### **Список экзаменационных вопросов по математическому и имитационному моделированию**



**I. Список вопросов по математическому моделированию**

### **1. Линейное программирование, графическое решение.**

Линейное программирование — раздел математического программирования, применяемый при разработке методов отыскания экстремума линейных функций нескольких переменных при линейных дополнительных ограничениях, налагаемых на переменные. По типу решаемых задач его методы разделяются на универсальные и специальные. С помощью универсальных методов могут решаться любые задачи линейного программирования. Специальные методы учитывают особенности модели задачи, ее целевой функции и системы ограничений. Особенностью задач линейного программирования является то, что экстремума целевая функция достигает на границе области допустимых решений. Классические же методы дифференциального исчисления связаны с нахождением экстремумов функции во внутренней точке области допустимых значений. Экономико-математическая модель — математическое описание исследуемого экономического процесса или объекта. Эта модель выражает закономерности экономического процесса в абстрактном виде с помощью математических соотношений. Использование математического моделирования в экономике позволяет углубить количественный экономический анализ, расширить область экономической информации, интенсифицировать экономические расчеты. Выделяем 3 этапа: 1) ставятся цели и задачи исследования, проводится качественное описание объекта в виде экономической модели. 2) формируется математическая модель изучаемого объекта, осуществляется выбор методов исследования, проводится программирование модели на ЭВМ, подготавливается исходная информация. 3) На основном этапе осуществляются анализ математической модели, реализованной в виде программ для ЭВМ, проведение машинных расчетов, обработка и анализ полученных результатов. Ограничения-равенства разрешают только те точки, которые лежат на соответствующей прямой, поэтому выделите на графике такие прямые: 1.Определите ОДР как часть плоскости, принадлежащую одновременно всем разрешенным областям, и выделите ее. При отсутствии ОДР задача не имеет решений, о чем сделайте соответствующий вывод. 2.Если ОДР - не пустое множество, то постройте целевую прямую, т.е. любую из линий уровня с1х1 + с2х2 = L, где L - произвольное число, например, кратное с1 и с2, т.е. удобное для проведения расчетов. Способ построения аналогичен построению прямых ограничений. 3.Постройте вектор C = (c1,с2), который начинается в точке (0;0), заканчивается в точке (c1,с2). Если целевая прямая и вектор С построены верно, то они будут перпендикулярны. 4. При поиске max ЦФ передвигайте целевую прямую в направлении вектора С, при поиске min ЦФ - против направления вектора С. Последняя по ходу движения вершина ОДР будет точкой max или min ЦФ. Если такой точки (точек) не существует, то сделайте вывод о неограниченности ЦФ на множестве планов сверху (при поиске шах) или снизу (при поиске min). Определите координаты точки max (min) ЦФ X = (х1\* ; х2\* ) и вычислите значение ЦФ l(x\* ). Для вычисления координат оптимальной точки X\* решите систему уравнений прямых, на пересечении которых находится X\* .

### **2. Линейное программирования, основная идея симплекс метода.**

Идея - обобщение графического метода для больших размерностей.

В отличие от графического метода, пространство многомерное. Целевая функция и ограничения содержат более двух переменных. Но они линейны.

Этапы:

1) Приводим систему ограничений к каноническому виду

2) Строим систему ограничений многомерного пространства.

Получаем многомерный многоугольник - симплекс.

3) Находим градиент целевой функции.

4) Перебираем вершины многогранника в направлении целевой функции и

5) Определяем оптимальное решение из этих вершин.

Подставляем значения в целевую функцию и находим ответ

### **3. Нелинейное программирование, особенности. Графическое решение.**

**Нелинейное** **программирование** – раздел математического **программирования**, изучающий задачи отыскания глобального экстремума фиксированной (целевой) функции при наличии ограничений в ситуации, когда целевая функция и ограничения имеют общий характер (не предполагаются линейными).

Особенности:

* Задачи НЛП значительно ближе к реальным ситуация, чем линейные;
* Решение задач нелинейного программирования может давать два или более экстремума, тогда как решение задач линейного программирования дает один экстремум;
* Задачи НЛП могут быть с ограничениями и без них;
* Множество допустимых планов *D* может иметь очень сложную структуру (например, ***невыпуклым, несвязным***, иметь бесконечное число крайних точек.);
* Глобальный максимум (минимум) может достигаться как внутри множества *D*, так и на его границах (где он, вообще говоря, будет не совпадать ни с одним из локальных экстремумов);
* Целевая функция *f* может быть недифференцируемой, что затрудняет применение классических методов математического анализа;
* Задачи НЛП настолько разнообразны, что для них не существует общего метода решения.

Если число переменных в задаче не превышает трех, то можно попытаться решить задачу графически.

Графический метод можно использовать для решения задачи НП, которая содержит две переменных *х*1 и *х*2, например задачи следующего вида:

*Z* = *f*(*x*1, *x*2)→ min (max);

*gi*(*x*1, *x*2)≤ *bi*, *.*

Чтобы найти ее оптимальное решение, нужно выполнить следующие действия:

1. Найти ОДР, определяемую ограничениями задачи. Если окажется, что эта область пуста, то это означает, что задача не имеет решения.
2. Построить семейство линий уровня целевой функции *f*(*х*1, *х*2) = *C* при различных значениях числового параметра *С*.
3. При решении задачи на минимум определить направление убывания, а для задачи на максимум — направление возрастания линий уровня ЦФ.
4. Найти точку ОДР, через которую проходит линия уровня с наименьшим в задаче на минимум (соответственно, наибольшим в задачи на максимум) значением параметра *С.* Эта точка будет оптимальным решением. Если ЦФ не ограничена снизу в задаче на минимум (сверху — в задаче на максимум), то это означает, что задача не имеет оптимального решения.
5. Найти координаты точки оптимума и определить в ней значение ЦФ.

3 вида задач нелинейного программирования:

1) Нелинейная целевая функция, множество ограничений - линейно Максимум в точке касания

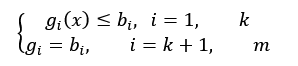
2) Нелинейная система ограничений, линейно заданная целевая функция Максимум в точке касания

3) Нелинейная система ограничений и целевая функция

### **4. Нелинейное программирование. Решение с помощью множителей Лагранжа, интерпретация решения.**

**/здесь несусветная жесть, но она была взята с его лекций/**

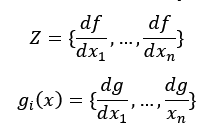
Z = R(x) которая стремится к min или max



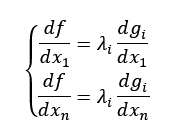
**1) в точке качания коэфф наклона y ограничения и целевой функции совпадают (касательные параллельны)**

**2) нормаль к этой касательной – это градиент**

**3) нормали к параллельным касательным точке параллельны**

****

**4) параллельность градиентов**

****

n уравнений, n+1 переменных

**5) добавляем целевое уравнение**

точка касания находится на i-том ограничении



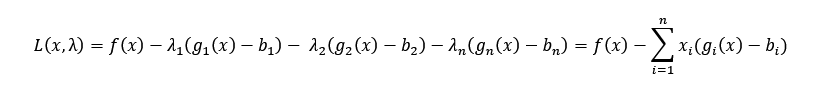
Для ограничений  (равенство) ограничения должны оставаться в системе.

Формальная запись метода ограничения. Все ограничения – равенства Лагранжа.

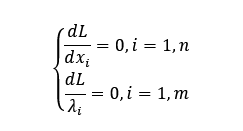


i=1, m

Z=f(x)→max



Составляем систему уравнений:



Не все полученные системы могут быть решены. Следует брать произвольную стартовую точку, рассчитывать градиент в этой точке, делать небольшой шаг в направлении градиента. Проверять ограничения, если они выполняются, делать очередной шаг, иначе делать шаг в обратную сторону меньшей длины.

### **5. Динамическое программирование. Условия применения, основное функциональное уравнение Беллмана.**

*Динамическое программирование - метод оптимизации, приспособленный к операциям, в которых процесс принятия решений может быть разбит на отдельные этапы, шаги.*

Такие операции называются многошаговыми. *Если управление сводится к однократному принятию решения, то соответствующая задача называется одноэтапной или одношаговой.Если управление требует некоторой последовательности принятых решений, то такая задача называется многоэтапной или многошаговой.*

Условия применения динамического программирования:

\*отсутствие последствия. Качество управления зависит только от текущего состояния, выбранного в этом состоянии.

\*аддитивность целевой функции. Выигрыш за все шаги есть сумма выигрышей на каждом шаге.

Введем обозначения.

k-номер шага

{S^k}-множество состояний на каждом шаге

S^k-одно из состояний из множества {S^k}

В каждом состоянии S^k возможно множество управлений {US^k}.

W(S^k, {US^k})-качество управления, когда состояние S^k выбирает управление US^k.

*Любая последовательность действий для каждого шага, переводящая систему из начального состояния в конечное, называется стратегией управления. Допустимая стратегия управления, доставляющая функции цели экстремальное значение, называется оптимальной.*

**Принцип оптимальности:** *если некоторая последовательность решений оптимальна, то на любом шаге последующие решения образуют оптимальную стратегию по отношению к результату предыдущих решений.*

На основе принципа оптимальности Беллмана строится схема решения многошаговой задачи, состоящая из 2-х частей:

**1) Обратный ход:**от последнего шага к первому получают множество возможных оптимальных («условно-оптимальных») управлений.

**2) Прямой ход:**от известного начального состояния к последнему из полученного множества «условно-оптимальных» управлений составляется искомое оптимальное управление для всего процесса в целом.

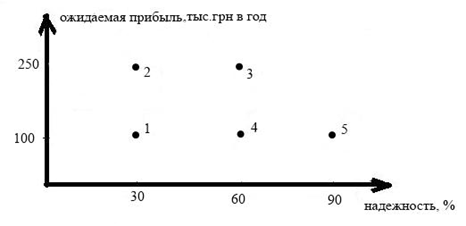
Оптимальную стратегию управления можно получить, если сначала найти оптимальную стратегию управления на n-м шаге, затем на двух последних шагах, затем на трех последних шагах и т.д., вплоть до первого шага.

Составляем уравнение Беллмана:

F(S^k)-оптимальный выигрыш, который можно получить из состояния S^k и все последующие шаги.

F(S^k)=max(внизу под max- US^k,также вместо max мб min)[ W(S^k, {US^k})+ F(S^(k+1))].

### **6. Многокритериальная оптимизация.**

Задача многокритериальной оптимизации состоит в поиске вектора целевых переменных, удовлетворяющий наложенным ограничениям и оптимизирует векторную функцию, элементы которой соответствуют целевым функциям. Эти функции образуют математическое описание критерия удовлетворительности и, как правило, взаимно конфликтуют. Отсюда, «оптимизировать» означает найти такое решение, при котором значение целевых функций были бы приемлемыми для постановщика задачи.  
**Оптимизация по Парето**  
Среди всех исходов находим множество исходов, которое лучше остальных. Останутся точки, которые не сравнимы. Из них не сможем определить какая из них лучше.  
Оптимальность по Парето — такое состояние системы, при котором значение каждого частного показателя, характеризующего систему, не может быть улучшено без ухудшения других.  
   
Пример: Стратегия 2 в среднем дает больше прибыли, чем стратегия 1, при той же надёжности. Стало быть, стратегия 1 не может быть лучшей. Стратегия 3 по ожидаемой прибыли равноценна стратегии 2, но надёжнее. Стало быть, стратегия 2 тоже невыгодна. Стратегия 3 прибыльнее стратегии 4 при той же надёжность, то есть стратегия 4 тоже неоправданно. Остаются только стратегии 3 и 5. По одному критерию превосходит одна, по-другому - другая.  
**Метод весовых коэффициентов**  
Z = w1 \* f1 + w2 \* f2 + ...+ wn\* fn (w - вес)  
Сумма всех wi равна 1. Выбор весов остается за лицом, которое принимает решение.  
Целевые функции смещаются по пропорциям.  
**Метод приближения к желаемой оптимальной точке**Мы знаем лучшую точку, но она находится за областью допустимых значений.  
Опускаем из этой точки перпендикуляр на область допустимых значений. Точка пересечения с областью допустимых значений будет наиболее близкой к желаемой.

### **7. Сетевые методы, поиск потока минимальной стоимости.**

Типы задач:

1) поиск кратчайшего пути

2) задача коммивояжера

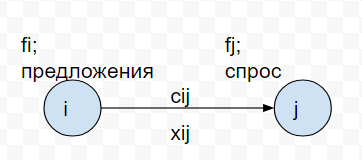
3) построение минимального остовного дерева

4) задача поиска потока минимальной стоимости

5) сетевое планирование процессов

6) Марковские процессы

Есть множество узлов, часть узлов является точками предложения, другая часть -точки спроса, есть еще транзитные узлы. Пункты спроса и пункты предложения могут быть транзитными.



xij - количество товара, движущегося от i в j (пропускная способность). cij - стоимость одной транспортировки товара.

Целевая функция - минимум стоимости потоков. Нужно ее минимизировать. Переменными являются значениями потоков между узлами.

Рассматриваем каждый узел и записываем уравнение по принципу - все что втекает, все и вытекает). Составляем систему ограничений. При асимметрии спроса и предложения добавляем новые точки или изменяем равенство на неравенство. Равенство выбираем в зависимости от того что больше: спрос или предложение.

Сумма всех входящих потоков = сумме всех вытекающих потоков

Z = ΣiΣj cij \* xij -> min

Находим все xij и подставляем в целевую функцию.

Частным случаем задачи о потоке минимальной стоимости является ***транспортная задача***, в которой имеется *двудольный граф* (двудольным называется граф, множество вершин которого может быть разбито на два непересекающихся подмножества, причем ребра (дуги) графа соединяют вершины только из разных подмножеств)

### **8. Сетевые методы, сетевое планирование проектов, метод критического пути.**

Методы сетевого планирования — методы, основная цель которых заключается в том, чтобы сократить до минимума продолжительность проекта. Основываются на разработанных практически одновременно и независимо методе критического пути МКП (СРМ — Critical Path Method) и методе оценки и пересмотра планов ПЕРТ (PERT — Program Evaluation and Review Technique). Критический путь — максимальный по продолжительности полный путь в сети называется критическим; работы, лежащие на этом пути, также называются критическими. Именно длительность критического пути определяет наименьшую общую продолжительность работ по проекту в целом. Длительность выполнения всего проекта в целом может быть сокращена за счет сокращения длительности работ, лежащих на критическом пути. Соответственно любая задержка выполнения работ критического пути повлечет увеличение длительности проекта. Метод критического пути позволяет рассчитать возможные календарные графики выполнения комплекса работ на основе описанной логической структуры сети и оценок продолжительности выполнения каждой работы, определить критический путь для проекта в целом.

### **9. Марковские цепи. Основные задачи, решаемые с помощью Марковских цепей. Формулы.**

Цепью Маркова называют такую последовательность случайных событий, в которой вероятность каждого события зависит только от состояния, в котором процесс находится в текущий момент и не зависит от более ранних состояний. Конечная дискретная цепь определяется:

1. **множеством состояний** *S* = {*s*1, …, *sn*}, событием является переход из одного состояния в другое в результате случайного испытания

2. **вектором начальных вероятностей** (начальным распределением) *p*(0) = {*p*(0)(1),…, *p*(0)(*n*)}, определяющим вероятности *p*(0)(*i*) того, что в начальный момент времени *t* = 0 процесс находился в состоянии *si*

3. **матрицей переходных вероятностей** *P* = {*pij*}, характеризующей вероятность перехода процесса с текущим состоянием *si* в следующее состояние *sj*, при этом сумма вероятностей переходов из одного состояния равна 1:

∑*j*=1…*n* *pij* = 1

Предположим, что проводится серия экспериментов с возможными исходами s1,s2,s3,…sns1,s2,s3,…sn. Назовём эти исходы **состояниями**.

· p(0)ipi(0) — вероятность того, что мы начинаем в состоянии sisi;

· pijpij — вероятность того, что в результате эксперимента состояние было изменено от состояния sisi к состоянию sjsj;

Если p(1)ipi(1) вероятность того, что исходом эксперимента будет состояние sisi. Тогда

p(1)i=p(0)1p1i+p(0)2p2i+p(0)3p3i+…+p(0)npnipi(1)=p1(0)p1i+p2(0)p2i+p3(0)p3i+…+pn(0)pni . (∗)(∗)

Это означает, что вероятность исхода в состоянии sisi равна сумме вероятностей начать эксперимент в некотором другом состоянии и окончить в sisi. Также заметим, что:

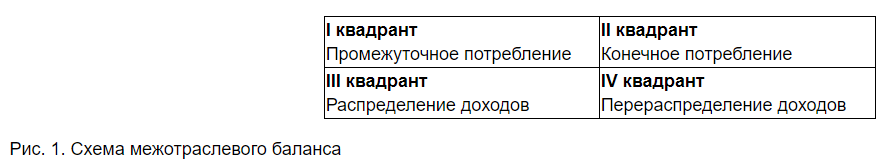
pj1+pj2+pj3+…+pjn=1pj1+pj2+pj3+…+pjn=1.

Цепи Маркова применяются при оценке будущих продаж. Например, сделав опрос среди покупателей той или иной марки автомобиля о их следующем выборе, можно составить матрицу P.

### **10. Модель межотраслевого баланса Леонтьева.**

Межотраслевой баланс - экономико-математическая балансовая модель, характеризующая межотраслевые производственные взаимосвязи между объектами экономической системы. Представляет из себя таблицу. Модель МОБ используется для макроэкономического анализа, так как охватывает весь процесс воспроизводства, отражает стоимостную и натуральную форму валового национального продукта, в нем представлены все основные показатели макроэкономики.

Модель МОБ В. Леонтьева отличается двояким рассмотрением отдельных отраслей - как покупателей материальных благ и услуг, предложенных другими отраслями, и как продавцов материальных благ и услуг, созданных ими самими. Данная характерная черта модели МОБ позволяет определить ее как модель "затраты-выпуск".



Модель В. Леонтьева может быть представлена уравнением

X = AX + Y, где

Х - объем производства какой-либо отрасли;

Y - конечный продукт данной отрасли;

А - матрица технологических коэффициентов аij, т.е. объем i-й отрасли для создания единицы продукции j-й отрасли.

### **11. Теория игр, позиционные игры с совершенной информацией.**

**/всё записано с его лекций/**

Теория игр изучает конфликтные ситуации между игроками.

Игры с совершенной информацией – каждый участник видит всю игру.

Теорема Куна в играх с совершенной информацией: есть хотя бы одно равновесие.

Метод обратной индукции анализирует дерево с конца в начало и определяет действия игроков.

В позиционных играх строится дерево игры T=(V,E), V-мн-во вершин, E-мн-во дуг, отображающих V само в себя (мн-во действий в каждой вершине).

V развито на подмножества V1, V2…Vn, Vn+1.



Vi *–* множество очередности игрока i

t=Vn+1 – множество терминальных вершин.

На множестве t определена функция выигрышей Hi(ν) игрока i.

I={1,2…, n} – множество игроков.

v = начальная вершина

G=<I,(V,E),Hi(v),V/t=UVi> - математическая запись игры

### **12. Теория игр, позиционные игры с несовершенной информацией.**

В игре с несовершенной информацией игроки могут не знать, в какой позиции они находятся (некоторые стохастические игры, в частности, карточные игры). К играм с несовершенной информацией сводятся игры с неполной информацией (также известные как байесовские игры). В отличие от игр с несовершенной информацией, где неполная информированность игроков возникает в процессе игры, в играх с неполной информацией неполная информированность некоторых игроков возникает еще до начала игры, как следствие ассимметричной информированности игроков (покупатель меньше знает о качестве товара, чем продавец, фирма точно не знает, какую технологию использует ее конкурент, и т. д.)

Если при выполнении очередного хода игрок не знает всей предыстории игры, т.е. всех выборов на предыдущих ходах (включая и случайные), то он фактически не знает в какой позиции осуществляется его выбор на данном ходе. Такие игры называются ***играми с неполной информацией***. Для описания неопределенности, с которой сталкивается *i*-й игрок на своем ходе, используется разбиение множества Pi на подмножества, называемые ***информационными множествами***, и принимается, что позиции, принадлежащие одному информационному множеству, для него неразличимы.

Таким образом, делая очередной ход, игрок знает, что он находится в одной из позиций данного информационного множества. При этом он может сделать разумный выбор лишь в том случае, когда число возможных альтернатив в каждой из позиций одно и то же и между ними есть однозначное соответствие.

### **13. Теория игр, игры N лиц, кооперативные игры, характеристическая функция.**

Теория игр — это раздел математической экономики, изучающий решение конфликтов между игроками и оптимальность их стратегий.

● Игры двух лиц занимают центральное место во всей теории игр. Основным понятием теории игр для игры двух лиц является обобщение весьма существенной идеи равновесия, которая естественно появляется в играх двух лиц. Что же касается игр n лиц, то одна часть теории игр посвящена играм, в которых сотрудничество между игроками запрещено. В другой части теории игр n лиц предполагается, что игроки могут сотрудничать для взаимной пользы.

● Кооперативная теория игр занимается изучением игр, в которых группы игроков — коалиции — могут объединять свои усилия. Этим она отличается от некооперативных игр, в которых коалиции неприемлемы и каждый обязан играть за себя.

● Согласно определению, кооперативной игрой называется пара (N, v), где N — это множество игроков, а v — это функция: 2N → R, из множества всех коалиций в множество вещественных чисел (так называемая характеристическая функция). Предполагается, что пустая коалиция зарабатывает ноль, то есть v(∅) = 0. Характеристическая функция описывает величину выгоды, которую данное подмножество игроков может достичь путем объединения в коалицию. Подразумевается, что игроки примут решение о создании коалиции в зависимости от размеров выплат внутри коалиции.

### **14. Теория игр, кооперативные игры, дележи, С – ядро.**

● Теория игр — это раздел математической экономики, изучающий решение конфликтов между игроками и оптимальность их стратегий.

● Кооперативная теория игр занимается изучением игр, в которых группы игроков — коалиции — могут объединять свои усилия. Этим она отличается от некооперативных игр, в которых коалиции неприемлемы и каждый обязан играть за себя.

Решением кооперативной игры является делёж, т. е. договор о распределении выигрыша коалиции между её членами.

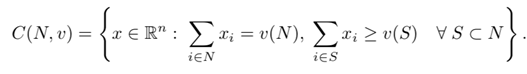
Дележ должен удовлетворять условиям:

1) Условие индивидуальной рациональности.x >=V(i), i∈ I. Каждому игроку должен быть предложен выигрыш не меньше, чем он мог бы выиграть самостоятельно.

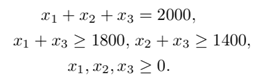
2) Условие коллективной (групповой) рациональности. Делёж реализует все потенциальные возможности данной игры

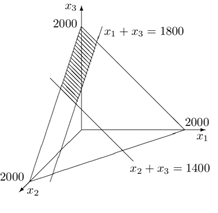
**С-ядро (многозначный дележ)** — принцип оптимальности в теории кооперативных игр, представляющий собой множество эффективных распределений выигрыша, устойчивых к отклонениям любой коалиции игроков(наличию переговорной силы у одних игроков над другими внутри коалиции)

Дележ x нестабилен из-за коалиции S ⊂ N, если . В таком случае суммарный выигрыш игроков коалиции S меньше того, что коалиция может получить действуя самостоятельно. Поэтому игроки коалиции S объединившись, смогут расстроить соглашение, приведшее к формированию дележа x.

Мы также говорим, что дележ нестабилен, если он нестабилен из-за какой-либо коалиции, иначе дележ называется стабильным. Множество C(N,v) всех стабильных дележей составляет с-ядро коалиционной игры. По определению 

Пример: Ядро игры задается следующими неравенствами:





Заштрихованная зона на этом рисунке и есть ядро.

### **15. Теория игр, кооперативные игры, дележи, НМ-решение.**

Рассмотрим некоторое множество дележей . Обозначим через  множество дележей, доминируемых дележами из  , а через L< множество дележей, доминирующих дележи из .

Решением игры по Нейману-Моргенштерну или НМ-решением будем называть такое множество дележей L, для которого , иначе говоря, если



Эти свойства называются соответственно внешняя устойчивость: для каждого дележа не из L найдется доминирующий его дележ из L, и внутренняя устойчивость: никакой дележ из L не доминируется никаким другим дележом из L.

ТЕОРЕМА 1. Если игра v линейно эквивалентна v', то всякому НМ-решению L игры v соответствует НМ-решение L' игры v'.

Доказательство. Пусть v' = rv + m. Рассмотрим биекцию

 существующую в силу теоремы 4 предыдущего параграфа. Так как эта биекция сохраняет отношение доминирования, то



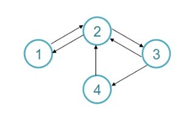
так что , а это означает, что L' есть НМ-решение игры v'.

### **16. Теория игр, кооперативные игры, дележи, вектор Шепли.**

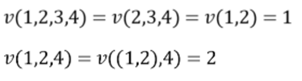
Вектор Шепли — принцип оптимальности распределения выигрыша между игроками в задачах кооперативной теории игр. Является распределением, в котором выигрыш каждого игрока равен среднему вкладу игрока в общий выигрыш всей коалиции:



Рассмотрим ситуацию, когда агенты объединяются в коалиции, таким образом, имеет место коллективного влияния. Для этого, используем следующую социальную сеть, которая состоит из 4 агентов:



Степень влияния агентов представлена следующими соотношениями:



Вычисляя вектор Шепли по формуле (1), получим следующие результаты:



### **17. Теория игр, кооперативные игры, дележи, сердцевина.**

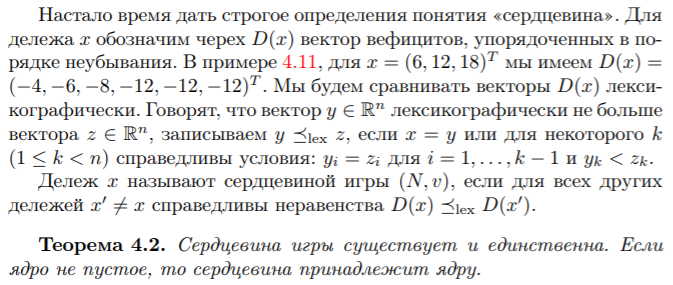
Кооперативной называется игра, в которой группы игроков - коалиции - могут объединять свои усилия. Подразумевается, что игроки примут решение о создании коалиции в зависимости от размеров выплат внутри коалиций.

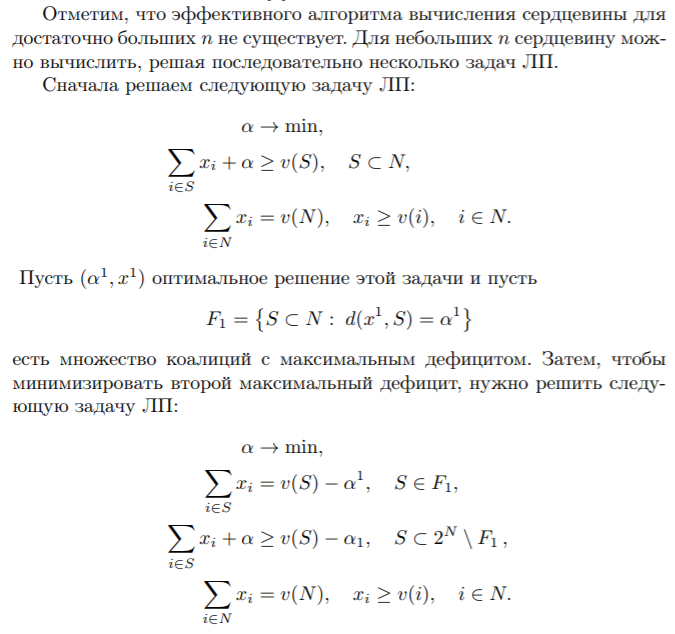
Исходом в кооперативной игре является дележ, возникающий не как следствие действия игроков, а как результат их соглашений.

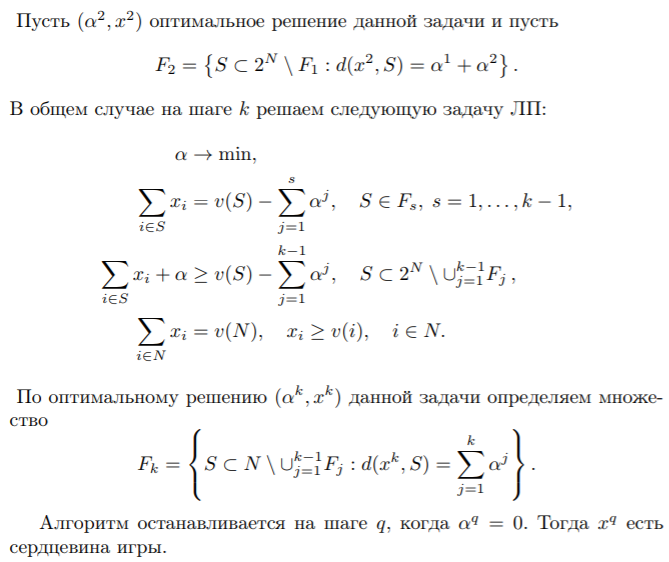
Дележ игроков должен удовлетворять следующим условиям:

* условие индивидуальной рациональности: xi>=v(i) для i є N, т.е. любой игрок должен получить выигрыш в коалиции не меньше, чем он получил бы, не участвуя в ней (в противном случае он не будет участвовать в коалиции)
* условие коллективной рациональности (условие эффективности): сумма выигрышей игроков должна соответствовать их возможностям. Другими словами, сумма выигрышей каждого из членов не должна превосходить уверенно получаемое ею количество.

Сердцевина - дележ, который вызывает наименьшую неудовлетворенность среди всех коалиций







### **II. Список вопросов по имитационному моделированию**

### **18. Суть имитационного моделирования.**

Имитационное моделирование – метод исследования системы с помощью замены реальной среды на компьютерную модель и дальнейшего проведения экспериментов над моделью системы. Модель – логические связи и функциональные отношения, описывающие логику работы элементов исследуемой системы.

Имитационное моделирование позволяет исследовать систему и получить приемлемый результат в том случае, когда невозможно проводить эксперименты на реальной системе и когда недоступно получение аналитического решения математической модели.

### **19. Механизмы продвижения модельного времени.**

При реализации имитационной модели используются обычно три представления времени:

· *реальное время* системы, функционирование которой имитируется;

· *модельное время*, по которому организуется синхронизация событий в модели;

· *машинное время* имитации, отражающее затраты ресурса времени компьютера.

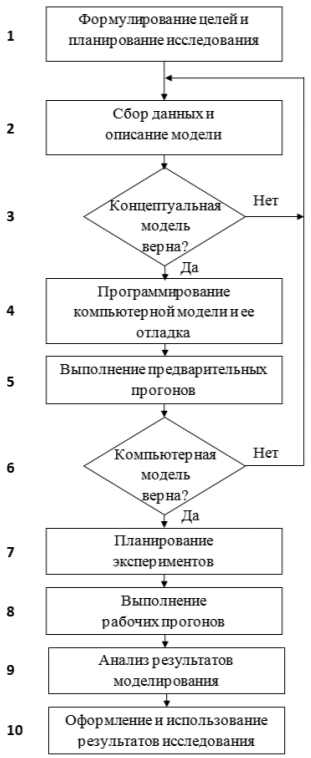
Продвижение времени в модели может быть организовано двумя способами:

· продвижение модельного времени с фиксированным переменным шагом ;

· продвижение модельного времени до очередного события (по принципу ).

### **20. Этапы построения имитационных моделей.**

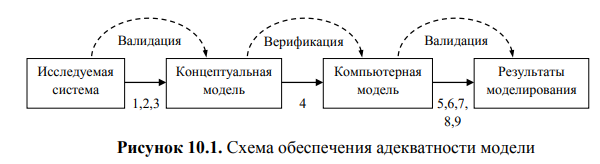
1. формулирование целей и планирование исследования
2. сбор данных и описание модели
3. концептуальная модель верна?
4. программирование компьютерной модели и ее отладка
5. выполнение предварительных прогонов
6. компьютерная модель верна?
7. планирование экспериментов
8. выполнение рабочих прогонов
9. анализ результатов моделирования
10. оформление и использование результатов исследования



### **21. Адекватность модели, схема обеспечения адекватности модели.**

Полученные в результате моделирования результаты и выводы должны быть эквивалентны результатам и выводам, полученным при проведении экспериментов на реальной системе. Процесс проверки такого соответствия называется проверка адекватности модели. Модель называется адекватной, если она является точным представлением реальной системы для конкретных целей исследования.

При проверке адекватности модели следует учесть некоторые особенности: - реальная система может еще не существовать, а лишь находиться в разработке; - абсолютно адекватных моделей быть не может, модель разрабатывается для определенных целей исследования; - модель адекватная для одних целей, может быть неадекватна для других целей; - процесс проверки адекватности модели все равно упирается в "человеческий фактор", критерии оценки правильности модели включают в себя лиц, принимающих решения ("а судьи кто?").



### **22. Адекватность модели и валидация.**

Модель называется адекватной, если она является точным представлением реальной системы для конкретных целей исследования.

Валидация - процесс позволяющий установить, является ли имитационная модель (концептуальное описание) точным представлением системы для конкретных целей исследования.

### **23. Адекватность модели и верификация.**

Модель называется адекватной, если она является точным представлением реальной системы для конкретных целей исследования.

В процесс проверки адекватности модели входят такие процессы, как валидация и верификация. Верификация - процесс проверки компьютерной программы (модели), позволяющий установить, правильно ли концептуальное описание модели преобразовано в компьютерную модель.

Верификация:

1) Лучше сперва писать основную часть программы, заменяя остальные подпрограммы заглушками (функции возвращают фиксированное значение), отлаживать и постепенно добавлять дополнительные подпрограммы.

2) Участие несколько человек в проверке компьютерной программы

3) Выполнять прогоны упрощенной программы с разными входными данными, а затем сравнить результаты с характеристиками, рассчитанными аналитически.

4) С помощью трассировки (каждая ветвь программы выдает сигнал при прохождении) желательно оценить правильность выполнения всех ветвей программы.

5) Полезно бывает посмотреть анимацию моделирования, чтобы выявить ошибки, которые были скрыты в числовых значениях.

6) Следует проверять правильность моделируемых в программе случайных чисел и их законов распределения, высчитывая среднее, дисперсию, или прибегая к более сложным критериям.

7) При использовании пакетов имитационного моделирования следует применять их с осторожностью, т.к., как правило, реализация некоторых операторов скрыта и не описана в документации.

### **24. Адекватность модели, схема сравнения выходных данных.**

**Адекватность модели**

Модель называется адекватной, если она является точным представлением реальной системы для конкретных целей исследования.

**Схема сравнения выходных данных** *(возможно текст лишний, но на всякий случай, у Кораблева это есть)*

* если система не существует, но есть возможность создать опытный образец, то можно сперва провести несколько экспериментов на опытном образце, а затем на имитационной модели, и в случае эквивалентности результатов в дальнейшем проводить эксперименты только на модели.
* стоит учесть, что классические статистические критерии могут быть не справедливы, так как выходные данные часто являются нестационарными и автокоррелированными;
* можно предложить на изучения специалистам собранные вперемешку реальные данные и данные моделирования, если специалисты не смогут отличить данные одни от других, то можно признать их эквивалентными (тест Тьюринга);
* если спустя время сбывается прогноз, полученный по результатам моделирования, то это в очередной раз показывает адекватность модели.
* анимация выходных данных поможет выявить ошибочные предположения, которые трудно разглядеть в цифровых данных;

При сравнении выходных характеристик реальной системы и модели необходимо запускать прогон модели не с новыми сгенерированными данными, а с реальными входными данными системы за прошлое время (метод общих случайных чисел).



### **25. Планирование и проведение экспериментов на имитационных моделях, виды факторов и откликов.**

После того как создана модель и установлена ее адекватность, можно переходить к проведению эксперимента. Вместо того, чтобы выполнять прогоны наугад, намного эффективнее распланировать проведение экспериментов, определить какие именно конфигурации системы следует моделировать, чтобы получить исчерпывающую информацию при наименьшем объеме моделирования.

**Факторы** – входные переменные или структурные допущения, принятые в модели.

**Отклики** – выходные показатели и характеристики работы системы.

Факторы делятся на:

· по характеру допущений

­ -качественные

­ -количественные

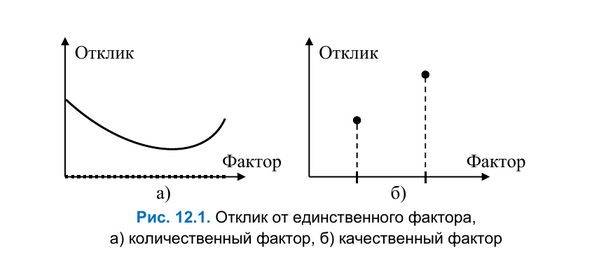
· по возможности влияния руководства

­ -управляемые

­ -неуправляемые

В зависимости от условий некоторые факторы могут быть как управляемые, так и неуправляемые.

В случае, если изменяемые фактор является единственным *(правый график)*, то можно построить график функции отклика от этого фактора. Участки, где отклик ведет себя нелинейно, следует моделировать более подробно. Однако если фактор является качественным *(левый график)*, то вместо графика функции можно получить лишь несколько точек отклика. (соединять их прямой неправильно, так как фактор не может изменяться непрерывно).



Так как выходные характеристики моделирования являются случайными числами, то следует выполнять несколько прогонов с разными случайными числами, чтобы получить средние значения и дисперсии этих характеристик.

### **26. Планирование и проведение экспериментов на имитационных моделях, поверхности откликов.**

Отклики - выходные показатели и характеристики работы системы.

Отклик от единственного фактора

Можно построить график функции отклика от этого фактора. С каким шагом изменять значение фактора предстоит выбрать разработчику, исходя из длительности прогонов и располагаемого времени. Участки, где отклик ведет себя нелинейно, следует моделировать более подробно.



Отклик от множества факторов

\*В случае если у нас имеется k количественных факторов, то отклик будет представлять функцию на k мерном пространстве.

\*Для случая k=2 график представляет рельеф поверхности.

\*Для случая k = 3 график может быть представлен как плотность вещества в объеме.

\*При большем количестве факторов график уже нарисовать не получится.

Так как выходные характеристики являются случайными числами, то следует выполнять несколько независимых прогонов. Сколько делать независимых прогонов при каждом значении фактора зависит от дисперсии выходных характеристик, требуемой точности измерений и от располагаемых ресурсов средств и времени.

### **27. Планирование и проведение экспериментов на имитационных моделях, факторный план 2k.**

Вместо того чтобы выполнять прогоны наугад, намного эффективнее спланировать проведение экспериментов, определить какие именно конфигурации системы следует смоделировать, чтобы получить исчерпывающую информацию при наименьшем объеме моделирования.

Выходом при большом числе факторов является уменьшение количества проверяемых значений по каждому фактору. Тогда минимальное количество значений, которое можно брать, будет равно двум и план эксперимента тогда называется - Факторный план **2k**.

• Каждый фактор выбирается всего на двух уровнях. Один уровень обозначим «−», а второй «+».

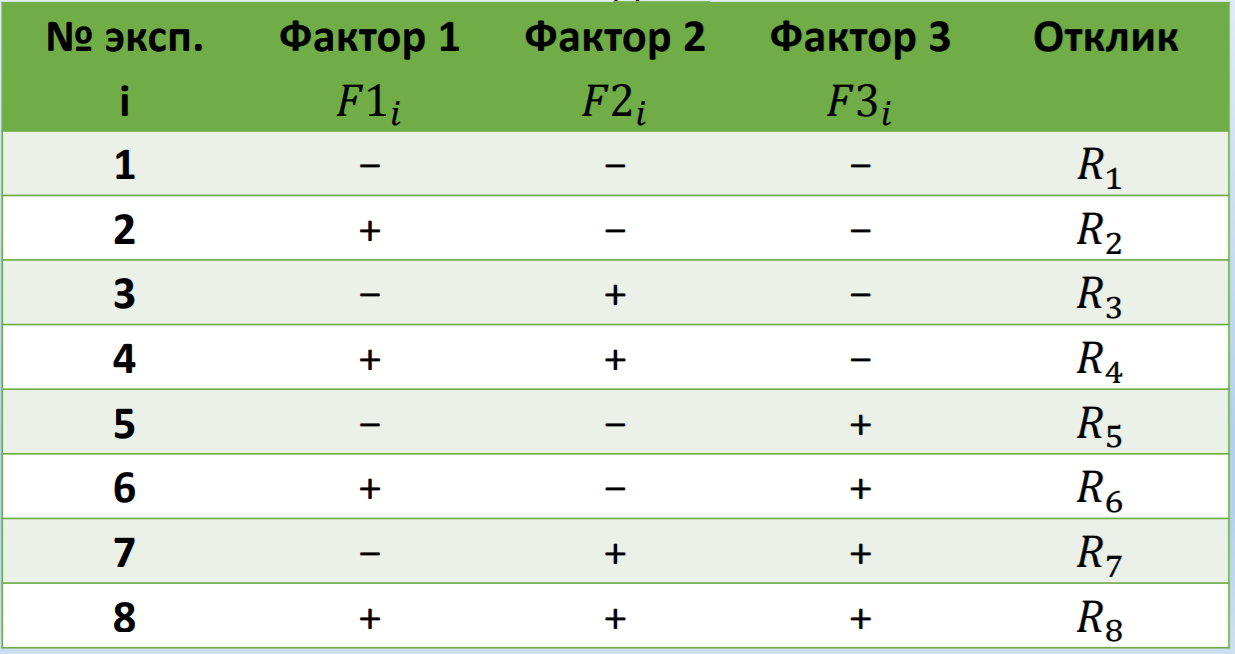
- во избежание путаницы лучше брать меньшие значения для уровня «−» и большие значения для фактора «+»;

- уровни «+» и «−» должны представлять собой противоположные по смыслу значения;

- не стоит брать значения уровней на очень большом нереалистичном расстоянии друг от друга

**Факторный план 2k представляет собой перечисление всех возможных комбинаций уровней факторов**

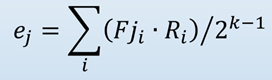
*Пример факторного плана* ***2k*** *для трех факторов(не обязательно, но пусть будет)*

**

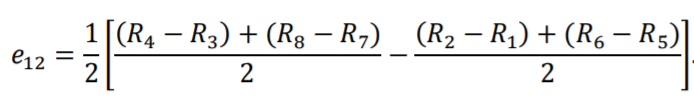
### **28. Планирование и проведение экспериментов на имитационных моделях, главные эффекты факторов.**

После того как создана модель и установлена ее адекватность, можно переходить к проведению эксперимента. Вместо того, чтобы выполнять прогоны наугад, намного эффективнее распланировать проведение экспериментов, определить какие именно конфигурации системы следует моделировать, чтобы получить исчерпывающую информацию при наименьшем объеме моделирования.

Главные эффекты факторов показывают, на сколько в среднем изменяется отклик при изменении фактора с уровня «−» на «+».



где Fji - знак уровня j стоящий в эксперименте i.

эффект взаимодействия первого и второго фактора можно вычислить 

### **29. Планирование и проведение экспериментов на имитационных моделях, эффекты взаимодействия факторов.**

Планирование экспериментов

Вместо того чтобы выполнять прогоны наугад, намного эффективнее распланировать проведение экспериментов, определить какие именно конфигурации системы следует моделировать, чтобы получить исчерпывающую информацию при наименьшем объеме моделирования.

Эффект взаимодействия факторов

Эффект взаимодействия двух факторов показывает насколько изменяется главный эффект фактора (половина) при изменении взаимодействующего фактора с уровня «−» на «+». = ()/

• Например, эффект взаимодействия первого и второго фактора можно вычислить следующим образом: =

Что представляет собой разницу главных эффектов первого фактора, когда второй фактор поменял уровень с «−» на «+» (половину разницы).

• Эффекты взаимодействия факторов являются симметричными характеристиками.

### **30. Планирование и проведение экспериментов на имитационных моделях, факторный план с дробными репликами.**

Даже при использовании факторного плана 2 при большом количестве факторов для проведения всех экспериментов может потребоваться неприемлемое количество времени. Факторный план 𝟐𝒌−𝒑 с дробными репликами может существенно уменьшить эту проблему при оценке главных эффектов факторов и эффектов взаимодействия. экспериментов в 2𝑝 раз, что очень существенно сократит общее время, необходимое на проведение экспериментов.

Например, для k=11 факторов можно составить факторный план 211−6 следующим образом. Первые 5 факторов принимают все возможные комбинации (всего 32 комбинации), знак уровня для оставшихся 6 факторов определяется как произведения некоторых первых 5 факторов.

### **31. Планирование и проведение экспериментов на имитационных моделях, сверхнасыщенные планы.**

**Планирование экспериментов**

Вместо того чтобы выполнять прогоны наугад, намного эффективнее распланировать проведение экспериментов, определить какие именно конфигурации системы следует моделировать, чтобы получить исчерпывающую информацию при наименьшем объеме моделирования.

**Сверхнасыщенные планы**

• Когда факторов очень много (больше 100), то даже факторные планы с дробными репликами будут требовать проведения неприемлемого количества экспериментов. Иногда количество экспериментов может быть строго ограничено (вне зависимости от числа факторов) и быть меньше чем количество самих факторов. Тогда экспериментирование придется осуществлять со сверхнасыщенным планом.

• В сверхнасыщенном плане столбец каждого фактора должен состоять на половину из уровней "+" и на половину из "−". Выбор, в каком из экспериментов брать уровень "+" или "−", можно делать случайно, в этом случае план экспериментов называется случайно уравновешенным планом, а можно систематически (систематический сверхнасыщенный план), так чтобы смешивание эффектов факторов стремилось к минимуму.

### **32. Планирование и проведение экспериментов на имитационных моделях, отсеивание факторов.**

После того как создана модель и установлена ее адекватность, можно переходить к проведению эксперимента. Вместо того, чтобы выполнять прогоны наугад, намного эффективнее распланировать проведение экспериментов, определить какие именно конфигурации системы следует моделировать, чтобы получить исчерпывающую информацию при наименьшем объеме моделирования.

Когда фактором очень много, будет требоваться проведение неприемлемого количества экспериментов. Иногда количество экспериментов может быть строго ограничено и быть меньше чем количество самих факторов. Тогда экспериментирование придется осуществлять со сверхнасыщенным планом.

Одним из подходов к уменьшению количества экспериментов является группировка факторов, когда группа факторов объединяется и рассматривается как один фактор. Можно группировать факторы по общему характеру, а можно объединять разные по характеру факторы, относящиеся к одному участку системы.

### **33. Планирование и проведение экспериментов на имитационных моделях, метамодели.**

Вместо того чтобы выполнять прогоны наугад, намного эффективнее распланировать проведение экспериментов, определить какие именно конфигурации системы следует моделировать, чтобы получить исчерпывающую информацию при наименьшем объеме моделирования.

Метамодель – это попытка заменить результаты моделирования алгебраической (математической) моделью (формулой).

В этом случае метамодель является оцененной функцией отклика от многих переменных. Метамодель будет являться приближенным результатом моделирования, когда для того чтобы получить значение отклика, достаточно будет подставить входные характеристики в некоторую формулу, а не проводить эксперимент. Возможность такой замены интересна когда моделирование приходится осуществлять для очень большой сложной системы, когда результат хочется получить быстро и нет времени на проведение очередного эксперимента.

### **34. Сравнение альтернативных конфигураций систем, Метод общих случайных чисел.**

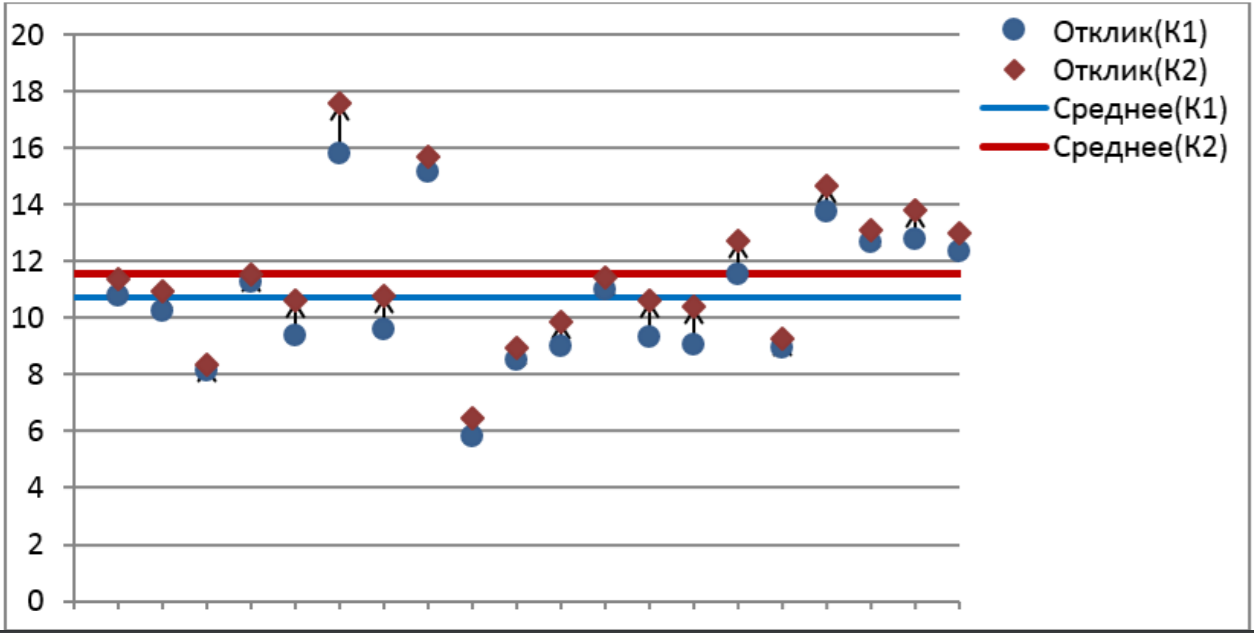
Вместо того чтобы выполнять прогоны наугад, намного эффективнее распланировать проведение экспериментов, определить какие именно конфигурации системы следует моделировать, чтобы получить исчерпывающую информацию при наименьшем объеме моделирования.

Идея метода - заключается в том, что разные конфигурации систем моделируются при одних и тех же наборах случайных чисел.

• Если в одно системе для генерации интервалов между поступлениями используется один поток случайных чисел, то для другой системы нужно использовать точно такой же поток случайных чисел при генерации интервалов между поступлениями.

• Точно также одинаковые потоки надо задавать для времени обслуживания, времени до поломки и т.д

• Различия между системами проявляются более явно, как правило с первых прогонов, и они не изменяют знак в других прогонах с новыми согласованными наборами случайных чисел (при других условиях).



### **35. Сравнение альтернативных конфигураций систем, проблема синхронизации случайных чисел.**

Первая проблема, с которой предстоит справиться, это необходимость задавать свой уникальный поток случайных чисел для каждого элемента системы, который использует случайные числа.

Другой проблемой может быть использование другого закона распределения или другого алгоритма преобразования случайных чисел в одном из элементов, использующих случайные числа, при сравнении двух разных систем.

Другая проблема, это ограниченность случайных чисел в каждом из потоков. При длительном моделировании случайные числа могут закончиться в некоторых потоках, следует предусмотреть достаточное расстояние между начальными позициями таких потоков, чтобы потоки не пересекались. Простым выходом из ситуации является следующее, вместо потоков 1,2 и 3, использовать потоки 1,3 и 5, т.е. используются потоки удвоенной длины.

### **36. Сравнение альтернативных конфигураций систем, доверительные интервалы при сравнении альтернативных систем (на основе критерия Стьюдента).**

Данные необходимо объединить в пары и вычислить разницу = − , для этого стоит отбросить те наблюдения, выборка которых оказалась больше, или провести серию дополнительных экспериментов, в результате чего объем выборок будет одинаков = = 𝑛.

E()=; D()=

Доверительный интервал разницы откликов

где - критическое значение распределения Стьюдента для 𝑛 − 1 степеней свободы и доверительной вероятности 90, 95 или 99 %.

Доверительный интервал для математического ожидания

разница лишь в том, что дисперсия для математического ожидания в 𝑛 раз меньше дисперсии самой случайной величины

D()=D=

### **37. Сравнение альтернативных конфигураций систем, доверительные интервалы Велча при сравнении альтернативных систем (предназначение).**

Этот доверительный интервал используется, когда размеры выборок не равны, но хочется использовать полностью набор данных. Вместо объединения в пары вычисляются математические ожидания и дисперсии каждой и характеристик. Затем вычисляется n\* оценка степеней свободы.   
Доверительный интервал Велча также можно использовать при проверке адекватности модели, когда имеется небольшая выборка реальных данных и намного больше выборка данных полученных с помощью моделирования. Если интервал разницы реальных данных и данных моделирования будет содержать ноль, то можно признать модель адекватной.

### **38. Сравнение альтернативных конфигураций систем, выбор лучшей из множества конфигураций систем.**

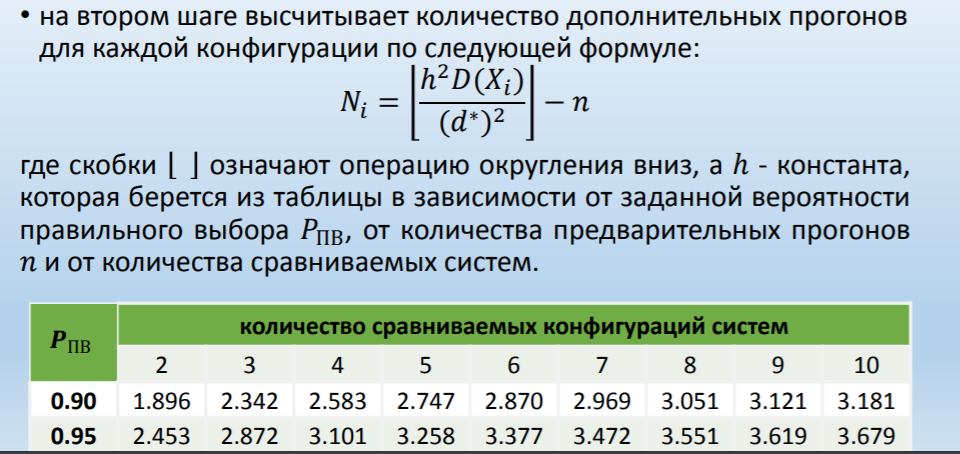
**Сравнение альтернативных конфигураций систем**

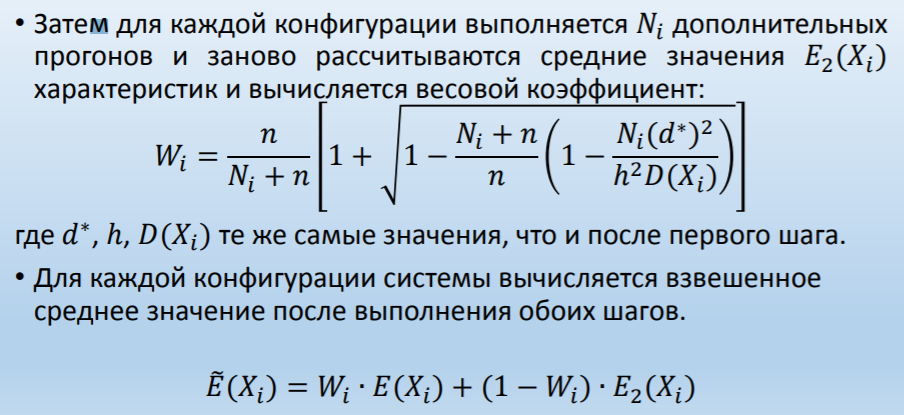
Каждый раз, когда выполняется моделирование какой-нибудь системы с использованием случайных чисел, выходные характеристики будут отличаться от прогона к прогону даже при неизменной конфигурации системы.*Как тогда сравнивать разные конфигурации системы?* Выходом будет многократное повторение эксперимента, расчет средних значений откликов и сравнение этих средних значений.

**Выбор лучшей из множества конфигураций**

Для выбора лучшей из множества конфигураций систем используется двухшаговая процедура(разработанная Дудевичем и Делалом в 1975 году), которая гарантирует заданную вероятность правильного выбора лучшей системы 𝑃пв, при заданной величине безразличия 𝑑∗ отклика (если разница меньше, то сравниваемые системы считаются эквивалентными).

1) На первом шаге выполняется некоторое число предварительных прогонов каждой из систем, например 𝑛 = 20. Цель первого шага получить предварительные оценки средних значений 𝐸(𝑋𝑖) и дисперсий 𝐷(𝑋𝑖) наблюдаемых откликов, при имеющемся наборе рассчитанных оценок 𝐸(𝑋𝑖) , 𝐷(𝑋𝑖) .

2) 



Среди всех конфигураций систем выбирается система с лучшим (наибольшим или наименьшим) значением 𝐸�(𝑋𝑖) характеристики. Вероятность того, что этот выбор был правильный, будет не меньше чем заданное значение правильного выбора 𝑃пв.

### **39. Метод Монте-Карло**

В статических моделях в отличии от динамических не происходит никаких изменений со временем или не предполагается само изменение времени. Одним из ярких примеров статических моделей является метод Монте-Карло.

Главная идея метода состоит в том, что вместо того, чтобы аналитически искать значение некоторой интересующей характеристики, подбирается такая случайная величина, математическое ожидание которой совпадает с искомым значением, после чего проводится большое количество экспериментов, в результате которых можно получить оценку этого математического ожидания. Из-за большого числа повторений случайного процесса метод Монте-Карло также называют методом статистических испытаний, а модели называют стохастическими.

*На всякий случай:*

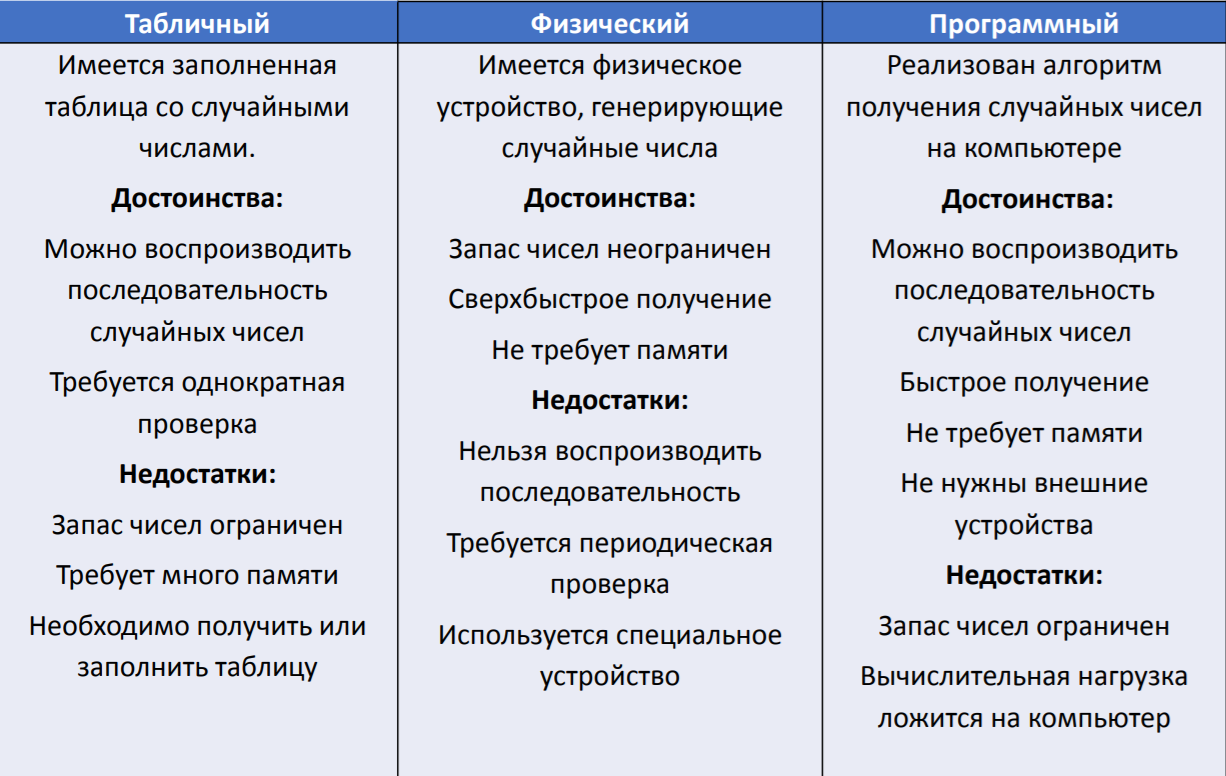
Широкое применение метод М-К получил при вычислении многократных интегралов. Метод М-К предполагает заключение многомерной фигуры в многомерный прямоугольник, генерации большого числа точек со случайными координатами, подсчета доли точек попавшие в область многомерной фигуры.

Метод М-К часто используется для определения квантилей (квантиль – значение, меньше которого находится заданная доля всех случайных чисел) сложных распределений у которых нет простого способа получить функцию со значениями квантилей.

### **40. Законы распределения, наиболее распространенные в практике статистических исследований.**

### В наших моделях могут быть использованы случайные величины для обозначения таких характеристик как время ожидания, время обслуживания, спрос на продукцию, доход за день, число сбоев и так далее. Также некоторые законы распределения используются в процессе статистической обработке результатов при построении доверительных интервалов. 1) Равномерное распределение (uniform, прямоугольное) 2) Экспоненциальное распределение (exponential, показательное) 3)Распределение Эрланга 4) Распределение Вейбулла 5) Нормальное распределение (normal, Гаусса, Gaussian) 6) Треугольное распределение (triangle).

### **41. Особенности способов получения случайных чисел: табличный, физический и программный способ.**



### **42. Генерирование псевдослучайных чисел. Метод серединных квадратов. Недостатки.**

Возьмем начальное четырехзначное число, например год рождения студентов 𝑍0=1996. Возведем его в квадрат 𝑍02=03𝟗𝟗𝟗𝟎16 (дополняя нулями слева, чтобы получилось восьмизначное число) и извлечем четырехзначное число из середины 𝑍1=9720,которое будет опять возводится в квадрат и т.д. Чтобы полученные случайные числа были от 0 до 1, разделим каждое число на 10000 (переместим запятую слева от числа)  
Недостатки метода: Последовательность чисел имеет тенденцию стремиться к нулю.

### **43. Линейный конгруэнтный генератор. Теорема о трех условиях для того, чтобы генератор обладал полным периодом.**

Получение нового числа происходит по рекурсивной формуле:

где 𝐴 множитель, 𝐶 - приращение, 𝑚 – модуль, все целые положительные числа. Чтобы получить случайное число:

=/m

Теорема о полном периоде (Халл и Добеллом ): Линейный конгруэнтный генератор имеет полный период тогда и только тогда, когда выполняются следующие условия: • 1) параметры 𝐶 и 𝑚 являются взаимно простыми (наибольший общий делитель НОД(𝐶,𝑚)=1); • 2) число 𝐴 − 1 должно делится на все простые числа, из произведения которых состоит модуль 𝑚; • 3) если модуль 𝑚 делится на 4, то 𝐴 − 1 тоже должно делиться на 4.

### **44. Конгруэнтный генератор с простым модулем. Механизм избежать явного деления.**

Генераторы с простым модулем:

• У таких генераторов параметр 𝐶 = 0.

• Теорема о полном периоде не выполняется (период < 𝑚).

• Если брать модуль 𝑚 = 231, то период составит максимум лишь одну четвертую 231−2.

• Однако если брать простой ,то период составит 𝑚 − 1. Например, самое первое простое число, которое меньше 231, является число 231 − 1 , и период генератора с таким модулем составит 231 − 2.

• Оказывается, что статистические свойства генераторов с полным периодом не превосходят свойств генераторов с простым модулем, которые более просты и понятны.

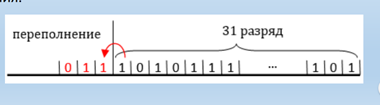
• Множитель 𝐴 должен быть выбран таким, чтобы число 𝐴𝑚−1 делилось на модуль 𝑚 (𝐴 первообразный элемент по модулю 𝑚)

Механизм избежать явного деления по модулю:

• Избежать деления по модулю помогает такое явление как переполнение разрядной сетки.

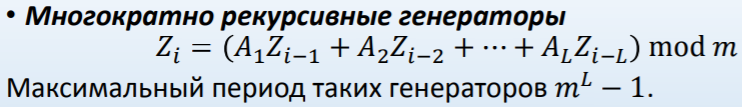
• Например, если в машинном слове 32 бита, и при этом 1 бит отводится под знак, то максимально возможный модуль 𝑚 = 231.

• Все биты, которые не влезут в разрядную сетку, будут отброшены, а то что останется и будет представлять собой результат остатка от деления.

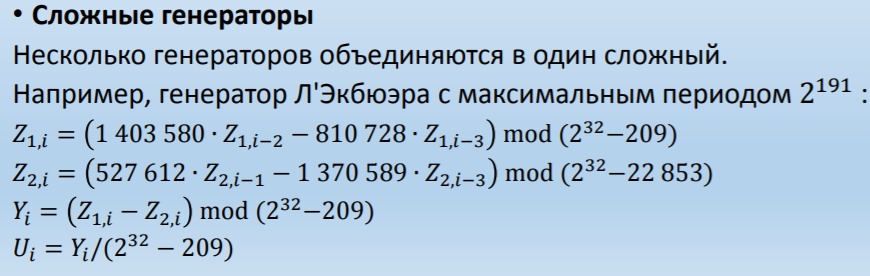
****

### **45. Многократные рекурсивные генераторы и сложные генераторы.**

**Многократный рекурсивный генератор** - генератор, в котором последующее число определяется через L предыдущих чисел.



Хороший **сложный генератор** можно получить, объединив два многократно рекурсивных генератора в один.



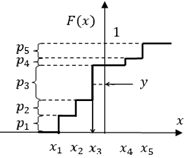
В данном примере сначала получаются 2 случайных числа разными генераторами, а третий выдает уже случайное число, опираясь на первые 2.

### **46. Моделирование законов распределения. Метод обратной функции. Получение дискретных распределений методом обратной функции.**

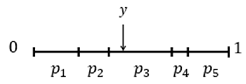
Метод обратного преобразования деформирует равномерное распределение в соответствии с законом распределения, который был выбран.

С помощью обратного преобразования можно получить случайные величины с дискретным распределением.

Функция распределения имеет вид ступенчатой функции.



Обратное преобразование сводится к тому, чтобы определить на какой интервал попала равномерно распределенная случайная величина 𝑦.



**Недостатки обратного преобразования:**

-Не для всех распределений можно выразить обратную функцию 𝑥=𝐹-1(𝑦).

- Для дискретных распределений с бесконечной областью определения требуются особые условия остановки.

- Метод обратного преобразования может быть не самым быстрым способом получения заданного закона распределения.

### **47. Недостатки и достоинства метода обратной функции. Генерирование усеченного распределения с помощью обратной функции.**

Недостатки обратного преобразования:

1. Не для всех распределений можно выразить обратную функцию x=F-1(y).

2. Для дискретных распределений с бесконечной областью определения требуются особые условия остановки.

3. Метод обратного преобразования может быть не самым быстрым способом получения заданного закона распределения.

Достоинства метода обратного преобразования:

1. Требуется только одно случайное число, в то время как другим методам могут потребоваться несколько.

2. Метод обратного преобразования может использоваться для понижения дисперсии в различиях между сравниваемыми системами, в которых используются разные законы распределения.

3. Если необходимо ограничить область получаемых случайных величин, то обратное преобразования позволяет сделать это очень просто, не меняя вида обратной функции.

Получение усеченного распределения. Например, если надо получить случайные числа 𝑥, ограниченные интервалом от 𝑎 до 𝑏, то для этого надо сделать следующее:

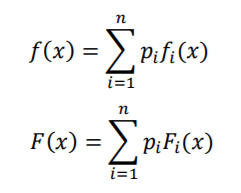
• 1) Генерируем 𝑦 как равномерное от 0 до 1.

• 2) Рассчитываем случайную величину *v=F(a)+[F(b)-F(a)]·y*.

• 3) Возвращаем случайную величину *x=F-1(v).*

### **48. Метод композиции для генерирования сложных законов распределения.**

Предназначен для моделирования таких сложных распределений, которые можно представить в виде выпуклой линейной комбинаций других более простых распределений, способы моделирования которых предполагаются нам известными.

где 𝑝𝑖 доли более простых распределений, причем 𝑝𝑖𝜖[0,1] и ∑ 𝑝𝑖 = 1. 

Моделировать такие распределения можно с помощью следующего подхода:

1. Генерируем равномерную случайную величину 𝑈𝜖[0,1].

2. С помощью случайной величины 𝑈 по вероятностям 𝑝𝑖 выбирается номер распределения 𝑖 (где 𝑝𝑖 показывает вероятность того, что будет выбрано распределение 𝑖).

3. Возвращается случайная величина 𝑥 в соответствии с законом распределения 𝐹𝑖(𝑥).

### **49. Метод принятия-отклонения для получения различных законов распределения.**

Метод принятия отклонения принципиально способен моделировать абсолютно произвольный закон распределения. Его можно назвать универсальным методом. Однако он также имеет недостатки, в основном связанные с производительностью и быстродействием.

(на всякий)

Алгоритм:  
1) Выбрать такую функцию ℎ(𝑥), которая бы ограничивала сверху заданную функцию 𝑓(𝑥) => ℎ(𝑥) ≥ 𝑓(𝑥).   
2) Вычисляем площадь под функцией 𝐻 = ∫ ℎ(𝑥)𝑑𝑥 𝑥𝑚𝑎𝑥 𝑥𝑚𝑖𝑛 . Теперь определим функцию, площадь под которой равна 1, 𝑟(𝑥) = ℎ(𝑥)/𝐻.   
3) Генерируем случайную величину 𝑋 с плотность распределения 𝑟(𝑥) (из-за этого следует подбирать такую верхнюю огибающую, в соответствии с которой мы можем генерировать случайные числа).   
4) Генерируем равномерную величину 𝑈𝜖[0,1] . Рассчитываем случайную величину 𝑌 = 𝑈 ∙ ℎ(𝑥).   
5) Если 𝑌 ≤ 𝑓(𝑋), то возвращаем полученную на шаге 3 случайную величину 𝑋, иначе повторяем шаги 3-5. Другими словами, если точка (𝑋, 𝑌) попала под график функции 𝑓(𝑥) , то точка принимается, иначе отбрасывается

### **50. Специальные свойства и метод свертки для генерирования случайных величин.**

Иногда можно представить случайную величину в виде некоторой связи других случайных величин, которых проще генерировать.

Общей формы и алгоритма для таких методов нет, все зависит от требуемого закона распределения.

В том случае, когда случайную величину можно выразить как сумму других случайных величин, этот метод называется методом свертки.

Метод свертки

Пример: распределение Эрланга порядка 𝒎 представляет собой сумму 𝑚 независимых случайных распределений с одинаковым экспоненциальным распределением. Тогда случайную величину 𝑋 можно получить следующим образом.

• 1) Генерируем ,,, … , как равномерные случайные величины.

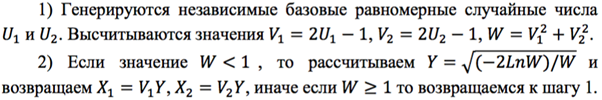
• 2) Получаем случайные величины с экспоненциальным законом распределения

• 3) Возвращаем случайную величину 𝑋 =или X=-

### **51. Генерирование нормального закона распределения.**

Для генерирования нормального распределения наиболее предпочтительный метод полярных координат.

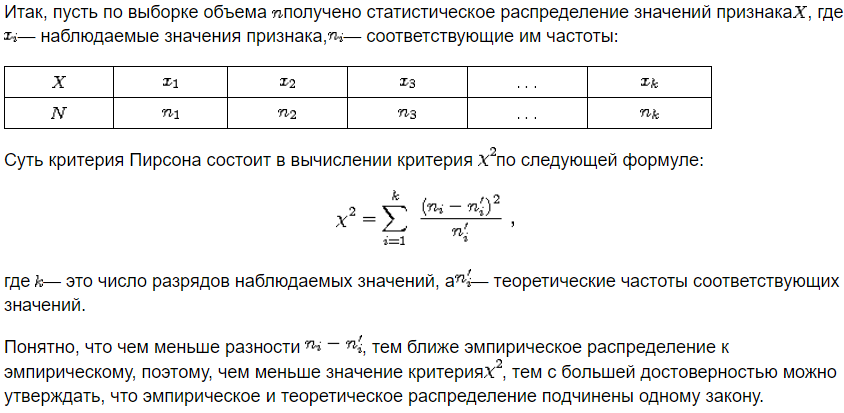
Метод полярных координат состоит из следующих шагов.



Вероятность того, что придется повторять действия шага 1, равняется значению 1 − 𝜋/4.

### **52. Критерий Пирсона проверки соответствия статистических данных заданному теоретическому распределению.**

Критерий согласия Пирсона используется для проверки соответствия располагаемой выборки случайных величин предполагаемому теоретическому закон распределения.



### **53. Критерий Колмогорова проверки соответствия статистических данных заданному теоретическому распределению.**

Критерий Колмогорова предполагает построение эмпирической функции распределения и сравнение ее с теоретической.

С помощью критериев Колмогорова можно проверять выборку случайных чисел, полученную с помощью генератора случайных чисел на соответствие равномерному закону распределения, проверять справедливость алгоритмов получения всевозможных законом распределения, определять закон распределения экспериментальных данных.

На всякий случай:

Имеется набор измерений случайной величины x1, x2,..,xn. Чтобы построить эмпирическую функцию распределения следует сперва упорядочить эту выборку по возрастанию, а затем для самого минимального наблюдения предполагаем Fэмп(xmin)=0, в каждой последующей точке функция распределения увеличивается на 1/(n-1). Затем рассчитываем значение K=D\*sqrt(n). Для распределения Колмогорова доля α всех наблюдений лежит левее значения , формула справедлива для α близких к единице. Если действительно выполняется K<Kкр α, то тогда различие между эмпирической и теоретической функцией распределения носит случайный характер. Если же неравенство не выполняется, тогда считаем, что разница обусловлена не случайным, а систематическим расхождением с теоретическим законом распределения.

### **54. Тестирование генераторов случайных чисел на равномерность заполнения многомерного пространства.**

В зависимости от параметров генератора количество гиперплоскостей и расстояние между ними может изменится. Если существуют большие пустые области, то моделирование пары координат с помощью одного такого генератора может привести к тому, что некоторые координаты никогда не будут получены.

Большое расстояние между гиперплоскостями нежелательно, следует проверять генератор случайных чисел по критерию серий, который не что иное, как обобщение критерия Пирсона на большие размерности. Каждую ось [0,1] надо разбить на 𝑘 интервалов, тогда весь гиперкуб разобьется на 𝑘𝑑 областей.

### **55. Тестирование генераторов случайных чисел на независимость случайных величин.**

Под требованием о независимости случайных величин будем понимать линейную независимость, то есть корреляция двух случайных чисел равна нулю 𝑝𝐿 = 𝐶𝑜𝑟(𝑈𝑖, 𝑈𝑖+𝐿 ) = 0 для разных 𝐿.

(Этот абзац на всякий случай) Корреляцию (выборочную) двух случайных величин разнесенных на 𝐿 позиций рассчитывают по формуле: , где математические ожидания одной и той же выборки.

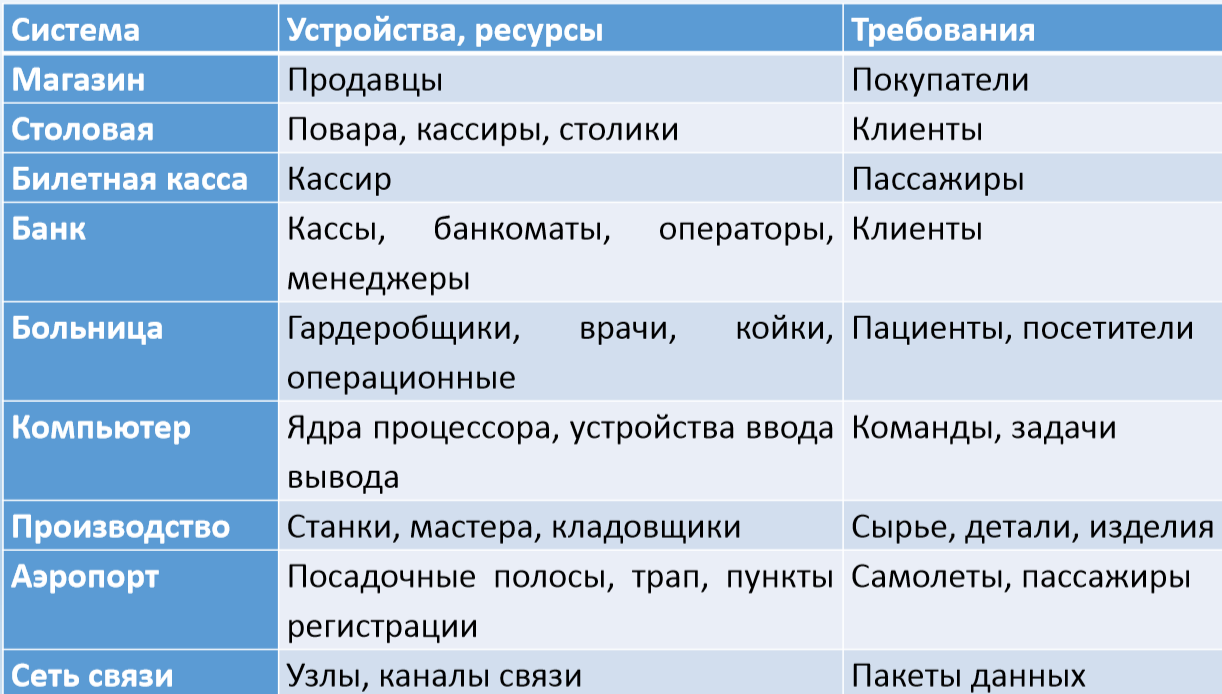
Для равномерного распределения от 0 до 1 дисперсию значения корреляции можно приближенно оценить по формуле: 

Тогда для проверки гипотезы равенства нулю корреляции 𝑝𝐿 = 0 формируем дробь 𝑡, которое должно иметь распределение Стьюдента со степенями свободы 𝑛 - 

Если выполняется −𝑡кр ≤ 𝑡 ≤ 𝑡кр для вероятности 95% (или 99%), то можем принять то, что числа 𝑈𝑖 , 𝑈𝑖+𝐿 независимы

### **56. Системы массового обслуживания. Примеры. Из каких характерных частей состоит система массового обслуживания.**

пример:



Для моделирования систем массового обслуживания необходимо задать и определить следующие основные части СМО:

1) входящий поток требований, поступающий на обслуживание

2) способ организации очереди

3) правила обслуживания требований

4) выходящий поток требований

5) режимы работы системы

### **57. Основные критерии (характеристики) оценки работы системы массового обслуживания.**

Основные критерии оценки работы системы массового обслуживания 1) Среднее время ожидания в очереди 2) Средняя длина очереди 3) Коэффициент загрузки устройства

Также иногда важно определить максимальное время ожидания в очереди и максимальный размер очереди.

Среднее время ожидание в очереди

• Среднее время ожидание в очереди определяется через множество задержек каждого требования в системе, перед тем как поступить на обслуживание.

=

где 𝑛- количество прошедших через систему требований, - время пребывания в очереди. • Если - случайные числа, то также будет случайным числом, со своим разбросом. Значение изменяется от прогона к прогону.

Средняя длина очереди

Среднее количество требований в очереди на протяжении всего моделирования.

=q\*P(q(t)=q)

где 𝑞 𝑡 - размер очереди в момент времени 𝑡, 𝑃(𝑞 𝑡 = 𝑞) - вероятность или доля времени, когда размер очереди равен 𝑞. Так как 𝑃(𝑞 𝑡 = 𝑞) неизвестно, используют другую формулу.

=

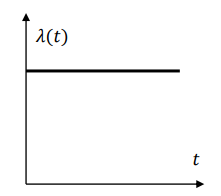
где - суммарное время когда размер очереди равняется 𝑞, 𝑇 – общее время моделирования.

Коэффициент загрузки устройства 𝐾 • Коэффициент загрузки устройства 𝑲 показывает долю времени, когда устройство занято обслуживанием требований ко всему времени моделирования.

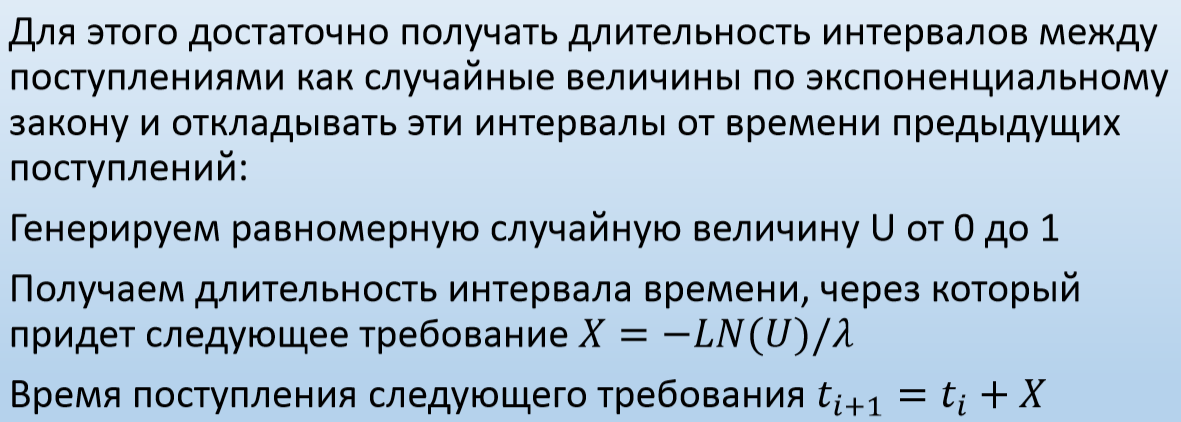
К==1- где и суммарное время когда устройство было занято и свободно.

### **58. Стационарный Пуассоновский процесс. Генерирование стационарного Пуассоновского процесса поступления требований.**

Стационарным Пуассоновским процессом называется такой процесс поступления, в котором интенсивность 𝜆 постоянна со временем. Интенсивность 𝜆(𝑡)=𝜆=const



Генерирование:

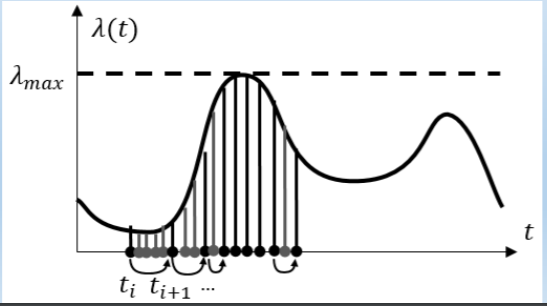


### **59. Нестационарный Пуассоновский процесс. Генерирование не стационарного пуассоновского процесса поступления.**

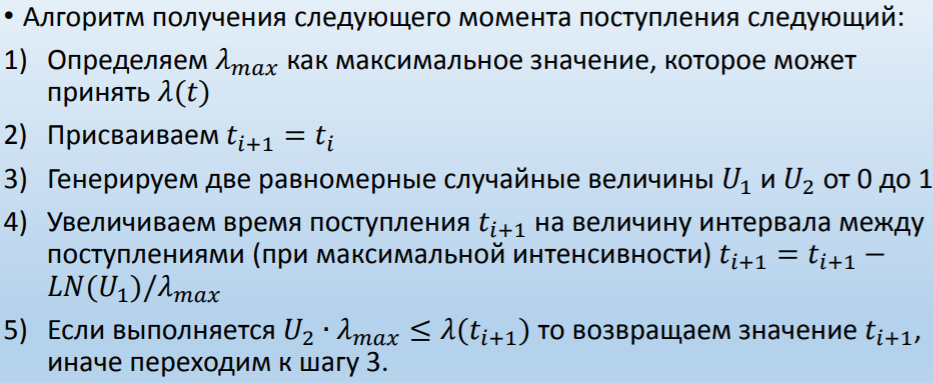
**Нестационарным Пуассоновским процессом** тогда называется такой процесс, у которого средняя интенсивность поступления представлена в виде некоторой функциональной зависимости от времени 𝜆(𝑡), где 𝜆 𝑡 - функция от времени. Например, есть разница в интенсивности приходов людей в парикмахерскую,( в час пик и и часы затишья).

**Моделирование нестационарного процесса методом прореживания:**

Основная идея заключается в том, чтобы моделировать интервалы поступления с максимальной постоянной интенсивностью, но прежде чем возвращать следующий момент поступления требования, пропускать некоторые моменты с определенной вероятностью.

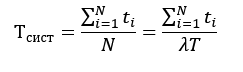


*Не уверена, что это нужно, но на всякий случай:*

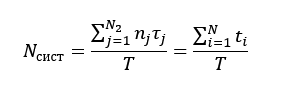
**

### **60. Формулы Литтла соотношения длины очереди и времени пребывания в очереди, числа требований в системе и времени пребывания в системе.**

Среднее время пребывания в системе:



Среднее количество требований в системе:



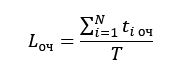
Первая формула Литтла (соотношение средней длины очереди и среднего времени пребывания в очереди):

Nсист=λTсист

Среднее время, проведенное в очереди:



Средняя длина очереди:



Вторая формула Литтла (соотношение средней длины очереди и среднего времени пребывания в очереди):

Lоч=λTоч

Связь между первой и второй формулой Литтла образуется из того, что время пребывания в системе состоит из суммы времени, проведенном в очереди и временем обслуживания:

Тсист=Точ+Тобслуж=Точ+1/μ

### **61. Аналитическое решение систем массового обслуживания. Правило составления уравнений Колмогорова. Особенности аналитического решения.**

Рассмотрим систему массового обслуживания с тремя устройствами обслуживания. Будем различать состояния системы по количеству требований. Состояние 𝑆0 будет означать пустую систему. Состояние 𝑆1 означать, что в системе всего одно требование. Состояние 𝑆𝑛 означает, что в системе n требований. Во время функционирования системы переход между состояниями происходит в зависимости от интенсивности поступления требований λ и от интенсивности обслуживания μ.

Введем вероятности 𝑃𝑖 того, что система находится в одном из возможных состояний (сумма вероятностей равна единице). Тогда для установившегося режима, т.е. при большом времени моделирования, для каждого состояния можно записать уравнение Колмогорова, которое представляет собой равенство произведения исходящих стрелок на вероятность текущего состояния, и произведения входящих стрелок на вероятность состояний, из которых они вышли.

Сформулируем правило составления уравнений Колмогорова. В левой части каждого из них стоит производная вероятности i-го состояния. В правой части — сумма произведений вероятностей всех состояний (из которых идут стрелки в данное состояние) на интенсивности соответствующих потоков событий, минус суммарная интенсивность всех потоков, выводящих систему из данного состояния, умноженная на вероятность данного (i-го состояния).

### **62. Непрерывные модели, способы решения.**

Название такого вида моделей говорит о том, что все процессы и время в модели должны протекать непрерывно, а не дискретно. Как правило непрерывные модели используются для демонстрации некоторых процессов, заданных системой дифференциальных уравнений.

Простейшие непрерывные модели (системы диф. уравнений) могу иметь аналитическое решение, полученное с помощью стандартных методов дифференциального исчисления, имеющее вид суммы многих гармонических затухающих колебаний. Более сложные непрерывные модели могут быть решены только методами численного интегрирования и как правило почти все непрерывные модели

Численные методы решения дифференциальных уравнений различаются по порядку точности. Наиболее популярные среди них это методы Рунге-Кутта от четвертого порядка и выше.

В непрерывных моделях должны быть заданы все начальные условия, например такие как начальные координаты 𝑥𝑖(0) , начальная скорость по каждой координате 𝑥̇𝑖(0), ускорение чаще всего задается функцией от текущих координат и скорости 𝑥̈𝑖(𝑡) = 𝑓𝑖(𝑥𝑖, 𝑥̇𝑖,𝑡).

### **63. Непрерывные модели, примеры.**

Одним из самых популярных примеров непрерывной модели, разбираемых в имитационном моделировании, является модель Хищник-Жертва. Эта модель показывает динамику популяции хищных животных и динамику популяции животных, являющихся жертвами для хищников, например будем рассматривать кроликов и волков. Обозначим численность кроликов как 𝑁1, а численность волков как 𝑁2.

Так как численность популяции не может быть бесконечной, то вводят ограничение на максимальную емкость ареала обитания 𝑀1, при достижении которой прирост прекращается, и в зависимости от количества волков, поедающих зайцев, прирост снижается.

𝑟1 - коэффициент рождаемости

𝑘1 - коэффициент смертности при встрече двух видов

Аналогично составляется уравнение для прироста численности хищного вида, лишь с одним отличием, что при встрече двух видов, прирост численности хищников увеличивается, а не уменьшается.

### **64. Модели системной динамики (Дж. Форрестера), примеры.**

Модели системной динамики можно отнести к непрерывным моделям, записываемых с помощью системы дифференциальных уравнений (как правило первой степени). • Модели состоят из: 1. Уровней (накопителей). 2. Потоков, обозначающих скорость изменения уровней. 3. Функций (от уровней), переключающих потоки или изменяющих их. 4. Линий задержки, для моделирования запаздывания во времени темпа потоков от значений уровней.

Модель Мир-2, Джей Форрестер 1974г.

• Основные переменные:

1. Население P

2. Основные фонды K

3. Доля Фондов в сельском хозяйстве X

4. Уровень загрязнения Z

5. Количество невозобновляемых природных ресурсов R

• Факторы (параметры)

1. 𝑃p – относительная численность населения к 1970г

2. 𝐾𝑝 – удельный капитал

3. 𝐶 – уровень жизни

4. 𝐹 – относительный уровень питания

5. – относительное загрязнение 6. – доля остающихся ресурсов

Система уравнений: 𝐵 = 𝐵(𝐶, 𝐹, 𝑃p, 𝑍S) и 𝐷 = 𝐷(𝐶, 𝐹, 𝑃p, 𝑍S) – темп рождаемости и смертности

Модель Мир-2, система уравнений

=𝑃(𝐵 − 𝐷)

=

=-

где 𝐵 = 𝐵(𝐶, 𝐹, , ) – темп рождаемости

D = D(𝐶, 𝐹, , ) – темп смертности

=(P,C)-скорость производства основных фондов

=(F,Q)– прирост сельскохозяйственных фондов

=(P,)– скорость загрязнения

=(P,C)-скорость потребления ресурсов

=40 лет – постоянная износа основных фондов

=15 лет – время выбытия доли сельскохозяйственных фондов

- характерное время естественного разложения

### **65. Агентно-ориентированные модели, классификация среды.**

В таких моделях в некоторой среде действует агент - разумный объект, действующий автономно, который пытается достичь своей определенной цели.

Среды могут отличатся по следующим параметрам: замкнутые и открытые, трансформируемые и не трансформируемые, детерминированные или стохастические. Открытые среды отличаются от замкнутых тем, что замкнутые могут быть полностью определены в начале моделировании, а открытые могут расширятся со временем, в них могут появляться новые агенты и уходить старые. Трансформируемые среды могут изменятся под действием агентов. Стохастические среды могут использовать случайные числа в процессе моделирования, когда в детерминированных все изменения происходят в соответствии с заданными условиями..

### **66. Агентно-ориентированные модели, классификация агентов.**

**Агентно-ориентированные модели.** В таких моделях в некоторой среде действует агент - разумный объект, действующий автономно, который пытается достичь своей определенной цели.   
Как правило агентные системы работают по шагам в дискретные моменты времени, когда агент осуществляет действие по очереди с другими агентами, а среда меняет свое состояние в дискретные моменты времени. Тем не менее агентные системы также могут быть непрерывными, с непрерывным изменением времени.   
Задача каждого агента - достижение целевого состояния (цели) посредством чередования актов восприятия среды, планирования (принятия) решения и осуществления выбранного действия.

**Классификация агентов**

Агенты могут отличатся по следующим признакам.

1. по степени информированности относительно состояния внешней среды

* агент полностью информирован о состоянии среды в каждый момент времени
* частично информирован благодаря некоторым сенсорам, осуществляющим восприятие среды;

2) по наличию памяти

* агент может помнить некоторую информацию о своих предыдущих действиях
* -//- о предыдущих изменениях среды;

3) по возможности прогнозирования будущих изменений среды

* агент может предсказывать дальнейшее развитие среды и действовать на опережение
* агент принимает решение как действовать исходя только из доступной информации на данный момент времени;

4) по возможности обучения по мере наблюдения за системой;

5) по возможности взаимодействия с объектами среды и с другими агентами.