Tạp chí Khoa học Trường Đại học Cần Thơ
Phần A: Khoa học Tự nhiên, Công nghệ và Môi trường
website: sj.ctu.edu.vn

DOI:10.22144/ctu.jvn.2019.132

THUẬT TOÁN QUY HOẠCH ĐỘNG CHO BÀI TOÁN XẾP BA LÔ CÂN BẰNG {0,1}

Võ Nguyễn Minh Hiếu[1*], Trần Thủ Lễ[2] và Nguyễn Ngọc Đăng Duy[3]

[1]*Sinh viên Sư phạm Toán học, Khóa 42, Trường Đại học Cần Thơ*

[2]*Lớp Cao học Giải tích, Khóa 25, Trường Đại học Cần Thơ*

[3]*Sinh viên Sư phạm Toán học, Khóa 43, Trường Đại học Cần Thơ*

[*]*Người chịu trách nhiệm về bài viết: Võ Nguyễn Minh Hiếu (email: hieub1609966@student.ctu.edu.vn)*

Thông tin chung:
Ngày nhận bài: 04/05/2019
Ngày nhận bài sửa: 27/07/2019
Ngày duyệt đăng: 30/10/2019

Title:
A linear time algorithm for the balanced {0,1}-knapsack problem

Từ khóa:
Bài toán cân bằng, bài toán xếp ba lô, quy hoạch động

Keywords:
Balance problem, dynamic programing, knapsack problem

ABSTRACT

In this paper, a variant of the balanced optimization problem, where the knapsack constraint is associated, is considered. To solve this problem, a special structure of the feasible solutions is explored. Based on this investigation, a dynamic approach is developed to solve the mentioned problem in linear time.

TÓM TẮT

Trong bài báo này, một biến thể của bài toán tối ưu cân bằng với ràng buộc có dạng xếp ba lô được nghiên cứu. Để giải quyết bài toán, một cấu trúc đặc biệt của tập các phương án chấp nhận được chỉ ra. Dựa vào đó, một thuật toán quy hoạch động được đề xuất để giải bài toán đã nêu trong thời gian đa thức.

Trích dẫn: Võ Nguyễn Minh Hiếu, Trần Thủ Lễ và Nguyễn Ngọc Đăng Duy, 2019. Thuật toán quy hoạch động cho bài toán xếp ba lô cân bằng {0,1}. Tạp chí Khoa học Trường Đại học Cần Thơ. 55(5A): 82-87.

1 MỞ ĐẦU

Tối ưu tổ hợp đóng một vai trò quan trọng trong lĩnh vực vận trù học với những ứng dụng đã được kiểm chứng trong thực tế. Một số hướng nghiên cứu quan trọng của tối ưu tổ hợp là bài toán quy hoạch tuyến tính (Kamarkar *et al.,* 1989; Danzig and Thapa, 2003), quy hoạch nguyên (Wolsey *et al.*, 1998), bài toán người bán hàng (Laporte and Martello, 1990; Arora *et al*., 1998; Applegate *et al.,* 2007; Zia *et al.,* 2017), bài toán xếp ba lô (Martello and Toth, 1990; Martello *et al*., 2000; Pisinger *et al.*, 2004), bài toán cây khung tối tiểu (Zhong *et al.,* 2015), bài toán vị trí (Kariv and Hakimi, 1979a, 1979b), bài toán ngược của bài toán vị trí (Nguyen and Vui, 2016; Nguyen *et al*., 2018; Nguyen, 2019; Nguyen *et al*., 2019) và một số bài toán tối ưu khác trên mạng lưới đồ thị (Danzig *et al*., 1959).

Trong bài báo này, bài toán tối ưu cân bằng với ràng buộc có dạng xếp ba lô được nghiên cứu. Mô hình của bài toán tối ưu cân bằng (balanced optimization problem) gồm có một tập các dự án, mỗi dự án nếu được triển khai cần đầu tư một lượng vốn tương ứng. Bài toán đặt ra đó là tìm một tập các dự án để triển khai sao cho độ lệch lớn nhất giữa các chi phí đầu tư là nhỏ nhất. Martello *et al.* (1984) đã chỉ ra rằng bài toán có thể giải quyết trong thời gian đa thức nếu bài toán tối ưu cổ chai (bottleneck optimization problem) tương ứng có thể giải được trong thời gian đa thức.

Một tình huống thực tế khác đó là bài toán xếp ba lô (knapsack problem). Cho trước *n* dự án, mỗi dự án gắn liền với một khoản lợi nhuận và chi phí. Câu hỏi đặt ra là, trong phạm vi ngân quỹ cho trước, nên đầu tư vào những dự án nào để lợi nhuận đạt được là lớn nhất. Bài toán này được gọi là bài toán xếp ba lô {0, 1} (Pisinger *et al.,* 2004).

Mặc dù bài toán xếp ba lô {0, 1} và bài toán tối ưu cân bằng là hai bài toán nổi tiếng trong lĩnh vực tối ưu tổ hợp và cấu trúc của chúng bằng cách này hay cách khác liên quan mật thiết đến nhau nhưng đến giờ mối quan hệ này vẫn chưa được nghiên cứu.

Bài báo này xem xét một bài toán với mô hình có dạng tổ hợp của hai bài toán trên, gọi là bài toán cân bằng với ràng buộc xếp ba lô. Ví dụ, xét mô hình đầu tư gồm n dự án. Mỗi dự án được cho tương ứng với một mức lợi nhuận a_i và một mức chi phí đầu tư c_i. Bài toán cân bằng với ràng buộc xếp ba lô sẽ chỉ ra cần đầu tư vào những dự án nào để mức chênh lệch lớn nhất giữa các chi phí đầu tư là nhỏ nhất trong khi vẫn giữ mức tổng lợi nhuận từ các dự án được đầu tư không nhỏ hơn một mức sàn định trước. Mô hình bài toán này giúp đảm bảo sự công bằng trong việc góp vốn đầu tư trong khi vẫn đảm bảo lợi ích tổng thể.

Bài báo được trình bày thành các phần như sau: phần 2 giới thiệu những khái niệm mở đầu và định nghĩa của bài toán xếp ba lô {0, 1} và bài toán tối ưu cân bằng nhằm giúp độc giả nắm được mô hình của các bài toán này cũng như ý nghĩa lịch sử của chúng; phần 3 là kết quả nghiên cứu chính của bài báo, trình bày thuật toán quy hoạch động cho bài toán cân bằng với ràng buộc xếp ba lô; ở cuối phần 3, một ví dụ số được trình bày nhằm minh họa chi tiết cho thuật toán; phần cuối cùng là kết luận về kết quả đạt được trong bài báo và đề xuất một số hướng nghiên cứu tiếp theo.

2 BÀI TOÁN XẾP BA LÔ {0, 1} VÀ BÀI TOÁN CÂN BẰNG

Trong phần này, mô hình toán học của bài toán xếp ba lô dạng {0, 1} và bài toán tối ưu cân bằng được giới thiệu và làm rõ ý nghĩa của chúng.

2.1 Bài toán xếp ba lô {0, 1}

Bài toán xếp ba lô (còn được biết đến với tên gọi bài toán cái túi) (knapsack problem) là một bài toán thuộc lĩnh vực tối ưu hóa tổ hợp. Bài toán này có nhiều phiên bản nhưng đều đề cập đến một vấn đề chung đó là cần chọn những món đồ nào để xếp vào trong một cái ba lô có giới hạn về khối lượng để mang đi sao cho một tiêu chí nào đó được tối ưu (như giá trị, nhu cầu sử dụng, …). Bài toán xếp ba lô dạng {0, 1} là bài toán xếp ba lô với số lượng mỗi đồ vật là bằng 1. Khi đó mỗi vật tương ứng với 1 nếu được chọn và tương ứng với 0 nếu không được chọn.

Bài toán được phát biểu dạng toán học như sau: Giả sử ta có n đồ vật: $1, 2, ..., n,$ mỗi đồ vật i có một giá trị c_i và một khối lượng a_i. Khối lượng tối đa mà ta có thể mang trong ba lô là b. Bài toán xếp balô {0,1} yêu cầu chọn ra một số đồ vật từ n đồ vật sao cho tổng khối lượng không vượt quá một khối lượng cho trước, đặt là b, và tổng giá trị của các đồ vật được chọn là lớn nhất.

Bài toán được phát biểu dưới dạng sau:

$$\text{Tìm } \max\left\{\sum_{i\in S} c_i\right\}, \text{ với } S \subset E,\ \sum_{i\in S} a_i \geq b\ .$$

Trong đó $E = \{1, ..., n\}$ là tập hợp tất cả các đồ vật ban đầu, S là tập các vật được chọn.

2.2 Bài toán tối ưu cân bằng

Chúng ta hãy bắt đầu với một tình huống như sau: Giả sử rằng một đại lý du lịch ở Hoa Kỳ đang lên kế hoạch để tổ chức một chương trình gồm các chuyến du lịch đến Châu Âu. Mỗi đại lý trong n đại lý ở Châu Âu đều đề nghị một chuyến du lịch đến mỗi nước trong n nước. Mỗi chuyến đi được đề nghị bởi đại lý i cho nước j tương ứng với một khoảng thời gian đã được lên kế hoạch trước. Lúc này, đại lý ở Hoa Kỳ mong muốn đưa ra một chương trình cho n chuyến du lịch theo cách như sau: mỗi chuyến đi được đề nghị tới một nước và được chọn từ danh sách các nước được đề nghị từ các đại lý ở Châu Âu. Bài toán đặt ra là tìm tập hợp các chuyến đi sao cho độ chênh lệch thời gian của chuyến đi dài nhất và chuyến đi ngắn nhất được cực tiểu hóa. Điều này là hoàn toàn cần thiết vì để đảm bảo sự công bằng về lợi ích của khách hàng và hạn chế tối đa sự so sánh khi lựa chọn chuyến đi.

Xét một mô hình khác gồm n công nhân và n công việc, trong đó mỗi công nhân được phân công để hoàn thành một công việc. Một câu hỏi được đưa ra đó là phải phân công công việc như thế nào để tổng thời gian hoàn thành tất cả các công việc là ít nhất. Tuy nhiên, trong thực tế có một vấn đề đặt ra là, nếu mức độ chênh lệch về thời gian hoàn thành công việc (do khối lượng công việc, độ khó công việc…) là quá lớn thì sự phân công của chúng ta có vẻ không công bằng đối với một số công nhân. Vì vậy, một câu hỏi khác được đặt ra đó là làm sao để bình đẳng hóa việc phân công.

Những ví dụ nêu trên với câu hỏi phải phân công như thế nào chính là bài toán gán (assignment problem), được nghiên cứu và giải quyết bởi thuật toán Hungarian (Kuhn, 1955, 1956). Một bài toán gán ứng với yêu cầu đạt được sự cân bằng nào đó được gọi là bài toán gán cân bằng. Bài toán gán cân bằng chính là một dạng đặc biệt của bài toán tối ưu cân bằng.

Bài toán tối ưu cân bằng (balanced optimization problem) lần đầu tiên được đề xuất bởi Martello *et al*. (1984). Các tác giả đã chỉ ra rằng bài toán cân bằng có thể giải được trong thời gian đa thức nếu bài toán tối ưu cổ chai tương ứng có thể giải trong thời gian đa thức. Hơn nữa, các tác giả còn đề xuất một thuật giải cho bài toán gán cân bằng với độ phức tạp $O\left(n^4\right)$.

Một cách tổng quát, bài toán tối ưu cân bằng đề cập đến các vấn đề có mô hình như sau: Giả sử rằng mỗi đối tượng được xem xét của bài toán đều được liên kết với một chi phí cho trước. Chúng ta mong muốn tìm được một phương án chấp nhận được sao cho độ chênh lệch giữa chi phí cao nhất và chi phí thấp nhất là bé nhất.

Ta phát biểu lại bài toán dưới dạng toán học. Giả sử ta được cho trước một tập hữu hạn E, và các chi phí $c_i, i \in E$ và họ F những tập con của E thỏa mãn những ràng buộc cho trước nào đó. Bài toán tối ưu cân bằng yêu cầu tìm một phương án chấp nhận được $S^* \in F$ là nghiệm của bài toán tối ưu:

$$\min\left\{\max\left\{c_i : i \in S\right\} - \min\left\{c_i : i \in S\right\} : S \in F\right\}.$$

3 BÀI TOÁN CÂN BẰNG VỚI RÀNG BUỘC XẾP BA LÔ VÀ THUẬT TOÁN

Trong phần này, bài toán tối ưu cân bằng với ràng buộc xếp ba lô được tập trung nghiên cứu. Xét một bài toán cụ thể với mô hình đầu tư kinh doanh đối với n dự án cho trước. Để đầu tư vào một dự án nào đó, nhà đầu tư cần bỏ ra một mức chi phí gọi là vốn đầu tư ban đầu tương ứng với dự án đó. Bài toán đặt ra đó là cần đầu tư vào những dự án nào sao cho mức chênh lệch lớn nhất giữa các chi phí đầu tư là nhỏ nhất, đồng thời, tổng lợi nhuận từ các dự án được đầu tư không thấp hơn một mức lợi nhuận đã lên kế hoạch trước. Việc đề xuất một phương án đầu tư như vậy là cần thiết nhằm giảm thiểu tối đa sự chênh lệch trong việc góp vốn đầu tư vào các dự án trong khi vẫn đảm bảo lợi ích tổng thể.

Cho một tập nền là tập các chỉ số $E = \left\{1, 2, ..., n\right\}$, trong đó mỗi chỉ số i được gắn tương ứng với một *chi phí* không âm c_i và một *trọng số (lợi nhuận)* không âm a_i. Mỗi tập con khác rỗng của E được gọi là một *phương án*. Trong bài toán tối ưu cân bằng, yêu cầu đặt ra là tìm một phương án S, làm cực tiểu hóa độ lệch lớn nhất của các chi phí tương ứng với tập S,

$$f(S) = \max_{i \in S} c_i - \min_{i \in S} c_i.$$

Một *phương án hữu hiệu* là một phương án S có tổng lợi nhuận không nhỏ hơn một số cho trước b, nghĩa là, $\sum_{i \in S} a_i \geq b$. Khi đó bài toán xếp ba lô cân bằng {0, 1} có thể được phát biểu: *Tìm một phương án hữu hiệu S làm cực tiểu hóa hàm $f(S)$*. Hay

$$\min\left\{f(S) = \max_{i \in S} c_i - \min_{i \in S} c_i : S \subset E, \sum_{i \in S} a_i \geq b\right\}$$

Phương án tối ưu là nghiệm của bài toán xếp ba lô cân bằng {0, 1}, hay có thể nói phương án tối ưu là một phương án hữu hiệu làm cực tiểu hàm $f(S)$, ký hiệu là S^*. Rõ ràng, bài toán là vô nghiệm nếu $\sum_{i \in E} a_i < b$. Do đó, từ đây trở về sau ta sẽ giả sử bài toán có nghiệm, nghĩa là, $\sum_{i \in E} a_i \geq b$. Hơn thế nữa, không mất tính tổng quát, ta có thể giả sử rằng $c_1 \leq c_2 \leq \ldots \leq c_n$. Mệnh đề sau đây chỉ ra cấu trúc đặc biệt của một lớp các phương án tối ưu.

Mệnh đề 2.1 Tồn tại một phương án tối ưu S^* của bài toán xếp ba lô cân bằng {0,1} sao cho nếu $r, s \in S^*, r \leq s$ thì $i \in S^*$ với mọi i thỏa $r \leq i \leq s$.

Chứng minh

Cho trước một phương án tối ưu S^*, ta xây dựng một phương án

$$S' := S^* \cup \{i \in E : r \leq i \leq s\}.$$

Khi đó, phương án S' là phương án hữu hiệu. Thật vậy, vì $S^* \subset S'$ nên

$$\sum_{i \in S'} a_i \geq \sum_{i \in S^*} a_i \geq b.$$

Theo giả sử, các giá trị c_i được sắp xếp theo thứ tự không giảm nên

$$f(S') = \max_{i \in S'} c_i - \min_{i \in S'} c_i = \max_{i \in S^*} c_i - \min_{i \in S^*} c_i = f(S^*)$$

.

Vậy S' cũng là phương án tối ưu cho bài toán xếp ba lô cân bằng *{0,1}*. □

Một phương án thỏa điều kiện trong Mệnh đề 2.1 được gọi là *phương án đầy đủ*, nghĩa là, trong phương án này các phần tử có các giá trị tương ứng nằm trong đoạn từ giá trị bé nhất đến giá trị lớn nhất đều được lựa chọn. Vậy rõ ràng mỗi phương án đầy đủ S sẽ tương ứng với một cặp chỉ số $\{r, s\}$ duy nhất sao cho $S = \{r, ..., s\}$.

Cho trước một chỉ số r, $1 \le r \le n$. Gọi s là chỉ số nhỏ nhất tương ứng với r sao cho $\sum_{i=r,..,s} a_i \ge b$. Một phương án đầy đủ ứng với cặp chỉ số $\{r, s\}$ như vậy sẽ được gọi là *phương án đầy đủ tối tiểu*. Mệnh đề sau đây chỉ ra rằng tồn tại một phương án tối ưu là một phương án đầy đủ tối tiểu.

Mệnh đề 2.2 Tồn tại một phương án tối ưu của bài toán xếp ba lô cân bằng $\{0,1\}$ sao cho phương án đó là phương án đầy đủ tối tiểu.

Chứng minh

Gọi $S^* = \{r, ..., s\}$ là một phương án tối ưu đầy đủ của bài toán xếp ba lô cân bằng $\{0,1\}$ và S^* không là một phương án đầy đủ tối tiểu. Ta sẽ chỉ ra rằng S^* có thể thu hẹp thành một tập S' sao cho S' vừa là một phương án tối ưu vừa là một phương án đầy đủ tối tiểu. Thật vậy, gọi s' là chỉ số nhỏ nhất sao cho $\sum_{i=r,..,s'} a_i \ge b$, $s' > r$. Do S^* không là một phương án đầy đủ tối tiểu nên $s' < s$. Khi đó, rõ ràng $f(S') = c_{s'} - c_r \le c_s - c_r = f(S^*)$. Vậy $S' := \{r, ..., s'\}$ là một phương án đầy đủ tối tiểu và tối ưu. □

Mệnh đề sau đây chỉ ra rằng các phương án đầy đủ tối tiểu được sắp 'thứ tự'.

Mệnh đề 2.3 Cho $S = \{r, ..., s\}$ và $S' = \{r', ...s'\}$ là các phương án đầy đủ tối tiểu sao cho $1 \le r \le r' \le n$. Khi đó $s' \ge s$.

Chứng minh

Chú ý rằng s là chỉ số nhỏ nhất sao cho $\sum_{i=r,..,s} a_i \ge b$. Ta giả sử phản chứng rằng $s' < s$. Khi đó, $\sum_{i=r}^{s'} a_i \ge \sum_{i=r'}^{s'} a_i \ge b$. Điều này mâu thuẫn với tính nhỏ nhất của chỉ số s. Mệnh đề được chứng minh. □

Trên cơ sở những mệnh đề vừa được chứng minh, chúng ta sẽ xây dựng một thuật toán để giải bài toán tối ưu cân bằng với ràng buộc xếp ba lô.

Ý tưởng thuật toán:

Cho bài toán xếp ba lô cân bằng {0, 1} với giả sử $\sum_{i=1,...,n} a_i \ge b$ và $c_1 \le c_2 \le ... \le c_n$. Với mỗi chỉ số r, $1 \le r \le n$, ta tìm một chỉ số s sao cho $S = \{r, ..., s\}$ là một phương án đầy đủ tối tiểu. Sau đó, với mỗi $r' > r$, ta tìm chỉ số s', $s \le s' \le n$ sao cho $S' = \{r', ..., s'\}$ cũng là một phương án đầy đủ tối tiểu, ... Cứ tiếp tục như vậy, ta sẽ tìm được tất cả các phương án đầy đủ tối tiểu, $S = \{r, ..., s\}$, $S' = \{r', ..., s'\}, \ldots$ Bằng cách chọn giá trị mục tiêu $f(S)$ nhỏ nhất trong số các phương án đầy đủ tối tiểu, ta thu được giá trị tối ưu của bài toán.

Thuật toán 2.4: Giải bài toán cân bằng {0,1}-knapsack.

Input: Một bài toán cân bằng {0,1}-knapsack thỏa mãn $\sum_{i=1,...,n} a_i \ge b$ và tập các chi phí đã được sắp $c_1 \le c_2 \le ... \le c_n$.

Set $s := 0$, $sum := 0$, $\min := +\infty$, $Sol := none$.

For $r = 1$ **to** n **do**

While $sum < b$ and $s < n$ do

$s := s + 1$

$sum := sum + a_s$

Endwhile

If $sum \ge b$ **do**

If $c_s - c_r < \min$ **do**

$\min := c_s - c_r$

$sol := \{r, s\}$

Endif

Else

If $s = n$ **do break**

Endif

$sum := sum - a_r$

Endfor

Output: Một giá trị cân bằng tối ưu (min) và một cặp chỉ số tối ưu (*Sol)*.

Do mỗi bước của thuật toán tốn thời gian hằng nên độ phức tạp của thuật toán là tuyến tính. Việc sắp thứ tự các chi phí c_i tốn thời gian $O(n \log n)$ (Hoare, 1962). Ta thu được kết quả về độ phức tạp của Thuật toán 2.4 trong định lí sau:

Định lí 2.5 Bài toán xếp ba lô {0, 1} cân bằng được giải trong thời gian $O(n\log n)$.

Ví dụ cụ thể sau đây nhằm minh họa cho Thuật toán 2.4.

Ví dụ 2.6 Cho dữ liệu đầu vào như sau: (dãy c_i đã được sắp thứ tự)

$c_1 = 1$	$c_2 = 3$	$c_3 = 4$	$c_4 = 6$	$c_5 = 8$	$c_6 = 11$
$a_1 = 5$	$a_2 = 2$	$a_3 = 3$	$a_4 = 5$	$a_5 = 4$	$a_6 = 6$

Với $b = 12$ và $n = 6$.

Ví dụ số minh họa cho thuật toán 2.4:

Input: Bài toán cân bằng {0,1}-knapsack thỏa mãn $\sum_{i=1,\dots,n} a_i \ge b$ và tập c_i đã được sắp $c_1 \le c_2 \le \dots \le c_n$.

Vòng lặp 1 ($r = 1$)

(While)

Vì $sum = 0 < b; s = 0 < n$ nên $s := s+1 = 1; sum := sum + a_1 = 5.$

Vì $sum = 5 < b; s = 1 < n$ nên $s := s+1 = 2; sum := sum + a_2 = 7.$

Vì $sum = 7 < b; s = 2 < n$ nên $s := s+1 = 3; sum := sum + a_3 = 10.$

Vì $sum = 10 < b; s = 3 < n$ nên $s := s+1 = 4; sum := sum + a_4 = 15.$

(Endwhile)

(If)

Vì $sum = 15 > b$ nên

Xét $c_s - c_r = c_4 - c_1 = 5 < \min$

Thực hiện: $\min := 5$; $Sol := \{1; 4\}$.

(Endif)

$sum := sum - a_1 = 10.$

Vòng lặp 2 ($r = 2$)

(While)

Vì $sum = 10 < b, s = 4 < n$ nên $s := s+1 = 5; sum := sum + a_5 = 14.$

(Endwhile)

(If)

Vì $sum = 14 > b$ nên

Xét $c_s - c_r = c_5 - c_2 = 5 = \min$

(Điều kiện không thỏa mãn, thuật toán chuyển đến bước tiếp theo)

(Endif)

$sum := sum - a_2 = 12.$

Vòng lặp 3 ($r = 3$)

Vì $sum = 12 = b$ nên lệnh **While** không update giá trị mới.

(If)

Vì $sum = 12 = b$ nên

Xét $c_s - c_r = c_5 - c_3 = 4 < \min$

Thực hiện: $\min := 4; Sol := \{3; 5\}$.

(Endif)

$sum := sum - a_3 = 9.$

Vòng lặp 4 ($r = 4$)

(While)

Vì $sum = 9 < b, s = 5 < n$ nên $s := s+1 = 6; sum := sum + a_6 = 15.$

(Endwhile)

(If)

Vì $sum = 15 > b$ nên

Xét $c_s - c_r = c_6 - c_4 = 5 > \min$

(Điều kiện không thỏa mãn, thuật toán chuyển đến bước tiếp theo)

(Endif)

$Sum := sum - a_4 = 10.$

Vòng lặp 5 ($r = 5$)

Vì $s = n$ nên lệnh **While** không cập nhật giá trị mới.

(If)

Vì $sum = 10 < b$ và $s = n$ nên thuật toán không cập nhật giá trị mới và dừng lại.

(Endif)

Ouput: $\min = 4$ và $Sol = \{3, 5\}$.

Vậy giá mục tiêu tối ưu của bài toán tối ưu cân bằng với dữ liệu cho trong Ví dụ 2.6 là $\min = 4$ ứng với phương án tối ưu $S^* = \{3, 4, 5\}$.

4 KẾT LUẬN

Trong bài báo này, nghiên cứu Bài toán tối ưu cân bằng với ràng buộc có dạng xếp ba lô cho thấy bài toán có thể giải trong thời gian $O(n \log n)$ do tác động của việc sắp thứ tự các chi phí. Một câu hỏi thú vị được đặt ra đó là liệu ta có thể cải thiện độ phức tạp của thuật toán về thời gian tuyến tính, tức là giải bài toán mà không thông qua việc sắp thứ tự. Hơn thế nữa, các kĩ thuật chứng minh trong bài báo có tiềm năng được tiếp tục nghiên cứu, phát triển, mở rộng và bổ sung,… để giải quyết những bài toán tối ưu cân bằng khác, ví dụ, bài toán gán cân bằng, bài toán cây bao trùm cân bằng, bài toán ngược của bài toán tối ưu cân bằng.

TÀI LIỆU THAM KHẢO

Arora, S., 1998. Polynomial time approximation schemes for Euclidean traveling salesman and other geometric problems. Journal of the ACM (JACM). 45(5): 753-782.

Applegate, D.L., Bixby, R.M., Chvátal, V. and Cook, W.J., 2007. The Traveling Salesman Problem: A Computational Study. Princeton University Press. Princeton, NJ, USA. 606 pages.

Dantzig, G.B. and Ramser, J.H., 1959. The Truck Dispatching Problem. Management Science. 6(1): 80-91.

Danzig, G. B. and Thapa, M. N., 2003. Linear Programing 2: Theory and Extensions. Springer – Verlag. The United States of America. 448 pages.

Hoare, C.A.R., 1962. Quicksort. The computer journal. 5(1): 10 -16.

Kamarkar, N., Adler, I., Resende, M. and Geraldo, V., 1989. An Implementation Of Karmarkar's Algorithm For Linear Programming. Mathematical Programming. 44(1): 297-335.

Kariv, O. and S.L. Hakimi, S.L., 1979a. An algorithmic approach to network location problems, I. The p-centers, SIAM Journal on Applied Mathematics. 37: 513-538.

Kariv, O. and Hakimi, S.L., 1979b. An algorithmic approach to network location problems, II. The p-medians. SIAM Journal on Applied Mathematics. 37: 536-560.

Kuhn, H. W., 1955. The Hungarian Method for the Assignment Problem. Naval Research Logistics Quarterly. 2: 83-97.

Kuhn, H. W., 1956. Variants of the Hungarian method for assignment problems. Naval Research Logistics Quarterly. 3: 253-258.

Laporte, G. and Martello, S., 1990. The selective travelling salesman problem. Discrete Applied Mathematics. 26(2-3):193-207.

Martello, S., Pulleyblank, W.R., Toth, P. and De Werra, D., 1984. Balanced Optimization Problems. Operations Research Letters. 3: 275-278.

Martello, S. and Toth, P., 1990. Knapsack problems: algorithms and computer implementations. John Wiley & Sons, Inc. New York, NY, USA. 296 pages.

Martello, S., Pisinger, D. and Toth, P., 2000. New trends in exact algorithms for the 0–1 knapsack problem. European Journal of Operational Research. 123(2): 325-332.

Nguyen, K. and Vui, P., 2016. The inverse p-maxian problem on trees with variable edge lengths. Taiwanese Journal of Mathematics, 20(6): 1437-1449.

Nguyen, K.T., Nguyen-Thu, H. and Hung, N.T., 2018. On the complexity of inverse convex ordered 1-median problem on the plane and on tree networks. Mathematical Methods of Operations Research, pp.1-13.

Nguyen, K.T., 2019. The inverse 1-center problem on cycles with variable edge lengths. Central European Journal of Operations Research, 27(1): 263-274.

Nguyen, K.T., Hung, N.T., Nguyen-Thu, H., Le, T.T. and Pham, V.H., 2019. On some inverse 1-center location problems. Optimization, 68(5): 999-1015.

Pisinger, D., Pferschy, U. and Kellerer, H., 2004. Knapsack Problems. Springer.

Wolsey, L.A., 1998. Integer Programming. John Wiley and Sons Inc, New York, United States.

Zhong, C., Malinen, M., Miao, D. and Fränti, P., 2015. A fast minimum spanning tree algorithm based on K-means. Information Sciences. 295: 1-17.

Zia, M., Cakir, Z. and Seker, D.Z., 2017. A New Spatial Approach for Efficient Transformation of Equality-Generalized TSP to TSP. Free and Open Source Software for Geospatial (FOSS4G) Conference Proceedings. 17(5): 14-22.