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.jsi.2017.006

# GIẢI THUẬT ƯỚC LƯỢNG SỐ CỤM DỮ LIỆU CẢI TIẾN CHO TẬP DỮ LIỆU LỚN

Dương Văn Hiếu, Trần Huy Long và Phạm Ngọc Giàu
*Khoa Công nghệ Thông tin, Trường Đại học Tiền Giang*

***Thông tin chung:***
*Ngày nhận bài: 15/09/2017*
*Ngày nhận bài sửa: 10/10/2017*
*Ngày duyệt đăng: 20/10/2017*

***Title:***
*A revised cluster number estimation algorithm for big datasets*

***Từ khóa:***
*Cây phủ tối thiểu, đồ thị tối ưu, tập dữ liệu lớn, tế bào hóa tập dữ liệu, ước lượng số cụm dữ liệu*

***Keywords:***
*Big datasets, Cell-MST-based, Cluster number estimation, Weighted-Cell-MST-based*

**ABSTRACT**

*This paper presents a revised version of a cluster number estimation algorithm for big datasets. This algorithm was designed to work on a standard personal computer. This is an improvemennt of the Cell-MST-Based cluster number estimation algorithm by appying weighted distance instead of using the Euclidean distance. This new algorithm was named Weighted-Cell-MST-based cluster number estimation algorithm. This revised version can provide more stable results compared to its former version when testing the same datasets in the same environment.*

**TÓM TẮT**

*Bài báo này trình bày một giải thuật ước lượng số cụm dữ liệu cải tiến dùng để ước lượng số cụm dữ liệu của tập dữ liệu lớn. Giải thuật được thiết kế chạy trên máy tính cá nhân có cấu hình cơ bản. Đây là một sự cải tiến của giải thuật ước lượng số cụm Cell-MST-Based bằng cách áp dụng khoảng cách có trọng số thay cho khoảng cách Euclid. Thuật toán cải tiến được đặt tên là Weighted-Cell-MST-based cluster number estimation algorithm. Thuật toán cải tiến cho kết quả ổn định hơn so với thuật toán ban đầu khi xét trên cùng các tập dữ liệu và trong cùng một điều kiện thực nghiệm.*

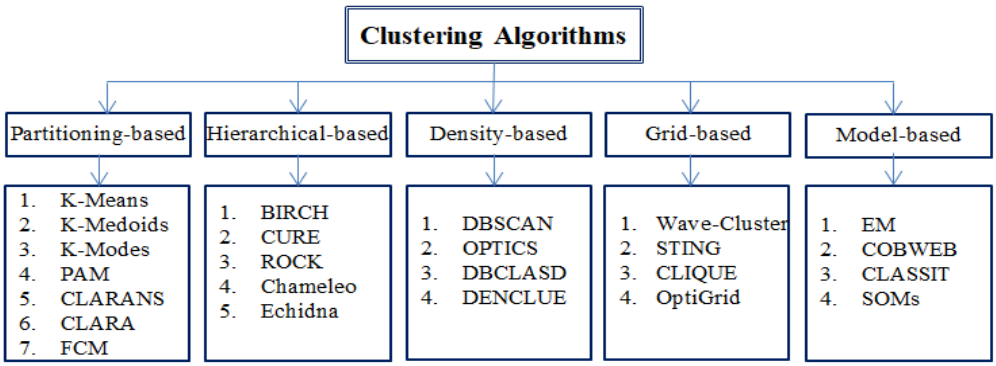
Trích dẫn: Dương Văn Hiếu, Trần Huy Long và Phạm Ngọc Giàu, 2017. Giải thuật ước lượng số cụm dữ liệu cải tiến cho tập dữ liệu lớn. Tạp chí Khoa học Trường Đại học Cần Thơ. Số chuyên đề: Công nghệ thông tin: 42-53.

## 1 GIỚI THIỆU

Trong thời đại cuộc cách mạng công nghiệp lần thứ tư, hầu hết dữ liệu đều được số hóa. Quá trình số hóa dữ liệu đã tạo ra một cuộc cách mạng về dữ liệu mà người ta thường gọi là big data hay dữ liệu lớn. Big data không chỉ là dữ liệu lớn về dung lượng mà còn đa dạng về cấu trúc, định dạng, nguồn phát sinh, mức độ thay đổi,... Trong lĩnh vực khai khoáng dữ liệu và khoa học dữ liệu, phân cụm dữ liệu được xem là công cụ quan trọng trong phân tích và xử lý các tập dữ liệu lớn và được ứng dụng nhiều trong các lĩnh vực kinh doanh, công nghệ, khoa học, giáo dục,... (Romero và Ventura, 2013; Kokol, 2015). Dựa vào phương pháp phân cụm, các giải thuật phân cụm dữ liệu được chia thành 5 nhóm như trong Hình 1 (Fahad *et al.*, 2014).

Hiện tại, có nhiều định nghĩa khác nhau về tập dữ liệu lớn, một định nghĩa được nhiều người chấp nhận và sử dụng là "tập dữ liệu lớn là tập dữ liệu mà chúng ta không thể xử lý bằng những phương pháp tuyền thống". Các tập dữ liệu mà Jesus *et al.* (2017) đã sử dụng trong các thí nghiệm có số lượng từ 1.025.000 mẫu tin đến 11.000.000 cũng được xem là các tập dữ liệu lớn.

Trong bài báo này, các tập dữ liệu được sử dụng để thí nghiệm có số lượng từ 300.000 mẫu tin đến 24.895.830 mẫu tin. Trong hoàn cảnh tối ưu hóa các thuật toán thì một số tập dữ liệu được sử dụng trong bài báo này đã được chấp nhận là các tập dữ liệu lớn (Van Hieu and Phayung, 2016).




**Hình 1: Phân nhóm các giải thuật phân cụm**

Các thuật toán phân cụm thuộc dạng partitioning clustering được biết đến nhiều là K-Means, K-Medoids, K-Modes, PAM, CLARANS, CLARA, FCM, CSO, PSO. Các thuật toán này phân chia một tập dữ liệu $\mathbf{X} = (\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N)$ thành $K < N$ cụm $\mathbf{C} = (\mathbf{c}_1, \mathbf{c}_2, \dots, \mathbf{c}_K)$ có thể giao nhau hoặc không giao nhau. Vì K-Means là một thuật toán phân cụm dữ liệu rất thông dụng nên bày báo này chọn thuật toán K-Means để làm nền tảng cho giải thuật ước lượng số cụm dữ liệu.

Gọi *dist* $(\mathbf{x}_i, \mathbf{x}_j)$ là khoảng cách từ đối tượng $\mathbf{x}_i$ đến đối tượng $\mathbf{x}_j$. Thuật toán K-Means có thể được biểu diễn như sau (Barioni *et al.*, 2014).

**Thuật toán 1:** K-Means $(\boldsymbol{X}, N)$

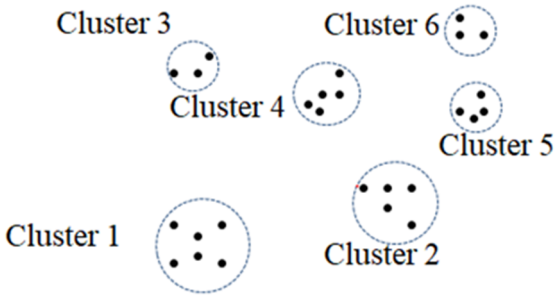
**Input:** Tập dữ liệu $\mathbf{X}$ có $N$ đối tượng trong không gian $D$ chiều, số lượng nhóm cần phân chia là $K$

**Output**: Tâm của $K$ nhóm và danh sách các đối tượng dữ liệu thuộc vào từng nhóm.

**Begin**

1. Khởi tạo $K$ tâm $\mathbf{Z} = (\mathbf{z}_1, \mathbf{z}_2, \dots, \mathbf{z}_K)$ cho $K$ nhóm.
2. Gán các đối tượng dữ liệu $\mathbf{x}_i = (x_{i1}, x_{i2}, \dots, x_{iD})$ với $i = 1, \dots, N$ vào nhóm $\mathbf{c}_j^*$ sao cho $dist(\mathbf{x}_i, \mathbf{z}_j^*) = \min(dist(\mathbf{x}_i, \mathbf{z}_j))$ với $i = 1, \dots, N$; $j = 1, \dots, K$; $\mathbf{z}_j$ là tâm của nhóm $\mathbf{c}_j$ và $dist(\mathbf{x}_i, \mathbf{z}_j)$ là khoảng cách từ $\mathbf{x}_i$ tới $\mathbf{z}_j$.
3. Tính lại tọa độ tâm của các nhóm $\mathbf{z}_j = \frac{1}{\|\mathbf{c}_j\|} \sum_{\mathbf{x}_i \in \mathbf{c}_j} \mathbf{x}_i$ với $j = 1, \dots, K$.
4. Kiểm tra điều kiện dừng đã được định nghĩa trước
   - Nếu chưa đạt điều kiện dừng thì lặp lại bước 2;
   - Ngược lại, trả về tâm của $K$ nhóm và danh sách các điểm thuộc từng nhóm.

**End**




**Hình 2: Ví dụ về kết quả phân cụm**

Để phân chia một tập dữ liệu $\mathbf{X}$ có $N$ phần tử thành $K$ nhóm thì các giải thuật phân nhóm cần được cung cấp giá trị của $K$ trước khi thực hiện việc phân nhóm.

Đối với các tập dữ liệu nhỏ hoặc có từ 1 đến 2 thuộc tính thì việc ước lượng số cụm dữ liệu có thể được thực hiện thông qua các công cụ trực quan hóa dữ liệu. Đối với các tập dữ liệu lớn và có nhiều hơn 2 thuộc tính thì có thể ước lượng cố cụm thông qua kiến thức chuyên gia về lĩnh vực của dữ liệu. Tuy nhiên, để ước lượng số cụm chính xác thì cần đến các giải thuật ước lượng số cụm dữ liệu.

Có nhiều thuật toán ước lượng số cụm được đề xuất: sử dụng ma trận khoảng cách giữa các đối tượng (Similarity matrix) (Shao *et al,.*2013), sử dụng lý thuyết tập thô (Decision-theoretic rough set) (Yu *et al.*, 2014), sử dụng phương pháp thống kê (Weighted gap statistic) (Yan and Ke, 2007), sử dụng cây phủ tối thiểu (Minimum-cost spanning trees) (Zhong *et al.*, 2015)... Tuy nhiên, các phương pháp đó khó áp dụng để hiện thực việc ước lượng số nhóm của các tập dữ liệu lớn khi thực hiện trên các máy tính cá nhân có cấu hình chuẩn vì lý do bộ nhớ khả dụng của máy tính cá nhân không thể đáp ứng được yêu cầu cấp phát và lưu trữ dữ liệu khi tính toán.

Ví dụ: Cần ước lượng số cụm dữ liệu cho một tập dữ liệu **X** có $N$=2.000.000 đối tượng dữ liệu. Trong trường hợp sử dụng kiểu dữ liệu *float* có kích thước 4 byte để lưu trữ giá trị số thập phân và kiểu dữ liệu *unsigned char* có kích thước 1 byte để lưu trữ giá trị số nguyên nhỏ. Yêu cầu về dung lượng bộ nhớ để lưu trữ dữ liệu cần thiết liên quan đến quá trình ước lượng số cụm có thể được tính như Bảng 1.

**Bảng 1: Minh họa yêu cầu dung lượng bộ nhớ với N=2.000.000**

| Phương pháp | Công thức tính dung lượng (Byte) | Kích thước bộ nhớ yêu cầu (Giga byte) |
|---|---|---|
| Ma trận kề (Shao *et al*., 2013) | $N^2 \times 1$ | 3.725,29 |
| Ma trận tương đương (Shao *et al*. 2013) | $N^2 \times 4$ | 14.901,16 |
| Lý thuyết tập thô (Yu *et al.*, 2014). | $N^2 \times 2 \times 4$ | 29.802,32 |
| Cây phủ tối thiểu (Zhong *et al.*, 2015) | $(N^2/2) \times 4$ | 7.450,58 |

Để cài đặt được thuật toán ước lượng số cụm dữ liệu theo phương pháp cây phủ tối thiểu thì cần phải giảm số đối tượng trong tập dữ liệu đến một giá trị nhất định tùy vào dung lượng khả dụng của bộ nhớ RAM.

Mặt khác, có nhiều giải thuật ước lượng số cụm dữ liệu không đòi hỏi phải sử dụng dung lượng bộ nhớ lớn nhưng lại tốn rất nhiều thời gian khi thực hiện việc phân cụm như thuật toán QEM (Kolesnikov *et al.*, 2015).

Thuật toán ước lượng số cụm dữ liệu cho tập dữ liệu lớn dựa trên sự kết hợp giữa phương pháp cây phủ tối thiểu và phương pháp tế bào hóa (Cell-MST-Based) (Van Hieu và Phayung, 2015) được đề xuất và được xem như một giải pháp tốt cho việc ước lượng số cụm của các tập dữ liệu lớn được thiết kế để thực thi trên các máy tính cá nhân có cấu hình chuẩn (Texas Tech University, 2015).

Kết quả thực nghiệm cho thấy thuật toán ước lượng số cụm dữ liệu Cell-MST-based cho kết quả tốt hơn rất nhiều khi so sánh với các thuật toán khác khi áp dụng cho các tập dữ liệu lớn. Tuy nhiên, khi áp dụng cho một vài tập dữ liệu đặc biệt thì kết quả có thể không được ổn định. Do đó, bài báo này trình bày một cách cải tiến thuật toán ước lượng số cụm dữ liệu Cell-MST-Based bằng cách áp dụng khái niệm khoảng cách có trọng số thay vì sử dụng khoảng cách Euclid. Thuật toán cải tiến được đặt tên là Weighted-Cell-MST-Based cluster number estimation algorithm.

Đóng góp của bài báo này là đề xuất khái niệm khoảng cách có trọng số giữa 2 đối tượng, đồng thời áp dụng khoảng cách có trọng số để cải tiến thuật toán ước lượng số cụm dữ liệu Cell-MST-Based để nhận được kết quả ổn định hơn cho một số tập dữ liệu đặc biệt.

Bài báo được cấu trúc thành 5 phần: Phần thứ hai là thuật toán ước lượng số cụm Cell-MST-based, phần thứ ba là đề nghị giải thuật cải tiến, phần thứ tư là kết quả thực nghiệm, phần cuối cùng là kết luận và hướng phát triển.

## 2 THUẬT TOÁN ƯỚC LƯỢNG SỐ CỤM CELL-MST-BASED

Phần này trình bày những nội dung chủ yếu về cơ sở lý thuyết có liên quan đến thuật toán ước lượng số cụm dữ liệu Cell-MST-Based. Nội dung trình bày gồm phương pháp biến đổi tập dữ liệu lớn **X** gồm $N$ đối tượng thành tập dữ liệu nhỏ gồm có $M^D$ tế bào, cách xây dựng đồ thị tối ưu từ tập tế bào, ước lượng số cụm dữ liệu bằng cây phủ tối thiểu, các chỉ số dùng để đánh giá kết quả phân cụm dữ liệu.

### 2.1 Quy ước ký hiệu

Để tránh hiểu nhầm và nhất quán các ký hiệu trong khi trình bày các nội dung tiếp theo của bài báo này, chúng tôi thống nhất sử dụng ký hiệu như sau:

– Ký tự thường in nghiên: sử dụng để ký hiệu cho biến hoặc giá trị vô hướng. Ví dụ các ký hiệu $i, j, k$ là các biến.

– Ký tự thường in đậm: sử dụng để ký hiệu cho vector hoặc đối tượng có cấu trúc. Ví dụ **x** là vector hoặc là đối tượng dữ liệu.

– Ký tự hoa in nghiên: sử dụng để ký hiệu cho hằng số. Ví dụ $N, D, M$ là hằng số.

– Ký tự hoa in đậm: sử dụng để ký hiệu cho ma trận hoặc tập hợp các đối tượng có cấu trúc. Ví dụ **X** là ma trận.

Hơn nữa, trong trường hợp ký hiệu là một chuỗi các ký tự thì cũng được quy ước giống như 1 ký tự.

### 2.2 Chỉ số so sánh kết quả phân cụm

Để xác định kết quả phân cụm dữ liệu là tốt hay chưa tốt, người ta thường sử dụng nhiều chỉ số khác nhau. Đối với bài toán ước lượng số cụm dữ liệu kết hợp với việc phân cụm thì các chỉ số thường được sử dụng là chỉ số Dunn, Davies-Boudlin, Krzanowski-Lai, Calinski-Harabasz, Intra-Inter. Khi so sánh 2 kết quả phân cụm với nhau; kết quả nào có giá trị Dunn, Krzanowski-Lai, Calinski-Harabasz lớn hơn thì tốt hơn; kết quả nào

có giá trị Davies-Boudlin, Intra-Inter nhỏ hơn thì tốt hơn (Starczewski and Krzyak, 2015).

Giá trị của các chỉ số được tính bằng các biểu thức (1) đến (5).

$$Dunn = \underbrace{min}_{1\le i\le K}\left\{\underbrace{min}_{\substack{1\le j\le K\\ j\neq i}}\left\{\frac{d(C_i,C_j)}{\underbrace{max}_{1\le t\le K}\{\Delta(C_t)\}}\right\}\right\} \quad (1)$$

$$Davies - Boudlin = \frac{1}{K}\sum_{i=1}^{K}\underbrace{max}_{i\neq j}\left\{\frac{S(C_i)+S(C_j)}{S(C_i,C_j)}\right\} \quad (2)$$

$$DIFF(K) = (K-1)^{2/p}\times WSS(K-1) - K^{\frac{2}{p}}\times WSS(K) \quad (3)$$

$$Krzanowski - Lai = \left|\frac{DIFF(K)}{DIFF(K+1)}\right| \quad (4)$$

$$Calinski - Harabasz = \frac{BSS(K-1)\times(N-K)}{WSS(K)\times(K-1)} \quad (5)$$

Tùy vào đặc điểm của các tập dữ liệu dùng để phân cụm hoặc ước lượng số cụm dữ liệu mà người ta chọn các chỉ số phù hợp.

### 2.3 Tế bào hóa tập dữ liệu lớn

Đây là thuật toán biến đổi tập dữ liệu lớn thành tập tế bào có kích thước nhỏ hơn rất nhiều so với tập dữ liệu ban đầu đã được đề xuất cho các thuật toán xử lý tập dữ liệu lớn (Van Hieu và Phayung, 2015).

#### *2.3.1 Ánh xạ đối tượng dữ liệu vào tế bào*

Một tế bào dữ liệu được định nghĩa như một nhóm các đối tượng dữ liệu gần giống nhau.

Gọi $\mathbf{X} = (\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N)$ là tập dữ liệu gốc có $N$ đối tượng trong không gian $D$ chiều, $\mathbf{v}_{min} = (v_{\min 1}, v_{\min 2}, \dots, v_{\min D})$ là giá trị nhỏ nhất của các thuộc tính của các đối tượng dữ liệu, $\mathbf{v}_{max} = (v_{\max 1}, v_{\max 2}, \dots, v_{\max D})$ là giá trị lớn nhất của các thuộc tính của các đối tượng dữ liệu. Mỗi đoạn $[v_{\min j},\ v_{\max j}]$ được chia thành $M$ đoạn để tạo thành các vector $\mathbf{w} = (w_1, w_2, \dots, w_D)$ và $\boldsymbol{m}_j = (m_{j0}, m_{j1}, m_{j2}, \dots, m_{jM})$ với $j = 1, \dots, D$ và $m_{j0} = v_{\min j}$. Giá trị của $v_{\min j}$, $v_{\max j}$, $w_j$, $m_{jk}$ được tính bằng các biểu thức (7) đến (10).

$$\mathbf{X} = \begin{pmatrix} x_{11} & x_{12} & \dots & x_{1D} \\ x_{21} & x_{22} & \dots & x_{2D} \\ \dots & \dots & \dots & \dots \\ x_{N1} & x_{N2} & \dots & x_{ND} \end{pmatrix} \quad (6)$$

$$v_{min\,j} = min(x_{ij}) \text{ với } = 1, \cdots, N\ ; j = 1, \cdots, D \quad (7)$$

$$v_{max\,j} = max(x_{ij}) \text{ với } i = 1, \cdots, N; j = 1, \cdots, D \quad (8)$$

$$w_j = \frac{v_{max\,j} - v_{min\,j}}{M} \text{ với } j = 1, \cdots, D \quad (9)$$

$$m_{j\,k} = v_{min\,j} + j \times w_j \text{ với} j = 1, \cdots, D;\ k = 1, \cdots, M \quad (10)$$

Thuật toán ánh xạ được biểu diễn như sau:

**Thuật toán 2:** Ánh xạ($\boldsymbol{X}, \boldsymbol{m}_j$ với $j = 1, \dots, D$)

**Input**: Tập dữ liệu $\mathbf{X}$, $D$ vector $\mathbf{m}_j$ *với* $j = 1, \dots, D$

**Output**: Tập đối tượng và địa chỉ ánh xạ $\mathbf{X}'$

**Begin**

```
X' ← X
For x'_i ∈ X' với i = 1, ..., N
  For j = 1, ..., D
    Stop ←0; k←0
    While (Stop < 1) và (k < M)
      If (x'_ij ≤ m_jk)
        x'_ij ← m_jk
        Stop ← 1
      Else
        k ← k + 1
      End if
    End while
  End for
End for
```

**End**

#### *2.3.2 Chuyển tập dữ liệu thành tập tế bào*

Mục đích của bước này là lấy địa chỉ của các tế bào. Mỗi tế bào là một nhóm các đối tượng dữ liệu. Gọi $C$ là dung lượng khả dụng của bộ nhớ RAM, $n$ là kích thước của kiểu dữ liệu dùng để lưu trữ khoảng cách giữa các tế bào, giá trị của $M$ được xác định bằng biểu thức (11).

$$\frac{M^D\times(M^D-1)}{2} \le \frac{C}{n} \quad (11)$$

Đặt $X = M^D$, giá trị của $X$ và $M$ được tính bằng các công thức (12) và (13).

$$X = \frac{-1+\sqrt{1+8C/n}}{2} \quad (12)$$

$$M = \lfloor X^{1/D} \rfloor \quad (13)$$

Thuật toán chuyển đổi tập dữ liệu thành tập tế bào được mô tả như sau:

**Thuật toán 3:** Tế bào ($\boldsymbol{X}, n$)

**Input:** Tập dữ liệu $\mathbf{X}$ có $N$ đối tượng trong không gian $D$ chiều, $n$ là kích cỡ của kiểu dữ liệu dùng để lưu giá trị khoảng cách.

**Output**: Tập hợp các tế nào $\mathbf{O}$ và véc tơ $\mathbf{w}$.

**Begin**

1. Lấy dung lượng khả dụng của bộ nhớ RAM, gọi là $C$.
2. Giải phương trình (12), (13) để tìm giá trị của $M$.

3. Tính vector $\mathbf{v}_{min}$ và $\mathbf{v}_{max}$ dựa vào biểu thức (7) và (8).
4. Tính vector **w** dựa vào biểu thức (9).
5. Tính vector $\mathbf{m}_j$ với j= $1, \dots, D$ dựa vào biểu thức (10).
6. Gọi hàm ánh xạ($\boldsymbol{X}, \boldsymbol{m}_j$ với $j = 1, \dots, D$).
7. Tính giá trị các thuộc tính của tế bào khác rỗng, gọi là **O**.
8. Trả về **O** và **w**.

**End**

### 2.4 Xây dựng đồ thị tối ưu và ước lượng số cụm dữ liệu

Sau khi có được tập hợp các tế bào, việc xây dựng đồ thị tối ưu để các định số lượng cụm được thực hiện qua 3 bước:

*2.4.1 Tính khoảng cách giữa các tế bào*

Gọi $N'$ là số lượng tế bào trong tập tế bào **O**. Một ma trận vuông có kích cỡ $N' \times N'$ được xây dựng để lưu trữ khoảng cách giữa các tế bào. Vì đây là ma trận đối xứng nên chỉ cần xây dựng ½ ma trận có dung lượng bộ nhớ được cấp phát thật sự là $\frac{(N\prime \times N\prime)}{2} \times 4$ byte.

*2.4.2 Xây dựng đồ thị tối ưu*

Mục đích của việc xây dựng đồ thị tối ưu là hạn chế việc cấp phát ô nhớ và lưu trữ những giá trị không cần thiết cho quá trình ước lượng số cụm dữ liệu. Thuật toán xây dựng đồ thị tối ưu từ tập tế bào có thể được tóm tắt như sau:

**Thuật toán 4:** Đồ thị tối ưu($\boldsymbol{O}$)

**Input:** Tập hợp các tế bào **O**.
**Output**: Đồ thị tối ưu $\mathbf{G}_0$ cùng với giá trị cầm canh $T_0$.
**Begin**

1. Tính ma trận khoảng cách giữa các tế bào gọi là **D**.
2. Tính trung bình và độ lệch của các khoảng cách gọi là $mean(\mathbf{D})$ and $std(\mathbf{D})$.
3. Nếu $std(\mathbf{D}) \leq 1$ thì
   $T_0 \leftarrow mean(\mathbf{D})$
   Ngược lại
   $T_0 \leftarrow min(mean(\mathbf{D}), std(\mathbf{D}))$.
4. Xây dựng đồ thị với các cạnh có độ dài $\leq T_0$ gọi là $\mathbf{G}_0$.
5. Xây dựng cây phủ tối thiểu từ $\mathbf{G}_0$, gọi là $\mathbf{T}_0$.
6. $T_0 = \max$(cạnh của $\mathbf{T}_0$).
7. Xóa các cạnh $> T_0$ từ đồ thị $\mathbf{G}_0$
8. Trả về $\mathbf{G}_0$ và $T_0$

**End**

Sau khi xóa bỏ các cạnh có độ dài lớn hơn $T_0$ thì đồ thị $\mathbf{G}_0$ được gọi là đồ thị tối ưu. Đồ thị tối ưu có thể là một đồ thị liên thông và cũng có thể là một đồ thị gồm nhiều thành phần liên thông.

*2.4.3 Ước lượng số cụm dữ liệu*

Bước thứ 3 của thuật toán ước lượng số cụm dữ liệu Cell-MST-based là ước lượng số cụm dữ liệu kết hợp với phân cụm bằng thuật toán K-Means.

Ban đầu, xem đồ thị tối ưu $\mathbf{G}_0$ là một rừng của các cây gọi là $\mathbf{F}_0$. Thực hiện việc lặp với $i = 1, \dots, \lfloor\sqrt{N}\rfloor$.

Tại bước lặp thứ $i$, xóa các cạnh có độ dài lớn hơn $T_i$ để tạo thành rừng $\mathbf{F}_i$. Gọi $K_i$ là số lượng cây của rừng $\mathbf{F}_i$. $K_i$ cũng được xem là số cụm các tế bào và tương đương với số cụm dữ liệu của tập dữ liệu **X**.

Tâm của từng cụm tế bào được xác định thông qua giá trị các thuộc tính của tế bào, địa chỉ này được ánh xạ ngược thành tâm của các cụm dữ liệu. Thuật toán phân cụm K-Means được sử dụng để phân cụm tập dữ liệu ban đầu **X**, các chỉ số Dunn, Davies-Boudlin, Intra-Inter được tính tương ứng với từng giá trị $K_i$.

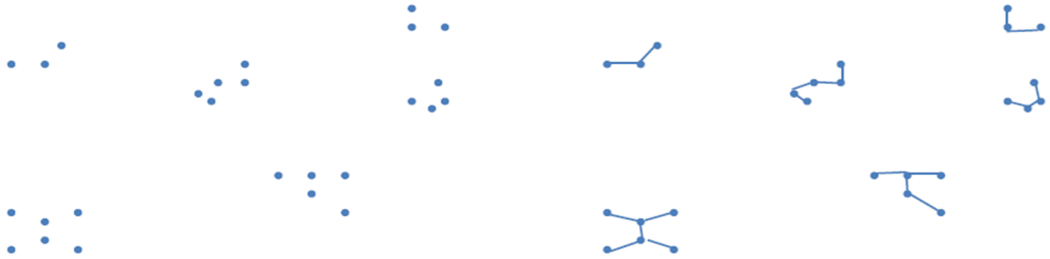
Giá trị của $T_i$ được xác định bằng biểu thức (14) với $i = 1, \dots, \lfloor 1/\alpha \rfloor$ và 0<α<1.

Tại bước $i$, gọi $\mathbf{z}^0{}_k = (z^0{}_{k1}, z^0{}_{k2}, \dots, z^0{}_{kD})$ với $k = 1, \dots, K_i$ là tâm của các cây trong rừng cây $\mathbf{F}_i$, và $\mathbf{z}_k = (z_{k1}, z_{k2}, \dots, z_{kD})$ với $k = 1, \dots, K_i$ là tâm ban đầu của các nhóm khi sử dụng thuật toán K-Means trên tập dữ liệu gốc **X**. Giá trị của tâm $\mathbf{z}_k$ được tính bằng biểu thức (15).

$$T_i = T_0 - i \times \alpha \quad (14)$$

$$z_{kj} = z^0{}_{kj} \times w_j + v_{min\,j} \text{ với } j = 1 \text{ tới } D \quad (15)$$

Khi $i = \lfloor\sqrt{N}\rfloor$, tất cả các giá trị chỉ số Dunn, Davies-Boudlin, Intra-Inter tương ứng với từng giá trị của $K_i$ được so sánh. Giá trị $K_i{}^*$ được chọn là giá trị có ít nhất 2 chỉ số tốt nhất trong 3 chỉ số. Trong trường hợp cả 3 chỉ số không giống nhau, giá trị $K_i$ được chọn tương ứng với trường hợp có giá trị MSE thấp nhất.



**a) Số cụm dữ liệu** **b) Số cây của rừng**

**Hình 3: Ví dụ mối tương quan giữa số cụm dữ liệu và số cây**

## 3 THUẬT TOÁN ƯỚC LƯỢNG SỐ CỤM DỮ LIỆU CẢI TIẾN WEIGHTED-CELL-MST-BASED

Để cải thiện kết quả của thuật toán ước lượng số cụm dữ liệu Cell-MST-Based khi áp dụng cho một số tập dữ liệu đặc biệt, nghiên cứu đề xuất sử dụng khoảng cách có trọng số thay gì khoảng cách Euclid, tiếp theo là thay đổi các thuật toán xây dựng đồ thị tối ưu, ước lượng số cụm cho phù hợp với khái niệm khoảng cách có trọng số.

### 3.1 Khoảng cách có trọng số

Khoảng cách Euclid dùng để đo khoảng cách giữa 2 điểm không có trọng số, trong khi tế bào dữ liệu có trọng số. Trọng số của tế bào là tổng trọng số của các đối tượng tạo nên tế bào nên về bản chất thì tế bào có trọng số lớn hơn sẽ mang lại một lượng thông tin khác hơn so với tế bào có trọng số nhỏ hơn.

Gọi $w_i$ là trọng số của tế bào $i$; $\mathbf{o}_i = (o_{i1}, o_{i2}, \cdots o_{iD}, w_i)$, $\mathbf{o}_j = (o_{j1}, o_{j2}, \cdots o_{jD}, w_j)$, $\mathbf{o}_t = (o_{t1}, o_{t2}, \cdots o_{tD}, w_t)$ là 3 tế bào khác nhau với $o_{ik}$ là giá trị của thuộc tính $k$ của các tế bào; Edist() là hàm tính khoảng cách Euclid giữa 2 tế bào, Wdist() là hàm tính khoảng cách có trọng số giữa 2 tế bào. Khoảng cách có trọng số được định nghĩa sao cho thỏa 3 tính chất sau:

$0 \le Wdist(\mathbf{o}_i, \mathbf{o}_j) < Edist(\mathbf{o}_i, \mathbf{o}_j)$.

$Wdist(\mathbf{o}_i, \mathbf{o}_j) = Wdist(\mathbf{o}_j, \mathbf{o}_i)$.

Nếu $Edist(\mathbf{o}_t, \mathbf{o}_i) = Edist(\mathbf{o}_t, \mathbf{o}_j)$ và $w_i > w_j$ thì $Wdist(\mathbf{o}_t, \mathbf{o}_i) < Wdist(\mathbf{o}_t, \mathbf{o}_j)$.

Goi $w_{max}$ là trọng số lớn nhất của các tế bào trong tập hợp tế bào $\mathbf{O}$ và $w(\mathbf{o}_i)$ là hàm lấy trọng số của tế bào $\mathbf{o}_i \in \mathbf{O}$. Để đáp ứng được 3 thuộc tính trên thì hàm tính khoảng cách có trọng số giữa 2 tế bào được định nghĩa như biểu thức (16)

$$Wdist(\mathbf{o}_i, \mathbf{o}_j) = Edist(\mathbf{o}_i, \mathbf{o}_j) - \frac{w(\mathbf{o}_i)+w(\mathbf{o}_j)}{2w_{max}} \quad (16)$$

Chứng minh các tính chất:

**Tính chất 1:**

Vì $w(\mathbf{o}_i) + w(\mathbf{o}_j) \le 2w_{max}$ và $w(\mathbf{o}_i) > 0$, $w(\mathbf{o}_j) > 0$ nên ta có $0 < \frac{w(\mathbf{o}_i)+w(\mathbf{o}_j)}{2w_{max}} \le 1$.

Mặt khác $Edist(\mathbf{o}_i, \mathbf{o}_j) \ge 1$ khi $i \ne j$

$\Rightarrow 0 \le Edist(\mathbf{o}_i, \mathbf{o}_j) - \frac{w(\mathbf{o}_i)+w(\mathbf{o}_j)}{2w_{max}} < Edist(\mathbf{o}_i, \mathbf{o}_j)$.

$\Rightarrow 0 \le Wdist(\mathbf{o}_i, \mathbf{o}_j) < Edist(\mathbf{o}_i, \mathbf{o}_j)$.

**Tính chất 2:**

Vì $Edist(\mathbf{o}_i, \mathbf{o}_j) = Edist(\mathbf{o}_j, \mathbf{o}_i)$ và $\frac{w(\mathbf{o}_i)+w(\mathbf{o}_j)}{2w_{max}} = \frac{w(\mathbf{o}_j)+w(\mathbf{o}_i)}{2w_{max}}$

$\Rightarrow Wdist(\mathbf{o}_i, \mathbf{o}_j) = Wdist(\mathbf{o}_j, \mathbf{o}_i)$.

**Tính chất 3:**

Vì $w(\mathbf{o}_i) > w(\mathbf{o}_j)$

ta có $\frac{w(\mathbf{o}_t)+w(\mathbf{o}_i)}{2w_{max}} > \frac{w(\mathbf{o}_t)+w(\mathbf{o}_j)}{2w_{max}}$.

Mặt khác $Edist(\mathbf{o}_t, \mathbf{o}_i) = Edist(\mathbf{o}_t, \mathbf{o}_j)$.

$\Rightarrow Edist(\mathbf{o}_t, \mathbf{o}_i) - \frac{w(\mathbf{o}_t)+w(\mathbf{o}_i)}{2w_{max}} < Edist(\mathbf{o}_t, \mathbf{o}_j) - \frac{w(\mathbf{o}_t)+wo_j)}{2w_{max}}$.

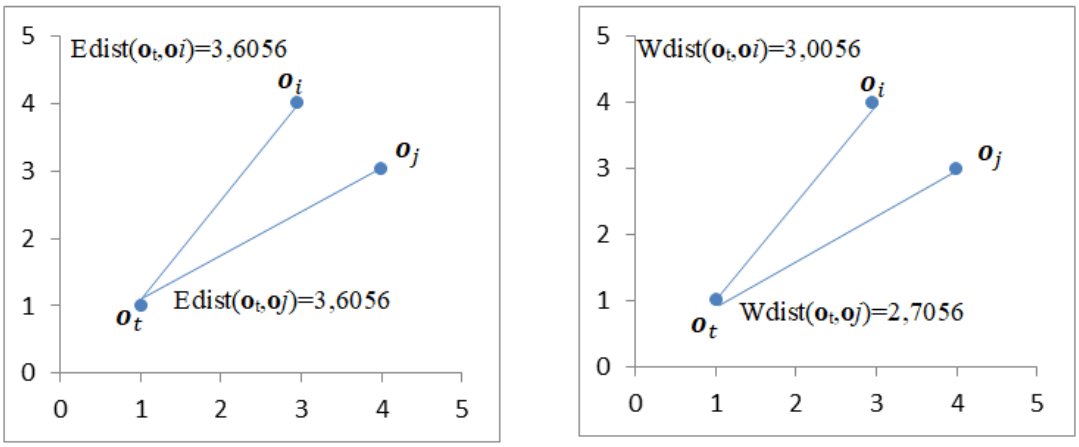
$\Rightarrow Wdist(\mathbf{o}_t, \mathbf{o}_i) < Wdist(\mathbf{o}_t, \mathbf{o}_j)$.

Ví dụ: có 3 tế bào $\boldsymbol{o}_t = (1,1,4)$, $\mathbf{o}_i = (3,4,2)$, $\mathbf{o}_j = (4,3,5)$. Ta có $w(\mathbf{o}_t) = 4$, $w(\mathbf{o}_i) = 2$, $w(\mathbf{o}_j) = 5$, $w_{max} = 5$, $Edist(\mathbf{o}_t, \mathbf{o}_i) = 3.6056$, $Edist(\mathbf{o}_t, \mathbf{o}_j) = 3.6056$.

Áp dụng biểu thức (16):

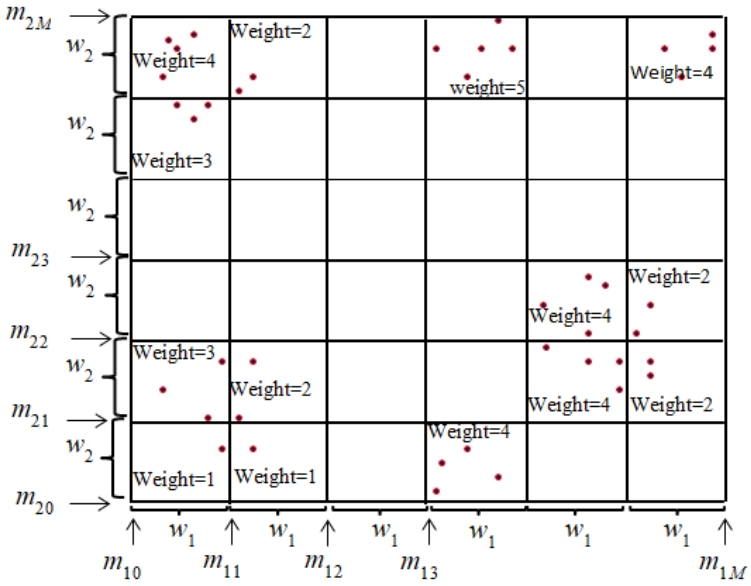
$Wdist(\mathbf{o}_t, \mathbf{o}_i) = 3.6056 - \frac{6}{10} = 3.0056$

$Wdist(\mathbf{o}_t, \mathbf{o}_j) = 3.6056 - \frac{9}{10} = 2.7056$.




**a) Khoảng cách Euclid** **b) Khoảng cách trọng số**

**Hình 4: Ví dụ minh họa sự khác nhau giữa khoảng cách Euclid và khoảng cách có trọng số**



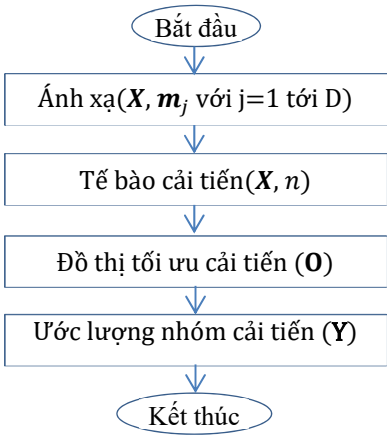

**Hình 5: Ví dụ minh họa chia 1 tập dữ liệu thành tế bào**

### 3.2 Thuật toán ước lượng số cụm dữ liệu cải tiến Weighted-Cell-MST-Based

Khái niệm khoảng cách có trọng số được trình bày trong mục 3.1 được sử dụng để tính khoảng cách giữa các tế bào khi xây dựng đồ thị tối ưu. Do đó, khoảng cách ngắn nhất giữa 2 đỉnh bất kỳ của đồ thị có thể nhỏ hơn 1. Điều này dẫn tới số cạnh của đồ thị tối ưu trong thuật toán Weighted Cell-MST-Based sẽ khác hơn so với số cạnh của đồ thị tối ưu trong thuật toán Cell-MST-Based. Sự thay đổi này dẫn đến kết quả ước lượng số cụm dữ liệu của thuật toán cải tiến có thể khác hơn so với thuật toán ban đầu.

*3.2.1 Mô hình tổng quát của thuật toán*

Thuật toán ước lượng số cụm dữ liệu cải tiến gồm 4 bước như sau:




**Hình 6: Minh họa thuật toán ước lượng số cụm dữ liệu cải tiến gồm 4 bước**

*3.2.2 Bước 1: Ánh xạ đối tượng dữ liệu vào tế bào*

Nội dung của bước này là chia tập dữ liệu lớn **X** gồm $N$ đối tượng thành một tập hợp các tế bào có tối đa $M^D$ tế bào. Mỗi tế bào chứa nhiều đối tượng dữ liệu.

Giải thuật cải tiến sử dụng lại giải thuật *Ánh xạ* ($\boldsymbol{X}, \boldsymbol{m}_j$ *với j=1 tới* D) của thuật toán Cell-MST-Based để ánh xạ các đối tượng dữ liệu trong tập dữ liệu **X** sang tế bào của các đối tượng dữ liệu.

*3.2.3 Bước 2: Lấy tập tế bào*

Mục đích của bước này là lựa chọn các tế bào có trọng số >0 từ kết quả của bước 1. Trọng số của mỗi tế bào được định nghĩa là tổng trọng số của các đối tượng dữ liệu nằm trong tế bào. Giá trị thuộc tính của tế bào được tính bằng trung bình cộng giá trị thuộc tính tương ứng của các đối tượng dữ liệu trong tế bào.

Cải tiến giải thuật *Tế bào* ($\boldsymbol{X}, n$) của thuật toán ước lượng số cụm Cell-MST-Based để tính trọng lượng của từng tế bào.

**Thuật toán 5:** Tế bào cải tiến ($\boldsymbol{X}, n$)

**Input:** Tập dữ liệu **X** có $N$ đối tượng trong không gian $D$ chiều, $n$ là kích cỡ của kiểu dữ liệu dùng để lưu giá trị khoảng cách.
**Output**: Tập hợp các tế nào **O** và véc tơ **w**.
**Begin**
1. Lấy dung lượng khả dụng của bộ nhớ RAM, gọi là $C$.
2. Giải phương trình (12), (13) để tìm giá trị của $M$.
3. Tính vector $\mathbf{v}_{min}$ và $\mathbf{v}_{max}$ dựa vào biểu thức (7) và (8).
4. Tính vector **w** dựa vào biểu thức (9).
5. Tính vector $\mathbf{m}_j$ với j= 1, … , $D$ dựa vào biểu thức (10).
6. Gọi hàm ánh xạ ($\boldsymbol{X}, \boldsymbol{m}_j$ với $j = 1, \dots, D$).
7. Tính trọng số của tế bào bằng tổng trọng số của đối tượng thuộc tế bào, xóa bỏ tế bào có trọng số =0.
8. Tính giá trị các thuộc tính cho các tế bào có trọng số >0 bằng trung bình giá trị thuộc tính tương ứng của các đối tượng thuộc tế bào.
9. Trả về tập hợp tế bào **O** và **w**.

**End**

*3.2.4 Bước 3: Xây dựng đồ thị tối ưu từ tế bào*

Mục đích của bước này là xây dựng đồ thị tối ưu từ kết quả của bước 2.

Cải tiến thuật toán *Đồ thị tối ưu* ($\boldsymbol{Y}$) của thuật toán ước lượng số cụm Cell-MST-Based để tìm đồ thị tối ưu từ tập các tế bào.

**Thuật toán 6:** Đồ thị tối ưu cải tiến($\boldsymbol{O}$)

**Input:** Tập hợp tế bào **O**.
**Output**: Đồ thị tối ưu $\mathbf{G}_0$, giá trị cầm canh $T_0$ và khoảng cách ngắn nhất giữa 2 tế bào.
**Begin**
1. Tính khoảng cách có trọng số giữa các tế bào gọi là **D**.
2. Tính các giá trị $min(\mathbf{D})$, $mean(\mathbf{D})$, and $std(\mathbf{D})$.
3. $T_0 \leftarrow max(mean(\mathbf{D}), std(\mathbf{D}))$.
4. Xây dựng đồ thị từ những khoảng cách $\leq T_0$, gọi là $\mathbf{G}_0$.
5. Xây dựng cây MST từ $\mathbf{G}_0$ gọi là $\mathbf{T}_0$.
6. $T_0 \leftarrow$ maximum(cạnh của $\mathbf{T}_0$)
7. Xóa các cạnh $> T_0$ từ $\mathbf{G}_0$
8. Trả về $\mathbf{G}_0$, $min(\mathbf{D})$ , $T_0$

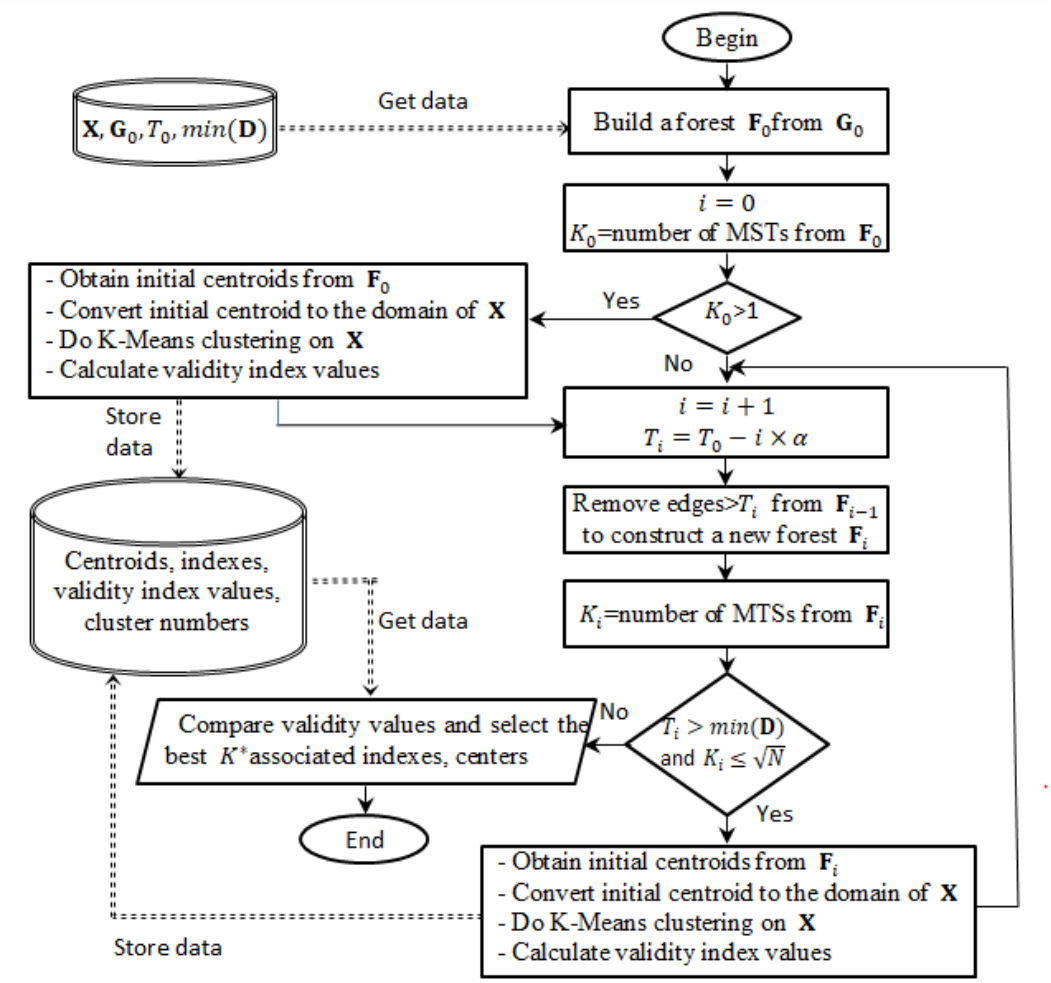
**End**

*3.2.5 Bước 4: Ước lượng số cụm dữ liệu kết hợp phân cụm*

Mục đích của bước này là ước lượng số cụm dữ liệu của tập tin ban đầu thông qua tập tế bào, kết hợp với việc phân cụm tập dữ liệu ban đầu. Vì có thể có nhiều giá trị số lượng cụm khác nhau nên giá trị số lượng cụm tối ưu là giá trị cho kết quả các chỉ số so sánh tốt nhất.

Cải tiến giải thuật *Ước lượng nhóm* (**Y**) của thuật toán ước lượng số cụm Cell-MST-Based để ước lượng số nhóm. Điểm khác biệt giữa bước 4 trong thuật toán Weighted-Cell-MST-Based so với bước 4 của thuật toán Cell-MST-Based là độ dài của cạnh trong rừng cây $\mathbf{F}_i$ của thuật toán Weighted-Cell-MST-Based có thể nhỏ hơn 1 trong khi độ dài các cạnh trong rừng cây $\mathbf{F}_i$ của thuật toán Cell-MST-Based có giá trị nhỏ nhất là 1.

Thuật toán *Ước lượng nhóm cải tiến* (**Y**) được thiết kế lại như sau:



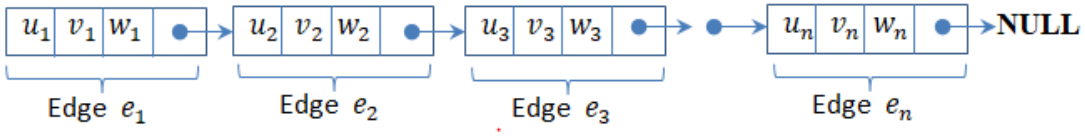

**Hình 7: Lưu đồ thuật toán ước lượng số cụm dữ liệu kết hợp phân cụm**

## 4 KẾT QUẢ THỰC NGHIỆM

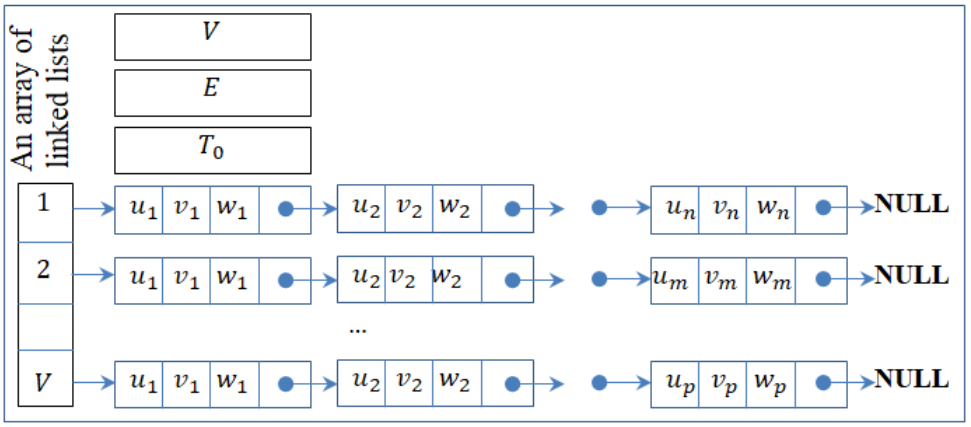
### 4.1 Cài đặt thuật toán

Các thuật toán được cài đặt bằng ngôn ngữ C, biên dịch bằng TDM-GCC 4.8.1 64 bit kết hợp với Dev C++ 5.9.2, chạy trên máy tính cá nhân có cấu hình: Intel processor core i5-3230M CPU 2.60 GH, 6GB of RAM, Windows 8.1. Thuật toán Eager Prim (Sedgewick và Wayne, 2011) được sử dụng để xây dựng cây phủ tối thiểu. Cấu trúc dữ liệu danh sách kề được sử dụng để lưu trữ đồ thị.

Một đồ thị được định nghĩa là một mảng của danh sách liên kết. Mỗi phần tử của mảng ứng với 1 nút. Mỗi danh sách liên kết là một danh sách của các cạnh nối với 1 nút. Mỗi thành phần của danh sách liên kết là một cấu trúc lưu thông tin của cạnh gồm nút đầu, nút cuối, trọng số, con trỏ chỉ đến cạnh tiếp theo.




**Hình 8: Minh họa một danh sách liên kết**




**Hình 9: Minh họa một đồ thị được lưu trữ bằng một mảng của danh sách liên kết**

Hàm *GlobalMemoryStatusEx*() của Hệ điều hành Windows được sử dụng để lấy dung lượng khả dụng của bộ nhớ RAM. Ba giá trị của α (5,0%; 7,5%; 10,0%) được sử dụng để kiểm tra.

### 4.2 Dữ liệu thực nghiệm

Dữ liệu dùng để thực nghiệm là các tập tin đã được sử dụng bởi thuật toán Cell-MST-Based gồm 13 tập dữ liệu (Van Hieu và Phayung, 2015). Trong 13 tập dữ liệu dùng để thực nghiệm, có 10 tập dữ liệu biết trước số cụm, 3 tập dữ liệu còn lại chưa biết trước số cụm dữ liệu.

Tất cả các tập dữ liệu dùng để thực nghiệm không chứa dữ liệu nhiễu nên kết quả của các thuật toán không bị ảnh hưởng bởi dữ liệu ngoại lệ.

**Bảng 2: Thông tin các tập dữ liệu dùng để thực nghiệm**

| Tập tin | Số thuộc tính | Số đối tượng dữ liệu | Số cụn |
|---|---|---|---|
| Simulation2D1 | 2 | 800.017 | 10 |
| Simulation2D2 | 2 | 800.017 | 16 |
| Simulation2D3 | 2 | 1.150.000 | 22 |
| Simulation2D4 | 2 | 1.450.000 | 29 |
| Simulation3D1 | 3 | 300.000 | 7 |
| Simulation3D2 | 3 | 371.000 | 7 |
| Simulation3D3 | 3 | 1.150.000 | 23 |
| Simulation3D4 | 3 | 1.450.000 | 25 |
| Transactions70k | 2 | 665.470 | 4 |
| Transactions90k | 2 | 855.367 | 4 |
| 3D road networks | 3 | 343.874 | Unknown |
| TDriveTrajectory | 2 | 17.760.778 | Unknown |
| GeolifeTrajectory | 2 | 24.895.830 | Unknown |

### 4.3 Kết quả thử nghiệm

So sánh kết quả thực hiện của giải thuật ước lượng số cụm dữ liệu ban đầu (Cell-MST-Based) và giải thuật cải tiến (Weighted-Cell-MST-Based) cho thấy:

Đối với 10 tập dữ liệu đã biết trước số cụm thì kết quả của giải thuật cải tiến giống với kết quả của giải thuật ban đầu đối với cả 3 giá trị của α.

Đối với 3 tập dữ liệu chưa biết trước số cụm thì có 1 trường hợp kết quả giống nhau (tập dữ liệu GeolifeTrajectory), 1 trường hợp giải thuật cải tiến ước lượng ít cụm hơn (tập dữ liệu 3D road networks). Đặc biệt, đối với tập dữ liệu TDriveTrajectory thì giải thuật cải tiến cho kết quả nhỏ hơn và ổn định hơn.

Để hiểu kỹ hơn kết quả của 2 tập dữ liệu 3D road networks và TdriveTrajectory, chúng ta xét tiếp chỉ số Intra-Inter. Khi so sánh các chỉ số Intra-Inter, những trường hợp mà 2 thuật toán cho kết quả khác nhau (3D road networks, TDriveTrajectory) thì giải thuật cải tiến cho kết quả tốt hơn giải thuật ban đầu vì kết quả phân cụm của giải thuật cải tiến có giá trị Intra-Inter nhỏ hơn kết quả phân cụm của giải thuật ban đầu.

Khi xét về mức độ chính xác thì giải thuật cải tiến cho kết quả tốt hơn giải thuật ban đầu khi áp dụng với 2 tập dữ liệu 3D road networks và TDriveTrajectory bởi vì 2 tập dữ liệu này có sự phân phối dữ liệu đặc biệt hơn so với các tập dữ liệu khác. Cụ thể là trọng số của các tế bào gần nhau có độ chênh lệch khá lớn. Đó chính là nguyên nhân tại sao chúng tôi đề xuất sử dụng khoảng cách có trọng số giữa các tế bào.

**Bảng 3: So sánh kết quả số cụm dữ liệu ước lượng được**

| Datasets | Cell-MST-Based | | | Weighted-Cell-MST-Based | | |
|---|---|---|---|---|---|---|
| | α=5,0% | α=7,5% | α=10,0% | α=5,0% | α=7,5% | α=10,0% |
| Simulation2D1 | 9 | 9 | 9 | 9 | 9 | 9 |
| Simulation2D2 | 16 | 16 | 16 | 16 | 16 | 16 |
| Simulation2D3 | 20 | 20 | 20 | 20 | 20 | 20 |
| Simulation2D4 | 29 | 29 | 29 | 29 | 29 | 29 |
| Simulation3D1 | 7 | 7 | 7 | 7 | 7 | 7 |
| Simulation3D2 | 5 | 5 | 5 | 5 | 5 | 5 |
| Simulation3D3 | 23 | 23 | 23 | 23 | 23 | 23 |
| Simulation3D4 | 25 | 25 | 25 | 25 | 25 | 25 |
| Transactions70k | 4 | 4 | 4 | 4 | 4 | 4 |
| Transactions90k | 4 | 4 | 4 | 4 | 4 | 4 |
| 3D road networks | 6 | 6 | 6 | *5* | *5* | *5* |
| TDriveTrajectory | 4 | 5 | 4 | 2 | 2 | 2 |
| GeolifeTrajectory | 2 | 2 | 2 | 2 | 2 | 2 |

**Bảng 4: So sánh chỉ số Intra-Inter đối với 2 tập dữ liệu có kết quả ước lượng số cụm dữ liệu khác nhau, với α=10%**

| Tập dữ liệu | Giải thuật Cell-MST-based | | Giải thuật Weighted Cell-MST-based | |
|---|---|---|---|---|
| | Số cụm | Intra-Inter | Số cụm | Intra-Inter |
| 3D road networks | 6 | 0,301750 | *5* | 0,289281 |
| TDriveTrajectory | 4 | 0,004220 | 2 | 0,001682 |

**Bảng 5: So sánh thời gian thực hiện (số phút) với α=10%**

| Datasets | Thuật toán Cell-MST-Based | Thuật toán Weighted-Cell-MST-Based |
|---|---|---|
| Simulation2D1 | 3,92 | 3,92 |
| Simulation2D2 | 3,47 | 1,94 |
| Simulation2D3 | 1,74 | 1,74 |
| Simulation2D4 | 1,59 | 1,59 |
| Simulation3D1 | 0,47 | 0,47 |
| Simulation3D2 | 1,20 | 0,88 |
| Simulation3D3 | 2,06 | 2,06 |
| Simulation3D4 | 2,36 | 2,37 |
| Transactions70k | 0,68 | 0,46 |
| Transactions90k | 0,45 | 0,97 |
| 3D road networks | 0,97 | 3,92 |
| TDriveTrajectory | 29,04 | 29,55 |
| GeolifeTrajectory | 9,05 | 6,92 |

## 5 KẾT LUẬN

Khi mật độ của các đối tượng dữ liệu tại những vùng gần nhau có sự chênh lệch quá lớn thì kéo theo sự chênh lệch lớn về trọng số của các tế bào gần nhau. Chính điều này làm cho việc áp dụng khoảng cách Euclid không còn phù hợp nữa bởi vì lượng thông tin mà các đối tượng đóng góp vào công thức tính khoảng cách Euclid bằng nhau trong khi trọng số thật sự cũng như mức độ ý nghĩa của các tế bào hoàn toàn khác nhau.

Do đó, khi áp dụng khoảng các có trọng số vào quá trình tính khoảng cách giữa 2 tế bào thì giải thuật ước lượng số cụm dữ liệu cải tiến Weighted-Cell-MST-Based cho kết quả tốt hơn so với thuật toán ban đầu là Cell-MST-Based.

Hiện tại, hầu hết các máy tính cá nhân đều là máy tính có đa bộ xử lý. Để tăng thời gian xử lý dữ liệu, giải thuật Weighted-Cell-MST-Based có thể thiết kế lại theo mô hình MapReduce để tận dụng khả năng của tất cả các bộ xử lý.

Mặt khác, các tập dữ liệu dùng để phân cụm có thể chứa các đối tượng dữ liệu ngoại lệ. Nên áp dụng thuật toán loại bỏ các đối tượng dữ liệu ngoại lệ Cell-DROS (Hieu and Phayung, 2016) để loại bỏ dữ liệu ngoại lệ trước khi áp dụng thuật toán ước lượng số cụm dữ liệu.

## TÀI LIỆU THAM KHẢO

Barioni, M. C. N., Razente, H., Marcelino, A. M. R., Traina, A. J. M. and Traina, C. (2014). Open Issues for Partitioning Clustering Methods: An Overview. Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery. 4.3, (2014) : 161-177.

Fahad, A., Alshatri, N., Tari, Z., Alamri, A., Khalil, I., Zomaya, A. Y., Foufou, S. and Bouras, A. (2014). A Survey of Clustering Algorithms for Big Data: Taxonomy and Empirical Analysis. IEEE Trans. on Emerging Topics in Computing. 2.3: 267-279.

Kokol, P. (2015). Introduction To Data Mining and Knowledge Discovery. In: Encyclopedia of Complexity and Systems Science. Robert A. Meyers (editor). New York: Springer Science+Business Media, pp 1-3.

Kolesnikov, A., Trichina, E. and Kauranne, T. (2015). Estimating the Number of Clusters in a Numerical Data Set Via Quantization Error Modeling. Pattern Recognition. 48.3: 941-952.

Romero, C. and Ventura, S. (2013). Data Mining in Education. Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery. 3.1: 12-27.

Sedgewick, R. and Wayne, K. (2011). Algorithms (4th Edition). Addison-Wesley Professional.

Jesus M., Julian L., and Salvador G. (2017). Exact fuzzy k-nearest neighbor classification for big datasets. Proceedings of 2017 IEEE International Conference on Fuzzy Systems

Shao, X., Pi, J. and Liu, L. (2013). A Method of Dynamically Determining the Number of Clusters and Cluster Centers. Proceedings of 2013 8th International conference on Computer Science Education (ICCSE):283-286.

Starczewski, A. and Krzyak, K. (2015). Performance Evaluation of the Silhouette Index. Artificial Intelligence and Soft Computing:49-58.

Stokes, K. (2014). Graph K-Anonymity through K-Means and as Modular Decomposition.

Texas Tech University (2015). Recommended Software and Hardware Configurations. 2015. https://www.depts.ttu.edu/ithelpcentral/configurations.php (ngày truy cập 17/8/2015).

Van Hieu, D. and Meesad, P. (2015). A Cell-MST-Based Method for Big Dataset Clustering on Limited Memory Computers. Proccedings of 2015 7th International Conference on Information Technology and Electrical Engineering. 632-637.

Van Hieu, D. and Meesad, P. (2016). Cell-RDOS: A Fast Outlier Detection Method for Big Datasets. International Jurnal of Advances in Soft Computing and Its Aplication. 8(3):1-15.

Yan, M. and Ye, K. (2007). Determining the Number of Clusters Using the Weighted Gap Statistic. Biometrics. 63.4, (2007) : 1031-1037.

Yu, H., Liu, Z. and Wang, G. (2014). An Automatic Method to Determine the Number of Clusters Using Decision-Theoretic Rough Set. International Journal of Approximate Reasoning. 55.1: 101-115.

Zhong, C., Malinen, M., Miao, D. and Frnti, P. (2015). A Fast Minimum Spanning Tree Algorithm Based on K-Means. Information Sciences. 295.0: 1-17.