LỜI CẢM ƠN

Để có thể hoàn thiện được luận văn thạc sỹ của mình, trước tiên em xin được gửi lời cảm ơn sâu sắc đến thày PGS.TS Hoàng Xuân Huấn. Thày đã tận tình định hướng, dìu dắt, chỉ bảo cho em trong những bước đầu nghiên cứu khoa học. Trong quá trình ấy thày luôn quan tâm, lo lắng, động viên, những điều đáng quý ấy em xin được ghi nhớ mãi trong lòng.

Em cũng xin được gửi lời chân thành cảm ơn đến các thày cô giáo trong bộ môn Hệ thống thông tin, bộ môn Khoa học máy tính – Khoa Công nghệ thông tin – Trường Đại học Công nghệ – Đại học Quốc gia Hà Nội và các thày cô đã tận tình dạy dỗ, nỗ lực, tâm huyết dạy từng môn học giúp em có được kiến thức về cuộc sống, về chuyên môn và hoàn thành khóa học tại trường.

Đồng thời em cũng xin được gửi lời cảm ơn đến các bạn học, người thân trong gia đình, đồng nghiệp đã giúp đỡ, động viên, tạo điều kiện cho em trong suốt khóa học tại Trường Đại học Công nghệ – Đại học Quốc gia Hà Nội.

Hà Nội, tháng 11 năm 2016

Học viên

Nguyễn Thị Thanh Tâm

LỜI CAM ĐOAN

Em xin cam đoan những nội dung kiến thức mà em trình bày trong quyển luận văn này là do em tự tìm hiểu, nghiên cứu, trình bày dưới sự hướng dẫn trực tiếp của thày PGS. TS Hoàng Xuân Huấn. Tất cả những phần nội dung mà em có tham khảo đã được trích dẫn đầy đủ, ghi rõ nguồn gốc ở phần Tài liệu tham khảo.

Em xin chịu trách nhiệm với lời cam đoan của mình, nếu có mọi phát hiện về sao chép không hợp lệ, vi phạm quy chế đào tạo em xin được hoàn toàn chịu trách nhiệm.

Hà Nội, tháng 11 năm 2016

Học viên

Nguyễn Thị Thanh Tâm

MỤC LỤC

DANH MỤC CÁC KÝ HIỆU, CHỮ VIẾT TẮT

Từ hoặc cụm từ	**Từ viết tắt**	**Từ Tiếng Anh**
Cơ sở dữ liệu	CSDL	DataBase
Thuật toán HAC	HAC	Hierarchical agglomerative clustering
Thuật toán BIRCH	BIRCH	Balanced Interative Reducing and Clustering using Hierarchies
Thuật toán PAM	PAM	Partition Around Mediods
Thuật toán STING	STING	A STatistical Information Grid approach
Giải thuật di truyền	GA	Genetic Algorithms
Nhiễm sắc thể	NST	Chromosomes
Thuật toán C-Mean mờ	FCM	Fuzzy C-Means
Thuật toán NSGA-II	NSGA-II	Non-dominated Sorting Genetic Algorithm-II

DANH MỤC CÁC BẢNG

DANH MỤC CÁC HÌNH VẼ

LỜI NÓI ĐẦU

Bước sang thế kỷ hai mươi mốt, cả thế giới đã cùng nhau chứng kiến sự bùng nổ của công nghệ thông tin. Ngày nay, vật dụng không thể thiếu đối với phần đông con người là chiếc điện thoại thông minh, máy tính bảng... Có thể thấy cùng với sự phát triển của công nghệ phần cứng, phần mềm thì dung lượng dữ liệu số do người dùng tạo ra đang là một vấn đề đáng được chú ý. Bên cạnh đó tất cả các lĩnh vực trong đời sống xã hội đều được tin học hóa cũng tạo nên một lượng dữ liệu khổng lồ. Từ đó có thể thấy nhu cầu cấp thiết là phải có những công cụ và kĩ thuật mới để có thể chuyển khối dữ liệu khổng lồ ấy thành những tri thức có ích. Do đó, lĩnh vực Khai phá dữ liệu ra đời đã đáp ứng được tính thời sự của ngành Công nghệ thông tin không chỉ ở Việt Nam mà trên toàn thế giới.

Lĩnh vực khai phá dữ liệu và phát hiện tri thức trong cơ sở dữ liệu là một lĩnh vực rộng lớn, đã cuốn hút các nhà nghiên cứu. Các công trình nghiên cứu từ nhiều chuyên ngành khác nhau như học máy, thu nhận mẫu, cơ sở dữ liệu (CSDL), thống kê, trí tuệ nhân tạo, thu nhận tri thức trong hệ chuyên gia, cùng hướng đến một mục tiêu thống nhất là trích lọc ra được các "tri thức" từ dữ liệu trong các kho chứa khổng lồ [2]. Và hiện nay nhiều người hiểu khai phá dữ liệu và một thuật ngữ khác - phát hiện tri thức trong cơ sở dữ liệu (Knowlegde Discovery in Databases – KDD) - là như nhau. Tuy nhiên, thực tế cho thấy khai phá dữ liệu chỉ là một bước trong phát hiện tri thức từ cơ sở dữ liệu.

Ngay từ khi mới xuất hiện, khai phá dữ liệu đã trở thành một trong những hướng nghiên cứu có tiềm năng trong lĩnh vực học máy và cơ sở tri thức. Một trong những bài toán khai phá dữ liệu điển hình là phân cụm dữ liệu (Data clustering). Phân cụm (Clustering) thực hiện việc nhóm dữ liệu thành các "cụm" (có thể coi là các lớp mới) để có thể phát hiện được các mẫu phân bố dữ liệu trong miền ứng dụng.Trong nhiều trường hợp, phân cụm còn được gọi là học máy không giám sát (unsupervised learning).

Trong thực tế, dữ liệu luôn có tính nhập nhằng, ranh giới giữa các cụm đôi khi không rõ ràng, khi đó phương pháp phân cụm rõ làm việc không hiệu quả và không mô tả được cấu trúc tự nhiên của tập dữ liệu. Do đó, lý thuyết tập mờ đã được áp dụng nhằm làm cho việc phân cụm dữ liệu được tốt hơn từ đó xây dựng nên phương pháp phân cụm dữ liệu mờ (gọi tắt là phân cụm mờ) [fuzzy clustering].

Tuy nhiên, không phải phương pháp phân cụm mờ nào cũng có thể áp dụng cho mọi bộ dữ liệu. Bởi các giá trị thuộc tính trong dữ liệu định danh là không có thứ tự do đó không áp dụng được các độ đo khoảng cách cơ bản như Euclide để tìm khoảng cách giữa hai véc tơ đặc trưng trong dữ liệu định danh. Vì vậy phải sử dụng một phương pháp khác cho dữ liệu này như K-mode mờ, K -medoid mờ, giải thuật di truyền, …

Hiện nay, lý thuyết toán học về tối ưu hóa đa mục tiêu ngày càng được sử dụng rộng rãi trong cuộc sống cũng như trong khoa học, ví dụ một cá nhân, một tổ chức, một

phương pháp, một kỹ thuật,… có thể sẽ có lúc phải quyết định việc lựa chọn phương án tối ưu để giải quyết một vấn đề nào đó. Tùy thuộc vào từng tình huống cụ thể mà các phương án đưa ra có thể giải quyết một hay nhiều vấn đề cùng một lúc. Khi đó chúng ta phải nghiên cứu, phân tích, trích chọn thông tin nhằm mục đích cuối cùng là đưa ra giải pháp để giải quyết vấn đề.

Tối ưu hóa đa mục tiêu là việc đi tìm phương án tốt nhất theo một nghĩa nhất định nào đó để đạt được nhiều mục tiêu cùng một lúc và một phương án như vậy gọi là một phương án lý tưởng. Trong một bài toán tối ưu đa mục tiêu, việc có hay không có phương án lý tưởng là việc mà chúng ta cần phải quan tâm, xem xét vì trong bài toán này các mục tiêu thường xung đột với nhau nên việc chúng ta cố gắng làm tăng giá trị cực đại hay cực tiểu của một mục tiêu sẽ có thể dẫn đến làm giảm giá trị cực đại hoặc cực tiểu của một mục tiêu khác. Do đó cách tốt nhất có thể là tìm ra một phương án nhằm thỏa mãn tất cả các yêu cầu đa mục tiêu trong một mức độ chấp nhận được và phương án mà chúng ta tìm ra đó được gọi là phương án thỏa hiệp của các hàm mục tiêu. Hiện nay có rất nhiều định nghĩa khác nhau đề cập đến phương án hay nghiệm tối ưu. Các định nghĩa này thường có sự tương quan nhất định với nhau và thường được biểu diễn qua các định lý, các mệnh đề và các tính chất như tối ưu Pareto [7]. Nhờ vào những ưu điểm và hiệu quả thực tế mà tối ưu hóa đa mục tiêu mang lại, nó đang trở thành một trong những lý thuyết toán học được ứng dụng rộng rãi trong nhiều lĩnh vực khoa học như: công nghệ, tài chính, hàng không, kinh tế,…

Bố cục của quyển luận văn chia làm 3 chương như sau:

CHƯƠNG 1. Nền tảng lý thuyết

Chương này trình bày tổng quan về phân cụm dữ liệu: khái niệm và ý nghĩa của việc phân cụm. Để hiểu rõ hơn về phân cụm đa mục tiêu nội dung đi từ khái niệm cơ bản đến sự khác nhau giữa phân cụm một mục tiêu và phân cụm đa mục tiêu. Đồng thời cũng đề cập và phân tích phân cụm rõ và phân cụm mờ, giải thuật GA sử dụng để tối ưu hóa cụm.

CHƯƠNG 2. Phân cụm đa mục tiêu mờ cho dữ liệu định danh

Chương này trình bày nội dung chính của luận văn. Chương này trình bày phương pháp phân cụm đa mục tiêu mờ cho dữ liệu định danh sử dụng giải thuật di truyền.

CHƯƠNG 3. Thử nghiệm

Chương này sẽ tập trung trình bày kết quả thực nghiệm phương pháp đã trình bày ở *CHƯƠNG 2*. Thuật toán được cài đặt và thử nghiệm trên các bộ dữ liệu, từ đó rút ra được một số bình luận, nhận xét và kết luận.

Cuối cùng, phần Kết luận trình bày tóm tắt những kết quả đã đạt được trong luận văn và đề xuất hướng nghiên cứu tiếp theo trong tương lai.

CHƯƠNG 1. NỀN TẢNG LÝ THUYẾT

1.1. Phân cụm dữ liệu là gì?

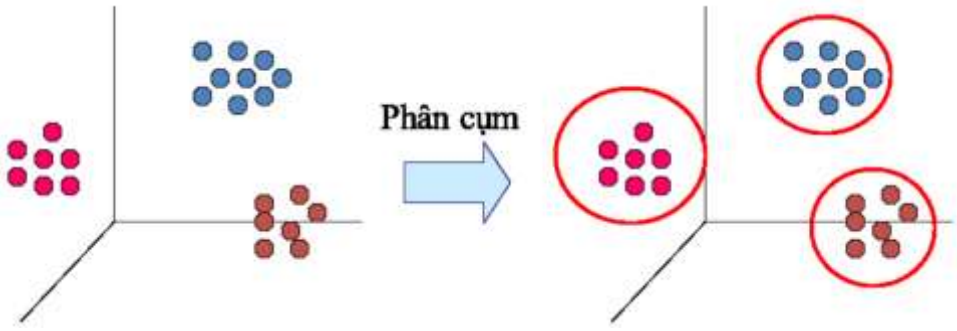
Phân cụm là một việc làm hết sức tự nhiên, nó được hiểu tương tự như việc người ta phân động, thực vật thành các loài, các họ… khác nhau (hay các nhóm có cùng một số đặc điểm nào đó và các đặc điểm này lại rất khác với các loài động, thực vật khác), hay như trong một lớp học người ta có thể phân ra các nhóm học sinh học tốt, học khá, học kém, …

Phân cụm được sử dụng rộng rãi trong rất nhiều lĩnh vực (hay bài toán) như nghiên cứu thị trường, nhận dạng mẫu, phân tích dữ liệu, xử lý ảnh, … Ví dụ trong lĩnh vực kinh doanh, phân cụm có thể giúp phân khách hàng thành các nhóm khác nhau đồng thời cũng có thể cho biết các đặc trưng của các nhóm người dùng này, từ đó công ty sẽ có các chính sách khác nhau dành cho các nhóm khách hàng này.

Vậy phân cụm dữ liệu là gì?

"Phân cụm (Clustering) thực hiện việc nhóm dữ liệu thành các "cụm" (có thể coi là các lớp mới) để có thể phát hiện được các mẫu phân bố dữ liệu trong miền ứng dụng. Phân cụm là một bài toán mô tả hướng tới việc nhận biết một tập hữu hạn các cụm hoặc các lớp để mô tả dữ liệu. Các cụm (lớp) cá thể tách rời nhau và toàn phần (tạo nên một phân hoạch cho tập dữ liệu) hoặc được trình bày đẹp hơn như phân lớp có thứ bậc hoặc có thể chồng lên nhau (giao nhau)" [2].

Do đó, quá trình phân cụm dữ liệu là quá trình phân chia một tập dữ liệu ban đầu thành các cụm dữ liệu để sao cho các phần tử trong cùng một cụm thì "tương tự" nhau và các phần tử trong các cụm khác nhau thì "kém tương tự" nhau. Việc xác định số các cụm dữ liệu có thể thực hiện xác định trước theo kinh nghiệm hoặc xác định tự động theo các phương pháp phân cụm.

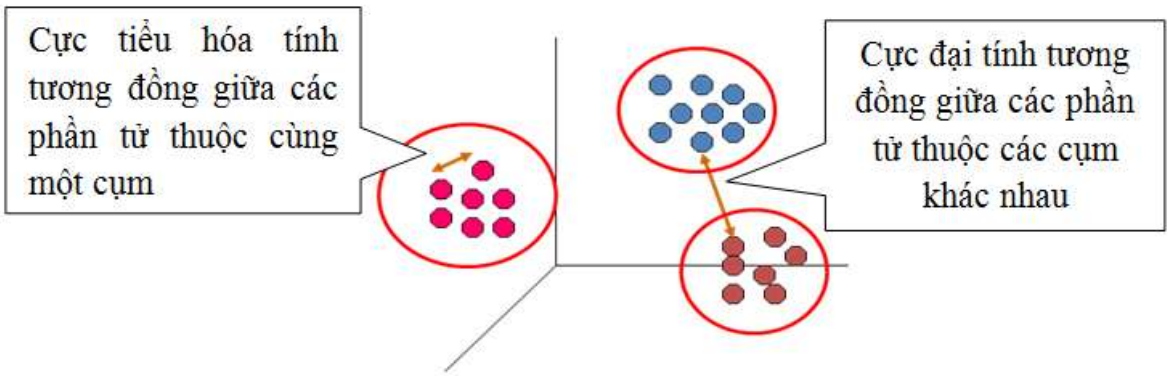


Hình 1.1. Ví dụ về phân cụm dữ liệu

Trong ví dụ ở Hình 1.1, ta có thể dễ dàng xác định được 3 cụm dựa vào dữ liệu đã cho, tiêu chí "tương tự" được nhắc đến ở trên để xác định số cụm trong trường hợp này là "khoảng cách": hai hoặc nhiều đối tượng thuộc cùng một nhóm được nhóm lại theo một khoảng cách nhất định. Ví dụ trên còn được gọi là phân cụm dựa trên khoảng cách.

Còn có một kiểu phân cụm dữ liệu khác như phân cụm dữ liệu dựa vào khái niệm: hai hay nhiều đối tượng sẽ thuộc vào cùng một nhóm nếu có một định nghĩa khái niệm chung cho tất cả các đối tượng trong đó. Hay, đối tượng của một nhóm phải phù hợp với nhau theo miêu tả của khái niệm đã được định nghĩa, không phải theo những biện pháp đơn giản tương tự.

Mục tiêu định hướng bài toán phân cụm đặt ra là cực đại tính tương đồng giữa các phần tử trong mỗi cụm và cực tiểu tính tương đồng giữa các phần tử thuộc các cụm khác nhau (Hình 1.2).



Hình 1.2. Tiêu chí để phân cụm

Trong học máy, phân cụm dữ liệu còn được coi là học máy không có giám sát (unsupervised learning), vì vấn đề mà nó phải giải quyết là tìm một cấu trúc trong tập hợp dữ liệu chưa biết trước các thông tin về cụm, các thông tin về tập huấn luyện hay các thông tin nhãn của các lớp. Trong nhiều trường hợp, nếu phân lớp được coi là học máy có giám sát thì phân cụm dữ liệu là một bước trong phân lớp dữ liệu, nó khởi tạo các lớp để phân lớp bằng cách xác định các nhãn cho các nhóm dữ liệu. [10]

1.2. Các khái niệm cần thiết khi tiếp cận phân cụm dữ liệu

1.2.1. Cấu trúc dữ liệu

Các thuật toán phân cụm dữ liệu thường sử dụng hai loại cấu trúc dữ liệu điển hình sau [6].

Ma trận dữ liệu (cách biểu diễn cấu trúc đối tượng theo biến): ma trận này biểu diễn *n* đối tượngvà *p* biến (hay còn gọi đó là các phép đo/ các thuộc tính) của đối tượng, có dạng ma trận *n* hàng và *p* cột. Trong đó, các hàng biểu diễn cho các đối tượng, các phần tử trong mỗi hàng dùng để chỉ giá trị thuộc tính tương ứng của đối tượng đó.

$$\begin{bmatrix} x_{11} & \dots & x_{1f} & \dots & x_{1p} \\ \dots & \dots & \dots & \dots & \dots \\ x_{i1} & \dots & x_{if} & \dots & x_{ip} \\ \dots & \dots & \dots & \dots & \dots \\ x_{n1} & \dots & x_{nf} & \dots & x_{np} \end{bmatrix}$$

Ma trận phi tương tự (cách biểu diễn cấu trúc đối tượng theo đối tượng): ma trận này lưu trữ khoảng cách của tất cả các cặp đối tượng được thể hiện bằng một ma trận vuông gồm n hàng và n cột. Trong đó, ký hiệu d(i,j): biểu diễn cho khoảng cách hay độ khác biệt giữa đối tượng i và đối tượng j và d(i,j) là một số không âm, d(i,j) gần tới 0 khi hai đối tượng i và j "gần" nhau hơn hay giữa chúng có độ tương đồng cao, d(i,j) càng lớn nghĩa là hai đối tượng i và j càng "xa" nhau hay giữa chúng có độ tương đồng thấp. Do d(i,i)=0 và d(i,j) = d(j,i), nên ma trận phi tương tự được biểu diễn như sau:

$$\begin{bmatrix} 0 & & & & \\ d(2,1) & 0 & & & \\ d(3,1) & d(3,2) & 0 & & \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ d(n,1) & d(n,2) & \dots & \dots & 0 \end{bmatrix}$$

Thông thường ma trận dữ liệu còn được gọi là ma trận 2 kiểu (two-mode matrix), còn ma trận phi tương tự được gọi là ma trận 1 kiểu (one-mode matrix). Trong thực tế hầu hết các thuật toán phân cụm thường sử dụng cấu trúc ma trận phi tương tự. Do đó, cần đưa về dạng ma trận phi tương tự trước khi tiến hành phân cụm nếu dữ liệu đầu vào cần phân cụm được tổ chức dưới dạng ma trận dữ liệu.

1.2.2. Các kiểu dữ liệu

Cho CSDL D có chứa n đối tượng trong không gian k chiều, trong đó x, y, z là các đối tượng thuộc D: $x=(x_1, x_2,\dots,x_k)$; $y=(y_1, y_2,\dots,y_k)$; $z=(z_1, z_2,\dots,z_k)$.Trong đó: x_i, y_i, z_i (i = 1..k) là các đặc trưng hoặc thuộc tính tương ứng của các đối tượng x, y, z.

Có hai đặc trưng cơ bản để phân loại kiểu dữ liệu là kích thước miền và hệ đo [13]:

1.2.2.1. Kiểu dữ liệu dựa trên kích thước miền

- *Thuộc tính liên tục (Continuous Attribute)*: nếu miền giá trị của nó là vô hạn, không đếm được, nghĩa là giữa hai giá trị có tồn tại vô số các giá trị khác, ví dụ như các thuộc tính về màu sắc, cường độ âm thanh,...

- *Thuộc tính rời rạc (Discrete Attribute)*: nếu miền giá trị của nó là tập hữu hạn, đếm được, ví dụ như *lớp học* là một thuộc tính rời rạc với tập các giá trị là: {lớp 1, lớp 2, lớp 3, lớp 4, lớp 5}.

- *Thuộc tính nhị phân (Binary Attribute)*: được coi là trường hợp đặc biệt của thuộc tính rời rạc vì miền giá trị của nó chỉ có hai phần tử được biểu diễn, ví dụ như: Yes/ No hoặc True/ False,...

1.2.2.2. Kiểu dữ liệu dựa trên hệ đo

- *Thuộc tính định danh (Nominal Scale)*: là dạng thuộc tính khái quát hoá của thuộc tính nhị phân, trong đó miền giá trị là rời rạc không phân biệt thứ tự và có nhiều hơn hai phần tử, tức là cho x và y là hai đối tượng thuộc tính thì chỉ có thể xác định là $x \neq y$ hoặc $x = y$. Ví dụ như thuộc tính về màu tóc, màu da...

- *Thuộc tính có thứ tự (Ordinal Scale)*: là thuộc tính định danh có thêm tính thứ tự, nhưng chúng không được định lượng, tức là cho x và y là hai thuộc tính thứ tự thì ta có thể xác định là $x \neq y$ hoặc $x = y$ hoặc $x > y$ hoặc $x < y$. Ví dụ như thuộc tính về thứ tự của cuộc thi học sinh giỏi quốc gia.

- *Thuộc tính khoảng (Interval Scale)*: thuộc tính khoảng dùng để xác định một thuộc tính là đứng trước hoặc đứng sau thuộc tính khác với một khoảng là bao nhiêu. Nếu $x_i > y_i$ thì ta nói x cách y một khoảng $x_i - y_i$ tương ứng với thuộc tính thứ i. Một ví dụ về thuộc tính khoảng như thuộc tính số serial của một đầu mã thẻ điện thoại. Thuộc tính này thường dùng để đo các giá trị theo xấp xỉ tuyến tính.

- *Thuộc tính tỉ lệ (Ratio Scale)*: là thuộc tính khoảng nhưng được xác định một cách tương đối so với điểm mốc, ví dụ như thuộc tính chiều cao/ cân nặng lấy điểm 0 làm mốc.

Trong các thuộc tính dữ liệu đã được nhắc đến ở phía trên, thuộc tính định danh (Categorical Scale) là thuật ngữ dùng để gọi chung cho thuộc tính định danh và thuộc tính có thứ tự, còn thuật ngữ thuộc tính số (Numeric Scale) thì dùng để gọi chung cho thuộc tính khoảng và thuộc tính tỉ lệ.

1.2.3. Độ đo tương tự và phi tương tự

Người ta phải đi tìm cách thích hợp để xác định "khoảng cách" giữa các đối tượng (hay là phép đo tương tự giữa các dữ liệu) để thực hiện việc phân cụm. Đó là các hàm để đo sự giống nhau giữa các cặp đối tượng dữ liệu và giữa các đối tượng dữ liệu thường thì các hàm này hoặc là để tính độ tương tự (similar) hoặc là để tính độ phi tương tự (dissimilar).

1.2.3.1. Không gian metric

Một không gian metric là một tập mà trong đó thực hiện việc xác định các "khoảng cách" giữa từng cặp phần tử, với những tính chất thông thường của khoảng cách hình học. Tức là, một tập X (các phần tử của X có thể là những đối tượng bất kỳ) các đối tượng dữ liệu trong CSDL D như đã đề cập ở trên được gọi là một không gian metric nếu:

- Với mỗi cặp phần tử x, y thuộc X đều có xác định, theo một quy tắc nào đó, một số thực δ(x,y), được gọi là khoảng cách giữa x và y.
- Quy tắc nói trên thoả mãn hệ tính chất sau :
  (i) δ(x,y) > 0 nếu x ≠ y;
  (ii) δ(x, y)=0 nếu x =y;
  (iii) δ(x,y) = δ(y,x) với mọi x,y;
  (iv) δ(x,y) ≤δ(x,z)+δ(z,y)

Hàm δ(x,y) được gọi là một metric của không gian, trong đó các phần tử của X gọi là các điểm của không gian này.

1.2.3.2. Thuộc tính khoảng cách

Sau khi chuẩn hoá, độ đo phi tương tự của hai đối tượng dữ liệu x, y được xác định bằng các metric khoảng cách như sau:

Khoảng cách Minskowski: $d(x,y) = (\sum_{i=1}^{n}|x_i - y_i|^r)^{1/r}$, q ≥ 1 (1.1)

Ba khoảng cách phổ biến sử dụng khoảng cách Minskowski được định nghĩa:

- *Khoảng cách Euclide*: $d(x,y) = (\sum_{i=1}^{n}|x_i - y_i|^2)^{1/2}$, (q = 2)
- *Khoảng cách Manhattan*: $d(x,y) = \sum_{i=1}^{n}|x_i - y_i|$, (q = 1) (1.3)
- *Khoảng cách cực đại*: $d(x,y) = Max_{i=1}^{n}|x_i - y_i|$, (q → ∞). (1.4)

Trong đó khoảng cách Euclide là chuẩn khoảng cách được dùng phổ biến nhất trong các chuẩn theo khoảng cách Minskowski.

1.2.3.3. Thuộc tính nhị phân

Xây dựng Bảng 1.1 sử dụng để tìm độ đo:

Bảng 1.1. Bảng giá trị tham số

Đối tượng y

Đối tượng x	y:1	y:0	Tổng
x:1	α	β	$\alpha+\beta$
x:0	γ	δ	$\gamma+\delta$
Tổng	$\alpha+\gamma$	$\beta+\delta$	τ

Với Bảng 1.1 ta có các thông tin sau:

- α là tổng số các thuộc tính có giá trị là 1 trong cả hai đối tượng x,y;
- β là tổng số các giá trị thuộc tính có giá trị là 1 trong x và 0 trong y;
- γ là tổng số các giá trị thuộc tính có giá trị là 0 trong x và 1 trong y;
- δ là tổng số các giá trị thuộc tính có giá trị là 0 trong x và y.

Trong đó: $\tau = \alpha + \gamma + \beta + \delta$

Khi đó độ đo tương tự được đo như sau:

Hệ số đối sánh đơn giản: $d(x,y) = \frac{\alpha + \delta}{\tau}$, có thể thấy hai đối tượng x và y có vai trò như nhau, tức là chúng đối xứng và có cùng trọng số.

Hệ số Jacard: $d(x,y) = \frac{\alpha}{\alpha + \beta + \gamma}$, (tham số này bỏ qua số các đối sánh giữa 0 – 0). Công thức tính này được sử dụng trong trường hợp mà trọng số của các thuộc tính có giá trị 1 của đối tượng dữ liệu có cao hơn nhiều so với các thuộc tính có giá trị 0, như vậy các thuộc tính nhị phân ở đây là không đối xứng.

1.2.3.4. Thuộc tính định danh

Độ đo phi tương tự giữa hai đối tượng x và y được định nghĩa như sau:

$$d(x,y) = \frac{p - m}{p} \qquad (1.5)$$

Trong đó: p là tổng số các thuộc tính,

m là số thuộc tính đối sánh tương ứng trùng nhau.

1.2.3.5. Thuộc tính có thứ tự

Phép đo độ phi tương tự giữa các đối tượng dữ liệu với thuộc tính thứ tự được thực hiện như sau: giả sử i là thuộc tính thứ tự có M_i giá trị (M_i là kích thước miền giá trị)

Các trạng thái M_i được sắp thứ tự: [1…M_i] và có thể thay thế mỗi giá trị của thuộc tính bằng giá trị cùng loại r_i, với $r_i \in \{1..\ M_i\}$.

Mỗi một thuộc tính có thứ tự có các miền giá trị khác nhau, vì vậy có thể chuyển đổi chúng về cùng miền giá trị [0,1] bằng cách thực hiện phép biến đổi sau cho mỗi thuộc tính:

$$z_i^{(j)} = \frac{r_i^{(j)} - 1}{M_i - 1} \qquad (1.6)$$

Sử dụng công thức tính độ phi tương tự của *thuộc tính khoảng* đối với các giá trị $z_i^{(j)}$, đây cũng chính là độ phi tương tự của thuộc tính có thứ tự.

1.2.3.6. Thuộc tính tỷ lệ

Có nhiều cách khác nhau để tính độ tương tự giữa các thuộc tính tỉ lệ. Một trong đó là sử dụng công thức tính *logarit* cho mỗi thuộc tính hoặc là loại bỏ đơn vị đo của các thuộc tính dữ liệu bằng cách chuẩn hoá chúng hoặc gán trọng số cho mỗi thuộc tính giá trị trung bình, độ lệch chuẩn. Độ tương đồng dữ liệu với mỗi thuộc tính dữ liệu đã được gán trọng số tương ứng w_i ($1 <= i <= k$), được xác định như sau:

$$d(x, y) = \sqrt{\sum_{i=1}^{n} w_i (x_i - y_i)^2} \qquad (1.7)$$

1.3. Phân cụm dữ liệu mờ

Phân cụm dữ liệu rõ (phân cụm rõ) là phương pháp chia tập dữ liệu ban đầu thành các cụm dữ liệu và mỗi phần tử dữ liệu chỉ thuộc về một cụm dữ liệu. Các kỹ thuật này thường phù hợp với việc phát hiện ra các cụm có mật độ cao và rời nhau, đường biên giữa các cụm được xác định tốt. Nhưng trên thực tế hiện nay rõ ràng có rất nhiều dữ liệu có tính nhập nhằng, đường biên giữa các cụm không rõ ràng tức là, một phần tử dữ liệu có thể thuộc nhiều cụm khác nhau [10]. Ví dụ như trong phân cụm tài liệu, một tài liệu có xu hướng có nhiều hơn một chủ để chứa trong tài liệu đó, như một tài liệu có thể chứa thông tin về máy tính, phần cứng, phần mềm, mạng máy tính. Vì vậy chương này sẽ đề cập, phân tích và làm rõ về phân cụm mờ. Đồng thời cũng trình bày lý thuyết về tối ưu hóa đơn mục tiêu và đa mục tiêu.

1.3.1. Tổng quan về tập mờ

Đối với những dữ liệu như đã nói ở trên các kỹ thuật phân cụm dữ liệu rõ làm việc không hiệu quả và không mô tả được cấu trúc thực của dữ liệu. Để có thể giải quyết được thì người ta sử dụng đến lý thuyết tập mờ vào việc phân cụm dữ liệu.

Lotfi A. Zadeh - người sáng lập ra lý thuyết tập mờ [15], ý tưởng trong lý thuyết tập mờ của ông là từ những khái niệm mang tính trừu tượng, không chắc chắn của thông tin mang lại nhưng "mờ" như: già – trẻ, lớn – bé, cao – thấp, xinh – xấu, … ông đã chỉ ra cách biểu diễn các thông tin "mờ" đó bằng một khái niệm toán học được gọi là tập mờ (Fuzzy set), như là một sự khái quát của khái niệm tập hợp.

1.3.1.1. Định nghĩa tập rõ

Định nghĩa 2.1 [15]: Cho tập nền X và x là phần tử thuộc tập X. Một tập C trên tập X là một tập hợp rõ, với x là phần tử của tập hợp C, chỉ có thể có $x \in C$ hoặc $x \notin C$. Có

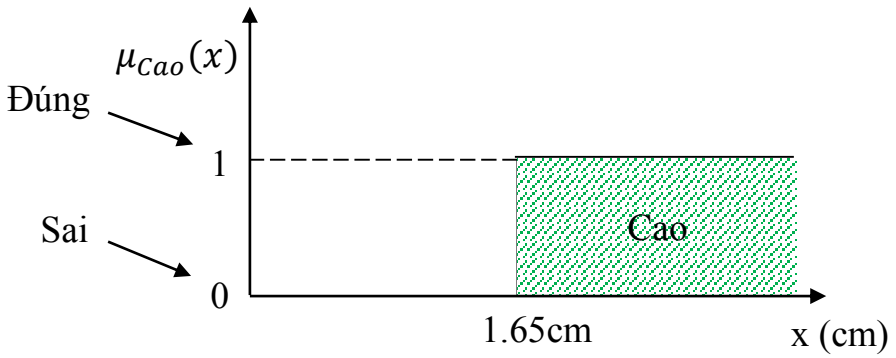
thể sử dụng hàm μ(x) để mô tả khái niệm thuộc về. Hàm μ(x) được gọi là hàm thuộc hay hàm đặc trưng của tập hợp C.

$$\mu(x) = \begin{cases} 1 & \text{if } x \in C \\ 0 & \text{if } x \notin C \end{cases} \quad (1.8)$$

Ví dụ:

Nếu chiều cao một người nào đó trên 1.65cm thì là cao, ngược lại là không cao. Hình bên dưới minh họa tập hợp "Cao" gồm tất cả những người có chiều cao từ 1.65cm trở lên.

$$Cao = \{x \in R | \ x \geq 1.65\}$$



Hình 1.3. Hình minh họa cho tập chiều cao của con người.

Từ Hình 1.3 cho thấy lý thuyết tập rõ không thể hiện được sự khác biệt giữa các phần tử trong cùng một tập hợp. Giữa hai người có chiều cao 1.70cm và 1.75cm không thể hiện được người nào cao hơn người nào.

Bên cạnh đó còn một vấn đề nữa mà lý thuyết tập rõ không giải quyết được, mà vấn đề nó không giải quyết được trong thực tế lại diễn ra khá phổ biến như nó không thể biểu diễn được dữ liệu mang tính mơ hồ, đại khái như: Hoa trông không cao lắm, Tú thì thấp thấp thôi. Câu hỏi đặt ra là: Hoa như vậy thì có thuộc tập hợp những người cao hay không? Có thể hiểu như thế nào là "thấp thấp"?

1.3.1.2. Định nghĩa tập mờ

Định nghĩa 2.2 [15]: Cho tập nền X và x là phần tử của tập X. Một tập mờ F trên tập X được định nghĩa bởi một hàm thành viên hay còn gọi là hàm thuộc $\mu_F(x)$ (*degree of membership)*, đo "mức độ" mà phần tử x thuộc về tập F thỏa mãn điều kiện với $\forall x \in X$, $0 \leq \mu_F(x) \leq 1$.

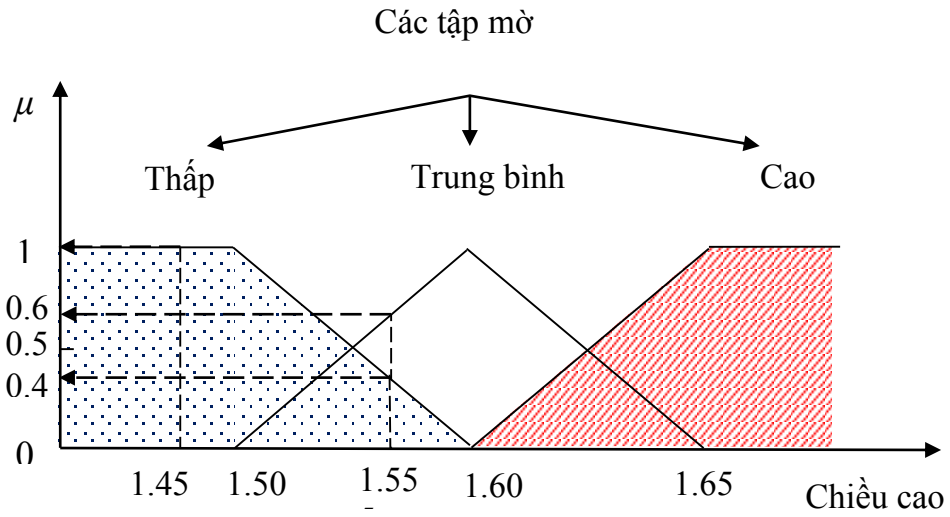
$$F = \{(x, \mu_F(x)) \mid x \in X\} \quad (1.9)$$

Khi $\mu_F(x) = 0$ thì $x \notin F$ hoàn toàn. Khi $\mu_F(x) = 1$ thì $x \in F$ hoàn toàn.

Tập mờ F rỗng nếu và chỉ nếu $\mu_F(x) = 0$ với $\forall x \in X$.
Tập mờ F toàn phần nếu và chỉ nếu $\mu_F(x) = 1$ với $\forall x \in X$.

Như vậy, khái niệm tập mờ là sự tổng quát hóa khái niệm tập rõ bởi hàm thuộc của nó có thể lấy giá trị bất kỳ trong khoảng [0, 1], tập rõ chỉ là một tập mờ đặc biệt vì hàm thuộc $\mu_F(x)$ chỉ nhận hai giá trị 0 hoặc 1.

Ví dụ: Cho các tập mờ: "Cao ", "Trung bình", "Thấp"



Hình 1.4. Ví dụ minh họa các tập mờ "Thấp", "Trung bình", "Cao"

Từ Hình 1.4, ta nhận thấy nếu cho biết chiều cao của một người thì có thể xác định mức độ người đó thuộc về lớp người thấp, trung bình hay cao. Ví dụ cụ thể như sau:

- Tú 1.45cm→ $\mu_{\text{Thấp}}$(Tú) = 1, $\mu_{\text{Trung bình}}$(Tú) = 0, μ_{Cao}(Tú) = 0;
- Hoa 1.55cm→ $\mu_{\text{Thấp}}$(Hoa) = 0.4, $\mu_{\text{Trung bình}}$(Hoa) = 0.6, μ_{Cao}(Hoa) = 0.

1.3.2. Phân cụm rõ và phân cụm mờ

Phương pháp phân cụm dữ liệu rõ được hiểu là khi phân cụm sẽ thực hiện phân chia các đối tượng dữ liệu thành các cụm loại trừ lẫn nhau, mỗi đối tượng chỉ thuộc về duy nhất một cụm. Ví dụ như khi phân cụm các học sinh tiểu học thành các lớp 1, 2, 3, 4, 5, khi đó mỗi học sinh chỉ có thể thuộc về một lớp.

Nhưng có những trường hợp khác, các cụm dữ liệu không thể tách biệt nhau một cách rõ ràng, một đối tượng dữ liệu có thể thuộc về nhiều hơn một cụm. Ví dụ, khi phân cụm sách, tài liệu tham khảo thành các chủ đề thì một tài liệu có thể liên quan tới nhiều chủ đề khác nhau. Khi đó để giải quyết vấn đề vừa nêu ở trên người ta thường dùng phương pháp phân cụm dữ liệu mờ.

1.3.2.1. Phân cụm rõ

Như đã nói ở trên, trong phương pháp phân cụm rõ thì mỗi đối tượng dữ liệu chỉ thuộc về chính xác một cụm. Mục tiêu của quá trình phân cụm là phân chia tập dữ liệu X gồm n đối tượng X=$\{x_1, x_2, \dots x_n\} \subset R^S$ thành c cụm. Trong phân hoạch rõ tập X có thể được xác định như là một họ các tập con $\{C_i \mid 1 \le i \le c\}$ thỏa mãn:

$$\cup_{i=1}^{c} C_i = X \tag{1.10a}$$

$$C_i \cap C_j = \emptyset, \quad 1 \le i \ne j \le c \tag{1.10b}$$

$$\emptyset \ne C_i \subset X, \quad 1 \le i \le c \tag{1.10c}$$

Hàm liên thuộc có thể được viết dưới dạng ma trận phân hoạch $U = [U_{ij}]_{cxn}$. Trong đó:

$$U_{ij} = \begin{cases} 1 & \text{nếu } x_j \in C_i \\ 0 & \text{nếu } x_j \notin C_i \end{cases} \quad 1 \le i \le c, 1 \le j \le n \tag{1.11}$$

Đặt M_{hc} là tập tất cả phân hoạch rõ của X:

$$M_{hc} = \left\{ U \in R^{cxn} \mid U_{ij} \in \{0,1\}, \forall i,j; \sum_{i=1}^{c} U_{ij} = 1, \forall j; 0 < \sum_{j=1}^{n} U_{ij} < n, \forall i \right\}. \tag{1.12}$$

$R^{c\,x\,n}$ là không gian của tất cả các ma trận thực cấp cxn.

Trong đó, một số thuật toán phân cụm rõ đã được nhắc đến ở chương 1 như: thuật toán k-means, thuật toán k-Medoids, DBSCAN, STING, ...

1.3.2.2. Phân cụm mờ

Trong khi đó, đối với phân cụm dữ liệu mờ các đối tượng dữ liệu có thể thuộc về nhiều hơn một cụm, tương ứng với các mức độ liên thuộc khác nhau, đặc trưng cho mức độ mà các điểm dữ liệu đó thuộc về các cụm.

Cho tập dữ liệu X gồm n đối tượng X=$\{x_1, x_2, \dots x_n\} \subset R^S$ tổ chức thành c cụm thể hiện qua các hàm liên thuộc U_{ij} mô tả mức độ đối tượng dữ liệu x_j thuộc về cụm i, với mọi$x_j \in$ X.

- Mức độ liên thuộc nhận giá trị giữa 0 và 1 (Công thức 1.13a). Đối tượng dữ liệu gần trung tâm cụm có mức độ thuộc cao hơn so với những đối tượng nằm ở gần biên của cụm.

- Đối tượng x_j càng xa tâm cụm i thì giá trị hàm liên thuộc U_{ij} càng dần về 0;
- Tương tự như vậy đối tượng x_j càng gần tâm cụm i thì giá trị hàm liên thuộc U_{ij} càng dần tới 1;
- Nếu đối tượng x_j nằm xa tất cả các cụm thì giá trị hàm liên thuộc U_{ij} dần tới 1/c.

- Tổng mức độ liên thuộc của một đối tượng tới tất cả các cụm là 1 (Công thức 1.13b)
- Điều kiện (công thức 1.13c) đảm bảo rằng không tồn tại một cụm nào mà không chứa bất kỳ đối tượng nào.

$$0 \le U_{ij} \le 1, \quad (1 \le i \le c, 1 \le j \le n) \tag{1.13a}$$

$$\sum_{i=1}^{c} U_{ij} = 1, (1 \le j \le n) \tag{1.13b}$$

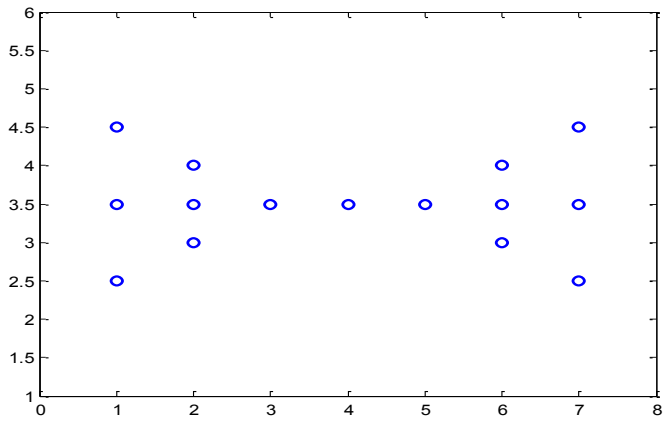
$$0 < \sum_{j=1}^{n} U_{ij} < n, \quad (1 \le i \le c) \tag{1.13c}$$

Đặt M_{fc} là tập tất cả phân hoạch mờ của X:

$$M_{fc} = \left\{ U \in R^{cxn} \,\middle|\, U_{ij} \in [0,1], \forall i,j; \sum_{i=1}^{c} U_{ij} = 1, \forall j; 0 < \sum_{j=1}^{n} U_{ij} < n, \forall i \right\}. \tag{1.14}$$

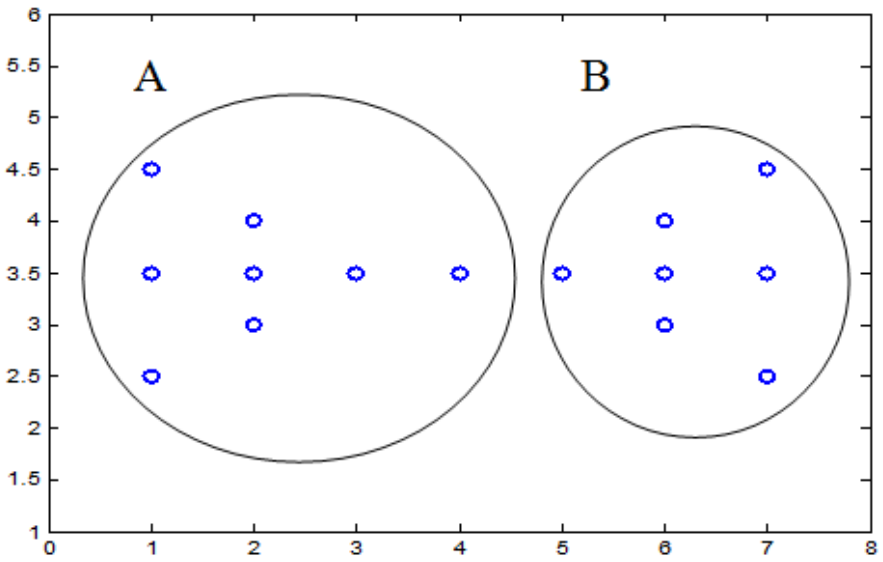
$R^{c\,x\,n}$ là không gian của tất cả các ma trận thực cấp c *x n*.

Để làm rõ hơn sự khác nhau giữa phân cụm rõ và phân cụm mờ ta xét ví dụ minh họa với tập dữ liệu hình cánh bướm (Butterfly) gồm 15 điểm (Hình 1.5).



Hình 1.5. Tập dữ liệu hình cánh bướm

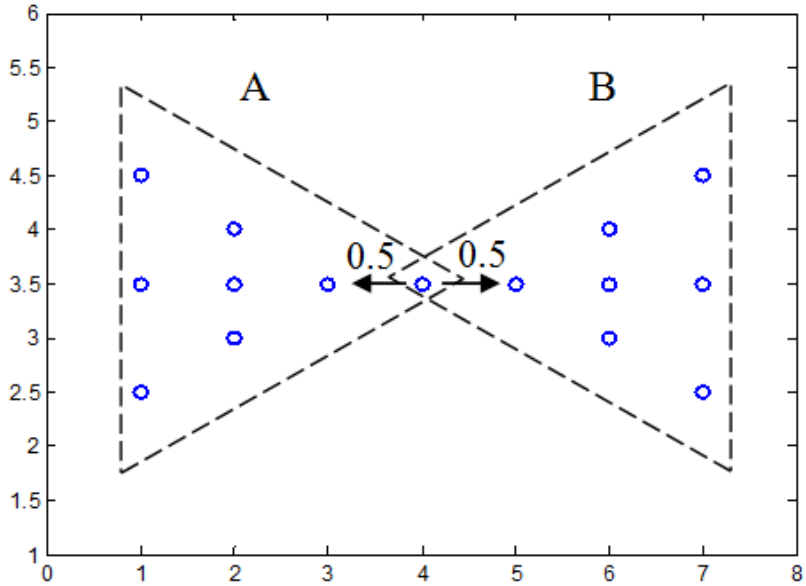
Sử dụng phương pháp phân cụm rõ để phân cụm những điểm dữ liệu trên, kết quả thu được hai cụm (xem Hình 1.6). Có thể thấy kết quả này không cho thấy cấu trúc tự nhiên của tập dữ liệu. Với điểm dữ liệu (4, 3.5) nằm ở giữa có khả năng thuộc về cả 2 cụm là như nhau, nhưng phương pháp phân cụm rõ đánh dấu điểm này thuộc về cụm A với độ thuộc bằng 1.



Hình 1.6. Kết quả phân cụm rõ với tập dữ liệu hình cánh bướm

Đối với tập dữ liệu hình cánh bướm trên phân cụm rõ không cho thấy sự khác biệt giữa các điểm dữ liệu trong cùng một cụm – những phần tử dữ liệu nằm ở trung tâm của cụm và những phần tử dữ liệu nằm ở gần biên của cụm. Ngược lại trong phân cụm mờ mỗi điểm dữ liệu được mô tả bởi một giá trị liên thuộc, tùy vào việc chúng có nằm gần các trung tâm cụm hay không mà chỉ ra mức độ thuộc của chúng với cụm đó.

Vẫn với tập dữ liệu hình cánh bướm ở trên, với phương pháp phân cụm mờ, điểm dữ liệu (4, 3.5) có giá trị mức độ liên thuộc về hai cụm A và B đều là 0.5, phản ánh đúng vị trí nằm giữa hai cụm của điểm dữ liệu này (Hình 1.7. Hai cụm mờ của tập dữ liệu hình cánh bướm)



Hình 1.7. Hai cụm mờ của tập dữ liệu hình cánh bướm

Bảng 1.2. Giá trị hàm liên thuộc của tập dữ liệu hình cánh bướm sử dụng thuật toán k-means và c-means mờ.

Dữ liệu	Phân cụm rõ Thuật toán k-means		Phân cụm mờ Thuật toán C-means mờ	
	U_1	U_2	U_1	U_2
1.0 2.5	1	0	0.0G7	0.9453
1.0 3.5	1	0	0.0226	0.9774
1.0 4.5	1	0	0.0547	0.9453
2.0 3.0	1	0	0.0161	0.9839
2.0 3.5	1	0	0.0024	0.9976
2.0 4.0	1	0	0.0161	0.9839
3.0 3.5	1	0	0.1242	0.8758
4.0 3.5	1	0	0.5000	0.5000
5.0 3.5	0	1	0.8759	0.1241
6.0 3.0	0	1	0.9839	0.0161
6.0 3.5	0	1	0.9976	0.0024
6.0 4.0	0	1	0.9839	0.0161
7.0 2.5	0	1	0.9453	0.0547
7.0 3.5	0	1	0.9774	0.0226
7.0 4.5	0	1	0.9453	0.0547

1.4. Tối ưu đa mục tiêu [1]

1.4.1. Bài toán tối ưu tổng quát

F(X) => max (min) với X ∈ D gọi là miền ràng buộc.

Trong đó:

- F(X) có thể là một hàm vô hướng hay hàm véc tơ, tuyến tính hay phi tuyến.

+ Nếu F là hàm vô hướng thì ta có mô hình quy hoạch (tối ưu) đơn mục tiêu,

+ Nếu F là vectơ thì có mô hình quy hoạch (tối ưu) đa mục tiêu.

- X có thể là một biến đơn lẻ hay một tập hợp nhiều biến tạo thành một vectơ hay thậm chí là một hàm của nhiều biến khác. Biến có thể nhận các giá trị liên tục hay rời rạc.

- D là miền ràng buộc của X, thường được biểu diễn bởi các đẳng thức, bất đẳng thức và được gọi là miền phương án khả thi hay phương án chấp nhận được.

1.4.2. Tối ưu đơn mục tiêu

Dạng chính tắc của bài toán tối ưu toàn cục một mục tiêu được biểu diễn như sau:

Max (Min) f(X) $X = (x_1, x_2, \ldots, x_n)$

với***(i) g(X)*** ≤ 0, $j=1, 2, \ldots, k$,

(ii) g(X) = 0, $j=k+1, k+2, \ldots, m$,

Trong các bài toán thực tế có thể bổ sung các ràng buộc dạng:

(iii) $a_i \leq x_i \leq b_i$, $i = 1, 2, \ldots, n$

Hàm mục tiêu f(x) và các hàm ràng buộc $g_j(x)$ với j=1, 2, …, m có thể là tuyến tính hay phi tuyến. Véctơ X có thể bao gồm các thành phần rời rạc hay liên tục hoặc là sự kết hợp giữa các thành phần rời rạc và các thành phần liên tục. Các dạng khác của bài toán tối ưu một mục tiêu đều có thể đưa về dạng chính tắc theo những quy tắc nhất định.

Nếu ký hiệu D là miền các phương án (miền ràng buộc) cho bởi các ràng buộc (i), (ii) hoặc (iii) thì bài toán trên đây có thể viết gọn hơn như sau**: f(x) →max (min) với x ∈ D**. Lúc này, x* ∈ D được gọi là phương án tối ưu toàn cục nếu $\forall$ x∈ D ta luôn có: f(x*) ≤ f(x). Trong trường hợp f(x*) ≤ f(x) chỉ đúng với $\forall$ x ∈ D trong một lân cận của x* thì x* được gọi là phương án tối ưu địa phương.

1.4.3. Tối ưu đa mục tiêu

1.4.3.1. Bài toán tối ưu đa mục tiêu

Bài toán tối ưu đa mục tiêu tổng quát có thể xem xét dưới dạng sau :

Cực đại hóa các hàm lợi ích :

$$f_i(x) \to \max, \ (i = \overline{1,k}) \qquad (1.15)$$

Với $x \in X \subset R^n$

Nói chung không có lời giải đồng thời đạt cực đại của cả k hàm f_i ($i = \overline{1,k}$). Lời giải của nó được tìm theo nghĩa tối ưu Pareto như sau:

Định nghĩa: Điểm $x^* \in X$ gọi là tối ưu Pareto của bài toán đa mục tiêu trên tập X nếu không tồn tại điểm y $\in X$ sao cho có ít nhất $i \leq k$ mà

$$f_i(y) > f_i(x^*) \qquad (1.16)$$

và $f_j(y) > f_j(x^*) \quad \forall j \neq i; \quad j \leq k$

1.4.3.2. Xử lý bài toán đa mục tiêu

Bài toán tối ưu đa mục tiêu hiện nay đang được rất nhiều người quan tâm nghiên cứu và có nhiều phương pháp để tìm tập lời giải Pareto. Trong quá trình đó, việc lựa chọn lời giải thường theo hướng "hỗ trợ quyết định" và có thể xử lý được nhờ đưa về các bài toán đơn mục tiêu.

a. Đưa các mục tiêu thứ yếu vào điều kiện ràng buộc

Theo phương pháp này, ta chọn hàm mục tiêu f_j mà ta cho là quan trọng nhất và xét bài toán:

$$f_j(x) \to \max \tag{1.17}$$

Với điều kiện

$$\begin{cases} f_i(x) \geq c_i \quad \left(i = \overline{1,k} \wedge i \neq j\right) \\ x \in X \end{cases} \tag{1.18}$$

Trong đó các c_i thay đổi theo ý muốn của người ra quyết định.

b. Chọn trọng số ưu tiên

Ta chọn các trọng số $\rho_i > 0 \quad \left(i = \overline{1,k}\right)$ sao cho

$$\rho_1 + \rho_2 + \cdots + \rho_k = 1$$

Độ lớn của ρ_i phụ thuộc vào mức độ quan trọng của hàm mục tiêu f_i. Với các $\rho_i \left(i = \overline{1,k}\right)$ đã có ta giải bài toán

$$\max\left\{\sum_{i=1}^{k} \rho_i f_i(x) \middle/ x \in X\right\} \tag{1.19}$$

Người ta quyết định tùy theo sự thay đổi khi chọn các trọng số ρ_i để lựa chọn lời giải.

1.4.4. Chọn phương án trong bài toán đơn mục tiêu và bài toán đa mục tiêu

Trong bài toán đơn mục tiêu thì các phương án so sánh được với nhau. Nếu 2 phương án x và y có hai giá trị hàm mục tiêu **f(y) ≤ f(x)** thì chấp nhận phương án x.

Trong bài toán đa mục tiêu một nghiệm x^* của bài toán (P1) được gọi là nghiệm lý tưởng nếu: $\mathbf{f_i(x^*) \leq f_i(x)}$ với $\forall x \in X$, i={1,...,k}. Nói một cách khác một nghiệm lý tưởng là một nghiệm mà nó phải thỏa mãn tất cả các hàm mục tiêu cần tối ưu ứng với miền chấp nhận được là X. Thực tế thì những nghiệm như vậy rất ít khi tồn tại. Nên ta đưa ra một số khái niệm khác về tối ưu có vẻ "mềm dẻo" hơn đó là nghiệm tối ưu Pareto.

- Một điểm $x^* \in X$ được gọi là một nghiệm tối ưu Pareto nếu không tồn tại một nghiệm $x \neq x^* \in X$ sao cho x trội hơn x*. Nghĩa là $f(x) < f(x^*)$.

- Một nghiệm $x= (x_1, x_2, \ldots, x_n)$ được gọi là trội hơn nghiệm $y= (y_1, y_2, \ldots, y_n)$ ký hiệu là: $x \leq y$, nếu:

$$\begin{cases} f_i(x) \leq f_i(y) \quad i \in \{1, \ldots, k\} \\ \exists j \in \{1, \ldots, n\} f_i(x) < f_i(y) \end{cases}$$

- $x = (x_1, x_2, \ldots, x_n)$ được gọi là nghiệm không trội hơn nghiệm $y = (y_1, y_2, \ldots, y_n)$ nếu $\forall x \in X, \nexists y \in X$ sao cho: $y >_X x$.

1.5. Giải thuật di truyền sử dụng để tối ưu hóa đa mục tiêu

1.5.1. Giới thiệu

Giải thuật di truyền (GA-Genetic Algorithms) [6] do D.E. Goldberg đề xuất, sau đó được L. Davis và Z. Michalevicz tiếp tục phát triển. GA được hình thành dựa trên quan niệm: *quá trình tiến hóa tự nhiên là quá trình hoàn hảo và hợp lý nhất, tự quá trình này đã mang tính tối ưu*. Quan niệm này là một tiên đề đúng, không chứng minh được nhưng phù hợp với thực tế khách quan.

GA là giải thuật tìm kiếm, chọn lựa các phương án tối ưu để giải quyết các bài toán thực tế khác nhau, dựa trên cơ chế chọn lọc của tự nhiên: từ tập lời giải ban đầu, thông qua nhiều bước tiến hoá, hình thành tập lời giải mới phù hợp hơn và cuối cùng dẫn đến lời giải tối ưu toàn cục.

Các giả thuyết thường được mô tả bằng các chuỗi bit, việc hiểu các chuỗi bit này tùy thuộc vào ứng dụng, ý tưởng các giả thuyết cũng có thể được mô tả bằng các biểu thức kí hiệu hoặc ngay cả các chương trình máy tính. Tìm kiếm giả thuyết thích hợp bắt đầu với một quần thể, hay một tập hợp có chọn lọc ban đầu của các giả thuyết. Các cá thể của quần thể hiện tại khởi nguồn cho quần thể thế hệ kế tiếp bằng các hoạt động chọn lọc, lai ghép và đột biến ngẫu nhiên – được lấy mẫu sau các quá trình tiến hóa sinh học.

GA đã được ứng dụng rộng rãi cho những bài toán cụ thể khác nhau và cho các vấn đề liên quan tới tối ưu hóa. Ví dụ, chúng đã được dùng để học tập luật điều khiển robot, để tối ưu hóa các bài toán đa mục tiêu và tôpô cho mạng nơron nhân tạo.

1.5.2. Các quy luật cơ bản

1.5.2.1. Quá trình chọn lọc

Các cá thể sẽ được chọn lọc theo độ thích nghi để tham gia vào quá trình tiến hóa tiếp theo. Các cá thể có độ thích nghi cao sẽ có cơ hội sống sót nhiều hơn và có thể có nhiều con trong thế hệ tiếp theo.

Phép chọn lọc các cá thể trong mỗi quần thể được thực hiện bởi bánh xe xổ số.

Quá trình chọn lọc được thực hiện như sau: quay bánh xe xổ số n lần, trong đó: n là số nghiệm của bài toán. Mỗi lần bánh xe dừng lại, một cá thể tương ứng sẽ bị rơi xuống rãnh, tức là đã được chọn. Cá thể đã được chọn sẽ có cơ hội sóng sót và di truyền lại cho thế hệ sau. Với cách thực hiện này, một số cá thể tốt sẽ được chọn nhiều lần và các cá thể xấu sẽ bị loại bỏ dần.

Mỗi quần thể P(t-1) gồm n nhiễm sắc thể P(t-1) = $\{v_1, v_2, \ldots v_n\}$. Khi đó xây dựng bánh xe xổ số sử dụng các tính toán:

- Để đánh giá độ thích nghi của quần thể, gọi là tổng độ thích nghi của quần thể, sử dụng:

$$F = \sum_{i=1}^{n} Eval(v_i) \qquad (1.20)$$

- Tính xác xuất chọn lọc của mỗi cá thể v_i:

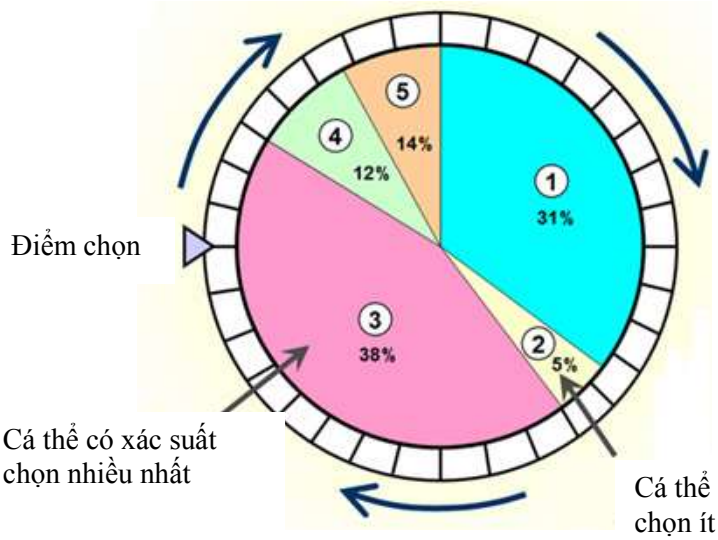
$$p_i = \frac{Eval(v_i)}{F} \qquad (1.21)$$

- Tính xác suất tích lũy q_i cho mỗi cá thể v_i:

$$q_i = \sum_{i=1}^{n} p_j, i = 1,2,...n \qquad (1.22)$$

- Thực hiện quá trình chọn lọc bằng các thực hiện chọn quần thể Q(t) từ quần thể p(t-1) dựa vào bánh xe xổ số:
  - Với mỗi số tự nhiên k = 1, 2…, n. Ta sinh một số thực ngẫu nhiên r_k trong đoạn [0,1].
  - Nếu $r_k < q_1$ thì chọn cá thể v_1, ngược lại, chọn cá thể v_i sao cho $q_{i-1} \le r_k \le q_i$, $2 \le i \le n$.

Xác suất chọn lọc cao nhất 38% là cá thể 1, tức là khi bánh xe xổ số quay, khả năng được chọn của nó là 0,38. Cá thể 2 có xác suất chọn lọc là 5%. Tương tự các cá thể 3, 4, 5 cũng có xác suất được xác định như trên (Hình 1.8).



Hình 1.8. Minh họa cho bánh xe xổ số với quần thể gồm 5 cá thể

1.5.2.2. Quá trình lai ghép

Quá trình này thể hiện bằng cách ghép 1 hay nhiều đoạn gen từ hai nhiễm sắc thể (NST) cha và mẹ để hình thành nên một NST mới mang đặc tính của cả cha và mẹ. Cụ thể như sau: Chọn ngẫu nhiên hai hay nhiều cá thể trong quần thể. Giả sử chuỗi NST của cha và mẹ đều có chiều dài là n. Ta sẽ tìm điểm lai bằng cách tạo ngẫu nhiên một con số từ 1 đến $n-1$, tức là, điểm lai vừa chọn sẽ chia hai chuỗi NST cha - mẹ thành hai nhóm NST con là $n_1$và n_2. Hai chuỗi NST con lúc này sẽ là $m_{11}+ m_{22}$ và $m_{21} + m_{12}$. Sau đó lại tiếp tục đưa hai NST con vào quần thể để tiếp tục tham gia quá trình tiến hóa.

Ví dụ: cho hai nhiễm sắc thể cha - mẹ như sau:

Chromosome1	10111 | 00100 | 110011
Chromosome 2	10101 | 11001 | 010110

Thực hiện lai ghép ở các đoạn như sau sẽ tạo ra hai con:

Offspring 1	10101 | 00100 | 010110
Offspring 2	10111 | 11001 | 110011

Lưu ý rằng, hai cá thể cha – mẹ với các đặc tính tốt cũng chưa chắc đã cho hai con có đặc tính tốt hơn so với cha – mẹ. Nhưng có thể thấy khả năng tạo ra các cá thể con tốt là rất cao. Nếu hai cá thể con không phải là một lời giải tốt thì có thể nó sẽ bị đào thải ở thế hệ tiếp theo.

1.5.2.3. Quá trình đột biến

Đưa nhiễm sắc thể con vào quần thể để tham gia tiếp vào quá trình tiến hóa. Quá trình đột biến là sự thay đổi một vài gen của một NST. Toán tử đột biến làm tăng nhanh

quá trình hội tụ, nhưng có thể sự tăng đột ngột không có tác dụng hoặc làm hội tụ sớm dẫn đến một lời giải kém tối ưu.

Trong GA cổ điển, mỗi cá thể biểu diễn bởi một chuỗi nhị phân. Do đó phép đột biến tại một vị trí nào đó là việc đảo bít tương ứng tại đúng vị trí đó.

Ví dụ:

Offspring	11101 10110 001100
Mutated Offspring	11111 10110 001000

1.5.2.4. Thủ tục GA

GA với mục đích giải quyết bài toán tối ưu như sau:

f(x) $\{x \in D \subset R^n\}$ -> max (min). Trong đó D là hình hộp trong không gian số thực n-chiều R^n, $f(x) > 0$ với $\forall x \in D$.

Mỗi x trong D được mã hóa bằng một chuỗi nhị phân (x_1, x_2,…, x_m) với m là độ dài của chuỗi. Một chuỗi biểu diễn một NST và mỗi x_i là một gen. Để đánh giá khả năng thích nghi của mỗi cá thể, xây dựng hàm :

$Eval(x_1,x_2,\ldots x_m) = f(x)$, với x là một vectơ tương ứng với $(x_1,x_2,\ldots.x_m)$

Thủ tục GA bao gồm các bước cụ thể:

Bước 1. Thế hệ T = 0; Khởi tạo ngẫu nhiên quần thể ban đầu P(0) gồm n cá thể;
Bước 2: Đánh giá độ thích nghi của quần thể ;
Bước 3: Lặp quá trình tiến hóa cho thế hệ T = T+1;
Bước 4: Chọn các cá thể tốt để sinh sản cho thế hệ (bởi bánh xe sổ số);
Bước 5: Tái tạo quần thể với các cá thể đã chọn bằng các toán tử di truyền;
Bước 6: Đánh giá quần thể vừa được tái tạo;
Bước 7: Kết thúc khi điều kiện được thỏa mãn.

```
Procedure GA
Begin
        t = 0;
        Khởi tạo P(t);
        Đánh giá (P(t));
        While (not E) do
        Begin
                t = t+1;
                Chọn Q(t) từ P(t-1)   {bằng bánh xe sổ số}
                Tái tạo P(t) từ Q(t)   {bằng các toán tử di truyền}
                Đánh giá P(t)
        End;
End;
```

CHƯƠNG 2. PHÂN CỤM ĐA MỤC TIÊU MỜ CHO DỮ LIỆU ĐỊNH DANH

Như đã giới thiệu, gần đây, vấn đề về phân cụm dữ liệu định danh đã thu hút sự quan tâm lớn của các nhà nghiên cứu. Một số thuật toán phân cụm với trọng tâm là phân cụm trên dữ liệu định danh đã được phát triển gần đây. Tuy nhiên, hầu hết những phương pháp này thường chỉ tối ưu hóa một mục tiêu của thuật toán phân cụm. Từ lý do này, [3, 4] đã đề xuất sử dụng thuật toán NSGA-II (Non-dominated Sorting Genetic Algorithm) dựa trên cơ sở tối ưu hóa đa mục tiêu. Hai mục tiêu là π và *Sep* [7] tương ứng là độ thuần nhất cụm mờ và độ phân tách cụm mờ cùng được tối ưu hóa. Trong phương pháp này đề xuất một phương pháp mới để lựa chọn một giải pháp phân cụm từ tập gần tối ưu Pareto ở thế hệ cuối cùng. Phương pháp chọn này dựa trên dựa trên kỹ thuật voting bởi các phương án gần tối ưu Pareto sau đó thực hiện phân lớp k-NN.

2.1. Giới thiệu

Như đã trình bày, phân cụm là hướng tiếp cận phân lớp không giám sát phổ biến trong đó một tập dữ liệu cho trước được phân thành các cụm khác nhau dựa trên một số độ đo mức độ giống nhau hoặc độ đo mức độ khác nhau. Nếu mỗi điểm dữ liệu được gán vào một cụm duy nhất, thì đó được gọi là phân cụm rõ. Còn nếu một điểm dữ liệu có độ thuộc nhất định vào từng cụm thì đó là phân cụm mờ.

Hầu hết các thuật toán phân cụm được thiết kế cho các tập dữ liệu trong đó khoảng cách giữa hai điểm dữ liệu bất kỳ có thể được tính bằng cách sử dụng các độ đo khoảng cách chuẩn ví dụ như độ đo Euclidean. Tuy nhiên nhiều bộ dữ liệu trong tự nhiên chứa thuộc tính định danh. Với dữ liệu định danh, các phần tử trong miền giá trị của thuộc tính không có thứ tự. Trong trường hợp đó, các thuật toán phân cụm như K-means [5], C-means mờ (fuzzy C-means - FCM) [8], … không thể áp dụng được. Thuật toán K-means tính trung tâm của một cụm bằng cách tính giá trị trung bình của tập các vectơ thuộc cụm đó. Tuy nhiên, với cơ sở dữ liệu danh, việc toán giá trị trung bình của một tập các vectơ là không có ý nghĩa. Một biến thể của thuật toán K -means, cụ thể là thuật toán PAM – *phân vùng quanh trọng tâm* hoặc K - medoids đã được đề xuất cho loại dữ liệu này. Trong giải thuật PAM, thay vì xác định các trung tâm cụm (*cluster center*), người ta đi xác định các trọng tâm cụm (*cluster medoid*) là các điểm nằm ở vị trí trung tâm nhất trong một cụm. Không giống như các cluster center, một cluster medoid phải là một điểm dữ liệu thực tế thuộc cụm. Một mở rộng khác của K -means là K - mode [11]. Ở đây, cluster center được thay thế bằng *cluster mode* (được mô tả trong phần 2.2). Một phiên bản mờ của thuật toán K – mode là K – mode mờ, cũng được đề xuất. Gần đây, một thuật toán phân cụm dữ liệu định danh dựa trên khoảng cách Hamming (HD) đã được phát triển. Tuy nhiên, tất cả các thuật toán nói trên đều dựa trên tối ưu đơn mục tiêu để phân vùng. Một hàm đơn mục tiêu có thể không mang lại kết quả

tốt và ổn định trên các loại dữ liệu định danh khác nhau. Do đó, một nhu cầu tự nhiên là tối ưu cùng lúc nhiều mục tiêu.

Giải thuật di truyền (GAs) [6] là một chiến lược tìm kiếm phương án tối ưu trên toàn cục dựa trên thuyết tiến hóa của Darwin. Mặc dù các thuật toán di truyền trước đó đã được sử dụng trong việc phân cụm dữ liệu, nhưng hầu hết chỉ tối ưu hóa một mục tiêu duy nhất do đó khó áp dụng cho tất cả các loại dữ liệu. Để giải quyết nhiều vấn đề thực tế, việc cần thiết là phải tối ưu hóa nhiều hơn một mục tiêu cùng lúc. Phân cụm là một vấn đề thực tế rất có ý nghĩa và các thuật toán phân cụm thường cố gắng tối ưu hóa một chỉ số quan trọng nào đó như độ thuần nhất cụm, độ tách cụm hoặc kết hợp cả hai. (Vấn đề phân cụm dữ liệu thêm phức tạp bởi không thể xác định được trung bình của các phần tử trong một cụm). Tuy nhiên, khi coi tầm quan trọng của các tiêu chí phân cụm có ý nghĩa như nhau, việc tối ưu hóa đồng thời độ thuần nhất và độ tách cụm một cách riêng rẽ sẽ tốt hơn việc kết hợp chúng lại thành một hàm mục tiêu duy nhất. Từ lý do trên, bài toán phân cụm mờ dữ liệu định danh được mô hình hóa như một trong những bài toán tối ưu hóa đa mục tiêu trong đó việc tìm kiếm được thực hiện trên một số hàm có mục tiêu thường là xung đột nhau. Về vấn đề này, các giải thuật di truyền đa mục tiêu được dùng để xác định các cluster centers (modes) và các ma trận độ thuộc. Giải thuật di truyền dựa trên thuật toán sắp xếp không vượt trội (Non-dominated sorting GA-II - NSGA-II) [12], là một giải thuật tốt được dùng rất phổ biến là một phương pháp tối ưu hóa cơ bản. Hai hàm mục tiêu là độ thuần nhất và độ tách cụm được tối ưu hóa đồng thời.

2.2. Thuật toán phân cụm mờ cho dữ liệu định danh [4]

Phần này mô tả thuật toán phân cụm K – mode mờ cho bộ dữ liệu định danh. Thuật toán K – mode mờ [14] là mở rộng của thuật toán nổi tiếng FCM cho dữ liệu định danh. Gọi $X=\{x_1, x_2, \ldots, x_n\}$ là một tập n đối tượng có thuộc tính định danh. Mỗi đối tượng x_i, $i = 1, 2, \ldots, n$ được mô tả bởi một tập gồm p thuộc tính $A_1, A_2, \ldots, A_p$. Gọi $DOM(A_j)$, $1 \le j \le p$, là miền giá trị của thuộc tính thứ j và nó chứa q_j giá trị định danh khác nhau: $DOM(A_j)=\{a_j^1, a_j^2, \ldots, a_j^{q_j}\}$. Khi đó, đối tượng định danh thứ i được xác định là $x_i=[x_{i1}, x_{i2}, \ldots, x_{ip}]$ trong đó $x_{ij} \in DOM(A_j)$, $1 \le j \le p$.

Các cluster centers trong thuật toán FCM được thay thế bởi các cluster modes trong phân cụm mờ K-modes. Một cluster mode được định nghĩa như sau: Gọi C_i là một tập các đối tượng định danh thuộc cụm thứ i, mỗi đối tượng được mô tả bởi các thuộc tính $A_1, A_2, \ldots, A_p$. Mode của cụm C_i là một vectơ $m_i = [m_{i1}, m_{i2}, \ldots, m_{ip}]$, $m_{ij} \in DOM(A_j)$, $1 \le j \le p$ sao cho đại lượng sau được tối thiểu hóa:

$$D(\mathrm{m_i}, \mathrm{C_i}) = \sum_{\mathrm{x}\in \mathrm{C_i}} \mathrm{D}(\mathrm{m_i}, \mathrm{x}) \qquad (2.1)$$

Ở đây, $D(m_i, x)$ là khoảng cách giữa m_i và x. Lưu ý rằng m_i không nhất thiết phải là một phần tử của tập C_i. Thuật toán K-mode mờ phân vùng bộ dữ liệu X thành K cụm để cực tiểu hóa hàm:

$$\mathrm{J_m}(\mathrm{U}, \mathrm{Z}:\mathrm{X}) = \sum_{\mathrm{k=1}}^{\mathrm{n}} \sum_{\mathrm{i=1}}^{\mathrm{K}} \mathrm{u_{ik}^m D}(\mathrm{z_i}, \mathrm{x_k}) \qquad (2.2)$$

Với các ràng buộc:

$$0 \le u_{ik} \le 1, \quad 1 \le i \le K, 1 \le k \le n \qquad (2.3)$$

$$\sum_{i=1}^{K} u_{ik} = 1, \quad 1 \le k \le n \qquad (2.4)$$

và

$$0 < \sum_{k=1}^{n} u_{ik} < n, \quad 1 \le i \le k \qquad (2.5)$$

trong đó m là số mũ mờ, $U = [u_{ik}]$ là ma trận độ thuộc K x n; u_{ik} là độ thuộc của đối tượng định danh thứ k vào cụm i. $Z = \{z_1, z_2, \ldots, z_K\}$ biểu diễn tập các modes.

Thuật toán K-mode mờ dựa trên một chiến lược tối ưu hóa ở đó lặp đi lặp lại việc ước lượng ma trận độ thuộc ở lần lặp tiếp theo và tính toán lại mode của các cụm mới. Bắt đầu bằng cách lấy ngẫu nhiên K modes. Sau đó, trong mỗi lần lặp ta tính các giá trị độ thuộc của từng điểm dữ liệu tới từng cụm bằng cách sử dụng công thức sau đây:

$$\mathrm{u_{ik}} = \frac{1}{\sum_{\mathrm{j=1}}^{\mathrm{K}} \left(\frac{\mathrm{D}(\mathrm{z_i}, \mathrm{x_k})}{\mathrm{D}(\mathrm{z_j}, \mathrm{x_k})} \right)^{\frac{1}{\mathrm{m-1}}}} \text{ với } 1 \le i \le K, 1 \le k \le n \quad (2.6)$$

Lưu ý rằng khi tính giá trị u_{ik} sử dụng công thức (2.6), nếu $D(z_j, x_k)$ bằng 0 đối với một số giá trị của j thì u_{ik} được gán bằng 0 cho tất cả các giá trị i = 1, 2, …, K, $i \neq j$, còn lại thì u_{jk} được gán bằng 1.

Dựa trên các giá trị độ thuộc, các modes được tính lại như sau [14]: $z_i = [z_{i1}, z_{i2}, \ldots, z_{ip}]$, trong đó $z_{ij} = a_j^r \in DOM(A_j)$ và

$$\sum_{k, x_{kj}=a_j^r} u_{ik}^m \ge \sum_{k, x_{kj}=a_j^t} u_{ik}^m, \quad 1 \le t \le q_j, \quad r \ne t \qquad (2.7)$$

Thuật toán kết thúc khi không cải thiện thêm được giá trị J_m (cố định số lần lặp). Cuối cùng, mỗi đối tượng được gán vào cụm mà nó có độ thuộc lớn nhất. Nhược điểm

chính của thuật toán phân cụm mờ K-modes là: (i) phụ thuộc rất lớn vào việc lựa chọn giá trị khởi tạo các mode và (ii) thường bị rơi vào tối ưu địa phương.

2.3. Tối ưu hóa đa mục tiêu và các giải thuật tối ưu hóa đa mục tiêu

2.3.1. Tối ưu hóa đa mục tiêu

Bài toán tối ưu hóa đa mục tiêu có thể phát biểu như sau. Tìm vectơ $\bar{x}^* = [x_1^*, x_2^*, \dots, x_n^*]^T$ của các biến quyết định thỏa mãn *m* ràng buộc bất đẳng thức:

$$g_i(\bar{x}) \geq 0, i = 1, 2, ..., m \qquad (2.8)$$

và p ràng buộc đẳng thức:

$$h_i(\bar{x}) = 0, i = 1, 2, ..., p \qquad (2.9)$$

và tối ưu hóa vectơ hàm:

$$\bar{f}(\bar{x}) = [f_1(\bar{x}), f_2(\bar{x}), ..., f_k(\bar{x})]^T \qquad (2.10)$$

Các ràng buộc (2.8) và (2.9) xác định miền khả thi F chứa tất cả các phương án có thể chấp nhận được. Bất kì phương án nào nằm ngoài miền này sẽ không được chấp nhận do nó vi phạm một hoặc nhiều ràng buộc. Vector $\bar{x}$* biểu diễn một phương án tối ưu trong F.

Khái niệm tối ưu Pareto rất quen thuộc trong lĩnh vực tối ưu hóa đa mục tiêu. Một định nghĩa hình thức về tối ưu Pareto từ góc nhìn của bài toán cực tiểu hóa có thể được phát biểu như sau: Một vector quyết định $\bar{x}$* được gọi là tối ưu Pareto khi và chỉ khi không có vector $\bar{x}$ vượt trội $\bar{x}$*, tức là, không có $\bar{x}$ mà

$$\forall i \in \{1,2,...,k\}, f_i(\bar{x}) \leq f_i(\bar{x}^*) \quad \text{và}$$
$$\exists i \in \{1,2,...,k\}, f_i(\bar{x}) < f_i(\bar{x}^*)$$

Nói cách khác, $\bar{x}$* là tối ưu Pareto nếu không tồn tại phương án chấp nhận được $\bar{x}$ nào làm giảm giá trị một số hàm mục tiêu mà không làm tăng giá trị một hàm mục tiêu nào khác. Ta cũng nói $\bar{x}$* không bị vượt trội trong miền F.

Tập tất cả các phương án chấp nhận được không bị vượt trội trong miền F được gọi là *tập tối ưu Pareto* (Pareto optimal set). Với tập tối ưu Pareto đã cho, các giá trị hàm mục tiêu tương ứng trong không gian mục tiêu được gọi là *Pareto Front*.

Mục tiêu của các giải thuật tối ưu đa mục tiêu là xác định các lời giải trong tập tối ưu Pareto. Thực tế, việc chứng minh một lời giải là tối ưu thường không khả thi về mặt tính toán. Vì vậy, một tiếp cận thực tế với bài toán tối ưu đa mục tiêu là tìm kiếm

tập các lời giải là thể hiện tốt nhất có thể của tập tối ưu Pareto, một tập các lời giải như vậy được gọi là *tập Pareto-được-biết-tốt-nhất* (Best-known Pareto set).

2.3.2. Việc sử dụng giải thuật di truyền giải quyết bài toán tối ưu đa mục tiêu

Khó khăn chính trong tối ưu đa mục tiêu là không tồn tại một một phương án tối ưu duy nhất và rất khó so sánh phương án này với phương án khác. Các bài toán này thường chấp nhận nhiều phương án mà mỗi phương án là chấp nhận được đối với mỗi hàm mục tiêu đồng thời đạt được sự cân đối giữa các mục tiêu. Giải thuật di truyền là phương pháp dựa vào quần thể, do đó nó có thể dễ được mở rộng để xử lý nhiều mục tiêu. Các giải thuật tiến hóa truyền thống có thể được cải tiến để tìm kiếm tập Pareto-được-biết-tốt-nhất trong bài toán tối ưu đa mục tiêu. Giải thuật tiến hóa là cách tiếp cận meta-heuristic được ưa chuộng nhất để giải bài toán tối ưu hóa đa mục tiêu. Trong số các phương pháp tối ưu hóa đa mục tiêu dựa vào meta-heuristic, 70% các phương pháp là dựa vào giải thuật di truyền [6].

Điểm khác biệt giữa các giải thuật GA đa mục tiêu nằm ở cách gán *độ thích nghi* (fitness assignment), cách *duy trì quần thể ưu tú* (elitism) và cách *tiếp cận nhằm đa dạng hóa quần thể* [6].

Phương pháp thường dùng để gán độ thích nghi là *xếp hạng Pareto* (Pareto ranking) được mô tả như sau: Phương pháp này làm việc bằng cách gán thứ hạng 1 cho các cá thể không bị vượt trội trong quần thể và đưa chúng ra ngoài vòng xem xét; rồi tìm tập cá thể không bị vượt trội mới để gán thứ hạng 2 và quá trình cứ tiếp tục như vậy.

Phương pháp thường dùng để đa dạng hóa quần thể là *chia sẻ độ thích nghi* (fitness sharing). Phương pháp chia sẻ độ thích nghi khuyến khích tìm kiếm trên những vùng chưa được thăm dò của *Pareto front* bằng cách giảm bớt độ thích nghi của các lời giải ở những vùng cá thể mật độ cao. Kỹ thuật chia sẻ độ thích nghi với *số đếm vùng lân cận* (niche count) và dùng *khoảng cách mật độ* (crowding distance) mà được mô tả như sau:

- Chia sẻ độ thích nghi dựa vào số đếm vùng lân cận.

Phương pháp này đòi hỏi phải giảm bớt độ thích nghi f_i của một cá thể i bằng cách chia nó cho số đếm vùng lân cận m_i được tính cho cá thể đó. Tức là độ thích nghi dùng chung được tính bằng f_i/m_i. Số đếm vùng lân cận m_i là giá trị ước lượng vùng lân cận của cá thể i đông đúc như thế nào. Nó được tính cho từng cá thể trong quần thể hiện hành theo công thức: $m_i = \sum_{j \in Pop} Sh[d[i, j]]$, với $d[i, j]$ là khoảng cách Euclid giữa hai cá thể i và j và $Sh[d]$ là hàm chia sẻ (sharing function). $Sh[d]$ là một hàm của $d[i, j]$ sao cho $Sh[0] = 1$ và $Sh[d \geq \sigma_{share}] = 0$. Thông thường $Sh[d] = 1\text{-}d/\sigma_{share}$ với $d \leq \sigma_{share}$ và $Sh[d] = 0$

với $d \geq \sigma_{share}$. Ở đây σ_{share} là *bán kính vùng lân cận*, được người dùng xác định để ước lượng độ cách biệt tối thiểu mong muốn giữa hai lời giải cuối cùng. Các cá thể có khoảng cách trong phạm vi σ_{share} bị giảm bớt độ thích nghi vì chúng ở trong cùng vùng lân cận.

• Phương pháp dùng khoảng cách mật độ

Phương pháp này đòi hỏi tính khoảng cách mật độ là giá trị ước lượng mật độ lời giải bao quanh một điểm được xét *i* trong quần thể. Đại lượng này là giá trị trung bình của hai điểm lấy hai bên của điểm được xét *i* dọc theo mỗi trục mục tiêu. Đại lượng này được dùng trong cơ chế chọn cha mẹ như sau: lấy ngẫu nhiên hai lời giải *x* và *y*; nếu chúng có cùng *mức không vượt trội* (non-domination rank) thì lời giải nào có khoảng cách mật độ cao hơn sẽ được chọn; ngược lại lời giải có mức không vượt trội thấp hơn sẽ được chọn.

Bên cạnh đó, việc duy trì quần thể ưu tú là một vấn đề quan trọng trong tối ưu hóa đa mục tiêu bằng giải thuật GA đa mục tiêu. Trong ngữ cảnh của giải thuật GA đa mục tiêu, tất cả những lời giải không bị vượt trội được phát hiện bởi giải thuật GA đa mục tiêu được coi như là những *lời giải ưu tú*. Có hai chiến lược thường dùng để hiện thực việc duy trì quần thể ưu tú: (i) lưu trữ các lời giải ưu tú trong chính quần thể và (ii) lưu trữ các lời giải ưu tú trong một danh sách thứ cấp bên ngoài quần thể và đưa chúng trở lại quần thể.

NSGA-II (Non-dominated sorting GA-II) là một trong những giải thuật GA đa mục tiêu được phát triển gần đây và được sử dung rộng rãi. Đặc điểm của giải thuật NSGA-II [12] được mô tả sơ lược như sau:

Gán độ thích nghi: xếp hạng dựa vào sắp thứ tự mức độ không vượt trội (non-domination sorting).
Cơ chế đa dạng hóa: phương pháp dùng khoảng cách mật độ (crowding distance).
Cách duy trì quần thể ưu tú: có.

2.4. Phân cụm đa mục tiêu mờ cho dữ liệu định danh sử dụng giải thuật di truyền

Phần này trình bày việc sử dụng thuật toán tối ưu đa mục tiêu dựa trên giải thuật di truyền NSGA - II để tiến hóa một tập các phương án gần tối ưu Pareto (near-Pareto-optimal) nhằm giải quyết bài toán phân cụm mờ cho dữ liệu định danh [4].

2.4.1. Thuật toán NSGA-II

Các bước chính của thuật toán NSGA-II:

1. Procedure NSGA-II
2. Begin
3. t = 0;
4. Khởi tạo P(t);
5. Sắp xếp không vượt trội và tính toán crowding distance cho P(t);
6. While (t<gen) do
7. Begin
8. t = t+1;
9. Chọn Q(t) từ P(t-1) dựa vào rank và crowding distance;
10. Q(t)=Lai ghép trên Q(t);
11. Q(t)=Đột biến trên Q(t);
12. Offspring=P(t-1) + Q(t);
13. Sắp xếp không vượt trội và tính toán crowding distance cho Offspring;
14. P(t)=Chọn N cá thể tốt nhất từ Offspring dựa vào rank và crowding distance;
15. End;
16. Trích lời giải;
17. End;

Trong đó:

- Khi áp dụng vào bài toán phân cụm đa mục tiêu mờ cho dữ liệu định danh, ta cần xác định cách thức biểu diễn nhiễm sắc thể tương ứng. Nội dung này được trình bày trong Phần 2.4.2.
- Cách khởi tạo quần thể (*dòng 4*) được trình bày trong Phần 2.4.3.
- Thủ tục sắp xếp không vượt trội và tính toán crowding distance (sử dụng ở *dòng 5* và *dòng 13*) được trình bày ở Phần 2.4.5. Việc sắp xếp không vượt trội và tính toán crowding distance dựa trên giá trị các hàm mục tiêu. Việc tính toán giá trị các hàm mục tiêu được trình bày trong Phần 2.4.4.
- Toán tử chọn lọc (dòng 9), lai ghép (dòng 10), và đột biến (dòng 11) được trình bày trong Phần 2.4.6.
- Ở dòng 9, Q(t) có kích thước đúng bằng kích thước quần thể (*N*). Do đó, ở dòng 12, Offspring có kích thước bằng *2N* (là gộp của P(t-1) và Q(t)). Cách làm này nhằm đảm bảo thế hệ sau chắc chắn thu được các cá thể tốt hơn (được chọn từ Offspring, thực hiện ở dòng 14).
- Cách chọn một cá thể từ thế hệ cuối cùng (ở dòng 16) được trình bày ở Phần 2.4.7.

2.4.2. Biểu diễn nhiễm sắc thể

Mỗi nhiễm sắc thể là một chuỗi các giá trị thuộc tính biểu diễn các điểm trung tâm (mode) của K cụm. Như vậy, mỗi nhiễm sắc thể biểu diễn một lời giải cho bài toán phân cụm. Nếu mỗi đối tượng định danh có p thuộc tính $\{A_1, A_2, ..., A_p\}$ thì chiều dài của một nhiễm sắc thể sẽ là K x p. Trong đó, p vị trí (gen) đầu tiên biểu diễn mode của cụm đầu tiên (là một véc tơ p-chiều); p vị trí tiếp theo biểu diễn mode của cụm thứ hai. Cứ như thế, từng véc tơ p-chiều của các mode của các cụm lần lượt được nối lại để tạo nên một nhiễm sắc thể. Để minh họa, xét ví dụ sau đây. Cho p = 3và K = 3. Khi đó, nhiễm sắc thể

$$c_{11} \quad c_{12} \quad c_{13} \quad c_{21} \quad c_{22} \quad c_{23} \quad c_{31} \quad c_{32} \quad c_{33}$$

biểu diễn một phương án phân cụm, là ghép của 3 véc tơ biểu diễn 3 mode của 3 cụm. Ba véc tơ đó là (c_{11}, c_{12}, c_{13}), (c_{21}, c_{22}, c_{23}) và (c_{31}, c_{32}, c_{33}), trong đó c_{ij} biểu diễn giá trị thuộc tính thứ j của mode của cụm thứ i; $c_{ij} \in DOM(A_j)$, $1 \leq i \leq K, 1 \leq j \leq p$.

2.4.3. Khởi tạo quần thể

Mỗi cá thể trong quần thể đầu tiên (quần thể khởi tạo) được sinh ra bằng cách chọn ngẫu nhiên K đối tượng thuộc cơ sở dữ liệu định danh đầu vào cần phân cụm. Mỗi cá thể được biểu diễn bởi một NST như trình bày trong phần 2.4.2. Nếu kích thước quần thể là P thì việc chọn ngẫu nhiên ra K đối tượng để hình thành một phương án phân cụm, tức là hình thành một nhiễm sắc thể, được lặp lại P lần.

2.4.4. Tính toán giá trị của các hàm mục tiêu

Độ thuần nhất tổng thể của các cụm π và độ phân tách cụm *Sep* được coi là hai mục tiêu cần phải được tối ưu hóa đồng thời. Như vậy, số lượng hàm mục tiêu trong bài toán tối ưu đa mục tiêu ở đây là 2. Việc tính toán giá trị của các hàm mục tiêu được thực hiện như mô tả dưới đây:

Giả sử ta có một NST biểu diễn một phương án phân cụm với K mode của các cụm là $z_1, z_2, \ldots, z_K$. Khi đó, các giá trị độ thuộc u_{ik}, $i=1, 2,..., K$ và $k=1, 2,..., n$, được tính như sau:

$$u_{ik} = \frac{1}{\sum_{j=1}^{K} \left(\frac{D(z_i, x_k)}{D(z_j, x_k)} \right)^{\frac{1}{m-1}}} \quad \text{với } 1 \leq i \leq K, 1 \leq k \leq n \qquad (2.11)$$

trong đó $D(z_i,x_k)$ là khoảng cách giữa z_i và x_k, $D(z_j,x_k)$ là khoảng cách giữa z_j và x_k (sử dụng khoảng cách Hamming đã được mô tả phía trước). m là trọng số. Lưu ý rằng khi

tính toán u_{ik} (công thức 3.11), nếu tồn tại j mà $D(z_j, x_k) = 0$ thì ta gán u_{ik} bằng 0 cho tất cả các giá trị $i=1,..., K, i \neq j$, còn u_{jk} được gán bằng 1. Sau đó, mode được cập nhật: $z_i = [z_{i1}, z_{i2},...,z_{ip}]$, trong đó $z_{ij} = a_j^r \in DOM(A_j)$ sao cho:

$$\sum_{k, x_{kj}=a_j^r} u_{ik}^m \geq \sum_{k, x_{kj}=a_j^t} u_{ik}^m \quad , \quad 1 \leq t \leq q_j, r \neq t \tag{2.12}$$

Điều này có nghĩa là giá trị thuộc tính định danh A_j của trung tâm cụm z_i được thiết lập để các giá trị định danh đạt giá trị tối đa của tổng u_{ij} (mức độ thuộc vào cụm thứ *i*) trên tất cả các định danh. Theo đó, các mức độ thuộc vào cụm được tính toán lại theo công thức (3.16).

Các biến δ_i và bản số mờ n_i của cụm thứ *i*, *i=1, 2,..., K* được tính toán bằng cách sử dụng phương trình sau:

$$\delta_i = \sum_{i=1}^{n} u_{ik}^m D(z_i, x_k) \quad , 1 \leq i \leq K \tag{2.13}$$

và

$$n_i = \sum_{k=1}^{n} u_{ik} \quad , 1 \leq i \leq K \tag{2.14}$$

Độ thuần nhất tổng thể π được đại diện bởi NST, được tính như sau:

$$\pi = \sum_{i=1}^{K} \frac{\delta_i}{n_i} = \sum_{i=1}^{K} \frac{\sum_{k=1}^{n} u_{ik}^m D(z_i, x_k)}{\sum_{k=1}^{n} u_{ik}} \tag{2.15}$$

Để tính toán các hàm tách cụm mờ phù hợp *Sep*, giả định z_i của cụm thứ *i* là trung tâm của tập mờ $\{z_i | 1 \leq j \leq K, j \neq i\}$. Do đó mức độ thành viên của mỗi z_j để $j \neq i$ được tính như sau:

$$\mu_{ij} = \frac{1}{\sum_{l=1, l \neq j}^{K} \left(\frac{D(z_j, z_i)}{D(z_j, z_l)}\right)^{\frac{1}{m-1}}}, i \neq j \tag{2.16}$$

Chỉ số tách cụm mờ được định nghĩa:

$$Sep = \sum_{i=1}^{K} \sum_{j=1, j \neq i}^{K} \mu_{ij}^m D(z_i, z_j) \tag{2.17}$$

Lưu ý rằng để thu được cụm nhỏ gọn, chỉ số π nên được tối thiểu. Ngược lại, để tách cụm được tốt, chỉ số tách cụm mờ *Sep* nên được tối đa. Vấn đề trong luận văn này nghiên cứu thì vấn đề tối ưu đa mục tiêu được đặt ra là giảm thiểu cả hai mục tiêu tức là giảm cả π và 1/Sep cùng một lúc.

Tương tự như vậy việc phân nhóm đa mục tiêu với việc tối ưu hóa đồng thời nhiều mục tiêu, hiệu quả phân cụm phụ thuộc rất lớn vào việc lựa chọn các mục tiêu này. Do đó phải rất cẩn thận khi lựa chọn các mục tiêu để có thể tạo ra kết quả khả quan, nếu tùy tiện hoặc không tính toán các lựa chọn có thể dẫn đến các tình huống xấu.

Nên lựa chọn các mục tiêu sao cho cân bằng và có thể dẫn đến mâu thuẫn một cách tự nhiên. Mâu thuẫn trong các hàm mục tiêu là có lợi vì nó dẫn đến một phương án tối ưu tổng quát. Nó cũng đảm bảo rằng không có phân cụm đơn mục tiêu nào được tối ưu hóa mà các mục tiêu quan trọng lại không được chú ý.

Mặc dù tồn tại một vài giá trị chỉ số cụm nhưng để tốt hơn nên xem xét độ thuần nhất cụm và độ tách cụm trong các mẫu. Do đó, trong nghiên cứu này đã lựa chọn để tối ưu hóa các độ thuần nhất cụm mờ tổng quát π (phản xạ của cụm đồng nhất) và độ tách cụm mờ *Sep* (phản xạ cụm tách). Mục đích của luận văn là thiết lập tính hiệu quả cho nguyên tắc cơ bản thực hiện việc phân cụm đa mục tiêu mờ cho dữ liệu định danh. Tuy nhiên hướng phát triển tiếp theo của luận văn là một nghiên cứu sâu hơn liên quan đến hai hoặc nhiều giá trị chỉ số cụm mờ.

2.4.5. Thủ tục sắp xếp không vượt trội và tính toán khoảng cách mật độ

Sắp xếp không không vượt trội:

Với một quần thể P, thủ tục sắp xếp không vượt trội được thực hiện như sau:

- ***Bước 1:*** Với mỗi cá thể p trong quần thể P ban đầu, làm như sau:
  - Khởi tạo $S_p = \emptyset$ là tập chứa tất cả các cá thể không vượt trội hơn p.
  - Khởi tạo $n_p = 0$ là số lượng cá thể vượt trội hơn p.
  - Với mỗi cá thể q thuộc P:

    * Nếu p vượt trội hơn q thì thêm p vào tập S_p tức là $S_p = S_p \cup q$

    * Ngược lại, nếu q vượt trội hơn p thì tăng số lượng cá thể vượt trội hơn p tức là $n_p = n_p + 1$

  - Nếu $n_p = 0$ tức là không có cá thể nào vượt trội hơn p thì p thuộc về front đầu tiên. Đặt xếp hạng của cá thể p là 1 tức là $p_{rank} = 1$. Cập nhật front đầu tiên, thiết lập bằng cách thêm p vào front 1 tức là $F_1 = F_1 \cup p$.
- ***Bước 2:*** Khởi tạo biến đếm front bằng 1, i = 1.
- ***Bước 3:*** Nếu front thứ i khác rỗng, tức là $F_i \neq \emptyset$, thực hiện:
  - $Q = \emptyset$ (khởi tạo tập chứa các cá thể của front thứ i+1)
  - Với mỗi cá thể p trong front F_i

    * với mỗi cá thể q trong S_p (S_p là tập các cá thể không vượt trội hơn p):

      . $n_q = n_q - 1$, giảm giá trị biến đếm vượt trội của cá thể q;

. sau khi giảm, kiểm tra nếu $n_q = 0$ thì không có cá thể nào trong front tiếp theo vượt trội hơn q, do đó đặt $q_{rank} = i + 1$; cập nhật cá thể q vào tập Q tức là Q = Q $\cup$ q.

- ***Bước 4:*** Tăng biến đếm front lên 1 đơn vị, i = i + 1. Ghi nhận front tiếp theo, F_i = Q; Quay lại ***Bước 3***.

Tính khoảng cách mật độ:

Sau khi việc sắp xếp không vượt trội được hoàn thành thì khoảng cách mật độ sẽ được tính toán. Thông tin xếp hạng và khoảng cách mật độ sẽ được dùng để phục vụ cho việc lựa chọn cá thể. Khi cần chọn một cá thể tốt, trước hết ta căn cứ vào thông tin xếp hạng. Nếu có nhiều hơn 1 cá thể có cùng hạng có thể chọn thì xét đến khoảng cách mật độ. Khoảng cách mật độ được tính như sau:

- Với mỗi front F_i gồm n cá thể
  - Khởi tạo khoảng cách bằng 0 cho tất cả các cá thể tức là $F_i(d_j) = 0$, trong đó j tương ứng với các thể thứ j trong front F_i.
  - Với mỗi hàm mục tiêu m
    * Sắp xếp các cá thể trong front F_i dựa trên mục tiêu m, tức là $I=sort(F_i,m)$
    * Gán khoảng cách bằng ∞ cho các giá trị biên của mỗi cá thể trong F_i, tức là $I(d_1) = \infty$ và $I(d_n) = \infty$.
    * Với k = 2 đến (n - 1)

$$I(d_k) = I(d_k) + \frac{I(k+1).m - I(k-1).m}{f_m^{max} - f_m^{min}} \qquad (2.18)$$

$I(k).m$ là giá trị của hàm mục tiêu thứ m của cá thể thứ k trong I

Ý tưởng cơ bản của khoảng cách mật độ là tìm khoảng cách euclidian của từng cá thể trong mỗi front cơ bản dựa trên m mục tiêu trong không gian đa chiều m. Các cá thể ở gần biên luôn được lựa chọn vì khoảng cách của chúng là vô hạn.

2.4.6. Chọn lọc, lai ghép và đột biến

Chọn lọc: Các cha – mẹ được chọn từ quần thể bằng cách sử dụng toán tử chọn lọc mật độ nhị phân dựa trên xếp hạng và khoảng cách mật độ (khoảng cách mật độ là một độ đo đo khoảng cách của một cá thể đến lân cận của nó). Từng cá thể được chọn lọc dựa trên việc xếp hạng của nó thấp hơn so với cá thể khác hoặc khoảng cách mật độ lớn hơn so với cá thể khác. Sau khi đã tìm ra các cá thể cho chúng vào hồ giao phối (matting pool) chuẩn bị cho lai ghép.

Lai ghép: sau quá trình chọn lọc, những cá thể được chọn cho quá trình lai ghép được tiến hành lai ghép đơn điểm (1 point) để sinh ra các cá thể mới. Số lượng cá thể tham gia vào lai ghép phụ thuộc tham số μ_c.

Đột biến: xác suất xảy ra đột biến là μ_m. Nếu một cá thể được lựa chọn để tiến hành đột biến thì quá trình đột biến sẽ xảy ra như sau: (i) chọn vị trí đột biến trong cá thể đó, (ii) tại vị trí đột biến, giá trị của vị trí đó sẽ được thay thế bởi 1 giá trị ngẫu nhiên trong miền giá trị (miền giá trị ở đây là giá trị mà mỗi gen có thể nhận được).

Quần thể ban đầu cùng với quần thể hiện tại được sắp xếp lại dựa trên tập không vượt trội và chỉ có cá thể tồn tại tốt nhất mới được lựa chọn trong đó N là kích thước của quần thể. Việc lựa chọn dựa trên việc xếp hạng và khoảng cách mật độ được thực hiện cuối cùng.

Điểm đặc trưng nhất của giải thuật NGSA-II là cách duy trì quần thể ưu tú (elitism operation). Trong đó, những cá thể trong quần thể hiện tại có khả năng được lựa chọn vào thế hệ tiếp theo. Dùng tập Pareto ở thế hệ cuối cùng sẽ cho ra những phương án khác nhau trong các bài toán về phân cụm.

2.4.7. Chọn một phương án từ các tập không vượt trội

Hiệu quả của một giải thuật di truyền đa mục tiêu là một tập đại diện các phương án không vượt trội trong thế hệ cuối cùng cân bằng giữa các mục tiêu khác nhau với hàm tối ưu hóa. Do đó cần có một phương pháp để xác định một phương án cuối cùng từ các tập không vượt trội. Vấn đề này đã được đề cập đến trong nhiều công trình nghiên cứu, trong đó tập trung vào các phương án nằm ở miền "knee" của front không vượt trội.

Các kỹ thuật đưa ra trong luận văn được sử dụng để tìm kiếm Pareto front xấp xỉ hoàn chỉnh và được dùng để xác định các phương án tốt nhất đó là phương án có nhiều điểm chung nhất trong các phương án ở thế hệ cuối cùng. Trong phương pháp này, tất cả các phương án được đưa ra có tầm quan trọng như nhau và ý tưởng là tìm ra phương án dựa trên việc kết hợp thông tin từ tất cả các phương án. Về vấn đề này, một kỹ thuật biểu quyết đa số bởi kỹ thuật phân lớp *k-nn* đã được đưa ra để chọn một phương án duy nhất từ tập các phương án không vượt trội.

Đầu tiên các vectơ nhãn phân cụm được tính từ phương án không vượt trội đưa ra bởi kỹ thuật đa mục tiêu. Thực hiện điều này bằng cách gán mỗi điểm dữ liệu vào các cụm có độ thuộc cao nhất. Sau đó, một kỹ thuật biểu quyết đa số được dùng để gán nhãn. Trước khi áp dụng biểu quyết đa số, phải đảm bảo sự thống nhất giữa các vectơ nhãn

của các phương án khác nhau, ví dụ cụm *i* của phương án đầu tiên phải phù hợp với cụm *i* của tất cả các phương án khác. Cách thực hiện như sau:

Đặt $X = \{l_1, l_2, \ldots, l_n\}$ là vector nhãn của phương án đầu tiên, trong đó mỗi $l_i \in \{1, 2, \ldots, K\}$ là nhãn cụm của điểm x_i. Đầu tiên, X được gán nhãn như vậy bắt đầu từ 1 và các điểm tiếp theo sẽ được gán các giá trị tiếp theo. Để gán lại nhãn cho X, đầu tiên 1 vectơ L có độ dài K được tạo ra mà các nhãn lớp xuất hiện duy nhất theo thứ tự. Vectơ L được tính như sau:

$k = 1, L_k = l_1, lab = \{L_1\}$
for $i = 2, \ldots, n$
  if $l_i / \in l_{ab}$ **then**
    $k = k + 1$.
  $L_k = l_i$.
  $l_{ab} = l_{ab} \cup \{l_i\}$.
  end if
end for

Sau đó một ánh xạ M: $L \rightarrow \{1, \ldots, K\}$ được xác định như sau:

$$\forall i = 1, \ldots, K, M[L_i] = i \qquad (2.19)$$

Tiếp theo một vectơ T tạm thời có độ dài n thu được bằng áp dụng các ánh xạ trên X như sau:

$$\forall i = 1, 2, \ldots, n, T_i = M[l_i] \qquad (2.20)$$

Tiếp theo, X được thay thế bởi T. Đây là cách X được dán nhãn. Ví dụ, khởi tạo đặt X = {33111442}. Sau khi dán nhãn lại nó sẽ là {11222334}.

Khi đó các vecto nhãn của từng phương án không vượt trội được sửa lại bằng cách so sánh nó với các vectơ nhãn của phương án đầu tiên như sau: Đặt N là tập các phương án không vượt trội (vectơ nhãn) được đưa ra bởi kỹ thuật phân cụm đa mục tiêu và X là vectơ nhãn cụm của phương án đầu tiên. Giả sử $Y \in N \backslash X$ (tức là, Y là một vectơ nhãn trong N khác X) là vectơ nhãn khác được dán nhãn phù hợp với X. Điều này được thực hiện như sau: đầu tiên, trong mỗi nhãn lớp l trong X, tất cả các điểm P_l được đánh dấu nhãn lớp l trong X được tìm thấy. Sau đó, quan sát các nhãn lớp của các điểm từ Y, chúng ta có được những nhãn lớp b từ Y, đánh dấu số điểm tối đa trong P_l. Sau đó một ánh xạ Map_b được định Map_b: $b \rightarrow l$. Quá trình này được lặp đi lặp lại cho mỗi nhãn lớp $l \in \{1, \ldots, K\}$ trong X. Sau khi đã nhận được tất cả các ánh xạ Map_b cho tất cả các nhãn lớp $b \in \{1, \ldots, K\}$ trong Y, chúng được áp dụng trên Y để dán nhãn Y theo X. Tất cả các phương án không vượt trội $Y \in N \backslash X$ được dán nhãn phù hợp với X như đã nói ở trên. Lưu ý rằng ánh xạ Map nên là ánh xạ 1-1 để đảm bảo rằng sau khi dán nhãn lại Y chứa tất cả các nhãn lớp K. Ràng buộc này có thể bị vi phạm trong khi tìm b. Tình trạng này được khắc phục như sau: Nếu một ánh xạ 1-1 không thể có được thì sẽ cố gắng duyệt

tất cả các khả năng gán nhãn, tức là *K!* khả năng của *Y* và tìm ra được *Y* phù hợp nhất với *X*. Nhãn phù hợp nhất của *Y* được lưu giữ.

Xét ví dụ sau: Đặt *X* là {11222334} và hai vectơ nhãn *Y* = {22444113} và *Z* ={42333221}. Nếu *Y* và *Z* được gán nhãn phù hợp với *X*, thì nhãn *Y* trở thành {11222334}và nhãn *Z* trở thành {13222334}.

Sau khi gán nhãn lại tất cả các vectơ nhãn, kỹ thuật biểu quyết đa số được áp dụng cho tùy từng điểm. Các điểm được chọn bởi ít nhất 50% các phương án thì nhãn được xác định. Những điểm này được sử dụng làm tập huấn luyện cho kỹ thuật *k-nn* để gán nhãn cho các điểm còn lại. Các điểm còn lại được gán nhãn lớp theo phân lớp *k-nn*. Đối với mỗi điểm chưa được xác định *k-nearest neighbords* được tính và các điểm được gán một nhãn lớp thu được bằng biểu quyết đa số *k-nearest neighbords*. Giá trị k được chọn là 5.

Áp dụng biểu quyết đa số theo phân lớp *k-nn* tạo ra một nhãn cụm vectơ *X* mới từ việc kết hợp thông tin phân cụm của tất cả các phương án không vượt trội. Sau đó, mỗi phương án được tính một giá trị tỉ lệ phù hợp với *X*. Phương án phù hợp nhất với *X* là phương án được chọn.

CHƯƠNG 3. THỬ NGHIỆM

3.1. Giới thiệu

Trong quá trình thực hiện đề tài, luận văn đã tiến hành cài đặt phương pháp được trình bày trong [3, 4]. Chương trình được thử nghiệm với một cơ sở dữ liệu trong [4] để kiểm chứng việc cài đặt chương trình. Sau đó, chương trình đã xây dựng được áp dụng cho 2 cơ sở dữ liệu khác, đó là: dữ liệu định danh SPECT heart và Hayes-Roth để đánh giá hiệu quả phân cụm của phương pháp [3, 4] đối với các cơ sở dữ liệu này. Dựa trên việc quan sát kết quả thử nghiệm, luận văn đã đưa ra một số nhận xét, kết luận và một số vấn đề tồn tại cần giải quyết.

3.2. Chương trình

Chương trình được cài đặt trên môi trường Matlab 2013. Các thử nghiệm được thực hiện trên máy tính Intel Core i52.5 GHz, 8 GB RAM, hệ điều hành Windows 7 64 bit. Chương trình được xây dựng dựa trên việc kế thừa có chỉnh sửa từ một mã nguồn Matlab cài đặt thuật toán NSGA-2 [16] để cài đặt phương pháp [3, 4].

3.3. Dữ liệu thử nghiệm

Ba cơ sở dữ liệu danh được dùng để thử nghiệm trong chương trình gồm dữ liệu định danh về đậu tương, SPECT heart và Hayes-Roth được lấy từ UCI Machine Learning Repository (www.ics.uci.edu/ ~mlearn/MLRepository.html).

Link thông tin về cơ sở dữ liệu đỗ tương:
http://archive.ics.uci.edu/ml/datasets/Soybean+%28Small%29
http://archive.ics.uci.edu/ml/machine-learning-databases/soybean/soybean-small.names

Link thông tin về cơ sở dữ liệu SPECT heart:
http://archive.ics.uci.edu/ml/datasets/SPECT+Heart
http://archive.ics.uci.edu/ml/machine-learning-databases/spect/SPECTF.names

Link thông tin về cơ sở dữ liệu SPECT heart:
http://archive.ics.uci.edu/ml/datasets/Hayes-Roth
http://archive.ics.uci.edu/ml/machine-learning-databases/hayes-roth/hayes-roth.names

Down dữ liệu chuẩn về CSDL này theo địa chỉ:
http://archive.ics.uci.edu/ml/machine-learning-databases/soybean/soybean-small.data
http://archive.ics.uci.edu/ml/machine-learning-databases/spect/SPECT.train
http://archive.ics.uci.edu/ml/machine-learning-databases/hayes-roth/hayes-roth.data

3.3.1. Cơ sở dữ liệu Soybean

Bộ dữ liệu này chứa 47 điểm dữ liệu về bệnh của đậu nành [xem Hình 3.1]. Mỗi điểm dữ liệu có 35 thuộc tính định danh và được phân loại vào 1 trong 4 bệnh: Diaporthe Stem, Charcoal, Rhizoctonia Root và Phytophthora, tức là, số cụm trong tập dữ liệu là 4. Mỗi loại bệnh có 10 bản ghi trừ bệnh Phytophthora có 17 bản ghi.

Các thuộc tính của và miền giá trị:

1. date: april,may,june,july,august,september,october.
2. plant-stand: normal,lt-normal.
3. precip: lt-norm,norm,gt-norm.
4. temp: lt-norm,norm,gt-norm.
5. hail: yes,no.
6. crop-hist: diff-lst-year,same-lst-yr,same-lst-two-yrs, same-lst-sev-yrs.
7. area-damaged: scattered,low-areas,upper-areas,whole-field.
8. severity: minor,pot-severe,severe.
9. seed-tmt: none,fungicide,other.
10. germination: 90-100%,80-89%,lt-80%.
11. plant-growth: norm,abnorm.
12. leaves: norm,abnorm.
13. leafspots-halo: absent,yellow-halos,no-yellow-halos.
14. leafspots-marg: w-s-marg,no-w-s-marg,dna.
15. leafspot-size: lt-1/8,gt-1/8,dna.
16. leaf-shread: absent,present.
17. leaf-malf: absent,present.
18. leaf-mild: absent,upper-surf,lower-surf.
19. stem: norm,abnorm.
20. lodging: yes,no.
21. stem-cankers: absent,below-soil,above-soil,above-sec-nde.
22. canker-lesion: dna,brown,dk-brown-blk,tan.
23. fruiting-bodies: absent,present.
24. external decay: absent,firm-and-dry,watery.
25. mycelium: absent,present.
26. int-discolor: none,brown,black.
27. sclerotia: absent,present.
28. fruit-pods: norm,diseased,few-present,dna.
29. fruit spots: absent,colored,brown-w/blk-specks,distort,dna.
30. seed: norm,abnorm.
31. mold-growth: absent,present.
32. seed-discolor: absent,present.
33. seed-size: norm,lt-norm.
34. shriveling: absent,present.
35. roots: norm,rotted,galls-cysts.

3.3.2. Cơ sở dữ liệu SPECT heart

Cơ sở dữ liệu SPECT heart có 80 bản ghi; mỗi bản ghi có 22 thuộc tính. Bộ dữ liệu mô tả thông tin chẩn đoán chụp cắt lớp hình ảnh tim (Single Proton Emission Computed Tomography - SPECT). Mỗi bệnh nhân được phân vào một trong hai loại: bình thường hoặc bất thường.

Các thuộc tính và miền giá trị:

1. OVERALL_DIAGNOSIS: 0,1 (class attribute, binary)
2. F1: 0,1 (the partial diagnosis 1, binary)
3. F2: 0,1 (the partial diagnosis 2, binary)
4. F3: 0,1 (the partial diagnosis 3, binary)
5. F4: 0,1 (the partial diagnosis 4, binary)
6. F5: 0,1 (the partial diagnosis 5, binary)
7. F6: 0,1 (the partial diagnosis 6, binary)
8. F7: 0,1 (the partial diagnosis 7, binary)
9. F8: 0,1 (the partial diagnosis 8, binary)
10. F9: 0,1 (the partial diagnosis 9, binary)
11. F10: 0,1 (the partial diagnosis 10, binary)
12. F11: 0,1 (the partial diagnosis 11, binary)
13. F12: 0,1 (the partial diagnosis 12, binary)
14. F13: 0,1 (the partial diagnosis 13, binary)
15. F14: 0,1 (the partial diagnosis 14, binary)
16. F15: 0,1 (the partial diagnosis 15, binary)
17. F16: 0,1 (the partial diagnosis 16, binary)
18. F17: 0,1 (the partial diagnosis 17, binary)
19. F18: 0,1 (the partial diagnosis 18, binary)
20. F19: 0,1 (the partial diagnosis 19, binary)
21. F20: 0,1 (the partial diagnosis 20, binary)
22. F21: 0,1 (the partial diagnosis 21, binary)
23. F22: 0,1 (the partial diagnosis 22, binary)

3.3.3. Cơ sở dữ liệu Hayes – Roth

Cơ sở dữ liệu Hayes – Roth liên quan đến chủ đề: đối tượng nghiên cứu: con người. Cơ sở dữ liệu này chứa 160 bản ghi, mỗi bản ghi có 5 thuộc tính và được phân vào 1 trong 3 nhóm.

Các thuộc tính bộ dữ liệu Hayes - Roth

Attribute Information:

-- 1. name: distinct for each instance and represented numerically
-- 2. hobby: nominal values ranging between 1 and 3

-- 3. age: nominal values ranging between 1 and 4
-- 4. educational level: nominal values ranging between 1 and 4
-- 5. marital status: nominal values ranging between 1 and 4
-- 6. class: nominal value between 1 and 3

3.4. Phương pháp biểu diễn dữ liệu

Để có cái nhìn trực quan về các bộ dữ liệu, có một phương pháp tốt dùng để đánh giá trực quan về cụm là phương pháp VAT (visual assessment of cluster tendency representation) [9]. Trong phương pháp này, dữ liệu theo một phương án phân cụm được biểu diễn như sau: đầu tiên các điểm được sắp xếp lại theo các nhãn lớp/cụm, sau đó ma trận khoảng cách giữa các điểm dữ liệu được tính toán. Cuối cùng, vẽ biểu đồ đồ họa của ma trận khoảng cách. Trong biểu đồ này, các hình hộp nằm trên đường chéo chính cho thấy các cấu trúc cụm.

3.5. Độ đo hiệu suất

Hiệu suất thuật toán phân cụm được đo bởi độ đo Adjusted Rand Index (ARI) [11]. Giả sử T là phân cụm đúng/thực tế của một tập dữ liệu và C là kết quả phân cụm cho bởi một số thuật toán phân cụm khác. Đặt a, b , c và d biểu thị tương ứng số lượng các cặp điểm thuộc cùng một cụm trong cả T và C, số lượng các cặp điểm thuộc vào cùng một cụm trong T nhưng khác cụm trong C, số lượng các cặp thuộc các cụm khác nhau trong T nhưng thuộc cùng một cụm trong C và số lượng các cặp thuộc các cụm khác nhau trong cả T và C. Khi đó chỉ số *(T,)* được xác định như sau:

$$ARI(T,C) = \frac{2(ad - bc)}{(a+b)(b+d) + (a+c)(c+d)} \quad (4.1)$$

Giá trị của *ARI(T, C)* nằm giữa 0 và 1 và giá trị *ARI* cao cho thấy rằng độ tương tự giữa T và C cao hơn. Khi T và C giống hệt nhau thì *ARI(T, C)* = 1.

3.6. Thủ tục thực nghiệm

Thực hiện lặp lại N lần, mỗi lần lặp lại chạy I lần thuật toán để tính *AvgARIB* như sau:

for i **=** 1 **to** N

for j **=** 1 **to** I

ARI[*j*] = giá trị *ARI* giữa kết quả của lần chạy *(i,j)* so với phân cụm thực tế;

end for

ARIB[*i*] = max {*ARI*[1], . . . , *ARI*[*I*]}.

end for

AvgARIB = avg{*ARIB*[1], . . . , *ARIB*[*M*]}.

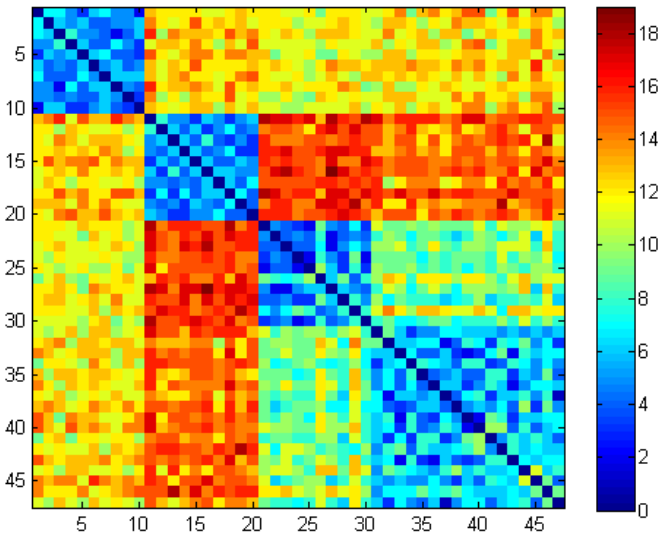
3.7. Các thông số đầu vào

Trong phần thử nghiệm, các thông số đầu vào được sử dụng tương tự [4]:

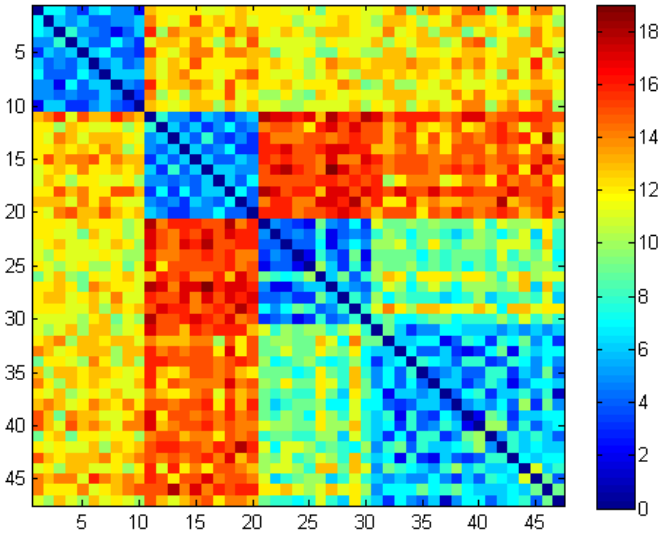
- Số thế hệ (số lần lặp của giải thuật di truyền): 100;
- Kích thước quần thể: 50;
- Xác suất lai ghép: 0.8;
- Xác suất đột biến: 1/*chiều dài NST;*
- Số mũ *m*: 2;

Đây là các giá trị được chọn sau một số thử nghiệm [4]. N và I được chọn là 50 và 100.

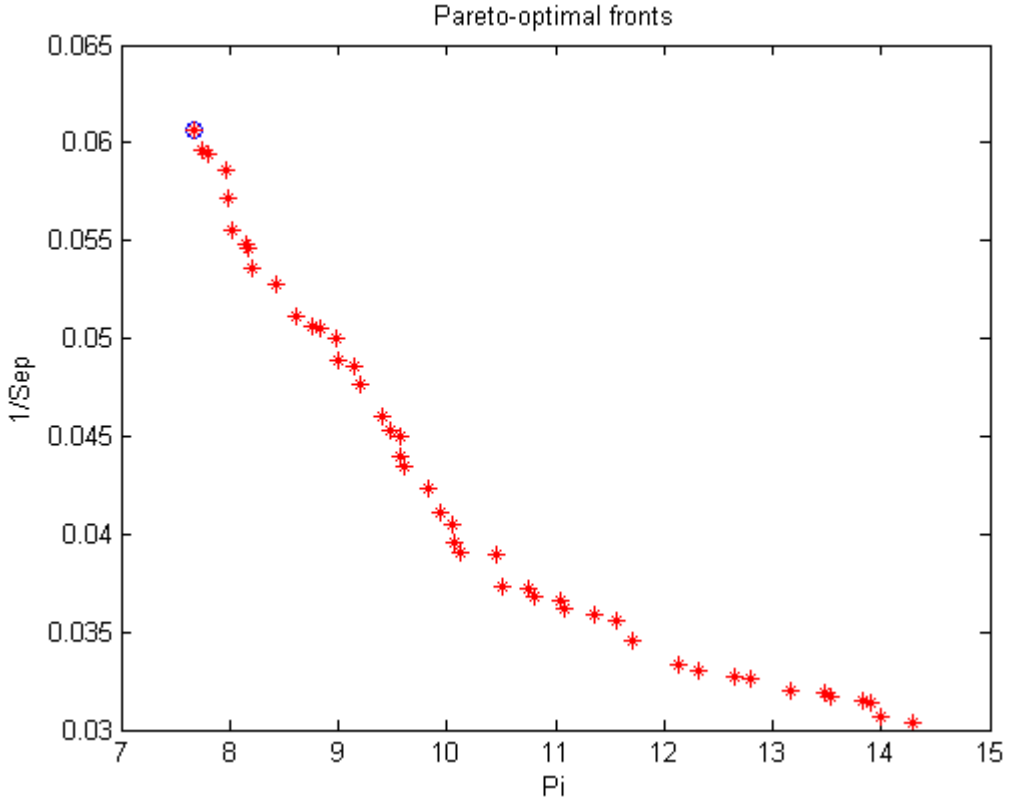
3.8. Kết quả thử nghiệm



Hình 3.1. Phân cụm thực tế của của bộ dữ liệu Soybean sử dụng biểu diễn VAT.

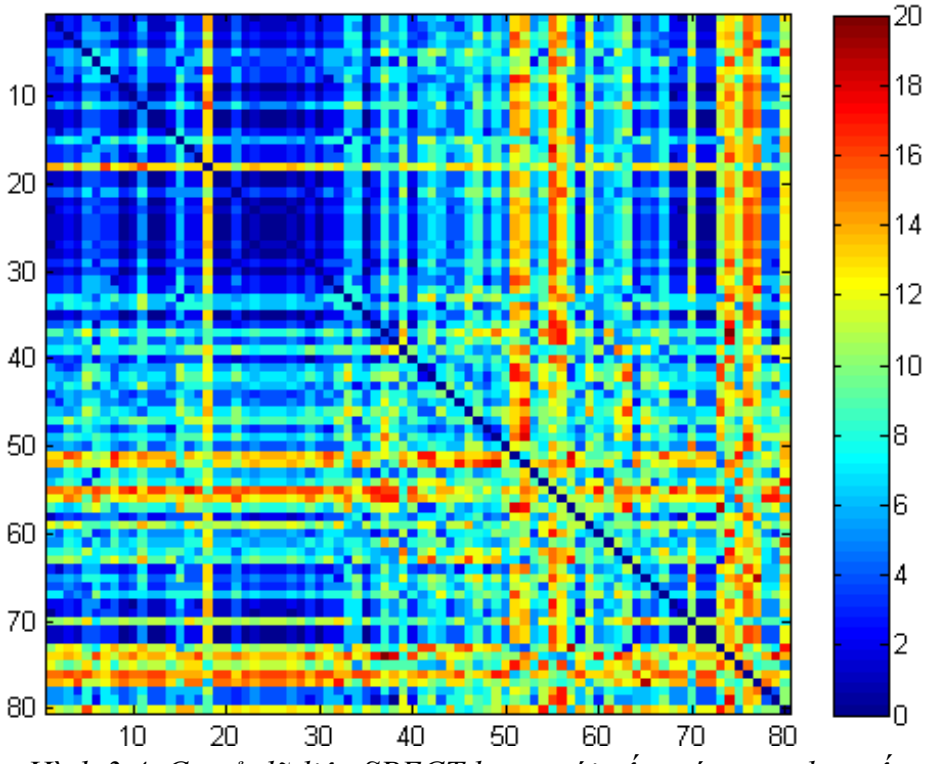


Hình 3.2. Kết quả phân cụm thực nghiệm lại phương pháp [4] trên dữ liệu Soybean.

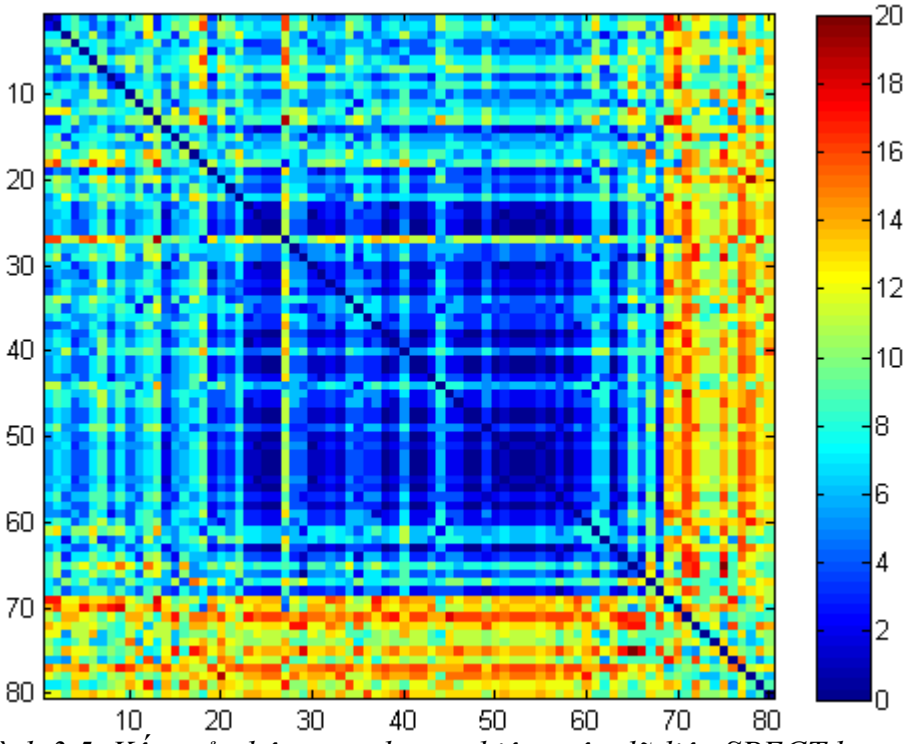


Hình 3.3. Lược đồ mối quan hệ Pi-1/Sep từ tập gần tối ưu Pareto thu được ở thế hệ cuối cùng của thuật toán NSGA-2 trên cơ sở dữ liệu đậu tương. Điểm được đánh dấu bằng hình tròn màu xanh là phương án được lựa chọn cuối cùng.

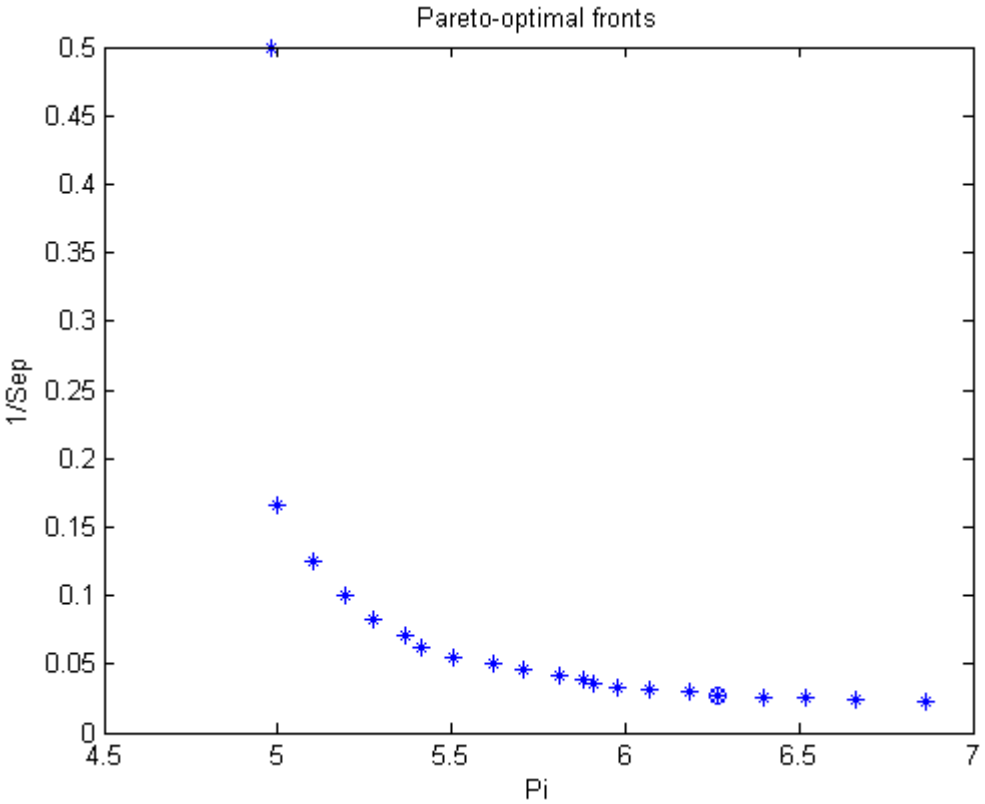
Kết quả thực nghiệm lại trên cơ sở dữ liệu Soybean phù hợp với kết quả trình bày trong [4] (*AvgARIB = 1*). Tương ứng, Hình 3.1 và Hình 3.2 biểu diễn một lần chạy cho kết quả *ARI = 1* cho thấy cấu trúc cụm thu được từ chương trình và cấu trúc cụm thực tế là giống nhau. Dưới đây là kết quả thực nghiệm trên các cơ sở dữ liệu SPECT heart và trên cơ sở dữ liệu Hayes-Roth cùng với một số nhận xét dựa trên quan sát các kết quả thực nghiệm.



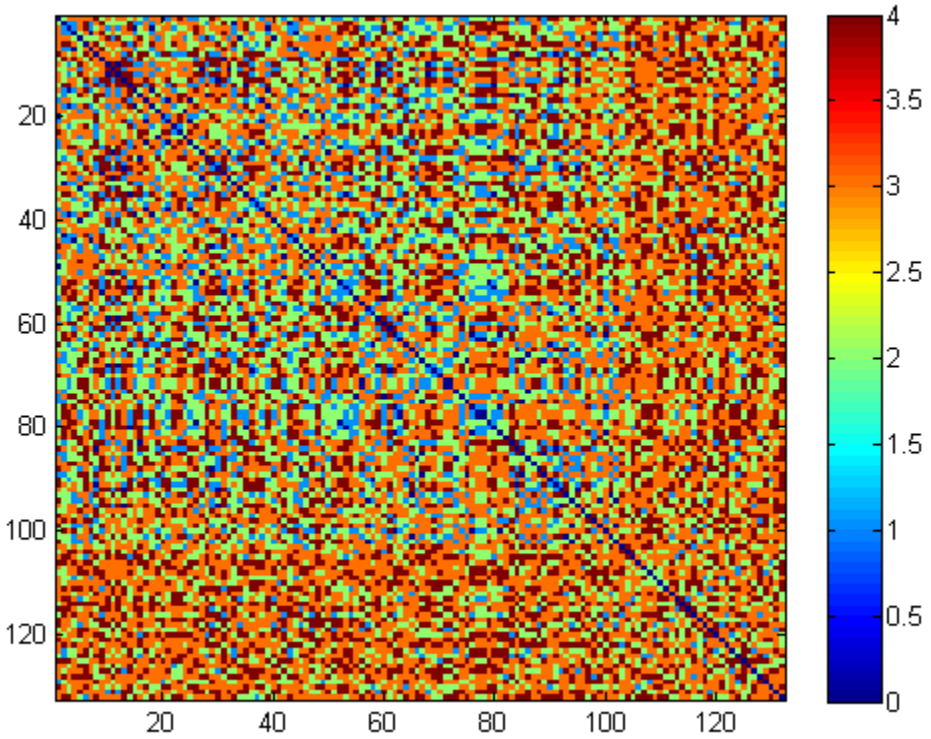
Hình 3.4. Cơ sở dữ liệu SPECT heart với cấu trúc cụm thực tế.



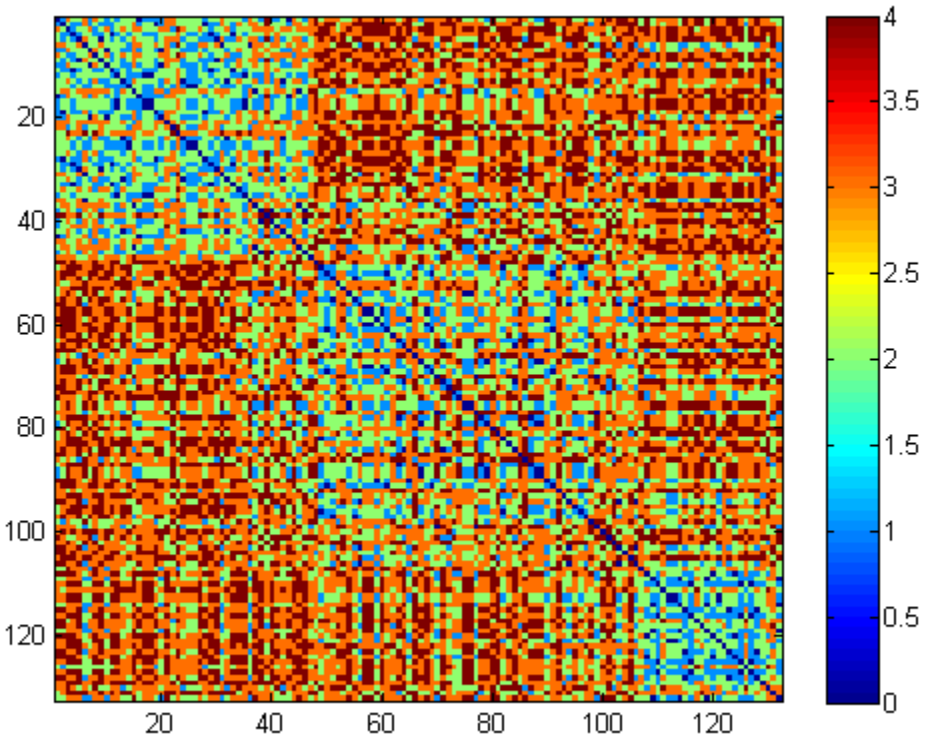
Hình 3.5. Kết quả phân cụm thực nghiệm trên dữ liệu SPECT heart.



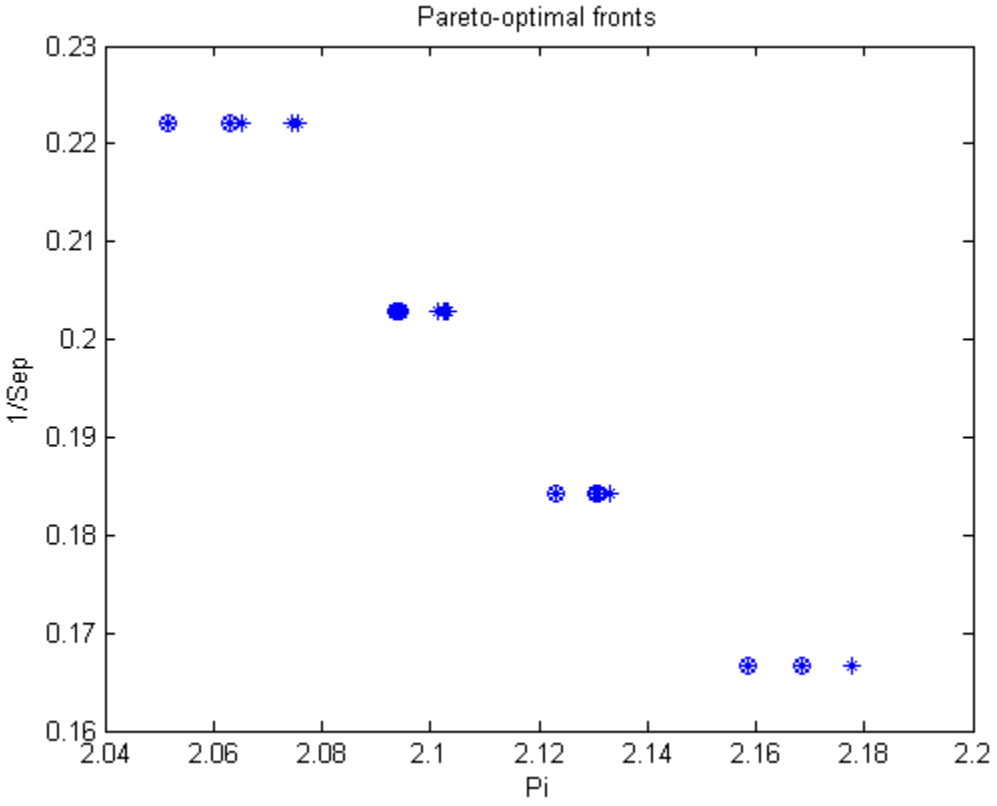
Hình 3.6. Lược đồ mối quan hệ Pi-1/Sep từ tập gần tối ưu Pareto thu được ở thế hệ cuối cùng của thuật toán NSGA-2 trên cơ sở dữ SPECT heart.



Hình 3.7. Cơ sở dữ liệu Hayes-Roth với cấu trúc cụm thực tế.



Hình 3.8. Kết quả phân cụm thực nghiệm trên dữ liệu Hayes-Roth.



Hình 3.9. Lược đồ mối quan hệ Pi-1/Sep từ tập gần tối ưu Pareto thu được ở thế hệ cuối cùng của thuật toán NSGA-2 trên cơ sở dữ Hayes-Roth.

Nhận xét:

Qua quan sát các kết quả mà luận văn này đã thực nghiệm nhiều lần đưa ra một số nhận xét như sau:

1. **Với mỗi bộ dữ liệu cụ thể ứng với mỗi bài toán thực tế, khi áp dụng phương pháp phân cụm thì cần thiết kế/lựa chọn hàm khoảng cách giữa các điểm dữ liệu phù hợp.** Như ta thấy trong Hình 3.1, khoảng cách Hamming mà ta đang sử dụng phù hợp với cơ sở dữ liệu đậu tương do đó ta có thể quan sát được rõ các cụm thực tế khi biểu diễn bằng phương pháp VAT. Trong trường hợp này, phương pháp sử dụng trong luận văn cho kết quả tốt (AvrARIB = 1). Tuy nhiên, đối với hai cơ sở dữ liệu SPECT heart (Hình 3.4) và Hayes-Roth (Hình 3.7), chúng ta không thể quan sát được cấu trúc các cụm thực tế trên lược đồ VAT với khoảng cách Hamming. Điều đó có nghĩa là khoảng cách Hamming không phù hợp với hai cơ sở dữ liệu này. Quan sát lược đồ VAT của kết quả phân cụm (Hình 3.5, Hình 3.8), chúng ta thấy cấu trúc các cụm đã rõ hơn. Điều đó có nghĩa là các cụm kết quả của phương pháp phân cụm có độ thuần nhất trong các cụm và độ phân tách giữa các cụm theo khoảng cách Hamming là tốt hơn các cụm thực tế. Do đó giá trị AvrARIB thu được rất thấp do có sự sai khác giữa kết quả phân cụm và các cụm thực tế (AvrARIB = 0.0244 đối với cơ sở dữ liệu SPECT heart; AvrARIB = -0.0050 đối với cơ sở dữ liệu Hayes-Roth).
2. ***Cần cải thiện phương pháp chọn phương án tốt từ thế hệ cuối cùng***. Mặc dù phương pháp chọn một phương án tốt từ thế hệ cuối cùng được báo cáo là một trong những đóng góp quan trọng của [4], tuy nhiên trong nhiều trường hợp, phương án chọn được không phải là phương án tốt nhất. Quan sát các thử nghiệm trên cơ sở dữ liệu đậu tương (là cơ sở dữ liệu mà hàm khoảng cách Hamming phù hợp để phân cụm) ta thấy có nhiều trường hợp trong 50 cá thể ở quần thể cuối cùng, có nhiều cá thể có *ARI* bằng 1 nhưng phương pháp chọn đưa ra phương án kém hơn (có $ARI < 1$).

KẾT LUẬN

Qua thời gian nghiên cứu, dưới sự hướng dẫn trực tiếp của thày PGS.TS Hoàng Xuân Huấn, em đã hoàn thành luận văn "Phân cụm đa mục tiêu mờ cho dữ liệu định danh". Luận văn đã đạt được hai kết quả chính là:

1. Nghiên cứu tài liệu và hệ thống lại các kiến thức có liên quan sau:
   – Phân cụm dữ liệu.
   – Các phương pháp chính sử dụng để phân cụm dữ liệu.
   – Phân cụm rõ, phân cụm mờ và giải thuật tối ưu hóa cụm.
   – Nghiên cứu giải thuật tối ưu đa mục tiêu thực hiện phân cụm mờ cho dữ liệu định danh.
2. Cài đặt thuật toán tối ưu đa mục tiêu NSGA – II phân cụm mờ cho dữ liệu định danh. Luận văn đã chạy thử nghiệm với 3 bộ dữ liệu thực tế từ đó đưa ra những bình luận, nhận xét và rút ra một số vấn đề cần tập trung nghiên cứu, giải quyết.

Trong thời gian tới, em định hướng tập trung nghiên cứu, thực hiện những vấn đề sau đây:

(i) Tìm hiểu các bài toán trong thực tế có liên quan đến cơ sở dữ liệu danh để áp dụng phương pháp mà luận văn đã nghiên cứu, tìm hiểu. Khi đó, một trong những vấn đề quan trọng cần thực hiện là phân tích đặc điểm của bài toán, đặc điểm về dữ liệu cũng như các cụm trong thực tế để thiết kế/lựa chọn hàm khoảng cách phù hợp.

(ii) Nghiên cứu để cải thiện hiệu quả của bước chọn phương án tốt từ thế hế cuối cùng, kết quả của thuật toán NSGA-II.

Thời gian qua mặc dù bản thân em cũng đã nỗ lực nhưng luận văn của em không tránh khỏi thiếu sót do năng lực của bản thân em còn hạn chế, em rất mong nhận được sự đóng góp của các Thày, Cô, bạn bè và những ai có cùng hướng quan tâm nghiên cứu.

Em xin được gửi lời cảm ơn chân thành nhất đến Thày PGS. TS Hoàng Xuân Huấn đã tận tình chỉ bảo, nhận xét, góp ý cho nghiên cứu của em. Em cũng xin được gửi lời cảm ơn sâu sắc đến tất cả các Thày, Cô đã tận tình giảng dạy cho em trong suốt khóa học tại Trường Đại học Công nghệ - Đại học Quốc Gia Hà Nội.

TÀI LIỆU THAM KHẢO

Tiếng Việt

[1] Hoàng Xuân Huấn (2012), *Giáo trình Nhận dạng mẫu*, Trường Đại học Công nghệ – Đại Học Quốc Gia Hà Nội.

[2] Nguyễn Hà Nam (2012), Nguyễn Trí Thành, Hà Quang Thụy, *Giáo trình Khai phá dữ liệu*, NXB Đại học Quốc gia Hà Nội.

Tiếng Anh

[3] Anirban Mukhopadhyay, Ujjwal Maulik and Sanghamitra Bandyopadhyay(*2013*), Hybrid Evolutionary Multiobjective Fuzzy C-Medoids Clustering of Categorical Data, *IEEE Workshop on Hybrid Intelligent Models and Applications (HIMA)*.

[4] Anirban Mukhopadhyay, Ujjwal Maulik and Sanghamitra Bandyopadhya (2009), Multiobjective Genetic Algorithm-Based Fuzzy Clustering of Categorical Attributes, IEEE transactions on evolutionary computation, vol. 13, no. 5, October.

[5] A. K. Jain and R. C. Dubes (1988), *Algorithms for Clustering Data*. Englewood Cliffs, NJ: Prentice-Hall.

[6] A. Konak, D. W. Coit, A. E. Smith (2006), "Multi objective optimization using genetic algorithms: A tutorial", J. Reability Engineering and System Safety, No. 91, pp. 992-1007.

[7] E. Zitzler and L. Thiele (1998), "An evolutionary algorithm for multiobjective optimization: The strength Pareto approach", Swiss Fed. Inst. Technol., Zurich, Switzerland, Tech. Rep. 43.

[8] J. C. Bezdek (1981), *Pattern Recognition with Fuzzy Objective Function Algorithms*. New York: Plenum.

[9] J. C. Bezdek and R. J. Hathaway, "VAT: A tool for visual assessment of (cluster) tendency," in Proc. Int. Joint Conf. Neural Netw., vol. 3. Honolulu, HI, 2002, pp. 2225–2230

[10] Jianhua Yang (2002), *Algorithmic engineering of clustering and cluster validity with applications to web usage mining*, School of Electrical Engineering and Computer Science, Australia.

[11] K. Y. Yip, D. W. Cheung, and M. K. Ng (2003), "A highly usable projected clustering algorithm for gene expression profiles," in *Proceedingsof 3rd ACM SIGKDD Workshop on Data Mining in Bioinformatics*, pp. 41–48.

[12] L. Kaufman and P. J. Rousseeuw (1990), *Finding Groups in Data: An GIntroduction to Cluster Analysis*. NY, US: John Wiley & Sons.

[13] Osmar R.Zaiane (2001), *Principles of knowledge discovery in databases*, University of Alberta, Fall.

[14] Z. Huang and M. K. Ng (1999), “A fuzzy k-modes algorithm for clustering categorical data,” *IEEE Trans. Fuzzy Syst.*, vol. 7, no. 4, pp. 446–452, Aug.

[15] Zadeh L.A.(1965), Fuzzy Sets, Information and Control, pp.338–353.

[16] https://www.mathworks.com/matlabcentral/fileexchange/10429-nsga-ii--a-multi-objective-optimization-algorithm