

## NGHIÊN CỨU BÀI TOÁN LẬP LỊCH BIỂU THEO HƯỚNG TIẾP CẬN MỤC TIÊU, ÁP DỤNG CHO TRƯỜNG ĐẠI HỌC

Trần Hải Thanh\*, Nguyễn Lan Oanh, Nguyễn Thu Phương, Nguyễn Thị Duyên  
Trường Đại học Công nghệ thông tin & Truyền thông - Đại học Thái Nguyên, Việt Nam

### ARTICLE INFORMATION TÓM TẮT

**Journal:** Vinh University  
Journal of Science  
**ISSN:** 1859-2228

**Volume:** 52  
**Issue:** 3A

**\*Correspondence:**  
ththanh@ictu.edu.vn

**Received:** 15 May 2023

**Accepted:** 27 June 2023

**Published:** 20 September 2023

#### **Citation:**

Trần Hải Thanh, Nguyễn Lan Oanh, Nguyễn Thu Phương, Nguyễn Thị Duyên (2023). Nghiên cứu bài toán lập lịch biểu theo hướng tiếp cận mục tiêu, áp dụng cho trường đại học. *Vinh Uni. J. Sci.* Vol. 52 (3A), pp. 105-115 doi: 10.56824/vujs.2023a063

### OPEN ACCESS

Copyright © 2023. This is an Open Access article distributed under the terms of the Creative Commons Attribution License (CC BY NC), which permits non-commercially to share (copy and redistribute the material in any medium) or adapt (remix, transform, and build upon the material), provided the original work is properly cited.

Bài báo trình bày kết quả nghiên cứu về bài toán lập lịch biểu cho trường đại học, một bài toán đòi hỏi phải lập lịch cho nhiều tiết học, các lớp học, các giảng viên và phòng học khác nhau, cùng với nhiều ràng buộc khác như giờ giảng dạy của giảng viên, sức chứa của phòng học và các tài nguyên khác. Giải thuật Tabu search được đề xuất sử dụng để giải quyết bài toán này. Các giải pháp tốt nhất được tìm kiếm dựa trên một bộ nhớ thích nghi và thăm dò phản ứng, và sử dụng các ràng buộc và hạn chế của bài toán để tạo ra một danh sách các điều kiện Tabu để đảm bảo không lặp lại các giải pháp không tối ưu. Sau khi khởi tạo và cập nhật bộ nhớ thích nghi, giải thuật Tabu search sẽ thực hiện tìm kiếm các lời giải tiếp theo dựa trên các điều kiện Tabu và bộ nhớ thích nghi. Kết quả cho thấy giải thuật Tabu search có thể tìm ra các lời giải tối ưu cho bài toán lập lịch biểu cho trường đại học, giúp cho quản lý giảng dạy đạt được hiệu quả cao hơn.

**Từ khóa:** Lập lịch biểu; trường đại học; tìm kiếm Tabu; tối ưu hoá; ràng buộc.

### 1. Giới thiệu

Bài toán lập thời khóa biểu là một nhánh của bài toán lập lịch [1-2] trong đó đưa ra một chuỗi các sự kiện (thường là các môn học, bài giảng hoặc môn thi) và bao gồm các giảng viên và học viên trong một khoảng thời gian định trước, thỏa mãn một tập hợp các ràng buộc của từng loại thời khóa biểu khác nhau.

Mỗi bài toán có các tính chất riêng. Khi sắp lịch thi, bài toán đặt ra là phải đáp ứng được yêu cầu về thời gian (như không được trùng hay quá sát nhau) giữa các lần thi liên tiếp của cùng một học sinh, sinh viên. Khi sắp lịch cho trường phổ thông thì ta cần quan tâm giờ rảnh mà giảng viên đăng ký và các tiết trống giữa giờ học của học sinh đóng vai trò rất quan trọng cho việc đánh giá kết quả của thời khóa biểu. Đối với trường đại học, bài toán cần giải quyết cũng là việc tránh xung đột giữa các thành phần tham gia trong thời khóa biểu (giảng viên, lớp học, phòng học và thiết bị). Vì thế, mục tiêu cuối cùng của việc sắp thời khóa biểu là tạo ra một thời khóa biểu với ít xung đột nhất.

Bài toán sắp lịch cho trường đại học có mục tiêu chính là việc sắp các phân công giảng dạy hàng tuần. Bài toán này bao gồm việc sắp các phân công giảng dạy vào các tiết theo một cách nào đó mà giảng viên (hay lớp học) có liên quan không được phép tham gia một lúc hai phân công, và các ràng buộc khác cần phải được thỏa mãn.

Bởi lý do nêu trên, nhu cầu đặt ra là cần một chương trình sắp thời khóa biểu tự động. Trong hơn bốn mươi năm qua, bắt đầu từ thập niên 60, Gotlieb và các nhà khoa học khác đã có nhiều bài báo liên quan đến việc sắp thời khóa biểu tự động đã xuất hiện ở các hội nghị và tạp chí khoa học, và các ứng dụng đã bắt đầu được phát triển cho ra các kết quả khá tốt [3-6].

Các kỹ thuật sơ khai được dựa trên việc giả lập quá trình sắp lịch của con người trong việc giải bài toán [7-10]. Các kỹ thuật này được gọi là *heuristics trực tiếp* (*direct heuristic*) được dựa trên việc *mở rộng liên tục* (*successive augmentation*). Nghĩa là, một phần thời khóa biểu sẽ được sắp, lần lượt từng phân công, cho đến khi tất cả các phân công đã được sắp hoặc không sắp được thêm phân công nào nữa do vi phạm các ràng buộc.

## 2. Phương pháp nghiên cứu

### 2.1. Các phương pháp tìm kiếm

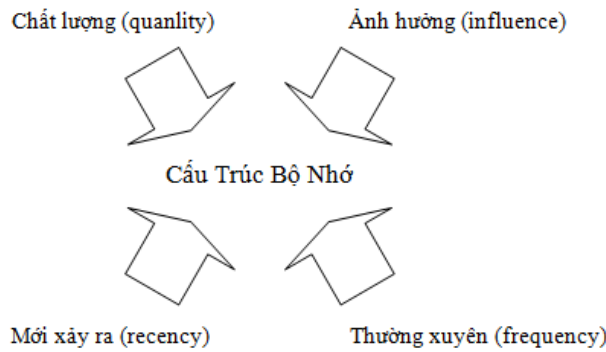
Các phương pháp tìm kiếm cục bộ dựa trên một ý tưởng tổng quát và đơn giản. Trong nghiên cứu này Tabu Search được lựa chọn bởi nó cho phép đi qua các lời giải cục bộ và tiếp tục khám phá không gian tìm kiếm mà không mắc phải vấn đề của việc quay trở lại các lời giải đã xem trước đó. Điều này làm tăng được lời giải tối ưu toàn cục. Gọi  $P$  là một bài toán tối ưu tổ hợp cần giải, và  $s$  là lời giải hiện hành giả sử là một lời giải khả thi của  $P$ , và có hàm chi phí  $f(s)$ . Miền lân cận  $N(s)$  được định nghĩa cho  $s$ , là tập những lời giải láng giềng khả thi  $s'$  của  $s$  sao cho từ  $s$  ta có thể đạt tới  $s'$  nhờ vào một bước chuyển  $m$ . Bước chuyển có tác dụng biến đổi  $s$  thành ra một lời giải láng giềng. Thao tác biến đổi này được lặp cho đến khi hội tụ về một lời giải tốt. Lời giải này là lời giải cận tối ưu, mà trong một số bài toán thực tế, không sai biệt gì nhiều với lời giải tối ưu [11-14].

### 2.2. Giải thuật Tabu

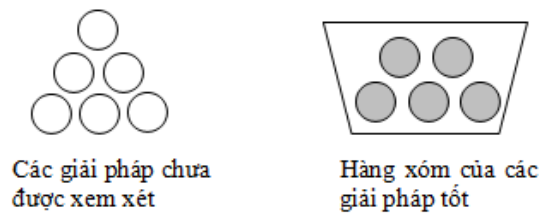
Tabu search [15] dựa trên giả thuyết vấn đề đã được giải, kết hợp chặt chẽ “bộ nhớ thích nghi” (adaptive memory) và “thăm dò phản ứng” (responsive exploration). Giống như việc leo núi, người leo núi phải nhớ có chọn lọc các thành phần của quãng đường đã qua (sử dụng adaptive memory) và lập ra các lựa chọn chiến lược trên đường (sử dụng responsive exploration). Bộ nhớ thích nghi này cho phép việc tìm kiếm trong không gian lời giải một cách tiết kiệm và hiệu quả.

Hình 1 mô tả cấu trúc bộ nhớ trong Tabu search hoạt động bằng việc tham khảo bốn chiều chính sau: tính chất “mới xảy ra” (recency), tính chất “thường xuyên” (frequency), “chất lượng” (quality) và “ảnh hưởng” (influence).

Hai thành phần rất quan trọng của Tabu search là chiến lược tăng cường (Intensification) và chiến lược đa dạng (Diversification) được mô tả như Hình 2. Chiến lược tăng cường dựa trên việc thay đổi các luật lựa chọn để tăng việc kết hợp các phép chuyển và các đặc tính của lời giải tốt trong quá trình tìm kiếm. Vì các lời giải tốt được ghi nhận lại để đánh giá các hàng xóm kề chúng, nên bộ nhớ hiện có liên hệ đến các cài đặt cho chiến lược tăng cường này.



**Hình 1:** Cấu trúc bộ nhớ trong Tabu search



**Hình 2:** Hai thành phần chiến lược Tăng cường và Đa dạng của Tabu search

### 3. Xây dựng mô hình bài toán

#### 3.1. Các khái niệm

- Tuần (*Week*): chu kỳ lặp lại của thời khóa biểu, có thể gồm nhiều tuần.
- Ngày (*Day*): các ngày dạy trong một tuần.
- Tiết học (*Period*): tiết dạy trong ngày.
- Môn học (*Subject*): môn học được giảng dạy trong trường.
- Giảng viên (*Teacher*): giảng viên tham gia vào thời khóa biểu.
- Lớp học (*Class*): lớp học tham gia vào thời khóa biểu.
- Buổi học (*Lesson*): buổi học của một lớp do một giảng viên được phân công dạy môn nào đó.
- Phòng học (*Room*): phòng học để sắp lịch cho các buổi học.
- Tài nguyên (*Resource*): các tài nguyên cung cấp cho thời khóa biểu. Bao gồm: như micro, máy chiếu...
- Phân công dự kiến (*Activity*): tập hợp thông tin nhằm xác định giảng viên dạy môn nào, dạy lớp nào, dạy ở phòng học nào.

Dễ dàng nhận thấy rằng việc xếp thời khóa biểu là việc sắp các buổi học bao gồm các thông tin được thể hiện trong Bảng 1

**Bảng 1:** Thông tin trong buổi học

<b>Biết trước, không cần xếp lịch</b>	<b>Phải xếp lịch</b>
Class	Room
Subject	Week-Day-Period
Teacher	

Trong đó:

- Class: lớp nào
- Subject: học môn gì
- Teacher: ai dạy
- Room: học ở đâu
- Week-Day-Period: học khi nào

Tuy vậy, khi sắp lịch thì người giáo vụ, Trưởng/Phó phòng Đào tạo thường sẽ biết trước, có sẵn thông tin nghiệp vụ rằng “tuần tiếp theo/tháng tiếp theo/học kỳ tiếp theo” thì lớp X sẽ học môn Y và do giảng viên Z giảng dạy. Trong thực tế thì kết quả của việc sắp lịch sẽ giúp cho họ và giảng viên giảng dạy biết lớp mình sẽ phụ trách, giúp cho sinh viên các lớp biết mình sẽ học môn gì, ở đâu, thầy cô nào dạy và mục đích chính là để phục vụ các đối tượng này.

### 3.2. Thiết kế mô hình bài toán

I - là tập các giảng viên (gồm cả giảng viên cơ hữu và giảng viên part time).

J - là tập các lớp học cần xếp lịch.

$J_i (i \in I)$  - là tập các lớp mà giảng viên  $i$  dạy.

K - là tổng số tiết học có thể sắp lịch trong tuần.

C - là tập các phòng học.

$C_j (j \in J)$  - là số phòng học còn trống có thể sắp lịch được cho buổi học  $j$ .

$C_k (k \in K)$  - là số phòng học còn trống có thể sắp lịch được trong tiết học  $k$ .

Ta có biến quyết định :

$$X_{cjk} = \begin{cases} Y > 0 & - \text{phòng } c \text{ có lớp } j \text{ học vào thời gian } k \\ 0 & - \text{phòng } c \text{ không có lớp } j \text{ học vào thời gian } k \end{cases}$$

$(c \in C, j \in J, k \in K)$

Ta có hàm mục tiêu là:

$$\sum_c \sum_j \sum_k X_{cjk}$$

Nhiệm vụ của bài toán đặt ra là xây dựng giải thuật để hàm mục tiêu này đạt MIN.

Trong đó giá trị Y của biến quyết định  $X_{cjk}$  được tính qua các ràng buộc, ta có 2 loại ràng buộc chính để mô tả lớp  $j$  do giảng viên  $i$  dạy vào giờ  $k$  tại phòng học  $c$  nếu không thỏa mãn mỗi:

Ràng buộc cứng: nếu vi phạm 1 tiêu chí + 100 điểm.

Ràng buộc mềm: nếu vi phạm 1 tiêu chí + 1  $\rightarrow$  10 điểm.

#### Các ràng buộc cứng

•  $\sum_c \sum_k X_{cjk} = 1 (j \in J)$  - ràng buộc mô tả mỗi lớp học  $j$  chỉ học vào 1 tiết học  $k$  trong tuần tại 1 phòng học  $c$  cụ thể - Class conflict

•  $X_{cjk} = 0 (c \notin C_k, j \in J_i, k \in K)$  - mô tả việc giáo vụ sẽ không lên lịch cho lớp  $j$  vào tiết học  $k$  nếu hiện phòng học  $c$  đang có lớp học - Period conflict.

•  $X_{cjk} = 0 (c \notin C_j, j \in J_i, k \in K)$  - mô tả sẽ không sắp lịch được cho lớp  $j$  học vào tiết học  $k$  nếu hiện phòng học  $c$  không đủ điều kiện cơ sở vật chất - Room conflict.

•  $X_{cjk} = 0 (c \in C_k | c \in C_j, j \notin J_i, k \in K)$  - mô tả không thể lên lịch cho lớp  $j$  vào tiết học  $k$  tại phòng học  $c$  do giảng viên  $i$  hiện đang bận - Teacher conflict.

### Các ràng buộc mềm

- $S$  - là tập các môn học.
- $S_i (i \in I)$  - là tập các môn mà giảng viên  $i$  có thể dạy.
- $C_j (j \in J)$  - số phòng học còn trống có thể sắp lịch được cho môn học  $j$ .
- $C_k (k \in K)$  - số phòng học còn trống có thể sắp lịch trong tiết học  $k$ .
- $L_i (i \in I)$  - số lượng lớp tối đa mà giảng viên  $i$  có thể dạy được ( $L_i \leq 30$ ).
- $0 \leq \sum_j \sum_k X_{cjk} \leq L_i (i \in I)$  - ràng buộc giảng viên có thể không dạy lớp nào và số lớp dạy không quá  $L_i$  (có thể dạy quá do thiếu giảng viên).
- $\sum_c \sum_j X_{cjk} \leq C_k (k \in K)$  - ràng buộc để kiểm tra với một tiết học  $k$  thì có các phòng học nào còn trống.
- $\sum_c \sum_k X_{cjk} \leq C_j (j \in J)$  - ràng buộc để kiểm tra với một môn học  $j$  thì có các phòng học nào còn trống.
- $\sum_k X_{ijk} \leq \begin{cases} 1 - PTU \\ 4 - NIIT \\ 3 - \text{các khóa ngắn hạn} \end{cases} (i \in I, k \in K)$  - mô tả một giảng viên  $i$  dạy một lớp  $j$  không quá số buổi qui định/tuần (có thể tăng hoặc giảm).

### 3.3. Thiết kế giải thuật

#### Bước 1: Khởi tạo lời giải ban đầu ngẫu nhiên

Khởi tạo ban đầu giúp ta tạo ra một danh sách các buổi học được sắp lịch một cách ngẫu nhiên bằng cách duyệt qua danh sách các lớp, mỗi lớp học một số môn nhất định, mỗi môn có thể học một vài buổi. Do đó ta sẽ duyệt từng buổi học/từng môn/từng lớp. Quy tắc khởi tạo giá trị ban đầu hoặc ngẫu nhiên cho lớp học này như sau:

Buổi học:

Lớp: lớp hiện đang duyệt.

Môn học: môn hiện đang xét.

Giờ học, ngày học, phòng học, giảng viên ta sẽ chọn ngẫu nhiên.

Việc sinh ra lời giải ban đầu ngẫu nhiên này có thể vi phạm nhiều ràng buộc, thậm chí không thỏa mãn một số ràng buộc cứng - hard constraints - nhưng lưu ý rằng đây chỉ là lời giải ban đầu mà từ đó ta sẽ dùng giải thuật tìm kiếm Tabu để tìm ra phương án sắp xếp tối ưu thỏa mãn tất cả các ràng buộc cứng và cố gắng giảm tối đa số lượng vi phạm các ràng buộc mềm - soft constraints.

#### Bước 2: Cải thiện chất lượng lời giải tìm được ở bước 1 bằng giải thuật tìm kiếm

##### Tabu

#### Tính hàm mục tiêu đánh giá chất lượng của lời giải

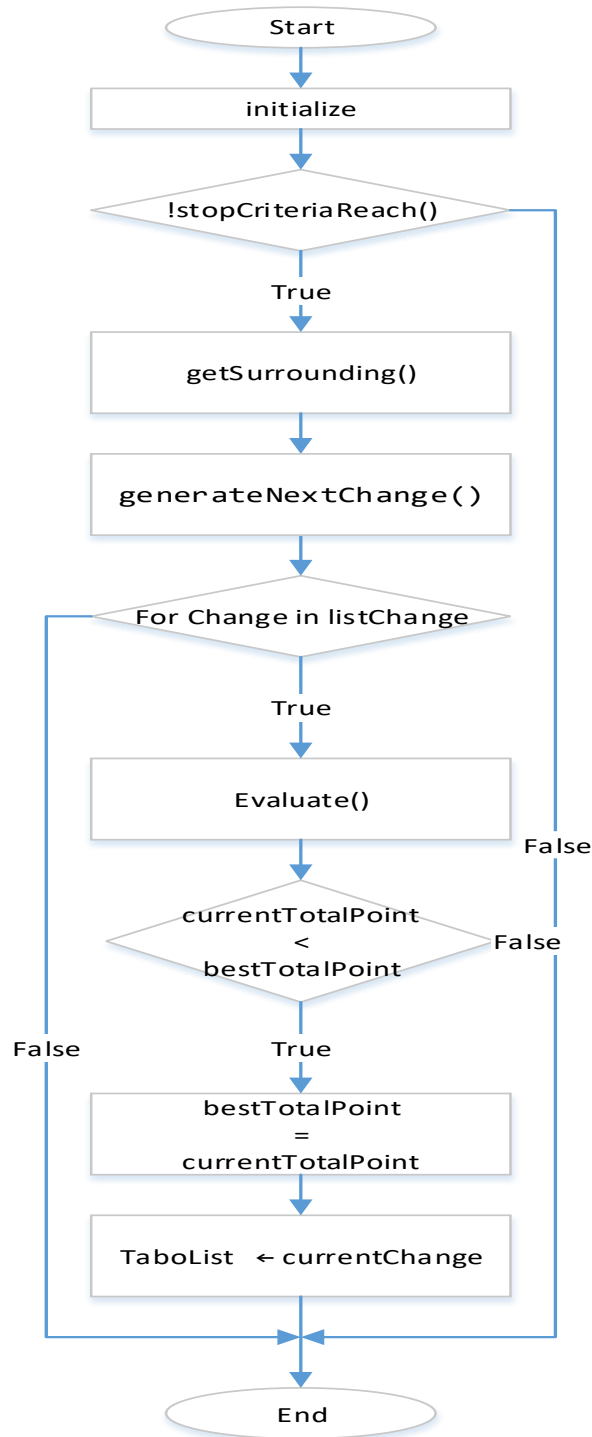
Với mỗi biến quyết định  $X_{cjk}$  như đã đề cập trong mô hình bài toán

$$\sum_c \sum_j \sum_k X_{cjk}$$

Hàm mục tiêu đánh giá chất lượng của lời giải dựa trên hai yếu tố:

- Vi phạm mỗi ràng buộc cứng: + 100 điểm.
- Vi phạm mỗi ràng buộc mềm: + 1 → 10 điểm.

Sơ đồ mô tả luồng xử lý khi ta cài đặt giải thuật tìm kiếm Tabu cho mô hình bài toán sắp thời khóa biểu đã được đề cập từ đầu đến nay trong bài báo được thể hiện trong Hình 3.



**Hình 3:** Sơ đồ cài đặt giải thuật

**Hàm mong đợi (Aspiration Function)**

Xét một phép chuyển m:

- Gọi Q là lời giải hiện tại

- $Q'$  là lời giải mới tạo thành sau khi thực hiện phép chuyển m lên  $Q'$
- $Q^*$  là lời giải tốt nhất hiện tại
- $f(Q)$  là hàm mục tiêu
- Phép chuyển m được gọi là thỏa hàm mong đợi chỉ khi  $f(Q) < f(Q^*)$

**Điều kiện dừng của thuật toán**

Thuật toán dừng khi một trong các điều kiện sau đây được thỏa mãn, hàm stopCriteriaReach:

- Sau 500 bước lặp.
- Hàm mục tiêu của lời giải tốt nhất hiện tại đạt  $MIN = 0$  (đã thỏa mãn mọi ràng buộc cứng và mềm).
- Sau một số bước lặp nhất định (ví dụ 100 bước lặp) liên tiếp mà chất lượng lời giải tốt nhất hiện tại không đổi.

**4. Kết quả thực nghiệm**

**Định dạng tập tin dữ liệu CSV đầu vào:**

Mô hình hệ thống phần mềm được xây dựng nhằm giúp đọc các file dữ liệu đầu vào theo định dạng CSV một cách dễ dàng. Do đó hệ thống đã xây dựng sẵn các class dưới đây để có thể mô hình hóa nghiệp vụ sắp xếp thời khóa biểu, từ đó dễ dàng cài đặt giải thuật tìm kiếm Tabu cho hệ thống.

**Bảng 2:** Bảng mô tả ánh xạ tập dữ liệu và mô hình hệ thống

Model	CSV
Model name	[0]
Name	[1]
ID	[2]
<i>OtherModel</i>	[3]
[ <i>OtherModel</i> <i>Constraint</i> ]	[4]
...	...

**Bảng 3:** Ví dụ ánh xạ từ tập dữ liệu vào mô hình hệ thống

Model	CSV
Model name: hour	[0]: hour
Name: 9h45-11h45	[1]: 9h45-11h45
ID: 8	[2]: 8

Dựa theo định dạng cấu trúc của Bảng 2 và Bảng 3, nhóm tác giả đã thực hiện thực nghiệm giải thuật với bộ dữ liệu gồm 7 phòng học; 25 lớp; 500 vòng lặp; 17 giảng viên; 28 môn học; 109 buổi học. 10 lần chạy thử nghiệm trên phần mềm đã được thực hiện. Kết quả thống kê các lần thực nghiệm được thể hiện trong Bảng 4, kết quả thực nghiệm lần chạy thứ 8 và 9 được thể hiện trên Hình 4. Từ kết quả thực nghiệm trên có thể thấy rằng, giải thuật tìm kiếm đạt điểm tối ưu toàn cục khi số lần lặp trong ngưỡng khoảng 350-500.

Điểm lời giải của các thực nghiệm đều cho kết quả tốt khi điểm cao khoảng 1400 điểm. Vì vậy giải pháp Tabu search này là khả thi cho bài toán lập lịch biểu cho trường đại học.



**Hình 4:** Kết quả thực nghiệm trên phần mềm



**Bảng 4: Kết quả thống kê các lần thực nghiệm giải thuật**

STT	Lời giải tốt nhất (điểm)	Tại vòng lặp số	Hết thời gian (giây)
1	1397	348	77,85
2	1414	491	87,08
3	1395	447	99,36
4	1368	430	82,62
5	1380	448	129,77
6	1434	500	82,79
7	1392	437	95,35
8	1349	500	92,41
9	1359	412	84,02
10	1337	384	83,67

## 5. Kết luận

Bài báo đã trình bày kết quả nghiên cứu về bài toán lập lịch biểu cho trường đại học và áp dụng giải thuật Tabu để giải quyết vấn đề này. Kết quả thực nghiệm đã cho thấy rằng phương pháp tiếp cận mục tiêu và giải thuật Tabu có thể tạo ra lời giải tối ưu với độ chính xác cao và tiết kiệm thời gian tính toán. Tuy nhiên, còn nhiều thách thức và hạn chế trong quá trình áp dụng giải thuật này, như số lượng lớn các ràng buộc cần tuân thủ, việc xác định tham số phù hợp và sự phụ thuộc vào kinh nghiệm của người lập trình. Do đó, việc phát triển các giải thuật và phương pháp tiếp cận mới để giải quyết vấn đề lập lịch biểu cho trường đại học là cần thiết và đầy thách thức trong tương lai.

**Lời cảm ơn:** Kết quả nghiên cứu này là sản phẩm của đề tài nghiên cứu khoa học cấp Cơ sở có mã số T2022-07-06, được tài trợ bởi kinh phí của Trường Đại học Công nghệ thông tin và Truyền thông - Đại học Thái Nguyên

## TÀI LIỆU THAM KHẢO

- [1] Schaerf, A *survey of automated timetabling*, Technical Report CSR9567, CWI, Amsterdam, The Netherlands, 1995.
- [2] Carter, M. W., And G. Laporte, "Recent developments in practical examination timetabling," In: *E.K. Burke and P. Ross (eds.), Practice and Theory of Automated Timetabling, First International Conference, Selected papers, Lecture Notes in Computer Science 1153*, Springer, 1996, 3-21. DOI: [10.1007/3-540-61794-9\\_49](https://doi.org/10.1007/3-540-61794-9_49)
- [3] Gotlieb, C. C., "The construction of class-teacher timetables," In *Popplewell, C. M. (Ed.), IFIP congress 62, (73-77)*, North-Holland, 1963.

- [4] Schmidt, G., Strohlein, T., “Timetable construction - an annotated bibliography,” *The Computer Journal*, 23(4), (307-316), 1979. DOI: [10.1093/comjnl/23.4.307](https://doi.org/10.1093/comjnl/23.4.307)
- [5] Junginger, W. *Timetabling in Germany - a survey*. Interfaces, (16, 66-74), 1986. DOI: [10.1287/inte.16.4.66](https://doi.org/10.1287/inte.16.4.66)
- [6] Tripathy, A., “School timetabling - A case in large binary integer linear programming,” *Management Science*, 30 (12), (1473-1489), 1984. DOI: [10.1287/mnsc.30.12.1473](https://doi.org/10.1287/mnsc.30.12.1473)
- [7] Schmidt, G., Strohlein, T., “Timetable construction - an annotated bibliography,” *The Computer Journal*, 23(4), (307-316), 1979. DOI: [10.1093/comjnl/23.4.307](https://doi.org/10.1093/comjnl/23.4.307)
- [8] Zbigniew Michalewicz and David B. Fogel, *How to Solve It: Modern Heuristics*, Springer, 2000. DOI: [10.1007/978-3-662-04131-4](https://doi.org/10.1007/978-3-662-04131-4)
- [9] Steven Minton, Mark D. Johnston, Andrew B. Philips and Philip Laird, “Minimizing conflicts: a heuristic repair method for constraint satisfaction and scheduling problems,” *Artificial Intelligence*, 58:161-205, 1992. DOI: [10.1016/0004-3702\(92\)90007-K](https://doi.org/10.1016/0004-3702(92)90007-K)
- [10] Costa, D., “A tabu search algorithm for computing an operational timetable,” *European Journal of Operational Research*, 76, (98-110), 1994. DOI: [10.1016/0377-2217\(94\)90009-4](https://doi.org/10.1016/0377-2217(94)90009-4)
- [11] Tripathy, A., “Computerised decision aid for timetabling - A case analysis,” *Discrete Applied Mathematics*, 35(3), (313-323), 1992. DOI: [10.1016/0166-218X\(92\)90253-7](https://doi.org/10.1016/0166-218X(92)90253-7)
- [12] Fred Glover, Manuel Laguna, *TabuSearch*, Kluwer Academic Publishers, United States of America, 1998.
- [13] Goldberg, David, *Genetic Algorithms in Search, Optimization and Machine Learning*. Reading, MA: Addison-Wesley Professional, 1989. ISBN: 978-0201157673.
- [14] Arabinda Tripathy, *School Timetabling - A Case In Large Binary Integer Linear Programming*, Management Science, vol. 30. no. 12. 12/1984. DOI: [10.1287/mnsc.30.12.1473](https://doi.org/10.1287/mnsc.30.12.1473)
- [15] Bardadym, V. A., “Computer-aided school and university timetabling: The new wave?,” in: *E.K. Burke and P. Ross (eds.), Practice and Theory of Automated Timetabling, First International Conference, Selected papers, Lecture Notes in Computer Science* 1153, Springer, 1996, 22–45. DOI: [10.1007/3-540-61794-9\\_50](https://doi.org/10.1007/3-540-61794-9_50)

**ABSTRACT**

**STUDY ON SCHEDULING PROBLEM  
TOWARDS TARGETED OUTREACH,  
APPLIED TO UNIVERSITY SCHEDULING**

**Tran Hai Thanh, Nguyen Lan Oanh, Nguyen Thu Phuong, Nguyen Thi Duyen**

*Thai Nguyen University of Information and Communication Technology,*

*Thai Nguyen University, Thai Nguyen, Vietnam*

Received on 15/5/2023, accepted for publication on 27/6/2023

This paper focuses on researching, analyzing, and evaluating universities scheduling problem which requires scheduling multiple classes, different teachers and classrooms, along with various constraints such as teacher's teaching hours, classroom capacity, and other resources. In order to solve this problem, Tabu search algorithm has been proposed, which based on searching for the best solutions for adaptive memory and reactive exploration. The constraints and limitations of the problem will be used to create a list of Tabu conditions to ensure that non-optimal solutions will not be repeated. After initializing and updating the adaptive memory, the Tabu search algorithm will search for the next solutions based on the Tabu conditions and adaptive memory. In this way, the Tabu search algorithm can find optimal solutions for this scheduling problem for universities. However, there are many challenges and limitations in the application of this algorithm, such as the large number of constraints to be follow, the determination of suitable parameters, and the dependence on the programmer's experience.

**Keywords:** Scheduling; university; Tabu search; optimization; constraints.