АВТОМАТИЗАЦИЯ ОПТИМИЗАЦИИ НЕЛИНЕЙНЫХ ФУНКЦИЙ МЕТОДОМ ГРАДИЕНТНОГО СПУСКА С ПЕРЕМЕННЫМ ШАГОМ

Урусова В. С., Сташкевич И. И., Бобух А. Н.

Рассмотрены области практического применения методов оптимизации. Подробно рассмотрен принцип градиентного спуска. Реализован алгоритм метода градиентного спуска с переменным шагом. Разработано приложение для нахождения экстремума функции нескольких переменных данным методом, с возможностью выполнения как безусловной, так и условной оптимизации. В приложении реализована подробная визуализация процесса решения, что сделано с целью облегчения понимания принципа работы метода. Намечены перспективные направления исследований, в частности: реализация дополнительных методов оптимизации; возможность выбора иной целевой функции; графическое представление процесса решения на плоскости.

Розглянуто області практичного застосування методів оптимізації. Докладно розглянутий принцип градієнтного спуску. Реалізовано алгоритм методу градієнтного спуску зі змінним кроком. Розроблено додаток для знаходження екстремуму функції декількох змінних даним методом, з можливістю виконання як безумовної, так й умовної оптимізації. У додатку реалізована докладна візуалізація процесу рішення, що зроблено з метою полегшити розуміння принципу роботи методу. Намічено перспективні напрями досліджень, зокрема: реалізація додаткових методів оптимізації; можливість вибору іншої цільової функції; графічне представлення процесу рішення на площості.

Areas of practical application of optimization methods are considered. The principle of graded-index descent is explicitly considered. The algorithm of the method of graded-index descent with variable step is implemented. The application for finding of an extremum of function of several variables by the given method, with possibility of performance both unconditional, and the conditional optimization is developed. In the application of the detailed visualization of process of the decision to facilitate understanding of a principle of operation of a method is implemented. Perspective directions of researches are set, in particular: the realization of additional methods of optimization; the possibility of choice of other objective function; graphic presentation of process of the decision on a plane.

Урусова В. С. — ст. преп. кафедры Инженерной графики ДГМА

Сташкевич И. И. — ассистент кафедры ИСПР ДГМА
stashkevich_dgma@ukr.net

Бобух А. Н. — студент ДГМА

ДГМА – Донбасская государственная машиностроительная академия, г. Краматорск

УДК 519.6

Урусова В. С., Сташкевич И. И., Бобух А. Н.

АВТОМАТИЗАЦИЯ ОПТИМИЗАЦИИ НЕЛИНЕЙНЫХ ФУНКЦИЙ МЕТОДОМ ГРАДИЕНТНОГО СПУСКА С ПЕРЕМЕННЫМ ШАГОМ

Под термином «методы оптимизации» понимают методы нахождения экстремумов функций нескольких переменных на множествах, которые задаются линейными или нелинейными равенствами и неравенствами.

Методы оптимизации являются одним из наиболее развивающихся разделов прикладной математики в силу большого многообразия и важности их приложений [1]. В качестве примеров можно указать следующие области применения методов оптимизации:

– технико-экономические системы: планирование, управление, транспортные задачи, распределение кадров, ресурсов;

– численный анализ: аппроксимация, регрессия, решение систем уравнений, численные методы решения задач математической физики;

– технические системы: распознавание образов, оптимальное управление, роботы, управление производством, проектирование систем связи;

– математическая экономика: анализ больших макроэкономических моделей, микроэкономических моделей или моделей предпринимательства, теория принятия решений, теория игр.

Как видим, сфера применения методов оптимизации чрезвычайно широка. Поэтому знание этих методов и умение их применять является необходимым навыком для выпускаемых техническими ВУЗами специалистов.

Изучению методов оптимизации в ВУЗах посвящен ряд учебных дисциплин, в рамках которых рассматриваются различные методы и их модификации [2, 3]. Недостатком является тот факт, что далеко не всегда излагаемые теоретические сведения сопровождаются конкретными практическими примерами. Это может существенно усложнить понимание материала, особенно если речь идёт о сложном алгоритме. Таким образом, имеется необходимость демонстрации работы метода на конкретных примерах с возможностью задания начальных условий, чтобы можно было изучить не только заведомо рабочие примеры, но и содержащие ошибки.

Целью данной работы является разработка учебного приложения, способного решать задачу оптимизации функции двух переменных выбранным методом и обеспечивающего подробный аналитический вывод каждого шага решения.

В качестве примера рассмотрим метод градиентного спуска. Семейство градиентных методов занимает ведущее место среди прямых методов оптимизации [2]. Идея данного метода основана на том, что градиент функции указывает направление ее наиболее быстрого возрастания в окрестности той точки, в которой он вычислен. Поэтому, если из некоторой текущей точки $x^{(1)}$ перемещаться в направлении вектора $\nabla \mathrm{f}(x^{(1)})$, то функция f будет возрастать, по крайней мере, в некоторой окрестности $x^{(1)}$. Следовательно, для точки $x^{(2)} = x^{(1)} + \lambda * \nabla \mathrm{f}(x^{(1)})$, $(\lambda > 0)$ лежащей в такой окрестности, справедливо неравенство $\mathrm{f}(x^{(1)}) \le \mathrm{f}(x^{(2)})$.

Продолжая этот процесс, мы постепенно будем приближаться к точке некоторого локального максимума (см. рис. 1).

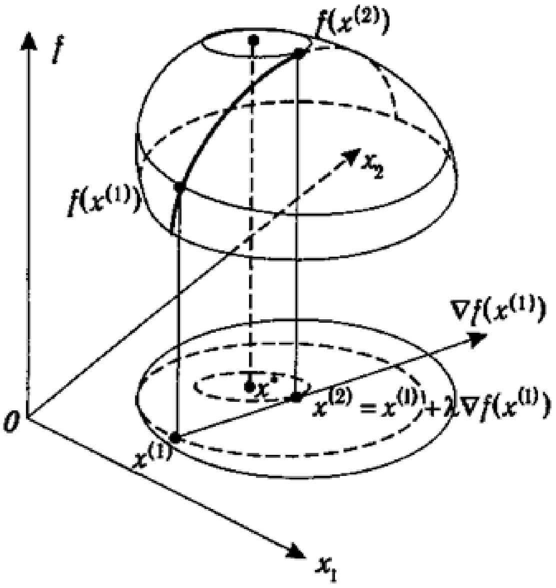

Рис. 1. Принцип градиентного спуска

Однако как только определяется направление движения, сразу же встает вопрос о том, как далеко следует двигаться в этом направлении или, другими словами, возникает проблема выбора шага λ в рекуррентной формуле:

$$x^{(q+1)} = x^{(q)} + \lambda * \nabla \mathrm{f}(x^{(q)}), \qquad (1)$$

задающей последовательность точек, стремящихся к точке максимума.

Существует ряд способов решения этой проблемы. В данной работе использован наиболее простой из них – метод дробления шага.

Выразим зависимость нового значения функции $\nabla \mathrm{f}(x^{(q+1)})$ от величины шага λ:

$$\mathrm{f}(x^{(q+1)}) = \mathrm{f}(x^{(q)} + \lambda * \nabla \mathrm{f}(x^{(q)})) = \varphi(\lambda). \qquad (2)$$

Тогда, согласно формуле (1), имеем ц$(0) = \mathrm{f}(x^{(q)})$. Суть метода дробления шага заключается в «подборе» такого значения λ, при котором выполняется условие ц$(\lambda) \geq$ ц(0). Задаются некоторым начальным значением λ_1 (например, $\lambda_1 = 1$) и проверяют условие ц$(\lambda_1) \geq$ ц(0). Если оно не выполняется, значит, шаг оказался слишком большим и метод «перешагнул» через точку экстремума. Тогда полагают $\lambda_2 = \frac{1}{2}\lambda_1$, снова проверяют условие, и т. д. до тех пор, пока не удается найти подходящий шаг, с которым переходят к следующей точке $x^{(q+1)}$.

После того как точка $x^{(q+1)}$ найдена, она становится текущей для очередной итерации. На практике признаком достижения точки экстремума служит достаточно малое изменение координат точек, рассматриваемых на последовательных итерациях. Одновременно с этим координаты вектора $\nabla \mathrm{f}(x^{(q)})$ должны быть близки к нулю.

Градиентные методы относятся к методам безусловной оптимизации, т. е. направлены на отыскание глобального оптимума. Если же необходимо найти некоторый локальный экстремум, нужно использовать дополнительные способы определения длины шага для контроля выхода шага за пределы области ограничений [3].

При решении задачи условной оптимизации решение может находиться либо внутри заданной области, либо на ее границе. В первом случае процесс решения ничем не будет отличаться от задачи безусловной оптимизации. Во втором случае потребуется корректировка найденных значений, о чём речь пойдёт ниже.

Следует отметить, что, говоря о задачах условной оптимизации в контексте базовой программы ВУЗа, будем рассматривать лишь наиболее простую их разновидность – с линейной системой ограничений. Для функции двух переменных это означает, что область решения ограничена отрезками прямых. Такая система ограничений имеет вид:

$$\sum_{j=1}^{n} a_{ij} x_j^{(k)} \le b_i, i = 1,...,m, \tag{3}$$

где n – число переменных в целевой функции; m – число ограничений в задаче; k – номер шага.

Допустим, в процессе решения задачи было определено некоторое направление шага $l = (l_1,...,l_n)$, и этот шаг привёл в точку $x^{(q+1)}$. Путём подстановки координат этой точки в (3), было установлено, что одно из неравенств не выполняется. Значит, нужно взять другую точку – лежащую на том же направлении, но не выходящую за пределы области решений, т.е. лежащую на ее границе. Следовательно, нужно уменьшить длину шага λ таким образом, чтобы выполнялось равенство:

$$\sum_{j=1}^{n} a_{ij} x_j^{(k+1)} = b_i, \tag{4}$$

где i – номер нарушенного ограничения.

Учитывая формулу (1), получаем:

$$\sum_{j=1}^{n} a_{ij} (x_j^{(k)} + \lambda * l_j) = b_i; \quad \sum_{j=1}^{n} a_{ij} x_j^{(k)} - b_i + \lambda * \sum_{j=1}^{n} a_{ij} l_j = 0.$$

Отсюда:

$$\lambda = -\frac{\sum_{j=1}^{n} a_{ij} x_j^{(k)} - b_i}{\sum_{j=1}^{n} a_{ij} l_j}. \tag{5}$$

Формула (5) позволяет вычислить длину шага, при которой новая точка попадёт на границу области ограничений. В случае, когда координаты точки $x^{(q+1)}$ не удовлетворяют нескольким ограничениям, следует вычислить λ по формуле (5) для каждого из этих ограничений, и взять наименьшее полученное значение. Подставляя его в (1), получаем исходную точку для следующего шага решения.

При этом следует понимать, что в найденной точке на границе области значение функции может не удовлетворять условию возрастания / убывания функции (в зависимости от того, решается ли задача на максимум или на минимум). В таком случае, следует применять основной метод контроля длины шага, например, метод дробления шага, о котором речь шла ранее. В качестве отправной величины шага следует взять значение λ, вычисленное по формуле (5).

Если же на границе области функция удовлетворяет условию возрастания / убывания, то в новой точке вычисляется градиент, и далее решение может продолжиться по одному из двух путей.

Если градиент укажет внутрь области, то, скорее всего, выход за границу был случайным, обусловленным слишком большой длиной шага, а решение всё же лежит внутри области. Тогда далее решение продолжится по обычному алгоритму градиентного метода.

Если же градиент вновь указывает за пределы области ограничений, то очевидно, что решение задачи также лежит вне области. В таком случае, все последующие точки будут определяться как проекции точек, полученных обычным методом градиентного спуска, на нарушенную границу. Поскольку в случае линейных ограничений проекция градиента будет лежать непосредственно на границе области, то и дальнейшее направление поиска берется на этой границе. Так, для функции двух переменных проекция градиента лежит на прямой $a_{i1}x_1 - a_{i2}x_2 = b_i$, и направление поиска будет определяться вектором этой прямой. Оптимум в таком случае достигается в той точке, в которой градиент перпендикулярен границе области.

С учётом всего вышесказанного, был разработано приложение, реализующее процесс поиска минимума функции методом градиентного спуска с переменным шагом. В программе рассматривается функция вида:

$$F(x, y) = (a * x + p) * x + b * x * y + (c * y + q) * y + r, \qquad (6)$$

где *a, b, c, p, q, r* – действительные числа.

В задаче может присутствовать произвольное число линейных ограничений. Также пользователь может задавать коэффициенты целевой функции, начальную точку поиска, начальную величину шага и желаемую точность вычислений.

В качестве среды реализации была выбрана среда Borland Dephi [4, 5].

На рис. 2 представлено окно разработанной программы с решенной задачей условной оптимизации.

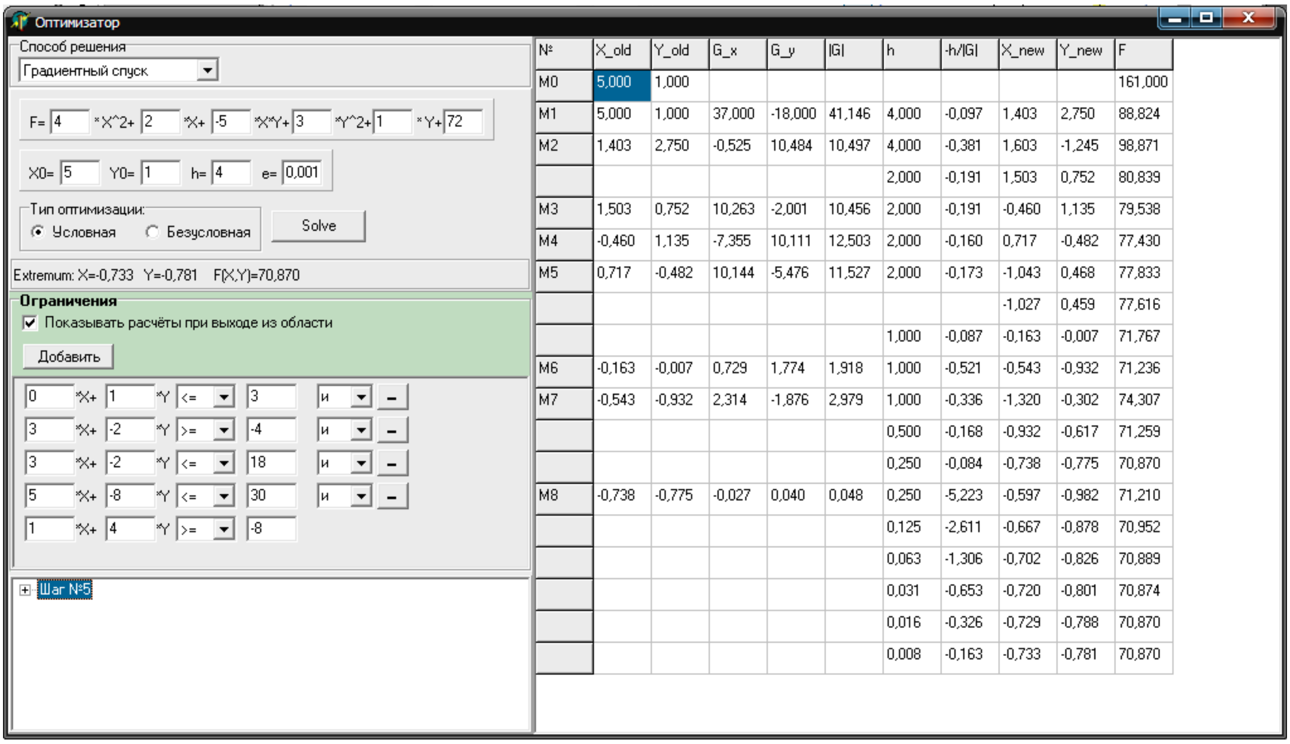

Рис. 2. Демонстрация работы приложения

В правой части окна находится таблица, где подробно расписан каждый шаг решения, с указанием результатов расчёта всех промежуточных значений, используемых для определения координат новой точки. Также в таблице видно, на каком шаге решения производилась коррекция длины шага, и на каком шаге решение выходило за пределы области ограничений.

Дополнительная информация о шагах, выходящих за пределы области, представлена на специальной информационной панели в левой нижней части окна приложения (см. рис. 3). В ней указаны: исходная и конечная точки шага; коэффициенты уравнения прямой, проходящей через эти точки; коэффициенты уравнения прямой, содержащей нарушенную границу области; точка пересечения этих прямых, являющаяся искомой точкой, лежащей на границе области.

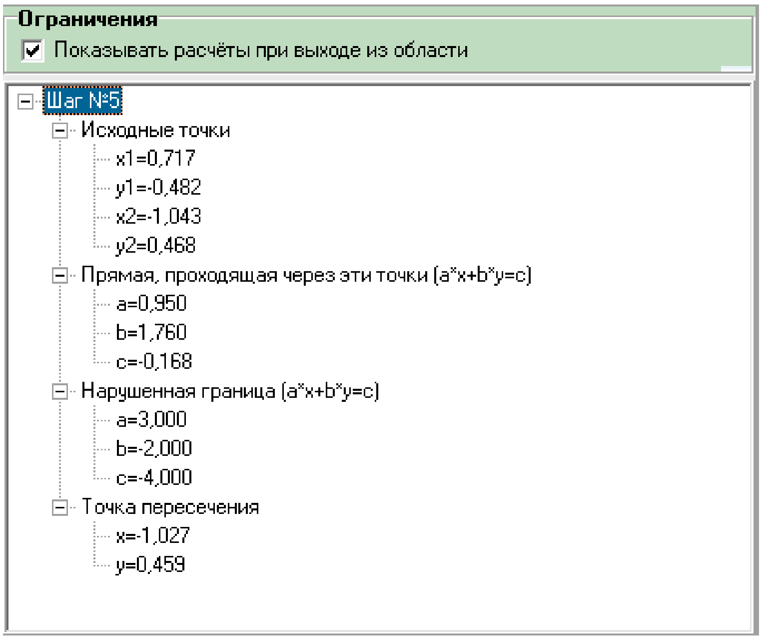

Рис. 3. Информация о нарушении границы области поиска решения

Эти сведения помогут получить более полное представление о ходе решения задачи, а также проконтролировать правильность расчётов при решении задачи вручную, если возникнет такая необходимость.

ВЫВОДЫ

Разработано приложение, осуществляющее полноценное аналитическое решение задачи оптимизации. Возможность ввода произвольных условий задачи даёт широкие возможности для исследования особенностей функционирования алгоритма в зависимости от тех или иных параметров задачи. Детальное отображение каждого шага расчётов позволяет получить подробную информацию о ходе решения и лучше уяснить принципы работы метода градиентного спуска.

Планируется дальнейшее развитие приложения и добавление новых возможностей, в частности: реализация дополнительных методов оптимизации; возможность выбора иной целевой функции; графическое представление процесса решения на плоскости.

ЛИТЕРАТУРА

1. *Минц М. Математическое программирование. Теория и алгоритмы / М. Минц. – М. : Наука, 1990. – 180 с.*
2. *Конюховский П. В. Математические методы исследования операций в экономике / П. В. Конюховский. – СПб: Питер, 2000. – 208 с.*
3. *Кремер Н. Ш. Математическое исследование операций / Н. Ш. Кремер. – Москва, Высшая школа, 2000. – 250 с.*
4. *Кэнту М. Delphi 5 для профессионалов / М. Кэнту. – СПб. : Питер, 2001. – 944 с.*
5. *Фаронов В. В. Delphi 5. Руководство программиста / В. В. Фаронов. – Москва : Нолидж, 2001. – 880 с.*