



# CUỘC THI TOÁN MÔ HÌNH 2024 VÒNG 1

02 - 28 tháng 7, 2024

Tên đội thi:  
Ngày thực hiện:

Đông Tây Bắc  
18-28/07/2024

## **Nhóm tác giả**

Nguyễn Trần Trung

Phạm Công Hoàng

Nguyễn Minh Tân

THPT Chuyên Lào Cai - Lào Cai

THPT Chuyên Lê Quý Đôn - Lai Châu

THPT Lê Hồng Phong - Thái Nguyên

Mục lục

<b>1</b>	<b>Giới thiệu</b>	<b>2</b>
1.1	Tổng quan vấn đề . . . . .	2
1.2	Lập trình ràng buộc . . . . .	2
<b>2</b>	<b>Nội dung bài toán</b>	<b>2</b>
2.1	Phát biểu bài toán . . . . .	2
2.2	Phân tích bài toán . . . . .	3
2.2.1	Mô hình hóa bài toán . . . . .	3
2.2.2	Điều kiện không thể vi phạm . . . . .	4
2.2.3	Điều kiện thực hiện theo khả năng . . . . .	4
<b>3</b>	<b>Giải bài toán</b>	<b>5</b>
3.1	Đưa các ràng buộc về biểu thức toán học . . . . .	5
3.2	Lập trình ràng buộc sử dụng công cụ OR-Tools . . . . .	7
3.3	Kết quả . . . . .	8
<b>4</b>	<b>Đánh giá - Hướng phát triển</b>	<b>8</b>
	<b>Lời cảm ơn</b>	<b>9</b>
	<b>Phụ lục</b>	<b>10</b>
	<b>Tài liệu</b>	<b>11</b>

# 1 Giới thiệu

## 1.1 Tổng quan vấn đề

Bệnh viện, trạm xá, trường học hay bất kỳ một tổ chức nào khác cũng đều cần có một kế hoạch phân bổ lịch trình làm việc hợp lý cho các nhân viên để đáp ứng một số nhu cầu nhất định về cả mặt tổ chức và nhu cầu cá nhân. Vấn đề này còn được biết đến là bài toán lập kế hoạch cho y tá, hay *Nurse Scheduling Problems - NSPs* [1] hoặc *Nurse Rostering Problem - NRPs* [2].

Theo bài báo "Nurse rostering problems – a bibliographic survey"[3], *NSP* liên quan đến việc lập bảng phân công nhiệm vụ định kỳ (hàng tuần, hai tuần một lần hoặc hàng tháng) cho nhân viên điều dưỡng, tuân theo nhiều ràng buộc cứng/mềm (*hard-constraints/soft-constraints*) [2] như pháp luật, chính sách nhân sự, sở thích của y tá và nhiều yêu cầu khác có thể dành riêng cho từng bệnh viện. Những hạn chế này có thể khác nhau giữa các bệnh viện, tổ chức trong khi các mục tiêu trong việc phân công cũng có thể khác nhau. Những điều này đã dẫn đến một loạt các mô hình *NSP* và do đó, một loạt các phương pháp tiếp cận giải pháp đã được phát triển cho các mô hình này.

## 1.2 Lập trình ràng buộc

*Lập trình ràng buộc* hay *Constraints Programming (CP)* [2] là một mô hình lập trình có khả năng thể hiện và thỏa mãn những ràng buộc cho trước, đồng thời tối ưu hóa một số mục tiêu khác. Lập trình ràng buộc có thể xử lý hiệu quả các bài toán *NSPs*, đặc biệt, nó được xây dựng trong các thư viện, bộ phần mềm mã nguồn mở có sẵn, cụ thể ở đây ta sử dụng bộ công cụ *Google OR-Tools* [4].

*Google OR-Tools* là bộ phần mềm mã nguồn mở và miễn phí do Google phát triển để giải các bài toán quy hoạch tuyến tính, lập trình số nguyên hỗn hợp, lập trình ràng buộc, định tuyến phương tiện và các vấn đề tối ưu hóa có liên quan.

# 2 Nội dung bài toán

## 2.1 Phát biểu bài toán

Trong cuộc thi này, chúng ta sẽ đi vào giải quyết bài toán sắp xếp các ca trực từ ngày 05/08/2024 đến ngày 01/09/2024 dành cho một trạm xá nhỏ gồm

8 nhân viên, trong đó:

- Có 4 điều dưỡng An, Bình, Châu và Dương làm việc toàn thời gian (full-time) 36 giờ một tuần, tương ứng 18 ca 8 tiếng trong 4 tuần.

- Có 4 điều dưỡng Linh, Kiệt, Giang và Hiếu thực tập bán thời gian (part-time) 20 giờ một tuần, tương ứng 10 ca 8 tiếng trong 4 tuần.

Các ca trực được lên lịch bắt buộc phải tuân theo điều kiện:

(a) Lịch làm việc trong ngày của trạm xá này được chia làm hai ca gồm ca ngày bắt đầu từ 8 giờ đến 5 giờ chiều và ca đêm bắt đầu từ 11 giờ đêm đến 7 giờ sáng hôm sau (được tính vào ca của ngày bắt đầu).

(b) Mỗi ngày, trạm xá cần 3 nhân viên cho một ca ngày và 1 nhân viên cho một ca đêm.

Bên cạnh đó, ta xem xét thực hiện theo khả năng những mong muốn sau của các nhân viên:

(c) Đối với mỗi tuần (chỉ tính từ thứ 2 đến chủ nhật), nhân viên làm việc full-time muốn làm đúng 4 đến 5 ca một tuần, trong khi nhân viên làm việc part-time muốn làm 2 đến 3 ca một tuần.

(d) Một chuỗi ngày liên tiếp làm việc của nhân viên full-time sẽ làm việc 4 đến 6 ngày còn nhân viên part-time sẽ là 2 đến 3 ngày.

## 2.2 Phân tích bài toán

### 2.2.1 Mô hình hóa bài toán

Có thể thấy, đề bài cho ta một yêu cầu về việc tìm ra một cách xếp lịch làm việc hợp lý cho các nhân viên của trạm xá, thỏa mãn các ràng buộc đã cho trước về thời gian trực cần thiết mỗi ngày của trạm xá và thời gian làm việc tối đa của nhân viên. Ngoài ra cũng cần đảm bảo một số quy tắc khá rõ ràng như mỗi nhân viên sẽ chỉ làm tối đa một ca mỗi ngày, hay các nhân viên chỉ có thể tham gia trực vào một trong hai ca đã cho trước.

Trước hết, ta nhìn nhận điều kiện ban đầu của bài toán như một bảng vuông  $8 \times 28$  với 8 hàng, 28 cột lần lượt tương ứng với 8 nhân viên và 28 ngày trong lịch làm việc (Xem [Bảng 1 - Phụ lục](#)). Như vậy, bài toán có thể quy về việc điền các ô trong bảng bởi ký tự "S" - nếu nhân viên đó trực ca sáng vào ngày tương ứng, ký tự "T" nếu nhân viên đó trực ca tối cũng như không điền gì cả nếu người này không tham gia trực vào ngày hôm đó thỏa mãn những điều kiện cho trước.

### 2.2.2 Điều kiện không thể vi phạm

Gọi  $X$  là tổng thời gian làm việc của tất cả nhân viên trong lịch trình 28 ngày. Ta có những bất đẳng thức sau:

- Vì mỗi nhân viên full-time làm việc tối đa 18 ca 8 tiếng trong 4 tuần còn của mỗi nhân viên part-time là 10 ca 8 tiếng trong 4 tuần. Vậy :

$$X \leq 18.4 + 10.4 = 112 \quad (1)$$

- Mỗi ngày trong lịch đều cần ít nhất 3 nhân viên để trực ca sáng, 1 nhân viên để trực ca tối. Suy ra:

$$X \geq 28.4 = 112 \quad (2)$$

Từ hai bất đẳng thức (1) và (2) ta được  $X = 112$ , tức tổng thời gian làm việc của các nhân viên phải bằng đúng 112, đồng thời mọi đánh giá phía trên đều phải xảy ra đẳng thức.

Như vậy, ta thêm điều kiện cho mô hình phía trên như sau:

- Mỗi hàng trong 4 hàng đầu tiên đều phải được đánh dấu đúng 18 lần, mỗi hàng trong 4 hàng còn lại được đánh dấu đúng 10 lần.
- Mỗi cột trong 28 cột sẽ có đúng 3 ô được đánh dấu "S" và 1 ô được đánh dấu "T".

### 2.2.3 Điều kiện thực hiện theo khả năng

Những điều kiện mềm (soft-constraints) được đưa ra có thể được thể hiện như sau:

- Ở điều kiện (c), ta chia 28 cột thành từng nhóm gồm 7 cột liên tiếp, tạo thành 4 bảng con tương ứng với 4 tuần trong lịch. Trên mỗi hàng của 4 hàng đầu tiên tại mỗi bảng con, nên có từ 4-5 ô được đánh dấu, tương tự ở mỗi hàng trên 4 hàng còn lại sẽ có từ 2-3 ô.
- Với điều kiện cuối cùng, trên mỗi hàng của 4 hàng đầu tiên, ta phân hoạch nó ra thành các chuỗi ô liên tiếp được đánh dấu (những chuỗi này cách nhau ra bởi một hoặc nhiều ô trống), như vậy mỗi chuỗi này nên có độ dài từ 4-6 ô. Tương tự con số này ở mỗi hàng của 4 hàng còn lại nên là từ 2-3 ô.

Sau khi đã thể hiện được các điều kiện qua mô hình toán học, ta sẽ tìm ra một lời giải có thể thỏa mãn toàn bộ những điều kiện không thể vi phạm và đáp ứng tối đa những điều kiện thực hiện theo khả năng.

### 3 Giải bài toán

Dựa trên những điều kiện được cho, ta có thể dễ dàng chọn được một mô hình thỏa mãn những yêu cầu của đề bài (Xem [Bảng 2 - Phụ lục](#))

Nếu chỉ để giải bài toán này, mô hình như trên là đủ để đáp ứng những điều kiện đã cho. Tuy vậy, trong nhiều tình huống, việc đơn thuần tự tay chọn ra một lời giải như trên đôi khi là bất khả thi khi những số liệu trở nên lớn hơn, các điều kiện trở nên phức tạp hơn.

Vì vậy, ở đây ta sẽ xây dựng nên một lời giải có thể giải quyết những bài toán ở cùng mức độ hoặc có các cấu trúc ràng buộc tương tự.

Ta đã tìm ra được một đáp án thỏa mãn tất cả các ràng buộc, kể cả ràng buộc mềm, nên khi này việc coi các ràng buộc cứng hay mềm không còn quan trọng để có thể tìm ra được ít nhất một đáp án, do đó, ta có thể coi tất cả các ràng buộc của bài toán là ràng buộc cứng, và đều gọi chung là ràng buộc trong các bước về sau.

#### 3.1 Đưa các ràng buộc về biểu thức toán học

Trước hết, ta quy ước thứ tự các điều dưỡng từ 1 đến 8 lần lượt tương ứng là An, Bình, Châu, Dương, Linh, Kiệt, Giang, Huệ; thứ tự các ngày từ 1 đến 28 tương ứng từ Thứ 2 ngày 05/08/2024 đến Chủ Nhật ngày 01/09/2024 và các ca sáng tương ứng với 1 và ca tối tương ứng với 2.

Ta đặt các biến  $x_{n,d,s}$  với  $n, d \in \mathbb{N}^*$ ,  $1 \leq n \leq 8$ ,  $1 \leq d \leq 28$  và  $s \in \{1; 2\}$  (3) được xác định bằng 1 khi và chỉ khi người điều dưỡng thứ  $n$  làm việc vào ca  $s$  vào ngày thứ  $d$  và được xác định bằng 0 trong trường hợp ngược lại.

Sau đây, ta đưa xây dựng các biểu thức toán học tương ứng với các ràng buộc mà bài toán đã đưa cho.

Đầu tiên, biểu thức điều kiện xác định của các biến  $x_{n,d,s}$  thỏa mãn (3) và:

$$x_{n,d,s} \in \{0; 1\} \quad (4)$$

Hiển nhiên, mỗi người không thể làm việc cả hai ca cùng một ngày nên số ca mỗi người làm trong một ngày nhiều nhất là 1:

$$x_{n,d,1} + x_{n,d,2} \leq 1 \text{ với mọi } n, d \text{ thỏa mãn (3)} \quad (5)$$

Từ dữ kiện bài toán, trong 28 ngày, mỗi điều dưỡng full-time đều làm 18 ca và 10 ca với các điều dưỡng part-time:

$$\sum_{d=1}^{28} (x_{n,d,1} + x_{n,d,2}) = 18 \text{ với } n \in \{1; 2; 3; 4\} \quad (6)$$

$$\sum_{d=1}^{28} (x_{n,d,1} + x_{n,d,2}) = 10 \text{ với } n \in \{5; 6; 7; 8\} \quad (7)$$

Tiếp theo là ràng buộc 3 nhân viên cho ca ngày và 1 nhân viên cho ca đêm:

$$\sum_{n=1}^8 x_{n,d,1} = 3 \text{ với } d \text{ thỏa mãn (3)} \quad (8)$$

$$\sum_{n=1}^8 x_{n,d,2} = 1 \text{ với } d \text{ thỏa mãn (3)} \quad (9)$$

Ràng buộc các điều dưỡng full-time làm việc 4-5 ca một tuần, các điều dưỡng part-time làm 2-3 ca một tuần:

$$4 \leq \sum_{i=1}^7 (x_{n,7w+i,1} + x_{n,7w+i,2}) \leq 5 \text{ với } n \in \{1; 2; 3; 4\}, w \in \{0; 1; 2; 3\} \quad (10)$$

$$2 \leq \sum_{i=1}^7 (x_{n,7w+i,1} + x_{n,7w+i,2}) \leq 3 \text{ với } n \in \{5; 6; 7; 8\}, w \in \{0; 1; 2; 3\} \quad (11)$$

Cuối cùng là ràng buộc về chuỗi ngày làm việc, với các điều dưỡng full-time làm việc liên tiếp 4-6 ngày và 2-3 ngày với các điều dưỡng part-time. Trước hết, ta xét các điều dưỡng full-time, dễ dàng có được biểu thức tương ứng với ràng buộc chuỗi ngày làm việc lớn nhất là 6, tức tổng số ca trong 7 ngày đều không quá 6 ca:

$$\sum_{d=i}^{i+6} (x_{n,d,1} + x_{n,d,2}) \leq 6 \text{ với } n \in \{1; 2; 3; 4\}, i \in \mathbb{N}^*, 1 \leq i \leq 22 \quad (12)$$

Với ràng buộc về chuỗi ngày làm việc nhỏ nhất là 4, ta xây dựng điều kiện cho từng ngày một sao cho không xảy ra trường hợp vi phạm nào như sau:

- Vì điều kiện chỉ yêu cầu đến độ dài chuỗi ngày làm việc, nên ta sẽ chỉ quan tâm đến những ngày đứng đầu của chuỗi làm việc. Hay có thể nói nhân viên được chọn sẽ làm việc vào ngày đó, đồng thời được nghỉ vào ngày đứng ngay trước.

- Tại ngày này, ta thấy để chuỗi làm việc kéo dài ít nhất là 4 ngày thì 3 ngày tiếp theo người được chọn đều phải có ca trực.

Như vậy, ta biểu diễn điều kiện này như sau :

$$x_{n,d,1} + x_{n,d,2} = 0 \vee x_{n,d-1,1} + x_{n,d-1,2} = 1 \vee \sum_{j=d}^{d+3} (x_{n,j,1} + x_{n,j,2}) = 4 \quad (13)$$

với  $n \in \{1; 2; 3; 4\}, 2 \leq d \leq 25$

Hai điều kiện đầu tiên là để loại ra những ngày đứng giữa của chuỗi làm việc hoặc không có lịch làm việc, còn điều kiện thứ 3 chính là lập luận được phát biểu ở trên.

Tương tự với các điều dưỡng part-time:

$$\sum_{d=i}^{i+3} (x_{n,d,1} + x_{n,d,2}) \leq 3 \text{ với } n \in \{5; 6; 7; 8\}, i \in \mathbb{N}^*, 1 \leq i \leq 25 \quad (12)$$

$$x_{n,d,1} + x_{n,d,2} = 1 \vee x_{n,d-1,1} + x_{n,d-1,2} = 0 \vee \sum_{j=d}^{d+1} (x_{n,j,1} + x_{n,j,2}) = 2 \quad (13)$$

với  $n \in \{5; 6; 7; 8\}, 2 \leq d \leq 25$

Biểu thức trên cần tách một số ngày và xét riêng như sau:

- Ngày thứ 1 sẽ không có điều kiện về ngày trước đó.
- Ngày 26 và 27 sẽ không có đủ 3 ngày ở đằng sau, nên ta sẽ chỉnh số ngày liền sau thành 2 với trường hợp của ngày 26, và 1 với trường hợp của ngày 27.

### 3.2 Lập trình ràng buộc sử dụng công cụ OR-Tools

Để giải được bài toán với các ràng buộc đã được thể hiện như trên, ta sẽ sử dụng công cụ Google OR-tools bằng ngôn ngữ lập trình Python.

Đầu tiên ta khởi tạo một số biến ban đầu, chú ý rằng biến  $x_{n,d,s}$  ta đặt ở mục trên tương ứng với biến `shifts[(n, d, s)]`:

```
1 num_nurses = 8 #Number of nurses
2 num_shifts = 2 #Number of shifts : day and night
3 num_days = 28 #Number of days
4 nurses_name_list = ['An', 'Binh', 'Chau', 'Duong', 'Linh', 'Kiet', 'Giang', 'Hieu']
5 shifts_name = ['S', 'T'] #Name of shifts
6 shifts = {}
7 for n in all_nurses:
8     for d in all_days:
9         for s in all_shifts:
10             shifts[(n, d, s)] = model.new_bool_var()
```

Mỗi người làm nhiều nhất 1 ca mỗi ngày tương ứng với biểu thức (5):

```
1 for n in all_nurses:
2     for d in all_days:
3         model.add_at_most_one(shifts[(n, d, s)] for s in all_shifts)
```

Mỗi nhân viên full-time sẽ làm đúng tổng cộng 18 ca, nhân viên part-time là 10 ca, tương ứng với biểu thức (6) và (7):



```
1 model.add(sum(shifts_fulltime_worked) == 18)
2 model.add(sum(shifts_parttime_worked) == 10)
```

Cùng với đó là số ca sáng và tối được yêu cầu ở mỗi ngày làm việc, tương ứng với biểu thức (8) và (9):

```
1 model.add(sum(day_shifts) == 3)
2 model.add(sum(night_shifts) == 1)
```

Tiếp theo là các ràng buộc về số ngày làm việc trong một tuần, tương ứng với biểu thức (10) và (11):

```
1 model.Add(fulltime_weekly_shifts >= 4)
2 model.Add(fulltime_weekly_shifts <= 5)
3 model.Add(parttime_weekly_shifts >= 2)
4 model.Add(parttime_weekly_shifts <= 3)
```

Kế đến ta thể hiện ràng buộc về chuỗi ngày làm việc, đầu tiên là biểu diễn chặn trên, tương ứng với biểu thức (12):

```
1 model.Add(sum(shifts[(n, j, s)] for j in range(d,d+7) for s in range(2)) <= 6)
```

Còn về ràng buộc chặn dưới, do biểu thức (13) gồm 3 điều kiện nằm trong *phép toán OR*, ta ghi chúng vào ba biến và sử dụng hàm `model.AddBoolOr` với 3 biến vừa tạo để thể hiện mối quan hệ "hoặc":

```
1 model.Add(sum(shifts[(n, d, s)] for s in range(2)) == 0)
2 .OnlyEnforceIf(condition_1_1)
3 model.Add(sum(shifts[(n, d-1, s)] for s in range(2)) == 1)
4 .OnlyEnforceIf(condition_1_2)
5 model.Add(sum(shifts[(n, j, s)] for j in range(d,d+4) for s in range(2)) == 4)
6 .OnlyEnforceIf(condition_1_3)
7 model.AddBoolOr([condition_1_1, condition_1_2, condition_1_3])
```

Đoạn mã tương ứng với các nhân viên làm việc part-time hoàn toàn được biểu diễn tương tự, nên ta sẽ không trình bày thêm ở đây.

### 3.3 Kết quả

Khi xây dựng mô hình (Xem trong tệp "[ToanMoHinh2024v1\\_code.ipynb](#)" hoặc [tại đây](#)), ta chỉnh tham số `solution_limit = 1` để giới hạn một kết quả in ra (vì có rất nhiều kết quả thỏa mãn) và chạy đoạn mã và thu được kết quả trong [Bảng 3 - Phụ lục](#) hoặc tệp "[ToanMoHinh2024v1\\_output.xlsx](#)".

## 4 Đánh giá - Hướng phát triển

Mặc dù không giống như lời giải ta đặt ra từ trước, tuy vậy kết quả vẫn thỏa mãn đầy đủ các *hard-constraints* và *soft-constraints* được đưa ra.

Ở lời giải trên, ta đã đưa các điều kiện mềm thành các điều kiện cứng để tối ưu hóa độ hiệu quả của mô hình. Tuy nhiên, trong đa số các trường hợp khác, ta không thể đáp ứng được toàn bộ những điều kiện này, dẫn đến mô hình không thể tìm ra lời giải nào. Khi đó ta sẽ phải coi các điều kiện mềm này như ban đầu và dùng đến các công cụ tối ưu hóa để giải quyết.

Trong thực tế, việc xếp lịch còn có thể bao gồm thêm nhiều yêu cầu như kế hoạch mong muốn của từng nhân viên, hay việc phân bổ các ca sáng-tối sao cho phù hợp với các nhân viên, làm cho bài toán trở nên phức tạp hơn. Dù vậy, nếu có thể diễn giải logic và mô hình hóa các điều kiện này, ta hầu như có thể giải quyết chúng và xây dựng mô hình như đã thực hiện ở trên.

Trong tương lai, với việc sử dụng một công cụ khá mạnh cho việc xử lý các bài toán ràng buộc đó là *Google Or-Tools*, mô hình được xây dựng hoàn toàn có thể được mở rộng ra và xử lý những vấn đề phức tạp hơn, lớn hơn với những số liệu trong thực tế. Chúng tôi sẽ phát triển mô hình để có thể đáp ứng được nhiều ràng buộc phức tạp hơn, nhằm có thể đưa mô hình vào ứng dụng trong thực tế tại các trường học, bệnh viện, trạm xá,...

## Lời cảm ơn

Trước tiên chúng tôi xin được cảm ơn ban tổ chức, ban giám khảo và quý bạn đọc đã dành thời gian để đọc bản báo cáo của nhóm chúng tôi. Bản báo cáo này là thành quả của một quá trình phân tích, tìm hiểu cũng như tham khảo các nguồn tài liệu, các kiến thức về toán học, lập trình cùng các vấn đề liên quan. Sau đó được nghiên cứu và sử dụng các kiến thức, công cụ đã nghiên cứu để trình bày ra một lời giải hiệu quả cho đề bài được nhận.

Xin cảm ơn ban tổ chức của cuộc thi vì đã tạo ra một sân chơi bổ ích cho chúng tôi được trải nghiệm, giao lưu và phát triển những kỹ năng quan trọng. Bản báo cáo có thể còn một số sai sót bị bỏ lại, mong ban giám khảo và bạn đọc có thể thông cảm cho chúng tôi.

Xin chân thành cảm ơn.

## Thông tin liên hệ

Các thắc mắc, góp ý về bài dự thi, xin vui lòng liên hệ nhóm tác giả:

Nguyễn Trần Trung - Email: [trung7clqd@gmail.com](mailto:trung7clqd@gmail.com)

Phạm Công Hoàng - Email: [hoangphamconglc2212@gmail.com](mailto:hoangphamconglc2212@gmail.com)

Nguyễn Minh Tân - Email: [minhtantn2006@gmail.com](mailto:minhtantn2006@gmail.com)

**Phụ lục - Các bảng sử dụng trong bài thi**

	01	02	03	04	05	...	26	27	28
1									
2									
...									
7									
8									

Bảng 1: Bảng 8x28

	01	02	03	04	05	06	07	08	09	10	11	12	13	14
1	S	S	S	S				S	S	S	S	S		
2	S	S	S	S				S	S	S	S	S		
3	S	S	S	S				S	S	S	S	S		
4	T	T	T	T				T	T	T	T	T		
5					S	S	S						S	S
6					S	S	S						S	S
7					S	S	S						S	S
8					T	T	T						T	T

	15	16	17	18	19	20	21	22	23	24	25	26	27	28
1	S	S	S	S				S	S	S	S	S		
2	S	S	S	S				S	S	S	S	S		
3	S	S	S	S				S	S	S	S	S		
4	T	T	T	T				T	T	T	T	T		
5					S	S	S						S	S
6					S	S	S						S	S
7					S	S	S						S	S
8					T	T	T						T	T

Bảng 2: Một mô hình thỏa mãn dễ dàng tìm ra

	01	02	03	04	05	06	07	08	09	10	11	12	13	14
1	S	S	S	S			S	S	S	S				S
2	S	S	T	S						S	S	S	T	
3			S	T	S	T			S	S	S	S	S	
4	T	T	S	S	S			S	T	T	S			
5					S	S		T	S					S
6					T	S	T					S	S	
7						S	S				T	T		S
8	S	S					S	S					S	T

	15	16	17	18	19	20	21	22	23	24	25	26	27	28
1	S	S	S			S	S	S	S	S	T			
2		S	T	T	S	S			S	S	S	T		T
3			S	S	T	T				S	S	S	S	S
4		S	S	S	S				S	T	S	S	S	
5	T						S	T				S	S	
6				S	S	S							T	S
7	S	T					S	S	T					
8	S						T	S						S

Bảng 3: Kết quả thu được

## Tài liệu

- [1] Kadir Büyüközkan **and** Ahmet Sarucan. “Applicability of artificial bee colony algorithm for nurse scheduling problems”. *in* *Int. J. Comput. Intell. Syst.*: 7 (2014), **pages** 121–136.
- [2] Ricardo Soto **and** others. “Nurse and Paramedic Rostering with Constraint Programming: A Case Study”. *in* *Romanian Journal of Information Science and Technology*: 16 (2013), **pages** 52–64.
- [3] B Cheang **and** others. “Nurse rostering problems—a bibliographic survey”. *in* *European Journal of Operational Research*: 151.3 (2003), **pages** 447–460.
- [4] Google. *Google OR-Tools*. 2024. URL: <https://developers.google.com/optimization/>.