

Article

The Application of the SubChain Salp Swarm Algorithm in the Less-Than-Truckload Freight Matching Problem

Yibo Sun ¹, Lei Yue ^{2,3}, Yi Liu ^{2,3}, Weitong Chen ¹ and Zhe Sun ^{2,3,*}

¹ Faculty of Sciences, Engineering and Technology, The University of Adelaide, Adelaide, SA 5000, Australia; yibo.sun@adelaide.edu.au (Y.S.); weitong.chen@adelaide.edu.au (W.C.)

² Post Big Data Technology and Application Engineering Research Center of Jiangsu Province, Nanjing University of Posts and Telecommunications, Nanjing 210003, China; 1221097414@njupt.edu.cn (L.Y.); 1220096706@njupt.edu.cn (Y.L.)

³ Post Industry Technology Research and Development Center of the State Posts Bureau (Internet of Things Technology), Nanjing University of Posts and Telecommunications, Nanjing 210003, China

* Correspondence: zhesunny@njupt.edu.cn

Abstract: The less-than-truckload (LTL) freight problem is a general pain point in logistics applications. Its challenge resides in the fact that these loads cannot be shipped in a timely manner due to their relatively small volumes. Traditional LTL matching methods are challenged by delays in updating logistic information and higher distribution costs. In order to solve LTL challenges, we developed a novel SubChain Salp Swarm Algorithm (SSSA) by improving the traditional Salp Swarm Algorithm with the utilization of a SubChain operation. Our method aims to find the optimal strategy for maintaining a balance between lower operating costs and customer satisfaction. Our SSSA method combines multiple disconnected SubChain points to separate individual chains to find local optima and obtain better convergence results in the final decision. We have compared our method with mainstream metaheuristic algorithms using logistics datasets from a road freight company in Hangzhou, and the results demonstrate that our method converges faster than other methods and has a lower variance. Our method solves the limitation of local optima observed in other optimization methods and improves customer service in relation to the transportation issue.



Academic Editor: Jose María Alvarez Rodríguez

Received: 13 March 2025

Revised: 9 April 2025

Accepted: 14 April 2025

Published: 17 April 2025

Citation: Sun, Y.; Yue, L.; Liu, Y.; Chen, W.; Sun, Z. The Application of the SubChain Salp Swarm Algorithm in the Less-Than-Truckload Freight Matching Problem. *Appl. Sci.* **2025**, *15*, 4436. <https://doi.org/10.3390/app15084436>

Copyright: © 2025 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

Keywords: less-than-truckload freight matching; SubChain Salp Swarm Algorithm; SubChain operation

1. Introduction

Less-than-truckload (LTL) freight, also known as “less-than-truckload cargo”, refers to goods with smaller volumes and weights that cannot fill an entire truckload. Given the small volume and weight of LTL cargo, it must be transported with other goods to save capacity resources and reduce costs. The distribution center usually serves as the loading point for LTL cargo in local surrounding areas, and trucks wait for the cargo to arrive at the distribution centers. The trucks are dispatched only after the loaded LTL cargo satisfies requirements.

LTL freight matching refers to the process of pairing goods awaiting delivery with appropriate transport vehicles to optimize the delivery process in logistics applications. Cargo owners will provide information, such as cargo volume, weight, destination, and delivery time windows, while transportation vehicles are characterized by attributes such as capacity, load, and speed. In the process of vehicle-and-cargo matching, it is necessary to

consider the attributes of the cargo and the lorry to maximize cost savings while ensuring that the cargo is delivered to its destination. Due to the special nature of LTL logistics, LTL goods do not arrive at the distribution center at the same time, so in addition to matching trucks and goods based on their attributes, LTL truck-and-cargo matching also needs to consider the truck's departure time.

Numerous studies have been conducted on the LTL freight matching problem. V. Jaya Surya [1] and others proposed and developed a web application to facilitate freight transportation and secure information about the shipment. The application efficiently minimizes the costs to both the shipper and the freight driver or the transport company. Rakpong Kaewpuang [2] and others proposed inter-related optimization and game models that allow us to analyze vehicle routing optimization for LTL carriers and carrier selection for customers, respectively. Xiao Yang [3] and his colleagues proposed a two-phase method for addressing the hub location and routing problem (HLRP) in long-distance less-than-truckload (LTL) freight transportation networks. They validated the effectiveness of their approach using instances based on the AP dataset. Lai et al. [4] utilized a time-space-based model to construct a scheme for lowering shipping transportation costs related to the LTL issue. Lyu et al. [5] designed an iterative request exchange module, which provides carriers with choices to find the optimal way to proceed in different situations. Along with handling multiple bundles of requests, the profits of transportation can be significantly improved. Herszterg et al. [6] proposed a novel time-expanded network system for the less-than-truckload problem. With the employment of timely load plan arrangements, schedules are thoroughly considered in different situations without increasing the costs of transportation. Qiao et al. [7] designed an optimization model to find a cost-effective solution to the less-than-truckload issue, which found the key elements among decision-making solutions related to destinations. Similarly, Qiao et al. [8] raised an issue in solving dynamic less-than-truckload problems with the development of cost pricing models, and bidding strategies were considered to guide decisions on appointing carriers. Tang et al. [9] designed an online matching algorithm by utilizing a time window to solve the issue of small-scale trucks. The system is available to solve real-life LTL problems. Similarly, the LTL matching problem can also be resolved by building dynamic models, such as a Bayesian network model or a neural network [10,11], which describe the matching problem hierarchically.

Although the LTL freight matching problem has multiple angles for optimization, matching accuracy remains a huge issue in designing algorithms. The Salp Swarm Algorithm (SSA) [12], initially proposed by Saremi et al. in 2017 [13], was originally designed to imitate the behavior of salps, where a leader navigates, and others are followers, to solve optimization problems. The first member is selected as the leader to find optimal solutions, and followers update their positions based on the position of their leader. The SSA method is a direct and effective swarming strategy that avoids the issue of gradient descent with the paces of followers. Singh et al. [14] modified the strategies used in Salp Swarm Algorithms by integrating particle swarm optimization (PSO) and largely solved the problems of local optima and early convergence. Faris et al. [15] summarized the detailed theory and applications of the SSA in different scenarios; with the advantages of efficiency and collectiveness, the accuracy of results in decision-making cases can be greatly improved. Tubishat et al. [16] improved the general SSA by adding the updated positions of individual salps and designed a novel search algorithm strategy for local places, which improves accuracy. Salgotra et al. [17] modified the inner mechanism of the Salp Swarm Algorithm (SSA) by altering adaptive parameters and the iteration period to maintain the balance between discovery and development. The issues of early local convergence and linear degradation can be largely resolved by hybridizing different mechanisms from other optimization methods. Zivkovic et al. [18] explored the influencing factors

of data extraction and noisy disturbance. By combining the Salp Swarm Algorithm with K-nearest neighbor methods, the authors considerably improved the classification accuracy and the algorithm's performance in less-than-truckload problems. Tubishat et al. [19] designed a novel version of the Salp Swarm Algorithm (SSA) method, with the utilization of opposition-based learning to improve the diversity of the search space and resolve the issue of local convergence. Zhao et al. [20] utilized automation technology along with the Salp Swarm Algorithm (SSA) to overcome the issue of local optima in transportation and improve the robustness of the original model.

Our proposed algorithm aims to provide matching services for truck drivers and cargo shippers and offer route planning services for truck drivers shipping cargo to different destinations. This paper's contributions can be summarized as follows: (1) We improved the original Salp Swarm Algorithm (SSA) by introducing the SubChain operation, resulting in our proposed SubChain Salp Swarm Algorithm (SSSA). (2) We designed an LTL matching model for a multi-objective optimization problem and verified the model's performance on data from a real logistics company. The results show that it has improved convergence speed and overcomes the issue of local convergence. (3) We fully considered the booking mechanism that matches goods that have not yet arrived at the distribution center to smoothly handle unexpected conditions and matching issues.

The structure of this paper is as follows: Section 2 introduces the LTL freight matching system. Section 3 presents the standard SSA and the SubChain Salp Swarm Algorithm and demonstrates the effectiveness of the improvement through six validation functions. Section 4 explains the application of the SubChain Salp Swarm Algorithm to the planning of the LTL freight matching system and provides the results of simulation experiments. Section 5 is the conclusion of this paper.

2. Introduction of the Less-Than-Truckload (LTL) Freight Matching System

The main research problem in our paper is the LTL truckload matching problem in a distribution center, where the distribution center serves as the loading point for LTL cargo. The locations of the distribution center and the receiving point for LTL goods are known. There are several trucks in the distribution center, and the LTL cargo needs to arrive at the distribution center for loading and distribution to the target receiving point. The different LTL cargos do not necessarily arrive at the distribution center at the same time. The major factors that we focused on are system operating cost and customer satisfaction.

2.1. Problem Assumptions

The problem studied in this paper is the less-than-truckload (LTL) freight matching issue at distribution centers. It is assumed that cargos that have not been sent to distribution centers can be assigned to the same trucks as shipments that have already arrived at the distribution centers. Once all the cargo assigned to a particular truck has arrived at the distribution center and been loaded, the truck departs for delivery. The following assumptions are made in this study:

- The locations of distribution centers and delivery points are known, and the truck needs to start from the distribution center and travel to the place of receipt of goods to distribute the cargo.
- Only cargo with less volume and weight than those of the remaining trucks can be loaded onto the truck; otherwise, the effect of shape is not considered.
- The basic cost is calculated when the truck is dispatched, and then the fuel consumption cost is calculated based on the driving distance. The distribution cost for a truck is the sum of the departure cost and fuel expenses.

- Delivery time is calculated based on the average speed of the vehicle, without considering the impact of road conditions. Each cargo has a maximum available distribution time. If the time cost of delivering goods to a distribution center exceeds the maximum permitted delivery time, an overtime penalty will be incurred.

2.2. System Operating Cost

Our study mainly focuses on reducing transportation costs and improving user satisfaction in the context of logistic problems. The cost of LTL logistics is mainly determined by the result of truck–cargo matching. The results of LTL matching largely depend on differences in transportation issues and operation costs. The operation costs in our LTL logistics study are mainly basic fees, fuel costs, handling costs, and overtime penalties.

(1) Basic Fees

Typically, the basic fees of vehicle dispatch mostly comprise costs of depreciation, maintenance, and human resources, which remain stable and have no relation with vehicle distances. The basic fees for vehicle departure refer to the depreciation and labor costs incurred due to the departure of the vehicle regardless of the cost generated by other logistics equipment and labor, and it is calculated according to Formula (1).

$$Cost_B = \sum_{v \in V} \sum_{j \in P} R_{0j}^v BC_v \quad (1)$$

where v represents the set of trucks (vehicles); P represents distribution centers and LTL (less-than-truckload) cargo receiving points; BC_v represents the fixed cost of vehicle v for departure; and R_{ij}^v means that vehicle v travels from point i to point j , and its value is 1 or 0.

(2) Fuel Costs

The cost of fuel consumption is influenced by the distance driven by the vehicle from the distribution center to its destination points. The truck departs from the distribution center and visits each destination point to deliver goods in sequence based on the matching schedule, and the fuel costs are calculated based on the average driving speed of the vehicle. The calculation method is shown in Formula (2).

$$Cost_O = \sum_{v \in V} \sum_{i \in P} \sum_{j \in P} R_{ij}^v D_{ij} OC_v \quad (2)$$

where D_{ij} represents the distance between point i and point j , and OC_v represents the fuel cost per kilometer for vehicle v .

(3) Handling Costs

To facilitate transportation, there will be packaging costs associated with the cargo. During loading and unloading at the distribution center and pickup points, handling costs are incurred. These costs mainly consist of packaging materials and labor costs, calculated as shown in Formula (3).

$$Cost_L = \sum_{v \in V} \sum_{c \in C} M_{cv} HC_c \quad (3)$$

where C represents the collection of all cargo, and M_{cv} means that cargo c is transported by vehicle v , and its value is 1 or 0.

(4) Overtime Penalties

Because trucks need to wait at the distribution center until all the cargo they are supposed to transport has arrived, there might be delays, and it may be necessary to pay overtime penalties to the cargo destination. The calculation method for overtime penalties is shown in Formula (4).

$$Cost_B = \sum_{v \in V} \sum_{j \in P} R_{0j}^v BC_v \quad (4)$$

2.3. Customer Satisfaction Analysis

During the delivery service, it is important to not only pursue the lowest distribution cost but also strive for the highest customer satisfaction. Customer satisfaction consists of two dimensions: minimizing the time waiting for cargo at the distribution center for departure and ensuring that the cargo arrives at its destination as early as possible.

2.3.1. Calculation of Cargo Waiting Time Satisfaction

Cargo waiting time should be within the acceptable range defined by the consignor. If it exceeds this acceptable time, satisfaction continuously decreases. The waiting time satisfaction for distribution is determined by the average waiting time satisfaction for all cargo. The calculation method is represented by Formula (5).

$$Sat_w^c = \begin{cases} 100\%, (CT_c - AT_c) \leq WT \\ \left(1 - \frac{CT_c - AT_c - WT}{CT_c - AT_c}\right) \times 100\%, \\ (CT_c - AT_c) > WT \end{cases} \quad (5)$$

$$Sat_w = \frac{\sum_{c \in C} Sat_w^c}{Num_{cargo}}$$

where CT_c represents the starting time of the delivery for cargo c , AT_c represents the arrival time of cargo c at the distribution center, Num_{cargo} represents the quantity of cargo, and WT is the maximum waiting time.

2.3.2. Analysis of Cargo Arrival Time Satisfaction

Cargo should arrive within the acceptable time defined by the consignor. If it exceeds this acceptable time, satisfaction continuously decreases. The arrival time satisfaction for distribution is determined by the average arrival time satisfaction of all cargo. The calculation method is represented by Formula (6).

$$Sat_F^c = \begin{cases} 100\%, FT_c \leq MST_c \\ 1 - \left(\frac{FT_c - MST_c}{DT_c}\right) \times 100\%, FT_c > MST_c \end{cases} \quad (6)$$

$$Sat_F = \frac{\sum_{c \in C} Sat_F^c}{Num_{cargo}}$$

where MST_c represents the maximum acceptable delivery time for cargo c to satisfy the customer, DT_c represents the allowed maximum delivery time for cargo c , and FT_c represents the delivery duration for cargo c (time taken to complete the delivery).

2.4. Objective Function

Based on the analysis in Section 2.3, the objective function for the LTL (less-than-truckload) vehicle–cargo matching problem at the distribution center is determined by Formula (7).

$$\begin{aligned} \min Z_1 &= \omega_1 Cost_B + \omega_2 Cost_O \\ &\quad + \omega_3 Cost_L + \omega_4 Cost_F \\ \max Z_2 &= \theta_2 Sat_W + \theta_3 Sat_F \end{aligned} \quad (7)$$

where ω_1 represents the weight of basic fees for vehicle departure, ω_2 represents the weight of fuel costs, ω_3 represents the weight of cargo packaging and handling costs, ω_4 represents the weight of cargo overtime penalty costs, θ_2 represents the weight of cargo waiting time satisfaction, and θ_3 represents the weight of cargo arrival time satisfaction.

3. SubChain Salp Swarm Algorithm Description and Experiment

In order to solve the less-than-truckload (LTL) problem, we designed our optimization method based on the traditional Salp Swarm Algorithm (SSA). The SSA method has the advantages of a simple algorithm structure and relatively few parameters, and it has been widely used for transportation issues. However, the traditional SSA method often relies too much on local optima and is slow to converge. In this section, we show how we improve the SSA method by introducing the SubChain operation, which separates local convergence results individually to acquire the global optimal result more accurately. The mainstream processes of our proposed method are shown below.

3.1. Standard Salp Swarm Algorithm

The Salp Swarm Algorithm (SSA) was proposed in 2017 to solve the optimization problems. Based on the foraging behavior of salp swarms, the algorithm forms a chain of salps, with the first part of the chain being the leaders and the rest of the chain being the followers. The best position is considered the location of the food. The leading salps continuously move toward the food, while the following salps update their positions based on the positions of the salps in front of them.

The update formula for the positions of the salps in the leader group is as follows:

$$x_j^i = \begin{cases} F_j + c_1 \cdot ((ub_j - lb_j) \cdot c_2 + lb_j) & c_3 < 0.5 \\ F_j - c_1 \cdot ((ub_j - lb_j) \cdot c_2 + lb_j) & c_3 \geq 0.5 \end{cases} \quad (8)$$

In the formula, x_j^i represents the value of the j -th variable at the position of the i -th sea squirt (leadership set); F_j represents the value of the j -th variable at the position of the food; ub_j and lb_j represent the upper and lower limit values of the j -th variable, respectively; and c_2 and c_3 are random numbers between 0 and 1.

The calculation formula for c_1 is as follows:

$$c_1 = 2e^{-(\frac{4l}{L})^2} \quad (9)$$

where l represents the current iteration number, and L represents the maximum number of iterations.

The update formula for the positions of the follower salps is as follows:

$$x_j^k = \frac{x_j^k + x_j^{k-1}}{2} \quad k = 2 \sim N \quad (10)$$

where N represents the population size. Based on Formulas (8) and (10), the chain of salps can continuously approach the position of the food, ultimately obtaining the optimal solution to the optimization problem.

3.2. SubChain Salp Swarm Algorithm

Although the SSA (Salp Swarm Algorithm) has certain advantages in convergence speed, it tends to get stuck in local optima, due to its heavy reliance on the state of the best individual, and lacks the ability to escape from it. To address this issue, we introduce the SubChain operation to improve SSA.

The SubChain operation [21] refers to a process where, during individual updates, a certain individual is selected to be disconnected from the chain. The disconnected individual adopts a new updating method, while the individuals following the leader continue to update based on the original updating strategy. The key factors in the SubChain operation include the selection of the disconnecting point and the updating strategy for the disconnected individual.

The selection strategy is largely driven by the observation of individuals closer to the top of the population, which are more effective in learning fine-grained exploitation strategies, while those farther from the top remain stable. Therefore, SubChain disconnection is applied to lower-ranked individuals to diversify their search behaviors. The disconnection points are typically decided using a randomization strategy; utilizing the stochastic element increases the algorithm's ability to escape local optima and enhances its global search capability. A detailed description of our SubChain design is provided in the subsections below.

3.2.1. Selection Strategy for Exercise Points

Generally, individuals with higher fitness levels can follow the best individual for a detailed search, while the search performance of the next individuals is poorer. With continuous iterations, the proportion of individuals in the higher and lower parts will gradually increase. The disconnection points will be selected according to Formula (11).

$$\text{SubIndex} = \text{Rand}\left(\frac{N \times 4 \times 4^{\frac{l}{L}}}{5}, \frac{N \times 4^{\frac{l}{L}}}{5}\right) \quad (11)$$

SubIndex refers to the index of the individual in the sorted population array on which the SubChain operation is performed. Rand refers to a randomly selected integer value within the given range of two numbers.

3.2.2. Updating Strategy for the SubChain Individual

The disconnected individual is randomly selected according to one of three updating strategies. The first scenario involves updating based on the same strategy as the best individual. The second scenario involves moving in a random direction in space. The third scenario involves moving toward the food source and randomly offsetting to the left or right by a certain angle. The specific updating method is shown in Formula (12).

$$x_j^{\text{SubIndex}} = \begin{cases} \begin{cases} F_j + c_1 \cdot ((ub_j - lb_j) \cdot c_2 + lb_j) & c_3 < 0.5 \\ F_j - c_1 \cdot ((ub_j - lb_j) \cdot c_2 + lb_j) & c_3 \geq 0.5 \end{cases} & 0 \leq q \leq \frac{1}{3} \\ c_4 \cdot (ub_j - lb_j), \frac{1}{3} < q \leq \frac{2}{3} \\ x_j^i + c_5 \cdot \sqrt{\sum_{j=1}^D (x_j^i - F_j)^2} \cdot \left(\frac{F - x^i}{\sqrt{\sum_{j=1}^D (x_j^i - F_j)^2}} + -1^{c_6} * \text{rand}\left(\frac{\pi}{36}, \frac{\pi}{12}\right) \right), \frac{2}{3} < q \leq 1 \end{cases} \quad (12)$$

where c_4 is a random number between 0 and 1, c_5 is a random number between 0.5 and 1.5, c_6 is a random natural number, and Rand is a random value taken within the given boundary values.

3.3. Numerical Experiments

To verify the performance of the proposed SSSA (SubChain Salp Swarm Algorithm), we used six well-known benchmark functions. During the testing process, we compared the SSSA with the standard SSA, standard Genetic Algorithm (GA), particle swarm optimization (PSO), artificial hummingbird algorithm (AHA) [22,23], Grasshopper Optimization Algorithm (GOA) [13,24], Moth–Flame Optimization Algorithm (MFO) [25,26], and Whale Optimization Algorithm (WOA) [27,28].

For a fair comparison, we set the population size to 100, the dimension to 10, and the number of iterations to 300 for all algorithms. Each algorithm was run 50 times on each test function. The average of the 50 runs was taken as the final result to eliminate uncertainties during the search process. The final iteration results were compared with the number of iterations required to achieve the optimized results, and the results are shown in Table 1.

The iteration plots are presented in Figure 1. Since the SSSA can reach the optimal solution very quickly in many cases, we selected verification functions with optimization space to demonstrate the optimization performance.

Table 1. Averages and standard deviations of the algorithm using different functions.

Functions	SSSA		SSA		GA		PSO	
	Avg	Std	Avg	Std	Avg	Std	Avg	Std
Michalewicz	−7.27	0.69	−6.69	0.71	−3.61	0.53	−3.46	0.40
Shekel	−10.15	1.58	−7.60	3.50	−2.39	0.75	−1.54	0.66
Hartmann_6	−3.01	4.23×10^{-2}	−2.99	5.02×10^{-2}	−2.68	0.11	−2.83	5.55×10^{-2}
Trid	−194.72	25.53	−71.39	32.54	−36.03	38.03	−65.73	52.13
Beale	4.61×10^{-7}	2.88×10^{-6}	4.89×10^{-2}	0.20	3.31×10^{-3}	5.07×10^{-3}	6.70×10^{-3}	8.34×10^{-3}
Styblinski_Tang	−26.83	0.08	−24.85	1.68	−9.18	4.10	9.64	49.50

Functions	AHA		GOA		MFO		GWO	
	Avg	Std	Avg	Std	Avg	Std	Avg	Std
Michalewicz	−3.86	0.33	−6.29	1.20	−3.55	0.72	−3.81	0.54
Shekel	−2.07	0.42	−5.61	3.48	−8.70	2.97	−7.55	0.003
Hartmann_6	−2.74	0.14	−3.00	0.04	−3.00	0.04	−3.00	0.04
Trid	391.12	145.92	−200.2	0.94	−195.4	11.95	−151.9	62.87
Beale	0.02	0.01	0.03	0.39	5.59×10^{-9}	0.01	6.79×10^{-7}	0.01
Styblinski_Tang	7.75	36.55	−20.12	1.71×10^{-6}	−20.12	2.64×10^{-8}	−19.74	1.72

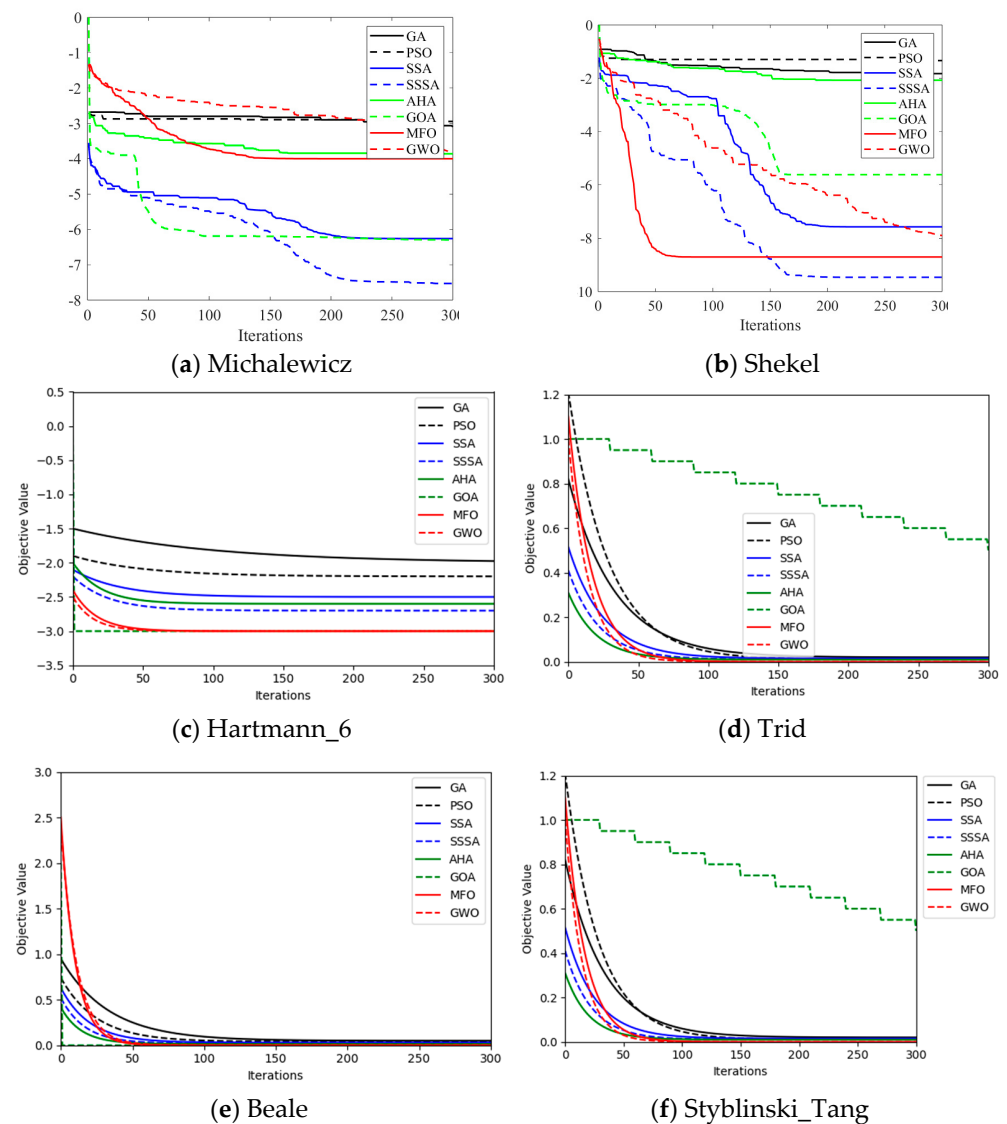


Figure 1. Results of algorithms with different functions.

From the table and figure, it can be observed that, for both functions where SSA performs well and those where it does not, the SSSA consistently achieves better results with faster convergence and smaller variance in the results. This validates the effectiveness of the improvement proposed in this paper.

4. The LTL Vehicle–Cargo Matching System Based on the SubChain Salp Swarm Algorithm: Design and Experiment

In this section, we illustrate the details of our improved SubChain Salp Swarm Algorithm, including the design of individual SubChains, which is the major modification in our proposed method, in Section 4.1, along with experimental settings and results in Sections 4.2 and 4.3. The results indicate that our method solves the issue of relying on local optima and has faster convergence.

4.1. Design of the Individual in the SSSA

Let the number of cargos be $\text{Num}_{\text{cargo}}$ and the number of trucks be $\text{Num}_{\text{truck}}$. In this study, the number of cargos is considered the dimension of the problem. The lower bound of the solution space is set to 1, and the upper bound is set to $\text{Num}_{\text{truck}} + 1$.

Initially, the algorithm generates a set of solutions within the solution space. Each solution consists of $\text{Num}_{\text{cargo}}$ values, where the i -th number represents the allocation plan for cargo i . Each number is split into an integer part and a fractional part. The integer part represents the truck number to which cargo i is assigned, and the fractional part represents the distribution weight of cargo i to the assigned truck's delivery destination.

When multiple cargos share the same delivery destination, the distribution weight for that destination is the sum of the fractional parts of these cargos' respective solutions. The trucks will prioritize visiting delivery destinations with higher distribution weights, indicating a higher preference for such destinations.

4.2. Data in the Experiment

The data used in this study were operational data from a certain highway freight transportation company. These data mainly contain indexes of the number of trucks, cargos selected for freight matching, and candidate features used for the dynamic scheduling phase. The details of these data are provided in Appendix A.

The weights applied in the objective function were determined based on the opinions of experts in the relevant field, and the specific numerical values are presented in Table 2. The related parameters for the algorithm are shown in Table 3.

Table 2. Parameters of proposed models.

Parameters	Value	Parameters	Value
ω_1	0.15	θ_2	0.366
ω_2	0.37	θ_3	0.634
ω_3	0.13	$\text{Num}_{\text{count}}$	40

Table 3. Parameters of experiment settings.

Parameters	Value	Parameters	Value
Iter_{max}	500	dim	40
limit low	1	popsize	30
limit up	21		

To maintain the accuracy and validity of the selected weights, it is necessary to consider the costs of packaging, loading, unloading goods, fuel consumption, and the overtime

penalty. The weights of these costs are added to the objective function to thoroughly calculate all factors in logistics. The detailed values are shown in the tables below.

4.3. Experimental Results and Analysis

To validate the effectiveness of the algorithm improvement, the SubChain Salp Swarm Algorithm, Genetic Algorithm, Particle Swarm Algorithm, and standard Salp Swarm Algorithm were applied to the model. Five experiments were conducted under the same conditions, and the average values were taken. The results are shown in Figure 2.



Figure 2. Results of LTL experiments.

As shown in Figure 2, a comparison of the solutions obtained from the four algorithms reveals that the proposed SubChain Salp Swarm Algorithm (SSSA) exhibits excellent performance. It achieves higher customer satisfaction while minimizing the total cost. The results indicate that the SSSA-based vehicle–cargo matching model can provide reasonable vehicle–cargo matching schemes, vehicle delivery sequences, and vehicle departure schedules. This ensures that undeparted cargo does not stay at the distribution center for extended periods and already dispatched cargo does not experience long delays. By saving distribution center costs, the model also guarantees the satisfaction of LTL consignors. The results of distribution matching are shown in Figure 3; after the truck's departure time and path are found, it will leave at the specified time and deliver the goods in turn according to the order of delivery points. The different matching paths are represented by different colors.

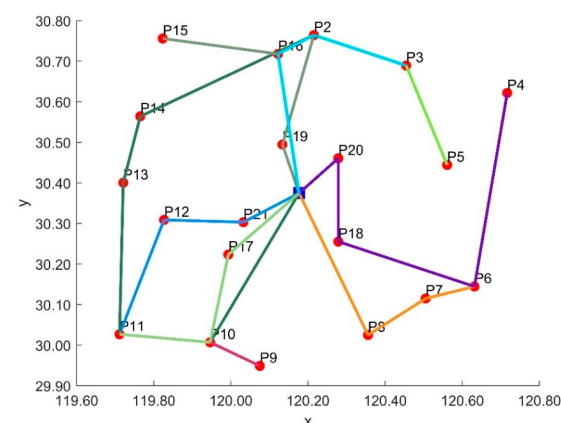


Figure 3. Matching paths of our algorithm.

The ideal matching results are represented in Figure 3 above, which represents the optimal solution in matching the distribution results of our proposed method. Our method can achieve high user satisfaction while keeping total costs to a minimum, which can be better optimized in comparison with other selected methods.

5. Discussion

The experimental results demonstrate that our proposed SubChain Salp Swarm Algorithm (SSSA) has the advantages of minimizing truckload costs while retaining the same level of customer satisfaction. Although the ratio of both waiting time and delivery time satisfaction is slightly lower than those of the other selected optimization methods (about 5%), transportation costs are greatly reduced (over 2 k degrees lower).

In contrast to traditional metaheuristic algorithms, our proposed method fully considers probable matching relationships between cargo and trucks, the sequences of receiving points, and the departure time of trucks. With the consideration of these basic conditions, our SSSA method is more robust when facing various unexpected conditions, and the matching strategy is more accurate and realistic. We have divided the operating costs into basic fees, fuel costs, handling costs, and overtime penalties, which constitute general costs in logistics applications.

In developing our method, we designed the SubChain mechanism with a selection strategy and an updating strategy. The design of the selection strategy utilizes a stochastic element to select the appropriate candidate for local selection and uses an updating strategy to change the locations of the disconnected individuals. By employing these SubChain operations, the connection between local and global optima is strengthened, and the reliance on local optima is greatly reduced. In this case, our proposed method is more applicable to solving the LTL issue.

Compared with other metaheuristic algorithms, our proposed method has advantages in dealing with timely issues to make better decisions regarding trucking loads, and utilizing our method in LTL problems can provide more accurate and robust suggestions for delivering cargo with limited trucks.

While our method has demonstrated its effectiveness on the selected datasets, it needs to be validated in practical applications in real-world logistics scenarios, as real-life logistics problems typically include dynamic and unpredictable issues such as traffic conditions, shipment delays, and fluctuating market demands, which are not fully captured by static simulated environments. In future research, we will need to pay more attention to actual freight routines with more complex real-time data inputs and evaluate the method's performance in dynamic situations, which will also help improve the scalability and robustness of our algorithm.

6. Conclusions

In order to solve the less-than-truckload (LTL) logistics problem, we designed a method based on the original Salp Swarm Algorithm with the utilization of SubChain operations. As LTL cargos have limited sizes and volumes, waste costs are incurred when delivering them with limited trucks. Our study fully considered factors such as waiting time satisfaction and delivering time satisfaction, and the model produces better strategies for matching trucks and cargo. With the utilization of the SubChain operation and the disconnection of individual SubChains, our method is less influenced by local optima. Experimental results illustrate that our method has faster convergence and saves transportation costs in the system matching problem, which has potential in real-life applications. However, the factors influencing the truckload, such as cargo shape, type, and temperature issues, have not been thoroughly considered and need to be accounted for in the future. Furthermore, real-life applications involve the monitoring status of drivers and cargo, so the algorithm should be tested in real-life scenarios to strengthen its managing and scheduling abilities. The SubChain operation may have non-trivial overhead when applied to real-time applications in high-frequency cycles, which should be considered in our future applications.

Author Contributions: Conceptualization, Y.S.; methodology, L.Y.; software, Y.L.; investigation, L.Y.; resources, Y.L.; writing—original draft preparation, Y.S.; visualization, W.C.; supervision, W.C.; project administration, Z.S.; funding acquisition, Z.S. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by Jiangsu Agriculture Science and Technology Innovation Fund (JASTIF) CX(22)1007.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data presented in this study are available on request from the corresponding author. The data are not publicly available due to privacy.

Conflicts of Interest: The authors declare no conflicts of interest.

Appendix A

Table A1. Truck data.

No.	Volume	Capacity	Speed	Basic Fee
1	54	12	63	141
2	47	11	65	114
3	53	12	73	154
4	49	15	65	183
5	52	13	75	112
6	49	14	66	128
7	49	12	69	104
8	47	13	64	105
9	4.6577	3.1337	24	07:30:07
10	14.0742	1.4090	20	10:37:25
11	10.2885	1.9472	22	09:41:36
12	5.1599	3.3614	20	10:25:18
13	11.9978	0.7438	12	10:59:45
14	2.8813	3.0847	27	07:59:37
15	9.1337	2.5067	20	08:35:48
16	1.0011	1.1360	17	08:16:08

Table A2. Cargo data.

No.	Volume	Weight	Packaging Cost	Arrival Time	Maximum Delivery Time	Target Delivery Destination	Overtime Penalty
1	10.3771	3.2584	13	08:50:42	23	3	50
2	5.3112	2.4297	22	11:42:57	23	13	27
3	3.4778	1.0340	17	08:57:27	22	11	37
4	4.8799	2.1599	16	09:27:29	5	10	20
5	10.4976	3.1631	18	07:28:22	5	19	34
6	10.7546	3.2597	30	08:01:52	22	15	42
7	5.0624	0.9055	21	11:28:59	13	5	19
8	11.2640	2.4548	22	07:01:19	23	6	21
9	4.6577	3.1337	24	07:30:07	5	6	33
10	14.0742	1.4090	20	10:37:25	15	5	36
11	10.2885	1.9472	22	09:41:36	18	15	31
12	5.1599	3.3614	20	10:25:18	24	10	32
13	11.9978	0.7438	12	10:59:45	20	6	30
14	2.8813	3.0847	27	07:59:37	22	4	49
15	9.1337	2.5067	20	08:35:48	24	19	17
16	1.0011	1.1360	17	08:16:08	12	5	32
17	13.6046	0.5339	26	07:00:16	15	5	16
18	4.2527	0.4867	20	08:20:02	17	3	31
19	3.9872	3.9430	28	09:43:56	11	16	42
20	1.3102	1.8544	23	09:32:33	12	8	49
21	14.1833	3.5235	19	09:04:17	11	5	24
22	13.8680	2.6240	29	07:37:52	9	10	21
23	8.9140	2.0293	24	07:23:40	11	14	48
24	8.2938	0.5900	29	09:41:20	23	18	14
25	3.4485	1.5086	16	10:14:09	6	13	25
26	10.8970	3.0812	30	08:17:04	18	11	17
27	2.2404	0.3642	18	10:43:26	5	11	42

Table A2. Cont.

No.	Volume	Weight	Packaging Cost	Arrival Time	Maximum Delivery Time	Target Delivery Destination	Overtime Penalty
28	11.9563	0.4347	25	10:37:00	21	20	38
29	14.8442	3.5172	27	11:13:30	12	4	29
30	14.2168	2.7373	15	10:53:42	21	2	34
31	9.1474	2.2252	17	10:37:09	6	7	13
32	11.1638	1.2771	22	07:05:46	20	12	38
33	2.7359	1.5288	15	09:39:25	9	7	33
34	9.9714	1.0141	29	07:10:39	8	18	22
35	8.5732	3.7996	25	09:35:02	13	15	35
36	12.0620	1.8673	18	11:31:15	16	17	36
37	10.6899	1.7073	18	10:23:08	13	9	37
38	4.2751	1.8803	27	08:32:28	14	6	38
39	0.8281	1.6879	16	07:18:05	11	7	39
40	13.7918	0.2964	23	07:13:38	14	10	40

References

1. Surya, V.J.; Muthurajkumar, S.; Jitiendran, K.; Ajithkumar, K.; Ibrahim, S.M. A Web Based Application for Optimization of Less than Truckload Problem. In Proceedings of the IEEE World Conference on Applied Intelligence and Computing (AIC), Sonbhadra, India, 17–19 June 2022; pp. 813–817.
2. Kaewpuang, R.; Sawadsitang, S.; Yu, H. Evolutionary Carrier Selection for Shared Truck Delivery Services. *IEEE Trans. Veh. Technol.* **2023**, *72*, 6778–6782. [\[CrossRef\]](#)
3. Yang, X.; Bian, Y.; Bostel, N.; Dejax, P. A Two-Phase Method for Solving the Hub Location and Routing Problem. In Proceedings of the IEEE International Conference on Cybernetics (Cybconf), Beijing, China, 5–7 July 2019; pp. 1–6.
4. Lai, M.; Cai, X.; Hall, N.G. Cost Allocation for Less-Than-Truckload Collaboration via Shipper Consortium. *Transp. Sci.* **2021**, *56*, 585–611. [\[CrossRef\]](#)
5. Lyu, X.; Chen, H.; Wang, N.; Yang, Z. A multi-round exchange mechanism for carrier collaboration in less than truckload transportation. *Transp. Res. Part E Logist. Transp. Rev.* **2019**, *129*, 38–59. [\[CrossRef\]](#)
6. Herszterg, I.; Ridouane, Y.; Boland, N.; Erera, A.; Savelsbergh, M. Near real-time loadplan adjustments for less-than-truckload carriers. *Eur. J. Oper. Res.* **2022**, *301*, 1021–1034. [\[CrossRef\]](#)
7. Qiao, B.; Pan, S.; Ballot, E. Revenue optimization for less-than-truckload carriers in the Physical Internet: Dynamic pricing and request selection. *Comput. Ind. Eng.* **2020**, *139*, 105563. [\[CrossRef\]](#)
8. Qiao, B.; Pan, S.; Ballot, E. Dynamic pricing model for less-than-truckload carriers in the Physical Internet. *J. Intell. Manuf.* **2019**, *30*, 2631–2643. [\[CrossRef\]](#)
9. Tang, W.; Chen, X.; Lang, M.; Li, S.; Liu, Y.; Li, W. Optimization of truck–cargo online matching for the less-than-truck-load logistics hub under real-time demand. *Mathematics* **2024**, *12*, 755. [\[CrossRef\]](#)
10. Shu, S.; Xu, Z.; Baldacci, R. Incorporating holding costs in continuous-time service network design: New model, relaxation, and exact algorithm. *Transp. Sci.* **2024**, *58*, 412–433. [\[CrossRef\]](#)
11. Sun, X.; Xing, D. Research on vehicle-cargo matching algorithm considering price heterogeneity. In Proceedings of the Fourth International Conference on Intelligent Traffic Systems and Smart City (ITSSC 2024), Xi'an, China, 23–25 August 2024; pp. 117–129.
12. Abualigah, L.; Shehab, M.; Alshinwan, M.; Alabool, H. Salp swarm algorithm: A comprehensive survey. *Neural Comput. Appl.* **2020**, *32*, 11195–11215. [\[CrossRef\]](#)
13. Saremi, S.; Mirjalili, S.; Lewis, A. Grasshopper optimisation algorithm: Theory and applications. *Adv. Eng. Softw.* **2017**, *105*, 50–67. [\[CrossRef\]](#)
14. Singh, N.; Singh, S.B.; Houssein, E.H. Hybridizing salp swarm algorithm with particle swarm optimization algorithm for recent optimization functions. *Evol. Intell.* **2022**, *15*, 23–56. [\[CrossRef\]](#)
15. Faris, H.; Mirjalili, S.; Aljarah, I.; Mafarja, M.; Heidari, A.A. Salp swarm algorithm: Theory, literature review, and application in extreme learning machines. In *Nature-Inspired Optimizers*; Springer: Berlin/Heidelberg, Germany, 2020; pp. 185–199. [\[CrossRef\]](#)
16. Tubishat, M.; Ja'afar, S.; Alswaitti, M.; Mirjalili, S.; Idris, N.; Ismail, M.A.; Omar, M.S. Dynamic salp swarm algorithm for feature selection. *Expert Syst. Appl.* **2021**, *164*, 113873. [\[CrossRef\]](#)
17. Salgotra, R.; Singh, U.; Singh, S.; Singh, G.; Mittal, N. Self-adaptive salp swarm algorithm for engineering optimization problems. *Appl. Math. Model.* **2021**, *89*, 188–207. [\[CrossRef\]](#)
18. Zivkovic, M.; Stoean, C.; Chhabra, A.; Budimirovic, N.; Petrovic, A.; Bacanin, N. Novel improved salp swarm algorithm: An application for feature selection. *Sensors* **2022**, *22*, 1711. [\[CrossRef\]](#) [\[PubMed\]](#)
19. Tubishat, M.; Idris, N.; Shuib, L.; Abushariah, M.A.; Mirjalili, S. Improved Salp Swarm Algorithm based on opposition based learning and novel local search algorithm for feature selection. *Expert Syst. Appl.* **2020**, *145*, 113122. [\[CrossRef\]](#)

20. Zhao, F.; Yang, S.; Liu, J.; Wu, B.; Hu, Y. Chaotic Salp Swarm Algorithm for Inverse Kinematics of a 5-DOF Drill Floor Robot. In Proceedings of the 5th International Conference on Intelligent Control, Measurement and Signal Processing (ICMSP), Chengdu, China, 19–21 May 2023; pp. 725–728.
21. Xu, P.; Li, B. Evolution design methodology of the metamorphic mechanisms inspired by gene deduction. *J. Mech. Des.* **2025**, *147*, 083302-1.
22. Mirjalili, S.; Mirjalili, S.M.; Lewis, A. Grey wolf optimizer. *Adv. Eng. Softw.* **2014**, *69*, 46–61. [[CrossRef](#)]
23. Hosseinzadeh, M.; Rahmani, A.M.; Husari, F.M.; Alsalam, O.M.; Marzougui, M.; Nguyen, G.N.; Lee, S.W. A survey of artificial hummingbird algorithm and its variants: Statistical analysis, performance evaluation, and structural reviewing. *Arch. Comput. Methods Eng.* **2025**, *32*, 269–310. [[CrossRef](#)]
24. Ikeagwuani, C.C.; Nnonyelu, C.J.; Oti, M.C.; Ude, E.J.; Alexander, T.C. A posteriori constrained bio-inspired algorithm for enhancing strength and resilient modulus of soft subgrade soil. *Road Mater. Pavement Des.* **2025**, 1–32. [[CrossRef](#)]
25. Mirjalili, S. Moth-flame optimization algorithm: A novel nature-inspired heuristic paradigm. *Knowl.-Based Syst.* **2015**, *89*, 228–249. [[CrossRef](#)]
26. Sahoo, S.K.; Premkumar, M.; Saha, A.K.; Houssein, E.H.; Wanjari, S.; Emam, M.M. Multi-objective quasi-reflection learning and weight strategy-based moth flame optimization algorithm. *Neural Comput. Appl.* **2024**, *36*, 4229–4261. [[CrossRef](#)]
27. Mirjalili, S.; Lewis, A. The whale optimization algorithm. *Adv. Eng. Softw.* **2016**, *95*, 51–67. [[CrossRef](#)]
28. Su, Y.; Liu, Y. A novel marine predator whale optimization algorithm for global numerical optimization. *Eng. Optim.* **2025**, 1–37. [[CrossRef](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.