Министерство образования и науки РФ

Федеральное государственное бюджетное образовательное учреждение высшего образования

Томский государственный университет систем управления и радиоэлектроники (ТУСУР)

Факультет систем управления (ФСУ)

Е.Б. Грибанова

Исследование операций и методы оптимизации

Методические указания по выполнению лабораторных работ для студентов заочной формы обучения

Томск – 2017

В методических указаниях представлены задания по лабораторным работам по дисциплине «Исследование операций и методы оптимизации в экономике». Темы лабораторных работ: «Минимизация функции одной переменно», «Минимизация функции нескольких переменных», «Условная оптимизация». Представлены примеры выполнения лабораторных работ с помощью пакетов Excel, MathCad, языка программирования Java.

СОДЕРЖАНИЕ

Оглавление

Введение

Данные методические указания предназначены для выполнения лабораторных работ по дисциплине «Исследование операций и методы оптимизации» и разработаны с учетом требований ФГОС ВО для направления подготовки 09.03.03 «Прикладная информатика».

Цель лабораторных работ: приобретение практических навыков для решения задач условной и безусловной оптимизации путем реализации алгоритмов на языках программирования, а также использования стандартных функций математических пакетов.

Лабораторные работы выполняются в соответствии с порядком, описанном в методических указаниях.

1. Минимизация функции одной переменной

1.1 Методы прямого поиска

1.1.1. Основные понятия

В данной лабораторной работе рассматриваются задачи, в которой целевая функция зависит от одной переменной, а допустимым множеством является отрезок вещественной оси

$$f(x) \to \min;$$

$$x \in [a;b].$$

Число $x^* \in [a;b]$ называется точкой глобального (абсолютного) минимума, или просто точкой минимума, функции $f(x)$ на отрезке $[a;b]$, если $f(x^*) \le f(x)$ для всех $x \in [a;b]$.

Число $x^* \in [a;b]$ называется точкой локального минимума функции $f(x)$ на отрезке $[a;b]$, если $f(x^*) \le f(x)$ для всех $x \in [a;b]$, достаточно близких к x^* (рис.1.1).

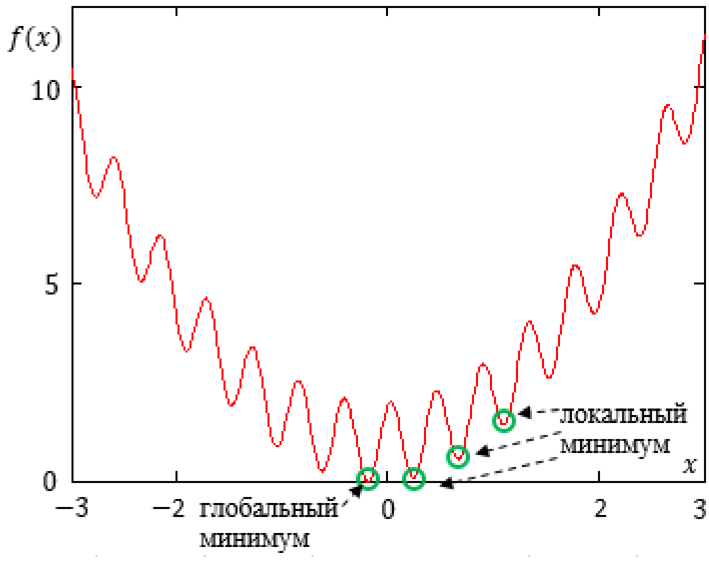


Рис.1.1 График функции $f(x) = \cos(14,5x - 0,3) + x(x + 0,2) + 0,01$

Функция $f(x)$ является унимодальной, если с увеличением x слева от x^* она монотонно убывает, справа - монотонно возрастает.

Если функция $f(x)$ – выпуклая на $[a;b]$, то на любом отрезке $[x';x'']\subset[a;b]$ ее график расположен не выше хорды, проведенной через точки графика с абсциссами x' и x''.

Всякая выпуклая непрерывная на отрезке $[a;b]$ функция является унимодальной (обратное неверно).

В данной работе будут рассмотрены методы поиска минимума выпуклой функции.

Все численные методы поиска минимума функции одной переменной можно разделить на прямые методы и методы первого и более высоких порядков, использующие производные.

Прямые методы используют значения функции в вычисленных точках:

1. метод равномерного поиска;
2. метод дихотомии;
3. метод золотого сечения;
4. Пауэлла;
5. метод Монте-Карло.

Методы первого порядка используют значение или знак производной функции в вычисленных точках:

1. метод Ньютона;
2. метод средней точки.

1.1.2. Метод равномерного поиска

Суть метода: интервал $[a_0, b_0]$ делится на N + 1 равных подинтервалов, в каждой точке (границах подинтервалов) вычисляется значение функции. Выбирается точка, в которой значение функции минимально.

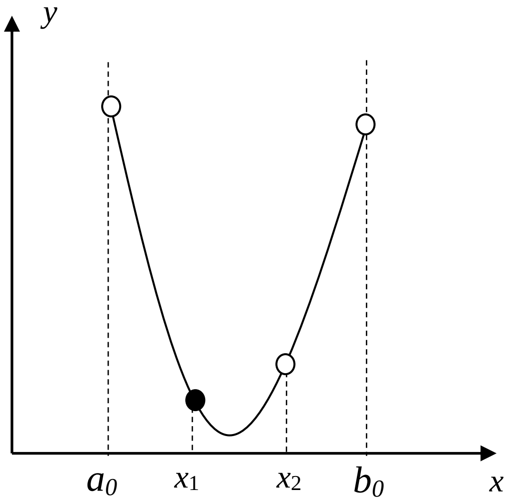


Рис.1.2 Метод равномерного поиска. Разбиение интервала на три под интервала

Алгоритм

Шаг 1. Задать исходные данные: начальный интервал неопределенности $L_0 = [a_0, b_0]$, N - количество вычислений функции.

Шаг 2. Вычислить точки $x_i = a_0 + i\dfrac{(b_0 - a_0)}{N+1}, i = 1..N$, равноотстоящие друг от друга.

Шаг 3. Вычислить значения функции в N найденных точках: $f(x_i)$, $i = 1..N$.

Шаг 4. Среди точек $x_i, i = 1..N$, найти такую, в которой функция принимает наименьшее значение: $f(x_k) = \min f(x_i)$.

1.1.3. Метод дихотомии

Суть метода: вычисляется середина интервала $[a_0, b_0]$ и две точки по обе стороны от этой середины, и рассчитывается значение функции в этих точках. Если значение функции в левой точке меньше, чем значение функции в правой точке, значит, функция возрастает на этом промежутке (рис.1.3), и происходит изменение правой границы. Иначе – функция убывает и происходит изменение левой границы.

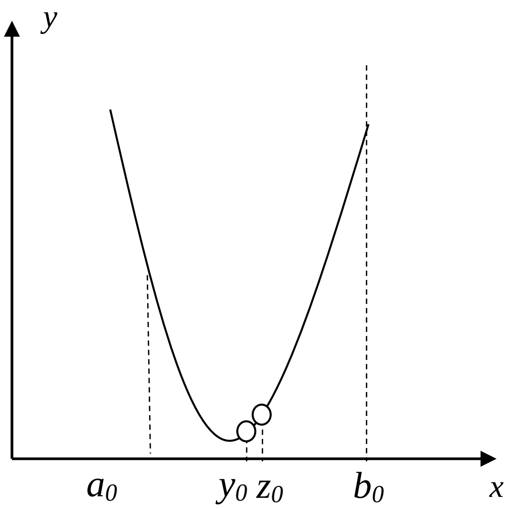


Рис.1.3 Метод дихотомии. Возрастание функции на интервале $[y_0;z_0]$, поэтому правая граница b_0 будет перемещена в точку z_0

Алгоритм

Шаг 1. Задать исходные данные: начальный интервал неопределенности $L_0 = [a_0, b_0]$, $\varepsilon > 0$ -малое число, $l > 0$ – точность ($\varepsilon \in (0,2l)$).

Шаг 2. Положить $k = 0$.

Шаг 3. Вычислить $y_k = \dfrac{a_k + b_k - \varepsilon}{2}$, $f(y_k)$, $z_k = \dfrac{a_k + b_k + \varepsilon}{2}$, $f(z_k)$.

Шаг 4. Сравнить $f(y_k)$ с $f(z_k)$:

а) если $f(y_k) \le f(z_k)$, положить $a_{k+1} = a_k, b_{k+1} = z_k$ и перейти к шагу 5;

б) если $f(y_k) > f(z_k)$, положить $a_{k+1} = y_k, b_{k+1} = b_k$.

Шаг 5. Вычислить $L_k = | b_{k+1} - a_{k+1} |$ и проверить условие окончания:

а) если $L_k \le l$, процесс поиска завершается и в качестве приближенного решения можно взять середину последнего интервала:

$$x^* = \frac{a_{k+1} + b_{k+1}}{2};$$

б) если $L_k > l$, положить $k = k + 1$ и перейти к шагу 3.

1.1.4. Метод золотого сечения

Суть метода: аналогична методу дихотомии, однако две точки определяются в пропорции золотого сечения (рис.1.4).

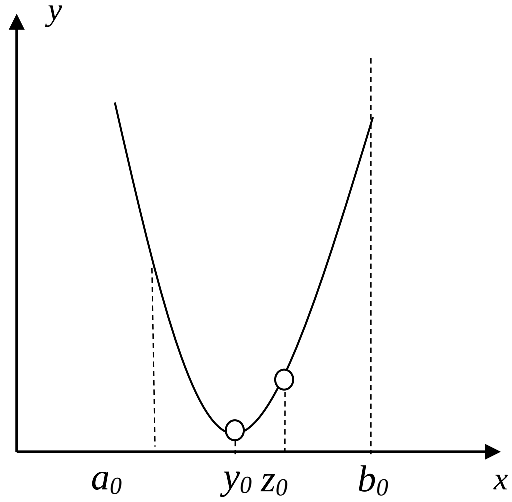


Рис.1.4 Метод золотого сечения. Возрастание функции на интервале [y_0;z_0], поэтому правая граница b_0 будет перемещена в точку z_0

Алгоритм

Шаг 1. Задать исходные данные: начальный интервал неопределенности $L_0 = [a_0, b_0]$, точность l>0.

Шаг 2. Положить k =0.

Шаг 3. Вычислить

$$y_0 = a_0 + \frac{3-\sqrt{5}}{2}(b_0 - a_0); z_0 = a_0 + b_0 - y_0.$$

Шаг 4. Вычислить $f(y_k)$, $f(z_k)$.

Шаг 5. Сравнить $f(y_k)$ с $f(z_k)$:

а) если $f(y_k) \le f(z_k)$, то положить $a_{k+1} = a_k, b_{k+1} = z_k$ и $y_{k+1} = a_{k+1} + b_{k+1} - y_k, z_{k+1} = y_k$. Перейти к шагу 6;

б) если $f(y_k) > f(z_k)$, положить $a_{k+1} = y_k, b_{k+1} = b_k$ и $y_{k+1} = z_k, z_{k+1} = a_{k+1} + b_{k+1} - z_k$.

Шаг 6. Вычислить $L_k = | b_{k+1} - a_{k+1} |$ и проверить условие окончания:

а) если $L_k \le l$, процесс поиска завершается и в качестве приближенного решения можно взять середину последнего интервала:

$$x^* = \frac{a_{k+1} + b_{k+1}}{2};$$

б) если $L_k > l$, положить $k = k + 1$ и перейти к шагу 4.

1.1.5. Метод Пауэлла

Суть метода: определить три точки в направлении уменьшения функции и рассчитать квадратичную аппроксимацию. Сравнить значение функции в наилучшей из трех точек и в точке квадратичной аппроксимации и если условие останова не выполняется, то выбирается наилучшая точка и две точки по обе стороны от неё. Так на рис. 1.5 будет выбрана точка $\bar{x}$ и две точки по обе стороны (x_1, x_2).

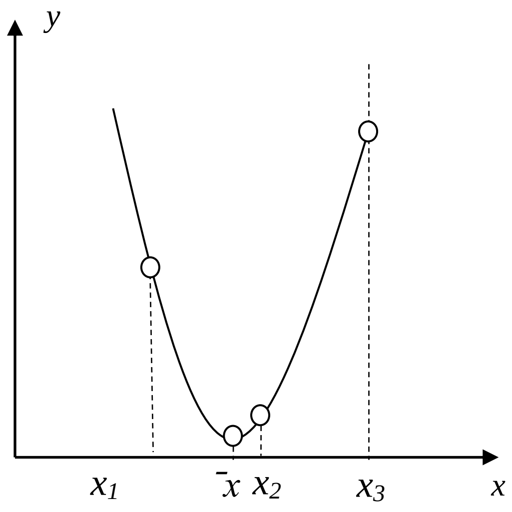


Рис.1.5 Определение точек методом Пауэлла

Алгоритм

Исходные данные: x_1 - начальная точка; Δx - выбранная величина шага по оси x.

Шаг 1: Вычислить $x_2 = x_1 + \Delta x$.

Шаг 2: Вычислить $f(x_1)$ и $f(x_2)$.

Шаг 3: Если $f(x_1) > f(x_2)$, положить $x_3 = x_1 + 2\Delta x$, если $f(x_1) \le f(x_2)$, то $x_3 = x_1 - \Delta x$. Если $x_3 < x_1$, то перенумеровать точки в естественном порядке: $x_1 = x_3$, $x_2 = x_1$, $x_3 = x_2$.

Шаг 4: Вычислить $f(x_3)$ и найти

$$F_{\min} = \min\{f_1, f_2, f_3\}.$$

$X_{\min}$ равно точке x_i, которая соответствует $F_{\min}$.

Шаг 5: По трем точкам x_1, x_2, x_3 вычислить $\bar{x}$, используя квадратичную аппроксимацию

$$a_1 = \frac{f(x_2) - f(x_1)}{x_2 - x_1},$$

$$a_2 = \frac{1}{x_3 - x_2}\left[\frac{f(x_3) - f(x_1)}{x_3 - x_1} - \frac{f(x_2) - f(x_1)}{x_2 - x_1}\right],$$

$$\bar{x} = \frac{x_2 + x_1}{2} - \frac{a_1}{2a_2}.$$

Шаг 6: Проверка на окончание поиска:

а) является ли разность $\left|F_{\min} - f(\bar{x})\right| \le \varepsilon$;

б) является ли разность $\left|X_{\min} - \bar{x}\right| \le \delta$,

где $\varepsilon > 0$ и $\delta > 0$ - заданные точности.

Если условия а) и б) выполняются одновременно, то закончить поиск (в качестве результата взять точку $\bar{x}$). Иначе переход на Шаг 7.

Шаг 7: Выбрать "наилучшую" точку ($X_{\min}$ или $\bar{x}$) и две точки по обе стороны от нее. Обозначить эти точки в естественном порядке и перейти на Шаг 4.

Замечание: после пятого шага необходимо провести дополнительную проверку, т.к. точка $\bar{x}$ может находиться за интервалом (x_1, x_3). В этом случае точка x_1 заменяется $\bar{x}$ и осуществляется переход к шагу 1.

1.1.6. Метод Монте-Карло

Суть метода: на интервале $[a_0, b_0]$ случайным образом генерируются точки, в каждой точке вычисляется значение функции. Выбирается точка, в которой значение функции минимально (рис.1.6).

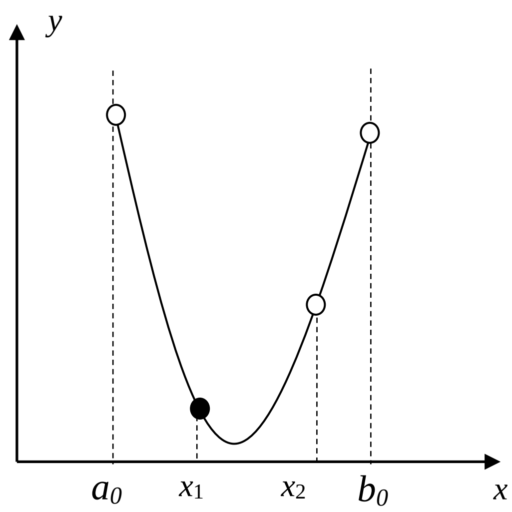


Рис.1.6 Метод Монте-Карло (точки x_1 и x_2 сгенерированы случайным образом)

Алгоритм

Шаг 1. Генерирование на интервале $[a_0, b_0]$ равномерно распределенной случайной величины x.

Шаг 2. Если $f(x) < f_{\min}$ ($f_{\min}$ - минимальное найденное значение функции), то происходит запоминание новой точки в качестве текущего решения $f_{\min} = f(x)$, $x_{\min} = x$.

1.2 Методы, основанные на использовании производных.

1.2.1. Метод Ньютона

Суть метода: для производной функции определяется корень с помощью касательных.

Алгоритм

Шаг 1. Определение начальной точки x_0, точности ε.

Шаг 2. Вычислить новую точку

$$x_{n+1} = x_n - \frac{f'(x_n)}{f''(x_n)}.$$

Шаг 3. Если $|f'(x_{n+1})| \le \varepsilon$, то в качестве решения задачи принимается число $x^* \approx x_{n+1}$, иначе – переход на шаг 2.

1.2.2. Метод средней точки (поиск Больцано)

Суть метода: определение середины z_0 интервала $[a_0, b_0]$ и определение знака производной в данной точке: если производная отрицательна, то функция убывает на интервале $[a_0, z_0]$ и левая граница перемещается в среднюю точку, иначе – функция возрастает $[z_0, b_0]$ и правая граница переходит в среднюю точку (рис.1.7).

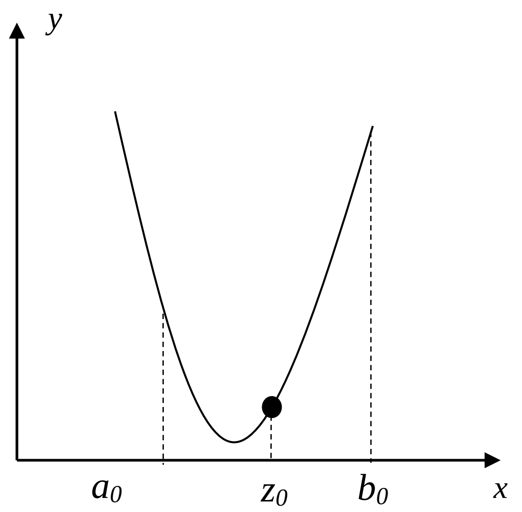


Рис.1.7 Производная в средней точке положительна, следовательно, правая граница будет смещена

Алгоритм

Шаг 1. Задать исходные данные: интервал $L_0 = [a_0, b_0]$, $\varepsilon > 0$ -точность.

Шаг 2. Положить $k = 0$.

Шаг 3. Вычислить $z_k = \dfrac{a_k + b_k}{2}$, $f'(z_k)$.

Шаг 4. Сравнить $f'(z_k)$ с нулем:

а) если $f'(z_k) < 0$, положить $a_{k+1} = z_k, b_{k+1} = b_k$ и перейти к шагу 5;

б) если $f'(z_k) > 0$, положить $a_{k+1} = a_k, b_{k+1} = z_k$.

Шаг 5. Проверить условие окончания:

а) если $|f'(z_k)| \le \varepsilon$, процесс поиска завершается и в качестве приближенного решения можно взять точку $x^* = z_k$;

б) если $|f'(z_k)| > \varepsilon$, положить $k = k + 1$ и перейти к шагу 3.

1.3. Простейшие формулы численного дифференцирования

Вычисление первой производной. В качестве приближенных формул первой производной можно использовать:

$$f'(x) \approx \frac{f(x+h) - f(x)}{h},$$

$$f'(x) \approx \frac{f(x) - f(x-h)}{h}.$$

Здесь h >0 – шаг.

Формула с большей точностью:

$$f'(x) \approx \frac{f(x+h) - f(x-h)}{2h}.$$

Вычисление второй производной. В качестве приближенных формул второй производной можно использовать:

$$f''(x) \approx \frac{f(x-h) - 2f(x) + f(x+h)}{h^2}.$$

1.4. Задание на лабораторную работу №1

1. В соответствии с вашим вариантом постройте график функции. Варианты заданий приведены в Приложении А.

2. Напишите программу определения минимума функции с использованием методов прямого поиска (см. вариант).
3. Напишите программу определения минимума функции с использованием методов с применением производных.
4. Для каждого метода выполните три итерации вручную.
   Для выполнения задания могут быть использованы пакеты (Excel, MathCad), языки программирования (С++, Pascal и др.). Недостающие данные выберите сами. Пример выполнения задачи приведен в Приложении Г.
5. Сравните методы по числу итераций.

2. Минимизация функции нескольких переменных

2.1. Основные понятия

В общем виде задача поиска минимума функции многих переменных может быть записана

$$f(x) \to \min,$$

где $f(x)$ - целевая функция многих переменных;

$x = (x_1, x_2, ..., x_n)$ - n -мерный вектор оптимизируемых параметров;

n - размерность задачи.

Методы многомерной оптимизации можно разделить на следующие группы:

1. Прямые методы, использующие значения функции в вычисленных точках.
2. Градиентные методы, использующие значения градиента.

2.2. Прямые методы

2.2.1. Метод Гаусса

Суть метода: последовательно осуществляется одномерная оптимизация по каждой переменной.

Алгоритм

Шаг 1. Задается начальная точка x_0, точность $\varepsilon_1, \varepsilon_2$.

Шаг 2. Выполняется одномерная оптимизация по каждой переменной.

Шаг 3. Проверка условия $\left\| x^j - x^{j-1} \right\| \le \varepsilon_1$, $\left| f(x^j) - f(x^{j+1}) \right| \le \varepsilon_2$. Если условия выполняются, то осуществляется завершение работы алгоритма, иначе – переход на шаг 2.

2.2.2. Метод Хука-Дживса

Суть метода: нахождение в окрестности текущей точки наилучшей и движение в этом направлении. Если значение в окрестных точках больше чем в текущей, то происходит уменьшение шага.

Процедура Хука–Дживса представляет собой комбинацию двух поисков:

а) **"Исследующий" поиск**: с заданным шагом Δ_i происходит расчет функции в пробных точках вокруг некоторой исходной точки x^0 ($f(x_0 \pm \Delta_j)$) (рис.2.1). Если значение ЦФ в пробной точке меньше значения ЦФ в исходной точке, то шаг поиска успешный. В противном случае из исходной точки делается шаг в противоположном направлении. После перебора всех n координат исследующий поиск завершается. Полученная точка называется базовой.

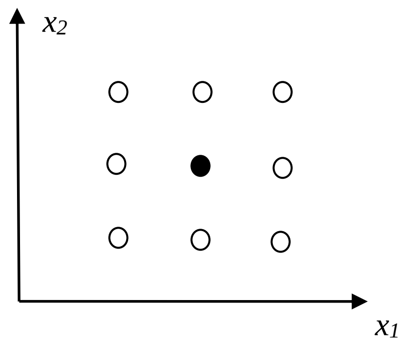


Рис.2.1 Исследующий поиск

б) **Ускоряющий поиск** по образцу: осуществляется шаг из полученной базовой точки вдоль прямой, соединяющей эту точку с предыдущей базовой (рис.2.2). Новая точка образца определяется по формуле:

$$x_p^{k+1} = x^k + (x^k - x^{k-1}) .$$

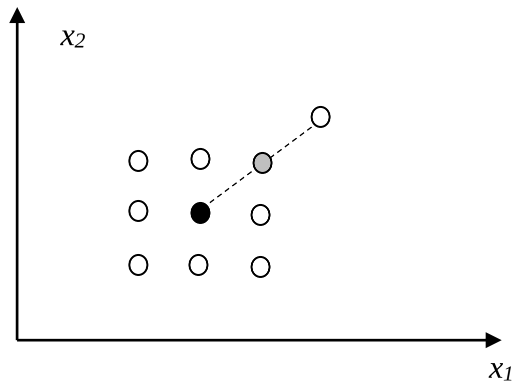


Рис.2.2 Поиск по образцу

Алгоритм

Введем следующие обозначения: x^k– текущая базовая точка; x^{k-1}– предыдущая базовая точка; x_p^{k+1}– точка, построенная при движении по образцу; x^{k+1}– следующая (новая) базовая точка.

Критерий останова: $\|\Delta x\| \le \varepsilon$.

Шаг 1. Определить начальную точку x^0; приращения (шаги) $\Delta_i,\ i = \overline{1, n}$; коэффициент уменьшения шага $\alpha > 1$; параметр окончания поиска ε.

Шаг 2. Провести исследующий поиск.

Шаг 3. Был ли исследующий поиск удачным (найдена ли точка с меньшим значением ЦФ)?

Да: переход на Шаг 5. Нет: продолжить, т.е. переход на Шаг 4.

Шаг 4. Проверка на окончание поиска. Выполняется ли неравенство $\|\Delta x\| \le \varepsilon$? Да: окончание поиска, т.е. текущая точка аппроксимирует точку экстремума x^*.

Нет: уменьшить приращение $\Delta_i / \alpha;\ i = 1, 2, \dots, n$. Переход на Шаг 2.

Шаг 5. Провести поиск по образцу: $x_p^{k+1} = x^k + (x^k - x^{k-1})$.

Шаг 6. Провести исследующий поиск, используя точку x_p^{k+1} в качестве временной базовой точки. Пусть в результате получена точка x^{k+1}.

Шаг 7. Выполняется ли неравенство: $f(x^{k+1}) < f(x^k)$?

Да: положить $x^{k-1} = x^k$; $x^k = x^{k+1}$. Переход на Шаг 5.

Нет: переход на Шаг 4.

2.2.3. Симплексный метод

Суть метода: приближение к минимальной точке с помощью изменения координат вершин симплекса (рис.2.3).

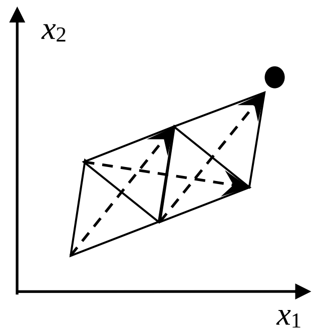


Рис.2.3 Движение симплекса

Шаг 1. Задается исходная вершина симплекса.

$$x^0 = (x_1^0, ..., x_n^0)$$

Задается коэффициент сжатия $\gamma \in [0,1]$ и размер симплекса L. Строится симплекс

$$(x_i^j) = \begin{pmatrix} x_1^0 & ... & x_n^0 \\ x_1^1 & ... & x_n^1 \\ ... & ... & ... \\ x_1^{n-1} & ... & x_n^{n-1} \\ x_1^n & ... & x_n^n \end{pmatrix}.$$

Здесь j-я строка – это координаты j-ой вершины V_j. ($j = 1,...,n+1$), где n - размерность пространства (размерность вектора x), i – номер координаты $i = 1,...,n$.

Определение координат x_i^j, начиная со второй, производится по формуле

$$x_i^j = x_i^0 + \tilde{x}_i^j,\ (j = 1,...,n;\ i = 1,...,n)\,, \qquad (2.1)$$

где $\tilde{x}_i^j$ - матрица размерности $(n+1)\times n$

$$(\tilde{x}_i^j) = \begin{pmatrix} 0 & 0 & 0 & ... & 0 \\ p_n & q_n & q_n & ... & q_n \\ q_n & p_n & q_n & ... & q_n \\ ... & ... & ... & ... & ... \\ q_n & q_n & q_n & ... & p_n \end{pmatrix},$$

где

$$p_n = \frac{L}{n\sqrt{2}}(\sqrt{n+1} + n - 1)\,, \qquad q_n = \frac{L}{n\sqrt{2}}(\sqrt{n+1} - 1)\,.$$

Векторы соответствующие вершинам $V_1,...,V_n$, определяемые формулой (2 .1), составят одинаковые углы с координатными осями $x_1,...,x_n$.

Шаг 2. В вершинах симплекса вычисляется ЦФ $f(x^j),\ j = 0,...,n$

Шаг 3. Проверяем условия: $\left\|x^j - x^{j-1}\right\| \le \varepsilon_1,\ \left|f(x^j) - f(x^{j+1})\right| \le \varepsilon_2$

Если «да», то конец; если «нет», то переходим на Шаг 4.

Шаг 4. Находится «наихудшая» вершина симплекса (при поиске минимума «наихудшая» вершина – та, в которой значение функции максимально).

$$f(x^p) = \max_j \left\{ f(x^j),\ j = \overline{1, n+1} \right\}$$

Шаг 5. Осуществляется расчет координат новой вершины (вершина отражения x^p):

$$\tilde{x}^p = \frac{2}{n}\left(\sum_{j=0}^{n} x^j - x^p\right) - x^p.$$

Шаг 6. Если точка $\tilde{x}^p$ оказывается «хуже» всех остальных точек симплекса, то осуществляется возврат к исходному симплексу с последующим его сжатием относительно «лучшей» из вершин x^k

$$f(x^k) = \min_j \left\{ f(x^j),\ j = \overline{1, n+1} \right\}$$

$$\tilde{x}^s = \gamma x^k + (1-\gamma) x^s,\ s = 0,1,\ldots,n;\ s \neq k\ ,$$

Переход на Шаг 2.

Если $\tilde{x}^p$ не является «худшей» в новом симплексе, то перейти на шаг 3.

2.3. Градиентные методы

Градиент функции $f(x)$ многих переменных в некоторой точке x - это вектор, координатами которого являются частные производные функции в этой точке:

$$\nabla f(x) = \left(\frac{\partial f(x)}{\partial x_1}, \frac{\partial f(x)}{\partial x_2}, \ldots, \frac{\partial f(x)}{\partial x_n}\right).$$

В малой окрестности точки x градиент указывает направление наискорейшего возрастания функции, а его норма характеризует скорость этого возрастания. Вектор-антиградиент указывает направление наискорейшего убывания функции.

В любой точке поверхности целевой функции $f(x)$ вектор-антиградиент перпендикулярен касательной к линии уровня $f(x) = const$ в этой точке.

Норма вектора-градиента

$$\|\nabla f(x)\| = \sqrt{\left(\frac{\partial f(x)}{\partial x_1}\right)^2 + \left(\frac{\partial f(x)}{\partial x_2}\right)^2 + \ldots + \left(\frac{\partial f(x)}{\partial x_n}\right)^2}\ .$$

В точке, где имеет место экстремум функции, вектор-градиент и все его компоненты обращаются в ноль $f'(x^*) = (0;0;\ldots;0)$.

Матрица Гессе функции $f(x)$ многих переменных - это матрица вторых производных:

$$H(x)=\nabla^2 f(x)=\begin{pmatrix} \frac{\partial^2 f(x)}{\partial x_1 \partial x_1} & \frac{\partial^2 f(x)}{\partial x_1 \partial x_2} & \dots & \frac{\partial^2 f(x)}{\partial x_1 \partial x_n} \\ \frac{\partial^2 f(x)}{\partial x_2 \partial x_1} & \frac{\partial^2 f(x)}{\partial x_2 \partial x_2} & \dots & \frac{\partial^2 f(x)}{\partial x_2 \partial x_n} \\ \dots & \dots & \dots & \dots \\ \frac{\partial^2 f(x)}{\partial x_n \partial x_1} & \frac{\partial^2 f(x)}{\partial x_n \partial x_2} & \dots & \frac{\partial^2 f(x)}{\partial x_n \partial x_n} \end{pmatrix}.$$

Критерии останова алгоритмов, использующих градиенты, (наиболее употребительные).

Пусть $\varepsilon > 0$ - заданная точность:

1) $\max\limits_{i=\overline{1,n}} \left| \frac{\partial f}{\partial x_i} \right| \le \varepsilon$; 2) $\|\nabla f\|^2 = \sum\limits_{i=1}^{n} \left(\frac{\partial f}{\partial x_i} \right)^2 \le \varepsilon$;

3) $\left| f(x^{k+1}) - f(x^k) \right| \le \varepsilon$; 4) $\|\nabla f\| = \sqrt{\sum\limits_{i=1}^{n} \left(\frac{\partial f}{\partial x_i} \right)^2} \le \varepsilon$;

5) $\|\Delta x\| = \left\| x^{k+1} - x^k \right\| \le \varepsilon$.

2.3.1 Метод градиентного спуска

Суть метода: происходит движение в направлении антиградиента с заданным параметром спуска.

Алгоритм

Шаг 1. Задать исходные данные: параметр спуска β, начальная точка x^0, точность ε.

Шаг 2. Вычислить новую точку по формуле:

$$x^{k+1} = x^k - \beta \nabla f\left(x^k\right).$$

Шаг 3. Выполнить проверку завершения алгоритма с использованием одного из критериев. Если условие выполняется, то работа алгоритма завершается, иначе – происходит переход на шаг 2.

2.3.2. Метод Коши

Суть метода: движение в направлении антиградиента путем осуществления одномерной минимизации.

Алгоритм

Шаг 1. Выбрать начальную точку x^0.

Шаг 2. На k-ой итерации, где $d_k = -\nabla f\left(x^k\right)$, найти такое λ_k, что

$$f\left(x^k + \lambda_k d_k\right) = \min_{\lambda \geq 0} f\left(x^k + \lambda d_k\right).$$

Положить $x^{k+1} = x^k + \lambda_k d_k$.

Шаг 3. Проверка критерия останова.

Да: окончание поиска $\rightarrow$ конец. Нет: $k = k+1$, $\rightarrow$ Ш. 2.

2.3.3. Метод Ньютона

Суть метода: решение системы уравнений (приравненные к нулю частные производные) методом Ньютона.

Алгоритм

Шаг 1. Задать исходные данные: начальная точка x^0, точность ε.

Шаг 2. Выполнить расчет новой точки по итерационной формуле:

$$x^{k+1} = x^k - \left[\nabla^2 f\left(x^k\right)\right]^{-1} \nabla f\left(x^k\right).$$

Шаг 3. Проверить условие останова. При выполнении условия работа алгоритма завершается, иначе – переход на шаг 2.

2.4. Задание

1. В соответствии с вашим вариантом постройте график функции. Варианты заданий приведены в приложении Б.

2. Напишите программу определения минимума функции с использованием метода прямого поиска (см. вариант).
3. Напишите программу определения минимума функции с использованием градиентного метода.
4. Для каждого метода выполните три итерации вручную.

   Для выполнения задания могут быть использованы пакеты (Excel, MathCad), языки программирования (C++, Pascal и др.). Пример выполнения лабораторной работы приведен в Приложении Г.

3. Условная оптимизация

3.1. Задача линейного программирования

3.1.1. Постановка задачи о диете

Задача о диете заключается в определении такого набора продуктов (количества продуктов каждого вида), чтобы его стоимость была минимальной и при этом удовлетворялась потребность человека в питательных веществах. Примем следующие обозначения: x_j $(j = 1,...,n)$ – число единиц товара j- го вида; $b_i\,(i = 1,...,m)$ – необходимый минимум содержания в рационе питательного вещества S_i; a_{ij} – число единиц питательного вещества S_i в единице продукта j-го вида; c_j – стоимость единицы продукта j-го вида.

Тогда экономико-математическая модель задачи примет вид:

найти такой рацион $X = (x_1, x_2,..., x_n)$, удовлетворяющий системе

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + ... + a_{1n}x_n \ge b_1, \\ a_{21}x_1 + a_{12}x_2 + ... + a_{2n}x_n \ge b_2, \\ .... \\ a_{m1}x_1 + a_{m2}x_2 + ... + a_{mn}x_n \ge b_m \end{cases}$$

и условию

$$x_1 \ge 0,\ x_2 \ge 0,..., x_n \ge 0,$$

при котором функция

$$f(x) = c_1x_1 + c_2x_2 + ... + c_nx_n$$

принимает **минимальное** значение.

3.1.2 Постановка транспортной задачи

Транспортная задача заключается в определении такого плана перевозок (количества товара, доставляемого из пунктов производства в пункты потребления), чтобы стоимость доставки была минимальной. В m пунктах отправления А1, А2, …, Аm, которые в дальнейшем будем называть поставщиками, сосредоточено определенное количество единиц некоторого однородного продукта, которое обозначим a_i $(i = 1,...,m)$. Данный продукт потребляется в n пунктах B1, B2, … , Bn, которые будем называть потребителями; объем потребления обозначим b_j $(j = 1,...,n)$. Если объем производства равен объему потребления, то такая модель называется сбалансированной. Обозначим количество продукта, перевозимого из пункта A_i в пункт B_j, через x_{ij}. Совокупность всех переменных x_{ij} для краткости обозначим x, тогда целевая функция задачи будет иметь вид

$$f(x) = \sum_{i=1}^{m}\sum_{j=1}^{n} c_{ij}x_{ij} \to \min$$

а ограничения выглядят следующим образом:

$$\sum_{j=1}^{n} x_{ij} = a_i;\ i = 1,...,m,$$

$$\sum_{i=1}^{m} x_{ij} = b_j;\ j = 1,...,n$$

$$x_{ij} > 0.$$

3.2. Задание

3.2.1. Задача о диете

1) В таблице 3.1 представлена информация о некоторых продуктах: количестве белка, жиров и углеводов, содержащихся в них, а также калорийность и цена (за 100 г.). Необходимо сформировать дневной рацион из 10-15 продуктов (выбрать на свое усмотрение), считая, что суточная потребность человека в белке, жирах, углеводах и энергии составляет соответственно 60 г., 70 г., 280 г. и 1826 килокалорий.

2) Добавьте условие разнообразия рациона, согласно которому человек не может в день употреблять более 400 г. продукта одно вида.

3) Рассчитайте стоимость полученного набора продуктов в первом и во втором случае.

Для решения этих задач необходимо построить математическую модель и реализовать ее в пакете MathCad, Excel или с помощью математических библиотек.

Пример выполнения лабораторной работы представлен в приложении Г. В приложениях Д, Ж приводится описание решения оптимизационных задач в Excel и MathCad.

Таблиц 3.1 Состав продуктов

	Название продукта	**Белки**	**Жиры**	**Углеводы**	**Ккал**	**Цена**
1	Яйцо куриное	12.7	11.5	0.7	157	15
2	Арахис	26.3	45.2	9.7	550	21
3	Горох цельный	23.0	1.2	53.3	316	30
4	Грецкий орех	13.8	61.3	10.2	647	44
5	Крупа гречневая	12.6	2.6	68.0	345	6
6	Пшено	12.0	2.9	69.3	351	3,2
7	Рис	8.0	1.0	76.0	345	6
8	Творог	7.1	23.0	27.5	345	13
9	Сыр	27.0	40.0	0.0	468	18
10	Колбаса вареная Любительская	12.2	28.0	0.0	300	17
11	Колбаса варено-копченая Сервелат	28.2	27.5	0.0	360	18
12	Сосиски Молочные	12.3	25.3	0.0	276	18
13	Говядина	18.9	12.4	0.0	187	21
14	Свинина	16.4	27.8	0.0	315	23
15	Кабачки	0.6	0.3	5.7	27	2
16	Капуста белокочанная	1.8	0.0	5.4	28	1,8
17	Картофель	2.0	0.1	19.7	87	2,5
18	Морковь	1.3	0.1	7.0	34	2
19	Огурцы	0.8	0.0	3.0	15	3
20	Перец красный сладкий	1.3	0.0	5.7	28	6,5
21	Свекла	1.7	0.0	10.8	50	2
22	Горбуша	21.0	7.0	0.0	147	12
23	Икра осетровая зернистая	28.9	9.7	0.0	202	164
24	Скумбрия	18.0	9.0	0.0	153	15
25	Шоколад темный	5.4	35.3	52.6	549	30
26	Груша	2.3	0.0	62.1	257	9
27	Персики	3.0	0.0	68.5	286	10

28	Яблоки	3.2	0.0	68.0	284	7
29	Апельсин	0.9	0.0	8.4	37	7,5
30	Бананы	1.5	0.0	22.0	94	4
31	Черешня	1.1	0.0	12.3	53	18
32	Макаронные изделия	11.0	0.9	74.2	348	3
33	Хлеб пшеничный из муки 1 сорта	7.7	2.4	53.4	266	3

3.2.2. Транспортная задача

Заводы производственной фирмы (производство офисных кресел) расположены в городах Омск, Новосибирск, Томск. Центры распределения расположены в городах Нижний Новгород, Пермь, Краснодар. Объемы производства и величина спроса в пунктах представлены в приложении В. Одно изделие имеет вес 3 кг. и объем 0,8 м3. Стоимость перевозки рассчитайте с помощью онлайн-калькулятора http://www.jde.ru/calc.

Составьте экономико-математическую модель задачи. С помощью пакета *MathCAD, Excel* или с помощью математических библиотек найдите оптимальное распределение поставок и минимальные затраты на перевозку.

Пример выполнения лабораторной работы представлен в приложении Г. В приложениях Д, Ж приводится описание решения оптимизационных задач в Excel и MathCad.

Литература

1. Мицель А.А. Исследование операций и методы оптимизации в экономике. Часть 1. Лекционный курс. Томск. 2016. – 146 с. [Электронный ресурс]. Режим доступа –https://edu.tusur.ru/publications/6474.
2. Мицель, А. А. Исследование операций и методы оптимизации в экономике. Лабораторный практикум: Методические указания по выполнению лабораторных работ [Электронный ресурс] / Мицель А. А. — Томск: ТУСУР, 2016. — 62 с. — Режим доступа: https://edu.tusur.ru/publications/6475.
3. Мицель А.А., Шелестов А.А. Методы оптимизации: Учебное пособие. – Томск: Изд-во Томск. гос.ун-та систем управления и радиоэлектроники, 2004. – 256 с.
4. Кремер Н.Ш., Путко Б.А., Тришин И.М., Фридман М.Н. Исследование операций в экономике: Учеб.пособие для вузов. – М.: Юрайт, 2011. – 430 с.

Приложение А. Варианты заданий к лабораторной работе №1 «Минимизация функции одной переменной»

Варианты	Задача	Методы прямого поиска	Методы с использованием производным
1	$f(x)=(x-4)^2,\ x\in[0;10],\ \varepsilon=0{,}1,\ \ x_0=0$	Равномерный поиск, Пауэлла	Средней точки
2	$f(x)=(x-4)^2,\ x\in[0;10],\ \varepsilon=0{,}1,\ \ x_0=0$	Дихотомии, Пауэлла	Ньютона
3	$f(x)=(x-5)^2,\ x\in[0;15],\ \varepsilon=0{,}1,\ \ x_0=0$	Золотого сечения, Пауэлла	Средней точки
4	$f(x)=(x-5)^2,\ x\in[0;15],\ \varepsilon=0{,}1,\ \ x_0=0$	Монте-карло, Пауэлла	Ньютона
5	$f(x)=(x-3)^2,\ x\in[0;10],\ \varepsilon=0{,}1,\ \ x_0=0$	Равномерный поиск, Пауэлла	Средней точки
6	$f(x)=(x-3)^2,\ x\in[0;10],\ \varepsilon=0{,}1,\ \ x_0=0$	Дихотомии, Пауэлла	Ньютона
7	$f(x)=(x-9)^2,\ x\in[0;15],\ \varepsilon=0{,}1,\ \ x_0=0$	Золотого сечения, Пауэлла	Средней точки
8	$f(x)=(x-9)^2,\ x\in[0;15],\ \varepsilon=0{,}1,\ \ x_0=0$	Монте-карло, Пауэлла	Ньютона
9	$f(x)=(x-7)^2,\ x\in[0;10],\ \varepsilon=0{,}1,\ \ x_0=0$	Равномерный поиск, Пауэлла	Средней точки
10	$f(x)=(x-7)^2,\ x\in[0;10],\ \varepsilon=0{,}1,\ \ x_0=0$	Дихотомии, Пауэлла	Ньютона
11	$f(x)=(x-10)^2,\ x\in[0;15],\ \varepsilon=0{,}1,\ \ x_0=0$	Золотого сечения, Пауэлла	Средней точки
12	$f(x)=(x-10)^2,\ x\in[0;15],\ \varepsilon=0{,}1,\ \ x_0=0$	Монте-карло, Пауэлла	Ньютона
13	$f(x)=(x-2)^2,\ x\in[0;7],\ \varepsilon=0{,}1,\ \ x_0=0$	Равномерны	Средней точки

		й поиск, Пауэлла	
14	$f(x)=(x-2)^2,\ x\in[0;8],\ \varepsilon=0{,}1,\ \ x_0=0$	Дихотомии, Пауэлла	Ньютона
15	$f(x)=(x-1)^2,\ x\in[0;6],\ \varepsilon=0{,}1,\ \ x_0=0$	Золотого сечения, Пауэлла	Средней точки
16	$f(x)=(x-1)^2,\ x\in[0;6],\ \varepsilon=0{,}1,\ \ x_0=0$	Монте-карло, Пауэлла	Ньютона
17	$f(x)=(x-11)^2,\ x\in[0;15],\ \varepsilon=0{,}1,\ \ x_0=0$	Равномерный поиск, Пауэлла	Средней точки
18	$f(x)=(x-11)^2,\ x\in[0;15],\ \varepsilon=0{,}1,\ \ x_0=0$	Дихотомии, Пауэлла	Ньютона
19	$f(x)=(x-8)^2,\ x\in[0;10],\ \varepsilon=0{,}1,\ \ x_0=0$	Золотого сечения, Пауэлла	Средней точки
20	$f(x)=(x-8)^2,\ x\in[0;10],\ \varepsilon=0{,}1,\ \ x_0=0$	Монте-карло, Пауэлла	Ньютона
21	$f(x)=(x-14)^2,\ x\in[0;20],\ \varepsilon=0{,}1,\ \ x_0=0$	Равномерный поиск, Пауэлла	Средней точки
22	$f(x)=(x-14)^2,\ x\in[0;20],\ \varepsilon=0{,}1,\ \ x_0=0$	Дихотомии, Пауэлла	Ньютона
23	$f(x)=(x-8)^2,\ x\in[3;10],\ \varepsilon=0{,}1,\ \ x_0=0$	Золотого сечения, Пауэлла	Средней точки
24	$f(x)=(x-8)^2,\ x\in[3;10],\ \varepsilon=0{,}1,\ \ x_0=0$	Монте-карло, Пауэлла	Ньютона
25	$f(x)=(x-2)^2,\ x\in[0;5],\ \varepsilon=0{,}1,\ \ x_0=0$	Равномерный поиск, Пауэлла	Средней точки
26	$f(x)=(x-2)^2,\ x\in[0;7],\ \varepsilon=0{,}1,\ \ x_0=0$	Дихотомии, Пауэлла	Ньютона
27	$f(x)=(x-3)^2,\ x\in[0;8],\ \varepsilon=0{,}1,\ \ x_0=0$	Золотого сечения, Пауэлла	Средней точки
28	$f(x)=(x-3)^2,\ x\in[0;8],\ \varepsilon=0{,}1,\ \ x_0=0$	Монте-карло, Пауэлла	Ньютона

29	$f(x)=\left(x-4\right)^2,\ x\in[0;9],\ \varepsilon=0{,}1,\ \ x_0=0$	Равномерный поиск, Пауэлла	Средней точки
30	$f(x)=\left(x-4\right)^2,\ x\in[0;9],\ \varepsilon=0{,}1,\ \ x_0=0$	Дихотомии, Пауэлла	Ньютона

Приложение Б Варианты заданий к лабораторной работе №2 «Минимизация функции нескольких переменных»

Вариант	Задача	Метод прямого поиска	Градиентный метод
1	$f(x_1,x_2)=4(x_1-5)^2+(x_2-6)^2$; $x^0=[0,0]$; $\varepsilon=0{,}1$	Гаусса	Коши
2	$f(x_1,x_2)=2(x_1-4)^2+(x_2-6)^2$; $x^0=[0,0]$; $\Delta x=[2{,}2]$; $\alpha=2$; $\varepsilon=0{,}1$	Хука-Дживса	Коши
3	$f(x_1,x_2)=2(x_1-5)^2+(x_2-6)^2$; $x^0=[0,0]$; $\beta=0{,}1$; $\varepsilon=0{,}1$	Гаусса	Градиентный спуск
4	$f(x_1,x_2)=2(x_1-7)^2+(x_2-6)^2$; $x^0=[0,0]$; $\Delta x=[2{,}2]$; $\alpha=2$; $\varepsilon=0{,}1$	Хука-Дживса	Ньютона
5	$f(x_1,x_2)=3(x_1-5)^2+(x_2-4)^2$; $x^0=[0,0]$; $\beta=0{,}1$; $\gamma=0{,}5$ $\varepsilon=0{,}1$, $L=1$	Симплексный	Градиентный спуск
6	$f(x_1,x_2)=2(x_1-7)^2+(x_2-6)^2$; $x^0=[1,2]$; $\gamma=0{,}5$; $L=1$; $\varepsilon=0{,}1$	Симплексный	Коши
7	$f(x_1,x_2)=2(x_1-4)^2+(x_2-6)^2$; $x^0=[7,7]$; $\varepsilon=0{,}1$	Гаусса	Ньютона
8	$f(x_1,x_2)=3(x_1-5)^2+(x_2-4)^2$; $x^0=[1,1]$; $\Delta x=[2{,}2]$; $\alpha=2$; $\beta=0{,}1$; $\varepsilon=0{,}1$	Хука-Дживса	Градиентный спуск
9	$f(x_1,x_2)=2(x_1-4)^2+(x_2-6)^2$; $x^0=[0,1]$; $\gamma=0{,}5$; $L=1$; $\varepsilon=0{,}1$	Симплексный	Ньютона
10	$f(x_1,x_2)=4(x_1-3)^2+(x_2-6)^2$; $x^0=[0,0]$; $\varepsilon=0{,}1$	Гаусса	Коши
11	$f(x_1,x_2)=2(x_1-2)^2+(x_2-6)^2$; $x^0=[0,0]$; $\Delta x=[2{,}2]$; $\alpha=2$; $\varepsilon=0{,}1$	Хука-Дживса	Коши
12	$f(x_1,x_2)=2(x_1-7)^2+(x_2-4)^2$; $x^0=[0,0]$; $\beta=0{,}1$; $\varepsilon=0{,}1$	Гаусса	Градиентный спуск
13	$f(x_1,x_2)=2(x_1-1)^2+(x_2-6)^2$; $x^0=[0,0]$; $\Delta x=[2{,}2]$; $\alpha=2$; $\varepsilon=0{,}1$	Хука-Дживса	Ньютона
14	$f(x_1,x_2)=3(x_1-5)^2+(x_2-9)^2$; $x^0=[0,4]$; $\beta=0{,}1$; $\gamma=0{,}5$ $\varepsilon=0{,}1$, $L=1$	Симплексный	Градиентный спуск
15	$f(x_1,x_2)=2(x_1-3)^2+(x_2-4)^2$; $x^0=[1,2]$;	Симплексный	Коши

	$\gamma = 0,5$; $L = 1$; $\varepsilon = 0,1$		
16	$f(x_1, x_2) = 2(x_1 - 7)^2 + (x_2 - 1)^2$; $x^0 = [0, -7]$; $\varepsilon = 0,1$	Гаусса	Ньютона
17	$f(x_1, x_2) = 3(x_1 - 3)^2 + (x_2 - 5)^2$; $x^0 = [1,1]$; $\Delta x = [2,2]$; $\alpha = 2$; $\beta = 0,1$; $\varepsilon = 0,1$	Хука-Дживса	Градиентный спуск
18	$f(x_1, x_2) = 2(x_1 - 7)^2 + (x_2 - 7)^2$; $x^0 = [0,1]$; $\gamma = 0,5$; $L = 1$; $\varepsilon = 0,1$	Симплексный	Ньютона
19	$f(x_1, x_2) = 4(x_1 - 4)^2 + (x_2 - 4)^2$; $x^0 = [1, 2]$; $\varepsilon = 0,1$	Гаусса	Коши
20	$f(x_1, x_2) = 2(x_1 - 4)^2 + (x_2 - 3)^2$; $x^0 = [1, 0]$; $\Delta x = [2,2]$; $\alpha = 2$; $\varepsilon = 0,1$	Хука-Дживса	Коши
21	$f(x_1, x_2) = 2(x_1 - 9)^2 + (x_2 - 8)^2$; $x^0 = [3,3]$; $\beta = 0,1$; $\varepsilon = 0,1$	Гаусса	Градиентный спуск
22	$f(x_1, x_2) = 2(x_1 - 7)^2 + (x_2 - 7)^2$; $x^0 = [2,3]$; $\Delta x = [2,2]$; $\alpha = 2$; $\varepsilon = 0,1$	Хука-Дживса	Ньютона
23	$f(x_1, x_2) = 3(x_1 - 5)^2 + (x_2 - 5)^2$; $x^0 = [1, 2]$; $\beta = 0,1$; $\gamma = 0,5$ $\varepsilon = 0,1$, $L = 1$	Симплексный	Градиентный спуск
24	$f(x_1, x_2) = 2(x_1 - 4)^2 + (x_2 - 5)^2$; $x^0 = [0,1]$; $\gamma = 0,5$; $L = 1$; $\varepsilon = 0,1$	Симплексный	Коши
25	$f(x_1, x_2) = 2(x_1 - 4)^2 + (x_2 - 4)^2$; $x^0 = [0, 0]$; $\varepsilon = 0,1$	Гаусса	Ньютона
26	$f(x_1, x_2) = 3(x_1 - 2)^2 + (x_2 - 4)^2$; $x^0 = [0,1]$; $\Delta x = [2,2]$; $\alpha = 2$; $\beta = 0,1$; $\varepsilon = 0,1$	Хука-Дживса	Градиентный спуск
27	$f(x_1, x_2) = 2(x_1 - 7)^2 + (x_2 - 7)^2$; $x^0 = [2,1]$; $\gamma = 0,5$; $L = 1$; $\varepsilon = 0,1$	Симплексный	Ньютона
28	$f(x_1, x_2) = 4(x_1 - 4)^2 + (x_2 - 7)^2$; $x^0 = [0, 4]$; $\varepsilon = 0,1$	Гаусса	Коши
29	$f(x_1, x_2) = 2(x_1 - 7)^2 + (x_2 - 4)^2$; $x^0 = [1,1]$; $\Delta x = [2,2]$; $\alpha = 2$; $\varepsilon = 0,1$	Хука-Дживса	Коши
30	$f(x_1, x_2) = 2(x_1 - 7)^2 + (x_2 - 5)^2$; $x^0 = [0, 2]$; $\beta = 0,1$; $\varepsilon = 0,1$	Гаусса	Градиентный спуск

Приложение В. Варианты заданий к лабораторной работа №3 «Условная оптимизация». Транспортная задача

Вариант	Объем производства в пункте			Величина спроса в пункте		
	Омск	Новосибирск	Томск	Нижний Новгород	Пермь	Краснодар
1	1000	2000	1200	2000	1100	1100
2	1500	1000	800	2000	900	400
3	1400	700	1000	900	800	1400
4	1300	1200	1100	1000	1500	1100
5	700	600	500	1000	200	60
6	1400	1700	1600	2000	1000	1700
7	1300	1700	1600	1900	1000	1700
8	1400	1800	1600	2000	1800	1700
9	2000	1700	1600	2000	1000	2300
10	1000	1700	1600	1600	1000	1700
11	1000	1200	1600	1600	1000	1200
12	1500	1700	1600	2100	1000	1700
13	1000	1700	1600	1600	1000	1700
14	700	1700	1600	1600	1000	1400
15	1000	2000	1600	1600	1300	1700
16	2000	1100	1100	1000	2000	1200
17	2000	900	400	1500	1000	800
18	900	800	1400	1400	700	1000
19	1000	1500	1100	1300	1200	1100
20	1000	200	60	700	600	500
21	2000	1000	1700	1400	1700	1600
22	1900	1000	1700	1300	1700	1600
23	2000	1800	1700	1400	1800	1600
24	2000	1000	2300	2000	1700	1600
25	1600	1000	1700	1000	1700	1600
26	1600	1000	1200	1000	1200	1600
27	2100	1000	1700	1500	1700	1600
28	1600	1000	1700	1000	1700	1600
29	1600	1000	1400	700	1700	1600
30	1600	1300	1700	1000	2000	1600

Приложение Г. Примеры отчетов по лабораторным работам по дисциплине «Исследование операций и методы оптимизации»

Примеры отчетов по лабораторной работе №1

Министерство образования и науки Российской Федерации

Федеральное государственное бюджетное образовательное учреждение
высшего профессионального образования

ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ СИСТЕМ УПРАВЛЕНИЯ И РАДИОЭЛЕКТРОНИКИ

Факультет систем управления (ФДО)

Кафедра автоматизированных систем (АСУ)

Минимизация функции одной переменной

Отчет по лабораторной работе № 1 по дисциплине
«Исследование операций и методы оптимизации»

Выполнил:
Студент гр.____
И.О. Фамилия
«   »__________2017 г.

Руководитель:
И.О. Фамилия руководителя
«   »__________2017 г.

2017

Вариант 1

Задание

1. В соответствии с вашим вариантом постройте график функции.
2. Напишите программу определения минимума функции с использованием методов прямого поиска (см. вариант).
3. Напишите программу определения минимума функции с использованием методов с применением производных.
4. Для каждого метода выполните две итерации вручную.
5. Сравните методы по числу итераций.

Для выполнения задания могут быть использованы пакеты (Excel, MathCad), языки программирования (C++, Pascal и др.).

Функция: $f(x) = (x-2)^2$, интервал [0;10], точность ε =0,1.

Методы: равномерный поиск, Пауэлла, средней точки.

Построим график функции с помощью Excel (рис.1.1).

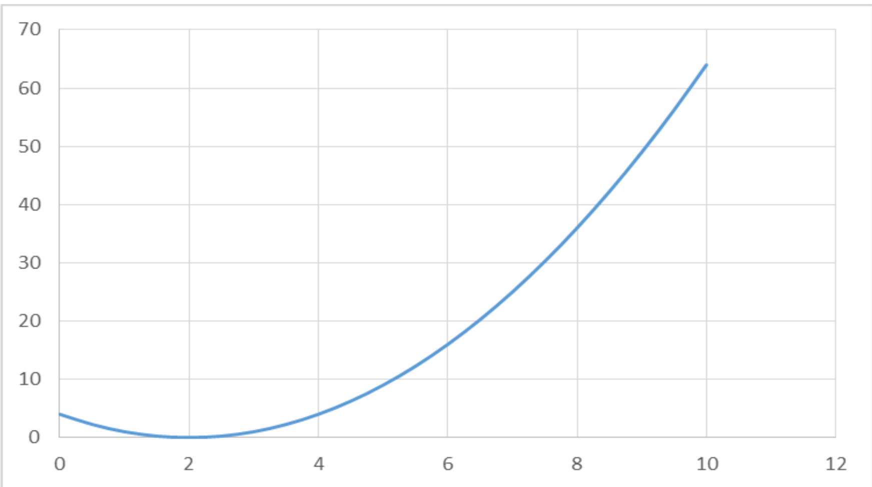


Рис.1.1 График функции $f(x) = (x-2)^2$ на интервале [0;10]

Для построения графика один столбец заполним значениями аргументов (от 0 до 10 с шагом 0,5):

[H1]=0

[H2]=H1+0,5

…

[H21]=H20+0,5

В следующий столбец запишем значения функции в данных точках:

[I1]=(H1-2)^2

…

[I21] =(H21-2)^2

Далее выделяем значения в двух столбцах и выбираем в главном меню «Вставка»-«Диаграммы»-«Точечная с прямыми отрезками».

Из рисунка 1.1 видно, что минимум функции находится в точке x=2.

Решим данную задачу с помощью макроса Excel.

Код метода *равномерного поиска* (для 20 точек) представлен на рис.1.2.

```
Function f(x) As Double
    f = (x - 2) ^ 2 'Определение заданной функции
End Function
Sub Кнопка1_Щелчок()
    'определим границы интервала
    a = 0
    b = 10
    N = 20 'число точек
    i = 0 'переменная для перебора точек
    'начальные значения точки минимума
    xmin = a
    fmin = f(xmin)
    'цикл перебора точек
    Do While (i <= N)
        x = a + i * (b - a) / (N + 1) 'расчет значения новой точки
        'если значение функции в новой точке меньше минимального,
        'то запомнить его в качестве минимального
        If f(x) < fmin Then
            xmin = x
            fmin = f(x)
        End If
        i = i + 1
    Loop
    'Вывод минимального значения функции и найденного арумента
    Cells(2, 1) = xmin
    Cells(2, 2) = fmin
End Sub
```

Рис. 1.2 Код метода равномерного поиска

Результат работы макроса представлен на рис.1.3.

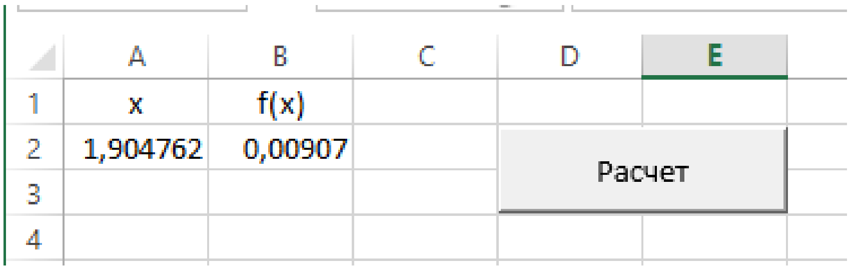


Рис.1.3 Результат программы

Выполним две итерации вручную.

1 итерация. i=0. Значение аргумента равно:

$$x = a + i \cdot \frac{b-a}{N+1} = 0 + 0 \cdot \frac{10-0}{21} = 0\,.$$

Значение функции в этой точке: $f(0) = (0-2)^2 = 4$.

2 итерация. i=i+1=1. Значение аргумента равно:

$$x = a + i \cdot \frac{b-a}{N+1} = 0 + 1 \cdot \frac{10-0}{21} = 0{,}48\,.$$

Значение функции в этой точке: $f(0{,}48) = (0{,}48-2)^2 = 2{,}32$.

Полученное значение функции меньше, чем значение на предыдущей итерации, поэтому $x_{\min} = 0{,}48$.

Код метода *Пауэлла* (для $\Delta x = 1{,}5$) представлен на рис. 1.4.

Результат работы макроса представлен на рис.1.5.

```
Sub Кнопка2_Щелчок()
    'определим расположение точек x1,x2,x3
    x1 = 0
    dx = 1.5
    x2 = x1 + dx
    If f(x2) < f(x1) Then
        x3 = x1 + 2 * dx
    Else: x3 = x1 - dx
    End If
    'если был выполнен шаг влево, то осуществляется нумерация в
    'естественном порядке
    If x3 < x1 Then
        xp = x3
        x3 = x2
        x2 = x1
        x1 = xp
    End If
    'цикл по итерациям
    Do
        'расчет значений функции в найденных точках
        f1 = f(x1)
        f2 = f(x2)
        f3 = f(x3)
        'нахождение минимального значения функции
        fmin = f1
        xmin = x1
        If f2 < fmin Then
            xmin = x2
            fmin = f2
        End If
        If f3 < fmin Then
            fmin = f3
            xmin = x3
        End If
        'вычисление квадратичной аппроксимации
        a1 = (f2 - f1) / (x2 - x1)
        a2 = 1 / (x3 - x2) * ((f3 - f1) / (x3 - x1) - (f2 - f1) / (x2 - x1))
        xs = (x2 + x1) / 2 - a1 / (2 * a2)
        'проверка условия завершения работы алгоритма
        If (Abs(fmin - f(xs)) < 0.1) And (Abs(xmin - xs) < 0.1) Then Exit Do
        'при заданной величине шага минимальное значение будет в точке x2 либо xs.
        'В зависимости от из расположения возьмем эти две точки и одну из границ
        If x2 < xs Then
            x1 = x2
            x2 = xs
        ElseIf x2 > xs Then
            x1 = xs
        End If
    Loop
    'выберем из xmin и xs точку, в которой функция минимальна и отобразим полученное
    'решение
    If fmin < f(xs) Then
        Cells(2, 1) = xmin
        Cells(2, 2) = fmin
    Else
        Cells(2, 1) = xs
        Cells(2, 2) = f(xs)
    End If
End Sub
```

Рис.1.4 Код метода Пауэлла

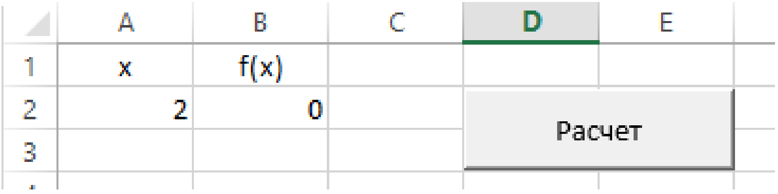


Рис.1.5 Результат метода Пауэлла

Выполним две итерации вручную.

1 итерация. Выполним расчет следующей точки:

$x_1 = 0$, $x_2 = 0 + 1{,}5 = 1{,}5$.

Значение функции:

$f(x_1) = (x_1 - 2)^2 = 4$,

$f(x_2) = (x_2 - 2)^2 = (1{,}5 - 2)^2 = 0{,}25$.

Так как $f(x_2) < f(x_1)$, то: $x_3 = 0 + 2 \cdot 1{,}5 = 3$.

Значение функции $f(x_3) = (x_3 - 2)^2 = (3 - 2)^2 = 1$.

Минимальное значение функции: $f_{\min} = 0{,}25$, $x_{\min} = 1{,}5$.

Вычислим квадратичную аппроксимацию:

$$a_1 = \frac{f(x_2) - f(x_1)}{x_2 - x_1} = \frac{0{,}25 - 4}{1{,}5 - 0} = -2{,}5.$$

$$a_2 = \frac{1}{x_3 - x_2}\left[\frac{f(x_3) - f(x_1)}{x_3 - x_1} - \frac{f(x_2) - f(x_1)}{x_2 - x_1}\right] = \frac{1}{3 - 1{,}5}\left[\left(\frac{1 - 4}{3 - 0}\right) - \left(\frac{0{,}25 - 4}{1{,}5 - 0}\right)\right] = 1$$

$$\bar{x} = \frac{x_2 + x_1}{2} - \frac{a_1}{2a_2} = \frac{1{,}5 + 0}{2} - \frac{-2{,}5}{2 \cdot 1} = 2.$$

Значение функции: $f(\bar{x}) = (2 - 2)^2 = 0$.

Выбираем точку $\bar{x}$, в которой функция минимальна и две точки по обе стороны: $x_1 = 1{,}5$; $x_2 = 2$; $x_3 = 3$. Переходим к следующей итерации.

2 итерация

Минимальное значение функции: $f_{\min} = 0$, $x_{\min} = 2$.

Вычислим квадратичную аппроксимацию:

$$a_1 = \frac{f(x_2) - f(x_1)}{x_2 - x_1} = \frac{0 - 1,5}{2 - 1,5} = -0,5.$$

$$a_2 = \frac{1}{x_3 - x_2}\left[\frac{f(x_3) - f(x_1)}{x_3 - x_1} - \frac{f(x_2) - f(x_1)}{x_2 - x_1}\right] = \frac{1}{3-2}\left[\left(\frac{1 - 0,25}{3 - 1,5}\right) - \left(\frac{0 - 0,25}{2 - 1,5}\right)\right] = 1$$

$$\bar{x} = \frac{x_2 + x_1}{2} - \frac{a_1}{2a_2} = \frac{2 + 1,5}{2} - \frac{-0,5}{2 \cdot 1} = 2.$$

Значение функции: $f(\bar{x}) = (2-2)^2 = 0$

$f_{\min} - f(\bar{x}) = 0$

$x_{\min} - \bar{x} = 2 - 2 = 0$

Условие $\left|f_{\min} - f(\bar{x})\right| \le 0,1$ и $\left|x_{\min} - \bar{x}\right| \le 0,1$ выполняется, следовательно, работа алгоритма завершается.

Выполним теперь решение задачи методом *средней точки*.

Код метода средней точки представлен на рис.1.6.

Результат работы программы представлен на рис. 1.7.

Выполним вручную две итерации.

1 итерация. Вычислим среднюю точку ($a = 0, b = 10$):

$$c = \frac{a + b}{2} = \frac{0 + 10}{2} = 5.$$

Значение производной $f'(x) = 2(x-2)$ в этой точке:

$f'(3) = 2(3-2) = 2$.

Значение положительно, поэтому смещаем правую границу: $b = c = 5$.

```
Function f(x) As Double
    f = (x - 2) ^ 2 'исходная функция
End Function
Function fp(x) As Double
    fp = 2 * (x - 2) 'производная функции
End Function
Sub Кнопка3_Щелчок()
    'определяем начальные границы
    a = 0
    b = 10
    'определяем точность
    e = 0.1
    'цикл по итерациям
    Do
        c = (a + b) / 2 'рассчитываем среднюю точку
        'если производная отрицательна, то смещаем левую границу
        If fp(c) < 0 Then
            a = c
        'если производная положительна, то смещаем правую границу
        Else
            b = c
        End If
        'если модуль производной в точке меньше заданной точности,
        'то завершаем вычисления
        If Abs(fp(c)) < e Then Exit Do
    Loop
    'рассчитываем среднюю точку и значение функции в ней
    xmin = (a + b) / 2
    fmin = f(xmin)
    'выводим полученные значения в ячейки
    Cells(2, 1) = xmin
    Cells(2, 2) = fmin
End Sub
```

Рис.1.6. Код метода средней точки

	A	B	C	D	E
1	x	f(x)			
2	2,1875	0,035156		Расчет	
3					
4					

Рис. 1.7 Результат оптимизации методом средней точки

2 итерация. Вычислим среднюю точку:

$$c = \frac{a+b}{2} = \frac{0+5}{2} = 2{,}5.$$

Значение производной в этой точке:

$$f'(2{,}5) = 2(2{,}5-2) = 1.$$

Значение положительно, поэтому смещаем правую границу: $b = c = 2{,}5$.

Условие останова ($|f'(2{,}5)| < 0{,}1$) не выполняется, поэтому будут выполнены ещё итерации.

5) Выполним сравнение алгоритмов по числу реализаций. Для этого добавим в код переменную (k), значение которой увеличивается на единицу в каждой итерации цикла (рис.1.8).

```
Function f(x) As Double
    f = (x - 2) ^ 2 'исходная функция
End Function
Function fp(x) As Double
    fp = 2 * (x - 2) 'производная функции
End Function
Sub Кнопка3_Щелчок()
    'определяем начальные границы
    a = 0
    b = 10
    'определяем точность
    e = 0.1
    'цикл по итерациям
    k = 0 'число итераций
    Do
        c = (a + b) / 2 'рассчитываем среднюю точку
        'если производная отрицательна, то смещаем левую границу
        If fp(c) < 0 Then
            a = c
        'если производная положительна, то смещаем правую границу
        Else
            b = c
        End If
        'если модуль производной в точке меньше заданной точности,
        'то завершаем вычисления
        k = k + 1
        If Abs(fp(c)) < e Then Exit Do
    Loop
    'рассчитываем среднюю точку и значение функции в ней
    xmin = (a + b) / 2
    fmin = f(xmin)
    'выводим полученные значения в ячейки
    Cells(2, 1) = xmin
    Cells(2, 2) = fmin
    Cells(2, 3) = k
End Sub
```

Рис.1.8 Код программы с подсчетом числа итераций

В таблице 1.1 предствлено определенное число итераций для метода Пауэлла и метода средней точки. Метод Пауэлла позволил найти решение за меньшее число итераций, однако реализация этого метода более сложная.

Таблица 1.1 Сравнение методов по числу итераций

	Метод Пауэлла	Метод средней точки
Число итераций	2	5

Заключение

Выполнена оптимизация функции с помощью трех методов. Наиболее точное решение было получено с помощью метода Пауэлла, наиболее простым методом является метод равномерного поиска.

Вариант 2

Задание

1. В соответствии с вашим вариантом постройте график функции.
2. Напишите программу определения минимума функции с использованием методов прямого поиска (см. вариант).
3. Напишите программу определения минимума функции с использованием методов с применением производных.
4. Для каждого метода выполните две итерации вручную.

   Для выполнения задания могут быть использованы пакеты (Excel, MathCad), языки программирования (C++, Pascal и др.).
5. Сравните методы по числу итераций.

Функция: $f(x)=(x-2)^2$, интервал [0;10], точность ε =0,1.

Методы: равномерный поиск, дихотомии, Ньютона.

Построим график функции с помощью MathCad (рис.1.9).

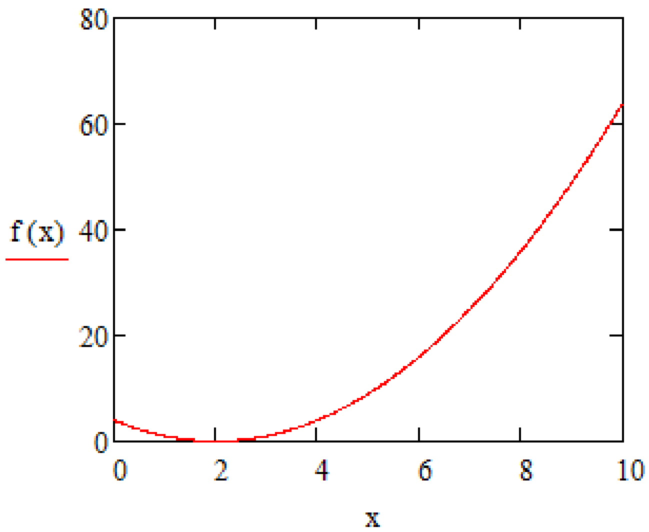


Рис.1.9 График функции $f(x)=(x-2)^2$

Для построения графика нужно задать функцию f(x), затем выбрать в главном меню Insert-Graph-X-Y Plot. На месте маркера, расположенного в центре внизу, напишем «x», на месте маркера, расположенного в центре, слева «f(x)» (рис.1.10).

Теперь нужно изменить значения по оси x. Для этого нажимаем на график и в маркере, расположенном внизу слева напишем 0.

$$f(x) := (x-2)^2$$

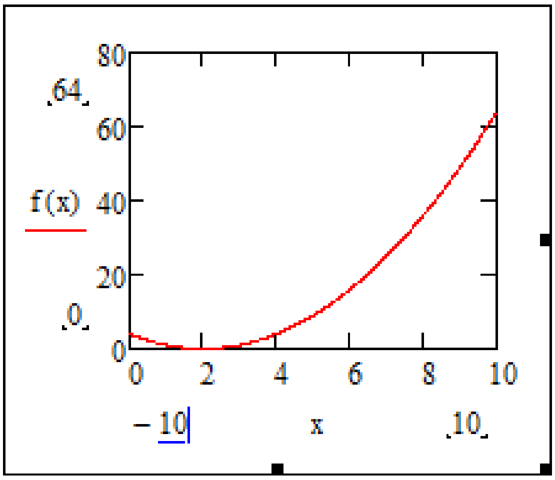


Рис. 1.10 Построение графика

Из рисунка видно, что минимум функции находится в точке *x*=2.

Решим данную задачу с помощью MathCad методом равномерного поиска (для 6 точек) (рис.1.11). Полученное решение: *x*=1,429.

```
ravnomer(a,b) := | a ← 0
                 | b ← 10
                 | N ← 6
                 | i ← 0
                 | xmin ← a
                 | while i ≤ N
                 |   | x ← a + i·((b − a)/(N + 1))
                 |   | xmin ← x  if  f(x) < f(xmin)
                 |   | i ← i + 1
                 | res_0 ← xmin
                 | res_1 ← f(xmin)
                 | res_2 ← i
                 | res
```

$$\mathrm{ravnomer}(a,b)=\begin{pmatrix}1.42857\\0.32653\\7\end{pmatrix}$$

-значение аргумента
- значение целевой функции
- количество итераций

Рис.1.11 Реализация алгоритма равномерного поиска в MathCad

Выполним две итерации вручную.

1 итерация. i=0. Значение аргумента равно:

$$x=a+i\cdot\frac{b-a}{N+1}=0+0\cdot\frac{10-0}{7}=0.$$

Значение функции в этой точке:

$$f(0)=(0-2)^2=4.$$

2 итерация. i=i+1=1. Значение аргумента равно:

$$x=a+i\cdot\frac{b-a}{N+1}=0+1\cdot\frac{10-0}{7}=1{,}43.$$

Значение функции в этой точке:

$$f(1{,}43)=(1{,}43-2)^2=0{,}33.$$

Полученное значение функции меньше, чем значение на предыдущей итерации, поэтому $x_{\min} = 1{,}43$.

На рис.1.12 представлено решение задачи методом дихотомии. Полученное решение: *x*=1,92.

```
half(a, b, eps) := | e ← 0.1
                   | k ← 0
                   | while  b − a > eps
                   |     | x ← ((a + b) − e) / 2
                   |     | y ← ((a + b) + e) / 2
                   |     | a ← y  if  f(x) > f(y)
                   |     | b ← x  otherwise
                   | res_0 ← (a + b) / 2
                   | res_1 ← f(res_0)
                   | res
```

$$\mathrm{half}(0, 10, 0.1) = \begin{pmatrix} 1.92266 \\ 5.98206 \times 10^{-3} \end{pmatrix} \begin{matrix} \text{-значение аргумента} \\ \text{- значение функции} \end{matrix}$$

Рис.1.12 Решение задачи с помощью метода дихотомии

Выполним вручную две итерации.

1 итерация. Вычислим точку *x* ($a = 0, b = 10$):

$$x = \frac{a + b - e}{2} = \frac{0 + 10 - 0.1}{2} = 4{,}95.$$

Значение функции в точке *x*:

$$f(4{,}95) = (4{,}95 - 2)^2 = 8{,}7025.$$

Вычислим точку *y*:

$$y = \frac{a+b+e}{2} = \frac{0+10+0.1}{2} = 5,05\,.$$

Значение функции в точке *y*:

$$f(5,05) = (5,05-2)^2 = 9,3025\,.$$

Т.к. $f(x) < f(y)$, то смещаем правую границу: $b = y = 5,05$.

2 итерация. Вычислим точку *x*:

$$x = \frac{a+b-e}{2} = \frac{0+5,05-0.1}{2} = 2,475\,.$$

Значение функции в точке *x*:

$$f(2,475) = (2,475-2)^2 = 0,2256\,.$$

Вычислим точку *y*:

$$y = \frac{a+b+e}{2} = \frac{0+5,05+0.1}{2} = 2,575\,.$$

Значение функции в этой точке:

$$f(2,575) = (2,575-2)^2 = 0,3306\,.$$

Т.к. $f(x) < f(y)$, то смещаем правую границу: $b = y = 2,575$.

Условие останова ($|b-a| = 2,575-0 = 2,575 < 0,1$) не выполняется, поэтому будут выполнены ещё итерации.

На рис.1.13 представлено решение задачи с помощью метода Ньютона.

Полученное решение: *x*=2.

$$n(xn) := \left| \begin{array}{l} \text{while } (1) \\ \quad \left| \begin{array}{l} x \leftarrow xn - \dfrac{\left[\dfrac{d}{dxn}(xn-2)^2\right]}{\dfrac{d^2}{dxn^2}(xn-2)^2} \\ \text{break if } \left|\dfrac{d}{dxn}(xn-2)^2\right| \le 1 \\ xn \leftarrow x \end{array} \right. \\ m \leftarrow x \end{array} \right.$$

$$n(0) = 2 \qquad f(n(0)) = 0$$

Рис.1.13 Решение задачи методом Ньютона

Выполним вручную две итерации.

1 итерация (xn=0).

Значение производной $f'(x) = 2(x-2)$ в начальной точке:

$f'(0) = 2(0-2) = -4$.

Значение второй производной:

$f''(x) = \left(2(x-2)\right)' = 2$.

Новое значение точки x равно:

$x = 0 - \dfrac{-4}{2} = 2$.

Значение производной в точке:

$f'(2) = 2(2-2) = 0$.

Выполняется условие останова ($\left|f'(2)\right| < 0{,}1$), значит, решение найдено.

5) Выполним сравнение алгоритмов по числу реализаций. Для этого добавим в код переменную (k), значение которой увеличивается на единицу в каждой итерации цикла (рис.1.14).

$$\mathrm{half}(a,b,eps) := \left|\begin{array}{l} e \leftarrow 0.1 \\ k \leftarrow 0 \\ \text{while } b - a > eps \\ \quad \left|\begin{array}{l} x \leftarrow \dfrac{(a+b)-e}{2} \\ y \leftarrow \dfrac{(a+b)+e}{2} \\ a \leftarrow y \ \text{ if } f(x) > f(y) \\ b \leftarrow x \ \text{ otherwise} \\ k \leftarrow k+1 \end{array}\right. \\ res_0 \leftarrow \dfrac{(a+b)}{2} \\ res_1 \leftarrow f(res_0) \\ res_2 \leftarrow k \\ res \end{array}\right.$$

$$\mathrm{half}(0,10,0.1) = \begin{pmatrix} 1.92266 \\ 5.98206 \times 10^{-3} \\ 6 \end{pmatrix} \begin{array}{l} \text{-значение аргумента} \\ \text{- значение функции} \\ \text{–число итераций} \end{array}$$

Рис.1.14 Определение числа итераций

В таблице 1.2 представлены значения числа итераций для метода дихотомии и метода Ньютона. С помощью метода Ньютона решение было получено всего за одну итерацию, однако данный метод требует вычисления первой и второй производной исходной функции (т.е. функция должна быть дважды дифференцируемой).

Таблица 1.2 Сравнение методов по числу итераций

	Метод дихотомии	Метод Ньютона
Число итераций	6	1

Вывод

Выполнена оптимизация функции с помощью трех методов. Наиболее точное решение было получено с помощью метода Ньютона, наиболее простым методом является метод равномерного поиска. Также с помощью метода Ньютона решение было найдено всего за одну итерацию, однако необходимо было вычислить первую и вторую производную.

Вариант 3

Задание

1. В соответствии с вашим вариантом постройте график функции.
2. Напишите программу определения минимума функции с использованием методов прямого поиска (см. вариант).
3. Напишите программу определения минимума функции с использованием методов с применением производных.
4. Для каждого метода выполните две итерации вручную.

   Для выполнения задания могут быть использованы пакеты (Excel, MathCad), языки программирования (C++, Pascal и др.).
5. Сравните методы по числу итераций.

Функция: $f(x)=(x-2)^2$, интервал [0;10], точность ε =0,1.

Методы: Монте-Карло, золотого сечения, средней точки. Решение задач выполним с помощью языка программирования Java.

Построим график функции с помощью библиотеки JFreeChart (рис.1.15).

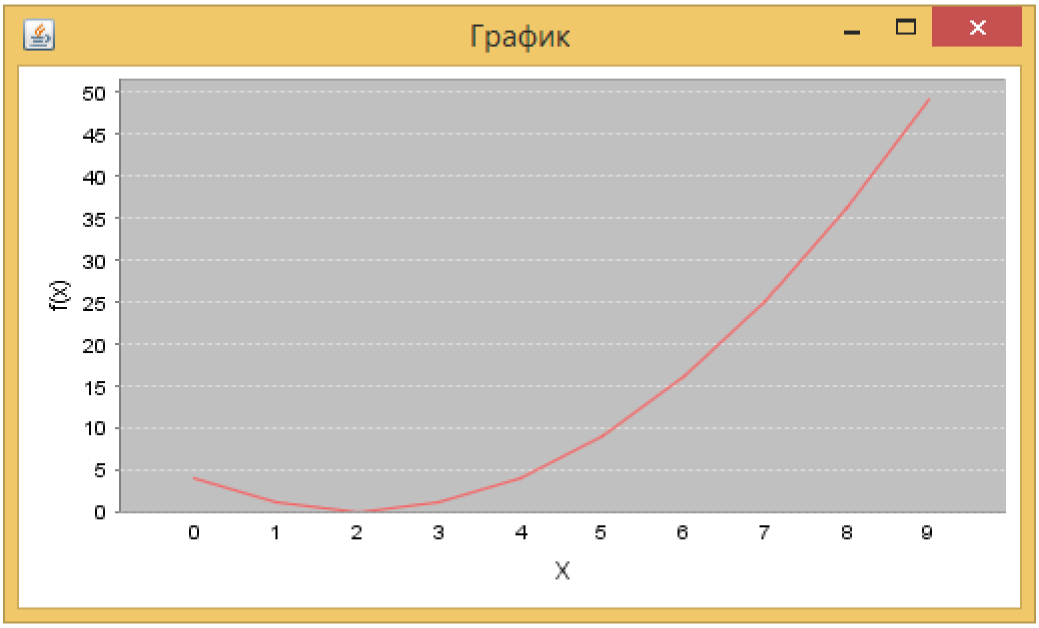


Рис.1.15 График функции $f(x)=(x-2)^2$ на интервале [0;10]

Код графика представлен на рис.1.16.

```java
public class JF2 extends JFrame {
    public JF2(final String title) {
        super(title);
        final CategoryDataset dataset = createDataset();
        final JFreeChart chart = createChart(dataset);
        final ChartPanel chartPanel = new ChartPanel(chart);
        chartPanel.setPreferredSize(new Dimension(500, 270));
        setContentPane(chartPanel);
    }
   // Создание набора данных
    private CategoryDataset createDataset() {
        final DefaultCategoryDataset dataset = new DefaultCategoryDataset();
        final String series1 = "First";
        //подписи по оси x
        final String type0 = "0";
        final String type1 = "1";
        final String type2 = "2";
        final String type3 = "3";
        final String type4 = "4";
        final String type5 = "5";
        final String type6 = "6";
        final String type7 = "7";
        final String type8 = "8";
        final String type9 = "9";
        // значения функции f(x)=(x-2)^2 в точках x
        dataset.addValue(4.0, series1, type0);
        dataset.addValue(1.0, series1, type1);
        dataset.addValue(0.0, series1, type2);
        dataset.addValue(1.0, series1, type3);
        dataset.addValue(4.0, series1, type4);
        dataset.addValue(9.0, series1, type5);
        dataset.addValue(16.0, series1, type6);
        dataset.addValue(25.0, series1, type7);
        dataset.addValue(36.0, series1, type8);
        dataset.addValue(49.0, series1, type9);
        return dataset;
    }
```

```
  /
private JFreeChart createChart(final CategoryDataset dataset) {

    // создание графика, подписей осей
    final JFreeChart chart = ChartFactory.createLineChart(
        "",         // заголовок графика
        "X",                        // подпись оси x
        "f(x)",                     // подпись оси y
        dataset,                    // данные
        PlotOrientation.VERTICAL,   // расположение осей графика
        false,                       // показывать легенду
        true,                       // подсказки значений
        true                        // urls
    );

    //установка цвета фона
    chart.setBackgroundPaint(Color.white);
    final CategoryPlot plot = (CategoryPlot) chart.getPlot();
    plot.setBackgroundPaint(Color.lightGray);
    plot.setRangeGridlinePaint(Color.white);
    return chart;
}

/**
 * Точка входа для построения графика
 *
  */
public static void main(final String[] args) {
    final JF mychart = new JF("График");
    mychart.pack();
    mychart.setVisible(true);

}
```

Рис.1.16 Построение графика

Из рисунка 1.15 видно, что минимум функции находится в точке x=2.

Решим задачу поиска минимума функции с помощью метода Монте-Карло (рис.1.17).

```
package linearpr;

public class rrun {

    public static void main(String[] args) {
        // TODO Auto-generated method stub
        //границы интервала
        double a=0;
        double b=10;
        //начальное значение решения
        double fmin=f(0);
        double xmin=0;
        //сгенерируем 100 точек
        for (int i=1; i<= 100; i++) {
            //генерирование случайной точки на интервале [a;b]
            double z=b*Math.random();
            //если значение функции в сгенерированной точке
            //меньше текущего решения, то запомнить эту точку
            if (f(z)<fmin) {
                fmin=f(z);
                xmin=z;
            }
    //вывод на экран решения
        }
        System.out.println("f="+fmin);
        System.out.println("x="+xmin);
    }
    //расчет заданной функции
    static double f(double x){
        double f = Math.pow(x-2, 2);
        return f;
    }

}
```

Рис.1.17 Код программы метода Монте-Карло

Полученное решение представлено на рис.1.18.

```
Console
<terminated> rrun [Java Application] C:\P
f=0.00949369842393122
x=1.902564386170183 5
```

Рис.1.18 Результат программы

Выполним две итерации вручную.

1 итерация. Возьмем на интервале [0;10] случайное число. Пусть оно равно 3. Рассчитаем значение функции в этой точке:

$$f(3) = (3-2)^2 = 1.$$

Данное значение запомним в качестве текущего решения ($x_{\min} = 3,\ f_{\min} = 1$).

2 итерация. Возьмем на интервале [0;10] случайное число. Пусть оно равно 5. Рассчитаем значение функции в этой точке:

$$f(3) = (5-2)^2 = 9 .$$

Данное значение больше, чем текущее решение, поэтому текущее решение остается неизменным ($x_{\min} = 3,\ f_{\min} = 1$).

Решим задачу методом золотого сечения. Код программы представлен на рис.1.19.

Результат работы программы представлен на рис.1.20.

Выполним вручную две итерации.

1 итерация. Вычислим точку y ($a = 0, b = 10$):

$$y = a + \frac{3-\sqrt{5}}{2}\left(b-a\right) = 0 + 0{,}382 \cdot (10-0) = 3{,}82 .$$

Значение функции в этой точке:

$$f(3{,}82) = (3{,}82-2)^2 = 3{,}31.$$

Вычислим точку z:

$$z = a + b - y = 0 + 10 - 3{,}82 = 6{,}18 .$$

Значение функции в этой точке:

$$f(6{,}18) = (6{,}18-2)^2 = 17{,}47 .$$

Т.к. $f(y) < f(z)$, то смещаем правую границу: $b = z = 6{,}18$; $z = y = 3{,}82$; $y = a + b - y = 0 + 6{,}18 - 3{,}82 = 2{,}36$.

```java
package linearpr;
public class golden {
    public static void main(String[] args) {
        // TODO Auto-generated method stub
        double a=0; double b=10; //границы интервала
        double fmin; double xmin;
        //начальные значения точек золотого сечения
        double y=a+(3-Math.sqrt(5.0))/2*(b-a);
        double z=a+b-y;
        //расчет величины интервала
        double L=b-a;
        //пока величина интервала больше точности выполнять цикл
        while(L>0.1) {
            //сравнение значения функции в точках золотого сечения
            if (f(y)<=f(z)) {
                b=z;//изменение правой границы
                //новые точки золотого сечения
                z=y; y=a+b-y;
            }else{
                a=y; //изменение левой границы
                //новые точки золотого сечения
                y=z; z=a+b-z;
            }
            L=b-a;//расчет новой величины интервала
        }
        //средняя точка интервала берется в качестве решения
        xmin=(a+b)/2;
        fmin=f(xmin);
        //вывод на экран решения
        System.out.println("f="+fmin); System.out.println("x="+xmin);
    }
    //расчет заданной функции
    static double f(double x){
        double f = Math.pow(x-2, 2);
        return f;
    }
}
```

Рис.1.19. Код программы (метод золотого сечения)

```
Console
<terminated> golden [Java Application]
f=5.949626521666265E-4
x=1.975608143732659
```

Рис.1.20 Результат работы программы

2 итерация. Вычислим значения функции в точках золотого сечения:

$f(2,36) = (2,36 - 2)^2 = 0,1296$.

$f(3,82) = (3,82 - 2)^2 = 3,31$.

Т.к. $f(y) < f(z)$, то смещаем правую границу:

$b = z = 3,82$;

$z = y = 2,36$;

$y = a + b - y = 0 + 3,82 - 2,36 = 1,46$.

Условие останова ($|b - a| = 3,82 - 0 = 3,82 < 0,1$) не выполняется, поэтому будут выполнены ещё итерации.

Решим данную задачу методом средней точки. Код программы представлен на рис.1.21.

Результат работы программы представлен на рис.1.22.

```
package linearpr;

public class middle {

    public static void main(String[] args) {
        // TODO Auto-generated method stub
        double a=0; double b=10; //границы интервала
        double c=(a+b)/2; //расчет средней точки
        //пока модуль производной больше точности выполнять цикл
        while(Math.abs(fp(c))>0.1) {
            //определение знака производной
            if (fp(c)<0) {
                a=c;//изменение левой границы
            }else{
                b=c; //изменение правой границы
            }
            c=(a+b)/2;//расчет новой средней точки
        }
        //вывод на экран решения
        System.out.println("f="+f(c)); System.out.println("x="+c);
    }
    //расчет заданной функции
    static double f(double x){
        double f = Math.pow(x-2, 2);
        return f;
    }
    //расчет производной
    static double fp(double x){
        double fp = 2*(x-2);
        return fp;
    }
}
```

Рис.1.21 Код программы (метод средней точки)

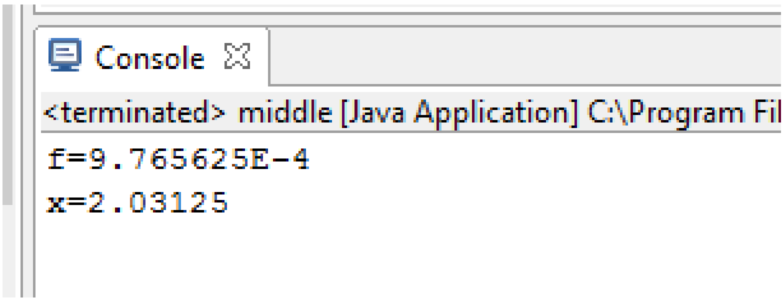


Рис.1.22 Результат работы программы

Выполнение итераций вручную методом средней точки представлено в варианте 1.

5) Выполним сравнение алгоритмов по числу реализаций. Для этого добавим в код переменную (k), значение которой увеличивается на единицу в каждой итерации цикла.

Результат представлен в таблице 1.3. С помощью метода средней точки решение было найдено за меньшее число итераций, однако данный метод требует вычисление значения производной.

Таблица 1.3 Сравнение методов по числу итераций

	Метод золотого сечения	Метод средней точки
Число итераций	10	5

Заключение

Выполнена оптимизация функции с помощью трех методов: двух методов прямого поиска и одного метода с использованием производной. Наиболее точное решение было получено с помощью метода средней точки, наиболее простым методом является метод Монте-Карло. Также метод средней точки позволяет найти решение за меньшее число итераций, однако требуется вычисление производной.

Пример отчета по лабораторной работе №2

Министерство образования и науки Российской Федерации

Федеральное государственное бюджетное образовательное учреждение
высшего профессионального образования

ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ СИСТЕМ УПРАВЛЕНИЯ И РАДИОЭЛЕКТРОНИКИ

Факультет систем управления (ФДО)

Кафедра автоматизированных систем (АСУ)

Минимизация функции нескольких переменных

Отчет по лабораторной работе № 2 по дисциплине
«Исследование операций и методы оптимизации»

Выполнил:
Студент гр.____
И.О. Фамилия
«   »__________2017 г.

Руководитель:
И.О. Фамилия руководителя
«   »__________2017 г.

2017

Вариант 1

Задание

1. В соответствии с вашим вариантом постройте график функции.
2. Напишите программу определения минимума функции с использованием метода прямого поиска (см. вариант).
3. Напишите программу определения минимума функции с использованием градиентного метода.
4. Для каждого метода выполните две итерации вручную.

Для выполнения задания могут быть использованы пакеты (Excel, MathCad), языки программирования (C++, Pascal и др.). Пример выполнения задания приведен в Приложении.

Функция: $f(x_1, x_2) = (x_1 - 5)^2 + (x_2 - 6)^2$, $\varepsilon = 0{,}1$, $\alpha = 2$.

Методы: Хука-Дживса, градиентный спуск.

График функции представлен на рис.1.23.

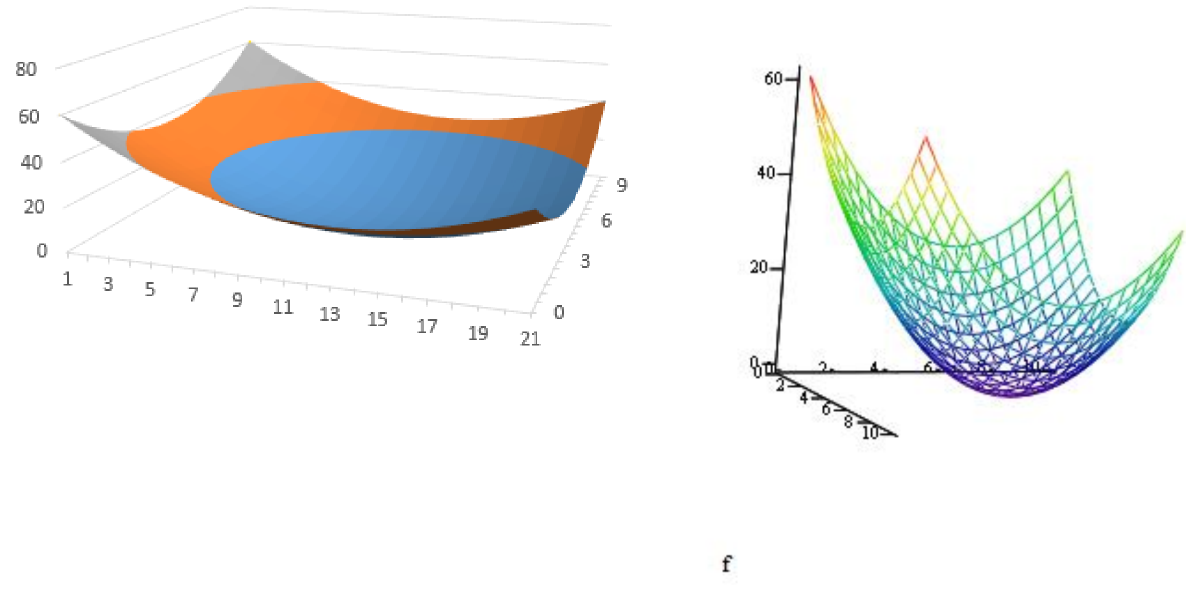
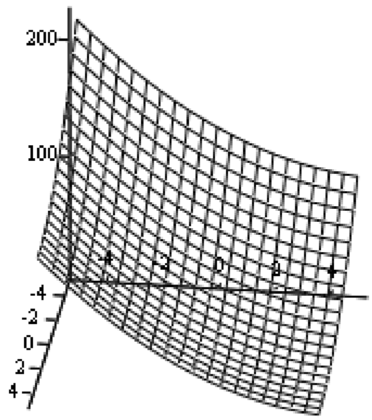


Рис.1.23 График функции $f(x_1, x_2) = (x_1 - 5)^2 + (x_2 - 6)^2$: a)построенный в Excel; b) построенный в MathCad

Для построения графика в Mathcad нужно задать функцию, а затем выбратьв главном меню «Insert»-«Graph»-«Surface Plot» и в левом нижнем маркере написать название функции (рис.1.24).

$$f(x1,x2) := (x1-5)^2 + (x2-6)^2$$



f

Рис.1.24 Построение графика

Чтобы закрасить график нужно нажать на нем правой кнопкой и выбрать «Properties». Далее нужно перейти на вкладку «Appearance» и поставить переключаель ColorOptions на «Colormap» (рис.1.25). Если нужно, чтобы график был закрашен, то переключатель Fill Options переводится в положение Fill Surface.

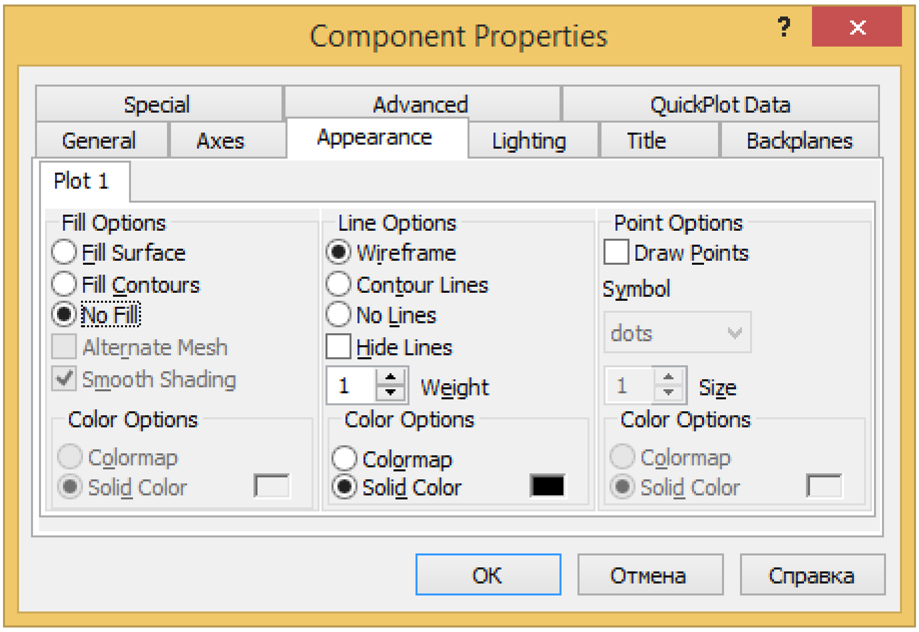


Рис. 1.25 Параметры для закрашивания графика

Для того, чтобы увеличить диапазон аргументов, нужно перейти на вкладку QuickPlot Data и установить начальные и конечные значения для двух аргументов (рис.1.26).

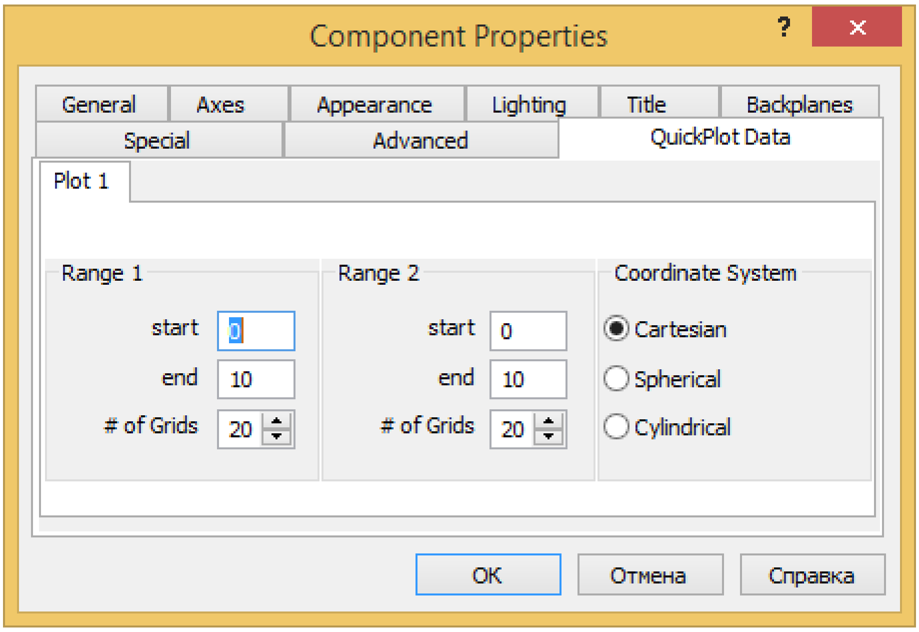


Рис.1.26 Определений значений аргументов

Для того, чтобы построить контурный график, нужно на вкладке General переключатель Plot1 поставить в позицию Contour Plot (рис.1.27). Полученный график представлен на рис.1.28. Он представляет собой вид графика сверху, значения целевой функции представлены разными цветами, значения, расположенные на одном срезе, равны. Можно сделать вывод, что минимум функции расположен в области, где аргументы принимают значения из интервалов [2;8] и [3;9].

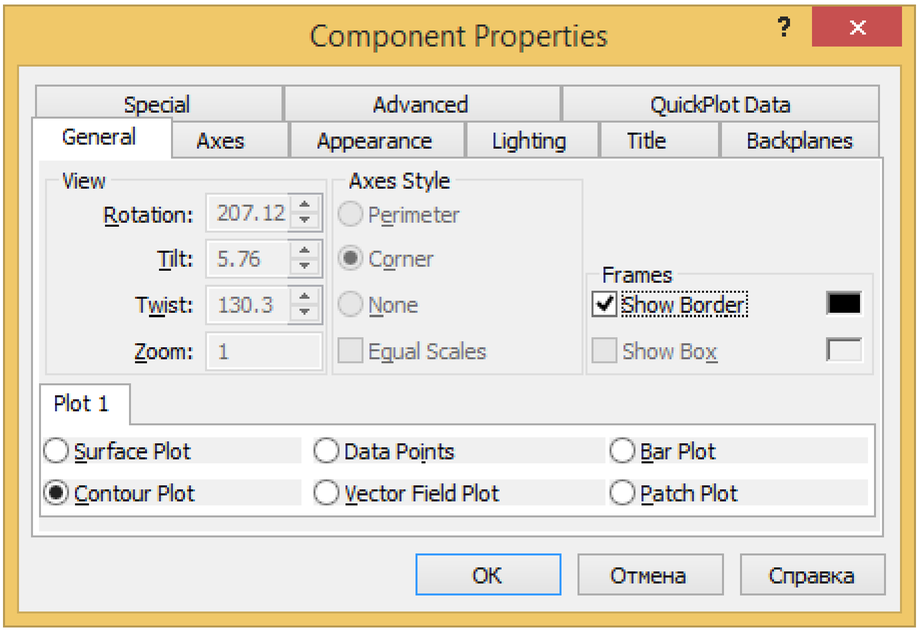


Рис.1.27 Построение контурного графика

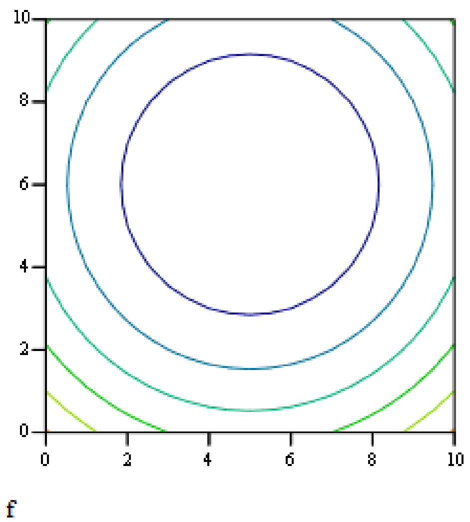
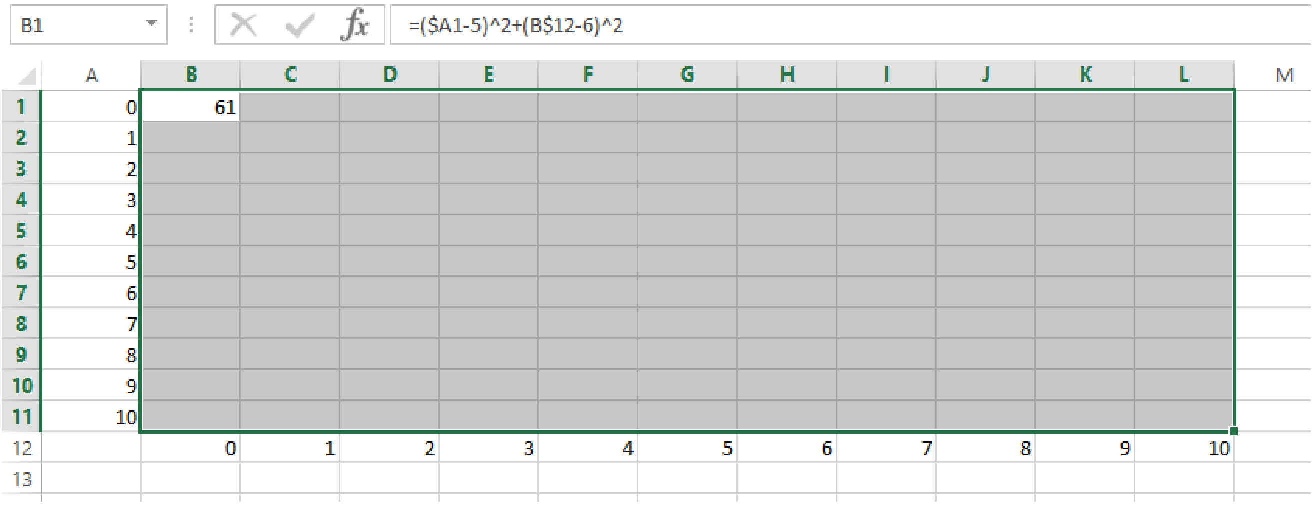


Рис.1.28 Контурный график

Для построения поверхности в Excel нужно в ячейках A1:A11 привести значения первого аргумента, в ячейках B12:L12 – второго аргумента. Затем выделить область внутри и записать формулу расчета функции для ячейки B1 (расположенной в верхнем левом углу) (рис.1.29):

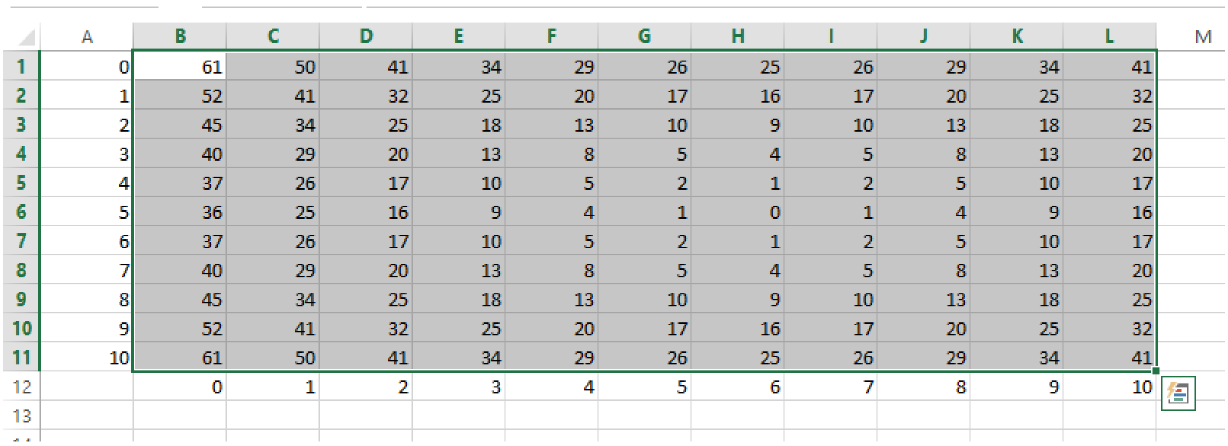
[B1]=($A1-5)^2+(B$12-6)^2

Далее нажимаем Ctrl+Enter. Полученные значения приведены на рис.1.30.



	A	B	C	D	E	F	G	H	I	J	K	L	M
1	0	61											
2	1												
3	2												
4	3												
5	4												
6	5												
7	6												
8	7												
9	8												
10	9												
11	10												
12		0	1	2	3	4	5	6	7	8	9	10	
13													

Рис.1.29 Заполнение таблицы



	A	B	C	D	E	F	G	H	I	J	K	L	M
1	0	61	50	41	34	29	26	25	26	29	34	41	
2	1	52	41	32	25	20	17	16	17	20	25	32	
3	2	45	34	25	18	13	10	9	10	13	18	25	
4	3	40	29	20	13	8	5	4	5	8	13	20	
5	4	37	26	17	10	5	2	1	2	5	10	17	
6	5	36	25	16	9	4	1	0	1	4	9	16	
7	6	37	26	17	10	5	2	1	2	5	10	17	
8	7	40	29	20	13	8	5	4	5	8	13	20	
9	8	45	34	25	18	13	10	9	10	13	18	25	
10	9	52	41	32	25	20	17	16	17	20	25	32	
11	10	61	50	41	34	29	26	25	26	29	34	41	
12		0	1	2	3	4	5	6	7	8	9	10	
13													

Рис.1.30 Рассчитанные значения функции

Далее вставим после столбца A новый столбец и заполним его значениями подписей (рис.1.31):

[B1]=""&A1

…

[B11]=""&A11

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	0	0	61	50	41	34	29	26	25	26	29	34	41	
2	1	1	52	41	32	25	20	17	16	17	20	25	32	
3	2	2	45	34	25	18	13	10	9	10	13	18	25	
4	3	3	40	29	20	13	8	5	4	5	8	13	20	
5	4	4	37	26	17	10	5	2	1	2	5	10	17	
6	5	5	36	25	16	9	4	1	0	1	4	9	16	
7	6	6	37	26	17	10	5	2	1	2	5	10	17	
8	7	7	40	29	20	13	8	5	4	5	8	13	20	
9	8	8	45	34	25	18	13	10	9	10	13	18	25	
10	9	9	52	41	32	25	20	17	16	17	20	25	32	
11	10	10	61	50	41	34	29	26	25	26	29	34	41	
12			0	1	2	3	4	5	6	7	8	9	10	
13														

Рис.1.31 Добавление столбца

Наконец, выделяем ячейки B1:M11 и выбираем в главном меню «Вставка»–«Диаграммы»–«Поверхность». В итоге получим график, представленный на рис.1.23.

Если вместо поверхности выбрать «Контурная», то получим диаграмму, которая показывает вид графика сверху (рис.1.32) и позволяет также сделать вывод о расположении точки минимума. Значения целевой функции здесь представлены соответствующими цветами.

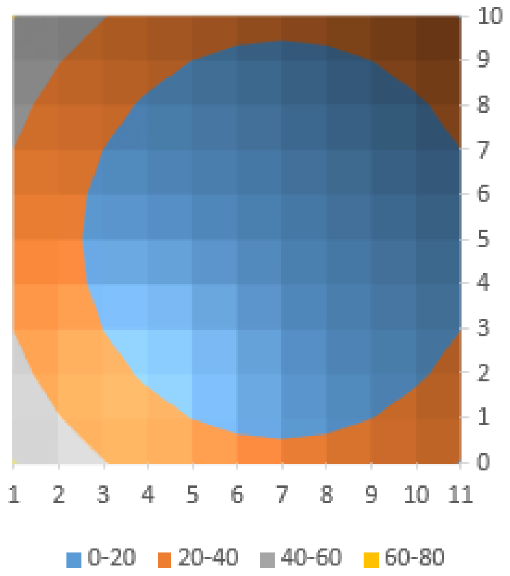


Рис.1.32 Контурный график

Решим задачу нахождения минимума функции методом ук-Дживса. Код метода Хука-Дживса представлен на рис.1.33.

```
Function f(x, y)
    f = (x - 5) ^ 2 + (y - 6) ^ 2 'заданная функция
End Function
Sub ХукаДживс_Кнопка1_Щелчок()
    h = 2 'величина шага
    a = 2 'коэффициент уменьшения шага
    'начальная точка
    x0 = 0
    y0 = 0
    'счетчик итераций
    iter = 0
    'цикл выполняется, пока величина шага не станет меньше заданной точности
    Do While (h > 0.1)
        ud = 0
        'вычисляется значение функции в соседних точках
        f1 = f(x0, y0)
        f2 = f(x0 + h, y0)
        f3 = f(x0 - h, y0)
        f4 = f(x0, y0 + h)
        f5 = f(x0, y0 - h)
        x0p = x0
        f6 = f1
        If (f2 < f1) Then 'исследующий поиск: если увеличение одного аргумента
            'приводит к уменьшению функции
            x0p = x0 + h + (x0 + h - x0) 'поиск по образцу: переход в новую точку
            ud = 1 'поиск удачный
            f6 = f(x0 + h, y0 + h)
            f7 = f(x0 + h, y0 - h)
            If f6 < f2 Then 'исследующий поиск: если увеличение второго аргумента
                'приводит к уменьшению функции
                y0p = y0 + h + (y0 + h - y0) 'поиск по образцу: переход в новую точку
            ElseIf f7 < f2 Then 'исследующий поиск: если уменьшение второго аргумента
                'приводит к уменьшению функции
                y0p = y0 - h + (y0 - h - y0) 'поиск по образцу: переход в новую точку
            End If
        ElseIf (f3 < f1) Then 'исследующий поиск: если уменьшение одного аргумента
            'приводит к уменьшению функции
```

```
        x0p = x0 - h + (x0 - h - x0) 'поиск по образцу: переход в новую точку
        ud = 1 'поиск удачный
        f6 = f(x0 - h, y0 + h)
        f7 = f(x0 - h, y0 - h)
        If f6 < f2 Then 'исследующий поиск: если увеличение второго аргумента
            'приводит к уменьшению функции
            y0p = y0 + h + (y0 + h - y0) 'поиск по образцу: переход в новую точку
        ElseIf f7 < f2 Then 'исследующий поиск: если уменьшение второго аргумента
            'приводит к уменьшению функции
            y0p = y0 - h + (y0 - h - y0) 'поиск по образцу: переход в новую точку
        End If
    End If
    y0p = y0
    If (ud = 0) And (f4 < f1) Then 'исследующий поиск: если увеличение второго
        'аргумента приводит к уменьшению функции
        y0p = y0 + h + (y0 + h - y0) 'поиск по образцу: переход в новую точку
        ud = 1 'поиск удачный
    ElseIf (ud = 0) And (f5 < f1) Then 'исследующий поиск: если уменьшение второго
        'аргумента приводит к уменьшению функции
        y0p = y0 - h + (y0 - h - y0) 'поиск по образцу: переход в новую точку
        ud = 1 'поиск удачный
    End If
    f1 = f(x0p, y0p)
    f2 = f(x0p + h, y0p)
    f3 = f(x0p - h, y0p)
    f4 = f(x0p, y0p + h)
    f5 = f(x0p, y0p - h)
    If ud = 1 Then 'если поиск удачный, то выполняется новый исследующий поиск
        If (f2 < f1) And (f2 <= f3) And (f2 <= f4) And (f2 <= f5) Then
            x0 = x0p + h 'переход в новую точку
            y0 = y0p
        ElseIf (f3 < f1) And (f3 <= f2) And (f3 <= f4) And (f3 <= f5) Then
            x0 = x0p - h 'переход в новую точку
            y0 = y0p
        ElseIf (f4 < f1) And (f4 <= f2) And (f4 <= f3) And (f4 <= f5) Then
            x0 = x0p 'переход в новую точку
            y0 = y0p + h
        ElseIf (f5 < f1) And (f5 <= f2) And (f5 <= f3) And (f5 <= f4) Then
            x0 = x0p 'переход в новую точку
            y0 = y0p - h
        Else
            x0 = x0p
            y0 = y0p
        End If
    Else
        h = h / 2 'если наилучшая точка не найдена, то шаг уменьшая в 2 раза
    End If
  Loop
  'вывод результата в ячейки
  Range("J1").Value = x0
  Range("K1").Value = y0
End Sub
```

Рис. 1.33 Код метода Хука-Дживса

Результат работы программы представлен на рис.1.34.

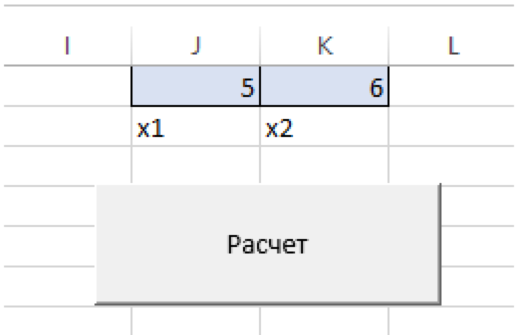


Рис.1.34 Результат работы алгоритма Хука-Дживса

Выполним вручную две итерации.

1 итерация.

Значение функции в начальной точке (0;0):

$f(0,0)=(0-5)^2+(0-6)^2=61$

Уменьшим значение первого аргумента:

$x_1=x_1-2=0-2=-2$.

Вычислим значение функции в новой точке:

$f(x_1,x_2)=(-2-5)^2+(0-6)^2=85$ (не успех).

Увеличим значение первого аргумента:

$x_1=x_1+2=0+2=2$.

Вычислим значение функции в новой точке:

$f(x_1,x_2)=(2-5)^2+(0-6)^2=45$ (успех).

Увеличим значение второго аргумента:

$x_2=x_2+2=0+2=2$.

Вычислим значение функции в новой точке:

$f(x_1,x_2)=(2-5)^2+(2-6)^2=25$ (успех).

Выполним поиск по образцу:

$x_1^p=x_1+h+(x_1+h-x_1)=2+2=4$

$x_2^p=x_2+h+(x_2+h-x_2)=2+2=4$

Значение функции в этой точке:

$f(x_1,x_2)=(4-5)^2+(4-6)^2=5$.

Проводим поиск по образцу вокруг полученной точки.

Увеличим значение первого аргумента:

$x_1^p = x_1^p + h = 4 + 2 = 6$.

Вычислим значение функции:

$f(x_1, x_2) = (6-5)^2 + (4-6)^2 = 5$ (не успех).

Увеличим значение второго аргумента:

$x_1^p = x_2^p + h = 4 + 2 = 6$.

Вычислим значение функции:

$f(x_1, x_2) = (4-5)^2 + (6-6)^2 = 1$ (успех).

Полученная точка первой итерации – (4;6).

2 итерация.

Уменьшим значение первого аргумента:

x_1=x_1-2=4-2=2.

Вычислим значение функции:

$f(x_1, x_2) = (2-5)^2 + (6-6)^2 = 9$ (не успех)

Увеличим значение первого аргумента:

x_1=x_1+2=4+2=6.

Вычислим значение функции:

$f(x_1, x_2) = (6-5)^2 + (6-6)^2 = 1$ (не успех).

Увеличим значение второго аргумента:

x_2=x_2+2=6+2=8.

Вычислим значение функции:

$f(x_1, x_2) = (4-5)^2 + (8-6)^2 = 5$ (не успех).

Уменьшим значение второго аргумента:

x_2=x_2-2=6-2=4.

Вычислим значение функции:

$f(x_1, x_2) = (4-5)^2 + (4-6)^2 = 5$ (не успех).

Поскольку исследующий оказался неудачным, то происходит уменьшение величины шага: $h = \frac{h}{2} = \frac{2}{2} = 1$ и выполняется переход к следующей итерации.

Рассмотрим теперь решение задачи методом градиентного спуска. Код программы представлен на рис.1.35.

```
Sub Кнопка1_Щелчок()
    'начальная точка
    x1 = 0
    x2 = 0
    'параметр спуска
    a = 0.1
     'зададим 1000 итераций
    For i = 1 To 1000
        dx1 = 2 * (x1 - 5) 'значение первой частной производной в точке
        dx2 = 2 * (x2 - 6) 'значение второй частной производной в точке
        'переход в новую точку
        x1 = x1 - a * dx1
        x2 = x2 - a * dx2
    Next
    'вывод полученного результата в ячейки
    Range("J1").Value = x1
    Range("K1").Value = x2
End Sub
```

Рис.1.35 Код метода градиентного спуска

Результат программы представлен на рис.1.36.

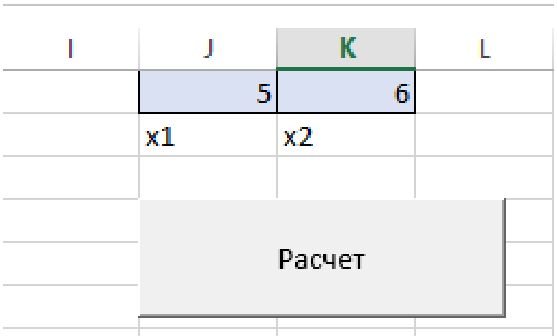


Рис.1.36 Результат работы програмы

Выполним вручную две итерации.

1 итерация.

Вычислим частные производные функции:

$$\frac{\partial}{\partial x_1} f(x_1, x_2) = 2(x_1 - 5).$$

$$\frac{\partial}{\partial x_2} f(x_1, x_2) = 2(x_2 - 6).$$

Значения частные производных в начальной точке (0;0): $2(0-5) = -10$;

$2(0-6) = -12$.

Новые координаты точки:

$$x_1 = x_1 - \alpha \frac{\partial}{\partial x_1} f(x_1, x_2)$$

$$x_2 = x_2 - \alpha \frac{\partial}{\partial x_2} f(x_1, x_2)$$

$$x_1 = 0 - 0{,}1 \cdot 2(0-5) = -1$$

$$x_2 = 0 - 0{,}1 \cdot 2(0-6) = -1{,}2$$

2 итерация.

Новые координаты точки:

$$x_1 = -1 - 0{,}1 \cdot 2(-1-5) = -2{,}2$$

$$x_2 = -1{,}2 - 0{,}1 \cdot 2(-1{,}2 - 6) = -2{,}64.$$

Заключение

В работе выполнено построение графика функции, найдена точка минимума методом Хука-Дживса и методом градиентного спуска. Преимуществом метода градиентного спуска является простота реализации, однако необходимо вычислять значение производной и определять значение параметра α.

Примеры отчетов по лабораторной работе №3

Министерство образования и науки Российской Федерации

Федеральное государственное бюджетное образовательное учреждение
высшего профессионального образования

ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ СИСТЕМ УПРАВЛЕНИЯ И РАДИОЭЛЕКТРОНИКИ

Факультет дистанционного образования (ФДО)

Кафедра автоматизированных систем (АСУ)

Условная оптимизация

Отчет по лабораторной работе № 3 по дисциплине
«Исследование операций и методы оптимизации»

Выполнил:
Студент гр.____
И.О. Фамилия
«   »__________2017 г.

Руководитель:
И.О. Фамилия руководителя
«   »__________2017 г.

2017

Вариант 1

Задание 1

1) В таблице 1.4 представлена информация о некоторых продуктах: количестве белка, жиров и углеводов, содержащихся в них, а также калорийность и цена (за 100 г.). Необходимо сформировать дневной рацион из 10-15 продуктов, считая, что суточная потребность человека в белке, жирах, углеводах и энергии составляет соответственно 60 г., 70 г., 280 г. и 1826 килокалорий.

2) Добавьте условие разнообразия рациона, согласно которому человек не может в день употреблять более 300 г. продукта одно вида.

3) Рассчитайте стоимость полученного набора продуктов в первом и во втором случае.

Для решения этих задач необходимо построить математическую модель и реализовать ее в пакете MathCad, Excel или с помощью математических библиотек.

Таблица 1.4 - Характеристики продуктов

	Название продукта	**Белки**	**Жиры**	**Углеводы**	**Ккал**	**Цена**
1	Яйцо куриное	12.7	11.5	0.7	157	15
2	Арахис	26.3	45.2	9.7	550	21
3	Горох цельный	23.0	1.2	53.3	316	30
4	Грецкий орех	13.8	61.3	10.2	647	44
5	Крупа гречневая	12.6	2.6	68.0	345	6
6	Пшено	12.0	2.9	69.3	351	3,2
7	Рис	8.0	1.0	76.0	345	6
8	Творог	7.1	23.0	27.5	345	13
9	Сыр	27.0	40.0	0.0	468	18
10	Колбаса вареная Любительская	12.2	28.0	0.0	300	17
11	Колбаса варено-копченая Сервелат	28.2	27.5	0.0	360	18
12	Сосиски Молочные	12.3	25.3	0.0	276	18
13	Говядина	18.9	12.4	0.0	187	21
14	Свинина	16.4	27.8	0.0	315	23
15	Кабачки	0.6	0.3	5.7	27	2
16	Капуста белокочанная	1.8	0.0	5.4	28	1,8

17	Картофель	2.0	0.1	19.7	87	2,5
18	Морковь	1.3	0.1	7.0	34	2
19	Огурцы	0.8	0.0	3.0	15	3
20	Перец красный сладкий	1.3	0.0	5.7	28	6,5
21	Свекла	1.7	0.0	10.8	50	2
22	Горбуша	21.0	7.0	0.0	147	12
23	Икра осетровая зернистая	28.9	9.7	0.0	202	164
24	Скумбрия	18.0	9.0	0.0	153	15
25	Шоколад темный	5.4	35.3	52.6	549	30
26	Груша	2.3	0.0	62.1	257	9
27	Персики	3.0	0.0	68.5	286	10
28	Яблоки	3.2	0.0	68.0	284	7
29	Апельсин	0.9	0.0	8.4	37	7,5
30	Бананы	1.5	0.0	22.0	94	4
31	Черешня	1.1	0.0	12.3	53	18
32	Макаронные изделия	11.0	0.9	74.2	348	3
33	Хлеб пшеничный из муки 1 сорта	7.7	2.4	53.4	266	3

Задание 2

Заводы производственной фирмы расположены в городах Омск, Новосибирск, Томск. Центры распределения расположены в городах Нижний Новгород, Пермь, Краснодар. Объемы производства и величина спроса в пунктах представлены в таблице 1.5. Одно изделие имеет вес 3 кг. и объем 0,8 м3. Стоимость перевозки рассчитайте с помощью онлайн-калькулятора http://www.jde.ru/calc.

Составьте экономико-математическую модель задачи. С помощью пакета *MathCAD, Excel* или с помощью математических библиотек найдите оптимальное распределение поставок и минимальные затраты на перевозку.

Таблица 1.5 – Варианты заданий

Вариант	Объем производства в пункте			Величина спроса в пункте		
	Омск	Новосибирск	Томск	Нижний Новгород	Пермь	Краснодар
1	1000	2000	1200	2000	1100	1100

Решение

Задание 1.

1) Выберем следующие 5 продуктов, представленные в таблице 1.6.

Таблица 1.6 Выбранные продукты

№ п/п	Наименование продукта	Белки	Жиры	Углеводы	Ккал	Стоимость, 100 гр продукта руб
1	Яйцо куриное	12,07	11,50	0,70	157,00	15,00
2	Арахис	26,30	45,20	9,70	550,00	21,00
3	Горох цельный	23,00	1,20	53,30	316,00	30,00
4	Грецкий орех	13,80	61,30	10,20	647,00	44,00
5	Крупа гречневая	12,60	2,60	68,00	345,00	6,00
	Необх. минимум	60	70	280	1826	

Обозначим через $x_1,...,x_5$ - количество продуктов, входящих в дневной рацион. Из условия задачи получаем следующую систему неравенств

$$\begin{cases} 12,07x_1 + 26,30x_2 + 23,00x_3 + 13,80x_4 + 12,60x_5 \ge 60; \\ 11,5x_1 + 45,20x_2 + 1,2x_3 + 61,3x_4 + 2,60x_5 \ge 70; \\ 0,7x_1 + 9,7x_2 + 53,3x_3 + 10,2x_4 + 68x_5 \ge 280; \\ 157x_1 + 550x_2 + 316x_3 + 647x_4 + 345x_5 \ge 1826. \end{cases}$$

Кроме того, значения переменных должны удовлетворять следующему условию

$x_1, x_2, ..., x_{15} \ge 0$

Целевая функция, характеризующая общую стоимость, будет иметь следующий вид

$$f = 15x_1 + 21x_2 + 30x_3 + 44x_4 + 6x_5 \to \min .$$

Реализуем данную модель в пакете Excel (рис.1.37). Для этого в ячейках D9:D12 запишем формулы для расчета содержания веществ в выбранных продуктах:

[D9]= B2*B17+B3*B18+B4*B19+B5*B20+B6*B21 (белки)

[D10]= C2*B17+C3*B18+C4*B19+C5*B20+C6*B21 (жиры)

[D11]= D2*B17+D3*B18+D4*B19+D5*B20+D6*B21 (углеводы)

[D12]= E2*B17+E3*B18+E4*B19+E5*B20+E6*B21 (калории)

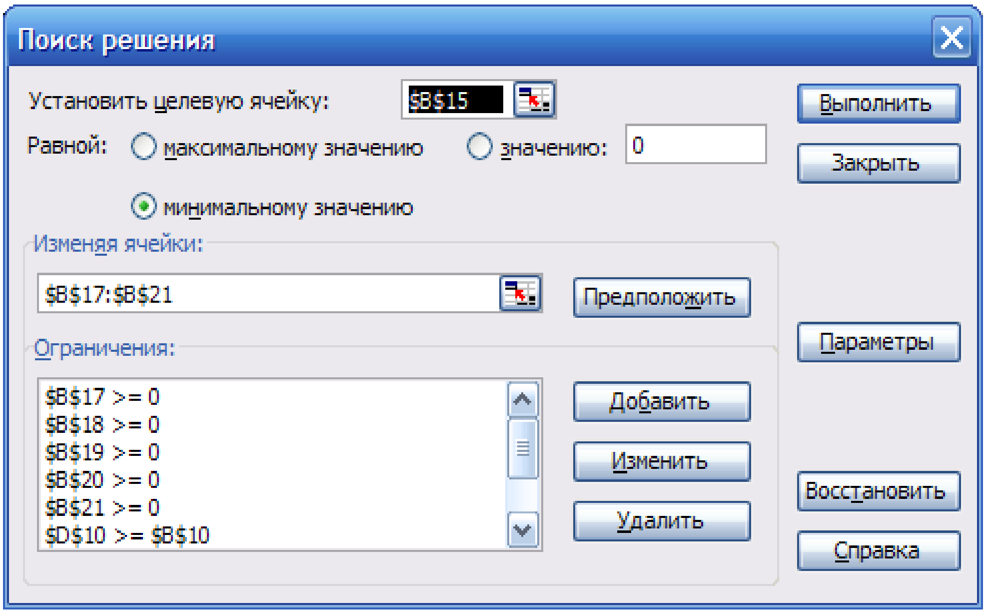
В ячейке B15 запишем расчет целевой функции:

[B15] =B17*F2+B18*F3+B19*F4+B20*F5+B21*F6

Далее нужно вызвать надстройку «Поиск решения» и заполнить значение целевой функции и ограничений (рис.1.38).

	A	B	C	D	E	F
1	**Название продукта**	**Белки**	**Жиры**	**Углеводы**	**Ккал**	**Цена**
2	Яйцо куриное	12,07	11,50	0,70	157,00	15,00
3	Арахис	26,30	45,20	9,70	550,00	21,00
4	Горох цельный	23,00	1,20	53,30	316,00	30,00
5	Грецкий орех	13,80	61,30	10,20	647,00	44,00
6	Крупа гречневая	12,60	2,60	68,00	345,00	6,00
7						
8	Ограничения			Формулы ограничения		
9	Белки	60,00		84,29126281		
10	Жиры	70,00		69,99999983		
11	Углеводы	280,00		280		
12	Энергия	1826,00		2082,963736		
13						
14						
15	Целевая функция	51,3498972				
16		Найденное решение				
17	X1	0				
18	X2	1,32266974				
19	X3	0				
20	X4	0				
21	X5	3,92897211				
22						

Рис.1.37 Реализация модели



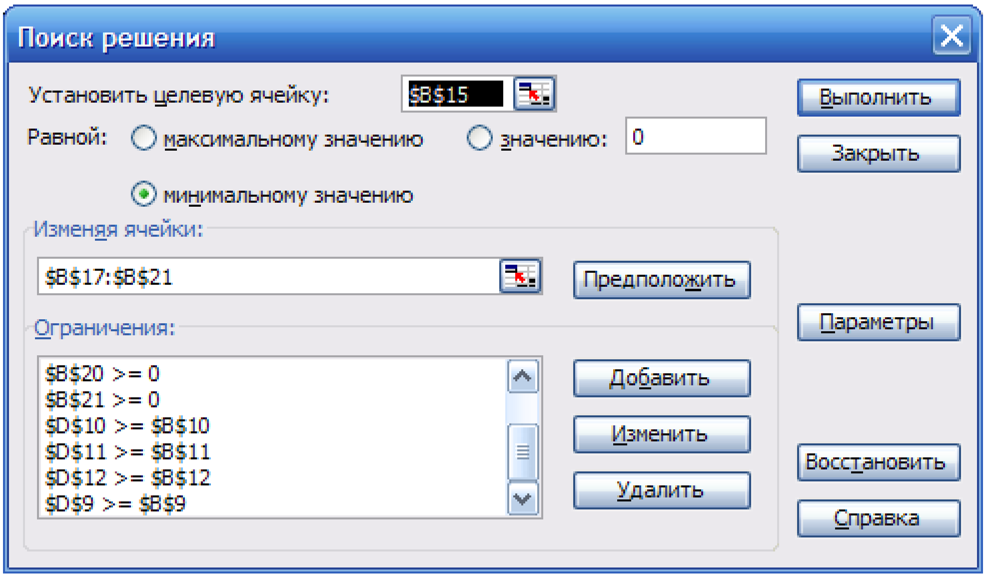


Рис.1.38 Надстройка «Поиск решения»

Результатом решения задачи будут следующий набор продуктов: 132 г. арахиса и 393 г. гречневой крупы.

2) Добавим условие разнообразия рациона, согласно которому человек не может в день употреблять более 300 г. продукта одно вида. Для этого в ограничениях запишем следующие формулы:

B17<=3; B18<=3; B19<=3; B20<=3; B21<=3; B22<=3.

Результат представлен на рис. 1.39.

14		
15	Целевая функция	81,674572
16		Найденное решение
17	X1	0
18	X2	1,34474791
19	X3	1,1811622
20	X4	0
21	X5	3
22		

Рис.1.39 Результат расчета

Результатом решения задачи будут следующий набор продуктов: 134 г. арахиса, 118 г. гороха и 300 г. гречневой крупы.

3) Стоимость полученного набора продуктов в первом и во втором случае рассчитана в ячейке B15 на рис.1.37,1.39:

Стоимость набора 1 = 51,35 руб.

Стоимость набора 2= 81,67 руб.

Задание 2.

Рассчитаем с помощью сайта http://www.jde.ru/calc стоимость доставки единицы товара (рис.1.40).

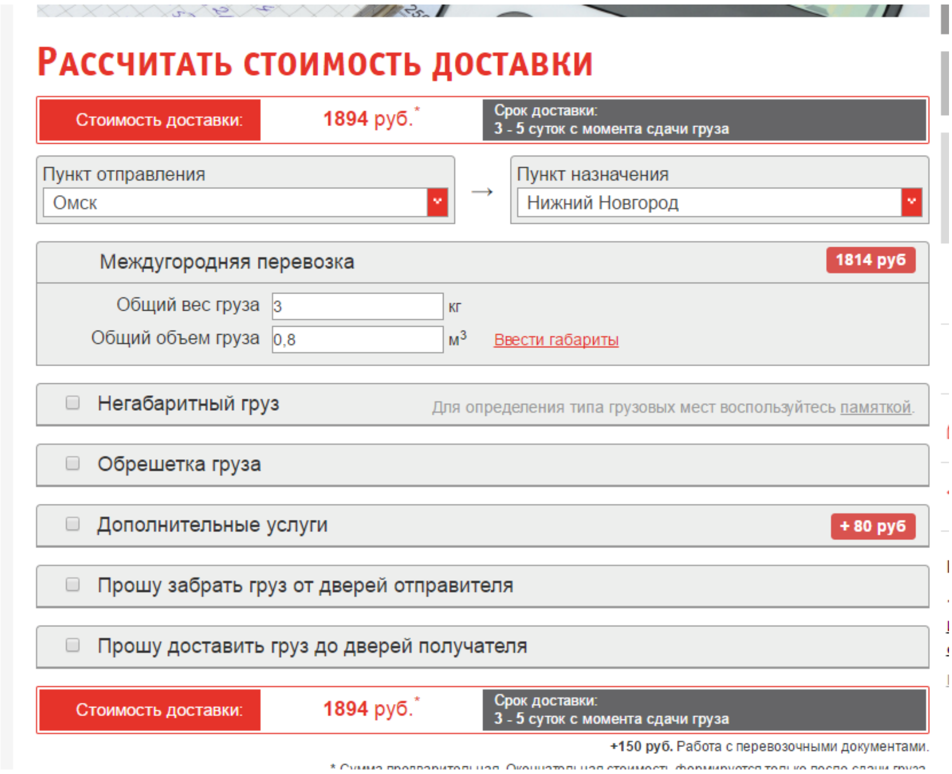


Рис.1.40 Расчет стоимости доставки

Рассчитанные значения представлены в таблице 1.7.

Таблица 1.7 Стоимость доставки

	Нижний Новгород	Пермь	Краснодар
Омск	1894	1552	3026
Новосибирск	2352	1748	3358
Томск	2630	2262	3358

Обозначим через x_{ij} - количество товара, перевозимого из пункта i в пункт j.

Из условия задачи (табл.2) получаем следующую систему неравенств

$$\begin{cases} x_{11}+x_{12}+x_{13}=1000; \\ x_{21}+x_{22}+x_{23}=2000; \\ x_{31}+x_{32}+x_{33}=1200; \\ x_{11}+x_{21}+x_{31}=2000; \\ x_{12}+x_{22}+x_{32}=1100; \\ x_{13}+x_{23}+x_{33}=1100. \end{cases}$$

Кроме того, значения переменных должны удовлетворять следующему условию

$$x_{11}, x_{12}, \ldots, x_{33} \geq 0$$

Целевая функция, характеризующая общую стоимость, будет иметь следующий вид

$$f = 1894x_{11} + 1552x_{12} + 3026x_{13} + 2352x_{21} + 1748x_{22} + \\ + 3358x_{23} + 2630x_{31} + 2262x_{32} + 3358x_{33} \to \min .$$

Реализуем данную модель в Excel (рис.1.41).

	A	B	C	D	E	F	G
1	Объем производства в пункте			Величина спроса в пункте			
2							
3	Омск	Новосибирск	Томск	Нижний Новгород	Пермь	Краснодар	
4	1000	2000	1200	2000	1100	1100	
5							
6							
7							
8		Нижний Новгород	Пермь	Краснодар			
9	Омск	1894	1552	3026			
10	Новосибирск	2352	1748	3358			
11	Томск	2630	2262	3358			
12							
14	Функция						
15	9890400						
16							
17	*Ограничения*						
18	Омск	1000					
19	Новосибирск	2000					
20	Томск	1200					
21	Нижний Новгород	2000					
22	Пермь	1100					
23	Краснодар	1100					

H	I	J	K	L	M	N
	Решение	Нижний Новгород	Пермь	Краснодар		
	Омск	1000	0	0		1000
	Новосибирск	900	1100	0		2000
	Томск	100	0	1100		1200
		2000	1100	1100		

Рис.1.41 Результат вычислений

Для этого в ячейках B18:B23 запишем формулы для расчета объема перевозимого товара для каждого пункта:

[B18]= J2+K2+L2

[B19]= J3+K3+L3

[B20] =J4+K4+L4

[B21] =J2+J3+J4

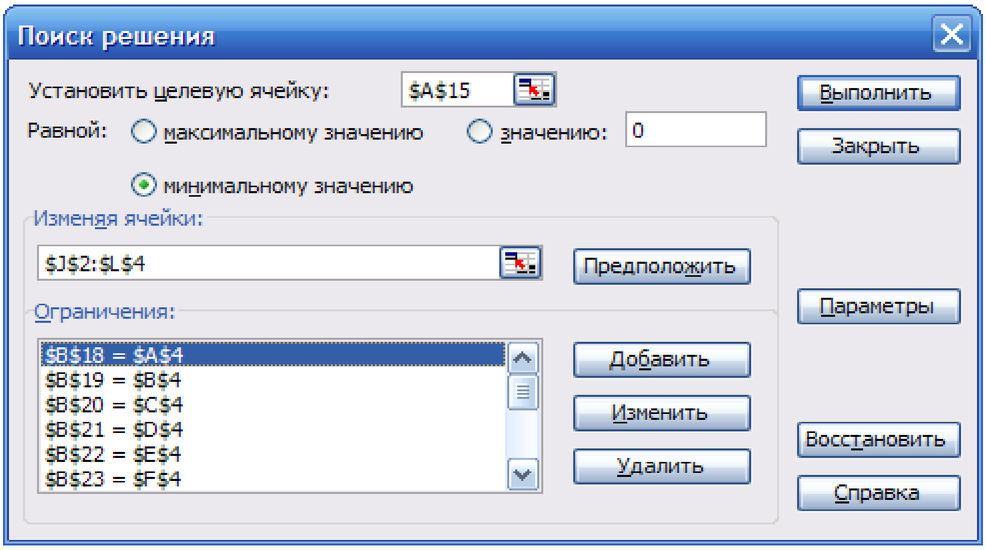
[B22] =K2+K3+K4

[B23] =L2+L3+L4

В ячейке A15 запишем расчет целевой функции:

[A15] =J2*B9+K2*C9+L2*D9+J3*B10+K3*C10+L3*D10+J4*B11+K4*C11+L4*D11

Далее вызываем надстройку «Поиск решения» и заполняем значение целевой функции и ограничений (рис.1.42).



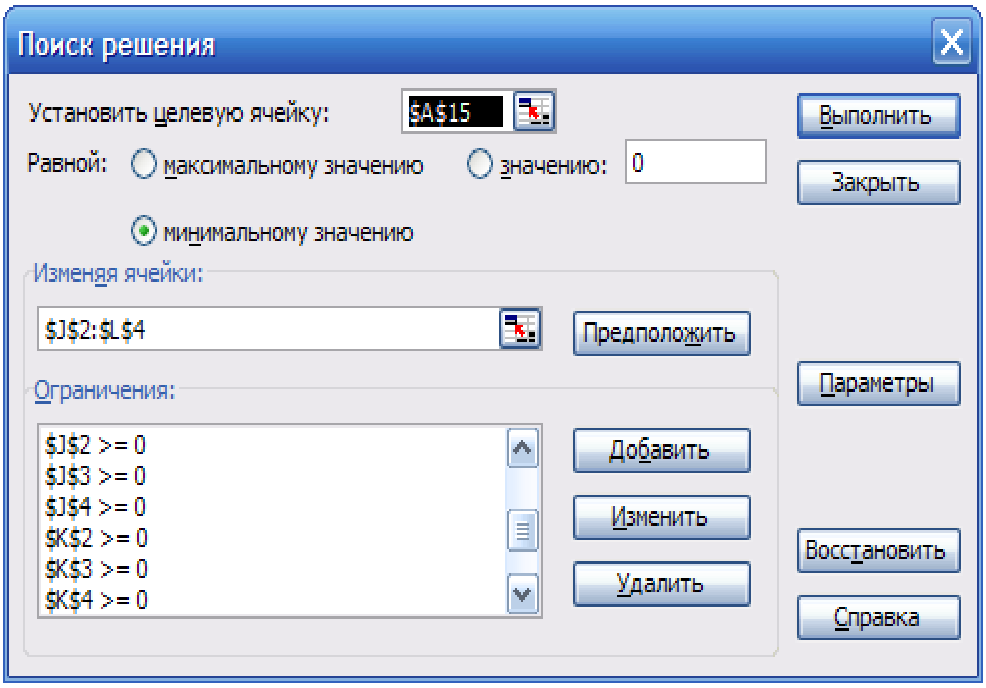


Рис.1.42 Надстройка «Поиск решения»

Таким образом, полученное решение:

- из Омска в Нижний Новгород нужно поставить 1000 изделий;
- из Новосибирска в Нижний Новгород нужно поставить 900 изделий;
- из Новосибирска в Пермь нужно поставить 1100 изделий;
- из Томска в Краснодар нужно поставить 1100 изделий;
- из Томска в Нижний Новгород нужно поставить 100 изделий.

Полученные значения удовлетворяют ограничениям.

Затраты на перевозку составляют 9890400 руб.

Заключение

Выполнено решение задач линейного программирования: задачи о диете и транспортной задачи. Реализация моделей выполнена в Excel. Для решения была использована надстройка Excel “Поиск решения”.

Вариант 2

Задание аналогично, представленному в варианте 1.

Решение

Задание 1

1) Выполним задание 1 варианта 1 с помощью пакета MathCad (рис.1.43).

$$A := \begin{pmatrix} 12.07 & 11.5 & 0.7 & 157 \\ 26.3 & 45.2 & 9.7 & 550 \\ 23 & 1.2 & 53.3 & 316 \\ 13.8 & 61.3 & 10.2 & 647 \\ 12.6 & 2.6 & 68 & 345 \end{pmatrix} \qquad B := \begin{pmatrix} 60 \\ 70 \\ 280 \\ 1826 \end{pmatrix} \qquad C := \begin{pmatrix} 15 \\ 21 \\ 30 \\ 44 \\ 6 \end{pmatrix}$$

$$n := 0..4$$

$$f(x) := \sum_{i=0}^{4} (x_i \cdot C_i) \qquad x_n := 0$$

Given

$$A^T \cdot x \ge B$$

$$x \ge 0$$

$$y2 := \text{Minimize}(f, x) \qquad y2 = \begin{pmatrix} 0 \\ 1.323 \\ 0 \\ 0 \\ 3.929 \end{pmatrix}$$

$$f(y2) = 51.35$$

Рис. 1.43 Реализация модели в MathCad

Результатом решения задачи будет следующий набор продуктов: 132 г. арахиса и 393 г. гречневой крупы.

2) Добавим условие, согласно которому объем продукта не может превышать 300 г. ($x \le 3$) (рис. 1.44).

Результатом решения задачи будут следующий набор продуктов: 134 г. арахиса, 118 г. гороха и 300 г. гречневой крупы.

3) Стоимость набора продуктов в первом варианте равна 51,35 руб. Стоимость набора продуктов во втором варианте равна 81,67 руб.

$$A := \begin{pmatrix} 12.07 & 11.5 & 0.7 & 157 \\ 26.3 & 45.2 & 9.7 & 550 \\ 23 & 1.2 & 53.3 & 316 \\ 13.8 & 61.3 & 10.2 & 647 \\ 12.6 & 2.6 & 68 & 345 \end{pmatrix} \qquad B := \begin{pmatrix} 60 \\ 70 \\ 280 \\ 1826 \end{pmatrix} \qquad C := \begin{pmatrix} 15 \\ 21 \\ 30 \\ 44 \\ 6 \end{pmatrix}$$

$$n := 0..4$$

$$f(x) := \sum_{i=0}^{4} \left(x_i \cdot C_i\right) \qquad x_n := 0$$

Given

$$A^T \cdot x \geq B$$

$$x \geq 0 \qquad x \leq 3$$

$$y2 := \mathrm{Minimize}(f, x) \qquad y2 = \begin{pmatrix} 0 \\ 1.345 \\ 1.181 \\ 0 \\ 3 \end{pmatrix}$$

$$f(y2) = 81.675$$

Рис.1.44 Решение задачи с ограничением на объем продуктов

Задание 2.

Выполним задание 2 варианта 1 с помощью пакета MathCad (рис.1.45).

Таким образом, полученное решение:

- из Омска в Нижний Новгород нужно поставить 1000 изделий;
- из Новосибирска в Нижний Новгород нужно поставить 900 изделий;
- из Новосибирска в Пермь нужно поставить 1100 изделий;
- из Томска в Краснодар нужно поставить 1100 изделий;
- из Томска в Нижний Новгород нужно поставить 100 изделий.

Полученные значения удовлетворяют ограничениям (табл.1.8).

Таблица 1.8 Проверка решения

	Нижний Новгород	Пермь	Краснодар	***Сумма***
Омск	1000	0	0	***1000***
Новосибирск	900	1100	0	***2000***

Томск	100	0	1100	***1200***
Сумма	***2000***	***1100***	***1100***	

Затраты на перевозку составляют 9890000 руб.

Заключение

Выполнено решение задач линейного программирования: задачи о диете и транспортной задачи. Реализация моделей выполнена в MathCad.

$$A := \begin{pmatrix} 1894 & 1552 & 3026 \\ 2352 & 1748 & 3358 \\ 2630 & 2262 & 3358 \end{pmatrix} \qquad B := \begin{pmatrix} 1000 \\ 2000 \\ 1200 \end{pmatrix} \qquad C := \begin{pmatrix} 2000 \\ 1100 \\ 1100 \end{pmatrix}$$

$$n := 0..2$$

$$f(x) := \sum_{i=0}^{2} \sum_{j=0}^{2} \left(x_{i,j} \cdot A_{i,j}\right)$$

$$x_{n,n} := 0$$

Given

$$\sum_{i=0}^{2} \left(x_{0,i}\right) = B_0 \qquad \sum_{i=0}^{2} \left(x_{1,i}\right) = B_1 \qquad \sum_{i=0}^{2} \left(x_{2,i}\right) = B_2$$

$$\sum_{i=0}^{2} \left(x_{i,0}\right) = C_0 \qquad \sum_{i=0}^{2} \left(x_{i,1}\right) = C_1 \qquad \sum_{i=0}^{2} \left(x_{i,2}\right) = C_2$$

$$x \geq 0 \qquad y := \mathrm{Minimize}(f, x)$$

$$\begin{array}{ll} & \begin{array}{ccc} \text{НижнийНовогород} & \text{Пермь} & \text{Краснодар} \end{array} \\ y = \begin{pmatrix} 1 \times 10^3 & 0 & 0 \\ 900 & 1.1 \times 10^3 & 0 \\ 100 & 0 & 1.1 \times 10^3 \end{pmatrix} & \begin{array}{l} \text{Омск} \\ \text{Новосибирск} \\ \text{Томск} \end{array} \end{array}$$

$$f(y) = 9.89 \times 10^6$$

Рис.1.45 Решение задачи в MathCad

Вариант 3

Задание аналогично, представленному в варианте 1.

Решение

Задание 1

1) Выполним задание 1 варианта 1 с помощью библиотеки common-math3 языка программирования java. Код программы представлен на рис.1.46.

```
import java.util.ArrayList;

public class linear {

    public static void main(String[] args) {
        // TODO Auto-generated method stub
        LinearObjectiveFunction f = new LinearObjectiveFunction(new
                double[] { 15, 21, 30, 44, 6 }, 0);//значения стоимости 100 г. продукта в целевой функции
                        Collection<LinearConstraint> constraints = new
                ArrayList<LinearConstraint>(); //ограничения, GEQ- отношение "больше или равно"
                        constraints.add(new LinearConstraint(new double[] { 12.07, 26.3, 23, 13.8, 12.6 },
                Relationship.GEQ,  60)); //добавление ограничения - норма белков,
                        constraints.add(new LinearConstraint(new double[] { 11.5, 45.2, 1.2, 61.3, 2.6 },
                Relationship.GEQ,  70)); //добавление ограничения - норма жиров
                        constraints.add(new LinearConstraint(new double[] { 0.7, 9.7, 53.3, 10.2, 68 },
                Relationship.GEQ,  280)); //добавление ограничения - норма углеводов
                        constraints.add(new LinearConstraint(new double[] { 157, 550, 316, 647, 345 },
                Relationship.GEQ,  1826)); //добавление ограничения - норма калорий
        //создание объекта для решения задачи линейного программирования
        SimplexSolver solver = new SimplexSolver();
        //решение задачи оптимизации
        PointValuePair solution = solver.optimize(new MaxIter(1000), f, new //макс. число итераций -1000
                LinearConstraintSet(constraints), //ограничения
                GoalType.MINIMIZE, new //направление целевой функции -минимизация
                NonNegativeConstraint(true));  //значения аргументов положительны
        double fs=solution.getValue();//получение значения целевой функции
        System.out.println(fs); //вывод значения целевой функции
        double[] fx=solution.getPoint(); //получение значений аргументов
        System.out.println(fx[0]); //вывод на экран значения первого аргумента
        System.out.println(fx[1]); //вывод на экран значения второго аргумента
        System.out.println(fx[2]); //вывод на экран значения третьего аргумента
        System.out.println(fx[3]); //вывод на экран значения четвертого аргумента
        System.out.println(fx[4]); //вывод на экран значения пятого аргумента
    }

}
```

Рис. 1.46 Код программы на языку Java

Результатом решения задачи будет следующий набор продуктов: 132 г. арахиса и 393 г. гречневой крупы (рис.1.47).

```
Console
<terminated> linear [Java Application] C:\Prog
51.3498973225123
0.0
1.3226697458978218
0.0
0.0
3.9289721097763404
```

Рис.1.47 – Результат работы программы

4) Добавим условие, согласно которому объем продукта не может превышать 300 г. ($x \le 3$) (ограничения представлены на рис. 1.48).

```
        Collection<LinearConstraint> constraints = new
 ArrayList<LinearConstraint>(); //ограничения, GEQ- отношение "больше или равно"
        constraints.add(new LinearConstraint(new double[] { 12.07, 26.3, 23, 13.8, 12.6 },
 Relationship.GEQ,  60)); //добавление ограничения - норма белков,
        constraints.add(new LinearConstraint(new double[] { 11.5, 45.2, 1.2, 61.3, 2.6 },
 Relationship.GEQ,  70)); //добавление ограничения - норма жиров
        constraints.add(new LinearConstraint(new double[] { 0.7, 9.7, 53.3, 10.2, 68 },
 Relationship.GEQ,  280)); //добавление ограничения - норма углеводов
        constraints.add(new LinearConstraint(new double[] { 157, 550, 316, 647, 345 },
 Relationship.GEQ,  1826)); //добавление ограничения - норма калорий
        constraints.add(new LinearConstraint(new double[] { 1, 0, 0, 0, 0 },
//добавление ограничения на количество продуктов, LEQ - меньше или равно
Relationship.LEQ,  3));
        constraints.add(new LinearConstraint(new double[] { 0, 1, 0, 0, 0 },
Relationship.LEQ,  3));
        constraints.add(new LinearConstraint(new double[] { 0, 0, 1, 0, 0 },
Relationship.LEQ,  3));
        constraints.add(new LinearConstraint(new double[] { 0, 0, 0, 1, 0 },
Relationship.LEQ,  3));
        constraints.add(new LinearConstraint(new double[] { 0, 0, 0, 0, 1 },
Relationship.LEQ,  3));
```

Рис.1.48 Новые ограничения

Результатом решения задачи будут следующий набор продуктов: 134 г. арахиса, 118 г. гороха и 300 г. гречневой крупы (рис.1.49).

```
Console
<terminated> linear [Java Applic
81.67457205779303
0.0
1.3447479061697083
1.18116220940972
0.0
3.0
```

Рис.1.49 Результат работы программы

5) Стоимость набора продуктов в первом варианте равна 51,35 руб. Стоимость набора продуктов во втором варианте равна 81,67 руб. (рис.1.47,1.49)

Задание 2.

Выполним задание 2 варианта 1 с помощью библиотеки common-math3 языка программирования java (рис.1.50).

Полученное решение представлено на рис.1.51:

- из Омска в Нижний Новгород нужно поставить 1000 изделий;
- из Новосибирска в Нижний Новгород нужно поставить 900 изделий;
- из Новосибирска в Пермь нужно поставить 1100 изделий;
- из Томска в Краснодар нужно поставить 1100 изделий;
- из Томска в Нижний Новгород нужно поставить 100 изделий.

Полученные значения удовлетворяют ограничениям (табл.1.8).

Затраты на перевозку составляют 9890000 руб.

```
public static void main(String[] args) {
    // TODO Auto-generated method stub
    //определение целевой функции - стоимости доставки
    LinearObjectiveFunction f = new LinearObjectiveFunction(new
            double[] { 1894, 1552, 3026, 2352, 1748, 3358, 2630, 2262, 3358}, 0);
                    Collection<LinearConstraint> constraints = new
            ArrayList<LinearConstraint>(); //ограничения, EQ- отношение "равно"
                  //добавление ограничений для пунктов производства и потребления
                    constraints.add(new LinearConstraint(new double[] { 1, 1, 1, 0, 0, 0, 0, 0, 0 },
            Relationship.EQ,  1000));
                    constraints.add(new LinearConstraint(new double[] { 0, 0, 0, 1, 1, 1, 0, 0, 0 },
            Relationship.EQ,  2000));
                    constraints.add(new LinearConstraint(new double[] { 0, 0, 0, 0, 0, 0, 1, 1, 1 },
            Relationship.EQ,  1200));
                    constraints.add(new LinearConstraint(new double[] { 1, 0, 0, 1, 0, 0, 1, 0, 0},
            Relationship.EQ,  2000));
                    constraints.add(new LinearConstraint(new double[] { 0, 1, 0, 0, 1, 0, 0, 1, 0 },
           Relationship.EQ,  1100));
                    constraints.add(new LinearConstraint(new double[] { 0, 0, 1, 0, 0, 1, 0, 0, 1},
           Relationship.EQ,  1100));
    //создание объекта для решения задачи линейного программирования
    SimplexSolver solver = new SimplexSolver();
    //решение задачи оптимизации
    PointValuePair solution = solver.optimize(new MaxIter(1000), f, new //макс. число итераций -1000
            LinearConstraintSet(constraints), //ограничения
            GoalType.MINIMIZE, new //направление целевой функции -минимизация
            NonNegativeConstraint(true));  //значения аргументов положительны
    double fs=solution.getValue();//получение значения целевой функции
    System.out.println(fs); //вывод значения целевой функции
    double[] fx=solution.getPoint(); //получение значений аргументов
    //вывод на экран аргументов
    System.out.println(fx[0]);  System.out.println(fx[1]);
    System.out.println(fx[2]);  System.out.println(fx[3]);
    System.out.println(fx[4]);  System.out.println(fx[5]);
    System.out.println(fx[6]);  System.out.println(fx[7]);
    System.out.println(fx[8]);
}
```

Рис.1.50 Код программы

Основные используемые классы и методы библиотеки common-math3:

Класс *LinearObjectiveFunction (double[] coefficients, double constantTerm)* – целевая функция вида $c_1x_1 + ...c_nx_n + d$, где c_i (*coefficients*) и d (*constantTerm*) - коэффициенты выражения, x_i - искомые координаты точки.

Класс *LinearConstraint(double[] coefficients, Relationship relationship, double value)* - определяет линейные ограничения задачи с использованием одного уравнения. Такие ограничения имеют вид:

$$c_1x_1 + ...c_nx_n = v$$

$$c_1x_1 + ...c_nx_n \le v$$

$$c_1x_1 + ...c_nx_n \ge v$$

Параметры:

coefficients – коэффициенты c_i ограничения (в левой части уравнения);

relationship – тип (не) равенства, используемого в ограничении;

value – значение *v* ограничения (число в правой части).

Класс *SimplexSolver* – класс, предназначенный для решения задачи линейного программирования с помощью двухфазного симплексного метода.

Метод *PointValuePair optimize(LinearObjectiveFunction f, Collection<LinearConstraint> constraints, GoalType goalType, boolean restrictToNonNegative)* – выполняет решение задачи оптимизации.

Параметры:

f – целевая функция

constraints – линейные ограничения

goalType – тип направления оптимизации: максимизация (GoalType.MAXIMIZE) или минимизация (GoalType.MINIMIZE)

restrictToNonNegative – определяет, должно ли найденное решение быть неотрицательным.

Возвращаемое значение:

PointValuePair – решение, определяемое оптимальное значение целевой функции.

Класс *PointValuePair* – класс, определяющий точку и значение целевой функции в этой точке. Для получения данных используются два метода:

метод public double[] *getPoint*() – получение точки.

метод public double[] *getValue*()– получение значения целевой функции.

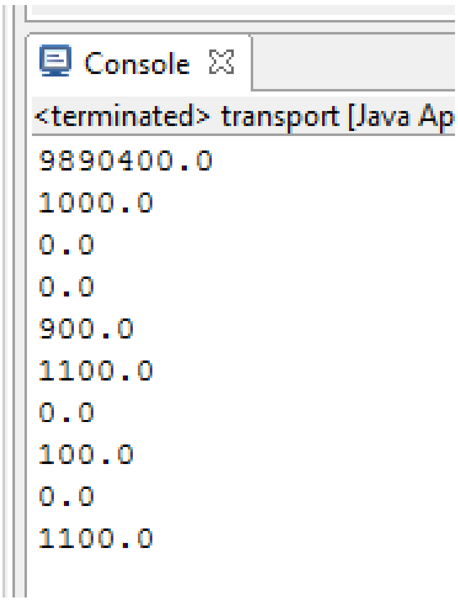


Рис.1.51 Результат программы

Заключение

Выполнено решение задач линейного программирования: задачи о диете и транспортной задачи. Реализация моделей выполнена с помощью языка программирования java и библиотеки common-math3.

Приложение Д. Надстройка Excel «Поиск решения»

Надстройка «Поиск решения» Excel позволяет решать различные оптимизационные задачи. Для её подключения необходимо выбрать «Файл»-«Параметры» (рис.2.1).

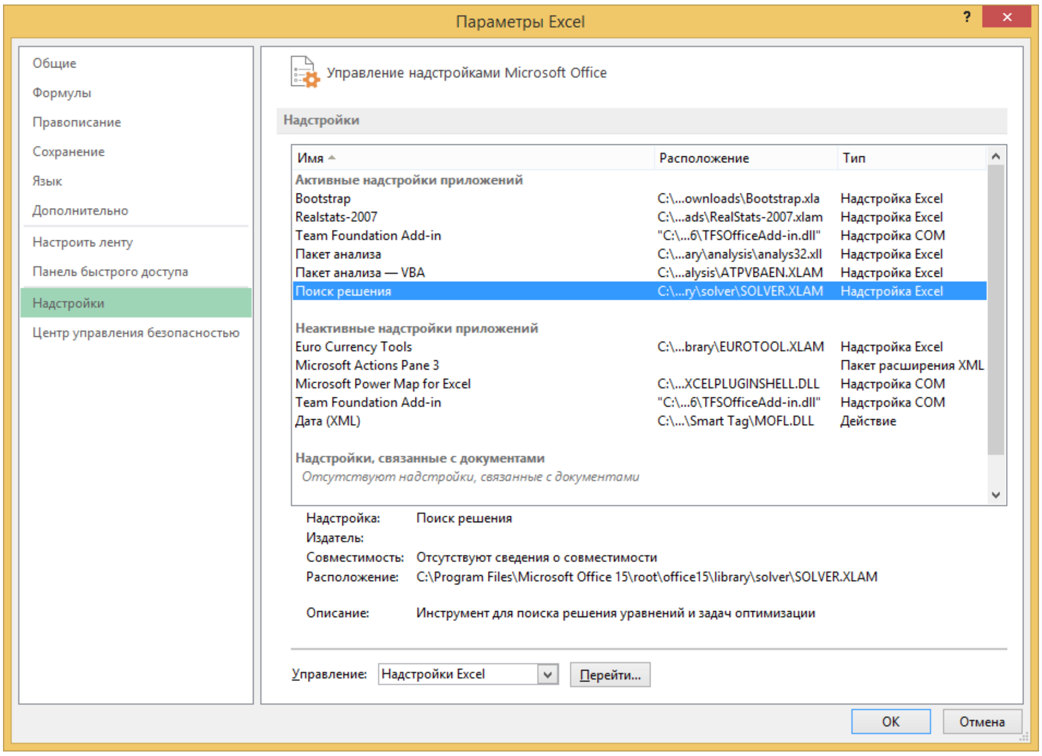


Рис.2.1 Параметры Excel

Затем нужно нажать кнопку «Перейти» и в открывшемся окне (рис.2.2) поставить галочку напротив надстройки «Поиск решения».

После этого надстройка появится в меню (рис.2.3).

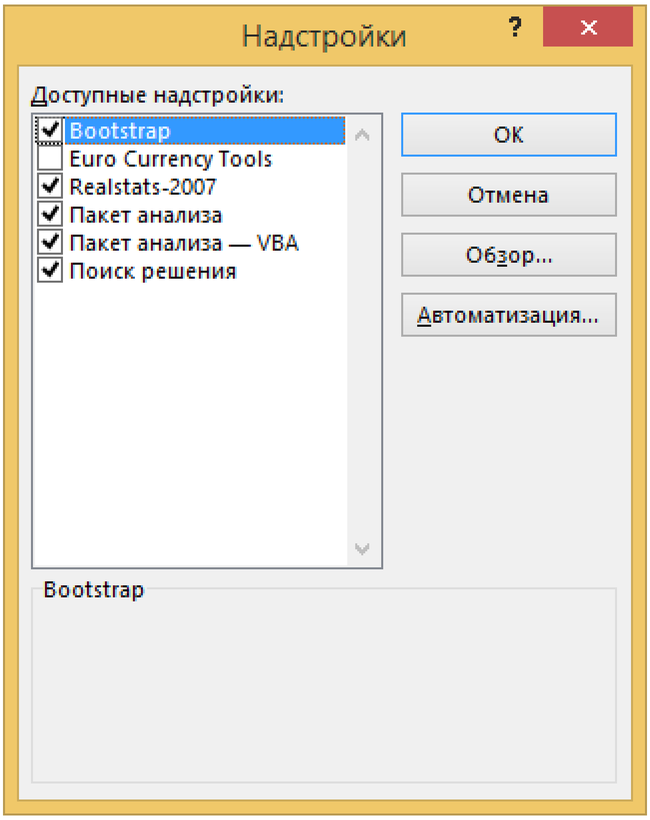


Рис.2.2 – Выбор надстройки «Поиск решения»

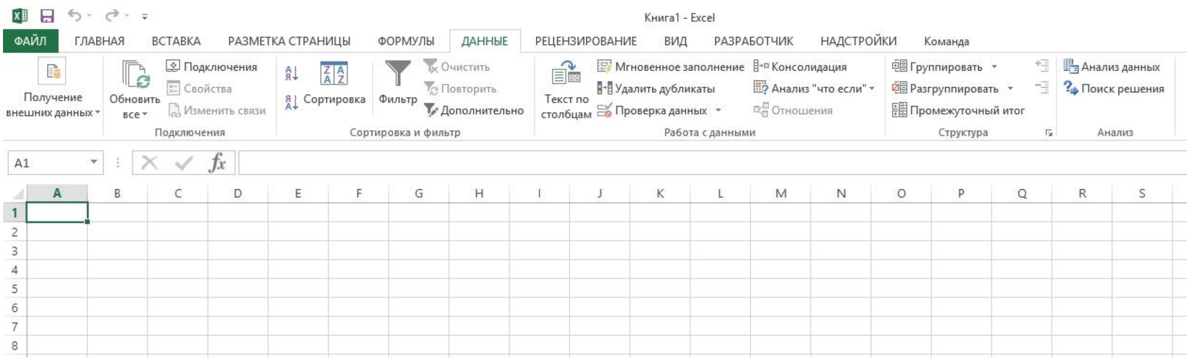


Рис.2.3 – Поиск решения в группе «Анализ»

Для решения задач необходимо указать следующие данные (рис.2.4):

1. Целевую функцию: указывается ссылка на ячейку с формулой;
2. Направление оптимизации: максимум, минимум или целевая функция должна принимать заданное значение;
3. Ячейки, в которых должны быть определены значения аргументов (изменяя ячейки переменных);
4. Ограничения;
5. Метод решения.

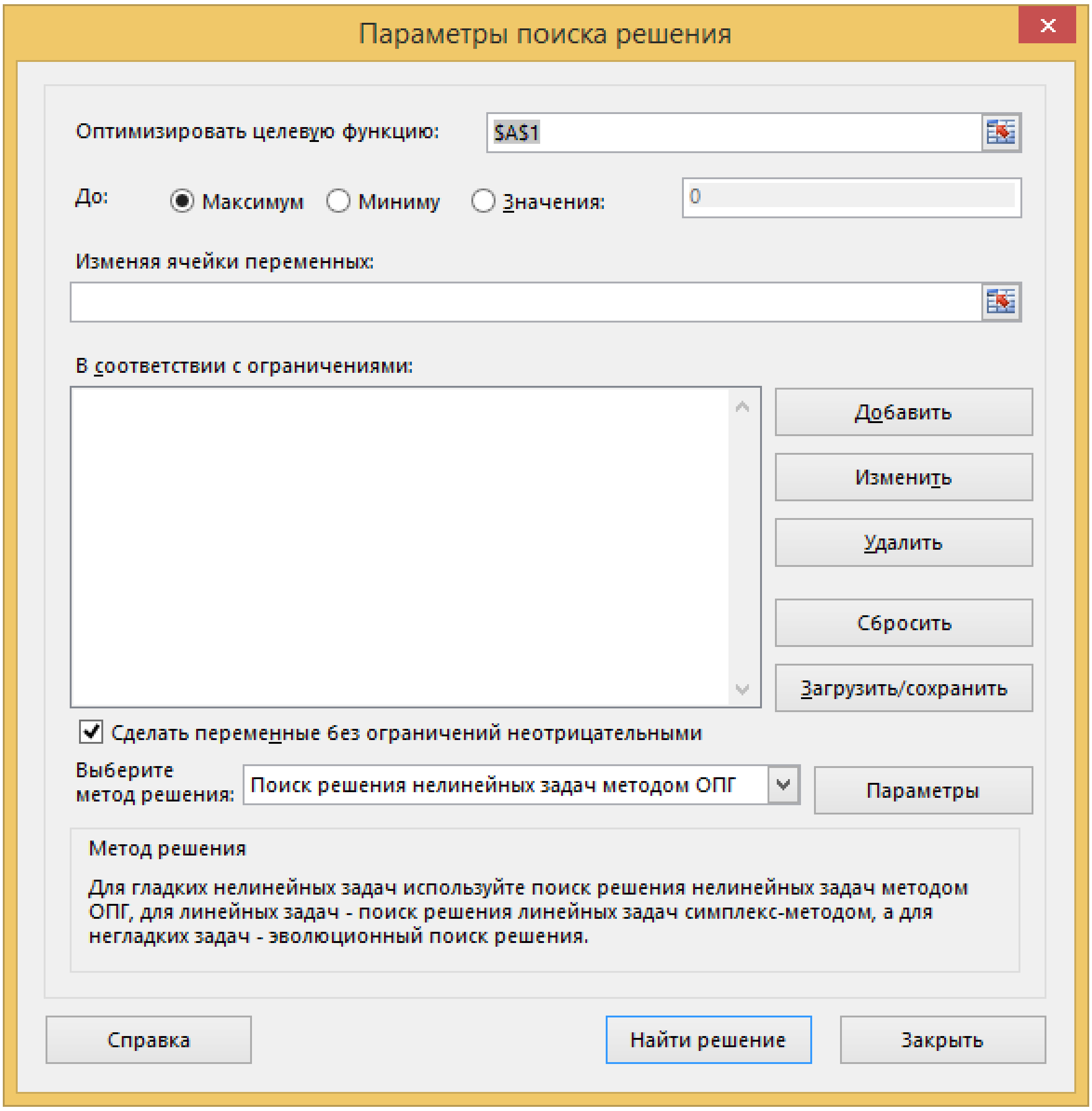


Рис.2.4 – Окно параметров

Рассмотрим несколько примеров решения оптимизационных задач с помощью данной надстройки.

Пример 1. Найти минимум функции $f(x) = (x-1)^2$.

Для решения этой задачи понадобится две ячейки. В одной из них (A2) будет рассчитано значение x, в другой (B2) содержится формула расчета целевой функции: =(A2-1)^2 (рис.2.5).

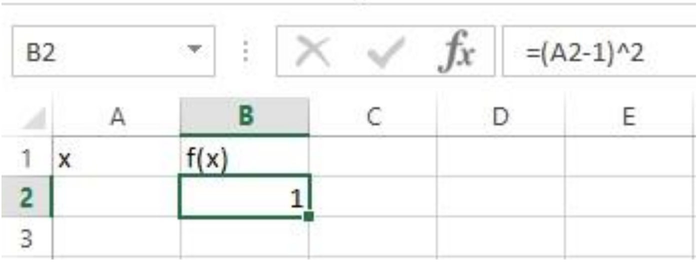


Рис. 2.5 – Заполнение ячейки

Далее вызываем «Поиск решения», в качестве целевой функции указываем ячейку B2, в качестве изменяемой ячейки – A2, и выбираем пункт «Минимуму» (рис.2.6).

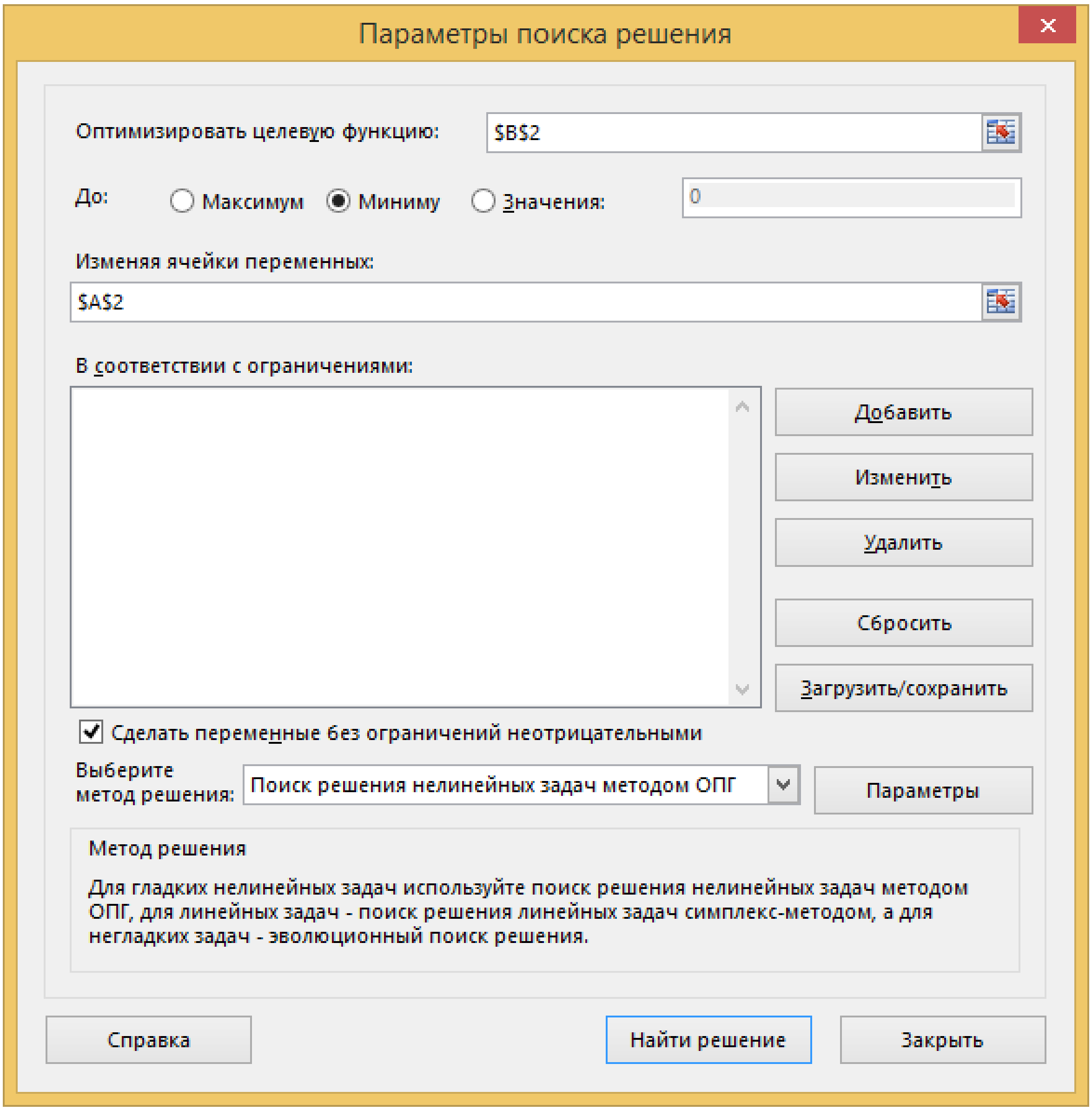


Рис.2.6 – Окно «Поиск решения»

Далее нажимаем кнопку «Найти решение». В ячейке появится найденное значение аргумента (рис.2.7).

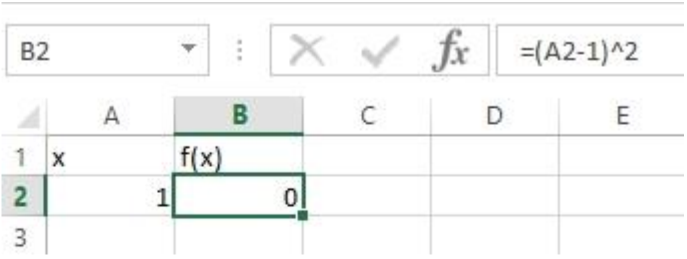


Рис.2.7 – Найденное решение

Пример 2. Найти минимум функции $f(x) = (x_1 - 1)^2 + 5(x_2 - 4)^2$.

Для решения этой задачи понадобится три ячейки. В одной из них (A2) будет рассчитано значение x1, в другой (B2) – x2, а в третьей содержится формула расчета целевой функции: =(A2-1)^2+5*(B2-4)^2 (рис.2.8).

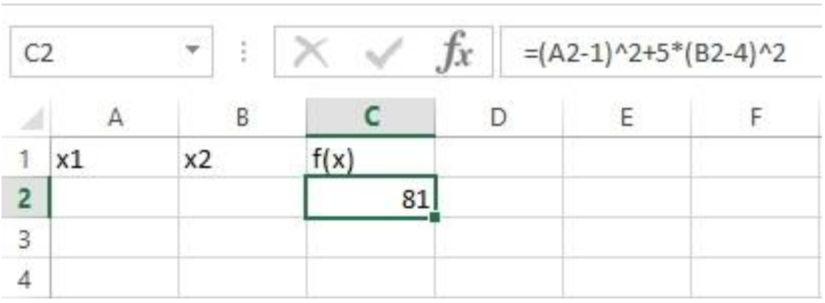


Рис.2.8 – Определение формулы расчета

Вызываем «Поиск решения», в качестве целевой функции указываем ячейку C2, в качестве изменяемых ячеек – A2:B2, и выбираем пункт «Минимуму» (рис.2.9).

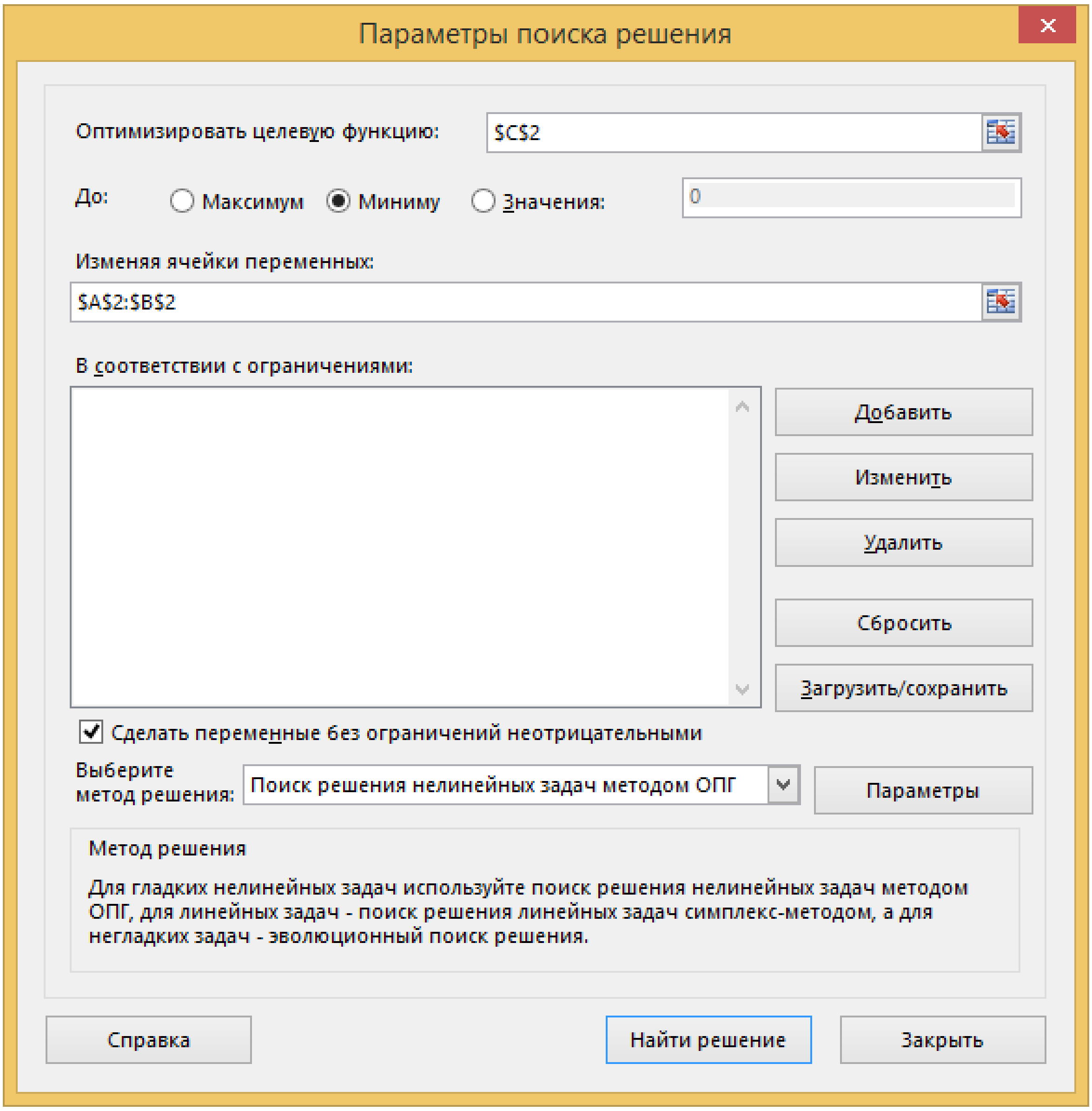


Рис.2.9 – Окно «Поиск решения»

Далее нажимаем кнопку «Найти решение». В ячейке появятся найденные значения аргументов (рис.2.10).

	A	B	C	D
1	x1	x2	f(x)	
2	0,999998	3,999998	3,07E-11	
3				

Рис.2.10 – Решение задачи

Пример 3. Решить задачу линейного программирования:

$$f(x) = 3x_1 + 5x_2 \to \min$$

$$x_1 + x_2 = 5$$

$$x \geq 0$$

Для решения этой задачи понадобится четыре ячейки. В одной из них (A2) будет рассчитано значение x1, в другой (B2) – x2, в третьей содержится формула расчета целевой функции: =3*A2+5*B2, а четвертой – формула ограничения: =A2+B2 (рис.2.11).

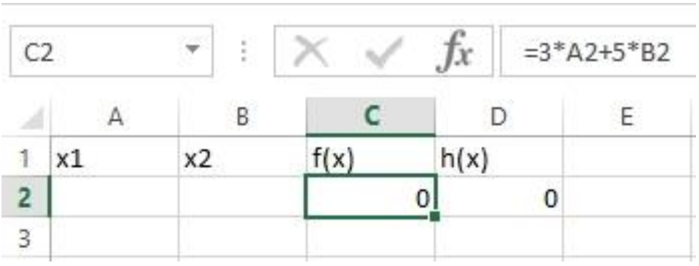


Рис.2.11 – Формула целевой функции

Вызываем «Поиск решения», в качестве целевой функции указываем ячейку C2, в качестве изменяемых ячеек – A2:B2, и выбираем пункт «Минимуму». Далее нажимаем кнопку «Добавить» для определения ограничения. В появившемся окне (рис.2.12) указываем ссылку на ячейку с ограничением и устанавливаем значение ограничения, равное 5 (рис.2.12).

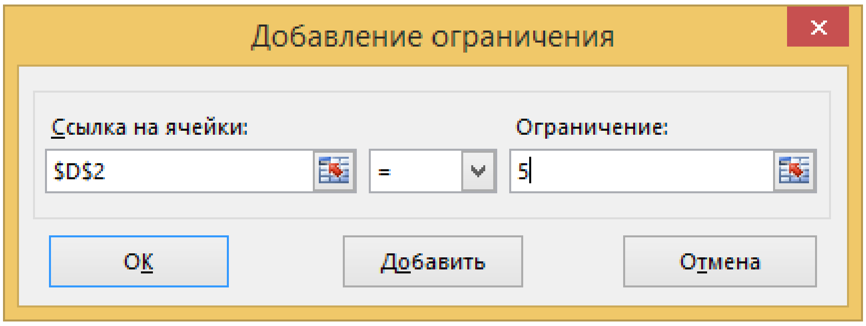


Рис.2.12 – Добавление ограничения

Созданное ограничение отобразится в окне «Параметры поиска решения» (рис.2.13). Далее нажимаем кнопку «Найти решение». В ячейке появятся найденные значения аргументов (рис.2.14).

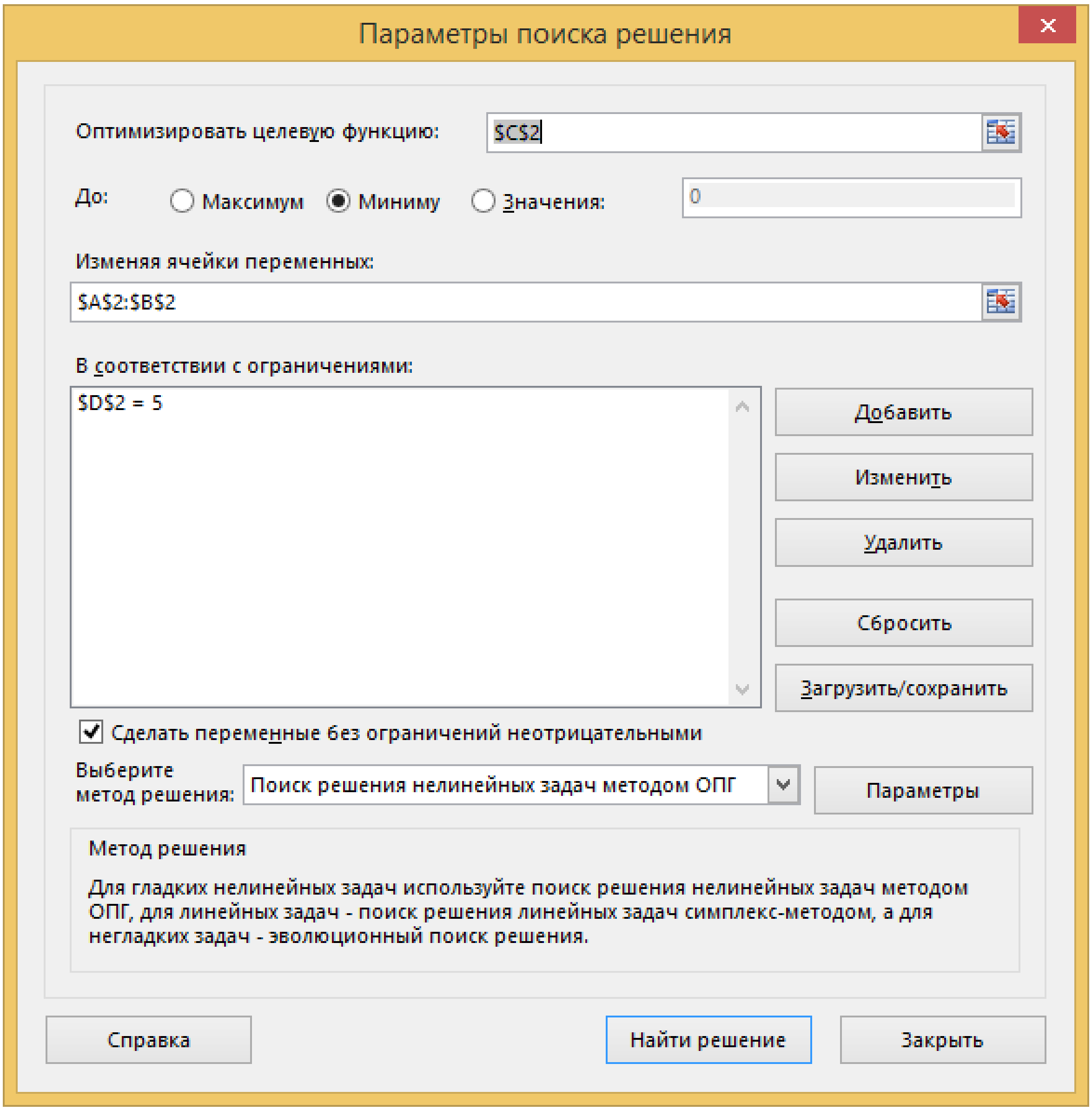
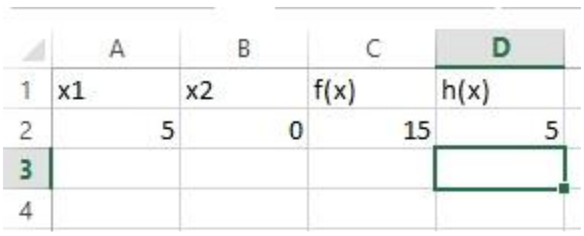


Рис.2.13 – Окно «Поиск решения»



	A	B	C	D
1	x1	x2	f(x)	h(x)
2	5	0	15	5
3				
4				

Рис.2.14 – Решение задачи

Пример 4. Решить задачу нелинейного программирования:

$$f(x) = 3x_1^2 + 5x_2^2 \to \min$$

$$x_1 + x_2 = 5$$

Для решения этой задачи понадобится четыре ячейки. В одной из них (A2) будет рассчитано значение x1, в другой (B2) – x2, в третьей содержится формула расчета целевой функции: =3*A2^2+5*B2^2, а четвертой – формула ограничения: =A2+B2 (рис.2.15).

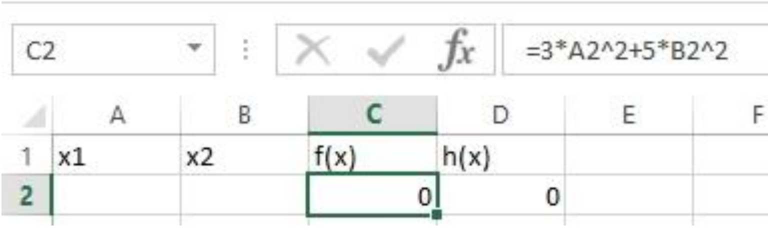


Рис.2.15 – Формула целевой функции

Вызываем «Поиск решения», в качестве целевой функции указываем ячейку C2, в качестве изменяемых ячеек – A2:B2, и выбираем пункт «Минимуму». Также добавляем ограничение способом, описанным в предыдущем примере. Созданное ограничение отобразится в окне «Параметры поиска решения» (рис.2.16). Далее нажимаем кнопку «Найти решение». В ячейке появятся найденные значения аргументов (рис.2.17).

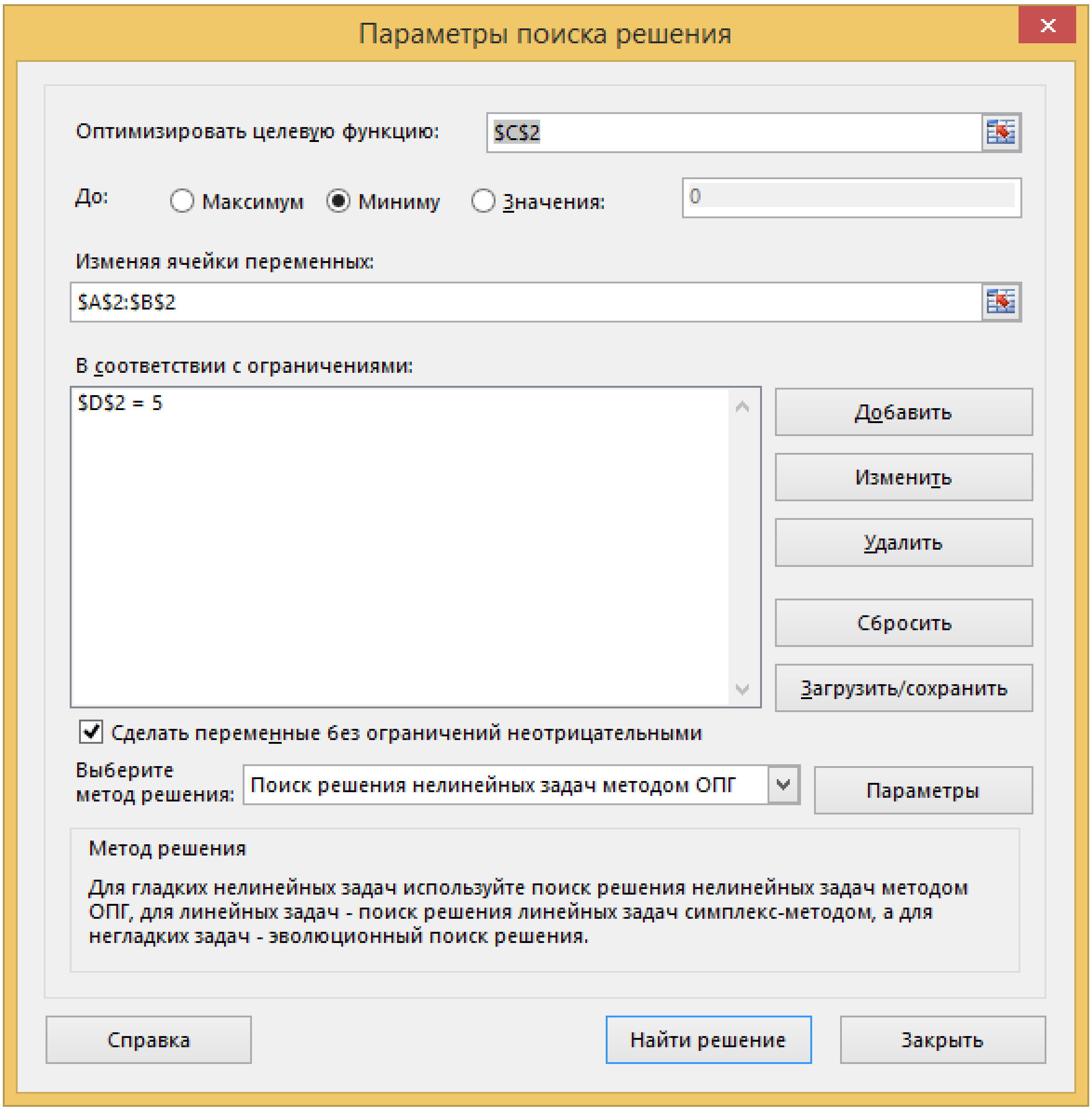


Рис.2.16 – Окно «Поиск решения»

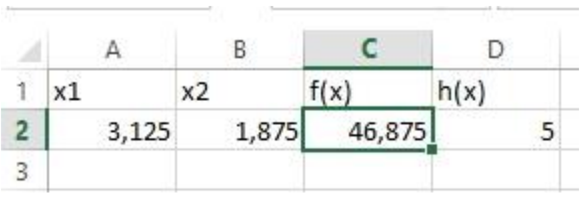


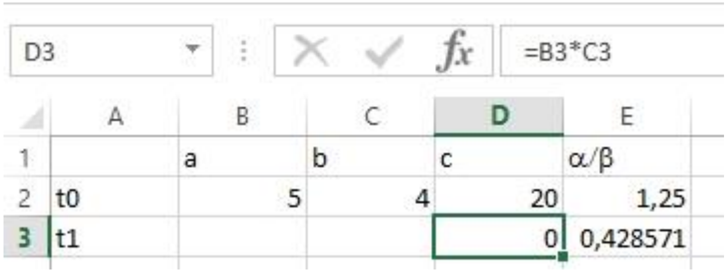
Рис.2.17 – Решение задачи

Пример 5. Решить обратную задачу:

$$c = a \cdot b$$

$$a_0 = 5, \quad b_0 = 4, \quad c_0 = 20, \quad c_1 = 30, \quad \alpha = 0,3, \quad \beta = 0,7$$

Для решения этой задачи понадобится восемь ячеек. В ячейки B2:D2 записываем начальные значения аргументов и функции. В ячейках B3:D3 будут определены новые значения (рис.2.18). В ячейке E2 запишем формулу отношения приростов: $\Delta a / \Delta b$, которое равно отношению коэффициентов относительной важности α / β: =(B3-B2)/(C3-C2). В ячейку E3 запишем отношение коэффициентов относительной важности: =0,3/0,7.


D3 =B3*C3

	A	B	C	D	E
1		a	b	c	α/β
2	t0	5	4	20	1,25
3	t1			0	0,428571

Рис.2.18 – Заполнение ячеек

Вызываем «Поиск решения», в качестве целевой функции указываем ячейку D3, в качестве изменяемых ячеек – B3:C3, и выбираем пункт «Минимуму». Также добавляем ограничение способом, описанным в предыдущем примере. Созданное ограничение отобразится в окне «Параметры поиска решения» (рис.2.19). Далее нажимаем кнопку «Найти решение». В ячейке появятся найденные значения аргументов (рис.2.20).

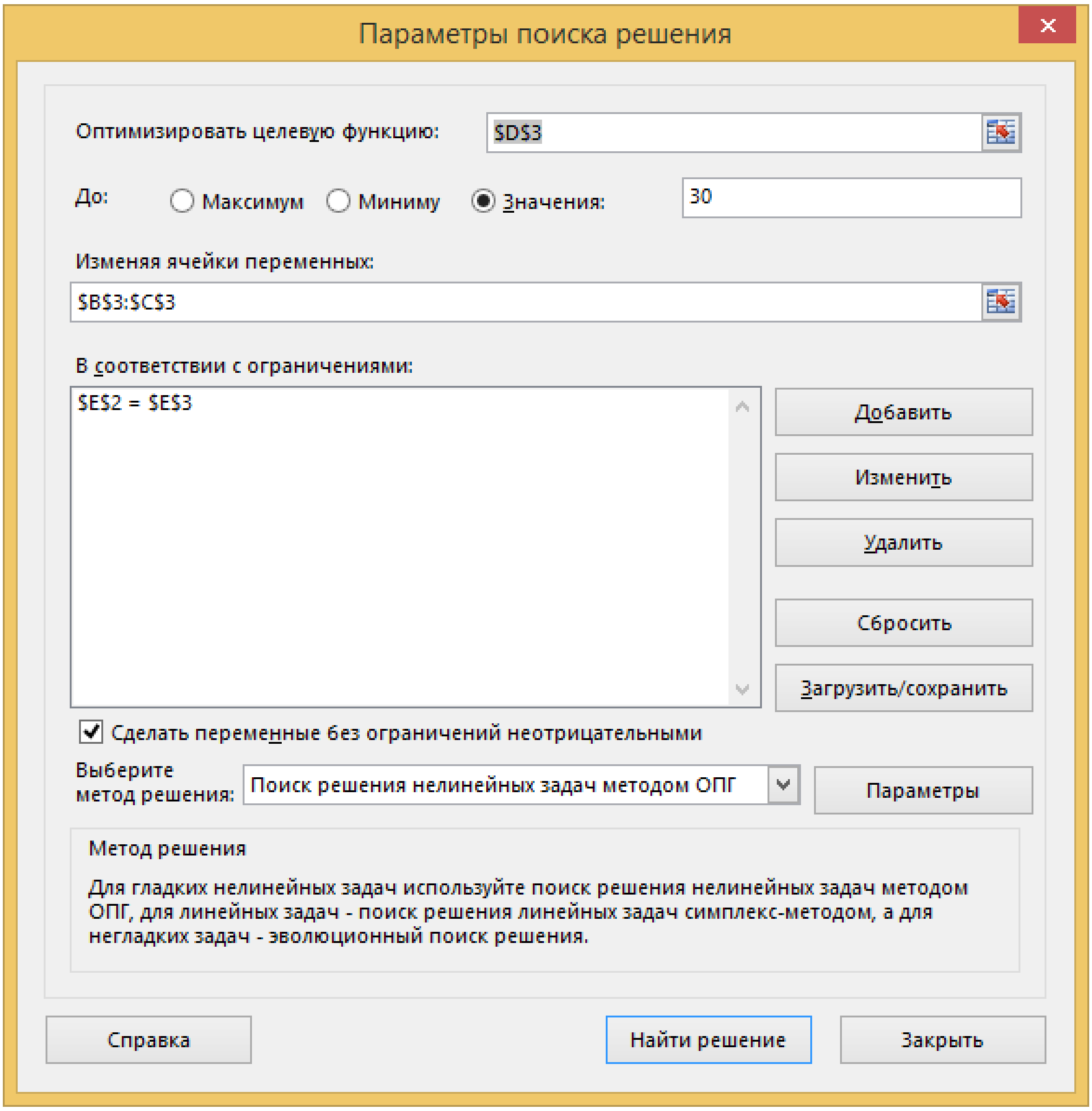


Рис.2.19 – Окно «Поиск решения»

	A	B	C	D	E
1		a	b	c	α/β
2	t0	5	4	20	0,428571
3	t1	5,586982	5,369624	29,99999	0,428571
4					

Рис.2.20 – Решение задачи

Приложение Ж. Решение оптимизационных задач в MathCAD.

Для решения оптимизационных задач в MathCad есть функция Minimize(y,x), которая осуществляет поиск значения x, соответствующего локальному минимуму функции y(x)**.**

Рассмотрим применение данной функции на примерах.

Пример 1. Найти минимум функции $f(x) = (x-1)^2$.

Решение задачи представлено на рис.3.1. Сначала определяем функцию, для которой нужно найти точку минимума. Далее задаем начальное значение аргумента (для того, чтобы присвоить значение нажимаем shift и знак «=», появится «:=»). После этого используем функцию Minimize, указав в качестве параметров обозначение функции и аргумента. Чтобы получить решение нажимаем знак «=» после функции. Решением задачи является точка $x = 1$.

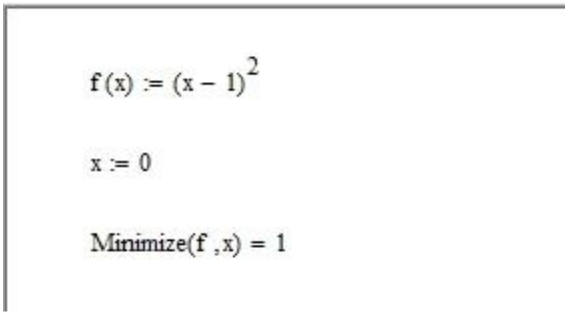


Рис.3.1 – Решение задачи одномерной оптимизации

Пример 2. Найти минимум функции $f(x) = (x_1 - 1)^2 + 5(x_2 - 4)^2$.

Решение задачи представлено на рис.3.2. Сначала укажем, что индексы будут начинается с 1, а не с 0 (по умолчанию). Для этого нужно написать ORIGIN:=1. Далее записываем функцию и начальное значение аргумента для решения задачи. После этого применяем функцию Minimize. Полученное решение: $x_1 = 1,\ \ x = 4$.

$$ORIGIN := 1$$

$$f(x) := (x_1 - 1)^2 + 5\cdot(x_2 - 4)^2$$

$$x_2 := 0$$

$$\mathrm{Minimize}(f, x) = \begin{pmatrix} 1 \\ 4 \end{pmatrix}$$

Рис.3.2 – Решение задачи многомерной оптимизации

Пример 3. Решить задачу линейного программирования:

$$f(x) = 3x_1 + 5x_2 \to \min$$
$$x_1 + x_2 = 5$$
$$x \geq 0$$

Решение задачи представлено на рис.3.3: $x_1 = 5$, $x_2 = 0$. Для того, чтобы указать ограничения нужно использовать блок «Given». После суммы аргументов нажимается Ctrl и знак “=” (появляется знак равенства жирным шрифтом).

$$f(x) := 3\cdot x_1 + 5\cdot x_2$$

$$x_2 := 0$$

Given

$$x_1 + x_2 \mathbf{=} 5$$

$$x \geq 0$$

$$\mathrm{Minimize}(f, x) = \begin{pmatrix} 5 \\ 0 \end{pmatrix}$$

Рис.3.3 – Решение задачи линейного программирования

Пример 4. Решить задачу нелинейного программирования:

$f(x) = 3x_1^2 + 5x_2^2 \to \min$

$x_1 + x_2 = 5$

$x \geq 0$

Решение задачи представлено на рис.3.4: $x_1 = 3{,}125$, $x_2 = 1{,}875$. Отличие от предыдущей задачи заключается только в формуле целевой функции.

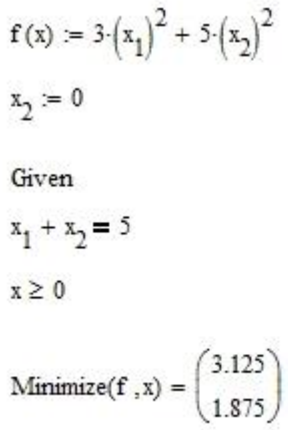


Рис.3.4 Решение задачи нелинейного программирования

Пример 5. Решить обратную задачу:

$c = a \cdot b$

$a_0 = 5,\ \ b_0 = 4,\ \ c_0 = 20,\ \ c_1 = 30,\ \ \alpha = 0{,}3,\ \ \beta = 0{,}7$

Решение задачи представлено на рис.3.5. Поскольку здесь необходимо найти новые значения аргументов, обеспечивающие заданное значение целевой функции (а не её минимум), то вместо функции Minimize используем функцию find, которая возвращает решение системы уравнения (в скобках указываются аргументы, которые необходимо найти). Все уравнения указываются после слова Given. В результате будут определены необходимые величины приростов.

Решение задачи:

$a_1 = 5{,}587$

$b_1 = 5{,}37$

$a0 := 5$

$b0 := 4$

$x := 1 \qquad y := 1$

Given

$(a0 + x)\cdot(b0 + y) = 30$

$$x = y\cdot\frac{0.3}{0.7}$$

$G := \mathrm{Find}(x, y)$

$$G = \begin{pmatrix} 0.587 \\ 1.37 \end{pmatrix}$$

$a0 + G_1 = 5.587$

$b0 + G_2 = 5.37$

Рис.3.5 – Решение обратной задачи