JKSCI

한국컴퓨터정보학회논문지
Journal of The Korea Society of Computer and Information
Vol. 25 No. 12, pp. 135-144, December 2020
https://doi.org/10.9708/jksci.2020.25.12.135

Shortest Path Search Scheme with a Graph of Multiple Attributes

Jongwan Kim*, KwangJin Choi*, Dukshin Oh**

*Professor, Smith College of Liberal Arts, Sahmyook University, Seoul, Korea
*Professor, Smith College of Liberal Arts, Sahmyook University, Seoul, Korea
**Professor, Dept. of Management Information Systems, Sahmyook University, Seoul, Korea

[Abstract]

In graph theory, the least-cost path is discovered by searching the shortest path between a start node and destination node. The least cost is calculated as a one-dimensional value that represents the difference in distance or price between two nodes, and the nodes and edges that comprise the lowest sum of costs between the linked nodes is the shortest path. However, it is difficult to determine the shortest path if each node has multiple attributes because the number of cost types that can appear is equal to the number of attributes. In this paper, a shortest path search scheme is proposed that considers multiple attributes using the Euclidean distance to satisfy various user requirements. In simulation, we discovered that the shortest path calculated using one-dimensional values differs from that calculated using the Euclidean distance for two-dimensional attributes. The user's preferences are reflected in multi attributes and it was different from one-dimensional attribute. Consequently, user requirements could be satisfied simultaneously by considering multiple attributes.

▸ **Key words**: Euclidean Distance, Shortest Path, Multiple Attributes, Dijkstra's Algorithm

[요 약]

그래프 이론에서 최소비용 경로는 시작 노드와 도착 노드 사이의 최단 경로를 탐색하여 구한다. 최소비용은 두 노드 사이의 거리나 가격의 차이를 1차원 값으로 계산하며 연결된 노드 사이의 최소비용의 합을 구성하는 노드와 간선이 최단 경로다. 그러나 각 노드가 다중속성을 갖는 경우에는 경로에서 나타날 수 있는 비용의 종류 또한 속성의 개수만큼이므로 최단 경로를 판단하기에는 어려움이 있다. 본 논문에서는 사용자의 다양한 요구사항을 만족할 수 있도록 유클리드 거리를 사용하여 다중속성을 반영한 최단 경로 탐색 기법을 제안한다. 실험에서는 1차원 값에 대한 최단 경로와 2차원 속성에 대한 유클리드 거리를 이용한 최단 경로가 다르게 탐색 되었다. 다중속성에서도 단일 속성과 차별화된 사용자의 선호 속성이 반영된 것으로 나타났다. 결과적으로 다중속성이 반영됨으로써 사용자의 다양한 요구사항을 만족시킬 수 있게 되었다.

▸ **주제어**: 유클리드 거리, 최단 경로, 다중속성, 데이크스트라 알고리즘

• First Author: Jongwan Kim, Corresponding Author: Dukshin Oh
*Jongwan Kim (kimj@syu.ac.kr), Smith College of Liberal Arts, Sahmyook University
*KwangJin Choi (kjchoi@syu.ac.kr), Smith College of Liberal Arts, Sahmyook University
**Dukshin Oh (ohds@syu.ac.kr), Dept. of Management Information Systems, Sahmyook University
• Received: 2020. 03. 31, Revised: 2020. 09. 25, Accepted: 2020. 10. 06.

http://www.ksci.re.kr pISSN:1598-849X | eISSN:2383-9945

I. Introduction

In subway travel, the shortest path between the starting point and destination represents a path involving the least cost when the path costs between stopovers are calculated. A path comprises nodes, which represent subway stations between the starting point and destination, and edges, which represent lines between stations. Two nodes that are connected by lines on a network or map can be routers or hosts. Moreover, buildings, mountains, or lakes on a map can correspond to products, tour destinations, and attractions in amusement parks.

Dijkstra's algorithm [1][2] can be used to identify this shortest path, which corresponds to optimizing a single objective. Identifying the path with the least cost generally involves examining one-dimensional costs such as distance, travel time, and the price difference between two nodes [3][4].

When traveling between two nodes, the travel speed of a user depends on the road conditions and the distance between the nodes. Therefore, it is difficult to satisfy user cost requirements using only one-dimensional attributes. In addition, when users search for a hotel, they do not make decisions based solely on the price of the room or its location. Consequently, the room price, location, and service quality should be considered simultaneously.

In addition to Dijkstra's algorithm [5][6], the Bellman-Ford shortest path algorithm [7][9] has been studied; however, it also calculates the connection line between two nodes as a cost. Therefore, it is difficult for any of the existing shortest path algorithms to calculate the least cost path while considering the tour destination attributes preferred by users because they cannot simultaneously consider multiple attributes such as price, distance, popularity, cleanliness, food, and convenience. Each attribute can be considered individually to create a series of individual shortest paths based on a single attribute, such as price or popularity. However, it is difficult to compare a particular path with the paths based on other attributes because the size and range of values may differ. For instance, if the shortest paths are calculated by considering the room rate and its convenience separately, the result obtained differs from that obtained when the two attributes are considered simultaneously. Therefore, multiple attributes must be considered to calculate the true least cost between two nodes that satisfies user requirements.

To address the disadvantages of existing shortest path search schemes that employ one-dimensional attributes, we propose a new shortest path calculation scheme that incorporates multiple attributes. Multiple attributes of each node are converted into one-dimensional costs between nodes to search for the shortest path based on user-preferred attributes. Multi-attribute models include additional items apart from the single attributes as they differ in terms of scale, units, and value direction. Multi attributes are used for users' preferences to satisfy of users' needs [10][11]. However, it is difficult to find the research of multi attributes and still applying one dimensional attributes in many studies. In this study, the attributes of nodes were compared using the Euclidean distance [7][8], and the paths were searched based on their similarity to user preferences.

The proposed scheme can be applied to areas other than the streets, such as subways, hotels, and amusement parks, where the user's satisfaction varies depending on the attributes.

The subway is affected by the distance between stations, travel time, arrival time, and ride location, and the hotel has multiple attributes such as price, quality of service and food or distance to the beach. In order to ride the rides, amusement parks determine the location, fare, boarding time, and level of fun of each ride from the user's location and explore the route that maximizes rides until exit time.

This paper is organized as follows. Section 1 explains the necessity for multiple attributes in the search for the shortest path. Section 2 discusses problems associated with existing shortest path schemes that do not reflect user preferences. Section 3 reviews scenarios and solutions for searching the short est path by reflecting user preferences using the Euclidean distance. Section 4 discusses the experimental results to determine the shortest path using the Euclidean distance. Finally, Section 5 discusses the results of the proposed scheme.

II. Problem Statements

This study aims to determine the shortest path from the starting point to the destination in network structures such as subways. Existing studies on shortest path search employ single attributes of each node for calculating the least cost. However, the path cost of travel between nodes is occasionally affected by multiple attributes. The travel cost may be affected by factors such as the distance between subway stations, waiting time, number of passengers, or differences in travel time, which depend on the number of trains.

The path cost between two nodes can be summarized using Heuristic 1 as follows:

Heuristic 2.1. Single attributes cannot reflect the path cost sufficiently.

When visiting multiple markets and determining the shortest path between them, calculating the path cost between two markets using only the distance may increase the time in the closer market because the road conditions and purchase time are not considered.

When single attributes are used, an attribute, such as the distance between markets, road conditions, or shopping time must be selected as the criterion for calculating the path cost and the value to be used as the least cost. However, these values are inappropriate for determining the shortest path because they belong to different domains. Therefore, it is more reasonable to determine the path using the cost calculated by considering these various attributes simultaneously.

Heuristic 2.2. Multiple attributes reflect user requirements.

To determine the least cost between stations in a subway, a reasonable path can be selected by simultaneously considering the distance, number of passengers using the station, and travel time, which depend on the number of passenger trains, rather than considering only the simple cost associated with distance.

Users generally include various attributes in their requirements. Hence, a user-oriented search for the shortest path can be accomplished via calculating the path cost by simultaneously comparing the multiple attributes of each node to the user requirements. In particular, considering various attributes related to the subway traffic information is necessary to accurately reflect user preferences in the path cost calculation.

Heuristic 2.3. Euclidean distance transforms multidimensional attributes to one-dimensional similarity.

The Euclidean distance is defined as the straight-line distance between two points, where each point has two attributes (x, y). When two or more attributes exist, the distance between the starting point and destination is calculated and represented as a one-dimensional dissimilarity by calculating the Euclidean distance for each attribute.

To obtain the Euclidean distance, the Pythagorean theorem is used to calculate the length of the hypotenuse that connects two points in a right triangle. These two points have two attributes (x, y), and even if the number of attributes increases, the distance between the attributes can be determined using the Euclidean distance. For example, if the attributes considered when a user wishes to buy a used car are car age, mileage, and price, five-dimensional attributes should be used. The proximity, which is the distance between a user and vehicle, can be determined by calculating the Euclidean distances between user-defi ned attributes and the attributes of each used car. Because the Euclidean distance is the distance between two points, the closer it is to zero, the closer it is to the attribute required by the user.

The Euclidean distance between two multi attributes can be expressed as shown in Eq. (1).

$$\sqrt{(p1-p2)^2+(q1-q2)^2}\,(p:property) \quad (1)$$

Table 1 lists the Euclidean distances for five used cars when the user searches for a used car that was used for 5 years, has a mileage of 50,000 km, and a price of 15 million won. Based on user requirements, A was discovered to have the shortest Euclidean distance for four attributes. This is called proximity or dissimilarity because the distance is used as the criterion.

Table 1. Euclidean distance for used cars

Car	Years	Km	Price	Euclidean Distance
A	3	60,000	1,000	10,012
B	5	80,000	1,200	30,001
C	5	100,000	800	50,005
D	4	35,000	1,800	15,003
E	5	20,000	2,000	30,004
User Attrib.	5	50,000	1,500	-

In this study, user requirements were incorporated by considering multiple attributes when searching for the shortest path.

When calculating the shortest path to the destination via stations in a subway, multiple attributes of each station must be determined to incorporate various user requirements. The existing shortest path scheme search for the shortest path use single attributes; however, they do not sufficiently consider the various user requirements. For example, the approximate value obtained using the one-dimensional values in Table 1 can be calculated as follows. If the user desires a vehicle that has been used for 5 years, vehicles B, C, and E satisfy this condition. However, the mileages of B and C are not close to the requirement, and the price of E is extremely high at 20 million won. Therefore, single attributes cannot be compared with the actual user requirements because they cannot reflect multiple attributes.

In this study, we attempt to provide services that satisfy user requirements by suggesting a shortest path search scheme for multiple attributes by reflecting the above mentioned Heuristics 1, 2, and 3.

III. Search for the Shortest Path using Multiple Attributes

The traditional shortest path problem finds the least-cost path to the destination node by calculating path costs between sub nodes that are fanned out from one node and selecting the sub nodes that provide paths to the destination that have the least cost. In this section, we propose a method to improve user satisfaction using a shortest path search scheme that considers multiple attributes simultaneously.

3.1. Multiple attributes and assumptions

In this study, it is assumed that a user is searching for the least-cost path to enjoy attractions at a festival. The user wishes to enjoy the festival by considering prices of attractions and distances between them. Distances and prices range from 0 to 100, and a value closer to zero indicates a higher level of user satisfaction. In this study, the shortest path was evaluated using one-dimensional (distance) and two-dimensional (distance, price) attributes.

Table 2 shows the dissimilarity calculated using the Euclidean distance for five attractions using distance and price as the attributes. The place of interest (POI) is an attribute that reflects user preferences regarding the attractions.

Table 2. Attributes and dissimilarity of attractions

Attrib.	A	B	C	D	E	POI
Dist.	30	50	70	45	95	35
Price	70	69	38	90	80	50
A	0.00	20.02	51.22	25.00	65.76	20.62
B	20.02	0.00	36.89	21.59	46.32	24.21
C	51.22	36.89	0.00	57.70	48.88	37.00
D	25.00	21.59	57.70	0.00	50.99	41.23
E	65.76	46.32	48.88	50.99	0.00	67.08

3.2. Shortest path search based on multiple attributes

The existing shortest path schemes start at a random node and subsequently visit nodes with the lowest path cost, based on a single attribute. The shortest path for nodes having multiple attributes can be determined by visiting nodes along the edge that can be visited with the least cost from one node.

In this study, user preferences were considered by calculating the path cost for multiple attributes as one-dimensional costs using the Euclidean dista nce.

Figure 1 shows examples of the path cost for one- and two-dimensional attributes in selecting attracti ons using the shortest path in a festival. The two attributes are the distance between attractions and the price. The distance is used as the path cost for visiting many attractions to minimize the travel time between attractions, whereas the price is used to minimize the additional cost of visiting the next attraction.

By evaluating the cost using only the distance between attractions, Figure 1 (a) demonstrates that the shortest path can be achieved by first visiting the closest attraction. Alternatively, considering only prices enables the least expensive path to be deter mined. However, if the two attributes are evaluated simultaneously, a path that enables visits to close attractions while spending the minimum amount can be obtained. Figure 1 (b) shows the shortest path calculated using the Euclidean distance for the (distance, price) pair, which are the attributes of the first selected node and its neighboring node.

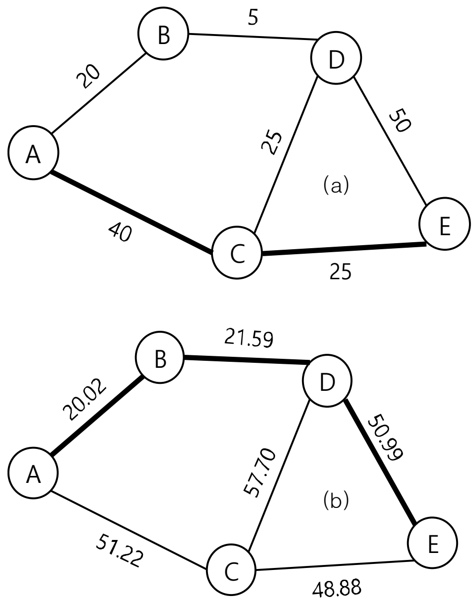

Fig. 1. (a) Shortest path using one-dimensional attributes and (b) two-dimensional attributes

As shown in Figure 1 (a), when distance was calculated using the path cost, the shortest distanc e was A-C-E. In Figure 1 (b), the A-B-D-E path was discovered by considering both the distance betwee n attractions and the price of each attraction. Figu re 1 (b) represents a path that considers multiple attributes by representing the dissimilarity between nodes B and C that are connected by links from node A in one dimension, which is then used as the path cost to obtain a path that reflects multiple attr ibutes. Finally, the node where the sum of paths is the lowest is connected to the edge, and more attr actions can be visited when the shortest path is co nfigured using only this path.

3.3. User preferences and shortest path

The existing shortest path scheme selects a rando m node as the starting point. However, when the start node is selected based on user preferences, the user requirements are considered because multiple attrib utes are evaluated.

3.3.1. Search for start node that reflects user preferences

User preferences can improve user satisfaction by considering multiple attributes from the first decision based on multiple attributes.

One shortest path search scheme that considers user preferences involves searching for a path by calculating the dissimilarity between the selected node and the pan-out node after selecting a node that is similar to the user preference. The charact eristics of this method can be explained by the following heuristic.

Heuristic 3.3.1. The similarity between a node close to the user preferences and its neighboring node reflect user preferences.

In existing shortest path one-dimensional search schemes that calculate the similarity between nodes regardless of user preferences, the starting node is selected randomly. If a start node that is close to the user preferences is selected, then the neighbor ing nodes whose Euclidean distance is also close to the user preferences are selected, instead of rando m nodes.

As shown in Figure 1, when the user's POI is (35, 50) for distance and price, the start node is obtained by calculating the Euclidean distance among all nodes. As shown in Table 2, the node that is closest to the user attributes is A, and when the dissimilarity between neighboring nodes is calculated using this node as the start point, we obtain Figure 1 (b). When multiple attributes are used, the service quality for the shortest path increases because detailed user requirements can be considered.

3.3.2. Path cost reflecting user preferences

Searching for a start node that is similar to user preferences incorporates additional user preferences, as compared with other methods. However, as shown in Figure 1 (b), the neighboring nodes calculate the Euclidean distance to the start node and become closer to the start node, which results in a few deviati ons from the user preferences. To search for a path that is closer to the user preferences, the Euclidean distance to the user attributes must be calculated to determine a path with a node whose path cost is close to the user's preference.

Figure 2 depicts the search for the shortest path based on user preferences. The cost of the edge connected to each node is calculated using the proximity between the user and each node instead of using the proximity between neighboring nodes. For example, the Euclidean distance associated with node B (24.21) is the value calculated between the user attribute POI (35, 50) and B (50, 69). When this method was used, the shortest distance was A-B-D-C-E.

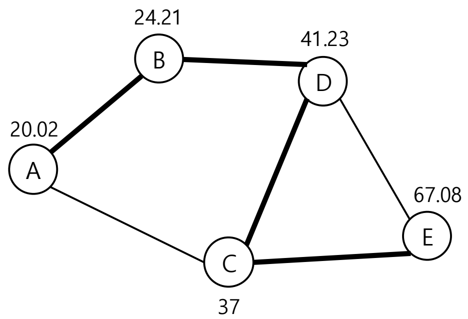

Fig. 2. Shortest path of user preferences

IV. Multi attributes and Shortest Path Experiment

In this section, we discuss the results of an experi ment to obtain the shortest path considering user preferences, using Euclidean distance.

4.1. Experimental environment

The shortest path algorithm is an important algorithm for determining the fastest path. In this algorithm, it is more practical to use approximately 10n data points that can be easily calculated rather than several different data points. Additional data can be used; however, the computation time increases with the number of data points. Therefore, we did not use additional data in this study.

Table 3. Experimental environment

Category	Contents
User's Preferences	Distance, Price: 35, 50
Nodes and Edges	10 nodes, 26 edges
Nodes' Attributes	(Distance, Price) in each object
Tool & Programming	MATLAB [12] & Script
System	Windows 10, Xeon 2 CPU, RAM 128 GB
Algorithm	Dijkstra's Algorithm, Euclidean Distance

The shortest path scheme is not the selection of a path that minimizes the cost of the edges linked to a node but the selection of a path that minimizes the total cost from the start node to the destination node. If a path that minimizes the edge cost of a node pair linked to each node is selected, the cost to the destination node may not be the lowest. Conversely, the cost to the destination node may be the lowest even if the cost of the linked nodes is not the lowest because the shortest path is the least-cost path from the start node to the destination node.

In this experiment, we used 10 nodes and 26 edges of the linked nodes as follows:
source_node = [1, 1, 1, 1, 2, 2, 2, 3, 3, 4, 4, 5, 5, 6, 6, 6, 6, 7, 7, 7, 8, 8, 9, 9, 10, 10]
linked_node = [2, 3, 8, 6, 3, 4, 9, 5, 10, 5, 3, 7, 8, 9, 3, 2, 4, 9, 4, 6, 3, 7, 3, 5, 9, 8]

4.2. Start node and user preferences

We examined two methods of selecting the start node by reflecting user preferences and obtaining the least-cost path described in the previous section. The first method compares only the start node of the shortest path with the user's multiple attributes, whereas the second method selects neighboring nodes by comparing all the nodes with user preferences. However, the results obtained via the first method differ significantly from the value expected from the user because the proximity with the previous (n-1)th node is calculated for the nodes selected after the first node, although the first node of the path is similar to the user preferences. Compared with the first method, the second method calculates the shortest path that is closer to the user preferences because it calculates the Euclidean distance based on the user's multiple attributes for the nodes linked to the start node.

In this experiment, we searched for the shortest path using the Euclidean distance for 10 nodes that had two attributes each and then represented the results in a graph. In this experiment, the evaluation of performance metrics such as search time and processing speed was excluded. One reason these data were excluded is that a considerably large number of visited nodes is required to measure the search time and processing speed. This increases the time required to compare multiple attributes, thereby degrading the practicality of the algorithm. Therefore, in this experiment, we focused on the proposed shortest path calculation scheme based on multiple attributes.

Figure 3 shows the shortest distances based on differences in the distance between each node. The difference in the distance between two nodes was the same as that of the existing schemes, and the shortest path from start node A was A-B-I-J.

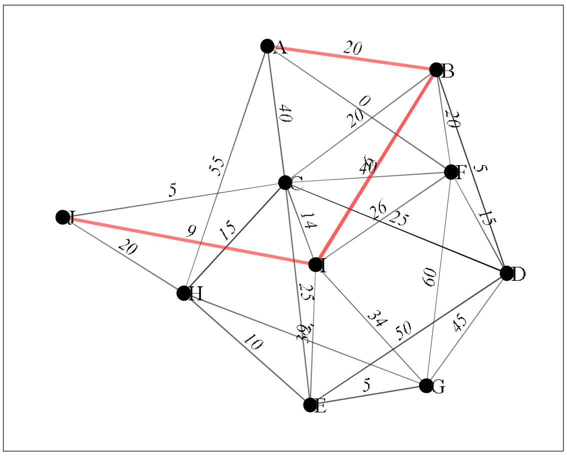

Fig. 3. Shortest path using distance

Figure 4 shows the shortest path determined by calculating the Euclidean distance for linked node pairs after selecting a random node A among 10 nodes having multiple attributes. The proximity to the node pairs linked with A was (A, B) = 20.024984, (A, F) = 7, (A, C) = 51.224994, and (A, H) = 66.2872 54. The shortest path to destination node J was A-B-I-J. The node closest to the neighboring node from node F was D = 19.8494, but path F~I was selected because the path cost to destination J was large. This means that the linked node with the least cost was not always selected when one moved from a node to its neighboring node. In Figure 4, the shortest path obtained by calculating the Euclidean distance with neighboring nodes was similar to the path reflecting the path cost used in Figure 3. However, it was clear that the shortest path was reset to a path that had a Euclidean distance similar to that of the start node by reflect ing the price attribute.

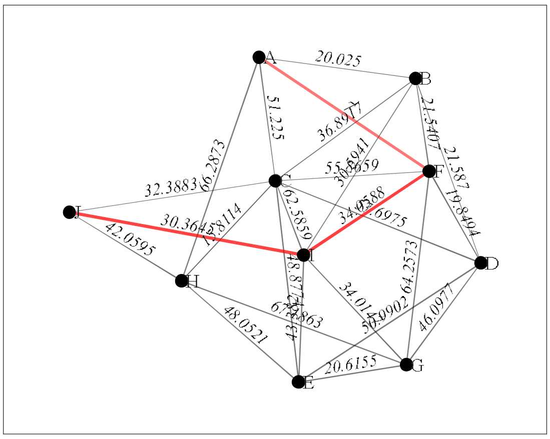

Fig. 4. Shortest path using the Euclidean distance of neighboring nodes

Figure 5 shows the path that minimizes the proxi mity to user preferences among the linked nodes B, C, F, and H and the following paths after selecting a random node among 10 possible nodes. The path cost of every node was the Euclidean distance betw een the user preferences and the multiple attributes of each node. The shortest path determined by calcul ating the POI value, which is a user attribute, and the Euclidean distance of every node was A-C-J; furthe rmore, (a, c) = 24.207437 was the proximity between the user's POI attribute and node C.

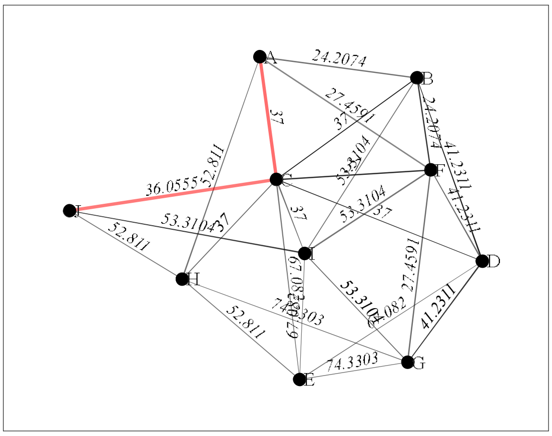

Fig. 5. Shortest path using user attributes and the Euclidean distance of each node

We verified that the shortest path of the single attributes differed from that of multiple attributes. Furthermore, the search result obtained using the proximity of neighboring nodes differed from that using the proximity of user attributes. This indicates that the shortest path search reflecting user prefere nces can be achieved by transforming user preferenc es to one-dimensional values.

V. Conclusions

The conventional shortest path scheme employing Dijkstra's algorithm calculated the path cost based on the single value of each node and determined the path that involved the least cost from the start node to the destination node. However, the nodes representing tour destinations, products, hotels, etc. are evaluated based on different attributes. Therefore, it is more practical to search for the shortest path using multiple attributes to satisfy user preferences.

This paper proposes a shortest path search scheme that calculates proximity using the Euclidean distanc es between user preferences and multiple attributes of data in terms of the services provided to users. The results showed that the shortest path obtained using only the distance differed from that obtained using the Euclidean distance. Hence, user satisfaction was considered during the search process by include ing multiple attributes in the path cost.

Typical practical applications of the results of this study are suitable for determining the route of the courier. The shortest path can be made up of mini mum cost depending on various attributes other than distance, as the courier service is affected not only by distance but also by travel time, apartment or housing conditions and road conditions. Using this algorithm, a courier can propose a route that maxim izes delivery within a day's working hours, focusing on various attributes.

This study calculates the shortest path close to the user's preference by additionally determining multi ple attributes based on distance. The assessment of objects without location properties was not applied because the target must include positions such as grades, price or size for the path of travel when deter mining the shortest path. However, the shortest path may be used to determine the path for product selection based on the similarity of attributes when selecting the product.

This study has a threshold for calculating the shortest path only by including location properties. The shortest path cannot be obtained because it is difficult to calculate similarity in colors, weather, etc., not location properties, which are difficult to compare, and is something to overcome in future studies.

Calculating users' preferences with multi attributes will affect influences to industry and it is important in research field to make researchers with new research field.

In the future, further research will be carried out in the direction of increasing multi-path based practical utilization by studying the shortest path for objects without location information.

ACKNOWLEDGEMENT

This paper was supported by the Sahmyook University Research Fund in 2020.

REFERENCES

[1] M. H. Alsuwaiyel, “Algorithms: Design Techniques and Analysis,” World Scientific, Vol. 14, 2016.
[2] M. A. Reba, and D. Shier, D. (2015). R., “Puzzles, Paradoxes, and Problem Solving: An Introduction to Mathematical Thinking,” CRC Press, 2015.
[3] R. Oosumi, K. Tamura, and K. Yasuda, “Novel single-objective optimization problem and Firefly Algorithm-based optimization method,” IEEE International Conference on Systems, Man, and Cybernetics (SMC), Budapest, pp. 1011-1015, 2016. DOI: 10.1109/SMC.2016.7844374
[4] O. Alam, and O. Faruq, “Finding Shortest Path for Road Network Using Dijkstra’s Algorithm,” Bangladesh Journal of Multidisciplinary Scientific Research, Vol. 1, No. 2, pp. 41-45, 2019. DOI: 10.46281/bjmsr.v1i2.366
[5] L. Parungao, F. Hein, and W. Lim, “Dijkstra Algorithm Based Intelligent Path Planning with Toplogical Map and Wireless Communication,” ARPN Journal of Engineering and Applied Sciences, Vol. 13, No. 8, pp. 2753-2763, 2018.
[6] M. Iqbal, K. Zhang, S. Iqbal, and I. Tariq “A Fast and Reliable Dijkstra Algorithm for Online Shortest Path,” SSRG International Journal of Computer Science and Engineering, Vol. 5, No. 12, pp. 24-27, 2018. DOI: 10.14445/23488387/IJCSE-V5I12P106
[7] J. Carey, S. Doshi, and P. Rajan, “C++ Data Structures and Algorithm Design Principles,” Packt Publishing, 2019.
[8] L. Morin, P. Gilormini, and K. Derrien, “Generalized Euclidean Distances for Elasticity Tensors,” J Elast 138, pp. 221-232, 2020. DOI: 10.1007/s10659-019-09741-z
[9] Y. Dinitz, and R. Itzhak, “Hybrid Bellman-Ford-Dijkstra algorithm,” Journal of Discrete Algorithms, Vol. 42, pp. 35-44, 2017. DOI: 10.1016/j.jda.2017.01.001
[10] K. Zhang, and K. Li, “A novel algorithm for TOP-K optimal path on complex multiple attribute graph,” International Journal of Embedded Systems, Vol. 12, No. 4, pp. 510-521, 2020. DOI: 10.1504/IJES.2020.107630
[11] L. Bidoux J. P. Pignon, and F. Bénabena, “Planning with preferences using Multi-Attribute Utility Theory along with a Choquet Integral,” Engineering Applications of Artificial Intelligence, Vol. 85, pp. 808-817, 2019. DOI: 10.1016/j.engappai.2019.08.002
[12] Matlab, *www.mathworks.com.*

Authors

Jongwan Kim received the Ph. D. degree in Computer Science and Engineering from Korea University, South Korea, in 2007. He researched at the University of New South Wales (UNSW), Sydney, Australia from 2012 to 2013 as a visiting fellow. Dr. Kim joined the faculty of the Smith College of Liberal Arts, at Sahmyook University, Seoul, Korea, in 2016. His research interests include Mobile&Streaming Data Management, Location-based Services, Big data and A.I.

KwangJin Choi received the B.S., M.S. and Ph. D. degrees in the Computer Engineering from Wonkwang University, Kwangwoon University and Joongbu University, Korea, in 1990, 1992 and 2005, respectively. Dr. Choi is working as a member of Korea Industrial Technology Promotion Agency Technology, Business Evaluation Committee and KISTI technical commercialization expert committee. He is currently a Professor in the Department of Smith College of Sahmyook University. He is interested in Big data, Artificial Intelligence, and Smart Factory. Awarded by the Minister of Education, Science and Technology.

Dukshin Oh received the Ph. D. degree in Management Information Systems from Sangmyung University in Korea. He is a professor of Management Information Systems at Sahmyook University in Korea. Dr. Oh has published many papers in several journals such as International Journal of Computer Science and Network Security, KIPS Transactions. His research interests include Management/Computer Information Systems, System Analysis and Design, e-Business and e-Learning systems.