ĐẠI HỌC QUỐC GIA HÀ NỘI
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ

TRẦN NGỌC HÀ

CÁC BÀI TOÁN TỐI ƯU TỔ HỢP VÀ TÍNH TOÁN MỀM

Chuyên ngành:Khoa học máy tính
Mã số:62.48.01.01

TÓM TẮT LUẬN ÁN
TIẾN SĨ CÔNG NGHỆ THÔNG TIN

NGƯỜI HƯỚNG DẪN KHOA HỌC:
PGS. TS.Hoàng Xuân Huấn
GS. TS.Thái Trà My

HÀ NỘI – 2017

Công trình được hoàn thành tại:

Trường Đại học Công nghệ, Đại học Quốc gia Hà Nội

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

GS.TS. Thái Trà My

Phản biện: .....
.....
Phản biện: .....
.....
Phản biện: .....
.....

Luận án sẽ được bảo vệ trước Hội đồng cấp Đại học Quốc gia chấm luận án tiến sĩ họp tại .....
vào hồigiờ ngàythángnăm

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

Các phương pháp tối ưu tổ hợp (TƯTH) đã được nghiên cứu rất sớm, từ thời Euler (thế kỷ 18), ngày nay, cùng với sự phát triển nhanh chóng của công nghệ thông tin, chúngđang được nhiều người quan tâm nghiên cứuvà ứng dụng rộng rãi trong các bài toán thực tế đặc biệt là trong tin-sinh học. Trong đó, chúng ta ngày càng gặp nhiều bài toán ưu tổ hợp TƯTH) thuộc loạiNP-khócỡ (size) lớn.

Trong tiếp cận truyền thống, các bài toán và thuật toán giải phải tuân thủ nhiều điều kiện toán học khắt khe:

- Bài toán phải được thiết lập đúng đắn (tồn tại duy nhất nghiệm và ổn định với điều kiện ban đầu) hoặc đã được chính quy hóa để trở nên đúng đắn, nếu có yếu tố không chắc chắn thì cần được xử lý dựa trên lý thuyết xác suất và thống kê.
- Các thuật toán giải phải chứng minh được tính hội tụ hoặc ước lượng được sai số/ tỷ lệ tối ưu, với các bài toán cỡ (size) lớn thì thuật toán phải có thời gian đa thức.

Vì có các đòi hỏi như vậy nên những thuật toán được đề xuất không đủ để đáp ứng nhu cầu ngày càng tăng trong ứng dụng. Các phương pháp tính toán mềm giải quyếtcác bài toán phức tạptheo tiếp cận mềm dẻo hơn. Kết quả thực nghiệm cho thấy hiệu quả tốt của các tiếp cận này nên chúng đang thu hút nhiều người nghiên cứu, ứng dụng.

Trong tiếp cận tính toán mềm, các thuật toán heuristics và metaheuristicsthường được đề xuất áp dụng cho cácbài toán TƯTHkhó cỡ lớn. Trong đó hiệu quả của các thuật toán được đánh giá bằng thực nghiệm và ý tưởng đề xuất. Các thuật toán heuristics cho phép tìm kiếm nhanh (thường theo kiểu tham lam) lời giải đủ tốt và thường hướng tới cực trị địa phương. Các thuật toán metaheuristics thường có thời gian chạy lâu hơn các thuật toán heuristics nhưng hướng tới cực trị toàn cục, thời gian chạy càng lâu thì lời giải tìm được càng tốt hơn.

Đa số các phương pháp metaheuristics dựa trên ý tưởng mô phỏng tự nhiênvới ngầm định rằng các quá trình phát triển tự nhiên thường mang tính tối ưu. Trong đó, cácthuật toán di truyền (GA), tối ưu đàn kiến (ACO), memetic đang được sử dụng rộng rãi cho các bài toán TƯTH khó. Đặc biệt, phương pháp ACO do Dorigo đề xuấtrất thích hợp cho các bài toán tối ưu tổ hợp trên đồ thị.

GA là phương pháp metaheuristics được đề xuất sớm và thông dụng nhất. Tuy nhiên, ở mỗi bước lặp của các thuật toán GA phải dùng lại nhiều lời giải của bước lặp trước đó nên thường kém hiệu quả hơn các thuật toán ACO. Trong phương pháp ACO, bài toán nguyên thủy đươc đưa thành bài toán tìm đường đi tối ưu trên đồ thị cấu trúc bằng thủ tục bước ngẫu nghiên dựa trên thông tin heuristics và thông tin học tăng cường. Bốn yếu tố ảnh hưởng nhiều đến chất lượng của thuật toán ACO là:

1) Quy tắc cập nhật mùi
2) Đồ thị cấu trúc
3) Thông tin heuristics
4) kỹ thuật tìm kiếm địa phương.

Ba yếu tố sau được xây dựng và xác định tùy theo từng bài toán cụ thể, chất lượng của chúng được xác định nhờ thực nghiệm. Các quy tắc cập nhật mùi có tính phổ dụng nhưng các tham số thích hợp phải được xác định bằng thực nghiệm. Khi áp dụng kỹ thuật tìm kiếm cục bộ cho các

thuật toán ACO theo lược đồ memeticta có các thuật toán ant-based.

Những phát hiện về cơ chế di truyền trong cơ thể sống đã thúc đẩy sinh học phân tử nói riêng và công nghệ sinh học nói chung phát triển mạnh mẽ trong nửa thế kỷ qua vàtrở nên lĩnh vực nghiên cứu và ứng dụng hấp dẫn. Tuy nhiên các nghiên cứu trong phòng thí nghiệm đòi hỏi nhiều thời gian và tốn kém. Cùng với sự phát triển của công nghệ thông tin, tin-sinh họcra đời và là công cụ trợ giúp hiệu quả cho các nghiên cứu sinh-y-dược.

Việc nghiên cứu tính tương đồng/khác biệt cấu trúc tuần tự là không đủ để phát hiện tính tương đồng/khác biệt về chức năng trong cơ thể sống. Nghiên cứu các mạng sinh học như mạng tương tác protein-protein (PPI), mạng điều hòa gen (gene regulatory), mạng các vị trí liên kết protein,mạng trao đổi chất…mang lại tiếp cận nghiên cứu hiệu quả hơn về phân tích chức năng trong sinh học phân tử. Đặc biệt, việc dóng hàng các mạng tương tác protein-protein và mạng các vị trí liến kết protein cho phép chúng ta dự đoán đặc điểm chức năng ởcác loài chưa nghiên cứu kỹ từcác tri thức của các loài đã biết, nhờ đó hiểu rõ hơn quan hệ tiến hóa sinh học, hỗ trợ thông tin để nghiên cứu thuốc điều trị các bệnh di truyền. Các bài toán này thuộc loại NP-khó và đang thu hút nhiều người nghiên cứu/ứng dụng do tính quan trọng của chúng.

Trong bối cảnh đó, chúng tôi chọn chủ đề nghiên cứu "***Các bài toán tối ưu tổ hợp và tính toán mềm***" với nội dung là nghiên cứu áp dụng các kỹ thuật TƯTH mềm để đề xuất một số thuật toán thông minh giải haibài toán *dóng hàng toàn cục mạng tương tác protein-protein* và *dóng hàng nhiềumạngvị trí liên kết protein* (sẽ gọi gọn là *bài toán dóng hàng nhiều đồ thị*) với chất lượng lời giải và thời gian tính toán tốt hơn so với các thuật toán mới nhất hiện nay.

2. Mục tiêu của luận án

Tìm hiểu các dạng bài toán dóng hàng các mạng protein nêu trên và các thuật toán giải chúng đã được đề xuất trong thời gian gần đây.

Tìm hiểu các kỹ thuật tính toán mềm để từ đó thấy rõ ưu và nhược điểm của từng phương pháp. Trên cơ sở đó, đề xuất các thuật toán mới với chất lượng lời giải tốt hơn các thuật toán hiện tại trong thời gian ngắn hơn cho các bài toán này.

3. Các đóng góp của luận án

Trong thời gian qua, cùng với cán bộ hướng dẫn vàcác cộng sự, tác giả luận án đã có đóng góp sau.

- Đề xuất ba thuật toán cho bài toán dóng hàng toàn cục mạng tương tác protein-Protein, bao gồm thuật toán heuristics FASTAN và hai thuật toán tối ưu đàn kiến: ACOGNA và ACOGNA++.
- Đề xuất ba thuật toán dựa trên tối ưu đàn kiến cho bài toán dóng hàng nhiều đồ thị, bao gồm ACO-MGA, ACO-MGA2 và ACOTS-MGA

Kết quả thực nghiệm cho thấy hiệu quả nổi trội củacác thuật toán đề xuất so với các thuật toán tiên tiến hiện có. Các kết quả của luận án đã được công bố trong 5 báo cáo hội nghị/hội thảo quốc gia/quốc tế bao gồm 4 báo cáo hội nghị quốc tế (Công trình 1,2,3,5) vàmột hội thảo toàn quốc "*Nghiên cứu cơ bản và ứng dụng công nghệ thông tin*" (Công trình 4), ngoài ra có một bài báo đang gửi đăng tạp chí.

4. Bố cục của luận án

Ngoài phần mở đầu và kết luận, luận án được tổ chức như sau:

Chương 1 giới thiệu bài toán tối ưu tổ hợp dạng tổng quát và các phương pháp metaheuristic bao gồm giải thuật di truyền và tính toán tiến hóa, các thuật toán memetic và phương

pháp tối ưu đàn kiến.

Chương 2 giới thiệu hai bài toán dóng hàng mạng tương tác protein-protein và dóng hàng nhiều đồ thị cùng một số vấn đề liên quan

Chương 3 trình bày ba thuật toán đề xuất để giải bài toán dóng hàng toàn cục 2 mạng tương tác protein-protein. Hiệu quả của các thuật toán được kiểm nghiệm trên các bộ dữ liệu chuẩn (IsoBase) được sử dụng bởi các thuật toán mới nhất hiện nay. Các thực nghiệm đã cho thấy hiệu quả nổi trội của các thuật toán đề xuất.

Chương 4 trình bày ba thuật toán dựa trên phương pháp tối ưu đàn kiến để giải bài toán dóng hàng nhiều mạng các vị trí liên kết của protein. Các kết quả thực nghiệm trên các bộ dữ liệu mô phỏng và dữ liệu thực cho thấy các thuật toán đề xuất tốt hơn hẳn so với các thuật toán mới nhất để giải bài toán dóng hàng nhiều đồ thị.

Chương 1. TỐI ƯU TỔ HỢP VÀ TÍNH TOÁN MỀM

Chương này phát biểu bài toán TƯTH tổng quát và các vấn đề liên quan, sau đó giới thiệu ngắn gọn các phương pháp tối ưu theo tiếp cận tính toán mềm, bao gồm GA, tính toán tiến hóa, các thuật toán memetic và phương pháp ACO.

1.1. Bài toán tối ưu tổ hợp

1.1.1. Phát biểu bài toán tổng quát

Một cách tổng quát, mỗi bài toán TƯTH có thể phát biểu như sau: Cho 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. Chẳng hạn với bài toán cực tiểu thì $f(s^*) \leq f(s)$ với mọi phương án chấp nhận được s.

1.1.2. Các ví dụ

Trong đời sống và trong các hệ thông tin, ta thường gặp nhiều bài toán tối ưu tổ hợp quan trọng. Chẳng hạn như: tìm đường đi ngắn nhất nối hai điểm trên một đồ thị đã cho, lập kế hoạch phân phối nguồn hàng tới nơi tiêu thụ với chi phí cực tiểu, lập thời khóa biểu cho giáo viên và học sinh thuận lợi nhất, định tuyến cho các gói dữ liệu trong Internet hay các bài toán trong lĩnh vực tin sinh học…

1.1.3. Các cách tiếp cận giải bài toán tối ưu tổ hợp

Với các bài toán TƯTHNP-khó có cỡ nhỏ, người ta có thể tìm lời giải tối ưu nhờ tìm kiếm vét cạn. Tuy nhiên, với các bài toán cỡ lớn thì đến nay chưa thể có thuật toán tìm lời giải đúng với thời gian đa thức nên chỉ có thể tìm lời giải gần đúng hay đủ tốt.

Theo cách tiếp cận truyền thống hay là tiếp cận cứng, các thuật toán gần đúng phải được chứng minh tính hội tụ hoặc ước lượng được tỷ lệ tối ưu. Với việc đòi hỏi khắt khe về toán học như vậylàm hạn chế số lượng các thuật toán công bố, khôngđáp ứng được nhu cầu ngày càng phong phú và đa dạng trong nghiên cứu và ứng dụng. Để khắc phục tình trạng này, người ta dùng tiếp cận *đủ tốt*để xây dựng các thuật toán *tối ưu mềm*.

1.2. Tính toán mềm

Tính toán mềmcho một cách tiếp cận để giải quyết các bài toán khó, thông tin không đầy đủ, thiếu chắc chắn và cho kết quả là những lời giải đủ tốt hoặc gần đúng mà tiếp cận truyền thông hay

tính toán cứng (hard computing) không giải quyết được. Tiếp cận này gồm các phương pháp sử dụng tập mờ/ tập thô, các phương pháp học máy như mạng nơ ron nhân tạo, máy tựa vector (SVM), các giải thuật tiến hóa như các giải thuật di truyền tối ưu bầy đàn, tối ưu đàn kiến, tối ưu bầy ong, giải thuật memetic, hệ miễn dịch nhân tạo…

Đối với các bài toán TƯTH khó, các phương pháp tính toán mềm được đánh giá chất lượng *dựa trên thực nghiệm* mà không nhất thiết phải chứng minh tính hội tụ hoặc ước lượng tỷ lệ tối ưu. Các thuật toán thường được xây dựng dựa trên một ý tưởng "*có lý*" và hiệu quả của chúng được đánh giá dựa vào kết quả thử nghiệm trên tập dữ liệu *đủ tin cậy*.

1.2.1. Các thuật toán dựa trên thực nghiệm.

Các phương pháp này phát triển theo hai hướng *heuristic* và *metaheuristics.* Các thuật toán *heuristic* đề xuất riêng biệt cho từng bài toán cụ thể, cho phép tìm nhanh một lời giải đủ tốt hoặc xấp xỉ tối ưu địa phương

Theo cách hiểu chung nhất, mỗi thuật toán *metaheuristics* tổng quát là một lược đồ tính toán đề xuất cho lớp bài toán rộng, khi dùng cho các bài toán cụ thể cần thêm các vận dụng chi tiết cho phù hợp. Nhờ các lược đồ này, người dùng có thể xây dựng được thuật toán cho bài toán trong thực tế mà không đòi hỏi có kiến thức tốt về toán học tính toán, vì vậy, hiện nay chúngđang được dùng phổ biến trong ứng dụng. Các thuật toán này thường có thời gian chạy lâu hơn các thuật toán truyền thống và tìm kiếm địa phương nhưng lời giải hướng tới tối ưu toàn cục.

1.2.2. Giải thuật di truyền

GA được J. H. Holland ở trường đại học Michigan giới thiệuđầu tiên vào năm 1975, là kỹ thuật mô phỏng quá trình tiến hoá tự thích nghi của các quần thể sinh học dựa trên học thuyết Darwin để tìm gần đúng lời giải tối ưu toàn cục.

Sau J.H. Holland,có rất nhiều người nghiên cứu về lý thuyết cũng như những ứng dụng của GA trong cáclĩnh vực khác nhau như sinh học, khoa học máy tính, kỹ thuật lai ghép, xử lý ảnh… và đang là thuật toán metaheuristics thông dụng nhất.

1.2.2. Tính toán tiến hóa và các thuật toán Memetic

Thuật ngữ tính toán tiến hóa ban đầu để chỉ các phương pháp tìm lời giải nhờ đưa về sử dụng GA. Ngày nay nó dùng để chỉ chung cho các phương pháp tối ưu dựa trên quần thể, trong đó quần thể của thế hệ sau được xây dựng dựa trên thông tintừ quần thể trước để tìm lời giải. Các thuật toán này thường được xây dựng dựa trên các lược đồ metaheuristics, chẳng hạn như các thuật toán tối ưu bầy đàn(Particle swarm optimization: PSO),đom đóm (Firefly algorithm), dơi (Bat algoritm)….

Memetic là cáckỹ thuật tìm kiếm dựa trên quần thể, ban đầu áp dụng cho giải thuật di truyền và nay ứng dụng hiệu quả cho các thuật toán khác.

Trong các thuật toán memetic,chẳng hạn GA hoặc ACO, cuối mỗi vòng lặp t, người ta tìm ra tập lời giải Ω(t) và tập thuật toán tìm kiếm địa phương $\mathcal{A}(t)$để áp dụng các thuật toán tìm kiếm tăng cường một cách linh hoạt phù hợp với đặc điểm từng bài toán. Kết quả thực nghiệm cho thấy việc áp dụng tìm kiếm địa phương đa dạng và linh hoạt ở mỗi bước lặp tùy theo các ràng buộc và đặc điểm hàm mục tiêu cải thiện đáng kể chất lượng thuật toán so với các thuật toán sử dụng đơn điệu một thuật toán tìm kiếm cho mọi bước lặp.

1.3. Phương pháp tối ưu đàn kiến

Phương pháp tối ưu đàn kiến (ACO) là thuật toán mô phỏng cách tìm đường đi tới tổ của kiến tự nhiên để giải các bài toán TƯTH khó. Phương pháp này được Dorigo giới thiệu vào năm 1991[6]

dưới dạng hệ kiến(Ant System) ngày nay đã được phát triển dưới nhiều biến thể và được ứng dụng rộng rãi

1.3.1. Kiến tự nhiên và kiến nhân tạo

Kiến tự nhiên.

Trên đường đi đến nguồn thức ăn và trở về tổ, mỗi con kiến thực để lại một vết hoá chất gọi là vết mùi (pheromone trail) và theo vết mùi của các con kiến khác để tìm đường đi. Đường có nồng độ vết mùi càng cao thì càng có nhiều khả năng được các con kiến chọn. Nhờ cách giao tiếp gián tiếp này đàn kiến tìm được đường đi ngắn nhất từ tổ tới nguồn thức ăn.

Việc tìm đường đi của các con kiến tự nhiên dựa trên nồng độ vết mùi làm taliên tưởng tới cách học tăng cường cho bài toán chọn tác động tối ưu, gợi mở một mô hình mô phỏng cho các con kiến thực để tìm đường đingắn nhất giữa hai nút (tương ứng là tổ và nguồn thức ăn) trên đồ thị. Trên cơ sở đó, mở rộng thành phương pháp ACO để giải các bài toán tối ưu tổ hợp khó

Kiến nhân tạo.

Khi mô phỏng hành vi của đàn kiến để giải các bài toán thực, người ta dùng đa tác tử (multiagent) làmđàn kiến nhân tạo, trong đó mỗi con kiến nhân tạo là một tác tử, có nhiều khả năng hơn kiến tự nhiên. Kiến nhân tạo (về sau sẽ gọi là kiến) có bộ nhớ riêng, có khả năng mở rộng, chẳng hạn,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 khả năng mở rộng mà mỗi đàn kiến có thể thực hiện lặp quá trình tìm lời giải nhờ thủ tục bước tuần tự trên đồ thị cấu trúc tương ứng của mỗi bài toán và cập nhật mùi theo phương thức học tăng cường để tìm lời giải chấp nhận được và xác định lời giải đủ tốt toàn cục.

1.3.2. Lược đồ chung của phương pháp ACO

```
Procedure Thuật toán ACO
Begin
    Initialize: Khởi tạo vết mùi, n_ants
    while Khi điều kiện dừng chưa thỏa mãn do
        for i=1 to n_ants do
                Xây dựng lời giải;
                Cập nhật lời giải tốt;
        end for
Cập nhật mùi
end while
End
```

Hình 1. 1. Đặc tả thuật toán ACO tổng quát

1.3.3. Thủ tục bước ngẫu nhiên xây dựng lời giải

Giả sử kiến đã phát triển được xâu $\langle u_0, \dots, u_m \rangle$ trong đó $u_m = i$nhưngchưa cho lời giải chấp nhận được và nhờ Ω ta xác định được tập đỉnh $J_k(i)$có thể phát triển thì thành phần $\dots u_{i+1} = j$ tiếp theo được chọn với xác suất

$$p_{ij}^k = \begin{cases} \frac{[\tau_{ij}(t)]^\alpha . [\eta_{ij}(t)]^\beta}{\sum_{l \in J_k(i)} [\tau_{il}(t)]^\alpha . [\eta_{il}(t)]^\beta} & nếu\ j \in J_k(i) \\ 0 & nếu\ j \notin J_k(i) \end{cases} \qquad (2.1)$$

trong đó α, β là các hằng số dương chọn trước. Thủ tục này tiếp tục cho đến khi xâu $\langle u_0, \dots, u_t \rangle$ tương ứng một với lời giải *s* trong *S*. Bằng cách này mỗi kiến xây dựng được lời giải trong mỗi vòng lặp và cùng thực hiện đánh giá lời giải để cập nhật mùi theo một quy tắc được chọn.

1.3.4. Các quy tắc cập nhật mùi

Việc cập nhật mùi, phản ánh cơ chếhọc tăng cường và ảnh hưởng quyết định chất lượng thuật toán nên thường dùng để làm tên gọi cho lớp thuật toán dùng nó. Để đảm bảo vết mùi hội tụ, người ta sử dụng hằng số bay hơi vết mùi $0<\rho\leq 1$ hay hệ số chiết khấu trong học tăng cường, khi một cạnh được cập nhật mùi thì vết mùi biến đổi theo công thức:

$$\tau_{ij} \leftarrow (1-\rho)\tau_{ij} + \Delta\tau_{ij} \qquad (1.4)$$

Điểm then chốt là cạnh nào được cập nhật và lượng thêm vào thế nào là tùy theo quy tắc được chọn. Có nhiều quy tắc cập nhật mùi đã được đề xuất, trong đó điển hình là các quy tắc hệ kiến (AS), hệ đàn kiến (ACS), hệ kiến *Max-Min*(Max–Min Ant System: MMAS) và hệ kiến *Max-min trơn* (Smooth Max-Min Ant System: SMMAS).

Quy tắc SMMAS

Quy tắc SMMAS lần đầu tiên được Đỗ Đức Đông và cộng sự dùng cho bài toán lập lịch sản xuất [7] và được trình bày chặt chẽ cho bài toán TSP trong [8].

SMMAS được đề xuất dựa trên nhận xét hai nhược điểm của MMAS:

- Thứ nhất, nếu chọn τ_{min}, τ_{max} lệch nhau ít thì làm triệt tiêu hiệu quả học tăng cường, còn nếu chọn lệch nhau nhiều thì vết mùi ở các cạnh ít được cập nhật sẽ nhanh chóng về τ_{min} làm hạn chế không gian tìm kiếm mặc dù có nhẹ hơn AS và ACS.
- Thứ hai là đại lượng $\Delta\tau_{i,j}$ phụ thuộc giá trị hàm mục tiêu làm cho thuật toán phải tính toán phức tạp, tuy nhiên trong học tăng cường thì không cần thiết.

Một trong các cải tiến là khi khởi tạo lại vết mùi, việc cập nhật mùi thường xuyên bằng lời giải tốt nhất tìm được mới nhất thay vì cố định.

Với nhận xét trên, SMMAS không giảm vết mùi ở các cạnh không thuộc lời giải tốt quá nhanh như quy tắc MMAS mà dùng quy tắc Max-Min trơn bằng cách cập nhật$\tau_{i,j}$toàn cục cho mọi cạnh với $\Delta\tau_{i,j}$ xác định bởi:

$$\Delta\tau_{i,j} = \begin{cases} \rho\tau_{min} nếu\ (i,j) \notin w(t) \\ \rho\tau_{max} nếu\ (i,j) \in w(t) \end{cases} \qquad (1.8)$$

Trong đó *w(t)* là lời giải tốt nhất mà các kiến xây dựng được. Quy tắc này cũng khởi tạo $\tau_0 = \tau_{max}$. Đây là một phương pháp cập nhật mùi dễ cài đặt và có độ phức tạp tính toán cũng thấp hơn so với các phương pháp trước nó. Thực nghiệm và phân tích toán học cho thấy nó tốt hơn MMAS.

1.3.5. Tìm kiếm địa phương

Thông thường thì các kỹ thuật tìm kiếm địa phương hội tụ đến cực trị địa phương nhanh hơn. Vì vậy người ta thường áp dụng kỹ thuật tìm kiếm địa phương để tăng cường chất lượng lời giải cho lời giải tốt nhất hoặc cho mọi lời giải trong mỗi bước lặp trước khi cập nhật mùi. Các kỹ thuật tìm kiếm có thể áp dụng linh hoạt theo lược đồ memetic được nêu trong mục 1.2.3.

1.3.6. Các ví dụ

Ví dụ 1. ACO cho bài toán TSP

Với bài toán TSP cho bởi G(V, E) và độ dài d_{ij} của các cạnh (*i,j*) đã biết như trong mục 1.1.2, ta có thể dùng luôn đồ thị này làm đồ thị cấu trúc với C_0=C=V. Với mỗi kiến *k* ở đỉnh *i* khi tìm kiếm, tậpp $J_k(i)$ là các đỉnh mà kiến chưa đi qua.Thông tin heuristics là nghịch đảo khoảng cách $\eta_{ij} = \frac{1}{d_{ij}}$.

Việc tìm kiếm địa phương cho một chu trình Hamintol được áp dụng cho các chu trình có *p* đỉnh liền nhau trong chu trình này được hoán vị (*p*-láng giềng), trong đó *p* chọn trước. Chiến lược tìm kiếm có thể là *tốt hơn* (better) hoặc *tốt nhất* (the best). Trong chiến lược *tốt hơn*, việc tìm kiếm

cho mỗi lời giải ở mỗi lần lặp sẽ dừng khi tìm được một lời giải tốt hơn nó, còn chiến lược tốt nhất sẽ tìm kiếm lời giải tốt nhất trong p-láng giềng. Một cách áp dụng memetic là chỉ áp dụng tìm kiếm địa phương có một số lời giải tốt trong một số lần lặp sau thay vì ở những vòng lặp đầu, khi lời giải chưa đủ tốt thì có cải tiến thì cũng chưa tốt bao nhiêu mà tốn thời gian.

Ví dụ 2. ACO cho bài toán tìm DNA-motif

1.3.7. Nhận xét về phương pháp ACO

So với GA, ở mỗi bước lặp, ACO không dùng lại nhiều lời giải của bước lặp trược như GA, hơn nữa việc kết hợp học tăng cường và thông tin heuristics sẽ tăng hiệu quả tìm kiếm.

Việc tìm kiếm ngẫu nhiên cho phép tìm kiếm linh hoạt, mềm dẻo trên miền rộng hơn phương pháp heuristics sẵn có. Để tăng cường khả năng khám phá, ACO có thể áp dụng khởi tạo lại vết mùi sau một số bước lặp mà không tìm được lời giải tốt hơn.

Thuật toán ACO dễ song song hóa để giảm thời gian chạy trên máy song song do mỗi con kiến tìm lời giải một cách độc lập trong mỗi vòng lặp.

Với những lý dó trên, luận án tập trung vào phát triển các thuật toán dựa trên đàn kiến.

Chương 2. TIN SINH HỌC VÀ DÓNG HÀNG CÁC MẠNG PROTEIN

Trong chương này, sau khi giới thiệu ngắn gọn bức tranh chung của tin sinh học sẽ giới thiệu bài toán dóng hàng mạng tương tác protein-protein và bài toán dóng hàng nhiều mạng vị trí liên kết protein được nghiên cứu trong các chương sau.

2.1. Giới thiệu về tin sinh học

2.1.1. Quá trình tổng hợp protein

2.1.2. Sinh học phân tử và phân tích các trình tự trong tin sinh học

2.1.3 Các mạng sinh học

Dóng hàng các chuỗi thuộc hệ gen đã tăng cường kiến thức y sinh học của nhờ phát hiện các vùng trình tự có sự tương đồng giữa các gen ở các loài khác nhau, các vùng đó có khả năng phản ánh các mối quan hệ chức năng và tiến hóa giữa các trình tự. Tuy nhiên, các gen hoặc các sản phẩm protein của chúng không hoạt động một cách độc lập mà chúng thực hiện các quá trình tế bào bằng cách tương tác với nhau.Các tương tác này được mô hình hóa bởi mạng sinh học, chẳng hạn như: mạng điều hòa gen (gene regulatory),mạng trao đổi chất, mạng tương tác protein-protein (protein-protein interactive network: PPI), mạng các vị trí liên kết/hoạt tính protein. Không giống như các nghiên cứu về các chuỗi gen, nghiên cứu mạng sinh học cho phép hiểu được các quá trình tế bào phức tạp phát sinh từ các hoạt động chung của các phân tử sinh học.

Những tiến bộ trong công nghệ sinh học hiện thời cung cấp nhiều dữ liệu cho phép ta nghiên cứu sâu hơn về các mạng sinh học và cho ta nhiều tri thức quý giá. Chẳng hạn, việc dóng hàng mạng sinh học nhằm tìm tương ứngđủ tốt giữa các nút mạng của các loài khác nhau cho phépxác định các vùng mạng có kiểu cấu trúc topology và cấu trúctrình tự, nhờ đó có thể chuyển một cách hiệu quảcác kiến thức về chức năng của tế bào từ các loài đã được nghiên cứu tốt sang những loài chưa được nghiên cứu nhiều hoặc khó làm thực nghiệm.Bởi vì việc nghiên cứu thực nghiệm trên con người gặp nhiều khó khăn bởi các rào cản đạo đức và pháp luật, nhờ dóng hàng mạng mà người ta có thể chuyển các tri thức đã biết từ nấm men (Saccharomyces cerevisiae), ruồi giấm (Drosophila melanogaster), hoặc sâu (Caenorhabditis elegans) sang tri thức của con người dựa trên phát hiện các vùng mạng được bảo tồn.

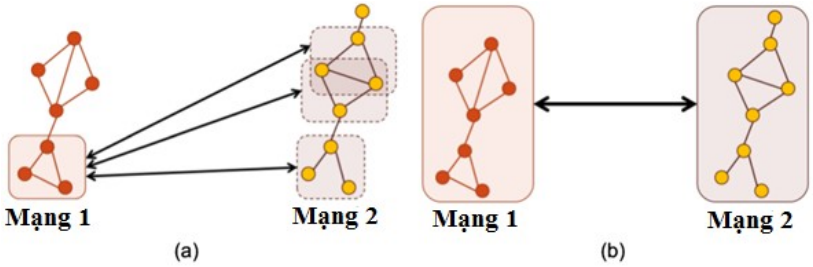
Luận án này tập trung nghiên cứu hai bài toán thời sự: dóng hàng toàn cục hai mạng tương tác protein-protein (về sau sẽ gọi gọn là mạng tương tác protein) và dóng hàng nhiều mạng các vị trí liên kết/hoạt tính protein.

2.2. Bài toán dóng hàng mạng tương tác protein

Các protein trong mỗi cơ thể sống không tồn tại một cách độc lập mà chúng tương tác với nhau. Dựa trên nghiên cứu thực nghiệm, người ta xây dựng được các CSDLvề các mạng tương tác protein (PPI). Việc dóng hàng hai mạng PPI cho phép chúng ta phát hiện các tương đồng chức năng giữa hai loài nhờ phát hiện các vùng tương tự giữa chúng.

Một mạng PPI được biểu thị bởi một đồ thị G(V,E) trong đó V là tập đỉnh mà mỗi nút ứng với một protein, E là tập cạnh, mỗi cạnh nối 2 nút biểu hiện tương tác của hai protein tương ứng. Ngoài tính topology thể hiện trên mạng, nhiều khi người ta còn quan tâm tới cả đặc tính cấu trúc của mỗi protein mà chúng không được biểu diễn trên đồ thị. Việc dóng hàng mạng được chia thành hai hướng tiếp cận: dóng hàng cục bộ và dóng hàng toàn cục.

Các nghiên cứu đầutiên về dóng hàng mạng PPI là dóng hàng cục.Dóng hàng cục bộ có mục tiêu là xác định các mạng/đồ thị con gần nhau về topologyvà về trình tự nhờ một ánh xạ từ mạng nọ vào mạng kia như minh họa trong hình 2.2 (a).



Hình 2.2. Dóng hàng cục bộ và dóng hàng toàn cục

Dóng hàng cục bộ có nhược điểm là khótìm ra các đồ thị con với kích thước lớn có cấu trúc và chức năng tương tự, kết quả của dóng hàng cục bộ là nhiều nhiều nên thường chứa nhiều các mạng con chồng lấn nhau nên thường dẫn tới sự nhập nhằng khó ứng dụng.

Một dóng hàng toàn cục mạng PPIlàmột đơn ánh từ mạng có số đỉnh nhỏ hơnvào mạng lớn (xem hình 2.2b),nhờ đó mà xác định các vùng bảo tồn. Việc xác định đơn ánh như vậy tránh được các nhập nhằng thường gặp ở phương pháp dóng hàng cục bộ.

Bài toán tối ưu dóng hàng toàn cục mạng PPI được chứng minh thuộc loại NP-khó nên đang là bài toán quan trọng trong sinh học phân tử và đã có nhiều thuật toán heuristics và metaheurristics đề xuất để giải chúng.

2.3. Bài toán dóng hàng nhiều mạng các vị trí liên kếtprotein.

Đã có các tiếp cận khác nhau được đề xuất để khám phá tính tương đồng cấu trúc, chủ yếu nhờ kỹ thuật đối sánh đúng các cặp đồ thị và nhận được những kết quả ý nghĩa khi nghiên cứu tiến hóa chức năng của các phân tử không thuần nhất. Tuy vậy, những phương pháp này khó khám phá các mẫu có ý nghĩa sinh học được lưu lại một cách gần đúng.

Weskamp và các cộng sự đầu tiên (2007) giới thiệu khái niệm dóng hàng nhiều đồ thị và dùng bài toán này để phân tích các vị trí hoạt tính protein (protein active sites), và đề xuất một thuật toán heuristic tìm lời giải theo chiến lược ăn tham greedy.

MGA là bài toán NP-khó, các thuật toán heuristic chỉ thích hợp cho các bài toán cỡ nhỏ, nên không phù hợp với các ứng dụng thực tế. Fober và các cộng sự đã mở rộng sử dụng bài toán này

cho phân tích cấu trúc phân tử sinh học và đề xuất một thuật toán tiến hóa với tên gọi GAVEO. Thực nghiệm cho thấy thuật toán này hiệu quả hơn thuật toán mà Weskamp đề xuất.

Trong chương 4, luận án đề xuất là ACO-MGA, ACO-MGA2 và ACOTS-MGA để giải bài toán dóng hàng nhiều đồ thị.

Chương 3.DÓNG HÀNG TOÀN CỤCMẠNG TƯƠNG TÁC PROTEIN

Chương này giới thiệu 3 thuật toán mà nhóm đề xuất để giải bài toán dóng hàng toàn cục mạng tương tác protein là FASTAN, ACOGNA và ACOGNA++.Các thực nghiệm đã chứng minh các thuật toán này cho chất lượng lời giải tốt hơn đáng kể so với các phương pháp mới nhất hiện nay. Bênh cạnh đó thời gian chạy của các thuật toán đề xuất cũng nhanh hơn so với các thuật toán metaheuristic khác có chất lượng lời giải tương đương.

3.1. Bài toán dóng hàng toàn cục mạng tương tác Protein

3.1.1. Phát biểu bài toán

Giả sử $G_1 = (V_1, E_1)$ và $G_2 = (V_2, E_2)$là 2 đồ thị mô tả 2 mạng tương tác protein, trong đó V_1, V_2 tương ứng là tập các đỉnh của các đồ thị G_1 và G_2; E_1, E_2 à tập các cạnh tương ứng củaG_1, G_2.Không mất tính tổng quát ta có thể giả sử $|V_1| \leq |V_2|$ trong đó |V| là ký hiệu cho số phần tử của tập V.

Định nghĩa 1.Dóng hàng toàn cục 2 mạng tương tác protein là xác định một đơn ánh $f: V_1 \to V_2$ trong đó mỗi đỉnh của V_1 được khớp với duy nhất 1 đỉnh $v_2 = f(v_1) \in V_2$.

Trong trường hợp $|V_1| = |V_2|$ thì f là một song ánh.

3.1.2. Đánh giá chất lượng dóng hàng toàn cục

Cho một dóng hàng mạng f ký hiệu $f(E_1) = \{(f(u), f(v)) \in E_2 : (u,v) \in E_1\}$ và $f(V_1) = \{f(v) \in V_2 : v \in V_1\}$. Các tiêu chuẩn dóng hàng được sử dụng phổ biến nhất trong các nghiên cứu về bài toán dóng hàng toàn cục mạng tương tác protein được trình bày như dưới đây:

Tiêu chuẩnGNAS được Aladaggiới thiệu được tính theo công thức sau:

$$\alpha|f(E_1)| + (1-\alpha)\sum_{u \in V_1} similar(u, f(u)) \qquad (4.1)$$

trong đó $\alpha \in [0.1]$ là tham số, $similar(u, f(u))$ là độ đo tương tự trình từ nào đó, chẳng hạn, BLAST bit-scores hay E-values. Ưu điểm của độ đo GNAS là thể hiện được cả mối tương quan về Topology và độ tương đồng về trình tự giữa 2 đồ thị được dóng hàng.

Kuchaiev và các cộng sựđề xuất dùng độ đo EC, Patro và các cộng sự đề xuất dùng độ đo ICS:

$$EC = \frac{|f(E_1)|}{|E_1|} \qquad (4.2)$$

$$ICS = \frac{|f(E_1)|}{|E(G_2[f(V_1)])|} \qquad (4.3)$$

trong đó $E(G_2[f(E_1)])$ là tập cạnh trong G_2 của đồ thị con có tập đỉnh là $f(V_1)$.

Saraph và các cộng sự nhận thấy khi mật độ cạnh ở hai mạng khác biệt thì hai độ đo này không phù hợp nên đề xuất độ đo S^3.

$$S^3 = \frac{|f(E_1)|}{|E_1| + |E(G_2[f(V_1)])| - |f(E_1)|} \qquad (4.4)$$

3.1.3. Dữ liệu thực nghiệm

Việc lựa chọn các bộ dữ liệu để so sánh các thuật toán rất quan trọng. Bộ dữ liệu được sử dụng so sánh để so sánh các phương pháp dóng hàng được đề xuất là bộ dữ liệu IsoBase, đây là bộ

dữ liệu thực gồm 4 mạng tương tác protein được sử dụng phổ biến khi đánh giá chất lượng các thuật toán dóng hàng mạng PPI. Đó là các mạng tương tác protein của các loài như: giun (caenorhabditis elegans), ruồi giấm (drosophila melanogaster), khỉ (saccharomyces cerevisiae) và người (homo sapiens). Các mạng tương tác này thu được từ [64]. Mô tả về các tập dữ liệu này được chỉ ra trong bảng 3.1. Từ các bộ dữ liệu đó chúng tôi tạo ra sáu cặp mạng tương tác để dóng hàng (*ce-dm, ce-hs,ce-sc,dm-hs, dm-sc,hs-sc)*.

Bảng 3. 1. Mô tả bộ dữ liệu

Tập dữ liệu	Ký hiệu	Số đỉnh	Số cạnh
C.elegans (Worm)	ce	2805	4495
D. melanogaster (fly)	dm	7518	25635
S.cerevisiae (yeast)	sc	5499	31261
H.sapiens (human)	hs	9633	34327

3.2. Một số thuật toán dóng hàng toàn cục mạng tương tác protein

Trong chương 2 luận án đã giới thiệu tổng quan về bài toán dóng hàng mạng tương tác protein và giới thiệu một số thuật toán liên quan. Để dễ theo dõi và tiện so sánh với các thuật toán đề xuất, mục này giới thiệu một số thuật toán dóng hàng toàn cục tiêu biểu được sử dụng để so sánh với các thuật toán đề xuất.

Thuật toán dóng hàng toàn cục đáng chú ý đầu tiên là IsoRank được Sing và các cộng sự đề xuất năm 2008, phát triển dựa trên dóng hàng cục bộ. IsoRank có ý tưởng xuất phát từ thuật toán PageRank của Google để định nghĩa hàm đánh giá sự tương đồng. Ý tưởng chính của IsoRank là hai nút được dóng hàng với nhau, nếu các nút kề với chúng tương ứng được dóng hàng.

Họ các thuật toán GRAAL bao gồm GRAAL, H-GRAAL, MI-GRALL và sau đó là C-GRAAL được phát triển song song với họ các thuật toán ISORAnk dựa trên kết hợp kỹ thuật tham lam với thông tin heuristics như: graphlet, hệ số phân nhóm, độ lập dị (eccentricities) và độ tương tự (giá trị E-values từ chương trình BLAST). Các thuật toán này đều đưa ra kết quả nhanh và tốt hơn so với các thuật toán trước đó.

Gần đây hơn là thuật toán GHOST, chiến lược dóng hàng của GHOST cũng tương tự như của MI-GRAAL, ngoại trừ việc thuật toán MI-GRAAL giải bài toán quy hoạch tuyến tính để tính toán độ tương tự giữa các nút trên các mạng khác nhau, trong khi GHOST giải bài toán quy hoạch bậc 2 theo phương pháp heuristic để tính toán độ tương tự giữa các nút trong cùng một mạng.

Những thuật toán đã nêu chỉ tối ưu cho độ chính xác (hàm mục tiêu) hoặc tính khả mở. Vì các mạng PPI thường có số đỉnh lớn nên cả tính chính xác và tính khả mở (thời gian chạy) cần được quan tâm. Sử dụng tiêu chuẩn GNAS, Aladag và các cộng sự [1] đề xuất thuật toán SPINAL cho lời giải tốt hơn các thuật toán trước đó cả về thời gian và chất lượng lời giải.

Gần đây, Saraph và các cộng sự đề xuất thuật toán MAGNA (2014) dựa trên giải thuật di truyền với quần thể ban đầu khởi tạo ngẫu nhiên hoặc kết hợp với lời giải được tìm bởi các thuật toán như: IsoRank, MI-GRAAL và GHOST. MAGNA và phiên bản cải tiến MAGNA ++ [84]sử dụng độ đo chất lương dóng hàng S^3, thực nghiệm cho thấy chúng cải thiện đáng kể chất lượng lời giải của các thuật toán được dùng để khởi tạo.

Somaye Hashemifar và các cộng sự (2016) giới thiệu 1 thuật toán tối ưu toàn cục mới tên là ModuleAlign, thuật toán này sử dụng thông tin tối ưu cấu trúc cục bộ để định nghĩa một hàm đánh giá tính tương đồng dựa trên module (module-based homology score). Dựa trên một thuật toán phân cụm chức năng của các protein có gắn kết về mặt chức năng vào trong cùng module, ModuleAlign

sử dụng một cơ chế lặp mới để tìm dóng hàng giữa 2 mạng. Các thực nghiệm đã cho thấy ModuleAlign cho kết quả chất lượng dóng hàng tốt hơn một số thuật toán đề xuất trước đó trong một số trường hợp.

3.3. Thuật toán nhanh giải bài toán dóng hàng mạng tương tác Protein

3.3.1. Đặc tả thuật toán

Thuật toán FASTAN gồm hai giai đoạn: giai đoạn thứ nhất *xây dựng dóng hàng ban đầu* và giai đoạn sau cải tiến nó nhờ thủ tụctối ưu cục bộ *Rebuild*.

3.3.1.1. Xây dựng dóng hàng ban đầu

Cho các đồ thị G_1, G_2; tham sốα và các độ tương tự của các cặp đỉnh <i,j> tương ứng của V_1, V_2 là similar(i, j). Ký hiệu V^i là tập các đỉnh đã được dóng hàng của đồ thị G_i và $RV_i = V_i - V^i$ là tập các đỉnh chưa được dóng hàng của đồ thị G_i. Gọi $A_{12}= (V_{12}, E_{12})$ là kết quả của phép dóng hàng đồ thị G_1 với đồ thị G_2, trong đó $V_{12} = \{< i, f(i) >: i \in V_1, f(i) \in V_2\}$; $E_{12} = \{(< u, f(u) >, < v, f(v) >) : (u,v) \in E_1, (f(u), f(v)) \in E_2\}$

Thủ tục FASTAN được thực hiện như sau:

Bước 1. Xác định cặp đỉnh $i \in V_1$ và $j \in V_2$ có độ tương tự *similar(i, j)* lớn nhất. Gán *f(i):=j;*

Bước 2. Thực hiện lặp với *k= 2* tới $|V_1|$

2.1. Tìm node $i \in RV_1$có số cạnh nối với các đỉnh trong V^1 lớn nhất (Thủ tục này gọi là *find_next_node*).

2.2. Tìm *f(i)* = $j \in RV_2$ mà khi dóng hàng j với i thì công thức $\alpha|f(E_1^*)| + (1 - \alpha)(\sum_{u \in V^1} similar(u, f(u)) + similar(i,j))$ đạt giá trị lớn nhất. Trong đó E_1^* là các cạnh của đồ thị G_1 có các đỉnh thuộc tập $V^1 \cup i$. (Thủ tục này gọi là *choose_best_matched_node)*.

Bước 3. Thực hiện lặp cải tiếnA_{12}nhờ thủ tục *Rebuild*.

Chú ý rằng ở các bước 2.1 và 2.2 có thể tìm được nhiều đỉnh *tốt nhất*, khi đó sẽ chọn ngẫu nhiên một đỉnh trong số đó.

Sau khi xây dựng được dóng hàng ban đầu FastAn chuyển sang giai đoạn 2. Trong giai đoạn này, thủ tục *Rebuild* được thực hiện lặp để cải tiến dóng hàng đã có.

3.3.1.2. Thủ tục Rebuild

Sau giai đoạn 1, đã xác định được dóng hàng thô A_{12}, để tăng chất lượng của lời giải, thuật toán sử dụng thủ tục tối ưu cục bộ ***rebuild.*** Ý tưởng của thủ tục này là sử dụng một tập giống gồm n_{keep} những cặp đỉnh đã được dóng hàng tốt của A_{12}, sau đó dóng hàng lại các cặp đỉnh khác, nếu lời giải mới tốt hơn sẽ thay thế cho lời giải trước đó. Chi tiết thủ tục rebuil như dưới đây.

Bước 1. Xác định tập $SeedV_{12}$ của V_1 gồm n_{keep}đỉnh cóscore tốt nhất của V_1 theo tiêu chí cho bởi công thức 3.5:

$$score(u) = \alpha \times w(u) + (1 - \alpha) \times similar(u, f(u)) \qquad (4.5)$$

trong đó *u* thuộc V_1 và $f(u)$ là đỉnh thuộc V_2 được ghép với *u* trongA_{12},*w(u)* là số lượng nút *v* thuộc V_1 mà (u,v) thuộc E_1 và (*f(u),f(v)*) thuộc E_2

Bước 2. Xác định V_{12} khởi tạo nhờ $SeedV_{12}$ và A_{12}

Bước 3. Thực hiện lặp như *bước 2* của phase 1 với *k* =$n_{keep} + 1$ tới $|V_1|$để xác định A_{12}

Sau mỗi lần thực hiện thủ tục *Rebuild* ta có một dóng hàng mới làm input G_{12}cholần lặp tiếp theo, quá trình này lặp lại cho đến khi không cải tiến được *GNAS(A_{12})* nữa.

3.3.2. Độ phức tạp của thuật toán FASTAN so với SPINAL

Trongnghiên cứu của Aladag và Erten, các tác giả cũng đã đề xuất thuật toán SPINAL có độ phức tạp với thời gian đa thức là:

$$SPINALComplexity = O\left(k \times |V_1| \times |V_2| \times \Delta_1 \times \Delta_2 \times log\left(\Delta_1 \times \Delta_2\right)\right) \quad (4.6)$$

Trong đó k là số lần lặp chính khi chạy thuật toán, Δ_1, Δ_2 lần lượt là bậc của đỉnh của đỉnh thuộc các đồ thị G_1 và G_2 có bậc lớn nhất.

Dễ dàng kiểm tra được độ phức tạp của giai đoạn 1 và mỗi bước lặp trong giai đoạn 2 của thuật toán FastAn (Số lần lặp của giai đoạn 2 trong thực nghiệm không vượt qúa 20 lần) là:

$$O(|V_1| \times (E_1| + |E_2|)) \quad (4.7)$$

Bởi vì $|V_1| \times \Delta_1 \geq E_1$ nên chú ý tới độ phức tạp của SPINAL trong công thức (3.6) ta có:

$$|V_1| \times |V_2| \times \Delta_1 \times \Delta_2 \geq E_1 \times E_2 > (|V_1| \times (E_1| + |E_2|)). \quad (4.8)$$

Như vậy độ phức tạp của FastAn so với độ phức tạp của SPINAL thấp hơn nhiều.

3.3.3. Kết quả thực nghiệm

Luận án so sánh thuật toán FASTAn và Spinal về chất lượng lời giải và thời gian chạy. Kết quả thực nghiệm chỉ ra rằng FASTAn có thể tìm ra lời giải (dóng hàng toàn cục) có điểm GNAS và $|E_{12}|$ tốt hơn nhiều so với Spinal (p-value $<2.2e^{-16}$ được tính sử dụng t-test dựa trên kết quả GNAS và$|E_{12}|$ values của 100 lần chạy) đối với cả 6 cặp mạng protein. Ngoài ra, kết quả kém nhất của FASTAn từ 100 lần chạy đối với tất cả các cặp mạng protein được dóng hàng đều tốt hơn các kết quả dóng hàng tạo ra bởi Spinal.

Khi so sánh về thời gian chạy, thuật toán FASTAn có thời gian chạy nhanh hơn Spinal đối với tất cả các bộ dữ liệu.

3.4. Thuật toán ACO cho bài toán dóng hàng toàn cục mạng tương tác Protein

3.4.1. Lược đồ chung

Thuật toán ACOGNA được xây dựng như dưới đây:

Bước 1. Khởi tạo ma trận vết mùi, và tập A gồm m kiến.

Bước 2. Thực hiện lặp trong khi chưa thoả mãn điều kiện dừng

Với mỗi kiến ta tiến hành các bước sau:

2.1. Gán *f(i)*=j trong đó *i, j*là cặp đỉnh có độ tương đồng *similar (i,j)* lớn nhất.

2.2 Thực hiện lặp với *k= 2* tới $|V_1|$

2.2.1. Tìm đỉnh $i \in RV_1$ có số cạnh tới các đỉnh trong V^1 lớn nhất;

*2.2.2.*Tìm đỉnh *f(i)*= $j \in RV_2$ theo thủ tục bước ngẫu nhiên (*thủ tục antMove*)

2.3. Thực hiện tìm kiếm cục bộ trên lời giải tốt nhất do các kiến tìm được để cải thiện chất lượng lời giải.

2.4. Cập nhật lại lời giải tốt nhất.

2.4. Cập nhật vết mùi theo quy tắc SMMAS dựa trên lời giải tốt nhất.

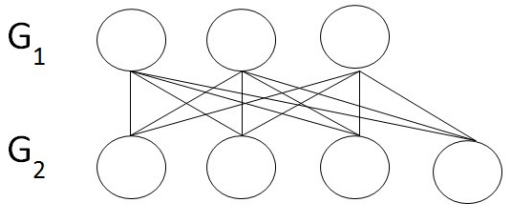
Bước 3. Lưu lại lời giải tốt nhất.

Chú ý rằng ở bước 2.2.1, việc tìm $i \in RV_1$ có số cạnh tới các đỉnh trong V^1 lớn nhất nhằm tăng số lượng các cạnh có thể được bảo toàn sau khi dóng hàng, nếu tìm được nhiều đỉnh tốt nhất thì sẽ lựa chọn ngẫu nhiên một đỉnh tìm được để dóng hàng.

3.4.2. Đồ thị cấu trúc

Đồ thị cấu trúc của thuật toán gồm 2 tầng, tầng thứ *i* thể hiện đồ thị G_i . Các đỉnh ở tầng trên được kết nối với tất cả các đỉnh ở tầng dưới. Hình 3.4 thể hiện đồ thị cấu trúc của thuật toán

ACOGNA. Khi xây dựng lời giải, kiến sẽ xuất phát từ một đỉnh thuộc tầng 1 và lựa chọn dóng hàng với 1 đỉnh thuộc tầng 2 theo công thức (3.10).



Hình 3. 1. Đồ thị cấu trúc của thuật toán ACOGNA

Một dóng hàng toàn cục của 2 đồ thị theo định nghĩa 1 là một đường đi xuất phát từ 1 đỉnh của G_1 dóng với 1 đỉnh của G_2 sau đó quay lại G_1 rồi tiếp tục dóng với 1 đỉnh của G_2, lặp lại cho tới khi tất cả các đỉnh của G_1 đã được dóng hàng.

3.4.3. Vết mùi và thông tin heuristic

Vết mùi τ_j^i trên cạnh $< i, j >$ dóng đỉnh $i \in V_1$ với đỉnh $j \in V_2$ được khởi tạo bằng τ_{max} và sau đó được cập nhật lại sau mỗi vòng lặp theo công thức 3.11

Thông tin heuristic η_j^i được tính theo công thức 3.9.

$$\eta_j^i = \alpha * f\left(E_1^*\right) + (1-\alpha) * similar(i,j) \quad (4.9)$$

Trong đó $f\left(E_1^*\right)$ là số cạnh được bảo tồn nếu tiếp tục dóng hàng đỉnh *j* với đỉnh *i*, α là hằng số thể hiện mối tương quan giữa độ tương đồng về cấu trúc và tính tương đồng về trình tự, $similar(i, j)$ là độ tương đồng giữa 2 đỉnh *i* và *j*.

3.4.4. Thủ tục bước ngẫu nhiên để xây dựng dóng hàng

Tại mỗi vòng lặp, sau khi chọn một đỉnh $i \in RV_1$ bằng thủ tục *find_next_node* tương tự thuật toán FASTAN, kiến chọn đỉnh $j \in RV_2$ theo xác suất được cho bởi công thức 3.10

$$p_j^i = \frac{(\tau_j^i)^a * [\eta_j^i]^b}{\sum_{k \in RV_2} (\tau_k^i)^a * [\eta_k^i]^b} \quad (4.10)$$

Sau khi lựa chọn được đỉnh $j \in RV_2$ để dóng với $i \in RV_1$, kiến quay lại lựa chọn đỉnh tiếp theo của đồ thị G_1 để tiếp tục dóng hàng. Quá trình lặp lại cho đến khi tất cả các đỉnh của G_1 được dóng hàng với các đỉnh của G_2

3.4.5. Quy tắc cập nhật vết mùi

Sau khi tất cả các kiến đã xây dựng lời giải, lời giải của kiến tốt nhất được áp dụng thủ tục tìm kiếm cục bộ để tăng chất lượng lời giải. Lời giải tốt nhất này được sử dụng để cập nhật vết mùi trên các cạnh theo quy tắc cập nhật mùi SMMAS, như dưới đây:

$$\tau_j^i = (1-\rho)\tau_j^i + \Delta_j^i \quad (4.11)$$

$$\Delta_j^i = \begin{cases} \rho * \tau_{max} & j = f(i) \\ \rho * \tau_{min} & j \neq f(i) \end{cases} \quad (4.12)$$

Trong đó τ_{max} và τ_{min} là các tham số được cho trước, $\rho \in (0,1)$ là tham số bay hơi cho trước quy định 2 thuộc tính, ρ nhỏ thể hiện việc tìm kiếm quanh thông tin học tăng cường, ρ lớn thể hiện tính khám phá.

3.4.6. Thủ tục tìm kiếm cục bộ

Trong mỗi vòng lặp, sau khi tất cả các kiến đã xây dựng xong lời giải, lời giải tốt nhất A_{12} được kiến xây dựng sẽ được áp dụng tìm kiếm cục bộ. Thủ tục tìm kiếm cục bộ được cải tiến từ thủ tục

rebuilt trong FASTAN.

Điểm khác biệt của ACOGNA so với FASTAN là khi chất lượng dóng hàng tăng lên khi gọi thủ tục dóng hàng cục bộ thì giá trị nkeep sẽ được điều chỉnh tăng lên để giữ được nhiều cặp đỉnh tốt hơn và giảm thời gian xây dựng lại các dóng hàng.

3.4.7. Kết quả thực nghiệm

Luận án so sánh thuật toán ACOGNA với thuật toán FASTAn theo tiêu chuẩn GNAS và giá trị $|E_{12}|$. Kết quả thực nghiệm cho thấy trong tất cả các trường hợp thì thuật toán ACOGNA đều cho các kết quả tốt hơn so với thuật toán FASTAn đối với tiêu chuẩn là GNAS và cả giá trị $|E_{12}|$.

Tiến hành thực nghiệm so sánh ACOGNA với thuật toán MAGNA++, các kết quả thực nghiệm chỉ ra rằng với tất cả các giá trị của tham số α và tất cả các bộ dữ liệu, điểm số EC của ACOGNA luôn luôn tốt hơn MAGNA++.Với tiêu chuẩn S^3 khi chạy với các bộ dữ liệu dm-hs, dm-sc, hs-sc với tất cả các giá trị của tham số **α**, thuật toán ACOGNA cho kết quả tốt hơn so với MAGNA++ khi chạy thuật toán này theo cả 3 tùy chọn tiêu chuẩn tối ưu là EC, ICS và S^3. Tuy nhiên đối với 3 bộ dữ liệu ce-dm, ce-sc, ce-hs MAGNA++ lại cho kết quả tốt hơn ACOGNA.

3.5. Thuật toán ACOGNA++

3.5.1. Mô tả thuật toán

Với đồ thị cấu trúc được xây dựng giống như thuật toán ACOGNA, để xây dựng một dóng hàng, các kiến sẽ thực hiện quả trìnhlặp để xác định 1 đỉnh thuộc tầng 1 của đồ thị cấu trúc và một đỉnh thuộc tầng 2 sẽ được dóng hàng với nó. Quá trình này kết thúc khi tất cả các đỉnh thuộc đồ thị G_1 đã được dóng hàng. Sau khi tất cả các kiến xây dựng xong dóng hàng, thủ tục tìm kiếm cục bộ sẽ được áp dụng trên lời giải tốt nhất của vòng lặp để nâng cao chất lượng.

Tùy theo tiêu chuẩn tối ưu được lựa chọn là GNAS, EC hay S^3, tiêu chuẩn được sử dụng để lựa chọn lời giải tốt nhất sẽ được thay đổi tương ứng theo các hàm mục tiêu này.

3.5.2. Vết mùi

Vết mùi lưu thông tin học tăng cường để đánh giá một cặp đỉnh được dóng hàng là tốt hay không. Thuật toán ACOGNA++ sử dụng 2 ma trận vết mùi.Vết mùi τ_1^i đặt trên các đỉnh của đồ thị G_1 để xác định các đỉnh sẽ được ưu tiên lựa chọn để dóng hàng trước.Vết mùi τ_j^i đặt trên cạnh (i,j) của đồ thị cấu trúc, dùng để xác định đỉnh $j \in G_2$ được dóng hàng với đỉnh $i \in G_1$. Các vết mùi được khởi tạo bằng giá trị τ_{max} và được cập nhật lại sau mỗi vòng lặp.

3.5.3. Thủ tục xác định cặp đỉnh dóng hàng

Thủ tục này gồm 2 bước, đầu tiên xác định đỉnh được dóng hàng trên đồ thị G_1 và sau đó là xác định ảnh của nó trên đồ thị G_2.

Xác định đỉnh được dóng hàng thuộc đồ thị nguồn

Khác với thủ tục*find_next_node* trong FASTAN và ACOGNA sử dụng để xác định đỉnh $i \in RV_1$sẽ được dóng hàng. Thuật toán ACOGNA++ sử dụng thuật toán ACO để xác định đỉnh i được dóng hàng như dưới đây.

Gọi tập T chứa các đỉnh i sao cho $i \in RV_1$ và có nhiều cạnh nối với các đỉnh của V^1nhất. Khi đó, đỉnh $i \in T$ được chọn ngẫu nhiên theo xác suất:

$$p_i = \frac{(\tau_1^i)^a * [\eta_i]^b}{\sum_{j \in T} (\tau_1^j)^a * [\eta_j]^b} \qquad (4.13)$$

Trong đó η_i là số lượng đỉnh kề của i trong đồ thị G_1, τ_1^i là vết mùi τ_1^i đặt trên các đỉnh của đồ thị G_1 như đã mô tả ở mục 3.5.2.

Việc sử dụng ACO để tìm đỉnh thuộc đồ thị nguồn được dóng hàng sẽ giúp khai thác tốt thông tin học tăng cường thông qua vết mùi mà các kiến để lại. Điều này giúp cải thiện chất lượng lời giải tốt hơn so với cách lựa chọn ngẫu nhiên trong FASTAN và ACOGNA.

Xác định ảnh của điểm được dóng hàng trên đồ thị đích G_2

Sau khi xác định được đỉnh $i \in V_1$ đỉnh $j \in V_2$ được các kiến lựa chọn theo xác suất.

$$p_j^i = \frac{(\tau_j^i)^c * [\eta_j^i]^d}{\sum_{k \in RV_2} (\tau_k^i)^c * [\eta_k^i]^d} \quad (4.14)$$

Khi chạy thuật toán ACOGNA++ để tối ưu theo hàm mục tiêu GNAS thì thông tin heuristic được sử dụng giống thuật toán ACOGNA. Trong trường hợp chạy thuật toán ACOGNA++ tối ưu theo hàm mục tiêu EC, hoặc S^3, thông tin heuristic η_j^i lần lượt được tính theo các công thức 3.15 hoặc 3.16.

$$\eta_j^i = \frac{\left| f\left(E(G_1[V^1 \cup i])\right)\right|}{|E_1|} \quad (4.15)$$

$$\eta_j^i = \frac{\left| f\left(E(G_1[V^1 \cup i])\right)\right|}{|E_1| + \left|E(G_2\left[f(V^1) \cup j\right])\right| - \left|f\left(E(G_1[V^1 \cup i])\right)\right|} \quad (4.16)$$

3.5.4. Quy tắc cập nhật vết mùi

Sau mỗi vòng lặp, lời giải tốt nhất được xác định được sử dụng để cập nhật lại vết mùi theo quy tắc cập nhật mùi SMMAS.

Vết mùi đặt trên các đỉnh của đồ thị G_1 được cập nhật theo công thức 3.17 và 3.18:

$$\tau_1^i \leftarrow (1-\rho)\tau_1^i + \Delta\tau_i \quad (4.17)$$

Trong đó

$$\Delta\tau_i = \begin{cases} \rho\tau_{min} \text{ nếu } < i, f(i) > \text{ không có đỉnh kề} \\ \rho\tau_{max} \text{nếu } < i, f(i) > \text{ có ít nhất một đỉnh kề} \end{cases} \quad (4.18)$$

Vết mùi đặt trên các cạnh của đồ thị cấu trúc được cập nhật theo công thức (3.19) và (3.20)

$$\tau_j^i = (1-\rho)\tau_j^i + \Delta_j^i \quad (4.19)$$

$$\Delta_j^i = \begin{cases} \rho * \tau_{max} & j = f(i) \\ \rho * \tau_{min} & j \neq f(i) \end{cases} \quad (4.20)$$

3.5.5. Thủ tục tìm kiếm cục bộ

Thủ tục tìm kiếm cục bộ của ACOGNA++ được sử dụng tương tự như trong ACOGNA.

3.5.6. Kết quả thực nghiệm

Luận án tiến hành so sánh chất lượng lời giải của các thuật toán theo các tiêu chuẩn S^3, GNAS, EC. Thuật toán ACOGNA++ được so sánh với các thuật toán ACOGNA, MAGNA++, và ModuleAlign.

Điểm mới của thuật toán ACOGNA++ so với ACOGNA là có thể tối ưu theo các hàm mục tiêu khác nhau (tương tự như MAGNA++). Khi so sánh theo hàm mục tiêu GNAS và EC, thì 2 thuật toán ACOGNA và ACOGNA++ có chất lượng tương đồng nhau còn khi so sánh thuật toán ACOGNA++ chạy với tiêu chuẩn tối ưu là S^3. Kết quả thực nghiệm cho thấy thuật toán ACOGNA++ cho chất lượng lời giải theo tiêu chuẩn S^3 vượt trội so với các thuật toán còn lại.

Chương 4. BÀI TOÁN DÓNG HÀNG CÁC MẠNG CÁC VỊ TRÍ LIÊN KẾT PROTEIN

Chương này giới thiệu các khái niệm liên quan đến bài toán dóng hàng nhiều đồ thị, một công cụ để phân tích cấu trúc protein. Bên cạnh đó giới thiệu 3 thuật toán phát triển dựa trên phương pháp tối ưu hóa đàn kiến: ACO-MGA, ACO-MGA2, ACOTS-MGA. Thuật toán đầu tiên ACO-MGA được xây dựng dựa trên phương pháp tối ưu đàn kiến thuần túy. Thuật toán ACO-MGA2 được xây dựng dựa trên lược đồ memetic theo 2 giai đoạn, giai đoạn đầu chỉ sử dụng ACO, không có tìm kiếm cục bộ, giai đoạn sau có áp dụng tìm kiếm cục bộ. Thuật toán thứ 3 là ACOTS-MGA có sự kết hợp giữa thuật toán ACO và tìm kiếm Tabu theo lược đồ memetic để tìm lời giải cho bài toán dóng hàng nhiều đồ thị.

4.1. Bài toán dóng hàng nhiều đồ thị

4.1.1. Tập nhiều đồ thị (multigraph)

Một multigraph là một tập hợp các đồ thị $G = \{G_1(V_1,E_1),\ldots,G_n(V_n,E_n)\}$, trong đócác đồ thị $G_i(V_i,E_i)$ liên thông, đỉnh (node) được gán nhãn thuộc tập L cho trước, các cạnh có trọng số biểu thị khoảng cách giữa các đỉnh. Trong mô hình các vị trí liên kết protein (protein binding sites), các nhãn của các nodes có thể là: hydrogen-bond donor, acceptor, mixed donor/acceptor, hydrophobic aliphatic, và aromatic. Trong các đồ thị có các toán tử soạn thảo (edit operations) được định nghĩa như sau.

Định nghĩa 4.1. (Các toán tử soạn thảo) Trên các đồ thị $G(V,E)$ của tập đồ thị G có các toán tử soạn thảo:

i) Chèn hoặc xóa bớt các nút: Một nút $v \in V$ và các cạnh liên kết với nó có thể bị xóa hoặc được chèn vào

ii) Thay đổi nhãn của một nút: Nhãn $l(v)$ của một nút $v \in V$ có thể thay bằng nhãn khác thuộc tập L

iii) Thay đổi trọng số của một cạnh: Trọng số $w(e)$ của một cạnh e có thể thay đổi tùy theo những hình thể khác nhau.

4.1.2. Dóng hàng nhiều đồ thị

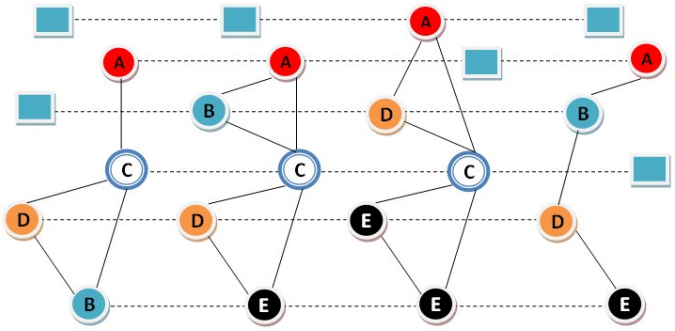
Cho tập đồ thị $G = \{G_1(V_1,E_1),\ldots,G_n(V_n,E_n)\}$, với mọi tập đỉnh $V_i$ta thêm vào nút dummy (ký hiệu là$\perp$) không có cạnh kết nối với các đỉnh khác, khi đó một dóng hàng của G được định nghĩa như sau.

Định nghĩa 4.2. (Multiple Graph Alignment).

Tập $A \subseteq \{V_1 \cup \{\perp\}\} \times \ldots \times \{Vm \cup \{\perp\}\}$ là một dóng hàng của đa đồ thị G nếu và chỉ nếu:

1. Với mọi $i=1,\ldots,n$ và với mỗi $v \in V_i$, tồn tại đúng một $a = (a_1,\ldots,a_n) \in A$sao cho $v = a_i$
2. Với mỗi $a = (a_1,\ldots,a_n) \in A$, tồn tại ít nhất một $1 \le i \le n$ sao cho$a_i \neq \perp$

Hình 4.1 minh họa một dóng hàng của một 4-đồ thị với các đỉnh dummy dạng hình vuông và các đỉnh có nhãn là các ô tròn có nhãn là các ký tự. Lưu ý rằng mỗi đồ thị chỉ dùng một đỉnh dummy nhưng để dễ hình dung, đồ thị thứ nhất và thứ tư ta để hai đỉnh có nhãn dummy với nghĩa rằng các nút ở hàng tương ứng được dóng với nút dummy ở đồ thị này.



Hình 4. 1. Một dóng hàng nhiều đồ thị của tập 4 đồ thị , đỉnh hình vuông là dummy còn các đỉnh tròn có nhãn là các ký tự tương ứng.

4.1.3. Hàm đánh giá chất lượng dóng hàng

Định nghĩa4.3(Hàm đánh giá chất lượng dóng hàng)

Với mỗi dóng hàng A của đa đồ thị G, hàm đánh giá chất lượng *s(A)* được xác định theo biểu thức (4.1):

$$s(A) = \sum_{i=1}^{n} ns(a^i) + \sum_{1 \le i < j \le n} \text{es}(a^i, a^j) \qquad (5.1)$$

trong đó ns là điểm đánh giá tính phù hợp của cột tương ứng và được tính theo biểu thức (4.2):

$$ns\begin{pmatrix} a_1^i \\ \vdots \\ a_m^i \end{pmatrix} = \sum_{1 \le j < k \le m} \begin{cases} ns_m & l(a_j^i) = l(a_k^i) \\ ns_{mm} & l(a_j^i) \ne l(a_k^i) \\ ns_{dummy} & a_j^i = \perp,\ a_k^i \ne \perp \\ ns_{dummy} & a_j^i \ne \perp,\ a_k^i = \perp \end{cases} \qquad (5.2)$$

còn es đánh giá tính tương thích của độ dài cạnh và được tính bởi biểu thức (4.3):

$$es\left(\begin{pmatrix} a_1^i \\ \vdots \\ a_m^i \end{pmatrix}, \begin{pmatrix} a_1^j \\ \vdots \\ a_m^j \end{pmatrix}\right) = \sum_{1 \le k < l \le m} \begin{cases} es_{mm} & (a_k^i, a_k^j) \in E_k, (a_l^i, a_l^j) \notin E_l \\ es_{mm} & (a_k^i, a_k^j) \notin E_k, (a_l^i, a_l^j) \in E_l \\ es_m & d_{kl}^{ij} \le \varepsilon \\ es_{mm} & d_{kl}^{ij} > \varepsilon \end{cases} \qquad (5.3)$$

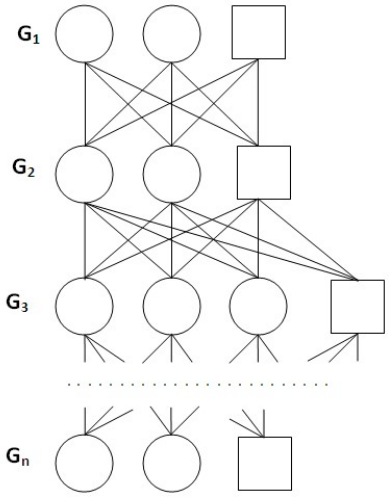
Trong công thức (4.3) $d_{kl}^{ij} = \left| w(a_k^i) - w(a_l^j) \right|$.

Các tham số (ns_m, ns_{mm} , ns_{dummy} , es_m , es_{mm}) được lấy như trong [10]: ns_m = 1.0; ns_{mm} = -5.0; ns_{dummy} = -2.5; es_m = 0.2; es_{mm} =-0.1.

Lời giải cần tìm của bài toán MGA là dóng hàng làm cực đại hàm đánh giá $s(A)$. Đây là bài toán NP-khó, nếu dùng phương pháp vét cạn độ phức tạp sẽ là $O((Vmax)!^n)$ với Vmax là số đỉnh của đồ thị có nhiều đỉnh nhất và n là số đồ thị.

4.2. Thuật toán ACO cho bài toán dóng hàng nhiều đồ thị

4.2.1. Đồ thị cấu trúc



Hình 4. 2. Đồ thị cấu trúc khi dóng hàng n đồ thị, trong đó mỗi đồ thị có 2 hoặc 3 node thực

Đồ thị cấu trúc gồm n tầng , tầng thứ i là đồ thi Gi của G, các đỉnh của tầng trên đều có cạnh kết nối với các đỉnh tầng dưới. Hình 4.2 minh họa đồ thị cấu trúc, trong đó không hiển thị các cạnh ở mỗi đồ thị trong mỗi tầng, nút hình tròn là nút thực còn nút biểu diễn bởi hình vuông là nút dummy.

Một dóng hàng của đồ thị theo định nghĩa 2 ở trên là một tập đường đi từ G_1qua mọitầng đến G_n sao cho mỗi đường chỉ đi qua một đỉnh của mỗi tầng và mỗi đỉnh thực của đồ thị cấu trúc đều có đúng một đường đi qua, riêng các đỉnh ảo thì cho phép có nhiều đường qua nó.Tập đường đi này có thể xem là chỉ 1 đường duy nhất như quan niệm của thuật toán ACO thông dụng với ngầm định rằng đường này khởi đầu từ một đỉnh của G_1 đi qua các đồ thị kế tiếp, khi đến tầng đầu hoặc tầng cuối thì "bước" sang đỉnh khác cùng tầng rồi quay lại cho đến khi qua hết mọi đỉnh thực mỗi đỉnh đúng một lần.

4.2.2. Thủ tục bước ngẫu nhiên để xây dựng một dóng hàng

Trong mỗi bước lặp, mỗi con kiến sẽ thực hiện lặp quá trình xây dựng các vectơ $a = (a_1,\ldots,a_n)$chomột dóng hàng A như sau.

Kiến chọn ngẫu nhiên một đỉnh thực trên đồ thị cấu trúc và dựa trên thông tin heuristics và pheromone trail để bước ngẫu nhiên xây dựng lời giải. Để dễ hình dung, ta giả thiết đỉnh thực này ở G_1 (được ký hiệu là a_1, kiến sẽ bước ngẫu nhiên qua các tầng để đến G_n như sau. Nếu kiến đã xây dựng được vec tơ $(a_1,\ldots,a_i)$ trong đó a_q là đỉnh j trong G_i thì nó chọn đỉnh k trong G_{i+1} với xác suất cho bởi công thức (4.4):

$$P^k_{ij} = \frac{{\tau^i_{j,k}}^{\alpha} * \left[\eta^i_{j,k}(a)\right]^{\beta}}{\sum_{s \epsilon R\_V_{i+1}} {\tau^i_{j,s}}^{\alpha} * \left[\eta^i_{j,s}(a)\right]^{\beta}}, \qquad (5.4)$$

trong đó R_V_i là số đỉnh còn lại chưa dóng hàng trên V_i kể cả nút dummy, $\tau^i_{j,k}$ là cường độ vết mùi của cạnh nối đỉnh j của G_i tới đỉnh k của G_{i+1}, còn $\eta^i_{j,k}(a)$ là thông tin heuristics được tính bởi công thức (4.5).

$$\eta^i_{j,k}(a) = \begin{cases} \frac{NL(k,a)}{i} & k \text{ là đỉnh thực} \\ \eta_{min} & k \text{ là đỉnh ảo} \end{cases} \qquad (5.5)$$

trong đó $NL(k,a)$ là số đỉnh trong $\{a_1,\ldots a_i\}$ có nhãn trùng với nhãn $l(k)$ của đỉnh k, $\eta_{min} > 0$ là giá trị đủ bé cho trước

Sau khi vectơ a được phát triển hết thành $a=(a_1,\ldots a_n)$ thì các đỉnh thực trong a bị loại ra khỏi đồ thị cấu trúc để tiếp tục lặp thủ tục dóng hàng của kiến đến khi mọi đỉnh thực đã được dóng hàng.

Lưu ý rằng nếu đỉnh thực được chọn ban đầu không thuộc G_1 mà là G_m thì thủ tục trên gồm hai quá trình dóng dần từ G_mtới G_n và dóng ngược từ G_m tới G_1

4.2.3. Qui tắc cập nhật mùi

Sau khi các con kiến đã tìm được lời giải, các lời giải của bước lặp được đánh giá và chọn lời giải tốt nhất để thực hiện tìm kiếm địa phương cải tiến chất lượng rồi thực hiên cập nhật mùi.

Vết mùi được cập nhật theo quy tắc cập nhật mùi SMMAS như trong công thức 4.6 và 4.7:

$$\tau^i_{j,k} = (1-\rho)\tau^i_{j,k} + \Delta^i_{j,k} \quad (5.6)$$

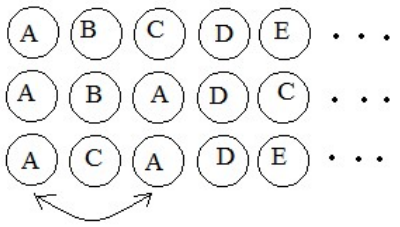
Trong đó: $\Delta^i_{j,k} = \begin{cases} \rho * \tau_{max} & (i,j,k) \in \ best\ solution \\ \rho * \tau_{min} & (i,j,k) \notin \ best\ solution \end{cases}$ (5.7)

Với τ_{max} và τ_{min} là các tham số cho trước.

4.2.4. Thủ tục tìm kiếm cục bộ

Thủ tục tìm kiếm địa phương được áp dụng cho lời giải tốt nhất theo nguyên tắc tốt nhất thì dừng. Trong thủ tục này, các cặp đỉnh cùng nhãn trong mỗi đồ thị G_i được chọn ngẫu nhiên sẽ đổi chỗ cho nhau trong vectơ dóng hàng của nó để cải thiện độ phù hợp của trọng số ở các cạnh liên quan. Nếu sau khi đổi chỗ, hàm đánh giá chất lượng tăng lên thì lời giải nhận được sẽ thay thế cho lời giải tốt nhất và dừng thủ tục tìm kiếm của lần lặp để cập nhật mùi.

Một phép hoán vị hai đỉnh cùng nhãn A được minh họa trong hình 4.4, trong đó các vetơ dóng hàng là vectơ cột, các chữ cái là nhãn của thành phần tương ứng .



Hình 4. 3. Một hoán vị cặp đỉnh có trong thủ tục Local Search

4.2.5. Kết quả thực nghiệm

Luận án tiến hành thực nghiệm so sánh ACO-MGA với hai thuật toán Greedy và thuật toán tiến hóa GAVEO về chất lượng lời giải và thời gian chạy. Dữ liệu thực nghiệm được sinh ngẫu nhiên là các tập đồ thị với các đồ thị có 20 và 50 đỉnh, số đồ thị lần lượt là 4,8,16 và 32. Các thực nghiệm cho thấy chất lượng lời giải vượt trội của ACO-MGA so với GAVEO và thuật toán Greedy. Ngoài ra, thời gian chạy của ACO-MGA cũng nhanh hơn GAVEO, và khi cho chạy 2 thuật toán ACO-MGA và GAVEO trên cùng bộ dữ liệu trong cùng khoảng thời gian thì chất lượng lời giải của ACO-MGA cũng luôn cao hơn GAVEO.

4.3. Thuật toán Memetic giải bài toán dóng hàng nhiều đồ thị

4.3.1. Lược đồ chung

Đầu tiên thuật toán khởi tạo các tham số và các kiến nhân tạo. Sau bước khởi tạo, thuật toán ACO-MGA2 thực hiện các vòng lặp theo 2 giai đoạn như mô tả trong thuật toán 4.1.

Giai đoạn đầu (áp dụng cho 70% vòng lặp đầu tiên), trong mỗi vòng lặp, các kiến xây dựng lời giải trên đồ thị cấu trúc dựa trên thông tin heuristic và vết mùi. Sau đó lời giải tốt nhất của các kiến được lựa chọn để cập nhật vết mùi theo quy tắc cập nhật mùi SMMAS, đồng thời cập nhật lại lời giải tốt nhất toàn cục.

Giai đoạn 2 của thuật toán (áp dụng cho 30% số vòng lặp cuối cùng). Trong mỗi vòng lặp, sau khi các kiến xây dựng xong các lời giải, 2 kỹ thuật tìm kiếm cục bộ được áp dụng để tìm lời giải tốt nhất của mỗi vòng lặp.

Thuật toán 4. 1: Thuật toán ACO-MGA2

*Input:*Tập các đồ thị $G = \{G_1(V_1,E_1),...,G_n(V_n,E_n)$

Output: Dóng hàng tốt nhất cho tập đồ thị G: $A \subseteq (V_1 \cup \{\perp\}) \times ... \times (V_n \cup \{\perp\})$

```
Begin
 Khởi tạo;
  while (Chưa thỏa mãn điều kiện dừng) do
        for each a ∈A do
            Kiến a xây dựng một dóng hàng cho tập các đồ thị;
        Tìm kiếm cục bộ trên lời giải tốt nhất //Chỉ áp dụng ở giai đoạn 2
              //Tìm kiếm bằng cách đổi vị trí của các đỉnh khác nhãn.
              //Tìm kiếm bằng cách đổi vị trí của các đỉnh cùng nhãn.   Cập   nhật
 vết mùi theo quy tắcSMMAS;
        Cập nhật lại lời giải tốt nhất;
  End while;
  Lưu lại lời giải tốt nhất;
 End;
```

4.3.2. Đồ thị cấu trúc

Đồ thị cấu trúc của thuật toán ACO-GMA2 được sử dụng giống như thuật toán ACO-MGA.

4.3.3. Vết mùi và thông tin heuristic

Thông tin Heuristic$\eta^i_{j,k}(a)$được tính bởi công thức4.8.

$$n^i_{j,k}(a) = \begin{cases} \dfrac{count(k,a)+1}{i} & k \text{ is a real node} \\ \dfrac{1}{n*Vmax} & k \text{ is a dummy node} \end{cases} \quad (5.8)$$

Trong đó c*ount(k,a)*là số lượng đỉnh trên véc tơ $\{a_1,...a_i\}$ có nhãn trùng với nhãn của đỉnh k trong trường hợp k là đỉnh thực, Vmax là số lượng đỉnh của đồ thị có nhiều đỉnh nhất..

4.3.4. Thủ tục bước ngẫu nhiên xây dựng một dóng hàng

Tại mỗi vòng lặp, mỗi kiến sẽ lặp lại quá trình xây dựng véc tơ$a = (a_1,..., a_n)$cho dóng hàng A như sau.

Kiến chọn ngẫu nhiên một đỉnh thực chưa được dóng hàng từ đồ thị cấu trúc làm đỉnh xuất phát. Kiến tiếp tục dựa trên thông tin heuristic và vết mùi để tuần tự xác định các đỉnh được dóng với đỉnh xuất phát trên các đồ thị ở các tầng tiếp theo. Các đỉnh này được lựa chọn một cách ngẫu nhiên với xác suất được cho bởi công thức 4.5 tương tự như thuật toán ACO-MGA.

4.3.5. Qui tắc cập nhật vết mùi

Việc cập nhật mùi của thuật toán ACO-MGA2 cải tiến so với thuật toán ACO-MGA ở điểm thuật toán ACO-MGA2 sử dụng 2 tham số ρ ở 2 giai đoạn khác nhau. Giai đoạn đầu không sử dụng tìm kiếm địa phương nên tham số ρ được thiết lập nhỏ hơn để khai thác thông tin học tăng cường, còn giai đoạn 2 khi áp dụng tìm kiếm cục bộ thì tham số này được thiết lập lớn hơn để tăng tính khám phá.

4.3.6. Thủ tục tìm kiếm cục bộ

Thủ tục tìm kiếm cục bộ thực hiện tuần tự trên đồ thị G_1 đến đồ thị G_n theo nguyên tắc tìm được kết quả tốt nhất thì dừng. Thủ tục này gồm hai kỹ thuật: *đổi các đỉnh cùng nhãn* và *đổi các đỉnh khác nhãn*.

1) *Đổi các đỉnh khác nhãn.* Đổi vị trítrên cặp vectơ dóng hàng tương ứng với mỗi cặp đỉnh khác nhãn của đồ thị G_i đang xét nếu việc đổi chỗ đó làm tăng số lượng các đỉnh cùng nhãn trên các vector dóng hàng.

2) *Đổi các đỉnh cùng nhãn.* Đổi vị trítrên cặp vectơ dóng hàng tương ứng với mỗi cặp đỉnh tcùng nhãn của đồ thị G_i đang xét nếu việc đổi vị trí đó cải thiện độ phù hợp của trọng số ở các cạnh liên quan.

Nếu sau khi đổi chỗ, hàm đánh giá chất lượng tăng lên thì lời giải nhận được sẽ thay thế cho lời giải tốt nhất lúc đó. Quá trình này được lặp lại cho đến khi tìm được lời giải tốt nhất.

Vì thủ tục tìm kiếm cục bộ tốn thời gian nên chỉ áp dụng cho giai đoạn hai, khi lời giải tốt nhất tìm được đủ tốt.

4.3.7. Các kết quả thực nghiệm

Luận án tiến hành thực nghiệm so sánh ACO-MGA2 với hai thuật toán Greedy và thuật toán tiến hóa GAVEO về chất lượng lời giải và thời gian chạy trên các bộ dữ liệu thực bao gồm 74 cấu trúc sinh ra từ cơ sở dữ liệu Cavbase.

Các thực nghiệm cho thấy chất lượng lời giải vượt trội của ACO-MGA so với GAVEO2 và Greedy. Thời gian chạy của ACO-MGA2 cũng nhanh hơn GAVEO với các bộ dữ liệu gồm 4,8 và 16 đồ thị, và chỉ chậm hơn GAVEO đối với bộ dữ liệu gồm 32 đồ thị. Tuy nhiên khi cho chạy 2 thuật toán ACO-MGA2 và GAVEO trên cùng bộ dữ liệu trong cùng khoảng thời gian thì chất lượng lời giải của ACO-MGA2 luôn cao hơn GAVEO.

4.4. Thuật toán memetic mới

4.4.1. Đồ thị cấu trúc

Đồ thị cấu trúc của thuật toán ACOTS-MGA được sử dụng giống như thuật toán ACO-MGA2.

4.4.2. Thông tin heuristic

Heuristic information $\eta^{i}_{j,k}(a)$là số điểm cạnh tính theo công thức (4.3) khi đỉnh k của đồ thị G_{i+1} được dóng với đỉnh j của đồ thị G_i

4.4.3. Thủ tục bước ngẫu nhiên xây dựng một dóng hàng

Tại mỗi vòng lặp, mỗi kiến sẽ lặp lại quá trình xây dựng các vector dóng hàng $a = (a_1, \ldots, a_n)$cho dóng hàng A như sau.

Kiến lựa chọn ngẫu nhiên một đỉnh thực ở tầng 1 là đỉnh khởi tạo. Tại các tầng tiếp theo, ký hiệu $label(\mathrm{a})$ là tập các nhãn của các đỉnh thuộc vector dóng hàng a, gọi $B_i = \{v \in G_i \mid label(\mathrm{v}) \in label(\mathrm{a})\}$ là tập các đỉnh thuộc đồ thị G_i có nhãn trùng với nhãn của các đỉnh thuộc vector dóng hàng. Trong trường hợp không có đỉnh nào có nhãn trùng với nhãn của các đỉnh đã được dóng hàng. B_i sẽ là tập các đỉnh còn lại chưa được dóng hàng. Kiến sẽ lựa chọn ngẫu nhiên 1 đỉnh trong B_i với xác suất được cho ở công thức 4.9.

Để dễ hình dung, giả sử vector dóng hàng đã được xây dựng từ đỉnh a_1của đồ thị G_1và thực hiện thủ tục bước ngẫu nhiên để phát triển đến đỉnh a_icủađồ thị $G_i$khi đó sẽ lựa chọn đỉnh thứ k thuộc đồ thị G_{i+1} với xác suất là:

$$p_{j,k}^{i} = \frac{(\tau_{j,k}^{i})^{\alpha} * [\eta_{j,k}^{i}(a)]^{\beta}}{\sum_{s \in B_{i+1}} (\tau_{j,s}^{i})^{\alpha} * [\eta_{j,s}^{i}(a)]^{\beta}} \quad (5.9)$$

Sau khi xây dựng đầy đủ vector $a=(a_1,\ldots,a_n)$,các đỉnh thực thuộc vector này sẽ bị loại bỏ khỏi đồ thị cấu trúc để tiếp tục quá trình xây dựng các vector dóng hàng cho đến khi tất cả các đỉnh đều được dóng hàng.

4.4.4. Qui tắc cập nhật vết mùi

Khác với thuật toán ACO-MGA2, việc cập nhật mùi của ACOTS-MGA được thực hiện theo các công thức 4.10 và 4.11.

$$\tau_{j,k}^{i} = (1-\rho)\tau_{j,k}^{i} + \Delta_{j,k}^{i} \quad (5.10)$$

$$\Delta_{j,k}^{i} = \begin{cases} \rho * \tau_{max} & (i,j,k) \in \textit{ gbest solution} \\ \rho * \tau_{mid} & (i,j,k) \in \textit{ ibest solution} \\ \rho * \tau_{min} & \textit{otherwise} \end{cases} \quad (5.11)$$

Các tham số τ_{max},τ_{min} và $\rho \in (0,1)$ được khởi tạo tương tự như thuật toán ACO-MGA2. Trong thuật toán ACOTS-MGA chúng tôi sử dụng thêm tham số τ_{mid}để cập nhật mùi trong trường hợp lời giải mới mà các kiến tìm được là lời giải tốt nhất của vòng lặp nhưng chưa phải là lời giải tốt nhất toàn cục. Tham số này được thiết lập nhỏ hơn τ_{max} với ý nghĩa là lời giải tốt nhất toàn cục sẽ để lại lượng vết mùi lớn hơn so với lời giải tốt nhất của vòng lặp.

4.4.5. Thủ tục tìm kiếm Tabu

Trong các vòng lặp cuối của thuật toán ACOTS-MGA, thuật toán Tabu Search được áp dụng để tăng cường chất lượng lời giải.

Thủ tục tìm kiếm Tabu sẽ duyệt lần lượt các đỉnh của các đồ thị, với mỗi đồ thị sẽ thực hiện việc hoán vị các cặp đỉnh trên các vector dóng hàng. Nếu việc hoán vị này làm tăng điểm đánh giá thì lời giải tốt nhất sẽ được cập nhật bằng lời giải hiện tại. Khác với thủ tục tìm kiếm thông thường, thủ tục Tabu Search này có sử dụng một danh sách Tabu để lưu lại các bước chuyển. Các bước chuyển nằm trong danh sách Tabu sẽ không được xét lại nữa để tránh lặp lại các bước chuyển.

Một khác biệt nữa so với thuật toán ACO-MGA2 là thủ tục local search của ACO-MGA2 chỉ được gọi một lần trong mỗi vòng lặp, còn trong thuật toán ACOTS-MGA, thủ tục tìm kiếm được gọi lặp lại nhiều lần cho đến khi không cải thiện được chất lượng lời giải nữa.

4.4.6. Các kết quả thực nghiệm

Luận án tiến hành thực nghiệm so sánh ACOTS-MGA với các thuật toán Greedy, GAVEO và ACO-MGA2trên các bộ dữ liệu thực bao gồm 74 cấu trúc sinh ra từ cơ sở dữ liệu Cavbase.

Các thực nghiệm cho thấy chất lượng lời giải vượt trội của ACOTS-MGA so với các thuật toán còn lại. Thời gian chạy của ACOTS-MGA nhanh hơn GAVEO và ACO-MGA2 với các bộ dữ liệu gồm 4,8 đồ thị, và chỉ chậm hơn GAVEO và ACO-MGA2 đối với bộ dữ liệu gồm 16 và 32 đồ thị. Tuy nhiên khi cho chạy các thuật toán trên cùng bộ dữ liệu trong cùng khoảng thời gian thì chất lượng lời giải của ACOTS-MGA luôn cao hơn các thuật toán còn lại.

KẾT LUẬN

Trong thực tế, ta thường gặp rất nhiều bài toán tối ưu tổ hợp. Hiện nay để giải các bài toán này người ta thường nghiên cứu đề xuất các thuật toán để giải các bài toán này dựa trên các kỹ thuật tính toán mềm. Luận án đã trình bày các khái niệm liên quan đến bài toán tối ưu tổ hợp và các kỹ thuật tính toán mềm. Trong đó tập trung trình bày chi tiết về phương pháp tối ưu hóa đàn kiến, phương pháp được chúng tôi sử dụng chủ yếu để đề xuất các thuật toán mới.

Luận án cũng đã trình bày về 2 bài toán có ý nghĩa rất lớn trong lĩnh vực tin sinh học là bài toán dóng hàng mạng tương tác protein và bài toán dóng hàng đồng thời nhiều mạng các vị trí liên kết protein. Với việc phân tích đặc điểm của các thuật toán mới nhất giải quyết các bài toán này, chúng tôi đã đề xuất các thuật toán mới giải quyết hiệu quả các bài toán trên.

Đối với bài toán dóng hàng mạng tương tác protein, chúng tôi đề xuất các thuật toán mới theo hướng tiếp cận dóng hàng toàn cục. Thuật toán thứ nhất là thuật toán FASTAN cho phép dóng hàng nhanh và cho chất lượng lời giải tốt so với các thuật toán mới nhất hiện nay. Thuật toán này phù hợp với các mạng tương tác protein có kích thước lớn và yêu cầu thời gian giải bài toán nhanh. Tuy nhiên khi tăng thời gian chạy thuật toán thì chất lượng của FASTAN được cải thiện không nhiều. Để khắc phục nhược điểm trên của FASTAN, chúng tôi tiếp tục đề xuất thuật toán giải bài toán dóng hàng toàn cục mạng tương tác protein dựa trên phương pháp tối ưu hóa đàn kiến có tên là ACOGNA. Các kết quả thực nghiệm trên các bộ dữ liệu sinh học thực đã chứng minh những hiệu quả nổi trội của phương pháp ACOGNA so với các thuật toán trước đó theo các tiêu chuẩn GNAS, EC, tuy nhiên với tiêu chuẩn S^3 thuật toán ACOGNA còn cho chất lượng lời giải kém hơn so với thuật toán MAGNA++. Thuật toán ACOGNA++ được đề xuất sau đó cho phép thay đổi hàm mục tiêu theo các tiêu chuẩn dóng hàng khác nhau và sử dụng thuật toán kiến trong cả 2 giai đoạn xác định thứ tự các đỉnh trên đồ thị nguồn và xác định ảnh của nó trên đồ thị đích. Vì vậy cho chất lượng lời giải tốt hơn ACOGNA, ModuleAlign, MAGNA++ đối với tất cả các bộ dữ liệu.

Với bài toán dóng hàng nhiều mạng các vị trí hoạt tính protein, luận án đề xuất 3 thuật toán để giải bài toán này là thuật toán ACO-MGA, ACO-MGA2 và ACOTS-MGA. Thuật toán ACO-MGA dựa trên phương pháp tối ưu hóa đàn kiến để giải bài toán dóng hàng nhiều mạng. Các kết quả thực nghiệm dựa trên các bộ dữ liệu mô phỏng đã chứng minh hiệu quả nổi trội của thuật toán này so với các thuật toán GAVEO và thuật toán heuristic để giải bài toán này. Nghiên cứu đặc tính biến thiên vết mùi của các thuật toán ACO, trong thuật toán ACO-MGA2, chúng tôi áp dụng lược đồ memetic cho thuật toán. Trong đó vết mùi của thuật toán ACO được cập nhật theo 2 giai đoạn khác nhau. Giai đoạn đầu tham số bay hơi được thiết lập nhỏ để khai thác thông tin học tăng cường. Giai đoạn này không áp dụng tìm kiếm cục bộ. Giai đoạn 2 có sử dụng tìm kiếm cục bộ nên tham số bay hơi được thiết lập lớn hơn để tăng tính khám phá của thuật toán. Các kết quả thực nghiệm trên các bộ dữ liệu thực đã cho thấy những ưu điểm nổi trội của thuật toán mới đề xuất này so với các thuật toán trước đó. Thuật toán ACO-MGA2 có nhược điểm là khi áp dụng tìm kiếm cục bộ, việc hoán đổi vị trí giữa các đỉnh bị lặp lại trong các lần gọi khác nhau, vì vậy luận án đề xuất thuật toán ACOTS-MGA sử dụng kết hợp phương pháp ACO và tìm kiếm Tabu theo lược đồ memetic. Thuật toán Tabu search sử dụng để thay thế cho thuật toán tìm kiếm cục bộ trong ACO-MGA2 sử dụng danh sách cấm để tránh xét lại các bước chuyển đã xét trước đó. Ngoài ra trong ACOTS-MGA, còn có sự cải tiến trong cách xác định thông tin heuristic và thủ tục bước ngẫu nhiên xây dựng một dóng hàng. Các thực nghiệm trên bộ dữ liệu thực đã chứng minh những ưu điểm nổi trội của phương

pháp này so với các phương pháp đề xuất trước đó.

Các kết quả nghiên cứu đã được công bố trong 5 bài báo công bố tại các hội nghị quốc tế cũng như trong nước có phản biện, trong đó có 3 bài được đưa vào danh mục Scopus.

Trong tương lai chúng tôi sẽ tiếp tục nghiên cứu sâu hơn về các phương pháp tính toán mềm để đề xuất các thuật toán mới hiệu quả hơn cho các bài toán mang tính thời sự khác trong lĩnh vực tin sinh học và lĩnh vực mạng xã hội.

DANH MỤC CÔNG TRÌNH CỦA TÁC GIẢ

1. Trần Ngọc Hà, Đỗ Đức Đông, Hoàng Xuân Huấn, *An Efficient Ant Colony Optimization Algorithm for Multiple Graph Alignment*, Proceedings of *International Conference on Computing, Management and Telecommunications (ComManTel), 2013, Ho Chi Minh City, Vietnam*, pp.386-391, 2013. **(Scopus)**
2. Trần Ngọc Hà, Đỗ Đức Đông, Hoàng Xuân Huấn (2014), "A Novel Ant Based Algorithm for Multiple Graph Alignment"**,***Proceedings of the 2014 International Conference on Advanced Technologies for Communications*, pp. 181-186. **(Scopus)**
3. Đỗ Đức Đông, Trần Ngọc Hà, Đặng Thanh Hải, Đặng Cao Cường, Hoàng Xuân Huấn (2015), "An efficient algorithm for global alignment of protein-protein interaction networks", *Proceedings of the 2015 International Conference on Advanced Technologies for Communications*, pp. 332-336. **(Scopus)**
4. Trần Ngọc Hà, Hoàng Xuân Huấn (2015), "Một thuật toán tối ưu đàn kiến dóng hàng toàn cục mạng tương tác protein"**,***Proceedings of Fundamental and Applied IT Research Conference 2015(FAIR 2015)*, Ha Noi, Viet Nam, pp. 471-477.
5. Ha Tran Ngoc, Huan Hoang Xuan (2016), "ACOGNA: An Efficient Method for Protein-Protein Interaction Network Alignment"**,** *Proceedings of the The Eighth International Conference on Knowledge and Systems Engineering* (KSE 2016), pp. 7-12, 2016.
6. Ha Tran Ngoc, Hien Le Nhu, Huan Hoang Xuan, "A new memetic algorithm for multiple graph alignment" (Submitted).