Tạp chí Tin học và Điều khiển học, T.21, S.2 (2005), 99–107

BÀI TOÁN LẬP LỊCH TRONG CÁC TRƯỜNG ĐẠI HỌC VÀ THUẬT TOÁN TABU

BÙI MINH TRÍ, NGUYỄN NHƯ HẠNH

Trường Đại học Bách khoa Hà nội

Abstract. This paper presents a reformation of Tabu search approach and its application to solving the timetabling problem for universities.

Tóm tắt. Bài báo này trình bày một cải tiến của phương pháp tìm kiếm Tabu và áp dụng vào bài toán lập lịch cho các trường đại học.

1. MÔ HÌNH BÀI TOÁN XẾP LỊCH CHO CÁC BÀI GIẢNG ([1])

Bài toán xếp lịch học là một bài toán phân bổ tài nguyên trong đó các hoạt động là các bài giảng $i(i = 1, ..., n)$, các tài nguyên là các thời điểm $j(j = 1, ..., m)$ trong tuần (mỗi thời điểm được xét là bắt đầu một tiết học). Mục tiêu của bài toán này là xác định một thời khoá biểu không mâu thuẫn và thoả mãn nhiều nhất nguyện vọng của giáo viên. Hai bài giảng được xem là xung đột nếu diễn ra đồng thời (ít nhất có một phần thời gian) và yêu cầu một trong các điều kiện sau: cùng một giáo viên; cùng một phòng học; hai bài giảng có chung sinh viên hoặc cùng cho một lớp. Để tính đến việc thoả mãn yêu cầu của giáo viên ta đưa vào biến x_{ij} nhận giá trị 1 nếu giáo viên dạy bài giảng i tại thời điểm j , bằng 0 nếu ngược lại và hệ số c_{ij} là điểm mà giáo viên dạy bài giảng i tại thời điểm j cho theo nguyện vọng tại các thời điểm đã chọn (theo thứ tự nguyện vọng tốt nhất thì cho điểm 1 rồi đến điểm 2 ...) Nó có ý nghĩa là giáo viên thích dạy bài giảng đó ở thời điểm nào và không thích ở thời điểm nào. Bài toán lúc này có mô hình là:

$$\begin{cases} F(x) = \sum_{i=1}^{n}\sum_{j=1}^{m} c_{ij}x_{ij} \to \min \\ \sum_{j=1}^{m} x_{ij} = 1,\ i = 1...n,\ x_{ij} \in \{0,1\}.....(1) \\ x_{ij} + x_{kl} \leqslant 1,\ l \in J_{ijk},\ i < k \leqslant n,\ i = 1..n,\ j = 1..m......(2) \end{cases} \qquad (P1)$$

trong đó J_{ijk} là tập các thời điểm l mà nếu phân cho bài giảng k thì sẽ xảy ra xung đột với bài giảng i được phân tại thời điểm j.

Mô hình bài toán trên là bài toán quy hoạch nguyên {0,1} với số lượng các ràng buộc là rất lớn. Điều kiện (1) đảm bảo rằng mỗi bài giảng chỉ được phân bổ vào một thời điểm trong tuần. Điều kiện (2) đảm bảo không xảy ra xung đột giữa hai bài giảng. Nhưng nếu giải bài toán này thì sẽ có trường hợp không tồn tại cách xếp do có xung đột giữa hai bài giảng. Nếu ta chấp nhận một phương án có xung đột với chi phí cực tiểu thì ta xét bài toán với kỹ thuật phạt như sau:

$$\begin{cases} F(x) = \sum_{i=1}^{n}\sum_{j=1}^{m} c_{ij}x_{ij} + \frac{1}{2}\sum_{k=1}^{n}\sum_{l=1}^{m} d_{ijkl}x_{ij}x_{kl} \to \min \\ \sum x_{ij} = 1,\ i = 1...n,\ x_{ij} \in \{0,1\}, i = 1...n, j = 1...m(3) \end{cases} \quad (P2)$$

trong đó $d_{ijk} = \begin{cases} Mt_{ijkl}...khi....l \in J_{ijk} \\ 0....khi....l \notin J_{ijk}, \end{cases}$ M là hệ số phạt, t_{ijkl} là khoảng thời gian xảy ra xung đột giữa hai bài giảng i xếp tại thời điểm j và bài giảng k xếp tại thời điểm l.

Thông tin đầu vào được cập nhật qua phiếu giáo viên ứng với mỗi bài giảng với các thông tin về tên giáo viên, tên bài giảng, độ dài bài giảng, lớp giảng dạy, kiểu phòng yêu cầu và điểm c_{ij} mà giáo viên đó cho bài giảng i tại các thời điểm j. Từ các thông tin trên ứng với một phương án xác định ta sẽ tính được khi nào thì xảy ra xung đột và sẽ tính toán được các hệ số phạt và hàm mục tiêu.

2. PHƯƠNG PHÁP TÌM KIẾM TABU([1])

2.1. Phương án liền kề

Cho một phương án hiện hành $x = \{x_{ij}\} \in X$ (X là tập tất cả các phương án) khi đó phương án liền kề của x nhận được từ x bằng cách chỉ định lại tài nguyên cho một và chỉ một hoạt động nào đó.

$x' = \{x'_{ij}\} \in X$ là một phương án kiền kề của x nếu:

$x'_{kl} = x_{kl},\ k = 1...n(k \neq i),\ l = 1...m$

$x_{iR(x,i)} = 0,\ x'_{iR(x',i)} = 1$

trong đó $R(x,i)$ là tài nguyên được chỉ định cho hoạt động i của phương án x. Với một phương án hiện hành cho trước thì số các phương án liền kề là khổng lồ bằng $(m-1)^n$.

2.2. Danh sách Tabu (Tabu List)

+ Hai điểm liên quan: hai điểm x và x' gọi là có liên quan tại (i,j) nếu $R(x,i) = R(x',i) = j$.

Tập tất cả các điểm có liên quan tại (i,j) ký hiệu là $[i,j]$ và được gọi là họ các điểm liên quan tại (i,j).

+ Danh sách Tabu (TL): được định nghĩa là một tập nào đó của các họ có liên quan.

+ Danh sách Tabu kèm theo (TLS): là tập tất cả các điểm thuộc một trong các họ có liên quan trong danh sách Tabu. $TLS = \{x : x \in [i,j], [i,j] \in TL\}$. Nói cách khác TLS là tập tất cả các điểm nằm trong các họ liên quan đang bị cấm thuộc danh sách Tabu.

2.3. Lân cận tìm kiếm

Cho x là phương án hiện hành khi đó gọi $z = x \oplus (i,j)$ là một phương án được sinh ra bằng cách tái phân bổ lại hoạt động i sử dụng tài nguyên j thay cho $R(x,i)$ (các hoạt động khác vẫn không thay đổi tài nguyên đã được phân) trong phương án hiện hành x là một phương án nằm trong lân cận tìm kiếm của x và ta định nghĩa lân cận tìm kiếm của x là $N(x)$:

$$N(x) = \{z \in X : \exists(i,j) \text{ sao cho } z = x \oplus (i,j),\ 1 \leqslant i \leqslant n,\ 1 \leqslant j \leqslant m,\ j \neq R(x,i)\}$$

trong đó X là tập tất cả các phương án phân bổ.

2.4. Thuật toán tìm kiếm Tabu

Từ một phương án hiện hành x thuật toán Tabu tìm kiếm các phương án liền kề trong lân cận của x để thực hiện các phép dịch chuyển. Chỉ những phương án liền kề nằm trong $N(x)\backslash TL$ mới là đối tượng của sự tìm kiếm. Nếu một phương án liền kề $a \in N(x)\backslash TL$ mà cho hàm mục tiêu nhỏ hơn x thì ta chuyển phương án hiện hành sang a và x được đưa vào TL với các điểm có liên quan và ta lại tiếp tục tìm kiếm trên lân cận của phương án hiện có mới. Trái lại thì phương án a được đưa vào $N^*(x)$ chính là các điểm đã được xét trong lân cận của x. Tuy nhiên vì cỡ lân cận của x là rất lớn cho nên ta không thể nào tìm hết các phương án liền kề của x được vì vậy chúng ta sẽ khống chế việc tìm kiếm trong một giới hạn nhất định. Khi $|N^*(x)| > \max NV$ ($\max NV$ là số cực đại các phần tử thuộc $N^*(x)$) thì x sẽ được đưa vào danh sách Tabu và phương án hiện hành sẽ được chuyển đến phương án khả quan nhất trong các phương án liền kề của x đã được tìm kiếm trong $N^*(x)$. Khi số phần tử của danh sách Tabu đến giới hạn thì ta sẽ loại bỏ phần tử cũ nhất của danh sách Tabu đi để có thể tái xét lại phần tử đó. Điều kiện dừng của bài toán là khi đã vượt quá thời gian cho phép hoặc vượt quá số bước lặp liên tiếp không cải thiện được hàm mục tiêu.

Mô hình bài toán phân bổ tài nguyên và kỹ thuật Tabu đã được trình bày kỹ qua bài báo của các tác giả Đỗ Xuân Dương - Đại học thương mại và Phạm Huy Điển - Viện Toán học, Viện Khoa học và Công nghệ Việt Nam đăng trên tạp chí Ứng Dụng Toán Học, tập II, số 1, trang 31-48, do vậy ở đây chúng tôi chỉ giới thiệu sơ qua.

3. PHƯƠNG PHÁP TABU CẢI TIẾN

Như chúng ta thấy ở trên thuật toán Tabu là thuật toán duyệt lần lượt các phương án liền kề để tìm ra phương án tốt hơn. Tuy nhiên do bài toán lập lịch học là một bài toán phân bổ tài nguyên với số lượng ràng buộc rất lớn và Thuật toán Tabu mới chỉ chú ý đến việc đưa thuật toán thoát khỏi các hố cực tiểu địa phương mà chưa chú ý đến tốc độ hội tụ của bài toán. Đây cũng là một vấn đề rất quan trọng không những ảnh hưởng đến thời gian chạy chương trình mà nó còn quyết định đến kết quả cuối cùng của thuật toán. Mặt khác thuật toán Tabu cũng chưa chú ý đến việc tìm phương án xuất phát. Nếu trước khi chạy thuật toán Tabu chúng ta tìm được một phương án xuất phát càng tốt thì chúng ta càng nhanh chóng tiến tới gần phương án tối ưu và cũng tiết kiệm được thời gian tính toán. Dưới đây chúng tôi trình bày hai phương pháp tìm phương án xuất phát mà cho chúng ta một kết quả khá tốt. Ngoài ra chúng tôi cũng đưa vào cải tiến thuật toán Tabu bằng cách đưa ý tưởng của thuật toán tham lam vào trong các bước lặp của thuật toán Tabu. Với định hướng làm giảm hàm mục tiêu nhiều nhất có thể trong từng bước lặp chắc chắn thuật toán này sẽ hội tụ nhanh hơn thuật toán Tabu. Chúng ta sẽ thấy được hiệu quả của chúng khi chạy thử và so sánh hai thuật toán trên cùng một bộ số liệu.

3.1. Tìm phương án xuất phát bằng thuật toán tham lam ([2,4])

Ta sẽ chọn thời điểm cho lần lượt các bài giảng từ bài giảng 1 đến bài giảng n. Ứng với bài giảng thứ $i(i = 1...n)$ ta xét tất cả các cách đặt thời điểm bắt đầu bài giảng và chọn ra thời điểm mà làm cho hàm mục tiêu tính từ bài giảng thứ nhất đến bài giảng thứ i đó là nhỏ nhất. Đây là một trong những phương án thường được sử dụng khi ta tính xấp xỉ các phương án tối ưu. Nó thường được sử dụng để tính giá trị ban đầu của các bài toán trước khi tinh chỉnh để đạt được một phương án tối ưu.

3.2. Tìm phương án xuất phát bằng thuật toán Quy hoạch động ([3])

Bước 1: Chia bài toán ban đầu ra làm N bài toán nhỏ có kích thước khác nhau, cùng dạng mà bài toán cơ sở là bài toán có kích thước 1, tức là bài toán có số bài giảng là 1 ứng với số các thời điểm không thay đổi là M.
Bước 2: Bài toán cơ sở là bài toán ta có thể dễ dàng tìm ra phương án tối ưu cho từng thời điểm. Lưu các phương án đó lại để làm dữ liệu tính các kết quả tối ưu của bài toán có cỡ lớn hơn. Cứ như vậy cho đến bài toán lớn ban đầu.
Bước 3: Từ các kết quả của bài toán cỡ nhỏ để tìm ra kết quả của bài toán cỡ lớn hơn ta làm như sau: Để tìm ra kết quả tối ưu tại mỗi thời điểm j của bước i ta sẽ xét với tất cả các kết quả của bài toán ở bước $i-1$ cộng thêm với chi phí sau khi đã kết hợp với bài giảng thứ i được xếp tại thời điểm j ở bước i rồi tìm ra kết quả tốt nhất. Cứ như vậy cho đến $i = N$.
Bước 4: Từ kết quả ở bước N và các thông số lưu trữ của các bước trước ta sẽ tiến hành lần ngược ra phương án tối ưu.

3.3. Chi tiết thuật toán Tabu cải tiến

Kế thừa từ ý tưởng dùng thuật toán tham lam để tìm phương án xuất phát cho bài toán, chúng tôi cũng áp dụng ý tưởng đó vào trong thuật toán Tabu với hy vọng sẽ cải thiện được phần nào tốc độ hội tụ của bài toán. Như đã trình bày ở trên ý tưởng cơ bản của thuật toán tham lam là tối ưu hoá ở các bước cục bộ của bài toán. Tuy không tìm được phương án tối ưu toàn cục nhưng nó cũng góp phần định hướng cho thuật toán tiến gần đến phương án tối ưu một cách nhanh hơn thuật toán Tabu thường. Chúng ta có thể thấy rõ hơn điều này nếu trong một bước cục bộ của thuật toán mà tìm ra được rất nhiều phương án có hàm mục tiêu nhỏ hơn hàm mục tiêu tối ưu tính cho đến lúc đó (giả sử là 10). Khi đó thuật toán Tabu sẽ phải thực hiện ít nhất là 10 bước chuyển đổi phương án tối ưu mà mỗi bước chuyển đổi có thể phải xét rất nhiều các phương án trung gian (cụ thể là số phương án liền kề được xét cực đại có thể lên đến vài nghìn hoặc vài chục nghìn). Việc làm đó sẽ trở nên đơn giản đối với thuật toán Tabu cải tiến khi nó chuyển đến phương án tối ưu mới mà làm hàm mục tiêu giảm nhiều nhất có thể được trong bước cục bộ đó. Đó chính là nhờ vào định hướng giảm thiểu hàm mục tiêu trong một bước cục bộ của thuật toán tham lam.
Bước 0: giả sử x là lời giải xuất phát của bài toán (P2) (phương án này được xác định dựa trên một trong hai thuật toán tìm phương án xuất phát mà chúng tôi đã trình bày ở trên) chúng ta xác định các thông số ban đầu của thuật toán.

MaxTime: thời gian tính toán cực đại cho phép.

MaxStep: số các bước lặp liên tiếp không cải thiện được hàm mục tiêu.

MaxNV: là số cực đại các phần tử thuộc $N^*(x)$.

MaxLTL: là độ dài tối đa của danh sách Tabu

TimeCount: là biến đếm thời gian tính toán.

StepCount: là biến đếm số phép lặp liên tiếp không cải thiện hàm mục tiêu.

NVCount: số các phần tử hiện có trong $N^*(x)$.

LTLCount: số các phần tử hiện có trong danh sách Tabu.

CS: phương án hiện hành.

BCS: điểm tốt nhất hiện có.

NVCount:=0, LTLCount :=0, $N^*(x) := \phi; TL := \phi; CS := x; BCS := x;$

NVCount:=0; LTLCount:=0;

Bước 1: nếu TimeCount<MaxTime và StepCount<MaxStep thì i :=Random(N)và chuyển sang Bước 2, trái lại ta dừng thuật toán.

Bước 2: ứng với mỗi i được sinh ra ở bước một ta xét lần lượt từng thời điểm $j = 1 \to M$.

- Nếu $[i, j] \not\subset TL$ thì đặt $z := CS \oplus (i, j)$ thì kiểm tra:

+ Nếu $(P(z) < P(bcp))$ and $(P(z) < Min)$thì:

$$Min := P(z); Zmin := z; Lg := true;$$

+ Nếu $(P(z) \geq P(bcp))$ thì thêm z vào $N^*(x)$ và tăng NVCount lên 1. Khi ta đã xét hết các gián trị của j thì ta kiểm tra:

- Nếu $Lg = true$ (có sự thay đổi hàm mục tiêu) thì $BCP := zmin, CS := Zmin; StepCount := 0, NVCount := 0$ và chuyển sang Bước 4. Trái lại chuyển sang Bước 3.

Bước 3: Nếu $|N^*(x)| < MaxNV$ (2).

Nếu $P(z) < P(z')$ thì $z' := z$ và quay lại Bước 1. Trái lại (2) chuyển sang Bước 3.

Bước 4: Đặt $CS = z'$(z' chính là phương án tốt nhất trong $N^*(x)$)$StepCount := StepCount + 1;$

Bước 5: Cập nhật danh sách Tabu bằng việc đưa $[i, R(CS, i)]$ vào TL.

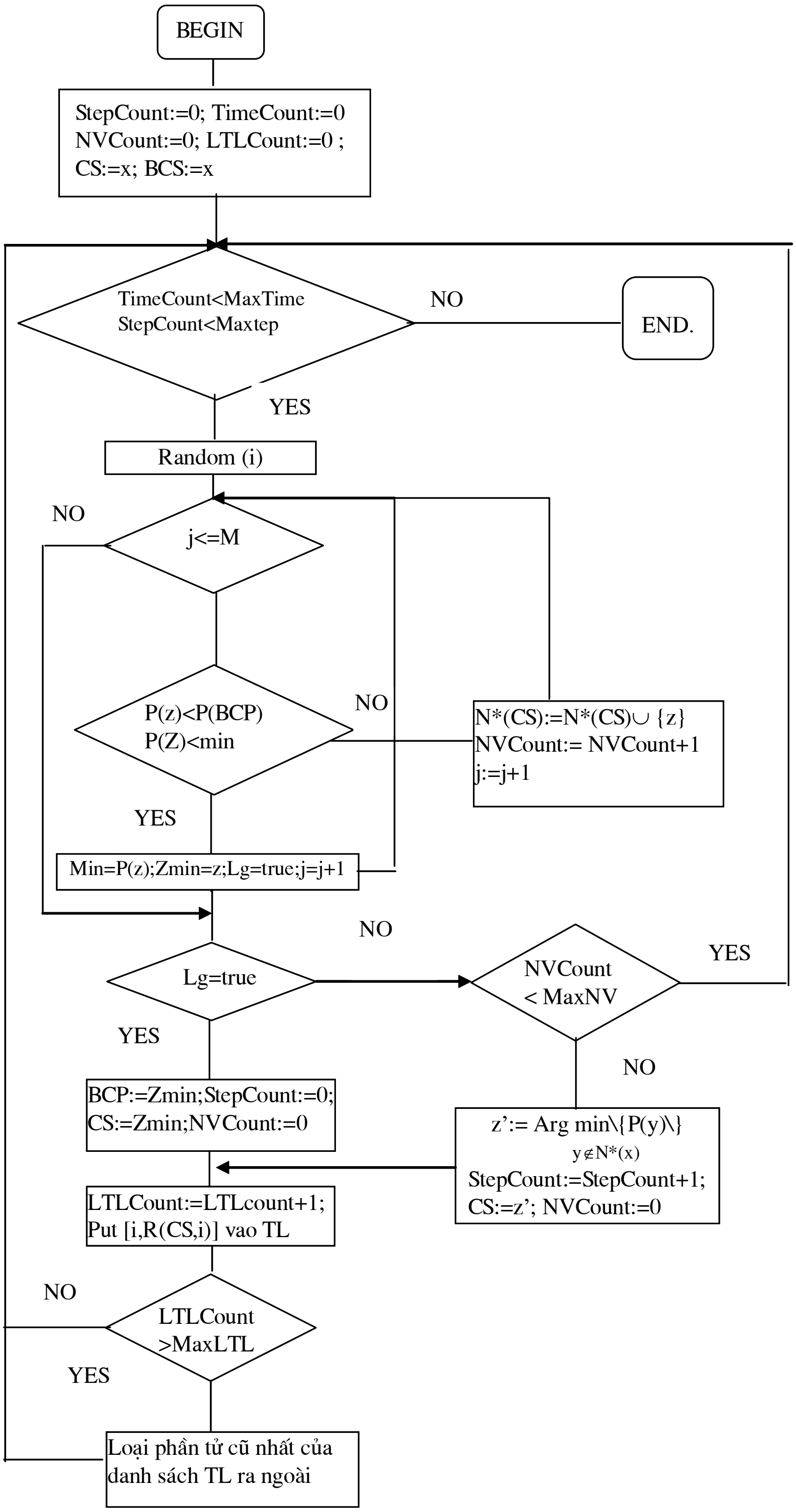
Nếu $LTLCount > MaxLTL$ thì loại phần tử cũ nhất của danh sách ra ngoài. Quay lại Bước 1.

Chúng ta sẽ hiểu rõ hơn thuật toán Tabu Cải tiến qua lưu đồ thuật toán dưới đây.

3.4. Kết quả chạy thử nghiệm của chương trình

Để so sánh được hiệu quả của hai thuật toán Tabu và Tabu cải tiến dưới đây tôi sẽ trình bày song song kết quả của hai thuật toán trên cùng một bộ số liệu.

• Số bài giảng $n = 500$, $MaxNV= 2000$, $MaxStep = 300, Maxtime = 60s$, độ dài danh sách Tabu thay đổi. Giá trị đầu của thuật toán Tabu: 2853718, giá trị đầu của thuật toán Tabu cải tiến: 549476.


BEGIN
StepCount:=0; TimeCount:=0
NVCount:=0; LTLCount:=0 ;
CS:=x; BCS:=x
TimeCount<MaxTime
StepCount<Maxtep
NO
END.
YES
Random (i)
NO
j<=M
P(z)<P(BCP)
P(Z)<min
NO
N*(CS):=N*(CS)∪ {z}
NVCount:= NVCount+1
j:=j+1
YES
Min=P(z);Zmin=z;Lg=true;j=j+1
NO
YES
Lg=true
NVCount
< MaxNV
YES
NO
BCP:=Zmin;StepCount:=0;
CS:=Zmin;NVCount:=0
z':= Arg min\{P(y)\}
y∉N*(x)
StepCount:=StepCount+1;
CS:=z'; NVCount:=0
LTLCount:=LTLcount+1;
Put [i,R(CS,i)] vao TL
NO
LTLCount
>MaxLTL
YES
Loại phần tử cũ nhất của
danh sách TL ra ngoài

MaxLTL	Giá trị cuối (Tabu)	Độ giảm (%) (Tabu)	Giá trị cuối (Tabu cải tiến)	Độ giảm (%) (Tabu cải tiến)
100	1190477	58.283%	548457	0.19%
500	1190444	58.284%	548457	0.19%
1000	1190460	58.283%	548454	0.19%
2000	1190465	58.283%	548455	0.19%
5000	1190475	58.283%	548456	0.19%

• N=500, MaxStep=300, Maxtime=60s, MaxLTL=1000, số lận cận tìm kiếm cực đại thay đổi. Giá trị đầu của thuật toán Tabu: 2853718, giá trị đầu của thuật toán Tabu cải tiến: 549476.

MaxNV	Giá trị cuối (Tabu)	Độ giảm (%) (Tabu)	Giá trị cuối (Tabu cải tiến)	Độ giảm (%) (Tabu cải tiến)
100	1190455	58.283%	548462	0.19%
500	1190455	58.283%	548446	0.19%
1000	1190471	58.283%	548460	0.19%
2000	1190468	58.283%	548448	0.19%
5000	1190506	58.282%	548459	0.19%

• Số bài giảng N=500, Maxstep=300, MaxNV=2000, MaxLTL=1000, thời gian chạy tối đa thay đổi. Giá trị đầu của thuật toán Tabu: 2853718, giá trị đầu của thuật toán Tabu cải tiến: 549476.

MaxTime	Giá trị cuối (Tabu)	Độ giảm (%) (Tabu)	Giá trị cuối (Tabu cải tiến)	Độ giảm (%) (Tabu cải tiến)
5	1190491	58.282%	548464	0.19%
10	1190488	58.282%	548462	0.19%
50	1190485	58.282%	548456	0.19%
100	1190464	58.283%	548454	0.19%
200	1190461	58.282%	548446	0.19%

• Để có thể thấy rõ hơn tốc độ hội tụ của hai thuật toán dưới đây chúng tôi đưa vào các kết quả chạy của hai thuật toán với cùng một bộ số liệu và với cùng một phương án xuất phát. Với các thông số cụ thể như sau: N=500, Maxstep=3000, MaxNV=2000, MaxLTL=1000, giá trị khởi đầu: 250072

MaxTime	Giá trị cuối (Tabu)	Độ giảm (%) (Tabu)	Giá trị cuối (Tabu cải tiến)	Độ giảm (%) (Tabu cải tiến)
10	225026	10.02%	223010	10.82%
30	223003	10.82%	222989	10.83%
60	220975	11.63%	218988	12.44%
100	219967	12.03%	217947	12.85%
200	219954	12.04%	217937	12.85%
300	218953	12.44%	216926	13.25%

Nhận xét:

+ Nhìn vào kết quả của thuật toán Tabu ta thấy giá trị cuối của hàm mục tiêu giảm khoảng 60% so với ban đầu và nó còn cho thấy kết quả cuối cùng cũng khá gần với kết quả tối ưu. Kết quả ta thu được sẽ càng chính xác nếu ta tăng thời gian tính toán của bài toán lên. Bởi vì đây là một bài toán có số lượng các ràng buộc cũng như số lượng các phép thử là rất lớn vì vậy để thu được kết quả tốt đòi hỏi phải chạy thuật toán với thời gian càng lớn càng tốt.

+ Nhìn vào kết quả của thuật toán Tabu cải tiến ta thấy rằng giá trị cuối của hàm mục tiêu không giảm nhiều so với ban đầu điều đó không có nghĩa là thuật toán sai mà có nghĩa là phương án xuất phát đã rất gần với phương án tối ưu. Có được điều đó có nghĩa là ta đã tìm được một phương án xuất phát tốt. Trong trường hợp ví dụ trên giá trị đầu tìm được bằng thuật toán Tabu cải tiến vẫn tốt hơn giá trị cuối của phương pháp Tabu.

+ Dựa vào kết quả chạy hai thuật toán với cùng một bộ số liệu, cùng phương án xuất phát và cùng các thông số khác chúng ta có thể thấy thuật toán Tabu cải tiến không những tốt hơn thuật toán Tabu về phương án xuất phát mà còn có tốc độ hội tụ nhanh hơn thuật toán Tabu. Có được điều này là do việc đưa ý tưởng của thuật toán tham lam vào kết hợp với thuật toán Tabu để từ đó cải thiện được tốc độ hội tụ của thuật toán. Kết qủa này càng khẳng định thêm những nhận xét mà chúng tôi đã nhận xét về thuật toán Tabu cải tiến là có cơ sở.

+ Mặt khác cũng từ kết quả chạy thử nghiệm hai thuật toán trên ta cũng có thể rút ra được kết luận chung sau: Độ tốt của kết quả cuối cùng của hai thuật toán phụ thuộc nhiều vào hai biến của điều kiện dừng là biến thời gian lặp cực đại và số bước lặp cực đại không cải thiện hàm mục tiêu. Độ dài của danh sách Tabu và số lân cận cực đại vào khoảng từ 1000 đến 5000 là phù hợp nhất.

TÀI LIỆU THAM KHẢO

[1] Đỗ Xuân Dương (Đại học thương mại) và Phạm Huy Điển (Viện Toán học và Công nghệ Việt Nam), Bài toán phân bổ tài nguyên và kỹ thuật Tabu, *Tạp Chí Ứng Dụng Toán Học tập* **II** (1) 31–38.

[2] Nguyên Đức Hiếu (Học viện kỹ thuật quân sự), Một thuật toán HEURISTIC giải bài toán phân lịch bay tiếp viên cho hãng hàng không quốc gia Việt Nam, *Tạp Chí Ứng Dụng Toán Học tập* **II** (1) 49–60.

[3] PGS. TS Bùi Minh Trí, *Quy hoạch toán học*, NXB Khoa học và kỹ thuật, Hà Nội, 2001.

[4] PGS.TS Bùi Minh Trí, *Tối ưu hoá tổ hợp*, NXB Khoa học và kỹ thuật, Hà Nội, 2003.

Nhận bài ngày 01 - 8 - 2005