## **«Задача линейного программирования; Симплекс-метод»**

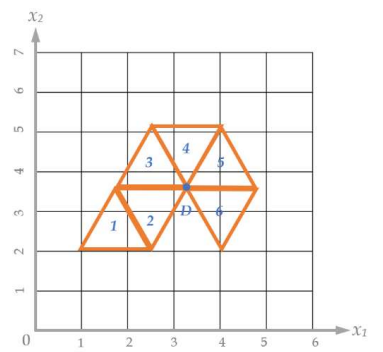
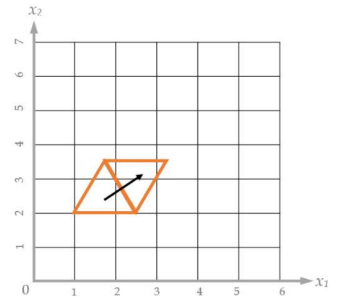
Задачей линейного программирования называют **поиск экстремума (минимума или максимума)** линейной целевой функции многих переменных.

Для того, чтобы составить математическую модель задачи линейного программирования, необходимо выполнить следующие действия, исходя из условий задачи:

1. Записать исходные данные.
2. Определить переменные.
3. Сформулировать целевую функцию.
4. Записать систему ограничений.

В общем случае задача ЛП имеет вид:

Для решения задачи ЛП применяют симплекс метод. **Симплекс** – это геометрическая фигура, имеющая  **угол в мерном пространстве** (треугольник в двухмерном) с произвольной длиной ребра. Симплекс помещается в произвольную начальную точку, далее для каждой вершины симплекса вычисляется значение целевой функции. В случае, если стоит задача минимизации, то отбрасывается вершина с наибольшим значением целевой функции, в случае задачи максимизации – наоборот. Далее симплекс «перекидывается» через ребро с оставшимися точками **до тех пор, пока симплекс не будет вращаться относительно одной из точек**.



## **«Производственная задача»**

Производственная задача является частным случаем задачи линейного программирования. Её **цель** – определение такого количества выпускаемой продукции каждого вида, чтобы прибыль от продажи была максимальной с учётом складских запасов.

Для каждого вида продукции необходимо определить количество компонентов каждого типа, используемого в изготовлении и стоимость реализации. В качестве ограничений используются запасы каждого типа компонентов на складе. Количество произведённой продукции обозначается переменными.

Функция прибыли имеет вид:

Где – стоимость товара, – количество товара.

Ограничения имеют вид:

Где – компонент, используемый в производстве, – количество данного компонента на складе. В сущности, ограничения говорят о том, что общее количество используемых в производстве компонентов должно быть меньше или равно количеству на складе.

Для решения задачи в MS Excel используется надстройка «Поиск решения» с указанными значениями и максимизацией целевой функции с использованием **симплекс-метода.**

## **«Классическая транспортная задача»**

Транспортная задача является одним из типов задачи линейного программирования. **Её цель** – в нахождении оптимального плана перевозок однотипной продукции от поставщиков потребителям, обеспечивающего минимальные расходы.

Задача называется замкнутой (сбалансированной), если запас на складах () полностью удовлетворяет спрос (), и открытой (несбалансированной) в противном случае. Данное равенство является ограничением транспортной задачи:

Далее формируются три матрицы: матрица-столбец, в котором записано количество товаров у каждого поставщика , матрица-строка, в которой записан спрос каждого потребителя и матрица, в ячейках которой записана стоимость доставки со склада потребителю - . Нулевая стоимость означает, что перевезти товар невозможно.

Следующим шагом составляется матрица переменных – количество перевезённого от i-го поставщика j-му потребителю.

Далее формируем целевую функцию:

И три ограничения:

1. Запасы всех поставщиков должны быть вывезены:
2. Потребности всех магазинов должны быть удовлетворены:
3. Количество перевезённого груза неотрицательно:

Задача решается **симплекс-методом**. Полученное решение является планом поставок из складов производителей потребителям.

В случае, если задача является открытой (несбалансированной), то:

1. Если возникает дефицит производственных мощностей, то вводится фиктивный склад.
2. Если возникает профицит производственных мощностей, то вводится фиктивный магазин.

## **«Задача о назначениях»**

Задача о назначениях является частным случаем производственной задачи и применяется при распределении задач, стажировок, назначений между некоторым количеством лиц, кредитном скоринге и распределению клиентов.

Формальная постановка задачи состоит из следующих элементов:

1. Матрица-строка означает количество задач \ стажировок \ назначений.
2. Матрица-столбец означает количество задач разных типов \ назначений \ претендентов
3. Матрица означает эффективность лица при решении данной задачи \ количество баллов по вступительным испытаниям и т. д.

Изображение выглядит как текст, шкафчик

Автоматически созданное описание

Далее формируется матрица переменных **булевского типа – главная особенность задачи о назначениях.**

Задача также должна быть сбалансирована:

Целевая функция должна стремиться к максимуму (наибольшей эффективности, а ограничения сформулированы на количество задач\назначений для каждого лица (по матрице-строке):

Задача решается симплекс-методом. Полученная матрица соответствует наиболее эффективному распределению задач\стажировок\т.д.

## **«Задача о кратчайшем пути на графе»**

Задача нахождения кратчайшего пути на графе может решаться не только алгоритмически с помощью Алгоритма Дейкстры, но и как задача линейного программирования:

Для этого преобразуем граф ( – количество вершин, – множество ребер) в матрицу, где вершины – номера столбцов\строк, вес ребер – элементы внутри матрицы. Вес ребра задается матрицей смежности:

Где -число вершин в графе.

Так как наша задача – найти по каким рёбрам проходит кратчайший маршрут из вершины А в вершину В.

Для этого создадим матрицу переменных искомого маршрута , где 1 будет означать выбор данного ребра. Таким образом сведём задачу к задаче линейного программирования. Для этого сформируем целевую функцию – минимальную длину пути, суммируя произведения единиц из матрицы маршрута и значений из матрицы смежности:

И зададим ограничения:

1. Маршрут должен начинаться в пункте А и заканчиваться в В (в строке А и в столбце В должна стоять 1):
2. Переход должен осуществляться только по существующим рёбрам:
3. Ограничение на связность маршрута: вход в вершину всегда должен соответствовать выходу из неё:
4. Вход и выход в каждую вершину производится не более одного раза (для избежания петель):

Данная задача решается симплекс-методом. Полученная матрица соответствует кратчайшему маршруту из А в В.

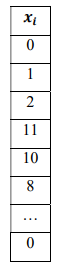
## **«Задача коммивояжера»**

Задача коммивояжера сводится к нахождению пути обхода всех вершин графа за минимальное время, посещая каждую вершину лишь один раз. Иными словами, необходимо найти на графе все гамильтоновы циклы минимальной длины. Для этого представим маршруты переезда между адресами в виде графа, где вершина А будет началом и концом маршрута, а вес ребра – время\стоимость переезда.

Пусть дан граф ( – количество вершин, – множество ребер). Вес ребра задается матрицей смежности:

Где -число вершин в графе.

Необходимо найти замкнутый цикл минимальной длины. Для этого составим вектор-столбец для переменных, в который мы будем записывать номер вершины в порядке её посещения. При этом первый и последний элемент вектора – начальная точка А, так как нам необходимо «замкнуть» маршрут



Следующим шагом сформируем целевую функцию – минимальную суммарную длину всех дуг пути:

Определим ограничения:

1. Маршрут должен начинаться и заканчиваться в А:
2. Переход должен осуществляться только по существующим ребрам:
3. Вершины не должны повторяться:
4. Искомые переменные целочисленные:

Для решения данной задачи симплекс метод или метод ОПГ не представляется возможным, поэтому задача решается **эволюционным методом.**

## **«Решение антагонистической игры»**

Антагонистические игры, являются играми «с нулевой суммой» игроки борются друг с другом, выигрыш одного равен проигрышу другого, то есть выигрыши игроков противоположны.

Для решения необходимо знать стратегии каждого игрока и составить матрицу, где будет отражен успех в процентном соотношении для каждого игрока при выборе той или иной стратегии. Такая матрица называется платежная. Количество строк и столбцов будет зависеть от количества выбранных стратегий игроками. При выборе Игроком А 3 стратегий и игроком В 4 стратегий матрица будет выглядеть таким образом:

Для того, чтобы нам найти чистую стратегию, т. е. такую стратегию, которая неизменно будет использоваться и приведет к победе воспользуемся принципом максимина. Данный принцип базируется на обеспечении максимально гарантированных выигрышей игрока А, какие бы стратегии ни применял в ответ игрок В и, а для игрока В такой принцип будет называться минимаксом, в общем виде математические формулы выглядят следующим образом

Далее найдем оптимальную смесь стратегий для игроков А и В. Для этого нам потребуется определить цену игры. Цена игры – это совокупность выигрышей (может высчитываться, как сумма выигрышей каждого из ходов или назначаться в конце игры). Для решения придется транспонировать платежную матрицу. Оптимальное решение игры называется «седловой точкой» – точкой, в которой сходятся минимум функции АХВ и максимум функций DXC. Для того, чтобы узнать являются ли наша смесь стратегий оптимальной нам нужно будет воспользоваться симплекс методом.

Исходные данные будут браться из транспонированной матрицы Искомые переменные – (замененные вероятности). В качестве целевой функции выступает максимальная величина входного/выходного потока: 𝐿 = → min, а ограничения будут такие, что все строки матрицы должны быть . Далее осуществляется оптимизация с помощью симплекс-метода, выявляются искомые переменные, обеспечивающие минимум целевой функции. Производится обратная замена переменных, после которой мы получаем искомые вероятности смешивания стратегий и цену игры:

Решение данной задачи покажет оптимальный вариант для игрока А. Поиск оптимального решения для игрока В осуществляется через решение обратной задачи линейного программирования. Исходными данными в этом случае будет не транспонированная матрица А.

## **«Решение биматричной игры»**

Биматричные игры – игры с двумя матрицами, при которых у каждого игрока различный пул преференций и выход, а победа одного не всегда означает проигрыш другого.

Равновесие по Нэшу – такое сочетание стратегий, при котором игроки не доверяют друг другу и выбирают такой вариант, который гарантированно позволит им выйти с минимальными потерями. При этом каждому из них не выгодно отступать от выбранной стратегии вне зависимости от обстоятельств.

Игры протекают так же в 2 стратегиях. Чистых и смешанных.  
**Чистая стратегия.** Реализуется путем равновесия по Нэшу: игрок А ищет наиболее выгодные варианты в своей матрице для игрока В (максимумы по столбцам матрицы А). Игрок В также ищет наиболее выгодные для оппонента варианты в своей матрице (максимумы по строкам матрицы В). В тех ячейках, где отмеченные выгодные варианты совпали в обоих матрицах и наблюдается ситуация равновесия по Нэшу.

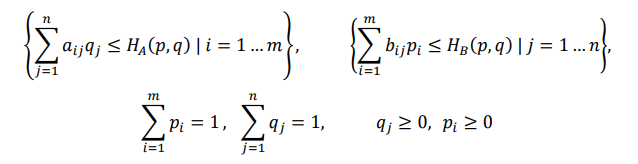
**Для смешанной стратегии** необходимо решить задачу нелинейного программирования. Целевая функция в данном случае нелинейного вида, и представляет собой сумму выигрышей обоих игроков.

где ситуации равновесия:

*Изображение выглядит как текст

Автоматически созданное описание*

Ограничения:

**

Решение задачи происходит методом ОПГ (обобщённого приведённого градиента). В результате для каждого игрока определяется оптимальная смесь стратегий.

## **«Решения матричной игры с природой в условиях риска»**

В данном типе игр в качестве второго игрока выступает «природа», т. е. такой игрок, который живет по своим законам, на которые мы повлиять никак не можем. От того, знаем ли мы поведение «природы», зависит тип игры:

1. Игра с риском (или с частичной неопределенностью) – в данном случае известно (хотя бы приблизительно) с какой вероятностью наступает каждое состояние природы.
2. Игра в условиях неопределённости – в данном случае вероятности состояний природы неизвестны.

В качестве исходных данных необходима платежная матрица, в которой будут расписаны стратегии игрока и «природы», а также вероятность их наступления.

Для решения задачи в условиях риска необходимо воспользоваться критериями Байеса, Лапласа и Гермейера.

**Критерий Байеса** выявляет оптимальное решение на основе средневзвешенных значений выигрыша каждой стратегии игрока А, для этого воспользуемся формулой^

**Критерий Лапласа** применяется в том случае, если вероятности наступления состояний природы неизвестны. Мы принимаем их как равновероятные и вычисляем среднеарифметическое (а не средневзвешенное, как в критерии Байеса) значение выигрыша игрока А по строке:

**Критерий Гермейера** требует построения новой матрицы. Элементы этой новой матрицы содержат значения выигрыша, умноженного на вероятность этого выигрыша - . Далее, используем принцип максимина для определения наибольшего гарантированного выигрыша.

Критерий Гермейера показывает гарантированный выигрыш в самом неблагоприятном состоянии природы.

## **«Решения матричной игры с природой в условиях неопределённости»**

В данном случае количество рассматриваемых критериев значительно больше, так как задача, стоящая перед нами, сложнее в силу того, что мы ничего не знаем об окружающей среде, т. е. вероятность состояний природы нам неизвестна. Для решения этой задачи используются следующие критерии: пессимизма, оптимизма, Вальда, Гурвица и Сэвиджа

**Критерий пессимизма и оптимизма** необходимы для оценки нижней и верхней границ игры – сколько мы максимум можем потратить и сколько максимум заработать. Это возможно определить с помощью критериев «Пессимизма» (для нижней границы) и «Оптимизма» (для верхней границы)

**Критерий Вальда** предлагает искать самый надежный сценарий по принципу «Максимина» (искать гарантированную прибыль от каждой стратегии).

**Критерий Гурвица** используется для учёта склонности человека к риску. Критерий основан на линейной свертке для нахождения компромиссного варианта, настраивая весовой коэффициент пессимизма и оптимизма, который регулирует отношение к риску.

Отношение к риску – это функция полезности из одноименной теории полезности, которая определяет, насколько какой-то результат является для нас более полезным.

**Критерий Сэвиджа** определяет стратегию с наименьшим недополученным выигрышем. Для реализации данного критерий потребуется построить матрицу упущенной выгоды (сколько могли бы получить максимально, но не получили). Среди элементов исходной матрицы ищем максимальное значение по столбцам. Заполняем матрицу упущенной выгоды (или матрицу рисков) соответствующими значениями – разностями максимума по столбцу и значения текущей ячейки:

Минимальная упущенная выгода определяется по критерию минимакса:

Если в ячейках матрицы стоят 0 – значит, упущенной выгоды нет. Полученная стратегия гарантирует минимальную упущенную выгоду.

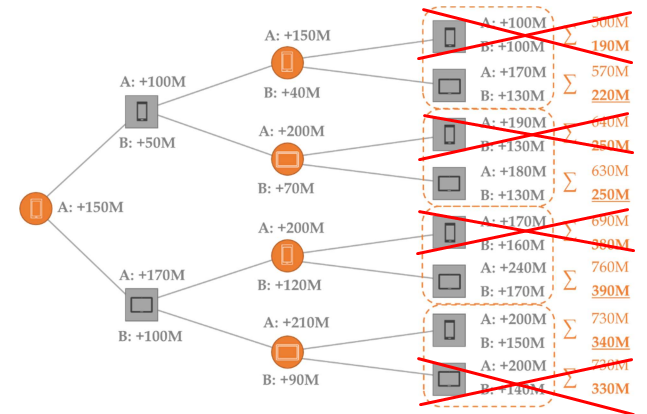
## **«Решения позиционной игры с противником на дереве решений»**

Позиционная игра – игра, в которой противники ходят по очереди. Для решения такого типа задач необходимо будет построить древо игры. Дерево игры – это совокупность всевозможных ходов игроков из всех возможных позиций и выигрышей/проигрышей, получаемых в результате этих ходов. Конечным элементам дерева игры (листьям) соответствует цена игры данной совокупности ходов, рассчитанная по формуле:

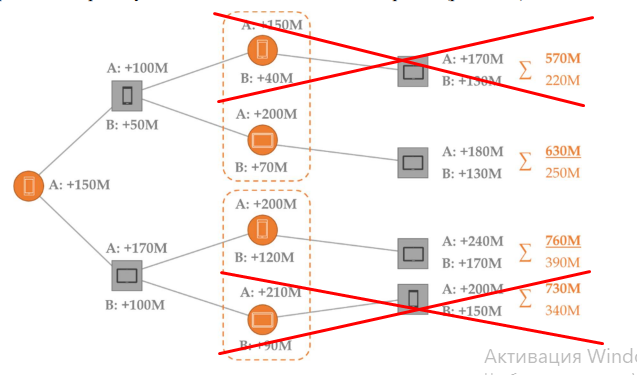
– цена игры (выигрыш) на k-том ходу, – число ходов.

Для игры с противником наиболее верным решением будет воспользоваться алгоритмом Куна. **Алгоритм Куна** – это алгоритм, который позволяет выявить оптимальную стратегию путем сравнения исходов игры на каждой развилке и отсечением непродуктивных веток.

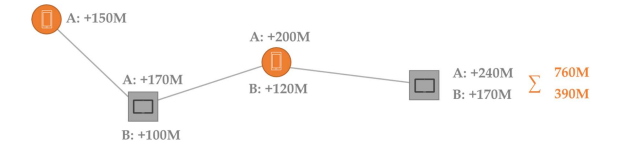
Прямым ходом алгоритма Куна (двигаемся от корня к листьям) суммируем все выигрыши каждого игрока на каждой ветке и указываем их на листьях (мы видим связанную пару исходов игры для двух игроков).



Обратным ходом алгоритма (направление движения – от листьев к корню) отсекаем наихудшие варианты на каждом уровне с позиции выбирающего игрока. При этом мы ориентируется не на сиюминутный выигрыш, а на суммарный исход. Если суммарные выигрыши на развилке равнозначны, то выбираем тот вариант, который будет наименее выгоден для соперника.

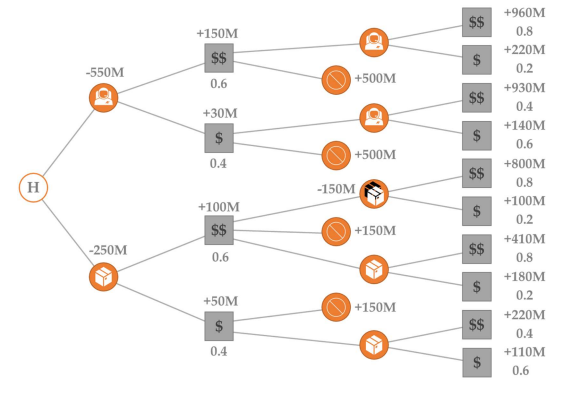


Чем выше к корню мы поднимаемся, тем больше веток мы отсекаем. По окончании работы алгоритма мы придём к корню и оставим единственную необходимую ветку стратегии, которая будет для нас оптимальной.



## **«Решения позиционной игры с природой на дереве решений**

В позиционной игре с природой присутствует один игрок и природа. Игрок имеет свои стратегии, а природа – состояния. Каждое состояние природы определено определённой вероятностью, которые в сумме дают единицу.

Для определения оптимальной стратегии строится дерево решения. Из начала дерева первым идут ходы игрока, изменяющие цену игры. Далее на каждое решение игрока приходятся все состояния природы. После – опять ход игрока (отвечаем на все состояния природы), затем – состояния природы. Пример дерева решений ниже: 

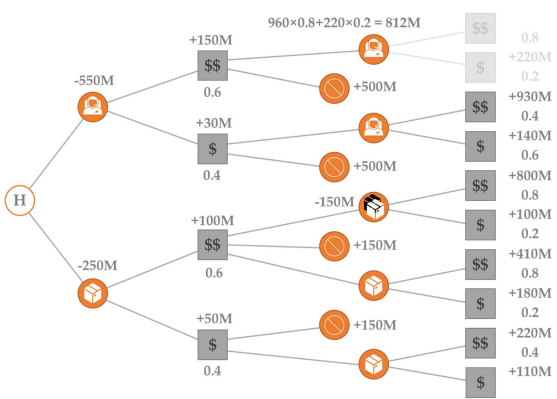
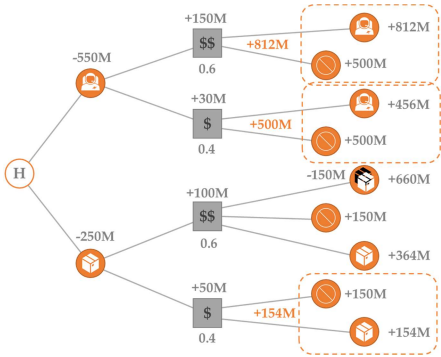
**Алгоритм поиска решения.**

Для решения позиционной игры с природой, в силу того, что во время ходов мы ничего не знаем наверняка применяются средневзвешенные значения выигрыша – все ветви природы «свертываются» по формуле:

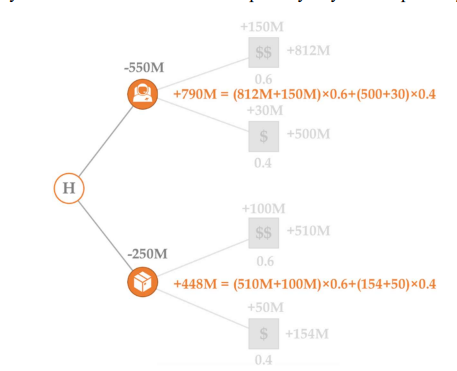
Где – вероятность состояния природы,

– выигрыш при этом состоянии проироды,

– количество состояний природы.

В результате все состояния природы оказываются «свёрнутыми» и мы можем выбрать стратегию с наибольшей ценой игры в каждом состоянии природы. После выбора наилучшего варианта в каждой паре вновь повторяем шаги до тех пор, пока не дойдём до корня. Средневзвешенные значениями руководствовуемся только тогда, когда действует природа (потому что игрок действует осознанно и мы можем выбрать лучшую альтернативу). В результате дерево решений сворачивается до корня и мы можем выбрать оптимальную стратегию игрока:



Выбирается она следующим образом: на первом ходу (игрок) выбираем ту стратегию, которая даст нам наибольший выигрыш. После ходит природа. В зависимости от того, какое состояние природы нам выпало, мы вновь выбираем стратегию с самым большим выигрышом.

## **«Понятие линейной регрессии, метод наименьших квадратов»**

Регрессия – односторонняя стохастическая зависимость, устанавливающая соответствие между случайными величинами. Существует множество видов регрессии: линейная, экспоненциальная, логарифмическая, полиноминальная, кубическая и другие. **Линейная регрессия** – простейший вид регрессии – линейная функция некоторой прямой, отражающей характер поведения всей выборки (статистики), имеющая следующий вид:

Где – тренд или истинная зависимость, а – шум (случайные колебания). Ключевым является вопрос нахождения коэффициентов и линейной регрессии , которые определяют зависимость между переменными и .

Для нахождения неизвестных коэффициентов используется **метод наименьших квадратов**. Суть метода в нахождении таких параметров прямой, чтобы сумма квадратов расстояний до этой прямой была минимальной. При этом сумма отклонений должна составлять нуль.

Для этого, для каждой точки, описываемой , применяется следующая функция – расстояние до каждой точки, которую необходимо свести к минимуму.

Для нахождения значений коэффициентов запишем функцию :

И найдём её минимумы. Для этого определим градиент , состоящих из частных производных. Определим точки экстремума, в которых . Для этого приравняем частные производные к нулю:

Запишем средние арифметические в следующем виде:

Подставляя средние арифметические, умножаем на и получаем:

Решая относительно коэффициентов, имеем:

Таким образом, уравнение линейной регрессии имеет вид:

## **«Доверительные интервалы, прогнозирование с помощью лин. регрессии»**

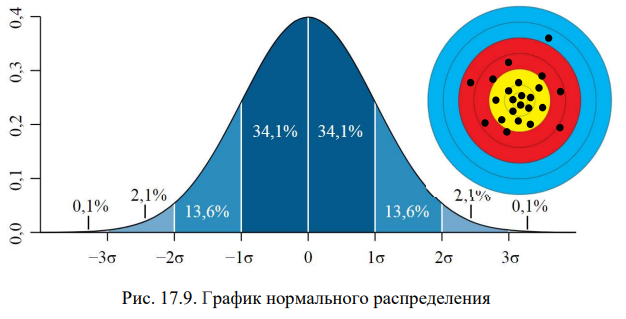
Прогнозирование с помощью линейной регрессии выполняется по следующему алгоритму:

1. Ввести исходные данные (выборку) и определить для них коэффициенты линейной регрессии вида
2. Провести фильтрацию выборки, исключив значения, не попадающие в допустимый интервал , где – среднеквадратичное отклонение по выборке.
3. Определить для отфильтрованной выборки коэффициенты линейной регрессии .
4. Для нового тренда определить доверительные интервалы , , .
5. Построить прогнозные значения исходя из полученного уравнения линейной регрессии и определить доверительные интервалы для полученных значений.

Рассмотрим подробнее доверительные интервалы. **Доверительный интервал** – это интервал, в который, с заданной вероятностью, попадают значения из некой выборки. Для вычисления доверительных интервалов необходимо определить дисперсию как меру разброса данных в выборке исходя из формулы:

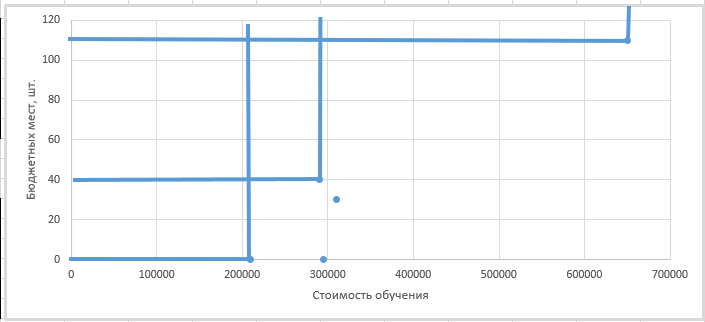
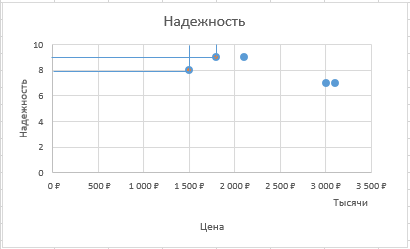
И среднеквадратичное (или стандартное) отклонение по формуле . Чем больше дисперсия, тем больше разброс, а значит полученный прогноз будет неточным и наоборот. Прогнозируемое значение будет колебаться по определённому закону распределения. По центральной предельной теореме Ляпунова, нормальное распределение Гаусса является самым часто встречающимся распределением, а значит возможно определить, сколько точек попадает в самые используемые доверительные интервалы:

1. В доверительный интервал [-σ; +σ] попадает ~68% всех точек.
2. В доверительный интервал [-2σ; +2σ] попадает ~95% всех точек.
3. В доверительный интервал [-3σ; +3σ] попадает ~99% всех точек.

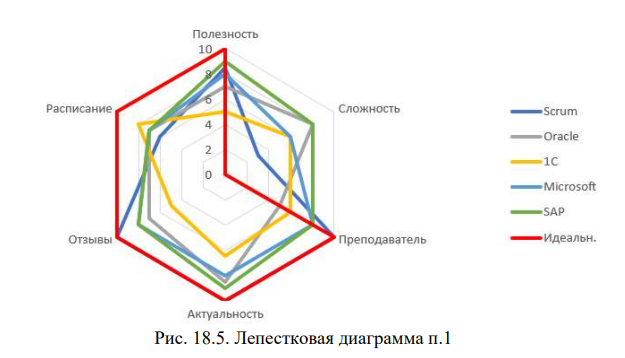


## **«Многокритериальная оптимизация по Парето»**

Для решения данной задачи вводим исходные данные. Затем определелим основные критерии выбора (удобно брать 2 критерия). Далее представим эти точки на плоскости. Выделим среди данных точек не улучшаемые, то есть для каждой точки выделим подпространство в направлении к максимуму или к минимуму, в зависимости от критерия



Затем задаем идеальные значения по каждому критерию и создаем лепестковую диаграмму.



По данным диаграммам можно определить наиболее подходящих кандидатов для выбора.

## **«Многокритериальная оптимизация методом линейной свёртки»**

Многокритериальная оптимизация методом линейной свёртки заключается в свёртывании всех (или нескольких) критериев в один.

В общем случае линейная свёртка имеет вид:

Где: – значимость критерия.

Для этого определяем значимость каждого критерия , выставив оценки по определённой шкале (от 1 до 10 или от 0 до 1 и т. д.). Далее необходимо нормировать оценки значимости каждого критерия по формуле:

Затем для каждого критерия необходимо определить параметр направленности :

И произвести линейную свёртку по формуле:

Затем необходимо выбрать вариант, получивший наибольшее значение по линейной свёртке. Он и будет оптимальным выбором.

## **«Многокритериальная оптимизация методом идеальной точки»**

Многокритериальная оптимизация методом «идеальной точки» позволяет определить выбор, наиболее близкий к некому «идеальному» значению.

Алгоритм действий следующий:

Во-первых, для каждого критерия определяются наиболее предпочтительные для нас значения. З

Во-вторых, определяем параметр , который минимизирует расстояние до идеальной точки по евклидовой метрике:

В-третьих, для каждого варианта выбора получаем некое значение параметра , означающего удалённость данного варианта от идеального. Следовательно, чем меньшее значение , тем ближе вариант к идеальному значению. Стало быть, выбираем тот вариант, у которого значение параметра минимально.

## **«Многокритериальная оптимизация методом контрольных показателей»**

Многокритериальная оптимизация методом контрольных показателей предполагает выбор такого варианта, который находится в определённых границах от контрольных значений.

Алгоритм действий следующий:

**Во-первых**, необходимо определить для каждого критерия контрольный показатель, находящийся в диапазоне минимального и максимального значений каждого критерия. Выбранные контрольные точки отражают границы, вне которых варианты рассматриваться не будут.

**Во-вторых**, введём глобальный критерий , максимизирующий минимальное расстояние от до границы :

Здесь для критериев, стремящихся к максимуму, применяется первая часть выражения в скобках, а к минимуму – вторая. То есть, варианты, у которых присутствуют значения рассматриваться не будут, так как значения их критериев выходят за границы.

**В-третьих**, вычисляя, находим такое , что все функции , наиболее удалены от своих границ . То есть оптимальным вариантом будет такой, у которого критерий максимален.

## **«Разработка регламента экспертных оценок, подбор экспертов»**

Для разработки **регламента проведения** экспертных оценок необходимо определить:

**Количество туров экспертизы:**

1. Один – эксперты однократно выбирают решение
2. Несколько – проводится несколько итераций экспертизы
3. Неопределенное количество – до тех пор, пока не будет найдено компромиссное решение (эксперты не сойдутся на чем-то одном)

**Степень общения экспертов:**

1. Отсутствует – мнения не зависят друг от друга
2. Заочное анонимное – исключается воздействие авторитетов
3. Заочное личностное – авторитеты могут иметь ограниченное влияние
4. Очное с ограничениями -
5. Очное без ограничений.

При разработке **регламента сбора** экспертных оценок необходимо учитывать:

1. Степень подключения экспертов к работе:
   1. Использование сразу всех экспертов.
   2. Поочередное подключение экспертов (например, постепенно подключая более дорогих/удаленных экспертов, если решение выработать не удается).
2. Степень рассогласованности мнений:
   1. Хорошо, если мнения экспертов не согласованны–мы получаем большее количество различных вариантов, среди которых найдется верный.2)
   2. Плохо, если мнения экспертов не согласованны получает отсутствие единства мнений, возможно эксперты некомпетентны или задача не точна.

При разработке **регламента интерпретации** результатов экспертных оценок, нужно помнить о двух исключениях из правил:

1. Догма согласованности – мнение «диссидентов» (малой части экспертов) может оказаться правильным.
2. Догма одномерности – не все результаты можно выразить одним числом или признаком.

Подбор экспертов включает в себя два основных этапа:

1. Составление списка возможных экспертов:
   1. Самостоятельное составление реестра экспертов;
   2. Набор по принципу «снежного кома» - каждый приглашенный эксперт рекомендует еще нескольких.
2. Выбор экспертов необходимой компетенции (в зависимости от сложности задачи, имеющегося бюджета и др.) следующими способами:
   1. По результатам научной деятельности (нужны эксперты для оценки экспертов);
   2. Самооценка экспертов (риск завышенной или заниженной);
   3. Оценка друг друга (возможна «клановость»)

## **«Метод средних баллов при проведении экспертных оценок»**

Метод средних баллов является наиболее простым при проведении экспертных оценок, который позволяет сравнивать объекты, назначая каждому из них определенное количество баллов по заданной шкале измерений. Подходит для сравнения простых объектов, для которых можно сказать, что один вариант лучше другого и указать количественно на сколько.

Обработка результатов проводится методами математической статистики в следующем алгоритме:

1. Определяется среднее количество баллов – матожидание выборки:
2. Определятся степень рассогласованности мнений экспертов – дисперсия по выборке:
3. Определяется доверительный интервал , где .
4. По полученным данным строится биржевая диаграмма и определяются выводы.

Данный метод обладает целым перечнем недостатков, основные из которых:

1. Невозможность выявить однозначно оптимальный вариант (верхние и нижние границы разных вариантов могут сливаться и наслаиваться).
2. Результат очень зависит от выбросов и степени рассогласованности экспертов.

## **«Методы медианных и средневзвешенных рангов при проведении экспертных оценок»**

**Метод медианных рангов** применяется при невозможности назначить баллы, но можно расставить по рангу – отранжировать объекты. Подходит для сравнения сложных объектов, для которых можно сказать, что один вариант лучше другого, но нельзя указать количественно на сколько.

Для этого метода необходимо отсортировать ранги по возрастанию (по строке), определить середину таблицы и вычислить медиану как среднее арифметическое двух соседних рангов, делящая ранжированные данные на две равные части, половина которых меньше медианы, а другая – больше:

Где – количество экспертов, – проставленный ранг. Полученные значения медианы независимы от выбросов. **Оптимальный вариант тот, у которого медиана минимальна.**

**Метод средневзвешенных рангов** позволяет учесть компетентность экспертов (степень отклонения его ответа от среднего значения). Для этого необходимо:

1. Определить математическое ожидание рангов вариантов
2. Определить степень отклонения ответа эксперта от среднего значения – дисперсию мнения каждого эксперта:
3. Чем больше дисперсия, тем сильнее мнение эксперта отличается от мнения большинства. Следовательно, его компетентность в данном вопросе можно считать ниже, чем у коллег. Компетентность эксперта будет обратно пропорциональна его дисперсии:
4. Нормируем значения компетентности:
5. Найдем средневзвешенное мнение экспертов относительно кандидатов с учетом коэффициента их компетентности (важности их мнения):
6. Определим оптимальный вариант как вариант с наименьшим значением .

## **«Методы бинарных отношений при проведении экспертных оценок»**

Метод попарных сравнений -один из наиболее сложных методов. Применяется в случаях, когда объекты настолько комплексные и многомерные, что нет возможности отранжировать их по одному признаку, но есть возможность попарно сравнить их между собой.

Для этого необходимо:

1. Каждый эксперт должен сравнить каждую пару вариантов друг с другом. Результат записывается в виде матриц бинарных отношений:

1. Решить задачу нелинейного программирования, определив значения медианы Кемени:
   1. Определить медиану Кемени , суммарное расстояние от которой до всех остальных бинарных матриц является минимальным
   2. Определить целевую функцию как сумму расстояний Кемени (характеризующих отличия между матрицами):
   3. Используя «Поиск решения» найти значения медианы Кемени.
2. Полученная медиана Кемени является усреднённым мнением экспертов, выраженным в бинарных отношениях. Для определения лучшего варианта необходимо просуммировать столбцы медианы Кемени и выбрать вариант с наименьшей суммой.

## **«Структура системы массового обслуживания, потоки заявок и вероятности состояний системы»**

Структуру любой системы массового обслуживания (СМО) можно представить в виде следующих компонентов:

1. Поступающие новые заявки (входящие покупатели)
2. Ожидающие в очереди заявки (очередь на кассу)
3. Обслуживающие устройства (кассы)
4. Обслуженные исходящие заявки (выходящие с покупками)

Рассмотрим два основных типа потоков: входящий и исходящий.

Скорость поступления заявок в систему характеризуется **интенсивностью входящего потока** – сколько заявок приходит в систему за определенный интервал времени

Где – среднее количество авто в период, – количество минут в периоде.

Скорость обработки заявок характеризуется **интенсивностью выходного потока** –сколько заявок обрабатывает один прибор в единицу времени

Где 𝑁–количество заявок, обработанное за период времени 𝑡 всеми 𝑛 приборами, -среднее время на обработку одной заявки одним прибором.

**Состояния системы** могут быть представлены как вершины графа Маркова. Всего у систем массового обслуживания бывает три основных состояния: бездействие, обслуживание, очередь. Каждому состоянию соответствуют определённое значение вероятности - % времени нахождения системы в этих состояниях. Определим эти вероятности по формулам:

Где: – количество заявок в системе,

–количество аппаратов обслуживания,

– максимальная длина очереди.

## **«Показатель нагруженности и сбалансированность системы; показатели для клиентов и владельцев систем массового обслуживания.»**

Показатель нагруженности 𝝆 характеризует насколько система справляется с потоком клиентов и вычисляется как отношение интенсивности входящего потока к выходящему:

Рассмотрим **типы сбалансированности системы**:

1. – система недогружена – выгодно для клиента (нет очередей), невыгодна для владельца (лишние кассы, большой простой по времени).
2. – система сбалансирована для клиента (приемлемые очереди, допустимый простой касс).
3. - система сбалансирована для владельца (большие очереди клиентов, кассы заняты почти полностью).
4. - система перегружена – выгодно для владельца (заявок больше, чем можно обработать), невыгодно для клиента (бесконечно растущая очередь).

Рассмотрим основные **показатели для клиента:**

1. Вероятность отказа в обслуживании – новая заявка не помещается в систему (заняты все приборы и заполнена вся очередь –последняя «правая» вероятность):
2. Вероятность встать в очередь –новая заявка встречает в системе очередь (заняты все приборы и есть место в очереди -сумма всех вероятностей очередей):
3. Средняя длина очереди –среднее количество заявок, ожидающих в очереди:
4. Среднее время ожидания в очереди –формула Литтла:

Рассмотрим основные **показатели для владельца:**

1. Абсолютная пропускная способность –скорость обслуживания заявок (сколько заявок успевает обрабатываться системой в единицу времени):
2. Относительная пропускная способность –процент обслуженных заявок(какой процент заявок успевает обрабатываться системой, а какой получает отказ):
3. Среднее количество занятых приборов –сколько приборов занято обслуживанием:
4. Коэффициент простоя –процент времени простоя обслуживающих приборов