|  |  |  |
| --- | --- | --- |
| **ĐẠI HỌC BÁCH KHOA HÀ NỘI**  **TRƯỜNG CÔNG NGHỆ THÔNG TIN TRUYỀN THÔNG**        **BÁO CÁO NHẬP MÔN TRÍ TUỆ NHÂN TẠO**    **ĐỀ TÀI: “PHÂN TÍCH TÌM NGHIỆM HỆ THỐNG BẰNG THUẬT TOÁN DI TRUYỀN”**  **Nhóm thực hiện: 23**           |  |  | | --- | --- | | **Sinh viên**  Ngô Đức Quang Anh  Hồ Tuấn Hải  Đặng Nhật Duy  Nguyễn Hùng Cường | **MSSV**  **20215259**  **20210304**  **20210269**  **20215264** |       **Hà Nội – 2024** |

## Phân tích tìm nghiệm biểu thức bằng thuật toán di truyền

## 4.1. Cơ sở lý thuyết giải thuật di truyền (Genetic Algorithm)

Giải thuật di truyền (GA) là một thuật toán tìm kiếm tối ưu dựa trên mô phỏng quá trình tiến hóa tự nhiên. GA bắt đầu với một quần thể các cá thể, mỗi cá thể đại diện cho một giải pháp tiềm năng cho bài toán. Quần thể này sau đó được tiến hóa qua nhiều thế hệ, với mỗi thế hệ bao gồm một tập hợp các cá thể mới được tạo ra từ các cá thể của thế hệ trước.

**Quá trình tiến hóa của GA bao gồm bốn bước chính:**

* Sàng lọc: Các cá thể trong quần thể được đánh giá dựa trên một hàm mục tiêu, và các cá thể có giá trị hàm mục tiêu cao hơn sẽ được ưu tiên chọn lọc.
* Tái tổ hợp: Hai cá thể được chọn lọc sẽ được tái tổ hợp với nhau để tạo ra một cá thể mới. Quá trình tái tổ hợp có thể được thực hiện theo nhiều cách khác nhau, chẳng hạn như tái tổ hợp điểm, tái tổ hợp tuyến tính, hoặc tái tổ hợp chia đôi.
* Đột biến: Một số cá thể trong quần thể có thể được đột biến, tức là một hoặc nhiều gen của chúng sẽ được thay đổi ngẫu nhiên. Đột biến giúp tạo ra sự đa dạng trong quần thể và giúp GA tránh bị mắc kẹt trong các giải pháp cục bộ tối ưu.
* Chọn lọc: Các cá thể trong quần thể mới được chọn lọc để tiếp tục tiến hóa cho thế hệ tiếp theo.

Quá trình này được lặp lại cho đến khi đạt được một giải pháp tối ưu hoặc cho đến khi đạt được một số giới hạn về số lượng thế hệ.

**Các thành phần của giải thuật di truyền**

Giải thuật di truyền có bốn thành phần chính:

* Quần thể: Quần thể là một tập hợp các cá thể, mỗi cá thể đại diện cho một giải pháp tiềm năng cho bài toán.
* Hàm mục tiêu: Hàm mục tiêu đánh giá chất lượng của các cá thể trong quần thể.
* Các toán tử di truyền: Các toán tử di truyền bao gồm sàng lọc, tái tổ hợp, và đột biến.
* Các tham số: Các tham số của GA bao gồm kích thước quần thể, tỷ lệ tái tổ hợp, và tỷ lệ đột biến.

**Ứng dụng của giải thuật di truyền**

Giải thuật di truyền có thể được áp dụng để giải nhiều loại bài toán khác nhau, bao gồm:

* Tối ưu hóa: GA có thể được sử dụng để tìm kiếm các giải pháp tối ưu cho các bài toán tối ưu hóa, chẳng hạn như quy hoạch tuyến tính, tối ưu hóa liên tục, và tối ưu hóa bộ số.
* Tìm kiếm: GA có thể được sử dụng để tìm kiếm các giải pháp cho các bài toán tìm kiếm, chẳng hạn như tìm kiếm đường đi ngắn nhất, tìm kiếm cấu trúc tốt nhất, và tìm kiếm dữ liệu.
* Lập kế hoạch: GA có thể được sử dụng để lập kế hoạch cho các hệ thống phức tạp, chẳng hạn như lập kế hoạch sản xuất, lập kế hoạch vận tải, và lập kế hoạch chiến lược.

**Ưu điểm và nhược điểm của giải thuật di truyền**

Ưu điểm:

* GA có thể giải quyết nhiều loại bài toán khác nhau.
* GA có thể tìm kiếm các giải pháp toàn cục tối ưu.
* GA có thể hoạt động tốt với các bài toán có hàm mục tiêu phức tạp.

Nhược điểm:

* GA có thể tốn nhiều thời gian để chạy.
* GA có thể bị mắc kẹt trong các giải pháp cục bộ tối ưu.

**Kết luận**

Giải thuật di truyền là một thuật toán tìm kiếm tối ưu mạnh mẽ có thể được áp dụng để giải nhiều loại bài toán khác nhau. GA có thể tìm kiếm các giải pháp toàn cục tối ưu, nhưng có thể tốn nhiều thời gian để chạy và có thể bị mắc kẹt trong các giải pháp cục bộ tối ưu.

### 4.2 Áp dụng vào bài toán giải phương trình f(x) = 0

### 4.2.1 Phương pháp giải

**Biểu diễn lời giải dưới dạng gen**

       Đầu tiên ta cần biểu diễn lời giải bài toán dưới dạng gen. Trong trường hợp này, ta biểu diễn lời giải dưới dạng một chuỗi nhị phân 32 bit tuân theo tiêu chuẩn Single precision floating point format. (1 8 23)

       Ví dụ chuỗi “01000001101110000000000000000000” là biểu diễn nhị phân biểu diễn cho lời giải x = 23.5.

**Khởi tạo quần thể ban đầu**

       Tiếp theo, ta cần khởi tạo một quần thể ban đầu gồm các cá thể. Mỗi cá thể trong quần thể là một chuỗi nhị phân biểu diễn cho một lời giải. Các cá thể trong quần thể có thể được khởi tạo ngẫu nhiên hoặc dựa trên một số thông tin ban đầu về bài toán.

       Trong bài toán này, quần thể bao gồm các cá thể, mỗi cá thể có độ dài 32 bit được tạo ngẫu nhiên.

Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ

Mô tả được tạo tự động

Hàm “init\_Population” khởi tạo một quần thể với kích thước đặt trước. Sử dụng đối tượng Random(), mỗi cá thể được tạo ngẫu nhiên bằng cách thêm 32 bit “0” hoặc “1” vào chuỗi.

**Đánh giá các cá thể**

Mỗi cá thể trong quần thể sẽ được đánh giá bằng cách tính giá trị hàm mục tiêu. Giá trị hàm mục tiêu càng nhỏ thì cá thể càng tốt.

Trong bài toán này, ta đang chọn hàm mục tiêu là lf(x)l với f(x) là hàm số ta cần tìm nghiệm. Giá trị của hàm mục tiêu càng gần 0 thì cá thể càng tốt.

A screenshot of a computer program

Description automatically generated

Hàm “func” sẽ trả về giá trị của hàm mục tiêu của x.

**Chọn lọc**

       Các cá thể trong quần thể sẽ được chọn lọc để tham gia vào quá trình sinh sản. Các cá thể có giá trị hàm mục tiêu càng nhỏ thì khả năng được chọn càng cao.

Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ

Mô tả được tạo tự động

Trong bài toán này, ta sắp xếp các cá thể theo đáp ứng của hàm mục tiêu và  số cá thể của quần thể đáp ứng hàm mục tiêu tốt nhất sẽ được chọn và tham gia vào quá trình tiếp theo.

**Sinh sản**

Các cá thể được chọn lọc sẽ được sinh sản để tạo ra một quần thể mới. Quá trình sinh sản có thể được thực hiện theo các phương pháp sau:

* Lai: Hai cá thể được chọn ngẫu nhiên sẽ được lai tạo để tạo ra một cá thể mới.
* Đột biến: Một cá thể được chọn ngẫu nhiên sẽ bị đột biến để tạo ra một cá thể mới.

Trong bài toán này, quá trình lai được thực hiện bởi cách chọn ngẫu nhiên một điểm  cắt trong bộ gen để tạo ra con từ hai cá thể cha mẹ. Hàm lai được thực hiện trong đoạn code dưới đây.

Ảnh có chứa văn bản, Phông chữ, ảnh chụp màn hình

Mô tả được tạo tự động

Để tăng sự đa dạng cũng như để tránh kẹt ở một điểm nào đó trong quá trình tiến hóa, quá trình đột biến được sử dụng trong một vài cá thể.

Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ

Mô tả được tạo tự động

  Hàm “mutate” ngẫu nhiên đảo ngược các bit trong chuỗi nhị phân của cá thể với một xác suất là “mutationRate”.

       Ở bài toán này ta sử dụng phương pháp “Dynamic mutation rate” hay “Xác suất đột biến động”, xác suất đột biến sẽ giảm dần theo số thế hệ nhằm đảm bảo sự ổn định của quần thể về sau – khi quần thể dần hội tụ. Xác suất đột biến được biểu diễn như sau



Sau đó các cá thể mới sẽ thay thế các cá thể bị loại bỏ để tạo thành quần thể mới, các cá thể mới được tạo bởi cách lai từ các cá thể cha mẹ từ quần thể ban đầu sau đó đột biến.

Ảnh có chứa văn bản, ảnh chụp màn hình, phần mềm, Phần mềm đa phương tiện

Mô tả được tạo tự động

**Thay thế**

Quần thể mới sẽ thay thế quần thể cũ. Quá trình này sẽ được lặp lại cho đến khi tìm được nghiệm hoặc đạt số thế hệ tối đa.



Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ

Mô tả được tạo tự động

**Kết quả hiển thị**

Xử Lý Kết Quả Cuối Cùng: Trả về kết quả cuối cùng, bao gồm giá trị sai số tốt nhất và giá trị x tương ứng.

### 4.2.2 Kết Luận

Ta đã triển khai một giải thuật di truyền để giải phương trình, sử dụng quần thể của các chuỗi bit để tìm kiếm giá trị x. Biểu đồ giúp theo dõi sự hội tụ của giải thuật và đánh giá hiệu suất của nó qua các thế hệ.

Giải thuật đã giúp giải gần đúng phương trình f(x) = 0 với sai số không vượt quá 10-4.

**Mô phỏng**

A screenshot of a calculator

Description automatically generated

Nhập hàm cần tìm nghiệm

A screenshot of a calculator

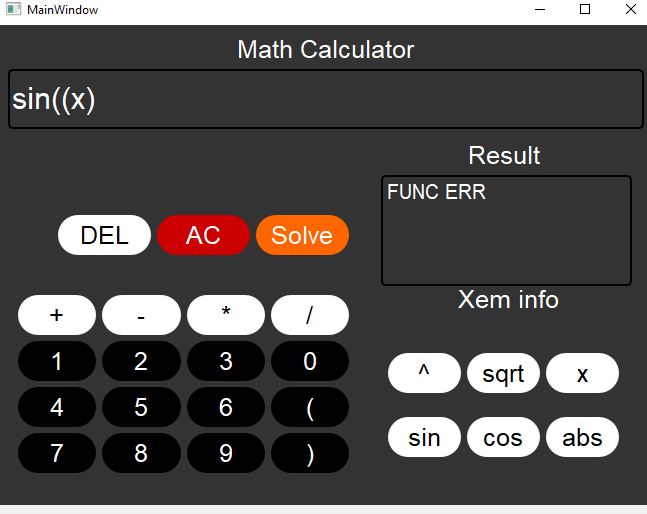
Description automatically generated

Ấn nút solve để hiển thị lên màn hình kết quả

A graph with a point

Description automatically generated

Ấn nút xem ìno để xem được quá trình tìm nghiệm



Nếu hàm có lỗi , kết quả sẽ hiển thị FUNC ERR