (из-за отсутствия надлежащего спроса на них), кроме того, министерство образования поставило условие, чтобы тираж художественных книг превышал тираж математических не более чем на 1000 штук (дабы население повышало уровень собственного образования). Одна математическая книга приносит доход в 5 рублей, а одна художественная – 4. Типография заботится не только об уровне математического образования населения, но и о собственной выгоде и хочет максимизировать свой доход.

Выпишем данные задачи в таблицу:

	математические книги	художественные книги	максимальный объем
бумага, тонн на 1000 штук	6	4	24
краска, бочек на 100 штук	1	2	6
доход, рублей	5	4	

Теперь попробуем в данной задаче выделить основные элементы.

(i) Переменные. Здесь это тиражи соответствующих типов книг.

x_1 – ежемесячный тираж математических книг (1000 штук).

x_2 – ежемесячный тираж книг художественного содержания (1000 штук).

(ii) Используя эти переменные мы можем построить целевую функцию. Логично предположить, что это будет функция суммарного дохода.

$$z = 5x_1 + 4x_2$$

Эта функция измеряется в тысячах условных рублей. Таким образом задача оптимизации следующая:

$$\text{Максимизировать } z = 5x_1 + 4x_2$$

(iii) Итак, нам осталось определить последний элемент – ограничения на переменные. То есть описать ограничения на количество продукции и объем сырья.

- (i) Во-первых, объем сырья (бумаги и краски) не должен превышать максимально допустимый объем. Из таблицы имеем следующее: Используется $6x_1 + 4x_2$ тонн бумаги и $x_1 + 2x_2$ бочек краски. Вспомнив, что Greenpeace требует ограничить использование сырья, получаем следующие ограничения

$$6x_1 + 4x_2 \leq 24$$

$$x_1 + 2x_2 \leq 6$$

- (ii) Кроме того, у нас есть ограничение на спрос (печатать не более 2000 художественных книг в месяц), которое запишется как $x_2 \leq 2$)
- (iii) А также ограничение со стороны минобразования; его можно сформулировать так: разность ежемесячных тиражей художественной и математической литературы не превышает единицы, то есть $x_2 - x_1 \leq 1$)
- (iv) Еще одно, сразу неочевидное, ограничение: типография не занимается утилизацией собственной продукции, значит, она выпускает неотрицательное число книг, что запишется как $x_1 \geq 0$; $x_2 \geq 0$.

Таким образом, задача получается следующая:

$$\begin{aligned} &\text{Максимизировать} \\ &z = 5x_1 + 4x_2 \\ &\text{при выполнении ограничений:} \\ &6x_1 + 4x_2 \leq 24 \\ &x_1 + 2x_2 \leq 6 \\ &x_2 \leq 2 \\ &x_2 - x_1 \leq 1 \\ &x_1 \geq 0 \\ &x_2 \geq 0 \end{aligned}$$

Определение 2 Назовем допустимым любое решение удовлетворяющее ограничениям.

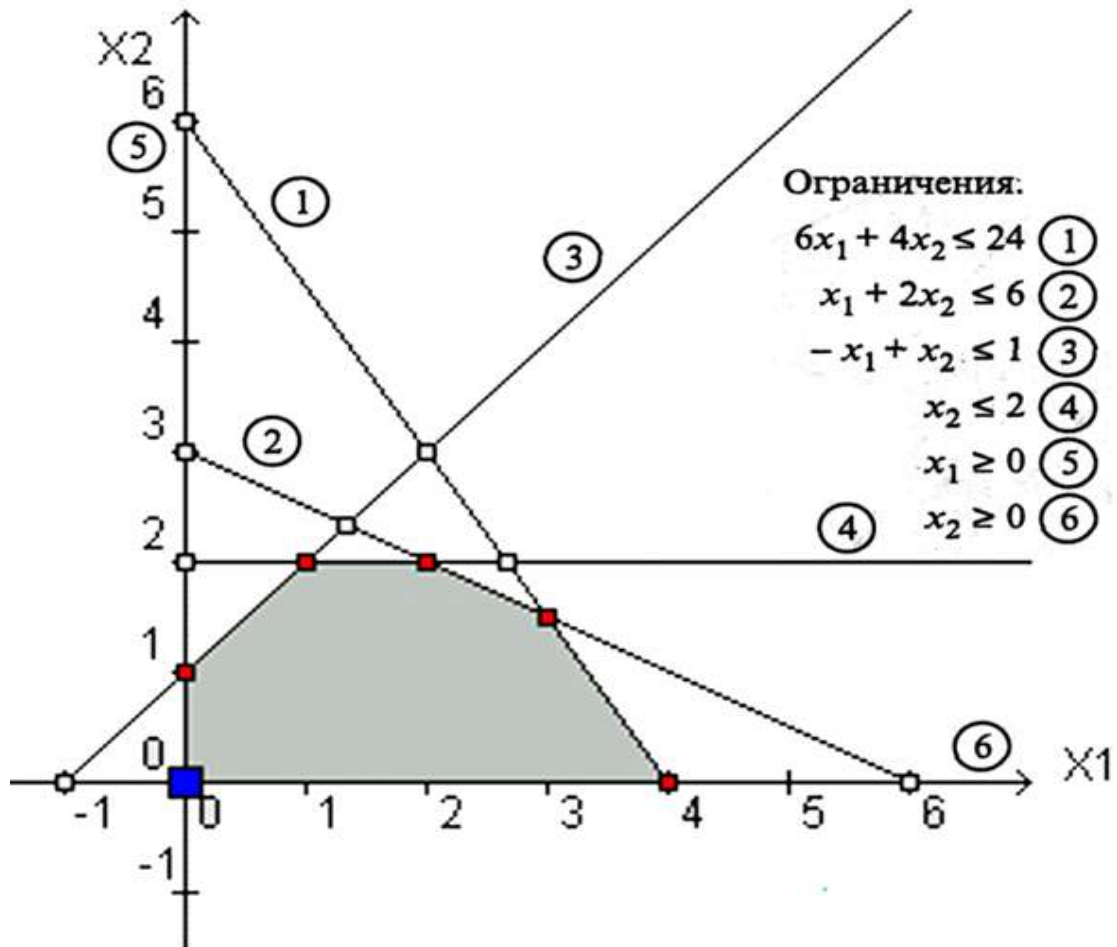
Например, решение печатать 3000 математических и 1000 художественных книг в месяц ($x_1 = 3$; $x_2 = 1$) будет допустимым, в чем легко удостоверится. Прибыль при этом составит 19 тысяч рублей. Итак, наша цель – нахождение **оптимального допустимого решения**, то есть допустимого решения максимизирующего нашу целевую функцию.

1.2 Графический метод

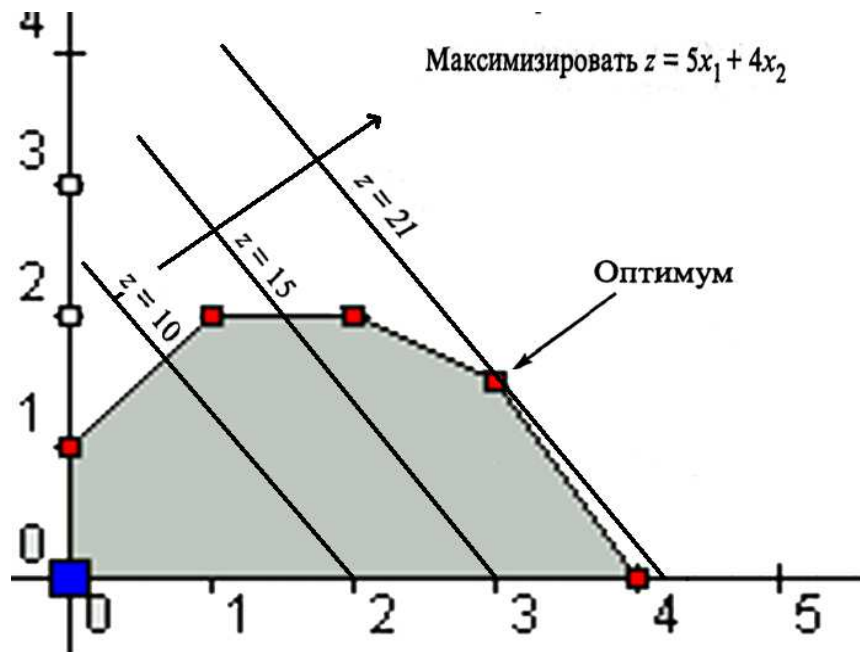
Здесь мы рассмотрим графический метод решения задачи линейного программирования.

***Замечание 2** Графический метод применим в случае двух переменных. Практических приложений у этого метода немного, но он важен для наглядности и понимания происходящего. При описании других методов мы часто будем ссылаться на графическое представление решения.*

Вернемся к нашей задаче о типографии 1. Сначала мы хотим построить **пространство решений**, то есть получить область, точки которой удовлетворяют всем ограничениям. Каждое ограничение задает полуплоскость ограниченную некоторой прямой, пересечение этих полуплоскостей есть пространство решений.



Далее, мы хотим максимизировать $z = 5x_1 + 4x_2$. Рассмотрим семейство соответствующих прямых. Направление возрастания функции $z = 5x_1 + 4x_2$ есть перпендикуляр к прямым соответствующего семейства. Соответственно, мы можем увеличивать значение целевой функции, пока соответствующая ей прямая семейства имеет хотя бы одну общую точку с пространством решений.



В нашем случае оптимальное решение соответствует точке С. То есть лучшим решением будет выпускать 3000 математических и 1500 художественных книг в месяц, получая при этом прибыль в 21000 рублей. Таким образом, мы “выиграли” 2000 рублей “недополученной” типографией прибыли (по сравнению с первым решением).

Замечание 3 (Об угловой точке) Очевидно, что если область задана произвольным многоугольником, а максимизируемая функция имеет общий вид $z = ax + by$, то решение (оптимум) все равно будет достигаться в угловой точке.

1.3 Дополнительные переменные

В нашем примере мы использовали неравенства (\leq , \geq), кроме того все переменные предполагались неотрицательными. Теперь введем несколько дополнительных понятий.

- (i) **Остаточная переменная.** Неравенства $\mathcal{F} \leq c$ (где \mathcal{F} — линейная комбинация переменных, а c — число) обычно можно интерпретировать как ограничения на использование некоторых ресурсов. В нашем примере неравенство $6x_1 + 4x_2 \leq 24$ задает ограничения на

использование бумаги. Для случая двух переменных неравенство $ax + by \leq c$ можно записать в виде $ax + by + s_1 = c$, где $s_1 \geq 0$, s_1 – **остаточная переменная**. (То есть s_1 можно интерпретировать как остаток соответствующего типа сырья.)

- (ii) **Избыточная переменная.** Аналогично, неравенства $\mathcal{F} \geq c$, интерпретируются, как ограничения на минимальный выпуск некоторой продукции (или минимальное использование некоторого сырья). Для случая двух переменных неравенство $ax + by \geq c$ можно записать в виде $ax + by - S_1 = c$, где $S_1 \geq 0$, S_1 – **избыточная переменная**. (Ее значение показывает избыток продукта по отношению к допустимому минимуму)
- (iii) **Свободная переменная.** Очевидно, допустима ситуация, когда переменная может принимать как положительные, так и отрицательные значения. Пусть переменная x принимает любые значения. В таком случае представим её как $x = x^+ - x^-$, где $x^+ \geq 0$ и $x^- \geq 0$.

1.4 Анализ чувствительности

Сама модель задачи линейного программирования содержит конкретные значения всех параметров. Но в реальной задаче, мы можем не быть полностью уверены в точных значениях параметров. (коэффициентов целевой функции и значений констант в правых частях неравенств ограничений). Соответственно мы хотим некоторый набор условий при выполнении которых решение достигается в той же угловой точке. Такое исследование называется **анализом чувствительности**.

- Изменение коэффициентов целевой функции.

Общий вид задачи: минимизировать (максимизировать) значение $z = ax + by$. Очевидно, что при изменении значений констант a и b решение может переместиться в другую угловую точку. Наша задача – определение интервалов изменения соответствующих параметров при сохранении решения в той же угловой точке.

Вспомним нашу задачу 1 о типографии. Оптимальным решением была точка С. При изменении коэффициентов целевой функции точка С останется точкой оптимума, пока угол наклона прямой $z = c_1x_1 + c_2x_2$

лежит между углами наклона прямых, точкой пересечения которых является точка С. Это прямые

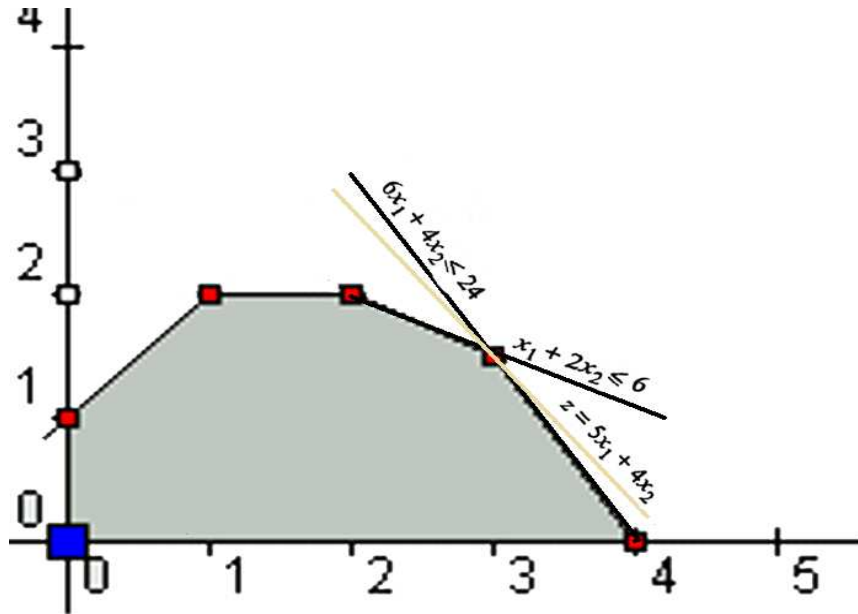
$$6x_1 + 4x_2 \leq 24$$

$$x_1 + 2x_2 \leq 6.$$

То есть при выполнении условия

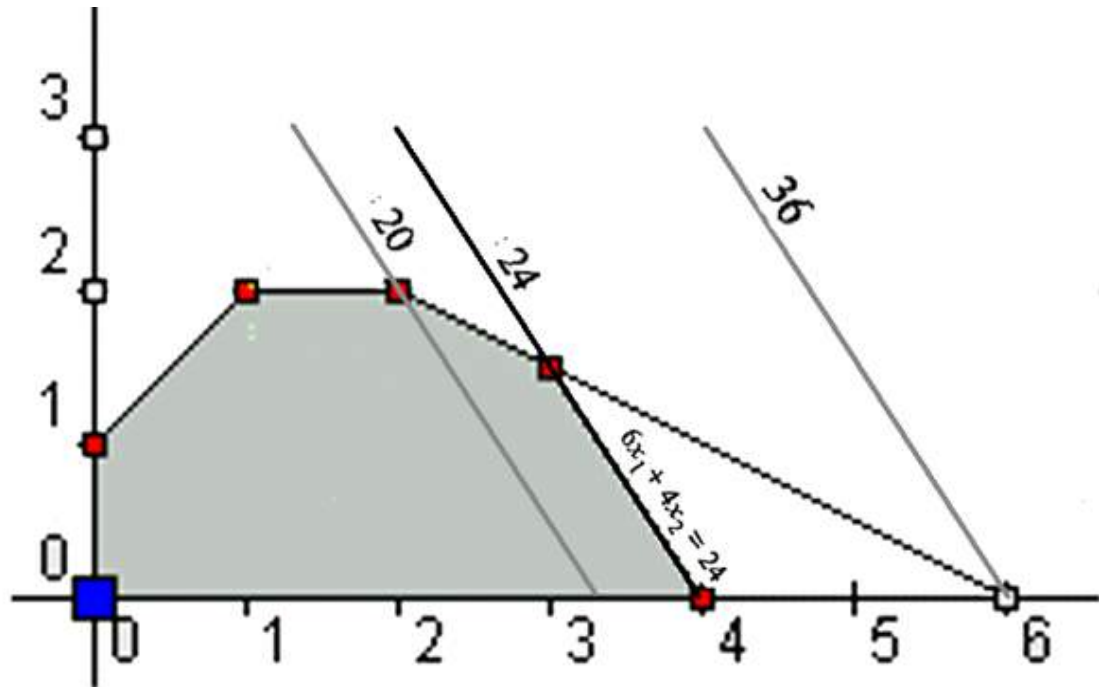
$$\frac{1}{2} \leq \frac{c_1}{c_2} \leq 1\frac{1}{2}$$

Точка С остается точкой оптимума.



- Стоимость ресурсов.

Теперь оценим влияние объема допустимых ресурсов на наше решение, точнее на сохранение точки оптимума. Ресурсами для нашей типографии являются бумага и краска. При изменении количества доступной бумаги точка С смещается вдоль отрезка DG.



Легко показать, что при изменении количества бумаги от 20 до 36 тонн решением остается точка С - точка пересечения прямых заданных ограничениями на ресурс. Отрезок $[20, 36]$ называется **интервалом осуществимости** для ресурса, здесь для бумаги. “Стоимостью” бумаги логично назвать следующую величину

$$cost_1 = \frac{\text{изменение значения целевой функции}}{\text{изменение количества соответствующего ресурса}}$$

на интервале осуществимости. Для бумаги получим 750 рублей за тонну, это означает, что изменение количества доступной бумаги на 1 тонну приведет к изменению значения целевой функции на 750 рублей. Аналогично можно рассчитать стоимость краски.

Ниже мы перейдем к более общему методу решения задач линейного программирования – к симплекс-методу.

2 Симплекс-метод

Из графического способа решения мы узнали, что решением всегда является угловая точка пространства решений. Этот факт важен при построении алгебраического метода решения задач линейного программирования.

2.1 Стандартная форма задачи

Задача дана в **стандартной форме** если она удовлетворяет следующим условиям.

- (i) Все ограничения – равенства с неотрицательной правой частью.
- (ii) Все переменные неотрицательны.
- (iii) Целевая функция минимизируется.

Следующие действия позволяют привести задачу к стандартной форме.

- (i) **Преобразование неравенств в равенства.** Неравенства преобразуются в равенства путем добавления остаточных или избыточных переменных к левой части, как это делалось в 1.3. Кроме того, правую часть можно сделать неотрицательной умножив все неравенство на -1 .
- (ii) **Освобождение задачи от свободных переменных** Свободную переменную x_j представим в виде разности двух неотрицательных переменных

$$x_j = x_j^+ - x_j^-$$

После решения задачи выполняется обратная подстановка.

- (iii) **Преобразование задачи максимизации в задачу минимизации.** Осуществляется минимизация функции $f' = -f$

Рассмотрим пример приведения задачи к стандартной форме.

ЗАДАЧА 1 *Максимизировать*

$$z = 2x_1 + 3x_2 + 5x_3$$

при выполнении условий

$$x_1 + x_2 - x_3 \geq -5$$

$$-6x_1 + 7x_2 - 9x_3 \leq 4$$

$$x_1 + x_2 + 4x_3 = 10$$

$$x_1, x_2 \geq 0$$

x_3 — свободная переменная

Выполним следующие действия

- (i) Вычтем из левой части первого неравенства избыточную переменную и домножим неравенство на -1 .
- (ii) Добавим остаточную переменную к левой части второго неравенства.
- (iii) Выполним замену $x_3 = x_3^+ - x_3^-$.

Таким образом получим следующую форму:

ЗАДАЧА 2 *Максимизировать*

$$z = 2x_1 + 3x_2 + 5x_3^+ - 5x_3^-$$

при выполнении условий

$$-x_1 - x_2 + x_3^+ - x_3^- + s_1 = 5$$

$$-6x_1 + 7x_2 - 9x_3^+ + 9x_3^- + s_2 = 4$$

$$x_1 + x_2 + 4x_3^+ - 4x_3^- = 10$$

$$x_1, x_2, x_3^+, x_3^-, s_1, s_2 \geq 0$$

2.2 Базисные решения

Записанная в стандартной форме задача линейного программирования содержит m линейных равенств и n неизвестных переменных, причем $m < n$. Разделим эти переменные на два множества: $n - m$ равных нулю и m , определенных как решение соответствующей системы. Если это решение единственное – назовем эти переменные **базисными**, а остальные – **небазисными**. В этом случае получившееся решение называется **базисным решением**. Если при этом все переменные неотрицательны, то такое решение называем **допустимым**, а в противном случае – **недопустимым**.

Несложно показать, что для свободной переменной невозможно, чтобы обе переменные ее разложения x_j^+ и x_j^- были базисными. (То есть одна из них обязательно должна быть нулевой). Действительно, если они обе ненулевые, то решение гарантировано не единственное, так как при подстановке $x_j^{+'} = x_j^+ + 1$, $x_j^{-'} = x_j^- + 1$ решение остается истинным.

2.3 Алгоритм симплекс-метода

Доказательство нижеследующего утверждения будет дано позднее.

УТВЕРЖДЕНИЕ 1 *Решение задачи линейного программирования может быть найдено путем перебора всех базисных (допустимых) решений.*

Далее мы построим алгоритм симплекс-метода, который будет перебирать не все базисные решения. Этот алгоритм начинает с некоторого базисного решения и пытается его улучшить. То есть удалив некоторую базисную переменную (она называется **исключаемой** из базиса), и соответственно добавив в базис другую переменную (она называется **вводимой** в базис), мы пытаемся увеличить (уменьшить) значение целевой функции.

Рассмотрим работу этого алгоритма подробнее на примере. Вспомним задачу о типографии и запишем ее в стандартной форме.

Максимизировать

$$z = 5x_1 + 4x_2 + 0s_1 + 0s_2 + 0s_3 + 0s_4$$

при выполнении ограничений:

$6x_2 + 4x_2 + s_1 = 24$	ограничение на бумагу
$x_1 + 2x_2 + s_2 = 6$	ограничение на краску
$x_2 + s_4 = 20$	ограничение на спрос
$x_2 - x_1 + s_3 = 1$	ограничение от минобразования
$x_1, x_2, s_1, s_2, s_3, s_4 \geq 0$	

Здесь мы не будем затрагивать примеры работы алгоритма, а перейдем к его обоснованию. Для подробного ознакомления следует воспользоваться визуализатором работы симплекс-метода. Рассмотрим подробнее результат работы. Мы получили следующую таблицу:

x_1	следует печатать 3000 математических книг в месяц
x_2	следует печатать 1500 художественных книг в месяц
z	доход при этом составляет 21000 рублей

Замечание 1 Мы уже умеем вычислять стоимость ресурсов и рассмотрели проблему чувствительности. Теперь рассмотрим вопрос о статусе ресурсов. Ресурсы бывают **дефицитными** (переменная ассоциируемая с соответствующим ресурсом равно 0) и **недефицитными**. В нашей задаче ограничения на сырье – дефицитны, а ограничения на спрос – нет.

2.4 Искусственные начальные решения

По прочтении предыдущего параграфа возникает вопрос: а как получить это допустимое базисное решение для произвольной задачи? Для этого существуют различные искусственные приемы.

• М-метод

Рассмотрим задачу, записанную в стандартной форме. Теперь в каждое равенство не содержащее дополнительной остаточной переменной введем некоторую новую переменную R_i , которая является искусственной и войдет начальное решение. Так как эта переменная не несет никакого смысла для данной задачи, то следует добиться того чтобы ее значение в результате равнялось нулю (и тогда нам не важен ее смысл, ведь никто не запрещает прибавить к 21 тысяче рублей 0 динозавров). То есть возьмем некий большой штраф M (данный метод также называют **методом больших штрафов**) и для каждой “неосмысленной” переменной прибавим (в случае максимизации) или вычтем (в противном случае)

из целевой функции выражение MR_i . Если размер штрафа достаточно велик, то процесс симплекс-метода приведет к нулевым значениям переменных R_i .

Пример использования данного метода также есть в упомянутом выше визуализаторе.

Замечание 2 *Использование метода больших штрафов может и не привести к обнулению всех новых переменных. Если условие несовместно (исходная задача не имеет допустимого решения, то одна или несколько введенных переменных R_i могут иметь ненулевые значения). Это индикатор некорректной постановки задачи.*

Замечание 3 *Теоретически требуется, чтобы $M \rightarrow \infty$, с учетом условий компьютерной реализации M должна быть достаточно велика. При этом могут возникать некоторые проблемы, вызванные возможной потерей точности при операциях над выражениями содержащими как большие так и малые числа. По этой причине в настоящее время чаще используется описанный ниже двухэтапный метод позволяющий избежать этой ошибки.*

- **Двухэтапный метод**

Дабы избежать описанных выше проблем разработан новый метод, заключающийся в следующем.

- Первый шаг делается по аналогии с М-методом. Записываем задачу в стандартной форме и добавляем искусственные переменные. Решаем задачу минимизации суммы искусственных переменных, при удовлетворении исходным ограничениям. Если минимальное значение этой искусственной целевой функции больше нуля то задача не имеет допустимых решений, в противном случае переходим ко второму этапу.
- Базисное решение полученное на предыдущем этапе используется для решения исходной задачи.

Этот метод несколько сложнее предыдущего, так как в ходе его использования может возникнуть другая проблема. Если все искусственные переменные в конце первого этапа небазисные, то имеет смысл просто удалить их. Если же какие-то из них остались в базисе (но имеют

нулевые значения) то необходимо действовать по-другому. Правило заключается в следующем: нужно оставлять искусственную переменную в базисе, если значение в ведущем столбце ненулевое. Действительно, пусть искусственная базисная переменная присутствует и она нулевая, то она не станет положительной. Если коэффициент искусственной переменной в ведущем столбце положителен, то на следующей итерации переменная станет небазисной. Если отрицательный – то мы просто не будем брать то отношение, которое ассоциируется с искусственной переменной, чтобы значение не стало положительным.

2.5 Особые случаи

- **Вырожденность**

Вырожденный случай – случай, в котором одна из базисных переменных имеет нулевое значение. Что означает, что в исходной задаче присутствует избыточное ограничение.

- **Альтернативные оптимальные решения**

Если гиперплоскости целевой функции и ограничивающего неравенства параллельны, то оптимальных решений может оказаться бесконечно много. Тогда они называются альтернативными оптимальными решениями. В приложениях такие решения позволяют сделать некоторый выбор, сохраняя при этом оптимальность условий.

- **Неограниченные решения**

Иногда, обычно в случае некорректной постановки задачи (не учтен некоторые параметры) решение может неограниченно возрастать (убывать – в случае минимизации). Если на некоторой итерации коэффициенты в ограничениях для небазисной переменной неположительны, а коэффициент в z -строке отрицателен, то значение целевой функции неограничено (в случае задачи максимизации).

- **Отсутствие допустимых решений**

Возможны также случаи, когда допустимых решений нет. То есть области заданные неравенствами имеют пустое пересечение. Это означает, что задача плохо сформулирована (или действительно не имеет решений :).

3 Теория симплекс-метода

3.1 Матричное представление задачи

Ниже мы обоснуем приведенный в предыдущей главе метод. Для начала нам понадобится другое (матричное) представление задачи.

Мы имеем m ограничений на n переменных

$$(\mathbf{A}, \mathbf{E})\mathbf{X} = \mathbf{b}, \mathbf{X} \geq \mathbf{0}.$$

и перед нами стоит задача минимизировать или максимизировать

$$z = \mathbf{C}\mathbf{X}$$

при этом задано начальное допустимое базисное решение.

Здесь \mathbf{E} – единичная матрица, под записью (\mathbf{A}, \mathbf{E}) будем понимать матрицу, полученную “дописыванием матрицы \mathbf{E} справа к матрице \mathbf{A} ”, далее будем обозначать $\mathbf{I} := (\mathbf{A}, \mathbf{E})$. $\mathbf{X} = (x_1, \dots, x_n)^T$, $\mathbf{C} = (c_1, \dots, c_n)$

Систему линейных ограничений (ограничения записаны в виде равенств $\mathbf{I}\mathbf{X} = \mathbf{b}$), можно записать и как

$$\sum_{j=1}^n x_j \mathbf{P}_j = \mathbf{b},$$

где \mathbf{P}_j – вектор, j – й столбец матрицы \mathbf{I} . Набор m из этих векторов является **базисом**, если определитель матрицы \mathbf{B} , состоящей из них отличен от 0.

Пусть $\det(\mathbf{B}) \neq 0$, тогда существует обратная матрица \mathbf{B}^{-1} . Для системы можно выбрать набор из m векторов (базисный), и соответственно набор $\mathbf{X}_\mathbf{B}$ из m элементов вектора \mathbf{X} . Всем остальным элементам вектора \mathbf{X} присвоим нулевые значения. Тогда решение данной системы ($\mathbf{B}\mathbf{X}_\mathbf{B} = \mathbf{b}$) будет единственным:

$$\mathbf{X}_\mathbf{B} = \mathbf{B}^{-1}\mathbf{b}.$$

$\mathbf{X}_\mathbf{B}$ – **базисное решение** системы $\mathbf{I}\mathbf{X} = \mathbf{b}$. Это решение допустимое, если $\mathbf{B}^{-1}\mathbf{b} \geq \mathbf{0}$, где $\mathbf{0} = (0, \dots, 0)^T$.

Требуемое представление задачи получается, если заданное начальное допустимое базисное решение состоит из переменных x_{n-m+1}, \dots, x_n (в противном случае мы можем просто перенумеровать переменные).

$$\mathbf{A} = \begin{bmatrix} a_{11} & \cdots & a_{1,n-m} \\ \vdots & \vdots & \vdots \\ a_{m1} & \cdots & a_{n,n-m} \end{bmatrix} \mathbf{B} = \begin{bmatrix} b_1 \\ \vdots \\ b_n \end{bmatrix}$$

Пример 1 Минимизировать $z = -2x_1 - 4x_2$ при ограничениях

$$3x_1 + 4x_2 \leq 1700$$

$$2x_1 + 5x_2 \leq 1600$$

$$x_1, x_2 \geq 0$$

После приведения к стандартной форме мы получаем задачу минимизации функции $z = -2x_1 - 4x_2$ при ограничениях

$$\begin{bmatrix} 3 & 4 & 1 & 0 \\ 2 & 5 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} 1700 \\ 1600 \end{bmatrix}$$

$$x_j \geq 0$$

В этой задаче 4 переменных (m) и 2 ограничения (n). Переменные $x_{3,4}$ – дополнительные. Соответственно в матричном виде:

$$\mathbf{X} = (x_1, x_2, x_3, x_4)^T,$$

$$\mathbf{C} = (-2, -4, 0, 0),$$

$$\mathbf{b} = (1700, 1600)^T,$$

$$\mathbf{A} = \begin{bmatrix} 3 & 4 \\ 2 & 5 \end{bmatrix}$$

$$\mathbf{I} = (\mathbf{A}, \mathbf{E}) = \begin{bmatrix} 3 & 4 & 1 & 0 \\ 2 & 5 & 0 & 1 \end{bmatrix}$$

3.2 Основные теоремы

Определение 1 Отрезком XY называется множество точек, определяемых соотношением $z = \lambda x + (1 - \lambda)y, \lambda \in [0, 1]$.

Определение 2 Множество называется **выпуклым**, если любые две его точки таковы, что соединяющих их отрезок содержит только точки множества.

Определение 3 Выпуклой комбинацией точек p_1, \dots, p_k метрического пространства называется точка

$$z = \lambda_1 p_1 + \lambda_2 p_2 + \dots + \lambda_k p_k,$$

$$\lambda_i \geq 0, \sum_{i=1}^k \lambda_i = 1.$$

Заметим, что выпуклая комбинация любых двух точек выпуклого множества – точка этого множества.

Определение 4 Крайняя точка выпуклого множества – это точка, не представимая в виде выпуклой комбинации каких-либо двух различных точек данного множества.

Рассмотрим множество допустимых решений задачи линейного программирования

$$Q = \{X | IX = b, X \geq 0\}.$$

Теорема 1 Если ограничения имеют допустимое решение, то они имеют и базисное решение.

Доказательство: Докажем это, построив базисное допустимое решение. Пусть в допустимом решении $n - r$ переменных равны 0, а остальные положительны. Тогда без потери общности можно считать нулевыми переменные x_{r+1}, \dots, x_n . А значит:

$$\sum_{j=1}^r x_j P_j = b$$

Если P_1, P_2, \dots, P_r линейно независимы, то $r \leq m$ (m – ранг матрицы I) и решение является базисным допустимым решением.

Если $\mathbf{P}_1, \mathbf{P}_2, \dots, \mathbf{P}_r$ линейно зависимы, то можно составить линейную комбинацию $\sum_{j=1}^r \mathbf{P}_j \alpha_j = \mathbf{0}$, причем не все $\alpha_j = 0$. Пусть $\alpha_k > 0$, тогда

$$\sum_{j \neq k}^r (x_j - \frac{\alpha_j x_k}{\alpha_k}) \mathbf{P}_j = \mathbf{b}.$$

Выбрав k так, что

$$\frac{x_k}{\alpha_k} = \min_{j=1, \dots, r} \frac{x_j}{\alpha_j}; \alpha_j > 0,$$

то значения

$$x'_j = x_j - \alpha_j \frac{x_k}{\alpha_k}, j = 1, \dots, r (j \neq k),$$

$$x'_j = 0, j = k, r+1, \dots, n,$$

неотрицательны и поэтому являются допустимыми решениями, в которых по крайней мере $r-1$ переменная имеет строго положительное значение. Продолжая рассуждения таким же образом (т. е. уменьшая количество строго положительных переменных), в конце концов придем к ситуации, при которой $r \leq m$, т. е. получим базисное допустимое решение.

Теорема 2 Множество Q – выпуклое.

Доказательство: Пусть точки \mathbf{x} и \mathbf{y} принадлежат допустимой области (т. е. $\mathbf{Ax} = \mathbf{b}, \mathbf{Ay} = \mathbf{b}, \mathbf{x} \geq \mathbf{0}, \mathbf{y} \geq \mathbf{0}$), тогда, если $\mathbf{z} = \lambda \mathbf{x} + (1 - \lambda) \mathbf{y}, \lambda \in [0, 1]$, то имеем:

$$\mathbf{Az} = \mathbf{A}(\lambda \mathbf{x} + (1 - \lambda) \mathbf{y}) = \lambda \mathbf{Ax} + (1 - \lambda) \mathbf{Ay} = \lambda \mathbf{b} + (1 - \lambda) \mathbf{b} = \mathbf{b},$$

т. е. $\mathbf{z} \in Q$, что и означает выпуклость Q .

Замечание 1 В выпуклом множестве любая содержащаяся в нем точка может быть представлена линейной комбинацией крайних точек.

Теорема 3 Точка \mathbf{x} – крайняя точка пространства решений Q , если и только если \mathbf{x} – базисное решение системы $\mathbf{Ix} = \mathbf{b}$, причем $\mathbf{x} \geq \mathbf{0}$.

Доказательство: Пусть $\mathbf{x}^0 = (x_1, \dots, x_m, 0, \dots, 0)^T$ – базисное допустимое решение. Тогда \mathbf{x}^0 является единственным решением уравнения $\mathbf{Ax} = \mathbf{b}$, где $\mathbf{x} \geq \mathbf{0}$ и последние $n - m$ координат вектора \mathbf{x} равны 0.

Если \mathbf{x}^0 – не вершина, то можно найти две другие точки $\mathbf{u} \in Q$ и $\mathbf{v} \in Q$, такие, что $\mathbf{x}^0 = \lambda \mathbf{u} + (1 - \lambda) \mathbf{v}$ для некоторого $\lambda \in (0, 1)$. Но тогда для последних $n - m$ координат имеем

$$\lambda u_i + (1 - \lambda) v_i = 0, i = m + 1, \dots, n.$$

Откуда, из неотрицательности всех "компонент" левой части $(\lambda, 1 - \lambda, \mathbf{u}_i, \mathbf{v}_i)$ следует, что $u_i = v_i = 0, i = m + 1, \dots, n$. А значит, в силу того, что \mathbf{x}^0 – базисное решение, получаем противоречие с выбором \mathbf{u} и \mathbf{v} , что и доказывает, что \mathbf{x}^0 – вершина.

Пусть теперь \mathbf{x}^0 – вершина допустимой области. Пусть r ее координат строго положительны. Покажем, что r не превосходит m , что и означает, что \mathbf{x}^0 – базисное допустимое решение. Пусть $r > m$, т. е. $x_j > 0, j = 1, \dots, r$. Тогда $\mathbf{P}_1, \dots, \mathbf{P}_r$ (столбцы матрицы \mathbf{I}) оказываются линейно независимыми. Найдем такие α_k , не все равные нулю, что $\sum_{j=1}^r \alpha_j \mathbf{P}_j = \mathbf{0}$. возьмем ρ такое, что

$$0 < \rho < \min_j \frac{x_j^0}{|\alpha_j|}, \alpha_j \neq 0.$$

А тогда векторы

$$\mathbf{x}^1 = \mathbf{x}^0 + \rho \mathbf{a}, \mathbf{x}^2 = \mathbf{x}^0 - \rho \mathbf{a},$$

где

$$\mathbf{a} = (\alpha_1, \dots, \alpha_r, 0, \dots, 0)^T,$$

удовлетворяют условиям области Q , действительно: $\mathbf{A} \mathbf{x}^{1,2} = \mathbf{A}(\mathbf{x}^0 + \rho \mathbf{a}) = \mathbf{A} \mathbf{x}^0 + \rho \mathbf{A} \mathbf{a} = \mathbf{b}$. А это означает, что $\mathbf{x}^0 = 1/2 \cdot (\mathbf{x}^1 + \mathbf{x}^2)$, т. е. \mathbf{x}^0 не является вершиной, что приводит к противоречию и доказывает, что $r \leq m$.

Теорема 4 *Оптимальное решение задачи линейного программирования достигается в крайней точке множества Q . (В случае, если это решение конечно).*

Доказательство: Не уменьшая общности будем считать, что перед нами стоит задача нахождения минимального решения. Пусть допустимым базисным решениям соответствуют точки $\mathbf{x}^1, \dots, \mathbf{x}^k$, тогда (с учетом замечания), любая точка $\mathbf{x} \in Q$ представима как линейная комбинация

этих точек. Пусть целевая функция $z = \mathbf{C}\mathbf{x}$ достигает минимума в некоторой точке \mathbf{x} , причем $\mathbf{x} = \sum_{i=1}^k \alpha_i \mathbf{x}^i$, $\sum_{i=1}^k \alpha_i = 1$. Но тогда

$$z = \mathbf{C}\mathbf{x} = \sum_{i=1}^k \alpha_i \mathbf{C}\mathbf{x}^i \geq \sum_{i=1}^k \alpha_i \mathbf{C}\mathbf{x}^0 = \mathbf{C}\mathbf{x}^0,$$

где $\mathbf{x}^0 = \min_{j=1,\dots,k} \mathbf{C}\mathbf{x}^j$. А это и означает, что оптимальное решение достигается в крайней точке множества Q – в точке \mathbf{x}^0

3.3 Обоснование симплекс-метода

Приведенные теоремы доказывают, что конечное оптимальное решение достигается в крайней точке пространства решений. Все точки этого пространства можно определить как базисные решения системы $\mathbf{I}\mathbf{X} = \mathbf{b}$, $\mathbf{X} \geq \mathbf{0}$, которые, таким образом, становятся главным объектом наших исследований.

При выполнении симплекс-метода мы начинали с допустимого базисного решения, затем переходили к следующему допустимому базисному решению, которое улучшает (по крайней мере не ухудшает) значение целевой функции, и так до тех пор, пока не будет достигнуто оптимальное решение. Таким образом, допустимое базисное решение – тот принципиальный элемент в симплекс-методе, вокруг которого выполняются все вычисления в симплекс-таблице. С этой точки зрения очевидна необходимость представления симплекс-таблицы в матричной форме.

В стандартной задаче линейного программирования (максимизировать $z = \mathbf{C}\mathbf{X}$ при ограничениях $\mathbf{I}\mathbf{X} = \mathbf{b}$, $\mathbf{X} \geq \mathbf{0}$) разобьем вектор \mathbf{X} на вектора \mathbf{X}_1 и \mathbf{X}_2 , причем \mathbf{X}_2 соответствует начальному базису \mathbf{B} (является начальным допустимым решением). Вектор \mathbf{C} также разделим на вектора \mathbf{C}_1 и \mathbf{C}_2 в соответствии с делением вектора \mathbf{X} . Тогда в матричном виде

$$\begin{bmatrix} 1 & -C_1 & -C_2 \\ 0 & A & E \end{bmatrix} \begin{bmatrix} z \\ X_1 \\ X_2 \end{bmatrix} = \begin{bmatrix} 0 \\ b \end{bmatrix}$$

Для текущей итерации \mathbf{X}_B – базисный вектор переменных, \mathbf{C}_B – вектор коэффициентов целевой функции, соответствующих этому базису. Так как небазисные переменные равны нулю, то задача эквивалентна задаче с целевой функцией $z = \mathbf{C}_B \mathbf{X}_B$ и ограничениями $\mathbf{B}\mathbf{X}_B = \mathbf{b}$, где решение

удовлетворяет уравнению $\mathbf{X}_B = \mathbf{B}^{-1}\mathbf{b}$, записываемому как

$$\begin{bmatrix} z \\ X_B \end{bmatrix} = \begin{bmatrix} 1 & -C_B \\ 0 & B \end{bmatrix}^{-1} \begin{bmatrix} 0 \\ b \end{bmatrix} = \begin{bmatrix} 1 & -C_B B^{-1} \\ 0 & B^{-1} \end{bmatrix} \begin{bmatrix} 0 \\ b \end{bmatrix} \begin{bmatrix} C_B B^{-1} b \\ B^{-1} b \end{bmatrix}$$

Симплекс-таблица получается из исходной задачи путем вычисления по следующей формуле

$$\begin{bmatrix} 1 & C_B B^{-1} \\ 0 & B^{-1} \end{bmatrix} \begin{bmatrix} 1 & -C_1 & -C_2 \\ 0 & A & E \end{bmatrix} \begin{bmatrix} z \\ X_1 \\ X_2 \end{bmatrix} = \begin{bmatrix} 1 & C_B B^{-1} \\ 0 & B^{-1} \end{bmatrix} \begin{bmatrix} 0 \\ b \end{bmatrix}$$

то есть имеет вид

Базис	\mathbf{X}_1	\mathbf{X}_2	Решение
z	$\mathbf{C}_B \mathbf{B}^{-1} \mathbf{A} - \mathbf{C}_1$	$\mathbf{C}_B \mathbf{B}^{-1} - \mathbf{C}_2$	$\mathbf{C}_B \mathbf{B}^{-1} \mathbf{b}$
\mathbf{X}_B	$\mathbf{B}^{-1} \mathbf{A}$	\mathbf{B}^{-1}	$\mathbf{B}^{-1} \mathbf{b}$

Таким образом вычислений требует лишь матрица \mathbf{B}^{-1} , она строится на каждом шаге алгоритма на основании принятого решения об исключаемом базисном и вводимом небазисном векторах.

Чтобы выяснить, какой вектор нам наиболее "выгодно" вводить в базис, рассмотрим коэффициент при переменной x_j в z -строке таблицы, который равен $z_j - c_j = \mathbf{C}_B \mathbf{B}^{-1} \mathbf{P}_j - c_j$, для базисных переменных эта разность равна нулю. Пусть Ω – множество небазисных переменных, тогда для целевой функции имеем

$$z + \sum_{j \in \Omega} (z_j - c_j) x_j = \mathbf{C}_B \mathbf{B}^{-1} \mathbf{b}.$$

Таким образом ясно, что увеличение значения переменной x_j приводит к увеличению целевой функции z выше текущего значения (равного $\mathbf{C}_B \mathbf{B}^{-1} \mathbf{b}$), если $z_j - c_j \leq 0$ (условие оптимальности).

Теперь выберем исключаемый вектор. Проверим равенство соответствующее i – й базисной переменной.

$$(\mathbf{X}_B)_i + \sum_{j \in \Omega} (\mathbf{B}^{-1} \mathbf{P}_j)_i x_j = (\mathbf{B}^{-1} \mathbf{b})_i$$

Пусть \mathbf{P}_k – вводимый вектор, x_k – вводимая переменная. Все прочие небазисные переменные равны 0, значит мы можем переписать ограничение $(\mathbf{X}_B)_i = (\mathbf{B}^{-1} \mathbf{b})_i - (\mathbf{B}^{-1} \mathbf{P}_k)_i x_k$, что означает, что базисная переменная $(\mathbf{X}_B)_i$ не станет отрицательной, если $\forall i ((\mathbf{B}^{-1} \mathbf{b})_i - (\mathbf{B}^{-1} \mathbf{P}_k)_i x_k \geq 0)$

(условие допустимости). Выпишем максимальное значение вводимой переменной

$$x_k = \min_i \left\{ \frac{(\mathbf{B}^{-1}\mathbf{b})_i}{(\mathbf{B}^{-1}\mathbf{P}_k)_i} \mid (\mathbf{B}^{-1}\mathbf{P}_k)_i > 0 \right\}$$

Базисная переменная, на которой минимум достигается, оказывается исключаемой.

4 Модификации симплекс-метода

После того как мы обсудили теорию симплекс-метода имеет смысл описать некоторые его модификации.

4.1 Мультипликативное представление обратной матрицы (модифицированный симплекс-метод)

Мы не будем производить преобразования строк методом Гаусса-Жордана, как это делалось в обычном симплекс-методе, а будем использовать обратную матрицу \mathbf{B}^{-1} . Последовательные базисы при итерациях симплекс-метода отличаются одним столбцом (идет замена исключаемого вектора вводимым). Положим $\mathbf{B}_{\text{next}}^{-1} = \mathbf{L}\mathbf{B}^{-1}$, и попытаемся вычислить матрицу \mathbf{L} . \mathbf{P}_j и \mathbf{P}_r – вводимый и исключаемый из базиса вектора, тогда \mathbf{L} – получается из единичной матрицы \mathbf{E} заменой r столбца на

$$\xi = \frac{1}{(\mathbf{B}^{-1}\mathbf{P}_j)_r} \begin{bmatrix} -(B^{-1}P_j)_1 \\ \vdots \\ -(B^{-1}P_j)_{r-1} \\ +1 \\ -(B^{-1}P_j)_{r+1} \\ \vdots \\ -(B^{-1}P_j)_m \end{bmatrix}$$

Доказательство: Рассмотрим матрицу \mathbf{F} , получаемую из \mathbf{E} заменой одного столбца $\mathbf{F} = (\mathbf{e}_1, \dots, \mathbf{e}_{r-1}, \mathbf{B}^{-1}\mathbf{P}_j, \mathbf{e}_{r+1}, \dots, \mathbf{e}_m)$. Легко проверить, что $\mathbf{B}_{\text{next}} = \mathbf{B}\mathbf{F}$. Теперь осталось,

$$\mathbf{B}_{\text{next}}^{-1} = (\mathbf{B}\mathbf{F})^{-1} = \mathbf{F}^{-1}\mathbf{B}^{-1},$$

где $\mathbf{L} = \mathbf{F}^{-1}$.

В нашем алгоритме начальный базис есть единичная матрица \mathbf{E} , тогда матрица B_i^{-1} на i итерации есть $\mathbf{B}_i^{-1} = \mathbf{L}_i \dots \mathbf{L}_1 \mathbf{E}$. Теперь мы не должны вычислять обратных матриц.

У модифицированного симплекс метода следующие преимущества:

- (i) Число арифметических действий меньше чем в обычном методе, например за счет разреженности матриц.
- (ii) Ошибку вычислений легко контролировать, так как основная погрешность приходится на вычисления обратной матрицы.

Пример 1 Еще раз вспомним о задаче 1 о типографии.

Максимизировать $z = (5, 4, 0, 0, 0, 0)(x_1, \dots, x_6)^T$ при ограничениях

$$\begin{bmatrix} 6 & 4 & 1 & 0 & 0 & 0 \\ 1 & 2 & 0 & 1 & 0 & 0 \\ -1 & 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ \vdots \\ x_6 \end{bmatrix} = \begin{bmatrix} 24 \\ 6 \\ 1 \\ 2 \end{bmatrix}$$

$$x_1, \dots, x_6 \geq 0$$

Пусть коэффициенты целевой функции $\mathbf{C} = (c_1, \dots, c_6)$, а столбцы матрицы ограничений $(\mathbf{A}, \mathbf{E}) = \mathbf{I} = (\mathbf{P}_1, \dots, \mathbf{P}_6)$. \mathbf{b} – вектор правых частей. Теперь будем действовать по итерациям:

• Итерация 0

$$\mathbf{X}_{\mathbf{B}_0} = (x_3, x_4, x_5, x_6)^T, \mathbf{C}_{\mathbf{B}_0} = \mathbf{0}.$$

$$\mathbf{B}_0 = (\mathbf{P}_3, \mathbf{P}_4, \mathbf{P}_5, \mathbf{P}_6) = \mathbf{L}, \mathbf{B}_0^{-1} = \mathbf{E}.$$

$$\text{Тогда } \mathbf{X}_{\mathbf{B}_0} = \mathbf{B}_0^{-1} \mathbf{b} = (24, 6, 1, 2)^T, z = \mathbf{C}_{\mathbf{B}_0} \mathbf{X}_{\mathbf{B}_0} = 0$$

Условие оптимальности

$$\mathbf{C}_{\mathbf{B}_0} \mathbf{B}_0^{-1} = \mathbf{0}, \{z_j - c_j\}_{j=1,2} = \mathbf{C}_{\mathbf{B}_0} \mathbf{B}_0^{-1} (\mathbf{P}_1, \mathbf{P}_2) - (c_1, c_2) = (-5, -4).$$

откуда вектор \mathbf{P}_1 – вводимый.

Условие допустимости

$$\mathbf{X}_{\mathbf{B}_0} = (\mathbf{x}_3, \mathbf{x}_4, \mathbf{x}_5, \mathbf{x}_6)^T = (24, 6, 1, 2)^T.$$

$$B_0^{-1}P_1 = (6, 1, -1, 0)^T.$$

$$x_1 = \min \left\{ \frac{24}{6}, \frac{6}{1} \right\} = 4$$

и вектор P_3 – исключаемый.

• Итерация 1

$$X_{B_1} = (x_1, x_4, x_5, x_6)^T, C_{B_1} = (5, 0, 0, 0).$$

$$B_1 = (P_1, P_4, P_5, P_6).$$

Вычислим B_1^{-1} , мы знаем что $B_0^{-1}P_1 = (6, 1, -1, 0)^T$. В матрице B_0 заменялся первый столбец ($r = 1$).

$$B_1^{-1} = LB_0^{-1} = LE = L = \begin{bmatrix} \frac{1}{6} & 0 & 0 & 0 \\ -\frac{1}{6} & 1 & 0 & 0 \\ \frac{1}{6} & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\text{Тогда } X_{B_1} = B_1^{-1}b = (4, 2, 5, 2)^T, z = C_{B_1}X_{B_1} = 20$$

Условие оптимальности

$$C_{B_1}B_1^{-1} = \left(\frac{5}{6}, 0, 0, 0 \right).$$

$$\{z_j - c_j\}_{j=2,3} = C_{B_1}B_1^{-1} - (c_2, c_3) = \left(-\frac{2}{3}, \frac{5}{6} \right)$$

Вектор P_2 – вводимый.

Условие допустимости

$$X_{B_1} = (x_1, x_4, x_5, x_6)^T = (4, 2, 5, 2)^T.$$

$$B_1^{-1}P_2 = \left(\frac{2}{3}, \frac{4}{3}, \frac{5}{3}, 1 \right)$$

$$x_2 = \min \left\{ 6, \frac{3}{3}, 3, 2 \right\} = \frac{3}{2}$$

Вектор P_4 – исключаемый.

• Итерация 2

$$\mathbf{X}_{\mathbf{B}_2} = (x_1, x_2, x_5, x_6)^T, \mathbf{C}_{\mathbf{B}_2} = (5, 4, 0, 0).$$

$$\mathbf{B}_2 = (\mathbf{P}_1, \mathbf{P}_2, \mathbf{P}_5, \mathbf{P}_6).$$

Для вычисления \mathbf{B}_2^{-1} имеем $\mathbf{B}_1^{-1}\mathbf{P}_2 = (\frac{2}{3}, \frac{4}{3}, \frac{5}{3}, 1)^T$, также $r = 2$, значит

$$\mathbf{B}_2^{-1} = \mathbf{L}\mathbf{B}_1^{-1} = \begin{bmatrix} 1 & -\frac{2}{4} & 0 & 0 \\ 0 & \frac{3}{4} & 0 & 0 \\ 0 & -\frac{5}{4} & 1 & 0 \\ 0 & -\frac{3}{4} & 0 & 1 \end{bmatrix} \begin{bmatrix} \frac{1}{6} & 0 & 0 & 0 \\ -\frac{1}{6} & 1 & 0 & 0 \\ \frac{1}{6} & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} \frac{1}{4} & -\frac{1}{2} & 0 & 0 \\ -\frac{1}{8} & \frac{3}{4} & 0 & 0 \\ \frac{3}{8} & -\frac{5}{4} & 1 & 0 \\ \frac{1}{8} & -\frac{3}{4} & 0 & 1 \end{bmatrix}$$

$$\text{Тогда } \mathbf{X}_{\mathbf{B}_2} = \mathbf{B}_2^{-1}\mathbf{b} = (3, \frac{3}{2}, \frac{5}{2}, \frac{1}{2})^T, z = \mathbf{C}_{\mathbf{B}_2}\mathbf{X}_{\mathbf{B}_2} = 21$$

Условие оптимальности

$$\mathbf{C}_{\mathbf{B}_2}\mathbf{B}_2^{-1} = \left(\frac{3}{4}, \frac{1}{2}, 0, 0\right).$$

$$\{z_j - c_j\}_{j=3,4} = \mathbf{C}_{\mathbf{B}_2}\mathbf{B}_2^{-1} - (c_3, c_4) = \left(\frac{3}{4}, \frac{1}{2}\right)$$

Таким образом решение $\mathbf{X}_{\mathbf{B}_2}$ оптимально:

$$x_1 = 3, x_2 = 1.5, z = 21$$

4.2 Метод решения задач с ограниченными переменными

В некоторых задачах линейного программирования на значения переменных могут накладываться явные положительные верхние и нижние ограничения (например, минимальные и максимальные значения спроса на определенную продукцию).

Рассмотрим простой случай, когда заданы только нижние границы на значения переменных ($\mathbf{X} \geq \mathbf{L}$). В этом случае подстановка $\mathbf{X} = \mathbf{L} + \mathbf{X}'$, $\mathbf{X}' \geq \mathbf{0}$ сводит задачу к обычной задаче линейного программирования, результат получается обратной подстановкой.

Если же в задаче присутствуют верхние границы ($\mathbf{X} \leq \mathbf{U}$), то прямая замена переменных ($X = U - X'', X'' \geq 0$) невозможна, т. к. обратная подстановка не гарантирует неотрицательность \mathbf{X} . Для решения задачи ЛП с верхними границами мы применим следующую модификацию для симплекс-метода: при проверки условия допустимости мы будем дополнительно учитывать ограничение $\mathbf{X} \leq \mathbf{U}$.

Из уравнения ограничения относительно базисной переменной x_i (заметим, что оно верно, только если все небазисные переменные равны нулю)

$$(\mathbf{X}_\mathbf{B})_i = (\mathbf{B}^{-1}\mathbf{b})_i - (\mathbf{B}^{-1}\mathbf{P}_j)_i x_j$$

мы получаем требуемые условия:

1) Базисная переменная $(\mathbf{X}_\mathbf{B})_i$ должна остаться неотрицательной. Это требование будет заведомо выполняться, если

$$x_j \leq \theta_1 = \min_i \left\{ \frac{(\mathbf{B}^{-1}\mathbf{b})_i}{(\mathbf{B}^{-1}\mathbf{P}_j)_i} \mid (\mathbf{B}^{-1}\mathbf{P}_j)_i > 0 \right\}.$$

2) Базисная переменная $(\mathbf{X}_\mathbf{B})_i$ не должна превышать своей верхней границы $((\mathbf{X}_\mathbf{B})_i \leq (\mathbf{U}_\mathbf{B})_i)$, что будет иметь место, если

$$x_j \leq \theta_2 = \min_i \left\{ \frac{(\mathbf{B}^{-1}\mathbf{b})_i - (\mathbf{U}_\mathbf{B})_i}{(\mathbf{B}^{-1}\mathbf{P}_j)_i} \mid (\mathbf{B}^{-1}\mathbf{P}_j)_i < 0 \right\}.$$

3) Вводимая переменная x_j не может принять значения, большего своей верхней границы, т. е. должно выполняться неравенство $x_j \leq u_j$.

Все три неравенства относительно x_j можно обобщить в виде следующего условия:

$$x_j = \min \theta_1, \theta_2, u_j.$$

Базис, формируемый для следующей итерации симплекс-метода, зависит от того, какое значение $(\theta_1, \theta_2, u_j)$ примет переменная x_j . Предполагая, что $(\mathbf{X}_\mathbf{B})_r$ - исключаемая переменная, имеем следующие правила формирования базиса:

1) Если $x_j = \theta_1$, то переменная $(\mathbf{X}_\mathbf{B})_r$ исключается из базиса и принимает нулевое значение. Новый базис формируется точно так же, как в обычном симплекс-методе с вводимой переменной x_j и исключаемой $(\mathbf{X}_\mathbf{B})_r$.

2) Если $x_j = \theta_2$, переменная $(\mathbf{X}_\mathbf{B})_r$ исключается из базиса и принимает значение своей верхней границы. Новый базис формируется точно так же, как в обычном симплекс-методе, но с учетом того, что переменная $(\mathbf{X}_\mathbf{B})_r$ равна верхней границе (а не нулю). Кроме того, поскольку значения θ_1 и θ_2 получены при условии, что все небазисные переменные равны нулю, для корректной работы алгоритма в дальнейшем необходимо выполнить преобразование над новой небазисной переменной $(\mathbf{X}_\mathbf{B})_r$, чтобы она также приняла нулевое значение. Это достигается с помощью

подстановки $(\mathbf{X}_B)_r = (\mathbf{U}_B)_r - (\mathbf{X}_B)'_r$, где $(\mathbf{X}_B)'_r \geq 0$. Не имеет особого значения, когда выполняется эта подстановка: до или после вычисления нового базиса.

3) Если $x_j = u_j$, то базисный вектор \mathbf{X}_B остается неизменным, поскольку в данном случае никакая из текущих базисных переменных не принимает ни нулевого значения, ни значения своей верхней границы. Таким образом, переменная x_j остается небазисной, но со значением своего верхнего предела. После подстановки $x_j = u_j - x'_j$ выполняется следующая итерация симплекс-метода.

В случае равенства значений θ_1, θ_2 и u_j выбор правила, в соответствии с которым переходим к следующей итерации симплекс-метода, произволен. Вместе с тем предпочтительнее использование третьего правила, так как в этом случае требуется выполнить меньший объем вычислений.

Замечание 1 Подстановка $x_j = u_j - x'_j$ изменяет исходные значения c_j, \mathbf{P}_j и \mathbf{b} на $c'_j = -c_j, \mathbf{P}'_j = -\mathbf{P}_j$ и $\mathbf{b}' = \mathbf{b} - u_j \mathbf{P}_j$. Это означает, что при выполнении модифицированного симплекс-метода на каждой итерации все вычисления (таких величин, как $\mathbf{B}^{-1}, \mathbf{X}_B$ и $z_j - c_j$) должны основываться на измененных значениях \mathbf{C}, \mathbf{A} и \mathbf{b} .

Пример 2 Решим следующую задачу линейного программирования вышеописанным методом решения задач с ограниченными переменными.

Максимизировать $z = 3x_1 + 5y + 2x_3$ при ограничениях

$$\begin{aligned} x_1 + y + 2x_3 &\leq 14, \\ 2x_1 + 4y + 3x_3 &\leq 43, \\ 0 \leq x_1 \leq 4, 7 \leq y \leq 10, 0 \leq x_3 \leq 3. \end{aligned}$$

Так как переменная y ограничена положительными константами не только сверху, но и снизу, делаем замену $y = x_2 + 7$, где $0 \leq x_2 \leq 10 - 7 = 3$.

• Итерация 0

Базис	x_1	x_2	x_3	x_4	x_5	Решение
z	-3	-5	-2	0	0	35
x_4	1	1	2	1	0	7
x_5	2	4	3	0	1	15

Имеем, $\mathbf{X}_B = \mathbf{B}^{-1}\mathbf{b} = (7, 15)^T$. Принимая переменную x_2 в качестве вводимой в базис переменной, получаем $\mathbf{B}^{-1}\mathbf{P}_2 = (1, 4)^T$. Откуда следует

$$\theta_1 = \min \left\{ \frac{7}{1}, \frac{15}{4} \right\} = 3.75, \text{ что соответствует переменной } x_5,$$

$$\theta_2 = \infty \text{ (поскольку } \mathbf{B}^{-1}\mathbf{P}_2 > \mathbf{0}).$$

Так как по условию $x_2 \leq 3$, получаем значение, принимаемое переменной x_2 .

$$x_2 = \min\{3.75, \infty, 3\} = 3 = u_2.$$

Поскольку $x_2 = u_2$ базис остается неизменным; переменная x_2 в базис не вводится (остается небазисной), но принимает значение своей верхней границы. После подстановки $x_2 = 3 - x'_2$ получаем новую симплекс-таблицу.

Базис	x_1	x'_2	x_3	x_4	x_5	Решение
z	-3	5	-2	0	0	50
x_4	1	-1	2	1	0	4
x_5	2	-4	3	0	1	3

• Итерация 1

Вводимая в базис переменная - x_1 . Базисный вектор \mathbf{X}_B и обратная матрица \mathbf{B}^{-1} такие же, как на предыдущей итерации. Вычисляем $\mathbf{B}^{-1}\mathbf{P}_1 = (1, 2)^T$. Отсюда следует

$$\theta_1 = \min \left\{ \frac{4}{1}, \frac{3}{2} \right\} = 1.5, \text{ что соответствует переменной } x_5,$$

$$\theta_2 = \infty \text{ (поскольку } \mathbf{B}^{-1}\mathbf{P}_1 > \mathbf{0}),$$

$$\theta_3 = 4.$$

Таким образом $x_1 = \min\{1.5, \infty, 4\} = 1.5 = \theta_1$.

В данной ситуации переменная x_1 становится базисной со значением 1.5, переменная x_5 выводится из базиса и принимает нулевое значение. Получаем следующую симплекс-таблицу.

Базис	x_1	x'_2	x_3	x_4	x_5	Решение
z	0	-1	5/2	0	3/2	109/2
x_4	0	1	1/2	1	-1/2	5/2
x_1	1	-2	3/2	0	1/2	3/2

• Итерация 2

Имеем новую обратную матрицу:

$$\mathbf{B}^{-1} = \begin{bmatrix} 1 & -\frac{1}{2} \\ 0 & \frac{1}{2} \end{bmatrix}$$

Значения базисных переменных вычисляются так: $\mathbf{X}_B = (x_4, x_1)^T = \mathbf{B}^{-1}\mathbf{b}' = (5/2, 3/2)^T$, где $\mathbf{b}' = (4, 3)^T$ - значения правых частей ограничений, вычисленные в конце нулевой итерации.

Определяем x'_2 как вводимую в базис переменную. Учитывая, что $\mathbf{P}'_2 = -\mathbf{P}_2$, получаем $\mathbf{B}^{-1}\mathbf{P}'_2 = (1, 2)^T$.

Далее вычисляем

$$\theta_1 = \min \left\{ \frac{5}{1}, - \right\}, \text{ что соответствует } x_4,$$

$$\theta_2 = \min \left\{ -, \frac{3-4}{-2} \right\} = 1.25, \text{ что соответствует } x_1,$$

$$\theta_3 = 3.$$

Таким образом $x'_2 = \min\{2.5, \mathbf{1.25}, 3\} = 1.25 = \theta_2$.

Таким образом, переменная x'_2 вводится в базис (со значением 1.25), из которого исключается переменная x_1 , с ненулевым значением, равным ее верхней границе. Получаем следующую симплекс-таблицу.

Базис	x_1	x'_2	x_3	x_4	x_5	Решение
z	$-1/2$	0	$7/4$	0	$5/4$	$215/4$
x_4	$1/2$	0	$5/4$	1	$-1/4$	$13/4$
x'_2	$-1/2$	1	$-3/2$	0	$-1/4$	$-3/4$

Поскольку переменная x_1 становится небазисной с ненулевым значением ($= 4$), применяем подстановку $x_1 = 4 - x'_1$, что приводит к следующей симплекс-таблице.

Базис	x_1	x'_2	x_3	x_4	x_5	Решение
z	$1/2$	0	$7/4$	0	$5/4$	$223/4$
x_4	$-1/2$	0	$5/4$	1	$-1/4$	$5/4$
x'_2	$1/2$	1	$-3/2$	0	$-1/4$	$5/4$

Данная таблица представляет допустимое оптимальное решение.

Оптимальные значения переменных x_1, x_2 и x_3 получаем обратной подстановкой: $x_1 = u_1 - x'_1 = 4 - 0 = 4, x_2 = u_2 - x'_2 = 3 - 5/4 = 7/4$ и $x_3 = 0$.

Теперь вычисляем значение переменной y : $y = 7 + x_2 = 7 + 7/4 = 35/4$.

Оптимальное значение целевой функции равно $z = 223/4$.

5 Марковские процессы с бесконечным числом этапов

Введем некоторые обозначения:

K - количество альтернативных стратегий;

S - количество стационарных стратегий;

\mathbf{S} - множество всех возможных стационарных стратегий;

\mathbf{P}^s и R^s - матрицы переходных(одношаговых) вероятностей и доходов, соответствующие применяемой стратегии s ;

ν_i^s - ожидаемый доход, получаемый за один этап при стратегии s для заданного состояния $i, i = 1, 2, \dots, m$;

π_i^s - долгосрочные стационарные вероятности матрицы переходных вероятностей (т. е. $\pi^s \mathbf{P}^s = \pi^s$ и $\pi_1^s + \pi_2^s + \dots + \pi_m^s = 1$).

Тогда оптимальная стационарная стратегия s^* выбирается из условия

$$\max_{s \in \mathbf{S}} \left\{ \sum_{i=1}^m \pi_i^s \nu_i^s \mid \pi^s \mathbf{P}^s = \pi^s, \pi_1^s + \pi_2^s + \dots + \pi_m^s = 1, \pi_i^s \geq 0, i = 1, 2, \dots, m \right\}.$$

Приведенная задача служит основой для формулировки марковской задачи принятия решений в виде задачи линейного программирования. Однако необходимо преобразовать переменные задачи таким образом, чтобы оптимальное решение автоматически определяло оптимальное действие(альтернативу) k , когда система находится в состоянии i . Совокупность всех оптимальных действий определяет оптимальную стратегию s^* .

Введем обозначение: q_i^k - условная вероятность выбора альтернативы k , когда система находится в состоянии i . Тогда задачу можно представить в виде.

$$E_{max} = \max \sum_{i=1}^m \pi_i \left(\sum_{k=1}^K q_i^k \nu_i^k \right)$$

при ограничениях

$$\begin{aligned} \pi_j &= \sum_{i=1}^m \pi_i p_{ij}, j = 1, 2, \dots, m, \\ \pi_1 + \pi_2 + \dots + \pi_m &= 1, \\ q_i^1 + q_i^2 + \dots + q_i^K &= 1, i = 1, 2, \dots, m, \\ \pi_i \geq 0, q_i^k &\geq 0, i = 1, 2, \dots, m, k = 1, 2, \dots, K. \end{aligned}$$

Заметим, что при каждом i должно быть только одно k , что $q_i^k = 1$.

Обозначим $w_{ik} = \pi_i q_i^k$, т. е. w_{ik} - это совместная вероятность пребывания в состоянии i и принятия решения k , а это означает, что $\pi_i =$

$\sum_{k=1}^K w_{ik}$. Следовательно,

$$q_i^k = \frac{w_{ik}}{\sum_{k=1}^K w_{ik}}.$$

При таком способе определения q_i^k ограничения $\sum_{i=1}^m \pi_1 = 1$ и $\sum_{k=1}^K q_i^k = 1$ выполняются автоматически.

После замены переменных исходная задача преобразуется к виду:

Максимизировать $E = \sum_{i=1}^m \sum_{k=1}^K \nu_i^k w_{ik}$ при ограничениях

$$\begin{aligned} \sum_{k=1}^K w_{jk} - \sum_{i=1}^m \sum_{k=1}^K p_{ij}^k w_{ik} &= 0, j = 1, 2, \dots, m, \\ \sum_{i=1}^m \sum_{k=1}^K w_{ik} &= 1, \\ w_{ik} &\geq 0, i = 1, 2, \dots, K. \end{aligned}$$

Сформулированная в таком виде задача представляет собой задачу линейного программирования с переменными w_{ik} .

Покажем теперь, что ее оптимальное решение автоматически гарантирует, что $q_i^k = 1$ только для одного k при любом i . Заметим, что в задаче линейного программирования имеется m независимых уравнений. Следовательно, задача должна включать m базисных переменных. Но, кроме того, w_{ik} строго положительна по меньшей мере при одном k для каждого i . Из этих двух утверждений можно заключить, что величина

$$q_i^k = \frac{w_{ik}}{\sum_{k=1}^K w_{ik}}$$

может принимать только два значения (0 и 1), что и требовалось доказать.

Рассмотрим теперь марковскую задачу принятия решений с дисконтированием. Обозначим коэффициент дисконтирования как α . Для решения этой задачи нужно лишь немного изменить ограничения, а именно они запишутся в виде:

$$\sum_{k=1}^K w_{jk} - \alpha \sum_{i=1}^m \sum_{k=1}^K p_{ij}^k w_{ik} = 0, j = 1, 2, \dots, m,$$

6 Транспортная задача

Дабы не создавать впечатление, что линейное программирование отождествляется с симплекс-методом изложим еще один, довольно специфический алгоритм. Алгоритм решения одной из транспортных задач.

ЗАДАЧА 1 *Задача о перевозке товара. m поставщиков товара поставляют $\mathbf{A} = (a_1, \dots, a_m)$ единиц товара, и n потребителей потребляющих $\mathbf{B} = (b_1, \dots, b_n)$ единиц товара. Причем $\sum_{i=1}^m a_i = \sum_{j=1}^n b_j$. Задана стоимость перевозки c_{ij} , нужно вычислить сколько товара каждый поставщик должен поставлять каждому потребителю для минимизации общей стоимости перевозок.*

Или, в терминах линейного программирования:

Минимизировать

$$\sum_{i=1}^m \sum_{j=1}^n c_{ij} x_{ij}$$

При ограничениях

$$\begin{aligned} \forall i \sum_{j=1}^n x_{ij} &= a_i \\ \forall j \sum_{i=1}^m x_{ij} &= b_j \end{aligned}$$

6.1 Вспомогательные теоремы

Рассмотрим прямоугольную решетку A , некоторые узлы которой выделены. Множество всех узлов назовем Φ , а множество выделенных узлов Ψ . Пусть A^v – множество столбцов решетки, A^g – множество строк. Назовем рядом столбец или строку и обозначим множество рядов как A^r .

Определение 1 *Цепь – ломаная состоящая из горизонтальных и вертикальных отрезков с изломами только в выделенных узлах.*

Определение 2 *Цикл – замкнутая цепь.*

Положим $X = \#\Psi$ – количество выделенных узлов, а X_{max} – максимальное количество выделенных узлов при котором не образуется цикл.

Лемма 1 Если $X = X_{max}$, то в любом ряду есть выделенный узел.

Действительно, если мы выделим любой узел в ряду где нет выделенных узлов, то в этом узле не будет излома.

Лемма 2 Если $X = X_{max}$, то любая пара выделенных узлов принадлежит некоторой цепи.

Теорема 1

$$X_{max} = m + n - 1$$

Доказательство: Индукция по m . Для $m = 1$ — очевидно. Пусть для k строк $X_{max}^k = k + n - 1$, тогда добавим строку. В каждом столбце матрицы есть выделенный узел (1). Добавить два выделенных узла в новую строку не удастся (2). Можно добавить один узел.

6.2 Решение транспортной задачи

Условия транспортной задачи запишем в виде таблицы.

	b_1	\dots	b_n
a_1	c_{11}	\vdots	c_{1n}
\vdots	\vdots	\vdots	\vdots
a_n	c_{n1}	\dots	c_{nn}

Наша задача заключается в построении соответствующей таблицы перевозок

	b_1	\dots	b_n
a_1	x_{11}	\vdots	x_{1n}
\vdots	\vdots	\vdots	\vdots
a_n	x_{n1}	\dots	x_{nn}

Будем считать клетки таблицы узлами, а множество выделенных узлов Ψ — множество клеток с $x_{ij} \geq 0$.

Теорема 2 Оптимальное решение достигается на ациклическом наборе.

Доказательство: Пусть мы нашли допустимый набор перевозок, содержащий цикл. Тогда занумерует клетки цикла (в порядке обхода) и разобьем цикл на два множества odd и even с суммами c_{ij} равными C_{odd} C_{even} . Если $C_{odd} \leq C_{even}$, то возьмем перевозку с минимальным весом r в четном полуцикле, теперь для $x_{ij} = x_{ij} - r (ij \in even)$, $x_{ij} = x_{ij} + r (ij \in odd)$. Тогда цикл разрушится, а общая стоимость перевозки не увеличится. \square

6.3 Метод потенциалов

Теперь перейдем к описанию метода решения транспортной задачи.

- Инициализация

Воспользуемся жадным алгоритмом по строкам (берем в строке перевозку минимальной стоимости и исчерпываем её ресурс). Для всех строк кроме последней, в случае исчерпания ресурсов как потребителя так и поставщика, заполняем фиктивной (нулевой) перевозкой клетку минимальной стоимости.

Теорема 3 В ходе инициализации получен ациклический набор.

Теорема 4 В ходе инициализации получен набор из $n + m - 1$ клеток.

- Выбор клетки для загрузки

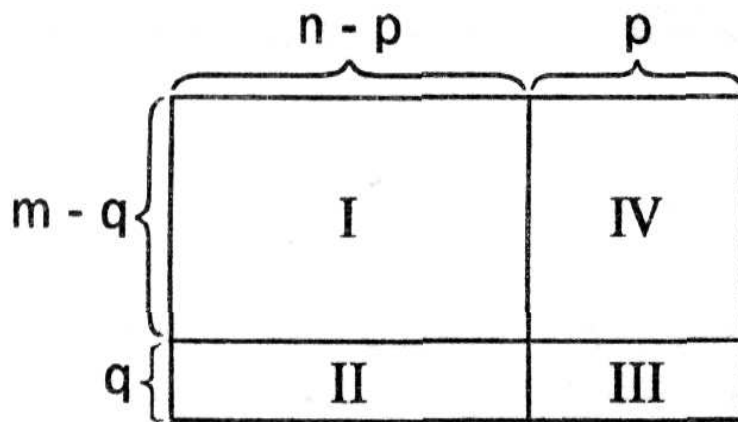
Введем числа u_i и v_j , сопоставив их строкам и столбцам соответственно. Причем так, чтобы для загруженной клетки $u_i + v_j = c_{ij}$. Положим дополнительно $u_1 = 0$, тогда таблица примет вид.

b_j	8	10	7	10	15	6	
a_i							
10	3	5	2	7	3	6	0
	3		7		*		
15	4	∞	2	4	∞	8	1
	5			10			0
30	2	5	8	10	6	7	0
		10			15	5	
	3	5	2	3	6	7	

Для удобства мы записываем выбранные перевозки x в левом нижнем углу клетки.

Теорема 5 Все u, v вычисляются однозначно.

Доказательство: Пусть удалось вычислить не все, тогда можно переставить местами некоторые ряды и получить



Тогда в блоках II и III нет загруженных клеток. Вследствие ацикличности набора в блоке IV их не более $p + q - 1$, тогда в блоке I более $(n - p) + (m - q)$ загруженных клеток, что приводит к наличию цикла.

Выберем теперь незагруженную клетку для которой $u_i + v_j > c_{ij}$. Пусть эта клетка – первая при обходе, тогда $C_{odd} > C_{even}$, найдем минимальную перевозку r на четном полуцикле вычтем её из перевозок четного полуцикла и прибавим к нечетному. Будем продолжать действия пока есть клетка для разгрузки.

b_i		8	10	7	10	15	5	
a_j								
10	3	5	2	7	3	6		0
	3		7		0			
15	4	∞	2	4	∞	8		1
	5			10				
30	2	5	8	10	6	7		3
	*	10			15	5		
		3	2	2	3	3	4	

b_i		8	10	7	10	15	5	
a_j								
10	3	5	2	7	3	6		0
	3		2		8			
15	4	∞	2	4	∞	8		0
	5		5	10				
30	2	5	8	10	6	7		3
	8	10			7	5		
		-1	2	2	4	3	4	

Теорема 6 Если для всех незагруженных клеток потенциал $p_{ij} = u_i + v_j - c_{ij}$, неположителен то получена наилучшая система перевозок.

Список литературы

- [1] Хэмди А. Таха, Введение в линейное программирование.
- [2] В. И. Рублинецкий, Введение в мир алгоритмов.