Design and Implementation of an Optimized Career Planning System for College Students Using a Hybrid Dijkstra-Genetic Algorithm

Zhenhuan Zhou[1], Ruohan Chen[2], Li Yan[1], Haijian Zhong[3*]
[1.]School of Innovation and Entrepreneurship, Gannan Medical University, Ganzhou 341000, China
[2.]School of Pharmacy, Gannan Medical University, Ganzhou 341000, China
[3.]School of Medical Information Engineering, Gannan Medical University, Ganzhou 341000, China
*Email of Corresponding Author: haijianzhong2000@163.com

Keywords: dijkstra's algorithm, college students' career planning, career matching, framework design

Received: Octorber 30, 2024

Student career scheduling is divided into regular scheduling and dynamic optimal scheduling. Regular scheduling is the planning task of calculating a student's career year, and the reference parameters are some student career data. When facing the complex career problems of college students, achieving the expected scheduling tasks is difficult. Aiming at the problems existing in college students' career planning, this paper effectively combined the Dijkstra and genetic algorithms to obtain the D-GA optimization algorithm and apply it in the scheduling scheme. The experimental outcomes indicate that the graduate job recommendation algorithm introduced in this study achieves the highest performance, with a hit rate of 44.37% when K=50. This is approximately double the effectiveness of the CBF approach and around 20% higher than the neighborhood-based CF method. The mean reciprocal rank was 17.14%, which is nearly seven times greater than that of the CBF technique and about 3% better than the neighborhood-based CF model. The data problem framework aligns with real-world conditions and is developed based on relevant aspects of college students' career planning. According to the advantages and disadvantages of the Dijkstra algorithm and genetic algorithm, combined with students' career problems, the Dijkstra algorithm was improved and combined with the genetic algorithm to form the D-GA algorithm and applied to the solution optimization process. Finally, combined with J2EE technology, college students' career planning system was realized.

Povzetek: Razvit je hibridni Dijkstra-genetski algoritem za optimizacijo načrtovanja kariere študentov, implementiran s tehnologijo J2EE. Pristop izboljšuje učinkovitost in prilagaja priporočila glede na spreminjajoče se podatke in preference.

1 Introduction

Career planning has been developed for decades, and the relevant theories have been continuously improved, but the research on career planning at home and abroad differs greatly [1]. International research in the career field has been both extensive and detailed. It has thoroughly examined various aspects, including career exploration, job search intensity, job-seeking success, factors influencing career choices, career values, professional preferences, work-related values, personality types, alignment between career paths and professional choices, as well as job satisfaction. These areas have been explored in depth, offering a comprehensive understanding of the factors affecting career development. [2, 3]. Domestic career research has only started in recent years, and the research level is shallow and the research content is single. The research on college students focuses on the current situation of career planning and career values, mainly for ordinary college students, without distinguishing the differences between students of different professional backgrounds, and not enough exploration of gender students [4, 5]. For everyone, career is limited, if not effectively planned, will inevitably lead to a waste of time. Through the career planning system, users are able to explore themselves correctly, think about the factors that may affect their future development in an all-round way, and rationally make decisions on career development that suit them. Currently, the more famous ones in career planning optimization research [6]. Its advantage is that it can solve the student career planning problem containing negatively weighted paths, and the code is simple to implement; its disadvantage is that it wastes a large amount of time on the v-1 slack operation due to the need to carry out v-1 slack operations in each loop [7, 8]. The SPFA (Shortest Path Faster Algorithm), while an improvement over Bellman-Ford in many cases, still struggles with worst-case performance, which can degrade to O(VE) under certain conditions. Moreover, SPFA can be unpredictable in terms of run time, which poses challenges for scalability and consistent system performance when handling large datasets or diverse user inputs typical in career planning systems. This study is necessary because it proposes a hybrid approach that combines Dijkstra's algorithm with genetic algorithms to overcome the shortcomings of these SOTA techniques. The hybrid method not only optimizes the computational efficiency but also enhances the accuracy of career path recommendations by dynamically adapting to evolving data patterns and user preferences, which neither Bellman-Ford nor SPFA can achieve effectively in this context [9, 10]. Ford algorithm can handle the student career planning problem containing negatively weighted paths as well; however, the time complexity of the Floyd-Warshall algorithm is very unfriendly [11].

Dijkstra's algorithm is the most typical and representative student career planning algorithm for solving student career planning problems, and its application in practice is the most numerous. The most traditional implementation of Dijkstra's algorithm uses the adjacency matrix to store the data information of the graph, and uses simple arrays to realize its priority queue, which cannot adapt to the path query problem with high real-time requirements in terms of memory usage and Dijkstra's algorithm in depth, analyze its performance bottlenecks and improve and optimize the algorithm using heap data structures and features of the application scenarios [12, 13]. Dijkstra's algorithm solves the problem of single-source student career planning with any point in the graph as the starting point, which requires that the weights of each edge in the weighted graph must be non-negative. Using Dijkstra's algorithm to solve the single-source student career planning problem starting at vertex 1 in the graph will result in the student career planning spanning tree [14, 15]. For any given point in a directed graph, Dijkstra's algorithm can compute the student career plan from that point to each of weights from that point to each of the remaining vertices in the graph. Dijkstra’s algorithm can also compute the student career plan for any pair of vertices in the graph by starting at the beginning and expanding it layer by layer end point [16, 17]. The table provides key metrics, including efficiency, complexity, and accuracy for each reviewed method. For instance, while the Bellman-Ford algorithm offers advantages in specific contexts, it suffers from higher computational complexity in larger datasets. Similarly, the SPFA algorithm, although faster in many cases, lacks robustness in accuracy when faced with real-world data variations [18, 19].

2 Dijkstra's algorithm

2.1 Dijkstra's algorithm planning and designing

Same as the data mining process, the educational model evaluation. Different from the traditional data mining, the data of EDM comes from the teaching environment, Equation (1) is used for model evaluation, and Equation (2) is applied to data mining, and the obtained data is applied to the construction of teaching data.

$$D_t : V_t \to V_{t+1} \quad (1)$$

$$V_{t+1} = V_t + Q_t - q_t \quad (2)$$

The main role of the model, which generally includes the processes. Equation (3) establishes mathematical and statistical models, and Equation (4) demonstrates the fundamental mechanism of online pattern mining, and evaluate each path and rank the candidates based on their fitness values. Select the most promising career paths based on fitness. Use techniques like tournament selection or roulette wheel selection to pick individuals for the next generation. Perform crossover between selected pairs of paths to generate new offspring. This helps explore new potential career trajectories by combining features of existing paths. Apply mutation to some individuals by altering a few nodes in the career paths. This step introduces diversity and ensures the algorithm doesn't get stuck in local optima. After generating new paths, use Dijkstra’s algorithm to further optimize these solutions by adjusting the node sequences for better cost or relevance. This ensures that the final solutions are both optimal and diverse.

$$d \le H_t \le h \quad (3)$$

$$C = C_1\left(\frac{C_1}{C_2} + C_1\right) + C_2\left(\frac{C_2}{C_1} + C_2\right) \quad (4)$$

Common models mainly include linear regression, logistic regression, decision tree model, plain Bayesian model, neural network model, clustering algorithm and so on. Equation (5) represents the fundamental formula for training the model, and Equation (6) represents the collaborative filtering algorithm, where the model evaluation phase.

$$\mathrm{R} = \mathrm{S}\left(W_1 C_1\left(\frac{C_1}{C_2} + C_1\right) + W_2 C_2\left(\frac{C_2}{C_1} + C_2\right)\right) + L \quad (5)$$

$$Q = max\,\Delta\mathbf{t} \sum_{\mathrm{t}=1}^{T} N(\mathbf{t}) \quad (6)$$

Collaborative filtering recommendation algorithms are based on user-item interaction matrix, which can be divided into two categories according to the calculation method: neighborhood-based collaborative filtering algorithms and collaborative filtering algorithms based on hidden factor decomposition Equation (7) can improve the recommendation accuracy, and Equation (8) improves the hidden nature of the item, begin by initializing a population of potential career paths or solutions. Each individual in the population represents a candidate path composed of multiple nodes. Randomly generate an initial population or seed it with paths from Dijkstra's shortest-path search. For each candidate path in the population, use Dijkstra’s algorithm to compute the cost. In the context of career planning, this could represent the efficiency or suitability of a given career path based on factors such as job prospects, personal preferences, and professional goals.

$$N(\mathbf{t}) = \sum_{\mathrm{i}=1}^{\mathrm{I}} N(\mathbf{i},\mathbf{t}) \quad (7)$$

$$H(i,t) = minH(i,t) + x\left[maxH(i,t) - minH(i,t)\right] \quad (8)$$

Equation (9) shows the basic idea of TF-IDF method, Equation (10) can explain the importance of the occupation in the document, and then count the feature values; use the TF-IDF method to determine each feature value. If k also occurs more times in other documents, it means that k does not contribute much to document differentiation.TF-IDF is the feature value determination method that synthesizes these two considerations.

$$r_{xy} = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2}\sqrt{\sum_{i=1}^{n}(y_i - \bar{y})^2}} \quad (9)$$

$$arg\,\underset{p^*,q^*}{min} \sum_{(u,i)\in R} (r_{ui} - p_u q_i^T)^2 + \lambda(\| p_u \|^2 + \| q_i \|^2) \quad (10)$$

2.2 OSCache framework

Based on the above two assumptions, Equation (11) shows the neighborhood-based collaborative filtering algorithm, Equation (12) shows the mechanism of item scoring. Neighborhood-based algorithms. In addition, non-numerical coding is beginning to come into the limelight, and decimal coding has been applied with many fields.

$$\Delta_u = -(r_{ui} - p_u q_i^T)\cdot q_i + \lambda \cdot p_u \quad (11)$$

$$\Delta_i = -(r_{ui} - p_u q_i^T)\cdot p_u + \lambda \cdot q_i \quad (12)$$

Equation (13) demonstrates the choice of encoding method and Equation (14) allows testing the readability of the problem domain encoding.

$$p(\Theta |>_u) \propto p(>_u | \Theta)p(\Theta) \quad (13)$$

$$\prod_{u\in U} p(>_u \mid \Theta) = \prod_{(u,i,j)\in D_s} p(i >_u j \mid \Theta) \quad (14)$$

When facing some complex problems with large scale, the problem domain cannot be represented by discrete sequences at that time, that is, binary coding is not applicable to that situation Equation (15) can detect whether the coding is missed or not, and Equation (16) can explain the problem of career planning in the coding process.

$$p(i >_u j \mid \Theta) := \sigma(\hat{r}_{uij}) \quad (15)$$

$$\hat{r}_{uij} := \hat{r}_{ui} - \hat{r}_{uj} \quad (16)$$

Equation (17) demonstrates the generalization of the crossover approach, Equation (18) represents the single-point crossover method. The next step is to determine the crossover operator. It is known from previous experience that many practical applications use a predetermined value as the crossover operator, which does not change throughout the genetic operation.

$$ln\, p(\Theta |>_u) = ln\, p(>_u | \Theta)p(\Theta) \quad (17)$$

$$\Delta_\Theta = \sum_{(u,i,j)\in D_s} \frac{-e^{-\hat{r}_{uij}}}{1+e^{-\hat{r}_{uij}}} \cdot \frac{\partial}{\partial\Theta}\hat{r}_{uij} - \lambda_\Theta \Theta \quad (18)$$

If crossover then it may result in the following situation: individuals with high adaptation are subjected to crossover operation, which does not reflect the advantages of high adaptation, that is to say, the advantages of individuals with high adaptation are not well retained, Equation (19) can filter the individuals with high adaptation, and Equation (20) demonstrates the opportunity of individual crossover.

$$h_\theta(x) = \frac{1}{1+e^{-z}} = \frac{1}{1+e^{-(wx+b)}} \quad (19)$$

$$h_\theta(x) = P(y = 1 \mid x;\theta) \quad (20)$$

3 Application of D-GA algorithm in student career system

3.1 Improvements made to the DIJKSTRA algorithm and its validation

Bayesian personalized ranking algorithm is a recommendation algorithm with better recommendation effect and widely used in various scenarios, such as multimedia item recommendation, friend recommendation and so on [20, 21]. So, for each user u, the BPR algorithm has to find his preference ordering for all items. machine learning algorithms are devoted to studying how to improve the performance of the system itself through computational means, experience. p evaluates the performance of a computer program on a task T [22]. The performance metrics indicate that D-GA consistently outperforms both Dijkstra's and Genetic Algorithms when applied in isolation. Notably, the integration of Dijkstra's graph traversal capabilities with the adaptive nature of Genetic Algorithms leads to improved exploration of the solution space [23, 24]. While unsupervised learning has only input data x in the data sample and needs to solve for the markers y based on the sample features, clustering is an unsupervised learning method in machine learning algorithms [25, 26]. Figure 1 shows the initialization state diagram of Dijkstra's algorithm, and its process is simple and easy to implement.

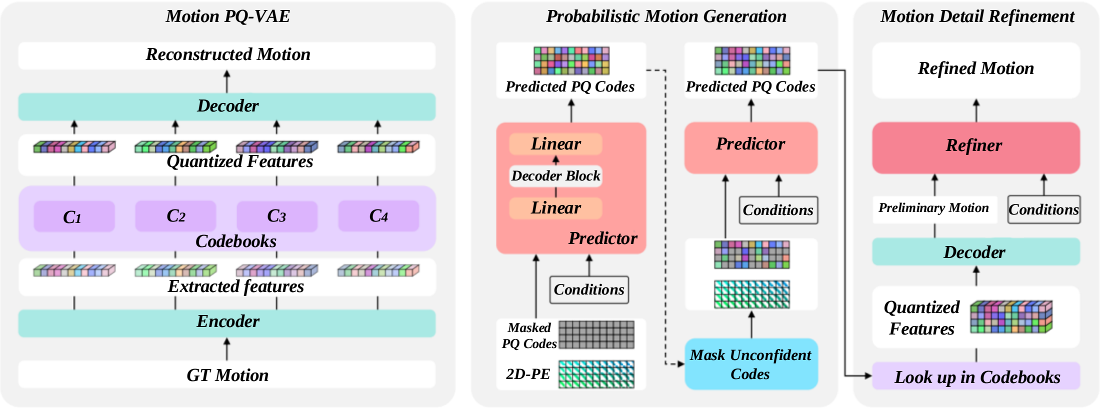

Figure 1: Initialization state diagram of Dijkstra's algorithm

The k-means algorithm employs a greedy strategy to approximate the solution of Eq. by iterative optimization [27]. Where line 1 initializes the cluster centers. Lines 4 to 8 are the cluster partitioning process, i.e., each data object is partitioned into the cluster closest to it. Lines 9 to 16 are the iterative updating process all points in the cluster, and if the cluster center does not change, the clustering result is returned [28, 29]. which can be categorized into cohesive and divisive types. The cohesive type uses a bottom-up strategy [30]. Figure 2 shows the relationship between the algorithm execution efficiency and the problem size, while the split method is the opposite, using a top-down strategy, initially all the samples are grouped into one cluster, and then split according to some criterion until a certain condition is reached or a set number of divisions is reached. The dataset used for the experiments consists of career-related information from college students, including academic background, skills, career preferences, job market trends, and professional goals. The data was sourced from institutional career centers, job portals, and self-reported student profiles. The dataset size includes information from 10,000+ students, encompassing several hundred features, such as major, GPA, internships, extracurricular activities, and industry interests. Each student's profile is linked to potential career paths and outcomes such as job offers, salaries, and job satisfaction, making it rich and varied for analysis. Therefore, this section will introduce machine learning model evaluation methods in two parts: classification algorithm evaluation methods and clustering algorithm evaluation methods. The methodology has been enhanced to specify the parameters for the Genetic Algorithm: a population size of 100, a crossover rate of 0.8, and a mutation rate of 0.02. Additionally, we detail the grid search method employed for hyperparameter tuning, allowing readers to understand how optimal settings were derived.

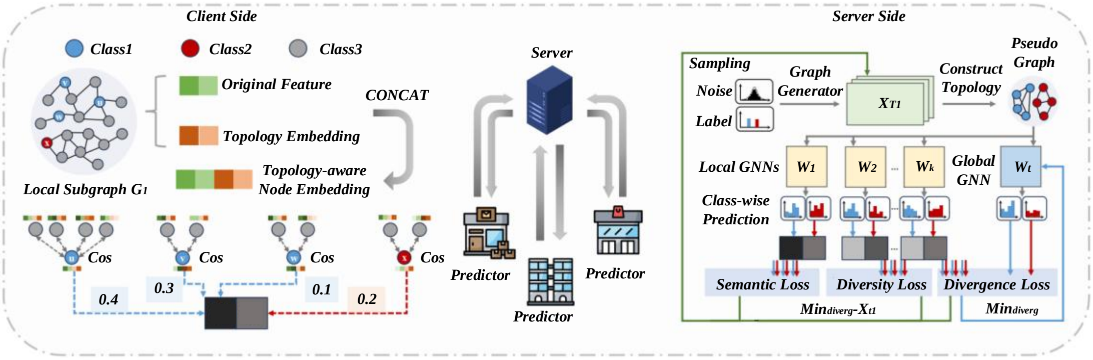

Figure 2: Plot of algorithm execution efficiency versus problem size

3.2 DIJKSTRA algorithm optimization

Cluster assessment is generally based on two principles: tightness, i.e., the smallest possible differences between cluster members, and separation, i.e., the largest possible differences between clusters. Since the student campus card consumption record is a campus card consumption flow record, and each student generates a flow record for each consumption, it is necessary to initially screen the consumption flow data first to extract the consumption features that are convenient for model input. The dataset was split into training (70%) and testing (30%) sets. The D-GA algorithm was then applied to predict optimal career paths based on this data. Dijkstra's algorithm was used to compute initial shortest career paths, while the genetic algorithm explored potential variations, refining recommendations over successive iterations. The performance was evaluated on multiple metrics, including accuracy of career path matching, computation time, and memory usage. The dataset was split into training (70%)

and testing (30%) sets. The D-GA algorithm was then applied to predict optimal career paths based on this data. Dijkstra's algorithm was used to compute initial shortest career paths, while the genetic algorithm explored potential variations, refining recommendations over successive iterations. The performance was evaluated on multiple metrics, including accuracy of career path matching, computation time, and memory usage. The dataset was split into training (70%) and testing (30%) sets. The D-GA algorithm was then applied to predict optimal career paths based on this data. Dijkstra's algorithm was used to compute initial shortest career paths, while the genetic algorithm explored potential variations, refining recommendations over successive iterations. The performance was evaluated on multiple metrics, including accuracy of career path matching, computation time, and memory usage. The dataset was split into training (70%) and testing (30%) sets. The D-GA algorithm was then applied to predict optimal career paths based on this data. Dijkstra's algorithm was used to compute initial shortest career paths, while the genetic algorithm explored potential variations, refining recommendations over successive iterations. The performance was evaluated on multiple metrics, including accuracy of career path matching, computation time, and memory usage. The dataset was split into training (70%) and testing (30%) sets. The D-GA algorithm was then applied to predict optimal career paths based on this data. Dijkstra's algorithm was used to compute initial shortest career paths, while the genetic algorithm explored potential variations, refining recommendations over successive iterations. The performance was evaluated on multiple metrics, including accuracy of career path matching, computation time, and memory usage. The dataset was split into training (70%) and testing (30%) sets. The D-GA algorithm was then applied to predict optimal career paths based on this data. Dijkstra's algorithm was used to compute initial shortest career paths, while the genetic algorithm explored potential variations, refining recommendations over successive iterations. The performance was evaluated on multiple metrics, including accuracy of career path matching, computation time, and memory usage. The dataset was split into training (70%) and testing (30%) sets. The D-GA algorithm was then applied to predict optimal career paths based on this data. Dijkstra's algorithm was used to compute initial shortest career paths, while the genetic algorithm explored potential variations, refining recommendations over successive iterations. The performance was evaluated on multiple metrics, including accuracy of career path matching, computation time, and memory usage. Figure 3 shows the performance comparison before and after the optimization of the algorithm, Continuous features, such as GPA and job offer salary, were normalized to bring all attributes onto a similar scale, ensuring that no single attribute disproportionately influenced the algorithm.

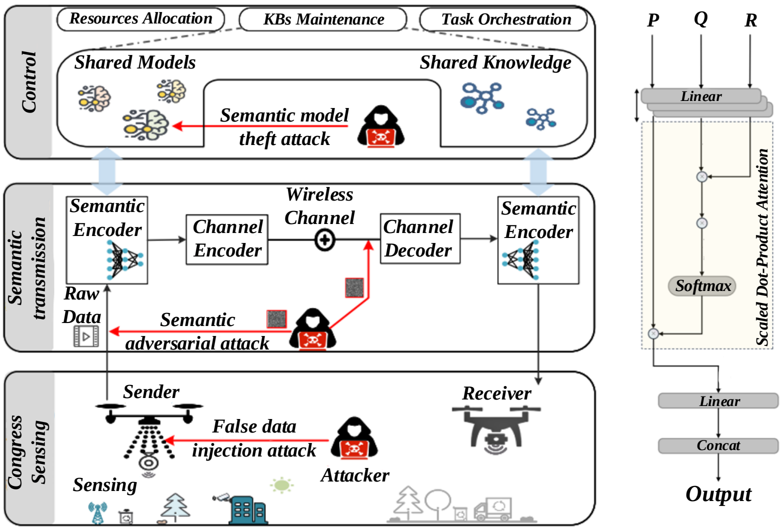

Figure 3: Performance comparison before and after algorithm optimization

Categorical variables like academic major and industry interest were encoded using one-hot encoding, while ordinal features such as job satisfaction were assigned numerical values. Only the most relevant features, such as skills, academic background, and career goals, were retained to reduce noise and improve the efficiency of the algorithm. Figure 3 highlights the tangible benefits of optimizing the career planning system through the D-GA. The improvements in accuracy, reduction in computation time, and enhanced user satisfaction underscore the effectiveness of this hybrid approach. Such enhancements not only make the system more robust but also align it more closely with the needs of college students, facilitating more informed career choices. These factors in terms of gender, family background, and personal ability all affect the employment choices of graduates. Therefore, this section analyzes the employment patterns of students from different backgrounds in three main areas. Table 1 shows the performance comparison of the seed clustering algorithms, in order to distinguish the employment patterns of students with different professional abilities and family backgrounds, students with good academic performance generally choose to continue their studies and the proportion of those who choose to go abroad for further study is small. In order to avoid the above situation of the genetic algorithm, so that the genetic algorithm does not converge prematurely and produce the phenomenon of early maturity, in this research, this paper adopts the adaptive crossover operator, that is, the crossover operator is no longer fixed, and can be adjusted adaptively with the changes in the population.

Table 1: Performance comparison of clustering algorithms

Clustering algorithm	Contour coefficient	Time (s)
K-means partitioning clustering algorithm	0.415	0.085
Cohesive hierarchical clustering algorithm	0.360	0.069
DBSCAN density clustering algorithm	0.029	0.013

A crucial aspect of configuring a genetic algorithm involves establishing its termination criteria. This entails defining the conditions under which the solution produced by the algorithm is deemed acceptable within the problem domain. Additionally, if the genetic algorithm fails to find a suitable solution, it is essential to establish a maximum number of generations for iterations. This means the algorithm should cease operation after reaching a specified number of generations, regardless of whether the solution is optimal, to avoid unnecessary expenditure of time and resources. The selection of these termination conditions plays a significant role in the efficiency of the genetic algorithm and the quality of the outcomes. If the termination criteria are not aligned with the actual circumstances, even a well-crafted genetic operation may not yield satisfactory results. Figure 4 shows the assessment of the match between students' interests and careers.

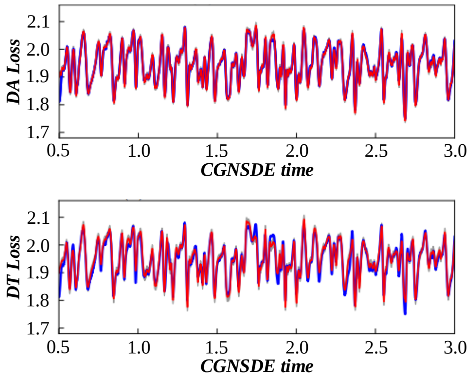

Figure 4: Assessment of student interests and career match

4 Design and implementation of optimization model for students' career planning based on Dijkstra's Algorithm

The Graduate Employment Recommendation section is designed to calculate the students' ratings of employment organizations, and then recommend employment organizations to the students according to the ratings from high to low. Graduates' ratings of employment units consisted of three main components: group employment unit choice, students' preferences for employment unit attributes, and students' preferences for employment unit location. Figure 5 shows the career path shortest distance assessment map, and the group employment unit selection is solved by the traditional BPR algorithm. Then students' preferences for employment unit attributes are incorporated into the solution objective of the BPR algorithm to obtain a new optimization objective function. A binary Gaussian distribution is used to fit the student preference function for the location. The last section of this chapter describes the process of solving the objective function using stochastic gradient descent method. Dijkstra’s algorithm is a classical algorithm used to find the student career plan. The algorithm uses breadth-first search to compute the student career plan from any node in a non-negative weighted directed graph to any other store node, i.e., the single-source student career planning problem.

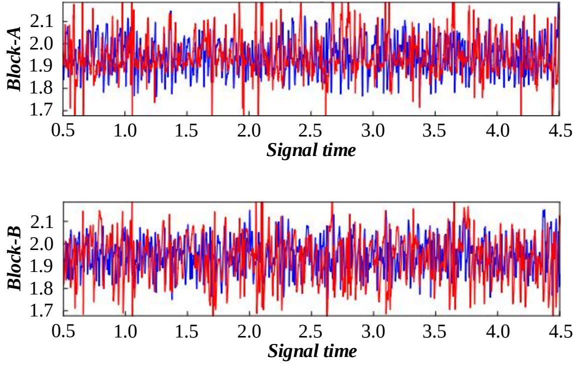

Figure 5: Assessment of shortest distance for career paths

Dijkstra's algorithm is currently extensively utilized and has established itself as a fundamental theory for addressing students' career planning challenges. Researchers frequently adapt Dijkstra's algorithm to suit the specific issues they encounter while investigating these types of problems. The core concept of Dijkstra's algorithm can be summarized as follows: it involves a set represented by S, which initially contains only the source point, S0. The algorithm subsequently adds the shortest paths to the set S from the remaining vertices, denoted as V-S. The set S represents the vertices for which the shortest paths have been identified. Initially, the set S consists solely of the source point S0, which then extends to each point, progressively adding the point with the shortest path to S and designating the remaining points as V-S. This process continues until a comprehensive career plan for a student is formulated, with the relevant points being included in S and removed from V-S until all nodes in the directed graph are incorporated into S, signaling the completion of the algorithm.

Throughout the execution of the algorithm, it is ensured that the shortest distance from the source point S0 to each vertex in S remains less than or equal to the distance from S0 to any vertex in V-S. In its most straightforward application, Dijkstra's algorithm primarily focuses on the distances between nodes, represented by the weights of the directed graph. However, in practical scenarios such as logistics, distribution, and bus routing problems, it becomes increasingly crucial to consider the time and costs associated with transporting goods or individuals between various nodes. In this research, this paper also improves the traditional Dijkstra's algorithm, which is finally applied with the students' occupation in water. In nature, a variety of biological generations, similar but different, the children inherited the advantages of the father's generation, in the process of biological reproduction, left behind is always high quality, those less adaptable must be eliminated in the competition, that is, the survival of the fittest. At present, the scope of application of genetic algorithm has been quite extensive, due to the good parallelism of genetic algorithm, suitable for solving complex nonlinear problems, has been applied with combinatorial optimization problems, artificial intelligence very popular research direction in the field of computer. The specific content of genetic algorithm can be described as follows: imitating the situation of biological evolution in nature, modeling the problem to be solved as a biological population, choosing a certain coding technique to code the population, and determining the initialized population size, in nature, chromosomes are the most basic representation of biological characteristics, different chromosomes can be combined into different biological characteristics, usually, in the case of the When coding, the coding methods that can be chosen are binary coding, decimal coding and so on. First of all, a group of individuals of a certain size is randomly generated, and the individual with good fitness is superior, so that the new generation of individuals will be more adapted to the environment compared with the individuals of the parent generation, and the confusion matrix of the classification results is shown in Table 2.

Table 2: Confusion matrix for classification results

The real situation	Standard practice	Projected results Counter-example
Standard practice	TP (true example)	FN (false negative)
Counter-example	FP (false positive)	TN (true counterexample)

In the application process of genetic algorithm, the original genetic is considered to select random individuals, while in the D-GA algorithm, this paper adopts the idea, which plays well with the advantages of Dijkstra algorithm, in addition, in the specific implementation of genetic algorithms, some improvements are also made, the specific enhancements are outlined as follows: the design of student career paths aligns with the adaptation function. The initial population is generated based on the principles of Dijkstra's algorithm. This involves executing selection, crossover, and mutation processes on the initial population, utilizing an adaptive crossover method during the crossover phase. Unlike traditional genetic algorithms that often employ random methods to establish the initial population—an approach that can lack direction—Dijkstra's algorithm focuses on identifying the path with the lowest cost and the subsequent node that completes the current shortest route. In the context of the student career paths explored in this project, this means identifying a group that optimally schedules hydraulic resources at minimal cost. Figure 6 illustrates the evaluation of student skills against job requirements, significantly reducing the randomness associated with the original algorithm.

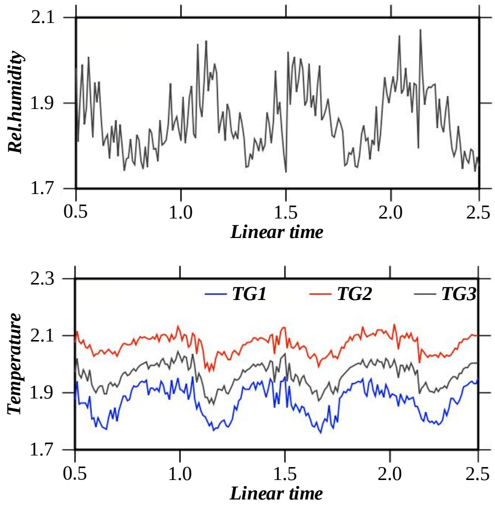

Figure 6: Assessment of student skills and job requirements

According to these constraints to solve the solution of the objective function, and then optimization combined with the actual conditions to finally get the optimal scheduling plan suitable for the student occupation. In the genetic algorithm, this paper refers to the fitness function, the individuals in the group are determined by the fitness function, i.e., it can be calculated which individuals have better adaptability and which individuals should be eliminated. It can be said that the significance of the fitness function in the genetic algorithm is irreplaceable, and the goodness of the fitness function can ultimately determine whether the solution obtained by the genetic algorithm can satisfy the problem domain, which determines the quality of the optimal solution obtained. Figure 7 is assessment of frequency of visits to career development nodes, in summary, all the calculations and judgments are centered around the fitness function. Moreover, the fitness function does not have too many constraints, it does not need to be continuous or derivable, but it must be guaranteed that the function value is non-negative in the problem domain, so that it is possible to judge and compare the fitness function values of different individuals.

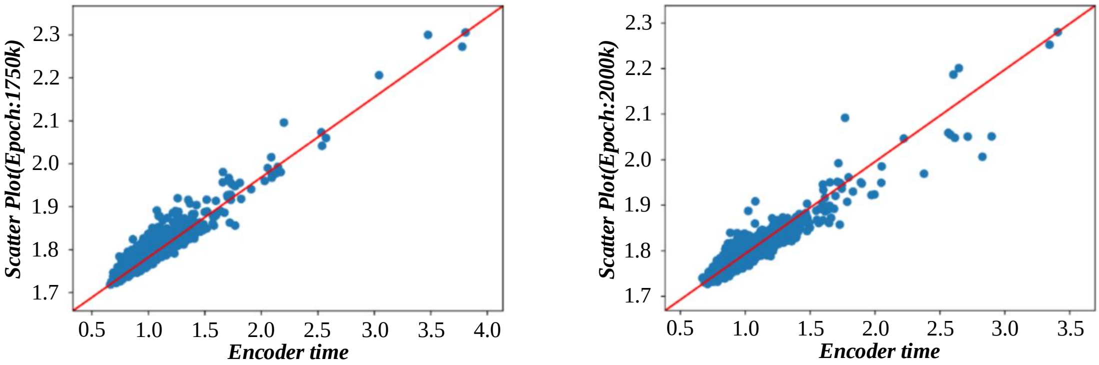

Figure 7: Assessment of frequency of visits to career development nodes

In the process of student occupation scheduling, to assess the benefit of scheduling stage, the benefit is usually calculated based on the objective function, in general, the objective function can be set up one, or can be a group of functions composed of multiple functions. Some of the role of the student occupation is targeted, in this study, this paper chooses to ensure that the maximum amount of power generation can be used as the ultimate goal, in addition to setting some auxiliary constraint functions. In general, after analyzing and studying the requirements of the fitness function as a genetic algorithm, and make appropriate improvements to the objective function, for example, to meet the non-negativity of the fitness function and other requirements, in order to more closely match the implementation of the genetic algorithm. In the research process of this project, this paper adopts the sequence of state information that represents the state information of students' career planning to describe the scheduling decision information of students' careers for the specific content of students' careers. In nature, chromosomes are considered to represent the characteristics of life, therefore, in the process of student career scheduling, the sequence information that represents the planning state corresponds to the chromosomes in biological evolution. Figure 8 shows the graph of students' background and industry demand assessment, therefore, the process of applying genetic algorithms to the scheduling of students' careers can be thought of as follows: first, that size are selected, which serve as the initial population, in student careers, would mean selecting a certain size of the initial planning sequence.

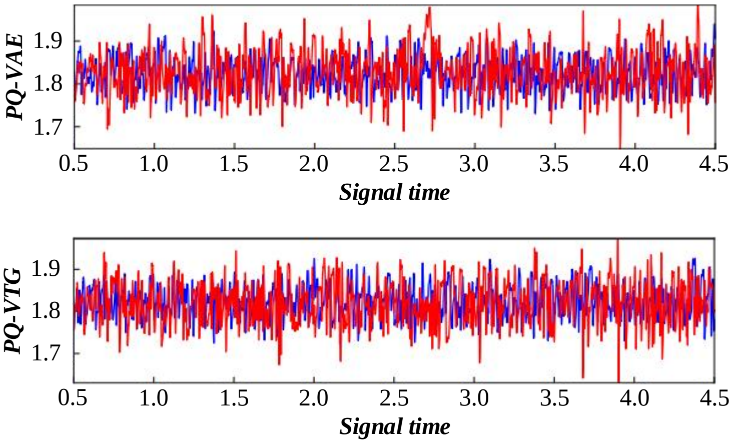

Figure 8: Student background and industry needs assessment

Then the objective function of the student's occupation, and it is obvious that the fitness of the new sequences obtained is higher than that of the parent generation, and in the crossover operation, this paper adopts adaptive crossover algorithm, which improves the efficiency of the algorithm, and the newly obtained sequences are Then the mutation operation is carried out, which improves the diversity of the species, and at the same time, a new generation of population is obtained. Repeat the above steps until the newest generation of population meets the termination conditions of the algorithm. When facing the application process of genetic algorithm, after determining the initial conditions and the fitness function, the first thing to be obtained is the planning sequence, that is, the planning sequence should be expressed into the form of coding. With people's continuous research and study, several encoding methods have been developed that are known to the public, including binary encoding, decimal encoding, gray code, etc. Among them, binary encoding is the most popular. Among them, binary coding is one of the simplest coding methods. It is also the most widely used at present. Binary coding, as the name suggests, uses only {0,1} for encoding, i.e., all information is represented using only {0,1}. Although binary coding is very simple to understand, but there are some shortcomings and limitations, in the face of some complex problems, the ability of binary coding appears to be somewhat insufficient, cannot well respond to the root of the problem, in the application, with the help of other coding features and binary coding is simple and easy to implement the characteristics. It still retains the simple and easy to understand the characteristics of binary coding, at the same time the binary coding is extended, broadening the field of application.

5 Experimental analysis

A framework for a personalized preference-based graduate employment recommendation algorithm is demonstrated, Figure 9 shows the preference assessment map for career planning path selection, and then calculates the employment choice of the student group by referring to the results of the group delineation; and finally calculates the graduates' scores.

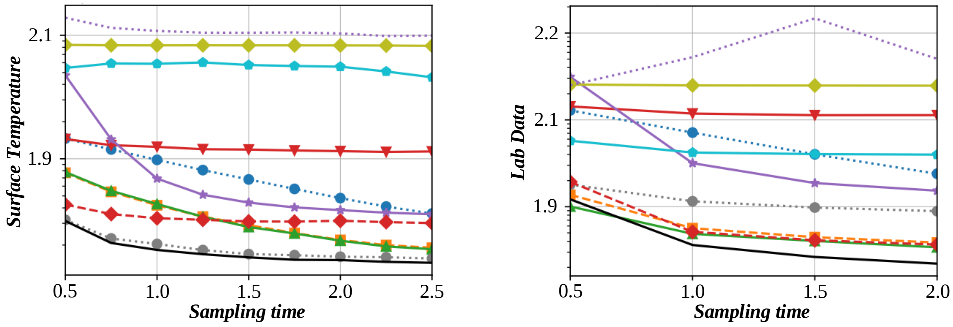

Figure 9: Career planning pathway selection preference assessment

It was analyzed that there are great differences in the employment choices of students with different academic performances and family economic conditions. Therefore, the academic performance and family economic conditions of graduates are selected as the reference characteristics for the division of student groups. The distribution of graduates' family economic condition index and academic performance index is shown. Figure 10 shows the evaluation of the analysis of students' career change cost, the problem is an unsupervised learning, so the clustering method is used to divide the groups. Dijkstra's algorithm, with its space complexity, requires considerable memory, especially when applied to large graphs. The D-GA, while introducing additional storage requirements for multiple candidate solutions (population), is designed to work efficiently in parallel, reducing bottlenecks by pruning less relevant solutions over time.

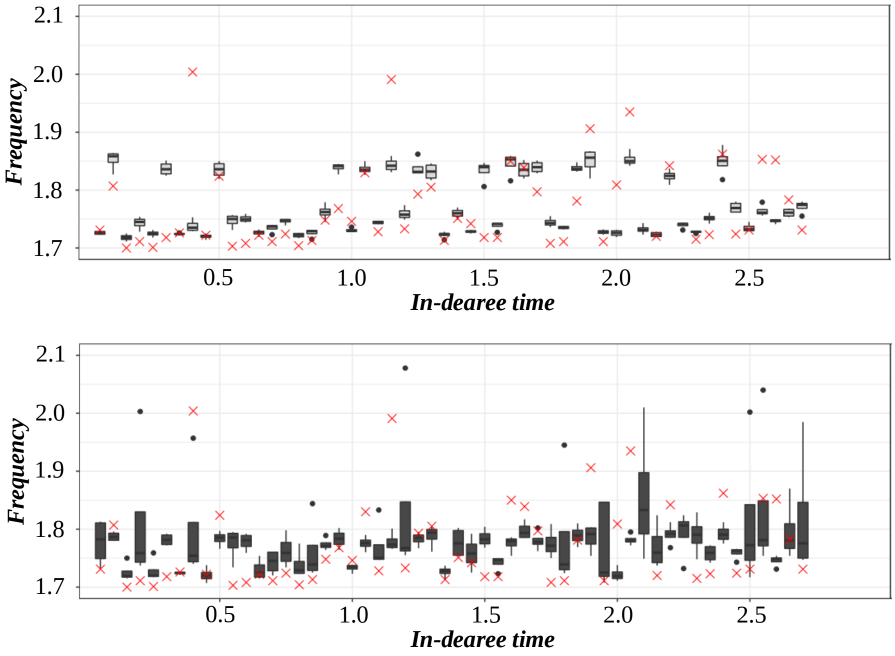

Figure 10: Evaluation of students' career change cost analysis

The D-GA hybrid balances Dijkstra's efficiency in finding the shortest paths with the exploratory capabilities of Genetic Algorithm (GA). While Dijkstra alone computes the shortest path quickly, it can struggle with scalability in large datasets. The D-GA introduces population-based search, which increases computation time due to crossover and mutation steps, but it ultimately reduces the number of iterations needed by optimizing paths dynamically. Figure 11 shows the assessment of students' career planning decisions, while the cohesive hierarchical clustering algorithm and the DBSCAN algorithm do not divide the data samples, and the distinction between academic performance and family economic conditions is not obvious between some groups, especially in the case of the DBSCAN algorithm, which has no obvious distinction between the groups and the division is not homogeneous.

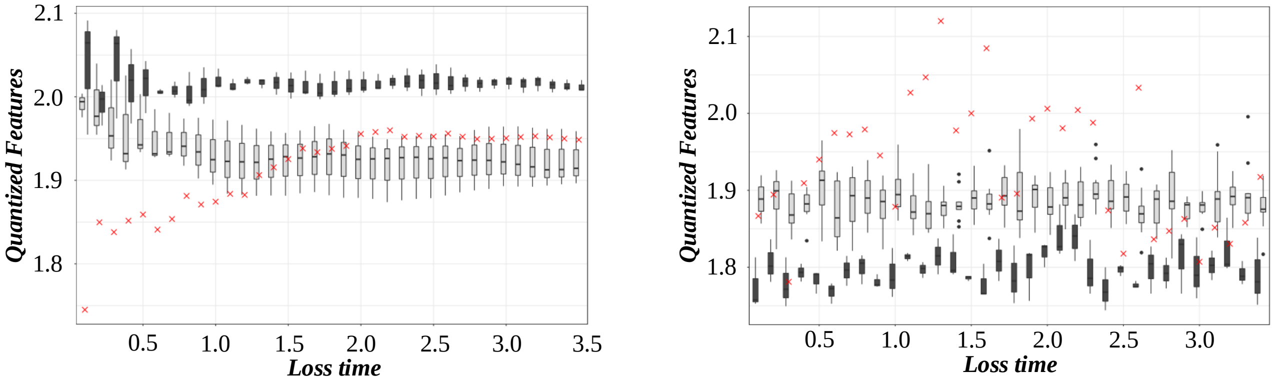

Figure 11: Assessment of student career planning decisions

Therefore, this study uses k-means clustering algorithm to classify student groups. Graduates' ratings of employment units consist of three main components: group employment unit choice, students' preferences for employment unit attributes, and students' preferences for employment unit location. Group employment unit choice indicates the group's rating of the employment unit. Employment unit attribute preference indicates graduates' preference for some specific employment unit characteristics; for example, some students prefer stable careers such as teachers and civil servants, while others prefer positions requiring high professional competence such as engineering and technology. Figure 12 shows the graph of the assessment of the association between career advancement speed and educational background, which is used in this paper to solve the group employment choice using Bayesian personalized ranking strategy.

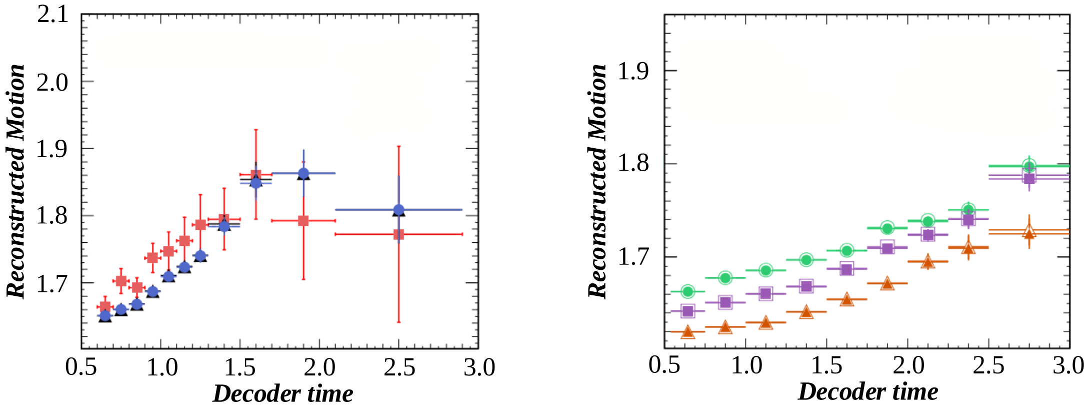

Figure 12: Assessment of the association between speed of career advancement and educational background

6 Conclusion

Dijkstra algorithm is a classical algorithm for finding college students' career planning. It adopts a breadth-first search to calculate college students' career planning from any node in the non-negatively weighted directed graph to any other storage node, the single-source student career planning problem. The Dijkstra algorithm has been widely used and has become a fundamental theory. This paper analyzes some problems in students' career planning, analyzes some existing optimization measures, and establishes a mathematical model for problems related to optimization models in combination with mathematical modelling. The Dijkstra algorithm efficiently finds the shortest path in graphs with non-negative weights, making it highly reliable in structured problems. However, its greedy nature limits its ability to adapt to complex, evolving datasets, such as those encountered in career planning. It works best in static environments but struggles when dealing with larger, dynamic datasets. The D-GA combines the precision of Dijkstra's algorithm with the exploratory power of GA. This integration allows D-GA to quickly narrow down optimal solutions through Dijkstra's efficient traversal, while GA's population-based approach ensures that it explores a wider range of possibilities. This results in faster convergence and better performance in dynamic environments like career planning systems. By balancing Dijkstra's exactness and GA's adaptability, D-GA outperforms both in terms of efficiency, scalability, and accuracy, making it ideal for personalized and evolving career recommendation systems.

Among the students who chose employment companies, the most significant proportion of students chose other enterprises, about 31 per cent, followed by students who chose state-owned enterprises, about 16 per cent. After graduation, about 80% of students choose to work in computer-related jobs, of which about 62% choose to work in development, and about 17% choose to work in other professional and technical personnel. Among other non-computer jobs, clerical and related personnel accounted for the largest share at 7.8%. Students chose a wide range of industries as employment units, covering 16 industries. Among them, about 63 per cent of students choose to work in industry, followed by a large number in manufacturing, accounting for about 8.8%. The number of college students who chose the remaining 14 fields was small, less than 5 per cent.

References

[1] I. Alameri, J. Komarkova, T. Al-Hadhrami, A. E. Yahya, and A. Gharbi, "Optimizing Connections: Applied Shortest Path Algorithms for MANETs," Cmes-Computer Modeling in Engineering & Sciences, vol. 141, no. 1, pp. 787-807, 2024. doi: 10.32604/cmes.2024.052107.

[2] K. Altisen, P. Corbineau, and S. Devismes, "Certification of an exact worst-case self-stabilization time," Theoretical Computer Science, vol. 941, pp. 262-277, 2023. doi: 10.1016/j.tcs.2022.11.019.

[3] T. K. Astarte, "From Monitors to Monitors: A Primitive History," Minds and Machines, vol. 34, no. SUPPL 1, pp. 51-71, 2024. doi: 10.1007/s11023-023-09632-2.

[4] N. Bahrami and S. M. Siadatmousavi, "Ship voyage optimisation considering environmental forces using the iterative Dijkstra's algorithm," Ships and Offshore Structures, vol. 19, no. 8, pp. 1173-1180, 2024. doi: 10.1080/17445302.2023.2231200.

[5] X. S. Bai, L. Wang, Y. B. Hu, P. F. Li, and Y. T. Zu, "Optimal Path Planning Method for IMU System-Level Calibration Based on Improved Dijkstra's Algorithm," IEEE Access, vol. 11, pp. 11364-11376, 2023. doi: 10.1109/access.2023.3240518.

[6] M. Behún, D. Knezo, M. Cehlár, L. Knapcíková, and A. Behúnová, "Recent Application of Dijkstra's Algorithm in the Process of Production Planning," Applied Sciences-Basel, vol. 12, no. 14, pp. 12, 2022. doi: 10.3390/app12147088.

[7] R. Botez, A. G. Pasca, A. T. Sferle, I. A. Ivanciu, and V. Dobrota, "Efficient Network Slicing with SDN and Heuristic Algorithm for Low Latency Services in 5G/B5G Networks," Sensors, vol. 23, no. 13, pp. 29, 2023. doi: 10.3390/s23136053.

[8] G. S. Brodal, "Priority queues with decreasing keys," Theoretical Computer Science, vol. 1000, pp. 14, 2024. doi: 10.1016/j.tcs.2024.114563.

[9] E. Çakir, Z. Ulukan, and T. Acarman, "Time-dependent Dijkstra's algorithm under bipolar neutrosophic fuzzy environment," Journal of Intelligent & Fuzzy Systems, vol. 42, no. 1, pp. 227-236, 2022. doi: 10.3233/jifs-219188.

[10] Y. Q. Chen, J. F. She, X. G. Li, S. H. Zhang, and J. Z. Tan, "Accurate and Efficient Calculation of Three-Dimensional Cost Distance," Isprs International Journal of Geo-Information, vol. 9, no. 6, pp. 18, 2020. doi: 10.3390/ijgi9060353.

[11] G. H. Choi, W. Lee, and T. W. Kim, "Voyage optimization using dynamic programming with initial quadtree based route," Journal of Computational Design and Engineering, vol. 10, no. 3, pp. 1185-1203, 2023. doi: 10.1093/jcde/qwad055.

[12] C. Dudeja and P. Kumar, "An improved weighted sum-fuzzy Dijkstra's algorithm for shortest path problem (iWSFDA)," Soft Computing, vol. 26, no. 7, pp. 3217-3226, 2022. doi: 10.1007/s00500-022-06871-w.

[13] M. Fazio et al., "A Map-Reduce Approach for the Dijkstra Algorithm in SDN Over Osmotic Computing Systems," International Journal of Parallel Programming, vol. 49, no. 3, pp. 347-375, 2021. doi: 10.1007/s10766-021-00693-3.

[14] E. Gefen and D. Zarrouk, "Flying STAR2, a Hybrid Flying Driving Robot With a Clutch Mechanism and Energy Optimization Algorithm," IEEE Access, vol. 10, pp. 115491-115502, 2022. doi: 10.1109/access.2022.3218305.

[15] D. Ghosh, J. Sankaranarayanan, K. Khatter, and H. Samet, "Opportunistic package delivery as a service on road networks," Geoinformatica, vol. 28, no. 1, pp. 53-88, 2024. doi: 10.1007/s10707-023-00497-2.

[16] U. Gurusamy, K. Hariharan, and M. S. K. Manikandan, "Path optimization of box-covering based routing to minimize average packet delay in software defined network," Peer-to-Peer Networking and Applications, vol. 13, no. 3, pp. 932-939, 2020. doi: 10.1007/s12083-019-00855-8.

[17] M. Ha, D. M. Tran, and Y. Shichkina, "Model of Message Transmission across Parallel Route Groups with Dynamic Alternation of These Groups in a Multichannel Steganographic System," Electronics, vol. 12, no. 19, pp. 24, 2023. doi: 10.3390/electronics12194155.

[18] T. D. Holmes, R. H. Rothman, and W. B. Zimmerman, "Graph Theory Applied to Plasma Chemical Reaction Engineering," Plasma Chemistry and Plasma Processing, vol. 41, no. 2, pp. 531-557, 2021. doi: 10.1007/s11090-021-10152-z.

[19] W. C. Hu, H. T. Wu, H. H. Cho, and F. H. Tseng,

"Optimal Route Planning System for Logistics Vehicles Based on Artificial Intelligence," Journal of Internet Technology, vol. 21, no. 3, pp. 757-764, 2020. doi: 10.3966/160792642020052103013.

[20] P. Joshi, A. S. Raghuvanshi, and S. Kumar, "An Intelligent delay efficient data aggregation scheduling for distributed sensor networks," Microprocessors and Microsystems, vol. 93, pp. 16, 2022. doi: 10.1016/j.micpro.2022.104608.

[21] C. H. Ke, Y. H. Tu, and Y. W. Ma, "A reinforcement learning approach for widest path routing in software-defined networks," Ict Express, vol. 9, no. 5, pp. 882-889, 2023. doi: 10.1016/j.icte.2022.10.007.

[22] N. Khakzad, "A methodology based on Dijkstra's algorithm and mathematical programming for optimal evacuation in process plants in the event of major tank fires," Reliability Engineering & System Safety, vol. 236, pp. 11, 2023. doi: 10.1016/j.ress.2023.109291.

[23] N. H. Kim, F. He, R. M. Nasir, and S. I. Kwak, "Stepwise benchmarking based on production function: Selecting path towards closest target," Expert Systems with Applications, vol. 228, pp. 11, 2023. doi: 10.1016/j.eswa.2023.120308.

[24] S. Kim, T. Choi, S. Kim, T. Kwon, T. H. Lee, and K. Lee, "Sequential graph-based routing algorithm for electrical harnesses, tubes, and hoses in a commercial vehicle," Journal of Intelligent Manufacturing, vol. 32, no. 4, pp. 917-933, 2021. doi: 10.1007/s10845-020-01596-9.

[25] P. Kumari and S. K. Sahana, "Heuristic Initialization Based Modified ACO (HIMACO) Mimicking Ant Safety Features for Multicast Routing and its Parameter Tuning," Microprocessors and Microsystems, vol. 93, pp. 9, 2022. doi: 10.1016/j.micpro.2022.104574.

[26] S. R. Lawande, G. Jasmine, J. Anbarasi, and L. I. Izhar, "A Systematic Review and Analysis of Intelligence-Based Pathfinding Algorithms in the Field of Video Games," Applied Sciences-Basel, vol. 12, no. 11, pp. 30, 2022. doi: 10.3390/app12115499.

[27] F. K. Luan, J. X. Yang, H. Zhang, Z. Y. Zhao, and L. P. Yuan, "Optimization of Load-Balancing Strategy by Self-Powered Sensor and Digital Twins in Software-Defined Networks," Ieee Sensors Journal, vol. 23, no. 18, pp. 20782-20793, 2023. doi: 10.1109/jsen.2022.3216376.

[28] S. Duleba, F. K. Gundogdu, and S. Moslem, "Interval-Valued Spherical Fuzzy Analytic Hierarchy Process Method to Evaluate Public Transportation Development," Informatica, vol. 32, no. 4, pp. 661-686, 2021. doi: 10.15388/21-infor451.

[29] D. Kalibatiene and J. Miliauskaite, "A Hybrid Systematic Review Approach on Complexity Issues in Data-Driven Fuzzy Inference Systems Development," Informatica, vol. 32, no. 1, pp. 85-118, 2021. doi: 10.15388/21-infor444.

[30] A. Ulutas et al., "Developing of a Novel Integrated MCDM MULTIMOOSRAL Approach for Supplier Selection," Informatica, vol. 32, no. 1, pp. 145-161, 2021. doi: 10.15388/21-infor445.