3 thuật toán giải quyết bài toán Maximal Clique Problem

1. Thuật toán Bees
2. Giải thuật cơ bản được đề xuất như sau :

1: Khởi tạo quần thể ban đầu với n vùng; mỗi vùng chứa một cá thể ong;

2: Đánh giá độ thích nghi của quần thể;

3: **while** ( điều kiện đừng chưa thỏa)

4: Trong **n** vùng ban đầu, chọn ngẫu nhiên p vùng (**p<n**) để thực hiện việc tìm kiếm lân cận và trong **p** vùng này chọn tiếp ra **e** vùng có độ thích nghi cao nhất.

5: Tuyển thêm **nep** ong để thực hiện việc tìm kiếm lân cận cho mỗi vùng quanh nó trong **e** vùng được chọn;

6: Tuyển thêm **nsp** ong để thực hiện việc tìm kiếm lân cận cho mỗi vùng quanh nó trong **p – e** vùng (**nep>nsp**);

7: Mỗi ong trong **n – p** vùng còn lại sẽ được thay thế bằng một ong ngẫu nhiên;

8: Đánh giá lại độ thích nghi cho các ong trong từng vùng;

9: Mỗi vùng sẽ chọn ra duy nhất một ong có độ thích nghi cao nhất để xây dựng quần thể ong ở thế hệ tiếp theo;

10: **end while.** \* Giải thuật **Bees** bao gồm một tập các tham số:

**n**: số ong do thám (tương ứng số vùng);

**p**: số vùng được chọn trong n vùng được thăm; e: số vùng tốt nhất trong p vùng được chọn;

**nep**: số ong được cử đến mỗi vùng trong e vùng;

**nsp**: số ong được cử đến p – e vùng; np: số ong được cử đến **n – p** vùng còn lại.

1. Thuật toán Local Search
2. Thuật toán KLS gốc

Input : Đồ thị G(V,E)

Output : ω(G) và tập đỉnh clique

* Giai đoạn 1: Giai đoạn khởi tạo (l-put)
* Giai đoạn 2: Giai đoạn tìm kiếm lân cận(k-put)

1. Thuật toán KLS cải tiến

* Giai đoạn 1: Khởi tạo một clique ngẫu nhiên
* Giai đoạn 2: Tìm kiếm lân cận tốt hơn
* Giai đoạn 3: Khai thác hiệu quả lân cận tốt nhất

1. Nội dung cải tiến:

* Mở rộng không gian thêm đỉnh
* Khai thác hiệu quả các lân cận

1. Thuật toán Heuristic

* Ý tưởng chung của các heuristic giải bài toán clique lớn nhất là bắt đầu từ một clique rỗng; sau đó lặp lại việc thêm dần các đỉnh để tạo thành clique lớn hơn đến khi nào không thể thêm đỉnh để tạo thành các clique lớn hơn nữa thì dừng. Các thuật toán heuristic dạng tham lam này dựa vào thông tin quan trọng nhất chính là bậc của các đỉnh ứng viên. Điểm mấu chốt của các cải tiến của chúng tôi là dựa vào yếu tố ngẫu nhiên của các lời giải tiềm năng và đồng thời cho chạy nhiều lần một bộ dữ liệu thực nghiệm để hy vọng tìm được lời giải có chất lượng tốt hơn.

1. Thuật toán Heuristic 1(Nguyên bản)

1. **procedure** MAXCLIQUEHEU1 (G=(V,E)) // viết tắt là HEU1

2. **begin**

3. Đặt U = ∅;

4. **while** (V ≠ ∅) do

5. **begin**

6. Duyệt các đỉnh thuộc tập V; tìm đỉnh v ∉ U và v có bậc là lớn nhất;

7. U=U ∪ {v};

8. V=V \ {v};

9. Xóa tất cả các đỉnh u không kề với đỉnh v ra khỏi tập V;

10. Cập nhật bậc của các đỉnh liên quan đến các đỉnh u vừa bị xóa;

11. **end**;

12. Xuất kết quả U;

13. **end**;

1. Thuật toán Heuristic 1 cải tiến

1. **procedure** MAXCLIQUEHEU1\_improve(G = (V,E)) **begin** // viết tắt là HEU1\_improve

2. Bestsolution=0;

3. T=∅;

4. **while** (điều kiện dừng chưa thỏa) **do**

5. **begin**

6. Đặt U = ∅;

7. **while** (V≠ ∅) **do**

8. **begin**

9. Đặt tập Q={k đỉnh thuộc tập V có bậc lớn nhất};

10. Chọn ngẫu nhiên một đỉnh v ∈ Q;

11. U=U ∪{v};

12. V=V \ {v};

13. Xóa tất cả những đỉnh u không kề với đỉnh v ra khỏi tập V;

14. Cập nhật bậc của các đỉnh liên quan đến các đỉnh u vừa bị xóa;

15. **end**;

16. if |U|> Bestsolution **then**

17. **begin**

18. Bestsolution=|U|;

19. Cập nhật T=U;

20. **end**;

21. **end**;

22. Xuất kết quả Bestsolution và các đỉnh của clique T tương ứng có kích thước Bestsolution;

23. **end**;

1. Thuật toán Heuristic 2
2. Thuật toán Heuristic 2 cải tiến