**Chương 5: Quy hoạch động và giải thuật tham lam**

* **Quy hoạch động**
  + Quy hoạch động giải các bài toán tối ưu hóa bằng cách kết hợp các lời giải của các bài toán con của bài toán đang xét, phương pháp này khả dụng khi khi các bài toán con không độc lập với nhau, quy hoạch động giải các bài toán “cháu” dùng chung một lần rồi lưu lời giải trong 1 bảng và sau đó không phải tính lại. Quy hoạch động thường dùng giải các bài toán tối ưu hóa
  + 4 bước quy hoạch động: (b1) đặc trưng hóa cấu trúc của lời giải tối ưu. (b2) định nghĩa giá trị của lời giải tối ưu một cách đệ quy. (b3) tính trị của lời giải tối ưu theo kiểu từ dưới lên. (b4) cấu tạo lời giải tối ưu từ những thông tin đã được tính toán và lưu trữ
  + 2 thành phần của quy hoạch động: tính chất tiểu cấu trúc tối ưu (optimal substructure) và các bài toán con trùng lắp (overlapping subproblems)
  + Các bài toán liên quan: (1) Nhân chuỗi ma trận (tính chi phí tối ưu MATRIX-CHAIN-ORDER có độ phức tạp là O(n3) với n là số lượng ma trận, giải thuậtt MATRIX-CHAIN-MULTIPLY) tính tích ma trận từ kết quả ngoặc tối ưu từ MATRIX-CHAIN-ORDER; (2) Chuỗi con chung dài nhất, với đầu vào là 2 chuỗi có độ dài lần lượt là m,n (tìm lời giải LCS-LENGTH thì độ phức tạp là O(mn), in ra kết quả PRINT-LCS có độ phức tạp là O(m+n)); (3) Bài toán cái túi (knapsack problem), với cái túi có sức chứa tối đa là M và N mặt hàng thì độ phức tạp của giải thuật là O(MN); (4) Giải thuật Warshall tính bao đóng truyền của 1 đồ thị được biểu diễn bằng ma trận kế cận (độ phức tạp O(V3), với đồ thị có V đỉnh); (5) Giải thuật Floyd tìm lối đi ngắn nhất cho tất cả các cặp đỉnh dựa trên đồ thì có trọng số (có hướng hoặc không), độ phức tạp là O(V3)
* **Giải thuật tham lam**
  + Các giải thuật tối ưu hóa thường đi qua một số bước với tập các khả năng lựa chọn tại mỗi bước, một giải thuật tham lam thường chọn 1 khả năng được xem là tốt nhất tại lúc đó - tức là chọn 1 khả năng tối ưu cục bộ với hy vọng dẫn đến 1 lời giải tối ưu toàn cục
  + 2 thành phần chính của giải thuật tham lam: 1) Tính chất lựa chọn tham lam (greedy choice property): lựa chọn được thực hiện tùy thuộc vào những lựa chọn đã làm cho đến bây giờ nhưng nó không phụ thuộc vào các lựa chọn ở tương lai hay những lời giải của bài toán con, như vậy tiến hành từ trên xuống. 2) Tiểu cấu trúc tối ưu
  + Các bài toán liên quan: (1) Bài toán xếp lịch các hoạt động GREEDY-ACTIVITY-SELECTOR; (2) Bài toán cái túi dạng phân số GREEDY-KNAPSACK có độ phức tạp là O(n) với n là số mặt hàng; (3) Bài toán nén file dùng mã Huffman (độ phức tạp là O(nlgn) trên tập n ký tự); (4) Cây bao trùm tối thiểu – giải thuật Prim (đồ thị biểu diễn bằng tập ds kế cận), với Q là hàng đợi bằng cấu trúc Heap từ dưới lên phải là min-heap) là O(ElgV), với Q là một mảng là O(V2); (5) Giải thuật Dijkstra tìm lối đi ngắn nhất từ 1 đỉnh đến các đỉnh còn lại (biễu diễn bằng tập ds kế cận), độ phức tạp khi biểu diễn hàng đợi Q là min-heap O(V+E)lgV, khi biểu diễn hàng đợi Q là mảng O(V2+E); (6) Bài toán tô màu đồ thị SAME-COLOR, độ phức tạp là O(n2) với n là số đỉnh của đồ thị
* **So sánh giải thuật tham lam với quy hoạch động:** 
  + Giải thuật quy hoạch động đem lại lời giải tối ưu, còn giải thuật tham lam không nhất thiết đem lại lời giải tối ưu
  + Quy hoạch động tiếp cận từ dưới lên (các bài toán con). Giải thuật tham lam tiếp cận từ trên xuống
  + Ở bài toán cái túi ta nên dùng giải thuật tham lam cho dạng phân số, dùng quy hoạch động cho dạng 0-1

**Chương 6: Giải thuật quay lui**

* **Giải thuật quay lui**
  + Thiết kế giải thuật tìm lời giải cho 1 bài toán không phải là bám theo 1 tập quy luật tính toán đã xác định mà là bằng cách *thử và sửa sai (trial and error)*
  + Quay lui (backtracking) là kiểu giải quyết vấn đề có đặc điểm: bước hướng về lời giải đầy đủ và ghi lại thông tin về bước này mà sau đó nó có thể bị tháo gỡ và xóa đi khi phát hiện rằng bước này dẫn đến tình thế bế tắc*.* Thời gian tính toán của giải thuật quay lui là O(an) với a là số nút con mà 1 nút trên cây tìm kiếm có được và n là chiều dài của lối đi trên cây
  + Các thuật ngữ nhận biết: Thử và sửa sai; Chọn bước đi tới; Bế tắc “dead-end”; Quay lui; Đệ quy; Cây không gian trạng thái
  + Các bài toán liên quan: (1) Bài toán đường đi của quân mã; (2) Bài toán 8 quân hậu
  + Sự mở rộng của giải thuật quay lui đệ quy là tìm không chỉ 1 lời giải mà tất cả những lời giải của bài toán đã cho (phương pháp: một khi lời giải được tìm thấy và ghi lại, ta tiếp tục xét các ứng viên kế tiếp trong quá trình chọn ứng viên 1 cách có hệ thống), khuôn mẫu tổng quát cho sự mở rộng này còn gọi là giải thuật vét cạn
* **Cây không gian trạng thái :**
  + Được dùng để diễn tả quá trình làm việc của giải thuật quay lui, ghi nhận những lựa chọn đã được thực hiện.
  + Nút rễ diễn tả trạng thái đầu tiên trước khi quá trình tìm lời giải bắt đầu.
* **Giải thuật nhánh và cận**
  + Các thuật ngữ nhận biết: Chi phí của lời giải tốt nhất cho đến bây giờ; Cận của lời giải chưa đầy đủ; Tỉa nhánh; Giải thuật quay lui vét cạn
  + Các bài toán liên quan: (1) Bài toán thương gia du hành (Traveling Salesman Problem); (2) Bài toán sinh ra các hoán vị; (3) Bài toán cái túi dạng 0-1
* **So sánh giải thuật quay lui và nhánh cận**
  + Ý tưởng chính của giải thuật quy lui là tỉa ngay nhánh trong cây không gian trạng thái ngay khi ta nhận thấy rằng nó không thể dẫn đến lời giải. Ý tưởng nhánh-cận tăng cường hơn khi ta thực hiện với bài toán tối ưu hóa
  + Giải thuật nhánh-cận đòi hỏi thêm:1) một cách thức để xác định cận của mỗi nút trên cây không gian trạng thái trong 1 lời giải bất kỳ đạt được khi mở rộng thêm lời giải chưa đầy đủ. 2) Trị hàm mục tiêu của lời giải tốt nhất cho đến thời điểm đang xét

**Chương 7: Vấn đề NP – đầy đủ**

* **Lớp bài toán P và lớp bài toán NP**
  + Chúng ta nói 1 giải thuật giải 1 bài toán trong thời gian đa thức nếu độ phức tạp trong trường hợp xấu nhất thuộc về lớp O(p(n)) với p(n) là đa thức của kích thước dữ liệu nhập n. Tập hợp các bài toàn có thể giải trong thời gian đa thức gọi là lớp P (polynomial), tuy nhiên một cách định nghĩa hình thức hơn thì chỉ bao gồm trong lớp P là những bài toán quyết định (bài toán trả lời có hoặc không).
  + Bài toán dừng: Có 1 số bài toán quyết định không thể giải được bằng bất cứ giải thuật nào (bài toán bất khả quyết) thì gọi là bài toán dừng
  + Tính không tất định:Cho đến giờ các giải thuật ta dùng là loại giải thuật có tính tất định (deterministic), tất định là khi giải thuật đang làm gì thì cũng có 1 việc duy nhất sẽ phải làm kế tiếp. Một cách để mở rộng quyền năng của máy tính là cho nó có năng lực làm việc không tất định (non-deterministic). Không tất định là khi một giải thuật có quyền năng “tiên đoán” để chọn 1 khả năng thích đáng trong nhiều sự lựa chọn mà nó gặp.
  + Bài toán thỏa mãn công thức logic mệnh đề (CNF satisfiability problem) trả lời câu hỏi có tồn tại hay không 1 phép gán chân trị và các biến logic sao cho toàn công thức trở thành true, chúng ta có thể xây dựng 1 giải thuật không tất định để giải bài toàn này. Bài toán này là bài toán NP
  + Một giải thuật không tất định là 1 thủ tục gồm 2 chặng: 1) Chặng dự đoán (không tất định). 2) Chặng kiểm chứng (tất định)
  + Sau khi có khái niệm giải thuật không tất định, chúng ta có thể định nghĩa lớp bài toán NP
  + Lớp NP là tập các bài toán quyết định có thể giải bởi giải thuật không tất định trong thời gian đa thức, lớp bài toán này còn được gọi là lớp bài toán đa thức không tất định (non-deterministic polynomial)
* **Vấn đề NP đầy đủ**
  + Có 1 ds các bài toán mà ta đã biết là thuộc về lớp NP nhưng không rõ là thuộc lớp P hay không (tức là giải dễ dàng trên máy tính không tất định nhưng chưa ai có thể tìm ra được 1 giải thuật hữu hiệu trên máy tính thông thường). Nếu bất kỳ 1 bài toán nào trong những bài toán này có thể giải trong thời gian đa thức thì những bài toán thuộc lớp NP cũng sẽ giải được trong thời gian đa thức trên máy tất định. Và những bài toán này gọi là những bài toán NP-đầy đủ (NP-complete)
  + Lớp NP đầy đủ là lớp con của những bài toán khó nhất trong lớp NP. Công cụ chính để chứng minh 1 bài toán thuộc lớp NP-đầy đủ là ý tưởng về tính khả thu giảm đa thức (polynomial reducibility).
  + Bất kỳ giải thuật nào giải được bài toán mới thuộc lớp NP có thể dùng để giải 1 bài toán NP-đầy đủ nào đó bằng cách: biến thể 1 thể hiện bất kỳ của bài toán NP-đầy đủ đã biến thành 1 thể hiện của bài toán mới, giải bài toán này bằng giải thuật đã có để tìm ra 1 lời giải, rồi biến lời giải này trở về thành 1 lời giải của bài toán NP-đầy đủ đã biết. Do đó để chứng mình 1 bài toán mới thuộc lớp NP là NP đầy đủ, ta chỉ cần chứng minh rằng 1 bài toán NP-đầy đủ nào đó đã biết thì khả thu giảm đa thức về bài toán mới ấy.
  + Để chứng minh 1 bài toán mới L là NP-đầy đủ, ta cần chứng minh: 1) Bài toán L thuộc lớp NP. 2) Có 1 bài toán NP-đầy đủ nào đó đã biết thu giảm về L.
* **Định lý Cook:** nếu tồn tại 1 giải thuật thời gian đa thức để giải bài toán thỏa mãn biểu thức logic thì tất cả mọi bài toán trong lớp NP có thể được giải trong thời gian đa thức
* **Một số bài toán NP đầy đủ:** (1) Bài toán tìm đồ thị con đầy đủ lớn nhất trong 1 đồ thị; (2) Bài toán phân hoạch số; (3) Bài toán cái túi; (4) Bài toán quy hoạch nguyên; (5) Xếp lịch công việc trên đa bộ xử lý; (6) Bài toán phủ định; (7) Bài toán xếp thùng
* **Một số kỹ thuật đối phó với những bài toán NP-đầy đủ:**
  + - Dùng giải thuật xấp xỉ để tìm lời giải xấp xỉ tối ưu
    - Dựa vào hiệu quả của trường hợp trung bình để phát triển 1 giải thuật mà ta có thể dựa vào đó để tìm ra lời giải trong 1 số trường hợp
    - Nếu kích thước đầu vào nhỏ ta có thể dùng các giải thuật có độ phức tạp hàm mũ thì vẫn hữu hiệu, vd: giải thuật quay lui, nhánh và cận
    - Đưa ra các Heuristic vào giải thuật để tăng tính hữu hiệu
    - Sử dụng các metaheuristic chú giải thêm cho thuật ngữ này
* **Heuristic và Metaheuristic:**
  + Heuristic là tri thức về bài toán cụ thể được sử dụng để dẫn dắt quá trình tìm lời giải của giải thuật, nhờ sự thêm vào heuristic mà giải thuật trở nên hữu hiệu hơn
  + Metaheuristic là loại heuristic tổng quát có thể áp dụng cho nhiều lớp bài toán khác nhau
* **Đóng góp của vấn đề NP-đầy đủ**
  + - Có nhiều bài toán NP-đầy đủ trong nhiều lĩnh vực: giải tích số, sắp thứ tự và tìm kiếm, xử lý dòng ký tự, xử lý đồ thị, xử lý hình ảnh, khai phá dữ liệu
    - Nó cung cấp 1 cơ chế để xác định 1 bài toán mới trong các lĩnh vực trên là “dễ” hay “khó”
    - Có 4 lớp bài toán phân theo độ khó (giảm dần): Bài toán bất khả quyết (chưa hề có lời giải) -> Bài toán khó giải (chỉ tồn tại giải thuật thời gian hàm mũ để giải chúng) -> Bài toán NP-đầy đủ (là 1 lớp con đặc biệt của lớp bài toán NP) -> Bài toán P (tồn tại giải thuật có độ phức tạp đa thức để giải)

**Chương 8: Giải thuật xấp xỉ**

* **Sự cần thiết của giải thuật xấp xỉ**
  + Nếu bài toán thuộc lớp NP-đầy đủ thì chúng ta không thể tìm ra 1 giải thuật hữu hiệu có độ phức tạp đa thức để tìm ra lời giải tối ưu 1 cách chính xác. Tuy vậy vẫn có thể tìm ra 1 lời giải cận tối ưu (near optimal solution) – tức xấp xỉ tối ưu trong thời gian đa thức
  + Trong thực tế, tính chất cận tối ưu của lời giải là đủ tốt để dùng trong thực tế, như vậy có sự đánh đổi giữa tính tối ưu và tính dễ giải quyết.
  + Một giải thuật để tìm ra được lời giải cận tối ưu cho 1 bài toán tối ưu hóa được gọi là giải thuật xấp xỉ (approximation algorithm)
* **Các cận về độ hữu hiệu:**
  + Tùy bài toán, lời giải tối ưu có thể được định nghĩa như 1 lời giải có trị hàm chi phí là cực đại hay cực tiểu
* **Các bài toán liên quan:**
  + Bài toán phủ đỉnh (vertex cover): giải thuật APPROX-VERTEX-COVER có cận tỉ số là 2, tức là kích thước của tập phủ đỉnh trả về tối đa lớn gấp 2 kích thước của tập phủ đỉnh tối ưu
  + Bài toán phủ tập (set covering problem): giải thuật GREEDY-SET-COVER có cận tỉ số là H(max{|S|: với S thuộc F})
  + Bài toán người thương gia du hành. Giải thuật APPROX-TSP-TOUR xấp xỉ với cận tỉ số là 2 cho bài toán TSP thỏa bất đẳng thức tam giác, các bước thực hiện:
    - Tập hợp các thành phố có trên vị trí bản đồ
    - Lập cây bao trùm tối thiểu T
    - Sự duyệt cây theo tiền thứ tự
    - Lộ trình các thành phố được viếng
    - Lộ trình tối ưu
  + Bài toán xếp lịch các công tác độc lập
  + Bài toán xếp thùng
  + Bài toán cái túi