Гибридное моделирование. Экзамен

# 1. Системы и подсистемы. Поведение систем. Модели систем.

**Система** — это совокупность элементов, взаимодействующих друг с другом для достижения определенной цели. Каждый элемент в системе имеет свою функцию и взаимосвязь с другими элементами.

**Подсистема** — это часть общей системы, которая сама является системой и выполняет специфические функции, но при этом взаимодействует с другими подсистемами в рамках общей системы.

**Поведение системы** описывается как совокупность реакций системы на внешние воздействия. Эти реакции зависят от внутренней структуры системы и её составляющих элементов. Поведение системы можно изучать через различные модели и методы анализа:

* **Динамическое поведение:** Характеризуется изменениями системы во времени.
* **Статическое поведение:** Рассматривается в фиксированный момент времени или в условиях, когда изменения минимальны.

**Моделирование систем** — это процесс создания абстрактных представлений реальных систем для их анализа и оптимизации. Модели систем помогают понять структуру и поведение системы, предсказать её реакцию на различные воздействия и принять оптимальные решения.

# 2. Классификация моделей. Основные типы моделей.

Классификация моделей

По назначению

1. **Описательные модели:** Используются для представления структуры и поведения системы без принятия конкретных решений. Примеры включают структурные диаграммы и функциональные модели.
2. **Прогнозные модели:** Предназначены для предсказания будущего поведения системы на основе текущих данных и тенденций.
3. **Оптимизационные модели:** Используются для поиска наилучших решений в заданных условиях с целью максимизации или минимизации определенных параметров.

По уровню абстракции

1. **Концептуальные модели:** Обобщенные представления системы, которые подчеркивают основные элементы и их взаимодействия. Они часто используются на начальных этапах анализа для понимания общей структуры системы.
2. **Физические модели:** Реальные или уменьшенные копии системы, используемые для визуального или практического исследования.
3. **Математические модели:** Используют математические уравнения и формулы для точного описания взаимодействий и поведения системы.

По подходу к построению

1. **Аналитические модели:** Строятся на основе строгих математических методов и используются для точного описания и анализа систем.
2. **Имитационные модели:** Моделируют систему с помощью компьютера, позволяя проводить эксперименты и наблюдать за поведением системы в различных условиях.

Основные типы моделей

Вербальные модели

* Описательные модели, представленные в текстовой форме. Они используются для предварительного анализа и формулировки проблемы.

Графические модели

* **Диаграммы:** Визуальные представления, такие как блок-схемы и графики, которые отображают структуру и процессы внутри системы.

Математические модели

* **Дискретные модели:** Модели, описывающие систему в виде дискретных событий или состояний, часто используемые в дискретно-событийном моделировании.
* **Непрерывные модели:** Описывают системы с непрерывными изменениями во времени, применяются в системной динамике и управлении процессами.

Имитационные модели

* **Дискретно-событийное моделирование:** Используется для систем, где изменения происходят в определенные моменты времени, такие как системы массового обслуживания.
* **Агентное моделирование:** Включает моделирование поведения отдельных агентов и их взаимодействий, что полезно для социальных и экономических систем.
* **Системная динамика:** Применяется для моделирования и анализа сложных динамических систем, таких как экологические и экономические системы.

# 3. Динамические модели. Методы реализации.

**Динамические модели** описывают поведение систем, изменяющихся во времени. Эти модели учитывают изменения состояния системы и взаимодействия её компонентов на протяжении времени, позволяя анализировать и прогнозировать динамику системы под воздействием различных факторов.

Существует несколько основных **методов реализации** динамических моделей:

1. **Системная динамика**
   * **Описание:** Метод системной динамики используется для моделирования и анализа сложных систем с обратными связями и временными задержками. Он основан на использовании дифференциальных уравнений для описания изменений в системе.
   * **Реализация:** Реализация системной динамики включает построение диаграмм причинно-следственных связей и потоковых диаграмм, которые затем переводятся в математические уравнения.
   * **Пример:** Использование системной динамики для моделирования экономического роста или изменения популяции в экосистеме.
2. **Дискретно-событийное моделирование**
   * **Описание:** Метод, который описывает систему как последовательность дискретных событий, происходящих в определённые моменты времени. Он полезен для моделирования систем, где события происходят нерегулярно.
   * **Реализация:** Реализация дискретно-событийного моделирования включает создание моделей, которые управляют очередями событий и симуляцией времени.
   * **Пример:** Моделирование работы системы массового обслуживания или производственного процесса.
3. **Агентное моделирование**
   * **Описание:** Метод моделирования, который фокусируется на взаимодействии отдельных агентов (индивидуумов, организаций и т.д.) в системе. Каждый агент имеет свои правила поведения и может взаимодействовать с другими агентами.
   * **Реализация:** Реализация агентного моделирования требует определения правил поведения агентов и их взаимодействий, а также симуляции среды, в которой агенты существуют.
   * **Пример:** Моделирование поведения потребителей на рынке или распространения заболеваний в популяции.

# 4. Оптимизационные модели. Методы принятия решений.

Оптимизационные модели используются для нахождения наилучшего решения (оптимума) из множества возможных вариантов. Эти модели применяются в различных областях, таких как экономика, логистика, производство и управление проектами.

Основные типы оптимизационных моделей

1. **Линейное программирование (LP):**
   * **Описание:** Модели, где цель и ограничения выражены линейными уравнениями и неравенствами.
   * **Пример:** Оптимизация распределения ресурсов для максимизации прибыли или минимизации затрат.
2. **Целочисленное программирование (IP):**
   * **Описание:** Подвид линейного программирования, где некоторые или все переменные должны быть целыми числами.
   * **Пример:** Оптимизация маршрутов доставки, где количество грузовиков должно быть целым числом.
3. **Нелинейное программирование (NLP):**
   * **Описание:** Модели, в которых целевая функция или ограничения являются нелинейными.
   * **Пример:** Оптимизация параметров технологического процесса для максимизации выхода продукции.
4. **Стохастическое программирование:**
   * **Описание:** Модели, учитывающие неопределенность в данных путем включения вероятностных параметров.
   * **Пример:** Оптимизация портфеля инвестиций с учетом риска и неопределенности будущих доходов.
5. **Динамическое программирование:**
   * **Описание:** Метод решения многопериодных задач, где решение принимается в несколько этапов.
   * **Пример:** Оптимизация стратегий управления запасами с учетом временных изменений спроса.

**Методы принятия решений**

Принятие решений в оптимизационных задачах может осуществляться различными методами в зависимости от сложности и типа задачи.

Основные методы принятия решений

1. **Метод ветвей и границ (Branch and Bound):**
   * Используется для решения задач целочисленного программирования. Метод включает разделение задачи на более простые подзадачи и оценку их границ для исключения неэффективных решений.
2. **Метод симплекса:**
   * Основной метод для решения задач линейного программирования. Он последовательно перемещается от одной вершины многоугольника решений к другой, улучшая значение целевой функции до достижения оптимума.
3. **Градиентные методы:**
   * Применяются в нелинейном программировании. Основаны на использовании градиентов целевой функции для нахождения направления улучшения решения.
4. **Эвристические методы:**
   * Используются для сложных и больших задач, где точные методы неэффективны. Примеры включают генетические алгоритмы, алгоритмы муравьиной колонии и симуляцию отжига.
5. **Стохастические методы:**
   * Применяются для задач с неопределенностью. Включают методы Монте-Карло и сценарный анализ для оценки вероятностных исходов и принятия решений на основе ожидаемой полезности.

# 5. Принятие решений в условиях риска и неопределенности.

**Риск** — это ситуация, в которой исходы известны и можно определить вероятность каждого из них. Принятие решений в условиях риска включает использование вероятностных данных для оценки возможных исходов и выбора наилучшего решения.

Методы принятия решений в условиях риска:

1. **Ожидаемая полезность:**
   * **Описание:** Метод основан на максимизации ожидаемой полезности, которая вычисляется как сумма произведений вероятностей на полезности каждого исхода.
   * **Пример:** Инвестор выбирает инвестиции, основываясь на ожидаемой доходности и риске.
2. **Анализ чувствительности:**
   * **Описание:** Метод оценки влияния изменений параметров модели на конечный результат. Помогает понять, как неопределенность входных данных влияет на решение.
   * **Пример:** Компания оценивает влияние изменения цен на сырье на прибыльность продукции.
3. **Деревья решений:**
   * **Описание:** Графический метод, который использует дерево для представления решений и их возможных последствий, включая вероятности и стоимость каждого пути.
   * **Пример:** Менеджер рассматривает возможные стратегии развития бизнеса с учетом вероятности успеха и затрат на каждую стратегию.
4. **Метод Монте-Карло:**
   * **Описание:** Статистический метод, использующий случайные числа и распределения для моделирования и оценки рисков.
   * **Пример:** Оценка риска инвестиционного проекта путем моделирования возможных сценариев доходности.

Принятие решений в условиях неопределенности

**Неопределенность** — это ситуация, в которой исходы неизвестны и не могут быть оценены с определенной вероятностью. Принятие решений в условиях неопределенности требует использования методов, которые помогают минимизировать потенциальные риски и максимизировать выгоды при отсутствии точной информации.

Методы принятия решений в условиях неопределенности:

1. **Максимин (критерий Вальда):**
   * **Описание:** Выбор решения, которое максимизирует минимальную возможную выгоду (пессимистический подход).
   * **Пример:** Фермер выбирает урожай, который приносит наибольшую минимальную прибыль в условиях неопределенности погодных условий.
2. **Максимакс (критерий оптимизма):**
   * **Описание:** Выбор решения, которое максимизирует максимальную возможную выгоду (оптимистический подход).
   * **Пример:** Предприниматель выбирает проект с наибольшим потенциалом дохода, игнорируя возможные риски.
3. **Критерий Сэвиджа (минимизация максимального сожаления):**
   * **Описание:** Выбор решения, которое минимизирует максимальное возможное сожаление от неправильного выбора.
   * **Пример:** Инвестор выбирает стратегию, которая минимизирует возможные упущенные выгоды от альтернативных инвестиций.
4. **Критерий Гурвица:**
   * **Описание:** Комбинирует максимин и максимакс, взвешивая их с коэффициентом оптимизма.
   * **Пример:** Менеджер использует смешанный подход для выбора бизнес-стратегии, учитывая как риски, так и потенциальные выгоды.
5. **Анализ сценариев:**
   * **Описание:** Метод, который включает разработку различных сценариев будущего и оценку последствий каждого из них.
   * **Пример:** Компания разрабатывает различные сценарии экономического развития и оценивает их влияние на свои стратегические планы.

# 6. Критерий Байеса-Лапласа (вероятностный). Условия применимости.

Критерий Байеса-Лапласа, или вероятностный критерий Байеса, представляет собой метод принятия решений на основе априорных знаний о вероятностях возможных исходов и полученных данных. Он является частным случаем более общего принципа максимума правдоподобия и может использоваться в различных областях, включая статистику, теорию принятия решений и машинное обучение.

Условия применимости критерия Байеса-Лапласа:

1. **Априорные знания о вероятностях**: Для применения критерия Байеса-Лапласа необходимо иметь априорные представления о вероятностях возможных исходов. Эти априорные знания могут быть основаны на предыдущих исследованиях, экспертных оценках или предположениях.
2. **Соблюдение аксиоматики вероятности**: Вероятности должны быть определены для всех возможных исходов и должны соблюдать основные аксиомы вероятности, такие как неотрицательность и нормировка (сумма вероятностей всех исходов равна единице).
3. **Достоверные данные**: Должны быть получены данные или наблюдения, которые могут быть использованы для обновления априорных вероятностей и получения постериорных вероятностей.

# 7. Критерий Вальда (крайнего пессимизма). Условия применимости.

Критерий Вальда, также известный как критерий крайнего пессимизма или минимаксный критерий, является методом принятия решений в условиях неопределённости, при котором выбирается альтернатива, которая обеспечивает минимальный гарантированный результат в худших условиях. Этот критерий основывается на принципе минимизации максимальных потерь.

Условия применимости критерия Вальда:

1. **Риск и неопределённость**: Критерий Вальда применяется в ситуациях, где имеется значительная степень неопределённости или риска, связанного с возможными исходами.
2. **Информационная неоднородность**: Когда доступная информация о возможных исходах неоднородна и может варьироваться.
3. **Решения в условиях неопределённости**: Принятие решений в ситуациях, где невозможно оценить вероятности исходов или они сильно колеблются.

Пример применения критерия Вальда:

Предположим, у нас есть задача выбора проекта для инвестиций. У нас есть два проекта с возможными исходами прибыли (в миллионах долларов):

* Проект A: Возможные исходы прибыли: 10M,20M,30M10M, 20M, 30M10M,20M,30M
* Проект B: Возможные исходы прибыли: 5M,25M,35M5M, 25M, 35M5M,25M,35M

Используя критерий Вальда, мы выберем тот проект, который гарантирует минимально возможную максимальную прибыль. Для проекта A максимальная прибыль 30M (в худшем случае), а для проекта B — 35M. Следовательно, по критерию Вальда выбирается проект A, так как он обеспечивает более высокий гарантированный минимум при максимальных возможных рисках.

# 8. Критерий Сэвиджа (минимакс потерь). Условия применимости.

Критерий Сэвиджа (минимакс потерь) — это метод принятия решений в условиях неопределённости, который нацелен на минимизацию максимальных потерь. Он также известен как минимаксный критерий оптимизации. Этот метод разработан для ситуаций, когда вероятности возможных исходов неизвестны или сложно определить, но можно оценить возможные потери при различных решениях.

Условия применимости критерия Сэвиджа:

1. **Наличие альтернативных решений**: Должно быть несколько альтернативных решений или стратегий, которые можно сравнивать по возможным последствиям.
2. **Неопределённость или риск**: Критерий Сэвиджа подходит для принятия решений в условиях неопределённости, когда неизвестны вероятности возможных исходов или они сильно колеблются.
3. **Оценка потерь**: Должна быть возможность оценить потери или негативные последствия каждого решения в каждом из возможных исходов.

Пример применения критерия Сэвиджа:

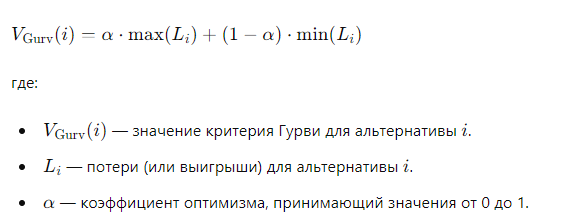
Рассмотрим пример принятия решения о том, инвестировать ли в новый проект. У нас есть две альтернативы с возможными исходами прибыли и их ожидаемыми потерями:

* Альтернатива X:
  + Потери при исходе A: 10M
  + Потери при исходе B: 5M
  + Потери при исходе C: 15M
* Альтернатива Y:
  + Потери при исходе A: 8M
  + Потери при исходе B: 6M
  + Потери при исходе C: 18M

Используя критерий Сэвиджа, мы выбираем ту альтернативу, для которой минимальные из максимальных потерь. Например, если максимальные потери для X составляют 15M (в случае исхода C), а для Y — 18M (в случае исхода C), то по критерию Сэвиджа предпочтительнее альтернатива X.

# 9. Критерии Гурвица (оптимизма - пессимизма) и Ходжа - Лемана (комбинация MM и BL). Условия применимости.

Критерий Гурвица использует коэффициент оптимизма α, чтобы найти компромисс между оптимистическим и пессимистическим подходами при принятии решений. Формула для критерия Гурвица выглядит следующим образом:

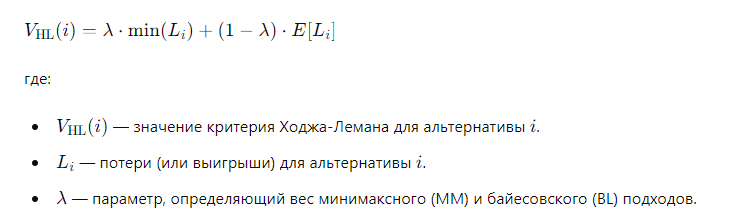


Условия применимости критерия Гурвица:

1. **Оптимизм и пессимизм**: Критерий Гурвица особенно полезен в ситуациях, когда принимающее решение человека или группы зависит от баланса между оптимистическими и пессимистическими взглядами на решение.
2. **Оценка альтернатив**: Должна быть возможность оценить потери или выигрыши для каждой альтернативы, чтобы применить формулу критерия Гурвица.
3. **Выбор коэффициента α**: Необходимо определить или оценить оптимальное значение коэффициента α, который наилучшим образом отражает предпочтения принимающего решение.

Критерий Ходжа-Лемана (комбинация MM и BL):

Критерий Ходжа-Лемана комбинирует минимаксный (MM) и байесовский (BL) подходы к принятию решений. Он использует параметр λ для взвешивания между этими двумя подходами. Формула для критерия Ходжа-Лемана выглядит следующим образом:



Условия применимости критерия Ходжа-Лемана:

1. **Комбинированный подход**: Критерий Ходжа-Лемана подходит для ситуаций, когда желательно учесть как минимаксный (максимальный пессимизм), так и байесовский (учёт вероятностей) подходы в принятии решений.
2. **Оценка потерь и вероятностей**: Необходимо иметь возможность оценить потери (или выигрыши) для каждой альтернативы и, возможно, вероятности возможных исходов для применения байесовского подхода.
3. **Выбор параметра λ**: Требуется определить или оценить оптимальное значение параметра λ, которое соответствует предпочтениям и условиям принятия решения.

# 10. Принятие решений в условиях конфликта. Теоретико-игровые модели. Чистые и смешанные стратегии.

**Теория игр**: Это математическая дисциплина, которая изучает стратегии принятия решений в условиях конфликта, где интересы различных сторон противоречат друг другу.

**Игроки**: Участники конфликта или ситуации, которые могут принимать решения.

**Стратегии**: Планы действий, которые игроки могут использовать для достижения своих целей.

**Чистая стратегия**: Это стратегия, при которой игрок выбирает одно определенное действие или решение с определенной вероятностью. Например, в игре, где игроки выбирают между двумя стратегиями A и B, чистая стратегия означает выбор одной из них с вероятностью 1.

**Смешанная стратегия**: Это стратегия, при которой игроки случайным образом выбирают свои действия в соответствии с вероятностным распределением. Таким образом, игрок может выбирать действия A и B с определенными вероятностями ppp и 1−p1-p1−p соответственно.

Принятие решений в условиях конфликта:

* **Равновесие по Нэшу**: Это ключевой концепт в теории игр, который описывает ситуацию, когда ни один игрок не может улучшить свою позицию, играя альтернативную стратегию, при условии, что другие игроки действуют оптимально. Равновесие по Нэшу выявляется через анализ стратегий и результатов игры.
* **Анализ игровых моделей**: Для анализа игровых моделей часто используются математические методы, такие как матрицы выигрышей, деревья игр и вероятностные модели. Эти методы позволяют определить оптимальные стратегии и прогнозировать исходы конфликтных ситуаций.

Пример:

Рассмотрим пример классической игры "Заключённый дилемма", где два заключённых выбирают между сотрудничеством и предательством. Выигрыши зависят от того, какие стратегии выбирают оба игрока:

* Если оба сотрудничают, каждый получает низкий, но ненулевой выигрыш.
* Если один предает, а другой сотрудничает, предатель получает высокий выигрыш, а сотрудник — высокий убыток.
* Если оба предают, каждый получает средний выигрыш.

Анализируя эту игру с использованием равновесия по Нэшу и стратегий, можно определить оптимальные действия для каждого игрока в зависимости от стратегий оппонента.

# 11. Антагонистическая игра двух лиц (игра с нулевой суммой). Минимаксная стратегия.

Антагонистическая игра двух лиц, или игра с нулевой суммой, является классическим примером в теории игр. В такой игре выигрыш одного игрока является прямой потерей для другого игрока, и наоборот. Основная задача каждого игрока — максимизировать свой выигрыш при условии, что выигрыш другого игрока является потерей для него.

Минимаксная стратегия в играх с нулевой суммой:

Минимаксная стратегия представляет собой стратегию, при которой каждый игрок выбирает действия таким образом, чтобы минимизировать потенциальные максимальные потери, которые он может понести, и максимизировать гарантированный выигрыш, если возможно.

Шаги по определению минимаксной стратегии:

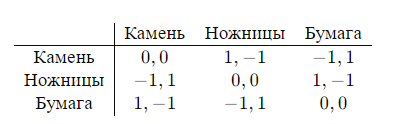
1. **Построение матрицы выигрышей**: В играх с нулевой суммой используется матрица выигрышей, где строки представляют стратегии первого игрока, а столбцы — стратегии второго игрока. Значения в матрице обозначают выигрыши (или потери) для каждой пары стратегий.
2. **Минимизация максимальных потерь**: Первый игрок выбирает стратегию таким образом, чтобы минимизировать максимальный возможный выигрыш противника. Это означает, что первый игрок рассматривает каждую свою возможную стратегию и выбирает ту, которая обеспечивает наименьший возможный выигрыш второму игроку (максимальную его потерю).
3. **Максимизация гарантированного выигрыша**: Второй игрок, зная выбранную первым игроком минимаксную стратегию, выбирает стратегию, которая максимизирует его гарантированный выигрыш при условии, что первый игрок будет играть оптимально.

Пример минимаксной стратегии:

Рассмотрим классический пример игры "Камень, ножницы, бумага" в формате с нулевой суммой, где каждый игрок может выбрать одно из трех действий. Матрица выигрышей может выглядеть следующим образом:

* Если игрок А выбирает камень, а игрок Б выбирает ножницы, А выигрывает.
* Если игрок А выбирает бумагу, а игрок Б выбирает ножницы, Б выигрывает.
* Если игроки выбирают одинаковые действия, это ничья.

Матрица выигрышей (выигрыш А / потеря Б) может быть такой:



Чтобы найти минимаксную стратегию первого игрока (А), он будет выбирать стратегию, которая минимизирует максимальный потенциальный выигрыш второго игрока (Б). Например, если А выбирает камень, то максимальная потеря Б будет 1 (в случае выбора ножниц Б), поэтому для А оптимально выбирать камень.

# 12. Бескоалиционная игра. Равновесие Нэша.

Бескоалиционная игра — это тип игры в теории кооперативных игр, где игроки действуют независимо друг от друга и не формируют коалиций или союзов. Каждый игрок принимает решение, опираясь на собственные интересы и цели. Основным концептом в бескоалиционных играх является **равновесие по Нэшу**.

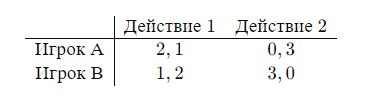
Равновесие по Нэшу в бескоалиционной игре:

Равновесие по Нэшу представляет собой такую ситуацию, при которой ни один игрок не может улучшить своё положение, изменяя свою стратегию, при условии, что все остальные игроки действуют по своим текущим стратегиям. Иными словами, это стратегический набор действий, при котором нет мотивации для индивидуального игрока изменять свою стратегию, учитывая стратегии остальных игроков.

Пример:

Рассмотрим простой пример бескоалиционной игры с двумя игроками (А и В). Каждый игрок может выбрать одно из двух действий (стратегий).

Представим игру с матрицей выигрышей:



Здесь первое число в каждой ячейке представляет выигрыш игрока А, а второе — выигрыш игрока В.

Шаги по нахождению равновесия по Нэшу:

1. **Исследование стратегий игроков**: Определите, какие стратегии выбирают игроки в настоящее время. Например, если оба игрока выбирают "Действие 1", это потенциальное равновесие.
2. **Оценка выигрышей при различных стратегиях**: Проверьте, может ли любой игрок улучшить свой выигрыш, изменяя свою стратегию при условии, что другой игрок остаётся при своей текущей стратегии. В данном примере:
   * Если игрок А перейдёт с "Действие 1" на "Действие 2", его выигрыш изменится с 2 до 0, при этом выигрыш игрока В останется тем же (3).
   * Если игрок В перейдёт с "Действие 1" на "Действие 2", его выигрыш изменится с 2 до 0, при этом выигрыш игрока А останется тем же (1).
3. В обоих случаях изменение стратегии приведёт к ухудшению выигрыша для соответствующего игрока, поэтому нет мотивации для изменения стратегий. Таким образом, пара стратегий (Действие 1, Действие 1) является равновесием по Нэшу в данной игре.

# 13. Рефлексивная игра. Моделирование информированности игрока на планшете.

**Рефлексивные игры** — это игры, в которых игроки принимают решения, учитывая не только свои собственные стратегии и выигрыши, но и предполагаемые стратегии и ожидания других игроков. Эти игры моделируют ситуации, где игроки осознают, что другие игроки также принимают решения на основе своих ожиданий о действиях других игроков.

Моделирование информированности игрока:

В рефлексивных играх важно учитывать, какой информацией обладает каждый игрок и как эта информация влияет на их стратегические решения. Для моделирования информированности игрока можно использовать несколько подходов, включая:

1. **Уровни рефлексии**:
   * **Нулевой уровень рефлексии**: Игроки принимают решения, основываясь только на своей собственной стратегии и выигрыше, не учитывая других игроков.
   * **Первый уровень рефлексии**: Игроки учитывают, что другие игроки тоже принимают решения, и строят свои стратегии на предположении о действиях других.
   * **Второй уровень рефлексии и выше**: Игроки осознают, что другие игроки также учитывают их действия и стратегии, и строят свои решения на более глубоком уровне осознания взаимодействий.
2. **Моделирование информированности на планшете**:
   * **Структура планшета**: Планшет или доска может быть использован для визуализации стратегии и информированности каждого игрока. Планшет отображает текущие стратегии, предположения о действиях других игроков и предполагаемые выигрыши.
   * **Информационные слои**: Планшет может быть разделён на несколько слоёв, каждый из которых отображает уровень информированности или рефлексии. Например, один слой может показывать стратегии первого уровня рефлексии, другой слой — второго уровня и так далее.

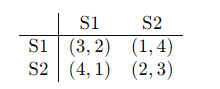
Пример рефлексивной игры:

Рассмотрим игру с двумя игроками (А и В), где каждый игрок выбирает стратегию, основываясь на своих предположениях о стратегии другого игрока. Например, каждый игрок может выбрать стратегию из набора {S1, S2}.

Информированность и стратегии:

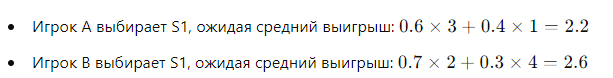
* **Игрок А** предполагает, что игрок В выберет S1 с вероятностью 0.6 и S2 с вероятностью 0.4.
* **Игрок В** предполагает, что игрок А выберет S1 с вероятностью 0.7 и S2 с вероятностью 0.3.

Выигрышная матрица:



Моделирование на планшете:

1. **Слой 1**: Отображение предположений игроков:
   * Игрок А: S1 (0.6), S2 (0.4)
   * Игрок В: S1 (0.7), S2 (0.3)
2. **Слой 2**: Отображение стратегий и выигрышей на основе предположений:



1. **Слой 3**: Адаптация стратегий на основе рефлексии:
   * Игрок А может пересмотреть свою стратегию, если считает, что игрок В также пересматривает свою стратегию на основе информации.

# 14. Методы многокритериальной оценки. Лексикографический порядок. Весовые коэффициенты.

Многокритериальная оценка (МКО) используется для принятия решений, когда необходимо учитывать несколько критериев или факторов. Методы многокритериальной оценки помогают определить наилучшую альтернативу среди множества возможных вариантов. Вот некоторые из наиболее распространенных методов и подходов:

1. Лексикографический порядок:

Лексикографический порядок (или метод лексикографической сортировки) — это метод, при котором критерии ранжируются по важности. Решения принимаются последовательно, начиная с наиболее важного критерия. Если две альтернативы равны по самому важному критерию, рассматривается следующий по важности критерий и так далее.

Пример:

Предположим, что у нас есть три альтернативы (A, B и C) и три критерия (K1, K2 и K3), ранжированные по важности следующим образом: K1 > K2 > K3.

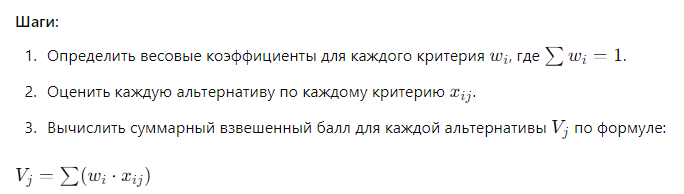
* Альтернатива A: (K1 = 7, K2 = 5, K3 = 8)
* Альтернатива B: (K1 = 8, K2 = 6, K3 = 7)
* Альтернатива C: (K1 = 8, K2 = 5, K3 = 6)

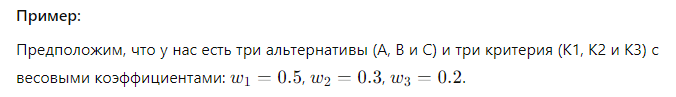
Порядок действий:

1. Сравниваем все альтернативы по K1. A имеет значение 7, B и C имеют значение 8.
2. A исключается, так как имеет наименьшее значение по K1.
3. Сравниваем B и C по K2. B имеет значение 6, C — 5.
4. Альтернатива B побеждает, так как имеет наибольшее значение по K2.

2. Весовые коэффициенты:

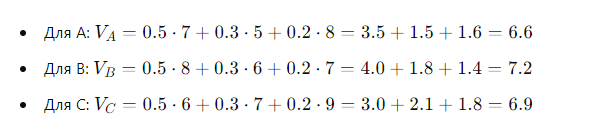
Метод весовых коэффициентов (или метод взвешенных сумм) предполагает назначение весов каждому критерию в зависимости от его важности. Затем для каждой альтернативы вычисляется суммарный взвешенный балл, и альтернатива с наибольшим баллом считается наилучшей.





* Альтернатива A: (K1 = 7, K2 = 5, K3 = 8)
* Альтернатива B: (K1 = 8, K2 = 6, K3 = 7)
* Альтернатива C: (K1 = 6, K2 = 7, K3 = 9)

Вычисляем суммарные взвешенные баллы:



Поэтому альтернатива B побеждает, так как имеет наибольший суммарный взвешенный балл (7.2).

Преимущества и недостатки методов:

* **Лексикографический порядок**:
  + **Преимущества**: Простота, не требует количественной оценки критериев.
  + **Недостатки**: Игнорирует критерии, которые не попадают в первые несколько по важности.
* **Весовые коэффициенты**:
  + **Преимущества**: Учитывает все критерии, позволяет гибко настраивать важность каждого критерия.
  + **Недостатки**: Требует определения весов, что может быть субъективным и сложным.

# 15. Построение оценок на основе анализа иерархий.

Построение оценок на основе анализа иерархий (Analytic Hierarchy Process, AHP) — это метод многокритериальной оценки, разработанный Томасом Саати, который позволяет систематизировать и упорядочить критерии и альтернативы при принятии решений. AHP используется в различных областях для управления и принятия решений, особенно когда необходимо учитывать комплексные системы критериев и их взаимосвязи.

Основные этапы построения оценок на основе анализа иерархий:

1. **Определение цели и иерархии критериев**:
   * **Цель**: Четко определяется цель принятия решения.
   * **Иерархия критериев**: Определяются основные критерии, влияющие на достижение цели, и подкритерии, разделяющиеся по уровням важности.
2. **Построение матриц парных сравнений**:
   * Для каждой пары критериев (или альтернатив) составляется матрица парных сравнений.
   * Эксперты оценивают относительную важность одного критерия по отношению к другому, используя шкалу сравнений Саати (обычно от 1 до 9).
3. **Вычисление весов критериев**:
   * Для каждой матрицы парных сравнений вычисляются собственные значения и векторы приоритетов.
   * На основе этих данных вычисляются весовые коэффициенты для каждого критерия.
4. **Синтезирование оценок**:
   * Вычисляется совокупная оценка для каждой альтернативы, учитывая весовые коэффициенты критериев.
   * Это может быть выполнено путем умножения оценок альтернатив на соответствующие весовые коэффициенты и их суммирования.
5. **Чувствительный анализ**:
   * Проводится оценка чувствительности результатов к изменениям в исходных данных (матрицы парных сравнений).
   * Это позволяет оценить стабильность и надежность полученных оценок.

Пример:

Рассмотрим простой пример применения AHP для выбора стратегии развития бизнеса:

1. **Цель**: Выбор наилучшей стратегии развития.
2. **Иерархия критериев**:
   * Главные критерии: Экономическая эффективность, Риск, Время окупаемости.
   * Подкритерии под каждым главным критерием.
3. **Матрицы парных сравнений**:
   * Эксперты оценивают важность каждого критерия по отношению к другому с помощью шкалы от 1 до 9. Например, оценка того, насколько экономическая эффективность важнее времени окупаемости.
4. **Вычисление весов критериев**:
   * Для каждой матрицы парных сравнений вычисляются собственные значения и векторы приоритетов.
   * На основе этого определяются весовые коэффициенты для каждого критерия.
5. **Синтез оценок**:
   * Оценки для каждой альтернативы умножаются на соответствующие весовые коэффициенты и суммируются для получения общей оценки каждой альтернативы.

Преимущества метода AHP:

* Структурированное и системное подход к принятию решений.
* Учет множества критериев и их взаимосвязей.
* Позволяет объективно определить важность каждого критерия на основе экспертных оценок.

Недостатки метода AHP:

* Требует значительных усилий и времени на сбор и обработку данных.
* Результаты могут зависеть от выбора экспертов и их квалификации.
* Чувствительность к изменениям в исходных данных (матрицы сравнений).

В целом, метод анализа иерархий (AHP) представляет собой мощный инструмент для принятия сложных многокритериальных решений, который широко применяется в различных областях, включая бизнес, инженерные и социальные науки.

# 16. Многошаговые процедуры принятия решений.

Многошаговые процедуры принятия решений представляют собой методологии или подходы, которые включают последовательность этапов или шагов для систематического решения проблемы или выбора наилучшей альтернативы. Эти процедуры разрабатываются с целью упрощения сложных решений, учета множества факторов и минимизации влияния субъективности.

Основные многошаговые процедуры принятия решений:

1. **Идентификация проблемы или цели**:
   * Определение и четкое формулирование проблемы или цели, которая требует принятия решения.
2. **Сбор информации и анализ**:
   * Сбор необходимых данных и информации, которые могут включать критерии, альтернативы, ограничения и другие соответствующие факторы.
   * Анализ данных для оценки текущего состояния и понимания взаимосвязей между различными факторами.
3. **Определение критериев и важности**:
   * Определение критериев, которые будут использоваться для оценки альтернатив.
   * Установление важности каждого критерия и их относительных приоритетов.
4. **Формулирование альтернатив**:
   * Генерация возможных альтернативных решений или стратегий, которые могут решить проблему или достичь поставленной цели.
5. **Оценка и сравнение альтернатив**:
   * Применение методов оценки и сравнения альтернатив на основе определенных критериев.
   * Использование математических моделей, методов многокритериального анализа или экспертных оценок для выявления наилучшей альтернативы.
6. **Принятие решения**:
   * Выбор наилучшей альтернативы на основе результатов оценки и сравнения.
   * Принятие окончательного решения с учетом всех факторов и критериев.
7. **Реализация и контроль**:
   * Разработка плана реализации выбранной альтернативы.
   * Мониторинг и контроль выполнения плана для достижения поставленных целей.

Примеры многошаговых процедур:

* **Процесс Six Sigma DMAIC** (Define, Measure, Analyze, Improve, Control) — используется для улучшения и оптимизации бизнес-процессов.
* **Метод анализа иерархий (AHP)** — включает в себя шаги сбора парных сравнений, расчета весов критериев и синтеза оценок.
* **Процесс SWOT-анализа** (Strengths, Weaknesses, Opportunities, Threats) — анализ внутренних и внешних аспектов организации для принятия стратегических решений.
* **Методология принятия решений на основе экспертных оценок** — использование знаний и опыта экспертов для оценки альтернатив.

# 17. Системы массового обслуживания. Основные элементы СМО и их характеристические свойства.

Система массового обслуживания (СМО) — это математическая модель, используемая для анализа и оптимизации процессов обслуживания, таких как обработка заявок, обслуживание клиентов в банке или на кассе, ожидание в очереди и т.д. Основные элементы СМО включают в себя:

Основные элементы СМО:

1. **Поток заявок (заявок)**:
   * Это множество событий, когда поступают заявки на обслуживание в систему. Заявки могут поступать как детерминированно (по фиксированному расписанию) или случайно (по случайному процессу).
2. **Очередь (если применимо)**:
   * Если приходят заявки быстрее, чем может быть выполнено обслуживание, они образуют очередь, в которой ожидают своей очереди на обслуживание.
3. **Устройство обслуживания (сервер)**:
   * Это элемент, который осуществляет обслуживание заявок. Например, это может быть кассир, работник обслуживания клиентов, сервер в сетевой системе и т.д.

Характеристические свойства СМО:

1. **Интенсивность поступления заявок (λ)**:
   * Это среднее количество заявок, поступающих в систему за единицу времени. Обозначается как λ (lambda).
2. **Интенсивность обслуживания (μ)**:
   * Это среднее количество заявок, которые устройство обслуживания способно обработать за единицу времени. Обозначается как μ (mu).
3. **Число каналов обслуживания (c)**:
   * Это количество параллельно работающих устройств обслуживания в системе. Если c = 1, то это система с одним сервером; если c > 1, то это система с несколькими серверами.
4. **Тип очереди**:
   * Определяет, как управляется очередь, если в системе есть больше заявок, чем может обслужить одно устройство. Например, это может быть FIFO (First In, First Out), LIFO (Last In, First Out), приоритетная очередь и т.д.
5. **Распределение времени обслуживания**:
   * Определяет, как распределено время, необходимое для обслуживания каждой заявки. Это может быть экспоненциальное, нормальное, равномерное распределение и т.д.

Важные характеристики СМО:

* **Интенсивность потока заявок (λ)** определяет загрузку системы. Если λ > μ, то система перегружена и очередь будет расти бесконечно.
* **Коэффициент использования сервера (ρ)** выражает степень загрузки устройства обслуживания и вычисляется как ρ = λ / μ. Если ρ близко к 1, то сервер используется почти на полную мощность.
* **Средняя длина очереди (Lq)** и **среднее время ожидания в очереди (Wq)** — важные показатели эффективности СМО, которые оценивают средний объем заявок в очереди и среднее время, которое заявки проводят в очереди.

# 18. Имитационное моделирование. Поведение системы и модели.Возможности, открывающиеся при ИМ.

Имитационное моделирование (ИМ) — это метод компьютерного моделирования, который используется для изучения поведения реальных систем или процессов путем создания моделей, которые могут воспроизводить основные характеристики и взаимодействия этих систем. Основной идеей имитационного моделирования является создание виртуальной модели, которая аппроксимирует поведение реальной системы и позволяет исследовать различные сценарии и варианты управления без непосредственного воздействия на реальные объекты.

Поведение системы и модели в имитационном моделировании:

1. **Модель**:
   * **Структурная модель**: Описывает структуру системы, включая компоненты (участники, ресурсы), их взаимодействия и потоки данных или ресурсов.
   * **Поведенческая модель**: Описывает, как каждый компонент системы взаимодействует с другими и каким образом система реагирует на внешние воздействия.
2. **Поведение системы**:
   * Имитационное моделирование позволяет воссоздать динамику системы во времени. Это включает изменения состояний системы, потоки данных или ресурсов, изменения параметров в ответ на внешние или внутренние события.
3. **Взаимодействие среды и модели**:
   * Виртуальная среда, созданная в рамках имитационной модели, взаимодействует с внешней средой через входные данные и параметры модели. Модель может быть настроена на изменение внешних условий для анализа их влияния на поведение системы.

Возможности, открывающиеся при использовании имитационного моделирования:

1. **Экспериментация с различными сценариями**:
   * ИМ позволяет проверять различные гипотезы и сценарии виртуально, не влияя на реальную систему. Это позволяет исследовать альтернативные стратегии управления и принимать обоснованные решения.
2. **Оптимизация процессов**:
   * Моделирование позволяет оптимизировать производственные процессы, управление запасами, распределение ресурсов и другие аспекты бизнеса путем итеративного тестирования различных конфигураций и параметров системы.
3. **Предвидение поведения системы**:
   * ИМ может быть использовано для прогнозирования поведения системы в различных условиях и предсказания будущих тенденций на основе текущих данных и предыдущего опыта.
4. **Обучение и обучающие цели**:
   * ИМ часто используется для обучения персонала в управлении сложными системами или процессами, создания симуляций для обучения реальных навыков и стратегий.
5. **Разработка и тестирование новых концепций**:
   * Виртуальные модели позволяют исследовать новые концепции и технологии, прежде чем внедрять их в реальные условия, что снижает риски и улучшает эффективность проектов.

# 19. Случайные и псевдослучайные числа. Критерии случайности. Методы генерации ПСП.

Случайные числа и псевдослучайные числа играют важную роль в компьютерных науках, статистике, криптографии и других облас

Случайные числа представляют собой значения, которые появляются как результат случайного процесса или явления. Они обладают следующими особенностями:

* **Недетерминированность**: Случайные числа не подчиняются определенным законам или закономерностям.
* **Индифферентность**: Вероятность появления любого из значений одинакова (равномерное распределение).
* **Невозможность предсказания**: Нельзя предсказать следующее случайное число на основе предыдущих.

Примеры случайных чисел в реальном мире могут включать результаты броска кубика или монеты, случайные флуктуации в физических процессах (например, термошум) или случайные события в квантовой механике.

Псевдослучайные числа (ПСП)

Псевдослучайные числа создаются при помощи алгоритмов, которые используют начальное значение (зерно, seed) для генерации последовательности чисел. Они обладают следующими особенностями:

* **Детерминированность**: Последовательность чисел генерируется детерминированным алгоритмом с заданным зерном.
* **Повторяемость**: При использовании одного и того же зерна последовательность чисел всегда будет одинаковой.
* **Приближение к случайности**: Хорошие генераторы ПСП стараются эмулировать случайность с высокой степенью равномерности и отсутствием предсказуемости.

Критерии случайности

Чтобы оценить качество случайных или псевдослучайных чисел, используются различные критерии:

1. **Равномерность распределения**: Каждое значение должно иметь одинаковую вероятность появления.
2. **Независимость и некоррелированность**: Каждое число должно быть независимым от предыдущих и не должно быть коррелировано с последующими.
3. **Отсутствие закономерностей**: Нельзя найти систему или правило, которое предсказывает последовательность чисел.
4. **Широкий диапазон**: Числа должны охватывать всю область значений, определенную типом данных.
5. **Энтропия**: Мера случайности, выражающая количество информации, содержащейся в последовательности чисел.

Методы генерации псевдослучайных чисел

Существует несколько методов для генерации ПСП, которые различаются по сложности и качеству:

1. **Линейный конгруэнтный метод (LCG)**: Простой и широко используемый алгоритм, основанный на рекуррентном соотношении.
2. **Методы на основе хеш-функций**: Применение хеш-функций к входным данным для получения случайных значений.
3. **Алгоритмы на основе шума**: Использование шумовых генераторов для создания случайных чисел.
4. **Криптографические генераторы**: Генераторы, которые используются в криптографии для создания криптографических ключей и случайных чисел высокой степени сложности.
5. **Смешанные методы**: Комбинация различных алгоритмов для улучшения качества случайных чисел.

# 20. Моделирование случайного события.

Случайное событие — это событие, которое может произойти или не произойти в результате какого-либо эксперимента или наблюдения. Вероятность случайного события — это мера его вероятности, выраженная числом от 0 до 1.

Методы моделирования случайных событий

1. **Метод Монте-Карло**
   * **Описание:** Использует генерацию случайных чисел и статистический анализ для моделирования и анализа сложных систем и процессов, в которых присутствуют случайные события.
   * **Пример:** Моделирование запасов в системе управления запасами, где спрос является случайным.
2. **Метод имитации (симуляция)**
   * **Описание:** Создание имитационной модели системы, которая включает случайные события и позволяет исследовать поведение системы в различных сценариях.
   * **Пример:** Имитация работы производственной линии с учетом случайных простоев оборудования.
3. **Метод дискретно-событийного моделирования**
   * **Описание:** Описывает систему как последовательность дискретных событий, каждое из которых изменяет состояние системы.
   * **Пример:** Моделирование работы системы массового обслуживания, такой как банк или магазин, с учетом случайных приходов клиентов.
4. **Статистическое моделирование**
   * **Описание:** Использование статистических методов для моделирования случайных процессов и событий на основе исторических данных и вероятностных распределений.
   * **Пример:** Моделирование финансовых рынков на основе исторических данных о ценах акций и их волатильности.

Примеры распределений вероятностей

* **Равномерное распределение:** Все возможные исходы имеют одинаковую вероятность.
* **Нормальное распределение:** Вероятности исходов распределены по колоколообразной кривой, симметричной относительно среднего значения.
* **Экспоненциальное распределение:** Моделирует время между случайными событиями, такими как время ожидания между прибытием клиентов.
* **Пуассоновское распределение:** Используется для моделирования числа событий, происходящих в фиксированный промежуток времени.

Шаги моделирования случайного события

1. **Определение цели и структуры модели**
   * Определите цель моделирования и основные элементы системы, которые необходимо включить в модель.
2. **Сбор и анализ данных**
   * Соберите исторические данные и определите вероятностные распределения для случайных событий.
3. **Разработка модели**
   * Создайте математическую или имитационную модель системы, включив в неё случайные события и их вероятности.
4. **Проведение экспериментов**
   * Проведите серию симуляций или экспериментов, используя модель для анализа поведения системы в различных сценариях.
5. **Анализ результатов**
   * Анализируйте результаты моделирования для принятия решений и оптимизации системы.

Программные инструменты для моделирования случайных событий

1. **MATLAB/Simulink**
   * Предлагает функции для генерации случайных чисел и моделирования случайных процессов.
2. **Arena**
   * Используется для дискретно-событийного моделирования и анализа систем массового обслуживания.
3. **AnyLogic**
   * Поддерживает агентное моделирование, системную динамику и дискретно-событийное моделирование, позволяя комбинировать различные подходы.
4. **R и Python (библиотеки SciPy, NumPy)**
   * Обеспечивают инструменты для статистического анализа и моделирования случайных процессов.

# 21. Моделирование случайной величины методом обратной функции.

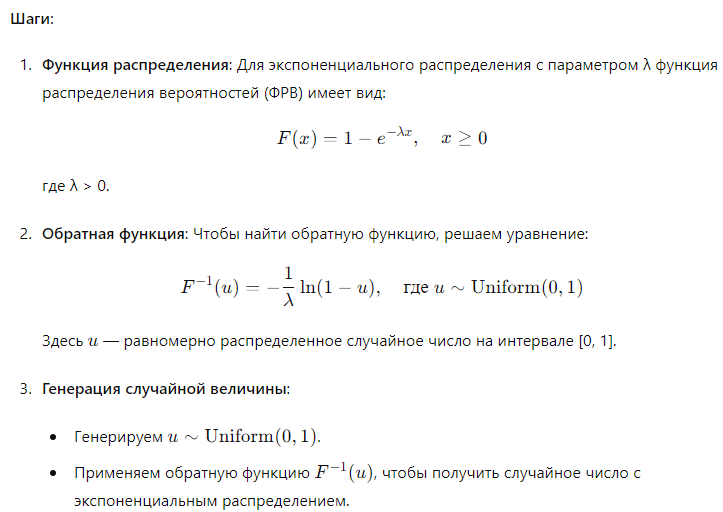
Метод обратной функции (Inverse Transform Method) — это один из классических методов для генерации случайных чисел с заданным вероятностным распределением. Он основывается на том, что если у нас есть функция распределения вероятностей (ФРВ) случайной величины, то можно использовать её обратную функцию для преобразования равномерно распределённых случайных чисел в числа с нужным распределением.

Основные шаги метода обратной функции:

1. **Выбор функции распределения**: Необходимо знать или определить функцию распределения вероятностей (ФРВ), для которой нужно сгенерировать случайные значения.
2. **Нахождение обратной функции**: Необходимо выразить случайную величину через обратную функцию распределения вероятностей.
3. **Генерация равномерно распределённых случайных чисел**: Сгенерировать случайные числа, равномерно распределенные на интервале [0, 1].
4. **Применение обратной функции**: Применить обратную функцию к сгенерированным случайным числам для получения значений случайной величины с заданным распределением.

Пример:

Давайте рассмотрим пример для генерации случайной величины с экспоненциальным распределением с параметром λ.



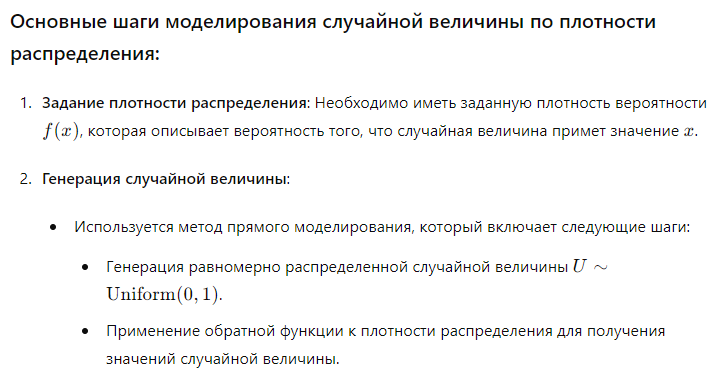
Этот метод широко используется из-за своей эффективности и простоты. Он позволяет генерировать случайные величины с различными распределениями, такими как экспоненциальное, нормальное, равномерное и другие, используя только генератор равномерного распределения.

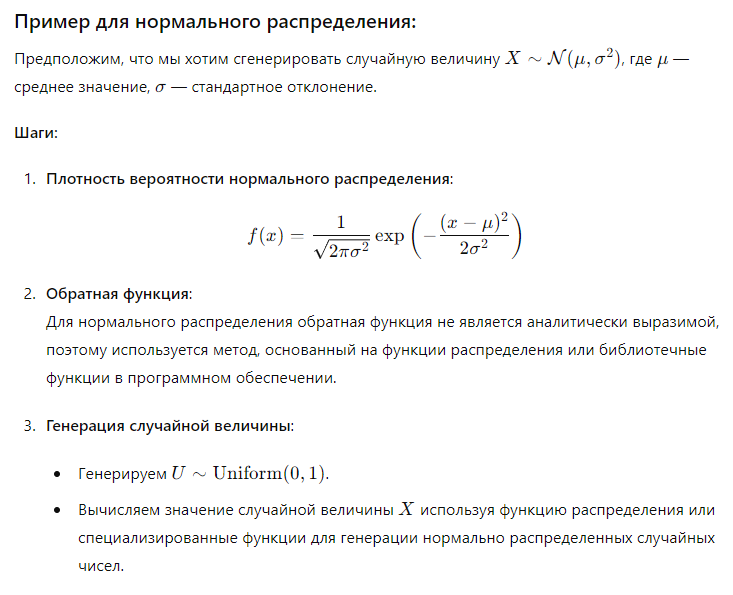
Преимущества метода обратной функции:

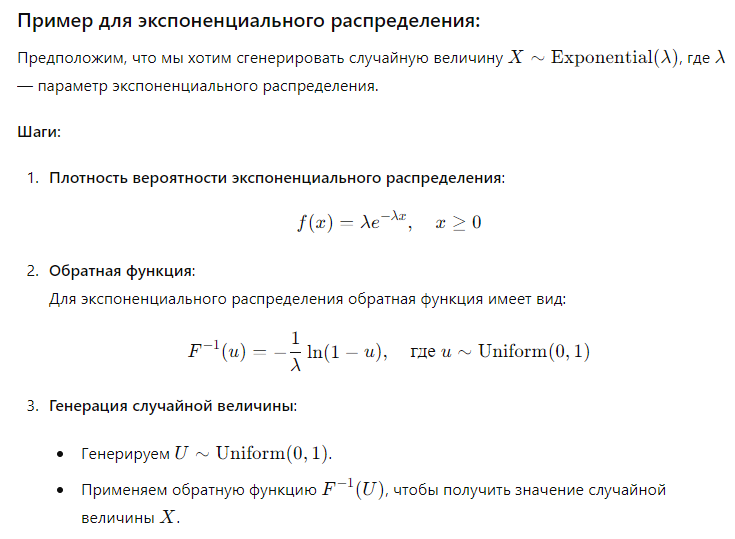
* **Простота**: Относительно прост в реализации и понимании.
* **Эффективность**: Генерация случайных чисел происходит быстро, особенно при использовании современных компьютеров.
* **Точность**: При правильном выборе функции распределения и генератора равномерного распределения метод обеспечивает высокую точность.

# 22. Моделирование случайной величины, заданной плотностью распределения.

Моделирование случайной величины на основе её плотности распределения — это процесс генерации случайных чисел, которые следуют заданному вероятностному распределению. В отличие от метода обратной функции, который часто используется для дискретных или монотонных распределений, моделирование случайной величины на основе плотности распределения чаще применяется для непрерывных распределений.







Применение моделирования случайной величины по плотности распределения

* **Метод Монте-Карло**: Широко используется для численного интегрирования и оценки интегралов с использованием случайных чисел.
* **Симуляции**: Используются для исследования случайных процессов, прогнозирования, анализа рисков и других приложений.
* **Статистические исследования**: Помогают в анализе данных и проверке гипотез.

# 23. Методы повышения эффективности моделирования СВ.

Для повышения эффективности моделирования случайных величин (СВ) могут быть использованы различные методы и подходы, которые направлены на улучшение точности, скорости и ресурсоэффективности генерации случайных чисел. Вот несколько методов, которые помогают достичь этой цели:

1. Выбор подходящего алгоритма генерации случайных чисел

* **Качество генератора псевдослучайных чисел**: Использование хороших алгоритмов генерации, которые обеспечивают равномерное распределение и большой период.
* **Криптографические генераторы**: В критических приложениях, где требуется высокая степень случайности, используются специализированные криптографические генераторы случайных чисел.

2. Параллельные вычисления

* **Многопоточность и распределенные вычисления**: Использование многопоточности или распределенных систем для параллельной генерации большого количества случайных чисел. Это может значительно ускорить процесс моделирования, особенно при выполнении больших объемов вычислений.

3. Применение специализированных алгоритмов моделирования

* **Методы статистического моделирования**: Использование методов, таких как методы Монте-Карло или методы Марковских цепей, для моделирования сложных случайных процессов и систем.
* **Уточнение методов оценки и анализа**: Применение точных численных методов для улучшения оценки интегралов и статистических показателей, получаемых в ходе моделирования.

4. Оптимизация кода и алгоритмов

* **Профилирование и оптимизация**: Анализ производительности программного кода и его оптимизация для устранения узких мест и повышения эффективности работы с генерацией случайных чисел.
* **Использование быстрых алгоритмов**: Выбор быстрых и эффективных алгоритмов генерации случайных чисел и вычислений, которые могут ускорить выполнение моделирования.

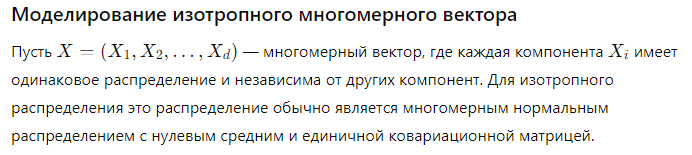
5. Предварительная обработка данных и кэширование

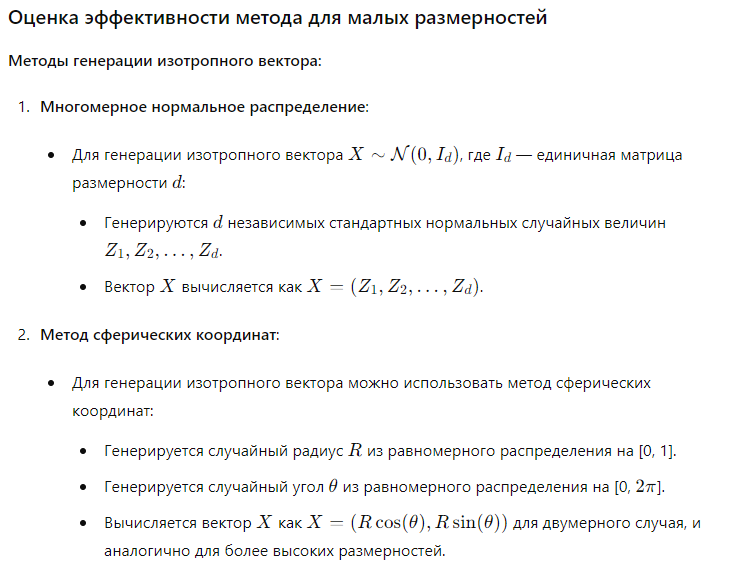
* **Кэширование результатов**: Хранение промежуточных результатов моделирования для повторного использования в случае многократного обращения к одним и тем же данным.
* **Оптимизация ввода-вывода**: Минимизация накладных расходов на чтение и запись данных в случае больших объемов операций с данными.

# 

# 24. Моделирование изотропного многомерного вектора. Оценка эффективности метода для малых размерностей.

Моделирование изотропного многомерного вектора представляет собой задачу генерации случайных величин в многомерном пространстве, где каждая компонента вектора независима и имеет одинаковое распределение. Изотропные распределения часто используются в статистике и вероятностных моделях, так как они предоставляют простой способ описания равномерно распределенных случайных величин в многомерном пространстве.





Оценка для малых размерностей:

* **Эффективность и точность метода** зависят от выбранного способа генерации и от размерности d вектора.
* **Многомерное нормальное распределение** обеспечивает высокую эффективность и точность даже для малых размерностей, так как генерация стандартных нормальных величин является быстрой и эффективной операцией.
* **Метод сферических координат** может быть менее эффективным для малых размерностей из-за необходимости вычисления тригонометрических функций, однако он предоставляет альтернативный подход для генерации изотропных векторов.

# 

# 25. Сети Петри. Моделирование параллельных процессов.

Сети Петри (Petri nets) — это математическая модель для описания параллельных и распределённых процессов. Они были разработаны Карлом Адамом Петри в 1962 году и нашли широкое применение в моделировании и анализе систем, включая параллельные вычисления, распределённые вычислительные сети, производственные процессы, программные системы и другие области.

Основные компоненты сетей Петри:

1. **Позиции (Places)**:
   * Позиции представляют собой состояния системы, где может храниться определённое количество маркеров (токенов). Они являются базовыми элементами модели.
2. **Переходы (Transitions)**:
   * Переходы определяют изменения состояния системы, когда выполняются определённые условия. Для активации перехода требуется определённое количество маркеров в связанных с ним позициях.
3. **Дуги (Arcs)**:
   * Дуги соединяют позиции с переходами и переходы с позициями, определяя поток маркеров в системе. Дуги могут быть направленными (от позиции к переходу или от перехода к позиции) и могут иметь веса, указывающие количество маркеров, необходимых для активации перехода.
4. **Маркеры (Tokens)**:
   * Маркеры представляют собой единицы состояния системы, которые перемещаются по сети от позиций к переходам и наоборот при активации переходов.

Принцип работы сетей Петри:

* **Активация переходов**: Переход активируется, когда все связанные с ним позиции содержат необходимое количество маркеров. После активации переход может переместить маркеры между позициями и изменить состояние системы.
* **Моделирование параллельности**: Сети Петри хорошо подходят для моделирования параллельных процессов благодаря своей способности представлять одновременное выполнение нескольких действий.

Применение сетей Петри:

* **Проектирование и анализ систем**: С помощью сетей Петри можно анализировать и предсказывать поведение системы, проверять на безопасность и пропускную способность.
* **Разработка программного обеспечения**: Сети Петри используются для моделирования параллельных алгоритмов и многопоточных приложений.
* **Производственные процессы и автоматизация**: Они помогают оптимизировать производственные процессы, планировать ресурсы и управлять логистикой.

Моделирование параллельных процессов с помощью сетей Петри предоставляет эффективный способ анализа и визуализации сложных взаимодействий между параллельными компонентами системы. Вот как сети Петри помогают в этом:

1. Представление параллелизма

Сети Петри позволяют чётко определить параллельные процессы в системе. Каждый переход в сети может представлять отдельный поток выполнения или параллельную задачу, а позиции — состояния, которые эти задачи могут занимать.

2. Управление ресурсами

Сети Петри могут моделировать и управлять ресурсами, необходимыми для параллельного выполнения процессов. Например, они могут отображать потоки данных или требования к доступу к общим ресурсам.

3. Анализ параллелизма

Анализ сетей Петри позволяет оптимизировать параллельные процессы, выявляя узкие места, конфликты за ресурсы, и потенциальные состояния гонки (race conditions).

4. Проверка безопасности и согласованности

Сети Петри могут использоваться для проверки безопасности параллельных систем, идентификации возможных дедлоков и живых блокировок, что помогает повысить надёжность системы.

5. Оптимизация производительности

Моделирование параллельных процессов позволяет исследовать и оптимизировать производительность системы, предсказывая её поведение в различных условиях и при различных нагрузках.

Примеры применения:

* **Программирование многопоточных приложений**: Сети Петри могут помочь разработчикам моделировать и анализировать взаимодействие между потоками в многопоточных приложениях.
* **Распределённые вычислительные системы**: Моделирование параллельных процессов в распределённых системах позволяет оптимизировать использование ресурсов и улучшить производительность.
* **Сети связи и коммуникаций**: Анализ параллельных процессов в сетевых системах помогает предотвратить перегрузки и оптимизировать пропускную способность.

# 26. Свойства сети Петри. Активность, достижимость, покрываемость, тупики.

1. Активность (Liveness)

Активность (или живость) сети Петри означает, что в системе всегда существует возможность выполнения переходов. То есть, если переход может быть активирован, то рано или поздно он будет активирован, если нет блокирующих условий.

2. Достижимость (Reachability)

Достижимость говорит о том, что из любого начального состояния сети Петри можно достичь любого другого состояния, выполняя последовательность действий (активаций переходов). Это свойство показывает, насколько система способна достичь различных состояний в процессе своей работы.

3. Покрытие (Coverability)

Покрытие связано с количеством маркеров (токенов) в различных состояниях сети Петри. Сеть Петри имеет конечное покрытие, если для каждого состояния существует конечное количество маркеров, которое достигается из любого начального состояния.

4. Тупики (Deadlocks)

Тупик (или дедлок) — это состояние, когда ни один из переходов не может быть активирован, так как ни одна из позиций не содержит достаточного количества маркеров для активации. Тупики могут возникать в сетях Петри, если не выполнены условия активации переходов или возникают конфликтные ситуации при доступе к ресурсам.

# 27. Расширенные сети Петри. Ингибиторные дуги.

Расширенные сети Петри (Extended Petri Nets) представляют собой расширенную версию классических сетей Петри, включающую дополнительные элементы для моделирования более сложных систем и взаимодействий. Одним из важных элементов расширенных сетей Петри являются ингибиторные дуги.

Ингибиторные дуги

Ингибиторная дуга в сети Петри — это тип дуги, который предотвращает активацию перехода, если выполнено определённое условие в связанной с ним позиции. Она указывает на то, что переход может быть активирован только в случае отсутствия маркера в позиции, с которой она связана.

Характеристики ингибиторных дуг:

1. **Наличие маркера в позиции**: Если в позиции, связанной с переходом ингибиторной дугой, есть хотя бы один маркер, то переход не может быть активирован. Таким образом, ингибиторные дуги представляют собой прямое отрицательное условие для активации перехода.
2. **Использование в моделировании**: Ингибиторные дуги полезны для моделирования условий, при которых выполнение некоторых действий зависит от отсутствия определённых ресурсов или условий. Например, в системе управления запасами ингибиторная дуга может представлять собой условие "не активировать заказ товара, если на складе ещё есть достаточное количество".

Пример использования ингибиторных дуг:

Предположим, у нас есть сеть Петри для управления производственным процессом, включающую следующие элементы:

* Позиции:
  + P1: "Доступность сырья"
  + P2: "Доступность рабочей силы"
* Переходы:
  + T1: "Запуск производства"

Чтобы активировать переход T1, необходимо, чтобы обе позиции P1 и P2 были маркированы (т.е. имели хотя бы по одному маркеру). Однако, если использовать ингибиторную дугу, то можно добавить дополнительное условие:

* Ингибиторная дуга:
  + Из P1 к T1 с условием "нет маркера"

Таким образом, если в P1 есть маркер, переход T1 будет заблокирован (неактивен), даже если в P2 есть маркер. Это позволяет учитывать условия, когда выполнение процесса зависит от отсутствия каких-то ресурсов или других условий.

# 28. Инструментальные среды для моделирования.

Инструментальные среды для моделирования (или программные инструменты для моделирования) – это программные пакеты, которые предоставляют функционал для создания, анализа и симуляции моделей систем. Они помогают инженерам, ученым и аналитикам разрабатывать модели, исследовать поведение систем и принимать обоснованные решения.

Основные инструментальные среды

1. **MATLAB/Simulink**
   * **Описание:** MATLAB является высокоуровневым языком программирования и интерактивной средой для численных вычислений, визуализации и программирования. Simulink - это надстройка MATLAB для моделирования, симуляции и анализа динамических систем.
   * **Применение:** Широко используется в инженерии, науке и экономике для моделирования и симуляции различных систем, включая механические, электрические и экономические модели.
2. **AnyLogic**
   * **Описание:** Универсальная среда для моделирования, которая поддерживает несколько парадигм моделирования: дискретно-событийное, агентное и системную динамику.
   * **Применение:** Используется для моделирования сложных систем, таких как логистические сети, системы массового обслуживания, производственные процессы и социальные системы.
3. **Arena**
   * **Описание:** Специализированный инструмент для дискретно-событийного моделирования, предоставляющий графический интерфейс для создания моделей.
   * **Применение:** Широко используется в промышленности и производстве для моделирования и оптимизации производственных процессов, логистики и систем обслуживания.
4. **Vensim**
   * **Описание:** Инструмент для системной динамики, который позволяет строить, симулировать и анализировать модели с обратными связями.
   * **Применение:** Применяется для моделирования бизнес-процессов, экономических систем, здравоохранения и экологических систем.
5. **SimEvents**
   * **Описание:** Надстройка MATLAB/Simulink для дискретно-событийного моделирования, позволяющая интегрировать события и временные задержки в модели.
   * **Применение:** Используется для моделирования систем массового обслуживания, логистических сетей и других дискретных систем.
6. **Python (с библиотеки SimPy и SciPy)**
   * **Описание:** Python - это язык программирования общего назначения, а SimPy и SciPy - это библиотеки для дискретно-событийного моделирования и научных вычислений соответственно.
   * **Применение:** Применяется для создания различных моделей, включая дискретно-событийные и статистические модели.
7. **GAMS (General Algebraic Modeling System)**
   * **Описание:** Высокоуровневый язык программирования и среда для математического моделирования, предназначенная для решения задач оптимизации.
   * **Применение:** Широко используется в экономике, логистике, энергетике и других областях для решения линейных, нелинейных и целочисленных задач оптимизации.

Критерии выбора инструментальной среды

* **Тип моделируемой системы:** Выбор среды зависит от того, какие аспекты системы нужно моделировать (динамические, дискретные, агентные и т.д.).
* **Уровень сложности модели:** Некоторые среды лучше подходят для простых моделей, другие для сложных и многокомпонентных систем.
* **Интерфейс и удобство использования:** Наличие графического интерфейса и поддержка визуального программирования могут значительно упростить процесс моделирования.
* **Поддержка и документация:** Наличие качественной документации, сообществ пользователей и технической поддержки является важным фактором при выборе среды.
* **Интеграция с другими инструментами:** Важно учитывать, насколько легко среда интегрируется с другими программными продуктами и инструментами, используемыми в организации.