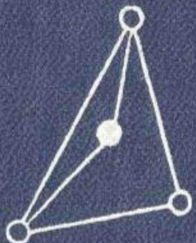
ВЫЧИСЛИТЕЛЬНЫЕ МЕТОДЫ ВЫБОРА ОПТИМАЛЬНЫХ ПРОЕКТНЫХ РЕШЕНИЙ



УДК 681.3:51

Вычислительные методы выбора оптимальных проектных решений. Михалевич В. С., Шор Н. З., Галустова Л. А., Журбенко Н. Г., Момот А. И., Сибирко А. Н., Трубин В. А., Юн Г. Н. К., «Наук. думка», 1977. 178 с.

В монографии рассмотрены новые эффективные вычислительные методы оптимизации и их использование при решении сложных задач выбора проектных решений. Изложены вопросы формализованного описания задач оптимального проектирования, методы последовательной оптимизации для объектов с протяженной или ветвящейся структурой и дискретными параметрами, методы градиентного типа с ускоренной сходимостью для оптимизации объектов с непрерывными параметрами, а также специальные методы оптимального проектирования комбинаторного характера. Подробное изложение современных математических методов оптимизации сочетается с большим числом примеров решения конкретных задач проектирования.

Книга предназначена для специалистов в области прикладной математики, а также для инженеров-проектировщиков, использующих математические модели и методы оптимизации.

Ил. 21. Табл. 4. Список лит.: с. 172—176 (108 назв.).

Авторы

В. С. Михалевич, Н. З. Шор, Л. А. Галустова, Н. Г. Журбенко, А. И. Момот, А. Н. Сибирко, В. А. Трубин, Г. Н. Юн

Под общей редакцией *В. С. Михалевича*

Рецензенты

В. Н. Кальченко, Г. Г. Гребенкин, Ю. М. Ермольев

Редакция физико-математической литературы

В $\frac{30501-249}{М221(04)-77}$ 155—77

ОГЛАВЛЕНИЕ

ПРИЛОЖЕНИЕ

Рассмотрим алгоритм и программу на языке ФОРТРАН метода градиентного типа с растяжением пространства в направлении разности двух последовательных градиентов (r-алгоритма).

Алгоритм

Пусть $f(x)$ — выпуклая функция, определенная в n-мерном эвклидовом пространстве E_n; $g(x)$ — обобщенный градиент функции $f(x)$ в точке x.

Первый шаг алгоритма задается начальным значением $x_0 \in E_n$; вычисляем $g(x_0)$, выбираем $h_1 > 0$ и находим

$$x_1 = x_0 - h_1 g(x_0).$$

Принимаем $\tilde{g}_1 = g(x_0)$, $B = E$ (единичная матрица). Переходим ко второму шагу. Пусть в результате вычислений после k шагов процесса ($k = 1, 2, \ldots$) получены определенные значения векторов x_k, $\tilde{g}_k \in E_n$ и матрицы B_k. Опишем $(k + 1)$-й шаг процесса.

Вычисляем следующие величины:

1. $g(x_k)$ — градиент функции $f(x)$ в точке x_k.

2. $g_k^* = B_k^* g(x_k)$.

3. $r_k = g_k^* - \tilde{g}_k$.

4. $\eta_{k+1} = r_k / \| r_k \|$.

5. β_{k+1} — величина, обратная α_{k+1} — коэффициенту растяжения пространства на $(k + 1)$-м шаге.

6. $B_{k+1} = B_k R_{\beta_{k+1}}(\eta_{k+1})$, где $R_{\beta_{k+1}}(\eta_{k+1})$

— оператор растяжения пространства в направлении η_{k+1} с коэффициентом β_{k+1} (матрица оператора $R_\alpha(\eta)$ имеет вид $r_{ij} = (\alpha - 1)\eta_i\eta_j + \delta_{ij}$, где δ_{ij} — символ Кронекера).

7. h_{k+1} — шаговый множитель в формуле спуска в заданном направлении.

8. $\tilde{g}_{k+1} = R_{\beta_{k+1}}(\eta_{k+1})\, g_k^*$.

9. $x_{k+1} = x_k - h_{k+1} B_{k+1} \tilde{g}_{k+1}$.

10. Переход к $(k+2)$-му шагу с запоминанием x_{k+1}, $\tilde{g}_{k+1}$ и B_{k+1} или окончание работы алгоритма при выполнении некоторого критерия остановки.

Приведенное описание класса алгоритмов порождает конкретные алгоритмы при уточнении способа выбора последовательности $\{h_k\}$, β_k и критерия остановки.

Хотя формальное описание r-алгоритма приведено для случая выпуклой функции, однако, как всякий алгоритм градиентного типа, он может применяться и для минимизации невыпуклых функционалов. Но следует заметить что в случае многоэкстремальной задачи r-алгоритм обеспечивает сходимость последовательности лишь к некоторому локальному минимуму функции $f(x)$.

r-Алгоритм в сочетании с идеей штрафных функций может быть использован для решения общей задачи математического программирования. Причем существенной особенностью изложенного алгоритма является то, что он допускает использование негладких штрафных функций. Например, пусть задача состоит в следующем:

найти

$$\min f(x), \quad i = 1, \ldots, m.$$
$$g_i(x) \leqslant 0 \tag{1}$$

Пусть λ_i — множители Лагранжа задачи (1).

Нетрудно показать, что задача (1) эквивалентна задаче безусловной минимизации $\min\left[f(x) + \sum_{i=1}^{m} M_i \max\{0, g_i(x)\}\right]$, где $M_i \gg \lambda_i$, для решения которой можно применить r-алгоритм. Изложенный алгоритм реализуется подпрограммой MINIM, написанной на языке ФОРТРАН.

Обращение к подпрограмме MINIM имеет вид

CALL MINIM (N, G, э, KR, DELTA,

ALPHA, k, INT, F, x, S, B, $G1$, $G2$, z).

Формальные параметры подпрограммы имеют следующий смысл: N — размерность пространства переменных; G — вектор градиента, вычисляемый в подпрограмме GRAD (размерность массива G равна N); э, KR, DELTA — числа, определяющие условие останова: счет прекращается, если на k-й итерации ($k = 1, 2, \ldots$) окажется выполненным хотя бы одно из условий:

1) $\|g(x_k)\| <$ э, э > 0,

2) $\|x_k - x_{k-1}\| < \text{DELTA}$ $(\text{DELTA} > 0)$,

3) $k = KR$;

ALPHA — коэффициент растяжения пространства, ALPHA > 1; k — счетчик итераций; формальный параметр k введен для возможности выдачи на АЦПУ информации о процессе работы алгоритма; INT — число, определяющее те итерации, при которых будет выдаваться на АЦПУ информация о процессе работы алгоритма; x — текущее значение переменных; при обращении к подпрограмме MINIM массиву x должно быть присвоено начальное приближение x_0; x_k — значение переменных, полученных на k-й итерации (размерность массива x равна N); S — предполагаемое расстояние от начальной точки x_0 до точки минимума x^*; B — двумерный рабочий массив размерности (N, N): $G1$, $G2$, z — одномерные рабочие массивы, каждый размерности N.

При обращении к подпрограмме MINIM необходимо придать определенные значения следующим формальным параметрам:

N, э, KR, DELTA, ALPHA, INT, x, S.

Значение остальных формальных параметров при обращении несущественно.

После окончания работы подпрограммы MINIM фактические параметры N, э, KR, DELTA, ALPHA, INT, S имеют те же значения, что и в момент обращения к процедуре. Остальные параметры изменяют свои значения:

x — полученные в результате работы подпрограммы значения переменных; F — значение функции в полученной точке; k — номер итерации, на которой процедура закончила свою работу; G — градиент функции в полученной точке. Для работы подпрограммы необходимо наличие в основной программе подпрограмм FUN (F, x), GRAD (G, x), PRIN (F, G, x, k), где FUN (F, x) — подпрограмма, вычисляющая значение минимизируемой функции $f(x)$ в точке x.

Таблица 4

Результаты решения тестового примера

К	$f(x_k)$	x_k
0	2.42_{10}^{+1}	—1.20000 1.00000
10	1.39_{10}^{+0}	—0.00195 —0.06203
20	1.14_{10}^{-1}	0.71829 0.49729
30	4.44_{10}^{-4}	1.01880 1.03700
40	2.29_{10}^{-8}	1.00008 1.00014
50	9.72_{10}^{-13}	1.00000 1.00000
54	6.14_{10}^{-14}	1.00000 1.00000

В результате работы подпрограммы FUN идентификатору F должно быть присвоено значение минимизируемой функции в точке, определяемой значением массива x. При этом значение x не должно измениться.

GRAD (G, x) — подпрограмма, вычисляющая значение градиента функции $f(x)$ в точке x.

В результате работы подпрограммы массиву G должно быть присвоено значение градиента функции $f(x)$ в точке x. При этом значение массива x не должно измениться.

PRIN (F, G, x, k) работает на той итерации k, для которой выполнено условие INT* ($k \div$ INT) $= k$. Подпрограмма служит для выдачи на АЦПУ информации о процессе работы алгоритма.

В результате работы подпрограммы PRIN ни один фактический параметр не должен измениться.

Опишем, из каких соображений выбирать значение параметра S. Если S выбрано слишком малым, это приведет к большому количеству вычислений значений функции на одной итерации, что может существенно увеличить время счета. Подпрограмма может вообще не сработать, если S выбрано настолько малым, что в результате ошибок округления представление в машине чисел $f(x_0)$, $f\left(x_0 - S\dfrac{g(x_0)}{\|g(x_0)\|}\right)$ окажется одним и тем же. При слишком больших значениях S возможна ошибка — «переполнение АУ». Такая же ошибка может произойти при неудачном выборе начального приближения x_0. Если указанной ошибки не произойдет, то выбор большого значения S не приведет к существенному увеличению времени счета.

Значение коэффициента растяжения ALPHA рекомендуется брать в интервале от 2 до 3. Заметим, что при ALPHA = 1 подпрограмма MINIM реализует обычный градиентный метод с регулировкой шага, подобной наискорейшему спуску.

Для работы процедуры требуется около N^2 ячеек памяти (N — размерность пространства переменных).

Приведем результаты решения тестового примера. Решалась задача минимизации функции

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

минимум которой достигается в $x^* = (0, 0)$, $f(x^*) = 0$. Фактические параметры подпрограммы MINIM имели следующие значения:

$$N = 2,\ э = 10^{-7},\ KR = 1000,\ \text{DELTA} = 10^{-6},$$
$$\text{ALPHA} = 2,\ \text{INT} = 10,\ x = (-1 \cdot 2{,}0),\ S = 1.$$

Значение остальных параметров, как указывалось, несущественно.

С заданной степенью точности алгоритм решил задачу за 54 итерации. Результаты работы программы приведены в табл. 4.

Программа

```
     SUBROUTINE MINIM (N, G, э, KR, DELTA,
     *ALPHA, K, INT, F, X, S, B, G1, G2, Z)
      DIMENSION  G (N),  X (N)
      DIMENSION B (N, N), G1 (N), G2 (N), Z (N)
      H = S
      NH = 3
      Б = 1/ALPHA — 1
      K = 1
      LL = INT
100   DO 20 I = 1, N
      DO 21 J = 1, N
21    B (I, J) = 0
20    B (I, I) = 1
      CALL FUN (F, X)
      CALL GRAD (G, X)
      DD = 0
      DO 22 I = 1, N
22    DD = DD + G (I)** 2
      DD = SQRT (DD)
      IF (DD.LT. э) GO TO 200
      DO 23 I = 1, N
      G1 (I) = G (I)
23    Z (I) = X (I)
      K2 = 0
      K1 = 1
104   D = H/DD
      DO 10 I = 1, N
10    X (I) = X (I) — G (I)* D
      CALL FUN (D, X)
      IF (D. GE. F) GO TO 11
      K1 = 0
      F = D
      K2 = K2 + 1
      IF (K2. LT.5) GO TO 104
      H = H* 10
      K2 = 0
      GO TO 104
11    IF (K1.NE.1) GO TO 12
      DO 13 I = 1, N
13    X (I) = Z (I)
      H = H/2
      GO TO 104
      12 IF (LL. NE. K) GOTO 24
      LL = LL + INT
      CALL PRIN (F, G, X, K)
24    D = 0
      DO 25 I = 1, N
25    D = D + (X (I) — Z (I))** 2
      D = SQRT (D)
      H = D/NH
      IF (D. LT. DELTA) GO TO 200
      CALL FUN (F, X)
      CALL GRAD(G,X)
```

```
      DD = 0
      DO 26 I = 1, N
   26 DD = DD + G (I)** 2
      DD = SQRT (DD)
      IF (DD.LT. э) GO TO 200
      IF (K.GE.KR) GO TO 200
  101 K = K + 1
      DO 30 I = 1, N
      G2 (I) = 0
      DO 31 J = 1, N
   31 G2 (I) = G2 (I) + B (J, I)* G (J)
   30 CONTINUE
      DO 32 I = 1, N
   32 G1 (I) = G1 (I) — G2 (I)
      D = 0
      DO 33 I = 1, N
   33 D = D + G1 (I)** 2
      D = SQRT (D)
      IF (D. GE.1.E — 18) GO TO 34
      DO 35 I = 1, N
   35 G1 (I) = G2 (I)
      GO TO 102
   34 DO 36 I = 1, N
   36 G1 (I) = G1 (I)/D
      DO 37 I = 1, N
      D = 0
      DO 38 J = 1, N
   38 D = D + B (I, J)* G1 (J)
      D = D* Б
      DO 39 J = 1, N
   39 B (I, J) = B (I, J) + G1 (J)* D
   37 CONTINUE
      D = 0
      DO 40 I = 1, N
   40 D = D + G1 (I)* G2 (I)
      D = Б*D
      DO 41 I = 1, N
   41 G1 (I) = G1 (I)* D + G2 (I)
  102 D = 0
      DO 42 I = 1, N
   42 D = D + G1 (I)** 2
      D = SQRT (D)
      DO 43 I = 1, N
      G (I) = 0
      DO 44 J = 1, N
   44 G (I) = G (I) + B (I, J)* G1 (J)
   43 CONTINUE
      IF (D. GE. 1. E — 18) GO TO 45
      D = 0
      DO 46 I = 1, N
   46 D = D + (X (I) — Z (I))** 2
      D = SQRT (D)
      H = D/NH
      GO TO 100
   45 DO 47 I = 1, N
   47 G (I) = G (I)/D
      DO 48 I = 1, N
   48 Z (I) = X (I)
      K1 = 0
  103 DO 49 I = 1, N
```

```
49 X (I) = X (I) — H* G (I)
   CALL FUN (D, X)
   K1 = K1 + 1
   IF (D. GE. F) GO TO 50
   F = D
   GO TO 103
50 H = K1*H/NH
   F = D
   CALL GRAD (G, X)
   IF (LL.NE.K) GO TO 56
   LL = LL + INT
   CALL PRIN (F, G, X, K)
56 IF (K, GE, KR) GO TO 200
   DD = 0
   DO 51 I = 1, N
51 DD = DD + G (I)**2
   DD = SQRT (DD)
   IF (DD. LT. э) GO TO 200
   D = 0
   DO 52 I = 1, N
52 D = D + (X (I) — Z (I))** 2
   D = SQRT (D)
   IF (D. LT. DELTA) GO TO 200
   IF (H. LE.1.E + 10) GO TO 53
   H = D/NH
   GO TO 100
53 D = H/D
   IF (D. LE.1000) GO TO 101
   DO 54 I = 1, N
   DO 55 J = 1, N
55 B (I, J) = D* B (I, J)
54 CONTINUE
   H = H/D
   GO TO 101
200 CONTINUE
   CALL PRIN (F, G, X, K)
   RETURN
   END
```

ВЛАДИМИР СЕРГЕЕВИЧ МИХАЛЕВИЧ
НАУМ ЗУСЕЛЕВИЧ ШОР
ЛЮСЯ АЛЕКСАНДРОВНА ГАЛУСТОВА
НИКОЛАЙ ГЕОРГИЕВИЧ ЖУРБЕНКО
АЛЕКСАНДРА ИВАНОВНА МОМОТ
АНТОНИНА НИКОЛАЕВНА СИБИРКО
ВЛАДИМИР АЛЕКСЕЕВИЧ ТРУБИН
ГЕННАДИЙ НИКОЛАЕВИЧ ЮН

ВЫЧИСЛИТЕЛЬНЫЕ МЕТОДЫ ВЫБОРА ОПТИМАЛЬНЫХ ПРОЕКТНЫХ РЕШЕНИЙ

Печатается по постановлению ученого совета Института кибернетики АН УССР

Редакторы *Т. С. Мельник, С. Д. Кошис*
Оформление художника *А. Л. Омельянюка*
Художественный редактор *И. П. Антонюк*
Технический редактор *И. Н. Лукашенко*
Корректор *И. В. Точаненко*

Информ. бланк № 654.
БФ 00231. Сдано в набор 27.X 1976 г. Подписано в печать 1. IV 1977 г. Формат 60×90$^1/_{16}$. Бумага типогр. № 1. Усл. печ. л. 11,25. Учетно-изд. л. 11,33. Тираж 3300. Изд. № 272. Зак. № 7-189.
Цена 1 руб 83 коп.

Издательство «Наукова думка»,
252601, Киев-601, ГСП, ул. Репина, 3.

Книжная фабрика «Коммунист» РПО «Полиграфкнига» Госкомиздата УССР. Харьков, ул. Энгельса, 11.