ITC 2/53

Information Technology
and Control
Vol. 53 / No. 2/ 2024
pp. 323-330
DOI 10.5755/j01.itc.53.2.35943

Optimization of Sewing Equipment Based on Improved Genetic-ant Colony Hybrid Algorithm

Received 2024/01/04	Accepted after revision 2024/02/07

HOW TO CITE: Rao, N., Jin, W., Yang, Y., Liao, Y., OuYang, L. (2024). Optimization of Sewing Equipment Based on Improved Genetic-ant Colony Hybrid Algorithm. *Information Technology and Control*, 53(2), 323-330. https://doi.org/10.5755/j01.itc.53.2.35943

Optimization of Sewing Equipment Based on Improved Genetic-ant Colony Hybrid Algorithm

Ning Rao, Wenbing Jin, Yuemei Yang

HangZhou Polytechnic, 311402, China

Yihui Liao

Quzhou Branch of China Mobile Communications Group Zhejiang Co., Ltd, 324000, China

Liangjing OuYang

JACK Sewing Machine Co., Ltd, 317700, China

Corresponding author: luckyrning@163.com

The optimization of the cutting path of the sample can effectively reduce the cutting time, thereby improving the production efficiency of numerical control processing. This paper comprehensively considers the impact of the cutting order and the position of the knife entry point on the cutting path, converts the cutting path problem into a type of traveling salesman problem (TSP), and proposes an improved genetic-particle swarm optimization algorithm. The selection mechanism of the algorithm combines the elitist retention strategy and roulette wheel selection method to accelerate the search for the optimal solution; the mutation strategy designs a linear decreasing mutation rate, which enhances the global search ability; at the same time, introduces the ant colony optimization algorithm to process the fitness function, adjusts the population evolution difference, and speeds up the optimization process. Through this hybrid algorithm, the cutting order of the sample can be quickly optimized, and the nearest neighbor algorithm is used to determine the position of the knife entry point. Tests are conducted on clothing patterning charts and standard examples. Compared with several commonly used algorithms, experimental results verify the feasibility and effectiveness of this algorithm.

KEYWORDS: sewing equipment, path optimization, genetic algorithm, traveling salesman problem, improved genetic-ant colony hybrid algorithm.

1. Introduction

Sewing equipment is a kind of numerical control system for cutting multi-contour flat samples. Compared with traditional manual cutting, it greatly improves the production efficiency and raw material utilization rate of the products. A cutting machine usually needs to process tens or even hundreds of samples. The total path of the cutting knife is composed of the effective cutting stroke and the empty cutting stroke between different samples. Among them, the length of the effective cutting stroke is the sum of the contour lengths of each sample, and its length is fixed; while the empty cutting stroke changes with the order of the samples and the position of the knife-in point. Therefore, the problem of cutting path optimization is to optimize the empty travel. By optimizing the empty travel and reducing the length of the empty travel, it will effectively shorten the knife-moving time and improve production efficiency.

In the whole processing process, the length of the cutting path of the Sewing equipment depends on the cutting order of the pieces and the position of the cutting point of each piece, which is a two-level optimization problem. Shi [12] proposed a path planning optimization of intelligent vehicle based on improved genetic and ant colony hybrid algorithm. Ayoade Akeem Owoade [1] proposed Efficient hybrid enhanced genetic algorithm and ant colony system model for rerouting multimedia message in multiple node–link failures within wireless network. Zulfa [15] proposed a hybrid cached data optimization based on the least used method improved using ant colony and genetic algorithms. Jaouachi [5] proposed assessment of jeans sewing thread consumption by applying metaheuristic optimization methods. Phan [10] proposed the optimal method of balancing the sewing Line with T-Shirt Product in the Garment Industry in Vietnam.

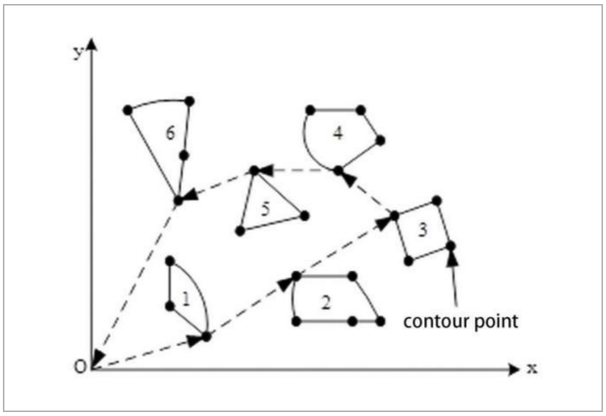
For the problem of optimizing clothing cutting paths, it can be regarded as a series of shortest path problems for "graphs", which can then be transformed into the GTSP for solution. Compared with holes, the outer contours of clothing samples have more corners and larger areas. The precise search of the target solution for the problem will inevitably lead to combinatorial explosion of the solution space, and traditional mathematical methods such as dynamic programming and enumeration are not suitable for solving this problem. In practical applications, approximation methods such as greedy and nearest neighbor algorithms have faster solving speeds, but the accuracy of the solutions is not high. To balance optimization effectiveness and algorithm running time, this paper proposes an improved genetic algorithm to solve the cutting path problem of cutting bed samples. Firstly, based on the initial knife entry point position, the cutting path problem is modeled as a type of traveling salesman problem, and the initial cutting order is generated using the selection, crossover, and mutation strategies of IGA. The selection strategy combines the elitist retention strategy and roulette wheel selection method, and the mutation strategy designs a linear decreasing mutation rate. Secondly, the nearest neighbor algorithm is used to obtain new knife entry points. The ant colony optimization algorithm is introduced to process the fitness function, adjust the population evolution difference, and speed up the optimization process. Finally, based on the new knife entry point position, a new cutting order is generated using IGA-ACO to obtain the final cutting path. The individuals constructed by IGA-ACO reflect the integer encoding, which can directly decode the cutting order of the sample. The update strategy effectively ensures the global search performance of the algorithm. The application example of clothing samples shows that the algorithm can effectively construct the cutting path of cutting bed samples. Compared with other related algorithms

2. Problem Description

The sample marker to be cut is composed of several independent samples, and the shape of the outer contour of the samples is irregular, including straight lines, arcs and curves. After the layout of the sample is determined, the feature points of the outer contour of each sample can be replaced with specific geometric information. When the sewing equipment is cutting, the cutter starts from the origin of the cutting machine, and the cutter selects a sample entry point. The entry point can be the vertex position of the outer contour of the sample, or the interpolation point position between two vertices, such as Shown as black dots on the swatch in Figure 1. After the cutter finishes cutting a sample, it returns to the entry point, thus completing the cutting of a sample. At this point, the entry point of the piece becomes the exit point, and the cutter moves from this point to another piece until the cutting of all pieces is completed, and finally returns to the origin of the cutting machine. The path shown by the dotted line in Figure 1 is a Complete empty travel path. In this paper, it is assumed that the total number of samples to be processed on a cutting machine is n, r represents the origin of the cutting machine, N_i represents the i sample, R_i represents the N_i number of corresponding contour points, R_{ij} represents the i contour point corresponding to the j sample, r_i represents the entry point of the i sample, $i = 1, 2, \cdots, n$, $j = 1, 2, \cdots, R_i$.

Figure 1

Schematic diagram of the outer contour clipping path of the sample



The total number of contour points calculated by the cutting entry point in the cutting path algorithm program is $\sum_{i=1}^{n} R_i$, so there are $n! \times \prod_{i=1}^{n} R_i$ cutting paths. As shown in Figure 1 there are $6! \times 3 \times 5 \times 4 \times 4 \times 3 \times 4 = 2073600$ cutting paths in the marker diagram. Even if the number of samples is small. As the number of samples increases, the number of entry points that need to be traversed will increase massively

Therefore, the problem is a typical NP-hard problem, that is, there is no algorithm that can guarantee an optimal solution in polynomial time. In practical applications, intelligent optimization algorithms or approximate algorithms are often used to convert them into problems that can be solved in polynomial time.

The cutting path problem of the cutting table can be simplified as the shortest path problem of a series of point sets, so that the shortest path problem can be modeled as a kind of traveling salesman problem [2-4, 6-9, 11, 13, 14]. Therefore, the cutting path problem of the cutting machine sample can be attributed to starting from the origin r of the cutting machine, passing through the entry point. $r_i (i = 1, 2, 3 \cdots, n)$ of each sample, and finally returning to the origin r of the cutting machine, the initial entry point of the sample can be given by the system or can be selected arbitrarily. Assuming that the cutting order of the samples to be processed is $S = \{N_1, N_2, \cdots, N_n\}$, the length of the empty travel path L traveled by the cutting machine can be expressed as the following:

$$L = d\left(r, r_1\right) + \sum_{i=1}^{n-1} d\left(r_i, r_{i+1}\right) + d\left(r_n, r\right). \quad (1)$$

Among them, $d(r, r_1)$ and $d(r_n, r)$ represent the distance from the origin of the cutting machine to the position of the first and last sample entry points respectively; $\sum_{i=1}^{n-1} d(r_i, r_{i+1})$ represents the total distance from the entry point i of the sample to the entry point of the sample $i+1$. The goal of this paper is to minimize the empty stroke length L and obtain the minimum empty stroke length $L_{\min}$:

$$L_{\min} = \min\left(L\right). \quad (1)$$

3. Improved Genetic-Ant Colony Hybrid Algorithm

In this paper, from the perspective of global optimality, the selection mechanism combines the elite retention strategy and the roulette selection method, and the mutation strategy designs a linearly decreasing mutation rate. Through these two steps, the genetic algorithm is easy to fall into the disadvantage of local optimality. Improvements are made to effectively ensure the global search performance of the algorithm and to find a better cutting order.

3.1. Individual Coding Design

The individuals are encoded using integer coding, the individual genes represent the serial number of each sample, the population number is m, and the samples to be cut are respectively represented by an integer number, then the individual code is represented as:

$$\left(v_1, v_2, \cdots, v_n\right), 1 \le i \le n, v_i \ne v_j. \quad (1)$$

Among them, v_i is the number of the i sample to be cut.

The population is initialized, and $(v_1, v_2, \cdots, v_n)$ is randomly arranged to generate m groups of different sequences. The fitness value i of the F_i individual is the reciprocal of the length L_i of the individual's clipping path, namely $F_i = 1/L_i$, obtained from Formula (1), $i = 1, 2, ..., n$.

3.2. Selection Mechanism

The selection mechanism in this paper combines the elite retention strategy and the roulette selection method, which not only avoids the destruction of individuals with high fitness in the current generation population due to crossover and mutation operations, but also increases the diversity of the population and improves the ability to find the global optimum. probability of solution.

The core idea of elite retention strategy is to copy the elite individuals that appear in the population process to the next generation. Before the genetic operation, the top 10% of the individuals with fitness in the current generation population are retained, and they do not participate in the crossover and mutation operations. The remaining 90% of the individuals are, and the roulette method is used to select two individuals.

The main content of the roulette selection method is that individuals with high fitness will have a higher probability of being selected. For the remaining individuals, the fitness of the individual is and the probability of being selected is calculated according to:

$$\rho_i = \frac{F_i}{\sum_{j=1}^{t} F_j}, i = 1, 2, \cdots, t. \quad (2)$$

Calculate the individual cumulative probability δ_i by ρ_i

$$\delta_i = \sum_{j=1}^{i} \rho_j, i = 1, 2, \cdots, t. \quad (3)$$

Roulette selection is performed according to δ_i, and the selection method is as follows: First, a random number ω in the range of [0,1] is generated, making the probability $p_i = \delta_i - \omega, i = 1, 2, \cdots, t$. Set the initial individual index $j = 1$, if $p_j < 0$, then $j = j + 1$, until $p_j \ge 0$, select the individual index H. According to this method, two different individual indexes are selected, and the two individuals corresponding to the indexes are selected.

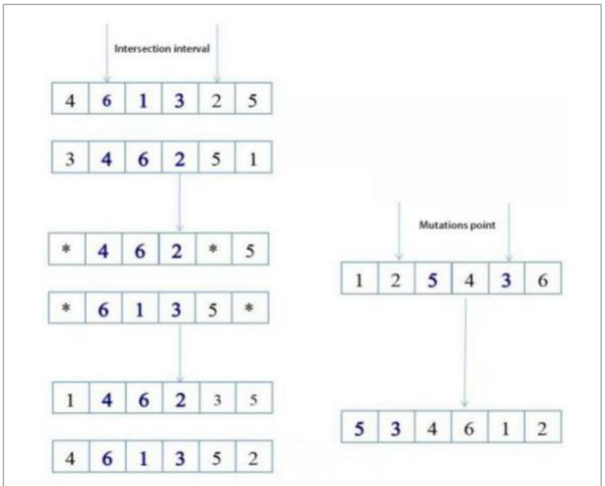
3.3. Crossover and Mutation Operators

The crossover rate p_c and mutation rate p_m will directly affect the convergence performance and optimization effect of the genetic algorithm. Not all selected individuals need to perform crossover operation, depending on the size of p_c, which determined whether crossover operation is required, p_c generally taking a decimal in the interval [0,1].

Whether an individual gene mutates depends on the

Figure 2

Schematic diagram of overlapping and mutation



(a) overlapping (b) Mutations

size of p_m. The algorithm degenerates into a random search process if p_m is too large; It is difficult to generate new individuals if p_m is too small. Therefore, a more feasible method is to take p_m a larger value at the initial stage and gradually reduce it to 0 as the search process progresses. Therefore, this paper adopts a linear decreasing mutation operator as follows:

$$p_m = p_{\max} - \frac{g^*\left(p_{\max} - p_{\min}\right)}{g_{\max}}. \quad (4)$$

Among them, p_{max}, p_{min} represent the maximum and minimum values of p_m, respectively, g represents the current number of iterations, and g_{max} represents the maximum number of iterations

3.4. Local Search

The ant colony algorithm is used for local search, imitating the foraging behavior of the ant colony. The solution obtained by the global search of the IGA algorithm is used as the initial solution of the ant colony algorithm, and local search is performed to improve the accuracy of the solution. The optimization steps of the ant colony algorithm are divided into the following parts:

Step 1: Place m ants on n samples, and randomly generate taboo tables for each ant;

Step 2: Each ant moves to the next piece according to the probability transition formula, and adds the piece to the taboo table of the ant; until each piece is traversed only once, record the shortest path;

Step 3: Update the pheromone according to the pheromone update rule; if the end condition is met, jump out of the loop and output the optimal solution, otherwise continue the search from step 2.

3.5. Stop Criteria

The search stops when the stopping criterion is reached during the algorithm cycle. The stopping criterion defined in this paper is as follows: the fitness value of the solution does not change significantly after multiple generations of continuous iterations or the current number of iterations exceeds the maximum number of iterations

3.6. Nearest Neighbor Algorithm

After using IGA-ACO to determine the cutting order of the samples, the nearest neighbor algorithm is used to determine the cutting entry point of each sample. The specific steps are as follows:

Step 1: Take the origin of the cutting machine r as the starting point of cutting, let $P_k = r$, the set of entry points $P = \{r\}$, and the sample serial number $i = 1$.

Step 2: In the cutting order S, find the contour point $r_i = R_{ij}$ that is closest to P_k in the contour point set $\{R_{i1}, R_{i2}, \cdots, R_{iR_i}\}$ corresponding to the i sample N_i, let $P_k = r_i$, $i = i + 1$, and add r_i to the set P.

Step 3: Determine whether all samples have been traversed, that is, whether exceeds. If it exceeds, add the origin of the cutting machine to the end position to obtain the set of entry points and end the search, otherwise, go to step 2 to continue the search.

Step 4: Determine whether all samples have been traversed, that is, whether exceeds. If it exceeds, add the origin of the sewing equipment to the end position to obtain the set of entry points and end the search, otherwise, go to step 2 to continue the search.

3.7. Algorithm Flow

In this paper, IGA-ACO combined with the nearest neighbor algorithm is used to solve the cutting path optimization problem, in which the initial entry point position is given or arbitrarily selected by the CAD software system, and the solution process is as follows:

Step 1: Initialize the parameters, let the current iteration number $g = 1$, and use integer coding to generate the initial population.

Step 2: Adopt the elite retention strategy to retain the top 10% individuals with fitness in the population.

Step 3: For the remaining 90% of the individuals, the roulette selection method is used to select two parent individuals each time, and according to the crossover rate and mutation rate, determine whether to perform crossover and mutation operations.

Step 4: Select the individual with the highest fitness value in the population according to the ant colony algorithm, update the cutting order $S = T$ and the number of iterations $g = g + 1$.

Step 5: Determine whether the stopping criterion is reached. If the stopping criterion is reached, the cutting order is obtained $S = \{N_1, N_2, \cdots, N_n\}$, and the search is ended; otherwise, go to step 2 to continue the search.

Step 6: Use the nearest neighbor algorithm to obtain the set of entry points $P = \{r, r_1, \cdots, r_n, r\}$.

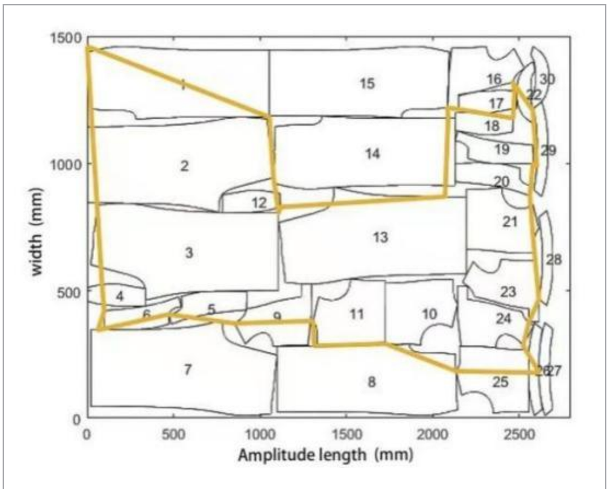
Step 7: Keep the cutting order obtained in step 5 as an elite individual, and use the position of the entry point obtained in step 6 to jump to step 1 to further search for the cutting order, and end at step 5 to obtain the final cutting order.

4. Simulation

To validate the effectiveness of the proposed algorithm in this article, a PC with Windows 7 operating system was used to conduct simulation verification using MATLAB 2016a software. Firstly, six pattern layouts generated by clothing CAD software were selected, and each layout was independently run five times. Figure 3 shows the optimized cutting path of clothing samples, with a total of 40 samples. The thick solid line in the figure represents the idle of the cutting knife. Table 1 summarizes the optimization results of the proposed algorithm on the six pattern layouts. It can be seen from the table that the algorithm achieves good optimization results, reducing the cutting time and material waste, and improving the efficiency of clothing production.

Figure 3

Optimization diagram of the cutting path of clothing samples



For the layout diagram in Figure 3, the algorithm adopted by the CAD software is to use the layout order as the cutting order and determine the position of the knife entry point through the nearest neighbor algorithm. As a result, the path length is shortened by 9381mm, and the optimization rate is 52.2%. Furthermore, the average running time is 15 seconds, and the accuracy and time are ideal, which verifies the algorithm's feasibility in this paper.

The time taken for the clipping path obtained by the CAD software is concise, usually within 1 minute, but the clipping effect is not ideal. It can be seen from Table 1 that the number of samples is from 30 to 144. The algorithm in this paper shows an optimization rate between 45% and 62%. The solution time is kept within 7 minutes when the number of samples is less than 100. The compromise between optimization accuracy and solution time can be achieved within three minutes, which verifies the algorithm's effectiveness in this paper.

To further evaluate the algorithm's performance in this paper, for solving the TSP model, IGA-ACO is compared with GA and the existing effective algorithms ACO and GNNA. The TSPLIB library. In each search process, the search is stopped when the stopping criterion is reached; at the same time, the time cost is considered, and the search is stopped after more than 5 minutes. Each calculation example uses each algorithm for five independent calculations. The shortest path length obtained by each algorithm is recorded in Table 2. The bold data represents the optimal results of the four algorithms in each calculation example, and represents the IGA-Deviation of the optimal solution of ACO from the best solution in several other algorithms.

It can be seen from Table 2 that in terms of accuracy, the optimal solution obtained by IGA-ACO is better than other algorithms in five of the six calculation examples. Compared with GA, the optimization effect of IGA-ACO is greatly improved. It can be seen from the result of deviation error that IGA-ACO has excellent

Table 1

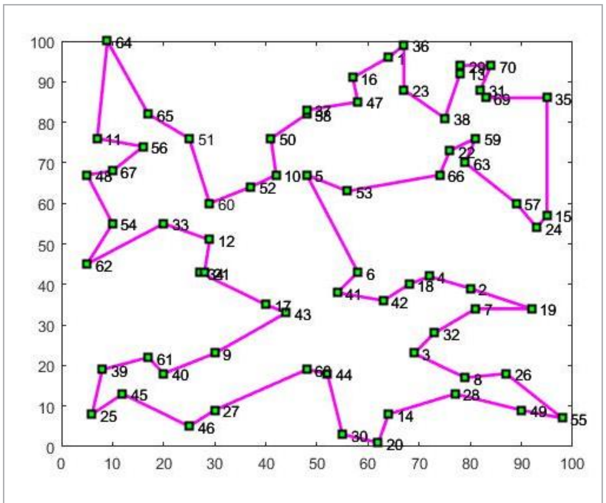
Optimization effect of the algorithm for different layout images

Sample	Number of samples	Before optimization	Optimized	Time (s)	Optimization rate
1	30	17973	8592	15	52.20%
2	40	19881	8148	42	59.00%
3	66	48876	18741	107	61.66%
4	72	33754	17304	118	48.73%
5	82	26860	13995	145	47.90%
6	144	74009	40303	395	45.54%

Table 2
Comparing the optimal results obtained by the algorithm with other algorithms

Examples	GA	ACO	G-NN	IGA-ACO	error
eil51	460.82	459.64	470.33	**450.22**	-2
berlin52	8848.5	7920.9	7959.568	**7894.81**	-0.3
st70	789.66	746.88	735.852	**709.16**	-5.1
eil76	659.3	581.11	611.418	**579.2**	-0.3
rat99	1788.4	**1299.03**	1309.027	1388.5	6.9
kraoA100	30134.4	23994.3	23438.084	23233.3	-3.2

Figure 4
Schematic diagram of st70 test results



solution performance. In addition, when the number of example data is relatively small, IGA-ACO has high solution accuracy, but as the number of nodes reaches a certain scale, the population diversity decreases, and the algorithm accuracy decreases. In terms of time, for six calculation examples, IGA-ACO can obtain a satisfactory solution to the problem within five minutes, so it is acceptable in terms of time. Figure 4 is a schematic diagram of the results of IGA-ACO in solving the calculation example st70.

5. Conclusion

In order to avoid the genetic algorithm from falling into local optimum prematurely, this paper improves the selection mechanism and mutation strategy of the genetic algorithm, introduces the ant colony algorithm for local search, obtains a better cutting order, and determines the position of the entry point with the nearest neighbor algorithm. The feasibility and effectiveness of the model and algorithm in this paper are demonstrated through actual clothing sample data and tests on standard problems. Although the model and algorithm in this paper are mainly based on clothing sample data, they can be extended to cutting path problems such as leather and mechanical parts in applications.

Declaration of Conflicting Interests

The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

Funding

This article was Supported by the Scientific Research Fund of Zhejiang Provincial Education Department, Grant number Y202145996; and the University Enterprise Cooperation Project for Domestic Visiting Engineers in Higher Education Institutions, Project name: Research on Key Technologies for Slice Transfer and Separation, Grant number: FG2021313.

References

1. Ayoade Akeem Owoade, Ogundile, O. O., Babalola, O. P., Balyan, V. Efficient Hybrid Enhanced Genetic Algorithm and Ant Colony System Model for Rerouting Multimedia Message in Multiple Node-Link Failures within Wireless Network. International Journal of Communication Systems, 2023, 36(11). https://doi.org/10.1002/dac.5507

2. Bogyrbayeva, A., Yoon, T., Ko, H., Lim, S., Yun, H., Kwon, C. A Deep Reinforcement Learning Approach for Solving the Traveling Salesman Problem with Drone. Transportation Research Part C: Emerging Technologies, 2023, 148, 103981. https://doi.org/10.1016/j.trc.2022.103981
3. Dell'Amico, M., Montemanni, R., Novellani, S. Exact Models for the Flying Sidekick Traveling Salesman Problem. International Transactions in Operational Research, 2021. https://doi.org/10.1111/itor.13030
4. Huerta, I. I., Neira, D. A., Ortega, D. A., Varas, V., Godoy, J., Asín-Achá, R. Improving the state-of-the-art in the Traveling Salesman Problem: An Anytime Automatic Algorithm Selection. Expert Systems with Applications, 2022, 187, 115948. https://doi.org/10.1016/j.eswa.2021.115948
5. Jaouachi, B., Khedher, F. Assessment of Jeans Sewing Thread Consumption by Applying Metaheuristic Optimization Methods. International Journal of Clothing Science and Technology. 2021. https://doi.org/10.1108/IJCST-01-2021-0005
6. Krishna, M. M., Panda, N., Majhi, S. K. Solving Traveling Salesman Problem using Hybridization of Rider Optimization and Spotted Hyena Optimization Algorithm. Expert Systems with Applications, 2021, 183, 115353. https://doi.org/10.1016/j.eswa.2021.115353
7. Mahmoudinazlou, S., Kwon, C. A Hybrid Genetic Algorithm for the Min-Max Multiple Traveling Salesman Problem. Computers & Operations Research, 2024, 162, 106455. https://doi.org/10.1016/j.cor.2023.106455
8. Mzili, T., Riffi, M. E., Mzili, I., Dhiman, G. A Novel Discrete Rat Swarm Optimization (DRSO) Algorithm for Solving the Traveling Salesman Problem. Decision Making: Applications in Management and Engineering, 2022, 5(2), 287-299. https://doi.org/10.31181/dmame0318062022m
9. Nurdiawan, O., Pratama, F. A., Kurnia, D. A., Kaslani, D. A., Ahaningsih, N. Optimization of Traveling Salesman Problem on Scheduling Tour Packages Using Genetic Algorithms. Journal of Physics: Conference Series, 2020, 1477, 052037. https://doi.org/10.1088/1742-6596/1477/5/052037
10. Phan, T. T., Le, T. M. A., Phan, D.-N., Tran, N. S. Researching the Optimal Method of Balancing the Sewing Line with T-Shirt Product in the Garment Industry in Vietnam. ECS Transactions, 2022, 107(1), 7869-7887. https://doi.org/10.1149/10701.7869ecst
11. Pop, P. C., Cosma, O., Sabo, C., Corina Pop Sitar. A Comprehensive Survey on the Generalized Traveling Salesman Problem. European Journal of Operational Research. 2023. https://doi.org/10.1016/j.ejor.2023.07.022
12. Shi, K., Huang, L., Du, J., Sun, Y., Tong, X., Xie, Y., Fang, Z. Path Planning Optimization of Intelligent Vehicle Based on Improved Genetic and Ant Colony Hybrid Algorithm. Frontiers in Bioengineering and Biotechnology, 2022, 10. https://doi.org/10.3389/fbioe.2022.905983
13. Shi, Y., Zhang, Y. The Neural Network Methods for Solving Traveling Salesman Problem. Procedia Computer Science, 2022, 199, 681-686. https://doi.org/10.1016/j.procs.2022.01.084
14. Zhang, Z., Han, Y. Discrete Sparrow Search Algorithm for Symmetric Traveling Salesman Problem. Applied Soft Computing, 2022, 118, 108469. https://doi.org/10.1016/j.asoc.2022.108469
15. Zulfa, M. I., Hartanto, R., Permanasari, A. E., Ali, W. LRU-GENACO: A Hybrid Cached Data Optimization Based on the Least Used Method Improved Using Ant Colony and Genetic Algorithms. Electronics, 2022, 11(19), 2978. https://doi.org/10.3390/electronics11192978