9.2. Các vấn đề minh họa

9.2.1 Bài toán The Job-Shop Scheduling

Bài toán The Job-Shop Scheduling là một trong những vấn đề được nghiên cứu nhiều nhất trong tối ưu hóa tổ hợp và một số lượng lớn các bài báo và sách đề xuất các thủ tục để giải, bao gồm một số triển khai tìm kiếm tabu, bao gồm một số cách triển khai tabu seach. Mặc dù một số lượng lớn các biến thể được tìm thấy trong tài liệu (và thậm chí nhiều hơn trong thế giới thực). Bài toán cổ điển được phát biểu như sau. Đầu tiên chúng ta giả định rằng **n** công việc phải được lên lịch trên **m** máy. Mỗi công việc tương ứng với một chuỗi **m** hoạt động cố định, một mỗi máy, trong đó mỗi hoạt động phải được xử lý trên một máy cụ thể trong một khoảng thời gian nhất định. Lưu ý rằng thứ tự xử lý trên các máy không nhất thiết phải giống nhau từ công việc này sang công việc khác. Mỗi máy có thể xử lý nhiều nhất một hoạt động tại một thời điểm và sau khi bắt đầu, một hoạt động phải được hoàn thành mà không bị gián đoạn. Mục đích là chỉ định các hoạt động cho các khe thời gian (time slot) trên máy để giảm thiểu thời gian hoàn thành công việc một cách tối đa, còn được gọi là the makespan. Một giải pháp cho vấn đề này có thể coi là một tập **m** hoán vị của **n** công việc, mỗi công việc cho mỗi máy, liên kết với lịch trình máy liên quan.

9.2.2. Bài toán The Capacitated Plant Location

Bài toán The Capacitated Plant Location là một trong những vấn đề cơ bản trong lý thuyết về vị trí. Gặp nhiều trong việc cài đặt các ứng dụng liên quan đến việc định vị các cơ sở với khả năng cung cấp dịch vụ giới hạn. Vấn đề có thể được mô tả như sau. Tập khách hàng ***I*** với các nhu cầu **Di**, **i ∈ I,** một số sản phẩm được cung cấp từ các nhà máy nằm trong một tập con các địa điểm của một tập **J** **các địa điểm tiềm năng.** Đối với mỗi địa điểm **j ∈ J**, chi phí cố định để mở nhà máy tại **j** là **f**j và công suất của nó là **Kj.** Chi phí vận chuyển một đơn vị sản phẩm từ địa điểm **j** đến khách hàng **i** là **Cij.** Mục tiêu là tối thiểu hóa tổng chi phí, tổng chi phí của chi phí cố định cho mở các nhà máy và các chi phí vận chuyển.

Gọi **Xij (i ∈ I, j ∈ J)** biểu thị số lượng vận chuyển từ địa điểm **j** đến khách hàng **i** (**Xij** gọi là các biến luồng (flow variable)) và **yj (j ∈ J)** là một biến 0–1 cho biết nhà máy ở địa điểm j có mở cửa hay không (**yj** là các biến vị trí). Bài toán có thể được xây dựng công thức dưới dạng chương trình toán học sau:

…Một công thức tính tổng đau đầu nào đó trong sách…

**Ghi chú 9.1**. Đối với bất kỳ vectơ **y˜** nào của các biến vị trí, giá trị tối ưu (cụ thể cho cấu hình của nhà máy này) cho các biến luồng **x(y˜)** có thể được lấy ra bằng cách giải quyết vấn đề vận chuyển liên quan:

… Công thức…

Nếu **y˜ = y∗**, vectơ biến vị trí tối ưu, lời giải tối ưu cho bài toán ban đầu chỉ đơn giản là cho bởi **(y∗, x(y∗))**.

**Ghi chú 9.2.** Một giải pháp tối ưu của bài toán ban đầu luôn có thể được tìm thấy tại một điểm cực trị của đa diện các vectơ luồng khả thi (chắc các nghiệm khả thi trong đồ thị được biêu diễn là đa điện nào đó) được xác định bởi các ràng buộc:

\_\_Vẫn là công thức\_\_

Tính chất này xuất phát từ thực tế là bài toán The Capacitated Plant Location có thể được hiểu như là là bài toán chi phí cố định được xác định trong không gian của các biến luồng. Bài toán chi phí cố định này có một hàm mục tiêu lõm luôn nhận vào một điểm cực tiểu. Các giá trị tối ưu cho các biến vị trí có thể dễ dàng thu được từ vectơ luồng tối ưu bằng cách đặt **yj** bằng 1 khi ∑**(i∈I) Xij > 0** và bằng 0 nếu ngược lại.

9.3.2 Tabu search

Năm 1986, Fred Glover đã đề xuất một cách tiếp cận mới, ông gọi là tabu search, cho phép hill climbing để vượt qua việc tối ưu cục bộ. Trong thực tế, nhiều yếu tố trong tabu search đầu tiên này và một số yếu tố của các công trình xây dựng sau đó, đã được giới thiệu (Glover 1977), bao gồm bộ nhớ ngắn hạn để ngăn chặn sự đảo ngược của các bước chuyển hiện tại, và bộ nhớ tần số dài hạn để củng cố các phần hấp dẫn (attractive components). Nguyên tắc cơ bản của tabu search là theo đuổi việc tìm kiếm mỗi khi gặp phải một điểm tối ưu cục bộ, bằng cách cho phép các bước chuyển không cải thiện (non-improving moves); ngăn chặn việc quay trở lại với các giải pháp trước bằng cách sử dụng bộ nhớ, được gọi là danh sách tabu, nơi lưu lại các lịch sử tìm kiểm gần. Ý tưởng chủ đạo trong việc khai thác thông tin để hướng dẫn việc tìm kiếm, có thể được kết nối với các phương pháp tìm kiếm thông tin được đề xuất vào cuối những năm 1980 trong lĩnh vực AI. Điều quan trọng cần lưu ý là Glover không xem tabu search là một phương pháp heuristic, mà là một phương pháp metaheuristic, một chiến lược chung đã được điều chỉnh cụ thể phù hợp với các vấn đề hiện tại để hướng dẫn và kiếm soát ở trong heuristics

9.3.3 Không gian tìm kiếm và cấu trúc vùng lân cận

Như vừa đề cập, tabu search mở rộng các phương pháp hill climbing. Trong thực tế, tabu search cơ bản có thể được xem như là sự kết hợp của hill climbing với bộ nhớ ngắn hạn. Theo sau đó là hai yếu tố cơ bản của bất kì phương pháp tabu search nào là định nghĩa không gian tìm kiếm và cấu trúc vùng lân cận.

Không gian tìm kiếm đơn giản là khoảng không gian của tất cả các giải pháp có thể xem xét (ghé thăm) trong quá trình tìm kiếm. Ví dụ, trong vấn đề job shop scheduling trong mục 9.2.1, không gian tìm kiếm có thể đơn giản là tập các giải pháp khả thi cho vấn đề, trong đó mỗi điểm trong không gian tìm kiếm tương ứng với một tập hợp m machine schedules thỏa mãn tất cả các ràng buộc được chỉ định. Mặc dù trong trường hợp đó, định nghĩa về không gian tìm kiếm có vẻ khá tự nhiên, nhưng không phải lúc nào cũng như vậy. Bây giờ hãy xem xét vấn đề the capacitated plant location của mục 9.2.2: không gian khả thi liên quan đến cả các biến vị trí nguyên (integer location) và các biến luồng liên tục (continuous flow variables) được liên kết bởi các điều kiện nghiêm ngặt; hơn thế nữa, như đã nêu, đối với bất kì tập giá trị khả thi nào cho các biến vị trí, người ta có thể khá dễ dàng lấy các giá trị tối ưu cho các biến luồng bằng cách giải quyết vấn đề vận chuyển liên quan. Trong bối cảnh này, rõ ràng ta có thể sử dụng như một không gian tìm kiếm khả thi đầy đủ: điều này sẽ liên quan đến thao tác (manipulating) các biến vị trí và lưu lượng, đó không phải là một nhiệm vụ dễ dàng. Một không gian tìm kiếm hấp dẫn hơn là tập các vectơ khả thi của các biến vị trí, tức là các vectơ khả thi trong {0, 1}^|J| ( |J| là lực lượng của tập J), bất kỳ giải pháp nào trong không gian đó đang được hoàn thiện để mang lại giải pháp khả thi cho vấn đề ban đầu bằng cách tính toán các biến luồng tối ưu liên quan. Điều thú vị là hai định nghĩa có thể này không phải là những định nghĩa duy nhất. Thật vậy, trên cơ sở Chú thích 9.2, ta cũng có thể quyết định việc tìm kiếm bằng cách thay thế tập các điểm cực trị của tập các luồng vectơ khả thi và lấy các biến vị trí có liên quan bằng việc đơn giản lưu ý rằng một nhà máy phải mở bất cứ khi luồng được phân bổ cho nó. Trên thực tế, kiểu tiếp cận này đã được sử dụng thành công bởi Crainic et al. (2000) để giải quyết vấn đề the fixed-charge multi-commodity network design, là một vấn đề tổng quát hơn bao gồm capacitated plant location là một trường hợp đặc biệt. Cũng cần lưu ý rằng không phải lúc nào giới hạn không gian tìm kiếm cho các giải pháp khả thi cũng là một ý kiến hay. Trong nhiều trường hợp, hãy cho phép việc tìm kiếm chuyển sang các giải pháp không khả thi là đáng mong đợi và đôi khi cần thiết (mục 9.4.3 để thêm chi tiết).

Liên kết chặt chẽ với định nghĩa của không gian tìm kiếm là cấu trúc vùng lân cận. Tại mỗi lần lặp lại tìm kiếm tabu, các biến đổi cục bộ có thể áp dụng cho giải pháp hiện tại, được kí hiệu là S, định nghĩa một tập các giải pháp lân cận trong không gian tìm kiếm, kí hiệu là N(S) (vùng lân cận của S) . Về mặt hình thức, N(S) là tập con của không gian tìm kiếm dược định nghĩa bởi

N(S) = {giải pháp thu được bằng cách áp dụng một biến đổi cục bộ duy nhất cho S}

Nói chung, đối với bất kỳ vấn đề cụ thể nào, có nhiều cấu trúc vùng lân cận có thể (và thậm chí thu hút) hơn so với các định nghĩa không gian tìm kiếm. Điều này xuất phát từ thực tế rằng có thể có một số cấu trúc lân cận hợp lý cho một định nghĩa nhất định về không gian tìm kiếm. Điều này dễ dàng được minh họa trong vấn đề job shop scheduling. Để đơn giản hóa cuộc thảo luận, chúng ta giả định rằng không gian tìm kiếm là không gian khả thi.

Cấu trúc vùng lân cận đơn giản cho vấn đề job shop scheduling thu được bằng cách xem xét trình tự công việc liên quan đến machine schedule. Theo trình tự, vị trí của một công việc tương ứng với thứ tự xử lý của nó trên máy. Ví dụ, ta có thể di chuyển một công việc tới một vị trí khác hoặc hoán đổi vị trí của hai công việc.

Trong khi các cấu trúc vùng lân cận này chỉ liên quan đến một hoặc hai công việc, các vùng lân cận mà chúng xác định chứa tất cả các lịch trình khả thi có thể đạt được từ lịch trình hiện tại bằng cách di chuyển bất kỳ công việc đơn lẻ nào ở bất kỳ vị trí nào khác hoặc bằng cách hoán đổi hai công việc bất kỳ. Việc kiểm tra các khu vực lân cận này có thể khá khắt khe. Trong thực tế, điều này có thể giảm gánh nặng tính toán, bằng cách xác định một tập con hạn chế các bước di chuyển khả thi và có thể dẫn tới cải thiện. Độc giả quan tâm có thể tham khảo Vaessens et al. (1996) và Anderson et al. (1997) để thảo luận chi tiết hơn về những vấn đề này.

Khi các định nghĩa khác nhau về không gian tìm kiếm được xem xét cho một vấn đề, các cấu trúc vùng lân cận sẽ khác nhau ở một mức độ đáng kể. Điều này có thể được minh họa trong vấn đề capacitated plant location. Nếu không gian tìm kiếm được xác định liên quan đến các biến vị trí, cấu trúc vùng lân cận thường sẽ liên quan đến thứ gọi là di chuyển Thêm / Thả và Hoán đổi để thay đổi trạng thái của một vị trí tương ứng (mở một cơ sở đang đóng hoặc đóng một cơ sở đang mở) và di chuyển một cơ sở mở từ địa điểm này sang địa điểm khác (điều này tương đương với việc thực hiện đồng thời di chuyển một Thêm và một Thả). Tuy nhiên, nếu không gian tìm kiếm là tập hợp các điểm cực trị được liên kết với các luồng vectơ khả thi, thì các di chuyển này trở nên vô nghĩa. Thay vào đó, ta nên xem xét các bước di chuyển được xác định bằng việc áp dụng các trục vảo công thức quy hoạch tuyến tính của bài toán the transportation, trong đó mỗi thao tác xoay quanh trục (pivot operation) điều chỉnh cấu trúc luồng để di chuyển giải pháp hiện tại đến một điểm cực trị liền kề.

Cuộc thảo luận trước lẽ ra phải làm rõ một điểm chính: chọn một không gian và một cấu trúc vùng lân cận cho đến nay là bước quan trọng nhất trong thiết kế của bất kỳ phương pháp tabu search heuristic nào. Ở bước này, người ta phải sử dụng sự hiểu biết và kiến thức tốt nhất mà mình có về vấn đề hiện tại.

9.3.4: Tabus:

Tabus là một trong những điểm đặc biệt của tabu search khi so với hill climbing. Như đã đề cập, tabus được sử dụng để ngăn chặn việc quay trở lại khi di chuyển khỏi tối ưu cục bộ thông qua các bước chuyển không cải thiện. Ý nghĩa chính ở đây là khi tình huống đó xảy ra, cần phải làm gì đó để ngăn việc tìm kiếm lần theo các bước của chính nó về điểm xuất phát. Điều này đạt được bằng cách thực hiện một số hành động tabu nhất định. Điều này có thể có nghĩa là không cho phép việc tìm kiếm quay trở lại điểm đã ghé thăm gần đây trong không gian tìm kiếm hoặc không cho phép các bước di chuyển gần bị đảo ngược. Để ví dụ, trong vấn đề job shop scheduling, nếu một job j đã bị di chuyển tới một vị trí mới trong machine schedule, người ta có thể khai báo tabu moving rằng công việc đó trở lại vị trí cũ trong một vài lần lặp lại (con số này được gọi tabu tenure of the move)

Tabus được lưu trữ trong bộ nhớ ngắn hạn của cuộc tìm kiếm (tabu list) và thường chỉ có một lượng thông tin nhất định và khá hạn chế được ghi lại. Trong bất kỳ bối cảnh cụ thể nào, có một số khả năng liên quan đến việc thông tin được ghi lại. Ta có thể ghi lại các giải pháp hoàn chỉnh, nhưng việc này yêu cầu một lượng lớn dung lượng lưu trữ và làm cho việc kiểm tra xem một bước di chuyển tiềm năng có phải là tabu hay không. Do đó nó hiếm khi được sử dụng. Cách sử dụng tabus phổ biến nhất liên quan đến việc ghi lại một số phép biến đổi cuối cùng trong giải pháp hiện tại và cấm các phép biến đổi ngược (như trong ví dụ trên); còn lại dựa trên nét đặc trưng riêng của các giải pháp hoặc các bước di chuyển.

Để hiểu hơn cách vận hành của tabus, ta hãy quay trở lại các vấn đề tham khảo. Trong bài toán job shop scheduling, ta có thể định nghĩa tabus theo nhiều cách. Để tiếp tục ví dụ của chúng ta trong đó một công việc j vừa được chuyển từ vị trí p1 sang vị trí p2, ta có thể khai báo tabu moves đặc biệt j về vị trí p1 từ vị trí p2 và ghi lại điều này trong bộ nhớ ngắn hạn dưới dạng bộ ba (j, p2, p1). Lưu ý rằng kiểu tabu này sẽ không hạn chế việc tìm kiếm nhiều, nhưng việc xoay vòng có thể xảy ra nếu sau đó j được chuyển đến vị trí p3 và sau đó từ p3 đến p1. Một tabu mạnh hơn sẽ liên quan tới việc cấm di chuyển j trở lại p1 (không tính đến vị trí hiện tại của nó) và được ghi lại là (j, p1). Một tabu mạnh hơn nữa sẽ không cho phép di chuyển j, và sẽ đơn giản được ghi chú là (j).

Trong bài toán the capacitated plant location, tabus trong di chuyển Thêm/Thả nên cấm việc thay đổi trạng thái của biến vị trí bị ảnh hưởng (the affected location variable) và có thể được ghi lại bằng cách ghi chú index của nó. Tabus cho di chuyển Hoán đổi thì phức tạp hơn. Chúng có thể được khai báo liên quan đến địa điểm cơ sở đã đóng cửa , địa điểm cơ sở mở cửa, cả hai địa điểm (tức là thay đổi trạng thái của cả hai biến vị trí là tabu) hoặc đối với việc hoán đổi cụ thể.

Nhiều danh sách tabu có thể sử dụng đồng thời và đôi khi được khuyến khích. Ví dụ, trong bài toán capacitated plant location, nếu ta sử dụng cấu trúc vùng lân cận có chứa cả di chuyển Thêm / Thả và Hoán đổi, có thể nên giữ một danh sách tabu riêng cho từng loại bước đi.

Tabu list tiêu chuẩn thường được triển khai dưới dạng danh sách hình tròn có độ dài cố định. Tuy nhiên, đã được chỉ ra rằng tabus có độ dài cố định không phải lúc nào cũng ngăn chặn việc lặp lại và một số tác giả đã đề xuất thay đổi độ dài danh sách tabu trong quá trình tìm kiếm. Một giải pháp khác là tạo ngẫu nhiên the tabu tenure of each move trong một khoảng thời gian nhất định. Sử dụng cách tiếp cận này yêu cầu một kế hoạch hơi khác để ghi lại tabus, thường được lưu trữ dưới dạng thẻ trong một mảng. Các mục trong mảng này thường ghi lại số lần lặp cho đến khi một bước đi là tabu. Được cung cấp chi tiết trong Gendreau et al. (1994).

9.3.5. Aspiration Criteria

Mặc dù là trung tâm của phương pháp tabu search, nhưng tabus đôi khi quá mạnh. Chúng có thể cấm các bước di chuyển hấp dẫn, ngay cả khi không có nguy hiểm của việc lặp lại chu kì, hoặc có thể dẫn đến sự trì trệ về tổng thể quá trình tìm kiếm. Do đó, cần phải sử dụng các thiết bị thuật toán (algorithmic devices) cho phép người ta thu hồi (hủy bỏ) tabus. Chúng được gọi là Aspiration Criteria. Cách phổ biến và đơn giản nhất để sử dụng Aspiration Criteria, được tìm thấy trong hầu hết việc triển khai tabu search, là cho phép một tabu moves khi nó dẫn tới một giải pháp có giá trị mục tiêu tốt hơn giá trị của giải pháp tốt nhất hiện tại (vì giải pháp mới rõ ràng là chưa được ghé thăm trước đây). Aspiration criteria phức tạp hơn đã được đề xuất và thực hiện thành công (ví dụ, de Werra and Hertz 1989, and Hertz and de Werra 1991), nhưng chúng hiếm khi được sử dụng. Quy tắc quan trọng là nếu không xảy ra việc lặp lại chu kì, tabus có thể được bỏ qua.

9.3.6 Một template cho một tabu search đơn giản.

Chúng tôi hiện đang ở vị trí để đưa một mẫu chung cho tabu search, tích hợp các yếu tố mà chúng ta đã thấy cho đến nay. Chúng tôi giả sử rằng ta đang cố tối thiểu hóa một hàm f(S) (đôi khi được biết tới là một mục tiêu hoặc một hàm evaluation) trên một miền và chúng tôi áp dụng thứ được gọi là phiên bản cải tiến tốt nhất của tabu search. Đó là phiên bản mà ta chọn ở mỗi lần lặp bước đi tốt nhất (ngay cả khi điều này dẫn đến việc tăng hàm f (S)). Đây là phiên bản tabu search được sử dụng phổ biến nhất:

Bên dưới là cái gì đó quan trọng trong sách nhưng toàn kí hiệu.

9.3.7. Tiêu chí dừng (Termination Criteria):

Lưu ý rằng chúng tôi chưa chỉ định trong template của mình một tiêu chí chấm dứt. Về lý thuyết, việc tìm kiếm có thể tiếp tục mãi mãi, trừ khi giá trị tối ưu của vấn đề được biết trước. Trong thực tế, chắc chắn, việc tìm kiếm phải dừng lại ở một số điểm. Một vài tiêu chí dừng phổ biến nhất

* Sau một số lần lặp cố định (hoặc thời gian CPU cố định)
* Sau một số lần lặp liên tiếp mà không cải thiện giá trị hàm mục tiêu (tiêu chuẩn được dùng trong hầu hết các triển khai)
* Khi hàm mục tiêu chạm tới một ngưỡng giá trị đã được chỉ định

9.3.8 Xác suất tabu search và Candidate Lists

Bình thường, ta phải đánh giá hàm mục tiêu cho mọi phần tử của vùng lân cận N (S) của giải pháp hiện tại. Điều này có thể cực kỳ tốn kém nếu xét từ quan điểm tính toán. Trong xác suất tabu search, chỉ một mẫu thử ngẫu nhiên N‘(S) của N(S) được xem xét, do đó giảm đáng kể chi phí tính toán. Một tính năng hấp dẫn khác là tính ngẫu nhiên được thêm vào có thể hoạt động như một cơ chế chống việc lặp lại chu kì. Điều này cho phép ta sử dụng tabu lists ngắn hơn mức cần thiết nếu ta đã thực hiện việc khám phá toàn bộ vùng lân cận. Mặt xấu của việc này là có thể bỏ lỡ các giải pháp tuyệt vời (chi tiết ở 9.6.3). Cũng có thể chọn theo xác suất khi nào áp dụng tiêu chí tabu (tabu criteria).

Một cách khác để kiểm soát số bước di chuyển được kiểm tra bằng chiến lược candidate list, cung cấp các cách chiến lược hơn để tạo ra một tập hợp con hữu ích N’(S) của N(S). Trên thực tế, cách tiếp cận theo xác suất có thể được coi là một ví dụ của chiến lược candidate list và cũng có thể được sử dụng để sửa đổi chiến lược đó. Việc không giải quyết đầy đủ các vấn đề liên quan đến việc tạo candidate lists hiệu quả là một trong những thiếu sót dễ thấy giúp phân biệt cách triển khai tabu search ngây thơ với cách triển khai có cơ sở vững chắc hơn. Các thiết kế liên quan cho các chiến lược candidate lists được thảo luận trong Glover and Laguna (1997). Chúng tôi cũng thảo luận về một phương pháp hữu ích trong việc tiếp cận việc candidate trong mục 9.4.4