ӘОЖ 510.67

Л.С.Фазылова, А.Е.Сланбекова

Е.А.Бөкетов атындағы Қарағанды мемлекеттік университеті

БІР АЙНЫМАЛЫ ФУНКЦИЯЛАР ҮШІН МИНИМУМДАУ ӘДІСТЕРДІ ПРОГРАММАЛАУ

В настоящей работе рассмотрены одномерные методы минимизации, относящиеся к последовательной стратегии, приведены программы вычисления точек минимума одномерной функции этими методами.

Present work explains one dimensional minimisation methods, which belong to consequent strategy, and contains programs for calculation points of minimum in one dimensional function using these methods.

Есептің қойылымы [1]. $J(u)$ функциясына минималды мән беретін, яғни

$$\min_{u \in R_1} J(u) = J(u_*)$$

шартын қанағаттандыратын, $u_* \in U$ нүктесін табу керек.

Мақалада бір айнымалы функциялардың (унимодалды функциялар [2]) минимум нүктелерін есептеудің келесі әдістері қарастырылады:

– асыра артық алу әдісі [1];

– дихотомия әдісі [2];

– «алтын қима» әдісі [1];

– Фибоначчи әдісі [1].

Егер барлық нүктелер алдын ала, есептеу басталғанға дейін, белгілі болса, бұл есеп пассивтік стратегияға жатады.

Тізбектік стратегияның өзі екі тәсілмен іске асырылатынын айта кеткен дұрыс. Бұл жұмыста тізбектік стратегия бірі-бірінің ішіне салынған интервалдар тізбегі жиыны түрінде құрылған. Интервалдың әрқайсысы минимум нүктесін қамтиды. Бастапқы интервалды таңдағанда оның шектері берілген функция осы интервалда унимодалды болатындай шартты қанағаттандыруы қажет. Әрі қарай қарастырылатын минимумдау әдістері унимодалды функциялардың минимумын есептеуге тиімді болғандықтан, алдын ала бастапқы интервалды таңдаудың осындай әдісін алу қажет.

Ондай әдістердің бірі — Свенн алгоритмі [1]. Оның көмегімен унимодалды функцияның минимум нүктесін алдын ала қамтитын интервал, анықталмаған бастапқы интервал табылады.

Әрі қарай анықталмаған интервалды қандайда бір тізбектік стратегиясы әдісін пайдалану арқылы кішірейту, яғни функцияны ағымдағы интервалдың екі нүктесінде есептеу негізінде, орындалады. Унимодалды функцияның қасиеті бөлінген ішкі интервалдардың қайсысында минимум нүктесі болмайтынын анықтайды. Жаңа интервал ретінде минимум нүктесі бар ішкі интервал қабылданады.

Қарастырылған анықталмаған интервалды кішірейтудің алгоритмін тиімді бағалау үшін келесі сипаттама енгізіледі:

$$R(N) = \frac{|L_n|}{|L_0|},$$

бұл қатынасты анықталмаған бастапқы интервалдың азаюының сипаттамасы деп атайды, мұнда $|L_n|$ — функцияның N есептеу нәтижесінде алынған интервалдың ұзындығы, ал $|L_0|$ — анықталмаған бастапқы интервалдың ұзындығы.

Бір айнымалы функция үшін туындыны есептемей минимумдау әдісіне кесіндінің *дихотомия әдісі* жатады [2, 3].

Тізбектік стратегияға жататын *алтын қима әдісін* қарастырамыз. Бұл әдісте анықталмаған бастапқы интервалы талап етілетін дәлдікпен беріледі. Интервалды кішірейту алгоритмі функцияның екі нүктедегі мәндерін талдауға сүйенеді. Бұндай нүктелер ретінде «алтын қима» нүктелері қабылданады.

1-анықтама [3]. $[a,b]$ кесіндісін тең емес екі бөлікке бөліп,

$$\frac{b-a}{b-c} = \frac{b-c}{c-a}$$

қатынастарын қанағаттандыратын c нүктесін *алтын қима* дейміз.

«Алтын қима» әдісінің алгоритмі [1].

1-қадам. Анықталмаған бастапқы интервалын $L_0 = [a_0,\ b_0]$, $l > 0$ қажетті дәлдігін беру.

2-қадам. $k = 0$ болсын.

3-қадам. $c_0 = a_0 + \frac{3-\sqrt{5}}{2}(b_0 - a_0)$, $d_0 = a_0 + b_0 - c_0$ есептеу,

мұнда $\frac{3-\sqrt{5}}{2} \approx 0{,}382$.

4-қадам. $J(c_k)$, $J(d_k)$ есептеу.

5-қадам. $J(c_k)$-ны $J(d_k)$-мен салыстыру:

а) егер $J(c_k) \le J(d_k)$ болса, онда

$$a_{k+1} = a_k,\ b_{k+1} = d_k,\ d_{k+1} = c_k,\ c_{k+1} = a_{k+1} + b_{k+1} - c_k$$

және 7-ші қадамға көшу;

b) егер $J(c_k) > J(d_k)$ болса, онда

$$a_{k+1} = c_k,\ b_{k+1} = b_k,\ c_{k+1} = d_k,\ d_{k+1} = a_{k+1} + b_{k+1} - d_k$$

және 6-шы қадамға көшу.

6-қадам. $\Delta = | a_{k+1} - b_{k+1} |$ есептеу және шарттың аяқталғанын тексеру керек:

а) егер $\Delta \le l$ болса, онда іздеу процесі аяқталады және $u_* \in [a_{N-1},\ b_{N-1}]$. Жуық шешімі ретінде соңғы интервалдың ортасын

$$u_* \cong \frac{a_{N-1} + b_{N-1}}{2}$$

алуға болады.

b) егер $\Delta > l$ болса, онда $k = k + 1$ және 4-ші қадамға көшу.

Жинақтылық [1]. «Алтын қима» әдісі үшін анықталмаған бастапқы интервалдың салыстырмалы азаюдың сипаттамасы келесі формула бойынша табылады:

$$R(N) = (0{,}618)^{N-1},$$

мұндағы N — функцияны есептеудің саны.

Ескерту [1]. 1) Анықталмаған интервалдар «алтын қима» әдісінде келесі түрде болады:

$$L_0,\ L_2,\ L_3,\ L_4\ \ldots,$$

яғни «алтын қима» әдісінде бірінші итерацияда функцияның екі мәндерін, ал әрбір келесі итерацияларда функцияның тек бір мәнін есептеу керек.

2) Анықталмаған интервал ұзындығының азаюы

$$\frac{|L_0|}{|L_2|} = \frac{|L_2|}{|L_3|} = \frac{|L_3|}{|L_4|} = \ldots = 1{,}618$$

тұрақты шама болады.

3) Егер $R(N)$ шамасы белгіленген болса, онда функцияны есептеудің саны $N \ge 1 + \frac{\ln R(N)}{\ln 0{,}618}$ шартын қанағаттандыратын ең кіші бүтін сан болып табылады.

Тізбектік стратегияға жататын Фибоначчи әдісін баяндаймыз. Ол әдіс бойынша функцияны есептеу нүктелері Фибоначчи сандары тізбегін пайдалану арқылы табылады.

Анықтама [1]. Фибоначчи сан тізбегі келесі формуламен анықталады:

$$F_0 = F_1 = 1,\ F_k = F_{k-1} + F_{k-2},\ k = 2,\ 3,\ 4, \ldots$$

Фибоначчи сандарының тізбегінің түрі мынадай:

1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, 233,…

Фибоначчи әдісінің алгоритмі [1].

1-қадам. Анықталмаған бастапқы интервалын беру $L_0 = [a_0,\ b_0]$; l — рұқсат етілетін ұзындықтың шектік аралығы; $\varepsilon > 0$ — айырымдылық тұрақтысы.

2-қадам. Функцияны есептеу N саны $F_N \ge \frac{|L_0|}{l}$ шартын және F_0, F_1, ..., F_N Фибоначчи санын қанағаттандыратын ең кіші бүтін санды табу.

3-қадам. $k = 0$ болсын.

4-қадам. $y_0 = a_0 + \frac{F_{N-2}}{F_N}(b_0 - a_0)$, $z_0 = a_0 + \frac{F_{N-1}}{F_N}(b_0 - a_0)$ есептеу.

5-қадам. $J(y_k)$, $J(z_k)$ есептеу.

6-қадам. $J(y_k)$-ны $J(z_k)$-мен салыстыру:

а) егер $J(y_k) \le J(z_k)$ болса, онда

$$a_{k+1} = a_k,\ b_{k+1} = z_k,\ z_{k+1} = y_k,\ y_{k+1} = a_{k+1} + \frac{F_{N-k-3}}{F_{N-k-1}}(b_{k+1} - a_{k+1})$$

және 7-ші қадамға көшу;

b) егер $J(y_k) > J(z_k)$ болса, онда

$$a_{k+1} = y_k,\ b_{k+1} = b_k,\ y_{k+1} = z_k,\ z_{k+1} = a_{k+1} + \frac{F_{N-k-2}}{F_{N-k-1}}(b_{k+1} - a_{k+1})$$

және 7-ші қадамға көшу.

7-қадам. Шарттың аяқталғанын тексеру керек және қажетті болса, шешімді алу үшін функцияның N-ші қорытынды есептеуін табу керек:

а) егер $k \ne N-3$ болса, онда $k = k+1$ және 5-ші қадамға көшу;

b) егер $k = N-3$ болса, онда әрқашан $y_{N-2} = z_{N-2} = \frac{a_{N-2} + b_{N-2}}{2}$, яғни функцияның жаңа есептеп шығарылған нүктесі жоқ болды.

$y_{N-1} = y_{N-2} = z_{N-2}$, $z_{N-1} = y_{N-1} + \varepsilon$ болсын. y_{N-1} және z_{N-1} нүктелерінде функцияның мәні есептеледі және анықталмаған интервалдың шеткі шекаралары табылады:

– егер $J(y_{N-1}) \le J(z_{N-1})$ болса, онда $a_{N-1} = a_{N-2}$, $b_{N-1} = z_{N-1}$;

– егер $J(y_{N-1}) > J(z_{N-1})$ болса, онда $a_{N-1} = y_{N-1}$, $b_{N-1} = b_{N-2}$ болады.

Іздеу процесі аяқталады және $u_* \in [a_{N-1},\ b_{N-1}]$. Жуық шешімі ретінде соңғы интервалдың кез келген нүктесін, мысалы, оның ортасын

$$u_* \cong \frac{a_{N-1} + b_{N-1}}{2}$$

алуға болады.

Жинақтылық [1]. Фибоначчи әдісі үшін анықталмаған бастапқы интервалдың салыстырмалы азаюдың сипаттамасы келесі формула бойынша табылады:

$$R(N) = \frac{1}{F_N},$$

мұнда N — функцияны есептеудің саны.

Ескерту [1]. 1) Функция есептеуде N саны берілген жағдайда Фибоначчи әдісі бұрын қарастырылған әдістерімен (асыра артық алу, дихотомия, «алтын қима» әдістері) салыстырғанда анықталмағандықтың шеткі интервалды ең кіші шамасын қамтамасыз етеді.

2) Анықталмаған интервалдың нөмірлеуі «алтын қима» әдісіндегідей: L_0, L_2, L_3, L_4 ...

3) k-ші итерацияда анықталмаған интервалдың ұзындығы $\frac{F_{N-k-1}}{F_{N-k}}$ ережесі бойынша қысқарады.

Қарастырылған бес әдісті Turbo Pascal тілінде орындалу процедураларын келтіреміз [3–5]:

```
function MetodSvenna (f:TFunc; u0,t:Real; var a,b:Real) : Boolean;
{ Свенн әдісі арқылы унимодальды кесіндіні табу.
 u0 — бастапқы нүкте; t — қадам;
 a, b — ізделіп отырған кесіндінің шектері}
var
  k: Integer;
  u,d: Real;
begin {MetodSvenna}
  k := 0;
  if (f(u0-t)>=f(u0)) and (f(u0)<=f(u0+t)) then
  {Бастапқы кесінді табылды}
```

```
    begin
      a := u0 - t;
      b := u0 + t;
      MetodSvenna := True;
      Exit
    end
  else
    if (f(u0-t)<=f(u0)) and (f(u0)>=f(u0+t)) then
    {Функция унимодалды функция емес}
      begin
        MetodSvenna := False;
        Exit
      end;
  if (f(u0-t)>=f(u0)) and (f(u0)>=f(u0+t)) then
    begin
      d := t;
      a := u0;
      u := u0 + t
    end
  else
    if (f(u0-t)<=f(u0)) and (f(u0)<=f(u0+t)) then
      begin
        d := -t;
        b := u0;
        u := u0 - t
      end;
  k := 1;
  repeat
    u0 := u;
    u := u0 + exp (k*ln(2)) * d;
    if f(u) < f(u0) then
      begin
        if d = t then a := u0
        else b := u0;
        k := k+1
      end
  until f(u) >= f(u0);
  if d = t then b := u
  else a := u;
  MetodSvenna := True
end; {MetodSvenna}
{-----------}
procedure MetodPerebora (f:TFunc; a,b,eps:Real; var x,effect:Real);
{Асыра артық алу әдісі.
 a, b – кесіндінің шектері; eps – дәлдік; x – ізделіп отырған минимум нүктесі;
 effect – ақырғы кесіндінің ұзындығы бастапқы кесіндінің ұзындығының қатынасына тең}
var
  N,i: LongInt;
  u: Real;
begin {MetodPerebora}
  N := round ((b-a)/eps); {нүктелер саны}
  x := a;
  for i:=1 to N do
    begin
      u := a + i * eps;
      if f(u) < f(x) then
```

```
        x := u
    end;
  effect := eps / (b-a);
end; {MetodPerebora}
{-----------}
procedure MetodDihotomii (f:TFunc; a,b,eps,delta:Real; var x,effect:Real);
{Дихотомия әдісі.
 a, b — кесіндінің шектері; eps — дәлдік;
 delta — айырушылықтың шамасы; x — ізделініп отырған минимум нүктесі;
 effect — ақырғы кесіндінің ұзындығы бастапқы кесіндінің ұзындығының қатынасына тең}
var
  c,d,L0: Real;
begin {MetodDihotomii}
  L0 := b - a;  {бастапқы кесіндінің ұзындығы}
  while (b-a)/2 > eps do
    begin
      c := (a + b - delta)/2;
      d := (a + b + delta)/2;
      if f(c) <= f(d) then b := d
      else a := c
    end;
  x := (a + b)/2;
  effect := (b - a) / L0
end; {MetodDihotomii}
{-----------}
procedure MetodFibonacci (f:TFunc; a,b,l,e:Real; var x,effect:Real);
{Фибоначчи әдісі.
 a, b — кесіндінің шектері; l — ақырғы аралықтың ұзындығының  рұқсат етілетін мағынасы;
 e — тұрақты; x — ізделіп отырған минимум нүктесі;
 effect — ақырғы кесіндінің ұзындығы бастапқы кесіндінің ұзындығының қатынасына тең }
type
  TArray = array [0..1000] of LongInt;
var
  k: Integer;
  N: LongInt;
  y,z,L0: Real;
  Fib: TArray;
{---}
procedure Fibonacci (var f:TArray; n:LongInt);
{ Фибоначчи N  сандарды табу}
var
  i: LongInt;
begin
  f[0] := 1;
  f[1] := 1;
  for i:=2 to n do
    f[i] := f[i-1] + f[i-2]
end;
{---}
begin {MetodFibonacci}
  Fibonacci (Fib,1000);
  N:= 0;
  L0 := b - a; {бастапқы кесіндінің ұзындығы }
  {Есептеулердің санын табамыз}
  while Fib[N] < L0/l do
    N := N + 1;
```

```
  y := a + Fib[N-2] / Fib[N] * (b-a);
  z := a + Fib[N-1] / Fib[N] * (b-a);
  for k:=0 to N-4 do
    if f(y) <= f(z) then
      begin
        b := z;
        z := y;
        y := a + Fib[N-k-3] / Fib[N-k-1] * (b-a);
      end
    else
      begin
        a := y;
        y := z;
        z := a + Fib[N-k-2] / Fib[N-k-1] * (b-a);
      end;
  z := y + e;
  if f(y) <= f(z) then b := z
  else a := y;
  x := (a + b)/2;
  effect := (b - a) / L0
end; {MetodFibonacci}
{-----------}
procedure MetodZolot (f:TFunc; a,b,eps:Real; var x,effect:Real);
{«Алтын қима» әдісі.
 a, b — кесіндінің шектері; eps — дәлдігі; x — ізделіп отырған минимум нүктесі;
 effect — ақырғы кесіндінің ұзындығы бастапқы кесіндінің ұзындығының қатынасына тең }
var
  c,d,L0: Real;
{---}
function x1 (a,b:Real) : Real;
begin
  x1 := 0.5 * ((sqrt(5)-1) * a + (3-sqrt(5)) * b);
end;
{---}
function x2 (a,b:Real) : Real;
begin
  x2 := 0.5 * ((sqrt(5)-1) * b + (3-sqrt(5)) * a);
end;
{---}
begin {MetodZolot}
  L0 := b - a; { бастапқы кесіндінің ұзындығы }
  c := x1 (a,b);
  d := x2 (a,b);
  while (b-a)/2 > eps do
    if F(c) >= F(d) then
      begin
        a := c;
        c := d;
        d := x2 (a,b);
      end
    else
      if F(c) < F(d) then
        begin
          b := d;
          d := c;
          c := x1 (a,b);
```

```
    end;
  x := (a + b)/2;
  effect := (b - a) / L0
end; {MetodZolot}
```

Тұжырым. Барлық қолданылған әдістердің ерекшелігі және артықшылығы болып олардың әрбір итерация қадамында функцияның туындыларын есептеуді қажет етпей, тек қана сәйкес нүктелердегі функция мәнін есептеумен шектелуінде.

Әдебиеттер тізімі

1. *Пантелеев А.В., Летова Т.А.* Методы оптимизации в примерах и задачах. — М.: Высш. шк., 2002. — 542 с.
2. Сборник задач по математике для втузов. Специальные курсы / Под ред. А.В.Ефимова. — М.: Наука, 1984. — 301 с.
3. *Аттетков А.В., Галкин С.В., Зарубин В.С.* Методы оптимизации. — М.: Изд-во МГТУ им. Н.Э.Баумана, 2001. — 436 с.
4. *Вирт Н.* Алгоритмы и структуры данных / Пер. с англ. — М.: Мир, 1989. — 360 с.
5. *Культин Н.Б.* Программирование в Turbo Pascal 7.0 и Delphi. — 2-е изд., перераб. и доп. — СПб.: БХВ-Санкт-Петербург, 1999. — 416 с.

ӘОЖ 510.67

Л.С.Фазылова, А.Е.Сланбекова

Е.А.Бөкетов атындағы Қарағанды мемлекеттік университеті

АҚЫРЛЫ ӨЛШЕМДІ КЕҢІСТІКТЕГІ МИНИМУМДАУДЫҢ САНДЫҚ ӘДІСТЕРІН ПРОГРАММАЛАУ

В настоящей работе рассмотрены две стратегии поиска минимума функции одной переменной, приведена программа вычисления точек минимума одномерной функции различными методами, проводится сравнительный анализ рассмотренных методов.

Present work explains two strategies of searching minimums in function with one variable, contains program for calculation points of minimum in one dimensional function using different methods and benchmark analysis of these methods.

Есептің қойылымы [1]. $J(u)$ функциясына минималды мән беретін, яғни

$$\min_{u \in R_1} J(u) = J(u_*)$$

шартын қанағаттандыратын, $u_* \in U$ нүктесін табу керек.

Мақалада минимум нүктелерін есептеудің екі стратегиясы қарастырылады:

– пассивтік (асыра артық алу әдісі);

– тізбектік (дихотомия, «алтын қима», Фибоначчи әдістері).

Егер барлық нүктелер алдын ала, яғни есептеуге дейін, берілсе — бұл *пассивтік стратегия* болады. *Асыра артық алу әдісі* іздеудің пассивтік стратегиясына жатады. Бірөлшемді минимизациялы есепті шешудің алгоритмін қарастырайық:

а) Айталық $[a_0,\ b_0]$ анықталмағандықтың бастапқы интервалы болсын, $u^i = a_0 + i\dfrac{(b_0 - a_0)}{N+1}$, $(i = 0,\ 1,\ \ldots,\ N+1)$ формуласы арқылы бір-бірінен бірдей қашықтықтағы нүктелерді есептейміз;

b) табылған нүктелерде функцияның мәнін есептеу

$$J(u^i),\ (i = 0,\ 1,\ \ldots,\ N+1);$$

c) u^i $(i = 0,\ 1,\ \ldots,\ N+1)$ нүктелері арасынан функцияның ең кіші мән қабылдайтын нүктені таңдап алу: