



Ứng dụng giải thuật di truyền trong xây dựng thời khóa biểu tại các Trường Đại học

Nguyễn Phồn Lữa

Khoa Công nghệ Thông tin, Trường Đại học Công nghệ Đông Á

Email: luanp@eaut.edu.vn

TÓM TẮT

Việc xây dựng thời khóa biểu (TKB) tại các trường đại học là một bài toán tối ưu hóa tổ hợp phức tạp, đòi hỏi thỏa mãn đồng thời nhiều ràng buộc về tài nguyên, thời gian và nhân sự. Nghiên cứu này đề xuất một phương pháp hai bước sử dụng giải thuật di truyền (GA) để phân bổ phòng học, ca học và ngày học cho các lớp, kết hợp với giải thuật phân việc tham lam (GAA) để phân công giảng viên dựa trên năng lực và lịch dạy ưa thích. Hàm mục tiêu được thiết kế nhằm giảm thiểu vi phạm các ràng buộc cứng như xung đột phòng học, ca học, năng lực giảng viên, đồng thời tối ưu hóa các tiêu chí mềm như sử dụng hiệu quả tài nguyên và ưu tiên giảng viên cơ hữu. Kết quả cho thấy phương pháp này tạo ra TKB khả thi, đáp ứng các ràng buộc phức tạp và nâng cao hiệu quả tổ chức giảng dạy tại các trường đại học.

Từ khóa: Giải thuật di truyền, lập lịch học, thời khóa biểu, tối ưu.

ABSTRACT

Course timetabling in universities is a complex combinatorial optimization problem that requires the simultaneous satisfaction of multiple constraints related to resources, time, and personnel. This study introduces a two-stage approach that employs a Genetic Algorithm (GA) to allocate classrooms, time slots, and teaching days, combined with a Greedy Assignment Algorithm (GAA) to assign lecturers according to their expertise and preferred schedules. The objective function is formulated to minimize violations of hard constraints—such as classroom and time-slot conflicts and lecturer qualifications—while optimizing soft criteria, including efficient resource utilization and prioritization of full-time faculty. Experimental results demonstrate that the proposed method generates feasible timetables, effectively addresses complex constraints, and improves the overall efficiency of academic scheduling in universities.

Keywords: Genetic Algorithm, scheduling, university timetabling, optimization

1. MỞ ĐẦU

Trong bối cảnh giáo dục đại học ngày càng phát triển về quy mô và mức độ đa dạng trong chương trình đào tạo, việc xây dựng thời khóa biểu (TKB) giảng dạy trở thành một nhiệm vụ



phức tạp và có tính chất quyết định đến hiệu quả vận hành của một cơ sở giáo dục. Bài toán này không chỉ yêu cầu sự phân phối hợp lý các nguồn lực hữu hạn như giảng viên, phòng học, và thời gian, mà còn phải tuân thủ nhiều ràng buộc đồng thời liên quan đến học phần, chuyên môn, và nhu cầu thực tế của người học. Đây là một bài toán tối ưu tổ hợp rất khó, thu hút sự quan tâm của nhiều nhà nghiên cứu. Bài toán lập thời khóa biểu cho các khóa học đại học đặc biệt khó giải quyết do quy mô lớn và sự tồn tại của nhiều ràng buộc cứng và mềm đầy thách thức. [1, tr. 1].

Theo cách truyền thống, công việc lập TKB thường được thực hiện thủ công hoặc bán tự động, phụ thuộc nhiều vào kinh nghiệm của cán bộ quản lý đào tạo. Tuy nhiên, khi quy mô sinh viên tăng lên, số lượng lớp học, học phần và giảng viên ngày càng lớn, việc áp dụng các phương pháp thủ công không còn khả thi. Điều này đặt ra nhu cầu cấp thiết phải nghiên cứu và phát triển các phương pháp tính toán hiệu quả để tự động hóa và tối ưu hóa quá trình xây dựng TKB.

Cho đến nay đã có nhiều nghiên cứu giải pháp cho bài toán này được thực hiện trên nhiều giải thuật khác nhau, trong đó ứng dụng GA trong xây dựng TKB có số lượng nghiên cứu nổi trội [2, tr. 10]. Từ kết quả khảo sát các nghiên cứu thực tế [2] cho thấy tầm quan trọng của bài toán TKB trong thực tiễn và mức độ ứng dụng rộng rãi của GA trong nghiên cứu để giải quyết bài toán này. Mặc dù vậy, các bài toán TKB bên cạnh việc có chung những ràng buộc về định độ tài nguyên (giảng viên, phòng học,...) còn có những ràng buộc riêng mang tính đặc thù mỗi nơi nên khó có một giải pháp có thể áp dụng phổ quát.

Trong nghiên cứu này, tác giả đề xuất một phương pháp xây dựng TKB giảng dạy cho trường đại học qua hai bước: Bước 1 sử dụng GA để xây dựng lịch học cho các lớp theo từng ngày, ca và phòng học cụ thể; bước 2 thực hiện phân công giảng viên giảng dạy cho các học phần tại các lớp bằng giải thuật phân việc tham lam (Greedy Assignment Algorithm - GAA). Về nội dung bài toán được nghiên cứu cũng đưa ra nhiều ràng buộc mang tính đặc thù trong môi trường giáo dục đại học tại Việt nam.

Bài báo được tổ chức như sau: Phần 2 - Mô tả bài toán trình bày chi tiết bài toán xây dựng TKB và các ràng buộc liên quan. Phần 3 giới thiệu các giải thuật được sử dụng trong nghiên cứu. Phần 4 trình bày cách thức triển khai GA, GAA trong bài toán. Phần 5 đánh giá thực nghiệm kết quả trên bộ dữ liệu mô phỏng, trao đổi thảo luận và định hướng nghiên cứu tiếp theo.

2. MÔ TẢ BÀI TOÁN

2.1. Các thành phần chính của bài toán



- **Lớp học:** Mỗi lớp tương ứng với một nhóm sinh viên học chung chương trình đào tạo.
- **Học phần:** Các môn học phải được giảng dạy cho từng lớp trong học kỳ.
- **Giảng viên:** Người chịu trách nhiệm giảng dạy các học phần, có giới hạn về thời gian và chuyên môn.
- **Phòng học:** Có giới hạn về sức chứa, thiết bị, và số lượng.
- **Thời gian học:** Các khung thời gian trong tuần, thường được chia theo các ngày học và ca học trong ngày.

2.2. Ràng buộc của bài toán

Bài toán bao gồm hai nhóm ràng buộc chính:

- **Ràng buộc cứng - bắt buộc phải thỏa mãn:**
 - Tất cả các học phần của mỗi lớp phải được xếp lịch đủ số giờ học theo số tín chỉ.
 - Không xếp lịch học vào các ngày lễ hoặc Chủ nhật.
 - Lịch học chỉ được xếp trong khung thời gian của đợt học.
 - Chỉ phân công giảng viên dạy các học phần họ có khả năng giảng dạy.
 - Phòng học được xếp phải có sức chứa đủ cho số học sinh của lớp.
 - Không để xảy ra xung đột về phòng học, giảng viên, hoặc lớp học tại cùng một thời điểm.
 - Mỗi học phần của một lớp chỉ được phân công cho một giảng viên duy nhất trong suốt đợt học.
- **Ràng buộc mềm - nên được tối thiểu hoá vi phạm:**
 - Lịch học nên đồng nhất giữa các tuần, trừ các tuần có lịch học bù.
 - Các ca giảng dạy trong một ngày của một giảng viên nên liên tục.
 - Các ca học trong một ngày của một lớp nên liên tục.
 - Các ca học liên tiếp của cùng một lớp nên được xếp vào cùng một phòng học.
 - Mỗi giảng viên không dạy quá k ca/ngày.
 - Phân bổ thời lượng giảng dạy đều giữa các giảng viên cơ hữu, tránh chênh lệch lớn.
 - Ưu tiên giảng viên cơ hữu hơn giảng viên thỉnh giảng.
 - Hạn chế tối đa sử dụng ca buổi tối.

2.3. Mục tiêu của bài toán

Mục tiêu của hệ thống là tạo ra một TKB hợp lệ, thỏa mãn tất cả các ràng buộc cứng và tối ưu hoá mức độ hài lòng tổng thể thông qua việc giảm thiểu số lượng vi phạm ràng buộc



mềm. Một TKB chất lượng cao không chỉ đảm bảo tính khả thi mà còn hỗ trợ cải thiện hiệu quả giảng dạy và học tập.

2.4. Tính chất của bài toán

- Không gian nghiệm lớn: Số lượng tổ hợp có thể có của các học phần, thời gian và phòng học là rất lớn.
- Nhiều ràng buộc tương tác: Các yếu tố như lớp, giảng viên và phòng học có thể xung đột lẫn nhau.
- Tính NP: Bài toán thuộc nhóm NP-khó, không có giải pháp đa thức tổng quát, do đó cần sử dụng các phương pháp giải gần đúng (heuristic hoặc metaheuristic).

2.5. Hàm mục tiêu đánh giá TKB

Chất lượng TKB được đánh giá dựa trên mức độ thỏa mãn ràng buộc cứng và tối ưu hóa ràng buộc mềm. Hàm mục tiêu sử dụng để đánh giá chất lượng TKB được định nghĩa như sau:

$$F = \sum_i w_i \cdot c_i$$

Trong đó:

- c_i : Số lần vi phạm ràng buộc r_i .
- w_i : Trọng số thể hiện mức độ quan trọng của ràng buộc r_i .
- **Mục tiêu:** tối thiểu hoá giá trị hàm F nhằm giảm thiểu vi phạm các ràng buộc. $F = 0$ khi tất cả ràng buộc được thỏa mãn.

3. CÁC GIẢI THUẬT ÁP DỤNG

3.1. Giải thuật di truyền (GA)

Giải thuật di truyền được John Holland giới thiệu vào những năm 1960 tại Đại học Michigan, lấy cảm hứng từ lý thuyết tiến hóa của Charles Darwin. GA mô phỏng các quá trình sinh học như chọn lọc tự nhiên, lai ghép và đột biến để tìm kiếm lời giải tối ưu trong không gian tìm kiếm lớn và phức tạp. Từ những năm 1980, GA đã được áp dụng rộng rãi trong các bài toán tối ưu hóa, từ kỹ thuật đến quản lý và khoa học dữ liệu. GA được sử dụng trong nhiều lĩnh vực như lập lịch, tối ưu hóa mạng, học máy, và thiết kế kỹ thuật.

- **Khái niệm cốt lõi:**



- Quần thể (Population): Là tập hợp các lời giải tiềm năng. Mỗi lời giải được gọi là một cá thể (individual), thường được mã hóa theo một cách nào đó và được gọi là nhiễm sắc thể (chromosome).
- Hàm thích nghi (Fitness Function): Đánh giá chất lượng của mỗi cá thể dựa trên mức độ thỏa mãn các ràng buộc.
- Các toán tử di truyền: Bao gồm chọn lọc (selection), lai ghép (crossover), và đột biến (mutation) để tạo ra các thế hệ mới với hy vọng cải thiện chất lượng lời giải.
- **Quy trình thực hiện:**
 - (1). **Khởi tạo:** Tạo ngẫu nhiên một quần thể ban đầu gồm các cá thể như những lời giải tiềm năng ban đầu của bài toán.
 - (2). **Đánh giá:** Sử dụng hàm thích nghi để đánh giá chất lượng của mỗi cá thể.
 - (3). **Chọn lọc:** Lựa chọn các cá thể tốt nhất (dựa trên giá trị thích nghi) để sinh sản.
 - (4). **Lai ghép:** Kết hợp các cá thể để tạo ra cá thể mới.
 - (5). **Đột biến:** Thay đổi ngẫu nhiên một phần của cá thể để duy trì tính đa dạng trong quần thể.
 - (6). **Lặp lại:** Tiếp tục các bước từ đánh giá đến đột biến cho đến khi đạt được lời giải tối ưu hoặc số thế hệ tối đa.

Ứng dụng trong xây dựng TKB: Trong bài toán này, GA được sử dụng để tạo lịch học cho các lớp. Mỗi cá thể là một thời khóa biểu hoàn chỉnh, mã hóa các thông tin về thời gian, phòng học và môn học. Hàm thích nghi đảm bảo các ràng buộc như không trùng lịch, phòng học phù hợp và cân bằng tải công việc. GA tìm kiếm lời giải gần tối ưu thông qua việc tiến hóa các lịch học qua nhiều thế hệ.

3.2. Giải thuật phân việc tham lam (GAA)

GAA có nguồn gốc từ các phương pháp tối ưu hóa đơn giản, được hệ thống hóa trong lý thuyết thuật toán. Ý tưởng cốt lõi là đưa ra lựa chọn tốt nhất tại mỗi bước với hy vọng đạt được lời giải tối ưu toàn cục. GAA được áp dụng trong các bài toán như lập lịch, phân bổ tài nguyên, nén dữ liệu (Huffman coding), và quản lý mạng. Trong bài toán thời khóa biểu, GAA được sử dụng để phân công giảng viên cho các học phần, đảm bảo tối ưu hóa theo các tiêu chí như trình độ chuyên môn và lịch trống của giảng viên.

- **Khái niệm cốt lõi:**



- Lựa chọn tham lam: Tại mỗi bước, giải thuật chọn lựa chọn tối ưu cục bộ dựa trên một tiêu chí cụ thể, ví dụ: chọn giảng viên có trình độ phù hợp nhất cho một học phần.
- Tính chất tối ưu cục bộ: Giải thuật giả định rằng các lựa chọn cục bộ sẽ dẫn đến lời giải tối ưu toàn cục, mặc dù điều này không luôn đúng với mọi bài toán.
- Tính không quay lui: Một khi lựa chọn được đưa ra, giải thuật không xem xét lại các quyết định trước đó.

▪ **Quy trình thực hiện:**

- (1). **Khởi tạo:** Xác định tập hợp các học phần cần phân công và danh sách giảng viên với thông tin về chuyên môn, lịch trống.
- (2). **Sắp xếp:** Sắp xếp các học phần hoặc giảng viên theo một tiêu chí ưu tiên, ví dụ: mức độ khó của học phần hoặc trình độ chuyên môn của giảng viên.
- (3). **Phân công:** Tại mỗi bước, chọn giảng viên tốt nhất (theo tiêu chí như chuyên môn, thời gian trống) để giảng dạy học phần hiện tại.
- (4). **Cập nhật:** Loại bỏ học phần đã được phân công và cập nhật lịch trống của giảng viên.
- (5). **Lặp lại:** Tiếp tục cho đến khi tất cả học phần được phân công.

Ứng dụng trong xây dựng TKB: GAA được sử dụng để phân công giảng viên cho các học phần sau khi lịch học đã được tạo bởi GA. Mỗi học phần được gán cho giảng viên dựa trên các tiêu chí như chuyên môn, số giờ giảng đã phân cho giảng viên, lịch trống. Giải thuật đảm bảo rằng mỗi học phần được phân công một cách nhanh chóng và hiệu quả, dù không đảm bảo tối ưu toàn cục.

3.3. Kết hợp GA và GAA

Trong bài toán xây dựng thời khóa biểu, GA và GAA được kết hợp để tận dụng ưu điểm của cả hai. GA tạo ra các lịch học tối ưu cho các lớp, xử lý các ràng buộc phức tạp về thời gian và phòng học. Sau đó, GAA phân công giảng viên cho các học phần trong lịch học này, đảm bảo tính khả thi và hiệu quả. Sự kết hợp này cho phép giải quyết bài toán một cách toàn diện, cân bằng giữa chất lượng lịch học và tính thực tiễn của việc phân công giảng viên.

4. GIẢI PHÁP THỰC HIỆN

4.1. Bước 1 - Xây dựng lịch học

4.1.1. Tổ chức dữ liệu



Để giải quyết bài toán lập thời khóa biểu, việc tổ chức và biểu diễn dữ liệu đóng vai trò quan trọng trong việc kiểm tra các ràng buộc và tối ưu hóa giải pháp. Đối với việc xây dựng lịch học, dữ liệu được thiết kế để phản ánh các tài nguyên thời gian, không gian, và yêu cầu của các lớp học, đồng thời hỗ trợ biểu diễn các giải pháp tiềm năng một cách hiệu quả. Trong bước này dữ liệu được chia thành bốn thành phần chính: thông tin tài nguyên cấp pháp cho các lớp tại mỗi thời điểm (slot), thông tin lớp học, cá thể (giải pháp), và quần thể. Các cấu trúc dữ liệu này được mô tả chi tiết dưới đây, kèm theo các biểu diễn toán học để minh họa.

- **Slot:** Slot là một đơn vị tài nguyên học tập có thể được gán cho một lớp học tại một thời điểm cụ thể. Mỗi slot bao gồm bốn thuộc tính: phòng học, ca học, ngày học, và sức chứa của phòng. Những slot này đại diện cho toàn bộ tài nguyên có sẵn trong suốt đợt học, được sử dụng để xây dựng thời khóa biểu và kiểm tra các ràng buộc, chẳng hạn như đảm bảo phòng học có đủ sức chứa cho số học sinh của lớp.

Thông tin slot thời gian được lưu trữ dưới dạng một tập hợp các mảng, mỗi mảng chứa thông tin về một thuộc tính cụ thể: ngày học, ca học, phòng học, sức chứa phòng.

Có thể mô tả cách biểu diễn dữ liệu Slot và các thành phần của nó như sau:

- $R = \{r_1, r_2, \dots, r_n\}$: Tập hợp các phòng học, sức chứa của phòng r_i là m_i .
- $C = \{c_1, c_2, \dots, c_m\}$: Tập hợp các ca học (ví dụ $m = 5$).
- $D = \{d_1, d_2, \dots, d_k\}$: Tập hợp các ngày học.
- $S = [(r_i, c_j, d_k, m_i)], r_i \in R, c_j \in C, d_k \in D, m_i \in N$
- $|S| = |R| \times |C| \times |D|$

- **Class:** Lớp học đại diện cho một đơn vị cần được xếp lịch, bao gồm thông tin về số học sinh trong lớp và số slot thời gian cần thiết trong đợt học. Mỗi lớp phải được gán đúng số slot yêu cầu, và các phòng học tại những slot này phải có sức chứa đủ cho số học sinh của lớp. Tập hợp tất cả các lớp học phản ánh toàn bộ nhu cầu lập lịch trong đợt học.

Thông tin lớp học được lưu trữ dưới dạng một danh sách, trong đó mỗi lớp được mô tả bởi hai thuộc tính: số học sinh, số slot yêu cầu. Mô tả toán học cho biểu diễn thông tin Class như sau:

- $c = (q, s)$: Thông tin một lớp, q là số lượng sinh viên trong lớp, s là số slot cần cấp phát trong đợt học cho lớp.
- $C = [c_i] = [(q_i, s_i)]$: Danh sách các lớp trong toàn trường.



▪ **Cá thể (giải pháp):** Cá thể là một giải pháp tiềm năng cho bài toán lập thời khóa biểu, thể hiện cách gán các slot thời gian cho các lớp học. Mỗi cá thể là một mảng, trong đó mỗi phần tử cho biết lớp học được gán vào một slot cụ thể hoặc để trống. Một giải pháp hợp lệ phải đảm bảo rằng mỗi lớp được gán đúng số slot yêu cầu, và phòng học tại các slot được gán có sức chứa không nhỏ hơn số học sinh của lớp. Cá thể là đơn vị cơ bản trong giải thuật di truyền, được đánh giá chất lượng thông qua hàm mục tiêu.

Cá thể được biểu diễn bằng một mảng có độ dài bằng số slot. Mỗi phần tử trong mảng là một số nguyên thể hiện slot được gán cho lớp nào. Ví dụ, nếu phần tử tại vị trí i trong mảng cá thể bằng k thì slot thứ i được gán cho lớp thứ k . Nếu phần tử bằng 0, slot i không được gán, tức là để trống.

- $C[k] \leftarrow S[i]$: Slot $S[i]$ được gán cho $C[k]$
- $I = [0, 1, \dots, n], n = |I| = |S|$
- $I[i] = k \Leftrightarrow C[k] \leftarrow S[i]$

▪ **Quần thể:** Quần thể là tập hợp nhiều cá thể, đại diện cho nhóm giải pháp tiềm năng tại một thời điểm trong giải thuật di truyền. Quần thể có kích thước cố định, ví dụ, 30 cá thể, để cân bằng tính đa dạng và hiệu quả tính toán. Quần thể được khởi tạo ngẫu nhiên, đánh giá từng cá thể qua hàm mục tiêu, và tiến hóa qua các thế hệ bằng toán tử di truyền (lựa chọn, lai ghép, đột biến). Quần thể liên kết các cá thể với dữ liệu slot và lớp học thông qua hàm mục tiêu, tăng khả năng tìm lịch học tối ưu.

4.1.2. Xây dựng GA

Việc thực hiện GA được bắt đầu bằng việc khởi tạo quần thể và sau đó là quá trình lặp lại qua nhiều thế hệ thực hiện các toán tử di truyền theo quy trình hoạt động của GA được giới thiệu ở trên. Phần giả mã dưới đây thể hiện cách khởi tạo quần thể và thực hiện của GA.

▪ **Khởi tạo quần thể**

```
// Khởi tạo quần thể
Function InitPopulation(n_p, n_s, C, S)
    P ← []
    For i ← 1 to n_p
        A ← Zeros(n_s)
        For k ← 1 to n_c
            valid ← {i | S.m[i] ≥ C[k].q}
            slots ← Random(valid, C[k].s)
            For i in slots do A[i] ← k
        Append Repair(A, S, C) to P
```



Return P
EndFunction

▪ Thực hiện GA

```
// Thuật toán di truyền lập TKB
Algorithm GeneticAlgorithm(S, C, n_d, n_p, n_s, n_g, p_c, p_m, t)
  P ← InitPopulation(n_p, n_s, C, S)
  cache ← {}, best_A ← null, best_f ← ∞, f_hist ← []
  For A in P
    f ← Fitness(A, S, C, n_d)
    cache[A] ← f
    If f < best_f then best_f ← f; best_A ← A
  Append best_f to f_hist
  For gen ← 1 to n_g
    new_P ← [best_A]; tasks ← []
    While Size(new_P) < n_p
      p1, p2 ← Select(P, t, cache)
      c1, c2 ← Cross(p1, p2)
      If Random() < p_m then c1 ← Mutate(c1); c2 ← Mutate(c2)
      c1 ← Repair(c1, S, C); Append c1 to new_P, tasks
      If Size(new_P) < n_p then
        c2 ← Repair(c2, S, C); Append c2 to new_P, tasks
    For A in tasks
      f ← Fitness(A, S, C, n_d)
      cache[A] ← f
      If f < best_f then best_f ← f; best_A ← A
    P ← new_P
    Append best_f to f_hist
  Return best_A, f_hist
EndAlgorithm
```

4.2. Bước 2 - Phân công giảng viên

Sau khi thời khóa biểu được xây dựng ở Bước 1, bước này tập trung vào phân công giảng viên cho các slot đã xác định. Bài toán yêu cầu gán mỗi slot cho một giảng viên phù hợp, đảm bảo các ràng buộc cứng như khả năng chuyên môn, lịch đề xuất, và giới hạn giờ dạy, giảm sử dụng ca tối. GAA được sử dụng nhờ hiệu quả tính toán cao, đưa ra các quyết định cục bộ tối ưu dựa trên tiêu chí ưu tiên.

4.2.1. Tổ chức dữ liệu

Dữ liệu cho bài toán phân công giảng viên gồm hai thành phần chính: slot đã giao cho các lớp và thông tin giảng viên. Thông tin về Slot đã được mô tả ở bước 1. Thông tin giảng viên bao gồm các thuộc tính như: Khả năng chuyên môn, danh sách các môn học mà giảng viên có thể dạy, ca học đề xuất.



4.2.2. Thực hiện GAA

Phần giả mã dưới đây thể hiện quá trình phân công giảng viên cho các lớp theo quy trình hoạt động của GAA được giới thiệu ở phần trên.

```
// Phân công giảng viên cho lớp
Algorithm AssignLecturers(classes, lecturers, period, max_slots_per_day)
    Set random seed
    groups ← []
    For class in classes
        For subject_id, slots in class.map_courses_slots(period)
            If slots ≠ ∅
                subject ← FindSubject(class, subject_id)
                Append {class_id, subject_id, slots, credits} to groups
    Sort groups by slots DESC
    lecturer_schedule ← {}, lecturer_slots ← {}
    lecturer_credits ← {}, subject_lecturers ← {}
    For l in lecturers
        For subject, _ in l.subjects
            Append l to subject_lecturers[subject.id]
        Init schedule, slots, credits for l
    unassigned ← []
    For g in groups
        candidates ← []
        For l in subject_lecturers[g.subject_id]
            If IsFeasible(l, g.slots, max_slots_per_day, lecturer_schedule,
                lecturer_slots)
                ability ← GetAbility(l, g.subject_id)
                cont ← Continuity(g.slots, lecturer_slots[l.id])
                priority ← (1 if l.permanent else 0) + ability/10 - cont/10
                Append (l, priority, lecturer_credits[l.id]) to candidates
        If candidates = ∅
            Append g to unassigned; Continue
    Sort candidates by (credits ASC, priority DESC)
    l ← candidates[0][0]
    For slot in g.slots
        Add slot to lecturer_schedule[l.id];
        Append slot.c to lecturer_slots[l.id][slot.d]
    lecturer_credits[l.id] += g.credits
    Append (g.class_id, g.subject_id) to l.assignments
    Find class by id and set lecturer_map[g.subject_id] ← l.name
EndAlgorithm
```

5. KẾT QUẢ VÀ THẢO LUẬN

5.1. Tập Dữ liệu thử nghiệm



Để đánh giá hiệu quả của mô hình đề xuất, tác giả xây dựng tập dữ liệu mô phỏng dựa trên các thông số thực tế như sau:

- **Phòng học:** 50 phòng, sức chứa 50–90 sinh viên.
- **Thời gian học:** Từ 14/4 đến 24/5/2025, học từ thứ 2 đến thứ 7 (mỗi ngày 5 ca, mỗi ca 3 giờ), trừ các ngày nghỉ lễ (30/4–03/5). Thời gian nghỉ lễ trong dữ liệu thử nghiệm được lấy từ ngày nghỉ lễ thực tế trong đợt học năm 2025.
- **Học phần:** Trích xuất từ chương trình ngành của ngành Công nghệ Thông tin, mỗi học phần 2–4 tín chỉ.
- **Lớp học:** 35 lớp, mỗi lớp 40–70 sinh viên, học 2–4 học phần.
- **Giảng viên:** 80 giảng viên (50% cơ hữu), mỗi người có danh sách học phần có thể giảng dạy kèm theo điểm đánh giá năng lực.

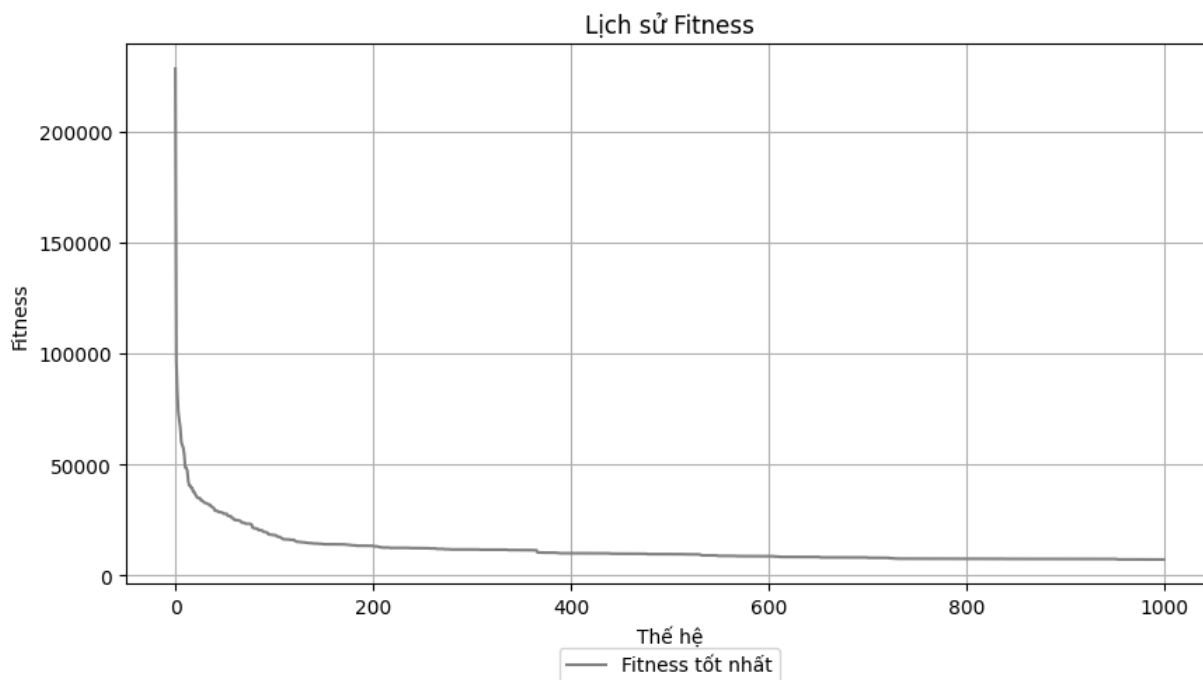
5.2. Tham số thực nghiệm

Ứng dụng được phát triển với ngôn ngữ lập trình Python và các thư viện hỗ trợ như numpy, pandas, matplotlib. Các tham số chính của giải thuật di truyền gồm: kích thước quần thể 100, số thế hệ 1000, xác suất lai ghép 0.8, xác suất đột biến 0.3. Trọng số hàm mục tiêu được thiết lập là: ($w_1 = 1000$, $w_2 = 500$, $w_3 = 200$, $w_4 = 50$, $w_5 = 20$).

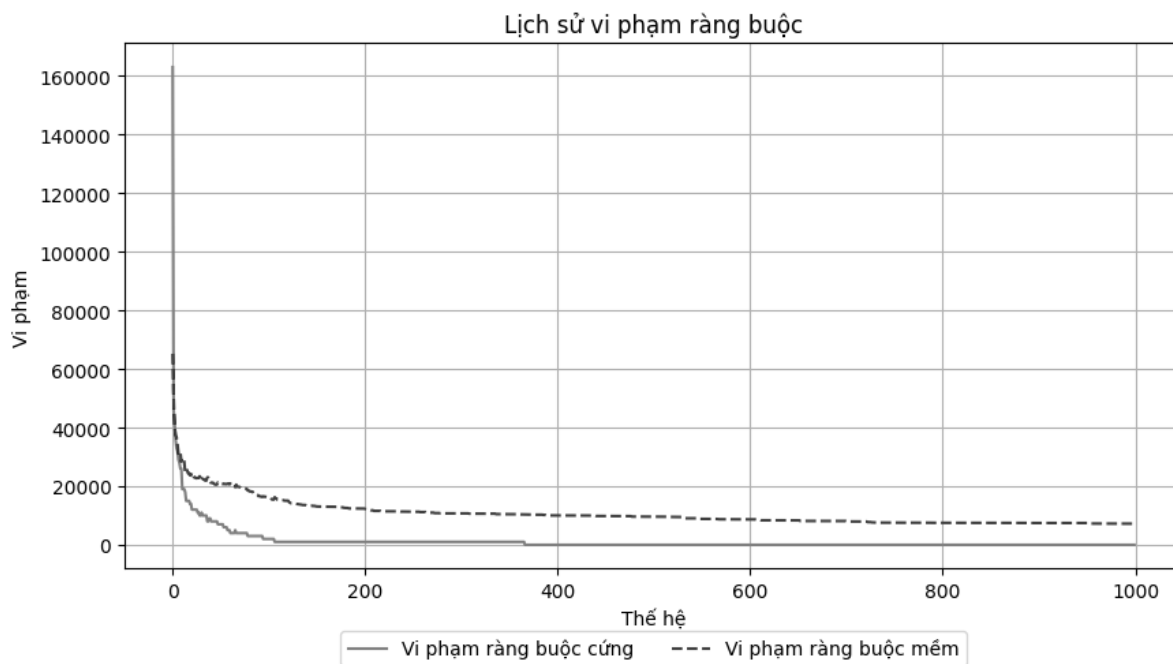
5.3. Kết quả hội tụ và đáp ứng ràng buộc

Hình 1 cho thấy giá trị fitness giảm mạnh từ khoảng 240,000 xuống gần 7,000 sau 1000 thế hệ, thể hiện quá trình hội tụ ổn định của giải thuật di truyền.

Hình 2 minh họa quá trình giảm vi phạm ràng buộc: các ràng buộc cứng được loại bỏ hoàn toàn sau khoảng 400 thế hệ, trong khi số vi phạm ràng buộc mềm duy trì ở mức ~7000 - phản ánh sự giới hạn trong việc tối ưu đồng thời tất cả tiêu chí mềm.



Hình 1. Biểu đồ lịch sử giá trị Fitness qua các thế hệ.



Hình 2. Biểu đồ lịch sử vi phạm ràng buộc qua các thế hệ.

5.4. Kết quả Phân công Giảng viên

Bước phân công sử dụng GAA đã gán thành công giảng viên cho toàn bộ học phần, đảm bảo 100% phù hợp năng lực chuyên môn. Giảng viên thỉnh giảng chỉ được sử dụng khi không



còn giảng viên cơ hữu đủ điều kiện. Kết quả được lưu vào tệp schedule.csv – có thể sử dụng làm TKB chính thức.

Bảng 1. Trích lịch học trong TKB cho ngày 15/4/2025

Lớp	SL SV	Ngày	Ca	Phòng	Sức chứa	Học phần	Giảng viên
1	59	15/4/2025	4	45	65	Toán rời rạc	GV_38
2	57	15/4/2025	2	45	84	Đại số tuyến tính	GV_22
3	54	15/4/2025	1	47	55	Toán rời rạc	GV_32
3	54	15/4/2025	2	50	76	Trí tuệ nhân tạo	GV_33
4	48	15/4/2025	3	50	73	An toàn thông tin	GV_8
4	48	15/4/2025	4	50	59	Trí tuệ nhân tạo	GV_31
5	63	15/4/2025	4	38	70	Đại số tuyến tính	GV_86
6	67	15/4/2025	1	45	74	Đồ án tốt nghiệp	GV_10
6	67	15/4/2025	2	40	71	Mạng máy tính	GV_18
6	67	15/4/2025	3	45	81	Kỹ năng mềm	GV_15

5.5. Thảo luận

Kết quả thực nghiệm cho thấy phương pháp đề xuất đạt hiệu quả cao trong việc xây dựng thời khóa biểu. Cụ thể, giải thuật di truyền đã thể hiện khả năng xử lý hiệu quả không gian tổ hợp lớn, loại bỏ hoàn toàn các xung đột về phòng học và ca học sau khoảng 400 thế hệ. Sự hội tụ nhanh của hàm fitness chứng tỏ độ phù hợp của giải thuật với bài toán lập thời khóa biểu phức tạp.

Bên cạnh đó, GAA được sử dụng trong bước phân công giảng viên cho thấy tính linh hoạt và hiệu suất cao, với thời gian tính toán thấp và khả năng phân công thành công toàn bộ học phần cho các giảng viên phù hợp. Thời khóa biểu cuối cùng đáp ứng đầy đủ các ràng buộc cứng (không có xung đột), đảm bảo tính đồng nhất giữa các tuần và phân bố hợp lý các tiết học trong tuần.

Tuy nhiên, phương pháp vẫn tồn tại một số hạn chế. Thứ nhất, mặc dù hiệu quả, giải thuật di truyền yêu cầu số thế hệ lớn (lên đến 1000), kéo theo thời gian xử lý tăng đáng kể khi quy mô dữ liệu (số lớp hoặc học phần) tăng. Thứ hai, GAA không đảm bảo tối ưu toàn cục, dẫn đến một số trường hợp phân công chưa hoàn toàn phù hợp với nguyện vọng giảng viên, đòi hỏi sự can thiệp thủ công – một yếu tố phổ biến trong thực tế do yêu cầu linh hoạt từ phía giảng viên. Cuối cùng, hiệu quả của mô hình phụ thuộc đáng kể vào việc lựa chọn trọng số trong hàm mục



tiêu. Nếu các trọng số này không được điều chỉnh hợp lý, thuật toán có thể ưu tiên sai các tiêu chí mềm, ảnh hưởng đến chất lượng thời khóa biểu.

Trong tương lai, nghiên cứu sẽ tập trung vào việc cải tiến các thao tác di truyền thông qua các chiến lược lai ghép và đột biến thông minh, đồng thời tích hợp giải thuật tìm kiếm cục bộ (local search) vào bước phân công giảng viên nhằm giảm số lượng học phần chưa được gán. Bên cạnh đó, việc tự động hóa quá trình tối ưu trọng số, thông qua kỹ thuật học máy hoặc các phương pháp thử nghiệm có hệ thống, được kỳ vọng sẽ nâng cao hơn nữa chất lượng và tính thực tiễn của giải pháp.

TÀI LIỆU THAM KHẢO

- [1]. M. C. Chen, S. L. Goh, N. R. Sabar and G. Kendall. *A survey of university course timetabling problem: Perspectives, trends and opportunities*. IEEE Access, vol. 9, pp. 106515–106529, 2021
- [2]. Bashab, Abeer, et al (2023). *Optimization Techniques in University Timetabling Problem: Constraints, Methodologies, Benchmarks, and Open Issues*. *Computers, Materials & Continua*, vol. 74, no. 3, 2023, pp. 6462–6481. <https://doi.org/10.32604/cmc.2023.034051>.
- [3]. Andrew Reid East. (2019). *Timetable Scheduling via Genetic Algorithm*. National University of Ireland, Galway, Computer Science and Information Technology
- [4]. Achini Kumari Herath (2017). *Genetic Algorithm For University Course Timetabling Problem*. University of Mississippi.
- [5]. Richard E. Neapolitan (2015). *Foundations of Algorithms 5th edition*. Jones & Bartlett Learning.