

BỘ GIÁO DỤC VÀ ĐÀO TẠO

**VIỆN HÀN LÂM KHOA HỌC
VÀ CÔNG NGHỆ VIỆT NAM**

VIỆN TOÁN HỌC

Lê Xuân Đoàn

**MỘT SỐ PHƯƠNG PHÁP TỐI ƯU
KHÔNG DÙNG ĐẠO HÀM**

LUẬN VĂN THẠC SĨ TOÁN HỌC

HÀ NỘI – 2015

BỘ GIÁO DỤC VÀ ĐÀO TẠO

VIỆN HÀN LÂM KHOA HỌC

VÀ CÔNG NGHỆ VIỆT NAM

VIỆN TOÁN HỌC

Lê Xuân Đoàn

**MỘT SỐ PHƯƠNG PHÁP TỐI ƯU
KHÔNG DÙNG ĐẠO HÀM**

Chuyên ngành: Toán ứng dụng

Mã số: 60460112

LUẬN VĂN THẠC SĨ TOÁN HỌC

NGƯỜI HƯỚNG DẪN KHOA HỌC

PGS.TS. Bùi Thế Tâm

HÀ NỘI – 2015

MỤC LỤC

	Trang
LỜI NÓI ĐẦU	4
Chương 1. PHƯƠNG PHÁP NELDER – MEAD CỰC TIỂU	
HÀM NHIỀU BIẾN	6
1. Mô tả thuật toán Nelder – Mead trong không gian R^2	6
2. Mô tả thuật toán Nelder – Mead trong không gian R^n	12
2.1. Phát biểu chung của thuật toán	13
2.2. Mô tả một bước lặp của thuật toán Nelder – Mead	13
2.3. Kiểm tra hội tụ	17
2.4. Xây dựng đơn hình xuất phát	17
2.5. Sơ đồ khối của thuật toán Nelder – Mead	18
3. Bài toán tối ưu với ràng buộc tổng quát	18
4. Thuật toán Nelder – Mead với các biến bị chặn	22
5. Các tính chất của thuật toán Nelder – Mead.....	23
6. Chương trình máy tính cho thuật toán Nelder – Mead	33
6.1. Bài toán	33
6.2. Các biến và mảng dùng trong chương trình	34
6.3. Văn bản chương trình	34
6.4. Giải ví dụ bằng số	42
6.5. Các ví dụ đã chạy chương trình	45

Chương 2. PHƯƠNG PHÁP TÌM KIẾM TRỰC TIẾP

HOOKE – JEEVES	49
1. Mô tả thuật toán Hooke - Jeeves trong không gian R^n	49
1.1. Phát biểu bài toán	49
1.2. Tư tưởng cơ bản của thuật toán	49
1.3. Mô tả một bước lặp của thuật toán Hooke – Jeeves	51
2. Ví dụ minh họa cho thuật toán Hooke – Jeeves trong không gian R^2 ...	53
3. Chương trình máy tính cho thuật toán Hooke – Jeeves	55
3.1. Bài toán	55
3.2. Các biến và mảng dùng trong chương trình	55
3.3. Văn bản chương trình	56
3.4. Giải ví dụ bằng số	61
3.5. Các ví dụ đã chạy chương trình	65
KẾT LUẬN	70
TÀI LIỆU THAM KHẢO	71

LỜI NÓI ĐẦU

Trong các vấn đề thực tế chúng ta thường gặp bài toán phải cực tiểu hay cực đại một hàm nhiều biến mà nó không có đạo hàm bậc nhất, không có đạo hàm bậc hai, không lồi, không lõm, không DC, không đơn điệu, không thỏa mãn điều kiện Lipchitz. Do đó các phương pháp tụt gradien, phương pháp Niu-ơn, phương pháp gradien liên hợp... đều không áp dụng được. Khi đó ta cần sử dụng các phương pháp tối ưu không dùng đạo hàm, ví dụ như phương pháp dò tìm theo các tọa độ Hooke – Jeeves 1960 [6] và phương pháp tụt theo các đơn hình Nelder – Mead 1965 [1], phương pháp Monte – Carlo...

Mục đích của luận văn này nhằm trình bày lại hai thuật toán không dùng đạo hàm Nelder – Mead và Hooke – Jeeves để cực tiểu một hàm nhiều biến và thử nghiệm các thuật toán này trên máy tính. Từ khi ra đời hai thuật toán trên đã được sử dụng rộng rãi trong các ngành kỹ thuật. Chính vì sự hiệu quả của hai thuật toán mà trong những năm gần đây nhiều tác giả đã cố gắng cải tiến các thuật toán này và xây dựng cơ sở lý thuyết chặt chẽ của các thuật toán, chứng minh sự hội tụ của chúng (xem [2], [3], [5]).

Các chương trình máy tính lập trình C trong luận văn này dùng các thuật toán cải tiến năm 2004 của các thuật toán Nelder – Mead và Hooke – Jeeves. Các thử nghiệm chỉ ra rằng các thuật toán này rất có hiệu quả để giải các bài toán cực tiểu các hàm nhiều biến không khả vi.

Luận văn bao gồm hai chương và tài liệu tham khảo.

Chương 1 trình bày chi tiết các bước của thuật toán Nelder – Mead trong không gian R^n (minh họa cụ thể trong không gian R^2), cho sơ đồ khối của thuật toán, mở rộng thuật toán trong trường hợp các biến bị chặn, các tính chất của thuật toán Nelder – Mead, cuối chương là chương trình máy tính và các thử nghiệm của thuật toán trên máy tính.

Chương 2 dành cho phương pháp tìm kiếm trực tiếp Hooke – Jeeves: mô tả các bước của thuật toán, sơ đồ khối của thuật toán, ví dụ minh họa của thuật toán, chương trình máy tính thể hiện thuật toán và các thử nghiệm trên máy tính.

Trong quá trình viết luận văn cũng như trong quá trình xử lý văn bản chắc chắn không tránh khỏi những thiếu sót nhất định. Tác giả luận văn rất mong nhận được sự góp ý của các thầy cô và các bạn đồng nghiệp.

Tác giả bày tỏ lòng biết ơn chân thành và sâu sắc tới thầy hướng dẫn PGS-TS Bùi Thế Tâm đã tận tình chỉ bảo, tạo điều kiện và giúp đỡ tôi có thêm nhiều kiến thức, khả năng nghiên cứu, tổng hợp tài liệu để hoàn thành luận văn này.

Tác giả xin chân thành cảm ơn các GS, PGS, TS của Viện Toán Học và trường Đại Học Khoa Học – Đại học Thái Nguyên đã giảng dạy và tạo mọi điều kiện thuận lợi trong quá trình tác giả học tập nghiên cứu.

Tác giả cũng xin chân thành cảm ơn gia đình và bạn bè đã quan tâm giúp đỡ động viên để tác giả hoàn thành luận văn này.

Tháng 5/ 2015

Tác giả

Chương 1

PHƯƠNG PHÁP NELDER – MEAD CỰC TIỂU HÀM NHIỀU BIẾN

Trong chương này trình bày thuật toán Nelder – Mead trong không gian hai chiều và thuật toán Nelder – Mead trong không gian nhiều chiều, thuật toán Nelder – Mead với các biến bị chặn. Nội dung của chương dựa chủ yếu trên tài liệu [1], [2], [3],[4] và [7] .

Vào năm 1965 hai nhà thống kê người Anh làm việc tại Trung tâm Nghiên cứu Thực vật Quốc gia đã phát minh ra phương pháp tìm kiếm trực tiếp theo đơn hình Nelder – Mead. Phương pháp này càng nổi bật lên khi người ta đặc biệt quan tâm tới lời giải số của các bài toán tối ưu phi tuyến phức tạp trong thực tế. Vì việc nhận được đạo hàm bậc nhất của $f(x)$ cần tối ưu thường là không làm được, nên ưa thích nhất của đa số những người làm thực tế là phương pháp tìm trực tiếp mà chỉ cần giá trị của hàm $f(x)$. Phương pháp mới Nelder – Mead đã đáp ứng điều đó. Từ đó phương pháp Nelder – Mead được xem như là một trong những phương pháp được trích dẫn và được dùng nhiều nhất để cực tiểu hàm phi tuyến không ràng buộc.

Để hiểu rõ tư tưởng của thuật toán ta hãy mô tả thuật toán trong không gian R^2 và sau đó trình bày thuật toán trong không gian R^n .

1. Mô tả thuật toán Nelder – Mead trong không gian R^2

Khởi tạo tam giác BGW

Giả sử $f(x, y)$ là hàm số lồi chặt cần cực tiểu. Khởi đầu cho ba đỉnh của tam giác. $V_k = (x_k, y_k)$, $k = 1, 2, 3$. Sau đó tính giá trị của hàm tại ba đỉnh

này là: $z_k = f(x_k, y_k)$ với $k = 1, 2, 3$. Sắp xếp giá trị hàm theo thứ tự tăng dần: $z_1 \leq z_2 \leq z_3$. Dùng kí hiệu:

$$B = (x_1; y_1), \quad G = (x_2; y_2), \quad W = (x_3; y_3), \quad (1)$$

trong đó B là đỉnh tốt nhất (best), G là đỉnh tốt (good, sát với đỉnh tốt nhất), W là đỉnh xấu nhất (worst).

Tính điểm giữa của cạnh tốt

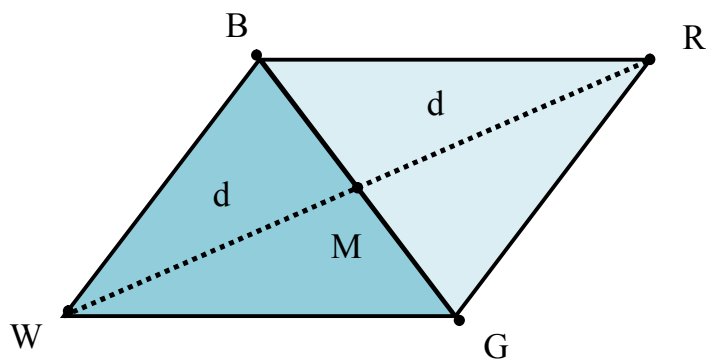
Quá trình tính toán dùng điểm giữa của đoạn thẳng nối B và G :

$$M = \frac{B + G}{2} = \left(\frac{x_1 + x_2}{2}, \frac{y_1 + y_2}{2} \right) \quad (2)$$

Phép phản xạ dùng điểm R

Hàm $f(x, y)$ giảm khi ta đi dọc theo cạnh của tam giác từ W tới B và cũng giảm khi đi dọc theo cạnh từ W tới G. Do đó hàm $f(x, y)$ sẽ nhận giá trị nhỏ hơn về phía đoạn BG khi xuất phát từ W. Để xác định R trước tiên tìm điểm giữa M của cạnh BG. Về đoạn thẳng từ W tới M, gọi độ dài của nó là d. Kéo dài đoạn thẳng này một đoạn bằng d nữa qua M ta được điểm R. Công thức véc tơ của điểm R là:

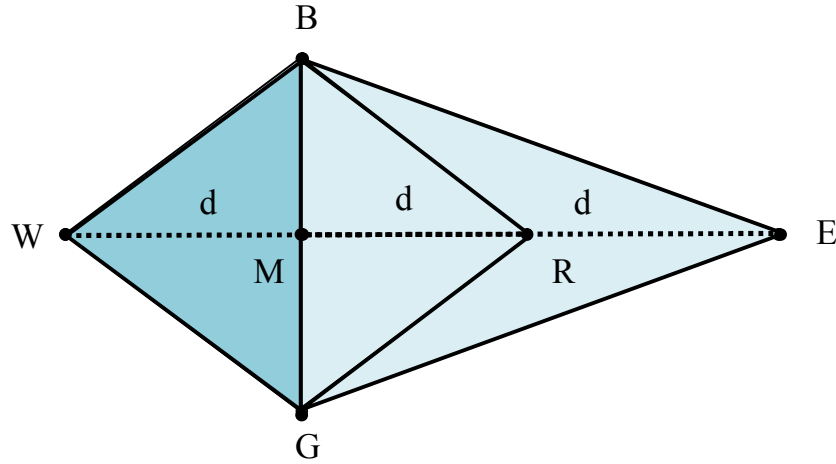
$$R = M + (M - W) = 2M - W. \quad (3)$$



Hình 1. Tam giác ΔBGW , trung điểm M và điểm phản xạ R.

Phép dẫn dùng điểm E

Nếu giá trị của hàm tại điểm R nhỏ hơn giá trị của hàm tại điểm W thì ta đã chuyển theo hướng tốt để cực tiểu hàm f . Tất nhiên điểm cực tiểu của f chưa chắc là điểm R. Ta sẽ đi tiếp theo đoạn thẳng MR tới một điểm E. Điều đó tạo nên tam giác BGE. Điểm E được tìm bằng cách đi tiếp một khoảng d bổ sung dọc theo đường nối M và R.



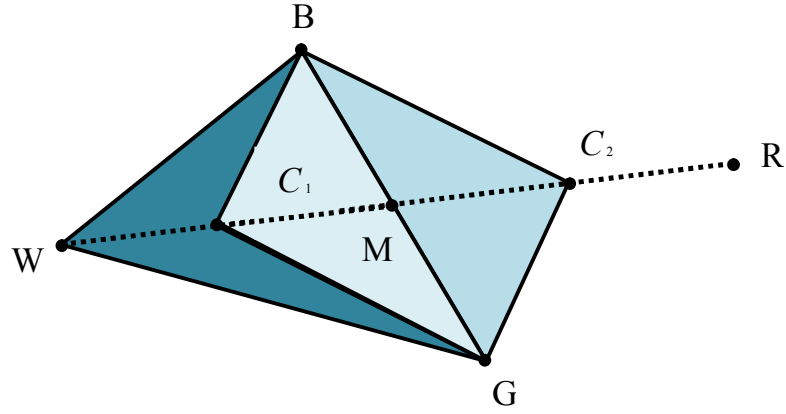
Hình 2. Tam giác ΔBWG , điểm R và điểm dẫn E.

Nếu giá trị của hàm tại E nhỏ hơn giá trị của hàm tại R thì ta đã tìm được một đỉnh tốt hơn R. Công thức véc tơ tính điểm E là:

$$E = R + (R - M) = 2R - M. \quad (4)$$

Phép co dùm điểm C

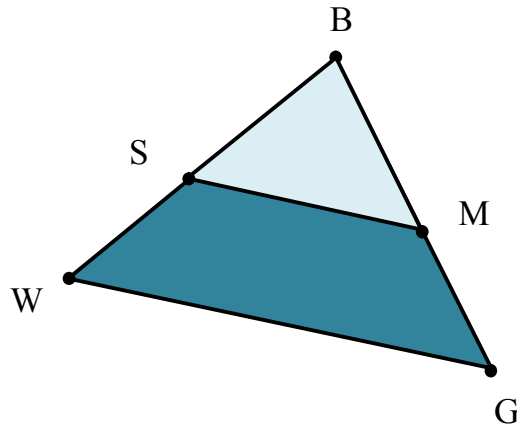
Nếu giá trị của hàm tại điểm R và W là như nhau, ta cần kiểm tra điểm khác. Xét hai điểm giữa C_1 và C_2 của $\overline{WM}$ và $\overline{MR}$ tương ứng. Gọi C là điểm có giá trị hàm nhỏ hơn trong hai điểm C_1 và C_2 , tam giác mới sẽ chọn là BGC.



Hình 3. Điểm co C_1 hoặc C_2 của phương pháp Nelder – Mead.

Thu hẹp đơn hình lại về B

Nếu giá trị hàm tại C không nhỏ hơn giá trị hàm tại W thì các điểm G và W cần phải co lại về B. Điểm G thay bởi M, W thay bởi điểm S là điểm giữa của $\overline{WB}$.



Hình 4. Thu hẹp tam giác ΔBGW về điểm B.

Mô tả logic của mỗi bước lặp

Thuật toán tính toán hữu hiệu sẽ chỉ tính giá trị của hàm tại các điểm cần thiết. Tại mỗi bước lặp một điểm mới được tìm để thay thế cho W. Khi W được thay thế thì nó không cần xét tiếp, bước lặp cũng kết thúc. Mô tả logic của thuật toán trong trường hợp 2 chiều [4], cho trong đoạn giả trình sau:

```

IF  $f(R) < f(G)$  THEN
    BEGIN {Trường hợp 1 : Phản xạ hoặc giãn}
        IF  $f(B) < f(R)$  THEN  $W \leftarrow R$ 
        ELSE
            Tính E và  $f(E)$ 
            IF  $f(E) < f(B)$  THEN  $W \leftarrow E$ 
            ELSE  $W \leftarrow R$ 
        ENDIF
    ENDIF
END
ELSE
    BEGIN {Trường hợp 2: Co hoặc thu hẹp}
        IF  $f(R) < f(W)$  THEN  $W \leftarrow R$ 
        Tính  $C = \frac{W+M}{2}$  hay  $C = \frac{M+R}{2}$  và  $f(C)$ 
        IF  $f(C) < f(W)$  THEN  $W \leftarrow C$ 
        ELSE
            Tính S và  $f(S)$ 
             $W \leftarrow S$ 
             $G \leftarrow M$ 
        ENDIF
    ENDIF
END

```

Ví dụ. Tìm cực tiểu hàm, [4] :

$$f(x, y) = x^2 - 4x + y^2 - y - xy.$$

Xuất phát từ ba đỉnh

$$V_1 = (0, 0), \quad V_2 = (1.2, 0.0), \quad V_3 = (0.0, 0.8).$$

Giá trị hàm f tại 3 đỉnh tương ứng là :

$$f(0, 0) = 0.0, \quad f(1.2, 0.0) = -3.36, \quad f(0.0, 0.8) = -0.16.$$

Với các kí hiệu dùng trong thuật toán thì

$$B = (1.2, 0.0), \quad G = (0.0, 0.8), \quad W = (0, 0).$$

Điểm W sẽ bị thay. Tọa độ điểm M và R là :

$$M = \frac{B + G}{2} = (0.6, 0.4)$$

$$R = 2M - W = (1.2, 0.8)$$

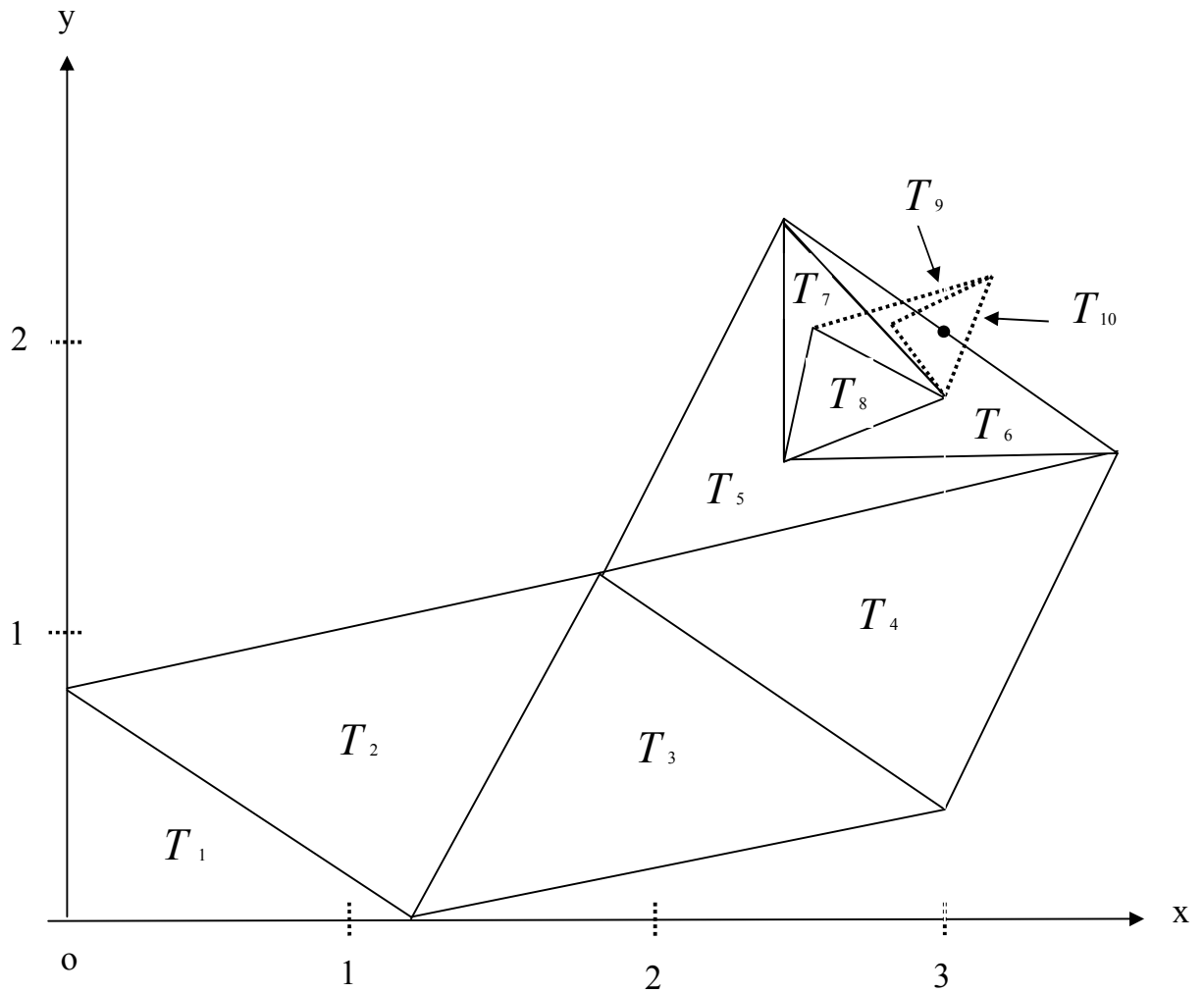
Giá trị của hàm $f(R) = f(1.2, 0.8) = -4.48$ là nhỏ hơn $f(G)$, trường hợp 1 xảy ra. Vì $f(R) \leq f(B)$ nên chúng ta di chuyển theo hướng đúng và véc tơ E được xây dựng theo công thức:

$$E = 2R - M = 2(1.2, 0.8) - (0.6, 0.4) = (1.8, 1.2).$$

Giá trị của hàm $f(E) = f(1.8, 1.2) = -5.88$ là nhỏ hơn $f(B)$ và tam giác mới có các đỉnh:

$$V_1 = (1.8, 1.2), \quad V_2 = (1.2, 0.0), \quad V_3 = (0.0, 0.8).$$

Quá trình tiếp tục và sinh ra một dãy các tam giác hội tụ tới điểm lời giải $(3, 2)$.



Hình 5. Dãy các tam giác $\{T_k\}, k = 1, 2, 3 \dots$ hội tụ tới điểm $(3, 2)$ theo phương pháp Nelder – Mead .

Bảng 1 cho các giá trị của hàm tại các đỉnh của tam giác đối với một số bước lặp đầu tiên của quá trình lặp. Đến bước lặp thứ 33 thì được đỉnh tốt nhất là $B = (2.99996456, 1.99983839)$

và $f(B) = -6.99999998$. Các giá trị này là xấp xỉ với $f(3,2) = -7$.

Bảng 1. Giá trị hàm tại các đỉnh của tam giác trong bài toán.

K	Điểm tốt nhất	Điểm tốt	Điểm xấu nhất
1	$f(1.2, 0.0) = -3.36$	$f(0.0, 0.8) = -0.16$	$f(0.0, 0.0) = 0.00$
2	$f(1.8, 1.2) = -5.88$	$f(1.2, 0.0) = -3.36$	$f(0.0, 0.8) = -0.16$
3	$f(1.8, 1.2) = -5.88$	$f(3.0, 0.4) = -4.44$	$f(1.2, 0.0) = -3.36$
4	$f(3.6, 1.6) = -6.24$	$f(1.8, 1.2) = -5.88$	$f(3.0, 0.4) = -4.44$
5	$f(3.6, 1.6) = -6.24$	$f(2.4, 2.4) = -6.24$	$f(1.8, 1.2) = -5.88$
6	$f(2.4, 1.6) = -6.72$	$f(3.6, 1.6) = -6.24$	$f(2.4, 2.4) = -6.24$
7	$f(3.0, 1.8) = -6.96$	$f(2.4, 1.6) = -6.72$	$f(2.4, 2.4) = -6.24$
8	$f(3.0, 1.8) = -6.96$	$f(2.55, 2.05) = -6.7725$	$f(2.4, 1.6) = -6.72$
9	$f(3.0, 1.8) = -6.96$	$f(3.15, 2.25) = -6.9525$	$f(2.55, 2.05) = -6.7725$
10	$f(3.0, 1.8) = -6.96$	$f(2.8125, 2.0375) = -6.95640625$	$f(3.15, 2.25) = -6.9525$

2. Mô tả thuật toán Nelder – Mead trong không gian R^n

Thuật toán Nelder – Mead dùng để cực tiểu hàm số thực $f(x)$ với $x \in R^n$. Trình bày của thuật toán dựa trên tài liệu [2]. Có 4 tham số cần xác định trong thuật toán Nelder – Mead: hệ số phản xạ ρ , hệ số dẫn χ , hệ số co γ và hệ số thu hẹp σ . Theo bài báo gốc của Nelder – Mead [1] các tham số này cần thỏa mãn:

$$\rho > 0, \chi > 1, \chi > \rho, 0 < \gamma < 1 \text{ và } 0 < \sigma < 1. \quad (5)$$

Cách chọn phổ biến nhất được dùng trong thuật toán Nelder – Mead chuẩn là:

$$\rho = 1, \chi = 2, \gamma = \frac{1}{2} \text{ và } \sigma = \frac{1}{2} \quad (6)$$

Các giá trị tham số này làm cho phương pháp trở nên hiệu quả, ngay cả khi làm việc trong những tình huống phức tạp.

2.1. Phát biểu chung của thuật toán

Lúc bắt đầu bước lặp thứ k ($k \geq 0$) ta có đơn hình không suy biến Δ_k với $n + 1$ đỉnh, mỗi một đỉnh là một điểm trong không gian R^n . Ta luôn luôn có thể giả thiết rằng bước lặp thứ k bắt đầu bằng việc sắp xếp và đánh nhãn các đỉnh này là $x^1, x^2, \dots, x^{n+1}$ sao cho:

$$f_1^{(k)} \leq f_2^{(k)} \leq \dots \leq f_{n+1}^{(k)} \quad (7)$$

trong đó $f_i^{(k)}$ kí hiệu cho $f(x^i)$. Bước lặp thứ k sinh ra một tập $n + 1$ đỉnh xác định một đơn hình mới $\Delta_{k+1} \neq \Delta_k$. Vì ta cần tính cực tiểu của hàm f nên ta coi x^1 là điểm tốt nhất, x^{n+1} là đỉnh xấu nhất, x^n là điểm gần xấu nhất. Tương tự ta coi $f_{n+1}^{(k)}$ là giá trị hàm xấu nhất.

Kết quả của mỗi bước lặp là hoặc (1) tìm được một đỉnh mới thay thế x^{n+1} trong tập hợp các đỉnh trong bước lặp tiếp, hoặc (2) nếu thực hiện việc thu hẹp thì một tập U đỉnh mới cùng với x^1 tạo nên đơn hình mới cho bước lặp tiếp theo.

2.2. Mô tả một bước lặp của thuật toán Nelder – Mead

Bước 1. Sắp xếp

Sắp xếp $n+1$ đỉnh thỏa mãn:

$$f(x^1) \leq f(x^2) \leq \dots \leq f(x^{n+1}).$$

Bước 2. Phép phản xạ

Dịch chuyển đơn hình từ điểm x^{n+1} . Tính điểm phản xạ x^r theo công thức:

$$x^r = \bar{x} + \rho(\bar{x} - x^{n+1}) = (1 + \rho)\bar{x} - \rho x^{n+1} \quad (8)$$

trong đó: $\bar{x} = \frac{\sum_{i=1}^n x^i}{n}$ là trọng tâm của n điểm tốt nhất (tất cả các đỉnh trừ đỉnh x^{n+1}). Tính giá trị $f_r = f(x^r)$. Điểm x^r là điểm đối xứng với x^{n+1} qua $\bar{x}$.

Nếu $f_1 \leq f_r < f_n$ thì chấp nhận điểm phản xạ x^r và kết thúc bước lặp.

Bước 3. Phép dẫn

Trường hợp $f_r < f_1$, tức là x^r là điểm tốt hơn n đỉnh của đơn hình. Khi đó hướng đi từ $\bar{x}$ tới x^r là hướng thuận tiện nhất để di chuyển. Vì vậy ta tiến hành dẫn theo hướng từ $\bar{x}$ tới x^r để được điểm x^e :

$$\begin{aligned} x^e &= \bar{x} + \chi(x^r - \bar{x}) \\ &= \bar{x} + \chi\rho(\bar{x} - x^{n+1}) \\ &= (1 + \rho\chi)\bar{x} - \rho\chi x^{n+1}, \end{aligned}$$

và tính giá trị $f_e = f(x^e)$.

Nếu $f_e < f_r$ thì ta chấp nhận x^e và kết thúc bước lặp, trái lại (nếu $f_e \geq f_r$) thì ta chấp nhận x^r và kết thúc bước lặp.

Bước 4. Phép co

Nếu $f_r \geq f_n$ thì ta tiến hành phép co.

a) Co bên ngoài

Nếu $f_n \leq f_r < f_{n+1}$ (tức là x^r tốt hơn thực sự x^{n+1}) ta thực hiện phép co ở ngoài đơn hình bằng cách tính:

$$\begin{aligned} x^c &= \bar{x} + \gamma(x^r - \bar{x}) \\ &= \bar{x} + \gamma\rho(\bar{x} - x^{n+1}) \\ &= (1 + \rho\gamma)\bar{x} - \rho\gamma x^{n+1}, \end{aligned}$$

và tính giá trị $f_c = f(x^c)$.

Nếu $f_c \leq f_r$ ta chấp nhận x^c và kết thúc bước lặp, trái lại chuyển sang bước 5 (thực hiện phép thu hẹp đơn hình).

b) Co bên trong đơn hình

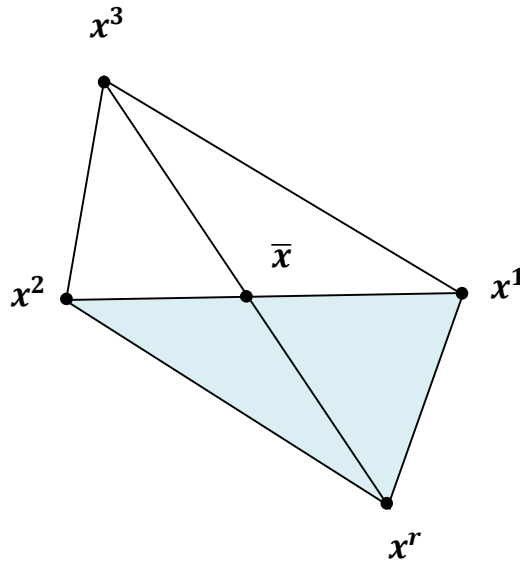
Nếu $f_r \geq f_{n+1}$ ta tiến hành co vào bên trong đơn hình bằng cách tính:

$$\begin{aligned}x^{cc} &= \bar{x} - \gamma(\bar{x} - x^{n+1}) \\ &= (1 - \gamma)\bar{x} + \gamma x^{n+1},\end{aligned}$$

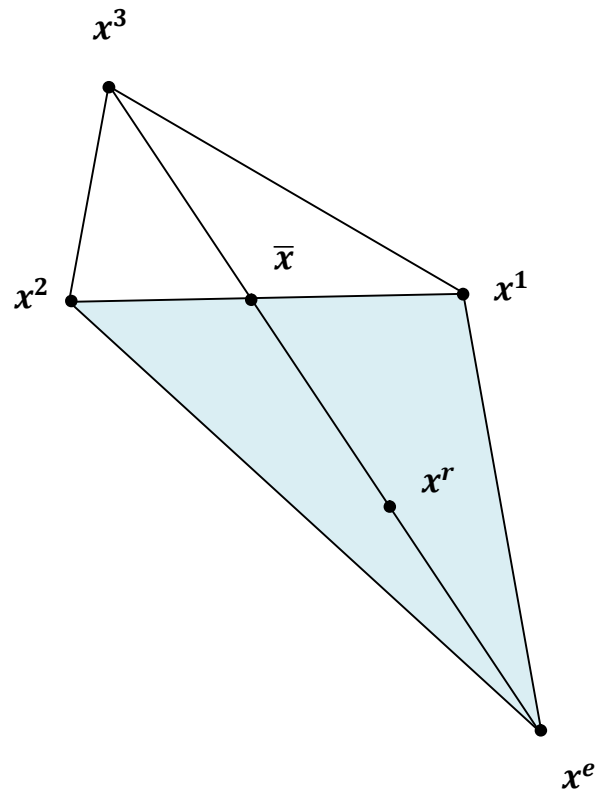
và tính $f_{cc} = f(x^{cc})$. Nếu $f_{cc} < f_{n+1}$ thì chấp nhận x^{cc} và kết thúc bước lặp, trái lại sang bước 5 .

Bước 5. Thực hiện thu hẹp đơn hình

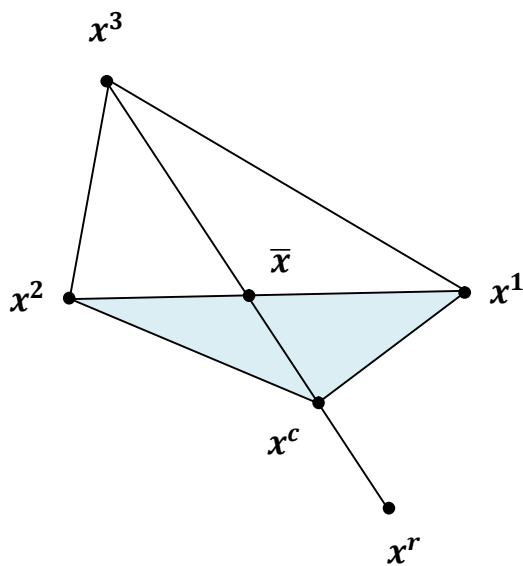
Tính hàm f tại n điểm $v^i = x^1 + \sigma(x^i - x^1)$, $i = 2, \dots, n + 1$. Các đỉnh chưa được sắp xếp của đơn hình trong bước lặp tiếp theo bao gồm $x^1, v^2, v^3, \dots, v^{n+1}$.



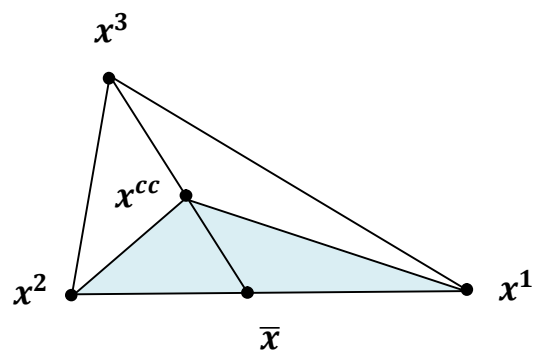
Hình 6a. Phép phản xạ.



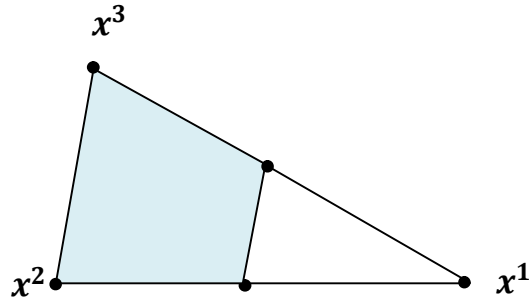
Hình 6b. Phép dẫn.



Hình 6c . Phép co ngoài.



Hình 6d .Phép co trong.



Hình 6e. Phép thu hẹp.

Hình 6a – 6e minh họa phép phản xạ, phép dẫn, phép co ngoài, phép co trong và phép thu hẹp với $\rho = 1$, $\chi = 2$, $\gamma = \frac{1}{2}$, $\sigma = \frac{1}{2}$.

2.3. Kiểm tra hội tụ

Trong thuật toán khi lập được đơn hình mới ta cần kiểm tra điều kiện hội tụ của dãy các giá trị hàm. Tính giá trị trung bình của hàm mục tiêu tại các đỉnh của đơn hình:

$$\bar{f} = \frac{\sum_{i=1}^{n+1} f_i}{n+1}$$

Kiểm tra độ lệch chuẩn của các giá trị hàm :

$$\sigma = \sqrt{\frac{\sum_{i=1}^{n+1} (f_i - \bar{f})^2}{n}} < \varepsilon \quad (9)$$

trong đó ε là số dương cho trước, chẳng hạn $\varepsilon = 0.0001$. Nếu điều kiện (9) thỏa mãn thì dừng tính toán.

2.4. Xây dựng đơn hình xuất phát

Đơn hình ban đầu thường chọn như sau : chọn x^1 tùy ý, sau đó chọn

$$x^2 = x^1 + ke^1,$$

$$x^3 = x^1 + ke^2,$$

.....

$$x^{n+1} = x^1 + ke^n,$$

với $k > 0$ là số chọn tùy ý, còn e^j là véc tơ đơn vị trên trục tọa độ thứ j của không gian R^n .

Vì thuật toán Nelder – Mead chỉ hội tụ về điểm thỏa mãn điều kiện dừng, nên để thoát khỏi điểm này trong thực hành tính toán trên máy tính ta cần chọn lại điểm x^1 một cách ngẫu nhiên và tạo đơn hình mới nhằm hi vọng tìm tối ưu toàn cục.

2.5. Sơ đồ khối của thuật toán Nelder – Mead

Hình 7 trình bày sơ đồ khối của thuật toán Nelder – Mead trong không gian R^n . Chú ý kí hiệu “+” trong hình để chỉ điều kiện được thỏa mãn, dấu “–” chỉ điều kiện không thỏa mãn.

3. Bài toán tối ưu với ràng buộc tổng quát

Cho tập D xác định bởi

$$D = \{x \in R^n : h_i(x) = 0, i = 1, 2, \dots, m, g_j(x) \leq 0, j = 1, 2, \dots, p\}$$

Cho hàm số $f : D \rightarrow R$, hàm số $h_i : R^n \rightarrow R$, hàm số $g_j : R^n \rightarrow R$.

Xét bài toán tối ưu với ràng buộc tổng quát dạng:

$$\min \{ f(x) : x \in R^n \} \quad (10)$$

với các điều kiện: $h_i(x) = 0, i = 1, 2, \dots, m$

$$g_j(x) \leq 0, j = 1, 2, \dots, p.$$

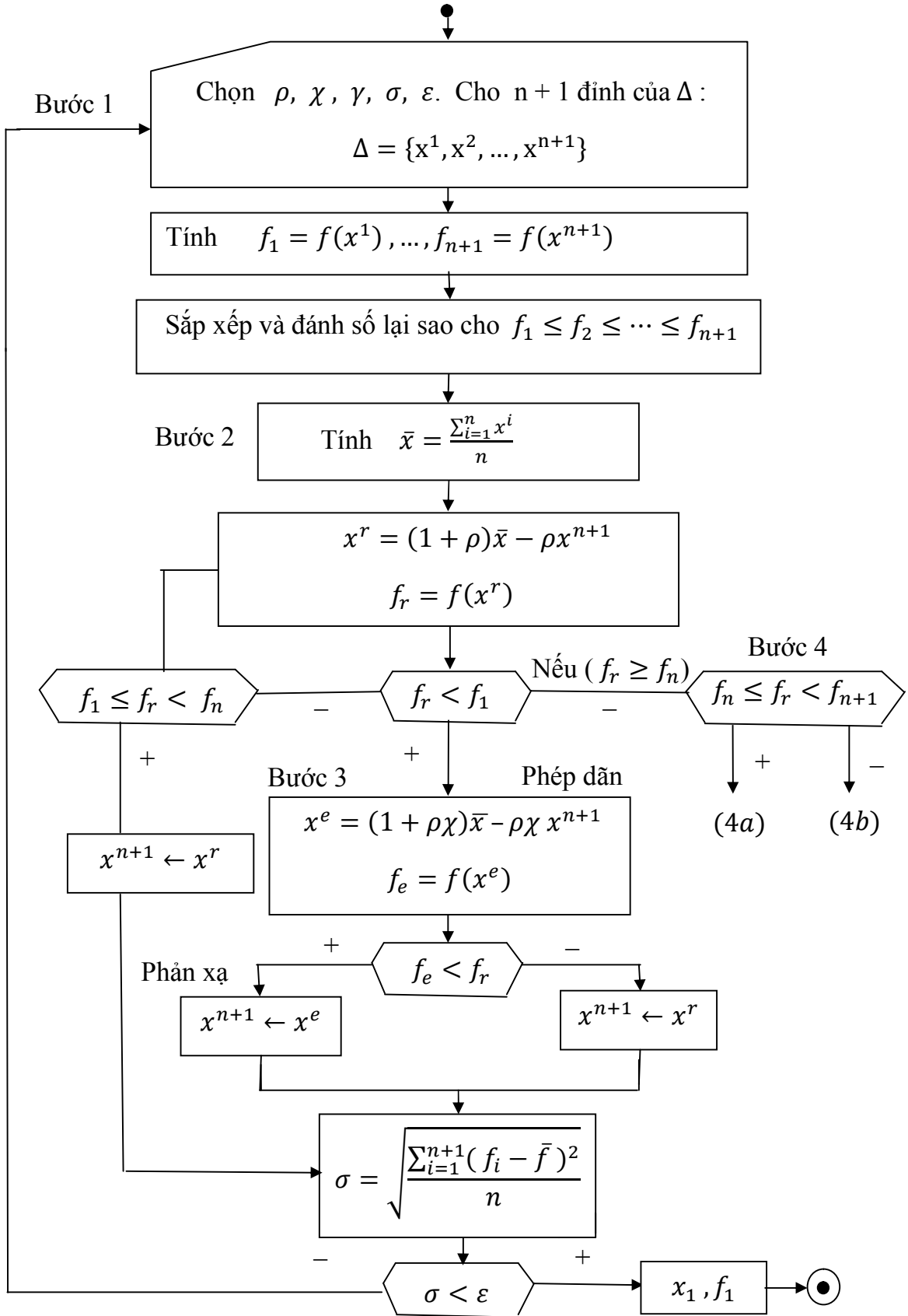
Dùng phương pháp hàm phạt có thể đưa bài toán (10) về giải một dãy các bài toán cực tiểu không ràng buộc:

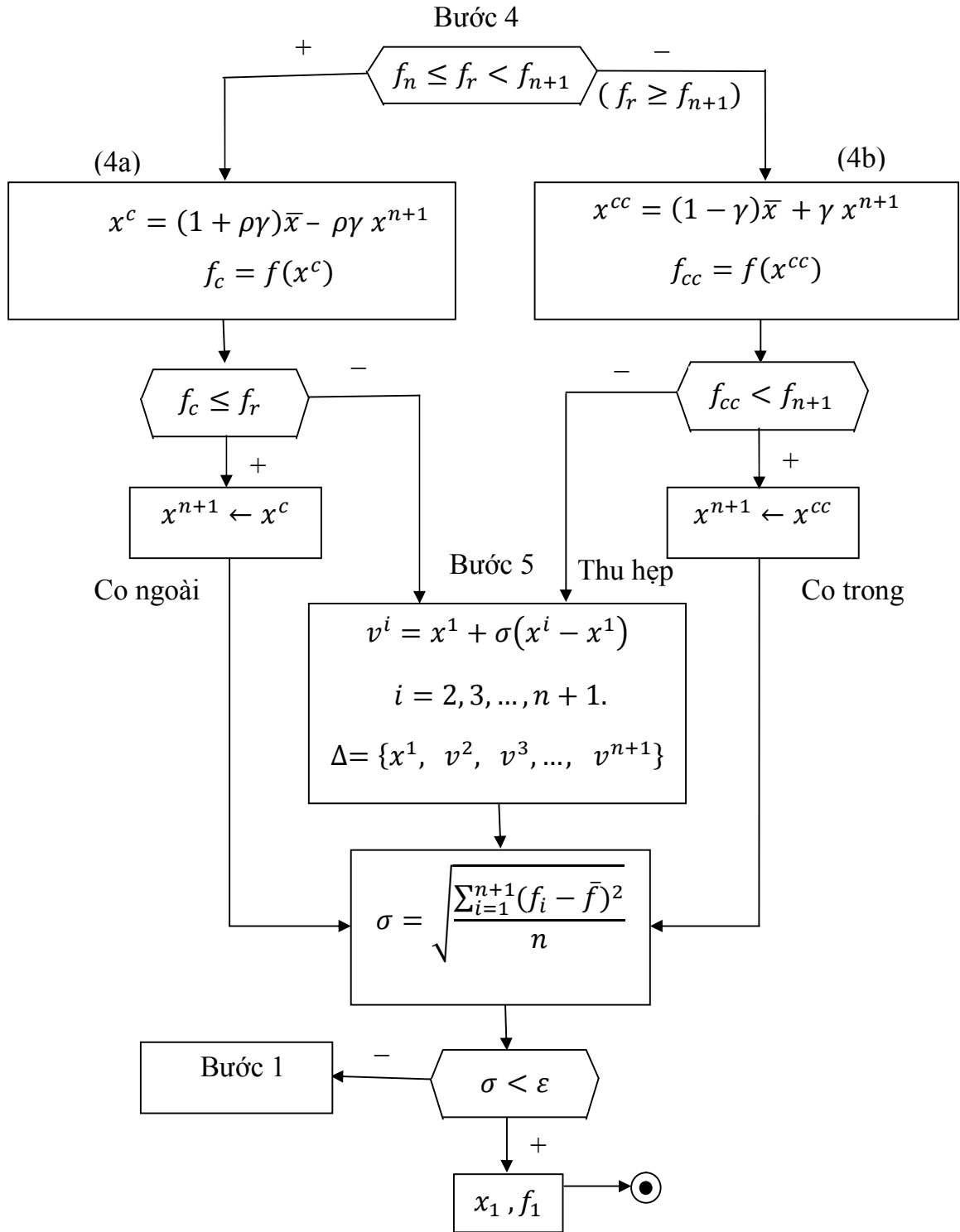
$$\min Q(x, t),$$

trong đó: $Q(x, t) = f(x) + \frac{1}{2}t \sum_{i=1}^m h_i^2(x) + \frac{1}{2}t \sum_{j=1}^p [\max\{0, g_j(x)\}]^2$.

Giả sử mỗi x^k là điểm cực tiểu toàn cục chính xác của hàm $Q(x, t_k)$ với dãy $\{t_k\}$ tăng dần và hội tụ tới $+\infty$. Khi đó mọi điểm tụ x^* của dãy $\{x^k\}$ là điểm cực tiểu toàn cục của bài toán (10).

Để cực tiểu các hàm $Q(x, t)$ ta có thể dùng phương pháp Nelder – Mead.





4. Thuật toán Nelder – Mead với các biến bị chặn

Cho hàm số $f : R^n \rightarrow R$. Xét bài toán

$$\min \{ f(x) : x \in R^n \}$$

với véc tơ x thỏa mãn : $x_i^{\min} \leq x_i \leq x_i^{\max}$, $i = 1, 2, 3, \dots, n$, (11)

trong đó x_i là tọa độ thứ i của véc tơ x .

Thuật toán được trình bày theo tài liệu [3]. Thuật toán ở mục này khác với thuật toán ở mục trước ở chỗ: nếu các điểm tạo được bởi phép phản xạ và phép dẫn vượt ra khỏi hình hộp (11) thì ta chiếu nó lên hình hộp.

Trong thuật toán đơn hình cỡ a khởi tạo tại điểm x_0 dựa trên qui tắc

$$x^i = x^0 + p e^i + \sum_{k=1, k \neq i}^n q e^k, \quad i = 1, \dots, n \quad (12)$$

trong đó e^k là véc tơ đơn vị thứ k và

$$p = \frac{a}{n\sqrt{2}}(\sqrt{n+1} + n - 1)$$

$$q = \frac{a}{n\sqrt{2}}(\sqrt{n+1} - 1).$$

Các đỉnh của đơn hình thay đổi qua các phép phản xạ, phép dẫn và phép co nhằm tìm một điểm tốt hơn. Thuật toán kết thúc khi giá trị hàm tại các đỉnh trở nên xấp xỉ bằng nhau (công thức (9)).

Ta bổ sung thêm hai tính chất của thuật toán. Thứ nhất thuật toán Nelder–Mead có thể không đảm bảo sự hội tụ tới cực tiểu địa phương, đặc biệt xảy ra khi đơn hình thu lại vào một không gian con. Thứ hai, nếu đơn hình là đủ lớn thì phương pháp có thể thoát ra ngoài vùng trũng của hàm. Cuối cùng, khi cỡ của đơn hình giảm, thuật toán trở thành địa phương.

Thuật toán Nelder–Mead nguyên thủy trình bày cho bài toán có miền không bị chặn. Với các biến bị chặn các điểm nhận được qua các phép phản xạ hoặc phép dẫn có thể nằm ở ngoài miền (10). Để đảm bảo tính bị chặn của các biến ta dùng phép chiếu:

$$\begin{cases} \text{if } (x_i < x_i^{\min}), & x_i = x_i^{\min} \\ \text{if } (x_i > x_i^{\max}), & x_i = x_i^{\max} \end{cases}$$

Lược đồ khối của thuật toán ở đây khác với thuật toán gốc Nelder–Mead sẽ ở cách khởi tạo đơn hình ban đầu (12) và bởi các biến bị chặn có dạng (11).

Ký hiệu x_i là đỉnh thứ i của đơn hình

f_i : giá trị hàm mục tiêu tại P_i

x^h : đỉnh của đơn hình có hàm mục tiêu lớn nhất

x^s : đỉnh của đơn hình có hàm mục tiêu lớn thứ hai.

x^l : đỉnh của đơn hình có giá trị hàm mục tiêu thấp nhất.

x^m : trọng tâm của các đỉnh đơn hình đơn hình (trừ x^h)

$r = 1$: hệ số phản xạ.

$\beta = \frac{1}{2}$: hệ số co.

$\gamma = \frac{1}{2}$: hệ số dẫn.

Sơ đồ khối của thuật toán Nelder – Mead với các biến bị chặn cho trong Hình 8 [3].

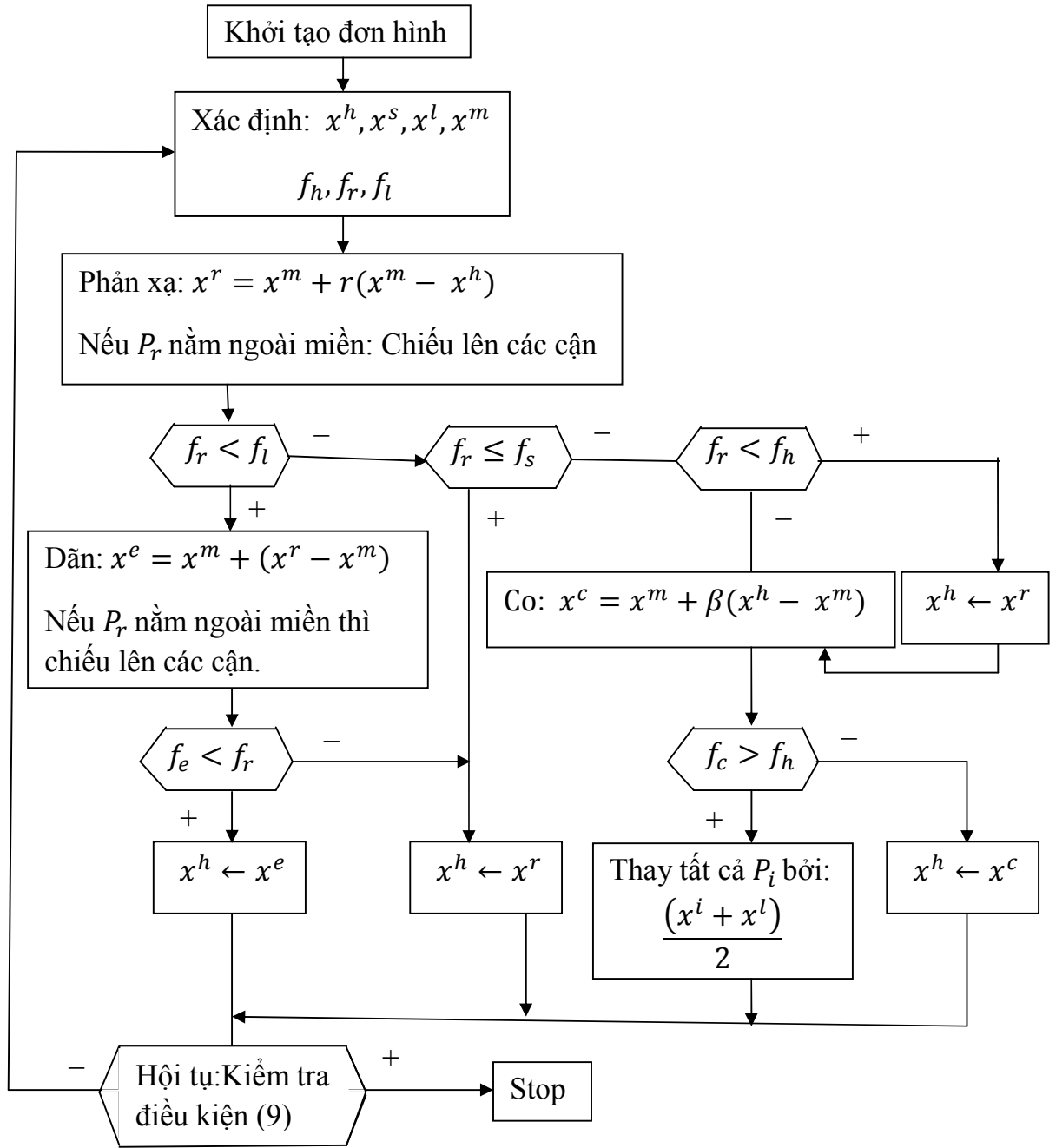
5. Các tính chất của thuật toán Nelder – Mead

Quy tắc sắp xếp bước không thu hẹp.

Khi bước không thu hẹp xảy ra, đỉnh xấu nhất $x_{n+1}^{(k)}$ bị loại bỏ, điểm chấp nhận tạo ra trong bước lặp k định nghĩa bởi $v^{(k)}$ trở thành đỉnh mới và nhận vị trí thứ $j + 1$ trong các đỉnh của đơn hình Δ_{k+1} , trong đó

$$j = \max_{0 \leq l \leq n} \{l : f(v^{(k)}) < f(x_{l+1}^{(k)})\};$$

tất cả các đỉnh khác được giữ nguyên thứ tự như ở bước lặp k .



Hình 8. Phương pháp Nelder – Mead với các biên bị chặn.

Quy tắc sắp xếp ở bước thu hẹp.

Nếu bước thu hẹp xuất hiện, duy nhất đỉnh được giữ lại từ $\Delta_k \rightarrow \Delta_{k+1}$ là $x_1^{(k)}$. Có một hoặc lớn hơn 1 đỉnh mới là đỉnh tốt nhất nếu:

$$\min \{f(v_2^{(k)}), \dots, f(v_{n+1}^{(k)})\} = f(x_1^{(k)}), \text{ khi đó: } x_1^{(k+1)} = x_1^{(k)}.$$

Ta định nghĩa chỉ số thay đổi k^* của bước lặp k là chỉ số nhỏ nhất của đỉnh mà khác nhau giữa bước lặp k và $k + 1$:

$$k^* = \min \{i : x_i^{(k)} \neq x_i^{(k+1)}\} \quad (13)$$

Trong bước không thu hẹp :

$$\begin{aligned} f_j^{(k+1)} &= f_j^{(k)} \quad \text{và} \quad x_j^{(k+1)} = x_j^{(k)}, \quad j < k^* \\ f_{k^*}^{(k+1)} &< f_{k^*}^{(k)} \quad \text{và} \quad x_{k^*}^{(k+1)} \neq x_{k^*}^{(k)} \\ f_j^{(k+1)} &= f_{j-1}^{(k)} \quad \text{và} \quad x_j^{(k+1)} = x_{j-1}^{(k)}, \quad j > k^* \end{aligned} \quad (14)$$

Như vậy, véc tơ $(f_1^{(k)}, \dots, f_{n+1}^{(k)})$ giảm chặt theo từ vựng tại mỗi bước lặp không thu hẹp.

Ví dụ, giả sử với $n = 4$ và các giá trị hàm thuộc các đỉnh tại bước lặp không thu hẹp k là $(1, 2, 2, 3, 3)$. Nếu $f(v^{(k)}) = 2$ thì các giá trị hàm tại bước lặp $k + 1$ là $(1, 2, 2, 2, 3)$, $x_4^{(k+1)} = v^{(k)}$, và $k^* = 4$.

Biểu diễn bằng ma trận

Để thuận tiện ta sử dụng kí hiệu ma trận để miêu tả cho bước lặp Nelder – Mead. Đơn hình Δ_k có thể biểu diễn bởi ma trận cỡ $n \times (n + 1)$ trong đó các cột là các đỉnh

$$\Delta_k = (x_1^{(k)} \dots x_{n+1}^{(k)}) = (B_k \ x_{n+1}^{(k)}), \quad B_k = (x_1^{(k)} \dots x_n^{(k)})$$

Với đơn hình bất kì Δ_k trong R^n , ta định nghĩa M_k là ma trận $n \times n$ mà cột thứ j của nó tương ứng là cạnh của Δ_k nối $x_j^{(k)}$ và $x_{n+1}^{(k)}$.

$$M_k \equiv \left(x_1^{(k)} - x_{n+1}^{(k)}, x_2^{(k)} - x_{n+1}^{(k)}, \dots, x_n^{(k)} - x_{n+1}^{(k)} \right) = B_k - x_{n+1}^{(k)} e^T \quad (15)$$

trong đó $e = (1, \dots, 1)^T$.

Trong không gian n chiều thể tích của Δ_k được cho bởi

$$\text{vol}(\Delta_k) = \frac{|\det(M_k)|}{n!} \quad (16)$$

Đơn hình Δ_k không suy biến nếu M_k không suy biến hoặc $\text{vol}(\Delta_k) > 0$. Thể tích của đơn hình chỉ phụ thuộc vào tọa độ của các đỉnh, không phụ thuộc vào thứ tự sắp xếp các đỉnh.

Ta định nghĩa bán kính của Δ_k là

$$\text{diam}(\Delta_k) = \max \left\| x_i^{(k)} - x_j^{(k)} \right\| \quad (i \neq j) \quad (17)$$

Trong bước lặp không thu hẹp, hàm được tính giá trị chỉ tại điểm thử có dạng :

$$z^{(k)}(\tau) := \bar{x}^{(k)} + \tau (\bar{x}^{(k)} - x_{n+1}^{(k)}) = (1 + \tau) \bar{x}^{(k)} - \tau x_{n+1}^{(k)} \quad (18)$$

trong đó hệ số τ là một trong bốn giá trị :

$$\tau = \rho \text{ (phản xạ)} ; \quad \tau = \rho \chi \text{ (dẫn)}. \quad (19)$$

$$\tau = \rho \gamma \text{ (co ngoài)} ; \quad \tau = -\gamma \text{ (co trong)}.$$

Kí hiệu τ_k là hệ số liên quan đến điểm chấp nhận của bước lặp thứ k . Như vậy, đỉnh mới $v^{(k)}$ thay thế đỉnh $x_{n+1}^{(k)}$ ở bước lặp thứ k được cho bởi công thức $v^{(k)} = z^{(k)}(\tau)$. Đôi khi ta gọi τ_k là kiểu dịch chuyển đối với bước lặp không thu hẹp k . Tại bước lặp thứ k của thuật toán Nelder – Mead ta chỉ ra rằng điểm thử (phản xạ , dẫn , co) có thể viết dưới dạng :

$$z^{(k)}(\tau) = \Delta_k t(\tau), \quad t(\tau) = \left(\frac{1+\tau}{n}, \dots, \frac{1+\tau}{n}, -\tau \right)^T. \quad (20)$$

Tiếp sau bước lặp thứ k , các đỉnh chưa sắp xếp của đơn hình tiếp theo là các cột của ma trận $\Delta_k S_k$, trong đó S_k là ma trận cấp $(n+1) \times (n+1)$ cho bởi

$\begin{pmatrix} I_n & \frac{(1+\tau_k)}{n} \mathbf{e} \\ \mathbf{0}^T & -\tau_k \end{pmatrix}$ với bước kiểu τ , và bởi $\begin{pmatrix} 1 & (1-\sigma)\mathbf{e}^T \\ \mathbf{0} & \sigma I_n \end{pmatrix}$ với bước kiểu thu

hẹp, trong đó $\mathbf{0}$ là cột gồm n số không và I_n là ma trận đơn vị n chiều. Sau khi sắp xếp ở đầu bước lặp $k + 1$, các đỉnh của Δ_{k+1} thỏa mãn :

$$\Delta_{k+1} = \Delta_k T_k, \text{ với } T_k = S_k P_k. \quad (21)$$

trong đó P_k là ma trận được chọn để đảm bảo thực hiện các quy tắc sắp xếp và chèn.

Các tính chất của thuật toán Nelder - Mead.

Kết quả chung. Các tính chất sau trực tiếp suy ra từ định nghĩa của thuật toán Nelder – Mead.

1. Bước lặp Nelder – Mead đòi hỏi tính một giá trị hàm khi bước lặp kết thúc ở bước 2, hai giá trị hàm khi kết thúc xảy ra ở bước 3 hoặc bước 4, và $n + 2$ giá trị hàm nếu bước thu hẹp xảy ra.

2. Bước “ phản xạ” được gọi tên như vậy vì điểm phản xạ x^r là phản xạ của điểm xấu nhất x_{n+1} qua điểm $\bar{x}$ trên đường thẳng nối x_{n+1} và $\bar{x}$. Hệ số phản xạ thường chọn $\rho = 1$.

3. Với hàm tổng quát, bước thu hẹp có thể dẫn tới việc tăng giá trị hàm tại các đỉnh trừ đỉnh f_1 , tức là : $f_i^{(k+1)} > f_i^{(k)}, 2 \leq i \leq n + 1$.

4. Tại bước dẫn, trong bài báo gốc của Nelder – Mead [1], chấp nhận x_e nếu $f(x_e) < f_1$, ngược lại thì chấp nhận x_r . Tiêu chuẩn thường dùng hiện nay, chấp nhận điểm tốt nhất giữa x_r và x_e nếu cả hai đều tốt hơn x_1 .

Tính chất 1 (Thể tích và tính không suy biến của đơn hình Nelder – Mead).

a) Nếu đơn hình ban đầu Δ_0 không suy biến, thì tất cả đơn hình tiếp theo của thuật toán Nelder – Mead cũng không suy biến.

b) Sau bước không thu hẹp kiểu τ , $\text{vol}(\Delta_{k+1}) = |\tau| \text{vol}(\Delta_k)$.

c) Sau bước thu hẹp tại bước lặp k , $\text{vol}(\Delta_{k+1}) = \sigma^n \text{vol}(\Delta_k)$.

Tính chất 2. Cho f là hàm bị chặn dưới trên R^n , thuật toán Nelder Mead áp dụng để tìm cực tiểu của hàm f , bắt đầu với đơn hình không suy biến Δ_0 , khi đó

- a) Dãy $\{f_1^{(k)}\}$ luôn hội tụ.
- b) Tại mỗi bước lặp không thu hẹp k , $f_i^{(k+1)} \leq f_i^{(k)}, 1 \leq i \leq n+1$, trong đó bất đẳng thức chặt với ít nhất một giá trị.
- c) Nếu tồn tại chỉ một số hữu hạn các bước thu hẹp, khi đó :
 - (i) Mỗi dãy $\{f_i^{(k)}\}$ hội tụ khi $k \rightarrow \infty$ với $1 \leq i \leq n+1$,
 - (ii) $f_i^* \leq f_i^{(k)}$ với $1 \leq i \leq n+1$, và tất cả giá trị của k , trong đó $f_i^* = \lim_{k \rightarrow \infty} f_i^{(k)}$
 - (iii) $f_1^* \leq f_2^* \leq \dots \leq f_{n+1}^*$.
- d) Nếu tồn tại chỉ một số hữu hạn bước lặp không thu hẹp, khi đó tất cả các đỉnh của đơn hình hội tụ về một điểm duy nhất.

Tính chất 3 (Hội tụ yếu)

Giả sử hàm f bị chặn dưới trên R^n , thuật toán Nelder – Mead áp dụng với hàm f , bắt đầu với đơn hình không suy biến Δ_0 và không xảy ra bước lặp thu hẹp. Nếu có số nguyên $j, 1 \leq j \leq n$, thỏa mãn

$$f_j^* < f_{j+1}^* \text{ trong đó } f_j^* = \lim_{k \rightarrow \infty} f_j^{(k)} \quad (22)$$

thì tồn tại chỉ số lặp K sao cho $\forall k \geq K$, chỉ số thay đổi thỏa mãn $k^* > j$, tức là các đỉnh j đầu tiên của tất cả các đơn hình trở nên cố định sau bước lặp K .

Chứng minh

Ta chứng minh bằng phương pháp phản chứng. Từ giả thiết (22), ta có $\exists \delta > 0$ sao cho $f_j^* + \delta = f_{j+1}^*$. Chọn $\varepsilon > 0$ thỏa mãn $\delta - \varepsilon > 0$.

Vì $f_j^* = \lim_{k \rightarrow \infty} f_j^{(k)}$ nên $\exists K$ thỏa mãn $\forall k \geq K$, $f_j^{(k)} - \varepsilon \leq f_j^*$. Khi đó

$$\forall k \geq K, f_j^{(k)} < f_j^{(k)} - \varepsilon + \delta \leq f_j^* + \delta = f_{j+1}^*.$$

Mặt khác, từ Tính chất 2, phần c(ii) với bất kỳ chỉ số l , $f_{j+1}^* \leq f_{j+1}^{(l)}$. Khi đó

$\forall k \geq K$, và bất kỳ l ,

$$f_j^{(k)} < f_{j+1}^{(l)} \quad (23)$$

Nhưng nếu $k^* \leq j$ với bất kỳ $k \geq K$ thì khi dùng hệ thức thứ ba trong (14)

ta có $f_{j+1}^{((k+1))} = f_j^{(k)}$, điều này mâu thuẫn với (23). Suy ra $k^* > j$, $\forall k \geq K$.

Kết quả với hàm lồi chặt

Định nghĩa lồi chặt. Hàm f là hàm lồi chặt trên R^n nếu, với mỗi cặp điểm $\mathbf{y}, \mathbf{z} \in R^n$ với $\mathbf{y} \neq \mathbf{z}$ và với số λ thỏa mãn $0 < \lambda < 1$,

$$f(\lambda \mathbf{y} + (1 - \lambda) \mathbf{z}) < \lambda f(\mathbf{y}) + (1 - \lambda) f(\mathbf{z}). \quad (24)$$

Khi f là hàm lồi chặt trên R^n và

$$\mathbf{c} = \sum_{i=1}^l \lambda_i \mathbf{z}_i, \text{ với } 0 < \lambda_i < 1 \text{ và } \sum_{i=1}^l \lambda_i = 1, \text{ thì}$$

$$f(\mathbf{c}) < \sum_{i=1}^l \lambda_i f(\mathbf{z}_i) \text{ và hiển nhiên } f(\mathbf{c}) < \max \{ f(\mathbf{z}_1), \dots, f(\mathbf{z}_l) \}. \quad (25)$$

Tính chất 4. Giả sử f là hàm lồi chặt trên R^n và thuật toán Nelder – Mead áp dụng với hàm f , bắt đầu với đơn hình không suy biến Δ_0 , khi đó không xuất hiện bước thu hẹp.

Chứng minh

Bước thu hẹp xuất hiện chỉ nếu thuật toán Nelder- Mead gặp bước 4 và không chấp nhận điểm co.

$$\text{Khi } n = 1, f(\bar{\mathbf{x}}) = f_n.$$

$$\text{Khi } n > 1, \text{ áp dụng (25) với } \mathbf{x}_1, \dots, \mathbf{x}_n \text{ suy ra } f(\bar{\mathbf{x}}) < f_n.$$

Phép co ngoài được xét nếu $f_n < f_r < f_{n+1}$. Vì hệ số co γ thỏa mãn $0 < \gamma < 1$ và $x^c = \bar{x} + \gamma(x^r - \bar{x}) = \gamma x^r + (1 - \gamma)\bar{x}$, nên x^c là tổ hợp lồi của $\bar{x}$ và điểm phản xạ x^r .

Từ (25) suy ra $f(c) < \max \{ f(\bar{x}), f_r \}$. Ta biết rằng $f(\bar{x}) < f_n$ và $f_n < f(x^r)$, suy ra $\max \{ f(\bar{x}), f_r \} = f_r$. Vì vậy $f(c) < f_r$, điểm x^c được chấp nhận và bước thu hẹp không xảy ra.

Chúng minh tương tự với bước co trong, vì $f_{n+1} \leq f_r$ và x^{cc} là tổ hợp lồi của $\bar{x}$ và x_{n+1} .

Tính chất 5. Giả sử f là hàm lồi chặt và bị chặn dưới trên R^n . Nếu thêm vào tính chất $\rho > 0$ và $0 < \gamma < 1$, hệ số phản xạ ρ và hệ số co γ thỏa mãn $\rho\gamma < 1$, khi đó

a) $f_n^* = f_{n+1}^*$ và

b) tồn tại vô hạn bước lặp có $x_n^{(k+1)} \neq x_n^{(k)}$

Chứng minh

Chúng minh bằng phương pháp phản chứng. Giả sử $f_n^* < f_{n+1}^*$. Từ Tính chất 3, tồn tại bước lặp chỉ số K sao cho chỉ số thay đổi $k^* = n + 1$ với $k \geq K$. Không mất tính tổng quát, giả sử $K = 0$, vì

$k^* = n + 1$ với mọi $k \geq 0$, n đỉnh tốt nhất có giá trị là hằng số trong tất cả các bước lặp, như vậy điểm trọng tâm $\bar{x}^{(k)} = \bar{x}$ là véc tơ hằng số, và

$$f(x_n) = f_n^* = \lim_{k \rightarrow \infty} f_n^{(k)}.$$

Vì f là hàm lồi ngặt, $f(\bar{x}) \leq f(x_n) = f_n^*$, (bất đẳng thức này là chặt nếu $n > 1$)

Chỉ số thay đổi sẽ là $n + 1$ tại mỗi bước lặp chỉ nếu điểm co được chấp nhận và trở thành điểm xấu nhất mới.

Hơn nữa,

$$x_{n+1}^{(k+1)} = (1 + \rho\gamma)\bar{x} - \rho\gamma x_{n+1}^{(k)} \text{ hoặc } x_{n+1}^{(k+1)} = (1 - \gamma)\bar{x} + \gamma x_{n+1}^{(k)} \quad (26)$$

(co trong)

(co ngoài)

Dạng thuần nhất của các phương trình này là:

$$y_{n+1}^{(k+1)} = -\rho\gamma y_{n+1}^{(k)} \text{ hoặc } y_{n+1}^{(k+1)} = \gamma y_{n+1}^{(k)} \quad (27)$$

Vì $0 < \gamma < 1$ và $0 < \rho\gamma < 1$ nên ta có $\lim_{k \rightarrow \infty} y_{n+1}^{(k)} = 0$, như vậy lời giải của cả hai phương trình (27) đều là 0 khi $k \rightarrow \infty$

Bây giờ ta chỉ cần tìm một nghiệm riêng của phương trình không thuần nhất (26). Cả hai được thỏa mãn bởi véc tơ hằng số $\bar{x}$, như vậy lời giải tổng quát của chúng được cho bởi $x_{n+1}^{(k)} = y_{n+1}^{(k)} + \bar{x}$, trong đó $y_{n+1}^{(k)}$ thỏa mãn một trong các hệ thức (27).

Vì $\lim_{k \rightarrow \infty} y_{n+1}^{(k)} = 0 \Rightarrow \lim_{k \rightarrow \infty} x_{n+1}^{(k)} = x_{n+1}^* = \bar{x}$, với $f_{n+1}^* = f(\bar{x})$.

Nhưng ta biết từ đầu chứng minh $f(\bar{x}) \leq f_n^*$, điều đó có nghĩa rằng $f_{n+1}^* \leq f_n^*$. Theo Tính chất 2, phần c(iii) ta có $f_n^* \leq f_{n+1}^*$. Suy ra $f_n^* = f_{n+1}^*$, do vậy phần (a) của Tính chất 5 đã được chứng minh.

Kết quả (b) của Tính chất 5 được suy ra trực tiếp vì chúng ta luôn chỉ ra mâu thuẫn nếu tồn tại K sao cho $x_1^{(k)}, \dots, x_n^{(k)}$ là không đổi với $k \geq K$.

Tiếp theo ta xét hàm lồi ngặt f với tập mức bị chặn. Cho tập mức $\Gamma(f)$ được xác định bởi $\Gamma(f) = \{x: f(x) \leq \mu\}$. Hàm f có tập mức bị chặn nếu $\Gamma(f)$ bị chặn với mỗi giá trị μ , đây là thu hẹp để loại bỏ hàm lồi chặt như e^{-x} . Một điểm của sự thu hẹp này là hàm lồi chặt với tập mức bị chặn có duy nhất điểm cực tiểu x_{min} .

Thuật toán Nelder – Mead trong không gian 1 chiều với hàm lồi chặt.

Tính chất 6 (Sự hội tụ của thuật toán Nelder – Mead trong không gian một chiều).

Nếu f là hàm lồi chặt trên R^1 với các tập mức bị chặn. Giả sử thuật toán Nelder – Mead áp dụng cho hàm f với các tham số thỏa mãn

$$\rho > 0, \chi > 1, \chi > \rho, \rho\chi \geq 1, 0 < \gamma < 1$$

và bắt đầu với đơn hình không suy biến Δ_0 . Khi đó cả hai đầu mút của khoảng Nelder – Mead đều hội tụ tới x_{min} .

Tính chất 7 (khoảng bao của x_{min}). Cho f là hàm lồi chặt trên R^1 với các tập mức bị chặn. Giả sử thuật toán Nelder – Mead áp dụng cho hàm f bắt đầu với đơn hình xuất phát không suy biến Δ_0 và hệ số phản xạ và hệ số co thỏa mãn $\rho > 0, \chi > 1, \chi > \rho, \rho\chi \geq 1$. Khi đó tồn tại một số nguyên nhỏ nhất K thỏa mãn

$$K \leq \frac{|x_{min} - x_1^{(0)}|}{diam(\Delta_0)}, \text{ sao cho } f_2^{(K)} \geq f_1^{(K)} \text{ và } f_1^{(K)} \leq f_e^{(K)}.$$

Trong trường hợp này $x_{min} \in \text{int}(x_2^{(K)}, x_e^{(K)})$ và chúng ta nói rằng x_{min} được bao bởi $x_2^{(K)}$ và $x_e^{(K)}$.

Hội tụ tuyến tính với $\rho = 1$.

Khi hệ số phản xạ được chọn là $\rho = 1$, phương pháp Nelder – Mead không hội tụ duy nhất đến điểm cực tiểu, nhưng tốc độ hội tụ của nó là tập M tuyến tính. Khoảng cách từ đỉnh tốt nhất tới điểm tối ưu giảm với mỗi tập M theo ít nhất một cấp số nhân với hằng số nhỏ hơn 1.

Tính chất 8 (Sự hội tụ tuyến tính của Nelder – Mead trong không gian một chiều với $\rho = 1$).

Cho f là hàm lồi chặt và có các tập mức bị chặn trên R^1 . Giả sử thuật toán Nelder – Mead với hệ số phản xạ $\rho = 1$, hệ số giãn $\chi > 1$, hệ số co $0 < \gamma < 1$, được áp dụng với hàm f bắt đầu từ đơn hình xuất phát không suy biến Δ_0 . Khi đó tồn tại một số nguyên M chỉ phụ thuộc vào χ và γ sao cho

$$diam(\Delta_{k+M}) \leq \frac{1}{2} diam(\Delta_k) \text{ với mọi } k \geq K.$$

trong đó K là chỉ số bước lặp được định nghĩa trong Tính chất 7.

Tiêu chuẩn của Nelder – Mead trong không gian 2 chiều với hàm lồi chặt

Trong phần này, xét tiêu chuẩn của thuật toán Nelder – Mead với các hệ số $\rho = 1$, $\chi = 2$, $\gamma = \frac{1}{2}$ áp dụng với hàm lồi chặt $f(x)$ trong R^2 với tập mức bị chặn. Kí hiệu cực tiểu duy nhất của f bởi x_{min} và $f_{min} = f(x_{min})$. Chú ý rằng tập mức $\{x: f(x) \leq \mu\}$ là rỗng nếu $\mu < f_{min}$, là một điểm x_{min} nếu $\mu = f_{min}$, và là một tập lồi đóng nếu $\mu > f_{min}$.

Tính chất 9 (Sự hội tụ của các giá trị hàm tại các đỉnh với $n = 2$).

Cho f là hàm lồi chặt trên R^2 với các tập mức bị chặn. Giả sử thuật toán Nelder – Mead với hệ số phản xạ $\rho = 1$, và hệ số co $\gamma = \frac{1}{2}$, được áp dụng với hàm f bắt đầu từ đơn hình xuất phát không suy biến Δ_0 . Khi đó 3 giá trị hàm tại 3 đỉnh tiến tới giới hạn là bằng nhau, tức là

$$f_1^* = f_2^* = f_3^*.$$

Tính chất 10 (Sự hội tụ của bán kính đơn hình tới không).

Cho f là hàm lồi chặt trên R^2 với các tập mức bị chặn. Giả sử thuật toán Nelder – Mead với hệ số phản xạ $\rho = 1$, hệ số dẫn $\chi = 2$ và hệ số co $\gamma = \frac{1}{2}$, được áp dụng với hàm f bắt đầu từ đơn hình xuất phát không suy biến Δ_0 . Khi đó các đơn hình $\{\Delta_k\}$ sinh ra bởi thuật toán thỏa mãn

$$\lim_{k \rightarrow \infty} diam(\Delta_k) = 0.$$

6. Chương trình máy tính cho thuật toán Nelder – Mead

6.1. Bài toán

Chương trình dùng để giải bài toán cực tiểu không ràng buộc :

$$\min \{ f(x) : x \in R^n \}, \text{ trong đó hàm số } f : R^n \rightarrow R.$$

6.2. Các biến và mảng dùng trong chương trình

Hàm mục tiêu được viết trong hàm *double ham(double x[], int n)*.

Biến n là số chiều của không gian.

Mảng 2 chiều $d[51][50]$ chứa đơn hình ban đầu.

Mảng $f[51]$ chứa giá trị hàm mục tiêu tại các đỉnh của đơn hình.

Các mảng xn, xr, xe, xc, xcc ứng với các điểm $\bar{x}, x^r, x^e, x^c, x^{cc}$ trong thuật toán, giá trị hàm mục tiêu tương ứng tại các điểm này là $f^n, f^r, f^e, f^c, f^{cc}$.

Biến tg có ý nghĩa : $tg = 1$ thì có in kết quả trung gian ra tệp *ketqua.cpp*, nếu $tg = 0$ thì không in kết quả trung gian.

Biến sbl : thuật toán đang chạy ở bước thứ mấy.

Biến eps : xác định sai số để dừng thuật toán.

Biến k : độ lớn của đơn hình xuất phát.

6.3. Văn bản chương trình

```
#include <stdio.h>
#include <math.h>
#include <windows.h>
double ham(double x[], int n);
int main()
{ FILE *fp;
double d[51][50], f[51],xr[50], xe[50], xc[50], xcc[50], x[50], xn[50];
double fr, fe, fc, fcc, fn, eps, xichma,t ;
int i, j, k, n, truonghop, tt,tg,sbl;
system("cls");
//    Tao don hình ban đầu
tg = 0; eps = 0.0000001; k = 1; n = 4;
```

```
// thay doi
printf("\n n= %d k= %d eps= %13.8f ",n,k,eps);
if (tg==1) { fp=fopen("ketqua.cpp","wb");
            fprintf(fp,"\n n= %d k= %d eps= %13.8f ",n,k,eps);}
for (j= 1; j <= n; j++) d[1][j]= 2 ; // thay doi
for (i= 2; i <= n+1; i++)
    { for (j= 1; j <= n; j++) d[i][j]=d[1][j];
      d[i][i-1] = d[i][i-1]+k ; }
    // In don hình ban đầu
printf ("\n Don hình ban đầu : ") ;
for (i=1; i <= n+1; i++)
{ printf ("\n Hàng thu %d ",i);
for (j= 1; j <= n; j++) printf("%13.5f ", d[i][j]); }
if (tg==1)
{ fprintf (fp,"\n Don hình ban đầu : ") ;
for (i=1; i <= n+1; i++)
{ fprintf (fp,"\n Hàng thu %d ",i);
for (j= 1; j <= n; j++) fprintf(fp,"%13.5f ", d[i][j]); }
}
// Tính giá trị hàm mục tiêu cho don hình ban đầu
for ( i= 1;i <= n+1; i++)
    { for ( j =1; j <=n; j++) x[j] = d[i][j] ;
      f[i] = ham(x,n) ; }
printf ("\n Mang f ban đầu :") ;
for (i= 1; i <= n+1; i++) printf ("%13.5f ",f[i]) ;
if (tg==1) { fprintf (fp,"\n Mang f ban đầu :") ;
for (i= 1; i <= n+1; i++) fprintf (fp,"%13.5f ",f[i]) ; }
sbl=1;
// Sắp xếp theo f[i] tăng dần, theo theo mang n+1 dinh
Buoc1: sbl++;
printf("\nBuoc lap thu %d ",sbl);
```

```
if (tg==1) fprintf(fp,"\\nBuoc lap thu %d ",sbl);
for (i=1; i<=n; i++) for (j=i+1; j<=n+1;j++)
if (f[i]>f[j])
{ t=f[i]; f[i]=f[j]; f[j]=t;
  for (k=1; k<=n;k++) x[k]=d[i][k];
  for (k=1; k<=n;k++) d[i][k]=d[j][k];
  for (k=1; k<=n;k++) d[j][k]=x[k]; }
printf ("\\n Don hinh sau khi sap xep : ") ;
for (i=1; i <= n+1; i++)
{ printf ("\\n Hang thu %d ",i);
for (j= 1; j <= n; j++) printf("%13.5f ", d[i][j]); }
printf ("\\n Mang f saukhi sap xep tang dan :") ;
  for (i= 1; i <= n+1; i++) printf ("%13.5f ",f[i]) ;
if (tg==1) {
fprintf (fp,"\\n Don hinh sau khi sap xep : ") ;
for (i=1; i <= n+1; i++)
{ fprintf (fp,"\\n Hang thu %d ",i);
for (j= 1; j <= n; j++) fprintf(fp,"%13.5f ", d[i][j]); }
fprintf (fp,"\\n Mang f saukhi sap xep tang dan :") ;
for (i= 1; i <= n+1; i++) fprintf (fp,"%13.5f ",f[i]) ;
}
// Tinh diem giua cua n dinh dau tien trong d
for (j=1;j<=n;j++)
{ xn[j]=0; for (i=1;i<=n;i++) xn[j]=xn[j]+d[i][j];
  xn[j]=xn[j]/n; }
printf("\\nMang xn : ");
for (j= 1; j <= n; j++) printf("%13.5f ", xn[j]);
if (tg==1){
fprintf(fp,"\\nMang xn : ");
for (j= 1; j <= n; j++) fprintf(fp,"%13.5f ", xn[j]);
}
```

```
// Tinh xr = 2*xn - d[n+1]
for (j=1;j<=n;j++) xr[j]= 2*xn[j] - d[n+1][j];
fr= ham(xr,n);
printf("\nMang xr : ");
for (j= 1; j <= n; j++) printf("%13.5f ", xr[j]);
printf("\nfr = %13.5f",fr);
if (tg==1) {
    fprintf(fp,"\nMang xr : ");
    for (j= 1; j <= n; j++) fprintf(fp,"%13.5f ", xr[j]);
    fprintf(fp,"\nfr = %13.5f",fr);
}
if(f[1] <= fr and fr < f[n]) truonghop = 1 ;
if(fr < f[1]) truonghop = 2 ;
if(f[n] <= fr and fr < f[n+1]) truonghop = 3 ;
if(fr >= f[n+1]) truonghop = 4 ;
switch(truonghop)
{
case 1 : {
    printf("\nTruong hop 1: ");
    for( j= 1; j <= n; j++) d[n+1][j] = xr[j] ;
    f[n+1]=fr;
    printf ("\n Hang thu %d cua d ",n+1);
    for (j= 1; j <= n; j++) printf("%13.5f ", d[n+1][j]);
    printf("\nf[%d]= %13.5f",n+1,f[n+1]);
    if (tg==1){
        fprintf(fp,"\nTruong hop 1: ");
        fprintf (fp,"\n Hang thu %d cua d ",n+1);
        for (j= 1; j <= n; j++) fprintf(fp,"%13.5f ", d[n+1][j]);
        fprintf(fp,"\nf[%d]= %13.5f",n+1,f[n+1]);
    }
}
goto ktdung ;
```

```
        } break;
case 2 : {
printf("\nTruong hop 2: ");
for( j= 1; j <= n; j++) xe[j] = xn[j] + 2*(xr[j] - xn[j]) ;
fe = ham(xe,n) ;
printf("\nMang xe : ");
for( j= 1; j <= n; j++) printf("%13.5f ", xe[j]);
printf("\nfe = %13.5f",fe);
if (tg==1){ fprintf(fp,"\nTruong hop 2: ");
        fprintf(fp,"\nMang xe : ");
        for( j= 1; j <= n; j++) fprintf(fp,"%13.5f ", xe[j]);
        fprintf(fp,"\nfe = %13.5f",fe);
        }
if(fe < fr) {
    for(j= 1; j <=n; j++) d[n+1][j] = xe[j] ;
    f[n+1]=fe;
    printf ("\n Hang thu %d cua d ",n+1);
    for( j= 1; j <= n; j++) printf("%13.5f ", d[n+1][j]);
    printf("\nf[%d]= %13.5f",n+1,f[n+1]);
    if (tg==1){ fprintf (fp,"\n Hang thu %d cua d ",n+1);
        for( j= 1; j <= n; j++) fprintf(fp,"%13.5f ", d[n+1][j]);
        fprintf(fp,"\nf[%d]= %13.5f",n+1,f[n+1]);
        }
    goto ktdung ; }
else {
    for(j= 1; j <= n; j++) d[n+1][j] = xr[j];
    f[n+1]=fr;
    printf ("\n Hang thu %d cua d ",n+1);
    for( j= 1; j <= n; j++) printf("%13.5f ", d[n+1][j]);
    printf("\nf[%d]= %13.5f",n+1,f[n+1]);
    if (tg==1){ fprintf (fp,"\n Hang thu %d cua d ",n+1);
```

```
        for (j= 1; j <= n; j++) fprintf(fp,"%13.5f ", d[n+1][j]);
        fprintf(fp,"\nf[%d]= %13.5f",n+1,f[n+1]);
    }
    goto ktdung ; }
    } break;
case 3 : {
printf("\nTruong hop 3: ");
for ( j= 1; j <= n; j++) xc[j]= xn[j] + 0.5*(xr[j] - xn[j]) ;
fc = ham(xc,n) ;
printf("\nMang xc : ");
for (j= 1; j <= n; j++) printf("%13.5f ", xc[j]);
printf("\nfc = %13.5f",fc);
if (tg==1) { fprintf(fp,"\nTruong hop 3: ");
    fprintf(fp,"\nMang xc : ");
    for (j= 1; j <= n; j++) fprintf(fp,"%13.5f ", xc[j]);
    fprintf(fp,"\nfc = %13.5f",fc);
}
if (fc <= fr){ for (j= 1; j <= n; j++) d[n+1][j] = xc[j] ;
    f[n+1]=fc;
    printf ("\n Hang thu %d cua d ",n+1);
    for (j= 1; j <= n; j++) printf("%13.5f ", d[n+1][j]);
    printf("\nf[%d]= %13.5f",n+1,f[n+1]);
    if (tg==1){ fprintf (fp,"\n Hang thu %d cua d ",n+1);
        for (j= 1; j <= n; j++) fprintf(fp,"%13.5f ", d[n+1][j]);
        fprintf(fp,"\nf[%d]= %13.5f",n+1,f[n+1]);
    }
    goto ktdung ; }
else goto Buoc5 ;
    } break;
case 4 : {
printf("\nTruong hop 4: ");
```



```
for( j= 1; j <= n; j++)
xccc[j] = xn[j] - 0.5*(xn[j] - d[n+1][j]) ;
fcc = ham(xccc,n) ;
printf("\nMang xccc : ");
for (j= 1; j <= n; j++) printf("%13.5f ", xccc[j]);
printf("\nfcc = %13.5f",fcc);
if (tg==1){ fprintf(fp,"nTruong hop 4: ");
    fprintf(fp,"nMang xccc : ");
    for (j= 1; j <= n; j++)
        fprintf(fp,"%13.5f ", xccc[j]);
    fprintf(fp,"nfcc = %13.5f",fcc);
}
if (fcc < f[n+1])
{ for(j= 1 ; j<= n; j++) d[n+1][j]= xccc[j] ;
  f[n+1]=fcc;
  printf ("n Hang thu %d cua d ",n+1);
  for (j= 1; j <= n; j++) printf("%13.5f ", d[n+1][j]);
  printf("\nf[%d]= %13.5f",n+1,f[n+1]);
  if (tg==1){
      fprintf (fp,"n Hang thu %d cua d ",n+1);
      for (j= 1; j <= n; j++)
          fprintf(fp,"%13.5f ", d[n+1][j]);
      fprintf(fp,"nf[%d]= %13.5f",n+1,f[n+1]);
      }
      goto ktdung ; }
else goto Buoc5 ;
    } break;
} // end cua switch
// Thu hep don hinh di 1 nua
Buoc5: for( i= 2; i <= n+1; i++) for(j= 1; j<= n; j++)
    d[i][j] = d[1][j] + 0.5*(d[i][j] - d[1][j]);
```

```
printf("\nDon hình thu hep cac canh di 1 nua : ") ;
for (i=1; i <= n+1; i++)
{ printf ("\n Hang thu %d ",i);
for (j= 1; j <= n; j++) printf("%13.5f ", d[i][j]); }
if (tg==1) {
    fprintf (fp,"\nDon hình thu hep cac canh di 1 nua : ") ;
    for (i=1; i <= n+1; i++)
        { fprintf (fp,"\n Hang thu %d ",i);
          for (j= 1; j <= n; j++) fprintf(fp,"%13.5f ", d[i][j]); }
    }
    //
ktdung: printf("\nTinh sai so va kiem tra dieu kien dung");
fn =0;
for(i= 1;i <= n+1; i++) fn = fn + f[i] ;
fn = fn/(n+1) ;
printf("\nfn = %13.5f",fn);
if (tg==1) { fprintf(fp,"\nTinh sai so va kiem tra dieu kien dung");
    fprintf(fp,"\nfn = %13.5f",fn);
    }
xichma=0;
for(j= 1;j<= n+1; j++) xichma = xichma + (f[j] - fn)*(f[j] - fn) ;
xichma = xichma/n ;
xichma = sqrt(xichma) ;
printf("\nxichma = %13.5f",xichma);
if (tg==1) fprintf(fp,"\nxichma = %13.5f",xichma);
if(xichma > eps) goto Buoc1;
// { printf("\nSang buoc lap moi khong 1/0 : "); scanf("%d%c", &tt);
// if (tt==1) goto Buoc1 ; else goto ketthuc; }
else { printf("\nKET QUA GIAI BAI TOAN");
    printf("\nSo buoc lap : %d",sbl);
    printf("\nMang x toi uu : ");
```

```
for (j= 1; j <= n; j++) printf("%13.5f ", d[1][j]);
printf("\nTri toi uu = %13.5f",f[1]);
if (tg==1) {
    fprintf(fp,"\nKET QUA GIAI BAI TOAN");
    fprintf(fp,"\nSo buoc lap : %d",sbl);
    fprintf(fp,"\nMang x toi uu : ");
    for (j= 1; j <= n; j++) fprintf(fp,"%13.5f ", d[1][j]);
    fprintf(fp,"\nTri toi uu = %13.5f",f[1]);
}
}
ketthuc: printf("\n"); system("pause"); fclose(fp);
return 0;
}
double ham(double x[], int n)
{ double r, r1, r2, r3, r4;
r1= x[1]+10*x[2]; r1= r1*r1;
r2=x[3]-x[4]; r2=5*r2*r2;
r3=x[2]-2*x[3]; r3=r3*r3*r3*r3;
r4=x[1]-x[4]; r4=10*r4*r4*r4*r4;
r=r1+r2+r3+r4;
return r;
}
```

6.4. Giải ví dụ bằng số

Cực tiểu hàm Powell [1962]:

$$f(x) = (x_1 + 10x_2)^2 + 5(x_3 - x_4)^2 + (x_2 - 2x_3)^4 + 10(x_1 - x_4)^4.$$

Đây là một hàm rất khó cực tiểu.

Thuật toán Nelder – Mead giải ví dụ này qua các bước lặp :

$n = 4$ $k = 1$ $\text{eps} = 0.0000001$. Điểm xuất phát $U(3; -1; 0; 1)$.

Đơn hình ban đầu :

2	2	2	2.
3	2	2	2.

2 3 2 2.

2 2 3 2.

2 2 2 3.

Mảng f ban đầu :

500 555 1025 745 515.

Bước lặp thứ 1

Đơn hình sau khi sắp xếp theo giá trị hàm mục tiêu tăng dần :

2 2 2 2.

2 2 2 3.

3 2 2 2.

2 2 3 2.

2 3 2 2.

Mảng f sau khi sắp xếp tăng dần :

500 515 555 745 1025.

Mảng xn : 2.25 2 2.25 2.25.

Mảng xr : 2.5 1 2.5 2.5.

$fr = 412.25$.

Trường hợp 2:

Mảng xe : 2.75 0 2.75 2.75.

$fe = 922.625$.

Hàng thứ 5 của d thay bằng : 2.5 1 2.5 2.5.

$f[5] = 412.25$.

Tính sai số và kiểm tra điều kiện dừng:

$fn = 545.45$; Xichma = 123.1326.

Bước lặp thứ 2

Đơn hình sau khi sắp xếp theo giá trị hàm mục tiêu tăng dần :

1.5 0 1.5 1.5

1 1 1 1

1 1 1 2

2 1 1 1

1 1 2 1.

Mảng f sau khi sắp xếp tăng dần:

83.25 122 137 155 207.

Mảng xn : 1.375 0.75 1.125 1.375.

Mảng xr : 1.75 0.5 0.25 1.75.

$fr = 56.81250$.

Trường hợp 2:

Mảng xe : 2.125 0.25 - 0.625 2.125.

$fe = 64.26563$.

Hàng thứ 5 của d thay bằng : 1.75 0.5 0.25 1.75.

$f[5] = 56.8125$.

Tính sai số và kiểm tra điều kiện dừng:

$fn = 110.8125$; $xichma = 40.12223$.

Bước lặp thứ 3

Đơn hình sau khi sắp xếp theo giá trị hàm mục tiêu tăng dần :

2.5 1 2.5 2.5

2 2 2 2

2 2 2 3

3 2 2 2

2 2 3 2.

Mảng f sau khi sắp xếp tăng dần:

412.25 500 515 555 745.

Mảng xn : 2.375 1.75 2.125 2.375.

Mảng xr : 2.75 1.5 1.25 2.75.

$fr = 327.3125$.

Trường hợp 2:

Mảng xe : 3.125 1.25 0.375 3.125.

$fe = 282.01563$.

Hàng thứ 5 của d thay bằng : 3.125 1.25 0.375 3.125.

$f[5] = 282.01563$

Tính sai số và kiểm tra điều kiện dừng:

$fn = 452.85312$; $xichma = 108.80385$.

.....
Bước lặp thứ 146

Đơn hình :

– 0.00018	0.00002	– 0.00982	– 0.00965.
– 0.00618	0.00066	– 0.01116	– 0.01114
0.00581	– 0.00063	– 0.00588	– 0.00589
0.00125	– 0.00015	– 0.0107	– 0.01074
0.00067	– 0.00014	– 0.00657	– 0.00642.

Hàm f : 0 0 0 0 0.

xn : 0.00017 – 0.00003 – 0.00939 – 0.00935.

xr : – 0.00032 0.00009 – 0.01221 – 0.01229.

$fr = 0$.

Trường hợp 4:

xcc : 0.00042 – 0.00009 – 0.00798 – 0.00789.

Hàng thứ 5 của d thay bằng: 0.00042 – 0.00009 – 0.00798 – 0.00789.

$f[5] = 0$.

Tính sai số và kiểm tra điều kiện dừng:

$fn = 0$; $xichma = 0$.

Kết quả giải bài toán

Số bước lặp: 146.

x tối ưu: (– 0.00018 ; 0.00002; – 0.00982 ; – 0.00965).

Giá trị tối ưu: $f = 0$.

6.5. Các ví dụ đã chạy chương trình

Ví dụ 1. Cực tiểu hàm Rosen brock [1960]

$$f(x_1, x_2) = 100(x_2 - x_1)^2 + 5(1 - x_1)^2.$$

Thuật toán Nelder – Mead giải ví dụ này qua các bước lặp với :

$n = 2$; $k = 1$; $\text{eps} = 0.0000001$. Điểm xuất phát $U(-1.2 ; 1)$.

Kết quả giải bài toán

Số bước lặp: 54.

x tối ưu: (1.000006 ; 1.00016).

Giá trị tối ưu: $f = 0$.

Ví dụ 2. Cực tiểu hàm mũ hai biến

$$f(x_1, x_2) = \sum_{\alpha} [(e^{-\alpha x_1} - e^{-\alpha x_2}) - (e^{-\alpha} - e^{-10\alpha})]^2$$

Với $\alpha = 0.1, 0.2, \dots, 1.0$.

Thuật toán Nelder – Mead giải ví dụ này qua các bước lặp với :

$n = 2$; $k = 1$; $\text{eps} = 0.00000001$. Điểm xuất phát $u = (3; 2)$.

Kết quả giải bài toán

Số bước lặp: 36.

x tối ưu: (0.99997; 10.00198).

Giá trị tối ưu: $f = 0$.

Ví dụ 3. Cực tiểu hàm Fletcher và Powell [1963]

$$f(x_1; x_2; x_3) = 100[x_3 - 10\theta(x_1; x_2)]^2 + \left[\sqrt{(x_1^2 + x_2^2)} - 1 \right]^2 + x_3^2$$

trong đó

$$2\pi\theta(x_1; x_2) = \begin{cases} \arctan\left(\frac{x_2}{x_1}\right), & x_1 > 0. \\ \pi + \arctan\left(\frac{x_2}{x_1}\right), & x_1 < 0. \end{cases}$$

Thuật toán Nelder – Mead giải ví dụ này qua các bước lặp với :

$n = 3$; $k = 1$; $\text{eps} = 0.00000001$. Điểm xuất phát $U(-1; 0; 0)$.

Kết quả giải bài toán

Số bước lặp: 122.

x tối ưu: (0.99993 ; - 0.00008 ; - 0.00011).

Giá trị tối ưu: $f = 0$.

Ví dụ 4. Cực tiểu hàm Griewank

$$f(x_1, \dots, x_n) = \frac{1}{400n} \sum_{i=1}^n x_i^2 - \prod_{i=1}^n \cos\left(\frac{x_i}{\sqrt{i}}\right)$$

trong đó: $n = 12$, $x_i \in [-1000, 1000]$.

Thuật toán Nelder – Mead giải ví dụ này qua các bước lặp với :

$n = 12$; $k = 1$; $\text{eps} = 0.00001$;

Điểm xuất phát: $u = (1; 1; 1; 1; 1; 1; 1; 1; 1; 1; 1; 1)$.

Kết quả giải bài toán

Số bước lặp: 336.

x tối ưu: (0,00049 ; - 0.00219 ; - 0.00266; 0.00398; - 0.0034; 0,0129;
0.01423; 0.00145; 0.00579; 0.00014; - 0.00953; - 0.02008).

Giá trị tối ưu: $f = -1$.

Ví dụ 5. Cực tiểu hàm có ràng buộc

$$\begin{aligned} f(x_1; x_2; x_3) = & 2x_1^2 + x_2^2 + 3x_3^2 - x_1x_2 - 2x_1x_3 - x_2x_3 + 10x_1 \\ & + 9x_2 - 26x_3. \end{aligned} \quad (*)$$

thỏa mãn $g_1(x) = x_1 - 1 \leq 0$,

$$g_2(x) = -x_2 \leq 0, \quad g_4(x) = x_1 + x_2 + x_3 - 4 \leq 0,$$

$$g_3(x) = -x_3 \leq 0, \quad g_5(x) = x_1 - 2x_2 - x_3 + 3 \leq 0.$$

Sử dụng phương pháp hàm phạt đưa bài toán (*) về giải bài toán cực tiểu không ràng buộc : $F(x_1; x_2; x_3) = f(x_1; x_2; x_3) + M \sum_{i=1}^5 [\max\{g_i(x); 0\}]^2$.

Thuật toán Nelder – Mead giải ví dụ này qua các bước lặp với :

$\text{eps} = 0.0000001$; $k = 1$; $n = 3$; $M = 1000000$. Điểm xuất phát $u = (2; 2; 2)$.

Kết quả giải bài toán

Số bước lặp: 127.

x tối ưu: $(-0.40034 ; 0 ; 4.20006) ;$

Giá trị tối ưu: $f = -56.60001.$

Chương 2

PHƯƠNG PHÁP TÌM KIẾM TRỰC TIẾP

HOOKE – JEEVES

Phương pháp tìm kiếm trực tiếp do Hooke – Jeeves đưa ra vào năm 1960. Nội dung của thuật toán được trình bày dựa trên tài liệu [7] và [8].

1. Mô tả thuật toán Hooke - Jeeves trong không gian R^n

1.1. Phát biểu bài toán

Bài toán đặt ra là cần tìm cực tiểu không ràng buộc của hàm n biến:

$$\min \{ f(x) : x \in R^n \}, \text{ trong đó hàm số } f : R^n \rightarrow R.$$

Kí hiệu x^t là phương án tốt nhất hiện biết với giá trị hàm mục tiêu tốt nhất hiện biết là f^t . Lúc bắt đầu thuật toán x^t lấy chính là điểm xuất phát của thuật toán.

1.2. Tư tưởng cơ bản của thuật toán

Thủ tục cơ bản của thuật toán Hooke – Jeeves là thủ tục dò tìm địa phương: xuất phát từ một điểm $x \in R^n$ theo $2n$ hướng dọc theo n trục tọa độ với độ dài bước h tìm điểm x' có giá trị hàm mục tiêu $f(x')$ tốt hơn $f(x)$. Mô tả chi tiết thủ tục dò tìm địa phương:

Bước 0. Có điểm x , bước dịch chuyển h , giá trị tốt nhất hàm mục tiêu hiện biết f^t ứng với phương án x^t . Biến $change = 0$ (biến $change$ để xác định gốc dò tìm có thay đổi không).

Bước 1. Xét tọa độ $j = 1$.

Bước 2. Tính điểm $z = x + he^j = (x_1, \dots, x_{j-1}, x_j + h, x_{j+1}, \dots, x_n)$, trong đó e^j là véc tơ đơn vị thứ j , $e^j = (0, \dots, 0, 1, 0, \dots, 0)$, số 1 ở vị trí j . Tính $t = f(z)$.

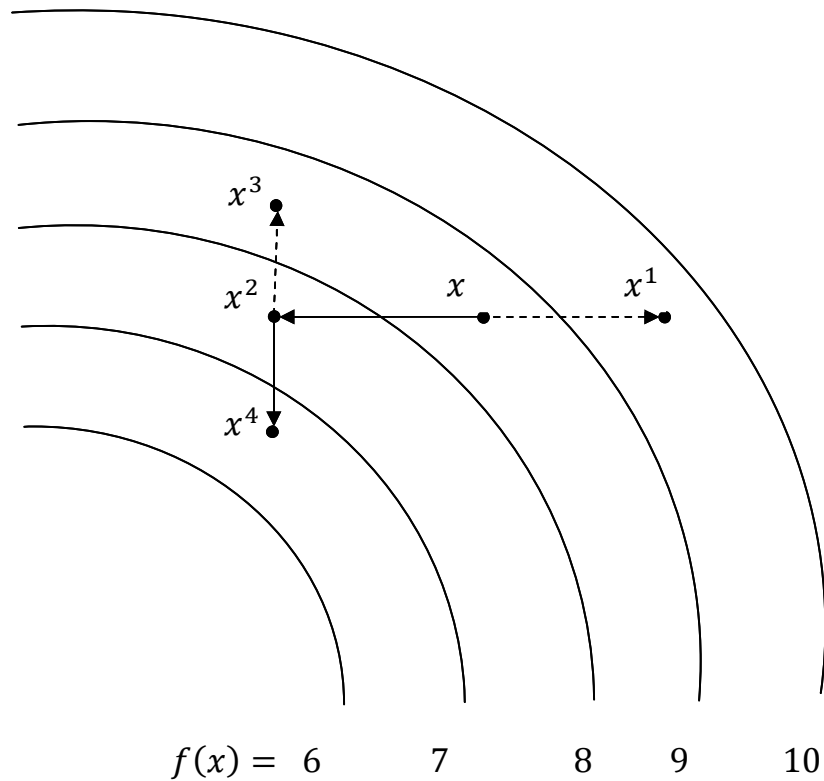
Bước 3. Nếu $t < f^t$ thì $f^t \leftarrow t$, $x \leftarrow z$, $change = change + 1$, và sang bước 6.

Bước 4. Tính điểm $z = x - he^j = (x_1, \dots, x_{j-1}, x_j - h, x_{j+1}, \dots, x_n)$, $t = f(z)$.

Bước 5. Nếu $t < f^t$ thì $f^t \leftarrow t$, $x \leftarrow z$, $change = change + 1$.

Bước 6. $j = j + 1$. Nếu $j \leq n$ thì chuyển lên bước 2.

Sau thủ tục trên nếu $change = 0$ thì ta không tìm được điểm x' có giá trị hàm mục tiêu tốt hơn điểm xuất phát x . Nếu $change > 0$ thì ta tìm được điểm x' (giá trị véc tơ này chứa trong biến x và x^t) có giá trị hàm mục tiêu tốt hơn điểm xuất phát x , giá trị tốt nhất hàm mục tiêu f^t và x^t được cập nhật tốt hơn.



Hình 9

Hình 9 minh họa cho thủ tục dò tìm địa phương trong không gian 2 chiều. Các đường cong biểu thị đường mức của hàm mục tiêu $f(x)$. Việc dò tìm xuất phát từ điểm $x = (x_1, x_2)$. Theo trục tọa độ e^1 ta xét điểm $x^1 = (x_1 + h, x_2)$, $f(x^1) > f(x)$. Xét điểm $x^2 = (x_1 - h, x_2)$, $f(x^2) < f(x)$, $f^t \leftarrow f(x^2)$, chuyển gốc dò tìm tới $x^2 = (x_1^2, x_2^2)$. Theo trục tọa độ e^2 ta xét điểm $x^3 = (x_1^2, x_2^2 + h)$, $f(x^3) > f(x^2)$. Xét điểm $x^4 = (x_1^2, x_2^2 - h)$, $f(x^4) < f(x^2)$, $f^t \leftarrow f(x^4)$. Kết quả được x^4 tốt hơn x ban đầu.

1.3. Mô tả một bước lặp của thuật toán Hooke – Jeeves

Thuật toán Hooke – Jeeves phát biểu chi tiết như sau:

Bước 0. Cho điểm xuất phát $u \in R^n$, cho bước dịch chuyển theo trục tọa độ h , cho hệ số giảm của h là α , cho số ε đủ nhỏ. Tính giá trị hàm mục tiêu tại điểm xuất phát $f(u)$, $f^t \leftarrow f(u)$, $x^t \leftarrow u$.

Bước 1. $x \leftarrow u$. Tiến hành thủ tục dò tìm địa phương theo điểm gốc x . Nếu $change > 0$ thì $v \leftarrow x$, chuyển sang Bước 3.

Bước 2. Trường hợp kết quả dò tìm có điểm gốc không thay đổi. Nếu $h < \varepsilon$ thì dừng thuật toán, in giá trị tối ưu f^t và phương án tối ưu x^t , trái lại ta giảm bước dịch chuyển h : $h \leftarrow h * \alpha$ và chuyển lên Bước 1.

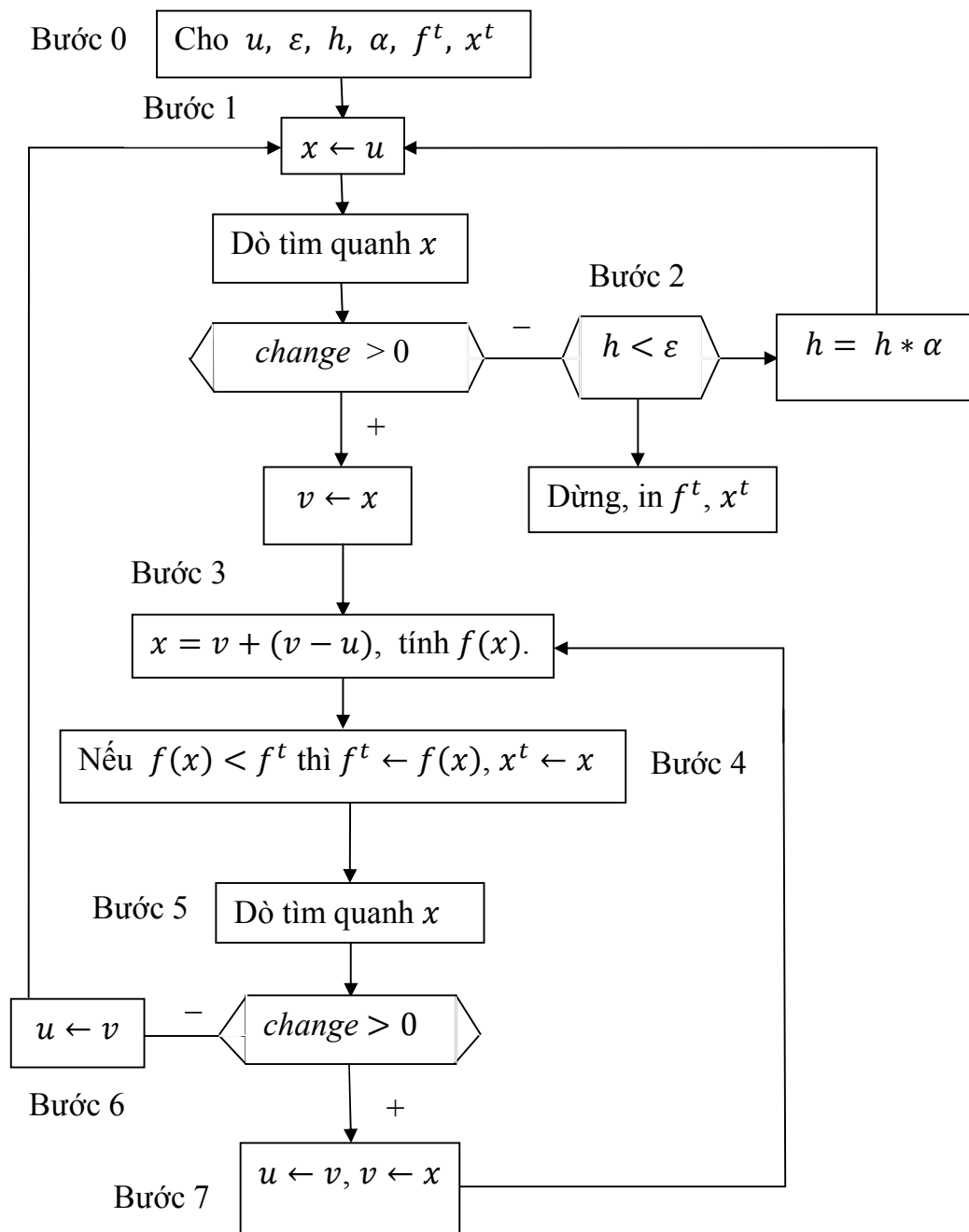
Bước 3. Xác định điểm $x = v + (v - u)$, tính $f(x)$.

Bước 4. Nếu $f(x) < f^t$ thì $f^t \leftarrow f(x)$, $x^t \leftarrow x$.

Bước 5. Thực hiện thủ tục dò tìm địa phương theo điểm gốc x . Nếu $change > 0$ thì chuyển sang Bước 7.

Bước 6. Trường hợp kết quả dò tìm có điểm gốc không thay đổi. Tiến hành $u \leftarrow v$, chuyển lên Bước 1.

Bước 7. Tiến hành $u \leftarrow v$, $v \leftarrow x$ và chuyển lên Bước 3.



Hình 10. Sơ đồ khối của thuật toán Hooke – Jeeves.

Chú ý kí hiệu “+” trong hình để chỉ điều kiện được thỏa mãn, dấu “-” chỉ điều kiện không thỏa mãn.

2. Ví dụ minh họa cho thuật toán Hooke – Jeeves trong không gian R^2

Dùng phương pháp Hooke – Jeeves tìm cực tiểu hàm

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

Chọn điểm xuất phát ban đầu $u^1 = (1; 1)$, độ dài bước $h = 1$, $\varepsilon = 10^{-3}$, hệ số giảm $\alpha = 10^{-1}$. Giá trị hàm mục tiêu tại điểm xuất phát $f(u^1) = 10$.

Giá trị tốt nhất hiện biết: $f^t = f(u^1) = 10$, với phương án $x^t = u^1 = (1; 1)$.

Bước lặp thứ 1

Dò tìm quanh điểm cơ sở (thực hiện theo bước 1). Đặt $x = u^1$.

Xét trục tọa độ e^1 . $x^1 = x + he^1 = (2; 1)$, $f(x^1) = 5 < f^t = f(u^1) = 10$.

Đặt $x = x^1 = (2; 1)$. Giá trị tốt nhất mới $f^t = f(x^1) = 5$.

Xét trục tọa độ e^2 . $x^2 = x + he^2 = (2; 2)$, $f(x^2) = 4 < f^t = 5$.

Đặt $u^2 = x^2 = (2; 2)$. Giá trị tốt nhất mới $f^t = f(x^2) = 4$.

Dò tìm theo mẫu (thực hiện theo bước 3 $\rightarrow$ 7).

Xác định điểm mẫu $x = u^2 + (u^2 - u^1) = (3; 3)$, $f(x) = 2 < f^t = 4$.

Giá trị tốt nhất mới $f^t = f(x) = 2$.

Xét trục tọa độ e^1 . $x^1 = x + he^1 = (4; 3)$, $f(x^1) = 1 < f^t = 2$.

Đặt $x = x^1 = (4; 3)$. Giá trị tốt nhất mới $f^t = f(x^1) = 1$.

Xét trục tọa độ e^2 . $x^2 = x + he^2 = (4; 4)$, $f(x^2) = 4 > f^t = 1$.

$$x^3 = x - he^2 = (4; 2), f(x^3) = 0 < f^t = 1.$$

Đặt $u^3 = x^3 = (4; 2)$. Giá trị tốt nhất mới $f^t = f(x^3) = 0$.

Dò tìm theo mẫu (tiếp).

Với $u^1 \leftarrow u^2 = (2; 2)$, $u^2 \leftarrow u^3 = (4; 2)$.

Xác định điểm mẫu $x = u^2 + (u^2 - u^1) = (6; 2)$, $f(x) = 4 > f^t = 0$.

Xét trục tọa độ e^1 và e^2 . $x^1 = x + he^1 = (7; 2)$, $f(x^1) = 9 > f^t = 0$.

$$x^2 = x - he^1 = (5; 2), \quad f(x^2) = 1 > f^t = 0.$$

$$x^3 = x + he^2 = (6; 3), \quad f(x^3) = 5 > f^t = 0.$$

$$x^4 = x - he^2 = (6; 1), \quad f(x^4) = 5 > f^t = 0.$$

Kết quả dò tìm có điểm gốc không thay đổi. Dừng tìm kiếm theo mẫu từ $u^2 = (4; 2)$. Đặt $u^1 \leftarrow u^2 = (4; 2)$.

Bước lặp thứ 2:

Dò tìm quanh điểm cơ sở. Đặt $x = u^1$.

Xét trục tọa độ e^1 và e^2 . $x^1 = x + he^1 = (5; 2)$, $f(x^1) = 1 > f^t = 0$.

$$x^2 = x - he^1 = (3; 2), \quad f(x^2) = 1 > f^t = 0.$$

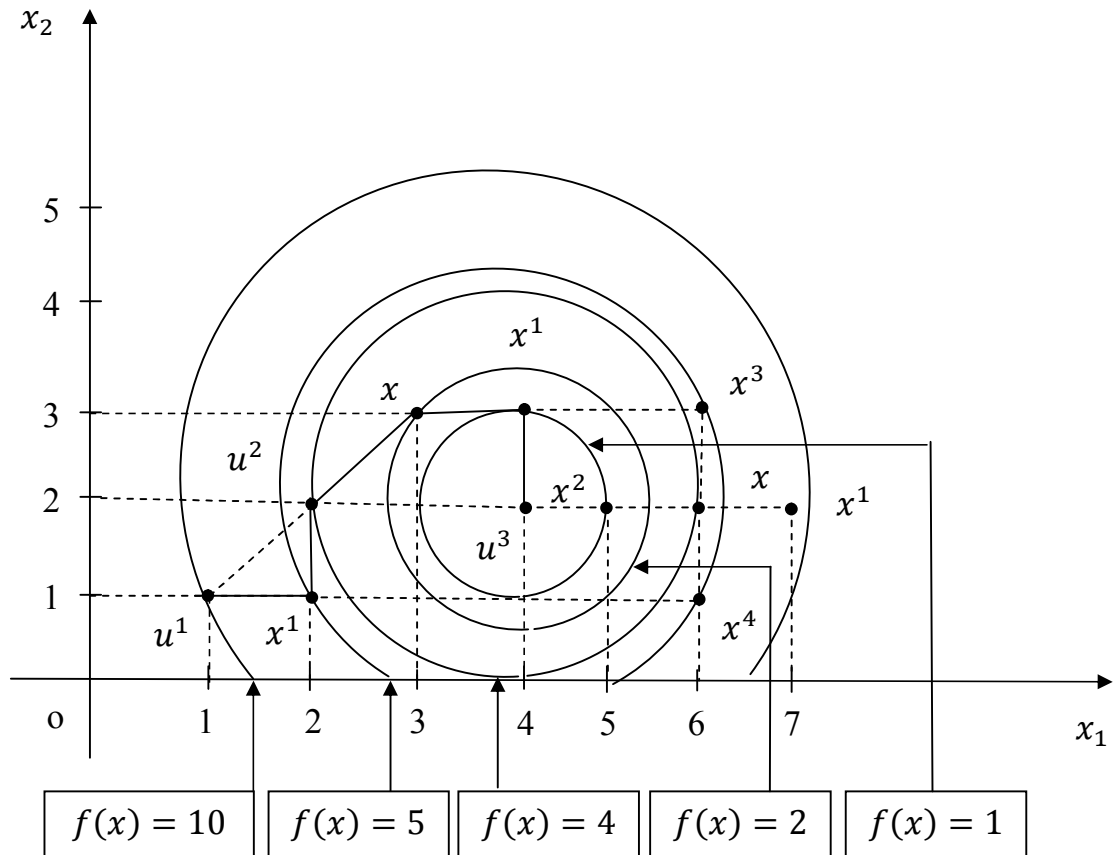
$$x^3 = x + he^2 = (4; 3), \quad f(x^3) = 1 > f^t = 0.$$

$$x^4 = x - he^2 = (4; 1), \quad f(x^4) = 1 > f^t = 0.$$

Kết quả dò tìm có điểm gốc không thay đổi. Dừng tìm kiếm quanh điểm cơ sở từ $u^1 = (4; 2)$. Do không tìm được điểm mới tốt hơn nên tiếp tục dò tìm quanh $u^1 = (4; 2)$ với độ dài bước $h := h * \alpha = 0.1$ (giảm 10 so với trước). Sau 3 bước nữa ta vẫn chỉ dò tìm quanh $u^1 = (4; 2)$. Với h giảm dần tới $h = 0.001 = \varepsilon$ thì dừng tính toán và kết luận:

Kết quả giải bài toán

Cực tiểu của hàm $f(x_1, x_2) = (x_1 - 4)^2 + (x_2 - 2)^2$ là $f_{min} = f^t = 0$ và điểm cực tiểu $x_{min} = x^t = (4; 2)$. Số lần thay đổi hàm mục tiêu là 5 (xem Hình 11).



Hình 11. Minh họa một bước lặp của thuật toán Hooke – Jeeves tìm cực tiểu của hàm : $f(x_1, x_2) = (x_1 - 4)^2 + (x_2 - 2)^2$.

3. Chương trình máy tính cho thuật toán Hooke - Jeeves

3.1. Bài toán

Chương trình dùng để giải bài toán cực tiểu không ràng buộc :

$$\min \{ f(x) : x \in R^n \}, \text{ trong đó hàm số } f : R^n \rightarrow R.$$

3.2. Các biến và mảng dùng trong chương trình

Hàm mục tiêu được viết trong hàm *double ham(double x[], int n)*.

Biến *change* : dò tìm có thay đổi phương án tốt nhất không. Nếu *change* = 0 thì dò tìm không đạt kết quả. Nếu *change* > 0 thì có thay đổi phương án tốt nhất.

sumchange : tổng số lần thay đổi phương án tốt nhất.

inter : tổng số bước lặp.

ft: giá trị hàm tốt nhất hiện biết.

h : độ dài bước dò tìm.

n : số chiều của không gian.

Biến *tg* có ý nghĩa : $tg = 1$ thì có in kết quả trung gian ra tệp *ketqua.cpp*,

nếu $tg = 0$ thì không in kết quả trung gian.

xt : phương án tốt nhất hiện biết.

x : điểm dò tìm.

Biến *eps* : xác định sai số để dừng thuật toán.

Biến *giam* : dùng để giảm độ dài bước *h* sau mỗi bước lặp, $h = h * giam$.

3.3. Văn bản chương trình

```
#include <stdio.h>
```

```
#include <math.h>
```

```
#include <windows.h>
```

```
#include <dos.h>
```

```
#define N 50
```

```
double ham(double x[], int n);
```

```
void search();
```

```
double ft,t,eps,h,giam,vocung, xt[N],u[N],v[N],x[N];
```

```
int i,j,n,tg,dung;
```

```
long int change,sumchange,dotimcoso,dotimv,inter;
```

```
FILE *f;
```

```
int main()
```

```
{
tg=1; dung=0; n=3; h=1; eps=0.00000001; u[1]=8.0; u[2]=9.0;u[3]=7.0;
giam=0.1; vocung=1000000.0;

system("cls"); sumchange=0; dotimcoso=0; dotimv=0; inter = 0;

if (tg==1) f= fopen("ketqua2.cpp","w");
printf("\n n = %d , h = %13.5f , eps = %13.5f",n,h,eps);
if (tg==1) fprintf(f,"\n n = %d , h = %13.5f , eps = %13.5f",n,h,eps);
ft = ham(u,n);
for (i=1;i<=n;i++) xt[i]=u[i];
printf("\nTri tot nhat: %13.5f - Phuong an: ",ft);
for (i=1;i<=n;i++) printf("%13.5f ",xt[i]);
if (tg==1) { fprintf(f,"\nTri tot nhat: %13.5f - Phuong an: ",ft);
            for (i=1;i<=n;i++) fprintf(f,"%13.5f ",xt[i]);
        }
lap1: inter++;
printf("\n\nBUOC LAP THU %ld",inter);
printf("\nDo dai buoc h = %13.5f",h);
printf("\n\nDo tim theo DIEM CO SO ");
if (tg==1) { fprintf(f,"\n\nBUOC LAP THU %ld ",inter);
            fprintf(f,"\nDo dai buoc h = %13.5f",h);
            fprintf(f,"\n\nDo tim theo DIEM CO SO ");
        }
for (i=1;i<=n;i++) x[i]=u[i]; dotimcoso++;
search(); sumchange += change;
if (change==0) {
    if (h<eps or ft<-vocung) {
        printf("\n KET QUA GIAI BAI TOAN ");
```

```
printf("\nSo buoc lap : %ld",inter);
printf("\nTri toi uu: %13.5f ",ft);
printf("\nPhuong an toi uu: ");
for (i=1;i<=n;i++) printf("%13.5f ",xt[i]);
printf("\nSo lan thay doi ham muc tieu : %ld",sumchange);
printf("\nSo lan do tim co so : %ld",dotimcoso);
printf("\nSo lan do tim theo huong giam : %ld",dotimv);
if (tg==1){ fprintf(f,"\n KET QUA GIAI BAI TOAN");
    fprintf(f,"\nSo buoc lap : %ld",inter);
    fprintf(f,"\nTri toi uu: %13.5f ",ft);
    fprintf(f,"\nPhuong an toi uu: ");
    for (i=1;i<=n;i++) fprintf(f,"%13.5f ",xt[i]);
    fprintf(f,"\nSo lan thay doi ham muc tieu : %ld",sumchange);
    fprintf(f,"\nSo lan do tim co so : %ld",dotimcoso);
    fprintf(f,"\nSo lan do tim theo huong giam : %ld",dotimv);
}
printf("\n"); system("pause"); fclose(f);
return 0; }
else { h=h*giam; goto lap1; } };
for (i=1;i<=n;i++) v[i]=x[i];
lap2: printf("\n\nDo tim theo DIEM MAU \n");
if (tg==1) fprintf(f,"\n\nDo tim theo DIEM MAU \n");
for (i=1;i<=n;i++) x[i]=v[i]+ (v[i]-u[i]);
t=ham(x,n);
printf("\n t = %13.5f va diem mau x : ",t);
for (i=1;i<=n;i++) printf("%13.5f ",x[i]);
```

```
if (tg==1) { fprintf(f,"\\n t = %13.5f va diem mau x : ",t);
            for (i=1;i<=n;i++) fprintf(f,"%13.5f ",x[i]); }
if (t < ft) { ft=t; sumchange++; for (i=1;i<=n;i++) xt[i]=x[i];
            printf("\\nThay tri tot nhat: %13.5f - phuong an: ",ft);
            for (i=1;i<=n;i++) printf("%13.5f ",xt[i]);
            if (tg==1) { fprintf(f,"\\nThay tri tot nhat: %13.5f - phuong an: ",ft);
                        for (i=1;i<=n;i++) fprintf(f,"%13.5f ",xt[i]); }
            }
dotimv++; search(); sumchange += change;
if (change==0) { for (i=1;i<=n;i++) u[i]=v[i]; goto lap1; }
else { for (i=1;i<=n;i++) u[i]=v[i];
        for (i=1;i<=n;i++) v[i]=x[i];
        goto lap2; }
}
void search()
{ change=0;
printf("\\n\\nVec to x truoc khi do tim: ");
for (i=1;i<=n;i++) printf("%13.5f ",x[i]);
if (tg==1){ fprintf(f,"\\n\\nVec to x truoc khi do tim: ");
            for (i=1;i<=n;i++) fprintf(f,"%13.5f ",x[i]); }
for (j=1; j<=n; j++)
{ printf("\\n\\n----- Xet toa do j = %d : ",j);
  if (tg==1) fprintf(f,"\\n\\n----- Xet toa do j = %d : ",j);
  x[j]=x[j]+h; t=ham(x,n);
  printf("\\nt = %13.5f va x + h: ",t);
  for (i=1;i<=n;i++) printf("%13.5f ",x[i]);
```

```
if (tg==1) { fprintf(f,"nt = %13.5f va x + h: ",t);
            for (i=1;i<=n;i++) fprintf(f,"%13.5f ",x[i]); }
if (t < ft) { ft=t; change++; for (i=1;i<=n;i++) xt[i]=x[i];
            printf("\nThay Tri tot nhat: %13.5f - Phuong an: ",ft);
            for (i=1;i<=n;i++) printf("%13.5f ",xt[i]);
            if (tg==1) { fprintf(f,"\nThay Tri tot nhat: %13.5f - Phuong an: ",ft);
                for (i=1;i<=n;i++) fprintf(f,"%13.5f ",xt[i]); }
            goto jtiep; }
else x[j]=x[j]-h;
x[j]=x[j]-h; t=ham(x,n);
printf("nt = %13.5f va x - h: ",t);
for (i=1;i<=n;i++) printf("%13.5f ",x[i]);
if (tg==1) { fprintf(f,"nt = %13.5f va x - h: ",t);
            for (i=1;i<=n;i++) fprintf(f,"%13.5f ",x[i]); }
if (t < ft) { ft=t; change++; for (i=1;i<=n;i++) xt[i]=x[i];
            printf("\nThay Tri tot nhat: %13.5f - Phuong an: ",ft);
            for (i=1;i<=n;i++) printf("%13.5f ",xt[i]);
            if (tg==1){ fprintf(f,"\nThay Tri tot nhat: %13.5f - Phuong an: ",ft);
                for (i=1;i<=n;i++) fprintf(f,"%13.5f ",xt[i]); }
            goto jtiep; }
else x[j]=x[j]+h;
jtiep: printf("\n\nVec to x sau j = %d: ",j);
        for (i=1;i<=n;i++) printf("%13.5f ",x[i]);
        if (tg==1){ fprintf(f,"\n\nVec to x sau j = %d: ",j);
            for (i=1;i<=n;i++) fprintf(f,"%13.5f ",x[i]); }
    }
```

```
printf("\n\nTri tot nhat sau search: %13.5f - Phuong an: ",ft);
for (i=1;i<=n;i++) printf("%13.5f ",xt[i]);
if (tg==1) { fprintf(f,"\n\nTri tot nhat sau search: %13.5f - Phuong an: ",ft);
            for (i=1;i<=n;i++) fprintf(f,"%13.5f ",xt[i]);}
if (dung==1) { printf("\n"); system("pause");}
}

double ham(double x[], int n)
{ double r;
tg= 1;
r= 4*(x[1]-5)*(x[1]-5)+(x[2]-6)*(x[2]-6)+3*(x[3]-4)*(x[3]-4);
return r;
}
```

3.4. Giải ví dụ bằng số

Cực tiểu hàm số sau:

$$f(x_1; x_2; x_3) = 4(x_1 - 5)^2 + (x_2 - 6)^2 + 3(x_3 - 4)^2.$$

Thuật toán Hooke – Jeeves giải ví dụ này qua các bước lặp:

tg = 1, dung = 0, n = 3, h = 1, eps = 0.0000001, giam = 0.1,
vocung = 1000000.

Giá trị tốt nhất: 72; Phương án: 8 9 7.

Bước lặp thứ 1

Độ dài bước h = 1.

Dò tìm theo điểm cơ sở

Véc tơ x trước khi dò tìm: 8 9 7.

Xét tọa độ j = 1 :

t = 100 và x + h: 9 9 7.

t = 52 và x - h: 7 9 7.

Thay giá trị tốt nhất: 52; Phương án: 7 9 7.

Véc tơ x sau j = 1: 7 9 7.

Xét tọa độ $j = 2$:

$t = 59$ và $x + h$: 7 10 7.

$t = 47$ và $x - h$: 7 8 7.

Thay giá trị tốt nhất: 47; Phương án: 7 8 7.

Véc tơ x sau $j = 2$: 7 8 7.

Xét tọa độ $j = 3$:

$t = 68$ và $x + h$: 7 8 8.

$t = 32$ và $x - h$: 7 8 6.

Thay giá trị tốt nhất: 32; Phương án: 7 8 6.

Véc tơ x sau $j = 3$: 7 8 6.

Giá trị tốt nhất sau tìm kiếm: 32;

Phương án: 7 8 6.

Dò tìm theo điểm mẫu

$t = 8$ và điểm mẫu x : 6 7 5.

Thay giá trị tốt nhất: 8; phương án: 6 7 5.

Véc tơ x trước khi dò tìm: 6 7 5

Xét tọa độ $j = 1$:

$t = 20$ và $x + h$: 7 7 5.

$t = 4$ và $x - h$: 5 7 5.

Thay giá trị tốt nhất: 4; phương án: 5 7 5.

Véc tơ x sau $j = 1$: 5 7 5.

Xét tọa độ $j = 2$:

$t = 7$ và $x + h$: 5 8 5.

$t = 3$ và $x - h$: 5 6 5.

Thay giá trị tốt nhất: 3; phương án: 5 6 5.

Véc tơ x sau $j = 2$: 5 6 5

Xét tọa độ $j = 3$:

$t = 12$ và $x + h$: 5 6 6.

$t = 0$ và $x - h$: 5 6 4.

Thay giá trị tốt nhất: 0; phương án: 5 6 4.

Véc tơ x sau $j = 3$: 5 6 4.

Giá trị tốt nhất sau tìm kiếm: 0.

Phương án: 5 6 4.

Dò tìm theo điểm mẫu

$t = 32$ và điểm mẫu x : 3 4 2.

Véc tơ x trước khi dò tìm: 3 4 2.

Xét tọa độ $j = 1$:

$t = 20$ và $x + h$: 4 4 2.

$t = 52$ và $x - h$: 2 4 2.

Véc tơ x sau $j = 1$: 3 4 2.

Xét tọa độ $j = 2$:

$t = 29$ và $x + h$: 3 5 2.

$t = 37$ và $x - h$: 3 3 2.

Véc tơ x sau $j = 2$: 3 4 2.

Xét tọa độ $j = 3$:

$t = 23$ và $x + h$: 3 4 3.

$t = 47$ và $x - h$: 3 4 1.

Véc tơ x sau $j = 3$: 3 4 2.

Giá trị tốt nhất sau tìm kiếm: 0

Phương án: 5 6 4.

Bước lặp thứ 2

Độ dài bước $h = 1$.

Dò tìm theo điểm cơ sở

Véc tơ x trước khi dò tìm: 5 6 4.

Xét tọa độ $j = 1$:

$t = 4$ và $x + h$: 6 6 4.

$t = 4$ và $x - h$: 4 6 4.

Véc tơ x sau $j = 1$: 5 6 4.

Xét tọa độ $j = 2$:

$t = 1$ và $x + h$: 5 7 4.

$t = 1$ và $x - h$: 5 5 4.

Véc tơ x sau $j = 2$: 5 6 4.

Xét tọa độ $j = 3$:

$t = 3$ và $x + h$: 5 6 5.

$t = 3$ và $x - h$: 5 6 3.

Véc tơ x sau $j = 3$: 5 6 4.

Giá trị tốt nhất sau tìm kiếm: 0.

Phương án: 5 6 4.

Bước lặp thứ 3

Độ dài bước $h = 0.1$

Dò tìm theo điểm cơ sở

Véc tơ x trước khi dò tìm: 5 6 4.

Xét tọa độ $j = 1$:

$t = 0.04$ và $x + h$: 5.1 6 4.

$t = 0.04$ và $x - h$: 4.9 6 4.

Véc tơ x sau $j = 1$: 5 6 4.

Xét tọa độ $j = 2$:

$t = 0.01$ và $x + h$: 5 6.1 4.

$t = 0.01$ và $x - h$: 5 5.9 4.

Véc tơ x sau $j = 2$: 5 6 4.

Xét tọa độ $j = 3$:

$t = 0.03$ và $x + h$: 5 6 4.1.

$t = 0.03$ và $x - h$: 5 6 3.9.

Véc tơ x sau $j = 3$: 5 6 4.

Giá trị tốt nhất sau tìm kiếm: 0.

Phương án: 5 6 4.

.....

Bước lặp thứ 10

Độ dài bước $h = 0$.

Dò tìm theo điểm cơ sở

Véc tơ x trước khi dò tìm: 5 6 4.

Xét tọa độ $j = 1$:

$t = 0$ và $x + h$: 5 6 4.

$t = 0$ và $x - h$: 5 6 4.

Véc tơ x sau $j = 1$: 5 6 4.

Xét tọa độ $j = 2$

$t = 0$ và $x + h$: 5 6 4.

$t = 0$ và $x - h$: 5 6 4.

Véc tơ x sau $j = 2$: 5 6 4.

Xét tọa độ $j = 3$:

$t = 0$ và $x + h$: 5 6 4.

$t = 0$ và $x - h$: 5 6 4.

Véc tơ x sau $j = 3$: 5 6 4.

Giá trị tốt nhất sau tìm kiếm: 0.

Phương án: 5 6 4.

Kết quả giải bài toán

Số bước lặp: 10.

Giá trị tối ưu: 0.

Phương án tối ưu: (5 ; 6 ; 4).

Số lần thay đổi hàm mục tiêu: 7.

Số lần dò tìm cơ sở : 10.

Số lần dò tìm theo hướng giảm: 2.

3.5. Các ví dụ đã chạy chương trình

Ví dụ 1. Cực tiểu hàm Rosenbrock [1960]:

$$f(x_1; x_2) = 100(x_2 - x_1)^2 + 5(1 - x_1)^2.$$

Thuật toán Hooke – Jeeves giải ví dụ này qua các bước lặp với :

$tg = 0$, $dung = 0$, $h = 1$, $giam = 0.01$, $eps = 0.00000001$, $vocung = 1000000$,

$n = 2$, Điểm xuất phát $u = (6.0; 6.0)$.

Kết quả giải bài toán

Số bước lặp: 10.

Giá trị tối ưu: 0.

Phương án tối ưu: (1.0007 ; 1.0014).

Số lần thay đổi hàm mục tiêu: 1179.

Số lần dò tìm cơ sở: 10.

Số lần dò tìm theo hướng giảm: 456.

Ví dụ 2. Cực tiểu hàm mũ hai biến:

$$f(x_1, x_2) = \sum_{\alpha} [(e^{-\alpha x_1} - e^{-\alpha x_2}) - (e^{-\alpha} - e^{-10\alpha})]^2$$

Với $\alpha = 0.1, 0.2, \dots, 1.0$.

Thuật toán Hooke – Jeeves giải ví dụ này qua các bước lặp với :

tg = 0, dung = 0, h = 1, giam = 0.01, eps = 0.00000001, vocung = 1000000,

n = 2. Điểm xuất phát $u = (3; 2)$.

Kết quả giải bài toán

Giá trị tối ưu: 0.

Phương án tối ưu: (1;10).

Số lần thay đổi hàm mục tiêu: 8.

Số lần dò tìm cơ sở : 7.

Số lần dò tìm theo hướng giảm: 4.

Ví dụ 3. Cực tiểu hàm Fletcher và Powell (1963):

$$f(x_1; x_2; x_3) = 100[x_3 - 10\theta(x_1; x_2)]^2 + \left[\sqrt{(x_1^2 + x_2^2)} - 1 \right]^2 + x_3^2$$

$$\text{Trong đó: } 2\pi\theta(x_1; x_2) = \begin{cases} \arctan\left(\frac{x_2}{x_1}\right), & x_1 > 0. \\ \pi + \arctan\left(\frac{x_2}{x_1}\right), & x_1 < 0. \end{cases}$$

Thuật toán Hooke – Jeeves giải ví dụ này qua các bước lặp với :

tg = 0, dung = 0, h = 1, giam = 0.01, eps = 0.00000001, vocung = 1000000,

n = 3. Điểm xuất phát $u = (-1; 0; 0)$.

Kết quả giải bài toán

Giá trị tối ưu: 0.

Phương án tối ưu : (1; 0.0014; 0.0022).

Số lần thay đổi hàm mục tiêu: 218.

Số lần dò tìm cơ sở : 13.

Số lần dò tìm theo hướng giảm : 68.

Ví dụ 4. Cực tiểu hàm Griewank:

$$f(x_1, \dots, x_n) = \frac{1}{400n} \sum_{i=1}^n x_i^2 - \prod_{i=1}^n \cos\left(\frac{x_i}{\sqrt{i}}\right)$$

Trong đó: $n = 12, x_i \in [-1000, 1000]$.

Thuật toán Hooke – Jeeves giải ví dụ này qua các bước lặp với :

tg = 0, dung = 0, h = 1, giam = 0.01, eps = 0.00000001, vocung = 1000000,

n = 12. Điểm xuất phát: $u = (1; 1; 1; 1; 1; 1; 1; 1; 1; 1; 1; 1)$.

Kết quả giải bài toán

Giá trị tối ưu: -1.

Phương án tối ưu: (0; 0; 0; 0; 0; 0; 0; 0; 0; 0; 0; 0).

Số lần thay đổi hàm mục tiêu: 12.

Số lần dò tìm cơ sở: 7.

Số lần dò tìm theo hướng giảm : 1.

Ví dụ 5. Cực tiểu hàm Powell [1962]:

$$f(x) = (x_1 + 10x_2)^2 + 5(x_3 - x_4)^2 + (x_2 - 2x_3)^4 + 10(x_1 - x_4)^4$$

Thuật toán Hooke – Jeeves giải ví dụ này qua các bước lặp với :

$n = 4$; $h = 1$; $\text{eps} = 0.0000001$, $\text{giam} = 10$.

Điểm xuất phát $u = (3; -1; 0; 1)$. $\text{vocung} = 1000000$.

Kết quả giải bài toán

Số bước lặp: 10.

Giá trị tối ưu: 0.

Phương án tối ưu: $(0; 0; 0; 0)$.

Số lần thay đổi hàm mục tiêu : 4.

Số lần dò tìm cơ sở : 10.

Số lần dò tìm theo hướng giảm : 1.

Ví dụ 6. Cực tiểu hàm có ràng buộc:

$$\begin{aligned} f(x_1; x_2; x_3) = & 2x_1^2 + x_2^2 + 3x_3^2 - x_1x_2 - 2x_1x_3 - x_2x_3 + 10x_1 \\ & + 9x_2 - 26x_3. \end{aligned} \quad (*)$$

thỏa mãn

$$g_1(x) = x_1 - 1 \leq 0.$$

$$g_2(x) = -x_2 \leq 0.$$

$$g_3(x) = -x_3 \leq 0.$$

$$g_4(x) = x_1 + x_2 + x_3 - 4 \leq 0.$$

$$g_5(x) = x_1 - 2x_2 - x_3 + 3 \leq 0.$$

Sử dụng phương pháp hàm phạt đưa bài toán (*) về giải bài toán cực tiểu không ràng buộc :

$$F(x_1; x_2; x_3) = f(x_1; x_2; x_3) + M \sum_{i=1}^5 [\max\{g_i(x); 0\}]^2.$$

Thuật toán Hooke - Jeeves giải ví dụ này qua các bước lặp với :

$h=1$; $giam = 0.01$; $eps = 0.00000001$; $vocung = 1000000$; $n = 3$.

Điểm xuất phát $u = (2; 2; 2)$; $M = 1000000$.

Kết quả giải bài toán

Số bước lặp: 16.

Giá trị tối ưu: -56.60001 .

Phương án tối ưu: $(-0.4; 0; 4.2)$.

Số lần thay đổi hàm mục tiêu: 78.

Số lần dò tìm cơ sở : 16.

Số lần dò tìm theo hướng giảm: 33.

KẾT LUẬN

Các phương pháp tối ưu không dùng đạo hàm đóng vai trò rất quan trọng trong việc giải các bài toán thực tiễn khoa học kỹ thuật. Các hàm cần cực tiểu thường không khả vi, không lồi, không lõm..., thậm chí giá trị hàm mục tiêu được cho bởi một quy trình tính toán mà không thể viết được ở dạng giải tích.

Luận văn này đã trình bày hai thuật toán cực tiểu hàm nhiều biến không dùng đạo hàm: thuật toán đơn hình Nelder – Mead và thuật toán dò tìm theo tọa độ Hooke – Jeeves.

Chương 1 trình bày thuật toán Nelder – Mead cực tiểu hàm nhiều biến, các tính chất về tính hữu hạn của thuật toán, chương trình máy tính thể hiện thuật toán và một số bài toán mẫu đã được thử nghiệm.

Chương 2 trình bày phương pháp tìm kiếm trực tiếp Hooke - Jeeves, chương trình máy tính thể hiện thuật toán và một số bài toán mẫu đã được thử nghiệm.

Tác giả đã cố gắng trình bày vấn đề một cách rõ ràng, đưa ra lập luận đơn giản nhất và một số ví dụ cụ thể để minh họa cho các thuật toán được đề cập trong luận văn.

TÀI LIỆU THAM KHẢO

- [1]. J.A. Nelder and R.Mead (1965). “A simplex method for function minimization”. *Computer Journal* **7** : 308 – 313.
- [2]. J.C. Lagarias, J.A. Reeds, M.H. Wright and P.E. Wright. (1998) “Convergence properties of the Nelder – Mead simplex method in low dimensions”, *SIAM Journal of Optimization* , **9**: 112 – 147.
- [3]. M.A. Luersen, R.L.Riche. Globalized Nelder – Mead method for engineering optimization. *Computers & Structures*, 2004, 1 – 9. Available outline at www.sciencedirect.com.
- [4]. Nelder – Mead method (2004), “chap.8 numerical optimization and sec.8.2 Nelder– Mead and Powell’s methods”, 430 – 436. Internet.
- [5]. R. Hooke and T.A. Jeeves (1961). “Direct search solution of numerical and statistical problems”, *Journal of ACM*, **8**: 212 – 229.
- [6]. B.L.Robertson, C.J.Price and M.Reale (2009). “A hybrid Hooke - Jeeves – Direct method for non-smooth optimization”. *AMO – Advanced Modeling and Optimization, Volume 11, Number 1, 2009*, 43 – 61.
- [7]. D.M. Himmelblau (1975). “Applied nonlinear programming”. Moscva, 157 – 163 (bản dịch tiếng Nga).
- [8]. A.I. Johnson. “Hooke and Jeeves algorithm”. Computer code developed by A.I. Johnson, University of Western Ontario, Canada, 309 – 317.