

# HỆ THỐNG RA QUYẾT ĐỊNH SẮP THỜI KHOÁ BIỂU THÔNG MINH<sup>1</sup>

TS. Quản Thành Thơ và TS. Nguyễn Tuấn Anh  
Trường Cao đẳng Nguyễn Tất Thành

Sắp xếp thời khoá giảng dạy hợp lý và hiệu quả luôn là một yêu cầu cấp thiết của tất cả các cơ sở đào tạo từ tiểu học, trung học cho đến cao đẳng, đại học. Trước đây, công việc này được làm bằng tay, do đó tốn rất nhiều chi phí nhân lực và thời gian, đồng thời hiệu quả tối ưu cũng còn bị hạn chế. Ngày nay, đã có nhiều chương trình ứng dụng xếp thời khoá biểu được giới thiệu, phần lớn là từ nước ngoài, vì vậy có chi phí cao và chưa thật phù hợp với yêu cầu với môi trường Việt Nam.

Hiện nay Trường Cao đẳng Nguyễn Tất Thành đang sử dụng chương trình quản lý đào tạo Edusoft. Mặc dù Edusoft đã bao phủ gần như đầy đủ các yêu cầu quản lý học vụ, nhưng quy trình sắp xếp thời khoá biểu của chương trình này vẫn cần được bổ sung thêm các chức năng để nâng cao tính tùy biến theo yêu cầu của Trường. Đây cũng là một nhu cầu chung thường gặp cho tất cả các đơn vị đào tạo khi sử dụng một tiện ích được xây dựng sẵn cho vấn đề sắp xếp thời khoá biểu.

Đáp ứng nhu cầu trên, Trường Cao đẳng Nguyễn Tất Thành đã phát triển một đề tài nghiên cứu xây dựng một hệ thống ra quyết định sắp thời khoá biểu thông minh, thông qua đó các đóng góp khoa học sau đã được hình thành: (i) Một module xếp thời khoá biểu thông minh dựa trên giải thuật Tabu và (ii) Một kiến trúc cho phép tích hợp module trên vào bất kỳ hệ thống có sẵn nào. Các đóng góp lý thuyết trên đã được hiện thực và bắt đầu đưa vào ứng dụng thực tế tại Trường.

## 1. GIỚI THIỆU

Vấn đề xây dựng Thời khoá biểu tự động hoặc bán tự động cho các loại công việc khác nhau, bao gồm cả các hoạt động dạy và học trong một cơ quan đào tạo là một vấn đề cấp thiết đã thu hút nhiều sự chú ý trong cả hai lĩnh vực nghiên cứu và ứng dụng trong thời gian qua. Tuy nhiên, các kết quả đạt được trong lĩnh vực này hiện nay vẫn chưa có được một ứng dụng rộng rãi do các nguyên nhân sau đây:

- Về mặt học thuật, bài toán sắp thời khoá biểu tự động là một bài toán có độ phức tạp giải thuật rất cao. Hiện nay vẫn chưa tồn tại giải thuật sắp thời khoá biểu tổng quát có thời gian chạy chấp nhận được trong môi trường thực tế.

- Khi áp dụng vào các đơn vị đào tạo, do mỗi đơn vị đào tạo sẽ có mô hình đào tạo, đặc thù đào tạo khác nhau, dẫn đến sự các mối quan hệ ràng buộc khác nhau. Việc đưa ra một mô hình biểu diễn đặc trưng tổng quát cho tất cả các mối quan hệ ràng buộc này là rất khó khăn.

- Khi triển khai, do mỗi đơn vị đào tạo có một hệ thống thông tin quản lý riêng sử dụng nhiều công nghệ khác nhau, việc đưa ra một giải pháp công nghệ để tích hợp module xử lý Thời Khoa Biểu vào hệ thống thông tin sẵn có của mỗi đơn vị đào tạo sao cho ít gây nên xáo trộn về quy trình và phương thức làm việc nhất cũng là một thách thức không nhỏ.

Mặt khác, hiện nay Trường Cao đẳng Nguyễn Tất Thành đang

triển khai phần mềm quản lý đào tạo Edusoft cho công tác quản lý đào tạo của mình.

Tuy phần mềm Edusoft hoạt động ổn định và đáp ứng các nhu cầu quản lý của trường, việc sử dụng phần mềm trên vẫn gặp các khó khăn sau:

- Cách đưa kế hoạch, sắp xếp Thời khoá biểu của Edusoft vẫn còn nhiều điểm chưa phù hợp với quy trình và mô hình đào tạo của Trường.

- Giao diện của Edusoft, cụ thể cho phần quản lý Thời Khoa Biểu, chưa thật thuận tiện và dễ sử dụng cho người dùng bình thường.

Các phân tích cho thấy việc phát triển một module xử lý Thời Khoa Biểu tích hợp vào phần mềm Edusoft hiện hành là một

nhu cầu cấp bách hiện nay trong việc quản lý điện tử các quy trình đào tạo ở Trường Cao đẳng Nguyễn Tất Thành. Đây cũng là một nhu cầu thường gặp tại các cơ sở đào tạo khi sử dụng một tiện ích sắp thời khóa biểu xây dựng sẵn. Vượt trên các nhu cầu sắp thời khóa biểu cơ bản, yêu cầu từ các cơ sở đào tạo ngày nay đang hướng đến một hệ ra quyết định đáp ứng các yêu cầu sau:

- Có khả năng import/export dữ liệu một cách tự động và tích hợp hoàn toàn vào quy trình vận hành của hệ thống đang dùng.
- Có khả năng đề nghị một thời khóa biểu hợp lý và linh hoạt.
- Có giao diện thuận tiện để dùng cho các thao tác xem, chỉnh sửa, xử lý dữ liệu thời khóa biểu.

Trong bài báo này, chúng tôi trình bày giải pháp cho các vấn đề trên. Cụ thể, trong phần 2, chúng tôi đề nghị một kiến trúc để tích hợp hệ ra quyết định vào một hệ thống quản lý đào tạo có sẵn. Phần 3 trình bày một giải pháp sắp thời khóa biểu thông minh dựa trên giải thuật Tabu. Phần 4 trình bày các cơ chế giao tiếp với người dùng. Phần 5 sẽ trình bày một số kết luận và hướng phát triển cho nghiên cứu này.

## 2. KIẾN TRÚC TÍCH HỢP HỆ THỐNG

Kiến trúc tích hợp của hệ thống được trình bày trong Hình 1. Để thuận tiện theo dõi, chúng tôi trình bày kiến trúc này trong ngữ cảnh thực tế của Trường Cao đẳng Nguyễn Tất Thành đang sử dụng hệ thống quản lý Edusoft. Tuy nhiên, kiến trúc trên có thể điều chỉnh dễ dàng để sử dụng cho tất cả các cơ sở đào tạo đang dùng các hệ thống quản lý khác.

Để thực hiện việc tích hợp, chương trình xếp Thời khóa biểu có khả năng nhập dữ liệu từ các tập tin định dạng csv sau đó thực hiện các bước xếp thời khóa biểu và có thể xuất dữ liệu thời khóa biểu đã xếp trở về dạng csv. Do đó chương trình hoạt động hoàn toàn độc lập với các chương trình quản lý có sẵn của các trường, dữ liệu của các trường chỉ cần chuyển đổi ra định dạng csv và nhập vào chương trình xếp thời khóa biểu. Sau khi xếp thành công, dữ liệu xuất ra csv và chuyển đổi vào dạng phù hợp của ứng dụng sẵn có trong trường.

## 3. GIẢI THUẬT XẾP THỜI KHOÁ BIỂU

### 3.1 Tóm lược

Việc giải bài toán xếp Thời khóa biểu (TKB) sẽ qua 2 bước chính như sau:

- Bước 1: Tìm một TKB khả thi (thỏa mãn tất cả ràng buộc cứng).
- Bước 2: Cải tiến TKB (cố gắng thỏa mãn tối đa các ràng buộc mềm).

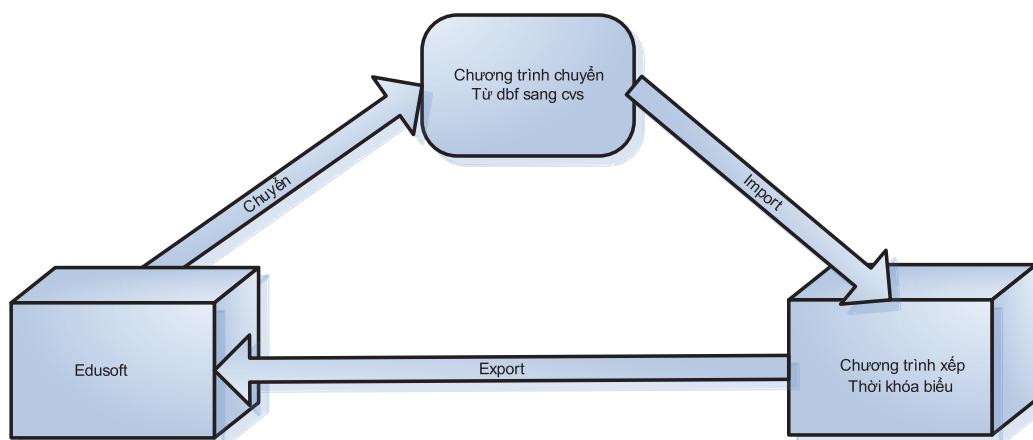
Để đơn giản không gian tính toán, tách biệt việc xét các ràng buộc cho các đối tượng: ghép giờ học với tiết, ghép phòng, và ghép giảng viên

### 3.2 Định nghĩa bài toán

Chúng ta có một số định nghĩa cơ bản như sau:

- Một meeting được định nghĩa là một giờ học của một lớp nào đó.
- Giả sử, mỗi lớp môn học trong một tuần có một hoặc nhiều meeting.
- Hai meeting được gọi là đụng độ nếu hai meeting này đều thuộc về một lớp môn học và chúng overlap nhau về mặt thời gian.
- Cluster đụng độ: tất cả các meeting trong TKB của một lớp tạo nên một đồ thị đụng độ (gọi là cluster đụng độ). Mỗi meeting trong này đều ko thể có thời gian overlap nhau.
- Multiple-conflict-cluster meeting: là meeting cùng thuộc về hai hay nhiều hơn conflict clusters (xảy ra ở trường hợp 1 nhóm lớp môn học được tạo từ hai lớp khác nhau).

Từ các định nghĩa trên bài toán xếp thời khóa biểu được đưa về dạng bài toán thỏa mãn điều kiện (Constraint Satisfaction Problem – CSP). Một CSP bao gồm ba thành phần (Variable, Domain, Constraint) trong đó V1-N: mỗi variable là một meeting, D1-N: domain cho mỗi biến là tích đề các: TimeSlots (tiết học) x Rooms (phòng).



Hình 1 – Kiến trúc tích hợp của hệ thống.

### 3.3 Định nghĩa các ràng buộc

#### Ràng buộc trên giờ học

Ràng buộc cứng	Ràng buộc mềm
<p><b>C1:</b> Các meetings đụng độ không được overlap thời gian với nhau.</p> <p><b>C2:</b> Một meeting chỉ được phép kéo dài trong hoặc buổi sáng, hoặc buổi chiều, hoặc buổi tối, không có meeting nào được kéo dài trong 2 buổi.</p> <p><b>C3:</b> Một số meeting không được xếp vào một khoảng timeslot nào đó (khoảng thời gian sinh viên không thể học).</p>	<p><b>M1:</b> Trong TKB của mỗi lớp, hạn chế tối đa số ngày vừa học sáng vừa học chiều. (đã thực hiện trong ràng buộc cứng khi xét Khối_buổi thích hợp).</p> <p><b>M2:</b> Không nên có tiết trống giữa hai meeting trong cùng một buổi đối với mỗi lớp (số tiết trống tối đa giữa hai lớp học trong ngày).</p> <p><b>M3:</b> Các meeting của cùng một môn học nên được rải đều trong tuần (Không cho phép hai cụm tiết của một môn học xếp trong cùng một ngày, Không cho phép hai cụm tiết của một môn học xếp trong cùng một buổi).</p> <p><b>M4:</b> Tập thời gian bắt đầu thích hợp cho một số môn học (thí dụ: Thể dục nên được sắp vào các tiết đầu buổi sáng hoặc cuối buổi chiều).</p>

#### Ràng buộc trên phòng học

Ràng buộc cứng	Ràng buộc mềm
<p><b>C4:</b> Tại một thời điểm, không thể có nhiều hơn một meeting được tổ chức tại một phòng.</p> <p><b>C5:</b> Meeting của một lớp chỉ có thể được gán vào một phòng khi sức chứa của phòng đó lớn hơn hoặc bằng số số lớp.</p> <p><b>C6:</b> Một số meeting yêu cầu một số phòng học đặc biệt (như giờ thực hành, thực tập).</p>	<p><b>M5:</b> Các meetings trong cùng buổi của một lớp nên được xếp gần nhau (tốt nhất nên cùng một phòng)</p> <p><b>M6:</b> Độ căng của phòng (mật độ sử dụng của phòng) nên là cao nhất có thể.</p>

#### Ràng buộc trên giảng viên

Ràng buộc cứng	Ràng buộc mềm
<p><b>C8:</b> Một giảng viên không thể ở hai meeting khác nhau tại cùng một thời điểm.</p> <p><b>C9:</b> Một giảng viên không thể ở hai room khác nhau tại cùng một thời điểm.</p> <p><b>C10:</b> Một giảng viên không thể dạy ở một thời khoảng không sẵn sàng định trước.</p>	<p><b>M5:</b> Các meetings trong cùng buổi của một lớp nên được xếp gần nhau (tốt nhất nên cùng một phòng)</p> <p><b>M6:</b> Độ căng của phòng (mật độ sử dụng của phòng) nên là cao nhất có thể.</p>

### 3.4 Thuật giải

#### Giải bài toán với ràng buộc cứng

- Trọng số khả thi (Distance to feasibility) = Số ràng buộc cứng bị vi phạm

- Giải thuật chung cho việc xếp với ràng buộc cứng (Backtracking-free Constructive method with Local Search and Look-Ahead):

1. Select an unassigned variable v
2. Select a value to assign to v
3. if the assignment causes a conflict then  
    perform local search  
    if the local search is not successful then  
        goto step 5
4. if termination condition is false then  
    go to step 1
5. stop

### *Giải bài toán với ràng buộc mềm*

- Mục đích: Cải tiến TKB để thỏa mãn ràng buộc mềm từ một TKB “khả thi” (thỏa tất cả ràng buộc cứng).
- Sử dụng giải thuật Local Repair – Tabu Mechanism: giải thuật dựa trên nguyên lý leo đồi và thoát khỏi tối ưu cục bộ bằng cách dùng bộ nhớ tạm thời – tabu list.

- Giải thuật được mô tả như sau:

```

1. start with some initial timetable s. Set best = s
2. scan all the stable-classes to select the stable-class k of which
timetable is with worst objective value.
3. if k is in tabu list then
    ignore k in the next scan of the list of stable-classes
and go to Step 2
4. repair(k, s') // repair the stable-class k to get a new timetable s'
5. update tabu list with k
6. if f(s') is better than f(s) then
    if f(s') is better than f(best) then
        set best = s'
        set s = s'
    else
        ignore k in the next scan of the list of stable-classes
and goto Step 2
7. if a stopping condition is met then
    stops.
else
    go to Step 2.

```

- Thủ tục repair(k, s') bao gồm các thao tác sau:

- List các ràng buộc mềm bị vi phạm trong TKB của lớp k, gọi nó là SC(k)
- Với mỗi ràng buộc c trong SC(k)
  - Thực hiện repair\_constraint(c, varlist(c)), với varlist(c) là danh sách tất cả các meeting bao gồm trong c:

```

/* repair các biến trong varlist để thỏa mãn nhiều ràng buộc c nhất */
procedure repair_constraint(c, varlist)
begin
    if varlist = <v_last> then // chỉ còn meeting cuối
        for each free timeslot of the meeting v_last do
            find suitable room at this timeslot for
the meeting v_last;
ing v_last;
been achieved
else
    let v_first = head(varlist)
    for each free timeslot of the meeting v_first do
        find suitable room at this timeslot for
the meeting v_first;
ing v_first
        assign <timeslot, room> to the meet-
        repair_constraint(c, tail(varlist))
    endfor
endif
end

```

▪ Nhằm giới hạn không gian tính toán trong local repair, ta chỉ quan tâm đến những meeting vi phạm ràng buộc c trong những lớp có “worst objective value”

• Sử dụng một tabu list để tránh tính lại những lớp đã áp dụng repair trước đó. Kích thước của tabu list không nên vượt quá  $2/3^*$  (tổng số lớp).

- Hàm đánh giá:

$$\begin{aligned}
F(cl) = & a1 * \text{all\_day} + a2 * \text{hole} + \\
& a4 * \text{scatter\_meeting} \\
& + a3 * \text{room\_samesession} + a5 * \text{subject\_suitabletime}
\end{aligned}$$

• a1, .., a5 là các hệ số được quyết định bởi người dùng (lập trình viên)

•  $\text{all\_day} = 4^n + 2^{lp-6l}$

▪ n: tổng số ngày có các meetings vừa ở buổi sáng và chiều

▪ p: tổng số timeslots của các meetings trên

• hole: tổng timeslot trống giữa 2 meeting trong cùng buổi sáng/chiều.

• scatter\_meeting: tổng số meetings được xếp vi phạm ràng buộc Si

▪ Si: Trong TKB của mỗi sv, số buổi học và buổi nghỉ nên được cân bằng để sv có thời gian tự học ở nhà

• room\_samesession: các meetings của cùng lớp được xếp trong một buổi nằm khác phòng.

• subject\_suitabletime: các meetings có môn học được gán tiết thích hợp nhưng không được xếp vào đúng tiết đó.

## 4. CÁC CHỨC NĂNG CỦA CHƯƠNG TRÌNH

Với mục tiêu nâng cao tính tiện dụng cho người sử dụng, chương trình hỗ trợ ra quyết định xếp thời khóa biểu cung cấp các tính năng sau:

- Cho phép xét thời gian giảng dạy thích hợp cho giảng viên,

tính năng này dùng để xét thời gian dạy của các giảng viên mà trường thỉnh giảng và có giờ dạy cố định trước.

- Hỗ trợ người sử dụng có thể xếp thời khóa biểu bằng tay theo mong muốn và chương trình sẽ kiểm tra lại để loại bỏ những vi phạm mà khi người dùng xếp bằng tay không kiểm soát được.

- Hỗ trợ giao diện kéo thả các cụm tiết học sau khi đã được xếp thời khóa biểu để thay đổi giờ học theo mong muốn của người dùng.

- Tối ưu việc sử dụng phòng học.

Hệ thống các chức năng được phát triển theo như mô tả trong Hình 2.

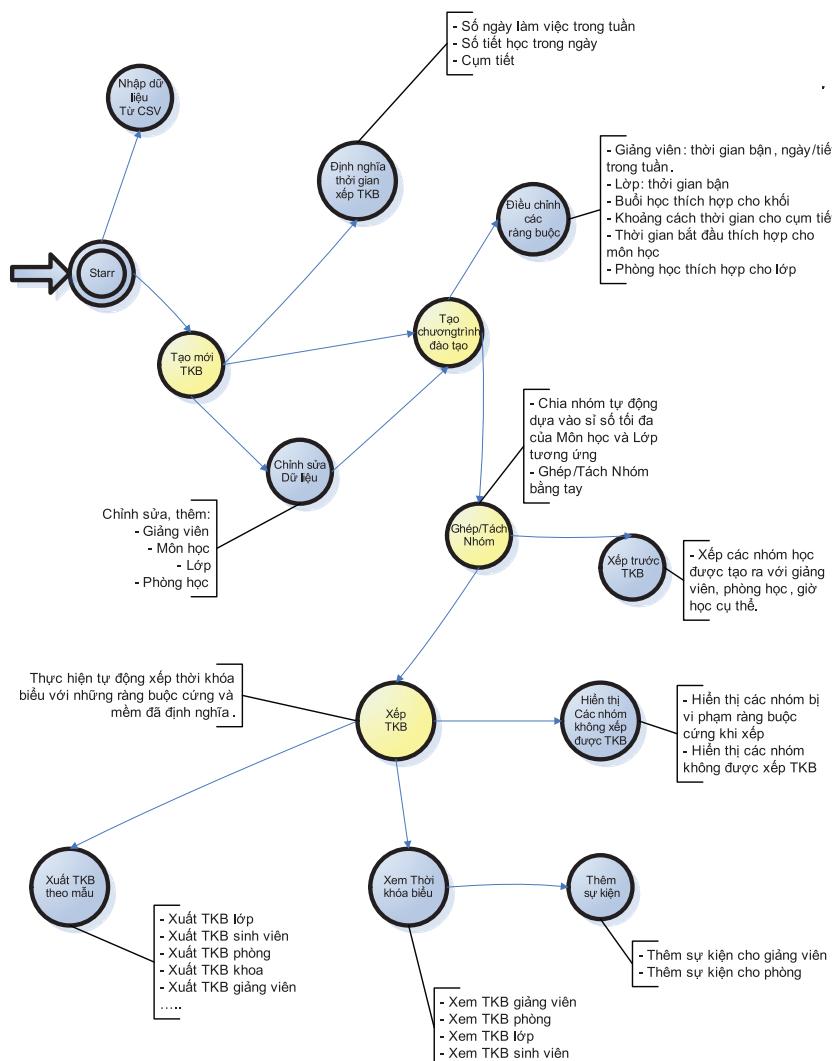
## 5. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

Trong bài báo này, chúng tôi đã trình bày một hệ thống hỗ trợ ra quyết định sắp xếp thời khóa biểu một cách thông minh. Mặc dù được trình bày trong ngữ cảnh thực tế của Trường Cao đẳng Nguyễn Tất Thành, giải pháp đề nghị trong hệ thống này có thể áp dụng rộng rãi cho nhiều cơ sở đào tạo khác nhau.

Sau giai đoạn thử nghiệm với kết quả tốt, trong giai đoạn tiếp theo, chúng tôi sẽ bước đầu đưa vào sử dụng hệ thống trong các Khoa của Trường. Kết quả đóng góp từ các Khoa sẽ được sử dụng để hoàn chỉnh chương trình, qua đó có thể giới thiệu chương trình sử dụng rộng rãi, góp phần vào sự phát triển chung của các cơ sở đào tạo giáo dục tại Việt Nam.

### LỜI CẢM ƠN

Chúng tôi xin cảm ơn Trường Cao Đẳng Nguyễn Tất Thành đã hỗ trợ tài chính để hoàn thành nghiên cứu này.



Hình 2 – Hệ thống chức năng của chương trình

### TÀI LIỆU THAM KHẢO

[1] V. H. Tam, N. Q. V. Hung and D. T. Anh, "Generating Complete University Course Timetables by Using Local Search Methods", IEEE, 2006.

[2] A. Schaerf, "Combining local search and look-ahead for scheduling and Constraint Satisfaction Problem", IJCAI'97, 1997.

[3] S. Milton, M.D. Johnston, A.B. Philips and P. Laird, "Minimizing Conflicts: a Heuristic Repair Method for Constraint Satisfaction and Scheduling Problems", Artificial Intelligence, vol. 58, 1992.

[4] J. Thornton, "Constraint Weighting Local Search for Constraint Satisfaction", Ph.D. Dissertation, School of Computing and Information Technology, Griffith University, Australia, 2000.

<sup>1</sup> Công việc trình bày trong nghiên cứu này là kết quả của đề tài nghiên cứu khoa học cấp Trường có mã số T-NCKH-2009-1 của Trường Cao đẳng Nguyễn Tất Thành