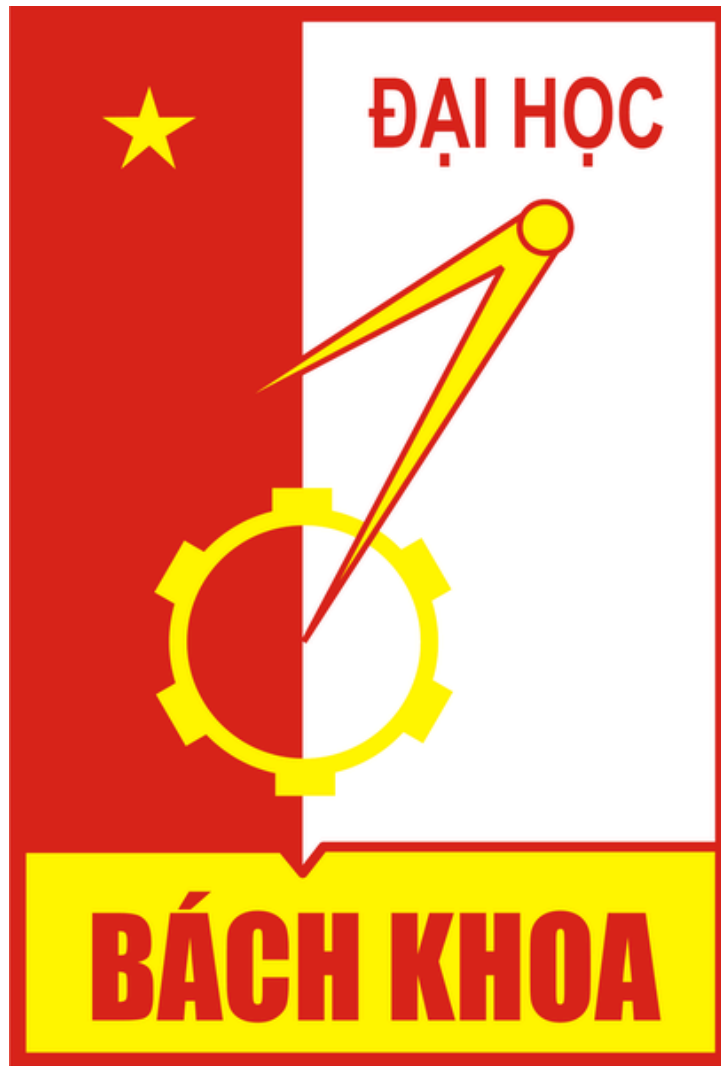


TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI



Tiểu luận

# Về vấn đề Tuyển Dụng Trợ Lý Hiring Assistant Problem

Đỗ Duy Quang - 20173325

*Hanoi, 8/2021*

# MỤC LỤC

<b>Phần I. Hiring Assistant Problem</b>	<b>3</b>
Giới thiệu vấn đề	3
Thuật toán	3
Các tiên đề xác suất	3
Vấn đề tồn đọng	4
<b>Phần II. Các hướng mở rộng</b>	<b>5</b>
Vấn đề ngẫu nhiên	5
Vấn đề muốn thuê người giỏi nhất	5
<b>Phần III. Ứng dụng</b>	<b>7</b>
Bài toán thuê nhà tại Hà Nội	7
Bài toán lựa chọn mô hình học máy	7
Bài toán lựa chọn bạn gái	7
<b>Phần IV. Kết luận và tài liệu tham khảo</b>	<b>8</b>

# Phần I. Hiring Assistant Problem

## 1. Giới thiệu vấn đề

Hiring Assistant Problem - Vấn đề tuyển dụng trợ lý là một vấn đề nổi tiếng trong các bài toán liên quan đến các thuật toán về xác suất. Vấn đề gốc được đưa ra cần phải giải quyết của bài toán này là:

- Cần thuê nhân viên (Assistant)
  - Bộ phận tuyển dụng HR sẽ gửi 1 ứng viên mỗi ngày đến phòng vấn
  - Phòng vấn người này, có thể thuê hoặc không thuê
  - Luôn luôn thuê người tốt nhất - tức thuê người tốt hơn người trước

## 2. Thuật toán

```
HIRE-ASSISTANT(n) best <- 0
  for i <- 1 to n do
    if candidate[i] is better than candidate[best]
      best <- i
  hire candidate
```

Có thể thấy thuật toán trên sẽ lặp lại việc thuê các ứng viên tốt hơn những ứng viên đã thuê. Có thể sa thải người cũ hoặc không.

Gọi chi phí phỏng vấn ứng viên  $i$  là  $C_i$   
Gọi chi phí tuyển dụng là  $Ch$   
Tổng chi phí:  $Cost = O(C_i * n + Ch * m)$

=> Worst case:  $C_i * n + Ch * n$   
Best case:  $C_i * n + Ch$

## 3. Các tiền đề xác suất

Gọi  $X_i = 1$  nếu ứng viên  $i$  được thuê, ngược lại  $X_i = 0$   
Gọi  $X$  tổng các ứng viên được thuê =>  $X = X_1 + X_2 + \dots + X_i$   
Khi đó, kỳ vọng số ứng viên được thuê sẽ là:

$$\begin{aligned}
E[X] &= E\left[\sum_{i=1}^n X_i\right] \\
&= \sum_{i=1}^n E[x_i]
\end{aligned}$$

Thực ra, trong  $i$  ứng viên, để ứng viên thứ  $i$  được thuê thì ứng viên đó phải là tốt nhất trong  $i$  người đã duyệt. Vậy nên ta có :  $E[X_i]$  chính là xác suất để ứng viên  $i$  được thuê và có giá trị là  $1/i$ .

Suy ra kỳ vọng thực sự sẽ là:

$$\begin{aligned}
E[X] &= E\left[\sum_{i=1}^n X_i\right] \\
&= \sum_{i=1}^n E[x_i] \\
&= \sum_{i=1}^n 1/i \\
&= \ln n + O(1) \\
&\in O(\lg n)
\end{aligned}$$

Đây là kỳ vọng nếu các ứng viên có thứ tự ngẫu nhiên.

Thực ra, kỳ vọng này là hoàn toàn chấp nhận được và có ý nghĩa. Nếu áp dụng công thức vào việc tuyển dụng 100 người, thì kỳ vọng số người được tuyển dụng sẽ là 5 người. Đây là con số khá tốt, không quá lớn không quá nhỏ.

Tuy nhiên, thuật toán này vẫn còn một số vấn đề ta sẽ bàn ở ngay sau đây.

## 4. Vấn đề tồn đọng

- Thứ tự của người phỏng vấn có thể không thực sự là ngẫu nhiên?
- Trường hợp muốn tìm ra **một người giỏi nhất** và **chỉ thuê một lần** (Tức quá trình phỏng vấn sẽ dừng ngay khi ta thuê một ứng viên), làm sao để tăng tỷ lệ và giảm chi phí?
- Một số vấn đề mở rộng liên quan đến ranking, ...

## Phần II. Các hướng mở rộng

### 1. Vấn đề ngẫu nhiên

Khi ta đã có một tập các ứng viên cần phỏng vấn, ta có thể xáo trộn lại thứ tự các ứng viên đó một cách ngẫu nhiên. Từ đó, ta sẽ có một tập các ứng viên giả ngẫu nhiên. Trong đó, một số cách ngẫu nhiên có thể được sử dụng:

**Cách 1:** Sử dụng các hoán vị ngẫu nhiên dựa trên độ ưu tiên

- Gán cho mỗi ứng viên một độ ưu tiên
- Sắp xếp dựa trên độ ưu tiên

```
n <- length(A)
for i <- 1 to n do
  Priority[i] = Random(1,n*n*n)
sort A (using Priority as keys)
```

**Cách 2:** Sử dụng hoán vị ngẫu nhiên các ứng viên dựa trên tổng số ứng viên

```
n <- length(A)
for i <- 1 to n do
  swap(A[i], A[Random(i,n)])
```

### 2. Vấn đề muốn thuê người giỏi nhất

Một bài toán mới đặt ra từ Hiring Assistant Problem: Bây giờ, tôi cần thuê một người giỏi nhất và chỉ thuê một lần. Cần phải làm sao để tối ưu hoá khả năng thuê được người giỏi nhất trong tổng số các ứng viên?

Một bài toán được phát triển: Online Hiring Problem. Đó là bài toán với nội dung đã nói như trên. Thuật toán giải quyết nó sẽ là:

- Phỏng vấn  $k$  ứng viên đầu tiên, từ chối hết họ
- Sau đó phỏng vấn các ứng viên tiếp theo, thuê người đầu tiên mà ta thấy tốt hơn tất cả các ứng viên đã từng phỏng vấn

Khi đó, ta cần phải tối ưu  $k$  đối với tập  $n$  ứng viên:

- Nếu  $k$  quá lớn, khả năng ta bỏ qua best candidate ở tập  $k$  là lớn
- Nếu  $k$  quá nhỏ, khả năng ta dừng lại là rất nhanh vì chỉ cần tìm thấy người tốt hơn tất cả là ta sẽ dừng lại
- Lấy  $k$  sao cho phù hợp?

Ta sẽ tiến hành phân tích xác suất để tìm ra số  $k$  phù hợp.

Coi S là sự kiện ta lấy được ứng viên tốt nhất.  
 Coi  $S_i$  là sự kiện: Ứng viên tốt nhất là thứ  $i$ , và ta cũng lấy ứng viên  $i$   
 Từ đó, xác suất để ta lấy được ứng viên tốt nhất sẽ là:

$$Pr\{s\} = \sum_{i=k+1}^n Pr\{S_i\}$$

$i$  chạy từ  $k+1$  do  $k$  ứng viên đầu tiên ta từ chối hết.

Xét 2 sự kiện:

- $B_i$ : ứng viên  $i$  là tốt nhất
- $O_i$ : không có ai trong các ứng viên  $k-1 \dots i-1$  đã được chọn
- Từ đó, ta có:

$$\begin{aligned} Pr\{S_i\} &= Pr\{B_i \wedge O_i\} \\ &= Pr\{B_i\} * Pr\{O_i|B_i\} \\ &= Pr\{B_i\} * Pr\{O_i\} \\ &= (1/n) * k/(i-1) \end{aligned}$$

Vậy xác suất  $P(S)$  sẽ là:

$$\begin{aligned} Pr\{S_i\} &= (1/n) * k/(i-1) \\ Pr\{S\} &= \sum_{i=k+1}^n \frac{k}{n(i-1)} \\ &= \frac{k}{n} \sum_{i=k+1}^n \frac{1}{i-1} \\ &= \frac{k}{n} \sum_{i=k}^{n-1} \frac{1}{i} \end{aligned}$$

Nhận thấy một tính chất:

$$\begin{aligned} \frac{k}{n} \int_k^n \frac{1}{x} dx &\leq \frac{k}{n} \sum_{i=k}^{n-1} \frac{1}{i} \\ \frac{k}{n} (\ln n - \ln k) &\leq Pr\{S\} \end{aligned}$$

Tích phân phía trái luôn nhỏ hơn hoặc bằng Xác suất lấy được người giỏi nhất của chúng ta. Để tối ưu hàm này, ta cần tìm giá trị lớn nhất của vế trái. Ở đây, vế trái đạt giá trị lớn nhất tại :

$$k = \frac{n}{e}$$

Điều này là có ý nghĩa trong thực tiễn. Giả sử chúng ta có 100 ứng viên. Chúng ta sẽ phỏng vấn 33 ứng viên đầu và đánh trượt hết. Sau đó tìm ra ứng viên tốt nhất ở phía sau. Khi đó, tỷ lệ có được ứng viên tốt nhất là  $0.37 \sim 37\%$  - khá tốt.

## Phần III. Ứng dụng

### 1. Bài toán thuê nhà tại Hà Nội

Các bạn sinh viên khi mới ra Hà Nội có thể gặp nhiều vấn đề về thuê nhà cửa. Việc thuê nhà có thể áp dụng 2 cách:

Cách 1:

Nếu bạn có một khoảng thời gian rộng rãi, mỗi ngày xem ngẫu nhiên một nhà, sau đó nếu thấy ưng hơn những ngôi nhà đã xem trước => có thể đưa vào danh sách thuê. Qua một số lượng ngày, ta sẽ có một tập các ứng cử viên là các ngôi nhà tốt nhất và ta có thể thuê ngôi nhà với chất lượng tốt nhất

Cách 2:

Tuy nhiên, bạn có một khoảng thời gian hạn chế và cần quyết định việc thuê ngay (vì nhà cửa thường có thời gian trống khá ngắn), bạn sẽ lập ra danh sách các nhà trọ cần thăm, xáo trộn chúng. Sau đó, bạn thăm khoảng  $\frac{1}{3}$  ngôi nhà đầu tiên và từ chối thuê hết các ngôi nhà đó. Sau đó, đi thăm các ngôi nhà tiếp theo. Nếu thấy ngôi nhà nào tốt hơn tất cả các ngôi nhà đã từng xem, có thể thuê ngay với xác suất thuê được best house khoảng 0.4

### 2. Bài toán lựa chọn mô hình học máy

Với các mô hình học máy, việc nguồn thời gian và các bước testing của chúng ta là nhiều. Vì thế, để lựa chọn mô hình học máy tốt nhất cho 1 bài toán, ta có thể thử nhiều mô hình với kết quả tốt từ những bài toán cùng loại, sau đó chọn ra các mô hình tốt nhất. Sử dụng các mô hình đó sẽ cho ra các kết quả tốt, và có thể cải tiến để tìm ra mô hình tốt nhất với bài toán của mình sau này.

### 3. Bài toán lựa chọn bạn gái

Tất nhiên, tình yêu là thứ nằm ngoài khả năng lý giải của khoa học. Tuy nhiên, giả sử bạn là một người chung thủy, ít thay đổi, muốn tìm một người bạn gái tốt nhất trong tập các ứng cử viên. Bạn có thể sử dụng bài toán thứ 2 trong Phần II. Bạn cần liệt kê ra n ứng cử viên bạn có thể tiếp cận. Sau đó, bạn tiếp cận với khoảng  $n/e$  cô gái đầu, làm quen... tuy

nhiên không trở thành bạn trai của họ. Sau đó, với những cô gái tiếp theo, bạn sẽ tìm ra cô gái nào được bạn coi là “tốt hơn” tất cả các cô gái đã làm quen từ trước. Sau đó, bạn sẽ cố gắng tán tỉnh và trở thành bạn trai của cô gái đó, chứ không cần đi làm quen các cô gái khác. Điều này vừa đảm bảo tính chung thủy, vừa đưa ra một khả năng bạn chọn được cô gái tốt nhất lên đến gần 40%.

## Phần IV. Kết luận và tài liệu tham khảo

Với những ý tưởng tìm hiểu ở trên, ta đã có các phương thức để thực hiện trong các ngữ cảnh nhất định như : Khoảng thời gian, quy trình, cần quyết định ngay hay không... Và áp dụng được vào khá nhiều trường hợp trong thực tế.

Bài toán Online Assistant Problem được phát triển từ bài toán Hiring Assistant problem và được em tham khảo từ khoá học CS673. Em xin cảm ơn các giáo sư trong khoá CS673 đã viết tài liệu để em có thể học tập, đồng cảm ơn các giáo sư và đặc biệt là thầy Nguyễn Khanh Văn với bộ slide về các mô hình thuật toán xác suất để hiểu và là tài liệu chính cho bài tiểu luận này.

Nguồn tài liệu tham khảo:

CS673 - Hiring Assistant Problem:

<https://www.cs.usfca.edu/~galles/cs673/lecture/lecture2.printable.pdf>

Ph.D Van K. Nguyen, Probability for Computing:

<https://users.soict.hust.edu.vn/vannk/Prob4Computing/Prob4Computing.htm>