

## TIẾP CẬN LƯỜNG CỰC ĐẠI TRONG MẠNG CHO BÀI TOÁN XẾP LỊCH BIỂU

Phạm Nguyên Khang<sup>1</sup>, Võ Trí Thức<sup>1</sup>, Nguyễn Bá Diệp<sup>1</sup> và Bùi Lê Diễm<sup>2</sup>

<sup>1</sup> Khoa Công nghệ Thông tin & Truyền thông, Trường Đại học Cần Thơ

<sup>2</sup> Khoa Sư phạm, Trường Đại học Cần Thơ

### Thông tin chung:

Ngày nhận: 03/09/2013

Ngày chấp nhận: 21/10/2013

### Title:

Maximum flow network approach for scheduling problem

### Từ khóa:

Xếp lịch biểu, luồng cực đại, chi phí cực tiểu, tối ưu hoá

### Keywords:

Scheduling problem, maximum flow, minimum cost, optimization

### ABSTRACT

*In this paper, we present a solution for the scheduling problem of practical courses in Can Tho University using Maximum Flow Network approach. The scheduling problem for practical courses is concerned to assign students to groups/practical rooms under some constraints such as room capacity, teachers' schedule, etc. The problem attempts to optimize the performance criteria and distribute the students fairly to rooms depending on the ratio of room capacity and the number of students enrolled. We model the scheduling problem as a maximum flow problem where each student is modeled as a source node and each room as a sink node. Capacity constraints are represented as maximum capacity of edges connecting source nodes and sink nodes. We have also added constraints on minimum number of students assigned to a room as minimum capacity of edges. The approach is implemented on Google computing platform using Google Apps Script. A case study is experimented by enrollment data of practical courses in Can Tho University. Results show that our solution is tractable.*

### TÓM TẮT

*Trong bài viết này, chúng tôi trình bày một giải pháp cho bài toán xếp lịch các học phần thực hành tại các trường đại học sử dụng mô hình luồng cực đại trong mạng. Bài toán xếp lịch thực hành là một dạng của bài toán xếp thời khoá biểu tổng quát trong đó liên quan đến việc phân các sinh viên vào các nhóm/phòng thực hành sao cho thoả mãn các ràng buộc về lịch rảnh của sinh viên, giảng viên, sức chứa của phòng và quan trọng nhất là khai thác tối đa hiệu suất sử dụng của các phòng thực hành. Với các ràng buộc như trên, bài toán xếp lịch thực hành có thể được mô hình hoá về bài toán tìm luồng cực đại trong mạng với độ phức tạp giải thuật đa thức. Mỗi sinh viên là một đỉnh phát, mỗi phòng thực hành là một đỉnh thu. Sức chứa của phòng có thể được ràng buộc bằng khả năng thông qua của cung tương ứng. Ta cũng có thể thêm vào ràng buộc số lượng sinh viên tối thiểu cho một phòng bằng các cách đặt cận dưới cho các cung. Vấn đề tối ưu tỉ lệ giữa sức chứa của phòng và số sinh viên được gán vào phòng được giải quyết bằng tiếp cận bài toán luồng cực đại với chi phí thấp nhất (minimum cost maximum flow). Giải pháp được triển khai bằng công nghệ điện toán đám mây của Google sử dụng ngôn ngữ Google Apps Script. Kết quả thực nghiệm trên dữ liệu Đăng ký thực hành cho học phần Thực hành Tin học căn bản của bộ môn Sư phạm Toán, Khoa Sư phạm cho thấy rằng giải pháp mà chúng tôi đề xuất là phù hợp.*

## 1 GIỚI THIỆU

Bài toán xếp thời khóa biểu là một trường hợp cụ thể của bài toán lập kế hoạch được ứng dụng rộng rãi trong nhiều lĩnh vực. Đây là bài toán được xếp vào lớp NP-khó (không tồn tại giải thuật tìm được lời giải tối ưu trong thời gian đa thức). Để tìm được lời giải tối ưu thường mất rất nhiều thời gian nhất là khi gia tăng số lượng môn học, giáo viên và phòng. Hiện nay có hai giải pháp chính được sử dụng để giải bài toán này: sử dụng tiếp cận tìm kiếm cục bộ (local search) [2, 9, 10] và giải pháp xếp lịch bán tự động. Tuy nhiên do bản chất của giải thuật, các tiếp cận tìm kiếm cục bộ không thể cho lời giải tối ưu. Wasfy và các cộng sự đã trình bày trong [13] một tiếp cận để giải quyết bài toán xếp lịch biểu bằng phương pháp quy hoạch nguyên (ILP) [1] sử dụng bộ phương pháp thỏa mãn ràng buộc (SAT) [6, 8, 11, 12]. Mặc dù phương pháp quy hoạch nguyên sử dụng thỏa mãn ràng buộc có nhiều thành công trong việc giải một số bài toán, nhưng các giải thuật ILP vẫn sử dụng heuristics để tìm kiếm lời giải nên kết quả có thể sẽ không tối ưu. Giải pháp bán tự động thường phải kết hợp với một vài phương pháp hiện thị để giúp người sử dụng tự mình xếp lịch biểu tốt. Tuy nhiên nhược điểm chính của phương pháp này là người sử dụng phải thực hiện rất nhiều thao tác để có được một lịch biểu hoàn chỉnh (chương trình máy tính chỉ hỗ trợ kiểm tra trùng lịch).

Từ khi Trường Đại học Cần Thơ quyết định thay đổi toàn diện và triệt để sang đào tạo theo hệ thống tín chỉ, việc xếp thời khóa biểu các học phần đã được phân cấp về các Khoa. Giáo vụ khoa chịu trách nhiệm xếp thời khóa biểu cho các học phần mà khoa mình phụ trách. Xét về bản chất việc làm này đã giảm kích thước đầu vào của bài toán xếp lịch biểu. Với sự trợ giúp của hệ thống xếp lịch biểu đang được sử dụng, việc xếp thời khóa biểu cho các học phần lý thuyết được giải quyết khá thuận lợi, dễ dàng. Tuy nhiên, đối với các học phần thực hành, hoặc các học phần vừa có lý thuyết vừa có thực hành đều không được xếp lịch thực hành trước. Lý do cơ bản là lịch thực hành sẽ phụ thuộc rất nhiều số lượng sinh viên đăng ký môn học. Nếu xếp lịch trước có thể dẫn đến việc sử dụng không hiệu quả các phòng thực hành (do số sinh viên đăng ký quá ít chẳng hạn). Do đó, hiện nay việc xếp lịch thực hành thường chỉ được thực hiện sau khi sinh viên đã đăng ký các học phần và có thời khóa biểu của các học phần lý thuyết. Đối với các Khoa có số học phần có thực hành nhiều như: Khoa Sư phạm, Khoa Khoa học Tự nhiên và Khoa Công nghệ Thông tin và Truyền thông (thực hành

tin học), Khoa Công nghệ (thực hành kỹ thuật điện), việc xếp lịch thực hành cho vài trăm sinh viên trở thành vấn đề. Các giải pháp hiện nay được các Khoa thực hiện có thể phân thành 2 nhóm. Nhóm thứ nhất, thực hiện thủ công: giáo viên phân nhóm và dán thông báo, sinh viên đăng ký trực tiếp trên giấy và tùy vào tình hình đăng ký mà giáo viên phụ trách giảng dạy sẽ xếp lịch. Nhóm thứ hai cho sinh viên đăng ký thông qua một hệ thống nào đó (ví dụ một website tự tạo). Sau khi đã có nhu cầu đăng ký, việc xếp lịch được thực hiện thủ công. Tóm lại, cho dù cho sinh viên đăng ký bằng cách nào đi nữa thì việc xếp lịch sau cùng vẫn phải thực hiện thủ công dẫn đến mất nhiều thời gian công sức thậm chí có thể tồn tại sai sót trong khi xếp lịch.

Trong bài báo này, chúng tôi đề xuất một giải pháp xếp lịch thực hành cho các học phần có thực hành tại Trường Đại học Cần Thơ. Giải pháp đề xuất sẽ thực hiện việc phân tự động sinh viên vào các nhóm thực hành phù hợp (không bị trùng lịch) và sử dụng các phòng thực hành một cách hiệu quả (khai thác tối đa hiệu suất sử dụng của phòng). Các đóng góp chính của bài báo gồm: (i) đề xuất quy trình đăng ký và xếp lịch biểu, (ii) mô hình hoá bài toán xếp lịch biểu về bài toán tìm luồng cực đại trên mạng, (iii) đề xuất hàm mục tiêu để tối ưu hiệu quả sử dụng các phòng thực hành cùng với giải thuật cải tiến cho bài toán luồng cực đại với chi phí thấp nhất trong bài toán xếp lịch biểu. Ngoài ra, các lựa chọn của sinh viên cũng sẽ được xem xét ưu tiên theo thứ tự. Ví dụ như: lựa chọn 1 sẽ được xem xét trước lựa chọn 2.

Phần tiếp theo của bài viết được trình bày như sau: phần 2 trình bày ngắn gọn về Bài toán xếp lịch biểu; phần 3 trình bày mạng và luồng cực đại trong mạng; phần 4 trình bày ứng dụng của luồng cực đại trong mạng cho bài toán xếp lịch biểu. Kết quả thực nghiệm được trình bày trong phần 5, tiếp theo là kết luận và hướng phát triển.

## 2 GIẢI PHÁP CHO BÀI TOÁN XẾP LỊCH BIỂU

Bài toán xếp lịch biểu chúng tôi đề cập đến trong bài viết này phát sinh từ nhu cầu thực tế ở Khoa CNTT&TT, Bộ môn Toán Khoa Sư phạm và Khoa Công nghệ, trường Đại học Cần Thơ. Các học phần thực hành (hoặc học phần vừa có lý thuyết vừa có thực hành) không được xếp thời biểu trước mà giảng viên (hoặc giảng giáo viên) phụ trách học phần phải tự xếp lịch. Quy trình xếp lịch thực hành cho mỗi học phần như sau: (i) với số lượng sinh viên đăng ký tham gia học phần đã biết, giảng viên ước lượng số lượng nhóm sao cho phù

hợp số lượng sinh viên và sức chứa của các phòng thực hành (phòng máy tính, phòng thí nghiệm); (ii) chọn buổi thực hành cho các nhóm và phân công giảng viên hướng dẫn thực hành (iii) thông báo cho sinh viên đăng ký nhóm và (iv) giảng viên phân sinh viên vào các nhóm. Giai đoạn (i) được thực hiện khá dễ dàng dựa vào số lượng sinh viên và sức chứa các phòng thực hành. Việc chọn buổi thực hành và phân công giảng viên được thực hiện dựa trên lịch rảnh của giáo viên và của phòng thực hành. Giai đoạn cho sinh viên đăng ký nhóm là công việc khá phức tạp. Giai đoạn cuối cùng là giai đoạn khó nhất: phân sinh viên vào các nhóm sao cho số lượng sinh viên của mỗi nhóm không quá lớn (vượt quá khả năng của phòng thực hành tương ứng với nhóm) và cũng không quá nhỏ (khai thác tối đa hiệu suất sử dụng phòng thực hành). Với quy trình xếp lịch như thế, chúng ta đã chia nhỏ bài toán xếp thời khóa biểu tổng quát (rất khó) thành các bài toán nhỏ (dễ hơn).

### 3 MẠNG VÀ LUỒNG CỰC ĐẠI TRONG MẠNG

*Mạng* (network) là một đồ thị có hướng  $G = \langle V, E \rangle$  với tập đỉnh  $V$  và tập cung  $E$ . Trên tập cung  $E$  ta định nghĩa hàm *khả năng thông qua* (capacity function) có giá trị không âm  $c: E \rightarrow \mathbf{R}^+$ . Để đơn giản, ta có thể giả sử  $G$  không chứa đa cung. Nếu có một cung đi từ đỉnh  $u$  đến đỉnh  $v$ , cung này sẽ là cung duy nhất và được ký hiệu là  $(u, v)$ . Trên mạng ta phân biệt 2 đỉnh: *đỉnh phát* (source)  $s$  và *đỉnh thu* (sink)  $t$ .

*Luồng* (flow) là một hàm  $f: E \rightarrow \mathbf{R}^+$  thỏa mãn các ràng buộc sau:

– Luồng trên cung không thể vượt quá khả năng thông qua của cung:

$$f(u, v) \leq c(u, v) \tag{1}$$

– Bảo toàn luồng: tổng luồng đi vào một đỉnh bằng tổng luồng đi ra khỏi đỉnh đó:  $\forall u \notin \{s, t\}$ ,

$$\sum_{(k, u) \in E} f(k, u) = \sum_{(u, v) \in E} f(u, v) \tag{2}$$

Ngoài ra, ta còn có:

– Tổng luồng đi ra đỉnh phát bằng tổng luồng đi vào đỉnh thu

$$\sum_{(s, u) \in E} f(s, u) = \sum_{(v, t) \in E} f(v, t) \tag{3}$$

– Giá trị luồng (value of flow), kí hiệu  $|f|$  được định nghĩa bằng tổng luồng đi ra khỏi đỉnh phát hoặc tổng luồng đi vào đỉnh thu:

$$|f| = \sum_{(s, u) \in E} f(s, u) = \sum_{(v, t) \in E} f(v, t) \tag{4}$$

Bài toán tìm luồng cực đại (trên mạng) là bài toán cực đại hóa  $|f|$  với các ràng buộc trên. Giải thuật đầu tiên được sử dụng để tìm luồng cực đại là giải thuật *đường tăng luồng* (augmenting path algorithm) của Ford & Fullkerson [5, 8]. Kế đến, Dinic [3] và Edmonds & Karp [4] đã độc lập cải tiến giải thuật đường tăng luồng để có được giải thuật có độ phức tạp thời gian đa thức. Kể từ đó, đã có rất nhiều giải thuật hiệu quả hơn được phát triển có thể kể đến như phương pháp *đẩy/gán nhãn lại* (push/relabel) của Golberg & Tarjan [7].

### 4 ỨNG DỤNG LUỒNG CỰC ĐẠI CHO BÀI TOÁN XẾP LỊCH BIỂU

Như đã phân tích ở trên, bài toán xếp lịch thực hành được quy về các bài toán con đơn giản hơn. Bài toán phân nhóm cho sinh viên là bài toán khó nhất và đáng được quan tâm. Trong phần này chúng tôi sẽ mô hình hóa bài toán phân nhóm về bài toán tìm luồng cực đại trên mạng.

Gọi:

- $S = \{u_1, u_2, \dots, u_n\}$  là tập các sinh viên
- $G = \{v_1, v_2, \dots, v_m\}$  là tập các nhóm
- $s$ : đỉnh phát
- $t$ : đỉnh thu
- $h(v)$ ,  $v \in G$ : sĩ số tối đa của nhóm (ứng với sức chứa của phòng)
- $l(v)$ ,  $v \in G$ : sĩ số tối thiểu của nhóm (ứng với sức chứa của phòng)

$d(u, v)$ ,  $u \in S, v \in G$ : hàm đăng ký nhóm của sinh viên.  $d(u, v) = 1$  nếu sinh viên  $u$  đăng ký nhóm  $v$ , ngược lại  $d(u, v) = 0$ .

Đồ thị  $G = \langle V, E \rangle$  của mạng sẽ là:

- $V = S \cup G \cup \{s, t\}$
- $E = \{(s, u) \mid u \in S\} \cup \{(u, v) \mid u \in S, v \in G, d(u, v) = 1\} \cup \{(v, t) \mid v \in G\}$
- $\forall u \in S: c(s, u) = 1$
- $\forall (u, v) \in E, u \in S, v \in G: c(u, v) = 1$
- $\forall v \in G, c(v, t) = h(v)$

Mỗi sinh viên được mô hình thành một *đỉnh-sinh viên*, mỗi nhóm cũng tương ứng với một *đỉnh-nhóm*. Ta thêm vào một đỉnh phát  $s$  và một đỉnh thu  $t$ .

Nếu sinh viên  $u_i$  đăng ký nhóm  $v_j$  sẽ có một cung nối từ *đỉnh-sinh viên*  $u_i$  đến *đỉnh-nhóm*  $v_j$ , khả

năng thông qua của cung này là 1 (mỗi sinh viên chỉ được phân vào 1 nhóm). Thêm các cung nối từ đỉnh phát  $s$  đến các đỉnh-sinh viên, khả năng thông qua của cung này cũng là 1 (mỗi sinh viên chỉ được đăng ký tối đa 1 nhóm). Sau cùng, thêm các cung nối các đỉnh-nhóm  $v \in G$  đến  $t$  với khả năng thông qua là  $h(v)$ . Mạng kết quả được minh họa trong Hình 1.

**4.1 Ràng buộc sĩ số tối thiểu**

Để tránh trường hợp một nhóm có quá ít sinh viên ta có thể bổ sung thêm ràng buộc sĩ số tối thiểu của một nhóm: tổng số sinh viên của nhóm  $v$  không được nhỏ hơn  $l(v)$ . Với mô hình luồng cực đại trên mạng, ràng buộc này có thể dễ dàng tích hợp vào mô hình: với mỗi cung, ngoài khả năng thông qua lớn nhất ta bổ sung thêm khả năng thông qua nhỏ nhất. Cách đơn giản nhất để giải bài toán luồng cực đại trên mạng với ràng buộc khả năng thông qua nhỏ nhất của các cung là biến đổi về bài toán luồng cực đại chuẩn. Ta xây dựng đồ thị  $G' = \langle V', E' \rangle$  từ đồ thị  $G = \langle V, E \rangle$ :

- Thêm vào hai đỉnh mới  $s'$  và  $t'$ .
- Thêm cung nối  $s'$  đến tất cả các đỉnh trong  $V$ .
- Thêm cung nối các đỉnh trong  $V$  đến  $t'$ .
- Thêm cung nối  $t$  đến  $s$ .
- Khả năng thông qua mới của các cung được định nghĩa như sau:
  - Với mỗi đỉnh  $v \in V$ , đặt:

- $c'(s', v) = \sum_{u \in V'} d(u, v)$  và
- $c'(v, t') = \sum_{w \in V'} d(v, w)$
- Với mỗi cung  $(u, v) \in E$ , đặt  $c'(u, v) = c(u, v) - d(u, v)$
- Đặt  $c'(t, s) = \infty$

Một cách trực quan, đồ thị  $G'$  được xây dựng bằng cách thay mỗi cung  $(u, v)$  trong đồ thị  $G$  bằng 3 cung:

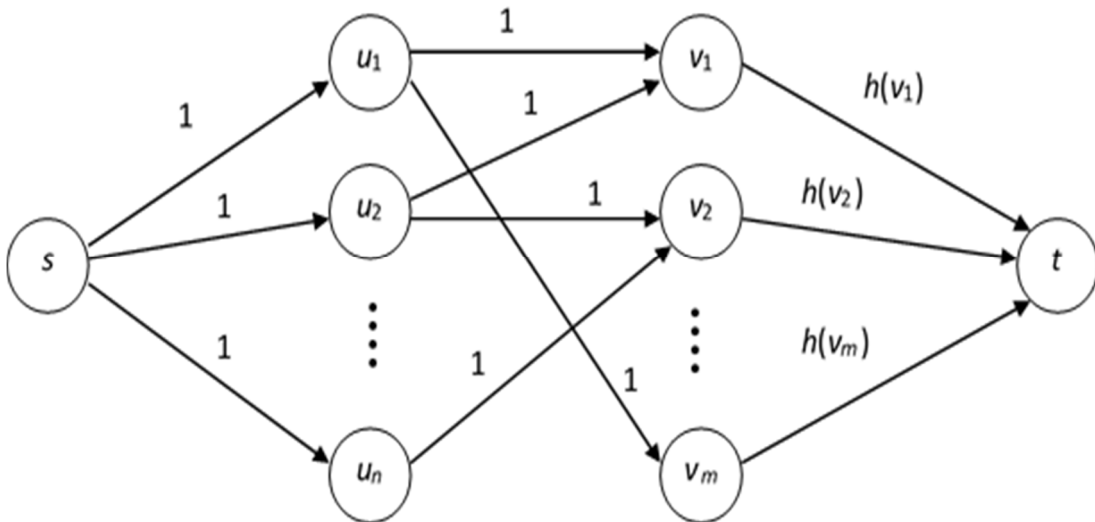
- $(u, v)$  với khả năng thông qua  $c(u, v) - d(u, v)$ ;
- $(s', v)$  với khả năng thông qua  $d(u, v)$  và
- $(u, t')$  với khả năng thông qua  $d(u, v)$

Trong  $G'$ , tổng khả năng thông qua của các cung xuất phát từ  $s'$  bằng tổng khả năng thông qua

của cả các cung đi vào  $t'$  và bằng  $\sum_{(u,v) \in E} d(u, v)$ . Ta gọi luồng có giá trị đúng bằng  $\sum_{(u,v) \in E} d(u, v)$  là

luồng bão hòa (saturating flow) vì nó bão hòa tất cả các cung rời khỏi  $s'$  hoặc đi vào  $t'$ . Nếu  $G'$  có luồng bão hòa, nó sẽ có luồng cực đại.

Người ta cũng đã chứng minh rằng: Mạng  $G$  sẽ có luồng khả thi (feasible flow) khi và chỉ khi  $G'$  có luồng bão hòa.



**Hình 1: Mạng phân nhóm sinh viên**

Giải thuật tìm luồng cực đại trong trường hợp khả năng thông qua của cung bị chặn dưới theo các bước như sau:

1. Xây dựng đồ thị  $G'$  và tìm luồng cực đại  $f'$  từ  $s'$  đến  $t'$  trên  $G'$ . Nếu tất cả các cung ra khỏi  $s'$  đều bão hòa (luồng đi qua cung này bằng khả năng thông qua lớn nhất của nó), ta tìm được *luồng khả thi* (admissible flow)  $f_a$  trên  $G$ :

$$f_a(u,v) = f'(u,v) + d(u,v)$$

với mọi cung  $(u,v) \in E$

2. Trên  $G$ , ta định nghĩa lại khả năng thông qua lớn nhất của các cung như sau:

- $c_r(u,v) = c(u,v) - f_a(u,v)$

3. Tìm luồng cực đại  $f_r$  từ  $s$  đến  $t$  trên  $G$  với khả năng thông qua lớn nhất  $c_r$ .
4. Luồng cực đại thỏa mãn ràng buộc khả năng thông qua lớn nhất và nhỏ nhất là:

- $f(u,v) = f_a(u,v) + f_r(u,v)$

#### 4.2 Tối ưu hoá hiệu quả sử dụng phòng

Như đã trình bày ở trên, mỗi nhóm được gán cho một phòng thực hành. Sĩ số tối đa của một nhóm thường là sức chứa của phòng thực hành tương ứng. Do đó, ngoài ràng buộc về sĩ số thấp nhất, ta cũng muốn số lượng sinh viên được xếp vào nhóm phải tương ứng với sĩ số tối đa của nhóm (ứng với sức chứa của phòng thực hành). Không xếp quá ít sinh viên vào một nhóm có sĩ số tối đa quá lớn. Bằng cách này, sẽ không có phòng quá tải trong khi các phòng khác thì quá trống. Như thế việc sử dụng phòng sẽ đạt hiệu quả hơn. Ràng buộc này có thể được mô hình hóa một cách đơn giản như sau:

$$\sum_{v \in G} \frac{c(v,t)}{f(v,t)} \rightarrow \min \tag{5}$$

Đó là tỉ số giữa sĩ số tối đa và số sinh viên thực sự được xếp vào nhóm  $v$  phải nhỏ nhất có thể. Như thế bài toán trở thành tìm luồng cực đại trên mạng thỏa mãn ràng buộc (5). Đây có thể xem như một biến thể của bài toán *luồng cực đại với giá thành cực tiểu* (minimum-cost maximum-flow). Giải thuật gốc cho bài toán *luồng cực đại với giá thành cực tiểu* được mô tả như sau:

1. Tìm luồng cực đại  $f$  trong mạng  $G$

2. Tìm chu trình giảm giá thành và cập nhật lại luồng  $f$ .
3. Lặp lại bước 2 cho đến khi không thể tìm được chu trình giảm giá thành nào nữa.

Do tính chất đặc biệt của bài toán phân nhóm sinh viên: có cấu trúc đồ thị đơn giản và hàm mục tiêu cần tối ưu khác với hàm mục tiêu của bài toán *luồng cực đại với giá thành cực tiểu*, chúng tôi đề xuất giải thuật giải quyết bài toán như sau:

1. Tìm luồng cực đại  $f$  trong mạng  $G$

2. Với mỗi đỉnh-sinh viên  $u \in S$

Xét các cặp  $v, w \in G$  có cung nối từ  $u$  đến  $v$  và cung nối từ  $u$  đến  $w$

Nếu  $f(u,v) = 1$  và  $f(u,w) = 0$

Nếu việc đổi luồng làm giảm hàm mục tiêu

Cập nhật lại luồng  $f$ .

Nếu  $f(u,w) = 1$  và  $f(u,v) = 0$

Nếu việc đổi luồng làm giảm hàm mục tiêu

Cập nhật lại luồng  $f$ .

3. Lặp lại bước 2 cho đến khi không thể tìm được chu trình giảm giá thành nào nữa.

Một ví dụ về tìm chu trình giảm giá thành được minh họa trong Hình 2. Chu trình tìm thấy là  $u \rightarrow w \rightarrow t \rightarrow v \rightarrow u$ . Đường liền nét là tăng luồng và đường không liền nét là đường giảm luồng.

#### 5 KẾT QUẢ THỰC NGHIỆM

Từ các phân tích được trình bày ở trên, giải pháp tổng thể cho quy trình xếp lịch được mô tả ngắn gọn như sau:

1. Chuẩn bị nhóm/phòng
2. Cho sinh viên đăng ký
3. Thu thập dữ liệu đăng ký và tiền xử lý
4. Xây dựng mạng
5. Tìm luồng cực đại trên mạng
6. Gán sinh viên vào nhóm dựa trên kết quả của luồng cực đại.

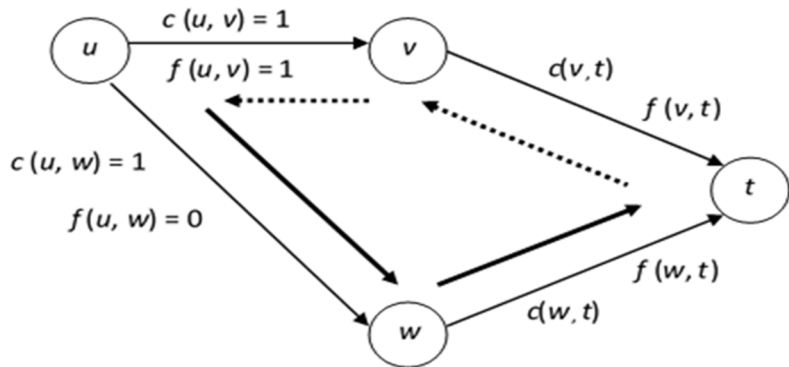
Chúng tôi đã triển khai quy trình đăng ký và xếp lịch thực hành trên nền tảng công nghệ điện toán đám mây của Google với ngôn ngữ lập trình ngữ Google Apps Script. Tất cả các thứ cần chuẩn bị là tạo một bảng tính (spreadsheet) dùng



làm nơi lưu trữ dữ liệu đăng ký, các thông số của hệ thống (số nhóm, sức chứa), các tham số của các giải thuật và cũng là nơi lưu kết quả xếp lịch. Việc tổ chức cho sinh viên đăng ký được thực hiện khá đơn giản với Google Form (bước 2). Tạo một biểu mẫu đăng ký với Google Form và gửi địa chỉ của biểu mẫu cho sinh viên. Một ví dụ về biểu mẫu đăng ký được trình bày trong Hình 3. Trong biểu mẫu này, phần quan trọng nhất là mã số sinh viên

và các lựa chọn của sinh viên. Tùy vào lịch rảnh của mình sinh viên sẽ lựa chọn các nhóm có lịch phù hợp. Đề bài toán để có lời giải, có thể khuyến khích sinh viên lựa chọn nhiều nhóm khác nhau. Trong thời hạn đăng ký, sinh viên có thể điều chỉnh các lựa chọn. Sau khi sinh viên đăng ký xong, dữ liệu đăng ký được tự động lưu trong bảng tính (bước 3).

**Hình 2: Tìm chu trình giảm giá thành trên mạng phân nhóm sinh viên**



Một phần của dữ liệu đăng ký thực hành môn Tin học căn bản ở Khoa Sư phạm, Đại học Cần Thơ được minh họa trong Bảng 1. Dữ liệu sẽ được xử lý trước khi chuyển qua bước xây dựng mạng và tìm luồng cực đại trong mạng. Trong phần tiền xử lý này, ta loại bỏ các sinh viên không có danh sách. Nếu sinh viên đăng ký nhiều lần ta ghi nhận lần đăng ký sau cùng và bỏ qua các lần đăng ký trước đó. Ngoài trừ các bước 1 và 2 ta phải thực hiện thủ công, các bước còn lại được lập trình để có thể thực hiện tự động. Bước 2 cũng có thể được thực hiện tự động nếu hệ thống được cung cấp thông tin về lịch rảnh của sinh viên.

còn lại là 480 được xếp vào 12 nhóm. Trường hợp không tối ưu tỉ lệ giữa sức chứa tối đa và số sinh viên được gán vào nhóm, sẽ có trường hợp 1 số nhóm có rất nhiều sinh viên, trong khi các nhóm khác có rất ít sinh viên (so với sức chứa tối đa của nhóm). Hàm mục tiêu (5) có giá trị 18.16. Trong trường hợp có tối ưu tỉ lệ sức chứa, ta thu được giá trị 16.25. Kết quả xếp lịch được lưu trữ lại bảng tính (xem Bảng 3) và có thể sử dụng để in kết quả theo từng nhóm như Bảng 4.

Sau khi dữ liệu đã được làm “sạch”, ta thiết lập các tham số cho hệ thống và giải thuật để bắt đầu xây dựng mạng và tìm luồng cực đại. Đầu tiên, có thể cho sức chứa tối thiểu của tất cả các nhóm bằng 0 và thực hiện việc xếp lịch. Tùy vào kết quả xếp nhóm, ta có thể điều chỉnh sức chứa tối thiểu của các nhóm. Ngoài ra ta cũng có thể bật/tắt chức năng tối ưu tỉ lệ của sức chứa tối đa và số sinh viên gán vào các nhóm. Nếu chức năng này được bật, hệ thống sẽ tìm luồng cực đại đồng thời tối ưu hàm mục tiêu (5).

**Lựa chọn nhóm**

**Lựa chọn 1 \***

**Lựa chọn 2 \***

**Lựa chọn 3 \***

**Lựa chọn 4 \***

**Hình 3: Một phần của Biểu mẫu đăng ký nhóm**

**Bảng 1: Dữ liệu đăng ký thực hành học phần tin học căn bản**

Timestamp	MSSV	Họ và tên	Lựa chọn 1	Lựa chọn 2	Lựa chọn 3	Lựa chọn 4
02/01/2013 14:57:05	B1200392	Nguyễn Dương Thanh Phú	Chiều thứ 2	Sáng thứ 3	Chiều thứ 5	Sáng thứ 5
02/01/2013 15:19:57	B1200352	Trương Gia Đại	Sáng thứ 6	Chiều thứ 6	Chiều thứ 7	Sáng thứ 7
02/01/2013 15:22:12	B1200962	Trương Thị Kiều Trang	Chiều thứ 2	Sáng thứ 6	Sáng thứ 2	Chiều thứ 6
02/01/2013 15:23:21	B1200467	Nguyễn Phi Vân	Sáng thứ 3	Chiều thứ 3	Sáng thứ 7	Chiều thứ 6
02/01/2013 15:27:56	B1200453	Lâm Minh Tài	Sáng thứ 3	Chiều thứ 6	Chiều thứ 7	Sáng thứ 7
02/01/2013 15:29:08	B1200370	Nguyễn Tùng Lâm	Chiều thứ 2	Sáng thứ 3	Chiều thứ 5	Sáng thứ 4
02/01/2013 15:33:01	B1207990	Dương Quang Khang	Sáng thứ 3	Chiều thứ 5	Sáng thứ 7	Chiều thứ 6
02/01/2013 15:34:09	B1208023	Phạm Thị Cẩm Tiên	Sáng thứ 3	Chiều thứ 6	Chiều thứ 7	Sáng thứ 7
02/01/2013 15:35:05	B1200471	Nguyễn Nguyễn Bá	Sáng thứ 3	Sáng thứ 3	Sáng thứ 7	Chiều thứ 6
02/01/2013 15:52:12	1117555	Hồ Trọng Nhân	Chiều thứ 4	Chiều thứ 3	Chiều thứ 7	Chiều thứ 6
02/01/2013 16:52:16	B1208188	Võ Văn Vũ	Sáng thứ 5	Chiều thứ 5	Sáng thứ 2	Sáng thứ 7
02/01/2013 16:55:49	B1200949	Châu Thị Kim Thi	Sáng thứ 5	Sáng thứ 4	Chiều thứ 5	Chiều thứ 4
02/01/2013 17:05:35	B1200973	Văn Thị Như Yến	Chiều thứ 2	Sáng thứ 6	Sáng thứ 6	Chiều thứ 2
02/01/2013 17:07:38	6116284	Nguyễn Hữu Thịnh	Sáng thứ 6	Sáng thứ 6	Sáng thứ 6	Sáng thứ 6
02/01/2013 17:09:33	6116251	Lý Tuyết Kha	Sáng thứ 6	Sáng thứ 6	Sáng thứ 6	Sáng thứ 6
02/01/2013 17:13:07	B1200386	Phạm Thị Hồng Nhân	Chiều thứ 2	Sáng thứ 3	Chiều thứ 5	Sáng thứ 5
02/01/2013 17:24:25	B1200421	LÊ MỘNG TUYẾN	Chiều thứ 2	Sáng thứ 3	Sáng thứ 2	Sáng thứ 5
02/01/2013 17:29:45	B1200386	Phạm Thị Hồng Nhân	Chiều thứ 2	Sáng thứ 3	Chiều thứ 5	Sáng thứ 5
02/01/2013 17:33:29	b1200394	QUÁCH THU PHƯƠNG	Chiều thứ 2	Sáng thứ 3	Sáng thứ 7	Sáng thứ 5
02/01/2013 17:34:29	B1200421	LÊ MỘNG TUYẾN	Chiều thứ 2	Sáng thứ 3	Sáng thứ 7	Sáng thứ 5

**Bảng 2: Kết quả xếp lịch trong trường hợp không tối ưu và tối ưu tỉ lệ sức chứa tối đa và số sinh viên trong nhóm**

STT	Nhóm	Sức chứa tối đa	Sức chứa tối thiểu	Không tối ưu		Tối ưu theo hàm mục tiêu (5)	
				SV/Nhóm	Tỉ lệ	SV/Nhóm	Tỉ lệ
1	Sáng thứ 2	75	40	40	1.88	40	1.88
2	Chiều thứ 2	75	0	74	1.01	45	1.67
3	Sáng thứ 3	40	39	39	1.03	39	1.03
4	Chiều thứ 3	72	0	35	2.06	44	1.64
5	Sáng thứ 4	40	0	25	1.60	33	1.21
6	Chiều thứ 4	40	0	28	1.43	33	1.21
7	Sáng thứ 5	75	70	70	1.07	70	1.07
8	Chiều thứ 5	75	0	55	1.36	45	1.67
9	Sáng thứ 6	40	0	40	1.00	32	1.25
10	Chiều thứ 6	40	0	40	1.00	33	1.21
11	Sáng thứ 7	40	0	18	2.22	33	1.21
12	Chiều thứ 7	40	0	16	2.50	33	1.21
<b>Tổng</b>				<b>18.16</b>		<b>16.25</b>	

**Bảng 3: Kết quả việc phân sinh viên vào các nhóm**

Timestamp	MSSV	Họ và tên	Email	Kết quả xếp nhóm	Lựa chọn 1	Lựa chọn 2	Lựa chọn 3	Lựa chọn 4
02/01/2013 15:22:12	B1200962	Trương Thị Kiều Trang	01693055217	Chiều thứ 2	Chiều thứ 2	Sáng thứ 6	Sáng thứ 2	Chiều thứ 6
02/01/2013 15:23:21	B1200467	Nguyễn Phi Vân	01259596990	Sáng thứ 7	Sáng thứ 3	Chiều thứ 3	Sáng thứ 7	Chiều thứ 6
02/01/2013 15:27:56	B1200453	Lâm Minh Tài	0985886645	Sáng thứ 7	Sáng thứ 3	Chiều thứ 6	Chiều thứ 7	Sáng thứ 7
02/01/2013 15:29:08	B1200370	Nguyễn Tùng Lâm	0939707015	Sáng thứ 4	Chiều thứ 2	Sáng thứ 3	Chiều thứ 5	Sáng thứ 4
02/01/2013 15:33:01	B1207990	Dương Quang Khang	0919493494	Chiều thứ 6	Sáng thứ 3	Chiều thứ 5	Sáng thứ 7	Chiều thứ 6
02/01/2013 15:34:09	B1208023	Phạm Thị Cẩm Tiên	0917525322	Sáng thứ 7	Sáng thứ 3	Chiều thứ 6	Chiều thứ 7	Sáng thứ 7
02/01/2013 15:35:05	B1200471	Nguyễn Nguyễn Bá	01655365883	Sáng thứ 7	Sáng thứ 3	Sáng thứ 3	Sáng thứ 7	Chiều thứ 6
02/01/2013 15:52:12	1117555	Hồ Trọng Nhân	nhan117555@student	BO QUA	Chiều thứ 4	Chiều thứ 3	Chiều thứ 7	Chiều thứ 6
02/01/2013 16:52:16	B1208188	Võ Văn Vũ	01286521844	BO QUA	Sáng thứ 5	Chiều thứ 5	Sáng thứ 2	Sáng thứ 7
02/01/2013 16:55:49	B1200949	Châu Thị Kim Thi	01882122021	BO QUA	Sáng thứ 5	Sáng thứ 4	Chiều thứ 5	Chiều thứ 4
02/01/2013 17:05:35	B1200973	Văn Thị Như Yến	0939698220	Chiều thứ 2	Chiều thứ 2	Sáng thứ 6	Sáng thứ 6	Chiều thứ 2
02/01/2013 17:07:38	6116284	Nguyễn Hữu Thịnh	01684823419	BO QUA	Sáng thứ 6	Sáng thứ 6	Sáng thứ 6	Sáng thứ 6
02/01/2013 17:09:33	6116251	Lý Tuyết Kha	01652238176	BO QUA	Sáng thứ 6	Sáng thứ 6	Sáng thứ 6	Sáng thứ 6
02/01/2013 17:13:07	B1200386	Phạm Thị Hồng Nhân	nhanB1200386@ctu	BO QUA	Chiều thứ 2	Sáng thứ 3	Chiều thứ 5	Sáng thứ 5
02/01/2013 17:24:25	B1200421	LÊ MỘNG TUYẾN	01258383095	BO QUA	Chiều thứ 2	Sáng thứ 3	Sáng thứ 2	Sáng thứ 5
02/01/2013 17:29:45	B1200386	Phạm Thị Hồng Nhân	nhanB1200386@ctu	Chiều thứ 5	Chiều thứ 2	Sáng thứ 3	Chiều thứ 5	Sáng thứ 5
02/01/2013 17:33:29	b1200394	QUÁCH THU PHƯƠNG	01699944396	Sáng thứ 7	Chiều thứ 2	Sáng thứ 3	Sáng thứ 7	Sáng thứ 5
02/01/2013 17:34:29	B1200421	LÊ MỘNG TUYẾN	01258383095	Sáng thứ 7	Chiều thứ 2	Sáng thứ 3	Sáng thứ 7	Sáng thứ 5

Xét về mặt thời gian thực hiện giải thuật tìm luồng cực đại, với số sinh viên khoảng 600-1000, thời gian thực hiện khoảng 20-30 giây. Lý do chính là do các giải thuật được cài đặt bằng Google scripts, nên thời gian thực hiện không thể so sánh với phiên bản cài đặt trên Desktop bằng các ngôn ngữ mạnh như C/C++. Tuy nhiên, thời gian thực hiện như trên vẫn có thể chấp nhận được. Hơn nữa, việc sử dụng nền tảng của Google để triển khai giải pháp lại giúp ta tiết kiệm được thời gian và công sức để xây dựng phần thu thập số liệu đăng ký.

**6 KẾT LUẬN**

Chúng tôi vừa đề xuất một giải pháp xếp lịch biểu cho các môn học sử dụng tiếp cận Luồng cực đại trong mạng. Chúng tôi đã mô hình hoá bài toán xếp lịch biểu về bài toán tìm luồng cực đại trên mạng. Ràng buộc mỗi sinh viên chỉ được gán vào một nhóm được mô hình bằng khả năng thông qua của cung xuất phát từ đỉnh-sinh viên bằng 1. Trong khi đó, ràng buộc về sức chứa của nhóm được mô hình hoá về khả năng thông qua của cung (cả cận trên và cận dưới). Chúng tôi cũng đã đề xuất một hàm mục tiêu phù hợp với việc tối ưu tỉ lệ giữa sức

chứa tối đa và số sinh viên vào nhóm cùng với một giải thuật tìm luồng cực đại với hàm mục tiêu nhỏ nhất. Tất cả các đề xuất đã được cài đặt trên nền tảng điện toán đám mây của Google. Chúng tôi đã thực nghiệm giải pháp đề xuất trên dữ liệu đăng ký thực hành của các sinh viên ở các Khoa của Trường Đại học Cần Thơ: Khoa Sư phạm (Tin học căn bản), Khoa Công nghệ Thông tin và Truyền thông (Xử lý ảnh), Khoa Công nghệ (Kỹ thuật điện). Kết quả cho thấy rằng giải pháp rất hiệu quả và tính đơn giản và chi phí thấp của nó.

Dựa vào các phân tích trong phần thực nghiệm, có thể thấy rằng có thể phát triển giải pháp được đề xuất bằng cách giải quyết một số vấn đề sau: (i) Tự động hoá quá trình thu thập dữ liệu đăng ký từ thời khoá biểu các môn học lý thuyết của sinh viên, (ii) Tổng quát hoá hàm mục tiêu tối ưu sao cho có thể mô hình được nhiều điều kiện tối ưu khác nhau ví dụ như: phân nhóm sao cho số nhóm sử dụng là ít nhất, (iii) và quan tâm đến vấn đề bảo mật khi đăng ký ví dụ như sử dụng tài khoản và mật khẩu để đăng nhập mới có thể đăng ký được. Chúng tôi dự định giải quyết các vấn đề trên trong một tương lai gần.

**Bảng 4: In danh sách sinh viên theo nhóm**

A	B	C	D
<b>Nhóm: Sáng thứ 2, Phòng: , Buổi: Sáng thứ 2</b>			
STT	MSSV	Họ và tên	Số điện thoại/email
1	1117535	võ thị mỹ xuyên	1643262750
2	7108781	Võ Thị Kim Ngọc	939564088
3	b1200299	Huỳnh Lâm Nhã	1654361913
4	B1200337	Nguyễn Hoàng Uyên	uyenB1200337@student.ctu.edu.vn
5	B1200373	NGUYỄN TÀI LINH	1222885531
6	b1200444	huỳnh viêm my	myb1200444@student.ctu.edu.vn
7	B1200486	Trần Đức Mạnh	manhB1200486@ctu.edu.vn
8	b1200506	Lý Mỹ Trinh	trinhb1200506@student.ctu.edu.vn
9	b1200538	Lê Thị Sơn	sonb1200538@student.ctu.edu.vn
10	b1200539	Bùi Thị Thanh Tâm	tamb1200539@student.ctu.edu.vn
11	b1200543	Nguyễn Thị Phương Thùy	thuyb1200543@student.ctu.edu.vn
12	B1200563	Lê Minh Bằng	bangB1200563@student.ctu.edu.vn
13	B1200600	Mai Phước Lộc	963880394
14	b1200667	trần chúc linh	linhb1200667@student.ctu.edu.vn
15	b1200708	Phan Ngọc Hà	963880061
16	B1200712	NGUYỄN THỊ HUYỀN	1629685510

**TÀI LIỆU THAM KHẢO**

1. Aloul, F., A. Ramani, I. Markov, and K. Sakallah, Generic ILP Versus Specialized 0-1 ILP: An Update, in *Proc. of the Int'l*

*Conference on Computer Aided Design*, 450-457, 2002.

2. Cosla, D., A Tabu Search Algorithm for Computing an Operational Timetable, in



- European Journal of Operational Research*, vol. 76, 98-110, 1994.
3. Dinic, E., A. Algorithm for solution of a problem of maximum flow in a network with power estimation. *Soviet Math. Doklady* 11: 1277–1280, 1970.
  4. Edmonds, J., Karp, Richard M., Theoretical improvements in algorithmic efficiency for network flow problems. *Journal of the ACM (Association for Computing Machinery)* 19 (2): 248–264, 1972.
  5. Ford, L. R., Fulkerson, D. R. Maximal flow through a network. *Canadian Journal of Mathematics* 8: 399–404, 1956.
  6. E. Goldberg and Y. Novikov, “BerkMin: A Fast and Robust SAT-solver,” in *Proc. of the Design Automation and Test Conference in Europe (DATE)*, 142-149, 2002.
  7. Goldberg, Andrew V., Robert E. Tarjan. A new approach to the maximum flow problem. *Annual ACM Symposium on Theory of Computing*, 136–146. ISBN 0-89791-193-8, 1986.
  8. Heineman, George T., Gary Pollice, and Stanley Selkow. Chapter 8: Network Flow Algorithms. *Algorithms in a Nutshell*. Oreilly Media. p. 226–250, 2008.
  9. Hertz, A., Tabu Search for Large Scale Timetabling Problems, in *European Journal of Operation Research*, 54(1), 39-47, 1991.
  10. Mooney, E., R. Dargen, and W. Parameter, Large-Scale Classroom Scheduling, in *IIE Trans.*, 28(5), 369-378, 1996.
  11. Moskewicz, K., C. Madigan, Y. Zho, and S. Malik, Chaff: Engineering an efficient SAT solver, in *Proc. of the Design Automation Conference (DAC)*, 503-535, 2001.
  12. Tam V., and D. Ting, Combining the Min-Conflicts and Look- Forward Heuristics to Effectively Solve a Set of Hard University Timetabling Problems, in *Proc. of the IEEE International Conference on Tools with Artificial Intelligence*, 2003.
  13. Wasfy A. and Fadi A. Aloul, Solving the University Class Scheduling Problem Using Advanced ILP Techniques, in *Proc. of IEEE GCC’07*, 2007.