

AI

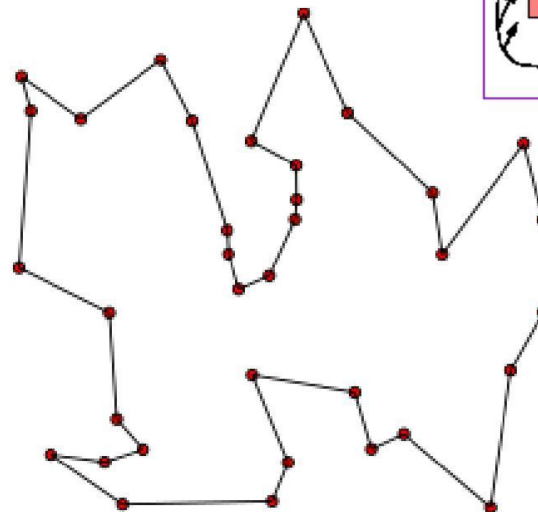
TÌM KIẾM ĐỊA PHƯƠNG VÀ TỐI ƯU HÓA CÁC VẤN ĐỀ

Nguyễn Ngọc Thảo - Nguyễn Hải Minh
{nnthao, nhminh}@fit.hcmus.edu.vn

Đề cương

- Các vấn đề về tìm kiếm và tối ưu hóa cục bộ
- Thuật toán tìm kiếm cục bộ
- Thuật toán tiến hóa

-
- An 8x8 chessboard with columns labeled 1 to 8 and rows labeled 1 to 8. The board has a standard alternating light and dark square pattern. The pieces are positioned as follows:
- Black King: e1
 - White King: d8
 - Black Pawns: a5, b2, c4, f3

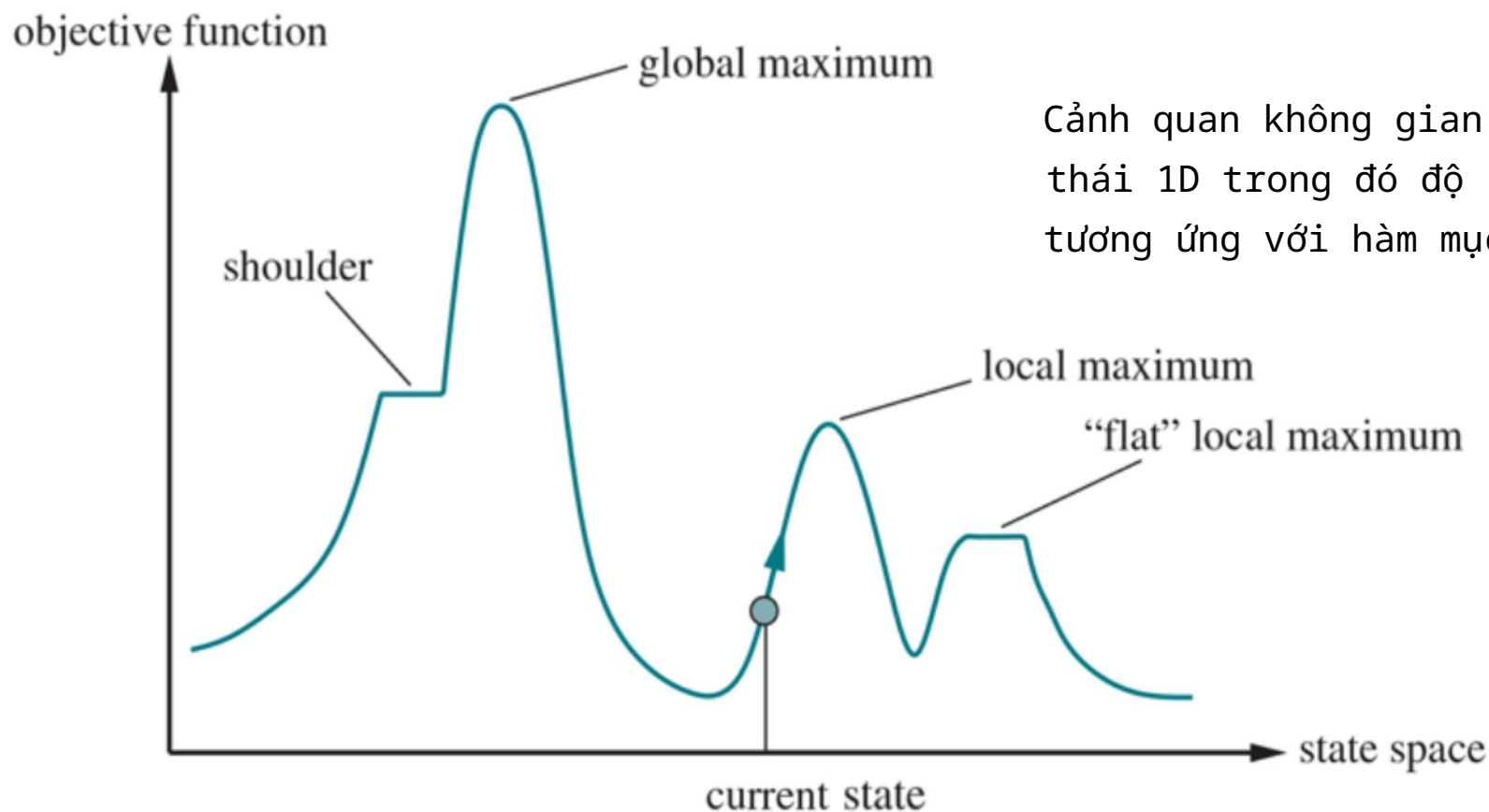


Thuật toán tìm kiếm cục bộ

- Các thuật toán này điều hướng từ trạng thái bắt đầu đến trạng thái lân cận mà không cần theo dõi đường đi cũng như tập hợp các trạng thái đạt được.
- Không có tính hệ thống
 - Họ có thể không bao giờ khám phá được một phần không gian tìm kiếm nơi có giải pháp.
- Tìm kiếm cục bộ có hai ưu điểm chính:
 - Sử dụng rất ít bộ nhớ
 - Tìm lời giải hợp lý trong không gian trạng thái lớn hoặc vô hạn mà các thuật toán hệ thống không phù hợp.

Thuật toán tìm kiếm cục bộ

- Tìm kiếm cục bộ có thể giải quyết các vấn đề tối ưu hóa, tìm thấy trạng thái tốt nhất theo hàm mục tiêu.



Tìm kiếm leo đồi

- Thuật toán đi theo hướng có độ dốc lớn nhất đi lên và kết thúc khi đạt đến “đỉnh”.

- Đỉnh = trạng thái không có hàng xóm nào có giá trị cao hơn.

hàm HILL-CLIMBING(sự cố) trả về trạng thái tối đa cục bộ

vấn đề hiện tại. INITIAL trong khi

đúng là làm

hàng xóm ý người kế thừa có giá trị cao nhất hiện tại nếu

VALUE(hàng xóm) > VALUE(hiện tại) sau đó trả về hiện tại ý hàng xóm

Leo đồi chỉ theo dõi một trạng thái hiện tại và trong mỗi lần lặp sẽ di chuyển đến bang lân cận có giá trị cao nhất.

Leo đòi cho bài toán 8 quân hậu

- Công thức trạng thái hoàn chỉnh: tất cả quân hậu trên bàn cờ, mỗi quân một cột
- Chức năng kế tiếp: di chuyển quân hậu sang ô khác trong cùng ô cột ý mỗi bang có $8 \times 7 = 56$ bang kế tiếp
- ý \Rightarrow (số) cặp quân hậu đang tấn công lẫn nhau ý mức tối thiểu toàn cầu có ý $() = 0$

18	12	14	13	13	12	14	14
14	16	13	15	12	14	12	16
14	12	18	13	15	12	14	14
15	14	14	♔	13	16	13	16
♔	14	17	15	♔	14	16	16
17	♔	16	18	15	♔	15	♔
18	14	♔	15	15	14	♔	16
14	14	13	17	12	14	12	18

Bảng hiển thị giá trị của mỗi khả năng

người kế vị có được bằng cách di chuyển một nữ hoàng trong cột.

Trạng thái hiện tại n có ý $= 17 \Rightarrow (4)_{3 \ 2 \ 5 \ 4}$
 $(\quad 1 \ 2 \ 3 \ 4 \ 5 \ 6 \ 7 \ 8 \quad)_{3 \ 2 \ 3}$

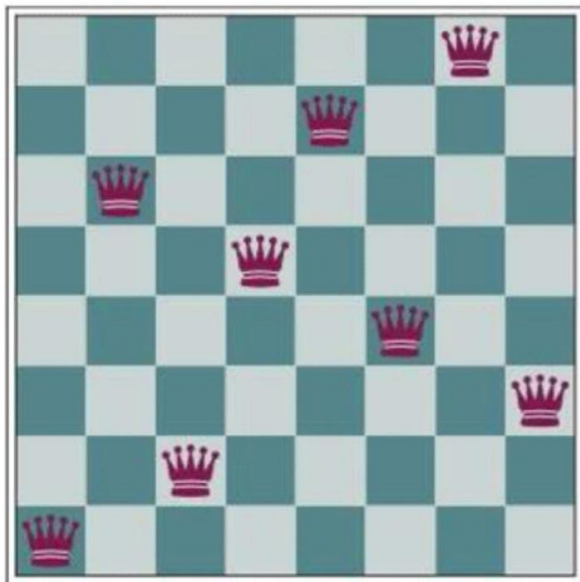
Có 8 nước đi hòa nhau nhất, với h =

12. Thuật toán leo đòi sẽ chọn một trong số đó.

Leo đòi: Chưa tối ưu

- Leo đòi còn được gọi là **tham lam tìm kiếm địa phương**.
 - Nó chộp lấy một quốc gia láng giềng tốt mà không cần nghĩ tới nơi tiếp theo.
- Nó có thể dễ dàng cải thiện tình trạng xấu và do đó đạt được tiến bộ nhanh chóng trong việc tìm ra giải pháp.
 - 8 nữ hoàng: 17 triệu bang, trung bình 4 bước khi thành công và 3 khi gặp khó khăn.
- Việc leo đòi có thể bị mắc kẹt ở **các cực đại, rặng núi** hoặc **cao nguyên** địa phương

Phần mở rộng và rặng núi địa phương

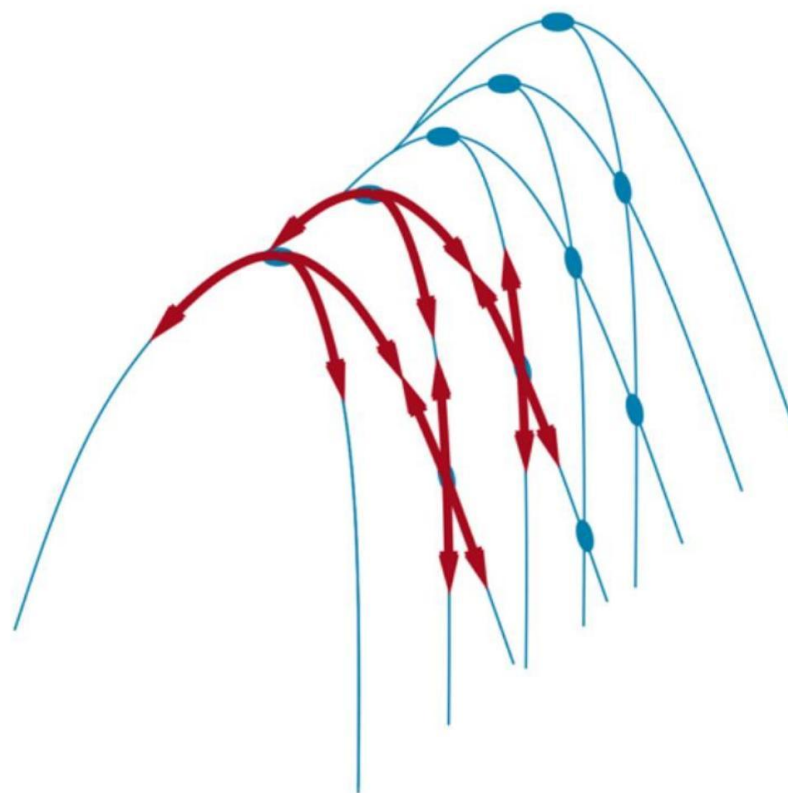


• Trạng thái hiện tại (1 6 2 5 7 4 8 3) có $\dot{y} = 1$

()

• Mọi người kế nhiệm đều có chi phí cao hơn

\dot{y} mức tối thiểu cục bộ

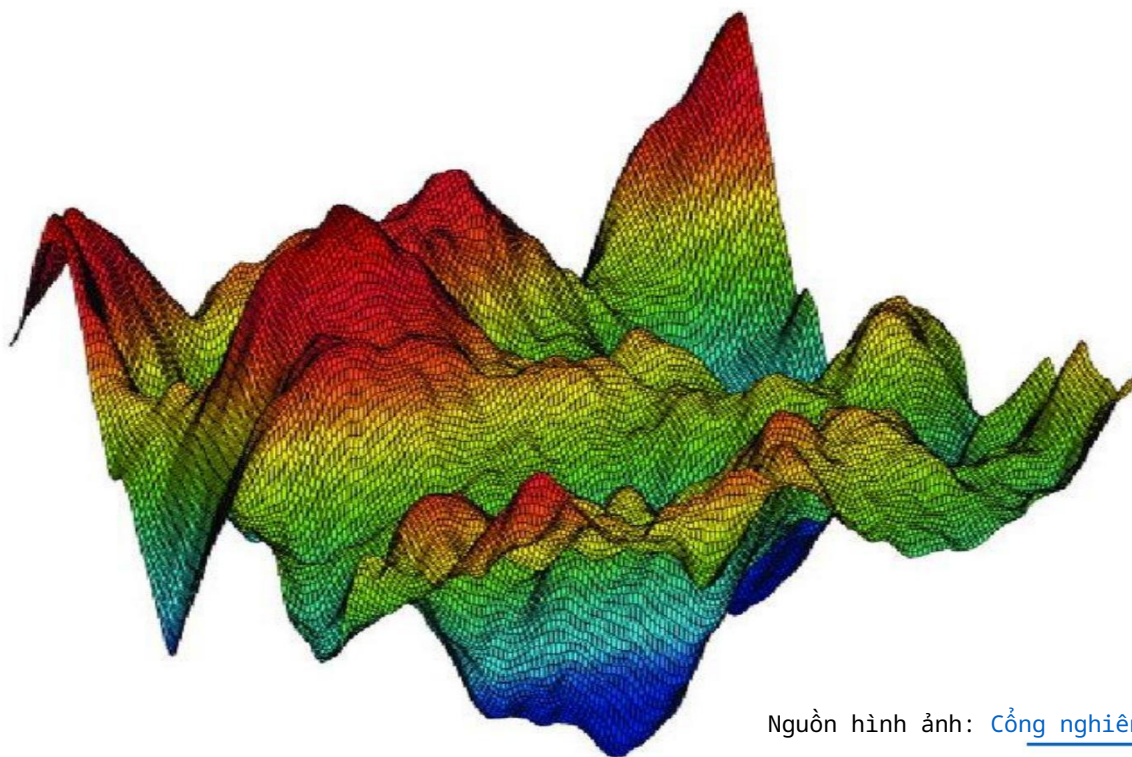


Lưới các trạng thái (vòng tròn tối) được đặt trên một sườn núi
tăng dần từ trái sang phải, tạo thành một chuỗi các trạng thái cục bộ.
cực đại

Từ mỗi mức tối đa cục bộ, tất cả các hành động sẵn có đều hướng xuống dưới.

Phần mở rộng và rặng núi địa phương

- Các bài toán trong thế giới thực hoặc các bài toán NP-khó thường có số cực đại cục bộ theo cấp số nhân cần giải quyết.



Nguồn hình ảnh: [Cổng nghiên cứu](#)

Vượt qua tình trạng dưới mức tối ưu

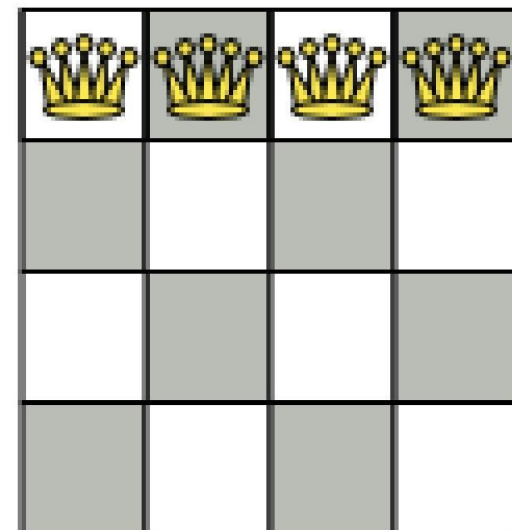
- Việc di chuyển sang một bên cho phép tác nhân tiếp tục đi ở vùng cao nguyên.
 - Nó không hoạt động ở mức tối đa cục bộ cố định.
- Cần hạn chế số lần di chuyển ngang liên tiếp để tránh tình trạng đi chệch hướng không ngừng.
- Cách tiếp cận này làm tăng tỷ lệ các trường hợp có vấn đề giải quyết bằng cách leo đồi.
 - Ví dụ, đối với bài toán 8 quân hậu: từ 14% đến 94%.
 - Thành công phải trả giá: khoảng 21 bước cho mỗi trường hợp thành công và 64 bước cho mỗi trường hợp sự thất bại

Vượt qua tình trạng dưới mức tối ưu

- Leo đồi ngẫu nhiên chọn ngẫu nhiên trong số các bước di chuyển lên dốc.
 - Xác suất lựa chọn có thể thay đổi tùy theo độ dốc của biến động.
 - Hội tụ chậm hơn nhưng giải pháp tốt hơn trong một số bối cảnh trạng thái
- Việc leo đồi được lựa chọn đầu tiên sẽ tạo ra những người kế nhiệm cho đến khi có được một cái tốt hơn tình trạng hiện tại.
 - Thích hợp khi một bang có nhiều (ví dụ hàng nghìn) người kế nhiệm
- Khởi động lại ngẫu nhiên leo đồi tiến hành một số tìm kiếm leo đồi từ trạng thái ban đầu ngẫu nhiên cho đến khi tìm thấy mục tiêu.
 - Nếu mỗi lần tìm kiếm đều có xác suất thành công thì số lượng dự kiến số lần khởi động lại được yêu cầu là $1/p$.

Câu đố 01: Bài toán 4 quân hậu

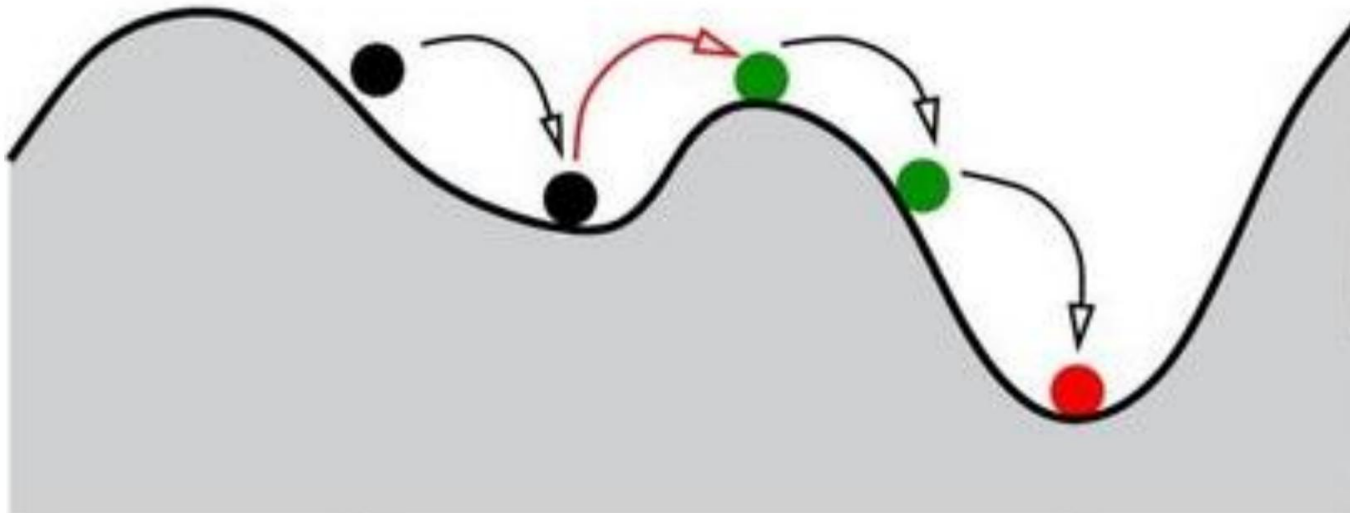
- Xét bài toán 4 quân hậu sau đây



- Áp dụng kỹ thuật leo đồi để tìm giải pháp bằng cách sử dụng phương pháp heuristic “The số lượng các cặp quân hậu tấn công lẫn nhau.”

Ủ mô phỏng

- Kết hợp leo đồi với đi bộ ngẫu nhiên theo một cách nào đó mang lại cả hiệu quả và sự hoàn thiện



Lắc mạnh (tức là ở nhiệt độ cao) rồi giảm dần cường độ lắc (tức là giảm nhiệt độ)

Ủ mô phỏng

một chức năng ánh xạ từ
thời gian đến “nhiệt độ”

hàm SIMULATED-ANNEALING(vấn đề, lịch trình) trả về trạng thái giải pháp

hiện tại y vấn đề. INITIAL cho $t = 1$ để y

làm

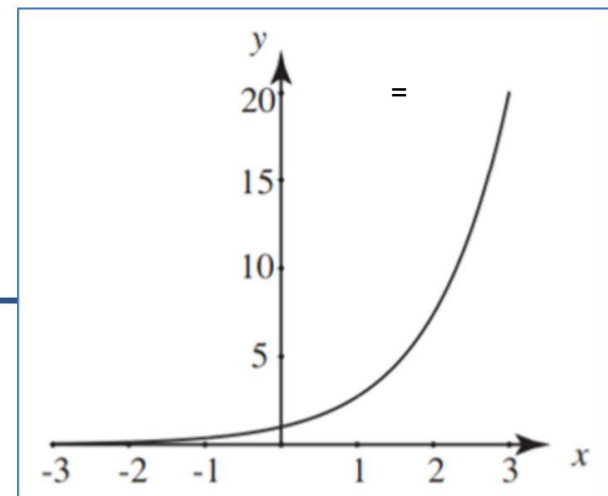
T lịch trình(t) nếu T

$= 0$ thì trả về dòng điện

tiếp theo y người kế thừa được chọn ngẫu nhiên của yE hiện tại y

$VALUE(next) - VALUE(current)$ nếu $yE > 0$ thì hiện tại y next

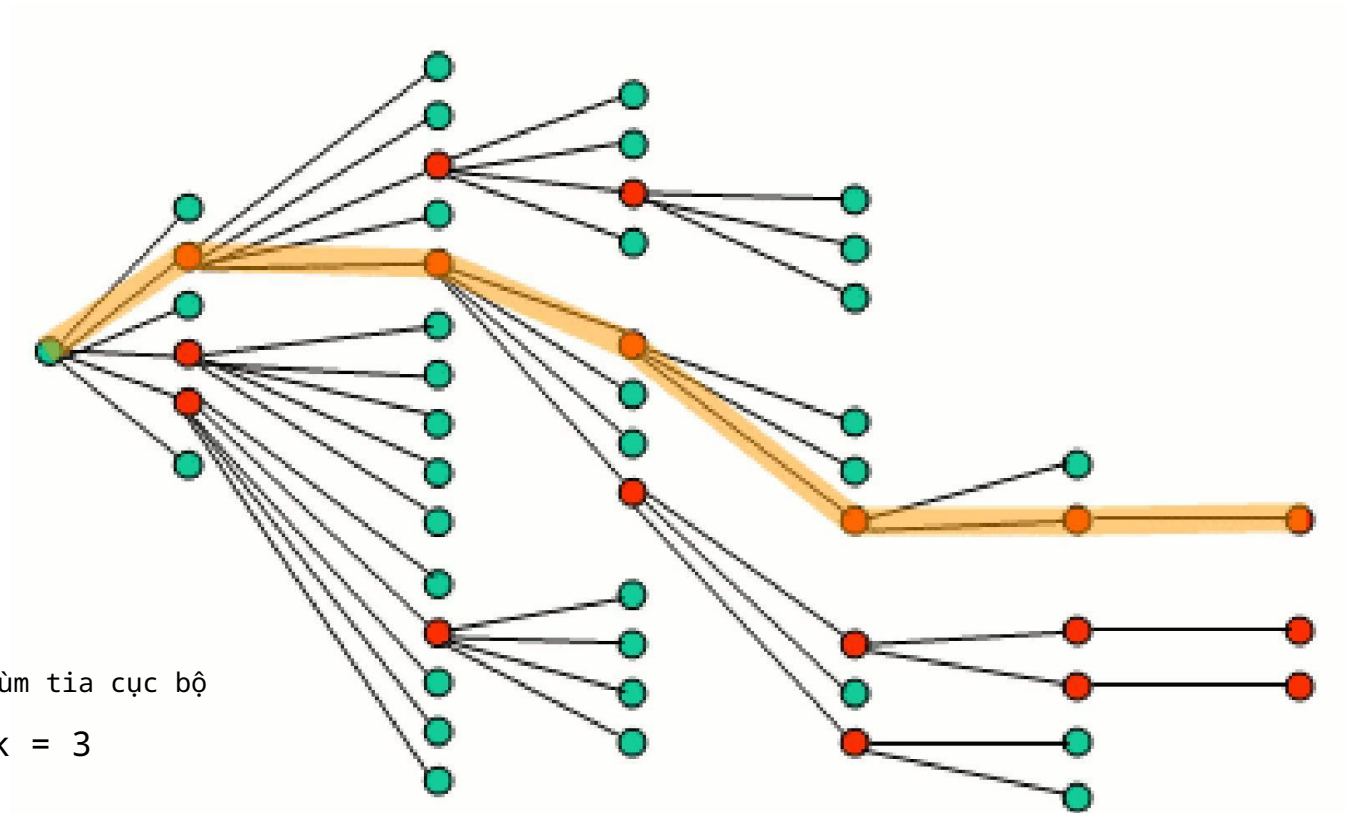
dòng điện khác y tiếp theo chỉ với xác suất $e^{-yE/T}$



Tìm kiếm chùm tia địa phương

- Chỉ giữ một nút trong bộ nhớ có vẻ là một điều cực đoan phản ứng với vấn đề hạn chế trí nhớ.
- Thuật toán **theo dõi các trạng thái** thay vì chỉ một trạng thái.
- Nó bắt đầu với các trạng thái được tạo ngẫu nhiên.
- Ở mỗi bước, tất cả trạng thái kế thừa của tất cả các trạng thái đều được tạo ra
- Nếu bất kỳ mục tiêu nào là mục tiêu thì thuật toán sẽ dừng lại. Ngược lại, nó sẽ chọn những người kế thừa tốt nhất từ danh sách đầy đủ và lặp lại.

Tìm kiếm chùm tia địa phương



Tìm kiếm chùm tia cục bộ
với $k = 3$

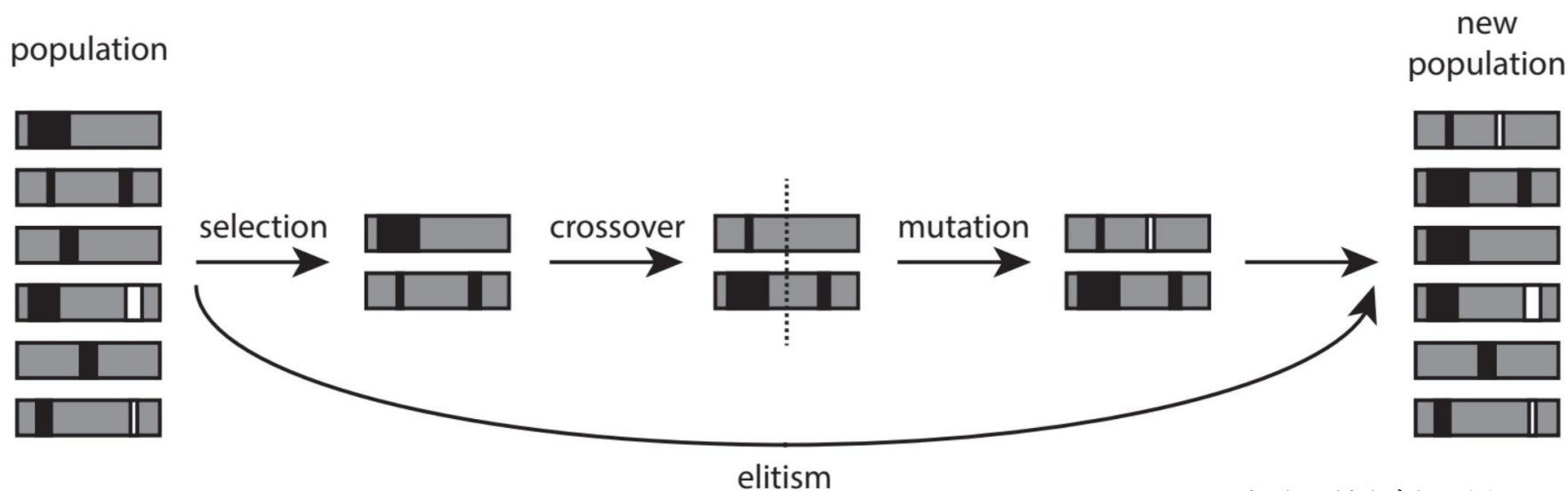
- Thuật toán nhanh chóng loại bỏ các tìm kiếm không hiệu quả và di chuyển nguồn lực đến nơi đạt được tiến bộ nhất.

Tìm kiếm chùm tia cục bộ

- Thông tin hữu ích được truyền giữa các luồng tìm kiếm song song ý sự khác biệt lớn so với tìm kiếm khởi động lại ngẫu nhiên
- Thuật toán có thể thiếu đa dạng giữa các các tiểu bang.
 - Các trạng thái có thể được nhóm lại trong một vùng nhỏ của không gian trạng thái ý một phiên bản đất tiền của leo đồi.
- Tìm kiếm chùm ngẫu nhiên có thể làm giảm bớt vấn đề trên bằng cách chọn ngẫu nhiên những phần tử kế tiếp theo giá trị của chúng.

Thuật toán tiến hóa

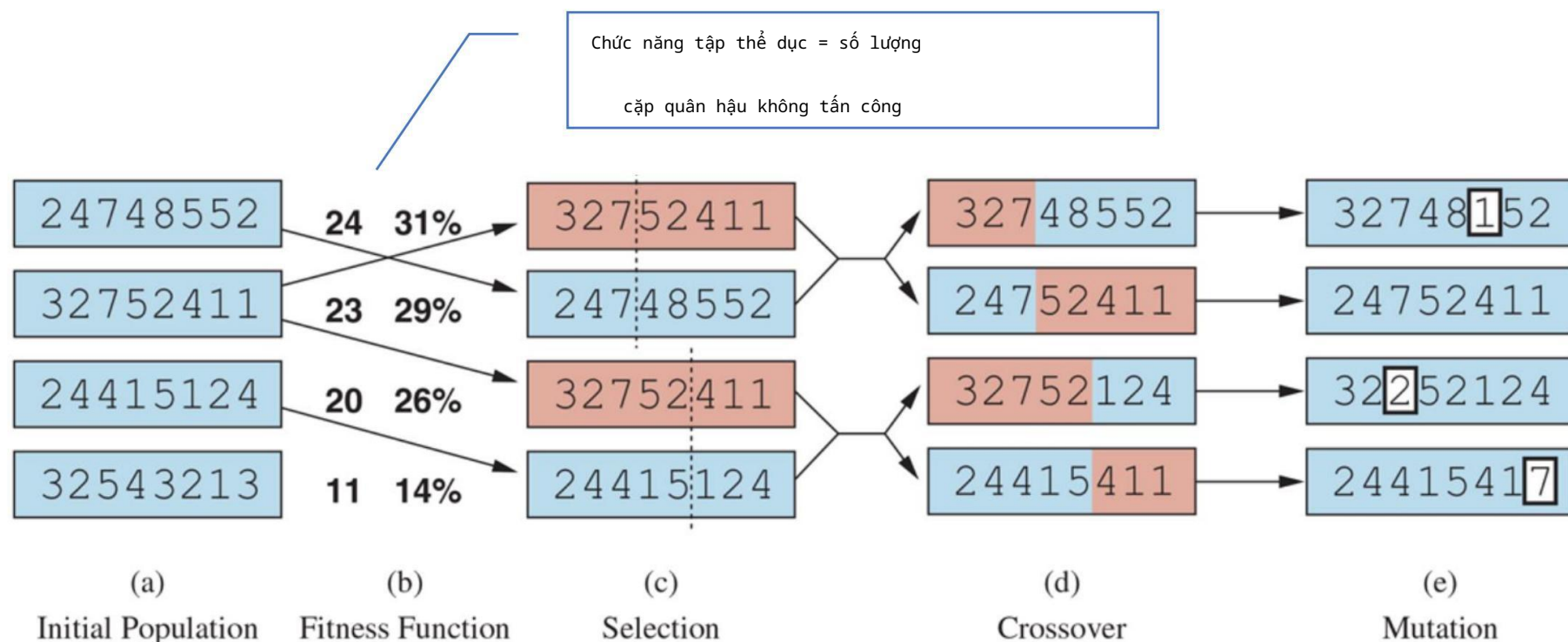
- Các biến thể của tìm kiếm chùm tia ngẫu nhiên, được thúc đẩy rõ ràng bởi ẩn dụ về chọn lọc tự nhiên trong sinh học



Tín dụng hình ảnh: [Mdpi](#)

Có một dân số gồm các cá nhân (tiểu bang). Các cá thể mạnh nhất (có giá trị cao nhất) sinh ra con cái (trạng thái kế tiếp) để sinh sống ở thế hệ tiếp theo.

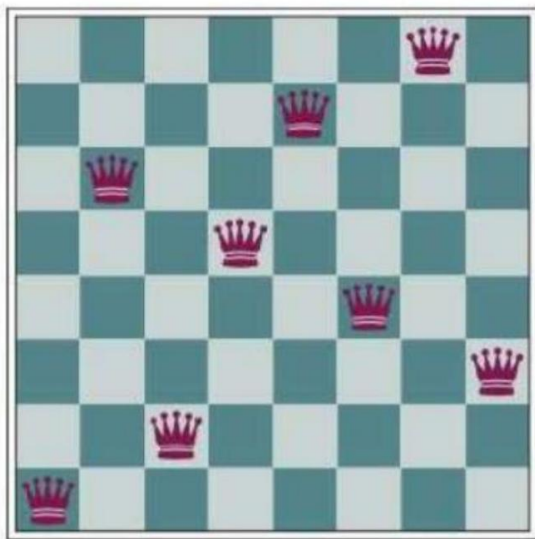
Thuật toán tiến hóa



Một thuật toán di truyền, được minh họa cho các chuỗi chữ số biểu thị trạng thái 8 quân hậu. Quần thể ban đầu ở (a) được xếp hạng theo hàm thích nghi ở (b) dẫn đến các cặp để giao phối ở (c). Chúng sinh ra con cái ở (d), chúng có thể bị đột biến ở (e).

Thuật toán tiến hóa

- Có vô số dạng thuật toán tiến hóa.
- **Sự đại diện của một cá nhân**
 - Thuật toán di truyền: một chuỗi trên một bảng chữ cái hữu hạn.
 - Lập trình di truyền: một chương trình máy tính. Chiến lược tiến hóa: a dãy số thực.



- Biểu diễn chữ số 1
 $\langle 6 \ 2 \ 5 \ 7 \ 4 \ 8 \ 3 \ \rangle$
- Biểu diễn nhị phân
 $\langle 000 \ 101 \ 001 \ 100 \ 110 \ 011 \ 111 \ 010 \ \rangle$

Thuật toán tiến hóa

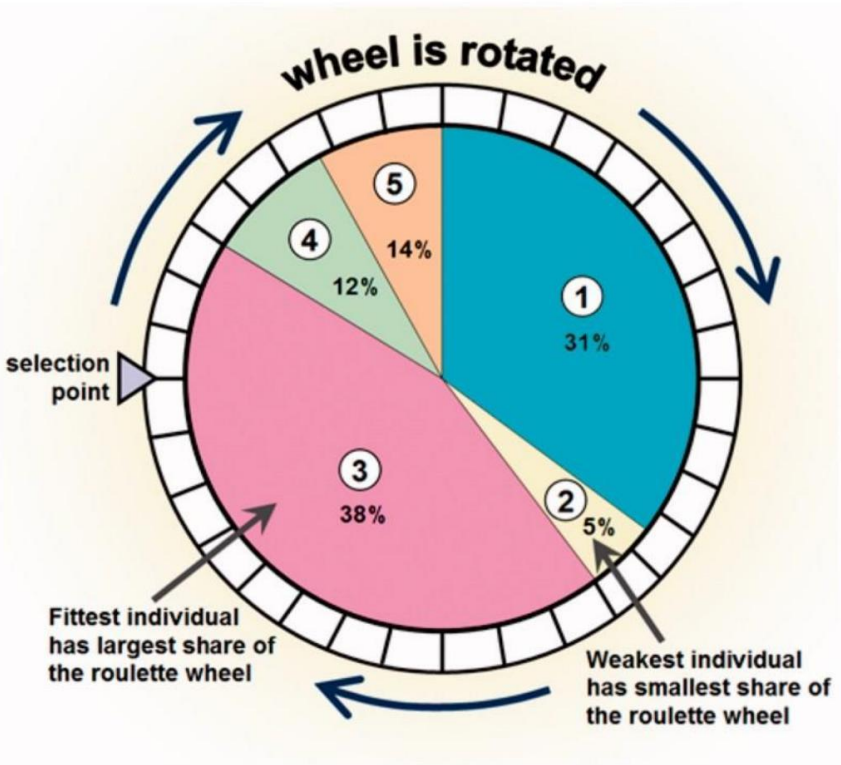
- Quy mô dân số
 - Ban đầu, một quần thể là một tập hợp các trạng thái được tạo ngẫu nhiên.
- Hàm thích hợp: hàm mục tiêu đánh giá từng trạng thái
 - Giá trị càng cao, trạng thái càng tốt
- Số trộn : số bố mẹ kết hợp với nhau để hình thành con cái
 - = 1: tìm kiếm chùm ngẫu nhiên. = 2: phổ biến nhất.

Thuật toán tiến hóa

- Quy trình tuyển chọn: chọn các cá nhân là cha mẹ của thế hệ kế tiếp
 - Các cá nhân có thể được chọn với xác suất tỷ lệ thuận với khả năng của họ điểm thể lực.

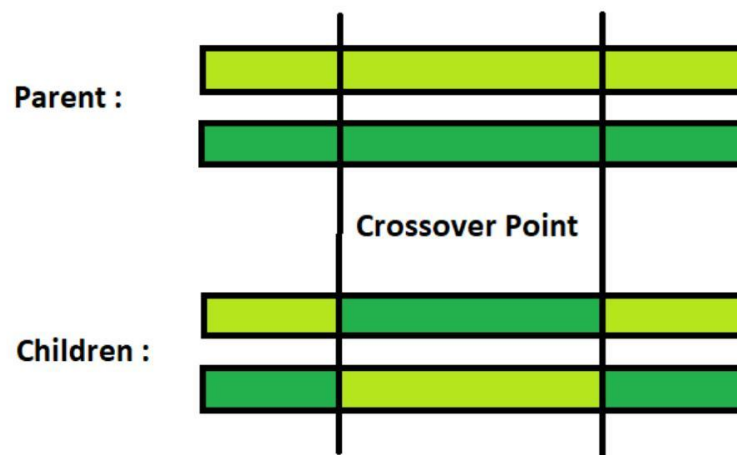
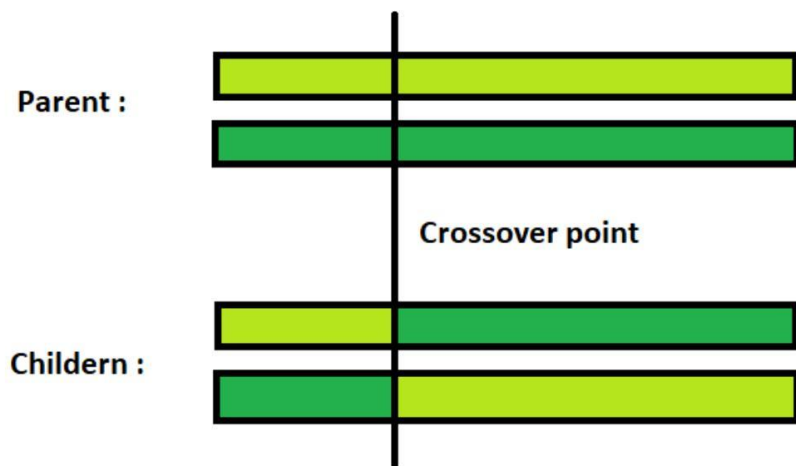
Phương pháp bánh xe Roulette

Cá nhân	Tỷ lệ thể lực (%)
1	31
2	5
3	38
4	12
5	14



Thuật toán tiến hóa

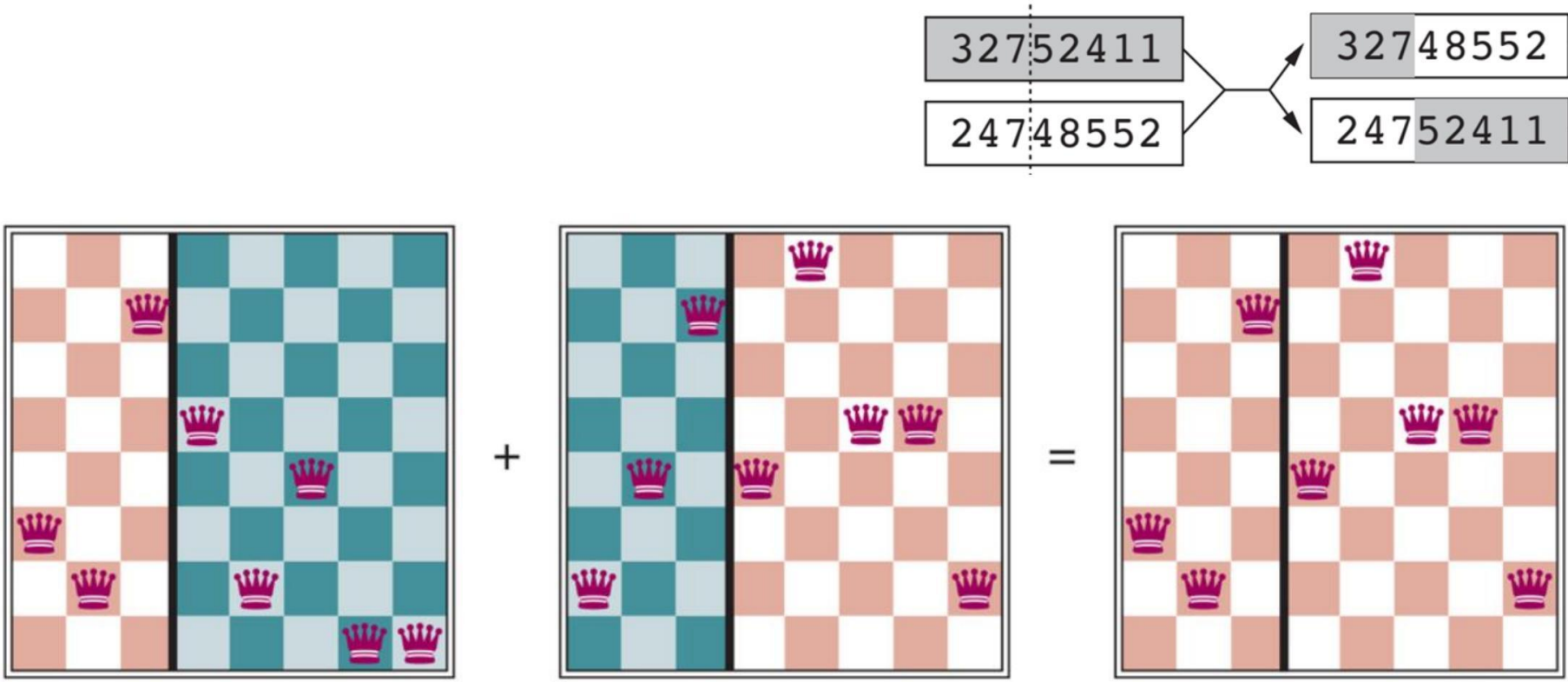
- Thủ tục tái hợp để hình thành con cái
 - Nó chọn ngẫu nhiên một **điểm giao nhau** để phân tách từng chuỗi cha và kết hợp lại các phần để tạo thành hai chuỗi con.



Tín dụng hình ảnh: [Geekforgeek](https://www.geekforgeek.org/)

- Crossover thường xuyên có những bước tiến lớn trong không gian trạng thái sớm trong quá trình tìm kiếm.

Bước tái hợp: Một ví dụ



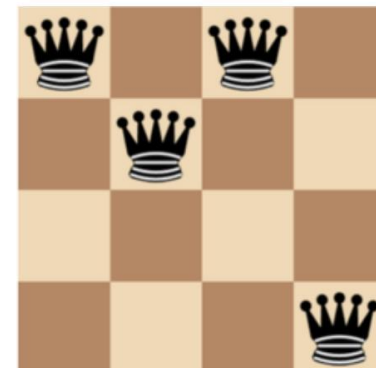
8 trạng thái hậu tương ứng với hai bố mẹ (trái và giữa) và con đầu lòng (phải). Các cột màu xanh lá cây bị mất trong bước chéo và các cột màu đỏ được giữ lại.

Thuật toán tiến hóa

- **Tỷ lệ đột biến:** xác định tần suất con cái có đột biến ngẫu nhiên đối với sự đại diện của chúng.
 - Mỗi bit trong thành phần của cá thể đều bị đảo lộn với xác suất bằng với tốc độ đột biến.
- **Thành phần của thế hệ tiếp theo:** chỉ bao gồm những đứa con mới được hình thành, hoặc một số cha mẹ đạt điểm cao nhất.
 - **Luyện tập tinh hoa:** đảm bảo thế lực tổng thể không bao giờ giảm tăng ca
 - **Thực hành loại bỏ:** Tất cả các cá thể dưới ngưỡng nhất định đều bị loại bỏ, có thể dẫn đến tăng tốc (Baum và cộng sự, 1995).

Câu đố 02: Tính điểm thể lực

- Xét bài toán 4 quân hậu, trong đó mỗi bang có 4 quân hậu, mỗi quân một trên bàn cờ. Trạng thái có thể được biểu diễn trong thuật toán di truyền dưới dạng một chuỗi gồm 4 chữ số, mỗi chữ số biểu thị vị trí của nữ hoàng trong cột riêng của nó (từ 1 đến 4).



- $()$ = số cặp quân hậu không tấn công

- Cho thể hệ hiện tại gồm 4 trạng thái:

$$S1 = 2341; S2 = 2132; S3 = 1232; S4 = 4321.$$

- Tính giá trị của $()$ đối với các trạng thái đã cho và xác suất để mỗi trạng thái đó sẽ được chọn trong bước "lựa chọn".

hàm GENETIC-ALGORITHM(population, Fitness) trả về một cá thể

lặp lại. lặp lại

trọng số ý TRỌNG LƯỢNG THEO(dân số, thể lực) dân số2

ý tập trống cho i = 1 đến

SIZE(dân số) làm

parent1, parent2 ý TRỌNG LƯỢNG-NGẪU NHIÊN-LỰA CHỌN(dân số, trọng số,2) con ý

TÁI SINH(parent1 , parent2) if (nhỏ ngẫu nhiên

xác suất) thì con ý BIẾN ĐỔI(con) thêm con vào quần thể 2

ý dân số2

Cho đến khi một số cá nhân đủ sức khỏe hoặc đủ thời gian trôi qua, hãy trả lại những gì tốt nhất

cá thể trong quần thể, theo mức độ thích nghi

hàm REPRODUCE(parent1, parent2) trả về một cá thể

n ý LENGTH(parent1) c ý

số ngẫu nhiên từ 1 đến n

trả về APPEND(SUBSTRING(parent1, 1, c), SUBSTRING(parent2, c+1, n)

Đánh giá các thuật toán di truyền

- Crossover mang lại khả năng khám phá ngẫu nhiên tốt hơn tìm kiếm cục bộ.
- Dựa vào rất ít kiến thức về miền
 - Số lượng lớn các tham số “có thể điều chỉnh”
 - Khó tái tạo hiệu suất từ vấn đề này sang vấn đề khác
- Thiếu các nghiên cứu thực nghiệm tốt so với các phương pháp đơn giản hơn
 - Hữu ích đối với một số nhóm vấn đề (nhỏ?), nhưng không có bằng chứng thuyết phục nào cho thấy GA tốt hơn việc leo đồi và khởi động lại ngẫu nhiên nói chung.

