

# MỘT THUẬT TOÁN TÌM KIẾM CỤC BỘ HIỆU QUẢ GIẢI BÀI TOÁN PHÂN CÔNG ĐỊA ĐIỂM THỰC TẬP CHO SINH VIÊN

**Hoàng Hữu Việt<sup>1</sup>✉, Nguyễn Thị Uyên<sup>1</sup>,  
Cao Thanh Sơn<sup>1</sup>, Lê Hồng Trang<sup>2</sup>**

---

(1) Viện Kỹ thuật và Công nghệ, Đại học Vinh

(2) Khoa Khoa học và Kỹ thuật, Đại học Bách Khoa Tp.HCM  
{viethh, uyennt, sonct}@vinhuni.edu.vn, lhtrang@hcmut.edu.vn



- 1 Giới thiệu
- 2 Nghiên cứu liên quan
- 3 Thuật toán đề xuất
- 4 Thực nghiệm
- 5 Kết luận

# Giới thiệu (1)

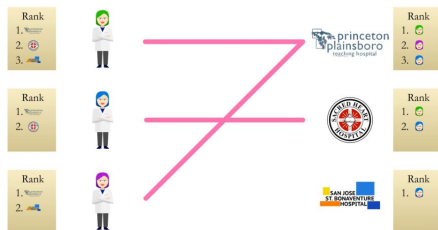
- Bài toán phân công địa điểm thực tập cho sinh viên, gọi là bài toán **Hospitals/Residents problem - HR** [Gale and Shapley (1962)].
- Bài toán **HR** bao gồm một tập các sinh viên (**Student**) và một tập các doanh nghiệp (**Enterprises**).

Danh sách ưu thích của sinh viên	Danh sách ưu thích của doanh nghiệp
$r_1: h_1 h_3 h_2$	$h_1: (2): r_3 r_7 r_5 r_2 r_4 r_6 r_1$
$r_2: h_1 h_5 h_4 h_3$	$h_2: (3): r_5 r_6 r_3 r_4 r_1$
$r_3: h_1 h_5 h_2$	$h_3: (1): r_5 r_2 r_6 r_1 r_7$
$r_4: h_1 h_2 h_4$	$h_4: (1): r_8 r_2 r_4 r_7$
$r_5: h_3 h_1 h_2$	$h_5: (1): r_3 r_7 r_6 r_8 r_2$
$r_6: h_3 h_2 h_1 h_5$	
$r_7: h_3 h_4 h_5 h_1$	
$r_8: h_5 h_4$	

- Mục đích là tìm một phép ghép ổn định (**stable matching**),  $M$ .

## Giới thiệu (2)

- Bài toán HR không được ứng dụng nhiều trong thực tế [Gusfield (1989)].
- The Hospitals/Residents problem with Ties (HRT) là một biến thể phổ biến của bài toán HR [Irving (2000)].
  - ▶ National Resident Matching Program (NRMP) in US,
  - ▶ Scottish Pre-registration house officer Allocations (SPA) in UK,
  - ▶ Canadian Resident Matching Service (CARMS) in Canada.



# I Giới thiệu (3)

- Bài toán **HRT** có chứa quan hệ ngang bằng (*ties*) trong danh sách ưu thích của sinh viên và doanh nghiệp.

Danh sách ưu thích của sinh viên	Danh sách ưu thích của doanh nghiệp
$r_1: h_1 h_3 h_2$	$h_1: (2): r_3 (r_7 r_5 r_2) r_4 r_6 r_1$
$r_2: h_1 (h_5 h_4) h_3$	$h_2: (3): r_5 r_6 (r_3 r_4) r_1$
$r_3: h_1 h_5 h_2$	$h_3: (1): (r_5 r_2) r_6 r_1 r_7$
$r_4: h_1 (h_2 h_4)$	$h_4: (1): r_8 r_2 r_4 r_7$
$r_5: h_3 h_1 h_2$	$h_5: (1): r_3 (r_7 r_6 r_8) r_2$
$r_6: (h_3 h_2) h_1 h_5$	
$r_7: h_3 h_4 h_5 h_1$	
$r_8: h_5 h_4$	

- Mục đích là tìm một phép ghép hoàn chỉnh (**perfect matching**).

# Giới thiệu (4)

Danh sách ưu thích của sinh viên	Danh sách ưu thích của doanh nghiệp
$r_1: h_1 h_3 h_2$	$h_1: (2): r_3 (r_7 r_5 r_2) r_4 r_6 r_1$
$r_2: h_1 (h_5 h_4) h_3$	$h_2: (3): r_5 r_6 (r_3 r_4) r_1$
$r_3: h_1 h_5 h_2$	$h_3: (1): (r_5 r_2) r_6 r_1 r_7$
$r_4: h_1 (h_2 h_4)$	$h_4: (1): r_8 r_2 r_4 r_7$
$r_5: h_3 h_1 h_2$	$h_5: (1): r_3 (r_7 r_6 r_8) r_2$
$r_6: (h_3 h_2) h_1 h_5$	
$r_7: h_3 h_4 h_5 h_1$	
$r_8: h_5 h_4$	

- Phép ghép (matching),  $M$ :

$$M = \{(r_1, \emptyset), (r_2, h_5), (r_3, h_1), (r_4, h_4), (r_5, \emptyset), (r_6, h_1), (r_7, h_3), (r_8, \emptyset)\}.$$

- Cặp khối (blocking pair):

▶  $(r_1, h_2), (r_2, h_1), (r_4, h_1), (r_5, h_1), (r_5, h_2), (r_5, h_3)$

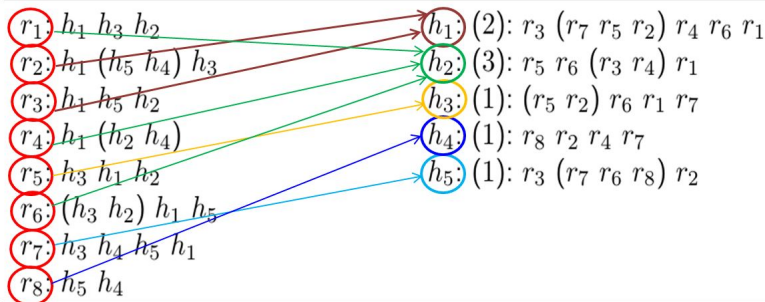
- Cặp khối trội nhất ((undominated blocking pair):

▶  $(r_5, h_3)$

# Giới thiệu (5)

Danh sách ưu thích của sinh viên

Danh sách ưu thích của doanh nghiệp



- Phép ghép hoàn chỉnh (**perfect matching**)

$$M = \{(r_1, h_2), (r_2, h_1), (r_3, h_1), (r_4, h_2), (r_5, h_3), (r_6, h_2), (r_7, h_5), (r_8, h_4)\}.$$

- **Manlove et al. (2012), Manlove et al. (2017)** đã đề xuất thuật toán SAT để tìm một phép ghép ổn định cho bài toán HRT.
- **Kwanashie et al. (2013), Drummond et al.(2015)** đã sử dụng mô hình IP (Integer Programming) để giải quyết bài toán HRT.
- **Gelain et al. (2013)** đã giới thiệu một thuật toán dựa trên tìm kiếm cục bộ (Local search) để tìm một phép ghép ổn định với kích thước tối đa (MAX-HRT).
- **Munera et al. (2015)** đã áp dụng lập trình thích nghi (Adaptive Search) để giải quyết bài toán MAX-HRT.



# Thuật toán đề xuất

- Đề xuất thuật toán **MIN-BPS** để giải bài toán MAX-HRT.
- Ý tưởng chính:
  - ▶ Bắt đầu từ một phép ghép chưa ổn định,
  - ▶ Chọn ngẫu nhiên một  $r_i \in \mathcal{R}$  theo thứ tự xoay vòng và tìm một  $h_j$  trong danh sách xếp hạng của  $r_i$  sao cho cặp  $(r_i, h_j)$  là một **cặp khối trội nhất** của phép ghép hiện thời.
  - ▶ Loại bỏ cặp  $(r_i, h_j)$  sao cho cải tiến được tính ổn định của phép ghép hiện thời.

# Thuật toán đề xuất

## Algorithm 1: MIN-BPS Algorithm

```

Input: - An instance  $I$  of HRT.
        - A maximum iter. number  $max\_iters$ 
Output: A matching  $M$ .
function Main( $I$ )
1   $M :=$  a random matching;
2   $M_{best} := M$ ;
3   $r_i := random(n)$ ;
4   $iter := 0$ ;
5  while ( $iter \leq max\_iters$ ) do
6    for ( $k = 1 \dots n$ ) do
7       $r_k := mod(r_i, n) + 1$ ;
8       $h_j := find(r_k, M)$ ;
9      if ( $h_j \neq \emptyset$ ) then
10       break;
11     end
12   end
13   if ( $h_j = \emptyset$ ) then
14     if ( $f(M_{best}) > f(M)$ ) then
15        $M_{best} := M$ ;
16     end
17     if ( $f(M_{best}) = 0$ ) then
18       break;
19     else
20        $M :=$  a random matching;
21       continue;
22     end
23   end
24    $M := M \cup \{(r_i, h_j)\}$ ;
25   if ( $h_j$  is over-subscribed) then
26      $s_w := h_j$ 's worst non-empty student;
27      $M := M \setminus \{(s_w, h_j)\}$ ;
28   end
29    $iter := iter + 1$ ;
30 end
31 return  $M_{best}$ ;
32 end function

```

## Thuật toán Min-BPs

## Thực hiện một ví dụ của thuật toán Min-BPs

- Bắt đầu từ phép ghép  $M_0 = \{(r_1, h_2), (r_2, h_1), (r_3, h_2), (r_4, \emptyset), (r_5, h_3), (r_6, h_1), (r_7, h_4), (r_8, h_5)\}$ .
- Xét ngẫu nhiên sinh viên  $r_2$
- Tại mỗi bước lặp, thuật toán thực hiện các bước như sau:

$i$	$M_i$	$r_i$	$h_j$	$M_{best}$
1	$M_0$	$r_3$	$h_1$	$M_1 = \{(r_1, h_2), (r_2, h_1), (r_3, h_1), (r_4, \emptyset), (r_5, h_3), (r_6, \emptyset), (r_7, h_4), (r_8, h_5)\}$
2	$M_1$	$r_4$	$h_2$	$M_2 = \{(r_1, h_2), (r_2, h_1), (r_3, h_1), (r_4, h_2), (r_5, h_3), (r_6, \emptyset), (r_7, h_4), (r_8, h_5)\}$
3	$M_2$	$r_5$	$\emptyset$	Xét sinh viên tiếp theo
		$r_6$	$h_2$	$M_3 = \{(r_1, h_2), (r_2, h_1), (r_3, h_1), (r_4, h_2), (r_5, h_3), (r_6, h_2), (r_7, h_4), (r_8, h_5)\}$
4	$M_3$	$r_7$	$\emptyset$	Xét sinh viên tiếp theo
		$r_8$	$\emptyset$	Xét các sinh viên tiếp theo
		$r_1, r_3, r_4, r_5, r_6, r_7, r_8$	$\emptyset$	

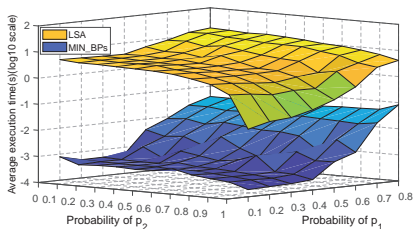
Min-BPs trả về một phép ghép hoàn chỉnh

$$M_3 = \{(r_1, h_2), (r_2, h_1), (r_3, h_1), (r_4, h_2), (r_5, h_3), (r_6, h_2), (r_7, h_4), (r_8, h_5)\}$$

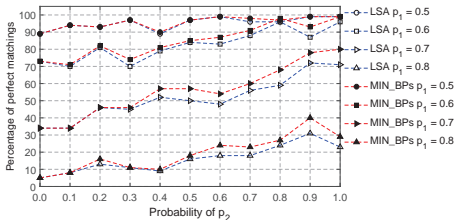
## Ví dụ

# Thực nghiệm (1)

- Thử nghiệm so sánh về thời gian thực hiện và chất lượng nghiệm tìm thấy bởi MIN-BPs và LSA.
- Tạo ngẫu nhiên HRT:  $n = 50, m = 10, p_1 \in [0.1, 0.8], p_2 \in [0.0, 1.0], c_j$  sinh ra ngẫu nhiên.



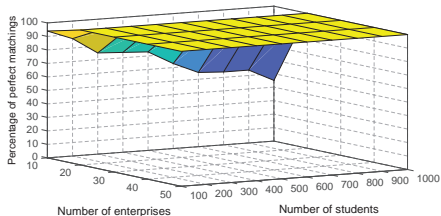
Hình 1. Trung bình thời gian thực hiện



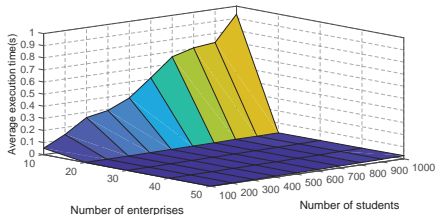
Hình 2. Phần trăm phép ghép hoàn chỉnh

# Thử nghiệm (2)

- Tạo ngẫu nhiên HRT:  $n \in [100, 700]$ ,  $m \in [10, 50]$ ,  $p_1 = p_2 = 0.5$ , và  $c_j \in [1, q]$ ,  $q$  là số lượng sinh viên được doanh nghiệp xếp hạng.

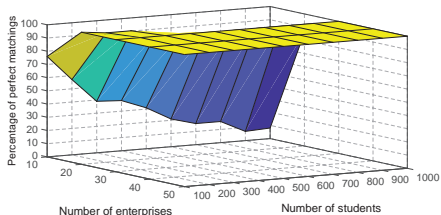


Hình 3. Phần trăm phép ghép hoàn chỉnh

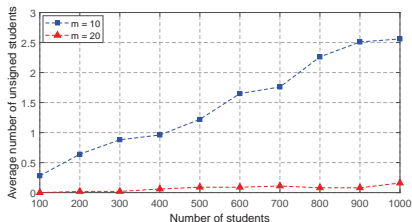


Hình 4. Trung bình thời gian thực hiện

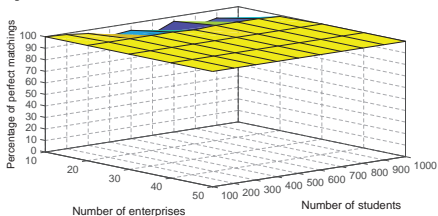
# Thử nghiệm (3)



Hình 5. Phần trăm phép ghép hoàn chỉnh khi  $c_j = n/m$



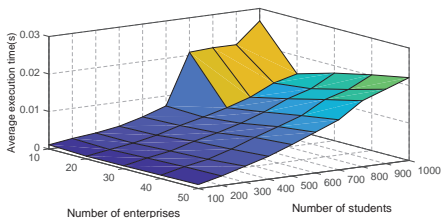
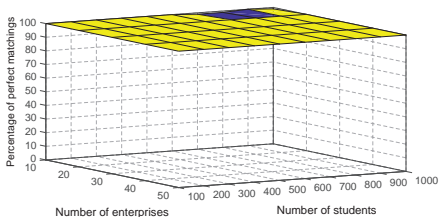
Hình 6. Trung bình số sinh viên chưa được ghép



Hình 7. Phần trăm phép ghép hoàn chỉnh khi tăng  $c_j = 1.1 * n.m$

# Thử nghiệm (4)

- Tạo ngẫu nhiên HRT:  $n \in [100, 700]$ ,  $m \in [10, 50]$ ,  $p_1 = p_2 = 0.5$ , và  $c_j = [0.2q, 0.5q]$



Hình 6. Phần trăm phép ghép hoàn chỉnh

Hình 7. Trung bình thời gian thực hiện

- Thuật toán **MIN-BPs** tìm phép ghép hoàn chỉnh cho bài toán phân công địa điểm thực tập sinh viên tại các trường đại học.
- Thực nghiệm chỉ ra rằng **MIN-BPs** hiệu quả cho bài toán kích thước lớn cả thời gian thực hiện và chất lượng nghiệm so với thuật toán LSA (**Gelain et al. 2013**).
- Hướng mở rộng:
  - ▶ Đánh giá các điều kiện ràng buộc về số lượng sinh viên và số chỗ thực tập của doanh nghiệp.
  - ▶ Cải tiến thuật toán và áp dụng cho các bài toán khác như SPA, SMTI.

**Xin chân thành cảm ơn!**

