Tạp chí Tin học và Điều khiển học, T. 15, S. 2 (1999), 61-68

PHƯƠNG PHÁP GIẢI MỘT LỚP BÀI TOÁN QUY HOẠCH NGUYÊN CÓ CẤU TRÚC ĐẶC BIỆT

VÕ VĂN TUẤN DŨNG, TRẦN VŨ THIỆU

Abstract. In this paper a finite algorithm for solving a class of integer problems with special structure is presented. It is based on techniques commonly used in solving transportation problems to improve feasible solutions. A technique for reducing the size of problems to be solved is also presented. Finally, some computational results are reported.

1. NỘI DUNG BÀI TOÁN

Xét bài toán quy hoạch nguyên phi tuyến:

(P)
$$f(x) = \max_{1 \le j \le n} \sum_{i=1}^{m} x_{ij} \to \min, \tag{1}$$

$$\sum_{j=1}^{n} x_{ij} = p_i, \quad i = 1, 2, \ldots, m, \tag{2}$$

$$0 \le x_{ij} \text{ nguyên} \le a_{ij}, \quad i = 1, 2, \ldots, m, \; j = 1, 2 \ldots, n, \tag{3}$$

trong đó $a_{ij} \in \{0, 1\}$ và $p_i \in \{1, 2, \ldots, n\}$ là những hằng số cho trước.

Bài toán (P) là mô hình toán học cho một số *bài toán lập lịch* thường gặp trong thực tiễn. Dưới đây là một ví dụ về *bài toán xếp lịch học tập*:

Có m sinh viên và n chuyên đề cho các sinh viên này. Với mỗi sinh viên i cho biết

$$a_{ij} = \begin{cases} 1, & \text{nếu sinh viên } i \text{ ưa thích chuyên đề } j, \\ 0, & \text{nếu ngược lại, } i = 1, 2, \ldots, m, \; j = 1, 2, \ldots, n \end{cases}$$

và p_i nguyên dương là số chuyên đề mà sinh viên i cần học, $i = 1, 2, \ldots, m$.

Hãy tìm cách bố trí sinh viên học các chuyên đề phù hợp với họ sao cho mỗi sinh viên học đủ số chuyên đề cần học và làm đồng đều đến mức tối đa số sinh viên học trong các chuyên đề này?

Đặt các biến số:

$$x_{ij} = \begin{cases} 1, & \text{nếu sinh viên } i \text{ học chuyên đề } j, \\ 0, & \text{nếu ngược lại, } i = 1, 2, \ldots, m, \; j = 1, 2, \ldots, n. \end{cases}$$

Khi đó, mô hình toán học cho bài toán đặt ra chính là bài toán (**P**).

Có thể thấy (**P**) tương đương với bài toán *qui hoạch nguyên tuyến tính* sau đây:

(P′)
$$\min \Big\{ t \,\Big|\, \sum_{j=1}^{n} x_{ij} \ge p_i, \ \forall i, \ \sum_{i=1}^{m} x_{ij} \le t, \ \forall j, \ 0 \le x_{ij} \le a_{ij}, \ \forall i, j, \text{ biến } t \text{ nguyên} \Big\}$$

Mô hình bài toán (**P**) đã được xét trong các tài liệu [3] và [4]. Trong [3] các tác giả đã nêu một thuật toán với thời gian đa thức để giải (**P**), nhờ giải một dãy hữu hạn các bài toán luồng cực đại trên mạng (xem [1] và [2]).

Dưới đây, trên cơ sở khai thác triệt để cấu trúc đặc biệt của bài toán (**P**) chúng tôi sẽ đề xuất một phương pháp khác cũng cho phép giải (**P**) trong thời gian đa thức, nhưng khá trực quan và đơn giản dựa trên việc điều chỉnh các phương án hiện có cho tới khi thu được phương án tối ưu (*lời giải*). Phương pháp này chỉ cần sử dụng một số kiến thức cơ bản tương tự như trong bài toán vận tải dạng bảng nên tiện cho việc lập trình trên máy tính.

2. CƠ SỞ PHƯƠNG PHÁP GIẢI

Ký hiệu

$$a_i = \sum_{j=1}^{n} a_{ij}, \; i = 1, 2, \dots, m, \quad b_j = \sum_{i=1}^{m} a_{ij}, \; j = 1, 2, \dots, n, \quad p = \sum_{i=1}^{m} p_i > 0,$$

thì trong [3] đã chứng minh rằng điều kiện cần và đủ để bài toán (P) có lời giải là

$$a_i \geq p_i \text{ với mọi } i = 1, 2, \dots, m. \tag{4}$$

Ta giả thiết bài toán (P) thỏa mãn điều kiện (4).

Giả sử $x = \{x_{ij}\}$ là một phương án của (P). Ứng với x, ta kẻ một bảng gồm m hàng và n cột, mỗi hàng ứng với một sinh viên, mỗi cột ứng với một chuyên đề. Ô nằm ở hàng i cột j được gọi là ô (i, j). Ta quy ước gọi ô (i, j) là *ô đen* nếu $a_{ij} = 0$. Các ô còn lại sẽ được phân thành hai loại: *ô trắng* nếu $x_{ij} = 0$ và *ô xanh* nếu $x_{ij} = 1$. Ký hiệu

$$t_j^x = \sum_{i=1}^{m} x_{ij}, \; \forall j = 1, 2, \dots, n, \quad t^x = \max_{1 \leq j \leq n} t_j^x. \tag{5}$$

Với bất kỳ phương án $x = \{x_{ij}\}$ của (P), theo (5) ta luôn luôn có

$$\sum_{j=1}^{n} t_j^x = \sum_{j=1}^{n} \sum_{i=1}^{m} x_{ij} = \sum_{i=1}^{m} \sum_{j=1}^{n} x_{ij} = \sum_{i=1}^{m} p_i = p. \tag{6}$$

Cột j được gọi là *cột đầy* nếu $t_j^x = t^x$, gọi là *cột gần đầy* nếu $t_j^x = t^x - 1$ và gọi là *cột vơi* nếu $t_j^x \leq t^x - 2$. Để ý rằng các khái niệm ô trắng, ô xanh, cột đầy, gần đầy, cột vơi đều gắn liền với một phương án cụ thể nào đó.

Từ các khái niệm nêu trên, ta có ngay quy tắc đơn giản sau đây cho phép nhận biết lời giải của (P).

Mệnh đề 1. *Nếu phương án x không có cột vơi, nghĩa là*

$$t^x = t_{j_0}^x, \; t_j^x \geq t^x - 1, \; \forall j \neq j_0, \tag{7}$$

thì x là phương án tối ưu của bài toán (P).

Chứng minh. Từ (6) và (7) suy ra

$$p = \sum_{j=1}^{n} t_j^x > n(t^x - 1). \tag{8}$$

Giả sử tồn tại phương án y tốt hơn x, nghĩa là

$$t_j^y \leq t^x - 1, \; \forall j = 1, 2, \dots, n. \tag{9}$$

Từ (6) và (9) suy ra $p = \sum_{j=1}^{n} t_j^y \leq n(t^x - 1)$, điều này trái với (8). Vậy x là phương án tối ưu và mệnh đề được chứng minh. □

Xét phương án $x = \{x_{ij}\}$ của (P). Giả sử C là một dây chuyền các ô xanh và trắng (*trong phương án x*) xen kẽ nhau nối cột j_0 với cột j_k:

$$C = \{(i_0, j_0), (i_0, j_1), \dots, (i_{k-1}, j_{k-1}), (i_{k-1}, j_k), \; k \geq 1\}, \tag{10}$$

trong đó (i_t, j_t), $t = 0, 1, \dots, k-1$, là các ô trắng, còn (i_t, j_{t+1}), $t = 0, 1, \dots, k-1$, là các ô xanh. Ta đưa vào phép biến đổi phương án x như sau:

Phép biến đổi A. Trên dây chuyền C đổi các ô trắng cũ thành xanh và ô xanh cũ thành trắng.

Bổ đề sau đây cho thấy phép biến đổi trên không làm thay đổi giá trị hàm mục tiêu (1) của phương án x.

Bổ đề 1. *Giả sử x' nhận được từ x nhờ thực hiện phép biến đổi A trên dây chuyền các ô xanh và trắng xen kẽ nhau nối hai cột không đầy. Khi đó, ta có $t^{x'} = t^x$.*

Chứng minh. Giả sử có dây chuyền (10) nối 2 cột không đầy j_0 và j_k. Do trên mỗi hàng i_t, $t = 0, 1, \dots, k-1$ có vừa đúng hai ô trắng và xanh thuộc C nên $x' = \{x'_{ij}\}$ thỏa mãn (2) và (3), nghĩa là x' cũng là một phương án của bài toán (**P**). Tương tự, do trên mỗi cột $j = j_1, j_2, \dots, j_{k-1}$ có vừa đúng hai ô trắng và xanh thuộc C nên

$$t_j^{x'} = t_j^x, \ \forall j \neq j_0, j_k. \tag{11}$$

Mặt khác, do cột j_0 chỉ có một ô thuộc C, đó là ô trắng (i_0, j_0) nên

$$t_{j_0}^{x'} = t_{j_0}^x + 1, \tag{12}$$

và do cột j_k chỉ có một ô thuộc C, đó là ô xanh (i_{k-1}, j_k) nên

$$t_{j_k}^{x'} = t_{j_k}^x - 1. \tag{13}$$

Do j_0, j_k là các cột không đầy nên từ (11)-(13) suy ra $t^{x'} = t^x$. □

Tương tự, giả sử C là một chu trình các ô xanh và trắng xen kẽ nhau:

$$(i_0, j_0), (i_0, j_1), \dots, (i_k, j_k), (i_k, j_0), (i_0, j_0) \quad (k \geq 1),$$

trong đó (i_t, j_t), $t = 0, 1, \dots, k$ là các ô trắng, còn (i_t, j_{t+1}), $t = 0, 1, \dots, k-1$ và (i_k, j_0) là các ô xanh. Ta đưa vào phép biến đổi:

Phép biến đổi B. Trên chu trình C đổi các ô trắng cũ thành xanh và ô xanh cũ thành trắng.

Bổ đề 2. *Giả sử x' nhận được từ x nhờ thực hiện phép biến đổi B trên chu trình các ô xanh và trắng xen kẽ nhau. Khi đó, ta có $t^{x'} = t^x$.*

Gán số cho các hàng và cột

Ta sẽ lần lượt gán số cho các hàng và cột của bảng như sau. Gán số 0 cho các cột đầy (cột j với $t_j^x = t^x$). Nếu cột j đã được gán số thì gán số j cho các hàng i chưa được gán số với $x_{ij} = 1$ ((i, j) là ô xanh). Nếu hàng i đã được gán số thì gán số i cho các cột j chưa được gán số với $a_{ij} - x_{ij} = 1$ (*tương đương với $a_{ij} = 1$, $x_{ij} = 0$, nghĩa là (i, j) là ô trắng*). Quá trình trên phải dừng sau không quá $m + n$ lần gán số cho các hàng và cột.

Nếu một cột vơi, chẳng hạn cột j_0 với $t_{j_0}^x \leq t^x - 2$, được gán số thì tồn tại một dây chuyền các ô xanh và trắng xen kẽ nhau nối cột đầy nào đó với cột vơi j_0. Cách tìm như sau: giả sử cột j_0 được gán số là i_0 ((i_0, j_0) là ô trắng) và hàng i_0 được gán số là $j_1 \neq j_0$ ((i_0, j_1) là ô xanh). Giả sử cột j_1 được gán số khác số 0, chẳng hạn $i_1 \neq i_0$ ((i_1, j_1) là ô trắng), và hàng i_1 được gán số là $j_2 \neq j_0, j_1$ ((i_1, j_2) là ô xanh). Nếu cột j_2 được gán số khác số 0, ta lại tiếp tục làm như trên. Do số cột trong bảng là hữu hạn (bằng n) nên cuối cùng, ta phải tìm được cột $j_k \neq j_t$, $t = 0, 1, \dots, k-1$, được gán số 0, nghĩa là j_k là cột đầy, và dây chuyền cần tìm là

$$C = \{(i_0, j_0), (i_0, j_1), \dots, (i_{k-1}, j_{k-1}), (i_{k-1}, j_k), \ (k \geq 1)\},$$

trong đó (i_t, j_t), $t = 0, 1, \dots, k-1$, là các ô trắng, còn (i_t, j_{t+1}), $t = 0, 1, \dots, k-1$ là các ô xanh. Ta có các mệnh đề sau đây.

Mệnh đề 2. *Nếu tồn tại dây chuyền các ô xanh và trắng xen kẽ nhau nối một cột đầy với một cột vơi thì ta có thể điều chỉnh phương án hiện có thành phương án mới tốt hơn hoặc có số cột đầy ít hơn.*

Chứng minh. Giả sử tìm được dây chuyền C nối cột đầy j_k với cột vơi j_0. Khi đó, ta thực hiện phép biến đổi A trên C. Lập luận tương tự như trong chứng minh Bổ đề 1, ta nhận được các hệ thức (11) - (13).

Do j_0 là cột vơi nên từ (11) - (13) suy ra rằng nếu j_k là cột đầy duy nhất trong phương án x thì $t^{x'} = t^x - 1$, nghĩa là phương án mới x' tốt hơn phương án cũ x. Nếu trái lại, ta có $t^{x'} = t^x$, nghĩa là x' không tồi hơn x, nhưng số cột đầy trong x' bớt đi một (bớt cột j_k). Mệnh đề được chứng minh. □

Mệnh đề 3. *Nếu không tồn tại dây chuyền các ô xanh và trắng xen kẽ nhau nối một cột đầy với một cột vơi thì phương án hiện có là phương án tối ưu.*

Chứng minh. Giả sử tồn tại phương án $y = \{y_{ij}\}$ tốt hơn phương án hiện có $x = \{x_{ij}\}$:

$$t^y < t^x, \tag{14}$$

trong đó t^x, t^y được tính theo (5). Ta sẽ dẫn đến điều vô lý. Thật vậy, từ (14) suy ra tồn tại chỉ số j sao cho $t_j^y < t_j^x$. Theo (6) ta có

$$\sum_{j=1}^{n} t_j^x = \sum_{j=1}^{n} t_j^y = \sum_{i=1}^{m} p_i = p,$$

nên tìm được cột j_0 sao cho

$$t_{j_0}^x < t_{j_0}^y \leq t^y < t^x. \tag{15}$$

Từ tính nguyên của các số trong dãy bất đẳng thức này cho thấy: $t_{j_0}^x \leq t^x - 2$, nghĩa là cột j_0 là cột vơi trong phương án x. Từ bất đẳng thức đầu trong (15) và từ định nghĩa (5) của $t_{j_0}^x$ và $t_{j_0}^y$ suy ra có tồn tại hàng i_0 sao cho $0 = x_{i_0 j_0} \neq y_{i_0 j_0} = 1$, lại do cả x lẫn y đều thỏa mãn (2) nên phải có

$$\sum_{j=1}^{n} x_{i_0 j} = \sum_{j=1}^{n} y_{i_0 j} = p_{i_0}.$$

Từ đó tìm được cột j_1 sao cho $((i_0, j_1)$ là ô xanh): $x_{i_0 j_1} = 1$, $y_{i_0 j_1} = 0$. Nếu có $t_{j_1}^x \leq t_{j_1}^y$ thì tồn tại hàng i_1 sao cho $((i_1, j_1)$ là ô trắng): $x_{i_1 j_1} = 0$, $y_{i_1 j_1} = 1$, lại do (2) nên tồn tại cột j_2 sao cho $((i_1, j_2)$ là ô xanh): $x_{i_1 j_2} = 1$, $y_{i_1 j_2} = 0$. Tiếp tục làm như vậy sẽ dẫn đến một trong hai tình huống sau:

a) Tìm được cột j_r sao cho $t_{j_r}^x > t_{j_r}^y$. Có hai khả năng xảy ra

a₁) $t_{j_r}^x = t^x$, nghĩa là j_r là cột đầy. Trong trường hợp này ta có dây chuyền các ô xanh và trắng xen kẽ nhau:

$$(i_0, j_0), (i_0, j_1), \ldots, (i_{r-1}, j_{r-1}), (i_{r-1}, j_r), \tag{16}$$

nối cột đầy j_r với cột vơi j_0. Điều này trái với giả thiết của Mệnh đề, vì thế khả năng này không thể xảy ra.

a₂) $t_{j_r}^x < t^x$, nghĩa là j_r không phải là cột đầy. Thực hiện phép biến đổi A trên dây chuyền (16), ta được phương án mới x' với $t^{x'} = t^x$ (Bổ đề 1) và số các thành phần của x', y khác nhau sẽ giảm ít nhất là 2.

b) Tìm được chu trình

$$(i_0, j_0), (i_0, j_1), \ldots (i_s, j_s), (i_s, j_0), (i_0, j_0) \quad (s \geq 1). \tag{17}$$

Thực hiện phép biến đổi B trên chu trình (17), ta được phương án mới x' với $t^{x'} = t^x$ (Bổ đề 2) và số các thành phần của x', y khác nhau sẽ giảm ít nhất là 4.

Nếu vẫn có $x' \neq y$, ta lặp lại cách làm trên, thay x bởi x'. Do số thành phần của x và y là hữu hạn và mỗi lần điều chỉnh số thành phần x, y khác nhau thực sự giảm, nên sau một số hữu hạn lần điều chỉnh ta phải được phương án $\hat{x} = y$, đồng thời $t^{\hat{x}} = t^x$, nghĩa $t^y = t^{\hat{x}} = t^x$. Điều này trái với giả thiết phản chứng $t^y < t^x$! Mệnh đề được chứng minh. □

3. THUẬT TOÁN GIẢI BÀI TOÁN (P)

Từ các kết quả nêu ở trên, ta đi đến thuật toán sau đây để giải (**P**).

Bước 0: Lập bảng gồm m hàng và n cột, mỗi hàng tương ứng với một sinh viên, mỗi cột tương ứng với một chuyên đề. Tô đen các ô (i,j) có $a_{ij} = 0$ (*ô đen không thay đổi trong suốt quá trình giải bài toán*), gọi các ô còn lại là ô trắng.

Bước 1: *Xây dựng phương án ban đầu.* Với mỗi hàng i, từ hàng 1 tới hàng m, ta lần lượt ghi số 1 vào các ô trắng, từ trái qua phải cho đến khi đủ p_i số (các ô còn lại của hàng xem như được ghi số 0), rồi chuyển sang hàng tiếp theo. Kết quả là ta nhận được phương án ban đầu $x^1 = \{x^1_{ij}\}$. Cũng có thể xuất phát từ một phương án đã biết bất kỳ của (**P**). Đặt $k = 1$. Chuyển sang Bước 2.

Bước 2: *Kiểm tra tối ưu.* Với phương án x^k nhận được, ta quy ước gọi các ô được ghi số 1 là ô xanh, các ô (khác ô đen) được ghi số 0 là ô trắng. Tính các số

$$t^k_j \equiv t^{x^k}_j = \sum_{i=1}^{m} x^k_{ij}, \quad j = 1, 2, \ldots, n \quad \text{và} \quad t^k \equiv t^{x^k} = \max_{1 \leq j \leq n} t^k_j.$$

Với phương án x^k, ta qui ước gọi cột j là cột đầy nếu $t^k_j = t^k$, gọi là cột gần đầy nếu $t^k_j = t^k - 1$ và là cột vơi nếu $t^k_j \leq t^k - 2$. Nếu không có cột vơi thì theo Mệnh đề 1, x^k là phương án tối ưu. Nếu trái lại, ta thực hiện gán số cho các hàng và cột của bảng như đã nêu ở mục trước. Sau khi gán số, nếu không có cột vơi nào được gán số thì x^k là phương án tối ưu (Mệnh đề 3). Trái lại, ta tìm được một dây chuyền C có dạng (10), gồm các ô xanh và trắng xen kẽ nhau nối một cột đầy j_k với một cột vơi j_0. Chuyển sang Bước 3.

Nếu trong quá trình gán số cho các hàng và cột, phát hiện có dây chuyền các ô xanh và trắng xen kẽ nhau nối một cột đầy với một cột vơi, thì ta có thể dừng quá trình gán số và chuyển ngay sang Bước 3 để điều chỉnh phương án.

Bước 3: *Điều chỉnh phương án.* Thực hiện phép biến đổi A trên dây chuyền C nhận được ở Bước 2. Kết quả là ta nhận được phương án mới x' hoặc tốt hơn phương án x^k ($t^{x'} < t^k$) hoặc x' sẽ có số cột đầy ít hơn số cột đầy của x^k (Mệnh đề 2). Đặt $x^{k+1} = x'$. Thay $k \leftarrow k + 1$, rồi quay lại Bước 2.

Mệnh đề 4. *Thuật toán nêu trên kết thúc sau một số hữu hạn bước.*

Chứng minh. Nếu thuật toán không dừng ở Bước 2, thì sau mỗi lần điều chỉnh ở Bước 3, ta sẽ nhận được phương án mới tốt hơn hoặc có số cột đầy ít hơn. Do hàm mục tiêu của bài toán chỉ có thể nhận một số hữu hạn giá trị nguyên, dương và do số cột trong bài toán cũng là hữu hạn (bằng n), nên các bước lặp nêu trên không thể kéo dài vô hạn. Mệnh đề được chứng minh. □

Độ phức tạp tính toán của thuật toán

Để đánh giá độ phức tạp tính toán của thuật toán đã nêu trên, ta tiến hành phân tích độ phức tạp của từng bước trong thuật toán:

Bước 1. Số phép toán cần thực hiện để xây dựng phương án ban đầu (kể cả việc tính tổng theo cột) tương đương với $O(m \times n)$.

Bước 2. Để tính lại các số t_j^{π} và t^k ta cần thực hiện số phép toán tương đương với $O(m+n)$ (tính truy hồi theo (11) - (13)). Phép gán số cho các hàng và cột trong bảng cần tối đa $O(m\times n)$ phép toán. Việc tìm dây chuyền "xanh - trắng" nối một cột đầy với một cột vơi cần thực hiện $O(m+n)$ phép toán (dựa theo số được gán cho các hàng và cột trong bảng). Như vậy, số phép toán cần ở Bước 2 tương đương với $O(m \times n)$.

Bước 3. Việc điều chỉnh luồng trên dây chuyền nối một cột đầy với một cột vơi cần tối đa $O(m+n)$ phép toán ($m+n$ bằng số tối đa các ô trên dây chuyền).

Các Bước 2 và 3 được lặp đi lặp lại một số lần. Sau mỗi lần lặp, hoặc giá trị hàm mục tiêu giảm đi 1 hoặc số cột đầy giảm đi 1. Do hàm mục tiêu (1) chỉ nhận nhiều nhất là $\max\limits_{1\le j\le n} b_j \le m$ giá trị và do số cột trong bảng bằng n nên số lần lặp tối đa là $m \times n$. Như vậy, số phép toán cần thực hiện theo thuật toán tương đương với $O((m\times n)\times(m\times n))$ hay $O(m^2\times n^2)$. Khi $m \approx n$ (số sinh viên xấp xỉ số chuyên đề) thì độ phức tạp của thuật toán tương đương với $O(n^4)$.

Tóm lại, thuật toán đã nêu cũng là một thuật toán đa thức, không những thế nó còn là đa thức mạnh, theo nghĩa: độ phức tạp tính toán chỉ phụ thuộc kích thước bài toán (m và n), chứ không phụ thuộc độ dài dữ liệu (các số a_i và p_i). Thực ra, trong bài toán (P) độ dài dữ liệu không vượt quá kích thước của bài toán!

Định lý. *Thuật toán giải nêu trên giải đúng bài toán sau thời gian* $O(n^2m^2)$.

4. GIẢM KÍCH THƯỚC BÀI TOÁN (P)

Giả sử $\underline{k}$ là một cận dưới đã biết cho giá trị tối ưu k^* của bài toán (P). Chẳng hạn, có thể lấy $\underline{k} = \left]\frac{p}{n}\right[$, trong đó $]x[$ biểu thị số nguyên nhỏ nhất lớn hơn hay bằng x (xem [3]). Ta có các nhận xét sau:

a) Nếu hàng i có $a_i = p_i$ thì từ (2) và (3) suy ra $x_{ij} = a_{ij}$, $\forall j = 1,2,\dots,n$. Điều này có nghĩa là nếu số chuyên đề mà một sinh viên ưa thích đúng bằng số chuyên đề mà anh ta cần học thì đương nhiên sinh viên đó được phân học tất cả các chuyên đề này.

b) Nếu cột j có $b_j \le \underline{k}$ thì ta có thể đặt $x_{ij} = a_{ij}$, $\forall i = 1,2\dots,m$. Nghĩa là nếu chuyên đề j có số sinh viên ưa thích không vượt quá cận dưới $\underline{k}$ thì đương nhiên ta sẽ bố trí học chuyên đề j cho tất cả các sinh viên ưa thích chuyên đề này, trừ ra những sinh viên đã phân học đủ số chuyên đề cần học. Cách bố trí này không ảnh hưởng tới giá trị tối ưu k^* của (P) (vì $k^* \ge \underline{k} \ge b_j \ge t_j^{\pi}$).

Từ những nhận xét nêu trên ta đi đến biện pháp sau đây làm giảm kích thước của bài toán ban đầu (P). Ký hiệu:

$$I = \{i : a_i = p_i\} \quad \text{và} \quad J = \{j : b_j \le \underline{k}\}. \tag{18}$$

Đặt

$$x_{ij} = a_{ij} \begin{cases} \forall i \in I,\ j = 1,2,\dots,n, \\ \forall i \notin I,\ \forall j \in J. \end{cases}$$

Như vậy, các sinh viên $i \in I$ do đã được phân học đủ số chuyên đề cần học nên không cần xét tiếp; các chuyên đề $j \in J$ do đã sắp xếp tối đa số sinh viên ưa thích chuyên đề đó nên cũng không cần xét tiếp.

Tính lại số chuyên đề mà mỗi sinh viên ($i \notin I$) cần phải học:

$$p'_i = p_i - \sum_{j \in J} a_{ij}, \ \forall i \notin I.$$

Đặt $I' = I \cup \{i : p'_i \le 0\}$. Tính lại số sinh viên ($i \notin I$) ưa thích các chuyên đề $j \notin J$:

$$b'_j = b_j - \sum_{i \in I'} a_{ij} = \sum_{i \notin I'} a_{ij}, \ \forall j \notin J.$$

Đặt $J' = J \cup \{j : b'_j = 0\}$. Số sinh viên đã được phân học chuyên đề $j \notin J'$ là $q_j = \sum_{i \in I} a_{ij} \ge 0$.

Loại bỏ các hàng $i \in I'$ và các cột $j \in J'$. Ta đi tới bài toán có dạng (sau khi đã đổi lại ký hiệu):

(Q)

$$\max_{1 \le j \le n} \Big(\sum_{i=1}^{m} x_{ij} + q_j \Big) \to \min, \tag{19}$$

$$\sum_{j=1}^{n} x_{ij} = p_i, \ i = 1, 2, \dots, m, \tag{20}$$

$$0 \le x_{ij} \text{ nguyên} \le a_{ij}, \ i = 1, 2, \dots, m, \ j = 1, 2 \dots, n, \tag{21}$$

với các dữ liệu thỏa mãn $a_{ij} \in \{0, 1\}$ và

$$a_i = \sum_{j=1}^{n} a_{ij} > p_i > 0, \ i = 1, 2, \dots, m, \tag{22}$$

$$b_j = \sum_{i=1}^{m} a_{ij} > 0, \ q_j \ge 0, \ j = 1, 2, \dots, n.$$

Điều kiện (22) đảm bảo cho (Q) có lời giải.

Chú ý. Trong các lập luận nêu trên, nếu $I = J = \emptyset$ (xem (18)), thì trong (Q) ta đặt $q_j = 0$, $\forall j = 1, 2, \dots, n$.

Có thể thấy bài toán (Q) tương đương với bài toán qui hoạch nguyên tính tính sau đây:

$$\min \Big\{ t \,\Big|\, \sum_{j=1}^{n} x_{ij} \ge p_i, \ \forall i, \ \sum_{i=1}^{m} x_{ij} \le t - q_j, \ \forall j, \ 0 \le x_{ij} \le a_{ij}, \ \forall i, j; \text{ biến } t \text{ nguyên} \Big\}$$

Như vậy, để giải bài toán (P) trước hết ta tiến hành giảm kích thước của bài toán ban đầu, nếu có thể được, để đưa (P) về dạng (Q) với m, n nhỏ hơn. Sau đó, để giải (Q) ta lần lượt thực hiện các bước như đã mô tả trong thuật toán đã nêu, chỉ có điểm khác duy nhất là các số t_j^k ở Bước 2 bây giờ được tính như sau:

$$t_j^k \equiv t_j^{x^k} = \sum_{i=1}^{m} x_{ij}^k + q_j, \quad j = 1, 2, \dots, n.$$

5. KẾT QUẢ THỬ NGHIỆM TRÊN MÁY VI TÍNH

Thuật toán giải nêu trên được lập trình bằng ngôn ngữ Pascal và chạy thử nghiệm trên máy vi tính AT 586, 100 MHz, 16 MB RAM với nhiều ví dụ bằng số cụ thể. Kết quả thử nghiệm nêu trong Bảng 1 cho thấy có thể dùng thuật toán này để giải bài toán xếp lịch học tập (P) cho các khóa học (đại học, cao học, NCS) như hiện nay: số người học mỗi khóa khoảng 100, số chuyên đề cần học khoảng 50. Thời gian tính chỉ khoảng vài chục giây.

Bảng 1. Kết quả thử nghiệm trên máy vi tính

$m \times n$	$m \times q$	p	t
25 × 15	25 × 210	156	0,11
40 × 15	40 × 336	258	0,17
60 × 15	60 × 501	379	0,50
80 × 15	80 × 665	506	0,82
100 × 15	100 × 832	632	1,21
120 × 15	120 × 1001	764	1,87
135 × 15	135 × 1127	874	2,36
30 × 25	30 × 413	365	0,33
40 × 25	40 × 543	483	0,60
60 × 25	60 × 806	724	1,26
80 × 25	80 × 1079	965	2,31
100 × 25	100 × 1349	1207	3,57
120 × 25	120 × 1622	1448	5,05
125 × 25	125 × 1690	1510	6,59
60 × 40	60 × 1164	1072	3,35
80 × 40	80 × 1560	1424	6,37
100 × 40	100 × 1949	1787	9,72
110 × 40	110 × 2147	1969	11,70
50 × 50	50 × 1195	1110	3,08
60 × 50	60 × 1441	1339	5,71
80 × 50	80 × 1928	1783	10,60
100 × 50	100 × 2410	2234	16,26

Các ký hiệu dùng trong bảng:
m : số sinh viên (số ràng buộc chính trong bài toán (**P**)),
n : số chuyên đề được xét trong bài toán (**P**),
p : tổng số chuyên đề các sinh viên cần học, $p = \sum_{i=1}^{m} p_i$,
q : số biến 0 - 1 trong bài toán (**P**) : $q = \sum_{i=1}^{m} a_i$,
t : thời gian tính toán (đơn vị: giây).

TÀI LIỆU THAM KHẢO

[1] R. K. Ahuja, T. L. Magnanti, and J. B. Orlin, *Network Flows: Theory, Algorithms and Applications*, Prentice Hall, N.J. 1993.

[2] S. Khuller, Y. J. Sussman, and W. Gasarch, Advanced Algorithms, *Lectures CMSC* 858, Jan. 30 - May 13, 1997.

[3] Nguyễn Đức Nghĩa, Võ Văn Tuấn Dũng, Thuật toán đa thức giải một lớp bài toán tối ưu rời rạc, *Tạp chí Tin học và Điều khiển học* **15** (1) (1999) 8-13.

[4] Nguyễn Huy Xương, *Mathématiques Discrètes et Informatiques*, Masson, Paris, 1992.

Nhận bài ngày 10 - 12 - 1998

Nhận lại sau khi sửa ngày 4 - 6 - 1999

Võ Văn Tuấn Dũng - Trường Đại học Bách khoa Hà Nội.

Trần Vũ Thiệu - Viện Toán học, Hà Nội.