



CUỘC THI TOÁN MÔ HÌNH 2024 VÒNG 1

Tên đội thi (Bảng B): Dynamic Trio
Ngày thực hiện: 12/07/2024 - 22/07/2024



Contents

1	Mở đầu	2
2	Giải pháp	2
2.1	Ý tưởng giải pháp	2
2.2	Thuật toán di truyền	3
2.3	Xử lý dữ liệu	5
2.4	Hàm thích nghi (Fitness Function)	6
3	Kết quả	8
3.1	Đánh giá mô hình	8
3.2	So sánh một số phương pháp lựa chọn và lai ghép phổ biến cho thuật toán di truyền:	9
4	Tính tổng quát và khả thi của giải pháp	11
4.1	Các trường hợp mở rộng	11
4.2	Đánh giá tính tổng quát của giải pháp	12
4.2.1	Ưu điểm của giải pháp	12
4.2.2	Nhược điểm của giải pháp	13
4.3	Các hướng phát triển và tối ưu cho thuật toán	13
5	Tài liệu tham khảo	14



1 Mở đầu

Trong thực tế, mô hình lập lịch là mô hình tối ưu được nhiều nhà nghiên cứu quan tâm do độ phức tạp và tính ứng dụng cao trong thực tế. Trong môi trường y tế, việc lập lịch và phân công kíp trực hợp lý cho các bác sĩ và điều dưỡng là một thách thức lớn nhằm đảm bảo chất lượng dịch vụ và sức khỏe cho bệnh nhân. Bài toán này không chỉ đơn giản là phân bổ thời gian làm việc mà còn bao gồm các yếu tố như kinh nghiệm chuyên môn, sự hợp tác giữa các thành viên trong nhóm y tế và sự quản lý hiệu quả tài nguyên nhân lực.

Bài toán thực tế mà BTC đưa ra là sắp xếp ca trực cho các bác sĩ và điều dưỡng ở bệnh viện Môn Hòa tháng 8, 9 năm 2024. Dữ liệu bao gồm các thành viên đến từ 3 nhóm (bác sĩ cộc I, bác sĩ cộc II, điều dưỡng) và 4 khoa khác nhau (Nội, Ngoại, Cấp cứu, Lâm sàng) (ở đây nhóm giả thiết các nhân sự có tên trùng nhau nhưng khác thâm niên và khoa là những người khác nhau)

2 Giải pháp

2.1 Ý tưởng giải pháp

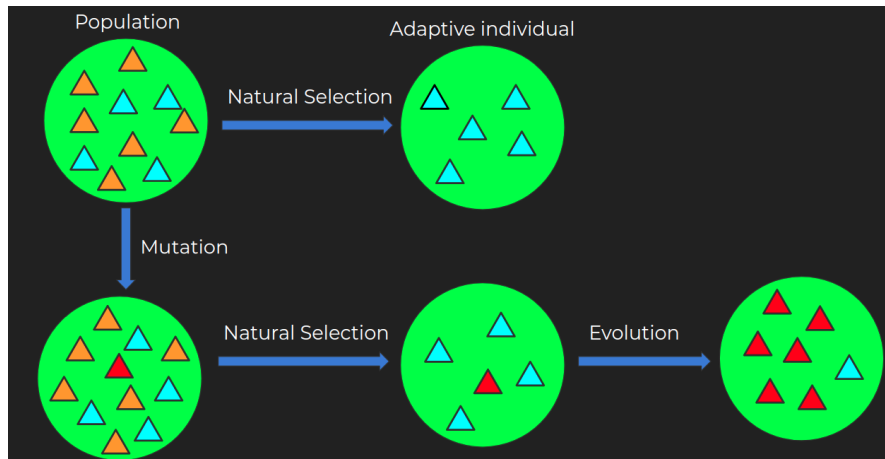
Thuyết tiến hóa của Charles Darwin đã mang đến một cuộc cách mạng trong hiểu biết của chúng ta về sự phát triển của sự sống trên Trái đất. Học thuyết này giải thích cách các sinh vật thích nghi và tiến hóa qua thời gian thông qua quá trình chọn lọc tự nhiên, di truyền, trao đổi chéo và đột biến. Những nguyên lý này không chỉ giới hạn trong sinh học mà còn truyền cảm hứng cho các nhà khoa học máy tính.

Genetic Algorithms (GA) - Giải thuật di truyền là một kỹ thuật khoa học máy tính dựa trên học thuyết tiến hóa nhằm giải quyết các bài toán tối ưu tổ hợp với những bài toán có không gian tìm kiếm rất lớn. Để có thể hiểu rõ hơn về những nguyên lý cơ bản của thuật toán và cách nó mô phỏng quá trình tiến hóa tự nhiên:

- **Quần thể (Population):** Tập hợp các cá thể với đặc tính đa dạng, quyết định khả năng sinh tồn và thích nghi.
- **Chọn lọc tự nhiên (Natural Selection):** Quá trình loại bỏ cá thể yếu, giữ lại những cá thể thích nghi tốt (Adaptive individual).
- **Đột biến (Mutation):** Tạo ra những thay đổi ngẫu nhiên trong gen, cung cấp nguyên liệu mới cho quá trình chọn lọc.
- **Trao đổi chéo (Cross-over):** Quá trình kết hợp thông tin di truyền từ hai cá thể cha mẹ để tạo ra cá thể con, tăng đa dạng gen trong quần thể.
- **Tiến hóa (Evolution):** Kết quả của quá trình chọn lọc tự nhiên tác động lên các cá thể đột biến và lai ghép. Những gen có lợi dần chiếm ưu thế trong quần thể qua thời gian.



Hình ảnh dưới đây sẽ mô tả cụ thể những thành phần trong quá trình tiến hóa được sử dụng trong thuật toán và mối liên hệ của chúng:

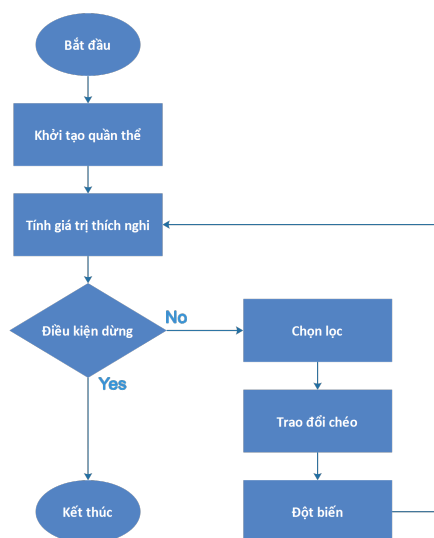


Hình 1: Các thành phần trong thuật toán và mối liên hệ

Giải thuật di truyền đã được sử dụng thành công cho một số bài toán phức tạp trong các ngành nghiên cứu như: xử lý ảnh, tối ưu hệ thống điều khiển, lập lịch trình... Trong bài cáo này, nhóm đề xuất giải pháp áp dụng thuật toán di truyền nhằm giải quyết bài toán lập lịch trực cho các nhân viên của bệnh viện Môn Hòa.

2.2 Thuật toán di truyền

Dưới đây là sơ đồ thuật toán của giải thuật di truyền:



Hình 2: Sơ đồ thuật toán



- **Khởi tạo quần thể (Initialization):**

- Tạo ra một quần thể ban đầu gồm các cá thể ngẫu nhiên.
- Mỗi cá thể đại diện cho một giải pháp tiềm năng của bài toán.

- **Đánh giá độ thích nghi (Fitness Evaluation):**

- Tính toán giá trị độ thích nghi cho mỗi cá thể trong quần thể.
- Độ thích nghi phản ánh chất lượng của giải pháp.

- **Chọn lọc (Selection):**

- Chọn các cá thể có độ thích nghi cao để tạo ra thế hệ tiếp theo, độ thích nghi này được xét bởi hàm đánh giá mô hình (Fitness Evaluation)
- Các phương pháp phổ biến bao gồm chọn lọc theo thứ hạng (Rank Selection), chọn lọc tournament (Tournament Selection), chọn lọc ưu tú (Elitism Selection) ...

- **Trao đổi chéo (Crossover):**

- Kết hợp thông tin di truyền từ hai cá thể cha mẹ để tạo ra cá thể con.
- Có nhiều phương pháp lai ghép như lai ghép một điểm, hai điểm, đồng nhất.

- **Đột biến (Mutation):**

- Thực hiện các thay đổi ngẫu nhiên nhỏ trên một số cá thể.
- Giúp duy trì đa dạng gen và khám phá không gian tìm kiếm mới.

- **Tạo thế hệ mới (New Generation):**

- Kết hợp các cá thể con mới được tạo ra với một số cá thể từ thế hệ trước.
- Đảm bảo kích thước quần thể không đổi qua các thế hệ.

- **Kiểm tra điều kiện dừng (Termination Check):**

- Kiểm tra xem thuật toán đã đạt được tiêu chí dừng chưa.
- Tiêu chí có thể là số thế hệ tối đa, thời gian chạy, hoặc độ hội tụ của quần thể.

- **Lặp lại:**

- Nếu chưa đạt điều kiện dừng, quay lại bước 2.
- Tiếp tục quá trình tiến hóa cho đến khi thỏa mãn điều kiện dừng.

- **Kết quả:**

- Trả về cá thể có độ thích nghi tốt nhất làm giải pháp cuối cùng.

Thuật toán Di truyền lặp lại các bước này qua nhiều thế hệ, dần dần cải thiện chất lượng của các giải pháp trong quần thể, hướng giải pháp tối ưu cho bài toán đang xét.



2.3 Xử lý dữ liệu

Bệnh viện Môn Hoà đang có nhu cầu sắp xếp lịch trực chuyên môn cho 2 tháng 8 và 9 năm 2024. Bệnh viện bao gồm 4 khoa: Nội, Ngoại, Cấp cứu và Lâm sàng với số lượng nhân sự như sau:

- 19 bác sĩ cộc I
- 17 bác sĩ cộc II
- 44 điều dưỡng

Ở đây nhóm tác giả quy định theo hàng ngang, hệ số 0,1,2 lần lượt là Bác sĩ cộc I, Bác sĩ cộc II và Điều dưỡng; theo hàng dọc, các hệ số 0,1,2,3 tương ứng với các khoa Nội, Ngoại, Cấp cứu và Lâm sàng.

Danh sách các thành viên như sau:

Khoa	Bác sĩ cộc I (0)	Bác sĩ cộc II (1)	Điều dưỡng (2)
Nội (0)	Hường(3), Thư(7) Sơn(8)	Hưng(4), Thắng(9)	Dũng(6), Tơ(7), Tiến(7), Dương(10), Huệ(8), Tuyền(16), Huyền(20), An(2), Linh(3)
Ngoại (1)	Thương(2), Tuấn(4)	Thanh Hà(6) Ngân Hà(10) Linh(13)	Thuận(8), Giang(10), Gấm(15), Phượng(1), Công(5), Anh(4) , Hoài(9), Huệ(12), Thông(12) , Thu(15), Yên(16), Văn Đức(19), Nam(18), Tú(22), Chí Đức(1)
Cấp cứu (2)	Quân(1), Long(5) Bắc(9), Thủy(11)(18) Tâm(17), Xuân(18)	Hùng(2), Dũng(6) Vân Anh(11) Út(13)	Minh(4), Dung(5), Lương(8), Oanh(11), Len(9), Quỳnh(13), Ngọc(14), Khánh Linh(13), Sang(17), Long(18), Phi(17), Thảo(21), Chung(1)
Lâm sàng (3)	Diệp(12), Giang(14) Bá Tuấn(16), Bằng(19)	Hà(3), Hương(5) Dung(7), Linh(12) Tâm(14) Minh Phương(16) Xuân(17)	Hương(2), Chi(3), Kim Anh(6), Ngân(10), Đông(11), Thu(14), Hưng(15), Vương(19), Thu Hằng(20), Huyền(21), Thành(22)

Ta lưu thông tin về khoa và các thành viên vào 2 danh sách trong file input (data.json):

- "name": Danh sách tên của các thành viên
- "employee_info": Danh sách thông tin của các thành viên, mỗi phần tử là một mảng chứa ba giá trị lần lượt là khoa, loại bác sĩ và thâm niên

Vd: Hướng, [0,0,3]

"POPULATION_SIZE": 300 tức số lượng cá thể trong một quần thể tại mỗi thế hệ là 300,

"P_CROSSOVER": 0.7, là tham số xác định tỷ lệ các cặp cha mẹ sẽ thực hiện quá trình lai ghép để tạo ra các cá thể con mới. P_CROSSOVER giúp điều chỉnh không gian tìm kiếm và tính đa dạng cho các cá thể mới.

"P_MUTATION": 0.15, cho biết tỷ lệ phần trăm các gene trong một cá thể có khả năng bị thay đổi trong quá trình đột biến. Nếu P_MUTATION là 0.15 nghĩa là 15% các gen trong mỗi cá thể có khả năng bị thay đổi ngẫu nhiên trong mỗi thế hệ

"MAX_GENERATIONS": 1000, quy định số thế hệ tối đa mà thuật toán sẽ thực hiện trước khi



dừng lại

Các điều kiện ràng buộc:

- Một kíp trực bao gồm 4 thành viên: 1 bác sĩ cộc I, 1 bác sĩ cộc II, 2 điều dưỡng (Hard Constraint)
- Một kíp trực nên có sự đa dạng thành viên đến từ cả 4 khoa cũng như có thành viên nhiều thâm niên (Soft Constraint)
- Lịch trực cần có tính hợp lý trong việc xoay vòng nhân sự, không ai phải trực quá nhiều cũng như quá ít (Soft Constraint)

Từ dữ liệu đề cho ta giải quyết yêu cầu bằng thuật toán di truyền như sau:

- Khởi tạo lịch trình ban đầu, trong quá trình chọn lọc ngẫu nhiên theo thứ tự người thứ nhất sẽ được chọn ngẫu nhiên trong số các bác sĩ loại I, người thứ 2 được chọn ngẫu nhiên từ trong nhóm các bác sĩ loại II, 2 người cuối cùng được chọn từ nhóm điều dưỡng. Do đó trong quá trình khởi tạo và đột biến, các cá thể (lịch trình được tạo ra) đã thỏa mãn Hard Constraint.
 - Dựa vào kết quả được đánh giá bằng hàm thích nghi (fitness function), cá thể (lịch trình) nào có kết quả càng thấp (vi phạm các ràng buộc càng ít) thì cá thể đó càng tốt. Các cá thể tốt hơn được chọn lọc bằng phương pháp chọn lọc giải đấu (tournament selection)
 - Các cá thể (lịch trình) được chọn sẽ trải qua quá trình lai ghép để tạo ra cá thể con theo phương pháp lai ghép đồng nhất (uniform crossover). Quá trình lai ghép này được kết hợp từ hai cá thể cha và mẹ (parent) để tạo ra cá thể con mới với xác suất $P_CROSSOVER = 0.7$
 - Một số cá thể con xảy ra quá trình đột biến để đảm bảo sự đa dạng của các cá thể dựa trên xác suất đột biến $P_MUTATION = 0.15$. Quá trình đột biến diễn ra trong từng nhóm bác sĩ nên vẫn cá thể mới vẫn đảm bảo Hard Constraint
- Quá trình này lặp đi lặp lại cho đến khi đạt được $MAX_GENERATIONS$, mục tiêu để tìm ra giải pháp tốt nhất cho bài toán

2.4 Hàm thích nghi (Fitness Function)

Trong quá trình chọn lọc ngẫu nhiên và đột biến ta đã thực hiện chọn theo từng nhóm bác sĩ (bác sĩ loại I, bác sĩ loại II, điều dưỡng) nên các cá thể sinh ra đã thỏa mãn ràng buộc cứng (Hard Constraint).

Đối với các ràng buộc mềm (Soft Constraints), ta đặt các điều kiện sau để đánh giá hàm thích nghi (fitness function):

- **Kiểm tra tính đa dạng thành viên đến từ 4 khoa:** Ta kiểm tra hệ số đầu tiên trong thông tin của các thành viên được phân bố lịch trong ngày, nếu trong một ngày các thành viên đến từ 1,2 hoặc 3 khoa sẽ được cộng một lượng phạt (penalty) tương ứng.
- **Kiểm tra thâm niên:** Bởi vì mỗi nhóm thành viên có chuyên môn khác nhau nên ta chỉ so sánh tuổi thâm niên của các thành viên trong cùng một nhóm (bác sĩ cộc I, bác sĩ cộc II, điều dưỡng). Dựa theo dữ liệu đề cho về thâm niên các thành viên, ta so sánh độ tuổi của các thành viên với phân vị 75% của nhóm tương ứng. Vì đề yêu cầu trong mỗi ngày có một thành viên có thâm niên nên ta chỉ cộng penalty bằng hiệu giữa thâm niên trung bình của cả bệnh viện và người có thâm niên lớn nhất của ngày trực đó vào các ngày không có thành viên nào thỏa mãn điều kiện.



- **Đánh giá tính hợp lý của lịch trực, độ phù hợp xoay vòng nhân sự:** Bởi vì số lượng nhân sự mỗi nhóm bác sĩ khác nhau và phụ thuộc vào dữ liệu bài toán nên ta đưa ra các trọng số phù hợp với thông tin đề cho. Ta đặt ra các điều kiện ràng buộc về 2 ngày trực liên tiếp nằm trong khoảng $[a, b]$, trong đó a và b quy định về số ngày nghỉ ít nhất và nhiều nhất giữa 2 ngày trực liên tiếp của một người.

Cụ thể như sau:

- Số bác sĩ cộc I là 19 bác sĩ, trực trong vòng 61 ngày (2 tháng 8,9 năm 2024) thì số buổi một người phải trực trong khoảng từ 3-4 ngày và 2 ngày trực sẽ cách nhau từ 16 - 19 ngày.

- Số bác sĩ cộc II là 17 bác sĩ, trong vòng 61 ngày thì mỗi người phải trực trong khoảng từ 3-4 ngày và 2 ngày trực sẽ cách nhau từ 13 - 17 ngày.

- Đối với điều dưỡng, trong vòng 61 ngày và mỗi ngày trực yêu cầu 2 người nên với số lượng 44 điều dưỡng, mỗi người phải trực từ 2-3 ngày và mỗi ngày cách nhau từ 15-20 ngày.

Ngoài ra, để đảm bảo tính chặt chẽ của ràng buộc này, ta đặt thêm các điều kiện để đảm bảo không có trường hợp nào mà một thành viên không trực trong suốt số ngày đề cho và số ngày trực vượt quá 4 ngày đối với bác sĩ loại I, II và 3 ngày đối với điều dưỡng.

Đối với ràng buộc về tính hợp lý của lịch trực và xoay vòng nhân sự, ta xét với từng thành viên được phân lịch trực trong ngày, nếu vi phạm điều kiện thì lượng penalty cộng vào sẽ bằng hiệu của $\frac{a+b}{2}$ và chênh lệch giữa 2 ngày trực liên tiếp của người đó.

Như vậy, ta có hàm thích nghi (fitness function) như sau, lượng phạt (penalty) chỉ tính cho các ràng buộc mềm vì trong quá trình khởi tạo, chọn lọc và đột biến đã thỏa mãn ràng buộc cứng:

$$F = S_1 \times \sum_{d=0}^{D-1} P_1 + S_2 \times \sum_{d=0}^{D-1} P_2 + S_3 \times \sum_{d=0}^{D-1} \sum_{i=0}^{I-1} P_3$$

Trong đó:

- D là số ngày trực đề cho, ở đây ta phân lịch trực theo ngày, mỗi ngày 1 nhóm gồm 4 người
- I là số người đề bài quy định trong 1 ca trực
- P_1, P_2, P_3 lần lượt là lượng phạt (penalty) cho các ràng buộc mềm về sự đa dạng thành viên của 4 khoa, thâm niên và tính hợp lý của lịch trực.
- S_1, S_2, S_3 lần lượt là hệ số của từng ràng buộc mềm (Soft Constraints)

Trong quá trình chạy thử và kiểm tra kết quả, nếu chỉ trả về kết quả của từng ràng buộc mềm (Soft Constraints) nhóm nhận thấy lượng phạt của P_1 (tính đa dạng 4 khoa) và P_2 (thâm niên) gần về 0, P_3 (tính hợp lý của lịch trình) có giá trị lớn cho thấy điều kiện này khó đáp ứng triệt để.

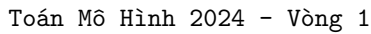
Trong thực tế, xét trong ca trực, ta nhận thấy sự hợp lý về lịch trực và xoay vòng nhân sự cùng với yêu cầu có thành viên nhiều thâm niên trong ca trực thường được quan tâm nhiều hơn. Phân lịch ca trực khác với các yêu cầu về công việc chuyên môn cụ thể nên ta giữ lượng phạt về tính đa dạng bé.

2 điều kiện về tính đa dạng nhân sự và thâm niên được tính lượng phạt (penalty) theo ngày trong khi sự hợp lý của lịch trực được xét từng người theo ngày nên lượng phạt cộng vào sẽ lớn hơn.

Sau khi thử nghiệm kết quả, dựa vào thông tin thành viên và phân bổ số lượng, thâm niên ở các khoa, các nhóm của dữ liệu đề cho, ta chọn được các hệ số sau:

$$S_1 = 0.7, S_2 = 0.1, S_3 = 0.2$$

Bởi vì lượng phạt (penalty) được của ràng buộc về tính hợp lý của lịch trình lớn trong khi lượng phạt (penalty) của ràng buộc về tính đa dạng và thâm niên bé nên sau khi nhân với các



3 Kết quả

Link kết quả: [output_main.csv](#)

3.1 Đánh giá mô hình

Dưới đây là bảng kết quả phân công lịch trực cho 2 tháng 8,9, bao gồm 61 ngày. Kết quả này sử dụng kết hợp phương pháp lai ghép Uniform Crossover, phương pháp chọn Tournament Selection và đột biến ngẫu nhiên.

Ngày	Bác sỹ cọc 1	Bác sỹ cọc 2	Điều dưỡng 1	Điều dưỡng 2
1/8/2024 Thanh Hà [1, 0, 6]	Hùng [0, 1, 4]	Quỳnh [2, 2, 13]	Vương [3, 2, 19]	
2/8/2024 Hùng [0, 0, 3]	Hương [3, 1, 5]	Nam [1, 2, 18]	Ngọc [2, 2, 14]	
3/8/2024 Linh [1, 0, 13]	Dũng [2, 1, 6]	An [0, 2, 2]	Linh [0, 2, 3]	
4/8/2024 Bá Tuấn [3, 0, 16]	Thống [0, 1, 9]	Lương [2, 2, 8]	Thần [3, 2, 22]	
5/8/2024 Thương [1, 0, 2]	Minh Phương [3, 1, 14]	Sang [2, 2, 17]	Danh [2, 2, 9]	
6/8/2024 Tâm [2, 0, 17]	Linh [3, 1, 12]	Tuyền [0, 2, 16]	Hoài [1, 2, 9]	
7/8/2024 Thu [0, 0, 7]	Hà [3, 1, 3]	Long [2, 2, 18]	Oanh [2, 2, 11]	
8/8/2024 Thủy [2, 0, 11]	Giang [1, 1, 10]	Công [2, 2, 10]	Huyền [3, 2, 21]	
9/8/2024 Tuấn [1, 0, 4]	Tâm [3, 1, 14]	Dương [1, 2, 5]	Chung [2, 2, 2]	
10/8/2024 Quân [2, 0, 1]	Vân Anh [2, 1, 11]	Thu [2, 2, 14]	Tú [1, 2, 22]	
11/8/2024 Diệp [3, 0, 12]	Gấm [1, 1, 15]	Phú [2, 2, 17]	Văn Đức [1, 2, 19]	
12/8/2024 Sơn [0, 0, 8]	Dung [3, 1, 7]	Khanh Linh [2, 2, 13]	Chi Đức [1, 2, 1]	
13-08-2024 Bức [2, 0, 9]	Xuân [3, 1, 17]	Anh [3, 2, 4]	Huê [0, 2, 8]	
14-08-2024 Xuân [2, 0, 18]	Hương [3, 1, 5]	Nam [1, 2, 18]	Huyền [0, 2, 20]	
15-08-2024 Trang [1, 0, 15]	Hùng [0, 1, 4]	Hưng [3, 2, 2]	Minh [2, 2, 1]	
16-08-2024 Hằng [3, 0, 19]	Út [2, 1, 13]	Long [2, 2, 18]	Yên [2, 2, 14]	
17-08-2024 Long [2, 0, 5]	Thống [0, 1, 9]	Lĩnh [2, 2, 16]	Thống [1, 2, 12]	
18-08-2024 Thu [0, 0, 7]	Hùng [2, 1, 2]	Thào [2, 2, 21]	Ngân [3, 2, 10]	
19-08-2024 Thanh Hà [1, 0, 6]	Tâm [3, 1, 14]	Quỳnh [2, 2, 13]	Tiến [0, 2, 7]	
20-08-2024 Linh [1, 0, 13]	Minh Phương [3, 1, 14]	Sang [2, 2, 17]	Linh [0, 2, 3]	
21-08-2024 Thương [1, 0, 2]	Dũng [2, 1, 6]	Vương [3, 2, 19]	Dũng [0, 2, 6]	
22-08-2024 Bá Tuấn [3, 0, 16]	Hà [3, 1, 3]	An [0, 2, 2]	Lương [2, 2, 8]	
23-08-2024 Hương [0, 0, 6]	Phương [1, 1, 1]	Huyền [3, 2, 21]	Dung [2, 2, 5]	
24-08-2024 Tâm [2, 0, 17]	Vân Anh [2, 1, 11]	Thần [3, 2, 22]	Hoài [1, 2, 9]	
25-08-2024 Thủy [2, 0, 11]	Giang [1, 1, 10]	Chung [2, 2, 1]	Tơ [0, 2, 1]	
26-08-2024 Tuấn [1, 0, 4]	Gấm [1, 1, 15]	Oanh [2, 2, 11]	Thu [3, 2, 14]	
27-08-2024 Quân [2, 0, 1]	Xuân [3, 1, 17]	Tuyền [0, 2, 16]	Văn Đức [1, 2, 19]	
28-08-2024 Sơn [0, 0, 8]	Hương [3, 1, 5]	Dung [3, 2, 0, 10]	Công [1, 2, 5]	
29-08-2024 Bức [2, 0, 9]	Hùng [0, 1, 4]	Chi [3, 2, 3]	Tú [1, 2, 22]	
30-08-2024 Xuân [2, 0, 18]	Vân Anh [2, 1, 11]	Tru Hằng [3, 2, 20]	Chi Đức [1, 2, 1]	
31-08-2024 Giang [3, 0, 14]	Út [2, 1, 13]	Nam [1, 2, 18]	Khanh Linh [2, 2, 13]	
			1/9/2024 Diệp [3, 0, 12]	Thống [0, 1, 9]
			2/9/2024 Long [2, 0, 5]	Danh [2, 2, 9]
			3/9/2024 Trang [1, 0, 15]	Huyền [2, 2, 1]
			4/9/2024 Bá Tuấn [3, 0, 16]	Dũng [2, 1, 6]
			5/9/2024 Thương [1, 0, 2]	Phương [1, 1, 1]
			6/9/2024 Thu [0, 0, 7]	Thần [1, 1, 8]
			7/9/2024 Tâm [2, 0, 17]	Tâm [3, 1, 14]
			8/9/2024 Thương [1, 0, 2]	Gấm [1, 1, 15]
			9/9/2024 Ngân Hà [1, 0, 10]	Giang [1, 1, 10]
			10/9/2024 Thủy [2, 0, 11]	Xuân [3, 1, 17]
			11/9/2024 Xuân [2, 0, 18]	Hà [3, 1, 3]
			12/9/2024 Giáp [3, 0, 14]	Hùng [0, 1, 4]
			13-09-2024 Quân [2, 0, 1]	Hương [3, 1, 5]
			14-09-2024 Tuấn [1, 0, 4]	Út [2, 1, 13]
			15-09-2024 Sơn [0, 0, 8]	Công [1, 2, 5]
			16-09-2024 Bức [2, 0, 9]	Phương [1, 1, 1]
			17-09-2024 Xuân [2, 0, 18]	Thống [0, 1, 9]
			18-09-2024 Diệp [3, 0, 12]	Hùng [2, 1, 2]
			19-09-2024 Trang [1, 0, 15]	Tâm Anh [2, 1, 11]
			20-09-2024 Long [2, 0, 5]	Vân [3, 1, 14]
			21-09-2024 Hằng [3, 0, 19]	Dũng [2, 1, 6]
			22-09-2024 Thu [0, 0, 7]	Thần [1, 1, 8]
			23-09-2024 Tâm [2, 0, 17]	Xuân [3, 1, 17]
			24-09-2024 Thương [1, 0, 2]	Gấm [1, 1, 15]
			25-09-2024 Ngân Hà [1, 0, 10]	Hà [3, 1, 3]
			26-09-2024 Hưng [0, 0, 3]	Giang [1, 1, 10]
			27-09-2024 Thủy [2, 0, 11]	Dung [3, 1, 7]
			28-09-2024 Giang [3, 0, 14]	Út [2, 1, 13]
			29-09-2024 Quân [2, 0, 1]	Phương [1, 1, 1]
			30-09-2024 Sơn [0, 0, 8]	Hùng [2, 1, 2]

(a) Tháng 8

(b) Tháng 9

Hình 3: Kết quả của nghiên cứu

- Faculty Penalty (đa dạng 4 khoa) = 35
- Seniority Penalty (thâm niên) = 22.0625
- Schedule Penalty (tính hợp lý của lịch trình) = 172.5

Do tính chất của bài toán, gần như không thể đưa ra một giải pháp tối ưu nhất nơi mà tất cả ràng buộc mềm đều được thoả mãn (fitness value bằng 0). Nhóm đã điều chỉnh các hệ số để phân bổ lượng phạt cho các ràng buộc mềm hợp lí và cân đối nhất có thể, dựa theo cách các lượng phạt này được tính toán và độ ưu tiên của các ràng buộc mềm này trong thực tế.



3.2 So sánh một số phương pháp lựa chọn và lai ghép phổ biến cho thuật toán di truyền:

Trong nghiên cứu này, nhóm đã thực hiện nghiên cứu nhiều phương pháp lựa chọn cá thể và lai ghép khác nhau trước khi rút ra được phương pháp tối ưu nhất. Một số phương pháp lựa chọn cá thể được đưa vào nghiên cứu bao gồm:

- **Roulette Wheel Selection:** Mỗi cá thể được gán một phần của vòng quay tỉ lệ với độ thích nghi của nó. Các cá thể có độ thích nghi cao hơn sẽ chiếm diện tích lớn hơn trên vòng quay, do đó có cơ hội được chọn cao hơn
- **Tournament Selection:** Một nhóm gồm k cá thể ngẫu nhiên được chọn và so sánh với nhau. Cá thể tốt nhất trong nhóm này sẽ được chọn để lai ghép.
- **Rank Selection:** Các cá thể trong quần thể được xếp hạng dựa trên độ thích nghi. Xác suất chọn mỗi cá thể phụ thuộc vào thứ hạng của nó thay vì giá trị độ thích nghi thực tế.
- **Random selection:** Lựa chọn ngẫu nhiên 2 cá thể để lai ghép.

Một số phương pháp lai ghép được đưa vào nghiên cứu bao gồm:

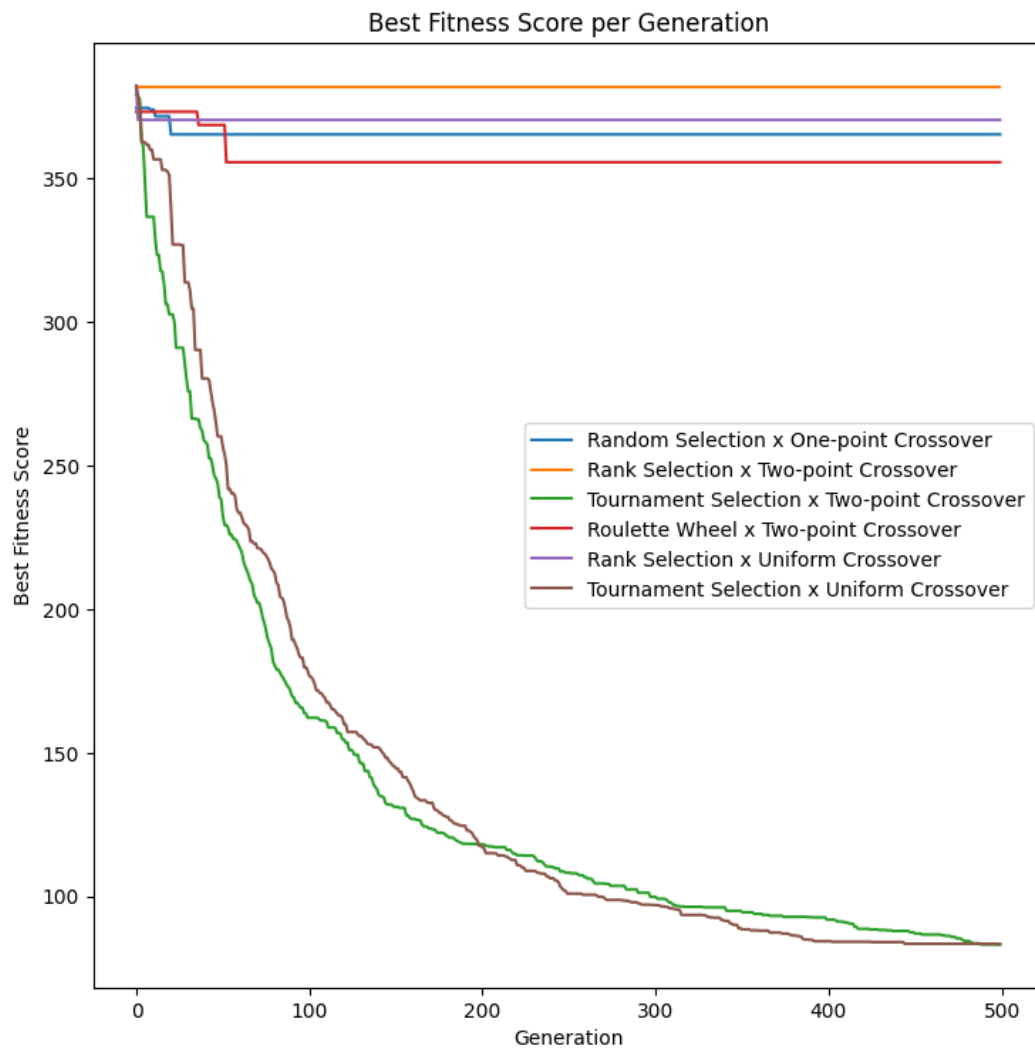
- **One-point Crossover:** Chọn một điểm ngẫu nhiên trên chuỗi gene và hoán đổi đoạn gene sau hai điểm này để tạo ra hai cá thể con.
- **Two-point Crossover:** Chọn hai điểm ngẫu nhiên trên chuỗi gene và hoán đổi đoạn gene trên hai điểm này giữa hai cá thể bố mẹ, tạo ra hai cá thể con.
- **Uniform Crossover:** Quyết định nguồn gốc của mỗi gene trên cá thể con được lấy từ bố hay mẹ theo một tỉ lệ nhất định (đối với nghiên cứu này, tỉ lệ này là 0.5).

Phương pháp Mutation (Đột biến) duy nhất được sử dụng là phương pháp chọn lựa ngẫu nhiên trong cùng loại bác sĩ để đảm bảo duy trì ràng buộc cứng (Hard Constraint). Sau đây là bảng so sánh độ hiệu quả của việc kết hợp các phương pháp khác nhau:

Phương pháp chọn	Lai ghép	Đột biến	Giá trị tốt nhất	Giá trị trung bình
Random Selection	One-point Crossover	Random	342.2	408.1
Roulette wheel	One-point Crossover	Random	371.8	570.2
Rank Selection	One-point Crossover	Random	389.8	850.9
Tournament Selection	One-point Crossover	Random	95.4	96.3
Random Selection	Two-point Crossover	Random	359	416
Roulette wheel	Two-point Crossover	Random	367.95	564.46
Rank Selection	Two-point Crossover	Random	374.6	872.3
Tournament Selection	Two-point Crossover	Random	81.5	82.2
Random Selection	Uniform Crossover	Random	355.93	435.72
Roulette wheel	Uniform Crossover	Random	363.9	544.9
Rank Selection	Uniform Crossover	Random	370.4	885.9
Tournament Selection	Uniform Crossover	Random	61.2	64.5

Bảng 1: A simple table

Đồ thị thể hiện sự hội tụ qua từng thế hệ của một vài phương án đáng chú ý:



Hình 4: Đồ thị so sánh sự hội tụ qua từng thế hệ của một vài phương pháp phổ biến

Từ bảng trên và đồ thị ta có thể rút ra các kết luận sau:

- Các phương pháp chọn khác ngoài Tournament Selection đều cho kết quả không khả quan và thất bại trong việc giảm fitness value của cả quần thể. Nguyên nhân của việc này là vì các phương pháp chọn đó không duy trì được độ đa dạng của quần thể, dẫn đến hội tụ sớm và không thể đạt được độ hiệu quả yêu cầu.
- Các phương pháp lai ghép thì lại cho kết quả khá tương đồng nhau và đều có độ hiệu quả nhất định. Trong các điều kiện của bài toán này, phương pháp Uniform Crossover cho ra kết quả vượt trội nhất.

Từ bảng trên ta có thể kết luận, phương pháp Tournament Selection kết hợp với Uniform Crossover đã được sử dụng ở trên cho ra giá trị fitness tốt nhất trong tất cả.



4 Tính tổng quát và khả thi của giải pháp

4.1 Các trường hợp mở rộng

Để đánh giá tính tổng quát và độ hiệu quả, khả thi của giải pháp, ta lần lượt thay đổi các trường hợp với bộ dữ liệu khác nhau để xem xét kết quả.

Trường hợp 1: Giả sử số lượng nhân sự giảm đi nhiều so với bộ dữ liệu đề cho và giữ nguyên số ngày trực (61 ngày) cùng với yêu cầu mỗi ngày 1 bác sĩ cộc I, cộc II và 2 điều dưỡng. Trong thực tế ở các bệnh viện nhỏ, số bác sĩ cộc I, cộc II và điều dưỡng ít dẫn tới yêu cầu cao về việc phân bổ lịch trực hợp lý và hiệu quả.

Bộ data nhóm thay đổi được lưu trong file (data_1.json), với số lượng thành viên mỗi nhóm như sau:

- 10 bác sĩ cộc I
- 9 bác sĩ cộc II
- 21 điều dưỡng

Bởi vì các điều kiện đối với các ràng buộc ta dựa trên dữ liệu đề cho, do đó nhóm phân tích lại về số lượng thành viên mỗi khoa và các nhóm để thay đổi khoảng $[a, b]$ của ràng buộc về tính hợp lý của lịch trình phù hợp và giữ nguyên hàm thích nghi ở phần 2.4.

- Bác sĩ cộc I có 10 thành viên nên mỗi người phải trực trong 6 ngày và có một người trực 7 ngày, số ngày nghỉ nằm trong khoảng 7-10 ngày.
- Số bác sĩ cộc II là 9 nên mỗi người phải trực trong khoảng từ 6-7 ngày và số ngày nghỉ của mỗi người nằm trong khoảng 6-9 ngày.
- Số điều dưỡng trong bệnh viện là 21 người, trực trong vòng 61 ngày, mỗi ngày yêu cầu 2 người thì mỗi người cần phải trực từ 5-6 ngày và số ngày nghỉ của mỗi người nằm trong khoảng 8-10 ngày.

Sau khi thay đổi tham số phù hợp và chạy thử mô hình, ta thu được kết quả lưu trong file (output_1.csv)

Link kết quả: [output_1.csv](#)

Kết quả này có giá trị theo hàm fitness là 39.54 lượng phạt (penalty) của các ràng buộc (soft constraints) là:

- Faculty Penalty (đa dạng 4 khoa) = 31
- Seniority Penalty (thâm niên) = 22.4
- Schedule Penalty (tính hợp lý của lịch trình) = 78

Sau khi giảm và có sự thay đổi phù hợp dựa vào số lượng nhân sự ở điều kiện sắp xếp lịch trình hợp lý, ta thu được kết quả tương đối tốt và không chênh lệch nhiều so với kết quả với bộ dữ liệu gốc.

Trường hợp 2: Giả định tăng số ngày cần lập thời khoá biểu lên 122 ngày thay vì 61 ngày. Thay đổi này được thực hiện thông qua tăng hệ số DAY_NUMS trong file data.json từ 61 lên 122.

Sau khi chạy thử mô hình, ta thu được kết quả lưu trong file (output_2.csv).

Kết quả này có giá trị theo hàm fitness là 239.9, lượng phạt (penalty) của các ràng buộc (soft constraints) là:

- Faculty Penalty (đa dạng 4 khoa) = 100
- Seniority Penalty (thâm niên) = 47
- Schedule Penalty (tính hợp lý của lịch trình) = 826

Vì tính chất của giải thuật này không thể cho ra được lời giải tốt nhất mà chỉ có thể cho ra được lời giải gần tốt nhất, nên khi số ngày cần xếp lịch tăng lên, các lỗi đã gặp trước đó sẽ tiếp tục



xảy ra lại với tỉ lệ tương đương. Do đó, càng tăng số ngày lên bao nhiêu thì fitness value sẽ tăng lên tương ứng. Tuy nhiên, sự gia tăng này của fitness value không hoàn toàn đồng nghĩa với việc giải thuật này trở nên kém hiệu quả khi số ngày tăng lên, bởi nó chỉ là những lỗi không thể bị loại bỏ trong quá trình chọn lọc và lai ghép được lặp lại với số lần nhiều hơn với mức độ không như cũ. Vì vậy, dù độ hiệu quả của giải thuật có giảm đáng kể, kết quả trả về của giải thuật di truyền trong trường hợp này vẫn có thể chấp nhận được.

Ta thu được kết quả lưu trong file (output_2.csv)

Link kết quả: [output_2.csv](#)

Trường hợp 3: Giả định trong trường hợp số lượng nhân viên trong mỗi ca trực phải được tăng lên gấp đôi, thành 2 bác sĩ cộc I, 2 bác sĩ cộc II và 4 điều dưỡng. Khi đó, sự đa dạng các khoa sẽ dễ dàng đạt được, ngược lại, các ràng buộc về độ tuổi và số ngày nghỉ của các bác sĩ sẽ giảm đi đáng kể. Tuy vậy, dù đã giảm số ngày nghỉ của các bác sĩ xuống, độ hiệu quả của mô hình vẫn giảm đi tương đối, thể hiện qua sự gia tăng của fitness value từ 61.2 lên 95.8.

Sau khi chạy thử mô hình, ta thu được kết quả lưu trong file (output_3.csv)

Kết quả này có giá trị theo hàm fitness là , lượng phạt (penalty) của các ràng buộc (soft constraints) là:

- Faculty Penalty (đa dạng 4 khoa) = 7
- Seniority Penalty (thâm niên) = 6.4
- Schedule Penalty (tính hợp lý của lịch trình) = 451.5

Sau khi điều chỉnh số ngày nghỉ của các bác sĩ cho phù hợp, độ gia tăng của fitness value trở nên khá nhỏ và nằm trong mức chấp nhận được nếu so với trường hợp cơ bản, thể hiện rằng giải thuật này vẫn hoạt động tốt khi tăng số lượng bác sĩ mỗi ngày lên. Tuy nhiên, thay đổi này đòi hỏi sự cập nhật các điều kiện cơ bản như số ngày nghỉ, số ngày trực tối đa, yêu cầu thâm niên cao cho phù hợp với tình hình mới để đạt được hiệu quả tốt nhất.

Ta thu được kết quả lưu trong file (output_3.csv)

Link kết quả: [output_3.csv](#)

4.2 Đánh giá tính tổng quát của giải pháp

Nhìn vào kết quả của 3 trường hợp mở rộng, ta thấy rằng sau khi giảm số lượng nhân sự phù hợp với bệnh viện nhỏ (trường hợp 1), tăng số ngày trực (trường hợp 2) và tăng số người trong mỗi ca trực (trường hợp 3) thuật toán đưa ra kết quả tương đối hợp lý, đảm bảo việc phân công lịch trực cho đội ngũ bác sĩ ở trường hợp 1 (đáp ứng các yêu cầu về thâm niên, đa dạng các khoa và xoay vòng nhân sự). Tuy nhiên ở 2 trường hợp còn lại, mặc dù lượng phạt (penalty) của thâm niên và đa dạng các khoa thấp nhưng vi phạm về xoay vòng nhân sự tương đối cao dẫn tới số ngày trực và khoảng cách ngày nghỉ chưa hợp lý.

Ta phân tích ưu, nhược điểm của giải pháp như sau:

4.2.1 Ưu điểm của giải pháp

- Áp dụng thuật toán di truyền vào bài toán lập lịch mang lại lợi ích rõ rệt nhờ khả năng tìm kiếm rộng lớn và hiệu quả. Trong quá trình chọn lọc và đột biến, thuật toán đã đảm bảo các ràng buộc cứng được thỏa mãn, do đó, các bước tiếp theo chỉ cần tập trung vào việc đánh giá hàm thích nghi để chọn ra các cá thể tối ưu nhất, đáp ứng đầy đủ các yêu cầu ràng buộc còn lại.

- Có sự thử nghiệm kết quả với các xác suất chọn lọc, đột biến và số lượng thế hệ, cũng như số lượng cá thể, nhằm đảm bảo lựa chọn tỉ lệ phù hợp giúp các cá thể con được sinh ra có sự đa dạng và hợp lý.



- Đối với các ràng buộc mềm, nhóm đã thực hiện đánh giá từng ràng buộc riêng biệt và thử nghiệm kết quả độc lập, cho phép ta điều chỉnh các điều kiện phù hợp nhất với bộ dữ liệu, đồng thời xác định mức độ phạt hợp lý để đảm bảo sự ưu tiên giữa các ràng buộc một cách tối ưu.

4.2.2 Nhược điểm của giải pháp

Tuy nhiên, giải pháp hiện tại vẫn còn một số nhược điểm đáng lưu ý:

- Việc điều chỉnh tham số phụ thuộc vào dữ liệu dựa vào sự thay đổi thành viên trong số lượng khoa và nhóm bác sĩ có thể dẫn đến những biến động lớn trong kết quả phân công. Kết quả ở trường hợp 2 và 3 cho ta thấy khi số lượng khoa và bác sĩ tăng lên, hoặc khi yêu cầu về số ca trực thay đổi, hệ thống hiện tại có thể không linh hoạt và hiệu quả như mong đợi.

- Do các điều kiện phụ thuộc vào dữ liệu cụ thể, mã nguồn hiện tại chưa đủ linh hoạt và cần phải điều chỉnh thường xuyên tùy vào bài toán cụ thể. Điều này dẫn đến việc độ chính xác và hiệu quả của hệ thống chưa đạt mức tối ưu.

- Hệ số của các điều kiện ràng buộc mềm hiện vẫn còn mang tính cảm tính và đòi hỏi phải thử nghiệm nhiều lần. Do đó, khi bộ dữ liệu thay đổi đáng kể (như số lượng khoa, nhóm) hoặc khi bài toán yêu cầu phân công công việc chuyên môn, thuật toán chưa hoạt động hiệu quả và chưa cung cấp được kết quả tối ưu.

4.3 Các hướng phát triển và tối ưu cho thuật toán

Đối với bài toán lập lịch trình bằng thuật toán di truyền truyền thống, nhóm đề xuất những hướng để phát triển và tối ưu về độ hiệu quả của thuật toán cũng như tốc độ và tài nguyên tính toán như sau:

- **Tích hợp các phương pháp Hyperparameters-optimization :** điều chỉnh các tham số của thuật toán để đạt được hiệu suất tốt nhất. Đối với giải thuật di truyền, các tham số cần điều chỉnh tối ưu bao gồm: Population Size, Mutation Rate, Crossover Rate.
- **Tối ưu hóa song song (Parallel Optimization):** Sử dụng các kỹ thuật tính toán song song và phân tán để tăng tốc độ tính toán và mở rộng khả năng tìm kiếm của GA, đặc biệt là khi áp dụng trên các bài toán lập lịch lớn. Ngoài ra cũng có thể tận dụng khả năng tính toán song song của GPU để tăng tốc độ tính toán trong thuật toán di truyền.
- **Thực hiện tự điều chỉnh (Self-Adaptive GA)** Các tham số của GA như tỷ lệ lai ghép và tỷ lệ đột biến có thể tự động điều chỉnh dựa trên tình hình hiện tại của quần thể. Ngoài ra, có thể sử dụng các kỹ thuật học máy để rút ra các quy luật từ quá trình tiến hóa và điều chỉnh chiến lược di truyền tương ứng. Một đề xuất mà nhóm khá tâm đắc là sử dụng các mô hình học tăng cường (reinforcement learning) để điều chỉnh các tham số của GA một cách động.
- **Tích hợp các phương pháp tìm kiếm địa phương:** Kết hợp các thuật toán tìm kiếm địa phương (local search) như Hill Climbing hoặc Simulated Annealing vào GA có thể giúp cải thiện khả năng khai thác cục bộ của thuật toán, từ đó tìm ra các giải pháp tốt hơn.



5 Tài liệu tham khảo

References

- [1] Tournament Selection (GA), GeeksforGeeks, <https://www.geeksforgeeks.org/tournament-selection-ga>, ngày truy cập 19/07/2024
- [2] Nero Phung, Genetic Algorithm - Giải thuật di truyền, Nero's Blog, <https://nerophung.github.io/2020/05/28/genetic-algorithm>, ngày truy cập 17/07/2024
- [3] Daniel W.Dyer, Evolutionary Computation in Java - A Practical Guide to the Watchmaker Framework, <https://watchmaker.uncommons.org/manual/index.html>, ngày truy cập 16/7/2024
- [4] Phạm Trường Duy, Genetic Algorithms - Giải Thuật Di Truyền - Python, Duy Phạm Blog, <https://duyphamdata.blogspot.com/2017/11/genetic-algorithm-giai-thuat-di-truyen.html>, 2017
- [5] Xiong Luo, Qian Qian and Yun Fa Fu, Improved Genetic Algorithm for Solving Flexible Job Shop Scheduling Problem, China, 3rd International Conference on Mechatronics and Intelligent Robotics, 2019
- [6] Burak Tagtekin, Mahiye Uluya ~ gmur Öztürk and Mert Kutay Sezer, A Case Study: Using Genetic Algorithm for Job Scheduling Problem, Turkey, Huawei Turkey RD Center Istanbul Turkey, 2021