**ЛАБОРАТОРНАЯ РАБОТА № 3.**

**Методы многомерного поиска.**

**Цель работы:**

1. **Изучение алгоритмов многомерного поиска 1-го и 2-го порядка.**
2. **Разработка программ реализации алгоритмов многомерного поиска 1-го и 2-го порядка.**
3. **Вычисление экстремумов функции.**

**Методические указания.**

**4.2.1. Метод наискорейшего градиентного спуска.**

**Стратегия поиска.**

Стратегия решения задачи состоит в построении последовательности точек  таких, что . Точки последовательности  вычисляются по правилу , где точка  задается пользователем; величина шага  определяется для каждого значения  из условия:

.

Решение задачи , где , может осуществляться с использованием необходимого условия минимума  с последующей проверкой достаточного условия минимума . Такой путь может быть использован либо при простой минимизирующей функции , либо при аппроксимации достаточно сложной функции  полиномом  (как правило, второй или третьей степени), и тогда условие  замещается условием , а условие  - условием .

Другой путь решения задачи  связан с использованием численных методов, когда ищется , т.е. с использованием методов одномерного поиска. Границы интервала  задаются пользователем. При этом степень близости найденного значения  к оптимальному значению , удовлетворяющему условиям  и , зависит от задания интервала  и точности методов одномерной минимизации.

Построение последовательности  заканчивается в точке , для которой , где  - заданное число, или если ,  - предельное число итераций, или при двукратном одновременном выполнении неравенств , где  - малое положительное число. Вопрос о том, может ли точка  рассматриваться как найденное приближение искомой точки локального минимума , решается путем дополнительного исследования.

**Алгоритм.**

**Ш.1.** Задать , предельное число итераций . Найти градиент функции в начальной точке .

**Ш.2.** Положить .

**Ш.3.** Вычислить .

**Ш.4.** Проверить выполнение критерия окончания :

 если неравенство выполнено, то  ;

 если нет, то перейти на Ш.6.

**Ш.5.** Проверить выполнение неравенства :

 если неравенство выполнено, то ;

 если нет, то перейти на Ш.6.

**Ш.6.** Вычислить величину шага , где .

**Ш.7.** Вычислить .

**Ш.8.** Проверить выполнение условий: :

 если оба условия выполнены при текущем значении  и , то ; расчет окончен;

 если хотя бы одно из условий не выполнено, то положить  и перейти на Ш.3.

**Замечание 3.2.**

Метод наискорейшего спуска гарантирует сходимость последовательности  к точке минимума для сильно выпуклых функций.

**4.2.2. Метод Флетчера-Ривза и Полака-Рибьера.**

***Постановка задачи***

Требуется найти безусловный минимум функции  многих переменных, т,е. найти такую точку  , что на множестве допустимых решений . При этом предполагается использование методов одномерного поиска для определения величины шага в направлении поиска .

**Стратегия поиска.**

Стратегия метода Флетчера-Ривза (FR) состоит в построении последовательности точек , таких, что . Точки последовательности  вычисляются по правилу:

 (4.1.1.)

 (4.1.2.)

 (4.1.3.)

. (4.1.4.)

Точка  задается пользователем, величина шага  определяется для каждого значения  из условия . Решение задачи одномерной минимизации может осуществляться либо из условия , либо численно, с использованием методов одномерной минимизации, когда решается задача:

. (4.1.5.)

При численном решении задачи определения величины шага степень близости найденного значения  к оптимальному значению , удовлетворяющему условиям , зависит от задания интервала  и точности одномерной минимизации.

Вычисление величины  по формуле (4.1.4.) обеспечивает для квадратичной формы  построение последовательности -сопряженных направлений , для которых . При этом в точках последовательности  градиенты функции  взаимно перпендикулярны, т.е. .

Для квадратичных функций  с матрицей  метод Флетчера-Ривза является конечным и сходится за число шагов, не превышающее - размерность  вектора переменных.

При минимизации неквадратичных функций метод не является конечным, при этом следует отметить, что погрешность в решении задачи (4.1.5.) приводит к нарушению не только перепендикулярности градиентов, но и -сопряженности направлений. Для неквадратичных функций, как правило, используется алгоритм Полака-Рибьеры, когда в формулах (4.1.1. – 4.1.3.) величина  вычисляется следующим образом:

, 

где . В отличие от алгоритма Флетчера-Ривза алгоритм Полака-Рибьера предусматривает использование итерации наискорейшего спуска через каждые  шагов. Построение последовательности  заканчивается в точке, для которой , где  - заданное число, или при  - предельное число итераций, или при двукратном одновременном выполнении двух неравенств  где  - малые положительные числа. Вопрос о том, может ли точка  рассматриваться как найденное приближение искомой точки минимума, решается путем проведения дополнительного исследования.

**Алгоритм.**

**Ш.1.** Задать  - предельное число итераций. Вычислить градиент .

**Ш.2.** Положить 

**Ш.3.** Вычислить

**Ш.4.** Проверить выполнение критерия окончания 

 если критерий выполнен,  расчет заканчивается;

 если нет, то перейти на Ш.5.

**Ш.5.** Проверить условие 

 если неравенство выполняется, то расчет окончен и 

 если нет, то при  перейти на Ш.6., а при  перейти на Ш.7.

**Ш.6.** Определить 

**Ш.7.** Определить

, или 

**Ш.8.** Определить 

**Ш.9.** Найти  из условия .

**Ш.10.** Вычислить 

**Ш.11.** Проверить выполнение условий 

 в случае выполнения обоих условий в двух последовательных итерациях с номерами  и  расчет окончен, найдена точка .

 если не выполняется хотя бы одно из условий, полагаем  и переход на Ш.3.

**4.2.3. Метод Девидона-Флетчера-Пауэлла.**

***Постановка задачи***

Пусть дана функция , ограниченная снизу на множестве  и имеющая непрерывные частные производные во всех его точках (т.е. ).

Требуется найти локальный минимум функции  на множестве допустимых решений , т.е. найти такую точку , что .

**Стратегия поиска.**

Стратегия метода Девидона-Флетчера-Пауэлла (DFP) состоит в построении последовательности точек , таких, что . Точки последовательности  вычисляются по правилу:

 , (4.2.1.)

где  - матрица размера , являющаяся аппроксимацией обратной матрицы Гессе. Она вычисляется по правилу:

 (4.2.2.)

 (4.2.3.)

где .

Точка  задается пользователем, величина шага  определяется из условия:

**.** (4.2.4.)

Решение задачи (4.2.4.) может выполняться как из условия  , либо численно, с использованием методов одномерной минимизации, когда решается задача:  оптимизации.

Формулы (4.2.2.), (4.2.3.) при аналитическом решении задачи (4.2.4.) обеспечивают построение последовательности  положительно определенных матриц, таких, что  при . Следствием этого для квадратичной функции , является тот факт, что направления  будут -сопряженными и, следовательно, алгоритм DFP сойдется не более чем за шагов.

Для неквадратичных функций  алгоритм DFP перестаёт быть конечным и его сходимость зависит от точности решения задачи (4.2.4.). Глобальную сходимость алгоритма можно гарантировать лишь при его обновлении через каждые  шагов, т.е. когда в формуле (4.2.1.):



Построение последовательности  заканчивается в точке , для которой  где  - заданное число, или при ( - предельное число итераций), или при двукратном одновременном выполнении двух неравенств: , где  - малые положительные числа. Вопрос о том, может ли точка  рассматриваться как найденное приближение искомой точки минимума, решается путем проведения дополнительного исследования.

**Алгоритм.**

**Ш.1.** Задать  - предельное число итераций. Найти градиент .

**Ш.2.** Положить 

**Ш.3.** Вычислить .

**Ш.4.** Проверить критерий окончания 

 если критерий выполнен,  расчет заканчивается;

 Если нет, то перейти на Ш.5.

**Ш.5.** Проверить условие 

 если неравенство выполняется, то расчет окончен и 

 если нет, то при  перейти на Ш.10., а при  перейти на Ш.6.

**Ш.6.** Вычислить .

**Ш.7.** Вычислить .

**Ш.8.** Вычислить .

**Ш.9.** Вычислить 

**Ш.10.** Определить .

**Ш.11.** Вычислить .

**Ш.12.** Вычислить .

**Ш.13.** Проверить условия :

 в случае выполнения обоих условий в двух последовательных итерациях с номерами  и  расчет окончен, найдена точка .

 если не выполняется хотя бы одно из условий, полагаем  и переход на Ш.3.

**4.2.4. Метод Бройдена-Флетчера-Гольдфарба-Шенно.(BFGS).**

Обозначим



Тогда



Отсюда



Такая замена обеспечивает более устойчивый процесс

поиска экстремума. Как видно из соотношений для и  формулы

пересчета для DFP и BFGS взаимнообратны.



****

**4.2.4. Метод Левенберга-Марквардта.**

***Постановка задачи***

Пусть дана функция , ограниченная снизу на множестве  и имеющая непрерывные вторые частные производные во всех его точках (т.е. ).

Требуется найти локальный минимум функции  на множестве допустимых решений , т.е. найти такую точку , что .

**Стратегия поиска.**

Стратегия метода Левенберга-Марквардта (LM) состоит в построении последовательности точек , таких, что . Точки последовательности  вычисляются по правилу:

, (4.3.1.)

где точка  задается пользователем,  - единичная матрица,  - последовательность положительных чисел, таких, что матрица  положительно определена. Как правило, число  назначается как минимум на порядок больше, чем самый большой элемент матрицы , а в ряде стандартных программ полагается . Если , то . В противном случае . Легко видеть, что алгоритм Левенберга-Марквардта в зависимости от величины  на каждом шаге по своим свойствам приближается либо к алгоритму Ньютона, либо к алгоритму градиентного спуска.

Построение последовательности  заканчивается, когда либо  , либо число итераций , где  - малое положительное число, а  - предельное число итераций.

Вопрос о том, может ли точка  рассматриваться как найденное приближение искомой точки минимума, решается путем проведения дополнительного исследования.

**Алгоритм.**

**Ш.1.** Задать  - предельное число итераций. Найти градиент  и матрицу Гессе .

**Ш.2.** Положить 

**Ш.3.** Вычислить .

**Ш.4.** Проверить критерий окончания 

 если критерий выполнен,  расчет заканчивается;

 если нет, то перейти на Ш.5.

**Ш.5.** Проверить условие 

 если неравенство выполняется, то расчет окончен и 

 если нет, то перейти на Ш.6.

**Ш.6.** Вычислить .

**Ш.7.** Вычислить .

**Ш.8.** Вычислить .

**Ш.9.** Вычислить .

**Ш.10.** Вычислить .

**Ш.11.** Проверить выполнение условия 

 если неравенство выполняется, то перейти на Ш.12;

 если нет, перейти на Ш.13.

**Ш.12.** Положить  и перейти на Ш.3.

**Ш.13.** Положить  и перейти на Ш.7.

**Замечание 4.1.** В окрестности точки минимума  метод Левенберга-Марквардта обладает скоростью сходимости, близкой к квадратичной.

**Задание.**

Требуется найти минимум тестовой функции Розенброка:

.

1. Методами сопряженных градиентов (методом Флетчера-Ривза и методом Полака-Рибьера).
2. Квазиньютоновским методом (Девидона-Флетчера-Пауэлла).
3. Методом Левенберга-Марквардта.

**Замечание 4.2.** В качестве методов одномерного поиска использовать любой из известных методов одномерного поиска.

**Варианты задания:**

1. ;

2. ;

3. ;

4. ;

5. ;

6. ;

7. ;

8. ;

9. ;

10. ;

11. ;

12. ;

13. ;

14.;

15. ;

16.;

17. ;

18.;

19. ;

20. ;

21.;

22.;

23.;

24.;

25.

1. **Найти все стационарные точки и значения функций соответствующие этим точкам.**
2. **Оценить скорость сходимости указанных алгоритмов и сравнить по времени получение результата оптимизации для разных методов.**
3. **Реализовать алгоритмы программированием на одном из языков высокого уровня (C++, C#,Python, Haskell и др.).**
4. **Отчет представить в стандартном виде (TEX, PDF).**

**Требования к отчету.**

1. Отчет должен содержать:

1.1. титульный лист;

1.2. цель работы;

1.3. постановку задачи;

1.4. проверку решения на допустимость.

2. Исследование выполнить с помощью написанной Вами программы с результатами в графическом виде.

3. Кроме текста исследования следует привести также текст исходного кода программ.

4. Отчет оформляется в формате PDF желательно в редакторе TEX.

**Генетический алгоритм.**

**Цель работы:**

1. **Изучение методов решения задач многоэкстремальной оптимизации на основе генетического алгоритма.**
2. **Разработка программы реализации генетического алгоритма.**
3. **Решение задачи многоэкстремальной оптимизации для заданных многоэкстремальных функций.**

**Постановка задачи**

Дана целевая функция , определенная на множестве допустимых решений. Требуется найти глобальные минимумы заданных функций  на допустимом множестве . То есть такую точку

, где . **(1)**

Задача поиска максимума целевой функции  сводится к задаче поиска минимума путем замены знака перед функцией на противоположный:

**.**

**Стратегия поиска**

Генетические алгоритмы имитируют природные способы оптимизации, присущие процессам эволюции живых систем. А именно:

* генетическое наследование;
* изменчивость;
* естественный отбор.

***Целевая функция***  соответствует природному понятию ***приспособленности*** живого организма. Вектор переменных  целевой функции называется ***фенотипо*м**, а отдельные его параметры – ***признаками*** .

Любой живой организм может быть представлен своим генотипом и фенотипом.

***Генотип***– это совокупность наследственных признаков, информация о которых заключена в хромосомном наборе генов.

***Фенотип***– совокупность всех признаков и свойств организма, формирующихся в процессе взаимодействия его генотипа и внешней среды. Каждый ген имеет своё отражение в фенотипе.

***Генетические алгоритмы***ведут поиск решения на уровне генотипа. Каждую координату  вектора  представляют в некоторой форме , удобной для использования в генетическом алгоритме и называется *геном*. Для этого необходимо выполнить преобразование, в общем случае не взаимно однозначное, вектора переменных  в некоторую структуру , называемую ***хромосомой******(генотипом, особью****): ,* где  функция кодирования, - пространство представлений (как правило )*.*

Для того, чтобы восстанавливать по хромосоме решение, необходимо задать обратное преобразование * ,* где  - функция декодирования.

В пространстве представлений  вводится так называемая ***функция приспособленности (функция фитнеса)*** *,* где  - множество вещественных чисел, аналогичная целевой функции  на множестве . Функция  может быть любая функция, удовлетворяющая условию:

.

При решении используются конечные наборы:



Возможных решений, называемых ***популяциями*,** где  - хромосомы с номером ,  - размер популяции,  - ген с номером - той популяции.

Затем осуществляется обратное преобразование:

.

Различают два способа кодирования:

1. Бинарное кодирование.
2. Вещественное кодирование.

Будем использовать второй вариант кодирования. В этом случае целевая функция может использоваться непосредственно как функция фитнеса. Тогда в качестве функции фитнеса  получается как преобразование целевой функции, т.е. *функция фитнеса*  где  - множество вещественных чисел, аналогична целевой функции .

Функцией  может быть любая функция, удовлетворяющая следующему условию:

.

Решение исходной оптимизационной задачи  сводится к поиску решения  другой оптимизационной задачи:

. (2)

В силу выбора функции , решение задач (1) и (2) (хромосома) совпадают:

. (3)

При решении задачи (2) используются конечные наборы  возможных решений, называемых *популяциями*, где  - хромосома с номером ,  - размер популяции,  - ген с номером .

**Генетический алгоритм.**

**Ш.1. Формирование исходной популяции.**

Ш.1.1.Задается номер популяции , максимальное количество популяций , номер итерации цикла , размер популяции .

Ш.1.2. Случайным образом выбирается начальная точка  - исходная хромосома. Она может быть выбрана как внутренняя точка гиперкуба области допустимых значений . Из этой точки формируется исходная популяция. Для этого с помощью равномерного распределения на единичном отрезке   раз генерируется последовательность из  случайных точек . Используя линейное преобразование, каждая точка отображается на соответствующий ей промежуток . Составляя векторы из точек последовательности  при фиксированных , получаем  начальных векторов , координаты которых  имеют равномерное распределение на отрезках . Таким образом, может быть сформирована начальная популяция .

Ш.1.3. Вычисляется значение функции фитнеса для каждой особи  и популяции  в целом .

**Ш.2. Отбор (селекция).**

*Селекция* – это операция, которая осуществляет отбор особь (хромосом)  в соответствии со значениями функции фитнеса  для последующего их скрещивания.

Ш.2.1. Вычислить кумулятивную вероятность .

Ш.2.2. Сформировать случайное действительное число  в интервале .

Ш.2.3. Выбрать  - ю хромосому  так, чтобы .

Ш.2.4. Перейти на Ш.2.2. до тех пор, пока не будет сформирована новая популяция ().

**Ш.3. Кроссинговер (скрещивание).**

Скрещивание – это операция, при которой из нескольких, обычно двух хромосом (особей), называемых родителями, порождается одна или несколько новых, называемых потомками.

Ш.3.1. Определяется параметр  как вероятность кроссинговера. Эта вероятность дает ожидаемое число  хромосом, подвергаемых операции кроссинговера.

Ш.3.2. Для операции кроссинговера выполняется процесс, повторяющийся от  до : формируется случайное действительное число из сегмента , при этом, если , то хромосома  выбирается как родительская.

Ш.3.3. Отбираются пары родительских хромосом , где . Действие оператора кроссинговера осуществляется следующим образом:

Ш.3.4. Формируется случайное число , затем оператор кроссинговера, действующий на исходные пары , производит две хромосомы потомки  и :

.

Ш.3.5. Если допустимое множество является выпуклым, то кроссинговер обеспечивает допустимость обоих потомков, в случае если допустимы оба родителя. Следует проверить допустимость каждого потомка перед тем, как он будет включен в новую популяцию. Если оба потомка являются допустимыми, тогда родители заменяются этими потомками. Если это не так, сохраняется допустимый потомок, если он существует, а затем вновь выполняется оператор кроссинговера с новым значением случайного числа  до тех пор, пока не будут получены два новых допустимых потомка или не будет превышено заданное число циклов. В этом случае осуществляется замена родителей только теми (сохраненными ранее) потомками, которые оказались допустимыми.

**Ш.4. Мутация.**

Мутация **–** это преобразование хромосомы, случайно изменяющее один или несколько из её генов. Оператор мутации предназначен для того, чтобы поддерживать разнообразие особей в популяции.

Ш.4.1. Определим параметр  как вероятность мутации. Эта вероятность дает ожидаемое число  хромосом, подвергаемых операции мутации.

Ш.4.2. Для операции мутации выполняется процесс, повторяющийся от  до : формируется случайное действительное число из сегмента , при этом, если , то хромосома  выбирается как родительская для операции мутации Для каждой выбранной родительской хромосомы , обозначенной как , производится мутация.

Ш.4.3. Поочередно рассматривается каждый потомок из ожидаемого числа  хромосом. Среди генов выбранной родительской хромосомы  случайно ( с вероятностью ) выбирается один с номером  подлежащий замене. Его новое значение  случайным образом выбирается из промежутка  изменения выбранной координаты .

**Ш.5. Формирование новой популяции.**

Ш.5.1. С равной вероятностью из потомков мутантов предыдущего шага выбирается один .

Ш.5.2. Выбранный потомок добавляется в популяцию вместо хромосомы, которой соответствует наименьшее значение функции фитнеса (наихудшее из допустимых значений).

Ш.5.3 Вычисляется значение функции фитнеса для мутантного потомка .

Ш.5.4. Проверка условий:

Ш.5.4.1. Если , то  и переход на Ш.2.

Ш.5.4.2. Если , то  и переход на Ш.6.

**Ш.6. Проверка условия останова генетического алгоритма.**

Условием окончания работы генетического алгоритма является формирование заданного количества популяций .

Ш.6.1. Если условие не выполнено, то полагаем  и переход на Ш.2.

Ш.6.2. Если условие окончания работы выполнено, то в качестве решения (приближенного) задачи  выбирается особь с лучшим значением функции фитнеса из текущей популяции: , а по нему определяется приближенное решение поставленной задачи .

**Замечание 1.** В качестве *функции фитнеса* можно использовать обратную целевую функцию.

**Замечание 2.** Обычно размер популяции выбирают в пределах 30-60 особей.

**Замечание 3.** Вероятность кроссинговера принимается равной , вероятность мутации .