Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Пермский национальный исследовательский политехнический университет»

Факультет: Прикладной математики и механики
Кафедра: Вычислительной математики, механики и биомеханики

Направление: 09.04.02 Информационные технологии и системная инженерия
Профиль: «Информационные технологии и системная инженерия»

КУРСОВАЯ РАБОТА

по дисциплине

«Математическое моделирование и 3D-визуализация сложных систем»

Выполнил
студент гр. ИТСИ-24-1м
Слаутин Максим Егорович

Принял Преподаватель кафедры ВММБ
Онискив Владимир Дмитриевич

Пермь 2025

Содержание

Лабораторная работа №1

Задание

Дано пространство: D = [0,10] × [0,10] × [0,10], а также набор функций, которые описываются следующими уравнениями:

1) $y_1 = (x_1 - 15)^2 + (x_2 - 3)^2 + x_3$

2) $y_2 = 2{x_1}^2 + 2x_1 + 4x_2 - 3x_3$

3) $y_3 = \frac{1}{3}(x_1 + 1)^3 + x_2 - 2x_1x_2 + {x_3}^3$

4) $y_4 = \frac{5}{2x_1} + \frac{9}{x_2} + 2x_1 + x_2$

Необходимо найти наиболее оптимальное решение, используя алгоритм Поиска с возвратом. Для решения была выбрана первая формула.

Теория

Поиск с возвратом — это метод решения задач, который заключается в последовательном переборе возможных вариантов с возвратом к предыдущим шагам, если текущий путь не приводит к решению. Этот подход полезен для задач, где нужно найти все возможные комбинации, удовлетворяющие заданным условиям. Поиск с возвратом работает по следующим пунктам:

Пошаговое построение решения: алгоритм начинает с пустого решения и на каждом шаге добавляет новый элемент, расширяя текущий вариант.

Проверка допустимости: после каждого добавления проверяется, удовлетворяет ли текущая частичная конфигурация условиям задачи. Если нет — алгоритм отказывается от этого пути.

Возврат: если дальнейшее развитие текущего варианта невозможно, алгоритм возвращается на предыдущий шаг и пробует альтернативные варианты.

Завершение: процесс продолжается, пока не будут найдены все возможные решения или пока не будет доказана их невозможность.

Результат

```
Стартовая точка: Point(0, 4, 9), её значение: 235
Промежуточная точка 0: Point(0, 4, 9), её значение: 235
Промежуточная точка 1: Point(0, 4, 9), её значение: 235
Промежуточная точка 2: Point(0, 4.0005543046303425, 8.999968390570642), её значение:
235.00107730708496
Промежуточная точка 3: Point(0, 4.0005543046303425, 8.999968390570642), её значение:
235.00107730708496
Промежуточная точка 4: Point(0, 4.0005152359016884, 9.000611981482544), её значение:
235.00164271875394
…
Промежуточная точка 99999: Point(0, 10, 10), её значение: 284
Конечная точка: Point(0, 10, 10), её значение: 284

Point(0, 10, 10)
```

Вывод

В ходе выполнения работы был реализован метод Поиска с возвратом, который проводит 100.000 попыток нахождения лучшей точки для решения выбранного уравнения.

Лабораторная работа №2

Задание

Имеются следующие координаты точек:

N	1	2	3	4	5	6	7	8	9	10
X	1	3	5	4	3	8	9	6	6	9
Y	1	7	2	5	3	4	1	8	4	7

Для выбранных точек построить оптимальный маршрут, в котором стартовой точкой является точка 1, а конечной точкой будет являться 10. В качестве результатов исследования получить оптимальную траекторию и длину траектории для 3 разных θ (тета)

Теория

Алгоритм имитации отжига — это вероятностный метод оптимизации, вдохновленный процессом отжига металлов. Он позволяет избегать застревания в локальных минимумах, постепенно уменьшая вероятность принятия худших решений. Основные этапы алгоритма:

Инициализация: задается начальный маршрут. Вычисляется его длина.

Генерация нового решения: производится случайная перестановка двух точек в маршруте. Вычисляется новая длина пути.

Критерий принятия решения: если новый маршрут короче, он принимается. Если длиннее, он может быть принят с подсчитанной вероятностью

Охлаждение (уменьшение θ): с течением итераций θ уменьшается, снижая вероятность принятия худших решений.

Остановка: алгоритм завершается после достижения заданного числа итераций или стабилизации результата.

Результат

```
Исходный маршрут:
Сумма: 51.227776731883786, точки: [Point #1 (1, 1), Point #2 (3, 7), Point #3 (5, 2), Point #4 (4, 5), Point #5 (3, 3), Point #6 (8, 4), Point #7 (9, 1), Point #8 (6, 8), Point #9 (6, 4), Point #10 (9, 7)]
Сумма: 49.959709113136306, точки: [Point #1 (1, 1), Point #2 (3, 7), Point #3 (5, 2), Point #4 (4, 5), Point #5 (3, 3), Point #8 (6, 8), Point #7 (9, 1), Point #6 (8, 4), Point #9 (6, 4), Point #10 (9, 7)]
Сумма: 51.97363315653524, точки: [Point #1 (1, 1), Point #8 (6, 8), Point #3 (5, 2), Point #4 (4, 5), Point #5 (3, 3), Point #2 (3, 7), Point #7 (9, 1), Point #6 (8, 4), Point #9 (6, 4), Point #10 (9, 7)]
Сумма: 42.88480141957296, точки: [Point #1 (1, 1), Point #5 (3, 3), Point #3 (5, 2), Point #4 (4, 5), Point #8 (6, 8), Point #2 (3, 7), Point #7 (9, 1), Point #6 (8, 4), Point #9 (6, 4), Point #10 (9, 7)]
Сумма: 42.88480141957296, точки: [Point #1 (1, 1), Point #5 (3, 3), Point #3 (5, 2), Point #4 (4, 5), Point #8 (6, 8), Point #2 (3, 7), Point #7 (9, 1), Point #6 (8, 4), Point #9 (6, 4), Point #10 (9, 7)]
Сумма: 42.567465784132665, точки: [Point #1 (1, 1), Point #5 (3, 3), Point #3 (5, 2), Point #7 (9, 1), Point #8 (6, 8), Point #2 (3, 7), Point #4 (4, 5), Point #6 (8, 4), Point #9 (6, 4), Point #10 (9, 7)]
Сумма: 42.567465784132665, точки: [Point #1 (1, 1), Point #5 (3, 3), Point #3 (5, 2), Point #7 (9, 1), Point #8 (6, 8), Point #2 (3, 7), Point #4 (4, 5), Point #6 (8, 4), Point #9 (6, 4), Point #10 (9, 7)]
Сумма: 39.600065109063884, точки: [Point #1 (1, 1), Point #5 (3, 3), Point #3 (5, 2), Point #7 (9, 1), Point #8 (6, 8), Point #2 (3, 7), Point #4 (4, 5), Point #9 (6, 4), Point #6 (8, 4), Point #10 (9, 7)]
…
Сумма: 37.14656966336835, точки: [Point #1 (1, 1), Point #5 (3, 3), Point #4 (4, 5), Point #2 (3, 7), Point #8 (6, 8), Point #9 (6, 4), Point #3 (5, 2), Point #7 (9, 1), Point #6 (8, 4), Point #10 (9, 7)]
Сумма: 37.14656966336835, точки: [Point #1 (1, 1), Point #5 (3, 3), Point #4 (4, 5), Point #2 (3, 7), Point #8 (6, 8), Point #9 (6, 4), Point #3 (5, 2), Point #7 (9, 1), Point #6 (8, 4), Point #10 (9, 7)]



Лучший маршрут: 1, 5, 4, 2, 8, 9, 3, 7, 6, 10
```

```
Сумма: 37.14656966336835, точки: [Point #1 (1, 1), Point #5 (3, 3), Point #4 (4, 5), Point #2 (3, 7), Point #8 (6, 8), Point #9 (6, 4), Point #3 (5, 2), Point #7 (9, 1), Point #6 (8, 4), Point #10 (9, 7)]
Лучшее значение тета = 74 среди: [50, 63, 74]
```

Вывод

В ходе выполнения лабораторной работы была написана программа, которая случайным образом меняет две точки маршрута и сравнивает результаты, с предыдущими значением, а после принимает решение о выборе нового маршрута в качестве текущего.

Лабораторная работа №3

Задание

Оптимальное заполнение одномерного ранца (Метод Отжига)

10 предметов

Z	1	2	3	4	5	6	7	8	9	10
W	4	6	2	8	9	1	3	5	4	3
C	5.0	4.5	6.0	4.0	3.5	7.0	4.0	3.0	6.5	5.5

W_{max}=32 кг из чего следует, что $\sum x_i W_i \leq 32$, где x_i= {0;1} (есть или нет), а $\sum \mathrm{C}_i W_i \rightarrow max$

Из заданного набора предметов Z каждый имеет свой вес W и ценность C. Рюкзак наполняется, так чтобы общая ценность была максимальной, а вес ≤ 32 килограммов.

Теория

Метод отжига — это вероятностный алгоритм оптимизации, для поиска глобального оптимума в задачах, где возможны множественные локальные экстремумы. Метод начинает с высокой «температуры», позволяя системе принимать как улучшающие, так и ухудшающие решения, затем постепенно «остывает», снижая вероятность принятия плохих вариантов. Основные этапы работы:

Генерация начального решения.

Итеративное улучшение: соседние решения создаются небольшими изменениями текущего. Лучшие решения принимаются всегда, худшие — с вероятностью, зависящей от температуры и степени ухудшения.

Схема охлаждения: температура снижается по заданному правилу.

Результат

```
Максимальная ценность: 152.50
Вес набора: 31
Оптимальный набор x: [1, 1, 1, 1, 0, 1, 1, 0, 1, 1]
```

Вывод

В ходе выполнения работы был реализован метод Отжига, благодаря которому нашлись наилучшие значения для максимизации выгоды с учетом весов каждого предмета.

Лабораторная работа №4

Задание

Выполнить 5 видов селекций, а именно:

1) Рулеточная (14 в 14)
2) Стохастическая универсальная (14 в 14)
3) Ранжированная селекция
4) Селекция усечения (14 в x при L=0.4)
5) Турнирный отбор (14 в 10)

Для написанных данных:

1	2	3	4	5	6	7	8	9	10	11	12	13	14
4.0	3.3	9.0	5.0	2.0	8.0	2.1	12.0	7.1	4.6	5.8	9.1	1.3	8.2

Теория

Селекция — это процесс выбора лучших особей (решений) в популяции для дальнейшего размножения или использования в алгоритмах оптимизации (например, генетических алгоритмах). Применяется в **искусственном интеллекте, машинном обучении, биологии, сельском хозяйстве** и других областях. Рассматриваемые 5 видов селекции:

Рулеточная: особи выбираются с вероятностью, пропорциональной их приспособленности. Чем выше качество особи, тем больше шансов быть выбранной.

Стохастическая универсальная: улучшенная рулеточная селекция: выбирается несколько особей за один "прогон" колеса рулетки, что уменьшает случайность и ускоряет сходимость.

Ранжированная: особи сортируются по приспособленности, и вероятность выбора зависит от их ранга, а не от абсолютного значения. Уменьшает влияние сверхуспешных особей.

Селекция усечения: отбираются только лучшие особи (например, топ-40% популяции), остальные отбрасываются. Просто и эффективно, но снижает разнообразие.

Турнирный отбор: случайно выбирается несколько особей (например, 2-3), и среди них отбирается лучшая. Позволяет балансировать между давлением отбора и разнообразием.

Каждый метод имеет свои преимущества и применяется в зависимости от задачи.

Результат

```
Вероятность фитнес от общей суммы
[0.049079754601226995,          0.040490797546012265,          0.11042944785276074,
0.06134969325153374,           0.024539877300613498,           0.09815950920245399,
0.025766871165644172, 0.147239263803681, 0.08711656441717791, 0.05644171779141103 6,
0.07116564417177913, 0.1116564417177914, 0.015950920245398775, 0.10061349693251533]

Рулетка
[14, 6, 10, 10, 4, 8, 9, 8, 7, 12, 9, 5, 8, 11]
```

```
Стохастическая универсальная
[6, 7, 8, 8, 9, 10, 11, 12, 13, 14, 1, 3, 3, 4]

Ранжированная селекция
Ранги:
[5, 4, 12, 7, 2, 10, 3, 14, 9, 6, 8, 13, 1, 11]
Выбор:
[12, 14, 3, 14, 9, 13, 3, 8, 11, 10, 14, 14, 12, 4]

Селекция усечения
[8, 12, 3, 14, 6, 9]

Турнирный отбор (бинарный)
[11, 8, 8, 9, 12, 12, 7, 8, 7, 6]
```

Вывод

В ходе выполнения лабораторной работы были реализованы все 5 метод селекций, которые хорошо отработали и по результатам видно, что значение фитнес-функции сильно влияет на выбор значений, однако даже у маленьких значений имеются хорошие шансы для выпадения в определённых выборках.

Лабораторная работа №5

Задание

Реализовать генетический алгоритм, направленный на поиск такого набора чисел из 4 чисел при котором линейная функция:

F = a + 3*b + 5*c + 7*d = 42

Основные этапы задачи:

1) Сгенерировать начальную популяцию случайных решений (начальная популяция равна 100).

2) Оценить пригодность каждой строки по расстоянию до целевого значения.

3) Выбрать пары решений пропорционально их пригодности (чем ближе к цели, тем выше шанс).

4) Создать новое поколение с помощью кроссовера — случайного смешивания частей двух родителей.

5) Применить мутацию к менее успешным решениям.

6) Повторять процесс до тех пор, пока не будет найдено точное решение, при котором F = 42.

Теория

Генетический алгоритм — это эвристический метод оптимизации и поиска решений, основанный на принципах естественного отбора и генетики. Он имитирует эволюцию популяции особей (решений), постепенно улучшая их качество. Основные этапы ГА:

Инициализация популяции: создается начальный набор случайных решений (хромосом).

Оценка приспособленности (Фитнесс-функция): каждая особь оценивается по функции приспособленности (например, чем лучше решение, тем выше значение).

Селекция (отбор): выбираются лучшие особи для размножения (например, рулеточная, турнирная селекция).

Кроссовер (скрещивание): родительские особи обмениваются частями своих хромосом, создавая потомков.

Мутация: случайные изменения в генах потомков (для поддержания разнообразия).

Формирование новой популяции: лучшие особи (родители и потомки) переходят в следующее поколение.

Проверка условия остановки: алгоритм завершается при достижении максимума поколений, нужного качества решения и т. д.

Результат

```
Изначальный массив:
1: [31, 21, 40, 18]
2: [28, 9, 32, 10]
3: [40, 14, 14, 33]
…
98: [28, 1, 3, 31]
99: [23, 19, 18, 7]
100: [4, 30, 19, 9]
```

```
1 итерация
Среднее расстояние было: 312.34
Среднее расстояние стало: 261.32

2 итерация
Среднее расстояние было: 261.32
Среднее расстояние стало: 203.19

3 итерация
Среднее расстояние было: 203.19
Среднее расстояние стало: 128.59

4 итерация
Среднее расстояние было: 128.59
Среднее расстояние стало: 78.27

5 итерация
Среднее расстояние было: 78.27
Среднее расстояние стало: 46.91

6 итерация
Среднее расстояние было: 46.91
Среднее расстояние стало: 19.49
Решение в строке 61: [11, 4, 1, 2]
```

Вывод

В ходе выполнения алгоритма был написан метод мутации для вещественных чисел (в разработанном случае число были использованы только целые числа 1 и -1) с изменением рандомного значения в строке, только для тех строк, в которых дистанция была больше среднего значения. Благодаря такому решению необходимые значения для решения функции F были найдены быстрее. Также в выводе программы видно, что с большой скоростью среднее расстояние приближалось к необходимому значению F=42.

Лабораторная работа №6

Задание

С помощью алгоритма роя частиц найти экстремум $\mathrm{F}(\mathrm{x},\mathrm{y}) = -(1 + sin^2(\mathrm{x})) * (1 + sin^2(\mathrm{y}))$, где $x \in [0; 4\pi]$. Число итераций равно 10, а рой равен 5.

Алгоритм**:**

1) Случайная генерация на квадрате $[0; 4\pi] \times [0; 4\pi]$ для $\bar{x}_i$ и $\bar{v}_i$ с ограничением скорости в 0.5;

2) Вычисление $\bar{P}_i$ и $\bar{g}$;

3) Вычисление α и β;

4) Вычисляем новые $\bar{x}_i$ и $\bar{v}_i$;

5) Вычисляем новые $\bar{P}_i$ и $\bar{g}$;

6) Либо финиш (все частицы достигли $\bar{g}$) либо по новой с пункта 3.

Теория

Основная идея заключается в том, что множество простых агентов, взаимодействующих по определенным правилам, способны совместно решать сложные задачи. Каждый агент обладает минимальной индивидуальной интеллектуальностью, но за счет обмена информацией и самоорганизации вся система демонстрирует разумное поведение.

Одним из самых известных алгоритмов роевого интеллекта является оптимизация роем частиц, имитирующая поведение стаи птиц. В этом методе виртуальные частицы движутся в пространстве решений, запоминая свои лучшие позиции и корректируя траекторию на основе опыта всей группы. Такой подход эффективен для поиска глобального оптимума в сложных задачах.

Роевой интеллект находит применение в самых разных областях. В робототехнике он используется для координации групп дронов, в логистике — для оптимизации маршрутов, в финансах — для анализа рынков, а в медицине — для обработки медицинских данных. Главные преимущества таких алгоритмов — это гибкость, масштабируемость и устойчивость к отказам отдельных агентов.

Результат

```
Изначальные значения:
1-я частица:
  Позиция: [x=11.337, y=0.992]
  Скорость: [vx=0.408, vy=-0.090]
  Значение F: -3.210

2-я частица:
```

```
  Позиция: [x=10.161, y=3.528]
  Скорость: [vx=-0.164, vy=0.409]
  Значение F: -1.657

3-я частица:
  Позиция: [x=2.903, y=8.837]
  Скорость: [vx=-0.048, vy=0.305]
  Значение F: -1.380

4-я частица:
  Позиция: [x=10.522, y=9.003]
  Скорость: [vx=-0.224, vy=-0.405]
  Значение F: -2.092

5-я частица:
  Позиция: [x=1.099, y=4.401]
  Скорость: [vx=-0.426, vy=0.062]
  Значение F: -3.419

Начальные лучшие значения:
Глобальный лучший: F=-3.419 при [x=1.099, y=4.401]

--- Итерация 1 ---
Текущие позиции:
Частица 1: [x=11.745, y=0.902] F=-2.482, [Vx=-0.500, Vy=0.500]
Частица 2: [x=9.661, y=4.028] F=-1.688, [Vx=-0.500, Vy=0.500]
Частица 3: [x=2.403, y=8.337] F=-2.592, [Vx=-0.500, Vy=-0.500]
Частица 4: [x=10.022, y=8.503] F=-2.152, [Vx=-0.500, Vy=-0.500]
Частица 5: [x=0.673, y=4.463] F=-2.693, [Vx=0.000, Vy=-0.000]

Лучшие локальные значения:
Частица 1: F=-3.210 при [x=11.337, y=0.992]
Частица 2: F=-1.688 при [x=9.661, y=4.028]
Частица 3: F=-2.592 при [x=2.403, y=8.337]
Частица 4: F=-2.152 при [x=10.022, y=8.503]
Частица 5: F=-3.419 при [x=1.099, y=4.401]

Глобальный лучший: F=-3.419 при [x=1.099, y=4.401]
--- Итерация 2 ---
…
--- Итерация 3 ---
…
--- Итерация 10 ---
Текущие позиции:
Частица 1: [x=10.946, y=1.246] F=-3.792, [Vx=0.302, Vy=0.500]
Частица 2: [x=11.246, y=1.504] F=-3.869, [Vx=-0.452, Vy=-0.499]
Частица 3: [x=5.403, y=5.609] F=-2.216, [Vx=0.500, Vy=-0.500]
Частица 4: [x=10.770, y=5.228] F=-3.426, [Vx=-0.013, Vy=-0.500]
Частица 5: [x=3.830, y=2.979] F=-1.440, [Vx=0.500, Vy=-0.500]

Лучшие локальные значения:
Частица 1: F=-3.824 при [x=11.245, y=1.402]
Частица 2: F=-3.991 при [x=11.008, y=1.504]
Частица 3: F=-2.592 при [x=2.403, y=8.337]
Частица 4: F=-3.762 при [x=11.022, y=7.503]
Частица 5: F=-3.419 при [x=1.099, y=4.401]

Глобальный лучший: F=-3.991 при [x=11.008, y=1.504]

Финальные результаты:
Лучшее найденное решение: F=-3.991
При координатах: [x=11.008, y=1.504]
```

Вывод

В результате выполнения лабораторной работы за 10 итераций для роя из 5 элементов были найдены близкие к лучшему значению на выбранном квадрате, а также видно динамику того, как все частицы движутся к более лучшим значениям, из чего можно сказать, что с большим количество итераций и числом роя значение будет все больше приближаться к минимальному.