BỘ GIÁO DỤC VÀ ĐÀO TẠO
VIỆN HÀN LÂM KHOA HỌC VÀ CÔNG NGHỆ VIỆT NAM

HỌC VIỆN KHOA HỌC VÀ CÔNG NGHỆ

NGUYỄN THỊ UYÊN

NGHIÊN CỨU MỘT SỐ BIẾN THỂ CỦA BÀI TOÁN HÔN NHÂN ỔN ĐỊNH THEO TIẾP CẬN HEURISTIC

Chuyên ngành: Khoa học máy tính

Mã số: 9 48 01 01

TÓM TẮT LUẬN ÁN TIẾN SĨ NGÀNH MÁY TÍNH

Hà Nội - 2023

Công trình được hoàn thành tại: Học viện Khoa học và Công nghệ - Viện Hàn lâm Khoa học và Công nghệ Việt Nam

Người hướng dẫn khoa học 1: PGS.TS. Hoàng Hữu Việt

Người hướng dẫn khoa học 2: PGS.TS. Nguyễn Long Giang

Phản biện 1:

Phản biện 2:

Phản biện 3:

Luận án được bảo vệ trước Hội đồng chấm luận án tiến sĩ, họp tại Học viện Khoa học và Công nghệ - Viện Hàn lâm Khoa học và Công nghệ Việt Nam vào hồi giờ, ngày tháng năm 2023.

Có thể tìm hiểu luận án tại:

- Thư viện Học viện Khoa học và Công nghệ
- Thư viện Quốc gia Việt Nam

Hà Nội - 2023

DANH MỤC CÁC CÔNG TRÌNH CÔNG BỐ

1. Các công bố sử dụng trong Luận án

[A.1]Hoang Huu Viet, **Nguyen Thi Uyen**, SeungGwan Lee,TaeChoong Chung, and Le Hong Trang, "*A Max-Conflicts based Heuristic Search for the Stable Marriage Problem with Ties and Incomplete Lists*", Journal of Heuristics (**SCIE - Q2**), vol. 27, no.3, pp. 439–458, 2021.

[A.2]Hoang Huu Viet, **Nguyen Thi Uyen**, Cao Thanh Sơn, and TaeChoong Chung: *A Heuristic Repair Algorithm for the Maximum Stable Mar- riage Problem with Ties and Incomplete Lists*, in Proceedings of the 34th Australasian Joint Conference on Artificial Intelligence 2022 (AI 2022), Sydney, Australia, Feb.2-4, 2022, pp.494-506, Lecture Notes in Artificial Intelligence 13151 (**SCOPUS**), Springer, ISBN 978-3-030- 97545-6.

[A.3] **Nguyen Thi Uyen**, Nguyen Long Giang, Nguyen Truong Thang, and Hoang Huu Viet: *A min-conflicts algorithm for maximum stable match- ings of the hospitals/residents problem with ties*, in Proceedings of the 14th International Conference on Computing and Communication Technologies (RIVF 2020), RMIT, Ho Chi Minh, Apr.6-7, 2020, pp.1- 6, Lecture Notes in Computer Science (**SCOPUS**), Springer, ISBN 978-1-7281-5377-3.

[A.4] **Nguyen Thi Uyen**, Nguyen Long Giang, Tran Xuan Sang and Hoang Huu Viet "*An efficient heuristics algorithm for solving the Student- Project Allocation with Preferences over Projects*", 24th Hội thảo Quốc gia (VNICT 2021), Thai Nguyen, Việt Nam, Dec. 13-14, pp. 1-6, 2021.

[A.5] **Nguyen Thi Uyen**, Giang L. Nguyen, Canh V. Pham, Tran Xuan Sang and Hoang Huu Viet: "*A Heuristic Algorithm for the Student-Project Allocation Problem with Lecturer Preferences over Students with Ties*, in Proceedings of the 11th International Conference on Computational Data and Social Networks (CSoNET 2022), Tampa, Florida, USA, Dec. 5-7, 2022, in Press, Lecture Notes in Computer Science (**SCOPUS**), Springer.

[B.1] **Nguyen Thi Uyen**, Giang L. Nguyen and Hoang Huu Viet, "*An effi- cient Heuristic search algorithm for the Hospitals/Residents with Ties problem*", Applied Artificial Intelligence (đang gửi tạp chí).

[B.2] **Nguyen Thi Uyen**, Nguyen Long Giang and Hoang Huu Viet "*Faster and Simpler Heuristic Algorithm for the Student-Project Allocation with Preferences over Projects*", International Journal of Fuzzy Logic and Intelligent Systems (đang gửi tạp chí).

2. Các công bố khác

[C.1] **Nguyen Thi Uyen** and Tran Xuan Sang, "*An efficient algorithm to find a maximum weakly stable matching for SPA-ST problem*, in Proceed- ings of the 21st International Conference on Artificial Intelligence and Soft Computing (ICAISC 2022), Zakopane, Poland, Jun. 18-22, 2022, in Press, Lecture Notes in Artificial Intelligence (**SCOPUS**), Springer.

[C.2] Hoang Huu Viet, **Nguyen Thi Uyen**, Cao Thanh Sơn, and Le Hong Trang, "*Một thuật toán tìm kiếm cục bộ giải bài toán phân công địa điểm thực tập cho sinh viên*", 23th Hội thảo Quốc gia (VNICT 2020), Hạ Long, Việt Nam, Nov. 5-6, pp. 271–276, 2020.

MỞ ĐẦU

1. Tính cấp thiết của đề tài luận án

Bài toán hôn nhân ổn định (*Stable Marriage Problem*, SMP) là một bài toán ghép cặp nổi tiếng được giới thiệu lần đầu tiên bởi Gale và Shapley năm 1962. Bài toán SMP gồm một tập n người nam và một tập n người nữ, trong đó mỗi người xếp hạng "thích" những người khác giới theo một thứ tự ưu tiên từ 1 đến n trong một danh sách xếp hạng. Mục đích của bài toán là tìm một phép ghép giữa nam và nữ sao cho thỏa mãn sự ổn định (*stable*) theo một tiêu chuẩn nào đó. Gần đây, bài toán SMP đã nhận được nhiều sự quan tâm của các nhà nghiên cứu trong các lĩnh vực Trí tuệ nhân tạo và Tính toán tối ưu.

- *Về mặt thực tiễn*: Năm 2012, Shapley và Roth đã được trao giải thưởng Nobel kinh tế vì các thành tựu đạt được dựa trên các mô hình xuất phát từ bài toán SMP trong lĩnh vực quản lý thị trường chứng khoán tại Mỹ. Ngoài ra, bài toán SMP có nhiều ứng dụng trong thực tế như: *(i)* bài toán phân bổ sinh viên thực tập tới các doanh nghiệp; *(ii)* bài toán phân công giảng viên hướng dẫn sinh viên thực hiện các đề tài; *(iii)* bài toán phân bổ nhà ở cho dân cư; *(iv)* bài toán tối ưu hóa các yêu cầu dịch vụ của người dùng Internet tới các nhà mạng viễn thông. Vì vậy, cần nghiên cứu bài toán hôn nhân ổn định và các biến thể để tìm ra các giải pháp tối ưu cho vấn đề ghép cặp trong các ứng dụng thực tế.

-*Về mặt khoa học*: Một số biến thể của bài toán SMP đã được đề xuất gần đây như: *(i)* bài toán hôn nhân ổn định với thứ tự ưu tiên ngang bằng (SMT), *(ii)* bài toán hôn nhân ổn định với danh sách không đầy đủ (SMI), và *(iii)* bài toán hôn nhân ổn định với thứ tự ưu tiên ngang bằng và không đầy đủ (SMTI). Trong bài toán SMT và SMTI, với việc xuất hiện xếp hạng ngang bằng trong danh sách xếp hạng, Irving và cộng sự (2002) đã chỉ ra có ba điều kiện về phép ghép ổn định được xem xét bao gồm: ổn định yếu (*weakly stable*), ổn định mạnh (*strongly stable*) và siêu ổn định (*super-stable*). Các tác giả đã chứng minh rằng một phép ghép ổn định yếu luôn tồn tại, trong khi phép ghép ổn định mạnh và siêu ổn định có thể không tồn tại với mọi thể hiện của các bài toán SMT và SMTI. Ngoài ra, các tác giả cũng đã chứng minh, việc tìm một phép ghép ổn định yếu với kích thước tối đa (MAX) là một bài toán NP-khó. Trong nghiên cứu này, luận án tập trung nghiên cứu tìm ra một phép ghép ổn định yếu với kích thước tối đa (MAX), do vậy để đơn giản, luận án gọi phép ghép ổn định yếu là phép ghép ổn định. Để nghiên cứu bài toán này, cần nghiên cứu các thuật toán heuristic để tìm một phép ghép ổn định yếu với kích thước tối đa cho MAX-SMTI và các biến thể.

2. Mục tiêu nghiên cứu: Mục tiêu nghiên cứu của luận án tập vào hai nội

dung chính như sau:
- Nghiên cứu tổng quan về bài toán hôn nhân ổn định và các biến thể.
- Nghiên cứu và đề xuất các thuật toán heuristic để giải quyết các bài toán MAX-SMTI, MAX-HRT và MAX-SPA.

3. Bố cục của luận án: Bố cục của Luận án gồm phần mở đầu và bốn chương nội dung, phần kết luận và danh mục các tài liệu tham khảo.

- ***Chương 1. Tổng quan về bài toán hôn nhân ổn định***. Trong chương này, luận án trình bày tổng quan cơ sở lý thuyết và tình hình nghiên cứu của bài toán hôn nhân ổn định và các biến thể.
- ***Chương 2. Đề xuất các thuật toán giải bài toán MAX-SMTI***. Trong chương này, luận án đề xuất 02 thuật toán để giải bài toán MAX-SMTI. Các kết quả được công bố tại tạp chí chuyên ngành SCIE và hội thảo quốc tế có chỉ số SCOPUS.
- ***Chương 3. Đề xuất các thuật toán giải bài toán MAX-HRT***. Trong chương này, luận án đề xuất 02 thuật toán để giải bài toán MAX-HRT. Các kết quả được công bố tại hội thảo quốc tế có chỉ số SCOPUS.
- ***Chương 4. Đề xuất các thuật toán giải bài toán MAX-SPA***. Trong chương này, luận án đề xuất 02 thuật toán để giải bài toán MAX-SPA. Các kết quả được công bố hội thảo trong nước và quốc tế có chỉ số SCOPUS.

CHƯƠNG 1. TỔNG QUAN VỀ BÀI TOÁN HÔN NHÂN ỔN ĐỊNH

1.1. Bài toán hôn nhân ổn định

Một thể hiện của bài toán SMP kích thước n, ký hiệu là I, bao gồm một tập gồm n người nam và người nữ. Mỗi người có một danh sách xếp hạng, trong đó mỗi người sẽ xếp hạng ưu tiên người khác giới theo một thứ tự nhất định. Cho một ví dụ của bài toán SMP gồm 8 nam và 8 nữ được thể hiện trong Bảng 1.1. Gale và Shapley (1962) đã trình bày một thuật toán nổi tiếng gọi

Bảng 1.1: *Một thể hiện của bài toán* SMP

Danh sách xếp hạng của $m_i \in \mathcal{M}$	Danh sách xếp hạng của $w_j \in \mathcal{W}$
m_1: w_4 w_3 w_1 w_5 w_2 w_6 w_8 w_7	w_1: m_4 m_7 m_3 m_8 m_1 m_5 m_2 m_6
m_2: w_2 w_8 w_4 w_5 w_3 w_7 w_1 w_6	w_2: m_5 m_3 m_4 m_2 m_1 m_8 m_6 m_7
m_3: w_5 w_8 w_1 w_4 w_2 w_3 w_6 w_7	w_3: m_2 m_8 m_6 m_4 m_3 m_7 m_5 m_1
m_4: w_6 w_4 w_3 w_2 w_5 w_8 w_1 w_7	w_4: m_5 m_6 m_8 m_3 m_4 m_7 m_1 m_2
m_5: w_6 w_5 w_4 w_8 w_1 w_7 w_2 w_3	w_5: m_1 m_8 m_5 m_2 m_3 m_6 m_4 m_7
m_6: w_7 w_4 w_2 w_5 w_6 w_8 w_1 w_3	w_6: m_8 m_6 m_2 m_5 m_1 m_7 m_4 m_3
m_7: w_8 w_5 w_6 w_3 w_7 w_2 w_1 w_4	w_7: m_5 m_2 m_8 m_3 m_6 m_4 m_7 m_1
m_8: w_4 w_7 w_1 w_3 w_5 w_8 w_2 w_6	w_8: m_4 m_5 m_7 m_1 m_6 m_2 m_8 m_3

là thuật toán Gale-Shapley để tìm một nghiệm tối ưu cho tập nam trong thời gian $O(n^2)$. Ngoài ra, một số thuật toán khác cũng đã được đề xuất như thuật toán xấp xỉ, thuật toán tìm kiếm heuristic, và một số thuật toán khác. Tuy nhiên, bài toán SMP ít được ứng dụng trong thực tế bởi các yêu cầu ràng buộc nghiêm ngặt của danh sách xếp hạng, tức là mỗi người nam phải xếp hạng đầy đủ các người nữ và ngược lại. Do đó, những năm gần đây một số biến thể mới của bài toán SMP đã được đề xuất và ứng dụng quan trọng trong thực tế.

1.2. Biến thể của bài toán hôn nhân ổn định

Biến thể đầu tiên của bài toán SMP được gọi là bài toán hôn nhân ổn định với danh sách xếp hạng ngang bằng (*Stable Marriage Problem with Ties*, SMT), nghĩa là một người có thể xếp hạng ưu tiên hai hay nhiều người khác giới theo một thứ tự bằng nhau. Biến thể khác của SMP được gọi là bài toán hôn nhân ổn định với danh sách xếp hạng không đầy đủ (*Stable Marriage Problem with Incomplete*, SMI), nghĩa là mỗi người chỉ xếp hạng ưu tiên với những người khác giới trong danh sách xếp hạng của họ. Nếu kết hợp hai biến thể SMT và SMI, gọi là bài toán hôn nhân ổn định với danh sách xếp hạng ngang bằng và không đầy đủ (*Stable marriage problem with Ties and*

Incomplete lists, SMTI). Mục tiêu của bài toán SMTI là tìm một phép ghép không chỉ ổn định mà còn có nhiều nhất số người nam được ghép, hay còn gọi là bài toán MAX-SMTI. Manlove và cộng sự (2008) đã chứng minh rằng MAX-SMTI là một bài toán NP-khó, ngay cả khi danh sách xếp hạng có thứ tự xếp hạng ưu tiên ngang bằng từ phía người nam hoặc người nữ. Vì vậy, luận án tập trung nghiên cứu bài toán SMTI và các biến thể của nó.

Định nghĩa 1.1 (Thể hiện SMTI)**.** *Một thể hiện I (instance) của bài toán* SMTI *kích thước n bao gồm một tập $\mathcal{M} = \{m_1, m_2, \cdots, m_n\}$ người nam và một tập $\mathcal{W} = \{w_1, w_2, \cdots, w_n\}$ người nữ, trong đó mỗi người có thể xếp hạng ưu tiên một số người khác giới theo các thứ tự có thể ngang bằng trong danh sách xếp hạng.*

Ký hiệu $rank(m_i, w_j)$ là thứ hạng của $w_j \in \mathcal{W}$ trong danh sách xếp hạng của $m_i \in \mathcal{M}$ và $rank(w_j, m_i)$ là thứ hạng của $m_i \in \mathcal{M}$ trong danh sách xếp hạng của $w_j \in \mathcal{W}$. Nếu $m_i \in \mathcal{M}$ thực sự thích $w_j \in \mathcal{W}$ hơn $w_k \in \mathcal{W}$ nghĩa là $rank(m_i, w_j) < rank(m_i, w_k)$ và nếu $m_i \in \mathcal{M}$ thích $w_j \in \mathcal{W}$ và $w_k \in \mathcal{W}$ như nhau nghĩa là $rank(m_i, w_j) = rank(m_i, w_k)$, tương tự các ký hiệu này cũng được dùng từ phía người nữ.

Định nghĩa 1.2 (Cặp chấp nhận)**.** *Một cặp $(m_i, w_j) \in \mathcal{M} \times \mathcal{W}$ là một cặp chấp nhận, nếu $rank(m_i, w_j) > 0$ và $rank(w_j, m_i) > 0$.*

Định nghĩa 1.3 (Phép ghép)**.** *Một phép ghép là một tập $M = \{(m_i, w_j) \in \mathcal{M} \times \mathcal{W} \mid rank(m_i, w_j) > 0$ và $rank(w_j, m_i) > 0\}$, trong đó mỗi $m_i \in \mathcal{M}$ chỉ được ghép với một $w_j \in \mathcal{W}$ và ngược lại. Nếu $(m_i, w_j) \in M$, thì m_i và w_j bạn ghép của nhau, ký hiệu $M(m_i) = w_j$ và $M(w_j) = m_i$. Nếu $m_i \in \mathcal{M}$ không được ghép trong M, thì m_i được gọi là độc thân và ký hiệu $M(m_i) = \varnothing$. Tương tự, nếu $w_j \in \mathcal{W}$ không được ghép trong M, thì w_j được gọi là độc thân và ký hiệu $M(w_j) = \varnothing$.*

Định nghĩa 1.4 (Cặp chặn)**.** *Một cặp $(m_i, w_j) \in \mathcal{M} \times \mathcal{W}$ là một cặp chặn cho một phép ghép M nếu thỏa mãn các điều kiện sau:*

1. *$rank(m_i, w_j) > 0$ và $rank(w_j, m_i) > 0$;*
2. *$M(m_i) = \varnothing$ hoặc $rank(m_i, w_j) < rank(m_i, M(m_i))$;*
3. *$M(w_j) = \varnothing$ hoặc $rank(w_j, m_i) < rank(w_j, M(w_j))$.*

Định nghĩa 1.5 (Cặp chặn vượt trội)**.** *Một cặp chặn $(m_i, w_j) \in \mathcal{M} \times \mathcal{W}$ vượt trội một cặp chặn $(m_i, w_k) \in \mathcal{M} \times \mathcal{W}$ nếu $rank(m_i, w_j) < rank(m_i, w_k)$.*

Định nghĩa 1.6 (Cặp chặn trội nhất). *Một cặp chặn $(m_i, w_j) \in \mathcal{M} \times \mathcal{W}$ gọi là cặp chặn trội nhất theo xếp hạng của m_i nếu không tồn tại cặp chặn (m_i, w_k) nào mà $rank(m_i, w_k) < rank(m_i, w_j)$.*

Định nghĩa 1.7 (Phép ghép ổn định). *Một phép ghép M là ổn định nếu không tồn tại bất kỳ cặp chặn $(m_i, w_j) \in \mathcal{M} \times \mathcal{W}$ cho M, ngược lại M được gọi là không ổn định.*

Định nghĩa 1.8 (Kích thước phép ghép). *Kích thước phép phép ghép ổn định M là tổng số cặp $(m_i, w_j) \in M$ và được ký hiệu là $|M|$.*

Định nghĩa 1.9 (Phép ghép hoàn chỉnh). *Một phép ghép ổn định M gọi là hoàn chỉnh nếu $|M| = n$, ngược lại M được gọi là không hoàn chỉnh.*

Một thể hiện của bài toán SMTI bao gồm 8 người nam và 8 người nữ được mô tả trong Bảng 1.2.

Bảng 1.2: *Ví dụ của một thể hiện* SMTI

Danh sách xếp hạng của $m_i \in \mathcal{M}$	Danh sách xếp hạng của $w_j \in \mathcal{W}$
m_1: w_1	w_1: m_1 $(m_5$ $m_6)$
m_2: w_5 $(w_3$ w_4 $w_6)$ $(w_7$ $w_8)$	w_2: $(m_3$ m_5 $m_6)$
m_3: w_4 $(w_2$ $w_5)$	w_3: m_6 $(m_7$ $m_8)$ m_5 m_2
m_4: $(w_5$ $w_6)$ w_8 w_7	w_4: m_3 $(m_2$ m_6 $m_7)$ m_5
m_5: $(w_1$ $w_3)$ $(w_4$ $w_5)$ w_2	w_5: $(m_5$ m_7 $m_8)$ $(m_3$ $m_4)$ m_2
m_6: $(w_4$ $w_7)$ w_1 $(w_2$ w_3 $w_8)$	w_6: m_2 m_7 $(m_4$ $m_8)$
m_7: w_4 w_6 $(w_3$ w_5 $w_7)$	w_7: $(m_2$ $m_6)$ m_7 m_4
m_8: w_5 w_6 w_3	w_8: $(m_2$ $m_4)$ m_6

Một số thuật toán đã được đề xuất để giải bài toán MAX-SMTI như sau:

i) Thuật toán xấp xỉ: Các thuật toán xấp xỉ đã đề xuất nhìn chung đã giải quyết khá tốt cho bài toán MAX-SMTI với chất lượng nghiệm tương đối tốt với tỷ lệ gần đúng tăng dần lên so với các thuật toán đề xuất trước đó. Hiện tại, thuật toán xấp xỉ có tỷ lệ gần đúng tốt nhất là $3/2$.

ii) Thuật toán heuristic: Các phương pháp lập trình ràng buộc để giải bài toán SMTI đã được một số nhà nghiên cứu đề xuất. Gent và Prosser (2002) đã đề xuất một nghiên cứu thực nghiệm về bài toán MAX-SMTI. Đầu tiên, các tác giả đề xuất một thuật toán để tạo ngẫu nhiên thể hiện SMTI với ba tham số (n, p_1, p_2), trong đó n là số lượng người nam hay nữ, p_1 là xác suất không đầy đủ và p_2 là xác suất của ưu tiên xếp hạng ngang bằng. Sau đó, họ áp dụng phương pháp lập trình ràng buộc để xem xét ảnh hưởng của các tham số p_1 và

p_2 đến chất lượng giải pháp. Gelain và cộng sự (2013) đã đề xuất thuật toán tìm kiếm cục bộ, LTIU để giải bài toán MAX-SMTI. Munera và cộng sự (2015) đã mô hình hóa bài toán SMTI và áp dụng lập trình ràng buộc để giải quyết cho bài toán MAX-SMTI.

iii) Các hướng tiếp cận khác: Ngoài ra, một số nhà nghiên cứu đã đề xuất phương pháp tiếp cận mới để giải bài toán SMTI như: sử dụng mô hình quy hoạch nguyên, sử dụng mô hình SAT model sử dụng đồ thị phân đôi, và sử dụng mô hình quy hoạch nguyên tuyến tính. Luận án này tập trung nghiên cứu các thuật toán theo hướng tiếp cận tìm kiếm heuristic để giải quyết bài toán MAX-SMTI hiệu quả hơn về thời gian và chất lượng nghiệm so với các nghiên cứu trước đó.

1.3. Một số ứng dụng mở rộng của bài toán SMTI

1.3.1. Bài toán Hospitals/Residents with Ties

Bài toán HRT là mở rộng của bài toán SMTI. Một thể hiện HRT kích thước $n \times m$ gồm một tập $\mathcal{R} = \{r_1, r_2, \cdots, r_n\}$ các sinh viên và một tập $\mathcal{H} = \{h_1, h_2, \cdots, h_m\}$ các doanh nghiệp, trong đó mỗi $r_i \in \mathcal{R}$ xếp hạng một tập con của $\mathcal{H}$ theo một thứ tự ưu tiên không nghiêm ngặt, mỗi $h_i \in \mathcal{H}$ xếp hạng một tập con của $\mathcal{R}$ theo một thứ tự ưu tiên không nghiêm ngặt và mỗi $h_j \in \mathcal{H}$ có một số lượng tối đa $c_j \in \mathbb{Z}^+$ sinh viên có thể nhận thực tập. Mục tiêu của bài toán HRT là tìm một phép ghép ổn định $M = \{(r_i, h_j) \in \mathcal{R} \times \mathcal{H}\}$ với kích thước lớn nhất, tức là bài toán MAX-HRT. Để mở rộng tính ứng dụng của bài toán HRT trong các vấn đề hỗ trợ tìm kiếm việc làm cho sinh viên vừa mới ra trường nói chung và các bác sĩ thực tập nói riêng, luận án xem xét bài toán HRT dưới dạng một bài toán ứng dụng phân bổ sinh viên thực tập tới các doanh nghiệp và do vậy luận án xem các *sinh viên* như là các bác sĩ thực tập và các *doanh nghiệp* như là các bệnh viện. Mục tiêu của bài toán HRT là tìm một phép ghép ổn định giữa sinh viên và các doanh nghiệp sao cho tất cả các sinh viên đều có thể nhận được vị trí thực tập phù hợp tại các doanh nghiệp, gọi là bài toán MAX-HRT.

Một thể hiện HRT gồm 8 sinh viên và 5 doanh nghiệp được mô tả trong Bảng 1.3.

Bảng 1.3: *Ví dụ một thể hiện* HRT

Danh sách xếp hạng của $r_i \in \mathcal{R}$	Danh sách xếp hạng của $h_j \in \mathcal{H}$
r_1: h_1 h_3 h_2	h_1: r_3 (r_7 r_5 r_2) r_4 r_6 r_1
r_2: h_1 (h_5 h_4) h_3	h_2: r_5 r_6 (r_3 r_4) r_1
r_3: h_1 h_5 h_2	h_3: (r_5 r_2) r_6 r_1 r_7
r_4: h_1 (h_2 h_4)	h_4: r_8 r_2 r_4 r_7
r_5: h_3 h_1 h_2	h_5: r_3 (r_7 r_6 r_8) r_2
r_6: (h_3 h_2) h_1 h_5	
r_7: h_3 h_4 h_5 h_1	
r_8: h_5 h_4	
Số sinh viên tối đa của $h_j \in \mathcal{H}$: $c_1 = 2$, $c_2 = 3$, $c_3 = c_4 = c_5 = 1$	

1.3.2. Bài toán Student-Project Allocation

Bài toán SPA đã được các nhà khoa học quan tâm nghiên cứu vì ứng dụng quan trọng trong các trường đại học. Một số biến thể của bài toán SPA đã được giới thiệu như SPA-S, SPA-P và SPA-ST. Gần đây, các biến thể SPA-P và SPA-ST được cộng đồng nghiên cứu quan tâm nhiều trong các ứng dụng thực tế. Vì vậy, luận án tập trung nghiên cứu hai biến thể của bài toán SPA là bài toán SPA-P. Mục tiêu của bài toán là tìm một phép ghép ổn định với kích thước tối đa, tức là có nhiều nhất số sinh viên nhận được đề tài thỏa mãn các điều kiện ràng buộc về số lượng sinh viên tối đa của giảng viên và đề tài (MAX-SPA).

Một thể hiện SPA-P gồm 5 sinh viên, 2 giảng viên và 5 đề tài được chỉ ra trong Bảng 1.4.

Bảng 1.4: *Ví dụ một thể hiện* SPA-P

Danh sách xếp hạng của $s_i \in \mathcal{S}$	Danh sách xếp hạng của $l_k \in \mathcal{L}$
s_1: p_1 p_3 p_4	l_1: p_1 p_2 p_3
s_2: p_5 p_1	l_2: p_4 p_5
s_3: p_2 p_5	
s_4: p_4 p_2	
s_5: p_5	
Số sinh viên tối đa của $p_j \in \mathcal{P}$: $c_1 = c_2 = c_3 = c_4 = 1, c_5 = 2$	
Số sinh viên tối đa của $l_k \in \mathcal{L}$: $d_1 = 3$, $d_2 = 2$	

Một thể hiện SPA-ST gồm 7 sinh viên, 3 giảng viên và 8 đề tài được chỉ ra trong Bảng 1.5.

Bảng 1.5: *Ví dụ một thể hiện* SPA-ST

Danh sách xếp hạng của $s_i \in \mathcal{S}$	Danh sách xếp hạng của $l_k \in \mathcal{L}$
s_1: $(p_1\ p_7)$	l_1: $(s_7\ s_4)\ s_1\ s_3\ (s_2\ s_5)\ s_6$
s_2: $p_1\ p_3\ p_5$	l_2: $s_3\ s_2\ s_7\ s_5$
s_3: $(p_2\ p_1)\ p_4$	l_3: $(s_1\ s_7)\ s_6$
s_4: p_2	
s_5: $p_1\ p_4$	l_1 đề xuất p_1, p_2, p_3
s_6: $p_2\ p_8$	l_2 đề xuất p_4, p_5, p_6
s_7: $(p_5\ p_3)\ p_8$	l_3 đề xuất p_7, p_8
Số sinh viên tối đa của $p_j \in \mathcal{P}$: $c_1 = 2, c_j = 1, (2 \leq j \leq 8)$	
Số sinh viên tối đa của $l_k \in \mathcal{L}$: $d_1 = 3, d_2 = 2, d_3 = 2$	

1.3.3. Các nghiên cứu liên quan

Một số thuật toán xấp xỉ khác nhau đã được đề xuất để giải quyết bài toán MAX-HRT trong các tài liệu với xấp xỉ tốt nhất là $3/2$. Ngoài ra, một số hướng tiếp cận khác dựa trên hướng tiếp cận heuristic, quy hoạch nguyên đã được đề xuất để giải quyết bài toán MAX-HRT. Munera và cộng sự (2015) chuyển bài toán HRT thành bài toán SMTI và áp dụng thuật toán tìm kiếm thích nghi để giải bài toán MAX-HRT. Tuy nhiên, các thuật toán đã đề xuất chưa giải quyết hiệu quả về chất lượng nghiệm và thời gian thực hiện cho bài toán HRT với kích thước lớn. Do đó, luận án này tập trung nghiên cứu và đề xuất các thuật toán theo hướng tìm kiếm heuristic để giải quyết bài toán MAX-HRT với kích thước lớn. Gần đây nhất, bài toán SPA được quan tâm theo nghĩa tìm một phép ghép ổn định yếu với kích thước tối đa hoặc phép ghép siêu ổn định. Cooper và Manlove (2018) đã đề xuất một thuật toán xấp xỉ với tỷ lệ $3/2$ để tìm một phép ghép ổn định yếu với kích thước tối đa cho bài toán SPA-ST.

CHƯƠNG 2. ĐỀ XUẤT CÁC THUẬT TOÁN GIẢI BÀI TOÁN MAX-SMTI

2.1. Đề xuất thuật toán Max-Conflicts

Algorithm 2.1: Thuật toán MCS

Input: - Thể hiện I của SMTI.
- Xác suất nhỏ, p.
- Bước lặp tối đa, max_iters.

Output: Phép ghép ổn định, M.

function `Main(`I`)`
  Khởi tạo ngẫu nhiên M;
  $M_{best} := M$;
  $f_{best} := n$;
  iter := 0;
  while $(iter \leq max\_iters)$ **do**
    $X :=$ `Find_UBPs(`M`)`;
    if $(X = \emptyset)$ **then**
      if $(f_{best} > f(M))$ **then**
        $M_{best} := M$;
        $f_{best} := f(M)$;
      if $(f_{best} > 0)$ **then**
        $M :=$ `Escape_Local_Minima(`M`)`;
        continue;
      else
        break;
    for (*mỗi* $m_i \in X$) **do**
      $ubp(w_j) := ubp(w_j) + 1$, với $(m_i, w_j) \in X$;
    for (*mỗi* $m_i \in X$) **do**
      $h(m_i) := n * ubp(w_j) - rank(w_j, m_i)$, với $(m_i, w_j) \in X$;
    if *(với xác suất nhỏ p)* **then** $m_j :=$ một $m_i \in X$ ngẫu nhiên;
    else $m_j := \text{argmax}(h(m_i)), \forall m_i \in X$;
    Xóa cặp chặn $(m_j, w_k) \in X$;
    $iter := iter + 1$;
  return M_{best};
end function

Phần này đề xuất một thuật toán tìm kiếm heuristic dựa trên các xung đột tối đa (Max-Conflicts based heuristic search, viết tắt MCS) được mô tả trong Thuật toán 2.1 cho bài toán MAX-SMTI. Ý tưởng chính của MCS là bắt đầu từ một phép ghép ngẫu nhiên, thuật toán sẽ tìm một tập các cặp chặn vượt trội cặp chặn trội nhất và định nghĩa một hàm heuristic để chọn một cặp chặn cặp chặn trội nhất tốt nhất và loại bỏ khỏi phép ghép hiện tại. MCS lặp lại cho đến khi tìm được phép ghép hoàn chỉnh hoặc đạt đến số lặp tối đa. Ký hiệu $X = \{(m_i, w_j) \mid m_i \in \mathcal{M}, w_j \in \mathcal{W}\}$ là một tập hợp các cặp chặn vượt trội cặp chặn trội nhất theo quan điểm của nam cho phép ghép không ổn định M. Với mỗi $(m_i, w_j) \in X$, không tồn tại cặp chặn nào vượt trội (m_i, w_j) theo quan điểm của m_i, nghĩa là m_i xuất hiện một lần, trong khi w_j có thể xuất hiện nhiều lần trong X. Gọi $ubp(w_j)$ là số cặp chặn trội nhất được tạo bởi $w_j \in X$, thuật toán định nghĩa một hàm heuristic như sau:

$$h(m_i) = n \times ubp(w_j) - rank(w_j, m_i), \forall (m_i, w_j) \in X. \tag{2.1}$$

Algorithm 2.2: Tìm tập các cặp chặn trội nhất, X

Input: Phép ghép M.
Output: Tập các cặp chặn trội nhất X của phép ghép M.

```
1.  function Find_UBPs(M)
2.      X := ∅;
3.      for (mỗi m_i ∈ M) do
4.          w_j := M(m_i);
5.          while (∃w_k ∈ W|rank(m_i, w_k) > 0) do
6.              w_k := argmin(rank(m_i, w_k) > 0);
7.              if (rank(m_i, w_k) = rank(m_i, w_j)) then
8.                  break;
9.              if ((m_i, w_k) là một cặp chặn) then
10.                 X := X ∪ (m_i, w_k);
11.                 break;
12.             else
13.                 Xóa w_k trong danh sách xếp hạng của m_i;
14.     return X;
15. end function
```

Algorithm 2.3: Vượt qua cục bộ địa phương

Input: Phép ghép M.
Output: Phép ghép M.

function `Escape_Local_Minima(`M`)`
  if *(với xác suất* $p \leq 0.5$*)* **then**
    $U := \{m_i \mid M(m_i) = \varnothing\}$;
    Chọn ngẫu nhiên một $m_j \in U$;
    for (*mỗi* $w_k \in$ *danh sách xếp hạng của* m_j) **do**
      if $(M(w_k) \neq \varnothing)$ **then**
        Xóa cặp $(M(w_k), w_k)$ thành hai người độc thân, $M(w_k)$ và w_k;
  else
    $V := \{w_i \mid M(w_i) = \varnothing\}$;
    Chọn ngẫu nhiên một $w_j \in V$;
    for (*mỗi* $m_k \in$ *danh sách xếp hạng của* w_j) **do**
      if $(M(m_k) \neq \varnothing)$ **then**
        Xóa cặp $(m_k, M(m_k))$ thành hai người độc thân, m_k và $M(m_k)$;
  return M;
end function

Đầu tiên, MCS tìm một tập hợp X của các cặp chặn trội nhất cho M bằng Thuật toán 2.2. Thứ hai, MCS kiểm tra nếu X là rỗng, tức là M là một phép ghép ổn định. Nếu phép ghép tìm được không phải là phép ghép hoàn chỉnh thì MCS gọi Thuật toán 2.3 để vượt qua điểm tối thiểu cục bộ và thực hiện vòng lặp tiếp theo, ngược lại, MCS trả về một phép ghép hoàn chỉnh. Thứ ba, MCS đếm số lượng các cặp chặn trội nhất, $ubp(w_j)$, được tạo bởi mỗi $w_j \in X$ và xác định các giá trị heuristic, $h(m_i)$, cho mọi $m_i \in X$. Thứ tư, MCS lấy ngẫu nhiên một $m_j \in X$ với xác suất p nhỏ hoặc lấy một $m_j \in X$ tương ứng với giá trị lớn nhất $h(m_j)$. Cuối cùng, MCS loại bỏ cặp chặn trội nhất $(m_j, M(m_j))$ cho M để nhận được một phép ghép mới. MCS thực hiện lặp lại cho đến khi tìm được phép ghép hoàn chỉnh hoặc đạt đến số bước lặp tối đa.

Các thuật toán này được thực hiện bằng ngôn ngữ lập trình Matlab R2017a trên môi trường một máy tính cá nhân với cấu hình Core i7-8550U CPU 1.8 GHz và 16 GB RAM hệ điều hành Windows-10. Các kết quả được chỉ ra rằng thuật toán MCS vượt trội so với LTIU và AS về thời gian thực hiện và chất lượng nghiệm tìm được cho bài toán MAX-SMTI kích thước lớn.

Algorithm 2.4: Thuật toán HR

Input: - Thể hiện I của SMTI.
- Bước lặp tối đa, max_iters.

Output: Phép ghép ổn định, M.

function `Main(`I`)`
  for (*mỗi* $m_i \in \mathcal{M}$) **do**
    $M(m_i) := \varnothing$;
    $a(m_i) := 1$; ▷ gán m_i hoạt động
    $c(m_i) := 0$; ▷ gán biến đếm cho m_i bằng 0
  $iter := 1$;
  while $iter \leq max\_iters$ **do**
    $m_i :=$ một người nam hoạt động, i.e., $a(m_i) = 1$;
    if $\nexists a(m_i) = 1$ **then**
      if $|M| = n$ **then** break;
      $iter := iter + 1$;
      $M :=$ `Improve(`M`)`;
      continue;
    if $(\nexists w_j \in \mathcal{W} | rank(m_i, w_j) > 0)$ **then**
      $a(m_i) := 0$; ▷ gán m_i không hoạt động
      $c(m_i) := c(m_i) + 1$; ▷ tăng biến đếm của m_i
      continue;
    if *tồn tại một người nữ độc thân* w_j *người mà* m_i *thích nhất* **then**
      $M(m_i) := w_j$;
      $a(m_i) := 0$;
    else
      $w_j :=$ một người nữ người mà m_i thích nhất;
      $m_k := M(w_j)$;
      if *tồn tại một người nữ độc thân* w_t *mà* $rank(m_k, w_t) = rank(m_k, w_j)$ **then**
        `repair(`m_i, m_k`)`;
      if $M(m_i) = \varnothing$ *và* $rank(w_j, m_i) < rank(w_j, m_k)$ **then**
        `repair(`m_i, m_k`)`;
        $rank(m_k, w_j) := 0$;
      else
        $rank(m_i, w_j) := 0$;
  return M;
end function

Algorithm 2.5: Cải tiến kích thước của phép ghép ổn định M

Input: Phép ổn định, M.
Output: Phép ghép M.

function `Improve`(M)
  for *mỗi người nam* $m_i \in M$ *sao cho* $M(m_i) = 0$ **do**
    Khôi phục lại danh sách xếp hạng của m_i;
    $X := \{\}$;
    for *mỗi* $w_j \in$ *danh sách xếp hạng của* m_i **do**
      $m_k := M(w_j)$;
      if $rank(m_i, w_j) \leq rank(m_k, w_j)$ *hoặc* $rank(w_j, m_i) = rank(w_j, m_k)$ **then**
        $X := X \cup \{w_j\}$;
    if $X = \emptyset$ **then** continue;
    for *mỗi* $w_j \in X$ **do**
      $m_k := M(w_j)$;
      $k :=$ số lượng w_t sao cho $rank(m_k, w_t) = rank(m_k, w_j)$;
      $h(w_j) :=$ $1/k + (rank(w_j, m_i) - rank(w_j, m_k)) \times (1 - c(m_k))$;
    $w_j := argmin(h(w_j)), \forall w_j \in X$;
    `repair`(m_i, m_k), với $m_k := M(w_j)$;
    $rank(m_k, w_j) := 0$;
  return M;
end function

2.2. Đề xuất thuật toán Heuristic-Repair

Thuật toán HR được đề xuất bao gồm thuật toán GS để tìm một phép ghép ổn định và một hàm heuristic để cải thiện kích thước của phép ghép tìm được bởi GS cho một thể hiện của SMTI được mô tả trong Thuật toán 2.4. Đầu tiên, HR tìm một phép ghép ổn định bằng cách cải tiến ý tưởng của thuật toán GS. Sau đó, thuật toán HR áp dụng lại thuật toán GS cải tiến. Nếu một phép ổn định tìm được mà chưa đạt kích thước tối đa, thì HR gọi Thuật toán 2.5 để cải thiện kích thước của phép ghép M bằng cách đề xuất một hàm heuristic. Thuật toán HR kết thúc khi tìm được một phép ghép ổn định với kích thước tối đa hoặc đạt đến số lần lặp tối đa. Các thuật toán HR, GSA2 và MCS thực hiện trên phần mềm Matlab R2017b trên máy tính cá nhân có CPU Core i7-8550U 1,8 GHz và RAM 16 GB, chạy trên Windows 10. Các kết quả thực nghiệm chỉ ra rằng thuật toán HR hiệu quả hơn thuật toán xấp xỉ GSA2 và MCS về thời gian thực hiện và chất lượng nghiệm cho bài toán MAX-SMTI.

CHƯƠNG 3. ĐỀ XUẤT CÁC THUẬT TOÁN GIẢI BÀI TOÁN MAX-HRT

3.1. Đề xuất thuật toán Min-Conflicts

Phần này đề xuất thuật toán Min-Conflicts, viết tắt MCA để giải bài toán MAX-HRT.

Algorithm 3.1: Thuật toán MCA

Input: - Thể hiện I của HRT.
- Xác suất nhỏ, p.
- Bước lặp tối đa, max_iters.

Output: Một phép ghép M.

```
1.  function MCA(I)
2.      Khởi tạo ngẫu nhiên M;
3.      M_best := M;
4.      f_best := n;
5.      iter := 0;
6.      while (iter ≤ max_iters) do
7.          iter := iter + 1;
8.          [f(M), X] := Find_Cost_And_UBPs(M);
9.          if (X = ∅) then
10.             if (f_best > f(M)) then
11.                 M_best := M;
12.                 f_best := f(M);
13.             if (f_best > 0) then
14.                 M := phép ghép được tạo ngẫu nhiên;
15.                 continue;
16.             else
17.                 break;
18.         if (xác suất nhỏ p) then
19.             r_j := một r_i ∈ X ngẫu nhiên;
20.         else
21.             r_j := argmin(rank(h_k, r_i)), ∀(r_i, h_k) ∈ X;
22.         Xóa cặp chặn (r_j, X(r_j));
23.     return M_best;
24. end function
```

Thuật toán MCA được mô tả trong Thuật toán 3.1. Bắt đầu từ một phép ghép ngẫu nhiên M, MCA tìm một phép ghép ổn định với kích thước tối đa. Tại mỗi bước lặp, MCA gọi Thuật toán 3.2 để tìm một tập UBP, $X = \{(r_i, h_j) \in R \times H\}$ và tính giá trị của hàm $f(M) = \#nbp(M) + \#nur(M)$, trong đó $\#nbp(M)$ là số cặp UBP cho M và $\#nur(M)$ là số sinh viên chưa được ghép trong M. Nếu M không hoàn chỉnh, thuật toán sẽ bắt đầu lại một phép ghép M mới và tiếp tục lần lặp tiếp theo. Sau đó, thuật toán kiểm tra xem một xác suất p nhỏ, MCA chọn ngẫu nhiên $r_j \in X$, ngược lại, MCA chọn $r_j \in X$, sao cho $h_k \in X$ ưa thích nhất. Thuật toán lặp lại cho đến khi M_{best} là hoàn chỉnh hoặc đạt đến số lần lặp tối đa. Trong trường hợp sau, thuật toán trả về phép ghép ổn định tối đa hoặc phép ghép không ổn định.

Algorithm 3.2: Tìm giá trị hàm $f(M)$ và cặp chặn vượt trội, X

Input: Phép ghép M.
Output: Giá trị hàm, $f(M)$, tập hợp các cặp chặn UBP, X.

function `Find_Cost_And_UBPs(`M`)`
  $X := \emptyset$;
  $\#nur := 0$;
  $\#nbp := 0$;
  for ($mỗi\ r_i \in \mathcal{R}$) **do**
    $ubp := false$;
    while ($\exists h_j \in \mathcal{H} | rank(r_i, h_j) > 0$) **do**
      $h_j := argmin(rank(r_i, h_j) > 0)$;
      if ($rank(r_i, h_j) = rank(r_i, M(r_i))$) **then**
        break;
      if ((r_i, h_j) *là một cặp chặn*) **then**
        $X := X \cup (r_i, h_j)$;
        $\#nbp := \#nbp + 1$;
        $ubp := true$;
        break;
      else
        Xóa h_j trong danh sách xếp hạng của r_i;
    if (($ubp = false$) *và* (r_i *là chưa được ghép*)) **then**
      $\#nur := \#nur + 1$;
  $f(M) := \#nbp + \#nur$;
  return $(f(M), X)$;
end function

Tất cả các thuật toán thực hiện trên phần mềm Matlab 2019a và thực hiện

Algorithm 3.3: Thuật toán HS

Input: - Thể hiện I của HRT.
- Bước lặp tối đa max_iter.

Output: Phép ghép ổn định M

function `HS` (I)
  Khởi tạo phép ghép ngẫu nhiên M;
  $M_{best} := M$;
  $a(r_i) := 1, \forall r_i \in \mathcal{R}$;
  $y(r_i) := 0, \forall r_i \in \mathcal{R}$;
  $z(h_j) := 0, \forall h_j \in \mathcal{H}$;
  $rank^{\dagger}(r_i, h_j) := rank(r_i, h_j), \forall r_i \in \mathcal{R}, h_j \in \mathcal{H}$;
  $iter := 0$;
  while $iter \leq max\_iter$ **do**
    $iter := iter + 1$;
    r_i := một sinh viên đang hoạt động;
    if *($\nexists r_i$ mà $a(r_i) = 1$)* **then**
      if $|M_{best}| < |M|$ **then**
        $M_{best} := M$;
        if $(|M_{best}| = n)$ **then** break;
      M' := `Improve(`M`)`;
      if $M' = M$ **then** break;
      $M := M'$;
      continue;
    $[h_j, W(r_i)]$:= `Find_UBP_and_Wait_List(`r_i, M`)`;
    if $h_j \neq \varnothing$ **then**
      $M := M \setminus \{(r_i, h_k)\} \cup \{(r_i, h_j)\}$, với $h_k := M(r_i)$;
      $a(r_i) := 0$;
      for *mỗi* $r_t \in \mathcal{R}$ *để mà* $W(r_t) = h_k$ **do**
        $rank(r_t, h_k) := rank^{\dagger}(r_t, h_k)$;
        $a(r_t) := 1$;
      if $|M(h_j)| > c_j$ **then**
        r_w := một sinh viên có thứ tự xếp hạng ưu tiên thấp nhất;
        $M := M \setminus \{(r_w, h_j)\}$;
        $a(r_w) := 1$;
    else
      $a(r_i) := 0$;
  return M_{best};
end function

trên máy tính có cấu hình CPU Core i7-8550U 1.8 GHz và RAM 16 GB trong Windows 10. Kết quả thực nghiệm chỉ ra rằng thuật toán MCA hiệu quả hơn so với LTIU về thời gian thực hiện và chất lượng nghiệm cho bài toán MAX-HRT.

3.2. Đề xuất thuật toán Heuristic-Search

Phần này trình bày một thuật toán tìm kiếm heuristic (Heuristic Search, viết tắt HS) được chỉ ra trong Thuật toán 3.3 để giải quyết bài toán MAX-HRT. Bắt đầu từ một phép ghép ngẫu nhiên M, tại mỗi lần lặp, nếu tồn tại một r_i sao cho $a(r_i) = 1$, HS tìm một h_j và một danh sách chờ $W(r_i)$ được chỉ ra trong Thuật toán 3.4 sao cho (r_i, h_j) là một cặp UBP và $W(r_i)$ là một tập hợp các $h_k \in \mathcal{H}$ mà $rank(r_i, h_k) < rank(r_i, M(r_i))$ hoặc $rank(r_i, h_k) = rank(r_i, h_j)$. Nếu không tồn tại bất kỳ h_j, sao cho (r_i, h_j) là một cặp chặn, thuật toán gán $a(r_i) = 0$. Nếu không tồn tại sinh viên nào đang hoạt động, có nghĩa là HS tìm được phép ghép ổn định M. Nếu phép ghép ổn định tìm thấy chưa đạt kích thước tối đa, HS sử dụng Thuật toán 3.5 để thoát khỏi điểm cực bộ và tiếp tục lặp lại cho các bước tiếp theo. Thuật toán HS kết thúc khi thuật toán tìm được một phép ghép hoàn chỉnh hoặc đạt đến số lần lặp tối đa tối đa.

Algorithm 3.4: Tìm cặp UBP và danh sách chờ $W(r_i)$

Input: r_i và M.
Output: - h_j trong đó (r_i, h_j) là một UBP.
- Danh sách chờ $W(r_i)$.

```
1.  function Find_UBP_and_Wait_List(r_i, M)
2.      for h_k ∈ danh sách xếp hạng của r_i do
3.          f(h_k) := rank(r_i, h_k) + |M(h_k)|/(c_k + 1);
4.      h_j := ∅;
5.      while (∃h_k ∈ H|rank(r_i, h_k) > 0) do
6.          h_k := argmin(f(h_k) ≥ 1), ∀h_k ∈ H;
7.          if rank(r_i, h_k) = rank(r_i, M(r_i)) then
8.              break;
9.          if (r_i, h_k) là cặp chặn then
10.             h_j := h_k;
11.             break;
12.         else
13.             W(r_i) := W(r_i) ∪ h_k;
14.             rank(r_i, h_k) := 0;
15.             f(h_k) := 0;
16.     return h_j và W(r_i);
17. end function
```

Tất cả các thuật toán thực hiện trên phần mềm Matlab 2019a và thực hiện trên máy tính có cấu hình CPU Core i7-8550U 1.8GHz và RAM 16GB trong Windows 10. Các kết quả thực nghiệm chỉ ra rằng thuật toán HS hiệu quả hơn về thời gian và chất lượng nghiệm so với thuật toán HP và AS cho bài toán MAX-HRT.

Algorithm 3.5: Cải thiện kích thước phép ghép ổn định

Input: Phép ghép ổn định, M.
Output: Phép ghép, M.

function `Improve(`M`)`
  for *mỗi* $r_u \in \mathcal{R}$, *trong đó* $M(r_u) = \varnothing$ **do**
    Tìm $(r_i, h_k) \in M$ sao cho $rank(h_k, r_i) = rank(h_k, r_u)$
    if $y(r_u) \geq y(r_i)$ **then**
      $M := M \setminus \{(r_i, h_k)\} \cup \{(r_u, h_k)\}$;
      $y(r_u) := y(r_u) + 1$;
      $a(r_u) := 0$;
      $a(r_i) := 1$;
      $rank(r_u, h_k) := rank^{\dagger}(r_u, h_k)$;
  for *mỗi* $h_t \in \mathcal{H}$, *trong đó* $|M(h_t)| < c_j$ **do**
    Tìm $(r_i, h_k) \in M$ sao cho $rank^{\dagger}(r_i, h_k) = rank^{\dagger}(r_i, h_t)$
    if $z(h_t) \geq z(h_k)$ **then**
      $M := M \setminus \{(r_i, h_k)\} \cup \{(r_i, h_t)\}$;
      $z(h_t) := z(h_t) + 1$;
      $rank(r_i, h_t) := rank^{\dagger}(r_i, h_t)$;
      for *mỗi* r_w *sao cho* $W(r_w) = h_k$ **do**
        $rank(r_w, h_k) := rank^{\dagger}(r_w, h_k)$;
        $a(r_w) := 1$;
  return M;
end function

CHƯƠNG 4. ĐỀ XUẤT CÁC THUẬT TOÁN GIẢI BÀI TOÁN MAX-SPA

4.1. Đề xuất thuật toán SPA-P-heuristic giải bài toán MAX-SPA-P

Algorithm 4.1: Thuật toán SPA-P-heuristic

Input: Thể hiện I của SPA-P.
Output: Phép ghép ổn định M.

function `SPA-P-heuristic(`I`)`
  $M := \emptyset$;
  $a(s_i) := 1, \forall s_i \in \mathcal{S}$;
  $h_{l_k}(s_i) := 0, \forall l_k \in \mathcal{L}, \forall s_i \in \mathcal{S}$;
  while $\exists$ s_i *với* $a(s_i) = 1$ **do**
    if $(\nexists p_j \in \mathcal{P} | rank(s_i, p_j) > 0)$ **then**
      $a(s_i) := 0$;
      continue;
    $p_j := argmin(rank(s_i, p_j) > 0)$;
    $l_k :=$ giảng viên đề xuất đề tài p_j;
    $M := M \cup \{(s_i, p_j)\}$;
    $a(s_i) := 0$;
    $y(s_i) :=$ số lượng đề tài được xếp hạng bởi s_i;
    $h_{l_k}(s_i) := rank(l_k, p_j) + y(s_i)/(q+1)$;
    if $|M(p_j)| > c_j$ **then**
      $s_t := \text{argmax}(h_{l_k}(s_t)), \forall s_t \in M(p_j)$;
      $M := M \setminus \{(s_t, p_j)\}$;
      $rank(s_t, p_j) := 0$;
      $a(s_t) := 1$;
      $h_{l_k}(s_t) := 0$;
    if $|M(l_k)| > d_k$ **then**
      $s_t := \text{argmax}(h_{l_k}(s_t)), \forall s_t \in M(l_k)$;
      $p_z := M(s_t)$;
      $M := M \setminus \{(s_t, p_z)\}$;
      $rank(s_t, p_z) := 0$;
      $a(s_t) := 1$;
      $h_{l_k}(s_t) := 0$;
  return M;
end function

Phần này trình bày một thuật toán heuristic, gọi là SPA-P-heuristic được mô tả trong Thuật toán 4.1 để giải quyết bài toán MAX-SPA-P. Bắt đầu từ một phép $M = \emptyset$. Ở mỗi lần lặp, khi một sinh viên s_i ghép cho đề tài thích nhất p_j trong danh sách xếp hạng của s_i để tạo thành một cặp $(s_i, p_j) \in M$, nếu đề tài p_j hoặc giảng viên l_k đề xuất p_j đã nhận quá số lượng sinh viên, thì một sinh viên tùy ý $s_r \in M(p_z)$, trong đó p_z là đề tài có sinh viên được ghép và xếp hạng ưu tiên thấp nhất trong l_k, bị loại khỏi M. Luận án đề xuất một hàm heuristic như sau:

$$h_{l_k}(s_t) = rank(l_k, p_z) + y(s_t)/(q+1). \tag{4.1}$$

trong đó l_k là giảng viên đề xuất đề tài p_z và $y(s_t)$ là số lượng đề tài được xếp hạng bởi s_t. Nếu p_j đã nhận quá số lượng sinh viên tối đa, thì sinh viên tồi nhất s_t trong $M(p_j)$ sẽ bị loại khỏi M. Nếu vậy, s_t sẽ xóa p_j trong danh sách xếp hạng của s_t và s_t sẽ hoạt động trở lại. Nếu l_k đã nhận quá số lượng sinh viên tối đa, thì sinh viên tồi nhất s_t trong $M(l_k)$ sẽ bị loại khỏi M. Nếu vậy, s_t sẽ xóa p_z trong danh xếp hạng của s_t, trong đó p_z được ghép cho s_t và s_t sẽ hoạt động trở lại. Khi một sinh viên s_t bị xóa khỏi M, giá trị heuristic $h_{l_k}(s_t)$ được gán bằng 0. Thuật toán được lặp lại cho đến khi tất cả sinh viên không hoạt động và trả về phép ghép ổn định với kích thước tối đa.

Các kết quả thực nghiệm chỉ ra rằng thuật toán SPA-P-heuristic vượt trội về thời gian và chất lượng nghiệm so với hai thuật toán xấp xỉ SPA-P-approx, SPA-P-promotion và thuật toán heuristic gần đây nhất SPA-P-MCH cho bài toán MAX-SPA-P.

4.2. Đề xuất thuật toán HAG giải quyết bài toán MAX-SPA-ST

Phần này trình bày một thuật toán tìm kiếm heuristic (Heuristic Algorithm, viết tắt HAG) để giải quyết bài toán MAX-SPA-ST. Thuật toán HAG được mô tả trong Thuật toán 4.2. Bắt đầu từ một phép ghép rỗng, $M = \emptyset$. Tại mỗi lần lặp, HAG xem xét một sinh viên $s_i \in \mathcal{S}$ chưa được ghép mà danh sách xếp hạng của s_i không rỗng và xác định hàm heuristic $h(p_j)$ cho mỗi đề tài $p_j \in \mathcal{P}$ trong danh sách xếp hạng s_i, trong đó p_j được đề xuất bởi l_k để chọn đề tài tốt nhất dựa vào giá trị nhỏ nhất của hàm $h(p_j)$ như sau:

$$h(p_j) = rank(s_i, p_j) - min(d_k - |M(l_k)|, 1)/2 - (c_j - |M(p_j)|)/(2 \times c_j + 1). \tag{4.2}$$

Algorithm 4.2: Thuật toán HAG

Input: Thể hiện SPA-ST, I
Output: Phép ghép ổn định, M.

function `HAG` (I)
  $M := \emptyset$
  $v(s_i) := 0, \forall s_i \in \mathcal{S}$
  while *true* **do**
    s_i := sinh viên chưa ghép và danh sách xếp hạng của s_i chưa rỗng
    if $\nexists\, s_i$ **then**
      if $|M| = n$ **then** break
      else
        $M' :=$ `Escape` (M)
        if $M' = M$ **then** break
        $M := M'$
        continue
    for *mỗi* $p_j \in A_i$ **do**
      l_k := giảng viên hướng dẫn đề tài p_j
      $h(p_j) = rank(s_i, p_j)$-$min(d_k$-$|M(l_k)|, 1)/2$-$(c_j$-$|M(p_j)|)$ $/(2 \times c_j + 1)$
    $p_j := \text{argmin}(h(p_j) > 0), \forall p_j \in \mathcal{P}$
    l_k := giảng viên hướng dẫn đề tài p_j
    if $|M(p_j)| < c_j$ *và* $|M(l_k)| < d_k$ **then**
      $M := M \cup \{(s_i, p_j)\}$
    else if $|M(p_j)| = c_j$ **then**
      $[s_t, g(s_t)] :=$ `Choose_Student` $(M(p_j), l_k)$
      if $g(s_t) > n + 1$ *hoặc* $rank(l_k, s_i) < rank(l_k, s_t)$ **then**
        $M := M \setminus \{(s_t, p_j)\} \cup \{(s_i, p_j)\}$
        if $g(s_t) < n + 1$ **then** $rank(s_t, p_j) := 0$
      else
        $rank(s_i, p_j) := 0$
    else
      $[s_w, g(s_w)] :=$ `Choose_Student` $(M(l_k), l_k)$
      if $g(s_w) > n + 1$ *hoặc* $rank(l_k, s_i) < rank(l_k, s_w)$ **then**
        $M := M \setminus \{(s_w, p_u)\} \cup \{(s_i, p_j)\}$, với $p_u = M(s_w)$
        `Repair` (p_u, l_k)
        if $g(s_w) < n + 1$ **then** $rank(s_w, p_u) := 0$
      else
        $rank(s_i, p_j) := 0$
  return M;
end function

Tiếp theo, với mỗi sinh viên $s_i \in \mathcal{S}$, s_i đề xuất tới p_j nếu p_j *đủ* dung lượng hoặc l_k là *đủ* dung lượng, thì thuật toán HAG xác định hàm $g(s_t)$ được mô tả trong Thuật toán 4.3 để chọn sinh viên s_t dựa vào giá trị lớn nhất của hàm heuristic $g(s_t)$ như sau:

$$g(s_t) = rank(l_k, s_t) + t(s_t) + r(s_t)/(q+1). \tag{4.3}$$

Algorithm 4.3: Hàm heuristic $g(s_t)$

Input: Tập các sinh viên X.
Output: Sinh viên s_t và $g(s_t)$.

function `Choose_Student` (X, l_k)
  for *mỗi* $s_t \in X$ **do**
    $t(s_t) := 0$;
    for *mỗi* $p_u | rank(s_t, p_u) = rank(s_t, M(s_t))$ **do**
      l_z := giảng viên hướng dẫn đề tài p_u;
      $t(s_t) = t(s_t) + min(d_z - |M(l_z)|, 1) \times min(c_j - |M(p_u)|, 1) \times n$;
    $r(s_t)$:= số lượng đề tài được xếp hạng bởi s_t;
    $g(s_t) := rank(l_k, s_t) + t(s_t) + r(s_t)/(q+1)$;
  $s_t := \text{argmax}(g(s_t))$;
  return $s_t, g(s_t)$;
end function

Algorithm 4.4: Phá vỡ các cặp chặn kiểu *(3bi)*

Input: Phép ghép M
Output: Phép ghép M.

function `Repair` (p_u, l_k)
  f := true;
  while f = *true* **do**
    f:= false;
    if $s_k \in M(l_k)$ *và* $rank(s_k, p_u) < rank(s_k, p_z) | p_z = M(s_k)$ **then**
      $M := M \setminus \{(s_k, p_z)\} \cup \{(s_k, p_u)\}$;
      $p_u := p_z$;
      f := true;
  return M;
end function

Thuật toán 4.4 được sử dụng để phá vỡ các cặp chặn khi một đề tài p_u bị

xóa khỏi M. Với mỗi $s_k \in M(l_k)$, nếu $rank(s_k, p_u) < rank(s_k, p_z)$, trong đó $p_z = M(s_k)$, thuật toán loại bỏ (s_k, p_z) và thêm (s_k, p_u) vào M. Quá trình này lặp lại cho mỗi đề tài bị xóa cho đến khi nó không thể tạo thành các cặp chặn. Nếu một phép phép ổn định tìm được chưa đạt kích thước tối đa, HAG gọi Thuật toán 4.5 cải thiện kích thước của phép ghép hiện tại. Thuật toán HAG dừng lại khi tìm được phép ghép ổn định với kích thước tối đa hoặc tất cả sinh viên chưa được ghép không thể tìm được bất kỳ đề tài nào để ghép.

Algorithm 4.5: Vượt qua tối thiểu cục bộ

Input: Phép ghép ổn định M.
Output: Phép ghép ổn định M.

```
1.  function Escape(M)
2.      for mỗi sinh viên chưa ghép s_u ∈ U do
3.          Khởi tạo lại danh sách xếp hạng của s_u;
4.          while danh sách xếp hạng của s_u không rỗng do
5.              p_z := argmin(rank(s_u, p_z) > 0), ∀p_z ∈ P;
6.              l_k := giảng viên hướng dẫn đề tài p_z;
7.              for ( mỗi s_i ∈ M(l_k) | rank(l_k, s_i) = rank(l_k, s_u)) do
8.                  if (|M(p_z)|) < c_z) hoặc (s_i ∈ M(p_z) và |M(p_z)| = c_z)
                      then
9.                      if v(s_u) ≥ v(s_i) then
10.                         p_j := M(s_i);
11.                         M := M \ {(s_i, p_j)} ∪ {(s_u, p_z)};
12.                         v(s_u) := v(s_u) + 1;
13.                         Repair (p_j, l_k) ;
14.                         break;
15.             if M(s_u) ≠ ∅ then
16.                 break;
17.             else
18.                 rank(s_u, p_z) := 0;
19.     return M;
20. end function
```

Tất cả các thuật toán thực hiện trên phần mềm Matlab 2019a và thực hiện trên máy tính có cấu hình CPU Core i7-8550U 1.8 GHz và RAM 16 GB trong Windows 10. Các kết quả thực nghiệm chỉ ra rằng thuật toán HAG hiệu quả về thời gian thực hiện và chất lượng nghiệm so với thuật toán xấp xỉ APX cho MAX-SPA-ST.

KẾT LUẬN

Luận án đã hoàn thành mục tiêu nghiên cứu đề ra và đạt được những kết quả như sau:

- Đề xuất 02 thuật toán giải quyết bài toán MAX-SMTI được công bố trong Chương 2.
- Đề xuất 02 thuật toán giải quyết bài toán MAX-HRT được công bố trong Chương 3.
- Đề xuất 02 thuật toán giải quyết bài toán SPA-P và SPA-ST được công bố trong Chương 4.

Các nghiên cứu được công bố trên các kỷ yếu hội thảo, các tạp chí uy tín thuộc chuyên ngành Heuristic, Khoa học máy tính và Trí tuệ nhân tạo trong nước và quốc tế, một số công bố thuộc danh mục SCIE và SCOPUS.

Từ các kết quả đạt được và những hạn chế trong luận án này, trong tương lai luận án sẽ tiếp tục nghiên cứu các thuật toán heuristic hiệu quả cho các biến thể khác của bài toán hôn nhân ổn định và các hướng tiếp cận khác cho bài toán hôn nhân ổn định và các biến thể.