

## **KẾT QUẢ XÂY DỰNG THUẬT TOÁN XẤP XỈ GIẢI MÔ HÌNH LẬP LỊCH TẠI BỆNH VIỆN**

**Vũ Vinh Quang<sup>1\*</sup>, Phạm Thanh Huyền<sup>2</sup>**

<sup>1</sup>*Trường Đại học Công nghệ thông tin và Truyền thông, Đại học Thái Nguyên*

<sup>2</sup>*Khoa Công nghệ thông tin, Trường Đại học Hạ Long*

\* *Email: vvquang@ictu.edu.vn*

*Ngày nhận bài: 22/9/2022*

*Ngày nhận bài sửa sau phản biện: 11/11/2022*

*Ngày chấp nhận đăng: 15/11/2022*

### **TÓM TẮT**

Trong thực tế, mô hình lập lịch là một mô hình tối ưu được nhiều nhà nghiên cứu quan tâm do độ phức tạp lớn và tính ứng dụng cao trong thực tế. Việc tìm lời giải tối ưu trong thời gian đa thức là một thách thức lớn, do đó trong thực tế người ta thường nghiên cứu một số lời giải gần tối ưu được thực hiện bằng các thuật toán xấp xỉ mà điển hình là các thuật toán tham lam và thuật toán tiến hóa dựa trên cơ chế của giải thuật di truyền. Nội dung chính của bài báo là trình bày các kết quả nghiên cứu về thuật toán xấp xỉ và giải thuật di truyền, trên cơ sở xây dựng và phân tích mô hình bài toán lập lịch tại các phòng khám của các bệnh viện, đề xuất các thuật toán xấp xỉ để giải quyết mô hình bài toán, tiến hành thử nghiệm trên mô hình cụ thể để khẳng định tính hiệu quả của các thuật toán đã đề xuất.

**Từ khóa:** *bài toán lập lịch, độ phức tạp thuật toán, giải thuật di truyền, thuật toán tham lam, tính toán tiến hóa.*

### **THE FINDINGS ON CONSTRUCTING A NEW APPROXIMATION ALGORITHM TO SOLVE THE HOSPITAL'S SCHEDULING MODEL**

#### **ABSTRACT**

In reality, the scheduling model is an optimal model that many researchers are interested in due to its great complexity and high applicability in practice. Finding out the optimal solution in polynomial time is a big challenge, therefore scientists frequently study some near-optimal solutions implemented by approximation algorithms, typically greedy and evolutionary algorithms based on the mechanics of genetic algorithms. The main content of this paper is to present the research findings on approximation algorithms and genetic algorithms. On the basis of constructing and analyzing the scheduling model in the hospitals' clinics, this paper proposed approximation algorithms to solve this problem and conducted experiments on specific models to confirm the effectiveness of the proposed algorithms.

**Keywords:** *algorithm complexity, evolutionary computation, genetic algorithm, greedy algorithm, scheduling problem.*

## 1. ĐẶT VẤN ĐỀ

Trong các lớp bài toán rời rạc thì mô hình các bài toán thuộc lớp NPC là lớp các bài toán chưa tìm được lời giải tối ưu trong thời gian đa thức. Điển hình trong lớp các bài toán đó thì mô hình lập lịch biểu tối ưu là mô hình được rất nhiều các nhà khoa học quan tâm do tính phức tạp của mô hình cũng như tính ứng dụng cao trong thực tế. Đối với mô hình lập lịch, với các hệ ràng buộc phức tạp thì việc tìm lời giải tối ưu trong thời gian đa thức là một thách thức lớn, do đó, trong thực tế, người ta thường nghiên cứu một số lời giải gần tối ưu. Các thuật toán này thường được thực hiện bằng các thuật toán xấp xỉ mà điển hình là các thuật toán tham lam (Martello & Toth, 1990; Wirth, 1976) và thuật toán tiến hóa dựa trên cơ chế của giải thuật di truyền (Eiben và cs., 1994; Goldberg, 1989; Nguyễn Đình Thúc, 2001).

Trong thực tế hiện nay, yêu cầu lập lịch biểu thường xuất hiện dưới dạng điều phối tối ưu các phương tiện vận tải, phân công giảng dạy tại các cơ sở giáo dục đào tạo, lập lịch trực tại các bệnh viện. Tùy thuộc vào yêu cầu cụ thể của từng đơn vị, tại Việt Nam đã có nhiều tác giả đưa ra các phương án lập lịch biểu tối ưu dựa trên các thuật toán về đồ thị, tính toán tiến hóa, thuật toán tham lam, trí tuệ nhân tạo (Mui và cs., 2012; Nguyễn Hữu Mùi & Vũ Đình Hòa, 2012; Trương Quốc Định & Nguyễn Thanh Hải, 2016). Tuy nhiên, xuất phát từ yêu cầu của mỗi đơn vị là khác nhau sẽ dẫn đến các hệ ràng buộc là khác nhau, do đó không thể tồn tại lời giải chung cho mọi mô hình lập lịch. Vì vậy, với mỗi một mô hình, xuất phát từ các hệ ràng buộc cụ thể, chúng ta cần đưa ra những giải pháp phù hợp để mô tả các hệ ràng buộc, xây dựng hàm mục tiêu, từ đó thiết kế thuật toán để xác định lịch biểu tối ưu cho mô hình tương ứng.

Trong bài báo này, chúng tôi trình bày các kết quả nghiên cứu về nguyên tắc thiết kế các thuật toán xấp xỉ và giải thuật di truyền. Trên cơ sở xây dựng mô hình, chúng tôi phân tích chi tiết các hệ ràng buộc mang tính chất đặc trưng của bài toán lập lịch tại

các phòng khám của các bệnh viện, từ đó đề xuất hai thuật toán trên cơ sở của thuật toán tham lam và thuật toán di truyền để tìm lịch biểu tối ưu cho bài toán lập lịch, cuối cùng tiến hành thử nghiệm các thuật toán trên bộ dữ liệu thực để khẳng định tính hiệu quả của các thuật toán đã đề xuất.

Cấu trúc của bài báo gồm phần 1 là phần đặt vấn đề, phần 2 trình bày một số kết quả nghiên cứu bao gồm nguyên tắc thiết kế thuật toán xấp xỉ và thuật toán di truyền, kết quả xây dựng mô hình lập lịch và đề xuất các thuật toán cùng kết quả thực hiện các thuật toán cho mô hình lập lịch tại bệnh viện, phần 3 đưa ra kết luận và hướng phát triển. Các kết quả tính toán trong bài báo được lập trình trên môi trường Matlab 7.1.

## 2. PHƯƠNG PHÁP NGHIÊN CỨU

Dựa trên lý thuyết về các thuật toán xấp xỉ, xuất phát từ yêu cầu tối ưu của bài toán lập lịch, chúng tôi phân tích chi tiết mối quan hệ ràng buộc của các thành phần tham gia (các bác sĩ, các y tá, các ca trực, các phòng khám) của bài toán, từ đó đề xuất hai thuật toán tìm nghiệm xấp xỉ cho mô hình. Để đánh giá tính hiệu quả của các thuật toán đã đề xuất, chúng tôi xây dựng mô hình lập lịch với các yêu cầu ràng buộc được mô tả bằng các ma trận nhị phân cho trước, từ đó tiến hành thử nghiệm các thuật toán đề xuất tìm lịch biểu tối ưu trên môi trường Matlab, so sánh và đánh giá hiệu quả của các thuật toán.

## 3. CÁC KẾT QUẢ NGHIÊN CỨU

### 3.1. Một số thuật toán xấp xỉ

#### 3.1.1. Thuật toán tham lam (Wirth, 1976)

Thuật toán tham lam (Greedy algorithms) là một thuật toán giải quyết theo tư tưởng heuristic để tìm kiếm lựa chọn tối ưu địa phương ở mỗi bước đi với hy vọng tìm được tối ưu toàn cục. Ưu điểm của thuật toán tham lam là dễ đề xuất theo tư tưởng tự nhiên. Trong hầu hết các bài toán tối ưu, để nhận được nghiệm tối ưu, chúng ta có thể đưa về một dãy các lựa chọn. Ý tưởng của chiến lược tham lam là tại mỗi bước ta sẽ lựa chọn một phương án được xem là tốt nhất. Tùy theo từng bài toán mà ta đưa ra tiêu chuẩn lựa chọn thích hợp.

Các thuật toán tham lam nói chung là đơn giản và hiệu quả, tuy nhiên nghiệm thu được không chắc là tối ưu. Phương pháp được tiến hành theo nhiều bước, tại mỗi bước theo một lựa chọn nào đó (xác định bằng một hàm chọn), lời giải của bài toán được bổ sung dần từng bước. Phương pháp tham lam cần tìm một trật tự hợp lý để duyệt dữ liệu nhằm đạt được mục tiêu một cách chắc chắn và nhanh chóng. Thông thường thuật toán tham lam có tốc độ tốt hơn hẳn so với các thuật toán tối ưu tổng thể.

#### Mô hình thuật toán tham lam

Kí hiệu tập  $X$  là tập nghiệm tối ưu,  $A$  là tập các phần tử ứng viên. Ta xây dựng tập  $X$  dần từng bước bắt đầu từ tập rỗng, tại mỗi bước ta sẽ chọn một phần tử “tốt nhất” trong các phần tử còn lại của  $A$  để đưa vào  $X$ . Việc lựa chọn một phần tử như thế ở mỗi bước được hướng dẫn bởi hàm chọn. Nếu khi thêm phần tử được chọn vào tập  $X$  mà  $X$  chưa đầy thì ta sẽ mở rộng  $X$  bằng các bước chọn tiếp sau.

Thuật toán tổng quát được mô tả như sau:

#### Procedure Greedy( $A, X$ )

Input  $A$  // Tập các đối tượng cho trước

Output  $X$  // Phương án tối ưu

Begin

+  $X \leftarrow \emptyset$ ;

+ While  $A \neq \emptyset$  do

  Begin

$x \leftarrow \text{Select}(A)$ ;  $A \leftarrow A - \{x\}$ ;

    if  $X \cup \{x\}$  thỏa mãn then

$X \leftarrow X \cup \{x\}$ ;

  end;

end.

Trong thủ tục tổng quát trên, Select là hàm chọn cho phép chọn từ tập  $A$  một phần tử được xem là tốt nhất để đưa vào  $X$ .

Mấu chốt của thuật toán là phương pháp xây dựng hàm Select. Độ phức tạp của thuật toán phụ thuộc vào độ phức tạp khi xây dựng hàm Select. Trong thực tế, đối với rất nhiều mô hình tối ưu rời rạc, người ta thường lựa chọn phương pháp thiết kế thuật toán theo tư tưởng tham lam để tìm nghiệm xấp xỉ tối ưu.

#### 3.1.2. Thuật toán di truyền (Goldberg, 1989)

Thuật toán di truyền (GA-Genetic Algorithm) do D.E. Goldberg đề xuất là kỹ thuật phỏng theo quá trình thích nghi tiến hóa của các quần thể sinh học dựa trên học thuyết Darwin. Trong GA, việc tìm kiếm nghiệm tối ưu được bắt đầu với một quần thể ban đầu, quần thể thế hệ kế tiếp được sinh ra bằng các hoạt động lai ghép và đột biến ngẫu nhiên trong các quá trình tiến hóa. Ở mỗi bước, cá thể nào tốt hơn sẽ tồn tại và ngược lại sẽ bị đào thải. Để thiết kế GA cần xác định được phương pháp khởi tạo quần thể ban đầu, nguyên tắc xây dựng hàm thích nghi (hàm mục tiêu) và các toán tử di truyền.

Trong công nghệ thông tin, GA là một lĩnh vực mới có tốc độ phát triển rất nhanh. Có thể chia thành các hướng:

+ Genetic Algorithm: Dựa vào quá trình di truyền trong tự nhiên để cải tiến lời giải qua các thế hệ bắt nguồn từ một tập các lời giải ban đầu.

+ Quy hoạch tiến hoá (Evolutionary Programming – EP): Dựa vào quy luật tiến hoá, tìm phương pháp kết hợp để giải quyết trọn vẹn một bài toán từ một lớp các phương pháp giải quyết được một số phần của bài toán.

+ Các chiến lược tiến hoá (Evolutionary Strategies – ES): Dựa trên một số chiến lược ban đầu, thực hiện tiến hoá để tạo ra những chiến lược mới phù hợp với môi trường thực tế một cách tốt nhất.

+ Lập trình di truyền (Genetic Programming – GP): Mở rộng GA trong lĩnh vực các chương trình của máy tính nhằm sinh ra một cách tự động các chương trình máy tính giải quyết một cách tối ưu một bài toán cụ thể.

+ Các hệ thống phân loại (Classifier Systems – CS): Các thuật toán GA đặc biệt được dùng trong việc học máy và việc phát hiện các quy tắc trong các hệ dựa trên các quy tắc.

Ngày nay, thuật toán GA càng trở nên quan trọng, đặc biệt là trong lĩnh vực tối ưu hoá với lớp các bài toán NP, NPC chưa có giải thuật hiệu quả.

### Các khái niệm cơ bản trong GA

+ Cá thể: biểu diễn một phương án của bài toán.

+ Quần thể: tập hợp các cá thể có cùng một số đặc điểm. Đây là một tập các lời giải của một bài toán.

+ Toán tử chọn lọc: chọn các cá thể có độ thích nghi tốt để đưa vào thế hệ tiếp theo. Thông thường việc chọn lọc được thông qua hàm mục tiêu.

+ Toán tử lai ghép: sinh ra một lời giải khác từ các lời giải cha mẹ (thế hệ sau).

+ Đột biến: tạo ra một cá thể mới tốt hơn hoặc xấu hơn cá thể ban đầu.

Khi đó thuật toán GA được mô tả như sau:

#### Procedure GA

Bước 1. Xác lập các tham số ban đầu của bài toán.

Bước 2. Khởi tạo: sinh ngẫu nhiên một quần thể gồm  $n$  cá thể (là  $n$  lời giải ban đầu của bài toán).

Bước 3. Xây dựng quần thể thế hệ sau:

3.1 Tiến hành lai ghép các cặp bố – mẹ với một xác suất lai ghép được chọn để tạo ra một cá thể mới đưa tiếp vào quần thể mới.

3.2 Tính độ thích nghi của tất cả các cá thể thuộc quần thể mới.

3.3 Chọn lọc  $n$  cá thể mới từ quần thể mới theo độ thích nghi để tạo ra quần thể cho thế hệ tiếp sau.

3.4 Đột biến  $k$  cá thể

Bước 4. Kiểm tra điều kiện dừng: nếu điều kiện được thỏa mãn thì thuật toán kết thúc và trả về lời giải tốt nhất chính là quần thể hiện tại, ngược lại quay lại bước 3.

#### Các cơ chế trong GA:

+ Mã hóa: Thông thường người ta sử dụng một trong các phương pháp (mã hoá nhị phân, mã hóa hoán vị, mã hóa số thực) xuất phát từ cách mô tả phương án tối ưu của từng bài toán.

+ Khởi tạo quần thể ban đầu: Tùy vào từng bài toán cụ thể mà ta có các phương pháp khởi

tạo khác nhau. Chất lượng của quần thể ban đầu càng cao thì lời giải mà GA đưa ra càng tốt. Thông thường chúng ta dùng phương pháp ngẫu nhiên để khởi tạo.

+ Xác định hàm thích nghi: Đối với bài toán tối ưu hóa thì hàm thích nghi chính là hàm mục tiêu của bài toán.

+ Cơ chế lựa chọn: Thường sử dụng phương pháp lựa chọn ngẫu nhiên theo xác suất từ quần thể lấy  $n$  cá thể để thực hiện lai ghép hoặc chọn tất cả các cá thể cùng tham gia lai ghép.

+ Toán tử lai ghép: Thường chọn các phương pháp lai ghép đơn điểm, đa điểm, mặt nạ hoặc lai ghép mã hóa số thực.

+ Toán tử đột biến: Thường thay đổi một giá trị tùy ý trong dữ liệu của cá thể theo một xác suất nào đó đủ nhỏ.

Thuật toán GA là một thuật toán được cài đặt đơn giản, độ phức tạp là đa thức. Hiện nay có rất nhiều các phần mềm trong các lĩnh vực tối ưu hiện nay được cài đặt theo tư tưởng GA.

### 3.2. Mô hình lập lịch trực tại phòng khám

Trong các bệnh viện hiện nay, vấn đề phân công trực tại các phòng khám là một công việc cấp thiết và thường xuyên. Do yêu cầu về công tác chuyên môn của bệnh viện cũng như lịch công tác của từng bác sĩ và y tá nên việc phân công lịch trực cần phải đáp ứng những yêu cầu rất phức tạp và hầu như không thể tìm được lời giải tối ưu.

Xuất phát từ tính phức tạp của mô hình, chúng tôi sẽ đề xuất một mô hình lập lịch trực và xây dựng thuật toán tìm lịch biểu tối ưu tại các phòng khám dựa trên thuật toán tham lam và giải thuật di truyền.

Chúng ta xét mô hình bài toán: giả sử có  $NBS$  bác sĩ và  $NYT$  các y tá, mỗi bác sĩ và y tá gồm các thông tin: họ tên và chuyên môn được đào tạo,  $NPK$  phòng khám và  $NCT$  các ca trực được phân biệt bởi số hiệu phòng khám và số hiệu ca trực.

Chúng ta đưa ra hai yêu cầu:

+  $YC_1$ : Mỗi bác sĩ và y tá chỉ được trực tại một số phòng khám phù hợp với chuyên môn đã được đào tạo.

+  $YC_2$ : Cho phép mỗi bác sĩ và y tá được đăng ký các buổi trực mà cá nhân sẵn sàng nhận trực theo lịch phân công.

Hãy xếp lịch trực cho các bác sĩ và y tá tại các phòng khám trong tất cả các ca trực để sao cho tại mọi ca trực, mỗi phòng khám phải có đầy đủ một bác sĩ và một y tá phù hợp với chuyên môn đồng thời số ca trực của các bác sĩ là tương đương nhau, các y tá là tương đương nhau. Giả thiết là xếp kín tất cả các ca trực trong lịch.

**Nhận xét:** Mô hình như trên sẽ đảm bảo yêu cầu về chuyên môn, đáp ứng nhu cầu của tất cả các đối tượng và đồng thời công bằng về việc phân công nhiệm vụ cho mọi bác sĩ và y tá. Đây là một mô hình phù hợp với tình hình thực tế ở các bệnh viện tại Việt Nam.

Xuất phát từ mô hình trên, chúng ta sẽ xây dựng mô hình toán học chi tiết cho bài toán.

Đưa vào các ký hiệu:

– Tập  $BS = \{1, 2, \dots, NBS\}$  số hiệu các bác sĩ.

– Tập  $YT = \{1, 2, \dots, NTY\}$  số hiệu các y tá.

– Tập  $PK = \{1, 2, \dots, NPK\}$  số hiệu các phòng khám.

– Tập  $CT = \{1, 2, \dots, NCT\}$  số hiệu các buổi trong lịch trực.

– Mảng  $PBS(s, p)$  biểu diễn sự phù hợp chuyên môn giữa bác sĩ và phòng khám:  $PBS(s, p) = 1$  khi bác sĩ số hiệu  $s$  phù hợp chuyên môn với phòng khám  $p$ ,  $PBS(s, p) = 0$  khi không phù hợp.

– Mảng  $PYT(s, p)$  biểu diễn sự phù hợp chuyên môn giữa y tá và phòng khám:  $PYT(s, p) = 1$  khi y tá số hiệu  $s$  phù hợp chuyên môn với phòng khám  $p$ ,  $PYT(s, p) = 0$  khi không phù hợp.

– Mảng trạng thái  $MBS(s, t)$  trạng thái sẵn sàng nhận trực của bác sĩ có số hiệu  $s$  trong ca trực thứ  $t$ :  $MBS(s, t) = 1$  khi sẵn sàng,  $MBS(s, t) = 0$  khi không sẵn sàng.

– Mảng trạng thái  $MYT(s, t)$  trạng thái sẵn sàng nhận trực của y tá có số hiệu  $s$  trong ca trực thứ  $t$ :  $MYT(s, t) = 1$  khi sẵn sàng,

$MYT(s, t) = 0$  khi không sẵn sàng.

Các biến số cần xác định:

– Các biến  $XBS(p, t) = k$  khi bác sĩ số hiệu  $k$  được xếp vào phòng khám  $p$  trong buổi trực  $t$ .

– Các biến  $XYT(p, t) = k$  khi y tá số hiệu  $k$  được xếp vào phòng khám  $p$  trong buổi trực  $t$ .

– Các biến  $CBS(s)$  ghi lại tổng số buổi trực của bác sĩ  $s$  trong toàn lịch trực.

– Các biến  $CYT(s)$  ghi lại tổng số buổi trực của y tá  $s$  trong toàn lịch trực.

Các điều kiện ràng buộc:

$R_1$ : Tại một thời điểm  $t$ , một bác sĩ chỉ được trực nhiều nhất tại một phòng khám.

$R_2$ : Tại một thời điểm  $t$ , một y tá chỉ được trực nhiều nhất tại một phòng khám.

$R_3$ : Chỉ xếp lịch trực cho các bác sĩ sẵn sàng trong ca trực.

$R_4$ : Chỉ xếp lịch trực cho các y tá sẵn sàng trong ca trực.

$R_5$ : Các bác sĩ và y tá chỉ được phép trực tại các phòng khám phù hợp về chuyên môn.

$R_6$ : Tại mọi thời điểm, các phòng khám đều phải có đầy đủ một bác sĩ và một y tá trực.

Như vậy, bài toán đưa về yêu cầu hãy xây dựng bảng phân công trực tại các phòng khám cho tất cả các bác sĩ và y tá trong bệnh viện thỏa mãn tất cả các ràng buộc từ  $R_1$  đến  $R_6$  sao cho tổng số các buổi trực của các bác sĩ được phân công là tương đương nhau, số các buổi trực của các y tá được phân công là tương đương nhau.

**Nhận xét:**

+ Chúng ta thấy rằng nếu phương án xếp lịch mà thỏa mãn tất cả các ràng buộc từ  $R_1$  đến  $R_6$  thì hai yêu cầu  $YC_1$  và  $YC_2$  đặt ra đối với mô hình luôn thỏa mãn.

+ Bài toán trên là một dạng bài toán lập lịch với nhiều ràng buộc phức tạp. Để nhận được lời giải đúng là rất khó thực hiện, trong nhiều trường hợp chúng ta không thể xác định được lịch biểu tối ưu. Sau đây, chúng ta sẽ đề xuất hai thuật toán xấp xỉ giải bài toán này.



### 3.3. Đề xuất thuật toán tham lam

#### 3.3.1. Tư tưởng

Do các bác sĩ và y tá trong bệnh viện là độc lập nên chúng ta cũng chỉ cần phân lịch  $L_1$  cho các bác sĩ và phân lịch  $L_2$  cho các y tá sau đó kết hợp lại chúng ta sẽ thu được lịch trực chung. Hiển nhiên, hai bài toán lập lịch  $L_1$  và  $L_2$  là tương đương. Như vậy chúng ta chỉ cần xét bài toán phân lịch cho các bác sĩ là đủ.

Kí hiệu ma trận  $L_1 = [L_1(i, j)]$  là lịch trực cần tìm trong đó  $L_1(i, j) = k$  được hiểu là bác sĩ  $k$  sẽ trực tại phòng khám  $i$  trong ca trực thứ  $j$  trong lịch ( $k = 1..NBS$ ;  $i = 1:NPK$ ;  $j = 1:NBT$ ).

Hiển nhiên bài toán cần xây dựng phương án trực  $L_1$  thỏa mãn hai ràng buộc  $MBS(k, i) = 1$  và  $PBS(k, j) = 1$  sao cho số buổi trực của các bác sĩ là xấp xỉ bằng nhau.

Bởi vì các vai trò của các bác sĩ là tương đương nên nếu chúng ta xếp lần lượt các bác sĩ sẵn sàng về chuyên môn và thời gian lần lượt vào các ca trực theo thứ tự tăng dần (khi hết số bác sĩ thì lại quay trở lại bác sĩ đầu tiên), quá trình sẽ dừng lại khi lịch đầy. Khi đó rõ ràng số buổi trực của các bác sĩ sẽ xấp xỉ bằng nhau.

Từ đó, chúng ta đề xuất thuật toán theo tư tưởng tham lam như sau:

#### 3.3.2. Thuật toán Greedy\_QH

Input: Các mảng:  $PBS, PYT, MBS, MYT$

Output: Lịch  $L_1$

Bước 1: Khởi động ma trận  $L_1$  là rỗng (chưa xếp lịch)

Bước lặp: Lần lượt xét các bác sĩ  $k = 1..NBS$

+ Xét một ô  $L_1(i, j)$  còn trống,

+ Nếu bác sĩ  $k$  thỏa mãn điều kiện  $MBS(k, i) = 1$  và  $PBS(k, j) = 1$  thì xếp bác sĩ  $k$  vào ô  $(i, j)$  tức là gán  $L_1(i, j) = k$ ;

+ Tăng  $k := k + 1$ ;

+ Nếu  $k = NBS$  thì quay lại  $k = 1$ .

Quá trình dừng lại khi tất cả các ô  $L_1(i, j)$  đã đầy.

Quá trình xếp lịch trực  $L_2$  cho các y tá là hoàn toàn tương tự.

Kết hợp hai lịch  $L_1$  và  $L_2$  ta thu được lịch trực chung.

Hiển nhiên độ phức tạp của thuật toán là  $O(NBS \times NPK \times NBT \times K)$  trong đó  $K$  là số lần lặp trong bước lặp.

#### Nhận xét:

+ Vì thuật toán được xây dựng theo tư tưởng tham lam nên chúng ta không cần xây dựng hàm mục tiêu, kết quả thu được của thuật toán sẽ đảm bảo số ca trực của các bác sĩ là xấp xỉ bằng nhau, số ca trực của các y tá cũng xấp xỉ bằng nhau.

+ Thuật toán dễ dàng được cài đặt bằng các ngôn ngữ lập trình Matlab hoặc C++.

### 3.4. Đề xuất thuật toán di truyền

#### 3.4.1. Xây dựng cấu trúc dữ liệu

+ Lựa chọn cấu trúc cá thể: kí hiệu một phương án xếp lịch là cặp hai ma trận  $Z = [ZBS(NPK, NCT); ZYT(NPK, NCT)]$ , trong đó  $ZBS(p, t) = k$  tương ứng với bác sĩ có số hiệu  $k$  trực phòng khám  $p$  tại ca trực thứ  $t$ ,  $ZYT(p, t) = k$  tương ứng với y tá có số hiệu  $k$  trực phòng khám  $p$  tại ca trực thứ  $t$ .

Như vậy, tập hợp các phương án chính là tập hợp các cặp ma trận các phần tử là các số nguyên dương có giá trị là các số hiệu của các bác sĩ hoặc của các y tá. Điều kiện phân công hợp lệ là trên một cột của các ma trận  $ZBS$  và  $ZYT$  không có các phần tử có giá trị trùng nhau tức là tại một thời điểm mỗi bác sĩ hoặc y tá chỉ được trực tại một phòng khám.

+ Hàm  $CBS(k)$  có giá trị bằng tổng tất cả các phần tử trong ma trận phương án  $ZBS$  thỏa mãn điều kiện  $ZBS(p, t) = k$ . Như vậy  $CBS(k)$  chính là tổng số buổi trực của bác sĩ  $k$  trong toàn lịch trực.

+ Hàm  $CYT(k)$  có giá trị bằng tổng tất cả các phần tử trong ma trận phương án  $ZYT$  thỏa mãn điều kiện  $ZYT(p, t) = k$ , như vậy,  $CYT(k)$  chính là tổng số buổi trực của y tá  $k$  trong toàn lịch trực.

### 3.4.2. Các ràng buộc của bài toán

+  $R_1$ : Tại một thời điểm  $t$ , một bác sĩ chỉ được trực nhiều nhất là một phòng khám sẽ tương đương với điều kiện: trên một cột của ma trận  $ZBS$  không tồn tại hai phần tử bằng nhau. Xây dựng hàm  $R_1(ZBS)$  để kiểm tra điều kiện  $R_1$ .

+  $R_2$ : Tại một thời điểm  $t$ , một y tá chỉ được trực nhiều nhất là một phòng khám sẽ tương đương với điều kiện: trên một cột của ma trận  $ZYT$  không tồn tại hai phần tử bằng nhau. Xây dựng hàm  $R_2(ZYT)$  để kiểm tra điều kiện  $R_2$ .

+  $R_3$ : Chỉ xếp lịch trực cho các bác sĩ sẵn sàng trong buổi trực sẽ tương đương với điều kiện: nếu  $ZBS(p, t) = s$  thì  $MBS(s, t) = 1$  hay  $MBS(ZBS(p, t), t) = 1$ .

+  $R_4$ : Chỉ xếp lịch trực cho các y tá sẵn sàng trong buổi trực sẽ tương đương với điều kiện: nếu  $ZYT(p, t) = s$  thì  $MYT(s, t) = 1$  hay  $MYT(ZYT(p, t), t) = 1$ .

+  $R_5$ : Các bác sĩ và y tá chỉ được phép trực tại các phòng khám phù hợp về chuyên môn đào tạo sẽ tương đương với điều kiện:

Nếu  $ZBS(p, t) = s$  thì  $PBS(s, p) = 1$  hay  $PBS(ZBS(p, t), p) = 1$ .

Nếu  $ZYT(p, t) = s$  thì  $PYT(s, p) = 1$  hay  $PYT(ZYT(p, t), p) = 1$ .

Kết hợp ba điều kiện  $R_3$ ,  $R_4$  và  $R_5$ , điều kiện thỏa mãn đồng thời chính là

$$MBS(ZBS(p, t), t) \times MYT(ZYT(p, t), t) \times PBS(ZBS(p, t), p) \times PYT(ZYT(p, t), p) = 1$$

với mọi  $1 \leq p \leq NPK; 1 \leq t \leq NCT$ ;

Xây dựng hàm  $R_{345}(Z)$  kiểm tra điều kiện  $R_3$ ,  $R_4$  và  $R_5$ .

+  $R_6$ : Tại mọi thời điểm, các phòng khám đều phải có bác sĩ trực và y tá sẽ tương đương với tất cả các phần tử trong cặp ma trận  $Z = [ZBS; ZYT]$  đều dương.

$$ZBS(p, t) > 0; ZYT(p, t) > 0;$$

$$1 \leq p \leq NPK; 1 \leq t \leq NCT;$$

Xây dựng hàm  $R_6(Z)$  kiểm tra điều kiện  $R_6$ .

### 3.4.3. Hàm mục tiêu của bài toán

Vì tổng số các buổi trực của các bác sĩ và y tá luôn bằng  $NPK \times NCT$ , do đó sử dụng bất đẳng thức Cauchy: “Tổng các phần tử là không đổi thì tích sẽ đạt giá trị lớn nhất khi tất cả các phần tử bằng nhau”, ta suy ra để đảm bảo yêu cầu của bài toán: số các buổi trực của các bác sĩ là xấp xỉ bằng nhau sẽ tương đương với điều kiện: Hãy xác định phương án  $Z = [ZBS; ZYT]$  thỏa mãn tất cả các ràng buộc để sao cho hai hàm mục tiêu:

$$FB(ZBS) = \prod_{s=1}^{NBS} CBS(s) \rightarrow \max$$

$$FY(ZYT) = \prod_{t=1}^{NYT} CYT(t) \rightarrow \max$$

Như vậy, bài toán lập lịch trực được đưa về bài toán: Hãy xác định phương án  $Z$  thỏa mãn các ràng buộc mô tả bởi các hàm  $R_1(Z)$ ,  $R_2(Z)$ ,  $R_{345}(Z)$ ,  $R_6(Z)$  để sao cho hàm mục tiêu:

$$F(Z) = FB(ZBS) + FY(ZYT) \rightarrow \max$$

### 3.4.4. Thuật toán Genetic\_QH

Input: các mảng  $PBS$ ,  $PYT$ ,  $MBS$ ,  $MYT$ .

Output: các mảng  $ZBS$ ,  $ZYT$ .

Bước 1: Khởi tạo quần thể ban đầu

1.1 Sử dụng phương pháp khởi tạo ngẫu nhiên  $N$  cặp ma trận  $Z = [ZBS; ZYT]$  kích thước  $NBS \times NYT$  các phần tử trong khoảng  $(0; NBS + 1)$ ,  $(0; NYT + 1)$ , trên các cột của ma trận thỏa mãn các điều kiện  $R_1$  và  $R_{345}$ .

1.2 Xác định độ thích nghi của các cá thể trong quần thể xuất phát.

1.3 Xác định giá trị hàm mục tiêu ban đầu

Bước 2: Quá trình lai ghép

+ Sử dụng phương pháp lai ghép hai điểm cắt theo chiều thời gian  $t$  giữa cá thể thứ  $i$  (bố) với cá thể thứ  $j$  ( $i < j$ ) (mẹ), chúng ta thu được quần thể gồm  $N + (N - 1) \times (N - 2)$  cá thể. Khi đó điều kiện  $R_1$  và  $R_2$  luôn thỏa mãn.

+ Kiểm tra độ thích nghi của tất cả các cá thể, ta thu được  $M$  cá thể thỏa mãn các điều kiện ràng buộc.

Bước 3: Quá trình chọn lọc

Tính giá trị hàm  $F(Z)$  với mọi cá thể trong quần thể gồm  $M$  cá thể, từ đó ta lựa chọn lấy  $N$  cá thể tốt nhất để sử dụng cho thế hệ kế tiếp

Bước 4: Đột biến

Sử dụng kỹ thuật đột biến  $m$  cá thể bất kỳ bằng cách thay đổi ngẫu nhiên một giá trị trong ma trận  $Z$  phù hợp với bài toán, xác suất đột biến  $q = \frac{m}{M} < 0.05$ .

Bước 5: Quay lại Bước 2 với  $N$  cá thể trong thế hệ tiếp sau.

Điều kiện dừng lặp của thuật toán là số lần lặp đạt tới số lần định hạn trước.

**Nhận xét**

+ Do thuật toán là ngẫu nhiên nên mỗi lần chạy khác nhau sẽ cho các kết quả khác nhau. Do đó ta cần chạy nhiều lần để lấy kết quả tốt nhất.

+ Độ phức tạp của thuật toán được đánh giá là  $O(NBS \times NPK \times NBT \times K)$  trong đó  $K$  là số lần lặp của giải thuật.

Thuật toán cũng dễ dàng cài đặt trên các ngôn ngữ lập trình cơ bản.

**4. CÁC KẾT QUẢ THỰC NGHIỆM**

Để kiểm tra độ chính xác của các thuật toán đã đề xuất, chúng tôi xét bài toán lập lịch với số bác sĩ là  $NBS = 15$ , số y tá  $NYT = 10$ , số phòng khám là  $NPK = 7$  và số ca trực là  $NCT = 14$ . Các điều kiện phù hợp chuyên môn của các bác sĩ và y tá đối với các phòng khám được cho trong Bảng 1 và Bảng 2 ( $1 - phù hợp, 0 - không phù hợp$ ). Các điều kiện sẵn sàng trực của các bác sĩ và y tá đối với các ca trực được cho trong Bảng 3 và Bảng 4 ( $1 - sẵn sàng, 0 - không sẵn sàng$ ). Các chương trình lập trình trong môi trường Matlab, thực hiện trên máy tính cấu hình: Intel(R) Core i7, CPU @ 2.50 GHz, RAM 8.0 GB, Win 10, 64 bit.

**Bảng 1. Bác sĩ phù hợp chuyên môn**

Phòng Bác sĩ	1	2	3	4	5	6	7
1	0	1	1	0	1	1	1
2	0	1	1	0	1	1	1
3	1	0	1	1	0	1	1
4	1	0	1	1	0	1	1
5	1	1	0	1	1	0	1
6	1	1	0	1	1	0	1
7	1	1	1	0	1	1	0
8	1	1	1	0	1	1	0
9	0	1	1	1	0	1	1
10	0	1	1	1	0	1	1
11	1	0	1	1	1	0	1
12	1	0	1	1	1	0	1
13	1	1	0	1	1	1	0
14	1	1	0	1	1	1	0
15	1	1	1	0	1	1	0

**Bảng 2. Y tá phù hợp chuyên môn**

Phòng Y tá	1	2	3	4	5	6	7
1	1	0	1	1	1	0	1
2	0	1	1	1	0	1	1
3	1	0	1	0	1	1	1
4	0	1	1	1	1	1	0
5	1	1	1	1	0	0	1
6	1	1	0	1	1	1	0
7	1	1	1	0	0	1	1
8	1	1	1	0	1	0	1
9	1	1	0	1	0	1	1
10	1	1	1	1	1	1	1

**Bảng 3. Bác sĩ sẵn sàng nhận buổi trực**

Ca BS	1	2	3	4	5	6	7	8	9	10	11	12	13	14
1	0	1	1	1	1	1	1	1	1	0	1	1	1	1
2	1	1	1	1	0	1	1	0	1	1	1	1	1	1
3	1	0	1	1	1	0	1	1	1	1	1	1	1	1
4	1	1	0	1	1	1	1	1	1	1	1	1	0	1
5	1	1	1	0	1	1	1	1	1	1	0	1	1	1
6	1	1	1	1	0	1	1	1	1	1	1	1	1	0
7	1	1	1	1	1	1	0	1	1	1	1	0	1	1
8	1	1	0	1	1	1	1	1	1	1	1	1	0	1
9	1	1	1	1	1	1	1	0	1	0	1	1	1	1
10	1	1	1	1	0	1	1	1	1	1	1	1	1	0
11	1	0	1	1	1	1	1	1	1	1	1	1	0	1
12	1	1	0	1	1	1	1	1	1	1	1	0	1	1
13	1	1	1	0	1	1	1	1	1	0	1	1	1	1
14	1	1	1	1	0	1	1	1	1	0	1	1	1	1
15	1	1	1	1	1	1	1	1	1	1	1	1	1	1



**Bảng 4. Y tá sẵn sàng nhận buổi trực**

Ca Y tá	1	2	3	4	5	6	7	8	9	10	11	12	13	14
1	0	1	1	1	1	1	1	1	1	1	1	1	1	1
2	1	1	1	1	0	1	1	1	1	1	1	1	1	1
3	1	1	1	1	1	1	0	1	1	1	1	1	1	1
4	1	1	1	1	1	1	1	1	1	1	1	1	0	1
5	1	1	1	1	1	1	1	1	1	1	1	1	1	0
6	1	1	1	1	1	0	1	1	1	1	1	1	1	1
7	1	1	1	1	1	1	1	1	1	1	1	0	1	1
8	1	1	1	1	1	1	1	1	1	1	0	1	1	1
9	1	1	1	1	1	0	1	1	1	0	1	1	1	1
10	1	0	1	1	1	1	1	1	1	1	1	1	1	1

**4.1. Kết quả chạy thuật toán Greedy\_QH**

Thời gian thực hiện: 0.02 giây

**Bảng 5. Lịch trực (Số hiệu bác sĩ – y tá)**

Phòng Ca	1	2	3	4	5	6	7
1	3-1	15-6	9-3	13-10	6-5	14-4	5-7
2	1-2	2-8	14-1	15-7	12-5	11-3	10-6
3	2-3	1-6	3-2	14-10	4-4	13-1	6-5
4	4-4	3-8	1-3	2-7	14-2	15-1	13-10
5	5-5	4-9	2-4	1-8	15-3	3-6	14-1
6	7-7	5-10	4-5	3-9	1-2	2-1	15-3
7	6-9	7-1	5-8	9-10	3-5	1-7	2-2
8	8-10	6-2	7-6	4-1	5-7	2-5	1-8
9	9-1	8-4	6-2	5-3	7-10	4-8	10-5
10	10-2	11-5	8-9	7-4	6-6	9-10	4-7
11	11-3	10-6	9-1	8-7	12-8	6-9	5-10
12	12-4	13-7	10-3	11-6	8-9	7-2	9-8
13	13-7	12-8	11-4	10-9	9-10	8-3	14-6
14	14-5	15-9	12-6	13-8	10-2	11-4	8-1

Số buổi trực bác sĩ: cao nhất: 7, thấp nhất: 5.

Số buổi trực y tá: cao nhất: 11, thấp nhất: 9.

**4.2. Kết quả chạy thuật toán Genetic\_QH**

Thời gian thực hiện: 0.12 giây

**Bảng 6. Lịch trực (Số hiệu bác sĩ – y tá)**

Phòng Ca	1	2	3	4	5	6	7
1	8-8	9-5	15-3	11-9	12-10	7-6	5-2
2	7-6	6-7	10-4	5-9	1-1	2-2	9-3
3	7-5	15-7	3-3	13-6	11-4	10-10	9-1
4	8-3	14-9	7-8	10-4	12-1	15-7	6-10
5	7-8	5-5	4-7	13-4	1-6	9-9	3-1
6	5-7	14-4	8-8	9-1	6-10	1-2	10-5
7	5-9	10-5	4-10	3-1	2-6	8-4	9-2
8	12-3	10-8	15-7	6-6	14-10	4-9	5-5
9	6-7	15-2	2-3	13-10	12-8	4-6	11-9
10	14-6	2-7	15-2	11-1	8-4	7-10	6-3
11	11-9	13-6	1-5	3-10	7-4	4-3	6-2
12	13-1	2-4	9-5	5-9	12-6	3-2	11-8
13	12-6	9-6	2-3	13-9	14-8	1-7	5-2
14	13-6	2-8	12-2	11-4	14-3	4-7	3-1

Số buổi trực bác sĩ: cao nhất: 7, thấp nhất: 6.

Số buổi trực y tá: cao nhất: 11, thấp nhất: 9.

### Nhận xét

+ Thông qua kết quả thực nghiệm qua một bộ số liệu cụ thể, chúng ta thấy rằng cả hai thuật toán đều đưa ra lịch biểu gần tối ưu.

+ So sánh với thuật toán Greedy\_QH thì thời gian thực hiện của thuật toán Genetic\_QH là chậm hơn do quá trình khởi tạo ngẫu nhiên các phương án cần thỏa mãn các điều kiện ràng buộc cùng với thời gian thực hiện lai ghép và chọn lọc. Tuy nhiên, thuật toán Genetic\_QH sẽ đảm bảo lời giải tối ưu tốt hơn, khi số lần lặp của thuật toán di truyền là đủ lớn vì thuật toán đã xét đến hàm mục tiêu trong quá trình thực hiện.

### 5. KẾT LUẬN

Nội dung chính của bài báo trình bày một số kết quả nghiên cứu về các thuật toán xấp xỉ. Trên cơ sở xây dựng và phân tích mô hình bài toán lập lịch trực tại phòng khám trong các bệnh viện, chúng tôi đề xuất hai thuật toán giải quyết bài toán. Các kết quả thực nghiệm đã chứng tỏ tính khả thi của các thuật toán đã đề xuất, các thuật toán này hoàn toàn có thể ứng dụng để lập lịch biểu cho tất cả các phòng khám tại các bệnh viện, với số liệu đầu vào xuất phát từ thực tế.

Các kết quả có thể mở rộng để nghiên cứu thiết kế các thuật toán xấp xỉ tìm nghiệm tối ưu cho các mô hình lập lịch biểu, với các hệ ràng buộc phức tạp hơn nữa trong thực tế.

### TÀI LIỆU THAM KHẢO

- Eiben, A. E., Raué, P. E., & Ruttkay, Zs. (1994). Genetic algorithms with multi-parent recombination. Trong Y. Davidor, H.-P. Schwefel, & R. Männer (B.t.v), *Parallel Problem Solving from Nature – PPSN III* (tr 78–87). Springer. [https://doi.org/10.1007/3-540-58484-6\\_252](https://doi.org/10.1007/3-540-58484-6_252)
- Goldberg, D. E. (1989). *Genetic Algorithms in Search, Optimization and Machine Learning* (13th ed. edition). Addison-Wesley Professional.
- Martello, S., & Toth, P. (1990). *Knapsack problems: Algorithms and computer implementations*. John Wiley & Sons, Inc.
- Mui, N. H., Hoa, V. D., & Tuyen, L. T. (2012). A parallel genetic algorithm for the job shop scheduling problem. *2012 IEEE International Symposium on Signal Processing and Information Technology (ISSPIT)*, 000019–000024. <https://doi.org/10.1109/ISSPIT.2012.6621254>
- Nguyễn Đình Thúc. (2001). *Lập trình tiến hóa*. Nxb Giáo dục.
- Nguyễn Hữu Mùi & Vũ Đình Hòa. (2012). Một thuật toán di truyền hiệu quả cho bài toán lập lịch job shop. *Vietnam Journal of Science and Technology*, 50(5), Art. 5. <https://doi.org/10.15625/0866-708X/50/5/9529>
- Trương Quốc Định & Nguyễn Thanh Hải. (2016). Giải thuật xếp thời khóa biểu ứng dụng vào bài toán quản lý xếp lịch thi kết thúc các lớp học phần tại Trường Đại học Cần Thơ. *Tạp chí Khoa học Trường Đại học Cần Thơ*, 43, Art. 43. <https://doi.org/10.22144/ctu.jvn.2016.254>
- Wirth, N. (1976). *Algorithms + Data Structures = Programs* (1st edition). Prentice Hall.