ĐẠI HỌC QUỐC GIA HÀ NỘI
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ

ĐỖ ĐỨC ĐÔNG

PHƯƠNG PHÁP TỐI ƯU ĐÀN KIẾN VÀ ỨNG DỤNG

Chuyên ngành: Khoa học máy tính
Mã số: 62.48.01.01

TÓM TẮT LUẬN ÁN TIẾN SĨ CÔNG NGHỆ THÔNG TIN

Hà nội - 2012

Công trình được hoàn thành tại: Trường Đại học Công nghệ - ĐH Quốc gia Hà nội.

Người hướng dẫn khoa học:
PGS.TS. Hoàng Xuân Huấn

Phản biện 1: PGS.TS. Phan Trung Huy
Trường Đại học Bách khoa Hà Nội

Phản biện 2: PGS.TS. Hà Quang Thụy
Trường Đại học Công nghệ, ĐHQGHN

Phản biện 3: PGS.TS. Đỗ Trung Tuấn
Trường Đại học Khoa học Tự nhiên, ĐHQGHN

Luận án sẽ được bảo vệ trước hội đồng cấp nhà nước chấm luận án tiến sĩ họp tại: Phòng 212-E3, Trường Đại học Công nghệ, 144 Xuân Thuỷ, Cầu Giấy, Hà Nội.
Vào hồi 9 giờ, ngày 18 tháng 12 năm 2012.

Có thể tìm hiểu luận án tại:
- Thư viện Quốc gia Việt nam
- Trung tâm Thông tin – Thư viện, Đại học Quốc gia Hà nội

MỞ ĐẦU

1. Tính cấp thiết của luận án

Trong thực tế và khi xây dựng các hệ thông tin, ta thường gặp các bài toán tối ưu tổ hợp (TƯTH). Trong đó phải tìm các giá trị cho các biến rời rạc để làm cực trị hàm mục tiêu nào đó. Đa số các bài toán này thuộc lớp NP-khó. Trừ các bài toán cỡ nhỏ có thể tìm lời giải bằng cách tìm kiếm vét cạn, còn lại thì thường không thể tìm được lời giải tối ưu.

Đối với các bài toán cỡ lớn không có phương pháp giải đúng, đến nay người ta vẫn dùng các cách tiếp cận sau:

1) Tìm kiếm heuristic để tìm lời giải đủ tốt;

2) Tìm kiếm cục bộ để tìm lời giải tối ưu địa phương;

3) Tìm lời giải gần đúng nhờ các thuật toán mô phỏng tự nhiên như: mô phỏng luyện kim, giải thuật di truyền, tối ưu bầy đàn,…

Hai cách tiếp cận đầu thường cho lời giải nhanh nhưng không thể cải thiện thêm lời giải tìm được, nên cách tiếp cận thứ ba đang được sử dụng rộng rãi cho các bài toán cỡ lớn.

Trong các phương pháp mô phỏng tự nhiên, tối ưu đàn kiến (*Ant Colony Optimization - ACO*) là cách tiếp cận metaheuristic tương đối mới, được giới thiệu bởi Dorigo năm 1991 đang được nghiên cứu và ứng dụng rộng rãi cho các bài toán TƯTH khó.

Các thuật toán ACO sử dụng kết hợp thông tin kinh nghiệm (heuristic) và học tăng cường qua các vết mùi của các con kiến nhân tạo để giải các bài toán TƯTH bằng cách đưa về bài toán tìm đường đi tối ưu trên đồ thị cấu trúc tương ứng của bài toán. Phương pháp này được áp dụng rộng rãi để giải nhiều bài toán khó và hiệu quả nổi trội của chúng so với các phương pháp mô phỏng tự nhiên khác đã được chứng tỏ bằng thực nghiệm.

Khi áp dụng các thuật toán tối ưu đàn kiến thông dụng như ACS và MMAS, người ta phải tìm một lời giải đủ tốt, trên cơ sở đó xác định các tham số cho cận trên và cận dưới của vết mùi. Điều này gây nhiều khó khăn khi áp dụng thuật toán cho các bài toán mới. Ngoài ra, lượng mùi cập nhật cho mỗi thành phần trong đồ thị tỷ lệ với giá trị hàm mục tiêu của lời giải chứa nó liệu có phản ánh đúng thông tin học tăng cường hay không cũng còn phải thảo luận.

Việc nghiên cứu sâu hơn về các thuật toán ACO và ứng dụng của nó đang được nhiều người quan tâm. Từ năm 1998 đến nay, cứ 2 năm thì có một hội nghị quốc tế về phương pháp này tổ chức ở Brussels.

2. Mục tiêu của luận án

1) Phân tích xu thế biến thiên của vết mùi trong các thuật toán ACO, trên cơ sở đó đề xuất các quy tắc cập nhật mùi dễ sử dụng và hiệu quả hơn.

2) Đề xuất các thuật toán giải một số bài toán thời sự.

3. Các đóng góp của luận án

Dựa trên các phân tích toán học, luận án đề xuất các quy tắc cập nhật mùi: Đa mức (MLAS), Max Min trơn (SMMAS). Ưu điểm nổi trội của thuật toán được kiểm định bằng thực nghiệm đối với các bài toán chuẩn như: lập lịch sản xuất (*Job Shop Scheduling - JSS*), người chào hàng (*Traveling Salesman Problem - TSP*), quy hoạch toàn phương nhị phân không ràng buộc (*Unconstrained Binary Quadratic Programming - UBQP*). Trường hợp các thông tin heuristic có ảnh hưởng nhiều tới kết quả tìm kiếm, luận án đề xuất quy tắc 3 mức (3-LAS) và kiểm định hiệu quả của nó qua bài toán người chào hàng. Thực nghiệm cho thấy hiệu quả của các quy tắc này như nhau nhưng quy tắc SMMAS đơn giản và dễ sử dụng hơn, thích hợp cho ứng dụng rộng rãi.

Nhờ quy tắc cập nhật mùi SMMAS, luận án đề xuất các thuật toán mới ứng dụng cho bài toán suy diễn haplotype, bài toán tìm tập hạt giống tối ưu. Ngoài ra, luận án cũng đưa ra lược đồ ứng dụng ACO, thuật toán di truyền xác định tham số khi dùng phương pháp SVM (*Support Vector Machine - SVM*) cho bài toán dự báo hoạt động điều hòa gen. Ưu điểm nổi trội của các đề xuất mới được kiểm nghiệm bằng thực nghiệm trên dữ liệu tin cậy.

4. Bố cục của luận án

Ngoài phần kết luận, luận án được tổ chức như sau.

Chương 1: Luận án giới thiệu một phát biểu bài toán tối ưu tổ hợp dạng tổng quát để tiện dụng về sau.

Chương 2: Những nét chính của phương pháp tối ưu đàn kiến được giới thiệu trong chương 2.

Chương 3: Dựa trên phân tích toán học về biến thiên vết mùi, luận án đề xuất các thuật toán mới MLAS, SMMAS và 3-LAS, hiệu quả của thuật toán được kiểm nghiệm trên hai bài toán cổ điển TSP và UBQP.

Chương 4: Trình bày thuật toán ACOHAP giải bài toán suy diễn haplotype.

Chương 5: Trình bày thuật toán AcoSeeD giải bài toán tìm tập hạt giống tối ưu ứng dụng trong tìm kiếm tương đồng của các chuỗi sinh học.

Chương 6: Giới thiệu thuật toán GASVM và ACOSVM để cải tiến dự báo hoạt động điều tiết gen.

Chương 1. Tối ưu tổ hợp

1.1. Bài toán tối ưu tổ hợp tổng quát

Về mặt hình thức, mỗi bài toán TƯTH ứng với một bộ ba (S, f, Ω), trong đó S là tập hữu hạn trạng thái (lời giải tiềm năng hay phương án), f là hàm mục tiêu xác định trên S còn Ω là tập các ràng buộc. Mỗi phương án $s \in S$ thỏa mãn các ràng buộc Ω gọi là phương án (hay lời giải) chấp nhận được. Mục đích của ta là tìm phương án chấp nhận được s^* tối ưu hóa toàn cục hàm mục tiêu f. Đối với mỗi bài toán, tồn tại một tập hữu hạn gồm n thành phần $C = \{c_1, \dots, c_n\}$ sao cho mỗi phương án s trong S đều biểu diễn được nhờ các liên kết của các thành phần trong nó. Cụ thể hơn, các tập S, C và Ω có các đặc tính sau.

1) Ký hiệu X là tập các vectơ trên C độ dài không quá h: $X = \{< u_0, \dots, u_k > u_i \in C \ \forall i \le k \le h\}$, khi đó mỗi phương án s trong S được xác định nhờ ít nhất một vectơ trong X như ở điểm 2.

2) Tồn tại tập con X^* của X và ánh xạ φ từ X^* lên S sao cho $\varphi^{-1}(s)$ không rỗng với mọi $s \in S$. Trong đó tập X^* có thể xây dựng được từ tập con C_0 nào đó của C nhờ mở rộng tuần tự dưới đây.

3) Từ C_0 mở rộng được thành X^* theo thủ tục tuần tự:

i) $x_0 = < u_0 >$ là mở rộng được với mọi $u_0 \in C_0$.

ii) Giả sử $x_k = < u_0, \dots, u_k >$ là mở rộng được và chưa thuộc X^*. Từ tập ràng buộc Ω, xác định tập con $J(x_k)$ của C, sao cho với mọi $u_{k+1} \in J(x_k)$ thì $x_{k+1} = < u_0, \dots, u_k, u_{k+1} >$ là mở rộng được.

iii) Với mọi $u_0 \in C_0$, thủ tục mở rộng nêu trên xây dựng được mọi phần tử của X^*.

Như vậy, mỗi bài toán TƯTH được xem là một bài toán cực trị hàm h biến, trong đó mỗi biến nhận giá trị trong tập hữu hạn C kể cả giá trị rỗng. Một cách nhìn khác, nó là bài toán tìm kiếm vectơ độ dài không quá h trên đồ thị đầy có các đỉnh có nhãn trong tập C.

1.2. Các ví dụ

Hai bài toán người chào hàng (TSP) và quy hoạch toàn phương nhị phân không ràng buộc (UBQP) được giới thiệu làm ví dụ cho các bài toán TƯTH.

1.3. Các cách tiếp cận

Các cách tiếp cận như tìm kiếm heuristic, tìm kiếm cục bộ, metaheuristic và thuật toán memetic cần dùng về sau được giới thiệu trong mục này.

Chương 2. Phương pháp tối ưu đàn kiến

Tối ưu đàn kiến (ACO) là một phương pháp metaheuristic dựa trên ý tưởng mô phỏng cách tìm đường đi từ tổ tới nguồn thức ăn của các con kiến tự nhiên. Đến nay nó được cải tiến đa dạng và có nhiều ứng dụng. Trước khi giới thiệu phương pháp ACO, luận án giới thiệu phương thức trao đổi thông tin gián tiếp của các con kiến thực và mô hình kiến nhân tạo.

2.1. Từ kiến thực đến kiến nhân tạo

Trên đường đi, mỗi con kiến để lại một chất hóa học gọi là vết mùi dùng để đánh dấu đường đi. Bằng cách cảm nhận vết mùi, kiến có thể lần theo đường đi đến nguồn thức ăn được các con kiến khác khám phá theo phương thức chọn ngẫu nhiên có định hướng theo nồng độ vết mùi để xác định đường đi ngắn nhất từ tổ đến nguồn thức ăn.

Mô phỏng kiến tự nhiên, người ta dùng đa tác tử (multiagent) làm đàn kiến nhân tạo, trong đó mỗi con kiến có nhiều khả năng hơn kiến tự nhiên. Mỗi con kiến nhân tạo (về sau sẽ gọi là kiến) có bộ nhớ riêng, có khả năng ghi nhớ các đỉnh đã thăm trong hành trình và tính được độ dài đường đi nó chọn. Ngoài ra các con kiến có thể trao đổi thông tin có được với nhau, thực hiện tính toán cần thiết, cập nhật mùi…

Nhờ các con kiến nhân tạo này (về sau cũng gọi đơn giản là kiến) Dorigo (1991) đã xây dựng hệ kiến (AS) giải bài toán người chào hàng, hiệu quả của nó so với các phương pháp mô phỏng tự nhiên khác như SA, GA đã được kiểm chứng bằng thực nghiệm và được phát triển, ứng dụng phong phú với tên gọi chung là phương pháp ACO.

2.2. Phương pháp ACO cho bài toán TƯTH tổng quát

Mục này giới thiệu tóm lược phương pháp tối ưu đàn kiến. Trước khi mô tả thuật toán tổng quát, ta cần tìm hiểu về đồ thị cấu trúc cho bài toán tối ưu tổ hợp.

2.2.1. Đồ thị cấu trúc

Xét bài toán TƯTH tổng quát được nêu trong mục 1.1 dưới dạng bài toán cực tiểu hoá (S, f, Ω), trong đó S là tập hữu hạn trạng thái, f là hàm mục tiêu xác định trên S còn Ω là các ràng buộc để xác định S qua các thành phần của tập hữu hạn C và các liên kết của tập này. Các tập S, C và Ω có các đặc tính đã nêu trong chương 1.

Như đã nói trong chương trước, mỗi bài toán TƯTH được xem như một bài toán tìm kiếm vectơ độ dài không quá h trên đồ thị đầy, các đỉnh có nhãn trong tập C. Để tìm các lời giải chấp nhận được, ta xây dựng đồ thị đầy với tập đỉnh V mà mỗi đỉnh của nó tương ứng với mỗi thành phần của C. Các lời giải chấp nhận được là

các vectơ xây dựng tuần tự theo thủ tục bước ngẫu nhiên như mô tả chi tiết trong mục 2.2.2.

Thông thường, đối với các bài toán thuộc loại NP-khó, người ta có các phương pháp heuristic để tìm lời giải đủ tốt cho bài toán. Các thuật toán ACO kết hợp thông tin heuristic này với phương pháp học tăng cường nhờ mô phỏng hành vi của đàn kiến để tìm lời giải tốt hơn.

Giả sử với mỗi cạnh nối các đỉnh $i, j \in C$ có trọng số heuristic $h_{i,j}$ để định hướng chọn thành phần mở rộng là j khi thành phần cuối của x_k là i theo thủ tục tuần tự ($h_{i,j} > 0 \ \forall(i,j)$). Ký hiệu H là vectơ các trọng số heuristic của cạnh tương ứng (trong bài toán TSP nó có thể là vectơ mà thành phần là nghịch đảo độ dài của cạnh tương ứng), còn τ là vectơ biểu thị các thông tin học tăng cường $\tau_{i,j}$ (về sau gọi là vết mùi, ban đầu được khởi tạo bằng τ_0>0) định hướng mở rộng x_k với thành phần cuối là i nhờ thêm thành phần j theo thủ tục tuần tự. Trường hợp đặc biệt, $h_{i,j}$ và $\tau_{i,j}$ chỉ phụ thuộc vào j thì các thông tin này chỉ để ở các đỉnh tương ứng. Không giảm tổng quát, ta sẽ xét cho trường hợp các thông tin này ở các cạnh.

Khi đó ta gọi đồ thị $G = (V, E, H, \tau)$ là *đồ thị cấu trúc* của bài toán tối ưu tổ hợp đang xét, trong đó V là tập đỉnh, H và τ là các thông tin đã nói ở trên còn E là tập cạnh của đồ thị sao cho từ các cạnh này có thể xây dựng được tập X^* nhờ mở rộng tập C_0 theo thủ tục tuần tự. Nếu không có thông tin heuristic thì ta xem H có các thành phần như nhau và bằng 1.

2.2.2. Mô tả thuật toán ACO tổng quát

Với điều kiện kết thúc đã chọn (có thể là số bước lặp hoặc và thời gian chạy cho trước), người ta dùng đàn kiến m con thực hiện lặp xây dựng lời giải trên đồ thị cấu trúc $G = (V, E, H, \tau)$ như sau. Trong mỗi lần lặp, mỗi con kiến chọn ngẫu nhiên một đỉnh $u_0 \in C_0$ làm thành phần khởi tạo $x_0 = \{u_0\}$ và thực hiện xây dựng lời giải theo thủ tục bước ngẫu nhiên để xây dựng lời giải. Dựa trên lời giải tìm được đàn kiến sẽ thực hiện cập nhật mùi theo cách học tăng cường.

Thủ tục bước ngẫu nhiên

Giả sử $x_k =< u_0, \ldots, u_k >$ là mở rộng được, từ các ràng buộc Ω xác định được tập con $J(x_k)$ của C sao cho với mọi $u_{k+1} \in J(x_k)$ thì $x_{k+1} =< u_0, \ldots, u_k, u_{k+1} >$ là mở rộng được hoặc $x_k \in X^*$ khi $J(x_k)$ là rỗng. Đỉnh $j = u_{k+1}$ để mở rộng được chọn với xác suất $P(j)$ như sau:

$$P(j) = \begin{cases} \frac{[\tau_{ij}]^{\alpha}[h_{ij}]^{\beta}}{\sum_{l \in J(x_k)}[\tau_{il}]^{\alpha}[h_{il}]^{\beta}} & j \in J(x_k) \\ 0 & j \notin J(x_k) \end{cases} \quad (2.1)$$

Quá trình mở rộng tiếp tục cho tới khi kiến r tìm được lời giải chấp nhận được $x(r)$ trong X^* và do đó $s(r) = \varphi(x(r)) \in S$.

Để tiện trình bày, về sau ta sẽ xem $x(r)$ và $s(r)$ như nhau và không phân biệt X^* với S.

Cập nhật mùi

Tùy theo chất lượng của lời giải tìm được mà vết mùi trên mỗi cạnh sẽ được điều chỉnh tăng hoặc giảm tùy theo đánh giá mức độ ưu tiên tìm kiếm về sau. Vì vậy, quy tắc cập nhật mùi được dùng làm tên gọi thuật toán và thường có dạng:

$$\tau_{i,j} \leftarrow (1-\rho)\tau_{i,j} + \Delta(i,j), \forall\ (i,j) \qquad (2.2)$$

Các bước thực hiện của các thuật toán ACO được mô tả trong hình 2.4.

```
Procedure Thuật toán ACO;
Begin
   Khởi tạo tham số, ma trận mùi, khởi tạo m con kiến;
   repeat
      for k = 1 to m do
         Kiến k xây dựng lời giải;
      end-for
      Cập nhật mùi;
      Cập nhật lời giải tốt nhất;
   until (Điều kiện kết thúc);
   Đưa ra lời giải tốt nhất;
End;
```

Hình 2.4: Thuật toán ACO

Nhận xét chung về các thuật toán ACO

Nhờ kết hợp thông tin heuristic, thông tin học tăng cường và mô phỏng hoạt động của đàn kiến, các thuật toán ACO có các ưu điểm sau:

1) Việc tìm kiếm ngẫu nhiên dựa trên các thông tin heuristic làm cho phép tìm kiếm linh hoạt và mềm dẻo trên miền rộng hơn phương pháp heuristic sẵn có, do đó cho ta lời giải tốt hơn và có thể tìm được lời giải tối ưu.

2) Sự kết hợp học tăng cường thông qua thông tin về cường độ vết mùi cho phép ta từng bước thu hẹp không gian tìm kiếm mà vẫn không loại bỏ các lời giải tốt, do đó nâng cao chất lượng thuật toán.

Chú ý. Khi áp dụng phương pháp ACO cho mỗi bài toán cụ thể, có ba yếu tố quyết định hiệu quả thuật toán:

1) *Xây dựng đồ thị cấu trúc thích hợp*. Việc xây dựng đồ thị cấu trúc để tìm được lời giải cho bài toán theo thủ tục tuần tự không khó. Khó khăn chính là với các bài toán cỡ lớn thì không gian tìm kiếm quá rộng, đòi hỏi ta sử dụng các ràng buộc Ω một cách hợp lý để giảm miền tìm kiếm cho mỗi con kiến. Cách xử lý bài toán suy diễn haplotype ở chương 4 minh họa cho điều này.

2) *Chọn thông tin heuristic*. Thông tin heuristic tốt sẽ tăng hiệu quả thuật toán. Tuy nhiên, nhiều bài toán ta không có thông tin này thì có thể đánh giá chúng như nhau. Khi đó lúc ban đầu, thuật toán chỉ đơn thuần chạy theo phương thức tìm kiếm ngẫu nhiên, vết mùi thể hiện định hướng của học tăng cường và thuật toán vẫn thực hiện được.

3) *Chọn quy tắc cập nhật mùi*. Quy tắc cập nhật mùi thể hiện chiến lược học của thuật toán. Nếu đồ thị cấu trúc và thông tin heuristic luôn phụ thuộc vào từng bài toán cụ thể thì quy tắc cập nhật mùi là yếu tố phổ dụng và thường dùng để đặt tên cho thuật toán. Có nhiều quy tắc cập nhật mùi đã được đề xuất, trong luận án này chúng tôi sẽ tìm quy tắc thích hợp cho hai loại bài toán tùy theo thông tin heuristic ảnh hưởng nhiều hay ít tới thủ tục tìm kiếm lời giải.

2.3. Phương pháp ACO giải bài toán TSP

Bài toán người chào hàng (*Traveling Salesman Problem - TSP*) là bài toán có nhiều ứng dụng trong thực tế, được phát biểu như sau: một người giới thiệu sản phẩm muốn tìm một hành trình ngắn nhất, xuất phát từ thành phố của mình, đi qua tất cả các thành phố mà khách hàng cần giới thiệu sản phẩm và sau đó trở về thành phố xuất phát với điều kiện các thành phố của khách hàng chỉ đi qua đúng một lần.

Bài toán TSP thuộc loại NP-khó và được xem là bài toán chuẩn để đánh giá hiệu quả của các thuật toán giải các bài toán TƯTH mới. Thuật toán ACO đầu tiên được gọi là hệ kiến (*Ant System - AS*), các thuật toán ACO về sau là cải tiến của AS và đều dùng bài toán TSP để thử nghiệm chất lượng.

Trong mục này giới thiệu các thuật toán chính để giải bài toán này như là ví dụ minh họa cho phương pháp ACO.

Hệ kiến (AS)

Trong mỗi bước lặp, sau khi tất cả các kiến xây dựng xong hành trình, vết mùi sẽ được cập nhật. Việc này sẽ thực hiện như sau: trước tiên tất cả các cạnh sẽ bị bay hơi theo một tỉ lệ không đổi, sau đó các cạnh có kiến đi qua sẽ được thêm một lượng mùi. Việc cập nhật mùi được thực hiện như sau:

$$\tau_{ij} \leftarrow (1-\rho)\tau_{ij} + \sum_{k=1}^{m} \Delta\tau_{ij}^{k} \ \forall (i,j) \in E, \tag{2.5}$$

trong đó $\Delta\tau_{ij}^{k}$ là lượng mùi do kiến k cập nhật trên cạnh mà kiến k đi qua. Giá trị này bằng:

$$\Delta\tau_{ij}^{k} = \begin{cases} \frac{1}{C^k} \ nếu \ cạnh \ (i,j) \ thuộc \ T^k \\ 0 \ \ ngược \ lại \end{cases} \tag{2.6}$$

trong đó C^k là độ dài hành trình T^k do kiến k xây dựng, giá trị này được tính bằng tổng độ dài các cạnh thuộc hành trình. Theo công thức (2.6), các cạnh thuộc hành trình tốt hơn sẽ được cập nhật nhiều hơn. Nói chung, cạnh nào càng có nhiều

kiến sử dụng và là cạnh thuộc hành trình ngắn sẽ càng được cập nhật mùi nhiều hơn và do đó sẽ được các kiến lựa chọn nhiều hơn trong các vòng lặp sau.

Hiệu quả của thuật toán AS so với các phương pháp metaheuristic khác có xu hướng giảm khi kích thước bài toán tăng vì vậy đã có nhiều nghiên cứu tập chung cải tiến thuật toán AS.

Hệ đàn kiến (ACS)

Thuật toán ACS (Dorigo & Gambardella, 1997) khác với AS ở ba điểm chính.

- Thứ nhất, đó là sự khai thác kinh nghiệm tìm kiếm mạnh hơn AS thông qua việc sử dụng quy tắc lựa chọn dựa trên thông tin tích lũy nhiều hơn.
- Thứ hai, việc bay hơi mùi và để lại mùi chỉ trên các cạnh thuộc vào lời giải tốt nhất đến lúc đó G-best (cập nhật mùi toàn cục).
- Thứ ba, mỗi lần kiến đi qua cạnh (i, j) để di chuyển từ i đến j, vết mùi sẽ bị giảm trên cạnh (i, j) để tăng cường việc thăm dò đường mới (cập nhật mùi cục bộ).

Hệ kiến Max-Min

Thuật toán MMAS (Stutzle & Hoos 2000) đề xuất với bốn điểm thay đổi so với AS.

- Thứ nhất, để tăng cường khám phá lời giải tốt nhất tìm được: chỉ kiến có lời giải tốt nhất tìm được trong lần lặp (I-best) hoặc tốt nhất đến lần lặp đó (G-best) được cập nhật mùi.
- Thứ hai, MMAS giới hạn vết mùi sẽ thuộc $[\tau_{min}, \tau_{max}]$.
- Thứ ba là vết mùi ban đầu được khởi tạo bằng τ_{max} và hệ số bay hơi nhỏ nhằm tăng cường khám phá trong giai đoạn đầu.
- Điểm thay đổi cuối cùng là vết mùi sẽ được khởi tạo lại khi tắc nghẽn hoặc không tìm được lời giải tốt hơn trong một số bước.

2.4. Một số vấn đề khác khi áp dụng ACO

Gutjahr khởi đầu cho nghiên cứu đặc tính hội tu của thuật toán MMAS không có thông tin heuristic. Ký hiệu $P(t)$ là xác suất tìm thấy lời giải của thuật toán MMAS trong vòng t phép lặp, $w(t)$ là lời giải tốt nhất ở bước lặp t. Nhờ sử dụng mô hình Markov không thuần nhất, Gutjahr đã chứng minh rằng với xác suất bằng 1 ta có :

1) $\lim_{t\to\infty} w(t) = w^*$, $\lim_{t\to\infty} P(t) = 1$ (2.12)

2) $\lim_{t\to\infty} \tau_{i,j} = \tau_{max}$ với mọi cạnh (i, j) thuộc lời giải tối ưu tìm được. (2.13)

Mô hình này của Gutjahr không áp dụng được cho ACS. Trường hợp MMAS không có thông tin heuristic, Stützle và Dorigo đã chứng minh rằng:

$$\forall \varepsilon > 0, \text{ với } t \text{ đủ lớn thì } P(t) > 1 - \varepsilon, \quad (2.14)$$

do đó $\lim_{t\to\infty} P(t) = 1.$ (2.15)

Các tác giả cũng suy luận rằng kết quả này cũng đúng cho ACS. Với giả thiết đã tìm được lời giải tối ưu sau hữu hạn bước, Stützle và Dorigo suy ra rằng vết mùi của các cạnh thuộc lời giải tối ưu tìm được hội tụ đến τ_{max} còn vết mùi trên các cạnh không thuộc lời giải này hội tụ về τ_{min} hoặc τ_0.

Tiếp theo trong luận án giới thiệu một số kỹ thuật nâng cao hiệu quả và giảm thời gian chạy của thuật toán như tìm kiếm cục bộ, thực hiện song song hóa, thông tin heuristic và chọn số lượng kiến.

Chương 3. Tính biến thiên của vết mùi và các thuật toán mới

Như đã nói trong chương trước, Gutjahr, Stützle và Dorigo đã xét tính hội tụ theo xác suất tới lời giải tối ưu của MMAS, ACS và sự hội tụ của cường độ vết mùi cho các biến thể của thuật toán MMAS mà chưa khảo sát cho ACS.

Tuy nhiên trong các bài toán tối ưu tổ hợp thì số phương án là hữu hạn nên kết quả về việc xác suất tìm thấy lời giải hội tụ về 1 khi số lần lặp dần ra vô hạn là không có nhiều ý nghĩa. Trong chương này luận án phân tích chi tiết hơn về các đặc tính biến thiên của vết mùi trong các thuật toán ACO thông dụng, trên cơ sở đó đề xuất các quy tắc cập nhật mùi mới. Kết quả thực nghiệm trên các bài toán TSP và UBQP cho thấy ưu điểm của các đề xuất này.

Trước khi phân tích toán học, ta biểu diễn lại thuật toán dưới dạng dễ khảo sát hơn.

3.1. Thuật toán tổng quát

Xét một bài toán TƯTH cực tiểu hoá (S, f, Ω) trong mục 2.2 với đồ thị cấu trúc: $G = (V, E, H, \tau)$, trong đó V là tập đỉnh, E là tập các cạnh, H là vectơ các trọng số heuristic của cạnh tương ứng, còn τ là vectơ vết mùi tích luỹ được (ban đầu được khởi tạo bằng τ_0>0), C_0 là tập đỉnh khởi tạo để xây dựng các lời giải chấp nhận được theo thủ tục bước ngẫu nhiên. Thuật toán sử dụng m kiến, thực hiện N_c bước lặp xây dựng lời giải nhờ thủ tục bước ngẫu nhiên như mô tả trong mục 2.2.

3.1.1. Quy tắc chuyển trạng thái

Giả sử kiến r đã xây dựng $x_k =< u_0, \dots, i >$ là mở rộng được, nó chọn đỉnh y thuộc $J(x_k)$ để mở rộng thành $x_{k+1} =< u_0, \dots, i, y >$ xác suất cho bởi công thức (3.1):

$$P(y/\tau, x_k) = \begin{cases} \frac{\tau_{i,y}^{\alpha} h_{i,y}}{\sum_{j\in J(x_k)} \tau_{i,j}^{\alpha} h_{i,j}} & y \in J(x_k) \\ 0 & y \notin J(x_k) \end{cases}. \quad (3.1)$$

Quá trình mở rộng tiếp tục cho tới khi kiến r tìm được lời giải chấp nhận được $x(r)$ với độ dài không quá h.

Chú ý. Quy tắc này khác một ít so với quy tắc chuyển trạng thái của thuật toán ACS và công thức 2.1, nhưng không ảnh hưởng tới các kết quả phân tích toán học về sau.

Ký hiệu $w(t)$ là lời giải tốt nhất các con kiến tìm được cho tới lần lặp thứ t và $w^i(t)$ là lời giải tốt nhất trong bước lặp thứ t, nếu $w^i(t)$ không tốt hơn $w(t-1)$ ta có $w(t) = w(t-1)$. Ta sẽ quan tâm tới các lời giải gần đúng $w(t)$ này.

3.1.2. Cập nhật mùi

Ở đây luận án xét hai quy tắc điển hình và được sử dụng phổ biến nhất hiện nay xuất phát từ ACS và MMAS. Giả sử g là một hàm giá trị thực xác định trên S sao cho $0 < g(s) < \infty \forall s \in S$ và $g(s) > g(s')$ nếu $f(s) < f(s')$ (trong bài toán TSP $g(s)$ là nghịch đảo độ dài đường đi tương ứng), khi đó ở mỗi bước lặp cường độ vết mùi sẽ thay đổi theo một trong các quy tắc sau đây.

Quy tắc ACS: Quy tắc này phỏng theo ACS, bao gồm cả cập nhật địa phương và toàn cục.

Cập nhật mùi địa phương. Nếu kiến k thăm cạnh (i,j), tức là $(i,j) \in s(k)$ thì cạnh này sẽ thay đổi mùi theo công thức:

$$\tau_{i,j} \leftarrow (1-\rho)\,\tau_{i,j} + \rho\tau_1 \tag{3.2}$$

Cập nhật mùi toàn cục. Cập nhật mùi toàn cục chỉ cho các cạnh thuộc $w(t)$:

$$\tau_{i,j} \leftarrow (1-\rho)\,\tau_{i,j} + \rho g\big(w(t)\big) \tag{3.3}$$

Quy tắc MMAS. Quy tắc này thực hiện theo MMAS. Sau khi mỗi con kiến đều xây dựng xong lời giải ở mỗi bước lặp, vết mùi được thay đổi theo công thức:

$$\tau_{i,j} \leftarrow (1-\rho)\,\tau_{i,j} + \Delta\tau_{i,j} \tag{3.4}$$

Trong đó,

$$\Delta\tau_{i,j} = \begin{cases} \rho g\big(w(t)\big) & (i,j) \in w(t) \\ \max\{\tau_1 - (1-\rho)\tau_{i,j}, 0\} & (i,j) \notin w(t) \end{cases} \tag{3.5}$$

ở đây τ_1>0 là tham số.

3.2. Phân tích toán học về xu thế vết mùi

Mục này chỉ nghiên cứu tính hội tụ của các thuật toán ACS và MMAS, sau khi ước lượng xác suất tìm thấy một phương án ở bước lặp t, luận án khảo sát sự thay đổi của vết mùi.

3.2.1. Ước lượng xác suất tìm thấy một phương án

Mệnh đề 3.1. Các khẳng định sau đúng.

a) Bài toán tổng quát luôn có lời giải tối ưu.

b) Với mỗi kết quả thực nghiệm, các giá trị $f(w(t))$ luôn hội tụ cho mỗi lần chạy khi t dần ra vô hạn.

c) Ta có đánh giá sau.

$$0 < \tau_{min} = min\{\tau_0, \tau_1, g(w(1))\} \le \tau_{i,j} \le max\{\tau_0, \tau_1, g(w(1))\} = \tau_{max} \quad (3.6)$$

Về sau ta sẽ giả thiết $\tau_1 < g(w(t)) \ \forall t$ và như vậy $\tau_{max} = g^*$.

Định nghĩa. Với mọi i thuộc V, đại lượng $k_*(i) = min\left\{\frac{h_{i,j}}{h_{i,k}} : j,k \in V\right\}$ được gọi là hệ số lệch heuristic của đỉnh i còn đại lượng $k_* = min\{k_*(i): i \in V\}$ được gọi là hệ số lệch heuristic của bài toán.

Với mọi $s \in S$, ta ký hiệu $p_s(t)$ là xác suất để m con kiến tìm được s ở bước lặp t, mệnh đề sau cho ta một ước lượng cận dưới của nó.

Định lý 3.1. Với mọi $s \in S$ và với mọi t, ta luôn có:

$$p_s(t) \ge p_{min} > 0 \quad (3.7)$$

trong đó p_{min} xác định bởi công thức: $p_{min} = 1 - \exp\left(-\frac{mk_*^h \tau_{min}^{h\alpha}}{n^h \tau_{max}^{h\alpha}}\right)$

Định lý 3.2. Với mọi $\varepsilon > 0$ bé tuỳ ý, tồn tại T sao cho với mọi $t > T$ ta đều có: $P(t) > 1 - \varepsilon$.

3.2.2. Đặc tính của vết mùi

Ta thấy rằng trong thực tế, ở các bước lặp t đủ lớn thì khả năng $g(w(t)) > g(w(t+1))$ (và do đó $w(t+1) \neq w(t)$) rất bé nên có thể từ bước lặp t_0 có các cạnh (i,j) không bao giờ thuộc vào $w(t)$ $\forall t > t_0$ hoặc luôn thuộc vào nó. Ta sẽ khảo sát đặc điểm của $\tau_{i,j}$ trong các trường hợp này.

Định lý 3.3. Giả sử cạnh (i,j) thuộc vào lời giải chấp nhận được s nào đó và tồn tại T sao cho $(i,j) \notin w(t); \forall t \ge T)$ thì các khẳng định sau đúng.

a) $\tau_{i,j}(t)$ hội tụ theo xác suất tới τ_1 nếu dùng quy tắc cập nhật mùi ACS.

b) $\tau_{i,j}(t) = \tau_1$ với mọi $t > T + \frac{ln\frac{\tau_1}{g^*}}{ln(1-\rho)}$ nếu dùng quy tắc cập nhật mùi MMAS.

Định lý 3.4. Giả sử cạnh $(i,j) \in w(t)$ $\forall t \ge T$ thì các khẳng định sau đúng.

a) Nếu cập nhật mùi theo ACS thì:

$$\lim_{t\to\infty} \tau_{i,j}(t) \ge \tau_1 + \rho \frac{g(w(T)) - \tau_1}{1-(1-\rho)^{m+1}} \quad (3.13)$$

b) Nếu cập nhật mùi theo MMAS thì:

$$\lim_{t\to\infty} \tau_{i,j}(t) \ge g(w(T)) \quad (3.14)$$

3.3. Thảo luận

Ta thấy chất lượng của thông tin heuristic tốt sẽ nâng cao hiệu quả thuật toán, tuy nhiên các quy tắc này không phải luôn có và rất khó can thiệp để thay đổi chất

lượng. Do vậy ta sẽ quan tâm tới cách cập nhật mùi để nâng cao chất lượng thuật toán. Dưới đây, sau khi nhận xét chung về đặc tính khai thác và khám phá của các thuật toán, luận án sẽ nhận xét về các quy tắc cập nhật mùi đã nêu ở trên và đưa ra một số đề xuất.

Tính khai thác là việc tập trung tìm kiếm lời giải quanh phạm vi của các cạnh (i,j) thuộc các lời giải tốt nhất đã biết tới thời điểm đang xét còn tính khám phá là tìm kiếm ở các phạm vi khác. Trong cách cập nhật mùi G-best, ta đã biết $w(t)$ nên việc tìm kiếm quanh nó sẽ hạn chế nhiều tính khám phá còn khi cập nhật theo I-best sẽ mở rộng miền này hơn. Vì vậy trong thực hành cập nhật theo I-best tốt hơn G-best.

Trong các bài toán tối ưu tổ hợp, thường thì xác suất để một phương án cho trước được các kiến tìm được trong mỗi phép lặp rất bé. Vì vậy có thể sau một số bước lặp cường độ vết mùi trên mỗi cạnh không thuộc $w(t)$ sẽ bé và giảm khả năng khám phá được chúng mặc dù chúng có thể vẫn rất hứa hẹn thuộc lời giải tốt. Chẳng hạn, với bài toán TSP ta có mệnh đề sau.

Mệnh đề 3.2. Trong bài toán TSP không định hướng, mỗi chu trình Hamilton (đường liền) qua cạnh (i,j) và không qua cạnh (k,h) có thể đổi nhiều nhất 7 cạnh để có được chu trình đi qua cạnh (k,h) mà không qua (i,j).

Các điểm hạn chế của ACO.

Mệnh đề trên cho thấy khi thuật toán mới bắt đầu, các vết mùi khởi tạo như nhau thì một cạnh (k,h) "tốt hơn" cạnh (i,j), do nó thuộc chu trình dài hơn có thể đảo ngược một cách rất ngẫu nhiên. Khi một cạnh do ngẫu nhiên mà không được cập nhật mùi sau một số bước thì cường độ mùi của nó nhanh chóng bị giảm xuống và khó được các con kiến chọn sau đó mặc dù "chất lượng" của nó chưa chắc đã là "xấu".

Nếu khởi tạo mùi như nhau và không dùng thông tin heuristic thì xác suất của mỗi cạnh được mỗi con kiến đã cho sử dụng trong lần lặp đầu là $\frac{2}{n-1}$, xác suất này rất bé khi n lớn. Như vậy tùy theo từng loại bài toán mà tỷ lệ giữa τ_0 và τ_1 rất có ý nghĩa để cân bằng giữa tính khám phá và khai thác của thuật toán.

Các lượng mùi cập nhật của ACS và MMAS phụ thuộc vào giá trị hàm mục tiêu của lời giải mà các con kiến xây dựng được trong các bước lặp. Việc xác định các giá trị τ_0, τ_1 hay τ_{min} , τ_{max} cũng phụ thuộc vào tương quan với các giá trị chưa được xác định trước này của từng bài toán thì thuật toán mới tốt được.

3.4. Đề xuất các phương pháp cập nhật mùi mới

Dựa trên các phân tích trên, luận án đề xuất các quy tắc cải tiến của ACS và MMAS.

a) Phương pháp cập nhật mùi đa mức: MLAS

Dựa vào nhận xét ở mục trước, thay cho việc bay hơi vết mùi ở các thành phần không thuộc các lời giải của mỗi con kiến trong mỗi lần cập nhật mùi ở mỗi bước lặp ta cho τ_1 và τ_{max} tăng dần. Độ lệch giữa τ_1 và τ_{max} cho phép ta điều khiển tính hội tụ và khám phá. Nếu thấy lời giải tốt ít thay đổi thì cho τ_1 gần τ_{max} để tăng tính khám phá và ngược lại cho τ_1 dịch xa τ_{max} để cho lời giải tập trung tìm kiếm quanh lời giải tốt nhất tìm được.

Quy tắc này đã thử nghiệm cho các bài toán TSP và JSS cho kết quả khả quan so với MMAS. Tuy nhiên việc điều khiển độ lệch giữa τ_1 và τ_{max} rất khó cho các bài toán cụ thể nên chúng tôi thay bởi phương pháp 3-LAS sẽ trình bày ở phần c) dưới đây.

b) Phương pháp Max-Min trơn: SMMAS

Dựa vào nhận xét ở mục trên, ta thấy không nên giảm vết mùi ở các cạnh không thuộc lời giải tốt quá nhanh như quy tắc MMAS mà nên dùng quy tắc Max-Min trơn như sau:

$$\tau_{i,j} \leftarrow (1-\rho)\tau_{i,j} + \Delta\tau_{i,j}$$
$$\text{với } \Delta\tau_{i,j} = \begin{cases} \rho\tau_{min} \text{ nếu } (i,j) \notin w(t) \\ \rho\tau_{max} \text{ nếu } (i,j) \in w(t) \end{cases} \quad (3.16)$$

Khi cài đặt, lấy $\tau_0 = \tau_{max}$.

c) Phương pháp 3-LAS

Đối với các bài toán mà thông tin heuristic ảnh hưởng nhiều tới chất lượng tìm kiếm lời giải, chẳng hạn như bài toán TSP thì phương pháp 3-LAS tương tự ACS nhưng dễ dùng hơn và hiệu quả tốt hơn. Phương pháp này dùng thêm tham số τ_{mid} thuộc khoảng (τ_{min}, τ_{max}) và cập nhật mùi tương tự SMMAS cho các cạnh có kiến sử dụng hoặc thuộc $w(t)$, cụ thể là:

$$\tau_{i,j} \leftarrow (1-\rho)\tau_{i,j} + \Delta\tau_{i,j}$$
$$\text{với } \Delta\tau_{i,j} = \begin{cases} \rho\tau_{max} & \text{nếu } (i,j) \in w(t) \\ \rho\tau_{mid} & \forall (i,j) \in w(t) \text{ và có kiến đi qua} \\ \rho\tau_{min} & \text{cho các cạnh còn lại} \end{cases} \quad (3.17)$$

3.5. Nhận xét về các thuật toán mới

Trong ba phương pháp cập nhật mùi ở trên, hai phương pháp SMMAS và 3-LAS đơn giản và dễ sử dụng hơn nên luận án sẽ nêu ra các ưu điểm của hai thuật toán này khi sử dụng và nhận xét về tính bất biến của chúng.

Ưu điểm khi sử dụng

Ta thấy thuật toán SMMAS và 3-LAS có một số ưu điểm nổi trội sau so với ACS và MMAS.

1) Với ACS và MMAS, để xác định τ_0 hay $\tau_{\min}$ và $\tau_{\max}$ người ta cần tìm một lời giải theo phương pháp heuristic và dựa vào giá trị hàm mục tiêu của nó. Vì giá trị hàm mục tiêu này nhận được ngẫu nhiên, nên khó xác định tốt tham số cho học tăng cường. Quy tắc cập nhật mới cho phép ta xác định các tham số này đơn giản và hợp lý hơn, cụ thể: trong SMMAS và 3-LAS ta không cần xác định chính xác giá trị $\tau_{\min}, \tau_{\max}$ mà chỉ cần xác định tỉ lệ giữa $\tau_{\min}, \tau_{\max}$. Trong thực nghiệm, luận án luôn thiết đặt $\tau_{\max} = 1.0$ và xác định $\tau_{\min}$ qua tỉ lệ giữa $\tau_{\min}, \tau_{\max}$. Cần nhấn mạnh rằng, việc chỉ cần lựa chọn tỉ lệ giữa $\tau_{\min}, \tau_{\max}$ đơn giản và mất ít thời gian thực nghiệm hơn rất nhiều so với việc lựa chọn cụ thể hai tham số $\tau_{\min}, \tau_{\max}$.

2) Việc thêm mùi cho các cạnh thuộc lời giải tốt ở mỗi bước lặp trong thuật toán ACS và MMAS, ta phải xây dựng hàm để tính lượng mùi được thêm dựa trên chất lượng lời giải do kiến xây dựng được. Ví dụ, trong bài toán TSP, ACS và MMAS sử dụng hàm nghịch đảo độ dài đường đi được kiến xác định. Điều này cũng là một trong những khó khăn khi áp dụng ACS (hoặc MMAS) đối với một bài toán mới. Tuy nhiên, trong SMMAS và 3-LAS không cần phải xây dựng hàm này.

3) Dễ dàng kiểm tra được các thuật toán này có cùng độ phức tạp như MMAS và ACS, nhưng ít phép toán hơn MMAS vì không phải tính hàm mục tiêu ở lượng mùi cập nhật và không phải so sánh để giới hạn vết mùi trong khoảng $\tau_{\min}, \tau_{\max}$. Theo cách cập nhật của SMMAS và 3-LAS, vết mùi luôn trong khoảng $\tau_{\min}, \tau_{\max}$.

Tính bất biến

Hai bài toán TƯTH (S, f, Ω) và (S, f', Ω), ta sẽ gọi chúng là hai thể hiện I và I' tương ứng của một bài toán nếu $f'(s) = g(f(s))$ với mọi s thuộc S trong đó g là hàm đơn điệu tăng chặt. Với giả thiết về tính lặp của máy tạo số giả ngẫu nhiên ta có kết luận.

Định lý 3.5. Giả sử I và I' là hai thể hiện của một bài toán TƯTH tùy ý thì khi giải bằng một trong hai thuật toán SMMAS hoặc 3-LAS với cùng số lần lặp nhờ dùng một máy phát lặp sẽ cho ta cùng một dãy lời giải và các vectơ vết mùi.

3.6. Kết quả thực nghiệm cho hai bài toán TSP và UBQP

Luận án thực nghiệm các thuật toán mới cho bài toán TSP và so sánh với MMAS. Ngoài ra, luận án cũng so sánh SMMAS với MMAS cho bài toán UBQP. Thực nghiệm cho thấy SMMAS đơn giản nhất mà tốt như MLAS, 3-LAS và các phương pháp mới đề xuất đều tốt hơn MMAS.

Chương 4. Thuật toán ACOHAP giải bài toán suy diễn haplotype

Suy diễn haplotype giúp ta hiểu được cấu trúc di truyền của quần thể dựa trên dữ liệu kiểu gen (genotype) của các tổ chức lưỡng bội. Theo tiêu chuẩn tìm tập haplotype nhỏ nhất (pure parsimony), bài toán suy diễn haplotype trở thành bài toán tối ưu tổ hợp thuộc lớp NP-khó. Chương này, luận án đề xuất một thuật toán hiệu quả có tên là ACOHAP giải bài toán suy diễn haplotype theo tiêu chuẩn pure parsimony. Thực nghiệm trên dữ liệu chuẩn và dữ liệu thực cho thấy ưu điểm nổi trội của nó so với các phương pháp tốt nhất hiện thời.

4.1. Bài toán suy diễn haplotype và tiêu chuẩn pure parsimony

Trong các tổ chức lưỡng bội, hầu hết các nhiễm sắc thể có hai "*bản sao*" không giống nhau. Một haplotype là một *bản sao* của một genotype trong một tổ chức lưỡng bội, nó mang các thông tin cho phép nghiên cứu các triệu chứng và tác nhân gây bệnh di truyền.

Bài toán suy diễn haplotype là từ một tập n genotype có độ dài m, xác định tập haplotype sao cho các cặp kết hợp từ chúng tạo nên được tập genotype đang xét. Hiện nay, bài toán suy diễn haplotpe là thách thức quan trọng trong nghiên cứu di truyền của các sinh vật lưỡng bội nói chung và con người nói riêng.

Trong biểu diễn dạng toán học của bài toán suy diễn haplotype, mỗi genotype được biểu diễu bằng một xâu độ dài m các ký tự thuộc tập {0, 1, 2}. Các ký tự 0 và 1 thể hiển allene của genotype ở vị trí tương ứng là đồng hợp tử, ký tự 0 biểu thị allen dạng tự nhiên (wild type) và ký tự 1 biểu thị allen dạng biến dị (mutant), còn ký tự 2 biểu thị cặp allen ở vị trí tương ứng là dị hợp tử. Mỗi haplotype là một xâu độ dài m các ký tự thuộc tập {0,1}. Tại vị trí dị hợp tử, genotype được kết hợp từ hai haplotype mà ở vị trí này một có dạng tự nhiên và một có dạng biến dị.

Với một genotype, ta cần tìm một cặp không thứ tự của haplotype có thể giải thích theo định nghĩa sau:

Định nghĩa 4.1. (Giải thích genotype)

Cho một genotype g, chúng ta nói rằng cặp haplotype không thứ tự $< h^a, h^b >$ giải thích g (hay g được giải thích bởi $< h^a, h^b >$) và ký hiệu là $< h^a, h^b > \triangleright g$ nếu chúng thỏa mãn điều kiện sau với mọi vị trí $i = 1,2, \dots, m$:

- nếu $g_i = 0$ thì $h_i^a = h_i^b = 0$,
- nếu $g_i = 1$ thì $h_i^a = h_i^b = 1$,
- nếu $g_i = 2$ thì $(h_i^a = 0 \wedge h_i^b = 1)$ hoặc $(h_i^a = 1 \wedge h_i^b = 0)$

Với một genotype, ký tự trên cặp haplotype ở vị trí các đồng hợp tử hoàn toàn xác định còn ký tự ở vị trí dị hợp tử thì có hai khả năng nhận giá trị. Nếu trong

genotype có l vị trí là dị hợp tử thì sẽ có 2^{l-1} cặp không thứ tự haplotype giải thích nó.

Với một danh sách n genotype $G = (g^1, \dots, g^n)$ có độ dài m đã cho, trong đó $g^s = (g_1^s, \dots, g_m^s)$ và $g_i^s \in \{0,1,2\}$ với mọi $s \leq n$ và $i \leq m$, ta định nghĩa các haplotype giải thích nó như sau.

Định nghĩa 4.2. (giải thích tập genotype)

Cho một danh sách n genotype $G = (g^1, \dots, g^n)$ có độ dài m, ta nói một danh sách $2n$ haplotype $H = (h^{1a}, h^{1b}, h^{2a}, h^{2b}, \dots, h^{na}, h^{nb})$ là một giải thích của G nếu g^s được giải thích bởi cặp haplotype $< h^{sa}, h^{sb} >$ với mọi $s \leq n$.

Suy diễn haplotype theo tiêu chuẩn pure parsimony

Như vậy, với một danh sách n genotype $G = (g^1, \dots, g^n)$ có độ dài m, bài toán suy diễn haplotype là tìm danh sách $2n$ haplotype $H = (h^{1a}, h^{1b}, h^{2a}, h^{2b}, \dots, h^{na}, h^{nb})$ giải thích *hợp lý* các genotype này.

Hiện có hai cách tiếp cận chính cho bài toán này là phương pháp tổ hợp và thống kê. Lời giải cho bài toán tùy thuộc vào mô hình di truyền là tiêu chuẩn cho xác định tập haplotype. Trong phương pháp tổ hợp, tiêu chuẩn *pure parsimony* nhằm tìm tập hapelotype nhỏ nhất giải thích G do Gusfield đề xuất đang được nhiều người sử dụng. Bài toán theo tiêu chuẩn này ký hiệp là HIPP (*Haplotype Inference by Pure Parsimony*)

4.2. Thuật toán ACOHAP giải bài toán HIPP

Trong các thuật toán ACO truyền thống, trong đó các con kiến xây dựng lời giải theo thủ tục bước ngẫu nhiên trên đường đi liên tục. Ở thuật toán này đồ thị cấu trúc là đồ thị con của cây nhị phân độ sâu m. Chúng được xác định động theo mỗi kiến ở từng bước lặp. Mỗi mức của đồ thị biểu thị cho một vị trí của các haplotype mà kiến xây dựng lời giải.

4.2.1. Đồ thị cấu trúc

Về hình thức, đồ thị cấu trúc là cây nhị phân đầy đủ có độ sâu m. Tuy nhiên để tránh bùng nổ tổ hợp khi m lớn, đối với mỗi kiến ở mỗi bước ta chỉ hiện thị một cây con T của cây nhị phân đầy đủ được trích nhờ quá trình xây dựng lời giải của nó với nút gốc ở mức 0 và các nút lá ở mức m. Các cây này biểu thị khác nhau (động) phù hợp với quá trình xây dựng lời giải của mỗi kiến trong mỗi lần lặp và có các đặc điểm sau.

- Mỗi nút trong X ở mức i có hai nút con tại mức $i + 1$. Nhánh từ X sang con bên trái có nhãn là 0 (gọi là nhánh 0). Tương tự, nhánh từ X sang con bên phải có nhãn là 1 (gọi là nhánh 1).

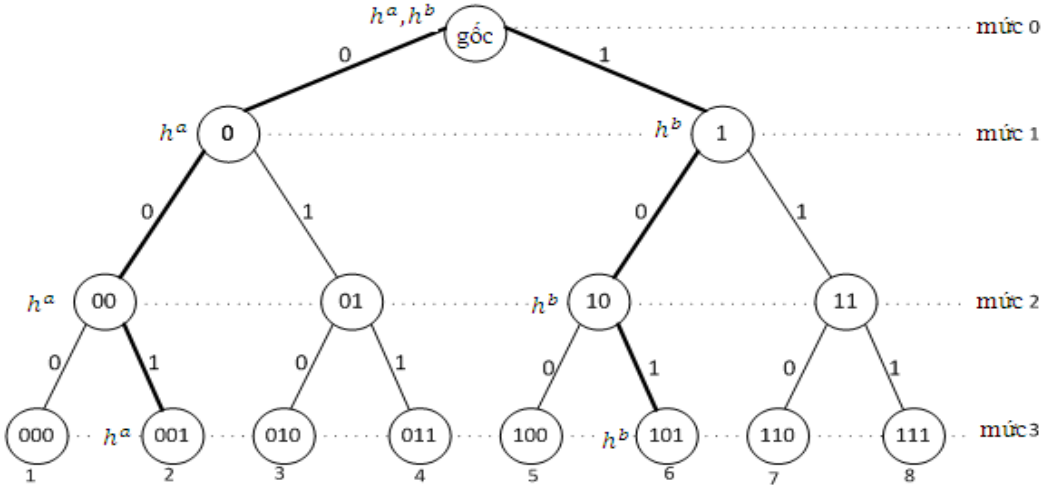
- Nhãn của nhánh trên đường đi từ nút gốc đến nút X tạo thành nhãn của nút X. Nhãn của nút X tại mức i là i ký tự đầu tiên của haplotype (nhãn của nút lá sẽ là một haplotype độ dài m)

- Mỗi nút có một danh sách kết hợp chỉ các haplotype được xây dựng nhờ đường đi đến nút này. Như vậy nút gốc luôn có danh sách kết hợp là $h_1^a, h_1^b, h_2^a, h_2^b, .., h_n^a, h_n^b$, các nút trên đường đi từ gốc đến lá có danh sách tương ứng giảm dần.

- Mỗi đường đi từ gốc đến lá xác định haplotype có trong danh sách tương ứng ở nút lá và nhãn của nút xác định nội dung của haplotype.

Như vậy đồ thị T này có nhiều nhất $2n$ nút lá biểu thị các haplotype cần tìm chứ không phải có 2^m nút như cây nhị phân đầy đủ. Đồ thị này không xác định ngay từ đầu mà được hiển thị dần theo quá trình xây dựng lời giải (sẽ được nói rõ hơn ở phần dưới). Hình 4.2 mô tả cây độ dài bằng 3 giúp xây dựng cặp haplotype giải thích genotype $g = 201$.

Thủ tục xây dựng lời giải của mỗi con kiến dưới đây sẽ giúp hiểu rõ hơn tính mềm dẻo của đồ thị cấu trúc và cách xây dựng.



Hình 4.2. Đồ thị cấu trúc giải bài toán HIPP

4.2.2. Thủ tục xây dựng lời giải của mỗi con kiến

Thuật toán xây dựng đồng thời $2n$ haplotype của mỗi con kiến lần lượt theo từng vị trí để suy diễn cả n genotype của G. Để thực hiện xây dựng lời giải, mỗi nút của cây sẽ có một danh sách haplotype kết hợp có ý nghĩa các haplotype trong danh sách sẽ nhận giá trị là nhãn của nút đó cho các vị trí từ đấy về trước.

Ban đầu, nút gốc được khởi tạo có một danh sách kết hợp gồm $2n$ haplotype $(h^{1a}, h^{1b}, .., h^{na}, h^{nb})$ rồi thực hiện m lần lặp, trong đó lần lặp thứ i sẽ xác định giá trị ở vị trí thứ i cho tất cả các haplotype và tạo danh sách kết hợp cho các nút ở mức i (trước đó danh sách này rỗng). Mỗi lần lặp, kiến thực hiện lần lượt hai bước: bước thứ nhất xử lý đồng hợp tử và bước thứ hai xử lý dị hợp tử.

Bước thứ nhất: xử lý đồng hợp tử. Các genotype mà vị trí thứ i là đồng hợp tử thì các cặp haplotype tương ứng ở vị trí thứ i sẽ nhận giá trị bằng giá trị vị trí thứ i trên genotype mà chúng giải thích. Cụ thể, nếu $g_i^s = 0/1$ thì h_i^{sa}, h_i^{sb} nhận giá trị 0/1. Khi đó, h^{sa}, h^{sb} được thêm vào danh sách nút con theo nhánh 0/1 tương ứng.

Bước thứ hai: xử lý dị hợp tử. Các genotype mà vị trí thứ i là dị hợp tử thì giá trị hai haplotype tương ứng ở vị trí thứ i sẽ có giá trị khác nhau, như vậy nếu xác định được giá trị thứ i của haplotype thứ nhất sẽ tính được giá trị thứ i của haplotype thứ hai. Cụ thể, nếu $g_i^s = 2$ thì h_i^{sa} sẽ lựa chọn 0 hoặc 1, h_i^{sb} sẽ bằng $1 - h_i^{sa}$. Nếu ở danh sách của nút mức $(i-1)$ chứa h^{sa} thì kiến lựa chọn ngẫu nhiên h_i^{sa} theo xác suất như sau:

$$P_i^s(v) = \frac{(\tau_{i,v}^s)^\alpha (\eta_{i,v}^s)^\beta}{(\tau_{i,0}^s)^\alpha (\eta_{i,0}^s)^\beta + (\tau_{i,1}^s)^\alpha (\eta_{i,1}^s)^\beta}$$

Trong đó α và β là hai tham số dương cho trước điều khiển ảnh hưởng giữa thông tin vết mùi và thông tin heuristic .

Thông tin heuristics. Ý tưởng chính để xác định thông tin heuristic cho nút đang xét là ước lượng số nút có danh sách kết hợp khác rỗng ở mức m sẽ tương thích với haplotype đang xét.

Cập nhật vết mùi. Sử dụng quy tắc SMMAS

Sử dụng tìm kiếm cục bộ. Để tăng hiệu quả thuật toán, trong mỗi lần lặp luận án sử dụng thuật toán tìm kiếm cục bộ cho lời giải tìm được theo chiến lược *tốt hơn* trong lân cận khoảng cách 1-Hamming (1-Hamming distance neighborhood) do Gaspero và Roli đề xuất.

4.3. Kết quả thực nghiệm

Luận án tiến hành làm thực nghiệm trên dữ liệu chuẩn (gồm 329 test) và bộ dữ liệu thực CEU (nhiễm sắc thể 20 của người da trắng châu Âu tại Utah) để so sánh với phương pháp RPoly và phương pháp CollHap. RPoly là phương pháp giải đúng tốt nhất hiện nay còn CollHap là phương pháp xấp xỉ tốt nhất hiện nay. Kết quả thực nghiệm cho thấy ACOHAP cho kết quả tối ưu như RPoly trong nhiều trường hợp và ACOHAP hiệu quả nổi trội hơn hẳn CollHap.

Chương 5. Thuật toán AcoSeeD tìm tập hạt giống tối ưu

Tìm kiếm các đoạn tương tự trong các chuỗi sinh học là một trong những công việc thường gặp và quan trọng nhất trong tin sinh học. Sử dụng tập hạt giống có cách đã nâng cao chất lượng tìm kiếm. Tuy nhiên, tìm tập hạt giống có cách tối ưu là bài toán thuộc lớp NP-khó. Chương này, luận án đề xuất thuật toán AcoSeeD có đồ thị cấu trúc hợp lý, dùng quy tắc cập nhật mùi SMMAS và kỹ thuật tìm kiếm

cục bộ được định hướng bằng một hàm mục tiêu xấp xỉ nhanh thay cho hàm mục tiêu chính trong phương pháp ACO. Kết quả thực nghiệm cho thấy AcoSeeD đã cải thiện đáng kể hiệu quả so với thuật toán tốt nhất hiện nay: SpEEDfast.

5.1. Bài toán tìm tập hạt giống có cách tối ưu và một số vấn đề liên quan

5.1.1. Bài toán tìm tập hạt giống tối ưu

Việc so khớp địa phương hai hay nhiều chuỗi sinh học được đưa về xét bài toán trên một miền tương đồng (homologous region) biểu diễn bằng xâu nhị phân R có độ dài N, ký tự 0 ở mỗi vị trí i của R biểu thị không khớp (mismatch) còn ký tự 1 biểu thị khớp (match). Chuỗi R sẽ gọi là chuỗi so khớp. Ta xét hạt giống biểu diễn bằng xâu ký tự gồm các ký tự 1 hoặc *, ký tự 1 biểu thị khớp còn ký tự * biểu thị khớp hoặc không khớp ở vị trí tướng ứng của hạt giống khi đối sánh với R.

Định nghĩa 5.1. (Tính hợp đúng được của hạt giống)

Với miền tương đồng biểu thị bởi chuỗi so khớp $R = r_1 r_2 .. r_N$ đã cho, hạt giống $s = s_1 s_2 .. s_l$ (l là độ dài hạt giống) được gọi là *hợp đúng được(hit)* R nếu tồn tại vị trí v của R sao cho với mọi $i = 1,2,..,l$ ta đều có:

$$r_{v+i-1} = \begin{cases} 1 & nếu\ s_i = 1 \\ 0\ hoặc\ 1 & nếu\ s_i = * \end{cases} \quad (5.1)$$

Số lượng ký tự 1 trong hạt giống s gọi là *trọng số* của nó.

Một tập hạt giống S gồm k hạt giống có cùng trọng số được gọi là *hợp đúng được* R nếu tồn tại một hạt giống hợp đúng được R.

Bây giờ ta xét chuỗi so khớp của hai chuỗi sinh học có xác suất khớp ở mỗi vị trí của chuỗi như nhau và bằng p, tức là các ký tự r_i ở mỗi vị trí i của chuỗi $R = r_1 r_2 .. r_N$ đều nhận giá trị 1 với xác suất p:

$$\mathrm{P}(r_i = 1) = p\ \forall i \le N, \quad (5.2)$$

khi đó p được gọi là mức tương tự (similarity level) của R.

Bài toán tìm tập giống tối ưu như sau: Với chuỗi so khớp R có mức tương tự p đã cho, tìm một tập S gồm k hạt giống có cùng trọng số w sao cho xác suất mà tập S hợp đúng được chuỗi này lớn nhất (xác suất để tập hạt giống S hợp đúng được chuỗi R gọi là độ nhạy của S).

Bài toán tìm tập hạt giống tối ưu được xét trong hai trường hợp: độ dài các hạt giống đã biết hoặc chưa biết. Trong cả hai trường hợp, Li và các cộng sự đã chứng minh các bài toán đều thuộc lớp NP-khó, đặc biệt ngay trong việc tính hàm mục tiêu cũng thuộc lớp NP-khó.

5.1.2. Các cách tiếp cận hiện nay

Bài toán tìm tập hạt giống tối ưu đã có nhiều thuật toán giải được công bố. Trong đó phải kể đến thuật toán heuristic tham ăn do Li và cộng sự đề xuất năm

2004, thuật toán leo đồi do Sun và Buhler đề xuất năm 2005. Do đặc điểm của bài toán là thời gian tính độ nhạy của tập hạt giống lớn nên Ilie và các cộng sự đã đề xuất thuật toán leo đồi sử dụng cách tính hàm mục tiêu xấp xỉ nhanh OC (Overlap Complexity) thay cho tính độ nhạy trong mỗi bước tính toán. Năm 2011, Ilie và cộng sự công bố phần mềm SpEED và được cho là phần mềm thể hiện thuật toán tìm tập hạt giống tốt nhất hiện nay. Phiên bản mới của phần mềm này là SpEEDfast được công bố năm 2012. Nhược điểm chính của thuật toán SpEED (hay SpEEDfast) là sử dụng thuật toán leo đồi đơn giản và chỉ sử dụng hàm mục tiêu OC trong tìm kiếm.

5.2. Thuật toán AcoSeeD giải bài toán tìm tập hạt giống

5.2.1. Mô tả thuật toán

Thuật toán AcoSeeD áp dụng phương pháp ACO theo lược đồ có sử dụng tìm kiếm cục bộ cho mỗi lời giải tìm được ở mỗi bước lặp. Vì thuật toán tính độ nhạy tốn nhiều thời gian chạy nên nó chỉ dùng để đánh giá chất lượng các lời giải sau khi đã áp dụng tìm kiếm cục bộ, còn trong quá trình tìm kiếm cục bộ thì hàm OC được áp dụng để định hướng tìm kiếm.

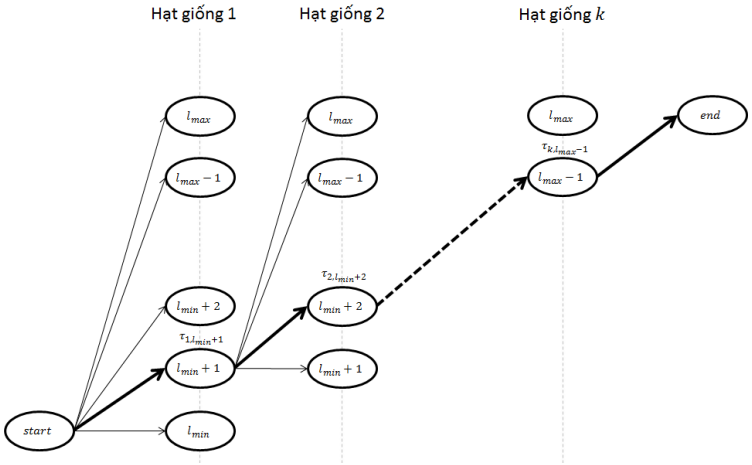
Với các tham số w, k, p, N đã cho và số vòng lặp N_c hoặc thời gian chạy xác định trước, thuật toán AcoSeeD xác định tập hạt giống tối ưu cho trường hợp chưa biết độ dài được mô tả trong hình 5.2. Trong trường hợp độ dài các hạt giống đã xác định thì thủ tục xác định độ dài các hạt giống được bỏ qua.

Procedure AcoSeeD;
Dữ liệu vào: w, k, p, N, độ dài các hạt giống nếu đã biết.
Kết quả ra: tập hạt giống và độ nhạy;
Begin
  Khởi tạo tập A gồm N_a kiến, ma trận mùi, các tham số $\rho, \tau_{max}, \tau_{min}$
  while (chưa kết thúc) **do**
    for i =1 **to** N_a **do**
      Kiến thứ i **x**ác định độ dài các hạt giống;
      Kiến thứ i xây dựng tập hạt giống;
      Cải tiến lời giải bằng tìm kiến cục bộ nhờ hàm mục tiêu OC;
      Tính độ nhạy của tập hạt giống do kiến i xây dựng;
    end-for
    Cập nhật mùi dựa trên lời giải có độ nhạy lớn nhất tìm được;
    Cập nhật lời giải tốt nhất;
  end-while
  Đưa ra lời giải tốt nhất;
End;

Hình 5.1: Thuật toán AcoSeeD

5.2.2. Thuật toán xác định độ dài các hạt giống

Trong trường hợp, độ dài các hạt giống chưa biết nhưng thuộc khoảng $[l_{min}, l_{max}]$ đã cho thì kiến phải thực hiện thủ tục xác định độ dài từng hạt giống trong tập nhờ đồ thị cấu trúc được mô tả trong hình 5.2. Ngoài hai đỉnh $start$ và end, đồ thị gồm k cột xếp từ phải sang trái, mỗi cột có $l_{max} - l_{min} + 1$ nút được gán nhãn từ l_{min} đến l_{max} biểu thị cho độ dài của hạt giống có thứ tự của cột tương ứng. Như vậy, các nút này được xếp thành $(l_{max} - l_{min} + 1)$ hàng và k cột. Ta xếp các hạt giống theo thứ tự tăng dần của độ dài, khi đó một đường xuất phát từ đỉnh $start$, đi qua k cột (chỉ sang ngang hoặc lên đỉnh trên ở cột tiếp theo) và kết thúc ở đỉnh end sẽ cho một phương án xác định độ dài của tập giống.



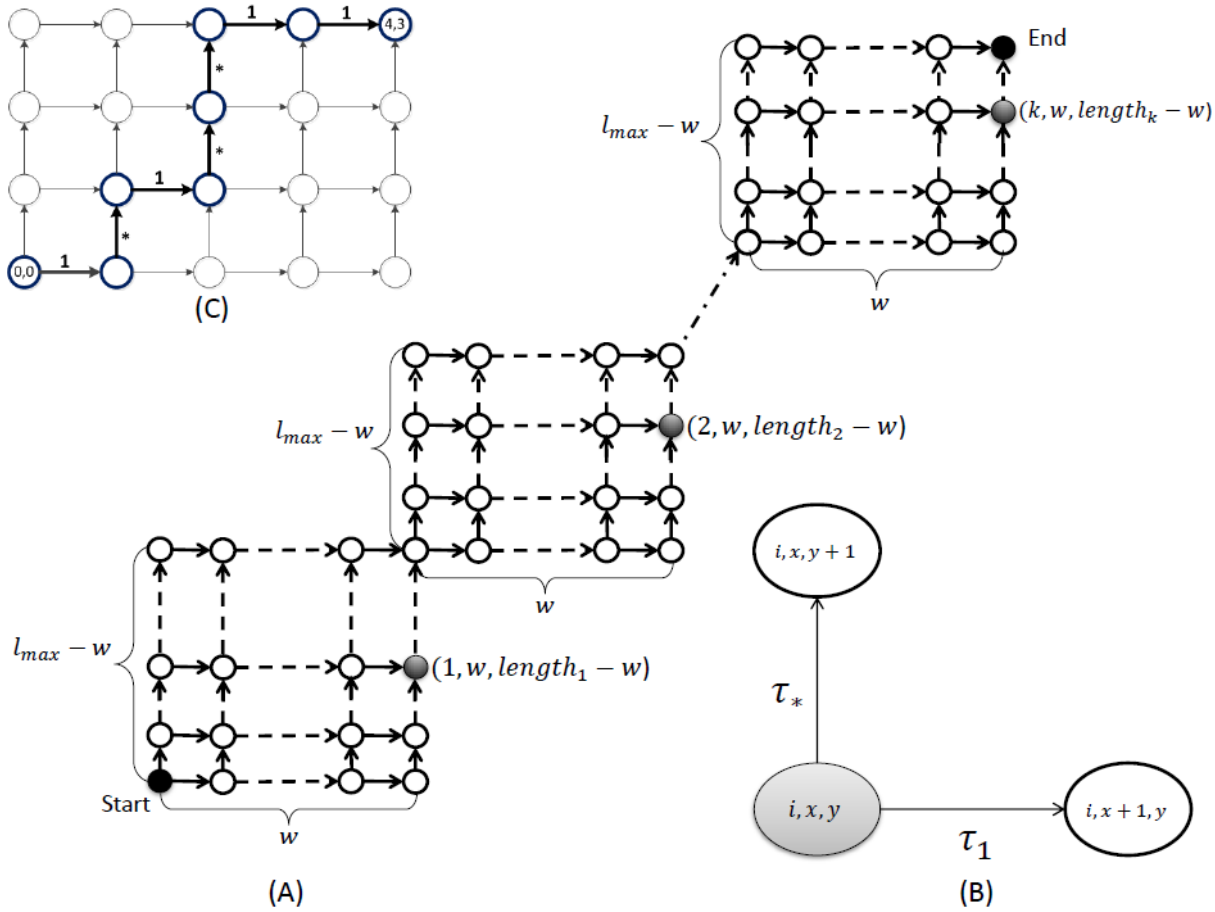
Hình 5.3: Đồ thị cấu trúc để xác định độ dài các hạt giống

5.2.3. Thuật toán xây dựng các hạt giống

Đồ thị cấu trúc để xây dựng lời giải tìm tập giống được mô tả trong hình 5.3.A, gồm k hình chữ nhật kích thước $w \times (l_{max} - w)$. Kiến xây dựng lần lượt k hạt giống bằng cách xuất phát từ đỉnh start (đỉnh trái dưới của hình chữ nhật thứ nhất) có toạ độ (1,0,0), trong đó chỉ số thứ nhất là chỉ số thứ tự hình chữ nhật, chỉ số thứ hai là chỉ số cột trên hình chữ nhật (đánh số từ trái qua phải), chỉ số thứ ba là chỉ số hàng trên hình chữ nhật (đánh số từ dưới lên trên) lần lượt di chuyển qua phải hoặc lên trên đến đỉnh End có toạ độ $(k, w, l_{max} - w)$ (đỉnh phải trên của hình chữ nhật thứ thứ k).

5.2.4. Cập nhật mùi

Sau khi tất cả các kiến xây dựng xong lời giải và các lời giải được áp dụng kỹ thuật tìm kiếm cục bộ sử dụng hàm mục tiêu OC thì lời giải có độ nhạy lớn nhất sẽ được dùng để cập nhật mùi cho cả hai giai đoạn xác định độ dài các hạt giống và giai đoạn xây dựng các hạt giống. AcoSeeD sử dụng cách cập nhật mùi mới SMMAS.



Hình 5.3: Đồ thị cấu trúc xây dựng các hạt giống.

Hình (A) Đồ thị cấu trúc xây dựng $\boldsymbol{k}$ hạt giống có trọng số $\boldsymbol{w}$. Hình (B) Hướng kiến di chuyển tại mỗi đỉnh. (C) Ví dụ xây dựng hạt giống trọng số 4 và độ dài 7.

5.3. Kết quả thực nghiệm

Hiệu quả của AcoSeeD được so sánh bằng thực nghiệm với hai phương pháp tốt nhất hiện nay là SpEED và SpEEDfast. Để khách quan với SpEED và SpEEDfast, AcoSeeD chạy trên các bộ dữ liệu và cùng số lời giải như Ilie đã làm. Kết quả thực nghiệm cho thấy AcoSeeD tốt hơn SpEED, SpEEDfast và AcoSeeD đã tìm được các tập hạt giống mới có độ nhạy cao hơn SpEEDfast tìm được.

Chương 6. Ứng dụng phương pháp ACO cải tiến hiệu quả dự đoán hoạt động điều tiết gen

6.1. Bài toán dự đoán hoạt động điều tiết gen

Hiểu cơ chế điều chỉnh biểu hiện gen qua các yếu tố phiên mã (*Transcription Factors-TFs*) là nhiệm vụ trung tâm của sinh học phân tử. Người ta biết rằng các trạng thái biểu hiện gen được thành lập thông qua sự tích hợp của mạng tín hiệu và phiên mã hội tụ trên các thành phần tăng cường, còn được gọi là mô-đun điều tiết (*Cis-Regulatory Module – CRM*). Các mô-đun điều tiết này là các đoạn DNA, nó

liên kết các yếu tố phiên mã để điều tiết biểu diễn gen liên quan. Mỗi mô-đun có thể điều tiết một hoặc nhiều gen. Gần đây, Zinzen và các cộng sự đã giới thiệu một mô hình dự báo điều tiết trên ruồi dấm Drosophila.

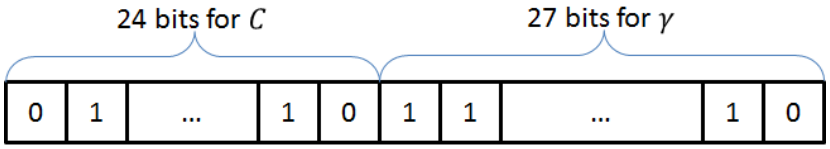
Ruồi giấm Drosophila là một mẫu sinh vật được dùng để nghiên cứu sự phát triển của phôi thai trong sinh học. Zinzen và các cộng sự đề xuất sử dụng phương pháp ChIP (*Chromatin Immunoprecipitation*) để thu được dữ liệu của yếu tố phiên mã quan trọng của ruồi giấm Drosophila (Twist, TinMan, Mef2, Bagpipe và Biniou) tại 5 thời điểm trong quá trình phát triển phôi.

Bài toán dự đoán được đưa về bài toán học có giám sát với đối tượng có 15 đặc trưng và nhận giá trị trong tập nhãn gồm 5 giá trị.

6.2. Thuật toán di truyền và tối ưu đàn kiến tìm tham số cho SVM dùng trong dự đoán hoạt động điều tiết gen

Zinzen đã sử dụng phương pháp tìm kiếm trên lưới để xác định tham hai tham số phát C và γ của hàm nhân dạng Gauss $e^{-\gamma\|u-v\|^2}$ trong phương pháp SVM (*Support Vector Machine - SVM*) để áp dụng cho bài toán dự đoán điều tiết này.

Luận án dùng mã nhị phân 51 bit để biểu diễn hai tham số C và γ. Tham số C nhận giá trị từ 10^{-2} đến 10^{5} được biểu diễn bằng một dãy 24 bit, và γ nhận giá trị 10^{-6} đến 10^{2} được biểu diễn bằng một dãy 27 bit.



Hình 6.3: Một nhiễm sắc thể biểu diễn C và γ

Dựa trên cách mã hóa này, luận án xây dựng áp dụng phương pháp ACO và thuật toán di truyền cổ điển cho xác định hai tham số này và thu được tương ứng hai hệ dự đoán ACOSVM và GASVM. Thực nghiệm cho thấy hai hệ này có hiệu quả hơn phương pháp của Zinzen và ACO tốt hơn so với GA.

Kết luận

Các bài toán TƯTH khó có nhiều ứng dụng quan trọng trong thực tiễn, đặc biệt là trong các bài toán sinh học. Phương pháp ACO kết hợp thông tin heuristic và thông tin học tăng cường nhờ mô phỏng hoạt động của đàn kiến có các ưu điểm nổi trội sau:

1) Việc tìm kiếm ngẫu nhiên dựa trên các thông tin heuristic cho phép tìm kiếm linh hoạt và mềm dẻo trên miền rộng hơn phương pháp heuristic sẵn có, do đó cho ta lời giải tốt hơn và có thể tìm được lời giải tối ưu.

2) Sự kết hợp học tăng cường thông qua thông tin về cường độ vết mùi cho phép ta từng bước thu hẹp không gian tìm kiếm mà vẫn không loại bỏ các lời giải tốt, do đó nâng cao chất lượng thuật toán.

Thực nghiệm đã chứng tỏ khả năng nổi trội của phương pháp ACO trong ứng dụng cho nhiều bài toán và phương pháp này đang được sử dụng rộng rãi.

Khi dùng phương pháp ACO, quy tắc cập nhật mùi đóng vai trò quan trọng, quyết định hiệu quả thuật toán được dùng. Luận án đề xuất các quy tắc cập nhật mùi mới: SMMAS, MLAS và 3-LAS. Các thuật toán này bất biến đối với phép biến đổi đơn điệu hàm mục tiêu, thực nghiệm trên các bài toán cơ bản như TSP, UBQP, lập lịch sản xuất với dữ liệu chuẩn cho thấy các thuật toán đề xuất có hiệu quả và dễ sử dụng hơn so với các thuật toán thông dụng nhất hiện nay như ACS và MMAS.

Trong các thuật toán này, *SMMAS đơn giản, dễ sử dụng hơn nên có thể dùng rộng rãi*. Thuật toán MLAS cho phép điều tiết linh hoạt khả năng khám phá và tăng cường của thuật toán theo từng thời điểm. Tuy thực nghiệm trên bài toán TSP cho kết quả hứa hẹn nhưng khó áp dụng hơn. Thuật toán 3-LAS thích hợp với các bài toán có thông tin heuristic tốt, khi sử dụng chúng ảnh hưởng nhiều tới chất lượng của kết quả tìm kiếm, chẳng hạn như bài toán TSP.

Bên cạnh phát triển thuật toán mới, luận án cũng đề xuất các giải pháp cho ba bài toán quan trọng trong sinh học phân tử: suy diễn haplotype, tìm tập hạt giống tối ưu và dự báo hoạt động điều tiết gen.

Đối với bài toán suy diễn haplotype, luận án đề xuất thuật toán ACOHAP. Kết quả thực nghiệm cho thấy ACOHAP cho kết quả tối ưu như RPoly (phương pháp chính xác tốt nhất hiện nay) trong nhiều trường hợp, hơn nữa, ACOHAP hiệu quả nổi trội hơn hẳn CollHap (phương pháp xấp xỉ tốt nhất hiện nay).

Đối với bài toán tìm tập hạt giống tối ưu, luận án đề xuất thuật toán AcoSeeD. Kết quả thực nghiệm cho thấy AcoSeeD cho kết quả tốt hơn hai phương pháp tốt nhất hiện nay là SpEED và SpEEDfast.

Đối với bài toán dự báo hoạt động điều tiết gen, dựa trên phương pháp đề xuất của Zinzen và các cộng sự, luận án đề xuất hai thuật toán metaheuristic: GASVM và ACOSVM. Các thuật toán này tương ứng sử dụng phương pháp GA hoặc ACO để tìm tham số tốt nhất cho bộ học SVM. Thực nghiệm cho thấy hiệu quả hơn cách tiếp cận áp dụng phương pháp tìm kiếm trên lưới của Zinzen.

Hiện tại hệ ACOHAP, AcoSeeD, GASVM và ACOSVM sẽ có ích cho các nhà nghiên cứu sinh học và những người quan tâm.

Trong tương lai, chúng tôi sẽ cùng với nhóm nghiên cứu Tin-Sinh của Đại học Công nghệ ứng dụng các đề xuất mới này cho các bài toán khác.

DANH MỤC CÁC CÔNG TRÌNH KHOA HỌC CỦA TÁC GIẢ LIÊN QUAN ĐẾN LUẬN ÁN

1) Huy Q. Dinh, Dong Do Duc, and Huan X. Hoang (2006), "Multi-Level Ant System - A new approach through the new pheromone update for Ant Colony Optimization", *Proc. of the 4th IEEE International Conference in Computer Sciences, Research, Innovation, and Vision for Future*, pp. 55-58.
2) D. Do Duc, Huy.Q. Dinh, and H. Hoang Xuan (2008), "On the pheromone update rules of ant colony optimization approaches for the job shop scheduling problem," *Proc. of the Pacific Rim Int. Workshop on Multi-Agents*, 2008, pp. 153-160.
3) Hoàng Xuân Huấn và Đỗ Đức Đông (2010), "Về vết mùi trong các thuật toán ACO và khung cảnh mới", *Kỷ yếu hội thảo quốc gia các vấn đề chọn lọc của CNTT lần thứ XII*, tr. 534-547.
4) Dong Do Duc, Huan Hoang Xuan (2010), "Smoothed and Three-Level Ant Systems: Novel ACO Algorithms for the Traveling Salesman Problem", *Ad. Cont. to the IEEE RIFV2010*, pp. 37-39.
5) Đỗ Đức Đông và Hoàng Xuân Huấn (2011), "Về biến thiên của vết mùi trong phương pháp ACO và các thuật toán mới", *Tạp chí Tin học và điều khiển học*, Tập 27, tr. 263-275.
6) Dong Do Duc and Hoang Xuan Huan (2011), "ACOHAP: A novel Ant Colony Optimization algorithm for haplotype inference problem", *Proc. of the Third International Conference on Knowledge and Systems Engineering*, pp. 128-134.
7) Dong Do Duc, Tri-Thanh Le, Trung Nghia Vu, Huy Q. Dinh, Hoang Xuan Huan (2012), "GA_SVM: A genetic algorithm for improving gene regulatory activity prediction", *Proc. of the 9th IEEE-RIVF International Conference on Computing and Communication Technologies*, pp. 234-237.
8) Dong Do Duc, Huan Hoang Xuan, and Huy Q. Dinh (2012), "META-REG: A computational meteheuristic method to improve the regulatory activity prediction", *Proc. of the 4th International Conference on the Development of Biomedical Engineering*, pp. 450-453.
9) Dong Do Duc, H. Q. Dinh, T.H. Dang, K. Laukens, and H. Hoang Xuan (2012), "AcoSeeD: an Ant Colony Optimization for finding optimal spaced seeds in biological sequence search", *Proc. Of the ANTS2012: Eighth Int. Conf. on swarm intelligence*, pp. 204-211.