

BÀI TẬP LỚN TRÍ TUỆ NHÂN TẠO: TÌM HIỂU GIẢI THUẬT DI TRUYỀN

LỜI NÓI ĐẦU

Trong ngành khoa học máy tính, tìm kiếm lời giải tối ưu cho các bài toán là vấn đề được các nhà khoa học máy tính đặc biệt rất quan tâm.

Mục đích chính của các thuật toán tìm kiếm lời giải là tìm ra lời giải tối ưu nhất cho bài toán trong thời gian nhỏ nhất. Các thuật toán như tìm kiếm không có thông tin / vét cạn (tìm kiếm trên danh sách, trên cây hoặc đồ thị) sử dụng phương pháp đơn giản nhất và trực quan nhất hoặc các thuật toán tìm kiếm có thông tin sử dụng heuristics để áp dụng các tri thức về cấu trúc của không gian tìm kiếm nhằm giảm thời gian cần thiết cho việc tìm kiếm được sử dụng nhiều nhưng chỉ với không gian tìm kiếm nhỏ và không hiệu quả khi tìm kiếm trong không gian tìm kiếm lớn.

Tuy nhiên, trong thực tiễn có rất nhiều bài toán tối ưu với không gian tìm kiếm rất lớn cần phải giải quyết. Vì vậy, việc đòi hỏi thuật giải chất lượng cao và sử dụng kỹ thuật trí tuệ nhân tạo đặc biệt rất cần thiết khi giải quyết các bài toán có không gian tìm kiếm lớn. Thuật giải di truyền (genetic algorithm) là một trong những kỹ thuật tìm kiếm lời giải tối ưu đã đáp ứng được yêu cầu của nhiều 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.

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. Hầu hết các nước có ngành CNTT phát triển đã và đang rất quan tâm đến lĩnh vực thiết kế robo nhằm giúp con người tiết kiệm sức lao động và giải phóng con người thoát khỏi các công việc nguy hiểm, đặc biệt hiện nay cuộc thi “Robocon” Châu Á_ Thái Bình Dương được các nước trong khu vực rất quan tâm. Ngoài phần cơ, để robo có thể tiến hành các hoạt động đơn giản nhất như đi, đứng... thì robo cần phải trang bị chương trình được lập trình dựa trên các thuật toán và ngôn ngữ thích hợp. Nhờ vào lịch trình được cài đặt cùng với một trí tuệ nhân tạo..., robo có thể định hướng thực hiện các hoạt động như con người. Tuy nhiên, việc tìm kiếm lời giải tốt nhất cho các hành động của robo không phải là đơn giản. Theo các nhà khoa học máy tính, thuật giải di truyền là một trong những thuật toán tối ưu giúp robo vạch lộ trình khi di chuyển. Với lý do trên, em chọn đề tài: “Thuật giải di truyền và ứng dụng”.

PHẦN I: THUẬT TOÁN DI TRUYỀN

I. 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.

II. 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:

2.1.1. 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 $m_1 + m_2$ và $m_2 + m_1$. Đưa hai chuỗi nhiễm sắc thể con vào quần thể để tiếp tục tham gia quá trình tiến hóa.

2.1.2. 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.

2.1.3. 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.

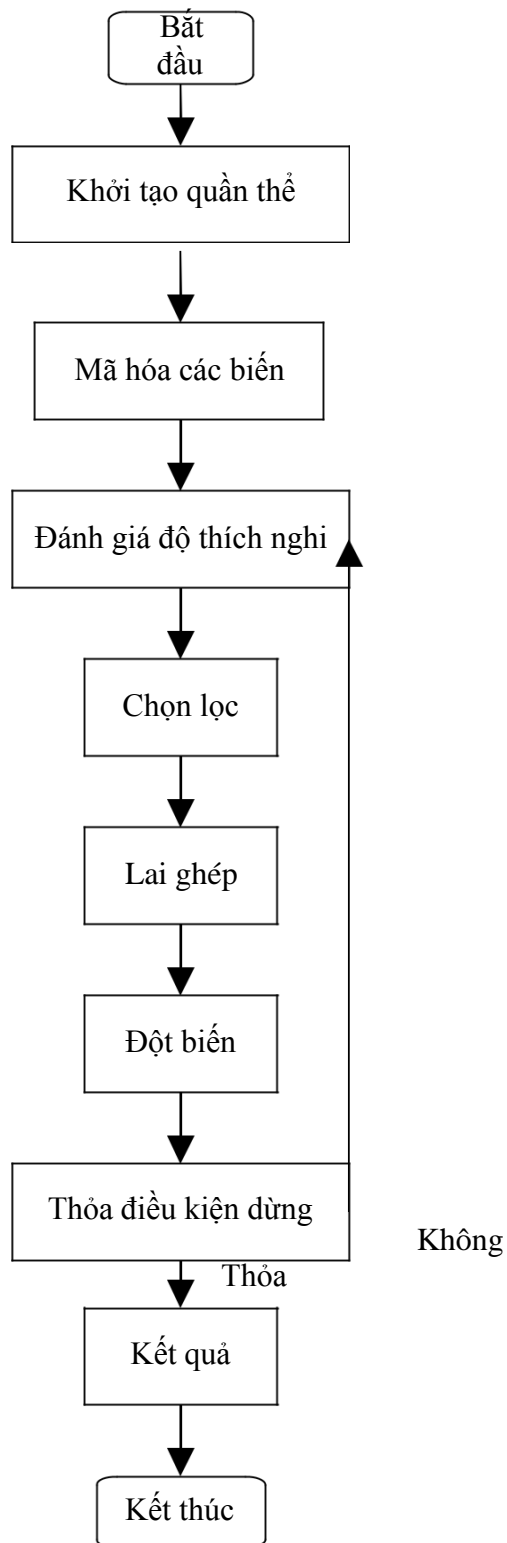
2.2 Cấu trúc thuật toán di truyền tổng quát

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:



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):

$$eval(v_i) = \frac{f(v_i)}{\sum_{i=1}^{kích\ thướ\ c\ q\ u\ a\ n\ h\ e} f(v_i)}$$

Tìm tổng giá trị thích nghi quần thể:

$$F = \sum_{i=1}^{kích\ thướ\ c\ q\ u\ a\ n\ h\ e} eval(v_i)$$

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

$$p_i = \frac{eval(v_i)}{\sum_{i=1}^{kích\ thướ\ c\ q\ u\ a\ n\ h\ e} eval(v_i)}$$

Tính xác suất tích lũy q_i cho mỗi nhiễm 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 nhiễm 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 nhiễm sắc thể v_1 , ngược lại chọn nhiễm sắc thể v_i ($2 \leq i \leq$ kích thước quần thể) sao cho $q_{i-1} < r \leq q_i$.

PHẦN II: ỨNG DỤNG

I. Ứng dụng

Tìm đáp số cho phương trình $X^2 = 64$. Đây là một bài toán đơn giản để giúp ta có thể hiểu rõ hơn các bước của thuật toán di truyền.

Giải bài toán di truyền theo các bước sau:

Bước 1: Chúng ta sử dụng hệ nhị phân để xây dựng mô hình bài toán. Ta dùng 4 bit nhị phân để mã hóa cho các đáp số của bài toán. Giả sử ta không biết đáp số của bài toán, ta sẽ chọn 4 số trong các đáp số có thể có và ký hiệu cho các đáp số đó.

Bảng chọn lựa:

Thứ tự	Nhị phân	Thập phân
1	0 0100	4
2	1 0101	21
3	0 1010	10
4	1 1000	24

Bước 2: Tìm hàm số thích nghi và tính hệ số thích nghi cho từng đáp số. Ta chọn hàm số thích nghi sau: $f(X) = 1000 - (X_2 - 64)$. Vậy, đáp số nào có hệ số thích nghi f gần bằng 1000 nhất thì đó là đáp số. Khảo sát kết quả tính được:

Thứ tự	Nhị phân	Thập phân (X)	$X_2 - 64$	Hệ số thích nghi $f(x)$
1	0 0100	4	- 48	952
2	1 0101	21	377	623
3	0 1010	10	36	964
4	1 1000	24	512	488

Bước 3: Ta thấy, hệ số thích nghi của các đáp số vẫn còn cách xa 1000. Do đó, cần tạo ra các đáp số mới bằng cách biến hóa các đáp số cũ. Ta thấy, số 4 và 10 có hệ số thích nghi cao hơn nên được chọn để tạo sinh và biến hóa. Đồng thời số 21 và 24 có hệ số thích nghi thấp sẽ bị loại.

Giả sử ta lai ghép hai số 4 và 10 theo hình sau :



Bước 4: Tính hệ số thích nghi cho quần thể mới

Thứ tự	Nhị phân	Thập phân (X)	$X_2 - 64$	Hệ số thích nghi f(x)
1	0 0100	4	- 48	952
2	0 1010	10	36	964
3	0 1000	8	0	1000
4	0 0110	6	28	968

Bước 5: May mắn chúng ta đã tìm được kết quả là $X = 8$ với hệ số thích nghi cao nhất là 1000.

Vậy kết quả của bài toán là $X = 8$.

II. Chương trình

_ Khai báo:

```
#include "stdio.h"
#include "conio.h"
#include "stdlib.h"
#include "math.h"
int CT[10];
int n=4;
int GTTN[10];
int kq;
```

```
int Max1,Max2;  
int Max1moi=0,Max2moi=0;  
int A[5],B[5];
```

_ Các đoạn code:

+ Code khởi tạo quần thể:

```
void Taoquantheth(int n)  
{  
    for(int i=0;i<n;i++)  
    {  
        printf("nhap ca the thu  
        %d:",i); scanf("%d",&CT[i]);  
    }  
}
```

+ Code tính giá trị thích nghi của quần thể bằng công thức: $F=1000-(x^2-$

```
64) void Giathichnghi(int n)  
{  
    for(int i=0;i<n;i++)  
    {  
        int gtn=1000-(CT[i]*CT[i] - 64);  
        GTTN[i]=gtn;  
    }  
}
```

+ Code kiểm tra giá trị thích nghi để suy ra kết

quả: int Kiemtra(int n)

```
{
    for(int i=0;i<n;i++)
    {
        if(GTTN[i]==1000)
            return (CT[i]);
    }
    return 0;
}
```

+ Tìm, chọn lọc cá thể để lai: (sắp xếp mảng giảm dần, lấy hai cá thể đầu tiên để lai với nhau)

void Timcathelai()

```
{
    for(int i=0;i<n;i++) for(int
        j=i+1;j<n;j++)
    {
        if(GTTN[i]<GTTN[j])
        {
            int a=GTTN[i];
            GTTN[i]=GTTN[j];
            GTTN[j]=a;
        }
    }
}
```

```
Max1=sqrt(1064-GTTN[0]);
Max2=sqrt(1064-GTTN[1]);
}
```

+ Lai cá thể với nhau: (mã hóa cá thể ra nhị phân, thay đổi cấu trúc nhị phân gây đột biến tạo cá thể mới, giải mã cá thể mới về thập phân)

```
void Laicathe()
{
    for (int i=0;i<5;i++)
    {
        A[i]=0;
        B[i]=0;
    }
    while(Max1!=0)
    {
        for(int j=0;j<5;j++)
        {
            A[j]=Max1%2;
            Max1=Max1/2;
        }
    }
    while(Max2!=0)
    {
        for(int k=0;k<5;k++)
        {
            B[k]=Max2%2;
            Max2=Max2/2;
        }
    }
}
```

```

        }
    }
    for(int l=4;l>1;l--)
    {
        int b=A[l];
        A[l]=B[l];
        B[l]=b;
    }
    for(int m=4;m>=0;m--)
    {
        Max1moi=Max1moi+A[m]*pow(2,m);
        Max2moi=Max2moi+B[m]*pow(2,m);
    }
}

```

+ Tạo quần thể mới: (kết hợp các thể mẹ và cá thể con) void Taoquanthemoi()

```

{
    CT[0]=Max1;
    CT[1]=Max2;
    CT[2]=Max1moi;
    CT[3]=Max2moi;
}

```

+ Hàm chính:

```

void main()
{

```



```
clrscr();
printf("Giai PT X*X=64, bang thuat toan di truyen");
printf("Tao quan the co 4 ca the");
Taoquanthe(n);
Giatrithichnghi(n);
kq=Kiemtra(n);
while(kq==0)
{
    Timcathelai();
    Laicathe();
    Taoquanthemoi();
    Giatrithichnghi(n);
    kq=Kiemtra(n);
}
printf(" Ket qua phuong trinh la : %d",kq);
getch();
}
```

PHẦN III: KẾT LUẬN

I. Ưu điểm

Trình bày và giới thiệu những khái niệm cơ bản, cơ sở lý thuyết về thuật giải di truyền. Trên cơ sở lý thuyết, đề tài đã cài đặt các phép toán cơ bản của thuật giải di truyền nhằm phục vụ cho việc thực hiện các ứng dụng. Sử dụng các phép toán của thuật giải di truyền để xây dựng ứng dụng cho bài toán người du lịch và bài toán vạch lộ trình đường đi cho robo.

II. Khuyết điểm

Đề tài chỉ giới thiệu những kiến thức chung nhất về thuật giải di truyền, chưa đi sâu vào các vấn đề nghiên cứu tối ưu khác. Phần ứng dụng vạch lộ trình đường đi cho robo chưa hoàn hảo. Đặc biệt là chưa giải quyết tốt việc robo tránh vật chướng ngại và kích thước quần thể thay đổi.

III. Ý kiến bản thân

Thuật toán di truyền đã chứng tỏ tính hữu ích của nó khi được ứng dụng rộng rãi trong nhiều lĩnh vực khác nhau của cuộc sống.

Trong lĩnh vực điều khiển tự động, thuật toán di truyền có thể được sử dụng để xác định thông số tối ưu cho các bộ điều khiển. Thông số bộ điều khiển được mã hóa thành các nhiễm sắc thể, thông qua mô phỏng, các nhiễm sắc thể này được đánh giá và lựa chọn thông qua mức độ thích nghi của chúng (cũng chính là các chỉ tiêu chất lượng của hệ thống). Kết quả của thuật toán sẽ cho một bộ điều khiển có thông số tốt nhất.

Trong y học, cấu trúc của các chất hóa học được mã hóa thành các nhiễm sắc thể hoặc đồ thị. Thuật toán di truyền sẽ lai ghép, lựa chọn để tạo ra các nhiễm sắc

thể mới (các chất hóa học mới). Và trong thực tế đã có rất nhiều loại thuốc mới được tạo ra như vậy.

