UDC 519.1:004.94:004.62
DOI https://doi.org/10.32782/tnv-tech.2024.1.5

USING THE MINIMUM SPANNING TREE PROBLEM FOR DATA MINING

Dymova H. O. *– Candidate of Technical Sciences,*
Associate Professor at the Department of Management,
Marketing and Information Technology
of the Kherson State Agrarian and Economic University
ORCID ID: 0000-0002-5294-1756

The article is devoted to an attempt to determine clusters using the optimization problem of a minimum spanning tree. The problem of the clustering problem is to group similar objects into many clusters in such a way that objects within one cluster are similar to each other, and objects between different clusters are different. One approach to solving this problem is to use the minimum spanning tree problem.

The idea of using the minimum spanning tree problem for clustering is to build a tree connecting all the objects, where the weights of the edges correspond to the distances between the objects. Then, using some pruning technique, a certain number of edges can be removed to divide the tree into clusters. To do this, it is necessary to remove the longest edges in order to divide the tree into individual components corresponding to clusters. This approach can be especially effective in cases where it is necessary to minimize overlaps: a minimum spanning tree combines vertices with the smallest distances, that is, the vertices that are most similar to each other will be located next to each other. This helps reduce the number of matches in clusters and improve the quality of clustering. A minimum spanning tree can be easily interpreted because it shows the closest connections between vertices, which allows us to understand which groups of data tend to cluster together. It can also be useful in cases where we are dealing with large amounts of data, since minimum spanning tree algorithms have low computational complexity. Minimum Spanning Tree allows you to ignore noise in the data because it builds a connection between nearby points and noisy data will usually be far away from other points, and also allows you to naturally determine the number of clusters. The number of clusters corresponds to the number of edges in the minimum spanning tree after removing the longest edges. However, be aware that this approach may not always be the best for all types of data and requires careful analysis and customization of parameters for each specific case.

Key words: *graph, minimum spanning tree, "greedy" algorithm, shortest path, clustering, data mining.*

Димова Г. О. Використання задачі про мінімальне остовне дерево для інтелектуального аналізу даних

Стаття присвячена спробі визначення кластерів за допомогою оптимізаційної задачі про мінімальне остовне дерево. Проблема задачі кластеризації полягає в групуванні подібних об’єктів у множину кластерів таким чином, щоб об’єкти всередині одного кластера були схожі між собою, а об’єкти між різними кластерами були відмінними. Одним із підходів до розв’язання цієї проблеми є використання задачі про мінімальне остовне дерево.

Ідея використання задачі про мінімальне остовне дерево для кластеризації полягає в тому, щоб побудувати дерево, яке з’єднує всі об’єкти, де ваги ребер відповідають відстаням між об’єктами. Потім, за допомогою деякого методу обрізання, можна видалити певну кількість ребер, щоб розділити дерево на кластери. Для цього необхідно видалити найбільш довгі ребра, щоб розділити дерево на окремі компоненти, які відповідають кластерам. Цей підхід може бути особливо ефективним у випадках, коли необхідно мінімізувати збіги: мінімальне остовне дерево об’єднує вершини з найменшими відстанями, тобто вершини, які найбільше схожі одна на одну, будуть розташовані поруч. Це допомагає зменшити кількість збігів у кластерах та поліпшити якість кластеризації. Мінімальне остовне дерево може бути легко інтерпретоване, оскільки воно показує найближчі зв’язки між вершинами, це дозволяє зрозуміти, які групи даних схильні до групування разом. Він також може бути корисним у випадках, коли маємо справу *з великими обсягами даних, оскільки алгоритми побудови мінімального остовного дерева мають низьку обчислювальну складність. Мінімальне остовне дерево дозволяє ігнорувати шум у даних, оскільки він будує з'єднання між найближчими точками, і шумові дані зазвичай будуть далеко від інших точок, а також дозволяє природним чином визначити кількість кластерів. Кількість кластерів відповідає кількості ребер у мінімальному остовному дереві після видалення найбільш довгих ребер. Однак, варто враховувати, що цей підхід може не завжди бути найкращим для всіх типів даних і вимагає уважного аналізу та налаштування параметрів для кожного конкретного випадку.*

Ключові слова: *граф, мінімальне остовне дерево, «жадібний» алгоритм, найкоротший шлях, кластеризація, інтелектуальний аналіз даних.*

Introduction. The minimum spanning tree problem is an important problem in graph theory, combinatorics, and optimization algorithms. A minimum spanning tree of a graph is a subgraph, which is a tree that includes all the vertices of the graph, but only some edges, so that the sum of the weights of these edges is minimal [1; 2]. This means that a minimum spanning tree is the most efficient way to connect all the vertices of a graph at a lower cost.

The minimum spanning tree problem has a wide range of applications in various industries such as telecommunications, transportation, electric power and others. Here are examples of several ways in which this task can be used:

– in transport and logistics networks. Determining the shortest route to transport goods or services can help reduce transportation costs and save time. The paths included in the minimum spanning tree can indicate the most efficient routes and optimal placement of warehouses or service facilities;

– in communication networks. In the telecommunications and information industries, the minimum spanning tree problem can be used to build a communication network. This will help reduce infrastructure costs and ensure efficient data flow;

– in optimization of production processes. Manufacturing plants can use the minimum spanning tree problem to optimize material flows and production logistics. This allows you to reduce warehousing costs and ensure optimal resource allocation;

– in the financial network. In the banking sector and financial institutions, the concept of a minimum tree can be used to analyze financial networks and optimize the path of funds or financial transactions.

Therefore, the minimum spanning tree problem can be an important tool for optimizing a variety of processes in any field of activity. It allows you to find optimal solutions taking into account constraints and efficiency requirements.

Formulation of the problem. The task of clustering is to group similar objects into many clusters in such a way that objects within one cluster are similar to each other, and objects between different clusters are different [3]. One approach to solving this problem is to use the minimum spanning tree problem.

The idea of using the minimum spanning tree problem for clustering is to construct a tree connecting all the objects, where the weights of the edges correspond to the distances between the objects.

The purpose of the article is to conduct data analysis to identify clusters using the minimum spanning tree problem.

Research analysis. In data mining, cluster analysis does not use a single algorithm, it is a general task using different approaches. Popular algorithms for identifying clusters include groups of resulting elements that are based on the distance between them, the density of areas in the data space, intervals, or specific statistical distributions [3].

Cluster analysis comes from anthropology, where it was started by Driver and Kroeber in 1932. It was introduced into psychology by Zubin in 1938 and Robert Tryon in 1939. Became famous for using Quettel to classify trait theory in personality psychology starting in 1943.

Presentation of the main material. A tree is a connected set of undirected edges (arcs) that does not contain cycles. Thus, if a set of m nodes connected by undirected edges is given, then to construct a tree it is necessary to select a subset consisting of m–1 arcs. In other words, each node is connected to another node in one and only way [1; 2; 4].

Consider a network containing n nodes, the collection of which forms the set S. A spanning tree is a connected set consisting of (n–1) arcs (edges) and n nodes. A tree can be formed from any proper subset of the set S, which, however, may not be the spanning tree of the original network. As before, we will assume that each arc connecting nodes i and j from the set S is assigned a number $c_{ij,}$ called the distance, or weight, of the arc. Now let us introduce the concept of a minimum spanning tree. A minimal skeleton is a network skeleton for which the sum of the weights $c_{ij,}$ of all its arcs is minimal [3].

The minimum spanning problem is one of the problems that can be solved using a "greedy" algorithm, which is very economical. Using the "absorption" scheme, we present the following algorithm.

The minimum spanning problem is to select such arcs of a given network such that their total cost is minimal and for any pair of nodes there is a path (or route) connecting them. This can be achieved by choosing arcs in such a way that the tree formed by them will connect all the nodes of a given network, that is, it is necessary to build a skeleton of minimal cost.

The minimum skeleton problem is solved quite simply. The algorithm begins its work by selecting an arbitrary network node and the shortest arc from the set of arcs connecting this node with other nodes. Let’s connect two nodes with the selected arc. Let’s select the third node closest to these nodes. We add this node and the corresponding arc to the network. We continue this process until all nodes are connected to each other [3; 4]. An algorithm based on the “absorption” of minimal arcs can be described as follows.

Algorithm for constructing a minimum spanning tree [5–8]:

Step 1. Using the nodes of the original network, define the following two sets: S – set of connected nodes; $\overline{S}$ is the set of unconnected nodes. Initially, all nodes will belong to the set $\overline{S}$.

Step 2. Select an arbitrary node from $\overline{S}$ and connect it to the nearest neighboring node. After completing this step, the set S will contain two nodes.

Step 3. Among all the arcs connecting nodes from the set S with nodes from the set $\overline{S}$, choose the minimal arc. The final node of this arc, belonging to $\overline{S}$, is denoted by δ. Remove node δ from set $\overline{S}$ and place it in set S.

Step 4. Perform step 3 until all nodes belong to the set S.

Let’s consider an example of finding a solution to a “greedy” algorithm for the network shown in Fig. 1.

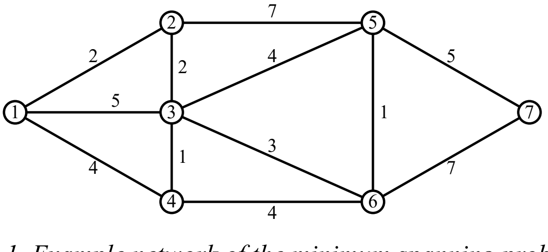
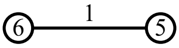
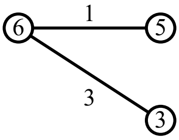
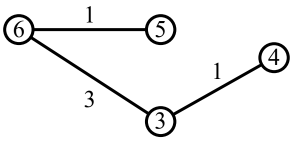
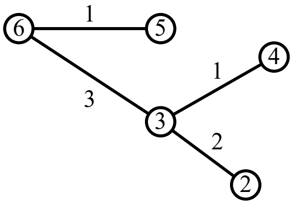
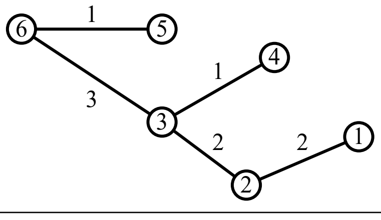
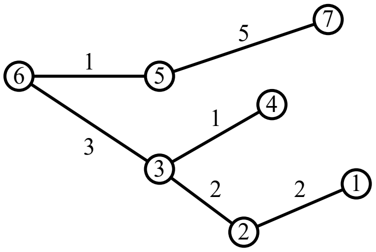


Fig. 1. Example network of the minimum spanning problem

We summarize the solution to the problem of the minimum skeleton of a network (Fig. 1) in Table 1.

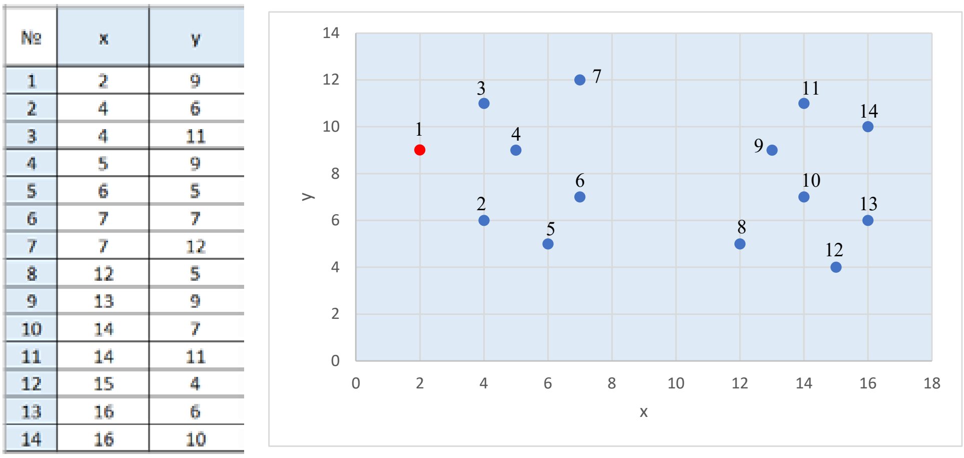
Table 1

Related problems about the minimum spanning tree

Step		**Set**	**Building a Minimum Spanning Tree**	**Cost, c_{ij}**
1		$S = \varnothing$ $\overline{S} = (1, 2, 3, 4, 5, 6, 7)$	–	–
2		$S = (6)$ $\overline{S} = (1, 2, 3, 4, 5, 7)$	–	–
3	*a*	$S = (6, 5)$ $\overline{S} = (1, 2, 3, 4, 7)$		1
	b	$S = (6, 5, 3)$ $\overline{S} = (1, 2, 4, 7)$		4
	c	$S = (6, 5, 3, 4)$ $\overline{S} = (1, 2, 7)$		5
	d	$S = (6, 5, 3, 4, 2)$ $\overline{S} = (1, 7)$		7
	e	$S = (6, 5, 3, 4, 2, 1)$ $\overline{S} = (7)$		9
	f	$S = (6, 5, 3, 4, 2, 1, 7)$ $\overline{S} = \varnothing$		14

The algorithm terminates because $\overline{S} = \varnothing$. The minimum core value is 14.

Let’s try to use the minimum spanning tree algorithm for the data clustering problem. A data set suitable for clustering is a set of points belonging to a certain space. A space is a universal set from which points in a data set are taken. An example of such a space would be Euclidean space, where the length of the vector is determined by the number of dimensions of the space, and the components of the vector are the coordinates of the corresponding points. To simplify the experiment, let’s take a two-dimensional space with some data set and try to divide this data into clusters using the minimum spanning tree problem. The data and their location on the plane are shown in Fig. 2 *a)–b)*.



№	x	y
1	2	9
2	4	6
3	4	11
4	5	9
5	6	5
6	7	7
7	7	12
8	12	5
9	13	9
10	14	7
11	14	11
12	15	4
13	16	6
14	16	10

a) *b)*

Fig. 2. Data set for clustering:
a) data; b) data layout on the plane

The algorithm can start from any point located in space. Let’s start from point 1 with coordinates [2; 9], highlighted in red. Clusters should be combined based on the proximity of the data sets. The join stops when further joins produce undesirable results.

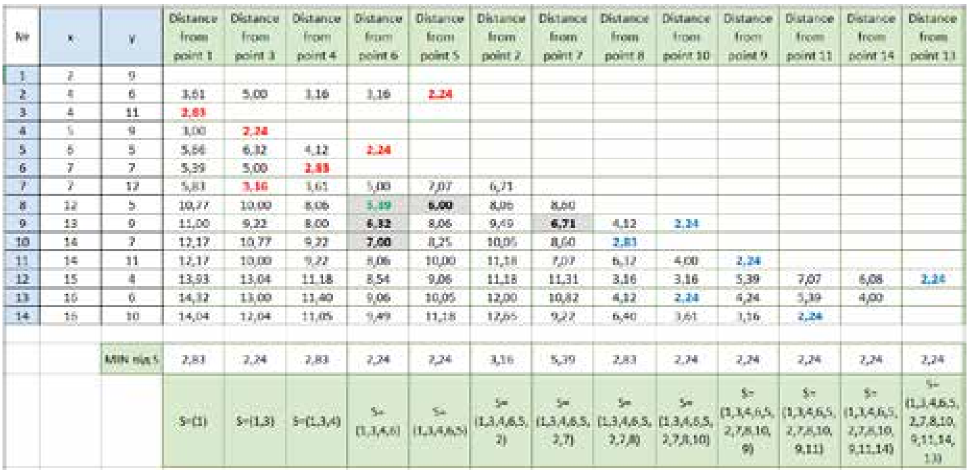
Let’s determine the distances between point 1 and other points on the plane and find the minimum value, which is 2,83. This is point 3.

After this, point 3 is included in the set S and then the minimum value of the distance from two points included in the set S is searched. This continues until all points are included in the set S, and the set $\overline{S}$ is equal to the empty set $\varnothing$ (Fig. 3).

Using the calculated data, we will construct a minimum spanning tree for a given data set (Fig. 4).

To divide the constructed spanning tree into clusters, we remove a certain number of edges that are the longest. If you analyze the calculation table (Fig. 3), you can see that the distance between the points of the tree parts highlighted in red and blue is minimal. Moreover, the distance between any points of the red and blue parts is much greater than the distance inside these parts (cells highlighted in gray in Fig. 3). Therefore, we can conclude that, in our case, we need to remove the edge between points 6 and 8. That is, there are two clusters of points consisting of points with the smallest distances between them. These are points 1–7, highlighted in red, and points 8–14, highlighted in blue.

Conclusions. The minimum spanning tree problem can be used for data mining in the field of data clustering, where there is a need to group similar objects together. It is



№	x	y	Distance from point 1	Distance from point 3	Distance from point 4	Distance from point 6	Distance from point 5	Distance from point 2	Distance from point 7	Distance from point 8	Distance from point 10	Distance from point 9	Distance from point 11	Distance from point 14	Distance from point 13
1	2	9													
2	4	6	3,61	5,00	3,16	3,16	2,24								
3	4	11	2,83												
4	5	9	3,00	2,24											
5	6	5	5,66	6,32	4,12	2,24									
6	7	7	5,39	5,00	2,83										
7	7	12	5,83	3,16	3,61	5,00	7,07	6,71							
8	12	5	10,77	10,00	8,06	5,39	6,00	8,06	8,60						
9	13	9	11,00	9,22	8,00	6,32	8,06	9,49	6,71	4,12	2,24				
10	14	7	12,17	10,77	9,22	7,00	8,25	10,05	8,60	2,83					
11	14	11	12,17	10,00	9,22	8,06	10,00	11,18	7,07	6,12	4,00	2,24			
12	15	4	13,93	13,04	11,18	8,54	9,06	11,18	11,31	3,16	3,16	5,39	7,07	6,08	2,24
13	16	6	14,32	13,00	11,40	9,06	10,05	12,00	10,82	4,12	2,24	4,24	5,39	4,00	
14	16	10	14,04	12,04	11,05	9,49	11,18	12,65	9,22	6,40	3,61	3,16	2,24		
		MIN від S	2,83	2,24	2,83	2,24	2,24	3,16	5,39	2,83	2,24	2,24	2,24	2,24	2,24
			S=(1)	S=(1,3)	S=(1,3,4)	S=(1,3,4,6)	S=(1,3,4,6,5)	S=(1,3,4,6,5, 2)	S=(1,3,4,6,5, 2,7)	S=(1,3,4,6,5, 2,7,8)	S=(1,3,4,6,5, 2,7,8,10)	S=(1,3,4,6,5, 2,7,8,10, 9)	S=(1,3,4,6,5, 2,7,8,10, 9,11)	S=(1,3,4,6,5, 2,7,8,10, 9,11,14)	S=(1,3,4,6,5, 2,7,8,10, 9,11,14, 13)

Fig. 3. Calculation of the algorithm for constructing a minimum spanning tree

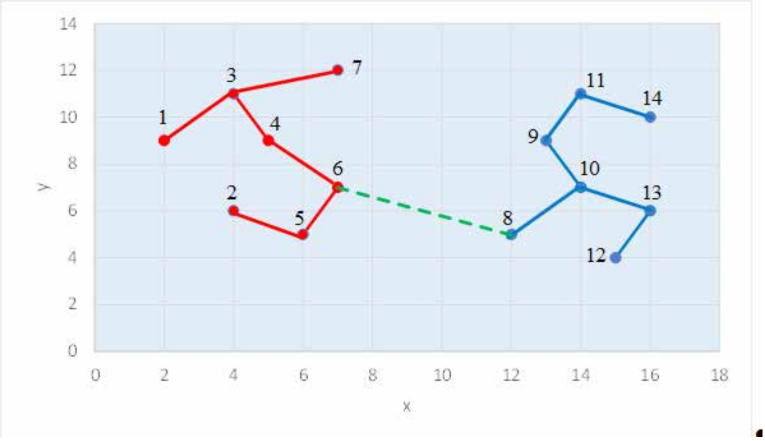


Fig. 4. Minimum spanning tree

also possible to use a minimum spanning tree to identify the most significant relationships between objects. This will help in discovering groups of objects that have similar characteristics or relationships.

BIBLIOGRAPHY:

1. Поповський, В., Сабурова, С., Олійник, В., Лосєв, Ю., Агеєв, Д. Математичні основи теорій телекомунікаційних систем. Харків : ТОВ «Компанія СМІТ», 2006. 564 с.

2. Капітонова Ю.В. Основи дискретної математики. Київ : Наукова думка, 2002. 580 с.

3. Димова Г.О., Ларченко О.В. Моделі і методи інтелектуального аналізу даних: навчальний посібник. Херсон : Книжкове видавництво ФОП Вишемирський В. С., 2021. 142 с.

4. Phillips D.T., Garcia-Diaz A. Fundamentals of Network Analysis. Prentice-Hall, Inc., Englewood Cliffs, N.J., 1981. 496 p.

5. Dymova H. Development Of A Software Application Algorithm For Solving Computer Network Optimization Problems. *Débats scientifiques et orientations prospectives du développement scientifique* : c avec des matériaux de la VI conférence scientifique et pratique internationale, Paris, 1er Mars 2024. Paris-Vinnytsia : La Fedeltà & UKRLOGOS Group LLC, 2024. Pp. 226–229. DOI 10.36074/logos-01.03.2024.051.

6. Димова Г., Ларченко О. Розробка комп’ютерної програми розв’язання задач мережевої оптимізації. Науковий журнал *«Комп’ютерно-інтегровані технології: освіта, наука, виробництво»*, (41), 2020. С. 142–150. DOI 10.36910/6775-2524-0560-2020-41-23.

7. Bhargava A. Grokking Algorithms. An Illustrated Guide For Programmers And Other Curious People. By Manning Publications Co. All Rights Reserved, 2016. 288 p.

8. Dymova H. Application of Characterization Analysis Methods to Investigation of Logical Networks Structures. *Theoretical and Empirical Scientific Research: Concept and Trends with Proceedings of the V International Scientific and Practical Conference*. Oxford, United Kingdom : European Scientific Platform, 2023. Pp. 124–128. DOI: 10.36074/logos-23.06.2023.34.

REFERENCES:

1. Popovs′kyy V., Saburova S., Oliynyk V., Losyev YU. & Aheyev D. (2006). Matematychni osnovy teoriy telekomunikatsiynykh system [Mathematical Foundations of Theories of Telecommunication Systems] “SMITH Company” LLC. [in Ukrainian].

2. Kapitonova Yu.V. (2002) Osnovy dyskretnoyi matematyky [Fundamentals of Discrete Mathematics] Scientific thought. [in Ukrainian].

3. Dymova H.O. & Larchenko O.V. (2021) Modeli i metody intelektual′noho analizu danykh : navchal′nyy posibnyk [Models and Methods of Data Mining: Tutorial] Publishing house FOP Vyshemyrskyy V.S. [in Ukrainian].

4. Phillips D.T. & Garcia-Diaz A. (1981) Fundamentals of Network Analysis. Prentice-Hall, Inc., Englewood Cliffs, N.J.

5. Dymova H. (2024) Development Of A Software Application Algorithm For Solving Computer Network Optimization Problems. *Débats scientifiques et orientations prospectives du développement scientifique: c avec des matériaux de la VI conférence scientifique et pratique internationale, Paris, 1er Mars 2024.* Paris-Vinnytsia : La Fedeltà & UKRLOGOS Group LLC. DOI 10.36074/logos-01.03.2024.051.

6. Dymova H. & Larchenko O. (2020) Rozrobka komp″yuternoyi prohramy rozv″yazannya zadach merezhevoyi optymizatsiyi [Development of a computer program for solving network optimization problems] *Scientific journal “Computer-integrated technologies: education, science, production”*, Vol. 41. DOI 10.36910/6775-2524-0560-2020-41-23. [in Ukrainian].

7. Bhargava A. (2016) Grokking Algorithms. An Illustrated Guide For Programmers And Other Curious People. By Manning Publications Co. All Rights Reserved.

8. Dymova H. (2023) Application of Characterization Analysis Methods to Investigation of Logical Networks Structures. *Theoretical and Empirical Scientific Research: Concept and Trends with Proceedings of the V International Scientific and Practical Conference*. Oxford, United Kingdom : European Scientific Platform. 124–128. DOI: 10.36074/logos-23.06.2023.34.