**Математическое программирование** - область математики, разрабатывающая теорию и численные методы решения многомерных экстремальных задач с ограничениями, т. е. задач на экстремум функции многих переменных с ограничениями на область изменения этих переменных.

**Комбинаторный анализ** – раздел математики посвященный решению задач выбора и расположения элементов некоторого, обычно конечного, множества в соответствии с заданными правилами

**Комбинаторная конфигурация** – способ построения конструкции из элементов исходного множества

|  |  |
| --- | --- |
| Имеется множество, состоящее из n элементов. Количество элементов множества всех подмножеств вычисляется по формуле | 2^n |
| Имеется множество, состоящее из n элементов. Количество сочетаний элементов мощностью m вычисляется по формуле | n!/((n-m)!m!) |
| Имеется множество, состоящее из n элементов. Количество перестановок данного множества рассчитывается по формуле | n! |
| Имеется множество, состоящее из n элементов. Множество размещений множества n по m элементов рассчитывается по формуле | n!/(n-m)! |

**ЛП** – метод математического моделирования, разработанный для оптимизации использования ограниченных ресурсов.

**Путь** - Не от исходной вершины к конечной, т.е. некоторое промежуточное ребро

**Полный путь** - От исходной к конечной

**Критический путь** – это максимально возможная суммарная пропускная способность

**Метод ветвей и границ** - это общий алгоритмический метод решения задач комбинаторной оптимизации. По существу, метод является вариацией полного перебора с отсевом подмножеств допустимых решений, заведомо не содержащих оптимальных решений. В основе метода лежат две процедуры: ***процедура ветвления***, позволяющая разбивать множество допустимых решений на непересекающиеся подмножества, и ***процедура*** ***вычисления*** нижней или верхней ***границы***.

**Симплекс метод** - алгоритм решения оптимизационной задачи линейного программирования путём перебора вершин выпуклого многогранника в многомерном пространстве.

**Метод потенциалов** - В методе потенциалов каждой стоке *i* и каждому столбцу *j* транспортной таблице ставится в соответствие числа (потенциалы) *Ui* (поставщики) и *Vj* (потребители). Для каждой базисной переменной *xij* потенциалы *Ui*  и *Vj* удовлетворяют уравнению:

*Ui +* *Vj* = *Cij*

**Теорема форда фалкерсона** - В любой сети максимальная величина потока из истока s в сток t равна минимальной пропускной способности разреза отделяющего s от t;

**Теорема джорджа-троттера** – Множества всех перестановок

**Транспортная задача** – специальный класс задач линейного программирования. Они описывают перевозку какого-либо товара из пункта направления в пункт назначения.

Назначение транспортной задачи – определение объёма перевозок из пунктов направления в пункт назначения с минимальной суммарной стоимостью перевозок.

Математическая модель: *Xij , i=~~1,m~~ , j=~~1,n~~  -* решение задачи.

Целевая функция: – целевая функция.

Алгоритм решения транспортной задачи:

1. Построение на0чального базисного решения:

* Метод северо-западного угла
* Метод наименьшей стоимости
* Метод Фогеля

1. Метод потенциалов

**Открытая тз** – число поставщиков не совпадает с числом потребителей и наоборот. Решение открытой задачи сводится к решению закрытой задачи

**Закрытая задача** – число поставщиков равно число потребителей

**Метод наименьшей стоимости**

Транспортная задача с *m* пунктами направления и с *n* пунктами назначения

Имеют *n+m* ограничений в виде равенств по одному на каждый пункт направления и назначения. Т. к. транспортная задача должна быть сбалансирована, то одно из этих равенств избыточно. Поэтому задача имеет *n+m+1* независимых ограничений и начальное базисное решение состоит из *n+m+1* базисных переменных.

Сначала в таблице находим ячейку с наименьшей стоимостью, затем переменной в этой ячейке присваивается наибольшее значение, допустимое ограничениями по спросу и предложению. Если таких ячеек несколько, то выбор произволен. Далее вычёркиваем соответствующий столбец или строка и корректируется спрос и предложения. Затем просматриваются не вычеркнутые ячейки и выбирается новая ячейка с min стоимостью.

**В методе потенциалов** каждой стоке *i* и каждому столбцу *j* транспортной таблице ставится в соответствие числа (потенциалы) *Ui* (поставщики) и *Vj* (потребители). Для каждой базисной переменной *xij* потенциалы *Ui*  и *Vj* удовлетворяют уравнению:

*Ui +* *Vj* = *Cij* Потенциалы *Ui , i=~~1,3~~ j=~~1,4~~*

>> Определяем потенциалы для всех базисных переменных…

>> Присваиваем одному из них произвольное значение (обычно 0), и считаем исходя их полученных ранее уравнений (*Ui +* *Vj* = *Cij*)

>> Определяем небазисные переменные для свободных клеток. Вычисляем по формуле *Ui +* *Vj* **-** *Cij*

>> Вводимой в базис переменной будет та, которая имеет наибольшее положительное значение

>> Обозначим через Т количество груза перевезённого через 2,2. Сначала строим замкнутый цикл, который начинается и заканчивается в искомой ячейке. Цикл состоит из последовательности вершин и горизонтальных отрезков, соединяющих ячейки с базисными переменными ввод. в ячейку(2,2).

>> Чтобы удовлетворять ограничениям по спросу и предложению надо поочерёдно отнимать и прибавлять Т к значению базисных переменных расположенных в угловых ячейках цикла.

>> Направление обхода не имеет значения.

>> Определяем допустимое решение…

>> Определяем значение функции цели…

>> Пункт 1….

**Динамическое программирование**

Если в процессе выполнения рекурсивного алгоритма одна и та же задача решается несколько раз, то говорят, что алгоритм содержит ***перекрывающиеся подзадачи***.

Метод решения задачи оптимизации, реализующей рекурсивный алгоритм с перекрывающимися подзадачами, в котором каждая такая подзадача решается один раз, а ее результат сохраняется для последующего применения, называется дм

**Линейное программирование –** описание действий, которые выполняются однократно в заданном порядке

[**Сетевое планирование**](https://math.semestr.ru/setm/examples_param.php)

**Путь** – любая последовательность работ, в которой конечное событие каждой работы совпадает с начальным событием следующий за ней работы.   
**Полный путь L** – любой путь, начало которого совпадает с исходным событием сети, а конец – с завершающим.   
**Наиболее продолжительный полный путь в сетевом графике называется критическим**. Критическими также называются работы и события расположенные на этом пути. Работы этого пути определяют общий цикл завершения всего комплекса работ, планируемых при помощи сетевого графика. И для сокращения продолжительности проекта необходимо в первую очередь сокращать продолжительность работ, лежащих на критическом пути.

[**Базисное решение**](Экзамен%20и%20презентации/Презентации%20к%20лекциям/Лекция12_13_Линейное_программирование.pptx)

**Опорный план** – план с определёнными через свободные базисными переменными.   
**Базисный план** – опорный план с нулевыми базисными переменными.   
**Оптимальный план** – базисный план, удовлетворяющий оптимальной функции цели (ФЦ).

**Ведущий (разрешающий) элемент** – коэффициент свободной неизвестной, которая становится базисной, а сам коэффициент преобразуется в единицу.   
**Направляющая строка** – строка ведущего элемента, в которой расположена с единичным коэффициентом базисная неизвестная, исключаемая при преобразовании (строка с минимальным предельным коэффициентом, см. далее).   
**Направляющий столбец** – столбец ведущего элемента, свободная неизвестная которого переводится в базисную (столбец с максимальной выгодой, см. далее).

Переменные x1, …, xm, входящие с единичными коэффициентами только в одно уравнение системы, с нулевыми – в остальные, называются базисными или зависимыми. В канонической системе каждому уравнению соответствует ровно одна базисная переменная. Переход осуществляется с помощью [метода Гаусса–Жордана](https://math.semestr.ru/gauss/jordan.php). Основная идея этого метода состоит в сведении системы m уравнений с n неизвестными к каноническому виду при помощи элементарных операций над строками.   
Остальные *n-m* переменных (xm+1,…, xn) называются **небазисными** или независимыми переменными.

**Базисное решение** называется допустимым базисным решением, если значения входящих в него базисных переменных xj≥0, что эквивалентно условию неотрицательности bj≥0.   
Допустимое базисное решение является угловой точкой допустимого множества S задачи линейного программирования и называется иногда опорным планом.   
Если среди неотрицательных чисел bj есть равные нулю, то допустимое базисное решение называется вырожденным (вырожденной угловой точкой) и соответствующая задача линейного программирования называется вырожденной.  
Вопрос про метод потенциалов(там ответ связан с тр. задачей, я выбрал 4-й по счету ответ в тесте)  
Снова тр. з., вопрос про цикл обхода, где ставятся + и -, крч там в ответах должен выбрать что выбирается минимальное значение груза в ячейках цикла имеющих знак «-»  
  
Про симплекс-метод, там чета типа что для него характерно  
варианты ответов были мол система может содержать равенства/неравенства, для этого метода характерна линейная или не линейная зависимость(из них я выбрал равенства/неравенства и линейная зависимость)  
  
переменные в линейной зависимости должны быть неотрицательными  
  
вопрос снова про линейную зависимость про графический метод (точно не помню формулировку)  
типа как найти оптимальное решение  
и варианты ответов будут точка, две точки, отрезок, интервал( я выбрал первые три)  
  
была такая формулировка вопроса про базисное решение :Пусть ограничение задачи линейного программирования представлено в виде m равенств с n переменными и m<n  
(почитай про базисный метод в линейной проге)  
  
модель что такое  
про сети крч будет два графа представлено, на одном маршрут надо будет выбрать(я взял 1-3-6) и еще на одном кратчайшее время или че(там 22 вроде ответ)  
  
еще вопросы про путь сетевой и полный сетевой путь

