

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«РОССИЙСКИЙ ГОСУДАРСТВЕННЫЙ
ПЕДАГОГИЧЕСКИЙ УНИВЕРСИТЕТ им. А. И. ГЕРЦЕНА»

**институт информационных технологий и технологического образования
кафедра информационных технологий и электронного обучения**

Основная профессиональная образовательная программа
Направление подготовки 09.03.01 Информатика и вычислительная техника
Направленность (профиль) «Технологии разработки программного обеспечения»
форма обучения – очная

Курсовая работа

по дисциплине «Технологии компьютерного моделирования»

Применение метода Монте-Карло для решения задач имитационного
моделирования, на примере задачи о случайном блуждании

Обучающейся 2 курса
Сорокиной Ирины Ивановны

Руководитель:
к.п.н, доцент
_____ Гончарова С. В.

« _____ » _____ 2021 г.

Санкт-Петербург
2021

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ	3
1 Имитационное моделирование	4
1.1 Имитационное моделирование как метод исследования	4
1.2 Изучение случайных процессов с применением методов Монте-Карло	6
1.3 Классическая задача о случайном блуждании	8
2 Компьютерная модель блужданий	11
2.1 Разработка программы для решения задачи	11
2.2 Проведение вычислительного эксперимента	12
ЗАКЛЮЧЕНИЕ	14
ЛИТЕРАТУРА	15
ПРИЛОЖЕНИЕ А. Стендовый доклад	16
ПРИЛОЖЕНИЕ Б. Презентация	17
ПРИЛОЖЕНИЕ В. Код программы	19

ВВЕДЕНИЕ

При изучении явлений окружающего мира мы часто сталкиваемся с процессами и явлениями, исход которых заранее предсказать нельзя. Эта непредсказуемость вызвана влиянием случайных факторов, воздействующих на ход процесса или явления.

Для исследования сложных систем, процессов, явлений, в которых случайные факторы играют существенную роль, очень часто невозможно построить аналитическую математическую модель, поэтому строят имитационную, при которой изучаемая система заменяется моделью, с достаточной точностью описывающей реальную систему, с которой проводятся эксперименты с целью получения информации об этой системе.

Актуальность этого метода познания заключается в возможности провести эксперимент в тех случаях, когда реальное взаимодействие невозможно или нецелесообразно. Также стоит отметить, что при использовании имитационной модели, для получения наиболее точного результата необходимо провести большое количество экспериментов, следовательно практически значимым является использование возможностей современной вычислительной техники. Можно сделать вывод - для изучения сложных систем, включающих в себя множество случайных факторов необходимо использовать методы статистических испытаний имитационного моделирования, реализованные средствами вычислительной техники, то есть программно.

Целью данной курсовой является решение задачи о случайном блуждании методом Монте-Карло в процессе имитационного моделирования. Для решения данной задачи необходимо изучить теорию по данному вопросу, создать программу имитирующую проведение реального эксперимента и провести эксперимент. Объект исследования - процесс имитационного моделирования, предмет исследования - задача о случайном блуждании.

1 Имитационное моделирование

1.1 Имитационное моделирование как метод исследования

Имитационное моделирование (ИМ) - один из самых мощных инструментов анализа при разработке сложных систем и анализа процессов их функционирования [5]. Использование этого инструмента позволяет провести эксперимент в тех случаях, когда реальное взаимодействие невозможно или нецелесообразно.

Этот метод предполагает использование технологий для имитации функционирования некой системы, описанной при помощи математических зависимостей. Чаще всего математические модели таких систем являются очень сложными и не поддаются обработке аналитическими методами. Моделирование позволяет собрать необходимую информацию о поведении системы путем создания ее компьютеризированной модели и дать оценку функциональным характеристикам исследуемого объекта.

Данный инструмент применяется в различных областях науки, техники и экономики, например при решении производственно-технологических задач: создании систем массового обслуживания, разработке систем связи, анализировании химических процессов.

Существует два типа имитационных моделей:

1. Непрерывные модели - используются для систем, поведение которых изменяется непрерывно во времени.
2. Дискретные модели - имеют дело с системами, поведение которых изменяется лишь в заданный момент времени.

Моменты времени в которые в системе происходят изменения, определяют события модели [2]. Интуитивно понятно, что для прогноза значений характеристик, развивающегося во времени процесса, необходимо знать значения этих характеристик в некоторый «начальный» момент времени.

На самом деле необходимость иметь значения внутренних характеристик в начальный момент времени для того, чтобы с помощью модели дать прогноз их значений в последующие моменты времени.

Имитационные модели относятся к классу моделей, которые являются системой соотношений между характеристиками описываемого процесса. Эти характеристики подразделяются на внутренние и внешние. Предполагается, что внутренние характеристики - это те, значения которых намереваются узнать с помощью средств ИМ; внешние - такие, от которых внутренние характеристики существенно зависят, но обратная зависимость не имеет места.

Модель должна быть замкнута, соотношения ее характеристик должны определять внутренние параметры, при известных внешних. Процедура определения внешних характеристик называется её идентификацией.

Этапы технологии имитационного моделирования [6]:

1. составление модели процесса;
2. проверка замкнутости и разработка процедуры вычисления внутренних характеристик по известным внешним характеристикам;
3. разработка компьютерной программы для вычисления внутренних характеристик по известным внешним характеристикам;
4. определение значений внешних характеристик;
5. выполнение имитационных экспериментов.

Таким образом, технология имитационного моделирования это способ извлечения новой информации, а именно значений внутренних характеристик модели, из той, которой уже располагаем — значений внешних характеристик[6].

Если в модели имеются случайные внешние характеристики, то и внутренние характеристики, и сам процесс являются случайными: в любой момент времени его характеристики это случайные величины. Имитационные эксперименты тогда воспроизводят реализации случайного процесса.

1.2 Изучение случайных процессов с применением методов Монте-Карло

Методы Монте-Карло - методы статистических испытаний, группа численных методов решения задач, использующих моделирование случайных величин и случайных процессов и построение на их основе статистических оценок искомых величин [10].

Основная идея состоит в использовании выборки случайных чисел для получения искомых оценок. Этот метод в известном смысле является предшественником современного имитационного моделирования.

Рассмотрим какой-нибудь естественный процесс, на протекание которого влияют различные случайные факторы. Законы распределения этих факторов будем считать известными: иногда эти законы можно вычислить теоретически, чаще - экспериментально. Так как компьютерные мощности позволяют нам находить значения случайных величин с любыми законами распределения, то, разыграв конкретные значения случайных факторов, мы сможем рассчитать конкретную случайную реализацию процесса. Такой расчет называют - имитацией и большинство реализаций метода Монте-Карло связано именно с этим.

Развитию методов способствовало бурное развитие ЭВМ. Алгоритмы Монте-Карло сравнительно легко программируются и позволяют рассчитывать многие задачи, недоступные для классических численных методов. Современный вариант метода сформировался в рамках Манхэттенского проекта, где он применялся для моделирования расстояний, которые могут пройти нейтроны в различных материалах. Идея моделирования на основе генерации набора случайных значений существовала уже в течение некоторого времени, но особое развитие получила при создании атомной бомбы, распространившийся затем во многих других областях знаний.

Областью применения группы методов Монте-Карло является способ оценки влияния неопределенности параметров системы в широком диапазоне

ситуаций. Метод обычно используют для оценки диапазона изменения результатов и относительной частоты значений в этом диапазоне для количественных величин, таких как стоимость, продолжительность, производительность, спрос [1].

Входными данными для моделирования методом Монте-Карло являются хорошо проработанная модель системы, информация о типе входных данных, источниках неопределенности и требуемых выходных данных. Входные данные и соответствующую им неопределенность рассматривают в виде случайных переменных с соответствующими распределениями. Часто для этих целей используют равномерные, треугольные, нормальные и логарифмически нормальные распределения.

Преимущества:

1. метод может быть адаптирован к любому распределению входных данных, включая эмпирические распределения, построенные на основе наблюдений за соответствующими системами;
2. модели относительно просты для работы и могут быть при необходимости расширены;
3. метод позволяет учесть любые воздействия и взаимосвязи, включая такие тонкие как условные зависимости;
4. для идентификации сильных и слабых влияний может быть применен анализ чувствительности;
5. модели являются понятными, а взаимосвязь между входами и выходами - прозрачной;
6. метод позволяет достичь требуемой точности результатов.

Недостатки:

1. точность решений зависит от количества итераций, которые могут быть выполнены (этот недостаток становится менее значимым с увеличением быстродействия компьютера);

2. большие и сложные модели могут представлять трудности для специалистов по моделированию и затруднять вовлечение заинтересованных сторон.

1.3 Классическая задача о случайном блуждании

Испытанием Бернулли называют случайный эксперимент с двумя возможными элементарными исходами: ω_1, ω_2 . Рассмотрим случайное блуждание, которое порождается схемой испытаний Бернулли. Предположим, что частица выходит из начала координат и через единицу времени перемещается на единицу вверх с вероятностью $p, 0 < p < 1$, или на единицу вниз с вероятностью $q = 1 - p$ (рис. 1).

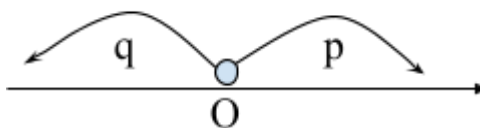


Рисунок 1. Несимметричное блуждание частицы

В каждый последующий момент времени повторяется та же история независимо от предыдущего положения частицы. Таким образом, за время n частица проходит некоторый путь, который можно изобразить графически. Всего путей движения частицы за время n будет 2^n , однако теперь эти пути нельзя считать равновероятными.

Последовательность перемещений частицы за время n может быть рассмотрена как последовательность n независимых испытаний с двумя исходами: движение вверх на единицу - “1”, движение вниз на единицу - “-1”.

Для иллюстрации метода Монте – Карло рассмотрим классическую задачу о случайном блуждании: “В городе в полночь из зоопарка сбежал крокодил, направление передвижения он выбирал случайным образом, так как

ему все равно куда идти. За час он проходил один квартал, блуждал он до 5 часов утра, потом началось движение транспорта, и крокодил спрятался. Определить район поиска крокодила”.

Это чисто математическая задача, ее решением является число. Если бы крокодил, выйдя из зоопарка, все время шел в одном направлении, то он ушел бы на расстояние пяти кварталов от зоопарка, в этом случае решение задачи находится сразу: «Крокодила нужно искать в пяти ближайших кварталах от зоопарка». Но крокодил выбирал направление движения случайным образом, поэтому, неизвестно, как далеко он мог уйти.

Будем решать эту задачу методом Монте – Карло. Искомой величиной X является число кварталов для поиска крокодила, это число зависит от такого случайного фактора, как выбор направления через каждый час блужданий. Экспериментом или испытанием в данной задаче является побег крокодила, очевидно, что при каждом побеге крокодил может спрятаться в разных местах города, поэтому значение X будет разным. И чем больше число испытаний, то есть, чем больше побегов мы отследим, тем точнее будет ответ.

Для использования этого метода нужно получить набор случайных чисел и связать с каждым числом такое случайное событие, как выбор направления движения. Будем определять случайные числа при помощи генератора случайных чисел с диапазоном значений от 1 до 36 и поставим в соответствие каждому числу, участвующему в розыгрыше, какое-либо направление движения (Таблица № 1).

Таблица №1

Сгенерированное число									Направление
1	5	9	13	17	21	25	29	33	Север
2	6	10	14	18	22	26	30	34	Юг
3	7	11	15	19	23	27	31	35	Запад
4	8	12	16	20	24	28	32	36	Восток

Сгенерируем 6 наборов по 5 значений и сопоставим числам направления движения (Таблица №2).

Таблица №2

№	Сгенерированные наборы	Направления движения
1	9, 15, 12, 8, 6	север, восток, восток, восток, юг
2	34, 11, 4, 5, 7	юг, запад, восток, север, запад
3	14, 29, 9, 6, 8	юг, север, север, юг, восток
4	22, 31, 32, 12, 1	юг, запад, восток, восток, север
5	16, 20, 32, 22, 18	восток, восток, восток, юг, юг
6	3, 13, 36, 11, 6	запад, север, восток, запад, юг

В третьем столбце Таблицы №2 составлена словесная, состоящая из направлений движения, имитационная модель шести вариантов блужданий крокодила. Уже из нее ясно, что крокодила надо искать в ближайших 3 кварталах от зоопарка, так как самый «длинный» пробег крокодила «восток-восток- восток» имеется в первом и пятом испытаниях.

2 Компьютерная модель блужданий

2.1 Разработка программы для решения задачи

Создадим программу, в основе которой будет лежать метод статистических испытаний, с использованием генератора случайных чисел.

Представим формальное описание алгоритма работы программы: пользователь загружает входные данные - время, прошедшее с побега животного из зоопарка и количество проводимых экспериментов, запускается цикл отсчитывающий количество экспериментов, затем запускается алгоритм проведения каждого отдельного испытания.

Создается переменная хранящая координаты животного в каждый момент времени, по умолчанию, на момент старта каждого из испытаний координаты животного, соответствуют координатам зоопарка $x, y = 0, 0$.

Далее запускается функция случайного выбора дальнейшего направления движения, используя полученное случайное значение функции координаты животного меняются, после окончания единичного испытания мы имеем координаты точки в которой оказался крокодил на текущий момент времени.

Между координатами по x и y выбирается абсолютное максимальное значение и записывается в переменную счетчик, после повторения эксперимента заданное количество раз, счетчик хранит сумму значений максимального удаления сбежавшего животного от зоопарка, разделим значение переменной счетчика на количество испытаний, округлим до целого и получим число кварталов на которое в среднем убегал крокодил за заданное время, это и будет результатом работы.

Для компьютерной реализации используем язык Python v3.8 и библиотеки данного языка: “random” - для выбора случайного направления движения, “matplotlib” - для создания графика (карты) блужданий,

“PySimpleGUI” - для создания интерфейса пользовательского приложения. Листинг кода программы приведен в приложении Д.

Так как данная программа, подразумевает ввод пользователем значений, необходимо создать пользовательский интерфейс, макет интерфейса представлен на рисунке 2.

01 Сколько времени прошло?

Часов: Минут:

02 Сколько раз вы хотите провести эксперимент?

Повторить раз.

Сохранить **Запустить**

Карта с направлениями

Результат эксперимента

Рисунок 2. Макет пользовательского интерфейса

2.2 Проведение вычислительного эксперимента

Решим поставленную задачу, запустим программу и введем время блуждания 5 часов, изначально число испытаний будет равно 100. Изучим полученный результат на рисунке 3.

В ходе проведения 100 экспериментов, путем статистических испытания выяснили, что вероятнее всего, крокодил спрятался в радиусе 2 кварталов от зоопарка, это и есть искомый ответ.

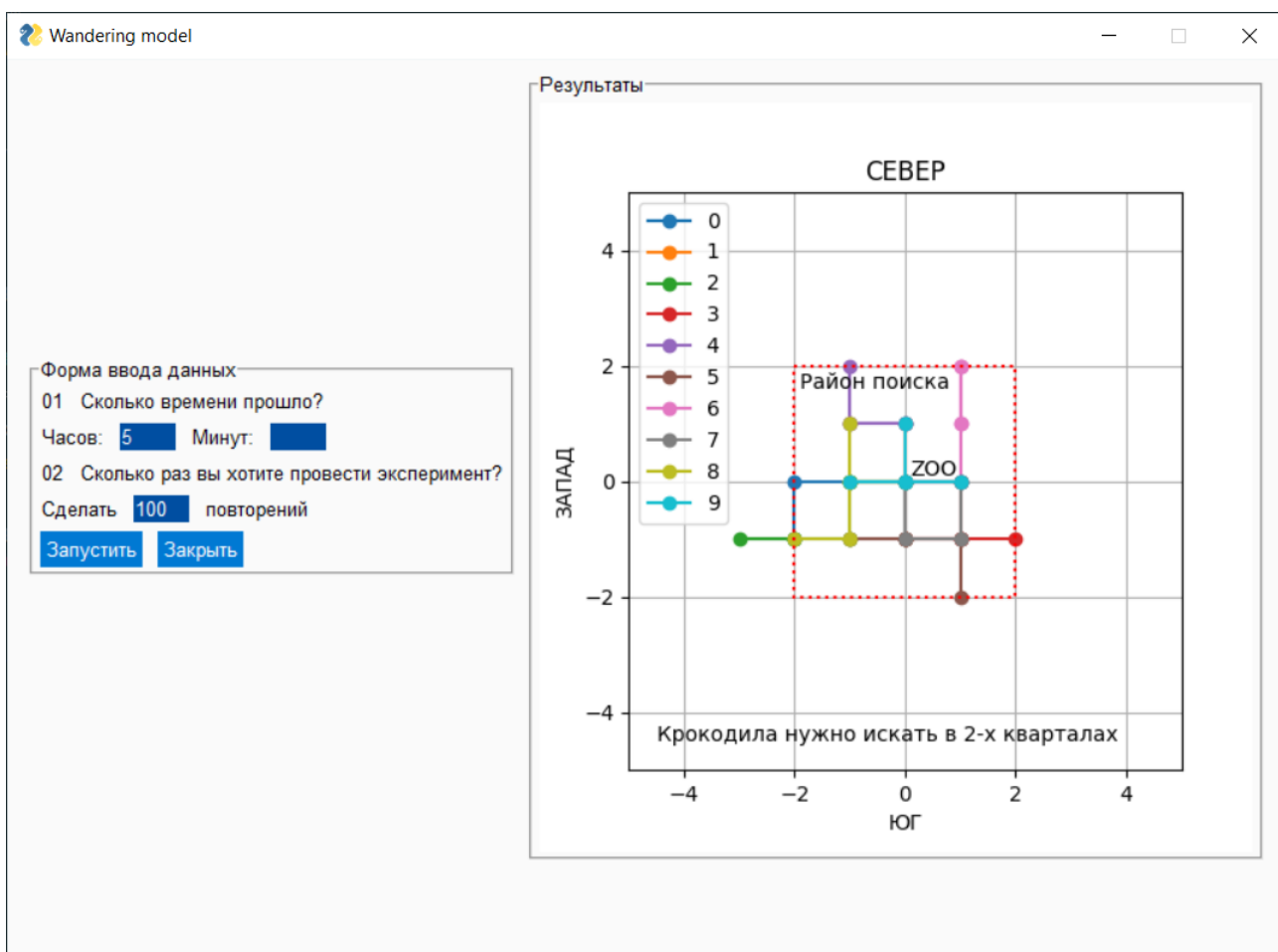


Рисунок 3. Решение задачи

Программа “Wandering model” строит компьютерную модель блужданий крокодила, основываясь на данных введенных пользователем. Результатом работы программы является графическая модель, в которой схематично изображены «траектории» блужданий крокодила по кварталам города. Каждая траектория задана пунктирной линией определенного цвета, кружком обозначены места, где крокодил менял направление блуждания. Точечной красной линией обведен квадратный район поиска крокодила. Число испытаний в программе можно задавать любое, но на модели для наглядности отображаются «траектории» только первых 10-ти блужданий, чтобы не перегружать рисунок. Не все места блужданий попадают в квадрат, это естественно, так как метод Монте-Карло обрабатывает данные статистически и район поиска определяется как среднее значение всех вариантов побегов.

ЗАКЛЮЧЕНИЕ

В ходе работы была реализована программа, моделирующая сложный процесс несимметричного блуждания, в основу реализации был положен метод статистических испытаний Монте-Карло. В процессе была получена дискретная имитационная модель, так как подразумевалось проведение эксперимента на определенном отрезке времени, а не постоянное наблюдение.

Изначально были идентифицированы внутренние характеристики системы - время блуждания и частота смены направления движения, внешние характеристики необходимо было выяснить в ходе решения задачи. После прохождения всех этапов технологии имитационного моделирования была составлена модель, которая легла в основу программной реализации.

Следующим этапом работы стала реализация метода Монте-Карло, необходимо было случайным образом определять направление, учитывать эти данные на большом количестве распределений и выдавать среднее значение от общего количества испытаний, реализовав данный метод, перешли к построению графика (карты) блужданий.

По итогу удалось составить компьютерную модель испытания Бернулли, вместо классических 2-х исходов (вправо, влево), полученная модель обладала 4-мя исходами, а частица перемещается на плоскости x, y .

Был проведен вычислительный эксперимент с применением возможностей вычислительной техники, так как в реальной жизни подобный опыт реализуем маловероятно. В ходе проведения эксперимента было найдено решение поставленной задачи.

В дальнейшем возможно доработка модели, добавление таких внутренних параметров как: скорость движения частицы, усложнение поведения, фактор состояния системы в конкретный момент времени.

ЛИТЕРАТУРА

1. Алябьева, Е. В. Имитационное моделирование : учебно-методическое пособие / Е. В. Алябьева. — Барнаул : АлтГПУ, 2016. — 48 с. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/112166> (дата обращения: 27.03.2021). — Режим доступа: для авториз. пользователей.
2. Введение в исследование операций. 6-е издание. : Пер. с англ. - М. Издательский дом "Вильямс", 2001. - 912с.: ил. - Парал. тит. англ.
3. Горностаева Т. Н., Горностаев О. М., Математическое и компьютерное моделирование. Учебное пособие – М.: Мир науки, 2019. – Сетевое издание. Режим доступа: <https://izd-mn.com/PDF/50MNNPU19.pdf>
4. Гущина А. А., Кемалов Б. К. К вопросу об имитационном моделировании //НиКа.2009.№.URL:<https://cyberleninka.ru/article/n/k-voprosu-ob-imitatsionnom-modelirovanii> (дата обращения: 27.03.2021).
5. Имитационное моделирование : учебное пособие / В. П. Строгалева, И. О. Толкачева. — 3-е изд. — Москва : Издательство МГТУ им. Н. Э. Баумана, 2017. — 295, [1] с. : ил.
6. Имитационное моделирование : учеб. пособие для студ. высш. учеб. заведений/Ю.Н. Павловский, Н.В. Белотелов, Ю. И. Бродский. - М. : Издательский центр "Академия", 2008. - 236 с.
7. Крупский Александр Юльевич, Монахов Анатолий Дмитриевич Инновации в имитационном моделировании // Вестник евразийской науки.2011.№2(7).URL:<https://cyberleninka.ru/article/n/innovatsii-v-imitatsionnom-modelirovanii> (дата обращения: 27.03.2021).
8. Пимонов, А. Г. Имитационное моделирование : учебное пособие / А. Г. Пимонов, С. А. Веревкин, Е. В. Прокопенко. — Кемерово : КузГТУ имени Т.Ф. Горбачева, 2015. — 139 с. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/69500> (дата обращения: 27.03.2021). — Режим доступа: для авториз. пользователей.
9. Плютенко А.С. Имитационное моделирование // Вестник МГУП. 2011. №1. URL: <https://cyberleninka.ru/article/n/imitatsionnoe-modelirovanie> (дата обращения: 27.03.2021).
10. Численные методы Монте-Карло/И. М. Соболев - Москва : Издательство "Наука", 1973. - 308 с.

ПРИМЕНЕНИЕ МЕТОДА МОНТЕ-КАРЛО ДЛЯ РЕШЕНИЯ ЗАДАЧ ИМИТАЦИОННОГО МОДЕЛИРОВАНИЯ

на примере задачи о случайном блуждании

ИМИТАЦИОННОЕ МОДЕЛИРОВАНИЕ (ИМ)

один из самых мощных инструментов анализа при разработке сложных систем и анализа процессов их функционирования. Использование этого инструмента позволяет провести эксперимент в тех случаях, когда реальное взаимодействие невозможно или нецелесообразно.



МЕТОДЫ МОНТЕ-КАРЛО

методы статистических испытаний, группа численных методов решения задач, использующих моделирование случайных величин и случайных процессов и построение на их основе статистических оценок искомых величин. Этот метод в известном смысле является предшественником современного имитационного моделирования.

ЗАДАЧА

Для иллюстрации метода Монте – Карло рассмотрим классическую задачу о случайном блуждании: "В городе в полночь из зоопарка сбежал крокодил, направление передвижения он выбирал случайным образом, так как ему все равно куда идти. За час он проходил один квартал, блуждал он до 5 часов утра, потом началось движение транспорта, и крокодил спрятался. Определить район поиска крокодила".



РЕШЕНИЕ



Будем решать эту задачу методом Монте – Карло. Искомой величиной X является число кварталов для поиска крокодила, это число зависит от такого случайного фактора, как выбор направления через каждый час блужданий. Экспериментом или испытанием в данной задаче является побег крокодила, очевидно, что при каждом побеге крокодил может спрятаться в разных местах города, поэтому значение X будет разным. И чем больше число испытаний, то есть, чем больше побегов мы отследим, тем точнее будет ответ.

СТУДЕНКА 2 ИВТ (1)/1
СОРОКИНА ИРИНА

ПРИЛОЖЕНИЕ Б. Презентация

Автор: Сорокина Ирина 1 июня 2021 года

ПРИМЕНЕНИЕ МЕТОДА МОНТЕ-КАРЛО ДЛЯ РЕШЕНИЯ ЗАДАЧ ИМИТАЦИОННОГО МОДЕЛИРОВАНИЯ

на примере задачи о случайном блуждании

Автор: Сорокина Ирина 1 июня 2021 года

ПОВЕСТКА ДНЯ

- 1 Изучение теоретического вопроса
- 2 Испытание Бернулли
- 3 Решение задачи
- 4 Демонстрация программы

Автор: Сорокина Ирина 1 июня 2021 года

ИМИТАЦИОННОЕ МОДЕЛИРОВАНИЕ (ИМ)

один из самых мощных инструментов анализа при разработке сложных систем и анализа процессов их функционирования. Использование этого инструмента позволяет провести эксперимент в тех случаях, когда реальное взаимодействие невозможно или нецелесообразно.


Существует два типа имитационных моделей:

- 1 Непрерывные модели - используются для систем, поведение которых изменяется непрерывно во времени.
- 2 Дискретные модели - имеют дело с системами, поведение которых изменяется лишь в заданный момент времени.

Автор: Сорокина Ирина 1 июня 2021 года

Этапы технологии имитационного моделирования:

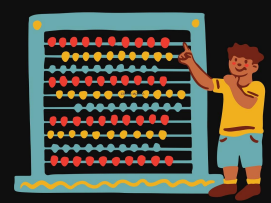
- 1 составление модели процесса;
- 2 проверка замкнутости и разработка процедуры вычисления внутренних характеристик по известным внешним характеристикам;
- 3 разработка компьютерной программы для вычисления внутренних характеристик по известным внешним характеристикам;
- 4 определение значений внешних характеристик; выполнение имитационных экспериментов.



Автор: Сорокина Ирина 1 июня 2021 года


МЕТОДЫ МОНТЕ-КАРЛО

Основная идея состоит в использовании выборки случайных чисел для получения искомых оценок. Этот метод в известном смысле является предшественником современного имитационного моделирования.



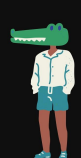
Автор: Сорокина Ирина 1 июня 2021 года

Испытанием Бернулли называют случайный эксперимент с двумя возможными элементарными исходами: и1, и2. Рассмотрим случайное блуждание, которое порождается схемой испытаний Бернулли. Предположим, что частица выходит из начала координат и через единицу времени перемещается на единицу вверх с вероятностью $p, 0 < p < 1$, или на единицу вниз с вероятностью $q=1-p$.



1 Что сделать

Для иллюстрации метода Монте – Карло рассмотрим классическую задачу о случайном блуждании: "В городе в полном из зоопарка сбежал крокодил, направление передвижения он выбирал случайным образом, так как ему все равно куда идти. За час он проходил один квартал, блуждал он до 5 часов утра, потом началось движение транспорта, и крокодил спрятался. Определить район поиска крокодила".



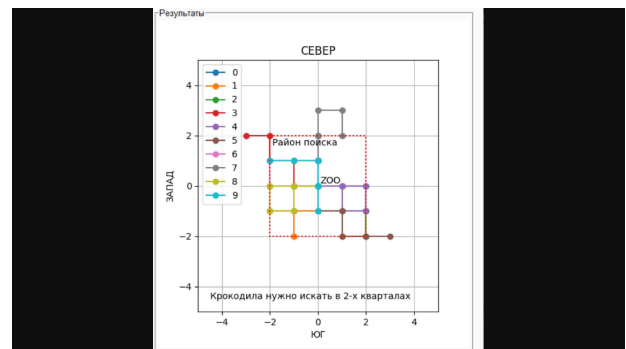
2 В процессе

Будем решать эту задачу методом Монте – Карло. Искомой величиной X является число кварталов для поиска крокодила, это число зависит от такого случайного фактора, как выбор направления через каждый час блужданий. Экспериментом или испытанием в данной задаче является побег крокодила, очевидно, что при каждом побеге крокодил может спрятаться в разных местах города, поэтому значение X будет разным. И чем больше число испытаний, то есть, чем больше побегов мы отследим, тем точнее будет ответ.



3
Готово
→

Форма ввода данных:
01 Сколько времени прошло?
Часов: 5 Минут:
02 Сколько раз вы хотите провести эксперимент?
Сделать 10000 повторений
Запустить Закреть



ПРИЛОЖЕНИЕ В. Код программы

Файл “main.py”

```
import matplotlib.pyplot as plt
import random

# функция выбора направления движения
def choosingWay():
    way_list = ['north', 'south', 'west', 'east']
    return random.choice(way_list)

# функция построения пути
def moving(time):
    x, y = 0, 0
    for _ in range(time):
        way = choosingWay()
        if way == 'north':
            y += 1
        elif way == 'south':
            y -= 1
        elif way == 'west':
            x -= 1
        else:
            x += 1
    return max(abs(x), abs(y))

# функция проведения эксперимента
def experiment(numberOfExp, pastTime):
    s = 0
    for _ in range(numberOfExp):
        s += moving(pastTime)
```

```

s = round(s / numberOfExp)
result = 'Вероятнее всего крокодила нужно искать в '
+ str(s) + '-х кварталах'

```

```

d = []
for __ in range(10):
    t_x, t_y = [0], [0]
    x, y = 0, 0
    for _ in range(pastTime):
        way = choosingWay()
        if way == 'north':
            y += 1
        elif way == 'south':
            y -= 1
        elif way == 'west':
            x -= 1
        else:
            x += 1
        t_x.append(x)
        t_y.append(y)
    d.append(t_x)
    d.append(t_y)
fig = draw_map(pastTime, s, d)
# fig.show()
return fig

```

```

def draw_map(n, s, d):
    fig, ax = plt.subplots(figsize=(5, 5))
    ax.plot(*d, marker='o')

```

```

    ax.plot((-s, s, s, -s, -s), (s, s, -s, -s, s),
linestyle='dotted', color='red')
    ax.set_xlim(-n, n)
    ax.set_ylim(-n, n)

    ax.set_title('СЕВЕР')
    ax.set_xlabel('ЮГ')
    ax.set_ylabel('ЗАПАД')

    ax.legend('0123456789', loc='upper left')

    ax.grid()

    ax.annotate('ZOO', xy=(0, 0), xytext=(0.1, 0.1))
    ax.text(-s + 0.1, s - 0.4, 'Район поиска')
    ax.text(-n + 0.5, -n + 0.5, 'Крокодила нужно искать в
' + str(s) + '-х кварталах')
    return fig

```

Файл “interface.py”

```

import PySimpleGUI as sg
import matplotlib
from matplotlib.backends.backend_tkagg import
FigureCanvasTkAgg
from main import *

matplotlib.use('TkAgg')

```

```

def draw_figure(canvas, figure):
    figure_canvas_agg = FigureCanvasTkAgg(figure, canvas)
    figure_canvas_agg.draw()
    figure_canvas_agg.get_tk_widget().pack(side='top',
fill='both', expand=1)
    return figure_canvas_agg

def delete_fig_agg(agg):
    agg.get_tk_widget().forget()
    plt.close('all')

sg.theme('Material2')

frame_left = [
    [sg.Text('01'), sg.Text('Сколько времени прошло?')],
    [sg.Text('Часов:'), sg.InputText(size=(5, 1),
key='hours'), sg.Text('Минут:'),
    sg.InputText(size=(5, 1), key='minutes')],
    [sg.Text('02'), sg.Text('Сколько раз вы хотите
провести эксперимент?', justification='center')],
    [sg.Text('Сделать'), sg.InputText(size=(5, 1),
key='number'), sg.Text('повторений')],
    [sg.Submit('Запустить'), sg.Cancel('Закрыть')]
]

frame_right = [
    [sg.Canvas(key='-CANVAS-', size=(450, 650))]
]

layout = [

```

```

[sg.Frame('Форма ввода данных', frame_left),
 sg.Frame('Результаты', frame_right)],
]

# Create the Window
window = sg.Window('Wandering model', layout,
finalized=True, size=(850, 600))

# Event Loop to process "events" and get the "values" of
the inputs
fig_agg = None
while True:
    event, values = window.read()
    if event == sg.WIN_CLOSED or event == 'Заккрыть': #
if user closes window or clicks cancel
        break
    if event == 'Запустить':
        pastTime = int(values['hours'])
        numberOfExp = int(values['number'])
        fig = experiment(numberOfExp, pastTime)

        if fig_agg is not None:
            delete_fig_agg(fig_agg)
        canvas_elem = window['-CANVAS-'].TKCanvas
        canvas_elem.Size = (100, 100)
        fig_agg = draw_figure(canvas_elem, fig)

window.close()

```