Лабораторная работа 5.

Прямые методы минимизации функций многих переменных.

Постановка задачи: Требуется найти безусловный минимум функции n переменных $f(x_1, x_2, ..., x_n)$, т.е. такую точку $x^* \in E_n$, что $f(x^*) = \min_{x \in E_n} f(x)$.

В данной работе рассматриваются прямые методы решения задачи, для применения которых не требуется информации о производных целевой функции.

Метод правильного симплекса.

Стратегия поиска: Симплексом в E_n называется выпуклая оболочка $n+1$ точек $x_1, x_2, ..., x_{n+1}$, не принадлежащих ни к какому $(n-1)$ - мерному подпространству E_n. Правильным симплексом называется множество из $n+1$ равноудаленной точки. Например, в E_2 правильный симплекс – правильный треугольник, в E_3 - правильный тетраэдр.

В начале метода строится некоторый правильный симплекс в пространстве E_n. На каждой итерации сравниваются значения $f(x)$ в вершинах симплекса. Затем преобразуется та вершина симплекса, в которой $f(x)$ достигает максимальное значение. Преобразование происходит путем отражения данной вершины симметрично относительно центра тяжести x_c остальных вершин. Если значение целевой функции $f(x)$ в полученной точке меньше, чем в исходной, то переходят к новому симплексу. Иначе пытаются осуществить процедуру отражения для остальных вершин исходного симплекса, занумерованных в правильном порядке. Если все попытки неудачны, повторяют процедуру отражения с уменьшенной длиной ребра до тех пор, пока длина ребра не станет меньше заданной точности ε.

Алгоритм:

1. Задать точность вычислений ε, выбрать начальное приближение x^0, ребро a.
2. Построить начальный правильный симплекс по заданному ребру a и точке x^0.
3. Вычислить значения $f(x)$ в вершинах симплекса.
4. Упорядочить вершины симплекса в порядке возрастания значений $f(x)$.
5. Найти $x_c^1 = \dfrac{1}{n}\sum_{i=1}^{n} x^{1,i}$ и выполнить отражение вершины $n+1$: $x^{2,n+1} = 2x_c^1 - x^{1,n+1}$.

Если $f(x^{2,n+1}) < f(x^{1,n+1})$, то положить $x^{1,n+1} = x^{2,n+1}$ и перейти к шагу 3. Иначе перейти к шагу 6.

6. По формулам, аналогичным формулам пункта 5, вычислить отражения вершин с номерами $j = n, n-1, ..., k$. Где k выбирается из условия $f(x^{2,k}) < f(x^{1,k})$, либо $k = 1$.
7. Если $k > 1$, то положить $x^{1,k} = x^{2,k}$ и перейти к шагу 3. Иначе перейти к шагу 8.
8. Выбрать новый размер ребра $a = a/2$. Если $a > \varepsilon$, перейти к пункту 2. Иначе остановка расчета и выбор в качестве точки минимума точки x^0.

Метод деформируемого симплекса
(Метод Нелдера-Мида).

Стратегия поиска: Метод является усовершенствованным вариантом метода правильного симплекса. В данном случае при построении нового симплекса, кроме операции отражения, допускаются операции сжатия и растяжения. А именно, положение новой вершины симплекса после k итераций находится путем сравнения и выбора наименьшего среди значений целевой функции в следующих четырёх пробных точках:

$$z^1 = x_c^k - \alpha(x_c^k - x^{k,n+1}) = \frac{1-\alpha}{n}\sum_{i=1}^{n} x^{k,i} + \alpha x^{k,n+1}, \quad 0<\alpha<1;$$

$$z^2 = x_c^k + \alpha(x_c^k - x^{k,n+1}) = \frac{1+\alpha}{n}\sum_{i=1}^{n} x^{k,i} - \alpha x^{k,n+1}, \quad 0<\alpha<1;$$

$$z^3 = x_c^k + \beta(x_c^k - x^{k,n+1}) = \frac{1+\beta}{n}\sum_{i=1}^{n} x^{k,i} - \beta x^{k,n+1}, \quad \beta\approx 1;$$

$$z^4 = x_c^k + \gamma(x_c^k - x^{k,n+1}) = \frac{1+\gamma}{n}\sum_{i=1}^{n} x^{k,i} - \gamma x^{k,n+1}, \quad \gamma>1.$$

На практике хорошо себя зарекомендовал следующий выбор параметров для нахождения пробных точек: $\alpha = 1/2,\ \beta = 1,\ \gamma = 2.$

При деформациях утрачивается свойство правильности исходного симплекса. Поэтому, не стремясь к правильности начального симплекса, его строят из произвольной базовой точки $x^0 \in E_n$ по формулам $x^i = x^0 + a e^i$, где e^i - i-й базисный вектор; a - параметр симплекса.

Для того чтобы избежать сильной деформации симплекса, алгоритм иногда дополняют процедурой обновления. Например, после N шагов алгоритма из одной вершины, пусть x^1, снова строят симплекс по формулам $x^{i+1} = x^1 + a e^i$, полагая $a = \left\| x^1 - x^{n+1} \right\|$.

Алгоритм:

1. Задать точность вычислений ε, выбрать начальное приближение x^0, ребро a, число обновления N. Положить счётчик итераций $k=0$.
2. Построить начальный симплекс по формуле $x_j = x_0 + a e_j$, где e_j - j-й базисный вектор.
3. Вычислить значения $f(x)$ в вершинах симплекса. Положить $k = k+1$.
4. Упорядочить вершины симплекса в порядке возрастания значений $f(x)$.
5. Найти $x_c^k = \frac{1}{n}\sum_{i=1}^{n} x^{k;i}$ и выполнить отражение вершины $n+1$ в ту из пробных точек (z^1, z^2, z^3, z^4), значение функции в которой минимально.

6.Посчитать среднее арифметическое длин рёбер треугольника $\rho_{cp} = \dfrac{\sum_{i=1}^{n+1} \rho(x^{k;i}, x^{k;i+1})}{n+1}$, если $\rho_{cp} < \varepsilon$ перейти к шагу 8, иначе перейти к шагу 7 $a = \rho_{cp}$, $x^0 = x^{k;1}$.

7.Если счётчик итераций k кратен N перейти к пункту 2, иначе к пункту 3.

8.Поиск завершён. Положить $x^* = x_c^k$.

Метод циклического покоординатного спуска.

Стратегия поиска: Метод заключается в последовательной одномерной минимизации целевой функции $f(x)$ с исчерпывающим спуском сначала по направлению первого базисного вектора e^1, затем - второго e^2 и так далее. После окончания минимизации по направлению последнего базисного вектора e^n цикл повторяется.

Критерием остановки метода является условие того, что за полный цикл норма изменения точки x не превышает заданной точности метода, то есть расстояние между промежуточной точкой минимума вначале цикла и в его конце меньше ε.

Алгоритм:

1.Задать начальную точку $x_0^0 \in E_n$, базисные вектора $(e_1, e_2, ..., e_j, ..., e_n)$, величины $\varepsilon_1, \varepsilon_2$ критерия достижения точности $\rho(x_0^k, x_n^k) < \varepsilon_1$ или $\left| f(x_0^k) - f(x_n^k) \right| < \varepsilon_2$. Найти $f(x_0^0)$, положить $j = 1$, счётчик итераций $k = 0$.

2.Решить задачу одномерной минимизации $\Phi(\alpha) = f(x_{j-1}^k + \alpha e_j) \to \min, \quad \alpha \in E$, т.е. найти α^*. Положить $x_j^k = x_{j-1}^k + \alpha^* e^j$, вычислить $f(x_j^k)$.

3.Если $j < n$, то положить $j = j + 1$ и перейти к шагу 2, иначе - к шагу 4.

4.Проверить условие достижения точности $\rho(x_0^k, x_n^k) < \varepsilon_1$ или $\left| f(x_0^k) - f(x_n^k) \right| < \varepsilon_2$. Если оно выполняется, то положить $x^* = x_n^k, \quad f(x^*) = f(x_n^k)$ и закончить поиск. Иначе – положить $x_0^{k+1} = x_n^k$, $k = k + 1$, $j = 1$ и перейти к шагу 2.

Для приближенного решения вспомогательной задачи одномерной минимизации на шаге 2 на практике удобно использовать метод поразрядного поиска. Алгоритм этого одномерного метода не требует обязательного задания границ интервала минимизации.

Метод Хука-Дживса.

Стратегия поиска: Эффективность прямого поиска точки минимума унимодальной целевой функции можно повысить, если на каждом k-м шаге поиска последовательно выбирать направление спуска. Для этого на каждом k-м шаге выделяют предварительный этап исследующего поиска. Целью этого этапа является выбор направления спуска путем исследования поведения целевой функции $f(x)$ в окрестности точки x^{k-1}, найденной на предыдущем шаге. В результате исследующего поиска находится точка $\bar{x}^k$, для которой $f(\bar{x}^k) < f(x^{k-1})$. Направление спуска, завершающего k-й шаг поиска, определяется вектором $\bar{x}^k - x^{k-1}$.

Алгоритм исследующего покоординатного спуска:

1. Задать точку x с приращениями по каждой координате $\Delta_j, \quad j = 1, ..., n$.

2. Положить $\bar{x} = x, \quad j = 1$.

3. Сделать пробный шаг $y = \bar{x} - \Delta_j e^j$, где e^j - j-й базисный вектор. Если $f(\bar{x}) \le f(y)$, то перейти к шагу 4, иначе – к шагу 5.
4. Сделать пробный шаг $y = \bar{x} + \Delta_j e^j$. Если $f(\bar{x}) \le f(y)$, то прейти к шагу 6, иначе – к шагу 5.
5. Положить $\bar{x} = y$.Перейти к шагу 6.
6. Положить $j = j + 1$. Если $j \le n$, то перейти к шагу 3. Иначе исследующий поиск окончен – получена точка $\bar{x}$, для которой $f(\bar{x}) < f(x)$, если $\bar{x} \ne x$.

Полный алгоритм Хука – Дживса:

1.Выбрать начальную точку x^0, вектор приращений $\Delta = (\Delta_1, ..., \Delta_n)$, коэффициент уменьшения шага $\gamma > 1$, параметр окончания поиска $\varepsilon > 0$.

2.Провести исследующий покоординатный поиск из точки x^0, т.е. найти точку $\bar{x}^0$. Если $\bar{x}^0 \ne x^0$, то перейти к шагу 4, иначе к шагу 3.

3.Проверка окончания поиска. Если $\|\Delta\| < \varepsilon$, то прекратить поиск и положить $x^* = x^0$. Иначе – положить $\Delta = \Delta / \gamma$ и перейти к шагу 2.

4.Перемещение из точки $\bar{x}^0$ в направлении убывания $\bar{x}^0 - x^0$:

$$x^1 = x^0 + a_k (\bar{x}^0 - x^0),$$

подбирая $a_k > 0$, чтобы найти такую точку x^1, чтобы $f(x^1) < f(\bar{x}^0)$.

5.Положить $x^0 = x^1$ и перейти к шагу 2.

Замечание 1. Значение a_k можно подбирать из условия исчерпывающего спуска целевой функции $f(x)$ при смещении точки $\bar{x}^0$ в направлении этого вектора.

Замечание 2. В простейшем варианте метода значение a_k не подбирают, а полагают $a_k = 2$. В этом случае формула, по которой осуществляется спуск, имеет вид

$$x^1 = x^0 + 2(\bar{x}^0 - x^0) = 2\bar{x}^0 - x^0.$$

Метод случайного поиска (алгоритм с возвратом при неудачном шаге).

Стратегия поиска: Задается начальная точка x^0. Каждая последующая точка находится по формуле

$$x^{k+1} = x^k + \alpha_k \frac{\xi}{\|\xi\|}, \quad k = 0,1,...$$

где $a_k > 0$ - величина шага, $\xi = (\xi_1, ..., \xi_n)$ - некоторая реализация n-мерного случайного вектора ξ. Будем считать, что координаты вектора ξ - это независимые случайные величины, равномерно распределенные на отрезке $[-1;1]$. На текущей итерации при помощи генерирования случайных векторов ξ получаются точки, лежащие на гиперсфере радиуса a_k с центром в точке x^k. Если значение функции в полученной точке не меньше, чем в центре сферы, шаг считается неудачным. Если число неудачных шагов из данной точки достигает заданного числа M, поиск повторяется из той же точки с уменьшенным шагом до тех пор, пока шаг не станет меньше заданной точности. Если же значение функции в полученной точке меньше, чем в центре, шаг считается удачным и полученное значение выбирается за новый центр поиска.

Алгоритм:

1.Выбрать параметр точности $\varepsilon > 0$, начальный шаг $\alpha > 0$, коэффициент уменьшения шага $\gamma > 1$, предельное число попыток M, точку x. Вычислить $f(x)$.

2.Положить счетчик числа неудачных попыток $j = 1$.

3.Получить реализацию случайного вектора ξ.

4.Найти пробную точку $y = x + \alpha \dfrac{\xi}{\|\xi\|}$, вычислить $f(y)$.

5.Если $f(y) < f(x)$, то положить $x = y,\ f(x) = f(y)$ и перейти к шагу 4. Иначе - перейти к шагу 6.

6.Положить $j = j+1$. Если $j \le M$, то перейти к шагу 3, иначе – к шагу 7.

7.Проверка условия достижения точности. Если $\alpha < \varepsilon$, то поиск завершить, полагая $x^* = x,\ f^* = f(x)$. Иначе – положить $\alpha = \alpha / \gamma$ и прейти к шагу 2.

Замечание. На практике предельное число неудачных попыток M обычно полагают равным $3n$, где n - число переменных целевой функции.

Задания

1. Написать в среде MATLAB функции, реализующие методы правильного симплекса, Нелдера-Мида, циклического покоординатного спуска, метод Хука-Дживса и метод случайного поиска.

2. Протестировать работу реализованных методов на примере овражной функции

$$f(x) = x_1^2 + a\,x_2^2,$$

при a = 1, 250, 1000. Задавая $\varepsilon = 10^{-3}$ и $\varepsilon = 10^{-5}$, сравнить скорость работы методов при различных значениях параметра a по числу итераций и по числу вычислений целевой функции.

3. Выбрать для выполнения работы тестовую функцию, номер которой соответствует номеру Вашего компьютера. Например, для компьютера №3 это будет функция 3), для компьютера №13 – функция 4): 13-9=4; для компьютера №23 это будет функция 5): 23-9×2=5.

1) $f(x) = 64x_1^2 + 126x_1x_2 + 64x_2^2 - 10x_1 + 30x_2 + 13$

2) $f(x) = 129x_1^2 - 256x_1x_2 + 129x_2^2 - 51x_1 - 149x_2 - 27$

3) $f(x) = 254x_1^2 + 506x_1x_2 + 254x_2^2 + 50x_1 + 130x_2 - 111$

4) $f(x) = 151x_1^2 - 300x_1x_2 + 151x_2^2 + 33x_1 + 99x_2 + 48$

5) $f(x) = 85x_1^2 + 168x_1x_2 + 85x_2^2 + 29x_1 - 51x_2 + 83$

6) $f(x) = 211x_1^2 - 420x_1x_2 + 211x_2^2 - 192x_1 + 50x_2 - 25$

7) $f(x) = 194x_1^2 + 376x_1x_2 + 194x_2^2 + 31x_1 - 229x_2 + 4$

8) $f(x) = 45x_1^2 - 88x_1x_2 + 45x_2^2 + 102x_1 + 268x_2 - 21$

9) $f(x) = 99x_1^2 + 196x_1x_2 + 99x_2^2 - 95x_1 - 9x_2 + 91$

4. Сравнить эффективность метода покоординатного циклического спуска и метода Хука–Дживса для задачи п.2 при a=250 и тестовой функции п.3 по числу итераций и по числу вычисленных значений целевой функции. Для метода Хука-Дживса величину шага исчерпывающего спуска выбирать по приближённой формуле и вычислять методом поразрядного поиска. *Объяснить полученные результаты.*

5. Используя методы прямого поиска, минимизировать функцию Розенброка

$$f(x) = 100(x_1^2 - x_2)^2 + (x_1 - 1)^2$$

с точностью $\varepsilon = 10^{-3}$, выбрав начальную точку $x^0 = (-1; 2)^T$.Установить, какие из примененных алгоритмов не позволяют при заданной точности поиска получить точку минимума $x^* = (1; 1)^T$ вследствие преждевременного окончания процесса поиска.

6. На примере функции Розенброка провести сравнение лучшего из прямых методов с методом сопряжённых градиентов, а также с квазиньютоновскими методами ДФП и БФГЖ. Определить сколько итераций (а также вычислений функции и производных) потребуется каждому методу, что бы разность между численным и точным решением была меньше ε.

7. На примере функции Химмельблау

$$f(x) = (x_1^2 + x_2 - 11)^2 + (x_1 + x_2^2 - 7)^2$$

рассмотреть особенности применения прямых методов для минимизации многомодальной функции. В качестве начального приближения взять точки $(0, 0)$ и $(-5, 0)$. Как зависит работа рассматриваемых алгоритмов от выбора начального приближения?

8. Встроенная функция Matlab fminsearch реализует метод Нелдера-Мида. Сравнить результаты ее работы с результатами, полученными Вашим алгоритмом. В каких случаях построенный вами алгоритм работает эффективнее функции fminsearch?

9. Сдать лабораторную работу преподавателю, *ответив предварительно на все следующие контрольные вопросы.*

Контрольные вопросы к Лабораторной работе 5.

1).Сформулировать стратегию построения алгоритма симплексного поиска.

2).Какая нумерация вершин симплекса называется правильной?

3).Описать алгоритм отражения вершины в методе правильного симплекса.

4).Зачем необходима и в чем заключается редукция правильного симплекса?

5).Сформулировать теоретическое обоснование минимизации целевой функции методом правильного симплекса.

6).В задачах минимизации с какими целевыми функциями метод правильного симплекса не может обеспечить высокой точности?

7).Сформулировать особенности минимизации целевой функции методом Нелдера-Мида по сравнению с ее минимизацией методом правильного симплекса.

8).Назвать класс целевых функций, при минимизации которых метод Нелдера-Мида имеет преимущество перед минимизацией по регулярному симплексу.

9).Сформулировать теоретическое обоснование минимизации целевой функции методом Нелдера-Мида.

10).Назвать класс унимодальных целевых функций, для которых эффективна минимизация методом циклического покоординатного спуска.

11).Как можно дополнительно повысить эффективность поиска точки минимума целевой функции, которая ищется методом циклического покоординатного спуска?

12).В чем состоит стратегия метода Хука-Дживса?

13).Какие подходы для реализации исследующего поиска в методе Хука-Дживса Вы знаете? В чем состоит метод исследующего покоординатного поиска?

14).Перечислите способы выбора ускоряющего множителя в методе Хука-Дживса при перемещении в направлении убывания.

15).Какие алгоритмы случайного поиска Вы знаете?

16).От какого параметра в наибольшей степени зависит эффективность алгоритмов случайного поиска?

17).На основе собственного опыта дать сравнительный анализ прямых методов.