

TRƯỜNG ĐẠI HỌC BÁCH KHOA

Bài tiểu luận
MÔN TRÍ TUỆ NHÂN TẠO NÂNG CAO

Học viên thực hiện: 1. Nguyễn Thị Hảo
2. Trần Thị Hải Yến

Lớp : 2010B CNTT – HV

Tháng 12 - 2010

MỤC LỤC

CHƯƠNG 1: MỞ ĐẦU	3
CHƯƠNG 2: THUẬT TOÁN DI TRUYỀN	5
1. Giới thiệu:	5
2. Nội dung:.....	5
2.1. Cơ sở lý thuyết:.....	5
2.2. Các bước quan trọng trong việc áp dụng thuật giải di truyền.....	7
2.3. Các công thức của thuật giải di truyền:	7
CHƯƠNG 3: ÁP DỤNG GIẢI THUẬT DI TRUYỀN TRONG	9
GIẢI BÀI TOÁN TỐI ƯU	9
1. Bài toán	9
2. Cách biểu diễn biến trong thuật giải di truyền.....	10
3. Thuật toán cực tiểu hóa hàm F với n biến	12
4. Hàm mục tiêu	13
5. Các hàm, thủ tục.....	13
6. Hàm chính	20
CHƯƠNG 4: KẾT LUẬN.....	22

CHƯƠNG 1: MỞ ĐẦU

Tối ưu hóa là một nội dung quan trọng của Tin học ứng dụng có liên quan đến mọi lĩnh vực của tự nhiên và xã hội.

Cho đến nay, tuy có khá nhiều phương pháp giải quyết bài toán tối ưu hàm số, nhưng nhìn chung các phương pháp chỉ dừng lại ở những lớp bài toán với thông tin rõ ràng hoặc với các thông tin bổ trợ khác. Do đó, việc tìm một phương pháp mới để giải bài toán tối ưu là cần thiết và có ý nghĩa thực tế.

Thuật giải di truyền (Genetic Algorithm = GA) là một trong những kỹ thuật tìm kiếm lời giải tối ưu đáp ứng được yêu cầu của bài toán và ứng dụng.

Hiện nay, thuật toán di truyền cùng với logic mờ được ứng dụng rất rộng rãi trong các lĩnh vực phức tạp. Thuật toán di truyền kết hợp với logic mờ chứng tỏ được hiệu quả của nó trong các vấn đề khó có thể giải quyết bằng các phương pháp thông thường hay các phương pháp cổ điển, nhất là trong các bài toán cần có sự lượng giá, đánh giá sự tối ưu của kết quả thu được. Chính vì vậy, thuật giải di truyền đã trở thành đề tài nghiên cứu thú vị và đem đến nhiều ứng dụng trong thực tiễn.

GA là phương thức giải quyết vấn đề bắt chước lối hành xử của con người để sinh tồn và phát triển. Nó giúp tìm ra giải pháp tối ưu hay tốt nhất trong điều kiện thời gian và không gian cho phép. Khác với chương trình giải tích, GA xét đến toàn bộ các giải pháp, bằng cách xét trước nhất một số giải pháp, sau đó loại bỏ những thành phần không thích hợp và chọn những thành phần thích nghi hơn để tạo sinh và biến hóa nhằm mục đích tạo ra nhiều giải pháp mới có hệ số thích nghi ngày càng cao.

Ngày nay, GA được ứng dụng khá nhiều trong các lĩnh vực như khoa học, kinh doanh và giải trí. Đầu tiên phải kể đến là các bài toán tối ưu bao gồm tối ưu số và tối ưu tổ hợp đã sử dụng GA để tìm lời giải như là bài toán người du lịch (Travelling Salesman Problems - TSP). Ứng dụng kế tiếp của GA là thiết kế và điều khiển robo....

Với những ưu điểm trên của GA, nhóm chúng em đã chọn “**Thuật giải di truyền**” làm đề tài nghiên cứu với ứng dụng “Giải bài toán tối ưu hàm nhiều biến”.

Các thành viên trong nhóm cùng công tác một nơi nên việc sưu tầm tài liệu, soạn thảo và chương trình demo cả nhóm cùng thực hiện.

Chúng em xin chân thành cảm ơn TS. Nguyễn Thanh Thủy cùng các thầy cô trong viện đã giúp đỡ chúng em hoàn thành học phần **Trí tuệ nhân tạo nâng cao** và bài tiểu luận này.

CHƯƠNG 2: THUẬT TOÁN DI TRUYỀN

1. Giới thiệu:

Thuật toán di truyền là thuật toán tối ưu ngẫu nhiên dựa trên cơ chế chọn lọc tự nhiên và tiến hóa di truyền. Nguyên lý cơ bản của thuật toán di truyền đã được Holland giới thiệu vào năm 1962. Cơ sở toán học đã được phát triển từ cuối những năm 1960 và đã được giới thiệu trong quyển sách đầu tiên của Holland, **Adaptive in Natural and Artificial Systems**. Thuật toán di truyền được ứng dụng đầu tiên trong hai lĩnh vực chính: tối ưu hóa và học tập của máy. Trong lĩnh vực tối ưu hóa thuật toán di truyền được phát triển nhanh chóng và ứng dụng trong nhiều lĩnh vực khác nhau như tối ưu hàm, xử lý ảnh, bài toán hành trình người bán hàng, nhận dạng hệ thống và điều khiển. Thuật toán di truyền cũng như các thuật toán tiến hóa nói chung, hình thành dựa trên quan niệm cho rằng, quá trình tiến hóa tự nhiên là quá trình hoàn hảo nhất, hợp lý nhất và tự nó đã mang tính tối ưu. Quan niệm này có thể xem như một tiên đề đúng, không chứng minh được, nhưng phù hợp với thực tế khách quan. Quá trình tiến hóa thể hiện tính tối ưu ở chỗ, thế hệ sau bao giờ cũng tốt hơn (phát triển hơn, hoàn thiện hơn) thế hệ trước bởi tính kế thừa và đấu tranh sinh tồn.

Thuật giải di truyền là kỹ thuật giúp giải quyết vấn đề bất chước theo sự tiến hóa của con người hay của sinh vật nói chung, trong điều kiện quy định sẵn của môi trường. Phương tiện để thực hiện cách giải quyết vấn đề này là chương trình tin học gồm các bước thi hành, từ việc chọn giải pháp tiêu biểu cho vấn đề, cho đến việc chọn các hàm số thích nghi hơn. Như vậy, GA không chú trọng đến giải pháp duy nhất và chính xác như phương pháp cổ điển, trái lại GA xét đến toàn bộ các giải pháp và chọn lấy giải pháp tương đối tốt nhất nếu không nói là tối ưu. GA tuy dựa trên tính ngẫu nhiên nhưng có hướng dẫn bởi hàm số thích nghi, do đó không có nghĩa là đoán mò như nhiều người hiểu lầm, trái lại GA có một nền tảng toán học vững chắc.

2. Nội dung:

2.1. Cơ sở lý thuyết:

Thuật toán di truyền gồm có bốn quy luật cơ bản là lai ghép, đột biến, sinh sản và chọn lọc tự nhiên như sau:

a) Quá trình lai ghép (phép lai):

Quá trình này diễn ra bằng cách ghép một hay nhiều đoạn gen từ hai nhiễm sắc thể cha-mẹ để hình thành nhiễm sắc thể mới mang đặc tính của cả cha lẫn mẹ. Phép lai này có thể mô tả như sau: Chọn ngẫu nhiên hai hay nhiều cá thể trong quần thể. Giả sử chuỗi nhiễm sắc thể của cha và mẹ đều có chiều dài là m . Tìm điểm lai bằng cách tạo ngẫu nhiên một con số từ 1 đến $m-1$. Như vậy, điểm lai này sẽ chia hai chuỗi nhiễm sắc thể cha-mẹ thành hai nhóm nhiễm sắc thể con là m_1 và m_2 . Hai chuỗi nhiễm sắc thể con lúc này sẽ là $m_{11}+m_{22}$ và $m_{21}+m_{12}$. Đưa hai chuỗi nhiễm sắc thể con vào quần thể để tiếp tục tham gia quá trình tiến hóa.

b) Quá trình đột biến (phép đột biến):

Quá trình tiến hóa được gọi là quá trình đột biến khi một hoặc một số tính trạng của con không được thừa hưởng từ hai chuỗi nhiễm sắc thể cha-mẹ. Phép đột biến xảy ra với xác suất thấp hơn rất nhiều lần so với xác suất xảy ra phép lai. Phép đột biến có thể mô tả như sau: Chọn ngẫu nhiên một số k từ khoảng $1 \leq k \leq m$ Thay đổi giá trị của gen thứ k Đưa nhiễm sắc thể con vào quần thể để tham gia quá trình tiến hóa tiếp theo.

c) Quá trình sinh sản và chọn lọc (phép tái sinh và phép chọn):

- **Phép tái sinh:** Là quá trình các cá thể được sao chép dựa trên độ thích nghi của nó. Độ thích nghi là một hàm được gán các giá trị thực cho các cá thể trong quần thể của nó. Phép tái sinh có thể mô phỏng như sau: Tính độ thích nghi của từng cá thể trong quần thể, lập bảng cộng dồn các giá trị thích nghi đó (theo thứ tự gán cho từng cá thể) ta được tổng độ thích nghi. Giả sử quần thể có n cá thể. Gọi độ thích nghi của cá thể thứ i là F_i , tổng dồn thứ i là F_t . Tổng độ thích nghi là F_m Tạo số ngẫu nhiên F có giá trị trong đoạn từ 0 đến F_m Chọn cá thể k đầu tiên thỏa mãn $F \geq F_t$ đưa vào quần thể của thế hệ mới.

- **Phép chọn:** Là quá trình loại bỏ các cá thể xấu và để lại những cá thể tốt. Phép chọn được mô tả như sau: Sắp xếp quần thể theo thứ tự độ thích nghi giảm dần Loại bỏ các cá thể cuối dãy, chỉ để lại n cá thể tốt nhất.

2.2. Các bước quan trọng trong việc áp dụng thuật giải di truyền

Để giải quyết vấn đề bằng thuật giải di truyền, chúng ta cần thực hiện 7 bước quan trọng sau đây:

- Bước 1: Chọn mô hình cho giải pháp của vấn đề. Chọn 1 số tượng trưng cho toàn bộ các giải pháp có thể có cho vấn đề.

- Bước 2: Chỉ định cho mỗi giải pháp một ký hiệu, ký hiệu có thể là dãy của những số 1 và 0 thuộc hệ nhị phân hay dãy số thập phân, dãy của chữ hay hỗn hợp của số và chữ. Trong giai đoạn mới làm quen với GA, chỉ nên dùng hệ nhị phân để làm ký hiệu cho giải pháp.

- Bước 3: Tìm hàm số thích nghi cho vấn đề và tính hệ số thích nghi cho từng giải pháp.

- Bước 4: Dựa trên hệ số thích nghi của các giải pháp để thực hiện sự tạo sinh (reproduction) và biến hóa các giải pháp. Các phương thức biến hóa gồm: lai ghép (cross over), đột biến (mutation).

- Bước 5: Tính các hệ số thích nghi cho các giải pháp mới là loại bỏ những giải pháp kém nhất để chỉ còn giữ lại một số nhất định các giải pháp.

- Bước 6: Nếu chưa tìm được giải pháp tối ưu hay tương đối khá nhất hay chưa hết hạn kỳ ấn định, trở lại bước thứ 4 để tìm giải pháp mới.

- Bước 7: Tìm được giải pháp tối ưu hoặc nếu thời gian cho phép để chấm dứt thì báo cáo kết quả tính được.

2.3. Các công thức của thuật giải di truyền:

Tính độ thích nghi $eval(v_i)$ của mỗi nhiễm sắc thể v_i ($i = 1..kích\;thước\;quần\;thể$):

$$eval(v_i) = \frac{f(v_i)}{\sum_{i=1}^{kíchthướcquanthể} f(v_i)} \quad \text{Với } f(v_i) \text{ là hàm mục tiêu.}$$

Tìm tổng giá trị thích nghi quần thể:

$$F = \sum_{i=1}^{kíchthướcquanthể} eval(v_i)$$

Tính xác suất chọn p_i cho mỗi nhiệm sắc thể v_i :

$$p_i = \frac{eval(v_i)}{\sum_{i=1}^{kíchthướcquầnthể} eval(v_i)}$$

Tính xác suất tích lũy q_i cho mỗi nhiệm sắc thể:

$$q_i = \sum_{j=1}^i p_j$$

Tiến trình chọn lọc được thực hiện bằng cách quay bánh xe rulet kích thước quần thể lần. Mỗi lần chọn ra một nhiệm sắc thể từ quần thể hiện hành vào quần thể mới theo cách sau: Phát sinh một số ngẫu nhiên r trong khoảng $[0, 1]$ Nếu $r < q_1$ thì chọn nhiệm sắc thể v_1 , ngược lại chọn nhiệm sắc thể v_i ($2 \leq i \leq$ kích thước quần thể) sao cho $q_{i-1} < r \leq q_i$.

Đó là các bước quan trọng trong việc áp dụng thuật giải di truyền. Một ứng dụng đơn giản và có ý nghĩa thực tế mà nhiều người quan tâm đó là bài toán tối ưu hàm số nhiều biến.

CHƯƠNG 3: ỨNG DỤNG GIẢI THUẬT DI TRUYỀN TRONG GIẢI BÀI TOÁN TỐI ƯU

1. Bài toán

Trong kỹ thuật, khi giải quyết bất kỳ nhiệm vụ nào chúng ta đều mong muốn có phương án tốt nhất theo một hoặc một vài tiêu chí nào đó. Có thể liệt kê rất nhiều những ví dụ cụ thể như: tiết kiệm thời gian nhất, chi phí nhỏ nhất, năng suất lớn nhất, quãng đường đi ngắn nhất, thiết kế kết cấu với trọng lượng vật liệu nhỏ nhất... Để giải được những bài toán này, toán học đã cho ra đời một ngành là “Quy hoạch toán học” hay “tối ưu hóa”. Bài toán tối ưu nói chung được viết dưới dạng toán học như sau:

Tìm giá trị cực tiểu (hoặc cực đại) hàm:

$$f(x) \rightarrow \min(\max); x \in R^n \quad (1)$$

Với các điều kiện: $g_i(x) \geq 0; \quad i = 1, 2, \dots, m$

$$h_i(x) = 0; \quad i = 1, 2, \dots, l$$

Bài toán đặt ra yêu cầu là tìm tập hợp các biến $x_i, i = 1, \dots, n$ thỏa mãn các điều kiện ràng buộc đồng thời hàm $f(x)$ đạt giá trị cực tiểu (hoặc cực đại). Thực ra tìm cực tiểu hoặc cực đại trong toán học không khác nhau nhiều (dùng phép biến đổi hàm ngược).

Hàm $f(x)$ trong biểu thức (1) được gọi là hàm mục tiêu hoặc tiêu chuẩn tối ưu, biểu diễn mối quan hệ giữa tiêu chuẩn chất lượng của quá trình khảo sát và các biến độc lập x .

Các hàm số $g_i(x), h_i(x)$ là các điều kiện ràng buộc của bài toán tối ưu dưới dạng đẳng thức và bất đẳng thức. Trong không gian các biến, các hàm số này tạo ra miền giới hạn D các khả năng cho phép của hàm $f(x)$.

Nếu như $D \equiv R^n$ (với R là số chiều của hàm mục tiêu), có nghĩa là không tồn tại bất kỳ một điều kiện giới hạn nào ta nói rằng bài toán quy hoạch phi tuyến không có điều kiện ràng buộc.

Thuật giải di truyền cho bài toán tối ưu một hàm F có n biến, $F(x_1, x_2, \dots, x_n)$. Biết rằng mỗi biến x_i có thể lấy các giá trị từ miền $D_i = [a_i; b_i]$ là tập con của tập các số thực R và yêu cầu độ chính xác là k chữ số thập phân đối với các giá trị biến.

2. Cách biểu diễn biến trong thuật giải di truyền

a) Biểu diễn các biến nhờ các vectơ nhị phân

- Mã hóa, ánh xạ một xâu với chiều dài hữu hạn sang các tham biến của bài toán tối ưu.

- Tham biến x thuộc $[U_{\min}; U_{\max}]$ sẽ được biểu diễn bởi chuỗi nhị phân có chiều dài L . L bit mã hóa x ứng với giá trị trong miền $[0; 2^L]$ ánh xạ lên miền $[U_{\min}; U_{\max}]$; từ đó có thể kiểm soát miền giá trị của các biến và tính chính xác của chúng. Tỷ lệ co giãn của ánh xạ $g = (U_{\max} - U_{\min}) / (2^L - 1)$ (1)

Giá trị x tương ứng với chuỗi nhị phân $String_2$, tính theo công thức :

$$X = U_{\min} + \text{decimal}(String_2) * g \quad (2)$$

Trong đó, $\text{decimal}(String_2)$ biểu diễn giá trị thập phân của chuỗi nhị phân $string_2$, g xác định bởi công thức (1).

Ví dụ, biểu diễn biến x_1 bởi chuỗi nhị phân 0001 thì $\text{decimal}(0001) = 1$.

b) Toán tử chọn cá thể

- Giá trị thích nghi $f(i)$ được xác định đối với mỗi quần thể ; giá trị này càng lớn thì cá thể được coi là hợp lý.

- Các bước thực hiện :

- + Tính tổng giá trị thích nghi của tất cả các thành viên quần thể và gọi nó là tổng thích nghi.

- + Phát sinh một số n là số ngẫu nhiên trong khoảng từ 0 đến tổng giá trị thích nghi.

- + Trả lại thành viên quần thể đầu tiên mà độ thích nghi của nó cộng với độ thích nghi các thành viên quần thể trước đây lớn hơn hoặc bằng n .

c) Toán tử lai ghép

Tác động lên các cá thể cha mẹ để tạo ra con lai tốt. Chúng được áp dụng lên cặp cha mẹ được lựa chọn với xác suất lai ghép ký hiệu P_{cross} cho biết số

lượng $P_{\text{cross}} * \text{pop_size}$ (kích thước của quần thể lai tạo) nhiễm sắc thể được dùng cho hoạt động lai ghép.

Với mỗi nhiễm sắc thể trong quần thể :

Phát sinh một số ngẫu nhiên r thuộc $[0 ; 1]$.

$r < p_{\text{cross}}$ thì chọn nhiễm sắc thể đó lai ghép.

Sau đó, kết hợp nhiễm sắc thể được chọn một cách ngẫu nhiên: chọn ngẫu nhiên vị trí lai ghép pos từ miền $[1 ; L]$ (L là tổng số bit trong nhiễm sắc thể).

$(b_1 b_2 \dots b_{\text{pos}} b_{\text{pos}+1} \dots b_L)$

$(c_1 c_2 \dots c_{\text{pos}} c_{\text{pos}+1} \dots c_L)$

Lai ghép được con cháu :

$(b_1 b_2 \dots b_{\text{pos}} c_{\text{pos}+1} \dots c_L)$

$(c_1 c_2 \dots c_{\text{pos}} b_{\text{pos}+1} \dots b_L)$

d) Toán tử đột biến

Nhằm tạo ra những thông tin mới trong quần thể lai tạo tại các vị trí bit nào đó. Số lượng bit đột biến là $P_{\text{mu}} * L * \text{pop_size}$. Mỗi bit có cơ hội đột biến như nhau và được thay đổi 0 thành 1 và ngược lại. P_{mu} là xác suất đột biến.

Với mỗi nhiễm sắc thể trong quần thể và mỗi bit trong nhiễm sắc thể :

Phát sinh một số ngẫu nhiên r trong miền $[0 ; 1]$.

Nếu $r < P_{\text{mu}}$ tiến hành đột biến tại bit đó.

Các thao tác xử lý trên được lặp lại cho đến khi các cá thể con cháu của chúng tăng trưởng tới kích cỡ mong muốn của quần thể.

e) Hàm thích nghi

* Ánh xạ giá trị hàm mục tiêu sang giá trị thích nghi

Giả sử cực tiểu hàm đánh giá $g(x)$, chuyển sang hàm thích nghi $f(x)$

$$f(x) = \begin{cases} C_{\text{max}} - g(x) & \text{khi } g(x) < C_{\text{max}} \\ 0 & \text{TH khác} \end{cases}$$

Với C_{max} là tham số đầu vào ; có thể là giá trị lớn nhất trong quần thể hiện tại hoặc lớn nhất sau k vòng lặp.

* Điều chỉnh độ thích nghi :

Điều chỉnh tuyến tính, giả sử độ thích nghi gốc là f , độ thích nghi biến đổi là f^* thì $f^* = a*f + b$

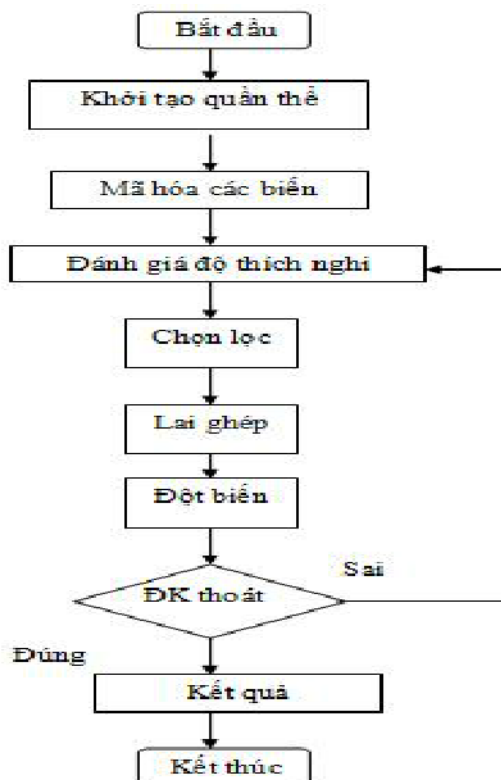
Và $f^*_{\max} = C_{\text{mult}} * f_{\text{avg}}$; trong đó C_{mult} là số các bản sao đối với 1 thành viên tốt nhất; số lượng quần thể n nhỏ thì $C_{\text{mult}} = 1.2$ đến 2 tỏ ra khá hiệu quả.

3. Thuật toán cực tiểu hóa hàm F với n biến

Thuật toán di truyền bao gồm các bước sau:

- Bước 1: Khởi tạo quần thể các nhiễm sắc thể.
- Bước 2: Xác định giá trị thích nghi của từng nhiễm sắc thể.
- Bước 3: Sao chép lại các nhiễm sắc thể dựa vào giá trị thích nghi của chúng và tạo ra những nhiễm sắc thể mới bằng các phép toán di truyền.
- Bước 4: Loại bỏ những thành viên không thích nghi trong quần thể.
- Bước 5: Chèn những nhiễm sắc thể mới vào quần thể để hình thành một quần thể mới.
- Bước 6: Nếu mục tiêu tìm kiếm đạt được thì dừng lại, nếu không trở lại bước 3.

Sơ đồ thuật toán:



4. Hàm mục tiêu

Tìm giá trị nhỏ nhất của $f(x_1, x_2, x_3, x_4) = (x_1 - 6)^2 + (x_2 - 4)^2 + (x_3 - 2)^2 + x_4^2$.

Với các biến x_1, x_2, x_3, x_4 thuộc $[-10; 10]$.

float f(float *k)

{

float s;

s=(k[0]-6)*(k[0]-6)+(k[1]-4)*(k[1]-4)+ (k[2]-2)*(k[2]-2) + k[3]*k[3];

return s;

}

5. Các hàm, thủ tục

a) Khởi tạo ngẫu nhiên trong miền $[0;1]$

float random01()

{

float s;

s=random(30000)/30000.0;

return s;

}

b) Hàm tạo giá trị ngẫu nhiên 0 hoặc 1 theo xác suất flip

Hàm cho vào một xác suất nào đó có thể là xác suất đột biến P_{mu} , xác suất lai ghép P_{cross} hay xác suất khởi tạo ban đầu.

int flip(float p)

{

if(random01()<p) return 1;

else return 0;

}

c) Xác định giá trị thích nghi, tổng thích nghi

Ánh xạ hàm mục tiêu sang hàm thích nghi và điều chỉnh độ thích nghi, xây dựng hàm thang quần thể (scalepop) tính giá trị thích nghi theo điều chỉnh ứng với từng cá thể trong quần thể, rồi tính tổng của chúng.

Dữ liệu vào: Giá trị hàm mục tiêu ứng với từng thành viên, cùng với kích cỡ của quần thể.

Dữ liệu ra: Giá trị thích nghi ứng với từng thành viên và tổng thích nghi của quần thể. Đây là kết quả vào cho thủ tục chọn lựa.

```
float scalepop(float*obj,float *fit,int popsize)
{
    float a,b,min,ave,max,sum,ob[100];
    int i;
    sum=max=-1.0e37;min=1.0e37;
    for(i=0;i<popsize;i++) if(sum<obj[i]) sum=obj[i];
    for(ave=0,i=0;i<popsize;i++)
    {
        obj[i]=sum-obj[i];
        if(max<obj[i]) max=obj[i];
        if(min<obj[i]) min=obj[i];
        ave+=obj[i];
    }
    ave/=popsize;
    if(min>(2*ave-max))
    {
        a=ave/(max-ave);
        b=a*(max-2*ave);
    }
    else
    {
        a=ave/(ave-min);
        b=-min*a;
    }
    for(sum=1,i=0;i<popsize;i++)
    {
```

```

        fit[i]=a*ob[i]+b;
        sum+=fit[i];
    }
    return sum;
}

```

d) Hàm chọn cá thể

Tìm ra những cá thể tốt nhất trong quần thể. Đầu vào là giá trị thích nghi, tổng giá trị thích nghi, kích cỡ của quần thể. Đầu ra là vị trí của cá thể tốt nhất trong quần thể.

Thuật:

```

int select(float *fit,float sum,int popsize)
{
    float partsum,rand;
    int i,j=popsize-1;
    rand=random01()*sum;
    for(partsum=0,j=0;j<popsize;j++)
    {
        partsum+=fit[j];
        if(partsum>=rand) {i=j;break;}
    }
    return i;
}

```

e) Hàm đột biến và hàm lai ghép

- Hàm đột biến: Kiểm tra từng ký tự trong chuỗi với xác suất đột biến P_{mu} , nếu thỏa mãn điều kiện không thì bỏ qua.

```

char mutation(char c,float pmu)
{
    char s;
    if(flip(pmu))
    {

```

```

        if(c) s=0;
        else s=1;
    }
    else s=c;
    return s;
}

```

- Hàm lai ghép: Đầu vào là 2 chuỗi cha mẹ, chiều dài của nhiễm sắc thể, xác suất lai ghép và tổng số biến n. Ra là chuỗi biểu diễn các con.

```

void crossover(char *parent1,char *parent2,char *child1,char *child2,
int lchrom,float pcross,float pmu,int n)
{
    int j,jcross,i;
    for(i=0;i<n;i++)
    {
        if(flip(pcross))
        { jcross=random(lchrom); }
        else jcross=lchrom;
        for(j=0;j<jcross;j++)
        {
            child1[i*lchrom+j]=mutation(parent1[i*lchrom+j],pmu);
            child2[i*lchrom+j]=mutation(parent2[i*lchrom+j],pmu);
        }
        for(j=jcross;j<lchrom;j++)
        {
            child1[i*lchrom+j]=mutation(parent2[i*lchrom+j],pmu);
            child2[i*lchrom+j]=mutation(parent1[i*lchrom+j],pmu);
        }
    }
}

```

f) Hàm khởi tạo quần thể

Mỗi nhiễm sắc thể biểu diễn tập vào x_1, x_2, \dots, x_n dưới dạng nhị phân. Do đó đầu vào là kích thước của quần thể pop_size; chiều dài của nhiễm sắc thể lchrom; đầu ra là giá trị các phần tử trong quần thể.

```
void initpop(char pop[100][250], int popsize, int lchrom)
{
    int i, j;
    for(i=0; i<popsize; i++)
    {
        for(j=0; j<lchrom; j++)
        {
            pop[i][j]=flip(0.5);
        }
    }
}
```

g) Hàm tạo sinh:

Đây là hàm quan trọng nhất trong thuật giải di truyền. Dựa vào giá trị thích nghi của từng cá thể trong quần thể, tổng giá trị thích nghi của quần thể, ứng dụng các toán tử di truyền gồm chọn lựa cá thể, đột biến, lai ghép lên quần thể các chuỗi biểu diễn các biến oldpop, từ đó sản sinh ra quần thể chuỗi biểu diễn mới.

Đầu vào: Tập hợp nhiễm sắc thể; kích cỡ quần thể; chiều dài nhiễm sắc thể biểu diễn một biến; tổng số biến nhập trong hàm mục tiêu n; xác suất lai ghép, xác suất đột biến; giá trị thích nghi và tổng giá trị thích nghi. Đầu ra là tập nhiễm sắc thể mới sau khi đã ứng dụng các toán tử.

Thủ tục:

```
void generate(char oldpop[100][250], char newpop[100][250], int popsize,
int lchrom, float *fit, float sum, float pcross, float pmu, int n)
{
    int j, mate1, mate2;
    j=0;
```

```

while(1)
{
    mate1=select(fit,sum,popsize);
    mate2=select(fit,sum,popsize);
    crossover(oldpop[mate1],oldpop[mate2],newpop[j],newpop[j+1],lc
    hrom,pcross,pmu,n);
    j+=2;
    if(j>=popsize) break;
}
}

```

h) Hàm nạp lại quần thể:

Sau khi một quần thể mới ra đời, nó được nạp vào quần thể hiện tại, vòng lặp mới được thực hiện để sản sinh ra những quần thể khác.

```

void reproduction(char oldpop[100][250],char newpop[100][250],int popsize,int
lchrom)
{
    int i,j;
    for(i=0;i<popsize;i++)
    {
        for(j=0;j<lchrom;j++)
            { oldpop[i][j]=newpop[i][j]; }
    }
}

```

i) Hàm chuyển đổi từ xâu sang số phẩy động gọi là hàm giải mã.

```

void decode(char *indi,int lchrom,float *up,float *down,double rate,float *k,int n)
{
    int i,j;
    double accum,power;
    char frac[30];
    for(j=0;j<n;j++)

```

```

{
    for(i=0;i<lchrom;i++)
        frac[i]=indi[j]*lchrom+i];
    power=1;
    for(accum=0,i=0;i<lchrom;i++)
    {
        if(frac[i])
            accum+=power;
        power*=2;
    }
    power=accum/rate;
    k[j]=down[j]+(up[j]-down[j])*power;
}
}

```

k) Hàm di truyền:

Đây là thủ tục chính của giải thuật.

Đầu vào: Miền giá trị của các biến x thuộc $[up; down]$, số vòng lặp (số lần tạo sinh), xác suất lai ghép, P_{mu} , tổng số biến x , kích cỡ quần thể, công thức hàm cần tối ưu.

Đầu ra: Tập giá trị số thực x với kết quả tối ưu của hàm.

```

float GEN(float *X,float *up,float *down,int gen,float ss(float *),
float pcross,float pmu,int n,int popsize)

```

```

{
    char oldpop[100][250],newpop[100][250];
    float obj[100],fit[100],sum,ty;
    int lchrom,i,j,N;
    float k[10],x[10];
    double rate;
    lchrom=250/n;
    if(lchrom>25) lchrom=25;

```

```

N=lchrom*n;
sum=1;
for(rate=0,i=0;i<lchrom;i++)
{
rate+=sum;
sum*=2;
}
randomize();
ty=1.0e37;
initpop(oldpop,popsize,N);
for(j=0;j<gen;j++)
{
for(i=0;i<popsize;i++)
{
decode(oldpop[i],lchrom,up,down,rate,k,n);
obj[i]=ss(k);
if(ty>obj[i])
{
for(int l=0;l<n;l++)
X[l]=k[l];
ty=obj[i];
}
}
sum=scalepop(obj,fit,popsize);
generate(oldpop,newpop,popsize,lchrom,fit,sum,pcross,pmu,n);
reproduction(oldpop,newpop,popsize,N);
}
return ty;
}

```

6. Hàm chính

```
int main()
{
    float x[3],y;
    float up[]={10,10};
    float down[]={-10,-10};
    y=GEN(x,up,down,50,f,.85,.05,2,40);
    printf("\f_min=%6.4f\n",y);
    for(int i=0;i<2;i++)
        printf("x[%d]=%4.2f ",i+1,x[i]);
    return 0;
}
```

CHƯƠNG 4: KẾT LUẬN

Qua quá trình nghiên cứu và thử nghiệm nhiều lần trên máy tính điện tử các thuật toán tối ưu tôi thấy, thuật giải di truyền có thể giải quyết bài toán tối ưu bất kỳ với n biến vào. Tuy nhiên, với số lượng biến vào khá nhiều, các giá trị hàm mục tiêu đạt được thường không gần với kết quả tối ưu thực sự. Để khắc phục, tăng số vòng lặp để hi vọng những lần sản sinh sau thu được kết quả như mong muốn.

TÀI LIỆU THAM KHẢO

- [1]. *Hoàng Kiếm*, Thuật giải di truyền, NXB Giáo dục, Hà nội, 2000.
- [2]. *Nguyễn Đình Thúc*, Trí tuệ nhân tạo - Lập trình tiến hóa, NXB Giáo dục, Hà nội, 2001.
- [3]. *Bùi Minh Trí*, Bài tập tối ưu hoá, NXB Khoa học Kỹ thuật, Hà nội, 2002.