Bài toán khuếch tán : Diffusion Limited Aggregation

Nhóm 10
Nguyễn Trung Hiếu - Trần Xuân Lương

Viện Công nghệ thông tin truyền thông
Đại học Bách khoa Hà Nội

12/12/2019

Nội dung

1. Bài toán Diffusion Limited Aggregation
2. Mô hình bài toán DLA
3. Thuật toán mô phỏng DLA
4. Phương pháp lặp Successive Over Relaxation
5. Song song hóa quá trình khuếch tán
6. Quá trình phát triển
7. Song song hóa quá trình phát triển

Bài toán Diffusion Limited Aggregation

- Diffusion Limited Aggregation (DLA) là mô hình phát triển không cân bằng và sự phát triển được xác định bởi sự khuếch tán.
- Ví dụ: sự phát triển của trực khuẩn
- Trực khuẩn sẽ ăn chất dinh dưỡng trong môi trường .
- Sự phát triển được xác định bởi nồng độ chất dinh dưỡng và nồng độ này sẽ được tính dựa trên sự khuyếch tán.

Mô hình bài toán DLA

- Cho một lưới ô vuông kích thước NxN, mỗi ô trên lưới có thể chứa thức ăn hoặc vật thể.
- Các ô $[0, j]$ $(0 \leq i \leq N-1)$ là các ô source $(c = 1)$.
- Các ô $[N-1, j]$ $(0 \leq j \leq N-1)$ hoặc bị vật thể chiếm là các ô sink $(c = 0)$
- Quá trình mô phỏng sẽ thực hiện theo thuật toán DLA.

Thuật toán mô phỏng DLA

Thuật toán DLA:

- Bước 1: Giải phương trình Laplace để tìm sự phân bố của chất dinh dưỡng, coi các điểm của vật thể là sink ($c = 0$).
- Bước 2: Cho vật thể phát triển.
- Bước 3: Quay lại bước 1.

Để giải phương trình Laplace, ta sử dụng phương pháp lặp Successive Over Relaxation (SOR).
Để phát triển vật thể, chúng ta cần thực hiện 3 bước sau:

- Xác định các candidate
- Xác định xác suất phát triển
- Phát triển

Phương pháp lặp Successive Over Relaxation

Công thức lặp:

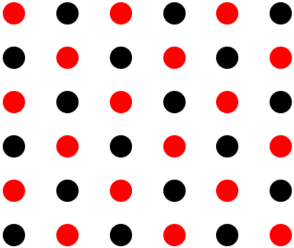
$$c_{l,m}^{(n+1)} = \frac{\omega}{4}[c_{l+1,m}^{(n)} + c_{l-1,m}^{(n+1)} + c_{l,m+1}^{(n)} + c_{l,m-1}^{(n+1)}] + (1-\omega)c_{l,m}^{(n)} \tag{1}$$

- $c_{l,m}^{i}$ là lượng dinh dưỡng ở ô $[l, m]$ sau bước lặp i
- ω là hệ số trộn
- Với $1 < \omega < 2$, ta gọi đây là phương pháp Successive Over Relaxation.

Điều kiện dừng lặp:

$$\max_{l,m} \mid c_{l,m}^{i+1} - c_{l,m}^{i} \mid < \epsilon \tag{2}$$

Song song hóa quá trình khuếch tán



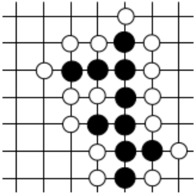
- Nhận xét:Ta có thể thấy các ô đỏ chỉ phụ thuộc vào các ô đen và ngược lại. Do đó, có thể tính toán song song các ô đỏ cũng như các ô đen. Ngoài ra, các ô đỏ sẽ tính dựa vào bước i còn các ô đen sẽ dựa vào bước $i + 1$.
- Để song song hóa quá trình khuếch tán:
  - Tính các ô đỏ trước các ô đen
  - Song song hóa quá trình tính toán các ô cùng màu theo hàng.

Song song hóa quá trình khuếch tán

- Khởi tạo m tiến trình.
- Chia bảng ra thành m khối, mỗi khối là một bảng gồm N/m hàng và N cột. Mỗi tiến trình đảm nhiệm 1 khối.
- Tại mỗi bước lặp, ta sẽ tính các ô đỏ trước tương đương các ô đỏ $c_{l,m}$ với $(l+m)\%2 = 0$, sau đó tính các ô đen $c_{l,m}$ với $(l+m)\%2 = 1$.
- Công thức tính dựa theo SOR:
  - $c_{l,m} = 0, \forall (l,m) \in Object$
  - $c_{N-1,m} = 0, \forall m \in [0, N-1]$
  - $c_{0,m} = 1, \forall m \in [0, N-1]$
  - $c_{l,m} = \frac{\omega}{4}[c_{l+1,m} + c_{l-1,m} + c_{l,m+1} + c_{l,m-1}] + (1-\omega)c_{l,m}, \forall l, m \in [1, N-2]$
- Sau khi cập nhật xong thì gửi hàng đầu cho tiến trình phía trên và gửi hàng dưới cùng cho tiến trình phía dưới.

Quá trình phát triển

- Xác định candidate:
  - Candidate là các ô mà một trong các ô kề với nó thuộc vật thể



- Xác định xác suất phát triển:
  - Công thức xác suất phát triển:

$$p_g((l,m) \in Can) = \frac{(c_{l,m})^\eta}{\sum_{(l,m)\in Can}(c_{l,m})^\eta} \tag{3}$$

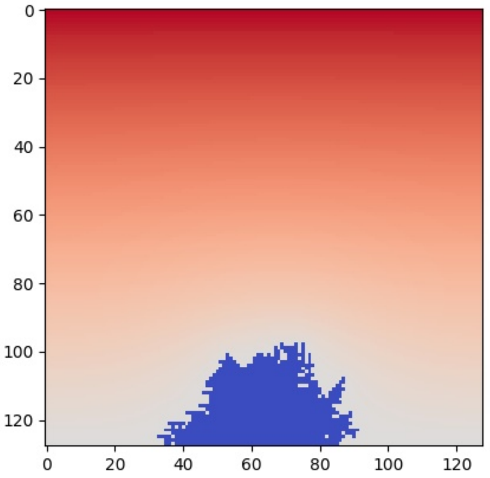
- Phát triển:
  - Công thức phát triển:

$$g_{l,m} = I\{rand(0,1) < p_g(l,m)\} \tag{4}$$

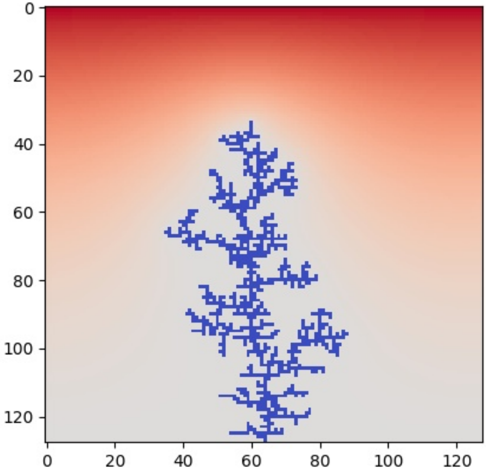
Song song hóa quá trình phát triển

- Song song hóa theo hàng
- Xác định candidate:
  - Tại mỗi ô, xác định xem các ô kề nó có thuộc vật thể hay không
  - Nếu có thì đánh dấu lại, không thì bỏ qua
- Tính xác suất phát triển:
  - Tính tổng $c_{l,m}$ trong tập candidate
  - Tính xác suất cho từng candidate
- Phát triển:
  - Phát triển dựa theo công thức

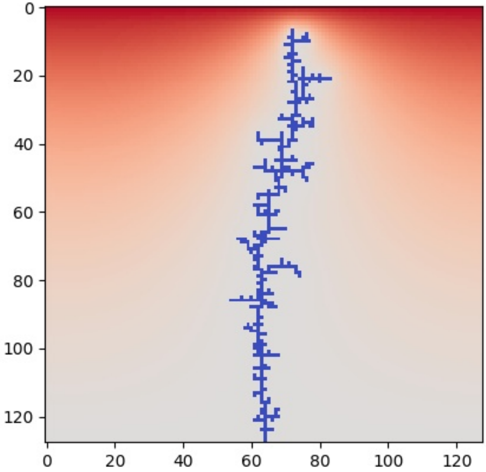
$N = 128$, $\eta = 0.0$, $iter = 1000$



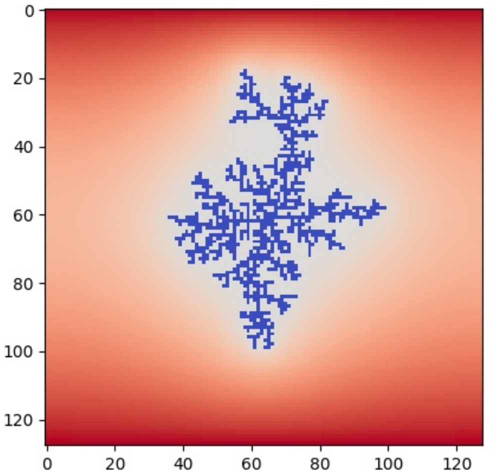
$N = 128$, $\eta = 1.0$, $iter = 1000$



$N = 128$, $\eta = 2.0$, $iter = 400$



$N = 128$, $\eta = 1.0$, $iter = 1000$



The End