

# Методы оптимальных решений

## Краткий конспект лекций

### Тема 1. Линейное программирование

Линейное программирование – направление математики, изучающее методы решения экстремальных задач, которые характеризуются линейной зависимостью между переменными и линейным критерием оптимальности.

Несколько слов о самом термине линейное программирование. Он требует правильного понимания. В данном случае программирование - это, конечно, не составление программ для ЭВМ. Программирование здесь должно интерпретироваться как планирование, формирование планов, разработка программы действий.

К математическим задачам линейного программирования относят исследования конкретных производственно-хозяйственных ситуаций, которые в том или ином виде интерпретируются как задачи об оптимальном использовании ограниченных ресурсов.

Круг задач, решаемых при помощи методов линейного программирования достаточно широк. Это, например:

задача об оптимальном использовании ресурсов при производственном планировании;

задача о смесях (планирование состава продукции);

задача о нахождении оптимальной комбинации различных видов продукции для хранения на складах (управление товарно-материальными запасами или "задача о рюкзаке");

транспортные задачи (анализ размещения предприятия, перемещение грузов).

Линейное программирование – наиболее разработанный и широко применяемый раздел математического программирования (кроме того, сюда относят: целочисленное, динамическое, нелинейное, параметрическое программирование). Это объясняется следующим:

математические модели большого числа экономических задач линейны относительно искомых переменных;

данный тип задач в настоящее время наиболее изучен. Для него разработаны специальные методы, с помощью которых эти задачи решаются, и соответствующие программы для ЭВМ;

многие задачи линейного программирования, будучи решенными, нашли широкое применение;

некоторые задачи, которые в первоначальной формулировке не являются линейными, после ряда дополнительных ограничений и допущений могут стать линейными или могут быть приведены к такой форме, что их можно решать методами линейного программирования.

Экономико-математическая модель любой задачи линейного программирования

включает: целевую функцию, оптимальное значение которой (максимум или минимум) требуется отыскать; ограничения в виде системы линейных уравнений или неравенств; требование неотрицательности переменных.

В общем виде модель записывается следующим образом:

целевая функция:

$$= c_1x_1 + c_2x_2 + \dots + c_nx_n \rightarrow \max(\min);$$

(2.1)

ограничения:

$$a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n \{ \leq = \geq \} b_1,$$

$$a_{21}x_1 + a_{22}x_2 + \dots + a_{2n}x_n \{ \leq = \geq \} b_2,$$

...

$$a_{m1}x_1 + a_{m2}x_2 + \dots + a_{mn}x_n \{ \leq = \geq \} b_m;$$

(2.2)

требование неотрицательности:

$$x_j \geq 0,$$

(2.3)

При этом  $a_{ij}$ ,  $b_i$ ,  $c_j$  ( ) - заданные постоянные величины.

Задача состоит в нахождении оптимального значения функции (2.1) при соблюдении ограничений (2.2) и (2.3).

Систему ограничений (2.2) называют функциональными ограничениями задачи, а ограничения (2.3) - прямыми.

Вектор , удовлетворяющий ограничениям (2.2) и (2.3), называется допустимым решением (планом) задачи линейного программирования. План , при котором функция (2.1) достигает своего максимального (минимального) значения, называется оптимальным.

Далее приведем примеры некоторых типовых задач, решаемых при помощи методов линейного программирования. Такие задачи имеют реальное экономическое содержание. Сейчас лишь сформулируем их в терминах ЗЛП, а методы решения подобных задач рассмотрим ниже.

1. Задача об оптимальном использовании ресурсов при производственном планировании.

Общий смысл задач этого класса сводится к следующему.

Предприятие выпускает  $n$  различных изделий. Для их производства требуется  $m$  различных видов ресурсов (сырья, материалов, рабочего времени и т.п.). Ресурсы ограничены, их запасы в планируемый период составляют, соответственно,  $b_1, b_2, \dots, b_m$  условных единиц.

Известны также технологические коэффициенты  $a_{ij}$ , которые показывают, сколько единиц  $i$ -го ресурса требуется для производства единицы изделия  $j$ -го вида ( ).

Прибыль, получаемая предприятием при реализации изделия  $j$ -го вида, равна  $c_j$ .

В планируемом периоде значения величин  $a_{ij}$ ,  $b_i$  и  $c_j$  остаются постоянными.

Требуется составить такой план выпуска продукции, при реализации которого прибыль предприятия была бы наибольшей.

## 2. Задача о смесях (планирование состава продукции).

К группе задач о смесях относят задачи по отысканию наиболее дешевого набора из определенных исходных материалов, обеспечивающих получение смеси с заданными свойствами. Иными словами, получаемые смеси должны иметь в своем составе  $m$  различных компонентов в определенных количествах, а сами компоненты являются составными частями  $n$  исходных материалов.

## 3. Транспортная задача.

Под транспортной задачей понимают целый ряд задач, имеющих определенную специфическую структуру. Наиболее простыми транспортными задачами являются задачи о перевозках некоторого продукта из пунктов отправления в пункты назначения при минимальных затратах на перевозку.

Три поставщика одного и того же продукта располагают в планируемый период следующими его запасами: первый – 120 условных единиц, второй – 100 условных единиц, третий – 80 условных единиц. Этот продукт должен быть перевезен к трем потребителям, потребности которых равны 90, 90 и 120 условных единиц, соответственно.

Обычно начальные условия транспортной задачи записывают в так называемую транспортную таблицу (см. таблицу 2.3). В ячейках таблицы в левом верхнем углу записывают показатели затрат (расходы по доставке единицы продукта между соответствующими пунктами), под диагональю каждой ячейки размещается величина поставки  $x_{ij}$  (т.е.  $x_{ij}$  - количество единиц груза, которое будет перевезено от  $i$ -го поставщика  $j$ -му потребителю).

Необходимо определить наиболее дешевый вариант перевозок, при этом каждый поставщик должен отправить столько груза, сколько имеется у него в запасе, а каждый потребитель должен получить нужное ему количество продукции.

## 2. Геометрическое решение ЗЛП

Если система ограничений задачи линейного программирования представлена в виде системы линейных неравенств с двумя переменными, то такая задача может быть решена геометрически. Таким образом, данный метод решения ЗЛП имеет очень узкие рамки применения.

Однако метод представляет большой интерес с точки зрения выработки наглядных представлений о сущности задач линейного программирования.

Геометрический (или графический) метод предполагает последовательное выполнение ряда шагов. Ниже представлен порядок решения задачи линейного программирования на основе ее геометрической интерпретации.

1. Сформулировать ЗЛП.

2. Построить на плоскости  $\{x_1, x_2\}$  прямые, уравнения которых получаются в результате замены в ограничениях знаков неравенств на знаки точных равенств.

3. Найти полуплоскости, определяемые каждым из ограничений задачи.

4. Найти область допустимых решений.

5. Построить прямую  $c_1x_1 + c_2x_2 = h$ , где  $h$  - любое положительное число, желательно такое, чтобы проведенная прямая проходила через многоугольник решений.

6. Перемещать найденную прямую параллельно самой себе в направлении увеличения (при поиске максимума) или уменьшения (при поиске минимума) целевой функции. В результате, либо отыщется точка, в которой целевая функция принимает максимальное (минимальное) значение, либо будет установлена неограниченность функции на множестве решений.

7. Определить координаты точки максимума (минимума) функции и вычислить значение функции в этой точке.

### 3. Основные теоремы линейного программирования

Для обоснования методов решения задач линейного программирования сформулируем ряд важнейших теорем, опуская их аналитические доказательства. Уяснить смысл каждой из теорем поможет понятие о геометрической интерпретации решения ЗЛП, данное в предыдущем подразделе.

Однако сначала напомним о некоторых понятиях, важных с точки зрения дальнейшего разговора.

Любые  $m$  переменных системы  $m$  линейных уравнений с  $n$  переменными ( $m < n$ ) называются основными, если определитель матрицы коэффициентов при них отличен от нуля. Тогда остальные  $m-n$  переменных называются неосновными (или свободными).

Базисным решением системы  $m$  линейных уравнений с  $n$  переменными ( $m < n$ ) называется всякое ее решение, в котором все неосновные переменные имеют нулевые значения.

**Теорема 1.** Множество всех допустимых решений системы ограничений задачи линейного программирования является выпуклым.

В частном случае, когда в систему ограничений входят только две переменные  $x_1$  и  $x_2$ , это множество можно изобразить на плоскости. Так как речь идет о допустимых решениях ( $x_1, x_2 \geq 0$ ), то соответствующее множество будет располагаться в первой четверти декартовой системы координат. Это множество может быть замкнутым (многоугольник), незамкнутым (неограниченная многогранная область), состоять из

единственной точки и, наконец, система ограничений-неравенств может быть противоречивой.

Теорема 2. Если задача линейного программирования имеет оптимальное решение, то оно совпадает с одной (двумя) из угловых точек множества допустимых решений.

Из теоремы 2 можно сделать вывод о том, что единственность оптимального решения может нарушаться, причем, если решение не единственное, то таких оптимальных решений будет бесчисленное множество (все точки отрезка, соединяющего соответствующие угловые точки).

Теорема 3. Каждому допустимому базисному решению задачи линейного программирования соответствует угловая точка области допустимых решений, и наоборот.

Следствием из теорем 2 и 3 является утверждение о том, что оптимальное решение (оптимальные решения) задачи линейного программирования, заданной (или приведенной) ограничениями-уравнениями, совпадает с допустимым базисным решением (допустимыми базисными решениями) системы ограничений.

Таким образом, оптимальное решение ЗЛП следует искать среди конечного числа допустимых базисных решений

## Тема 2. Транспортная задача

Транспортная задача (задача Монжа — Канторовича) — математическая задача линейного программирования специального вида о поиске оптимального распределения однородных объектов из аккумулятора к приемникам с минимизацией затрат на перемещение.[1][2] Для простоты понимания рассматривается как задача об оптимальном плане перевозок грузов из пунктов отправления в пункты потребления, с минимальными затратами на перевозки. Транспортная задача по теории сложности вычислений входит в класс сложности P. Когда суммарный объем предложений (грузов, имеющихся в пунктах отправления) не равен общему объему спроса на товары (грузы), запрашиваемые пунктами потребления, транспортная задача называется несбалансированной (открытой).

Транспортная задача (классическая) — задача об оптимальном плане перевозок однородного продукта из однородных пунктов наличия в однородные пункты потребления на однородных транспортных средствах (предопределённом количестве) со статическими данными и линейном подходе (это основные условия задачи).

Для классической транспортной задачи выделяют два типа задач: критерий стоимости (достижение минимума затрат на перевозку) или расстояний и критерий времени (затрачивается минимум времени на перевозку). Под названием транспортная задача, определяется широкий круг задач с единой математической моделью, эти задачи

относятся к задачам линейного программирования и могут быть решены оптимальным методом. Однако, спец.метод решения транспортной задачи позволяет существенно упростить её решение, поскольку транспортная задача разрабатывалась для минимизации стоимости перевозок.

Проблема была впервые формализована французским математиком Гаспаром Монжем в 1781 году[3]. Основное продвижение было сделано на полях во время Великой Отечественной войны советским математиком и экономистом Леонидом Канторовичем[4]. Поэтому иногда эта проблема называется транспортной задачей Монжа — Канторовича.

классическую транспортную задачу можно решить симплекс-методом, но в силу ряда особенностей её можно решить проще (для задач малой размерности).

Условия задачи располагают в таблице, вписывая в ячейки количество перевозимого груза из  $\sim A_i$  в  $\sim B_j$  груза  $\sim X_{ij} \geq 0$ , а в маленькие клетки — соответствующие тарифы  $\sim C_{ij}$ .

#### Нахождение опорного плана

Требуется определить опорный план и путём последовательных операций найти оптимальное решение. Опорный план можно найти следующими методами: «северо-западного угла», «наименьшего элемента», двойного предпочтения и аппроксимации Фогеля.

#### Метод северо-западного угла (диагональный или улучшенный)

На каждом этапе максимально возможным числом заполняют левую верхнюю клетку оставшейся части таблицы. Заполнение таким образом, что полностью выносятся груз из  $\sim A_i$  или полностью удовлетворяется потребность  $\sim B_j$ .

#### Метод наименьшего элемента

Одним из способов решения задачи является метод минимального (наименьшего) элемента. Его суть заключается в сведении к минимуму побочных перераспределений товаров между потребителями.

#### Алгоритм:

Из таблицы стоимостей выбирают наименьшую стоимость и в клетку, которая ей соответствует, вписывают большее из чисел.

Проверяются строки поставщиков на наличие строки с израсходованными запасами и столбцы потребителей на наличие столбца, потребности которого полностью удовлетворены. Такие столбцы и строки далее не рассматриваются.

Если не все потребители удовлетворены и не все поставщики израсходовали товары,

возврат к п. 1, в противном случае задача решена.

### Итерации

После нахождения опорного плана перевозок, нужно применить один из алгоритмов его улучшения, приближения к оптимальному.

Метод падающего камня (нем.)

Метод потенциалов.

### Решение с помощью теории графов

Рассматривается двудольный граф, в котором пункты производства находятся в верхней доле, а пункты потребления — в нижней. Пункты производства и потребления попарно соединяются рёбрами бесконечной пропускной способности и цены за единицу потока  $\sim C_{ij}$ .

К верхней доле искусственно присоединяется исток. Пропускная способность рёбер из истока в каждый пункт производства равна запасу продукта в этом пункте. Цена за единицу потока у этих рёбер равна 0.

Аналогично к нижней доле присоединяется сток. Пропускная способность рёбер из каждого пункта потребления в сток равна потребности в продукте в этом пункте. Цена за единицу потока у этих рёбер тоже равна 0.

Дальше решается задача нахождения максимального потока минимальной стоимости (mincost maxflow). Её решение аналогично нахождению максимального потока в алгоритме Форда — Фалкерсона. Только вместо кратчайшего дополняющего потока ищется самый дешёвый. Соответственно, в этой подзадаче используется не поиск в ширину, а алгоритм Беллмана — Форда. При возврате потока стоимость считается отрицательной.

Алгоритм «mincost maxflow» можно запускать и сразу — без нахождения опорного плана. Но в этом случае процесс решения будет несколько более долгим. Выполнение алгоритма «mincost maxflow» происходит не более чем за  $\sim O(v^2e^2)$  операций. ( $\sim e$  — количество рёбер,  $\sim v$  — количество вершин.) При случайно подобранных данных обычно требуется гораздо меньше — порядка  $\sim O(ve)$  операций.

При решении несбалансированной транспортной задачи применяют приём, позволяющий сделать ее сбалансированной. Для этого вводят фиктивные пункты назначения или отправления. Выполнение баланса транспортной задачи необходимо для того, чтобы иметь возможность применить алгоритм решения, построенный на

использовании транспортных таблиц.

#### Транспортная задача в сетевой постановке

В этом варианте пункты не делятся на пункты отправления и пункты потребления, все пункты равноправны, но производство задается положительным числом, а потребление — отрицательным. Перевозки осуществляются по заданной сети, в которой дуги могут соединять любые пункты, включая производитель — производитель, потребитель — потребитель.

Задача решается слегка измененным методом потенциалов, практически тем же, что и классическая постановка.

#### Транспортная задача с ограничениями пропускной способности

Вариант транспортной задачи в сетевой постановке, при котором задается максимальная пропускная способность некоторых дуг.

Задача решается слегка усложненным методом потенциалов.

#### Многопродуктовая транспортная задача

Вариант транспортной задачи, в которой присутствует несколько продуктов (пункты могут производить/потреблять несколько продуктов). Для некоторых дуг задается ограничение на пропускную способность (без этого ограничения задача распадается на отдельные задачи по продуктам).

Задача решается симплекс-методом (используется разложение Данцига-Вулфа, в качестве подзадач используются однопродуктовые транспортные задачи).

### **Тема 3. Целочисленное программирование**

Целочисленное программирование – один из наиболее молодых, перспективных и быстро развивающихся разделов математического программирования. Можно перечислить большое количество разнообразных задач планирования экономики, организации производства, исследования конфликтных ситуаций, синтеза схем автоматического регулирования, которые формально сводятся к выбору лучших, в некотором смысле, значений параметров из определенной дискретной совокупности заданных величин. К ним можно отнести и экстремальные комбинаторные задачи, возникающие в различных разделах дискретной математики.

Задачи и методы, относящиеся к перечисленному кругу вопросов, в литературе именуется по-разному. Наибольшее распространение получил термин «целочисленное программирование», однако встречаются и такие как «дискретное программирование»,



реже «комбинаторное (или диофантово) программирование».

Наиболее изученными задачами этого класса являются целочисленные задачи линейного программирования, в которых на все переменные (или на их часть) наложено дополнительное требование целочисленности. От них принято отличать так называемые дискретные задачи линейного программирования, в которых область допустимого изменения каждой переменной – не множество целых неотрицательных чисел, а некоторое заданное конечное множество.

Целочисленные задачи математического программирования могут возникать различными путями.

1. Существуют задачи линейного программирования, которые формально к целочисленным не относятся, но при соответствующих исходных данных всегда обладают целочисленным планом. Примеры таких задач – транспортная задача и ее модификации (задачи о назначениях, о потоках в сетях).

2. Толчком к изучению целочисленных задач в собственном смысле слова явилось рассмотрение задач линейного программирования, в которых переменные представляли физически неделимые величины. Они были названы задачами с неделимостью. Таковы, например, задачи об оптимизации комплекса средств доставки грузов, о нахождении минимального порожнего пробега автомобилей при выполнении заданного плана перевозок, об определении оптимального машинного парка и его оптимального распределения по указанным работам при условии минимизации суммарной стоимости (машинного парка и производимых работ), о нахождении минимального количества судов для осуществления данного графика перевозок и т. п.

3. Другим важным толчком к построению теории целочисленного программирования стал новый подход к некоторым экстремальным комбинаторным задачам. В них требуется найти экстремум целочисленной линейной функции, заданной на конечном множестве элементов. Такие задачи принято называть задачами с альтернативными переменными. В качестве примеров можно назвать задачи коммивояжера (бродячего торговца), об оптимальном назначении, теории расписания, или календарного планирования, и задачи с дополнительными логическими условиями (например, типа «или – или», «если – то» и т. п.).

Исторически первой задачей целочисленного типа является опубликованная в 1932 г. венгерским математиком Е. Эгервари задача о назначении персонала. В 1955 г. на Втором симпозиуме по линейному программированию была рассмотрена задача о бомбардировщике, известная как задача о ранце.

Приведем примеры целочисленных задач линейного программирования.

Пример. Задача о ранце. Имеется  $m$ -вектор ограниченных ресурсов  $b = (b_1, b_2, \dots, b_m)$ , которые можно использовать для перевозки различных по своим характеристикам грузов. Каждый  $j$ -й груз ( $j = \overline{1, n}$ ) обладает следующими свойствами: 1) неделимостью; 2) полезностью  $c_j$ ; 3) расходом  $i$ -го ресурса для перевозки

единицы  $j$ -го груза  $a_{ij}$  ( $i = \overline{1, m}; j = \overline{1, n}$ ). Выбрать такой номер груза для перевозки, при котором максимизируется общая полезность рейса (суммарная стоимость перевезенных за рейс грузов).

Составим математическую модель задачи. Обозначим через  $x_j$  количество выбранных для транспортировки предметов. Требованию неделимости соответствует условие

$$x_j \geq 0, x_j - \text{целые}, j = \overline{1, n}. \quad (13.1)$$

Сопоставление расходов ресурсов каждого типа для транспортировки единицы груза и их наличия приводит к ограничению

$$\sum_{j=1}^n a_{ij} x_j \leq b_i, i = \overline{1, m}. \quad (13.2)$$

Общая полезность рейса определяется значением функции

$$F(x) = \sum_{j=1}^n c_j x_j. \quad (13.3)$$

Частным случаем задачи (13.1) – (13.2) является задача о ранце, в которой любой из заданного набора предметов  $j = \overline{1, n}$  может быть выбран или нет. Тогда математическая модель примет следующий вид:

Максимизировать

$$F(x) = \sum_{j=1}^n c_j x_j$$

при условиях

$$\sum_{j=1}^n a_{ij} x_j \leq b_i, i = \overline{1, m},$$

$$x_j = \begin{cases} 0, \\ 1, j = \overline{1, n}. \end{cases}$$

Пример. В области

$$\begin{cases} 2x_1 + 11x_2 \leq 38, \\ x_1 + x_2 \leq 7, \\ 4x_1 - 5x_2 \leq 5, \\ x_1 \geq 0, x_2 \geq 0 \\ x_1, x_2 - \text{целые}. \end{cases}$$

найти максимум функции  $F(x_1, x_2) = x_1 + x_2$ .

Решим задачу геометрически (рис. 13.1). Область поиска экстремума – многоугольник OABCD. Так как линия уровня целевой функции параллельна стороне BC многоугольника, экстремум достигается в вершинах **B(13/3; 8/3)** и **C(40/9; 23/9)**, а также в любой точке отрезка BC и равен 7. Нас интересуют лишь точки с целочисленными координатами, поэтому ни B, ни C не являются допустимым решением задачи. Округлив значение координат точек B и C, получим точку (4; 3), не принадлежащую области поиска. Легко показать, что целочисленный оптимум достигается в точках M(2; 3) и N(3; 2) и равен 5. Обе точки находятся внутри области поиска.

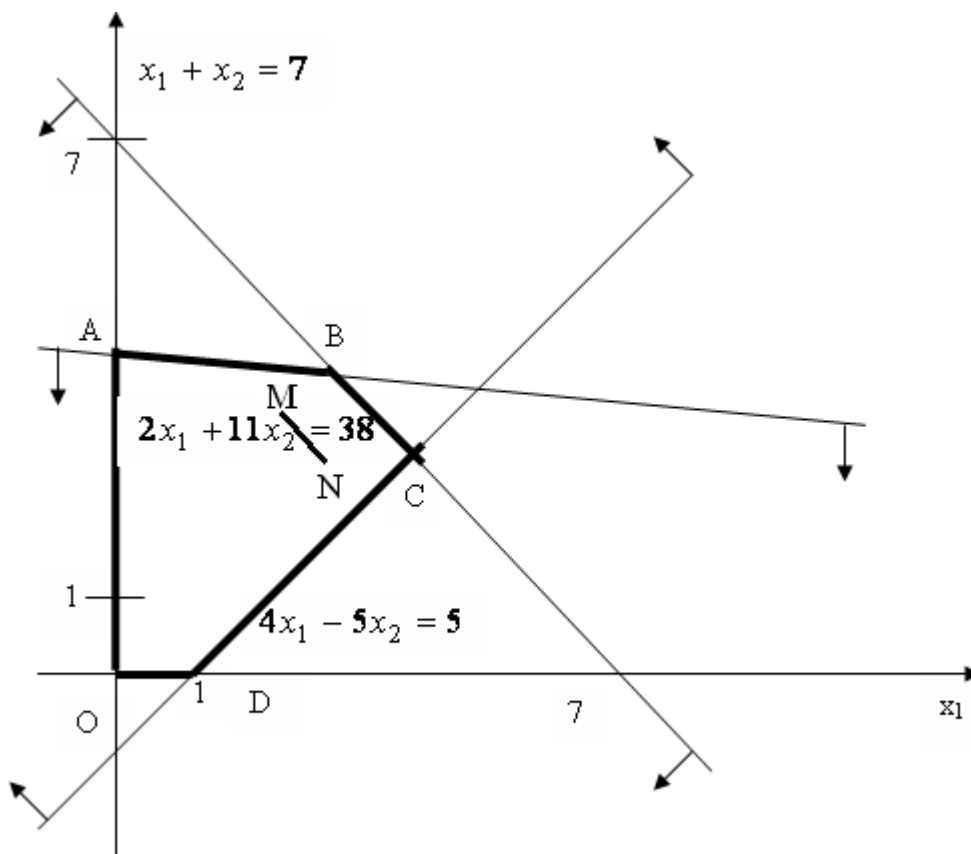


Рисунок 13.1

Несостоятельность округления подчеркивается также следующими соображениями. Несмотря на то, что целочисленные переменные обычно выражают количество неделимых предметов, возможны и другие типы спецификации этих переменных. Так в задаче о ранце решение представляется булевой переменной  $x$  ( $x = 0$  или  $x = 1$ ). В этом случае бессмысленно оперировать дробными значениями величины  $x$  и процедура округления является логически неприемлемой.

Из приведенного примера видно, что для решения задач целочисленного программирования необходимо рассмотреть особые методы оптимизации. Кроме того, оптимальное решение задач целочисленного программирования не обязательно принадлежит границе многогранника (многоугольника) условий, что характерно для задач линейного программирования.

## Тема 4. Теория игр

Теория игр — математический метод изучения оптимальных стратегий в играх. Под игрой понимается процесс, в котором участвуют две и более сторон, ведущих борьбу за реализацию своих интересов. Каждая из сторон имеет свою цель и использует некоторую стратегию, которая может вести к выигрышу или проигрышу — в зависимости от поведения других игроков. Теория игр помогает выбрать лучшие стратегии с учётом представлений о других участниках, их ресурсах и их возможных поступках.[1]

Теория игр — это раздел прикладной математики, точнее — исследования операций. Чаще всего методы теории игр находят применение в экономике, чуть реже в других общественных науках — социологии, политологии, психологии, этике, юриспруденции и других. Начиная с 1970-х годов её взяли на вооружение биологи для исследования поведения животных и теории эволюции. Очень важное значение она имеет для искусственного интеллекта и кибернетики, особенно с проявлением интереса к интеллектуальным агентам.

### Представление игр

Игры представляют собой строго определённые математические объекты. Игра образуется игроками, набором стратегий для каждого игрока и указания выигрышей, или платежей, игроков для каждой комбинации стратегий. Большинство кооперативных игр описываются характеристической функцией, в то время как для остальных видов чаще используют нормальную или экстенсивную форму. Характеризующие признаки игры как математической модели ситуации:

наличие нескольких участников;

неопределенность поведения участников, связанная с наличием у каждого из них нескольких вариантов действий;

различие (несовпадение) интересов участников;

взаимосвязанность поведения участников, поскольку результат, получаемый каждым из них, зависит от поведения всех участников;

наличие правил поведения, известных всем участникам.

### Экстенсивная форма

Игры в экстенсивной, или расширенной, форме[5] представляются в виде ориентированного дерева, где каждая вершина соответствует ситуации выбора игроком своей стратегии. Каждому игроку сопоставлен целый уровень вершин. Платежи записываются внизу дерева, под каждой листовой вершиной.

На рисунке слева — игра для двух игроков. Игрок 1 ходит первым и выбирает стратегию F или U. Игрок 2 анализирует свою позицию и решает — выбрать стратегию A или R. Скорее всего первый игрок выберет U, а второй — A (для каждого из них это оптимальные стратегии); тогда они получают соответственно 8 и 2 очка.

Экстенсивная форма очень наглядна, с её помощью особенно удобно представлять игры с более чем двумя игроками и игры с последовательными ходами. Если же участники делают одновременные ходы, то соответствующие вершины либо соединяются пунктиром, либо обводятся сплошной линией.

#### Нормальная форма

В нормальной, или стратегической, форме игра описывается платёжной матрицей. Каждая сторона (точнее, измерение) матрицы — это игрок, строки определяют стратегии первого игрока, а столбцы — второго. На пересечении двух стратегий можно увидеть выигрыши, которые получают игроки. В примере справа, если игрок 1 выбирает первую стратегию, а второй игрок — вторую стратегию, то на пересечении мы видим  $(-1, -1)$ , это значит, что в результате хода оба игрока потеряли по одному очку.

Игроки выбирали стратегии с максимальным для себя результатом, но проиграли, из-за незнания хода другого игрока. Обычно в нормальной форме представляются игры, в которых ходы делаются одновременно, или хотя бы полагается, что все игроки не знают о том, что делают другие участники. Такие игры с неполной информацией будут рассмотрены ниже.

#### Характеристическая функция

В кооперативных играх с трансферабельной полезностью, то есть возможностью передачи средств от одного игрока к другому, невозможно применять понятие индивидуальных платежей. Вместо этого используют так называемую характеристическую функцию, определяющую выигрыш каждой коалиции игроков. При этом предполагается, что выигрыш пустой коалиции равен нулю.

Основания такого подхода можно найти ещё в книге фон Неймана и Моргенштерна. Изучая нормальную форму для коалиционных игр, они рассудили, что если в игре с двумя сторонами образуется коалиция  $C$ , то против неё выступает коалиция  $N \setminus C$ . Образуется как бы игра для двух игроков. Но так как вариантов возможных коалиций много (а именно  $2^N$ , где  $N$  — количество игроков), то выигрыш для  $C$  будет некоторой характеристической величиной, зависящей от состава коалиции. Формально игра в такой форме (также называемая TU-игрой[7]) представляется парой  $(N, v)$ , где  $N$  — множество всех игроков, а  $v : 2^N \rightarrow \mathbb{R}$  — это характеристическая функция.

Подобная форма представления может быть применена для всех игр, в том числе без трансферабельной полезности. В настоящее время существуют способы перевести любую игру из нормальной формы в характеристическую, но преобразование в обратную сторону возможно не во всех случаях.

### Применение теории игр

Теория игр, как один из подходов в прикладной математике, применяется для изучения поведения человека и животных в различных ситуациях. Первоначально теория игр начала развиваться в рамках экономической науки, позволив понять и объяснить поведение экономических агентов в различных ситуациях. Позднее область применения теории игр была расширена на другие социальные науки; в настоящее время теория игр используется для объяснения поведения людей в политологии, социологии и психологии. Теоретико-игровой анализ был впервые использован для описания поведения животных Рональдом Фишером в 30-х годах XX века (хотя даже Чарльз Дарвин использовал идеи теории игр без формального обоснования). В работе Рональда Фишера не появляется термин «теория игр». Тем не менее, работа по существу выполнена в русле теоретико-игрового анализа. Разработки, сделанные в экономике, были применены Джоном Майнардом Смитом в книге «Эволюция и теория игр». Теория игр используется не только для предсказания и объяснения поведения; были предприняты попытки использовать теорию игр для разработки теорий этичного или эталонного поведения. Экономисты и философы применяли теорию игр для лучшего понимания хорошего (достойного) поведения. Вообще говоря, первые теоретико-игровые аргументы, объясняющие правильное поведения, высказывались ещё Платоном.

### Описание и моделирование

Первоначально теория игр использовалась для описания и моделирования поведения человеческих популяций. Некоторые исследователи считают, что с помощью определения равновесия в соответствующих играх они могут предсказать поведение человеческих популяций в ситуации реальной конфронтации. Такой подход к теории игр в последнее время подвергается критике по нескольким причинам. Во-первых, предположения, используемые при моделировании, зачастую нарушаются в реальной жизни. Исследователи могут предполагать, что игроки выбирают поведения, максимизирующее их суммарную выгоду (модель экономического человека), однако на практике человеческое поведение часто не соответствует этой предпосылке. Существует множество объяснений этого феномена — нерациональность, моделирование обсуждения, и даже различные мотивы игроков (включая альтруизм). Авторы теоретико-игровых моделей возражают на это, говоря, что их предположения аналогичны подобным предположениям в физике. Поэтому даже если их предположения не всегда выполняются, теория игр может использоваться как разумная идеальная модель, по аналогии с такими же моделями в физике. Однако, на теорию игр обрушился новый вал критики, когда в результате

экспериментов было выявлено, что люди не следуют равновесным стратегиям на практике. Например, в играх «Сороконожка», «Диктатор» участники часто не используют профиль стратегий, составляющий равновесие по Нэшу. Продолжаются споры о значении подобных экспериментов. Согласно другой точке зрения, равновесие по Нэшу не является предсказанием ожидаемого поведения, оно лишь объясняет, почему популяции, уже находящиеся в равновесии по Нэшу, остаются в этом состоянии. Однако вопрос о том, как эти популяции приходят к равновесию Нэша, остается открытым. Некоторые исследователи в поисках ответа на этот вопрос переключились на изучение эволюционной теории игр. Модели эволюционной теории игр предполагают ограниченную рациональность или нерациональность игроков. Несмотря на название, эволюционная теория игр занимается не столько вопросами естественного отбора биологических видов. Этот раздел теории игр изучает модели биологической и культурной эволюции, а также модели процесса обучения.

Нормативный анализ (выявление наилучшего поведения)

С другой стороны, многие исследователи рассматривают теорию игр не как инструмент предсказания поведения, но как инструмент анализа ситуаций с целью выявления наилучшего поведения для рационального игрока. Поскольку равновесие Нэша включает стратегии, являющиеся наилучшим откликом на поведение другого игрока, использование концепции равновесия Нэша для выбора поведения выглядит вполне обоснованным. Однако, и такое использование теоретико-игровых моделей подверглось критике. Во-первых, в некоторых случаях игроку выгодно выбрать стратегию, не входящую в равновесие, если он ожидает, что другие игроки также не будут следовать равновесным стратегиям. Во-вторых, знаменитая игра «Дилемма заключенного» позволяет привести ещё один контрпример. В «Дилемме заключенного» следование личным интересам приводит к тому, что оба игрока оказываются в худшей ситуации в сравнении с той, в которой они пожертвовали бы личными интересами.

## **Тема 5. Динамическое программирование**

Динамическое программирование в теории управления и теории вычислительных систем — способ решения сложных задач путём разбиения их на более простые подзадачи. Он применим к задачам с оптимальной подструктурой (англ.), выглядящим как набор перекрывающихся подзадач, сложность которых чуть меньше исходной. В этом случае время вычислений, по сравнению с «наивными» методами, можно значительно сократить.

Ключевая идея в динамическом программировании достаточно проста. Как правило, чтобы решить поставленную задачу, требуется решить отдельные части задачи (подзадачи), после чего объединить решения подзадач в одно общее решение. Часто

многие из этих подзадач одинаковы. Подход динамического программирования состоит в том, чтобы решить каждую подзадачу только один раз, сократив тем самым количество вычислений. Это особенно полезно в случаях, когда число повторяющихся подзадач экспоненциально велико.

Метод динамического программирования сверху — это простое запоминание результатов решения тех подзадач, которые могут повторно встретиться в дальнейшем. Динамическое программирование снизу включает в себя переформулирование сложной задачи в виде рекурсивной последовательности более простых подзадач.

#### Идея динамического программирования

Оптимальная подструктура в динамическом программировании означает, что оптимальное решение подзадач меньшего размера может быть использовано для решения исходной задачи. К примеру, кратчайший путь в графе из одной вершины (обозначим  $s$ ) в другую (обозначим  $t$ ) может быть найден так: сначала считаем кратчайший путь из всех вершин, смежных с  $s$ , до  $t$ , а затем, учитывая веса ребер, которыми  $s$  соединена со смежными вершинами, выбираем лучший путь до  $t$  (через какую вершину лучше всего пойти). В общем случае мы можем решить задачу, в которой присутствует оптимальная подструктура, проделывая следующие три шага.

#### Разбиение задачи на подзадачи меньшего размера.

Нахождение оптимального решения подзадач рекурсивно, проделывая такой же трехшаговый алгоритм.

Использование полученного решения подзадач для конструирования решения исходной задачи.

Подзадачи решаются делением их на подзадачи ещё меньшего размера и т. д., пока не приходят к тривиальному случаю задачи, решаемой за константное время (ответ можно сказать сразу). К примеру, если нам нужно найти  $n!$ , то тривиальной задачей будет  $1! = 1$  (или  $0! = 1$ ).

Перекрывающиеся подзадачи в динамическом программировании означают подзадачи, которые используются для решения некоторого количества задач (не одной) большего размера (то есть мы несколько раз проделываем одно и то же). Ярким примером является вычисление последовательности Фибоначчи,  $F_3 = F_2 + F_1$  и  $F_4 = F_3 + F_2$  — даже в таком тривиальном случае вычисления всего двух чисел Фибоначчи мы уже посчитали  $F_2$  дважды. Если продолжать дальше и посчитать  $F_5$ , то  $F_2$  посчитается ещё два раза, так как для вычисления  $F_5$  будут нужны опять  $F_3$  и  $F_4$ . Получается следующее: простой рекурсивный подход будет расходовать время на вычисление решения для задач, которые он уже решал.



Чтобы избежать такого хода событий мы будем сохранять решения подзадач, которые мы уже решали, и когда нам снова потребуется решение подзадачи, мы вместо того, чтобы вычислять его заново, просто достанем его из памяти. Этот подход называется кэширование. Можно проделывать и дальнейшие оптимизации — например, если мы точно уверены, что решение подзадачи нам больше не потребуется, можно выкинуть его из памяти, освободив её для других нужд, или если процессор простаивает и мы знаем, что решение некоторых, ещё не посчитанных подзадач, нам понадобится в дальнейшем, мы можем решить их заранее.

Подводя итоги вышесказанного можно сказать, что динамическое программирование пользуется следующими свойствами задачи:

- перекрывающиеся подзадачи;
- оптимальная подструктура;
- возможность запоминания решения часто встречающихся подзадач.

Динамическое программирование обычно придерживается двух подходов к решению задач:

нисходящее динамическое программирование: задача разбивается на подзадачи меньшего размера, они решаются и затем комбинируются для решения исходной задачи. Используется запоминание для решений часто встречающихся подзадач.

восходящее динамическое программирование: все подзадачи, которые впоследствии понадобятся для решения исходной задачи просчитываются заранее и затем используются для построения решения исходной задачи. Этот способ лучше нисходящего программирования в смысле размера необходимого стека и количества вызова функций, но иногда бывает нелегко заранее выяснить, решение каких подзадач нам потребуется в дальнейшем.

Языки программирования могут запоминать результат вызова функции с определенным набором аргументов ( мемоизация), чтобы ускорить «вычисление по имени». В некоторых языках такая возможность встроена (например, Scheme, Common Lisp, Perl), а в некоторых требует дополнительных расширений (C++).

Известны сериальное динамическое программирование, включённое во все учебники по исследованию операций, и несериальное динамическое программирование (НСДП), которое в настоящее время слабо известно, хотя было открыто в 1960-х годах.

Обычное динамическое программирование является частным случаем несериального динамического программирования, когда граф взаимосвязей переменных — просто путь. НСДП, являясь естественным и общим методом для учета структуры задачи оптимизации, рассматривает множество ограничений и/или целевую функцию как рекурсивно

вычислимую функцию. Это позволяет находить решение поэтапно, на каждом из этапов используя информацию, полученную на предыдущих этапах, причём эффективность этого алгоритма прямо зависит от структуры графа взаимосвязей переменных. Если этот граф достаточно разрежен, то объём вычислений на каждом этапе может сохраняться в разумных пределах.

Одним из основных свойств задач, решаемых с помощью динамического программирования, является аддитивность. Неаддитивные задачи решаются другими методами. Например, многие задачи по оптимизации инвестиций компании являются неаддитивными и решаются с помощью сравнения стоимости компании при проведении инвестиций и без них.

Классические задачи динамического программирования

Задача о наибольшей общей подпоследовательности: даны две последовательности, требуется найти самую длинную общую подпоследовательность.

Задача поиска наибольшей увеличивающейся подпоследовательности: дана последовательность, требуется найти самую длинную возрастающую подпоследовательность.

Задача о редакционном расстоянии (расстояние Левенштейна): даны две строки, требуется найти минимальное количество стираний, замен и добавлений символов, преобразующих одну строку в другую.

Задача о вычислении чисел Фибоначчи

Задача о порядке перемножения матриц: даны матрицы  $A_1, \dots, A_n$ , требуется минимизировать количество скалярных операций для их перемножения.

Задача о выборе траектории

Задача последовательного принятия решения

Задача об использовании рабочей силы

Задача управления запасами

Задача о ранце: из неограниченного множества предметов со свойствами «стоимость» и «вес» требуется отобрать некоторое число предметов таким образом, чтобы получить максимальную суммарную стоимость при ограниченном суммарном весе.

Алгоритм Флойда — Уоршелла: найти кратчайшие расстояния между всеми вершинами взвешенного ориентированного графа.

Алгоритм Беллмана — Форда: найти кратчайший путь во взвешенном графе между двумя заданными вершинами.

Максимальное независимое множество вершин в дереве: дано дерево, найти максимальное множество вершин, никакие две из которых не связаны ребром.

## Тема 6. Сетевое планирование

Сетевое планирование – метод управления, основанный на использовании математического аппарата теории графов и системного подхода для отображения и алгоритмизации комплексов взаимосвязанных работ, действий или мероприятий для достижения четко поставленной цели.

Разработаны в начале 50-х г. XX в.

Наиболее известны практически одновременно и независимо разработанные метод критического пути - МКП и метод оценки и пересмотра планов - PERT.

Применяются для оптимизации планирования и управления сложными разветвленными комплексами работ, требующими участия большого числа исполнителей и затрат ограниченных ресурсов.

Основная цель сетевого планирования - сокращение до минимума продолжительности проекта.

Задача сетевого планирования состоит в том, чтобы графически, наглядно и системно отобразить и оптимизировать последовательность и взаимозависимость работ, действий или мероприятий, обеспечивающих своевременное и планомерное достижение конечных целей. Для отображения и алгоритмизации тех или иных действий или ситуаций используются экономико-математические модели, которые принято называть сетевыми моделями, простейшие из них - сетевые графики. С помощью сетевой модели руководитель работ или операции имеет возможность системно и масштабно представлять весь ход работ или оперативных мероприятий, управлять процессом их осуществления, а также маневрировать ресурсами.

Наиболее распространенными направлениями применения сетевого планирования являются:

целевые научно-исследовательские и проектно-конструкторские разработки сложных объектов, машин и установок, в создании которых принимают участие многие предприятия и организации;

планирование и управление основной деятельностью разрабатывающих организаций;

планирование комплекса работ по подготовке и освоению производства новых видов промышленной продукции;

строительство и монтаж объектов промышленного, культурно-бытового и

жилищного назначения;

реконструкция и ремонт действующих промышленных и других объектов;

планирование подготовки и переподготовки кадров, проверка исполнения принятых решений, организация комплексной проверки деятельности предприятий, объединений, строительного-монтажных организаций и учреждений.

Использование методов сетевого планирования способствует сокращению сроков создания новых объектов на 15-20%, обеспечению рационального использования трудовых ресурсов и техники.

## **Тема 7. Теория массового обслуживания**

Наряду с другими экономико-математическими методами в экономическом анализе используется теория массового обслуживания. Она применяется, в частности, в розничной торговле при анализе количества обслуживаемых покупателей и продолжительности их обслуживания (при условии высокого качества их обслуживания). На эти показатели оказывают влияние различные факторы (переменные величины). Они взаимодействуют между собой в условиях процесса обслуживания покупателей, носящего стохастический характер.

На основе теории массового обслуживания выбирается оптимальный вариант организации торгового обслуживания населения, обеспечивающий минимальное время обслуживания при минимизации затрат и высоком качестве обслуживания населения.

Рассматриваемая теория находит применение и в других отраслях экономики. Теория массового обслуживания заключается в том, что на базе теории вероятностей выводятся математические методы анализа процессов массового обслуживания, а также методы оценки качества работы обслуживающих систем.

При всем своём разнообразии процессы в системах массового обслуживания имеют общие черты:

Требование на обслуживание не регулярно случайно поступает на канал обслуживания и в зависимости от его занятости, продолжительности обслуживания образуют очередь требований.

Теория массового обслуживания изучает статистические закономерности поступления. И на этой основе вырабатывает решения, то есть такие характеристики системы обслуживания, при которых затраты времени на ожидание в очереди и на простой каналов обслуживания были бы наименьшими. (если мало каналов обслуживания — то образуются большие очереди, и наоборот, если много каналов обслуживания, то очередей нет, но при этом каналы обслуживания работают не рационально, так как часть из них простаивает без работы).

Теория массового обслуживания — это прикладная область теории случайных процессов.

Предметом исследования теории массового обслуживания являются вероятностные

модели физических систем обслуживания, в которых случайные и не случайные моменты времени возникают заявки на обслуживание и имеются устройства на обработку данных заявок.

Теория массового обслуживания целиком базируется на теории вероятности и на математической статистике. В определенной степени она связана с распределением Пуассона, которое описывает вероятность числа появлений в заданном интервале времени какого-либо события. Например, появление покупателя у прилавка, если известно, что появление события зависит от того давно ли оно появлялось в последний раз и сколько раз и когда именно случалось до этого.