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 $\{x^k\}, k = 0,1,\dots,$ таких, что $f(x^{k+1}) < f(x^k), k = 0,1,\dots$. Точки последовательности $\{x^k\}$ вычисляются по правилу $x^{k+1} = x^k - \alpha^k \nabla f(x^k)$, где точка x^0 задается пользователем; величина шага α^k определяется для каждого значения k из условия:

$$\varphi(\alpha^k) = f(x^k - \alpha^k \nabla f(x^k)) \to \min_{\alpha^k}.$$

Решение задачи $\alpha^{*k} = Arg\min f(x^k + \alpha^k d^k)$, где $d^k = -\nabla f(x^k)$, может осуществляться с использованием необходимого условия минимума $\dfrac{d\varphi(\alpha^k)}{d\alpha^k} = 0$ с последующей проверкой достаточного условия минимума $\dfrac{d^2\varphi(\alpha^k)}{d\alpha^{k2}} > 0$. Такой путь может быть использован либо при простой минимизирующей функции $\varphi(\alpha^k)$, либо при аппроксимации достаточно сложной функции $\varphi(\alpha^{*k}) = f(x^k - \alpha^k \nabla f(x^k))$ полиномом $P(\alpha^k)$ (как правило, второй или третьей степени), и тогда условие $\dfrac{d\varphi(\alpha^k)}{d\alpha^k} = 0$ замещается условием $\dfrac{dP(\alpha^k)}{d\alpha^k} = 0$, а условие $\dfrac{d^2\varphi(\alpha^k)}{d\alpha^{k2}} > 0$ - условием $\dfrac{d^2P(\alpha^k)}{d\alpha^{k2}} > 0$.

Другой путь решения задачи $\alpha^{*k} = Arg\min f(x^k + \alpha^k d^k)$ связан с использованием численных методов, когда ищется $\min\limits_{\alpha^k \in [a,b]} \varphi(\alpha^{*k}) = \min\limits_{\alpha^k \in [a,b]} f(x^k - \alpha^k \nabla f(x^k))$, т.е. с использованием методов одномерного поиска. Границы интервала $[a,b]$ задаются пользователем. При этом степень близости найденного значения α^k к оптимальному значению α^{*k}, удовлетворяющему условиям $\dfrac{d\varphi(\alpha^k)}{d\alpha^k} = 0$ и $\dfrac{d^2\varphi(\alpha^k)}{d\alpha^{k2}} > 0$, зависит от задания интервала $[a,b]$ и точности методов одномерной минимизации.

Построение последовательности $\{x^k\}, k = 0,1,\dots,$ заканчивается в точке x^k, для которой $\|\nabla f(x^k)\| < \varepsilon_1$, где $\varepsilon_1 > 0$ - заданное число, или если $k \ge M$, M - предельное число итераций, или при двукратном одновременном выполнении неравенств $\|x^{k+1} - x^k\| < \varepsilon_1, |f(x^{k+1}) - f(x^k)| < \varepsilon_2$, где ε_2 - малое положительное число. Вопрос о том, может ли точка x^k рассматриваться как найденное приближение искомой точки локального минимума x^*, решается путем дополнительного исследования.

Алгоритм.

Ш.1. Задать x^0, $\varepsilon_1 > 0, \varepsilon_2 > 0$, предельное число итераций M. Найти градиент функции в начальной точке $\nabla f(x^0) = \left[\frac{\partial f(x^0)}{\partial x_1}, \ldots, \frac{\partial f(x^0)}{\partial x_n}\right]^T$.

Ш.2. Положить $k = 0$.

Ш.3. Вычислить $\nabla f(x^k)$.

Ш.4. Проверить выполнение критерия окончания $\|\nabla f(x)\| < \varepsilon_1$:

$a)$ если неравенство выполнено, то $x^* = x^k$;

$b)$ если нет, то перейти на Ш.6.

Ш.5. Проверить выполнение неравенства $k \ge M$:

$a)$ если неравенство выполнено, то $x^* = x^k$;

$b)$ если нет, то перейти на Ш.6.

Ш.6. Вычислить величину шага $\alpha^{*k} = Arg \min f(x^k + \alpha^k d^k)$, где $d^k = -\nabla f(x^k)$.

Ш.7. Вычислить $x^{k+1} = x^k - \alpha^k \nabla f(x^k)$.

Ш.8. Проверить выполнение условий: $\|x^{k+1} - x^k\| < \varepsilon_1, |f(x^{k+1}) - f(x^k)| < \varepsilon_2$:

$a)$ если оба условия выполнены при текущем значении k и $k = k - 1$, то $x^* = x^{k+1}$; расчет окончен;

$b)$ если хотя бы одно из условий не выполнено, то положить $k = k + 1$ и перейти на Ш.3.

Замечание 3.2.

Метод наискорейшего спуска гарантирует сходимость последовательности $\{x^k\}$ к точке минимума для сильно выпуклых функций.

4.2.2. Метод Флетчера-Ривза и Полака-Рибьера.

<u>Постановка задачи</u>

Требуется найти безусловный минимум функции $f(x)$ многих переменных, т,е. найти такую точку $x^* \in R^n$, что $f(x^*) = \min\limits_{x \in R^n} f(x)$ на множестве допустимых решений $X \in R^n$. При этом предполагается использование методов одномерного поиска $\alpha^{*k} = Arg \min\limits_{\alpha^k \in R^1} f(x^k + \alpha^k d^k)$ для определения величины шага в направлении поиска d^k.

Стратегия поиска.

Стратегия метода Флетчера-Ривза (FR) состоит в построении последовательности точек $\{x^k\}, k = 0,1,\ldots$, таких, что $f(x^{k+1}) < f(x^k), k = 0,1,\ldots$. Точки последовательности $\{x^k\}$ вычисляются по правилу:

$$x^{k+1} = x^k + \alpha^k d^k, k = 0,1,\ldots; \tag{4.1.1.}$$

$$d^k = -\nabla f(x^k) + w^{k-1} d^{k-1}; \tag{4.1.2.}$$

$$d^0 = -\nabla f(x^0); \tag{4.1.3.}$$

$$w^{k-1} = \frac{\|\nabla f(x^k)\|^2}{\|\nabla f(x^{k-1})\|^2}. \tag{4.1.4.}$$

Точка x^0 задается пользователем, величина шага α^{*k} определяется для каждого значения k из условия $\alpha^{*k} = Arg \min\limits_{\alpha^k \in R^1} f(x^k + \alpha^k d^k)$. Решение задачи одномерной минимизации может осуществляться либо из условия $\dfrac{d\varphi(\alpha^k)}{d\alpha^k}=0, \dfrac{d^2\varphi(\alpha^k)}{d{\alpha^k}^2}>0$, либо численно, с использованием методов одномерной минимизации, когда решается задача:

$$\varphi(\alpha^k) \to \min_{\alpha^k \in [a,b]} . \qquad (4.1.5.)$$

При численном решении задачи определения величины шага степень близости найденного значения α^k к оптимальному значению α^{*k}, удовлетворяющему условиям $\dfrac{d\varphi(\alpha^k)}{d\alpha^k}=0, \dfrac{d^2\varphi(\alpha^k)}{d{\alpha^k}^2}>0$, зависит от задания интервала $[a,b]$ и точности одномерной минимизации.

Вычисление величины w^{k-1} по формуле (4.1.4.) обеспечивает для квадратичной формы $f(x)=\sum_{i=1}^{n}\sum_{j=1}^{n}{}_{ij}x_i x_j$ построение последовательности H-сопряженных направлений $d^0,d^1,\ldots,d^k,\ldots$, для которых $\langle d^j, Hd^i\rangle = 0, \forall i,j=0,1,\ldots,k; i\neq j$. При этом в точках последовательности $\{x^k\}$ градиенты функции $f(x)$ взаимно перпендикулярны, т.е. $\langle \nabla f(x^{k+1}), \nabla f(x^k)\rangle = 0, k=0,1,\ldots$.

Для квадратичных функций $f(x)$ с матрицей $H>0$ метод Флетчера-Ривза является конечным и сходится за число шагов, не превышающее n - размерность x 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 H-сопряженности направлений. Для неквадратичных функций, как правило, используется алгоритм Полака-Рибьеры, когда в формулах (4.1.1. – 4.1.3.) величина w^{k-1} вычисляется следующим образом:

$$w^{k-1}=\begin{cases}\dfrac{\|\nabla f(x^k)\|^2}{\|\nabla f(x^{k-1})\|^2}, & k\notin J,\\ 0, & k\in J,\end{cases} \qquad w^{k-1}=\begin{cases}\dfrac{\langle \nabla f(x^k), [\nabla f(x^k)-\nabla f(x^{k-1})\rangle}{\|\nabla f(x^{k-1})\|^2}, & k\notin J,\\ 0, & k\in J,\end{cases}$$

где $J=\{0,n,2n,\ldots\}$. В отличие от алгоритма Флетчера-Ривза алгоритм Полака-Рибьера предусматривает использование итерации наискорейшего спуска через каждые n шагов. Построение последовательности $\{x^k\}$ заканчивается в точке, для которой $\|\nabla f(x^k)\|<\varepsilon_1$, где $\varepsilon_1>0$ - заданное число, или при $k\geq M$, M - предельное число итераций, или при двукратном одновременном выполнении двух неравенств $\|x^{k+1}-x^k\|<\delta_2, |f(x^{k+1})-f(x^k)|<\varepsilon_2,$ где δ_2,ε_2 - малые положительные числа. Вопрос о том, может ли точка x^k рассматриваться как найденное приближение искомой точки минимума, решается путем проведения дополнительного исследования.

Алгоритм.

Ш.1. Задать $x^0,\varepsilon_1,\delta_2,\varepsilon_2,M$ - предельное число итераций. Вычислить градиент $\nabla f(x^0)$.

Ш.2. Положить $k=0$.

Ш.3. Вычислить $\nabla f(x^k)$.

Ш.4. Проверить выполнение критерия окончания $\|\nabla f(x^k)\|<\varepsilon_1$:

$a)$ если критерий выполнен, $x^*=x^k$, расчет заканчивается;

$b)$ если нет, то перейти на Ш.5.

Ш.5. Проверить условие $k \geq M$:

$a)$ если неравенство выполняется, то расчет окончен и $x^* = x^k$;

$b)$ если нет, то при $k = 0$ перейти на Ш.6., а при $k \geq 1$ перейти на Ш.7.

Ш.6. Определить $d^0 = -\nabla f(x^0)$.

Ш.7. Определить

$$w^{k-1} = \frac{\left\|\nabla f(x^k)\right\|^2}{\left\|\nabla f(x^{k-1})\right\|^2}, \text{ или } \; w^{k-1} = \begin{cases} \dfrac{\left\langle \nabla f(x^k), [\nabla f(x^k) - \nabla f(x^{k-1})\right\rangle}{\left\|\nabla f(x^{k-1})\right\|^2}, & k \notin J, \\ 0, & k \in J. \end{cases}$$

Ш.8. Определить $d^k = -\nabla f(x^k) + w^{k-1} d^{k-1}$.

Ш.9. Найти α^{*k} из условия $\alpha^{*k} = Arg \min\limits_{\alpha^k \in R^1} f(x^k + \alpha^k d^k)$.

Ш.10. Вычислить $x^{k+1} = x^k + \alpha^k d^k$.

Ш.11. Проверить выполнение условий $\left\|x^{k+1} - x^k\right\| < \delta_2, \left| f(x^{k+1}) - f(x^k)\right| < \varepsilon_2$:

$a)$ в случае выполнения обоих условий в двух последовательных итерациях с номерами k и $k-1$ расчет окончен, найдена точка $x^* = x^k$.

$b)$ если не выполняется хотя бы одно из условий, полагаем $k = k+1$ и переход на Ш.3.

4.2.3. Метод Девидона-Флетчера-Пауэлла.

Постановка задачи

Пусть дана функция $f(x)$, ограниченная снизу на множестве R^n и имеющая непрерывные частные производные во всех его точках (т.е. $f(x) \in C^1(X),\ X = R^n$).

Требуется найти локальный минимум функции $f(x)$ на множестве допустимых решений $X = R^n$, т.е. найти такую точку $x^* \in R^n$, что $f(x^*) = \min\limits_{x \in R^n} f(x)$.

Стратегия поиска.

Стратегия метода Девидона-Флетчера-Пауэлла (DFP) состоит в построении последовательности точек $\{x^k\}$, таких, что $f(x^{k+1}) < f(x^k), k = 0,1,...$. Точки последовательности $\{x^k\}$ вычисляются по правилу:

$$x^{k+1} = x^k - \alpha^{*k} G^{k+1} \nabla f(x^k), k = 0,1,... , \quad (4.2.1.)$$

где G^{k+1} - матрица размера $n \times n$, являющаяся аппроксимацией обратной матрицы Гессе. Она вычисляется по правилу:

$$G^{k+1} = G^k + \Delta G^k, G^0 = E, \quad (4.2.2.)$$

$$\Delta G^k = \frac{\Delta x^k (y^k)^T}{(y^k)^T \Delta g^k} - \frac{G^k \Delta g^k (G^k \Delta g^k)^T}{\left(\Delta g^k\right)^T G^k \Delta g^k}, \quad (4.2.3.)$$

где $\Delta x^k = x^{k+1} - x^k, \Delta g^k = \nabla f(x^{k+1}) - f(x^k)$.

Точка x^0 задается пользователем, величина шага α^{*k} определяется из условия:

$$\alpha^{*k} = Arg \min_{\alpha^k \in [a,b]} \varphi(x^k + \alpha^k d^k). \quad (4.2.4.)$$

Решение задачи (4.2.4.) может выполняться как из условия $\dfrac{d\varphi(\alpha)}{d\alpha^k} = 0, \dfrac{d^2\varphi(\alpha)}{d\alpha^{k\,2}} > 0$, либо численно, с использованием методов одномерной минимизации, когда решается задача: $\varphi(\alpha^k) \to \min\limits_{\alpha^k \in [a,b]}$ оптимизации.

Формулы (4.2.2.), (4.2.3.) при аналитическом решении задачи (4.2.4.) обеспечивают построение последовательности $\{G^k\}$ положительно определенных матриц, таких, что $G^k \to H^{-1}(x^*)$ при $k \to \infty$. Следствием этого для квадратичной функции $f(x) = \frac{1}{2}\langle Hx, x\rangle + \langle b, x\rangle, H > 0$, является тот факт, что направления $d^k, k = 0,1,...,$ будут H -сопряженными и, следовательно, алгоритм DFP сойдется не более чем за n шагов.

Для неквадратичных функций $f(x)$ 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 n шагов, т.е. когда в формуле (4.2.1.):

$$G^{k+1} = \begin{cases} E, k \in J;\ J = \{0, n, 2n, ...\}, \\ G^k + \Delta G^k, k \notin J. \end{cases}$$

Построение последовательности $\{x^k\}$ заканчивается в точке x^{*k}, для которой $\nabla f(x^k) < \varepsilon_1$, где ε_1 - заданное число, или при $k \geq M$ (M - предельное число итераций), или при двукратном одновременном выполнении двух неравенств: $\|x^{k+1} - x^k\| < \delta_2, |f(x^{k+1}) - f(x^k)| < \varepsilon_2$, где $\delta_2 > 0, \varepsilon_2 > 0$ - малые положительные числа. Вопрос о том, может ли точка x^k рассматриваться как найденное приближение искомой точки x^* минимума, решается путем проведения дополнительного исследования.

Алгоритм.

Ш.1. Задать $x^0, \varepsilon_1, \delta_2, \varepsilon_2, M$ - предельное число итераций. Найти градиент $\nabla f(x^0)$.

Ш.2. Положить $k = 0, G^0 = E.$

Ш.3. Вычислить $\nabla f(x^k)$.

Ш.4. Проверить критерий окончания $\|\nabla f(x^k)\| < \varepsilon_1$:

$a)$ если критерий выполнен, $x^* = x^k$, расчет заканчивается;

$b)$ Если нет, то перейти на Ш.5.

Ш.5. Проверить условие $k \geq M$:

$a)$ если неравенство выполняется, то расчет окончен и $x^* = x^k$;

$b)$ если нет, то при $k = 0$ перейти на Ш.10., а при $k \geq 1$ перейти на Ш.6.

Ш.6. Вычислить $\Delta g^k = \nabla f(x^{k+1}) - \nabla f(x^k)$.

Ш.7. Вычислить $\Delta x^k = x^{k+1} - x^k$.

Ш.8. Вычислить $\Delta G^k = \dfrac{\Delta x^k (\Delta x^k)^T}{(\Delta x^k)^T \Delta g^k} - \dfrac{G^k \Delta g^k (\Delta g^k)^T G^{kT}}{(\Delta g^k)^T G^k \Delta g^k}$.

Ш.9. Вычислить $G^{k+1} = G^k + \Delta G^k$.

Ш.10. Определить $d^k = -G^{k+1} \nabla f(x^k)$.

Ш.11. Вычислить $\alpha^{*k} = Arg \min\limits_{\alpha^k \in [a,b]} f(x^k + \alpha^k d^k)$.

Ш.12. Вычислить $x^{k+1} = x^k + \alpha^{*k} d^k$.

Ш.13. Проверить условия $\|x^{k+1} - x^k\| < \delta_2, |f(x^{k+1}) - f(x^k)| < \varepsilon_2$:

$a)$ в случае выполнения обоих условий в двух последовательных итерациях с номерами k и $k-1$ расчет окончен, найдена точка $x^* = x^{k+1}$.

$b)$ если не выполняется хотя бы одно из условий, полагаем $k = k+1$ и переход на Ш.3.

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

Обозначим

$$y_k = \Delta g_k = \nabla \varphi_{k+1} - \nabla \varphi_k;\ s_k = \Delta x_k = x^{k+1} - x^k.$$

Тогда

$$\widetilde{H}^{k+1} y_k = s_k;\ и\ H^{k+1} s_k = y_k.$$

Отсюда

$$H^{k+1} = H^k + \frac{y_k y_k^T}{y_k^T s_k} - \frac{H^k s_k s_k^T H^k}{s_k^T H^k s_k}$$

Такая замена обеспечивает более устойчивый процесс поиска экстремума. Как видно из соотношений для H^{k+1} и $\widetilde{H}^{k+1}$ формулы

пересчета для DFP и BFGS взаимнообратны.

$$\widetilde{H}^{k+1} = \widetilde{H}^k + \frac{(s_k - \widetilde{H}^k y_k)s_k^T + s_k(s_k - \widetilde{H}^k y_k)^T}{s_k^T y_k} - \frac{(s_k - \widetilde{H}^k y_k)^T y_k}{(s_k^T y_k)^2} s_k s_k^T,$$

$$\widetilde{H}^{k+1} = \left(I - \frac{s_k y_k^T}{s_k^T y_k}\right)\widetilde{H}^k\left(I - \frac{y_k s_k^T}{s_k^T y_k}\right) + \frac{s_k s_k^T}{s_k^T y_k}.$$

$$H^{k+1} = H^k + \frac{(y_k - H^k s_k)y_k^T + y_k(y_k - H^k s_k)^T}{y_k^T s_k} - \frac{(y_k - H^k s_k)^T s_k}{(y_k^T s_k)^2} y_k y_k^T,$$

$$H^{k+1} = \left(I - \frac{y_k s_k^T}{y_k^T s_k}\right)H^k\left(I - \frac{s_k y_k^T}{y_k^T s_k}\right) + \frac{y_k y_k^T}{y_k^T s_k}.$$

4.2.4. Метод Левенберга-Марквардта.

Постановка задачи

Пусть дана функция $f(x)$, ограниченная снизу на множестве R^n и имеющая непрерывные вторые частные производные во всех его точках (т.е. $f(x) \in C^2(X),\ X = R^n$).

Требуется найти локальный минимум функции $f(x)$ на множестве допустимых решений $X = R^n$, т.е. найти такую точку $x^* \in R^n$, что $f(x^*) = \min\limits_{x \in R^n} f(x)$.

Стратегия поиска.

Стратегия метода Левенберга-Марквардта (LM) состоит в построении последовательности точек $\{x^k\}$, таких, что $f(x^{k+1}) < f(x^k), k = 0,1,...$. Точки последовательности $\{x^k\}$ вычисляются по правилу:

$$x^{k+1} = x^k - \left[H(x^k) + \mu^k E\right]^{-1} \nabla f(x^k), k = 0,1,..., \qquad (4.3.1.)$$

где точка x^0 задается пользователем, E - единичная матрица, μ^k - последовательность положительных чисел, таких, что матрица $\left[H(x^k) + \mu^k E\right]^{-1}$ положительно определена. Как правило, число μ^0 назначается как минимум на порядок больше, чем самый большой элемент матрицы $H(x^0)$, а в ряде стандартных программ полагается $\mu^0 = 10^4$. Если $f\left(x^k - \left[H(x^k) + \mu^k E\right]^{-1} \nabla f(x^k)\right) < f(x^k)$, то $\mu^{k+1} = \frac{\mu^k}{2}$. В противном случае $\mu^{k+1} = 2\mu^k$. Легко видеть, что алгоритм Левенберга-Марквардта в зависимости от величины μ^k на каждом шаге по своим свойствам приближается либо к алгоритму Ньютона, либо к алгоритму градиентного спуска.

Построение последовательности $\{x^k\}$ заканчивается, когда либо $\|\nabla f(x^k) < \varepsilon_1\|$, либо число итераций $k \geq M$, где ε_1 - малое положительное число, а M - предельное число итераций.

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

Алгоритм.

Ш.1. Задать $x^0, \varepsilon_1, \delta_2, \varepsilon_2, M$ - предельное число итераций. Найти градиент $\nabla f(x^0)$ и матрицу Гессе $H(x^0)$.

Ш.2. Положить $k = 0, \mu^k = \mu^0$.

Ш.3. Вычислить $\nabla f(x^k)$.

Ш.4. Проверить критерий окончания $\|\nabla f(x^k)\| < \varepsilon_1$:

$a)$ если критерий выполнен, $x^* = x^k$, расчет заканчивается;

$b)$ если нет, то перейти на Ш.5.

Ш.5. Проверить условие $k \geq M$:

$a)$ если неравенство выполняется, то расчет окончен и $x^* = x^k$;

$b)$ если нет, то перейти на Ш.6.

Ш.6. Вычислить $H(x^k)$.

Ш.7. Вычислить $H(x^k) + \mu^k E$.

Ш.8. Вычислить $\left[H(x^k) + \mu^k E\right]^{-1}$.

Ш.9. Вычислить $d^k = -\left[H(x^k) + \mu^k E\right]^{-1} \nabla f(x^k)$.

Ш.10. Вычислить $x^{k+1} = x^k - \left[H(x^k) + \mu^k E\right]^{-1} \nabla f(x^k)$.

Ш.11. Проверить выполнение условия $f(x^{k+1}) < f(x^k)$:

$a)$ если неравенство выполняется, то перейти на Ш.12;

$b)$ если нет, перейти на Ш.13.

Ш.12. Положить $k = k+1, \mu^{k+1} = \dfrac{\mu^k}{2}$ и перейти на Ш.3.

Ш.13. Положить $\mu^{k+1} = 2\mu^k$ и перейти на Ш.7.

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

Задание.

Требуется найти минимум тестовой функции Розенброка:

$$f(x) = \sum_{i=1}^{n-1} \left[a(x_i^2 - x_{i+1})^2 + b(x_i - 1)^2\right] + f_0 .$$

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. $a = 50, b = 2,\ f_0 = 10, n = 2$;
2. $a = 150, b = 2,\ f_0 = 100, n = 3$;
3. $a = 80, b = 3,\ f_0 = 110, n = 2$;
4. $a = 250, b = 2,\ f_0 = 50, n = 2$;
5. $a = 70, b = 5,\ f_0 = 30, n = 3$;
6. $a = 30, b = 2,\ f_0 = 80, n = 4$;
7. $a = 250, b = 2,\ f_0 = 300, n = 2$;
8. $a = 158, b = 2,\ f_0 = 40, n = 2$;
9. $a = 500, b = 2, f_0 = 10, n = 2$;
10. $a = 350, b = 2, f_0 = 110, n = 2$;
11. $a = 300, b = 5, f_0 = 15, n = 2$;
12. $a = 200, b = 1, f_0 = 25, n = 2$;
13. $a = 100, b = 15, f_0 = 15, n = 2$;
14. $a = 500, b = 5, f_0 = 35, n = 2$;
15. $a = 100, b = 3, f_0 = 15, n = 2$;
16. $a = 140, b = 2, f_0 = 24, n = 2$;
17. $a = 1000, b = 10, f_0 = 150, n = 2$;
18. $a = 100, b = 2, f_0 = 45, n = 3$;
19. $a = 220, b = 3, f_0 = 12, n = 2$;
20. $a = 500, b = 15, f_0 = 25, n = 2$;
21. $a = 30, b = 3, f_0 = 45, n = 3$;
22. $a = 180, b = 2, f_0 = 15, n = 2$;
23. $a = 200, b = 5, f_0 = 48, n = 3$;
24. $a = 300, b = 25, f_0 = 250, n = 2$;
25. $a = 10, b = 250, f_0 = 45, n = 3$.

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).**

<u>Требования к отчету.</u>

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 $f(x) = f(x_1, x_2, ..., x_n)$, определенная на множестве допустимых решений $D \subseteq R^n$. Требуется найти глобальные минимумы заданных функций $f(x)$ на допустимом множестве D. То есть такую точку

$$x^e = Arg\min_{x \in D} f(x), \text{ где } x = (x_1, ..., x_n)^T, D = \{x | x_i \in [\alpha, \beta], i = 1, ..., n.\}. \qquad \textbf{(1)}$$

Задача поиска максимума целевой функции $f(X)$ сводится к задаче поиска минимума путем замены знака перед функцией на противоположный:

$$f(x^e) = \max_{x \in D} f(x) = -\min_{x \in D}[-f(x)].$$

Стратегия поиска

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

- генетическое наследование;
- изменчивость;
- естественный отбор.

Целевая функция $f(x)$ соответствует природному понятию ***приспособленности*** живого организма. Вектор переменных $x = (x_1, ..., x_n)^T$ целевой функции называется ***фенотипо*м**, а отдельные его параметры – ***признаками*** $i = 1, 2, ..., n$.

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 x_i вектора $x = (x_1, ..., x_n)^T \in D$ представляют в некоторой форме s_i, удобной для использования в генетическом алгоритме и называется *геном*. Для этого необходимо выполнить преобразование, в общем случае не взаимно однозначное, вектора переменных $x = (x_1, ..., x_n)^T \in D$ в некоторую структуру $s = (s_1, s_2, ..., s_n)^T \in S$, называемую ***хромосомой (генотипом, особью):*** $D \xrightarrow{e} S$, где e функция кодирования, S - пространство представлений (как правило $D \neq S$).

Для того, чтобы восстанавливать по хромосоме решение, необходимо задать обратное преобразование $S \xrightarrow{e^{-1}} D$, где e^{-1} - функция декодирования.

В пространстве представлений S вводится так называемая ***функция приспособленности (функция фитнеса)*** $\mu(s): S \xrightarrow{\mu} R$, где R - множество вещественных чисел, аналогичная целевой функции $f(x)$ на множестве D. Функция $\mu(s)$ может быть любая функция, удовлетворяющая условию:

$$\forall x^1, x^2 \in D: s^1 = e(x^1),\ s^2 = e(x^2),\ s^1 \neq s^2,\ eclu\ f(x^1) > f(x^2),\ mo\ \mu(s^1) > \mu(s^2)\ .$$

При решении используются конечные наборы:

$$I = \left\{ s^k = (s_1^k, s_2^k, ..., s_n^k)^T,\ k = 1,2,...,m \right\} \subset S$$

Возможных решений, называемых ***популяциями,*** где s^k - хромосомы с номером k, m - размер популяции, s_i^k - ген с номером i k - той популяции.

Затем осуществляется обратное преобразование:

$$x^e = e^{-1}(s^*) .$$

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 $\mu(x)$ получается как преобразование целевой функции, т.е. *функция фитнеса* $\mu(x): D \xrightarrow{\mu} R,$ где R - множество вещественных чисел, аналогична целевой функции $f(x)$.

Функцией $\mu(x)$ может быть любая функция, удовлетворяющая следующему условию:

$$\forall x^1, x^2 \in D:\ x^1 \neq x^2,\ eclu\ f(x^1) > f(x^2),\ mo\ \mu(x^1) > \mu(x^2)\ .$$

Решение исходной оптимизационной задачи $f(x^*) = \min_{x \in D} f(x)$ сводится к поиску решения x_μ^* другой оптимизационной задачи:

$$\mu(x_\mu^*) = \max_{x \in D} \mu(x) . \qquad (2)$$

В силу выбора функции $\mu(x)$, решение задач (1) и (2) (хромосома) совпадают:

$$x^e = x^* = Arg \min_{x \in D} f(x) = Arg \max_{x \in D} \mu(x) = x_\mu^* . \qquad (3)$$

При решении задачи (2) используются конечные наборы $I = \left\{ x^k = \left(x_1^k, ..., x_n^k\right)^T,\ k = 1,2,..., Mp \right\} \subset D$ возможных решений, называемых *популяциями*, где x^k - хромосома с номером k, M - размер популяции, x_i^k - ген с номером i.

Генетический алгоритм.

Ш.1. Формирование исходной популяции.

Ш.1.1. Задается номер популяции $t = 0$, максимальное количество популяций Np, номер итерации цикла $k = 1$, размер популяции Mp.

Ш.1.2. Случайным образом выбирается начальная точка x^0 - исходная хромосома. Она может быть выбрана как внутренняя точка гиперкуба области допустимых значений D. Из этой точки формируется исходная популяция. Для этого с помощью равномерного распределения на единичном отрезке [0,1] Mp раз генерируется последовательность из n случайных точек $\{P_i^{0k}\}_{i=1,n}^{k-1,Mp},\ i = 1,...,n;\ k = 1,..., Mp$. Используя линейное преобразование, каждая точка отображается на соответствующий ей промежуток $[\alpha, \beta]: P_i^k = (\beta_i - \alpha_i) P_i^{0k} + \alpha_i$. Составляя векторы из точек последовательности $\left\{P_i^k\right\}$ при фиксированных k, получаем Mp начальных векторов $x^k = \left(x_1^k, ..., x_n^k\right)^T,\ x_i^k = P_i^k,\ i = 1,...,n$, координаты которых x_i имеют равномерное распределение на отрезках $[\alpha_i, \beta_i],\ i = 1,...,n$.

Таким образом, может быть сформирована начальная популяция
$I_0 = \left\{ x^k, k = 1,..., Mp \,\middle|\, x^k = \left(x_1^k, x_2^k, ..., x_n^k \right) \in D \right\}$.

Ш.1.3. Вычисляется значение функции фитнеса для каждой особи $x^k \in I_0 : \mu_k = \mu(x^k),\ k = 1,..., Mp$ и популяции I_0 в целом $\mu = \sum_{k=1}^{Mp} \mu_i$.

Ш.2. Отбор (селекция).

Селекция – это операция, которая осуществляет отбор особь (хромосом) x^k в соответствии со значениями функции фитнеса $\mu(x^k)$ для последующего их скрещивания.

Ш.2.1. Вычислить кумулятивную вероятность $q_i = \sum_{j=1}^{i} \mu_j(x^j),\ i = 1,2,..., Mp$.

Ш.2.2. Сформировать случайное действительное число r в интервале $(0, Mp]$.

Ш.2.3. Выбрать i - ю хромосому x^i $(1 \le i \le Mp)$ так, чтобы $q_{i-1} < r \le q_i$.

Ш.2.4. Перейти на Ш.2.2. до тех пор, пока не будет сформирована новая популяция ($while(i \le Mp)$).

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 $Pc \in (0,1]$ как вероятность кроссинговера. Эта вероятность дает ожидаемое число $Pc \cdot Mp$ хромосом, подвергаемых операции кроссинговера.

Ш.3.2. Для операции кроссинговера выполняется процесс, повторяющийся от $i = 1$ до $Pc \cdot Mp$: формируется случайное действительное число r из сегмента $[0,1]$, при этом, если $r < Pc$, то хромосома x^i выбирается как родительская.

Ш.3.3. Отбираются пары родительских хромосом (x^i, x^j), где $i \ne j$. Действие оператора кроссинговера осуществляется следующим образом:

Ш.3.4. Формируется случайное число $c \in (0,1)$, затем оператор кроссинговера, действующий на исходные пары (x^i, x^j), производит две хромосомы потомки X и Y :

$$X = c \cdot x^i + (1-c) \cdot x^j, \quad Y = (1-c) \cdot x^i + c \cdot x^j.$$

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 c 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 $Pm \in (0,1]$ как вероятность мутации. Эта вероятность дает ожидаемое число $Pm \cdot Mp$ хромосом, подвергаемых операции мутации.

Ш.4.2. Для операции мутации выполняется процесс, повторяющийся от $i = 1$ до $Pm \cdot Mp$: формируется случайное действительное число r из сегмента $[0,1]$, при этом, если $r < Pm$, то хромосома x^i выбирается как родительская для операции мутации Для каждой выбранной родительской хромосомы x^i, обозначенной как $Z = (x_1, x_2, ..., x_n)$, производится мутация.

Ш.4.3. Поочередно рассматривается каждый потомок из ожидаемого числа $Pm \cdot Mp$ хромосом. Среди генов выбранной родительской хромосомы $Z = (x_1, x_2, ..., x_n)$ случайно (с вероятностью $1/n$) выбирается один с номером $p \in (1,2,...,n)$ подлежащий замене. Его новое значение x_p^M случайным образом выбирается из промежутка $\lfloor \alpha_p, \beta_p \rfloor$ изменения выбранной координаты x_p.

Ш.5. Формирование новой популяции.

Ш.5.1. С равной вероятностью из потомков мутантов предыдущего шага выбирается один $x^M = (x_1, x_2, ..., x_p^M, ..., x_n)$.

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 $\mu_M = \mu(x^M)$.

Ш.5.4. Проверка условий:

Ш.5.4.1. Если $k < Mp$, то $k = k + 1$ и переход на Ш.2.

Ш.5.4.2. Если $k = Mp$, то $t = t + 1$ и переход на Ш.6.

Ш.6. Проверка условия останова генетического алгоритма.

Условием окончания работы генетического алгоритма является формирование заданного количества популяций $t = Np$.

Ш.6.1. Если условие не выполнено, то полагаем $k = 1$ и переход на Ш.2.

Ш.6.2. Если условие окончания работы выполнено, то в качестве решения (приближенного) задачи $\mu(x_\mu^*) = \max_{x \in D} \mu(x)$ выбирается особь с лучшим значением функции фитнеса из текущей популяции: $x_\mu^* \cong x_\mu^e = Arg \max \mu(x^k)$, а по нему определяется приближенное решение поставленной задачи $f(x^*) = \min f(x) : x^* = x_\mu^*$.

Замечание 1. В качестве *функции фитнеса* можно использовать обратную целевую функцию $\mu(x) = \dfrac{1}{f(x)}$.

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

Замечание 3. Вероятность кроссинговера принимается равной $Pc = 0.3 - 0.5$, вероятность мутации $Pm = 0.05 - 0.2$.