

фосфора являющимся стабильным р-элементом. Были получены соединения типа $M_3Ln(VO_4)_{2-x}$ где $0,1 < x < 1$. Были проведены температурно-частотные исследования. Как показали результаты исследований при $x = 0,8-0,9$ на зависимости $\lg\sigma = f(T^{-1})$ два участка: первый участок при температурах $>450K$ σ не зависит от температуры; второй высокотемпературный участок ток имеет энергию активации $\sim 0,9$ эВ. Также исследования на переменном сигнале показали, что в этих соединениях электропроводность не зависит от частоты при $x < 0,8$. Это показывает на то, что с увеличением доли в соединениях $M_3Ln(VO_4)_{2-x}(PO_4)$ все механизмы проводимости, которые наблюдались в $M_3Ln(VO_4)_2$ и $M_2MLn(VO_4)_2$ постепенно исчезают.

Литература

1. Dougier, P. Properties cristallographiques, magnetiques et electriques de l'orthovanatite de lanthane $LaVO_3$ / P. Dougier, P. Hagenmuller // I. Solid. State Chem. – 1974. – № 3. – P. 177–183.
2. Хайкс, Р. Механизм электропроводности в окислах переходных металлов с примесью лития / Р. Хайкс, У. Джонстон. – В кн.: Диэлектрическая спектроскопия. – М.: Мир, 1974.
3. Elliot, S. R. A theory of a.c. conduction in chalcogenide glasses / S. R. Elliot. – Phil. Mag. – 1977. – Vol. 36, № 6. – P. 129.
4. Электрофизические свойства и строение двойных ванадатов лантаноидов / Ш. Б. Ибрагимов // Узб. физ. журн. – 1993. – №1. – С. 79–81.

УДК 535.317

ПРИБЛИЖЕННОЕ РЕШЕНИЕ ЗАДАЧ НЕЛИНЕЙНОГО ПРОГРАММИРОВАНИЯ ГРАДИЕНТНЫМ МЕТОДОМ

Бокуть Л.В., Каменко Д.А.

*Белорусский национальный технический университет
Минск, Республика Беларусь*

Аннотация. В работе рассматривается приближенное решение задач нелинейного программирования градиентным методом. Для решения задачи безусловной оптимизации с несколькими переменными расчет шага осуществлялся методом дихотомии. Для реализации метода использован компьютерный пакет MatLab.

Ключевые слова: нелинейное программирование, градиентный метод, компьютерный пакет MatLab.

APPROXIMATE SOLUTION OF NONLINEAR PROGRAMMING PROBLEMS BY THE GRADIENT METHOD

Bokut L., Kamenko D.

*Belarusian National Technical University
Minsk, Republic of Belarus*

Abstract. In work the approximate solution of problems of nonlinear programming is considered by a gradient method. To solve the problem of unconditional optimization with several variables, the step calculation was carried out by the dichotomy method. For realization of a method the computer MatLab package is used.

Key words: nonlinear programming, gradient method, computer package MathLab.

*Адрес для переписки: Каменко Д.А., пр. Независимости, 65, Минск 220013, Республика Беларусь
e-mail: dimakamenko.2000@gmail.com*

В отечественной и зарубежной литературе подробно описаны градиентные и неградиентные методы оптимизации [1]. В статье [2] наглядно проиллюстрирован метод наискорейшего спуска с подробными доказательствами вспомогательных теорем. Существуют известные реализации алгоритма, например, на языке программирования Python [3, 4].

Целью данной работы является рассмотрение алгоритмов и их реализация в мощной компьютерной системе MatLab.

MatLab – одна из тщательно проработанных систем автоматизации математических и научно-технических расчетов. Она включает в себя:

– программные средства математических вычислений;

- специальные массивы;
- программные средства обработки данных;
- программные средства графики и др.

В основе градиентных методов лежит свойство градиента функции в точке. Он позволяет указать направление наибольшего роста функции в окрестности точки.

Необходимость решения задач нелинейного программирования связана с тем, что при исследовании реальных процессов не всегда можно использовать линейные функции для адекватной построенной математической модели.

Одной из разновидностей градиентного метода является метод наискорейшего спуска. Он предполагает выбор некоторой точки X_k и начального шага h_k .

Затем вычисляется градиент исходной функции $grad(X_k)$ и осуществляется переход к этому градиенту с выбранным шагом. Если значение функции F в последующей точке больше предыдущего значения, новая точка принимается за исходную точку, и повторяется предыдущий шаг.

Идея метода может быть проиллюстрирована примером спуска с горы, когда на каждом этапе спуска, будет выбираться направление самого крутого склона.

Пусть исходная задача формулируется следующим образом. Необходимо найти минимум функции:

$$y(x_1, x_2) = 21x_1^2 - 420x_1 \cdot x_2 + 21x_2^2 - 192x_1 + 50x_2 - 25.$$

Для вычисления значения вектора градиента находим частные производные:

$$\frac{\partial y}{\partial x_1} = 422x_1 - 420x_2 - 192.$$

$$\frac{\partial y}{\partial x_2} = 422x_2 - 420x_1 + 50.$$

Данная задача является задачей безусловной оптимизации с несколькими переменными. Для расчета шага выбран метод дихотомии. При использовании метода дихотомии считается, что исходная функция непрерывна и имеет на концах интервала разный знак. После вычисления значения функции в середине интервала одна часть интервала отбрасывается так, чтобы функция имела разный знак на концах оставшейся части.

Рассмотрим реализацию этого подхода в системе MatLab. Данная система оснащена графической составляющей, дополнена средствами компьютерной алгебры и библиотеками команд, предназначенных для эффективной работы со специальными классами задач.

Данная система позволяет представлять исходные данные в виде матриц и обрабатывать большие массивы данных.

Пусть исходная функция имеет вид:

$$\gg y = @(x) 211 * x(1).^2 - 420 * x(1) * x(2) + 211 * x(2).^2 - 192 * x(1) + 50 * x(2) - 25;$$

И заданы условия:

$$\gg \text{d}y_1 = @(x) 422 * x(1) - 420 * x(2) - 192;$$

$$\gg \text{d}y_2 = @(x) 422 * x(2) - 420 * x(1) + 50;$$

При задании функции использована операция поэлементного умножения. Она задана с помощью символов «*». А поэлементное возведение в степень с помощью символов «^».

Отсутствие «.» приводит к ошибке в выполнении программы, что связано с тем, что поэлементной операции требуются в компьютерном пакете MatLab дополнительные уточнения.

Зададим априорную погрешность алгоритма:

$$\gg E = 0.0001$$

Пусть параметр

$$\gg h = 1$$

Начальное значение вектора x :

$$\gg x = [0, 0];$$

$$\gg k = 0;$$

Организуем цикл:

```
>> while (sqrt(dyx1(x)^2 + dyx2(x)^2) > E)
f = @(h) y(x-h*[dyx1(x), dyx2(x)])
local_min = dichotomy(f, 0, 3, E);
h = local_min(1);
x = x - h*[dyx1(x), dyx2(x)];
k = k + 1;
end
```

Длина шага h вычислялась в этом случае следующим образом:

```
function [ret] = dichotomy(cb, a, b, eps)
d2 = eps/10;
n = 0;
x_min = h;
while (b-a) > eps
x_min = (b+a)/2;
fb = cb(x_min+d2);
fm = cb(x_min-d2);
if fb < fm
h = x_min - d2;
else
b = x_min + d2;
end
n = n + 1;
end
ret = [x_min, n];
end
```

Результат работы алгоритма может быть представлен в виде графика

```
[x1, x2] = meshgrid(-200:1:200);
z = 211 * x1.^2 - 420 * x1 .* x2 + 211 * x2.^2 - 192 * x1 + 50 * x2 - 25;
surf(x1, x2, z)
shading interp;
text(x(1), x(2), 'Минимум', 'color', 'r');
k
x
y(x)
```

Функция $[X, Y] = \text{meshgrid}(x, y)$ формирует массивы X и Y , которые определяют координаты узлов прямоугольника, задаваемого векторами x и y .

Функция $\text{surf}(X, Y, Z)$ создает 3D объемную поверхностную диаграмму, которая является 3D поверхностью с твердыми цветами обводки и точными цветами поверхности.

Описанный алгоритм позволяет находить минимальное значение сходимости многомерной функции в определенной точке.

На рис. 1 представлена поверхность решения.

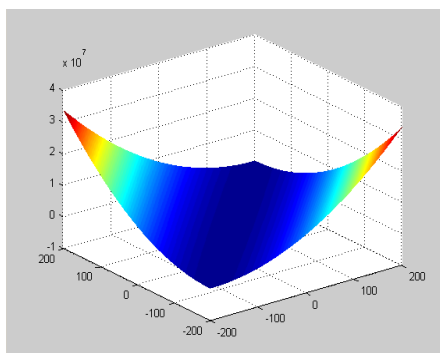


Рисунок 1 – Поверхность решения

Таким образом, входными параметрами описанного алгоритма являлись целевая функция,

начальная точка и погрешность, позволяющая завершить цикл.

Отметим, что скорость сходимости метода снижается при приближении к минимальному значению.

Литература

1. Iusem, A. N. A proximal regularization of the steepest descent method / A. N. Iusem, B. F. Svaiter. – 1995.
2. Ryan Tibshirani. Convex Optimization: Lecture 5: Gradient Descent [Электронный ресурс]. – Режим доступа: <https://www.stat.cmu.edu>. – Дата доступа : 29.09.2022.
3. Linge, S. Programming for Computations Python: A Gentle Introduction to Numerical Simulations with Python / S. Linge, Hans Petter Langtangen. – 2016.
4. Guttag, J. V / Introduction to Computation and Programming using Python / J. V. Guttag, J. Sussman. – 2016.

УДК 535.317

АНАЛИЗ МЕТОДОВ ФОРМИРОВАНИЯ ОПОРНОГО ПЛАНА ТРАНСПОРТНОЙ ЗАДАЧИ

Бокуть Л.В., Климович Т.А.

*Белорусский национальный технический университет
Минск, Республика Беларусь*

Аннотация. Работа посвящена проблеме формирования опорного плана транспортной задачи. Представлена сравнительная характеристика методов ее решения. Проведен также анализ формирования оптимального опорного плана различными методами.

Ключевые слова: транспортная задача, опорный план, оптимальный план, граф.

ANALYSIS OF METHODS FOR FORMING A REFERENCE PLAN FOR A TRANSPORT TASK

Bokut L., Klimovich T.

*Belarusian National Technical University
Minsk, Republic of Belarus*

Abstract. The work is devoted to a problem of formation of the basic plan of a transport task. Comparative characteristic of methods of its decision is presented. Also the analysis of formation of the optimum basic plan is carried out by various methods.

Key words: transport task, basic plan, optimum plan, graph.

*Адрес для переписки: Климович Т.А., пр. Независимости, 65, Минск 220113, Республика Беларусь
e-mail: tanya.tatina.klimovich@mail.ru*

Одной из главных задач математического моделирования транспортных перевозок является построение оптимального плана перевозок. Для решения транспортных задач можно применить симплексный метод. Так как матрица системы ограничений транспортной задачи очень своеобразна, то для нахождения ее решения применяют специфические методы. Они, как и симплексный метод, позволяют находить начальное опорное решение, а затем, улучшая его, получить оптимальное решение. В зависимости от способа постановки условий данной задачи, транспортная задача может быть представлена в сетевой и матричной (табличной) форме. Математическая модель сбалансированной транспортной задачи имеет следующий вид:

$$Z(x) = \sum_{i=1}^m \sum_{j=1}^n C_{ij} x_{ij} \rightarrow \min;$$

$$\sum_{j=1}^n x_{i,j} = a_i, \quad i \in \overline{1, m};$$

$$\sum_{i=1}^m x_{i,j} = b_j, \quad j \in \overline{1, n};$$

$$x_{i,j} \geq 0, \quad i \in \overline{1, m}; j \in \overline{1, n}.$$

Транспортная задача является канонической задачей линейного программирования с переменными x_{ij} и m n ограничениями. Транспортная задача может быть представлена в сетевой (графовой) форме [1]. На примере одной и той же задачи