1. Tree-Search

Là thuật toán cho phép mở lại các node đã mở trước đó, và kết quả sẽ là lặp vô hạn nếu cấu trúc frontier là LIFO áp dụng trên đồ thị vô hướng.

1. Graph-Search

Là thuật toán không cho phép mở laị các node đã mở trước đó.

* Cả 2 thuật toán trên chỉ dừng lại khi node lá được mở là nút đích

1. Tìm kiếm theo chiều rộng (BFS)

Là thuật toán có cấu trúc cho frontier là FIFO, bỏ qua tất cả các node đã mở

1. Tìm kiếm theo chiều sâu (DFS)

Là thuật toán có cấu trúc cho frontier là LIFO, bỏ qua tất cả các node đã mở

1. Tìm kiếm theo chiều sâu có giới hạn độ sâu

Là thuật toán tìm kiếm theo chiều sâu (DFS) tuy nhiên chỉ tìm kiếm ở độ sâu l nhất định mà không quan tâm đến các node vượt quá giới hạn đó

1. Tìm kiếm lặp sâu dần

Là thuật toán tìm kiếm theo chiều rộng nhưng có độ sâu của cây tăng dần từ 1.

* Các thuật toán trên chỉ dừng lại khi các node lá mở được node con là trạng thái đích

1. Tìm kiếm với giá cực tiểu UCS

Là thuật toán ưu tiên mở các node có giá thành g(n) nhỏ nhất trong đó g(n) là giá (khoảng cách) thực tế từ trạng thái ban đầu đến nút n.

1. Thuật toán tìm kiếm tham lam (Greedy best first search)

Là thuật toán ưu tiên mở các node có giá thành h(n) nhỏ nhất trong đó h(n) là giá (khoảng cách) ước lượng từ trạng thái ban đầu đến nút n.

1. Thuật toán tìm kiếm A\*

Là thuật toán ưu tiên mở các node có giá thành f(n) nhỏ nhất trong đó f(n) = g(n) + h(n) trong đó f(n) được coi là đánh giá chi phí từ nút ban đầu đến nút n.

* Thuật toán trên chỉ dừng lại khi node lá được mở là nút đích

1. Thuật toán leo đồi him climbing

Là thuật toán có 1 vòng lặp từ trạng thái ban đầu di chuyển về phía có giá trị tăng lên và dừng lại nếu không có lân cận nào có trạng thái tốt hơn gọi là đỉnh.

1. Thuật toán mô phỏng luyện kim

Là thuật toán leo đồi tuy nhiên nó cho phép di chuyển về hướng có giá trị tồi hơn với xác xuất sẽ giảm dần sau mỗi lần quay ngược

1. Thuật toán tìm kiếm Beam

Là thuật toán tại mỗi bước sẽ sinh ra k trạng thái ngẫu nhiên của mỗi trạng thái sau đó sinh ra tất cả các trạng thái của k trạng thái và chọn k trạng thái tốt nhất.

1. Thuật toán di truyền