Chapter 5.
Intelligent systems

Metaheuristic algorithms

References

- Xin-She Yang (2010). Engineering optimization. WILEY, pp.21-137.
- Xin-She Yang (2010). Nature-inspired metaheuristic algorithms. LUNIVER Press, pp.53–62.
- S.N. Sivanandam, S.N. Deepa (2008). Introduction to genetic algorithms. Springer, pp.15–82.
- Wassim Jaziri, Local Search Techniques: Focus on Tabuss search

HEURISTIC & METAHEURISTIC ALGORITHM

Thuật toán heuristic

- Thuật toán heuristic chỉ những kinh nghiệm riêng biệt để tìm kiếm lời giải cho một bài toán tối ưu cụ thể.
- Thuật toán heuristic thường tìm được lời giải có thể chấp nhận được trong thời gian cho phép nhưng **không chắc** đó là lời giải tối ưu.
- Cấu trúc của một thuật toán heuristic thường đơn giản và không khó để đề xuất.

Ví dụ 1

- Đề xuất một số thuật toán heuristic giải bài toán hành trình người bán hàng.
- SV tham khảo các thuật toán GTS1, GTS2

Thuật toán metaheuristic

- Thuật toán metaheuristic sử dụng nhiều heuristic kết hợp với các kỹ thuật phụ trợ nhằm khai phá không gian tìm kiếm;
- Thuật ngữ "*metaheuristic*" thường đề cập đến các thuật toán dạng **cá thể** như local search, simulated annealing, tabu search,... hoặc dạng quần thể như genetic algorithm, particle sarm optimization (PSO), ant, bee, bat,...

Các yếu tố chung của một thuật toán metaheuristic

- Cách thức tạo lời giải ban đầu,
- Điều kiện dừng,
- Cách thức mã hóa và giải mã lời giải,
- Chiến lược tăng cường hóa,
- Chiến lược đa dạng hóa,...

Sơ đồ thuật toán metaheuristic

- Khởi tạo lời giải ban đầu (lời giải có thể là một cá thể/hoặc một quần thể);
- Định nghĩa các chiến lược tăng cường hóa;
- Định nghĩa các chiến lược đa dạng hóa;
- Tính độ thích nghi cho mỗi lời giải;
- **while** (*điều kiện dừng chưa thỏa*{
- Thực hiện các chiến lược tăng cường hóa;
- Thực hiện các chiến lược đa dạng hóa;
- Cập nhật lời giải tốt nhất cho đến thời điểm hiện tại;
- }
- Trả về lời giải tốt nhất tìm được;

Khởi tạo lời giải ban đầu

- Khởi tạo ngẫu nhiên;
- Khởi tạo dựa vào một số heuristic riêng của bài toán;

Chiến lược tăng cường hóa lời giải

- Tăng cường hóa nhằm mục đích khai thác sâu hơn các vùng không gian lời giải tiềm năng để hy vọng tìm được lời giải có chất lượng tốt hơn,
- Tăng cường hóa thường được thực hiện nhờ áp dụng các tiếp cận tham lam (tìm kiếm được hướng đến phương án lân cận tốt nhất),
- Nếu tính tăng cường quá cao, nó có thể nảy sinh vấn đề hội tụ sớm; các lời giải tối ưu cục bộ tìm được có độ lệch cao hoặc vô nghĩa. Ngược lại, nếu tính tăng cường quá thấp, quá trình hội tụ sẽ chậm,
- Tính tăng cường hóa cho lời giải thường được thực hiện dựa vào các chiến lược tìm kiếm lân cận,
- Tùy theo đặc tính của bài toán mà có thể áp dụng các cách thức tìm kiếm lân cận khác nhau,
- Chiến lược tăng cường hóa được sử dụng xuyên suốt trong quá trình mà thuật toán diễn ra.

Chiến lược đa dạng hóa lời giải

- Đa dạng hóa nhằm khai phá những vùng không gian tìm kiếm mới nhằm tránh việc tìm kiếm rơi vào bẫy tối ưu cục bộ,
- Đa dạng hóa thường được thực hiện qua việc sử dụng tính ngẫu nhiên và có thể kết hợp với một số heuristic riêng biệt cho từng bài toán,
- Nếu tính đa dạng quá cao thì sẽ có nhiều vùng không gian lời giải được khai phá ngẫu nhiên, theo đó sẽ làm chậm quá trình hội tụ của thuật toán,

Chiến lược đa dạng hóa lời giải (….)

- Nếu tính đa dạng quá thấp thì không gian lời giải được khai phá sẽ bị giới hạn, các lời giải tìm được có khuynh hướng hội tụ cục bộ hoặc vô nghĩa,
- Một lời giải khi nó được cho thực hiện liên tiếp việc tìm kiếm lân cận theo một cách thức nào đó, thì sau một số lần lặp, chất lượng của lời giải đó sẽ không còn cải thiện được nữa (lời giải được gọi là đã khai thác cạn); nếu tiếp tục tìm kiếm lân cận theo hướng đó thì sẽ bế tắc,
- Chiến lược đa dạng hóa thường được sử dụng khi chất lượng của lời giải hiện tại không được cải thiện sau một số lần lặp xác định.

Điều kiện dừng

- Lời giải tốt nhất tìm được của bài toán không được cải thiện sau một số lần lặp định trước,
- Số lần lặp của thuật toán đạt đến một giá trị định trước,
- Thuật toán chạy hết một lượng thời gian định trước,...

Tiêu chí đánh giá chất lượng các thuật toán metaheuristic

- Các thuật toán metaheuristic thường được đánh giá dựa trên chất lượng lời giải (độ tốt của lời giải) và thời gian tính tương ứng để có lời giải đó thông qua việc thực nghiệm trên các bộ dữ liệu chuẩn (benchmarks) của bài toán,
- Tiêu chí đánh giá chất lượng lời giải được ghi nhận qua giá trị tốt nhất (và có thể cả giá trị trung bình và giá trị độ lệch chuẩn) sau một số lần chạy đối với mỗi bộ dữ liệu,
- Việc so sánh thời gian tính các thuật toán trên các máy tính có cấu hình khác nhau chỉ mang tính tương đối; tuy nhiên các bảng so sánh thời gian này cho ta một cái nhìn tham khảo về thời gian tính của các thuật toán.

LOCAL SEARCH ALGORITHM

Ý tưởng chung

- Từ một giải pháp hiện tại được chọn là *s*, thuật toán local search - LS sẽ tìm một giải pháp lân cận *s'* của *s*;
- Nếu *s'* tốt hơn *s* thì *s'* sẽ trở thành giải pháp hiện tại,
- Quá trình này sẽ dừng khi thuật toán lặp đến một số lần định trước hoặc khi giải pháp tốt nhất không được cải thiện qua một số lần lặp xác định,
- LS là nền tảng của nhiều thuật toán metaheuristic khác.

Có thể viết lại như sau:

-Gọi *S* là toàn bộ (hoặc một phần) của không gian lời giải của bài toán

-Khởi tạo ngẫu nhiên giải pháp *s* thuộc *S*;

while (**điều kiện dừng chưa được thỏa**)

{

+Tìm một giải pháp lân cận *s'* của *s* với s' thuộc *S*;

+Nếu *s'* tốt hơn *s* thì *s'* sẽ trở thành giải pháp hiện tại, tức đặt s=s';

+Thỏa dieu kien de thuc hien tim kiem ngau nhien

<cho thay doi ngau nhien 1 thanh phan cua lời giải>

}

-*s* là lời giải tìm được của bài toán;

Trong đó điều kiện dừng có thể sử dụng là:

-Khi thuật toán lặp đến một số lần định trước hoặc

-Khi giải pháp tốt nhất không được cải thiện qua một số lần lặp xác định.

Ví dụ 2: đề xuất thuật toán LS giải bài toán TSP

- Khởi tạo cá thể ban đầu,
- Đánh giá độ thích nghi,
- Thực hiện tìm kiếm lân cận (đề xuất một số chiến lược tìm kiếm lân cận),
- Điều kiện dừng,...

Tìm kiếm lân cận/tìm kiếm địa phuong cho TSP

- Ví dụ bài toán TSP có 9 đỉnh, ta biểu diễn một cá thể lời giải:
- 5 1 7 8 9 4 6 2 3.
- Chiến lược 1: ***đảo*** (chọn 2 điểm, và chuỗi con giữa hai điểm này bị đảo)
- Chiến lược 2: ***chèn (***chọn 1 thành phố và chèn nó vào một vị trí ngẫu nhiên)
- Chiến lược 3: ***dời chỗ*** **(**chọn một hành trình con rồi chèn nó vào một vị trí ngẫu nhiên)
- Chiến lược 4: ***trao đổi qua lại*** (hoán vị 2 thành phố)

MỘT CẢI TIẾN CỦA LOCAL SEARCH: variable neighborhood search algorithm (VNS)

- Ý tưởng của thuật toán lân cận biến đổi (Variable neighborhood search-VNS) là thực hiện lần lượt từng thuật toán lân cận, hết thuật toán lân cận này đến thuật toán lân cận khác,
- Trong quá trình thực hiện thuật toán VNS, ta luôn ghi nhận lời giải tốt nhất (kỷ lục),
- Khi thực hiện một thuật toán lân cận, nếu tìm được kỉ lục mới thì ta quay trở lại thực hiện thuật toán VNS từ đầu. Ngược lại, ta chuyển sang thuật toán lân cận tiếp theo,
- Quá trình được tiếp tục cho đến khi thực hiện hết tất cả các thuật toán lân cận mà lời giải tốt nhất không được cải thiện.

Có thể viết lại như sau:

-Gọi *S* là toàn bộ (hoặc một phần) của không gian lời giải của bài toán

-Khởi tạo ngẫu nhiên giải pháp *s* thuộc *S*;

-Gọi LS_1, LS_2,…,LS_k là các thuật toán tìm lân cận của bài toán đang xét;

while (điều kiện dừng chưa được thỏa)

{

for (int i=1;i<=k;i++)

{

+Gọi s’ là lận cận của s với s’ được tìm thấy bằng LS_i;

+Nếu s’ tốt hơn s thì break; // thoát khỏi vòng lặp for;

}

}

-*s* là lời giải tìm được của bài toán;

Trong đó điều kiện dừng có thể sử dụng là: Khi thực hiện hết tất cả các thuật toán lân cận mà lời giải tốt nhất không được cải thiện.

GENETIC ALGORITHM

Giới thiệu

- Thuật toán di truyền (Genetic Algorithm - GA) thuộc lớp các phương pháp tìm kiếm thông dụng có sự kết hợp với các yếu tố của kỹ thuật tìm kiếm ngẫu nhiên (stochastic) có định hướng giúp tạo nên sự cân bằng tương đối giữa tìm kiếm và khai phá trong không gian lời giải.
- Một thuật toán di truyền nhìn chung gồm các thành phần cơ bản sau: Biểu diễn di truyền cho các cá thể của bài toán, cách để tạo quần thể, một hàm lượng giá để đánh giá các lời giải theo độ thích nghi, các toán tử di truyền để thay đổi các thành phần thuộc gen của cá thể con (phép lai, đột biến, chọn lọc, …), các giá trị tham số mà thuật toán di truyền sử dụng (kích thước quần thể, xác suất thực hiện các toán tử di truyền lên quần thể, …).

- Thuật toán di truyền được mô phỏng dựa theo hoạt động di truyền của sinh vật trong tự nhiên; đó là lai ghép (crossover), đột biến (mutation) và chọn lọc (selection).
- Khi sử dụng thuật toán di truyền để giải quyết bài toán tối ưu thì có nhiều khía cạnh mấu chốt cần phải giải quyết:
  - ✓ Thứ nhất là cách biểu diễn di truyền của một lời giải hay nói cách khác là tìm một cấu trúc dữ liệu phù hợp biểu diễn cho các cá thể (từ đây trở về sau khái niệm lời giải/cá thể/gen được hiểu là như nhau),
  - ✓ Thứ hai là cách thức khởi tạo quần thể ban đầu,
  - ✓ Thứ ba là xác định một hàm thích nghi đóng vai trò môi trường (tính thích nghi),
  - ✓ Thứ tư là định nghĩa các phép toán di truyền phù hợp,
  - ✓ Thứ năm là xác định rõ các tham số di truyền chẳng hạn như kích thước của quần thể, xác suất áp dụng cho từng phép toán di truyền, số thế hệ cần lặp lại,…

Khởi tạo quần thể

- Nhìn chung, có hai cách để tạo quần thể bao đầu:
  - ✓ khởi tạo heuristic
  - ✓ khởi tạo ngẫu nhiên trong khi còn thỏa các điều kiện biên hoặc các ràng buộc của bài toán.
- Mặc dù độ thích nghi trung bình của các cá thể khởi tạo theo phương pháp heuristic là tương đối cao nhằm giúp thuật toán di truyền tìm kiếm lời giải nhanh hơn, nhưng trong phần lớn các bài toán có kích thước lớn, ví dụ, những bài toán thiết kế mạng, cách tiếp cận theo hướng heuristic chỉ có thể khai phá được một lượng nhỏ trong không gian lời giải và khó có thể tìm ra lời giải tối ưu toàn cục vì thiếu đi tính đa dạng của quần thể.
- Thường thì ta phải thiết kế riêng một thủ tục mã hóa dựa trên các nhiễm sắc thể để khởi tạo quần thể ban đầu từ đó.

Lượng giá độ thích nghi

- Độ tốt của một cá thể trong quần thể chỉ là một trong số những cơ sở để xác định tính thích nghi của cá thể đó đối với môi trường.
- Rõ ràng một cá thể tốt nhất ở thế hệ hiện tại vẫn có khả năng bị kẹt lại trong các thế hệ sau và ngược lại, một cá thể chưa tốt ở thế hệ hiện tại vẫn có khả năng tiềm tàng dẫn đến cá thể tốt hơn ở thế hệ sau.
- Mặc dù vậy, thường thì cá thể tốt ở thế hệ hiện tại vẫn sẽ có xác suất dẫn đến cá thể tối ưu cao hơn những cá thể xấu hơn.
- Do đó, chúng ta vẫn lấy độ tốt của cá thể làm một yếu tố căn bản nhất để xác định tính thích nghi của cá thể đó.
- Thông thường, độ thích nghi của cá thể cũng chính là xác suất để cá thể đó được chọn lọc hoặc lai ghép hoặc đột biến khi tiến hành sinh ra thế hệ kế tiếp
- Sau đây là hai phương pháp thường được sử dụng để xác định tính thích nghi của cá thể.

Độ thích nghi tiêu chuẩn

• Hàm mục tiêu của bài toán thường là hàm dùng để đánh giá mức độ tốt của một lời giải (hàm mục tiêu được sử dụng để lượng giá độ tốt của các cá thể và có thể xem như là hàm lượng giá).

Gọi Popsize là kích thước của quần thể. Mỗi cá thể i được ký hiệu là t_i. Độ thích nghi $f(t_i)$ của mỗi cá thể t_i (i=1..Popsize), ở đây $f(t_i)$ là hàm mục tiêu; với i=1..n-1

Tính tổng giá trị thích nghi của cả quần thể: $F = \sum_{i=1}^{Popsize} f(t_i)$

Tính xác suất chọn cho từng cá thể ti: $p_i = f(t_i)/F$

Tính vị trí xác suất q_i cho từng cá thể t_i: $q_i = \sum_{j=1}^{i} p_j$, (i = 1..Popsize)

Độ thích nghi xếp hạng

- Cách tính độ thích nghi tiêu chuẩn như trên chỉ thực sự hiệu quả đối với những quần thể mà các cá thể có độ tốt tương đối đồng đều.
- Nếu vì một lý do nào đó mà có một cá thể có độ tốt quá cao hoặc quá thấp, tách biệt hẳn các cá thể còn lại thì các cá thể của thế hệ sau sẽ bị "hút" về phía cá thể đặc biệt đó tạo nên hiện tượng di truyền cục bộ. Phương pháp xác định độ thích nghi xếp hạng sẽ loại bỏ hiện tượng di truyền cục bộ này; phương pháp này không làm việc trên giá trị độ lớn của hàm mục tiêu f mà chỉ làm việc dựa trên thứ tự của các cá thể trên quần thể sau khi đã sắp xếp các cá thể theo giá trị của hàm mục tiêu f.

Các phép toán di truyền

- Lai ghép (Crossover)
- Đột biến (Mutation)
- Chọn lọc (Selection)

Lai ghép (Crossover)

- Phép lai là quá trình hình thành nhiễm sắc thể mới trên cơ sở các nhiễm sắc thể cha-mẹ, bằng cách ghép một hay nhiều đoạn gen của hai (hay nhiều) nhiễm sắc thể cha-mẹ với nhau.
- Phép lai xảy ra với xác suất $p_{crossover}$.
- Có nhiều chiến lược lai ghép như one-point crossover (lai ghép một điểm), two-point crossover (lai ghép 2 điểm), uniform crossover (lai ghép đồng nhất).

- Phép lai là toán tử di truyền chính. Nó được thực hiện cùng lúc trên 2 nhiễm sắc thể và tạo ra cá thể con bằng cách tổ hợp lại các đặc điểm của cả 2 cá thể lai (cá thể cha).
- Cách đơn giản để thực hiện phép lai là chọn ra ngẫu nhiên một điểm cắt (cut-point) và tạo ra cá thể con bằng cách kết hợp phân đoạn bên trái điểm cắt của cá thể cha thứ 1 và phân đoạn bên phải điểm cắt của cá thể lai còn lại.
- Phương pháp này hoạt động tốt trên cách biểu diễn chuỗi các bit.
- Hiệu năng của GA lệ thuộc rất nhiều vào hiệu năng mà toán tử lai được sử dụng.

- Xác suất lai (ký hiệu là pC) được định nghĩa là tỉ lệ số lượng cá thể con được sinh ra ở mỗi thế hệ so với kích thước quần thể (popSize).
- Xác suất lai kiểm soát số lượng nhiễm sắc thể được thực hiện phép lai pC x popSize.
- Xác suất lai càng cao cho phép khai phá nhiều hơn trong không gian tìm kiếm, và giảm thiểu cơ hội hội tụ về lời giải tối ưu sai (false optimum); nhưng nếu xác suất lai quá cao, kết quả sẽ làm tốn nhiều thời gian tính toán trong việc khai phá những vùng không tìm năng trong không gian tìm kiếm.

Đột biến (Mutation)

- Đột biến là hiện tượng cá thể con mang một số tính trạng không có trong gen di truyền của cha-mẹ.
- Phép đột biến xảy ra với xác suất $p_{mutation}$, nhỏ hơn nhiều so với xác suất lai $p_{crossover}$.
- Đột biến là toán tử cơ bản giúp sinh ra những thay đổi cùng lúc trên nhiều nhiễm sắc thể khác nhau.
- Một cách đơn giản để tạo đột biến là thay đổi một hoặc nhiều gen.
- Trong GA, đột biến đóng vai trò then chốt trong quá trình thay thế các gen đã mất trong quá trình chọn lọc tự nhiên của quần thể nhằm giúp chúng có cơ hội được thử trong nhiều biến cố khác nhau hoặc cung cấp các gen chưa từng hiện diện trong quần thể ban đầu.

- Xác suất đột biến (ký hiệu là pM) được định nghĩa là % trong tổng số gen của quần thế tham gia đột biến.
- Xác suất đột biến kiểm soát xác suất mà các gen mới được tạo để thử trong quần thể.
- Nếu xác suất đột biến quá thấp, nhiều gen vốn hữu dụng sẽ không có cơ hội được thử, trong khi nếu xác suất đột quá cao, thì sẽ có gây ra nhiều thay đổi hỗn loạn trong quần thể, cá thể con sẽ không còn giống với cha mẹ chúng, và thuật toán mất đi khả năng lưu vết trong khi thực hiện tìm kiếm.

Chọn lọc (Selection)

Phép chọn lọc là quá trình loại bỏ các cá thể xấu trong quần thể để chỉ giữ lại trong quần thể các cá thể tốt.

Có nhiều chiến lược chọn lọc như: Ranking selection, Tournament selection.

+ Ranking Selection:Sắp xếp các cá thể theo giá trị giảm dần độ thích nghi tức là tăng dần hàm mục tiêu của lời giải và chỉ chọn những cá thể nào có độ thích nghi cao hơn.

+ Tournament Selection:Chọn ngẫu nhiên từng cặp cá thể, so sánh độ thích nghi, và chỉ chọn cá thể có độ thích nghi cao hơn.

- Chọn lọc giúp điều hướng trong GA. Nếu áp lực chọn lọc quá cao, thì sẽ làm chậm quá trình tìm kiếm.
- Thông thường, áp suất chọn lọc thấp sẽ giúp thuật toán mở rộng việc khai phá trong không gian lời giải, trong khi áp lực chọn lọc cao sẽ làm thu hẹp đi không gian tìm kiếm.
- Chọn lọc tự nhiên điều hướng thuật toán về những vùng tiềm năng trong không gian tìm kiếm.
- Đến nay nhiều phương pháp chọn lọc đã được đề xuất, thực nghiệm và so sánh.
- Có các phương pháp chọn lọc phổ biến mà điển hình là chọn lọc vòng quay Routlette (Roulette wheel).

Sơ đồ thuật toán di truyền

- Khởi tạo quần thể ban đầu với các lời giải ngẫu nhiên
- Đánh giá độ thích nghi của các thể của quần thể
- **while** (chưa thỏa điều kiện dừng)
- Chọn ngẫu nhiên một cặp cá thể cha-mẹ từ quần thể
- Lai ghép (Crossover: Nếu xảy ra xác suất lai ghép pcrossover thì lai ghép 2 cá thể cha mẹ để tạo thành các cá thể con; ngược lại,các cá thể con giống cá thể cha-mẹ.
- Đột biến (Mutation): Nếu xảy ra xác suất đột biến pmutation thì đột biến bằng cách thay đổi nhỏ trong cá thể con.
- Đánh giá lại độ thích nghi của các cá thể con.
- Chọn lọc (Selection): Chọn các cá thể có độ thích nghi cao hơn cho thế hệ sau.
- **end while**

VÍ DỤ 3: Đề xuất thuật toán GA giải bài toán TSP

Khởi tạo quần thể

- Mỗi cá thể được khởi tạo bằng một trong hai cách sau: Heuristic hoặc Ngẫu nhiên (mỗi hành trình của TSP xem là một cá thể; mỗi đỉnh của hành trình là gen của cá thể tương ứng).
- Khởi tạo một quần thể gồm N cá thể.

Đánh giá độ thích nghi

Tính chi phí của mỗi cá thể dựa vào hàm mục tiêu.

Phép đột biến

Như đã đề cập trong các thuật toán LS hoặc VNS

Phép lai

(nghiên cứu các phép lai PMX, OX, CX)

PMX (Partially Matched Crossover**):** Do Golberg va Lingle đề nghị;

Tạo ra con mới bằng cách chọn một chuỗi con của hành trình từ một cha-mẹ đồng thời bảo toàn thứ tự và vị trí của tối đa có thể các thành phố từ cha-mẹ còn lại.

Một chuỗi con của hành trình được chọn bằng cách chọn 2 điểm cắt ngẫu nhiên, được dùng làm giới hạn cho các thao tác hoán vị

p_1 = (1 2 3 | 4 5 6 7 | 8 9) và

p_2 = (4 5 2 | 1 8 7 6 | 9 3)

tạo ra con theo cách sau. Trước hết các đoạn giữa các điểm cắt được hoán vị ('x' được hiểu là 'hiện tại chưa biết'):

o_1 = (x x x | 1 8 7 6 | x x) và

o_2 =(x x x | 4 5 6 7 | x x)

Hoán vị này cũng định nghĩa một loạt các ánh xạ:

1↔4, 8↔5, 7↔6, và 6↔7.

Rồi ta có thể thêm vào các thành phố (từ các cha–mẹ gốc), mà không có xung đột:

o_1 = (x 2 3 | 1 8 7 6 | x 9) và

o_2 =(x x 2| 4 5 6 7 |9 3)

Cuối cùng, x đầu tiên trong con o_1 (lẽ ra phải là 1, nhưng không có xung đột) được thay bằng 4, vì ánh xạ (1↔4). Tương tự, x thứ hai trong con o_1 được thay bằng 5, còn x và x trong con o_2 là 1 và 8. Các con là:

o_1 = (4 2 3| 1 8 7 6| 5 9) và

o_2 =(1 8 2| 4 5 6 7 |9 3) .

Lai PMX khai thác các điểm tương đồng quan trọng trong giá trị và xếp bậc đồng thời khi được sử dụng với một kế hoạch sinh sản thích hợp.

(Order Crossover)

OX – do Davis đề nghị – tạo ra các con bằng cách chọn một chuỗi con của hành trình từ một cha–mẹ và bảo tồn thứ tự tương đối của các thành phố của cha–mẹ kia. Thí dụ, hai cha–mẹ (với hai điểm cắt được đánh dấu bởi '|')

p_1 = (1 2 3 | 4 5 6 7 | 8 9) và

p_2 = (4 5 2 | 1 8 7 6 | 9 3)

tạo ra các con như sau. Đầu tiên, các đoạn giữa các điểm cắt được sao chép vào các con:

o_1 = (x x x | 4 5 6 7 | x x) và

o_2 =(x x x | 1 8 7 6 | x x).

Kế đến, bắt đầu từ điểm cắt thứ hai của một cha–mẹ, các thành phố từ cha–mẹ kia được sao chép theo cùng thứ tự, bỏ đi các dấu hiệu đã có. Đến cuối chuỗi, ta tiếp tục từ vị trí đầu tiên của chuỗi. Thứ tự của các thành phố trong cha–mẹ thứ hai (từ điểm cắt thứ hai là:

9 – 3 – 4 – 5 – 2 – 1 – 8 – 7 – 6;

sau khi bỏ đi các thành phố 4, 5, 6 và 7 đã có trong con thứ nhất, ta có:

9–3–2–1–8.

Thứ tự này được đặt vào con thứ nhất (bắt đầu từ điểm cắt thứ hai):

o_1 = (2 1 8|4 5 6 7|9 3).

Tương tự ta có con kia là:

o_2 =(3 4 5|1 8 7 6|9 2).

Lai tạo OX khai thác thuộc tính về biểu diễn đường dẫn, mà thứ tự của các thành phố (chứ không phải là vị trí của chúng) rất quan trọng, nghĩa là, hai hành trình,:

9–3–4–5–2–1–8–7–6

4–5–2–1–8–7–6–9–3

thực ra là giống nhau.

(Cycle Crossover)

CX – do Oliver đề nghị – xây dựng con theo cách mỗi thành phố (và vị trí của nó) xuất phát từ một trong các cha–mẹ. Ta giải thích cơ chế của lai chu trình bằng thí dụ sau đây. Hai cha–mẹ:

p_1 = (1 2 3 4 5 6 7 8 9) và

p_2 = (4 1 2 8 7 6 9 3 5)

sẽ tạo ra con thứ nhất bằng cách lấy thành phố thứ nhất từ cha–mẹ thứ nhất:

o_1= (1 x x x x x x x x x)

Do mỗi thành phố trong con có thể bị lấy đi từ một trong các cha–mẹ của nó (từ cùng vị trí), nên giờ ta không có chọn lựa nào: thành phố kế tiếp được xét phải là thành phố 4, do thành phố từ cha–mẹ p_2 ngay "bên

dưới" thành phố 1 được chọn. Trong p_1 thành phố này ở tại vị trí '4', như vậy:

o_1= (1 x x 4 x x x x x)

Điều này, trở lại bao hàm thành phố 8, do thành phố từ cha–mẹ p_2 ngay "bên dưới" thành phố 4 được chọn. Như vậy:

o_1= (1 x x 4 x x x 8 x);

Theo luật này, các thành phố kế tiếp được gộp vào con thứ nhất là 3 và 2. Nhưng chú ý rằng việc chọn thành phố 2 đòi hỏi việc chọn thành phố 1, đã có trong danh sách – như vậy ta đã hoàn thành một chu trình:

o_1= (1 2 3 4 x x x 8 x).

Các thành phố còn lại được hoàn thành từ cha–mẹ kia:

o_1= (1 2 3 4 7 6 9 8 5).

Tương tự:

o_2 = (4 1 2 8 5 6 7 3 9).

CX bảo toàn vị trí tuyệt đối của các phần tử theo thứ tự của cha–mẹ.

Intelligent systems

Tài liệu tham khảo

[1].Hồ Cẩm Hà, "*Các hệ thống thông minh*",2012.

[2].Adrian A. Hopgood, "*Intelligent Systems for Engineers and Scientists: A Practical Guide to Artificial Intelligence*", 4th Edition, 2021, CRC Press.

[3].Yung C. Shin, Chengying Xu, "*Intelligent Systems: Modeling, Optimization, and Control*", 2017, CRC Press.

[4].Geoff Hulten, "*Building Intelligent Systems: A Guide to Machine Learning Engineering*", 2018, A.Press.

Topic 1:Giới thiệu về các hệ thống thông minh và các ứng dụng hiện đại

❖Câu hỏi thảo luận và bài tập:

- Định nghĩa trí tuệ nhân tạo, các lĩnh vực ứng dụng của trí tuệ nhân tạo.
- Định nghĩa hệ thống thông minh, phân loại các hệ thống thông minh.
- Nêu sự khác biệt cơ bản giữa các phần mềm tin học thông thường và các hệ cơ sở tri thức ?
- Anh (chị) đã từng dùng phần mềm thông minh nào ? Hãy chỉ ra những sản phẩm đó thông minh (câu 5, trang 16,ref1)
- Theo anh (chị) thế nào là một hệ thống (phần mềm) thông minh ? Cho một số ví dụ để làm rõ quan điểm của anh (chị) (câu 6, trang 16, ref1)
- Để có thể xây dựng được những hệ thống thông minh, chúng ta cần phải có những hiểu biết gì ? (câu 8, trang 16, ref1)

Topic 2: Biểu diễn tri thức

❖Câu hỏi thảo luận và bài tập:

- Một số phương pháp biểu diễn tri thức
- Theo anh (chị) một ngôn ngữ biểu diễn tri thức tốt cần phải có những khả năng gì ? (câu 1, trang 33, ref1)
- Bài tập 4, trang 34, ref1.

Topic 3: Các phương pháp lập luận

❖Câu hỏi thảo luận và bài tập:

- Định nghĩa suy diễn tiến (ref1)
- Định nghĩa suy diễn lùi (ref1)
- Bài tập 2,3,6 trang 60,61 (ref1)

Topic 4: Hệ chuyên gia

❖Câu hỏi thảo luận và bài tập:

- Trình bày khái niệm hệ chuyên gia (expert system).
- Trình bày cấu trúc cơ bản của một hệ chuyên gia.
- Câu 2,9,10 trang 92, ref1.

Topic 5: Hệ trợ giúp ra quyết định

❖Câu hỏi thảo luận và bài tập:

- Trình bày hệ hỗ trợ ra quyết định (decision support system).
- Nêu các đặc trưng và khả năng cơ bản của một hệ hỗ trợ ra quyết định.

Topic 6: Hệ tư vấn

❖Câu hỏi thảo luận và bài tập:

- Một số khái niệm liên quan

Topic 7: Quản trị dữ liệu: xây dựng kho dữ liệu, truy cập và hiển thị

❖Câu hỏi thảo luận và bài tập:

- Trình bày khái niệm kho dữ liệu (data warehouse).
- Trình bày khái niệm cơ sở dữ liệu thông minh.
- Trình bày khái niệm khai phá dữ liệu (data mining); nêu các chức năng của khai phá dữ liệu.
- Câu 2, Câu 6, trang 166, ref1

Topic 8: Các tác tử thông minh

❖Câu hỏi thảo luận và bài tập:

- Trình bày tác tử thông minh khái (intelligen agent).
- Trình bày cấu trúc cơ bản của một tác tử thông minh.
- Trình bày các đặc tính cơ bản của tác tử thông minh.

Topic 9: Xây dựng hệ thống thông minh

❖Câu hỏi thảo luận và bài tập:

- Một số khái niệm liên quan