**Математические методы принятия решений**

**Список примерных теоретических вопросов**

1. Задача линейного программирования; симплекс-метод.

Линейное программирование – метод произведения линейных вычислений, то есть с помощью сведения вычислений к целевой функции, которая является линейной.

Если целевая функция и ограничения, представленные в виде уравнений или неравенств, состоят из переменных в первой степени, то есть являются линейными, то мы имеем дело с задачей линейного программирования (программирования в значении – вычисления).

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

1. Записать исходные данные.

2. Определить переменные.

3. Сформулировать целевую функцию.

4. Записать систему ограничений.

ЭТАПЫ ПОСТРОЕНИЯ МАТ. МОДЕЛИ

Исходные данные: a, b, c, d

Искомые переменные: x, y, z

Составление целевой функции:

F = ax +by + cz + d

Введение ограничений:

ax + by ≤ N

by – cz ≤ M

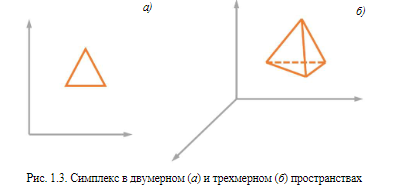
x, y, z › 0

Поиск максимума / минимума:

F = ax +by + cz + d → max

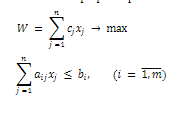
Для поиска максимума целевой функции существует множество различных приемов и методов, но, наверное, самый универсальный – симплекс-метод.

Симплекс – представляет собой геометрическую фигуру, количество углов которой на единицу больше размерности пространства. В двухмерном пространстве, симплексов будет являться треугольник.



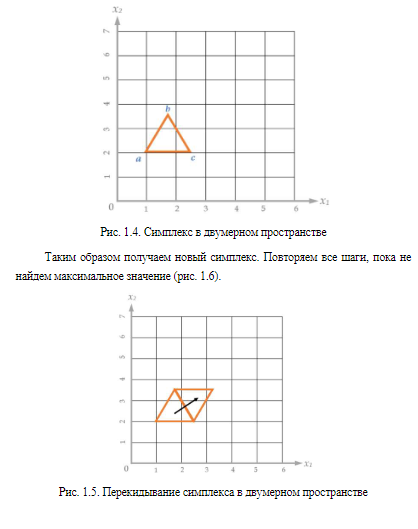
Симплекс-метод представляет собой алгоритм решения задачи линейного программирования посредством перебора вершин выпуклого многогранника в многомерном пространстве.

Поместим симплекс в наше рабочее пространство, поскольку в целевой функции у нас только две координаты - , то пространство будет плоское – двумерное, а симплекс будет треугольником. Длина стороны симплекса избирается произвольно, как и его первоначальное положение. Помещаем наш треугольник *abc* в какую-то начальную точку и дальше в каждой вершине считаем значение целевой функции по формуле:

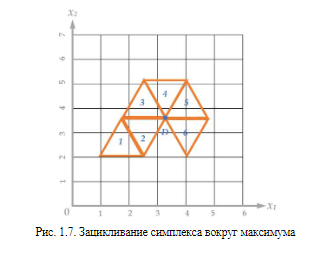


Так как каждая вершина треугольника – это точка на плоскости, значит она имеет свои координаты ().

Сравниваем значения целевой функции в каждой вершине симплекса. В случае поиска максимального значения целевой функции отбрасывается та вершина, в которой наименьшее, и наоборот. Поскольку наша целевая функция стремится к максимуму, ту вершину, где значение целевой функции самое маленькое, мы отбрасываем и перекидываем треугольник через противоположное ребро.



Показателем того, что мы нашли максимум является два варианта событий: систематический поворот относительно одной из сторон (позиции 2 и 3, рис. 1.7), вращение вокруг какой-либо точки, например, точки D на рисунке 1.7.



Координаты этой точки и будут решением нашей целевой функции.

1. Производственная задача.

Линейное программирование – метод произведения линейных вычислений, то есть с помощью сведения вычислений к целевой функции, которая является линейной.

Если целевая функция и ограничения, представленные в виде уравнений или неравенств, состоят из переменных в первой степени, то есть являются линейными, то мы имеем дело с задачей линейного программирования (программирования в значении – вычисления).

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

1. Записать исходные данные.

2. Определить переменные.

3. Сформулировать целевую функцию.

4. Записать систему ограничений.

Суть производственной задачи состоит в наличии некого предприятия, выпускающего несколько видов продукции. Обозначены ресурсы и их количество, необходимое для производства. Количество ресурсов ограничено. Известно также за какую цену можно продать ту или иную продукцию. Необходимо понять, какие производственные изделия следует произвести и в каком количестве, чтобы прибыль предприятия была максимальной.

Разберем задачу на примере. Рассмотрим производство мороженного двух типов: стаканчик и эскимо. Для двух типов продукции у нас используются следующие материалы: сухое молоко, сгущенное молоко, сахар, глазурь, стаканчик вафельный, палочка деревянная. В итоге имеем номенклатуру продукции и номенклатуру материалов. Теперь можем составить матрицу, в которой будут описаны расходы материалов на единицу каждого вида продукции.

***Исходная таблица материалов.***

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | Сухое молоко,кг | Сгущенное молоко,кг. | Сахар,кг. | Глазурь,кг. | Стаканчик вафельный,шт. | Палочка деревянная,шт. |
| "Стаканчик" |  |  |  |  |  |  |
| "Эскимо" |  |  |  |  |  |  |

Далее в производственной задаче задаются ***ограничения*** по складу, они характеризуют максимальное количество каждого материала, которое может хранится на складе предприятия.

Ограничения по складу.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | Сухое молоко,кг | Сгущенное молоко,кг. | Сахар,кг. | Глазурь,кг. | Стаканчик вафельный,шт. | Палочка деревянная,шт. |
| Ограничения по складу |  |  |  |  |  |  |

Далее, определим ***стоимость продажи*** готовой продукции.

Стоимость продажи готовой продукции.

|  |  |
| --- | --- |
|  | Стоимость продажи |
| "Стаканчик" |  |
| "Эскимо" |  |

Объединим все исходные данные в одну таблицу.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | Сухое молоко,кг | Сгущенное молоко,кг. | Сахар,кг. | Глазурь,кг. | Стаканчик вафельный,шт. | Палочка деревянная,шт. | Стоимость продажи |
| "Стаканчик" |  |  |  |  |  |  |  |
| "Эскимо" |  |  |  |  |  |  |  |
| Ограничения по складу |  |  |  |  |  |  |  |

Следующим шагом определим ***искомые переменные.***

В данном случае будем искать оптимальное количество каждого вида продукции, которое необходимо изготовить. Обозначим переменные – количество стаканчиков, – количество мороженных эскимо.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | Сухое молоко,кг | Сгущенное молоко,кг. | Сахар,кг. | Глазурь,кг. | Стаканчик вафельный,шт. | Палочка деревянная,шт. | Стоимость продажи | **Кол-во прод., шт.** |
| "Стаканчик" |  |  |  |  |  |  |  |  |
| "Эскимо" |  |  |  |  |  |  |  |  |
| Огранич по складу |  |  |  |  |  |  |  |  |

Находим оптимальное количество производимой продукции каждого вида. Это можно сделать используя разные критерии: стоимость, скорость изготовления, минимальное количество отходов. Исходя из исходных данных выберем ***критерий оптимальности*** – максимальная выручка от проданной продукции.

Вид целевой функции отвечает на вопрос: «Какое количество каждого вида продукции необходимо произвести, чтобы производство было оптимально по выручке?». В этом случае, целевая функция будет выглядеть следующим образом:

То есть, по формуле выручки, мы умножаем количество произведенного мороженного стаканчик - на его стоимость - . Аналогично перемножаем стоимость и количество произведенного мороженного эскимо - . Эти величины складываем и получаем совокупную выручку производства двух видов продукции, которая будет стремиться к максимуму.

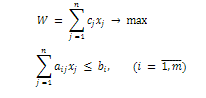
Теперь перейдем к описанию ***ограничений***, которые не будут давать целевой функции перебирать значения по бесконечности. В качестве ограничений имеет максимальное количество материалов которое может храниться на складе, и быль использовано в производстве продукции:

Первое ограничение говорит нам о том, что количество материала сухое молоко на складе израсходованное на производство двух видов продукции, не должно быть больше чем количество, данное материала на складе. Аналогично для всех других ограничений.

Также ограничением в нашем случае будет не отрицательность искомых переменных, так как производство не может произвести например – 3 мороженных. Таким образом, вторая группа ограничений будет выглядеть так:

На этом задачу можно считать сформулированной. Остается найти способ ее решения, то есть метод отыскания таких значений - , которые доставляют максимум ***целевой функции*** C и при этом не нарушают ограничений.

В общем виде задачу можно записать так:



1. Классическая транспортная задача.

Классическая транспортная задача является подвидом развития и модернизации производственной задачи – основополагающего элемента линейного программирования.

***Классическая транспортная задача -*** была сформулирована впервые под решение проблемы перевозки грузов.

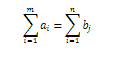
В классической постановке она звучит так: есть город, в нем располагаются несколько складов однотипной продукции. Однотипность – ключевой момент. И есть несколько объектов на территории города, которые являются потребителями этой продукции. У каждого объекта имеется определенный спрос измеряемый в количестве продукции. У каждого склада есть определенный запас продукции. Есть стоимость перевозки – везти со склада в ближайший магазин намного дешевле. Задача сводится к тому, что необходимо построить план перевозок продукции от поставщиков (складов) к потребителям, обеспечивая минимальные транспортные расходы.

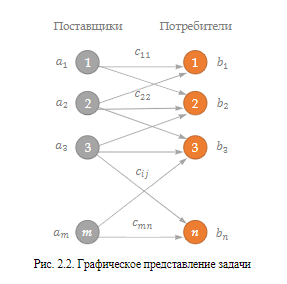
***Математическая постановка задачи.***

Эту задачу можно описать следующим образом – построить граф (рис. 2.2). Слева поставщики, справа потребители. Пусть имеется m поставщиков одной продукции, у которых сосредоточено единиц соответственно. Также имеется n потребителей продукции, потребности которых составляют единиц продукции соответственно.

Далее, от каждого поставщика к каждому потребителю, к которому существует дорога, строится ребро и обозначается его величина - стоимость перевозки единицы продукции от i-ого поставщика j-ому потребителю. Соответственно, необходимо найти маршруты и объемы перевозок продукции ***минимальной стоимости.***

В математической постановке задачи имеется одно условие, задача является замкнутой. Это значит, что спрос потребителей полностью удовлетворяется предложением складов, то есть все товары со складов полностью развозятся к потребителям. Записывается данное таким образом:

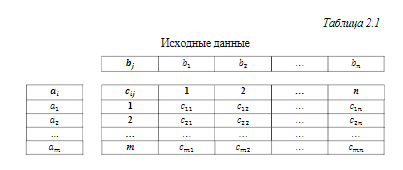




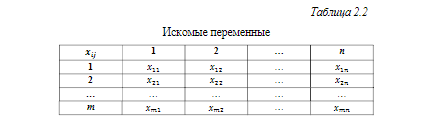
Сначала рассмотрим задачу в закрытом типе, далее разберем случаи дефицита и профицита.

После нарисовки графа, следует преобразовать его в набор матриц.

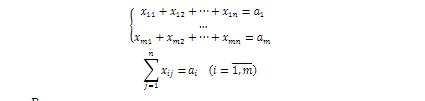
Граф с рисунка 2.2 преобразован в три матрицы: матрица – столбец, в которой записано количество товаров у каждого поставщика ; матрица – строка, в которой записаны потребности каждого магазина ; матрица, в ячейках которой записана стоимости перевозки от поставщика к потребителю . Нулевая стоимость говорит о том что такая перевозка невозможна.



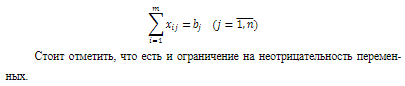
После обозначения исходных данных, приступим к обозначению переменных, то есть тех значений, которые не известны, но которые необходимо найти. Искомые переменные также образуют матрицу - это количество единиц продукции, перевезенной от i-ого поставщика j-ому потребителю. ( Табл. 2.2)



Следующим пунктом, обозначим ограничения. По вышеупомянутому условия, необходимо прописать данное ограничение: на от i-ом складе, запас продукции равен, количеству необходимой продукции для j-ого магазина. То есть все запасы должны быть вывезены.



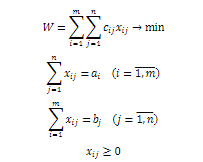
Второе ограничение аналогично первому – количество товаров, доставляемое j-ому потребителю с разных складов, должно быть равно общему потребности j-ого потребителя. Соответственно, потребности всех магазинов должны быть удовлетворены.



Далее, составим ***целевую функцию****.* Целевая функция должна отражать следующее: стоимость всех выбранных маршрутов в сумме должна быть минимальной.



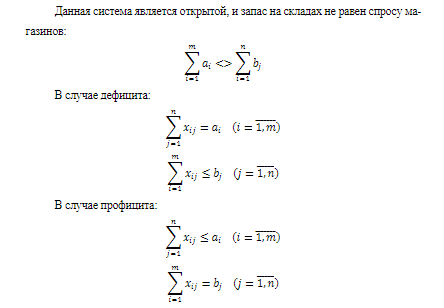
Данная задача может быть описана в виде классической записи задачи линейного программирования. То есть сначала целевая функция, а потом – ограничения:



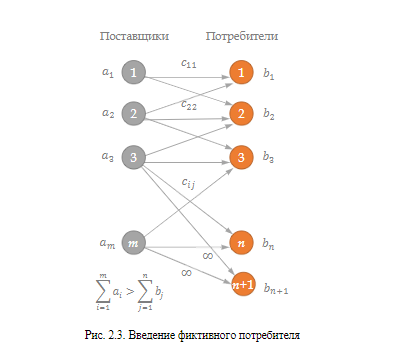
Решение задачи осуществим с помощью симплекс-метода.

**Открытая транспортная задача.**

Рассмотрим частный случай классической транспортной задачи. Если запасы на складах превышают спрос магазинов или наоборот если продуктов на складе не хватает.



Данная задача решает путем сведения ее от открытой к закрытой транспортной задачи, с помощью введения дополнительного – фиктивного магазина или склада и списания в него накопившихся издержек как показано на рисунке 2.3. (показано введение фиктивного магазина)



Спрос в этом магазине устанавливается как разница между положительным и отрицательным потенциалом товаров:



А стоимость перевозки в этот пункт полагаем равным бесконечности, чтобы в изначальных пунктах не возник недостаток.



Если существует обратный баланс и товаров больше чем магазинов, вводится фиктивный склад. Соответственно, когда спрос превышает предложение:



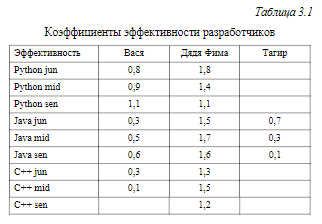
Таким образом, открытая задача сводится к закрытой и решать ее уже можно аналогично способу описанному выше.

1. Задача о назначениях.

Задача о назначениях является одной из модификаций производственной задачи.

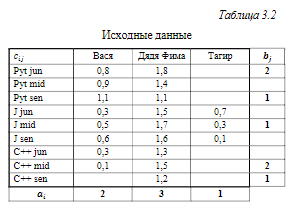
Разберем задачу на конкретном примере решения.

Пусть имеется фирма с командой разработчиков из 3-ёх человек (Вася, Фима и Таха), обладающих разными компетенциями в 9 различных сферах: Python junior, Python middle, Python senior, Java junior, Java middle, Java senior, C++ junior, C++ middle, C++ senior. Под компетенцией разработчика подразумевается его эффективность решения задачи, которая определяется указанным в таблице 3.1 коэффициентом.



При этом Вася способен решить за спринт 2 задачи, Фима – 3, а Таха – 1. Задача заключается в том, что очередной спринт состоит из 7 задач различного уровня сложности и руководителю проекта требуется выбрать 6 задач на спринт и назначить их разработчикам так, чтобы эффективность их выполнения была максимальна.

Запишем формальную постановку задачи. Исходные данные представлены в таблице 3.2 в виде совокупности матриц.

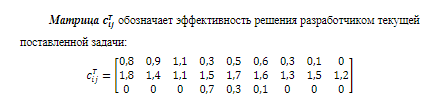


Матрица-строка обозначает количество задач на каждого разработчика:

(m = 3)

Матрица-столбец обозначает количество задач различных типов (степени компетентности) в текущем стеке:

(n = 9)



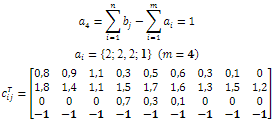
Основное отличие задачи о назначения от остальных заключается в том, что искомые переменные – булевого типа, то есть могут принимать только значения 1 или 0. Получается бинарная матрица переменных , где единицы стоят в тех позициях, которые назначены данному программисту:



Рассмотрим матерматическую запись этой задачи. Сначала проверим е ена сбалансированность: 7 задач в стеке и 6 задач для распределения между программистами.

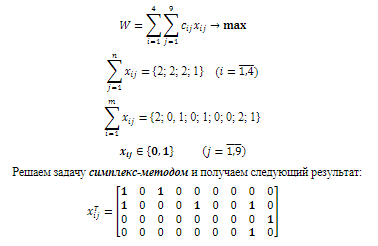


Значит существует дисбаланс и необходимо свести задачу из открытой к закрытой. Введем фиктивного программиста Лену с 1 задачей и эффективностью – 1.

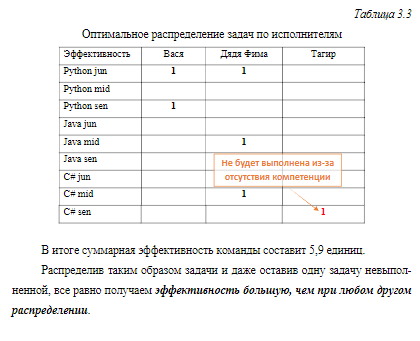


Модифицируем матрицы исходных данных.

Запишем целевую функцию и ограничения в виде строгих равенств, при этом целевая функция эффективности стремится к максимуму, а ограничения сформулированы на продуктивность каждого разработчика и на количество в стеке задач определенного уровня сложности. Стоит отметить, что теперь х принимает только два значения 0 и 1.



Для интерпретации запишем решение в таблице.



1. Задача о кратчайшем пути на графе.

В данном типе задач необходимо построить кратчайший путь из одной точки транспортной сети в другую. Критериями могут служить время перемещения, длина пройденного пути, стоимость проезда, вероятность затора.

Рассмотрим задачу поиска кратчайшего пути в центре Москвы (с района Якиманка на Китай город), которая представлена на рис 6.6. В качестве вершин возьмем перекрестки, в качестве дуг – дороги между перекрестками. Все ребра будут обозначать приблизительное время проезда между перекрёстками. Нам необходимо построить самый быстрый маршрут из пункта отправления А в пункт Б. На данном графе можно заметить и ребра, и дуги. Введение ориентированности связано с односторонним движением. Также особенной является вершина 7, в которую можно войти, но выйти нельзя.

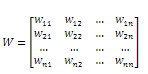


**Решение как задачи линейного программирования.**

Преобразуем граф в матрицу, где вершины – номера столбцов/строк, вес ребер – элементы внутри самой матрицы.

Пусть дан граф *G = (V,A),* где *V* – множество вершин графа, *A* – множество ребер графа.

Вес ребер , задается матрицей смежности:



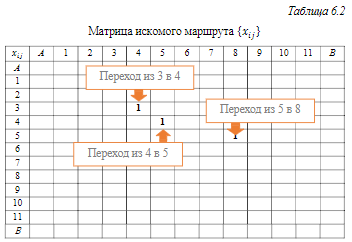
где *n* – количество вершин графа.

Матрица смежности может строиться на основании веса ребер, но может и иметь бинарный вид, то есть показывать наличие или отсутствие дороги (ребра на графе) (табл. 6.1).



Наша задача найти по каким ребрам графа , проходит кратчайший маршрут из вершины А в вершину В.

Также создадим матрицу искомого маршрута , где 1 будет означать наш выбор данного ребра. Иными словами, это матрица переменных. Например, он может выглядеть так:



Таким образом, мы сводим нашу задачу к задаче линейного программирования. Для решения зададим целевую функцию – минимальную суммарную длину пути:

То есть суммируем произведение единиц из матрицы искомого маршрута и веса ребер из матрицы смежности. Наша целевая функция должна стремиться к минимуму.

Теперь необходимо задать ограничения:

* Маршрут должен начинаться в пункте А и заканчиваться в пункте В:



Это значит, что в строке А и в столбце В обязательно должна быть единица.

* Переход должен осуществляться только по существующим ребрам:

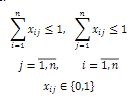


Это запись значит, что единицу мы имеем право ставить только в тех ячейках, в которых в матрице смежности стоит цифра отличная от нуля. Если стоит ноль, то ребро отсутствует.

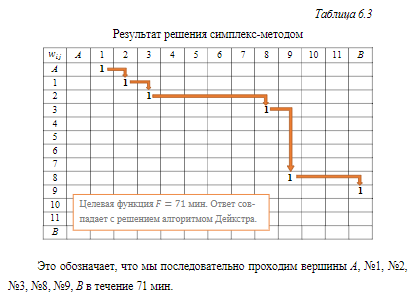
* Ограничение на связность маршрута: вход в вершину всегда должен соответствовать выходу из нее (сумма по строке j равна сумме по столбцу j), кроме начала А и конца В маршрута:



* Вход и выход в каждую вершину производится не более одного раза, чтобы не возникало петель:



Решается задача симплекс-методом, как и любая другая задача линейного программирования. В результате получаем матрицу значений:

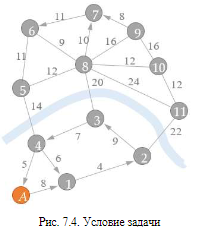


1. Задача коммивояжера.

**Постановка задачи.**

Представим маршруты между адресатами в виде графа (ориентированные дуги обозначают последовательность посещений). Возьмем, для примера, граф из предыдущей главы с небольшими изменениями.

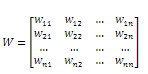
Вершина А является началом и концом маршрута (офис доставки). Вес ребра обозначает время переезда между адресатами. Необходимо построить замкнутый маршрут, проходящий через все вершины графа не более 1-ого раза и имеющий минимальную длину.



***Решение задачи как дискретного комбинаторного программирования.***

Пусть дан граф *G = (V,A),* где *V* – множество вершин графа, *A* – множество ребер графа.

Вес ребер , задается матрицей смежности:



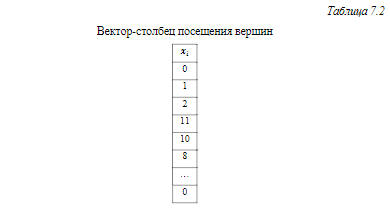
где *n* – количество вершин графа.

Необходимо найти замкнутый цикл минимальной длины.

Составим матрицу смежности графа :



В данном случае мы будем использовать другую вариацию матрицы маршрута. Так как нам необходимо обойти все вершины, то можно построить для переменных вектор-столбец, в который мы будем записывать номер вершины в определенной последовательности. В конце стоит та же вершина, что и в начале, так как нам следует замкнуть маршрут.



Для решения сформулируем целевую функцию – минимальную суммарную длину всех дуг пути:



Наша целевая функция должна стремится к минимуму:

Теперь запишем ограничения:

* Маршрут должен начинаться и заканчиваться в пункте А:



* Переход должен осуществляться только по существующим ребрам:



* Вершины не должны повторяться:

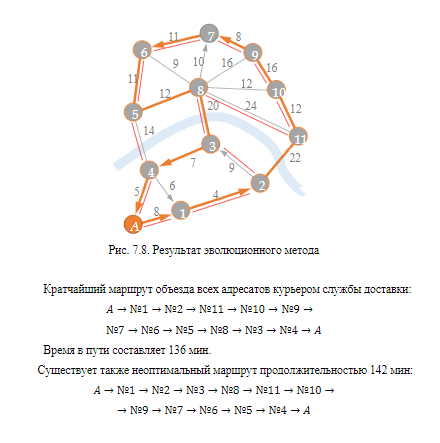


* Искомые переменные целочисленные:



Решить данную задачу симплекс-методом не представляется возможным, поэтому используем ***эволюционный метод.*** Результат решения задачи эволюционным методом в MS Excel (комбинаторный перебор вариантов).

Результат:



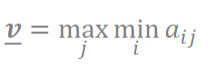
1. Решение антагонистической игры.

# **14.Решение антагонистической игры в чистых стратегиях.**

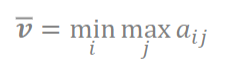
Антагонистические игры - игры с нулевой суммой.

Игроки борются друг с другом, выигрыш одного равен проигрышу другого, выигрыши игроков противоположны. Заполняем одну платежную матрицу.

Стратегия «Максимин» обеспечивает максимальный из гарантированных выигрышей игрока А, какие бы стратегии не применял в ответ игрок В – находим минимальное по строке и максимальное из полученных значений. Полученное значение называется нижней ценой игры и означает, что если игрок А будет придерживаться данной стратегии, то его гарантированный выигрыш не опустится ниже этого значения.



Стратегия «Минимакс» (максимальное по столбцу и минимальное из полученных) обеспечивает минимальный проигрыш игрока В, какие бы стратегии не применял игрок A (обратная стратегии «Максимин»). Полученное значение называется верхней ценой игры и означает, что выигрыш игрока В гарантированно не опустится ниже (макс.выигрыш – верхняя цена игры).



Когда верхняя и нижняя цены игры совпадают это называется чистая цена игры и означает, что найдена оптимальная пара стратегий для обоих игроков при которой игроки получают максимально возможное гарантированное число клиентов. Если кто-то из игроков отойдет от своей оптимальной стратегии, то он проиграет.

Оптимальное решение игры называется «седловой точкой»: точкой, в которой сходятся минимум функции AXB и максимум функции DXC( нижняя и верхняя цены игры равны). В таком случае игра имеет единственное решение, оптимальное для обоих игроков.

# **15.Решение антагонистической игры в смешанных стратегиях.**

Антагонистические игры - игры с нулевой суммой. Игроки борются друг с другом, выигрыш одного равен проигрышу другого, выигрыши игроков противоположны. Составляем платежную матрицу.

Вариант чередования ходов в игре называется смешанной стратегией, в отличие от рассмотренных ранее вариантов игры в чистых стратегиях.

Цена игры в смешанных стратегиях выше, чем в чистых. Находим смесь стратегий или делим бюджет так, чтобы получить максимальное количество клиентов на рынке.

1)Решаем прямую задачу для игрока А. Поиск оптимальных смешанных стратегий с помощью задачи линейного программирования.

Исходные данные – транспонированная платежная матрица. Вводим переменные по количеству вероятностей x1,x2,x3 (замененные вероятности p1, p2, p3) – доли бюджета на развитие.

Целевая функция –сумма переменных сведенная к минимуму: 𝐿=𝑥1+𝑥2+𝑥3→min.

Ограничения – сумма произведений элементов матрицы на значения переменных (больше или равно 1).

После нахождения x1,x2,x3 проводим обратную замену переменных:

Цена игры - v = 1/ Lmin (целевую функцию) –, p1 = x1\* v, p2 = x2\*v, p3 = x3\*v и тд.

С помощью поиска решений находим в каком процентном соотношении поделить бюджет (р1,р2,р3) и цену игры (количество клиентов, которое получим).

2) Далее записываем обратную задачу для игрока В.

Исходные данные – платежная матрица (нетранспонированная).

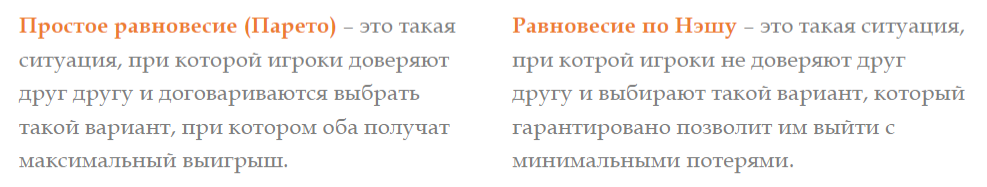
Далее, что касаемо ограничений и обратной замены переменных происходит все то же самое, единственное, что целевая функция стремится к максимуму.

На практике и при решении задач получалось, что смешанные стратегии дают более гибкий результат, так как мы не вкладываемся во что-то одно, а делим бюджет на несколько направлений и соответственно это дает больший выигрыш.

1. Решение биматричной игры.

# **16.Решение биматричной игры в чистых стратегиях.**

Игра с двумя матрицами. Игроки не борются друг с другом, а стремятся каждый к своему выигрышу. Матрицы игроков не противоположны. В данной ситуации у нас две платежные матрицы.



Равновесие по Нэшу – пара стратегий, которая принесет весомую прибыль каждому игроку и при этом каждому из которых будет не выгодно отклоняться от стратегии.

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.

Ситуации равновесия в чистых стратегиях отсутствуют. В любом случае кто-то проигрывает больше другого, поэтому необходимо обратиться к смешанным стратегиям.

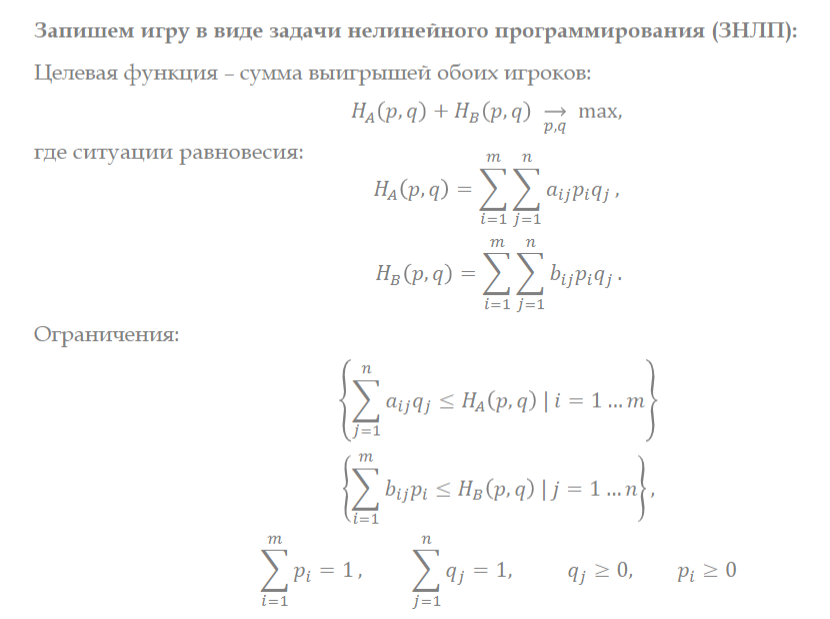
# **17.Решение биматричной игры в смешанных стратегиях.**

Игра с двумя матрицами. Игроки не борются друг с другом, а стремятся каждый к своему выигрышу. Матрицы игроков не противоположны.

Записываем нашу игру в виде задачи нелинейного программирования.

P1,p2,p3 – смесь стратегий игрока А.

Q1,q2,q3 -смесь стратегий игрока В.

добавляем два ограничения – сумма смесей стратегий игроков А и В равняется 100%.

Находим оптимальную смесь стратегий с помощью поиска решений методом ОПГ – обобщенного приведенного градиента.

1. Решения матричной игры с природой в условиях риска.

# **18.Решения матричной игры с природой в условиях риска.**

Игры с природой – это такой тип игр, в которых в качестве второго игрока выступает «природа». Природа не стремится сделать вам хуже или лучше, она живет по своим законам. Не все из этих законов известны, существует большая доля неопределенности и вариативности в поведении природы.

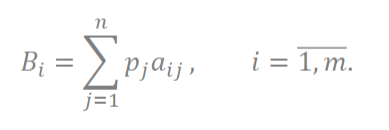
Примеры «природы»: • поведение диких животных; • реальные погодные условия; • изменение потребительского спроса; • колебание курса ценных бумаг и др.

Решение игр с природой в условиях риска означает, что нам известны вероятности наступления состояний природы.

Критерии выбора стратегии : Байеса, Лапласа, Гермейера.

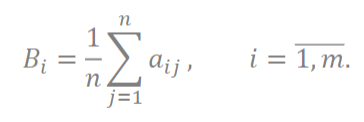
1. **Критерий Байеса:**

Вычисляем средневзвешенное значение выигрыша каждой стратегии игрока А– для этого используем формулу суммапроизведений и максимальное из найденных значений будет являться оптимальным.



1. **Критерий Лапласа:**

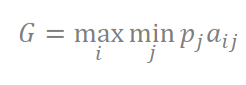
Если значения вероятностей неизвестны, то они принимаются как равновероятные (если 4 состояния, то ¼ = 25%) Вычисляем среднеарифметическое значение выигрыша игрока А (суммпроизв). Оптимальной является стратегия, в которой среднеарифметическое значение максимальное.



1. **Критерий Гермейера:**

Игрок может получить свой выигрыш с определенной вероятностью – на основании этого строим матрицу Гермейера (сумму выигрыша умножаем на вероятность).

Ценой игры является максимин матрицы Гермейера – находим минимум по каждой строке и из этих минимумов находим максимум.



Критерий Гермейера – гарантированный выигрыш в самом неблагоприятном состоянии природы.

1. Решения матричной игры с природой в условиях неопределенности.

# **19.Решения матричной игры с природой в условиях неопределенности.**

Игры с природой – это такой тип игр, в которых в качестве второго игрока выступает «природа». Природа не стремится сделать вам хуже или лучше, она живет по своим законам. Не все из этих законов известны, существует большая доля неопределенности и вариативности в поведении природы.

Примеры «природы»: • поведение диких животных; • реальные погодные условия; • изменение потребительского спроса; • колебание курса ценных бумаг и др.

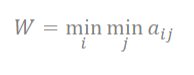
Решение игр с природой в условиях неопределенности означает, что нам неизвестны вероятности наступления состояний природы.

В данном случае для выбора стратегии мы используем критерии: оптимизма(максимакс), пессимизма(минимин), Вальда(максимин), Гурвица (линейная свёртка), Сэвиджа (на матрице рисков).

Рекомендуется решать задачу по каждому критерию и затем выбирать стратегию, на которую указывает наибольшее количество критериев.

**Критерий пессимизма:**

Критерий пессимизма – используем принцип «минимин». Пессимистичный сценарий – самые большие издержки.



**Критерий оптимизма:**

Критерий оптимизма - используем принцип «максимакс». Оптимистичный сценарий – максимизация прибыли.



**Критерий Вальда:**

Критерий Вальда - принцип «максимин» (минимальное по строке и потом максимум из полученных значений). Надежный сценарий – гарантированная прибыль.



**Критерий Гурвица (линейная свёртка):**

В данном случае мы составляем линейную свертку на основе коэффициента риска с использованием критериев оптимизма и пессимизма. Отношение к риску принимает значения от 0 до 1.

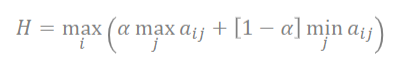
Существует три ситуации отношения к риску:

𝛼 ∈ [0; 0,5) – игрок склонен избегать риска;

𝛼 = 0,5 – игрок безразличен к риску;

𝛼 ∈ [0,5; 1) – игрок стремиться к риску.

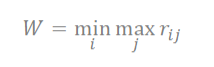
Используем формулу : коэффициент отношения к риску \* максимальное значение (по строке) + (1 - коэффициент отношения к риску) \* минимальное значение (по строке).



В итоге мы получаем оптимальные стратегии в зависимости от того, насколько хочет рискнуть игрок.

**Критерий Сэвиджа:**

Критерий Сэвиджа - критерий «минимакс» на матрице рисков. Компромиссный сценарий - минимизация упущенной выгоды (отставания от «идеального» варианта). Мы находим максимум прибыли в каждом состоянии природы (по столбцу) и принимаем его за некий идеальный вариант.



Далее строим матрицу рисков. Матрица рисков строится как разница максимума по столбцу и значения текущей ячейки.



1. Решение позиционной игры с противником на дереве решений.

# **20.Решение позиционной игры с противником на дереве решений.**

Игроки ходят последовательно, стратегии выбираются динамически на каждом ходу, могут быть как антагонистическими, так и с индивидуальным набором очков. Ходы представлены в виде дерева.

Изначально необходимо сформулировать задачу, на основе которой будет реализоваться игра. Далее мы строим дерево игры. Дерево игры – это совокупность всевозможных ходов игроков из всех возможных позиций и выигрыши или проигрыши, получаемые в результате этих ходов. Конечным элементам дерева (листьям) соответствует цена игры данной совокупности.

Варианты стратегий игры:

1. Выбирать ход худший для конкурента.
2. Выбирать ход, чтобы меньше отстать от конкурента - для магазина В выбираем наименьшую разницу отставания, а для магазина А берем наибольшую сумму отрыва).
3. Выбирать ход, лучший для себя - на каждом ходу выбираем шаг, который принесет максимальную прибыль.
4. Выбирать ход лучший для себя и оппонента.

Алгоритм Куна:

1. Выбираем лучшие альтернативы на последних элементах (листьях)
2. Далее переходим на предпоследний ход и выбираем лучшую альтернативу для второго игрока
3. По итогу остается одна ветка наиболее выгодная для обоих игроков

Сводим игру к матричной форме. Платежная матрица строится как комбинация всех возможных сочетаний стратегий обоих игроков и финального выигрыша от применения этих стратегий. В виде двух антагонистических игр для игрока А и В находим нижнюю и верхнюю цены игры.

Либо рассматриваем биматричную игру и находим равновесие по Нэшу.

1. Решение позиционной игры с природой на дереве решений.

# **21.Решение позиционной игры с природой на дереве решений.**

Изначально необходимо сформулировать задачу, на основе которой будет реализоваться игра. Для этого используем стратегии игрока и состояния, которые может принимать природа., выступающая в качестве второго игрока.

Строим дерево игры, указывая выигрыш или проигрыш на каждом ход, а также вероятности наступления состояния природы.

Для того, чтобы найти наиболее оптимальную стратегию мы сворачиваем дерево решений игры.

Когда свой ход в игре делает природа, мы находим средневзвешенное значение (сумма произведений выигрышей в каждом состоянии природы на вероятность наступления данных состояний природы), в виду того, что действия природы выполняются на основе вероятности.

Когда осуществляет свой ход человек, он руководствуется здравым умом, поэтому выбираем ход, который принесет наибольшую прибыль.

Соответственно, в ходе реализаций данных действий на ходу природы мы сворачиваем ветку, а затем на ходу человека отрезаем менее выгодные варинты.

При реализации данной игры у нас остается не одна ветка, как в позиционной игре с противником, а разветвленная ветка ввиду того, что у природы несколько состояний.

Сведение к матричной игре. Позиционную игру с природой можно представить в виде платежной матрицы, в строках которой будут все комбинации действий игрока, а в столбцах – комбинации состояний природы. Применяем критерии матричных игр и находим оптимальные стратегии: критерий Байеса, Гермейера и Гурвица (линейная свертка – отношение к риску).

1. Понятие линейной регрессии; метод наименьших квадратов.

*Регрессия* в теории вероятностей и математической статистике – односторонняя стохастическая зависимость, устанавливающая соответствие между случайными переменными. Это математическое выражение, отражающее связь между переменными y и x при наличии между ними статистической зависимости.

Линейная регрессия.

Линейная регрессия – функция некоторой прямой, которая отражает характер поведения всей статистики. Она имеет следующий вид:

где – истинная зависимость (тренд), - случайные колебания (шум). Из всех точек, расположенных на графике, для точного описания прямой проходят только не выбивающиеся из тренда точки. Изобразим полученную прямую функции на рис. 17.3.

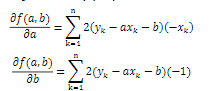
Благодаря нахождению коэффициентов a и b линейной регрессии y = ax + b становится возможным найти наиболее адекватно отражающие зависимость переменных и . Коэффициенты находятся методом наименьших квадратов.

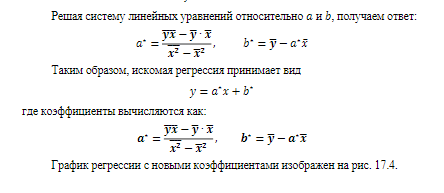
**Метод наименьших квадратов.**

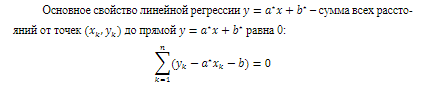
Для каждой точки, описываемой и , применяется следующая функция – наименьшее расстояние до каждой точки:

Для нахождения a и b запишем функцию :

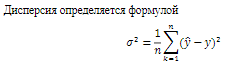
Найдем частные производные по a и b:







1. Доверительные интервалы; прогнозирование с помощью линейной регрессии.



Дисперсия описывает разброс данных в выборке. Если дисперсия вокруг тренда невелика, то результатам прогнозирования можно доверять. Если дисперсия велика – то вероятность того, что прогноз сбудется, мала.

Прогноз представляет из себя поиск примерного диапазона возможных значений, а не одного единственного. Таким образом, ошибка прогноза вычисляется из прогнозируемого диапазона и находится как квадратный корень из дисперсии:

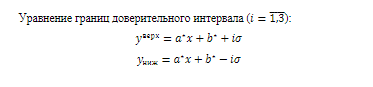


Прогнозируемое значение будет колебаться по определенному закону распределения. Самым часто встречающимся в природе является нормальное распределение Гаусса. Плотность вероятности зависит от среднеквадратичного отклонения следующим образом:

1. В доверительный интервал (+σ, -2σ) попадает ~68% всех точек.

2. В доверительный интервал (+2σ, -2σ) попадает ~95% всех точек.

3. В доверительный интервал (+3σ, -3σ) попадает ~99% всех точек.



**Прогнозирование.**

Итак, функция регрессии задает бесконечную прямую, которая продолжается и вне диапазона заданных значений. Значит, этот тренд можно продлить – экстраполировать на более широкий диапазон значений.

Экстраполяция – особый тип аппроксимации, при котором функция восстанавливаются вне границ заданного интервала. С помощью экстраполяции возможно прогнозировать значения для будущего диапазона.

1. Многокритериальная оптимизация по Парето.

Рассмотрим оптимизацию по Парето на примере задачи.

Необходимо выбрать лучших нападающих в футбольной команде по критериям максимальной личной результативности (голы) и командной сыгранности (голевые передачи).

Математическая запись задачи:

Найти множество игроков , лучших по критериям:

1) Наибольшего количества забитых голов:

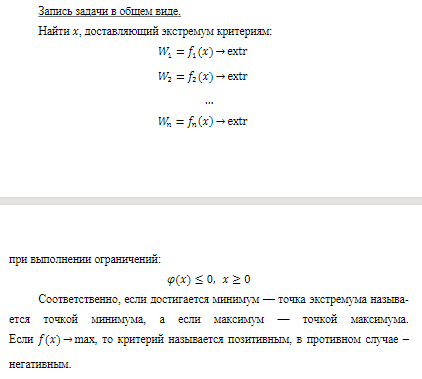
2) Выполненных передач:

Решением является множество Парето оптимальных игроков :

,

Доставляющее максимум критериям .

Так как решений много эксперту необходимо выбрать те, которые будут являться удовлетворительными исходя из соображений выгоды каждого из них в конкретный момент времени.



1. Многокритериальная оптимизация методом линейной свертки.

Линейная свертка критериев – способ сведения задач от множества до одного критерия оценки.

Необходимо выбрать лучших нападающих в футбольной команде по критериям максимальной личной результативности (голы) и командной сыгранности (голевые передачи).

Математическая запись задачи:

Найти множество игроков , лучших по критериям:

1) Наибольшего количества забитых голов:

2) Выполненных передач:

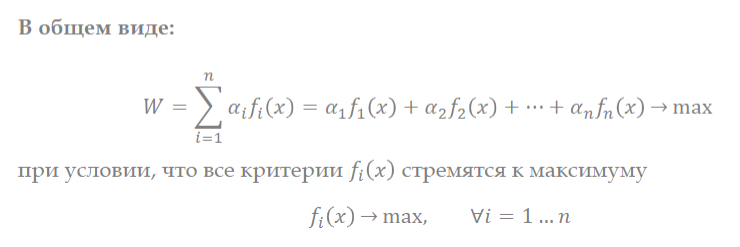
Переход двух критериев к одному:

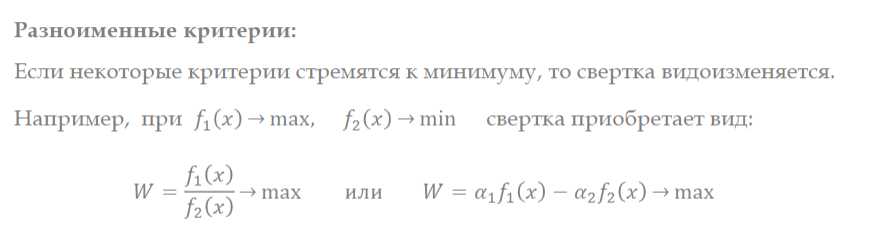
Производится путем их сворачивания в выражение:

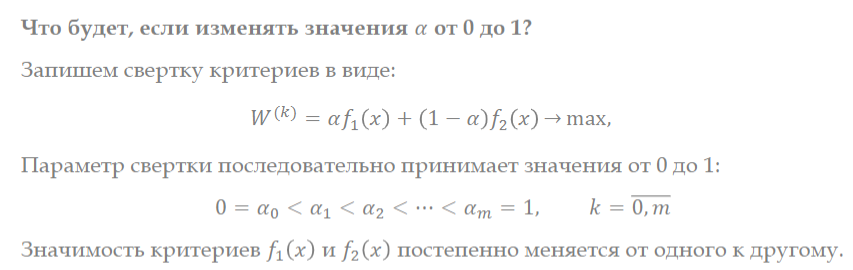
Где – весовые коэффициенты, отражающие значимость критериев.

= 1

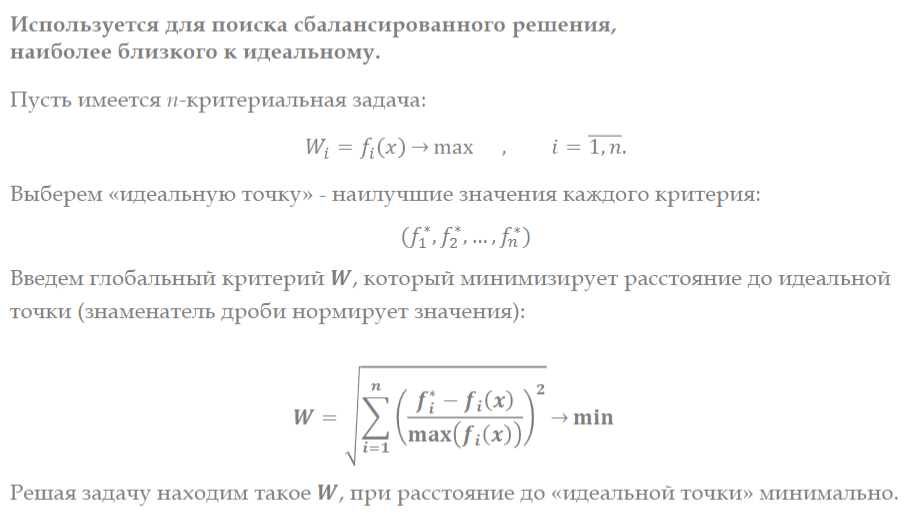
Далее решается классическая задача по одному обобщенному критерию.



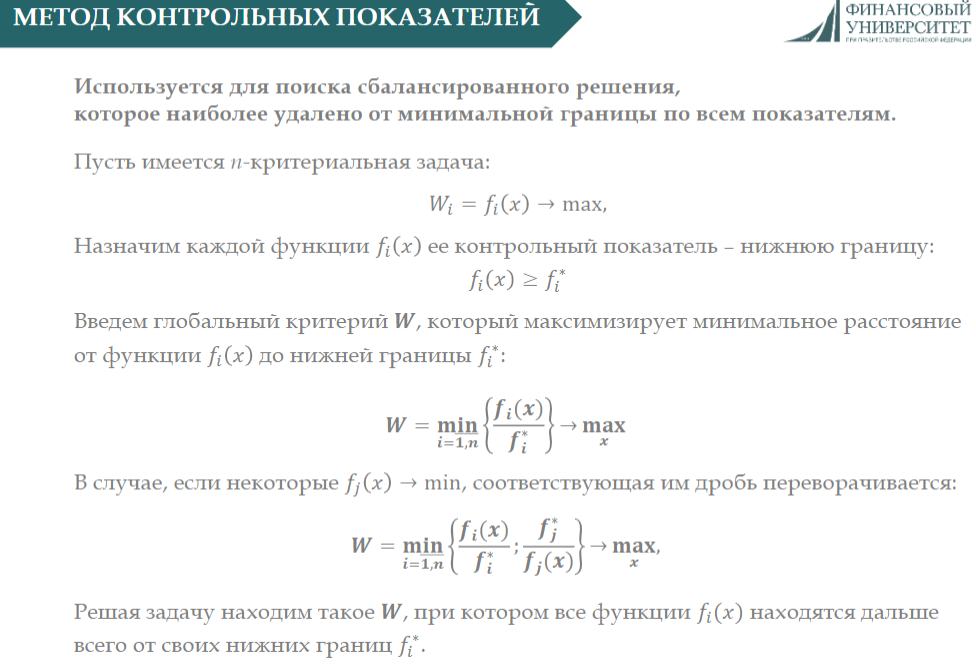




1. Многокритериальная оптимизация методом идеальной точки.



1. Многокритериальная оптимизация методом контрольных показателей.



Решая задачу, находим такое 𝑾, при котором все функции 𝑓𝑖(𝑥) дальше всего от своих нижних границ 𝑓𝑖 ∗ - то есть находим мин W по строке и выбираем максимум.

Визуальное представление результатов можно отразить на лепестковой диаграмме.

СВЕДЕНИЕ К ОДНОКРИТЕРИАЛЬНОЙ ЗАДАЧЕ применяется без участия человека – при автоматизированной обработке возможно единственное решение.

Ограничение: значение функции должно быть больше или равно нижней границе, те которые меньше отсекаются

1. Разработка регламента экспертных оценок; подбор экспертов.

***1)При разработке регламента проведения экспертных оценок определяется:***

Количество туров экспертизы:

1) один – эксперты однократно выбирают подходящее, по их мнению, решение;

2) несколько – проводится несколько итераций экспертизы для корректировки оценки на каждой следующей, с учетом итогов предыдущей;

3) неопределенное количество (пока не будет найдено компромиссное решение).

Степень общения экспертов:

1) отсутствует – эксперты не имеют каналов общения;

2) заочное анонимное – взаимодействуют заочно и анонимно;

4) очное с ограничениями – имеют возможность взаимодействовать очно, но в опред условиях;

3) заочное личностное – взаимодействуют заочно, но не анонимно;

5) очное без ограничений – имеют возможность взаимодействовать очно и свободно.

Допускаются различные комбинации регламентов исходя из условий задачи.

**При разработке регламента сбора экспертных оценок определяется:**

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).

Степень рассогласованности мнений:

1) Хорошо, если мнения экспертов несогласованы – мы получаем большее количество различных вариантов, среди которых найдется верный.

2) Плохо, если мнения экспертов несогласованы – мы получает отсутствие единства мнений, возможно эксперты некомпетентны или задача не точна.

**При разработке регламента интерпретации результатов экспертных оценок:**

Следует помнить два исключения из правил, которые встречаются на практике:

1) «Догма согласованности» – не обязательно большее количество экспертов выражают верное мнение, иногда мнение «диссидентов» - правильное.

2) «Догма одномерности» - не все результаты можно выразить одним числом или упорядочиванием по единственному даже интегральному признаку.

*2)Подбор экспертов.*

1. Составление списка возможных экспертов:

а) самостоятельное составление реестра экспертов;

б) 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.).

Способы определения компетентности экспертов:

а) по результатам научной деятельности (необходимы эксперты для оценки);

б) самооценка экспертов (завышенная или заниженная);

в) оценка друг друга (возможна «клановость»).

1. Метод средних баллов при проведении экспертных оценок.

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.

Пример – фокус-группа из 7 экспертов выбирает лучшее мороженное из 5 видов. Каждый эксперт назначает каждому виду оценку от 1 до 10.

Обработка результатов по каждому виду производится методами математической статистики:

Среднее количество баллов образца – Мат ожидание:



Данный показатель отражает средний балл, который получил образец с учетом оценок.

Степень рассогласованности мнений эксперт – дисперсия по выборке:



Если дисперсия – мала, то мнения экспертов согласованны, в противном случае, их мнения сильно отличаются.

Интервальная оценка баллов – доверительный интервал, например, ±2σ:



Интервальная оценка отражает, насколько сильно разбросаны мнения экспертов, относительно среднего значения. На основе этой информации можно судить, корректно ли в итоге оценены сравниваемые объекты.

1. Методы медианных и средневзвешенных рангов при проведении экспертных оценок.

Метод медианных рангов.

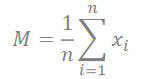
Метод сравнения более сложных объектов. Применяется в том случае, если невозможно назначить баллы, но можно расставить по рангу – отранжировать объекты. Подходит для сравнения сложных объектов, для которых можно сказать, что один вариант лучше другого, но нельзя указать количественно на сколько.

Медиана (Me) – это некоторая отметка, делящая ранжированные данные (отсортированные по возрастанию или убыванию) на две равные части. Половина исходных данных меньше этой отметки, а половина – больше.

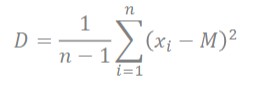
В сравнении с мат.ожиданием, медианный ранг не подвержен случайным выбросам и отражает мнение большинства.

Метод средневзвешенных рангов. Средневзвешенные ранги определяются с учетом компетентности экспертов:

1. Определяется среднее арифметическое рангов (матожидание) – формула СРЗНАЧ.



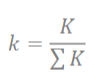
1. По каждому эксперту рассчитывается степень отклонения его ответа от среднего значения – то есть дисперсия мнений каждого эксперта (смотрим по одному эксперту) – формула СУММКВРАЗН (столбик оценок эксперт 1; столбик матожиданий всех объектов) / кол-во объектов.



1. Чем больше дисперсия, тем сильнее мнение эксперта отличается от мнения большинства. Следовательно, его компетентность в данном вопросе можно считать ниже, чем у коллег (исключая догму «согласованности»). Компетентность эксперта будет обратно пропорциональна его дисперсии – формула 1/D.



1. Выразим значимость мнения каждого эксперта в процентах (нормируем) и приведем его к виду коэффициента компетентности – компетентность эксперта/ сумму компетентностей всех экспертов.

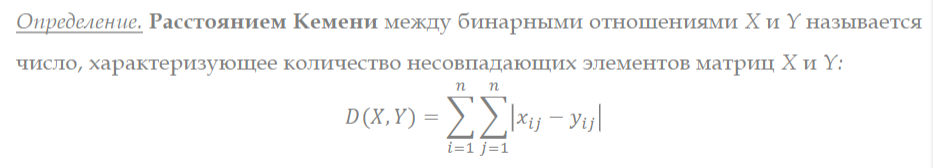
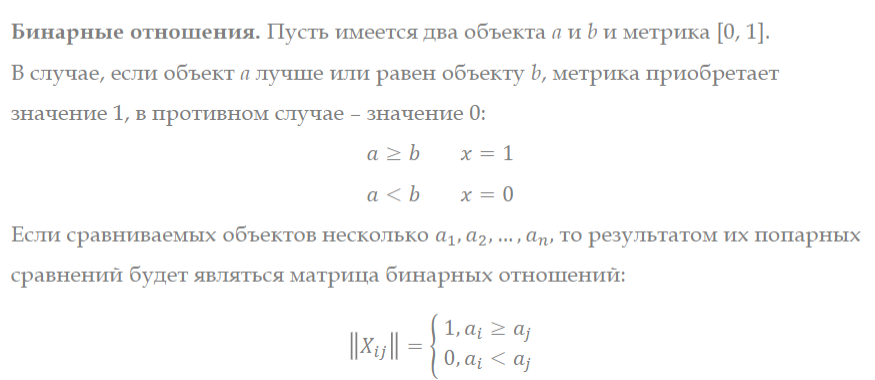


1. Далее найдем средневзвешенное мнение экспертов с учетом коэффициента их компетентности (важности их мнения) – формула СУММПРОИЗВ (оценки экспертов по объекту; коэффициенты компетентности экспертов).

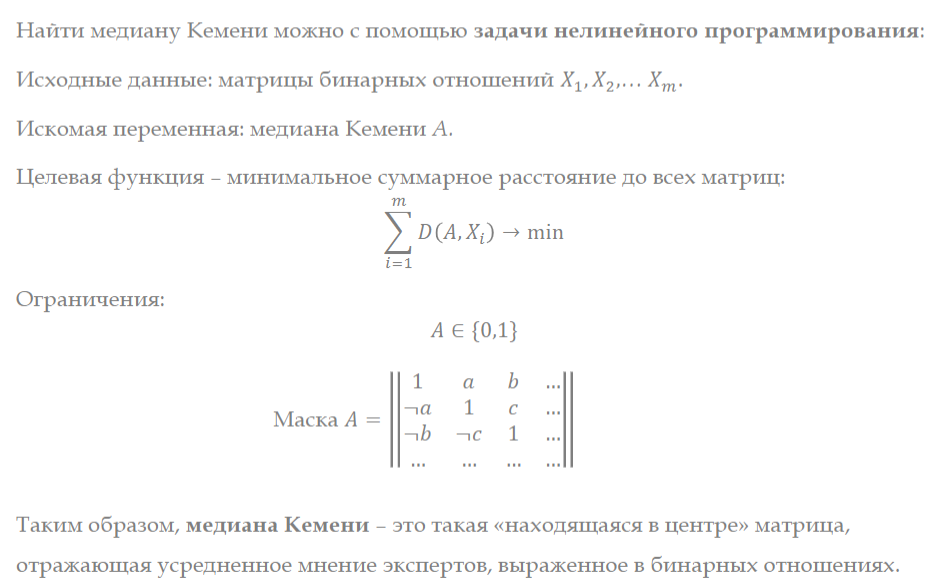
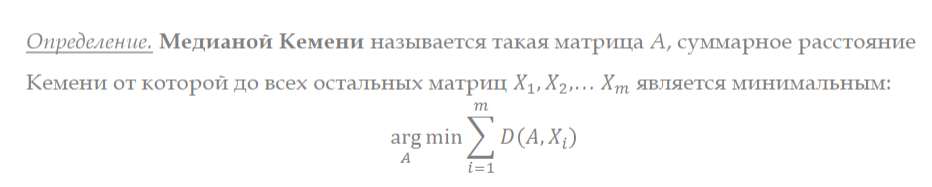


1. Методы бинарных отношений при проведении экспертных оценок.

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



Расстояние Кемени между мнениями экспертов 1 и эксперта 2 рассчитывается как вычитание матриц их бинарных отношений, взятие модуля и сложение элементов получившейся матрицы.

Если медиана Кемени совпала с мнением какого-либо эксперта, то его мнение можно принимать за усредненное мнение всех членов экспертной группы.

Вывод: в методе бинарных отношений внимание эксперта сужено до решения простой задачи – какой из двух вариантов лучше. Это сделать проще, чем охватить область всех вариантов, еще и назначив каждому из них баллы или ранги. Поэтому метод бинарных отношений дает более точные ответы, когда множество альтернатив велико или используются сложные объекты, но трудозатратен в обработке.

1. Структура системы массового обслуживания; потоки заявок и вероятности состояний системы.

Теория массового обслуживания / теория очередей (queueing theory) — это раздел математической экономики, занимающийся вопросами моделирования и оптимизации систем массового обслуживания с большим количеством клиентов.

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

• поступающие новые заявки (входящие покупатели);

• ожидающие в очереди заявки (очередь на кассу);

• обслуживающие устройства (кассы);

• обслуженные исходящие заявки (выходящие с покупками).

2) Интенсивность входного потока заявок.

Скорость поступления заявок в систему – сколько заявок приходит в систему за определенный интервал времени: 𝜆 = 𝑁 / T или 1/tвх (средний интервал времени между поступлением заявок).

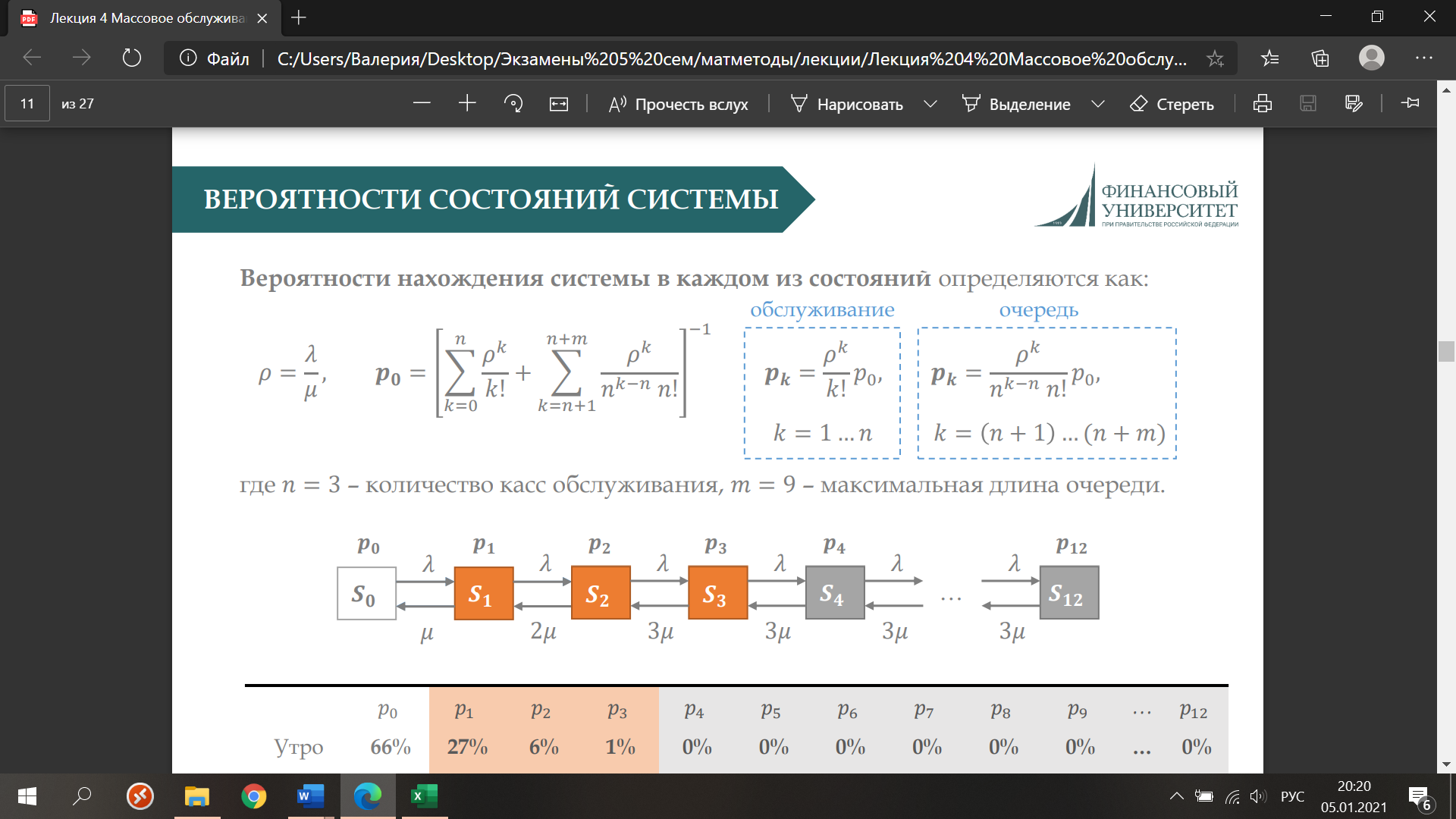
Интенсивность выходного потока заявок от одного прибора.

Скорость обработки заявок – сколько заявок обрабатывает один прибор в единицу времени:

𝜇 = 𝑁 /𝑛t, где 𝑁 – количество заявок, обработанное за период времени 𝑡 всеми 𝑛 приборами или 1/tвых (𝑡 вых - среднее время на обработку одной заявки одним прибором).

3) Вероятности нахождения системы в каждом из состояний определяются как:

𝜌 = 𝜆 / 𝜇 – показатель нагруженности системы = интенсивность входного потока/ интенсивность выходного потока.



P0 (вероятность простоя) рассчитывается по формуле = сумма в -1 степени ( всех p в степени k, деленных на факториал k и p в степени k,деленных на n в степени k-n,умноженное на факториал n)

Вероятность то, что заявка обслуживается или в очереди находится по формуле P0 умножить на p в степени k, деленное на факториал k)

n- количество обслуживающих приборов, m – длина очереди ( находили с помощью формулы округление вверх интенсивности входного потока(лямбды)).

1. Показатель нагруженности и сбалансированность системы; показатели для клиентов и владельцев систем массового обслуживания.

1)Показатель нагруженности 𝝆 – насколько система справляется с потоком клиентов: 𝝆 = 𝝀/ 𝝁

𝝆 ≪ 𝒏 – система недогружена – выгодно для клиента – нет очередей, невыгодна для владельца – лишние кассы, большой простой по времени;

𝝆 < 𝒏 – система сбалансирована для клиента – приемлемые очереди, допустимый простой касс;

𝝆 ≤ 𝒏 – система сбалансирована для владельца – большие очереди клиентов, кассы заняты практически полностью ;

𝝆 > 𝒏 – система перегружена – выгодно для владельца – заявок больше, чем можно обработать; невыгодно для клиента – бесконечно растущая очередь.

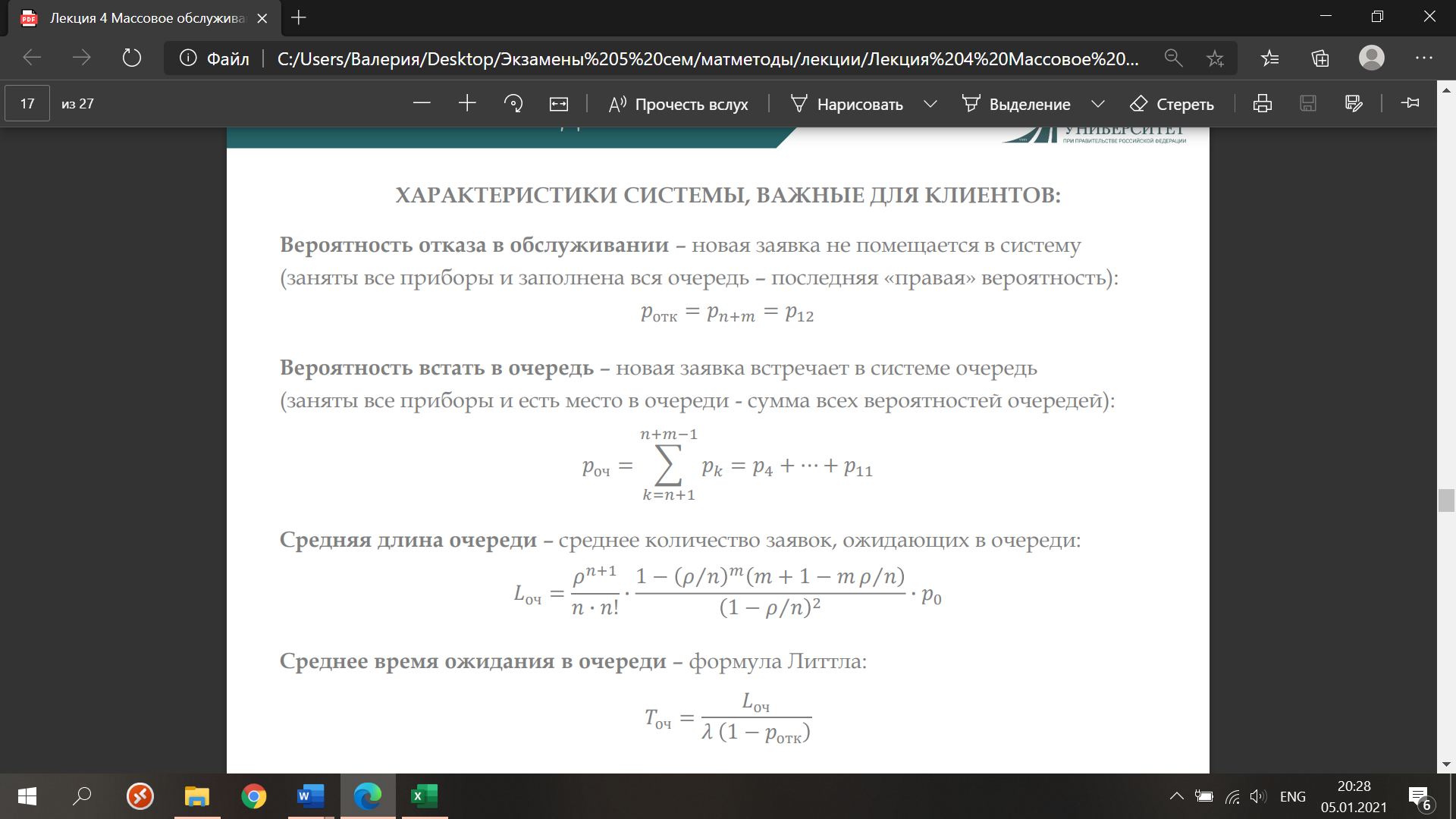
2)Показатели важные для клиентов:

Вероятность отказа в обслуживании - % потерянных заявок pотк= pn+m - n- количество обслуживающих приборов, m – длина очереди ( находили с помощью формулы округление вверх интенсивности входного потока(лямбды)).

Вероятность обслуживания – 1- вероятность отказа в обслуживании.

Средняя длина очереди – по формуле внизу

Среднее время ожидания в очереди – средняя длина очереди / абсолютную пропускную способность



3)Показатели для владельца СМО:

Относительная пропускная способность- % обслуженных заявок = вероятности обслуживания

Абсолютная пропускная способность – количество обработанных заявок в единицу времени – интенсивность входного потока \*относительную пропускную способность

Среднее количество занятых приборов –абсолютная пропускная способность / интенсивность выходного потока

Коэффициент простоя – 1-среднее количество занятых приборов/ количество всех приборов

