**LẬP LỊCH ĐƯỜNG ĐI CHO ROBOT TỰ TRỊ TRONG MÔI TRƯỜNG TĨNH CÓ CHƯỚNG NGẠI VẬT DỰA TRÊN PHÉP LAI TỐI ƯU HÓA BẦY ĐÀN LẠI VỚI GIẢI THUẬT TIẾN HÓA SAI KHÁC VI PHÂN**

Phạm Hoàng Hưng

*Trường Đại học Thủy lợi, email: hungph62@wru.vn*

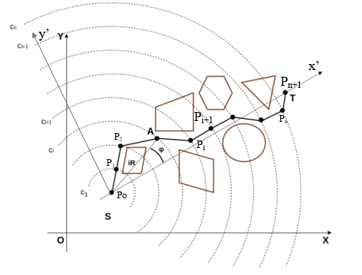
1. GIỚI THIỆU

Ngày nay cùng với sự phát triển của công nghệ, mobile robot đang là xu hướng phát triển của thế giới. Robot đang chứng tỏ có thể làm được nhiều thứ hơn và thay thế con người làm những công việc khó khăn nhất, như trong vận tải, y tế, cứ trợ cứu nạn hay thám hiểm. Tối ưu hóa năng lượng cho robot là một thách thức lớn đối với các nhà khoa học. Bài toán tìm đường đi của robot là một bài toán tìm kiếm và hoạch định đường đi cho robot sao đường đi ngắn nhất và không va chạm với chướng ngại vật. Bài toán được xếp vào lớp NP-khó. Trong bài báo này tôi xin giới thiệu đến phương pháp tìm đường đi dựa trên phép lai thuật toán tối ưu hóa bầy đàn (PSO) lai giải thuật tiến hóa sai khác vi phân.

1. KIẾN THỨC NỀN TẢNG
   1. **Mô hình hóa môi trường sử dụng cung tròn**

Để hoạch định đường đi hiệu quả, môi trường bản đồ của không gian làm việc trước tiên được chuyển đổi để làm cho robot hiểu

Có rất nhiều phương pháp mô hình hóa môi trường và trong đồ án này tôi sử dụng phương pháp mô hình hóa môi trường sử dụng cung tròn

**

**Hình 1.** *Mô hình hóa sử dụng cung tròn*

* 1. **Tính toán hàm mục tiêu**

Độ dài đường đi:

*fL*(Pa) = (𝑝0 , 𝑝𝑛+1)

Độ dài đoạn pipi+1 được tính theo góc 𝜑,𝑖+1 và 2 bán kính 𝑖𝑅, (𝑖 + 1)𝑅:

d(pipi+1) = R

Độ dài đoạn 𝑝𝑛, 𝑝𝑛+1:

d(pnpn+1) =

* 1. **Thuật toán tối ưu hóa bầy đàn**

Thuật toán tối ưu hóa bầy đàn (PSO) được giới thiệu năm 1995 bởi James Kennedy và Russell C. tại hội nghị IEEE, được lấy cảm hứng từ việc nghiên cứu hành vi xã hội của các loài động vật sống bầy đàn như chim, cá.

Công thức tính toán:

* + 𝑣𝑖(𝑡+1)=𝑤\*𝑣𝑖𝑗(𝑡)+𝑐1\*R1\*(Pbest𝑖𝑗(𝑡) − 𝑥𝑖𝑗(𝑡)) + 𝑐2\*𝑅2 \*(Gbest𝑖𝑗(𝑡) - 𝑥𝑖𝑗(𝑡))
  + x𝑖(𝑡 + 1) = 𝑥𝑖𝑗 (𝑡) + v𝑖𝑗 (𝑡 + 1)

Trong đó: 𝑖 = 1, 2, …, 𝑁; 𝑗 = 1, 2, …, 𝑛;

t: biến đếm vòng lặp

w: trọng số quán tính

c1, c2: hệ số gia tốc

* 1. **Giải thuật tiến hóa sai khác vi phân**

Giải thuật tiến hóa sai khác vi phân (DE) được Rainer Storn và Kenneth Price giới thiệu năm 1997, được sử dụng nhiều trong các bài toán tối ưu.

Vector đột điến được tạo ra như sau:

𝑣𝑖,+1 = 𝑥𝑟1,𝐺 + 𝐹 ∙ (𝑥𝑟2,𝐺 − 𝑥𝑟3,𝐺)

Với r1, r2, r3 ∈ {1, 2, … , 𝑁𝑃} là các số nguyên , ngẫu nhiên và khác nhau, F > 0

1. **THUẬT TOÁN TỐI ƯU HÓA BẦY ĐÀN LAI GIẢI THUẬT TIẾN HOA SAI KHÁC VI PHÂN**

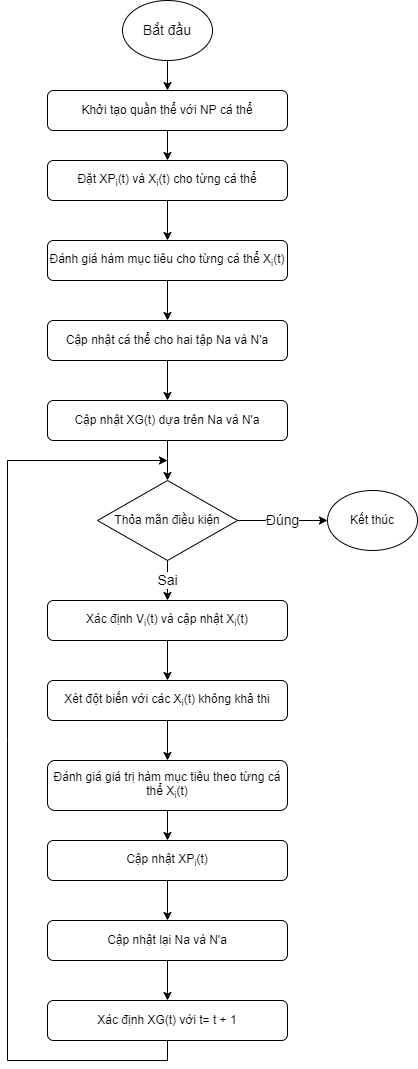
**Đầu vào:**

Môi trường tĩnh 2D, m chướng ngại vật, vị trí bắt đầu: S, vị trí kết thúc: T

**Đầu ra:**

Độ dài đường đi: 𝑃𝑎 = (𝑝0, 𝑝1, …, 𝑝𝑛, 𝑝𝑛+1)

**Lưu đồ thuật toán**



**Hình 2**. *Lưu đồ thuật toán*

1. **KẾT QUẢ THỰC NGHIỆM**

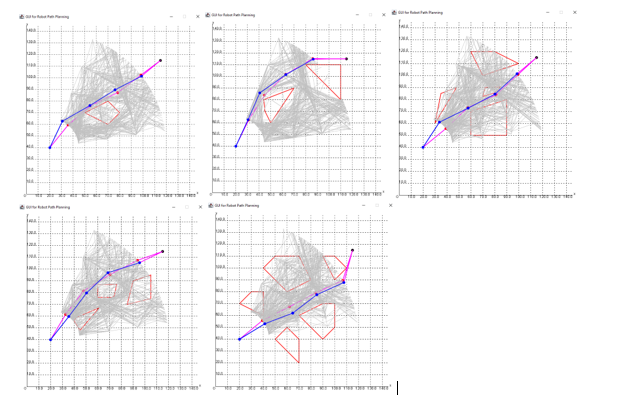
Đồ án thực hiện ba thực nghiệm để phân tích hiệu quả của giải thuật PSO lai DE dựa trên độ dài đường đi trung bình (L(tb))

**Thực nghiệm 1**: Ảnh hưởng của hình dạng chướng ngại vật

**Thực nghiệm 2**: Ảnh hưởng của mật độ chướng ngại vật

**Thực nghiệm 3:** Môi trường phòng thí nghiệm

### **Hình dạng chướng ngại vật**

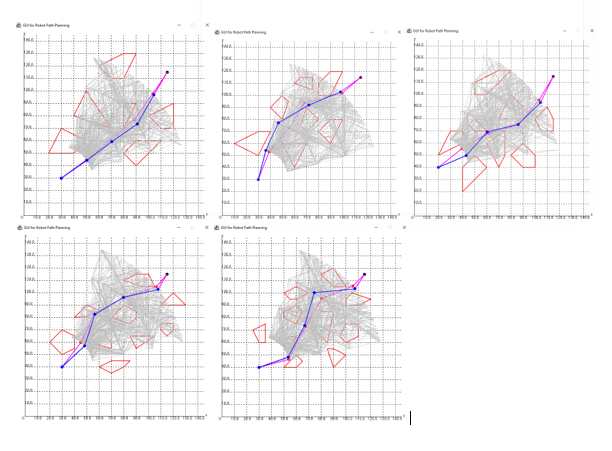


**Hình 3**. *Kết quả thực nghiệm 1*

|  |  |  |  |
| --- | --- | --- | --- |
| Bộ dữ liệu | Giải thuật | L(tb) | Tỉ lệ tc |
| Bộ dữ liệu 1 | PSO | 124.766 | 100% |
| PSODE | 124.6613 | 100% |
| Bộ dữ liệu 2 | PSO | 136.2387 | 100% |
| PSODE | 134.214 | 100% |
| Bộ dữ liệu 3 | PSO | 125.1215 | 100% |
| PSODE | 124.0198 | 100% |
| Bộ dữ liệu 4 | PSO | 123.7699 | 100% |
| PSODE | 122.6976 | 100% |
| Bộ dữ liệu 5 | PSO | 126.8717 | 100% |
| PSODE | 125.6711 | 100% |

Ở thưc nghiệm 1 ta thấy được hai thuật toán đều chạy ổn định nhưng thuật toán PSO lai DE có sự tốt hơn về đường đi với chung hình dạng chướng ngại vật, điểm bắt đầu, kết thúc

### **Mật độ chướng ngại vật**

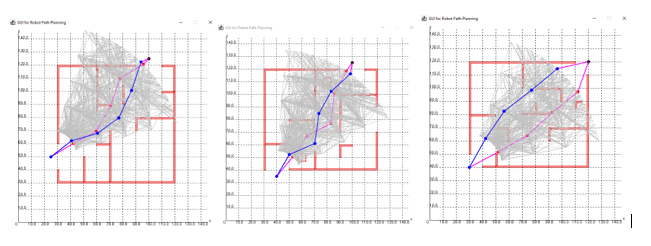


**Hình 4**. *Kết quả thực nghiệm 2*

|  |  |  |  |
| --- | --- | --- | --- |
| Bộ dữ liệu | Giải thuật | L(tb) | Tỉ lệ tc |
| Bộ dữ liệu 6 | PSO | 126.0912 | 100% |
| PSODE | 124.6364 | 100% |
| Bộ dữ liệu 7 | PSO | 126.4448 | 100% |
| PSODE | 125.1059 | 100% |
| Bộ dữ liệu 8 | PSO | 126.1613 | 100% |
| PSODE | 126.9431 | 100% |
| Bộ dữ liệu 9 | PSO | 124.6114 | 100% |
| PSODE | 128.6075 | 100% |
| Bộ dữ liệu 10 | PSO | 127.5019 | 100% |
| PSODE | 125.3432 | 100% |

Ở thực nghiệm 2 với mật độ chướng ngại vật tăng dần nhưng thuật toán PSO lai DE vẫn tốt hơn PSO về đường đi

* 1. **Trong môi trường phòng thí nghiệm**



**Hình 5**. *Kết quả thực nghiệm 3*

|  |  |  |  |
| --- | --- | --- | --- |
| Bộ dữ liệu | Giải thuật | L(tb) | Tỉ lệ tc |
| Bộ dữ liệu 1 | PSO | 112.2969 | 86,6666% |
| PSODE | 111.969 | 86,6666% |
| Bộ dữ liệu 2 | PSO | 111.7923 | 76.6666% |
| PSODE | 111.4091 | 76.6666% |
| Bộ dữ liệu 3 | PSO | 125.2149 | 73,3333% |
| PSODE | 124.5118 | 73,3333% |

Với môi trường phòng thí nghiệm và quá nhiều lần chạy thì thuật toán PSO lai DE vẫn cho thấy sư vượt trội về đường đi so với PSO

1. **TÀI LIỆU THAM KHẢO**

|  |  |
| --- | --- |
| [1] | J. Kennedy and R. Eberhart, "Particle swarm optimization", in *Proceedings of ICNN'95 - International Conference on Neural Networks*, Perth, WA, Australia, Australia, 1995. |
| [2] | Zhang et al," Path Planning of Mobile Robot Based on Hybrid Multi-Objective Bare Bones Particle Swarm Optimization With Differential Evolution" 2018. |
| [3] | K. Price et al, "Differential Evolution – A Simple and Efficient Heuristic for global Optimization over Continuous Spaces," 1997. |