Hindawi
Scientific Programming
Volume 2021, Article ID 8667945, 11 pages
https://doi.org/10.1155/2021/8667945

Research Article

Simulation and Modeling Algorithm for Terminal Container Handling Intelligent Management Based on Internet of Things and Big Data Technology

Lifen Zhang,[1,2] **Qiang Zhou**,[1] **and Huifang Ding**[1]

[1]*School of Transportation and Logistics Engineering, Wuhan University of Technology, Wuhan 430063, China*
[2]*Hubei Communications Technical College, Wuhan 430079, China*

Correspondence should be addressed to Lifen Zhang; zhanglifen@whut.edu.cn

Received 26 September 2021; Revised 18 October 2021; Accepted 23 October 2021; Published 9 November 2021

Academic Editor: Hai Dong

In order to improve the effect of intelligent terminal container management, this paper improves the Internet of Things and big data technology, analyzes the RFID middleware architecture based on the actual needs of container handling management, and proposes a new method of RFID middleware load balancing. Moreover, this paper combines the Internet of Things technology and big data technology to analyze the terminal container loading and unloading process and build a corresponding intelligent system. After constructing a terminal container handling intelligent management system based on the Internet of Things and big data technology, the performance of the system is verified, and multiple sets of simulation data are used to conduct research. The experimental research results show that the terminal container handling management system based on the Internet of Things and big data constructed in this paper basically meets the actual needs of use.

1. Introduction

With the rapid development of China's terminal container transportation industry, major port group companies are facing unprecedented challenges. At present, most domestic ports still adopt manual or semimechanical methods to complete the collection and recording of container data information, which is inefficient and slow [1]. Customs, national inspection, and other government supervision departments also use image processing pattern recognition technology to identify the number of containers. This processing method only has a recognition rate of 80% to 98%, the container traffic management and tracking monitoring are all in an isolated state, and it is impossible to realize the efficient and automated production of terminal containers. Moreover, it is also a heavy and slow fatigue job for government workers [2]. Improving the operational efficiency of terminal containers is directly related to the economic benefits of various enterprises. The use of manual implementation to identify and count the vehicle numbers and container numbers of terminal containers is likely to cause duplication of work, human errors, time delays, stagnation of goods, and low efficiency. Therefore, in order to avoid the waste of container resources in the terminal, it is an urgent issue to improve the automatic management capabilities of terminal containers and modern management facilities [3].

The main transportation method of the former port logistics is container transportation. With the continuous growth of port transportation, resource competition among ports has become more intense. The competitive advantage of each port comes from hard conditions such as port geography and hinterland economy and its management. The service level, operation and production efficiency, equipment and facilities conditions, and other automation, informationization, modernization, and technical indicators are also extremely important factors that reflect the competitiveness of the port. Since the 1990s, with the increasing requirements for port operation efficiency, production costs, and sustainable development, port automation has become a new era of port development due to its significant advantages such as efficient operation, low labor costs, and superior safety and reliability Since the 1990s, with the increasing requirements for port

operation efficiency, production costs, and sustainable development, port automation has become the general trend of the new era of port development by virtue of its significant advantages such as efficient operation, low labor cost, safety, and reliability. Up to now, more than 30 automated terminals have been applied and put into actual production in many countries around the world and have achieved good operational results. The emergence and in-depth development of automated terminals has become a major change in the sustainable development of ports in the future.

This paper combines the Internet of Things technology and big data technology to analyze the terminal container loading and unloading process and build a corresponding intelligent system to provide a reference for subsequent intelligent container loading and unloading and management.

2. Related Work

For a long time, domestic and foreign scholars have carried out many researches on the optimization of container ship loading. Aiming at the problem of container loading, literature [4] first established a 0–1 planning model based on the optimization goals of the smallest amount of container tipping in the yard and the shortest quay crane moving distance. However, due to the large number of variables in the model 0–1, it is difficult to solve the problem when the scale is large and the practicability is not strong. Aiming at the problem that the problem model is difficult to solve, literature [5] simplified the above planning model by ignoring the ship weight stability constraint. The optimization goal is also set to minimize the amount of container dumping in the yard, and the number of variables is reduced compared with the previous one. Literature [6] took the minimum amount of container dumping in the yard and the stability of ship weight as the optimization goals, established a multiobjective integer programming model, and used the weighting method to obtain the noninferior solution of the problem model. On the basis of the previous research on container dumping in the storage yard, literature [7] first proposed a simple method for estimating the amount of container dumping in the container yard in combination with the operation of the container yard. It is defined as the basic amount of container dumping, which simplifies the problem of loading and arranging containers. In the following research, this method will be used as the research basis to analyze the container transportation situation in the terminal in detail. Literature [8] proposed that when analyzing and studying the distribution and arrangement of containers in the yard of ships, factors such as the weight of the container and the distance of the destination port should be taken into consideration. Literature [9] summarized the main methods currently used to study the automatic stowage of container ships and used mathematical modeling methods to optimize the research progress of ship stowage from the perspective of the ship and the port. Literature [10] described in detail the main characteristics of container ship stowage and unilaterally optimized ship stowage based on the linear programming method from the perspective of ship weight stability. Literature [11] proposed three strategies for carrying containers for container yard operations, used the integer programming method to establish a mathematical model, and used the CPLEX solver to accurately solve small-scale cases. The above-mentioned research on the problem of container loading and packing is mainly based on the mathematical model method to solve the problem. After that, the researchers used heuristic algorithms based on rule setting to further study the problem of container loading and packing. Based on the in-depth analysis of the causes of the container tipping problem in the container yard, literature [12] established a rule-based heuristic algorithm for the container loading and arranging problem based on the known container yard storage status and the container ship stowage map, which takes minimizing the amount of box turnover as the optimization goal. The results show that the case-solving effect is good.

Literature [13] additionally considers the actual constraints of port customs clearance. Moreover, on this basis, it constructed a heuristic algorithm with the optimization objective of minimizing the amount of container dumping based on the known yard container storage status and container ship stowage map. In addition, it designed numerical experiments to verify the effectiveness and practicability of the algorithm. In recent years, the emergence of intelligent optimization algorithms such as genetic algorithm, tabu search algorithm, simulated annealing algorithm, and particle swarm algorithm has provided new ideas and methods for solving optimization problems. Literature [14] compared the container stowage problem with the traveling salesman problem. On this basis, it considered the constraint conditions of ship stability, heel, and trim moment and established a problem model with the optimization goal of minimizing the amount of tank tipping. Moreover, it used a genetic algorithm to solve the problem. However, there are deficiencies; that is, the algorithm code is too long, the solution space is too large, and so on.

Literature [15] constructed an optimization algorithm based on a parallel tabu search to find the optimal ship loading sequence based on the known container ship stowage map. Literature [16] used a genetic algorithm to solve the problem with the minimum amount of box dumping as the optimization objective. Literature [17] regards the influencing factors such as the volume of container dumps, ship stability, strength, and operability as evaluation strategies and also uses genetic algorithms to find the optimal ship loading sequence. Literature [18] combined the actual loading and unloading of the container terminal to analyze the main reasons for the tipping of containers during container loading in the yard and put forward related control methods based on the given ship loading plan and yard. For the storage status of containers, a mathematical model with the smallest amount of unloaded containers as the optimization objective is established, and the genetic algorithm is used to solve them under different unloading rules. Literature [19] regards the container turnover rate of the yard as a constraint, takes the weight stability of the ship after loading as the optimization objective, establishes an integer programming problem model for solving the container loading sequence, and uses an algorithm based on

particle swarm optimization to design and solve it. Literature [20] established a model based on the given container loading and stowage map, taking the minimum amount of unloading as the optimization objective, and proposed a parallel genetic particle swarm algorithm to solve it. The simulation experiment was carried out through Matlab, and the solution effect was better. Most of the above researches are based on known ship loading and loading plans from a single perspective of container loading sequence. Literature [21] considered the relationship between container loading sequence and ship stowage, based on the idea of two-stage hierarchical solution, and designed an algorithm based on SWO-HES two-stage optimization to solve the small-scale problem of a single shell position of a ship.

3. Internet of Things and Big Data Processing Algorithms Applied to Container Unloading Management

This paper analyzes the architecture of RFID middleware and proposes a new method of RFID middleware load balancing. This method is based on ALE (Application Level Events). This method defines the workload on the RFID middleware as ALE's ECSpecs, and ALE's ECSpecs are sent from the RFID application. Therefore, the workload of RFID middleware is to migrate ECSpecs from middleware with higher load to middleware with lower load by managing components.

ALE provides a set of flexible interfaces with standard functions, which are realized through RFID middleware. Therefore, ALE defines a set of standard interfaces with aggregation, filtering, and counting functions.

Figure 1 shows the role of ALE in an RFID system. An RFID application layer requests the RFID middleware to collect and transmit EPCs information through the ALE interface. When requesting EPCs information, the RFID application layer specifies EPCs related to ECSpec. The three fields of EPCs are actually sent to the RFID application layer. The reading field specifies the EPCs information collected from the reader. The boundary section specifies the time stamp of the collection and the report generated. The ECSpec report specifies a filter that removes EPCs that are not related to the RFID application layer.

The ALE interface uses the logical name for the meaning reader. Logical reader is an abstract name. This abstract name refers to one or multiple physical readers that can be regarded as a group. For example, there are three RFID readers on a door carrying a lock in a warehouse. It is very simple and convenient to compile these three physical readers into a group of logical readers. The RFID middleware maintains the mapping information between the logical reader and the physical reader. Figure 2 is a class diagram of the relationship between the concepts mentioned, shows the dependencies between the various components, and also shows the architectural relationship in ALE.

The load of the RFID middleware mentioned here is mainly caused by the collection and filtering of the tag data read by the connection with the reader. A large number of

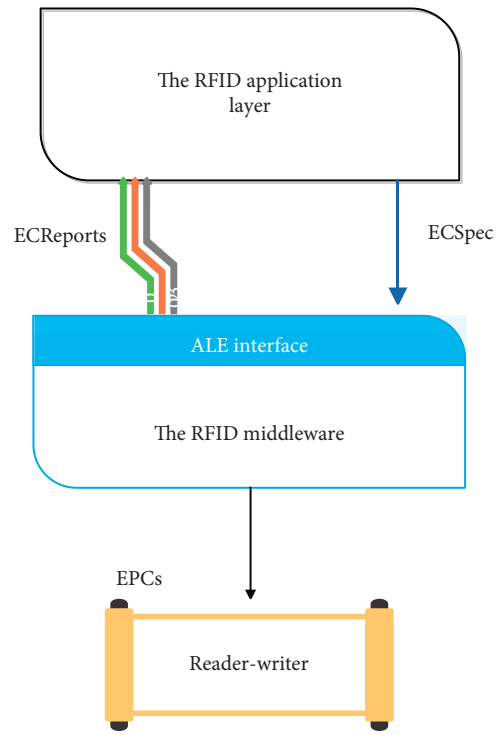

Figure 1: RID middleware architecture diagram with label data flow.

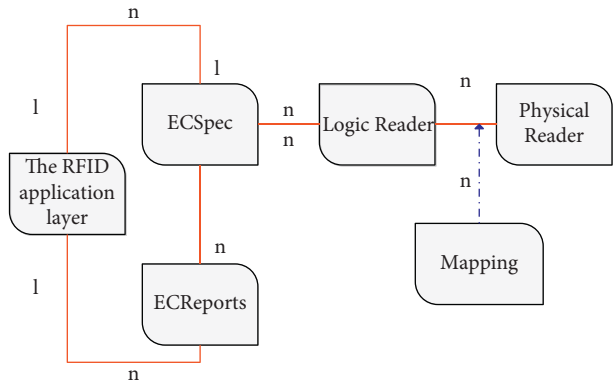

Figure 2: Basic concept diagram in ALE.

data tags have to be processed so as to cause the load of the middleware. Therefore, in order to reduce the load of the middleware, we need to effectively manage the large amount of raw label data processed by the middleware. As discussed in the previous section, the ECSpec sent by the RFID application layer defines a set of readers and tag data that is meaningful to the RFID application layer. In other words, an ECSpec determines a set of readers and notes data information processed by the RFID middleware. Therefore, we can manage the load status of the RFID middleware by controlling the ECSpec assigned to the middleware.

For the method of achieving load balancing of RFID middleware by migration, Figures 3(a) and 3(b) can be used to explain the overall process of this method. Figure 3(a) shows two application servers, and an RFID application is running in each application server. There are a total of six

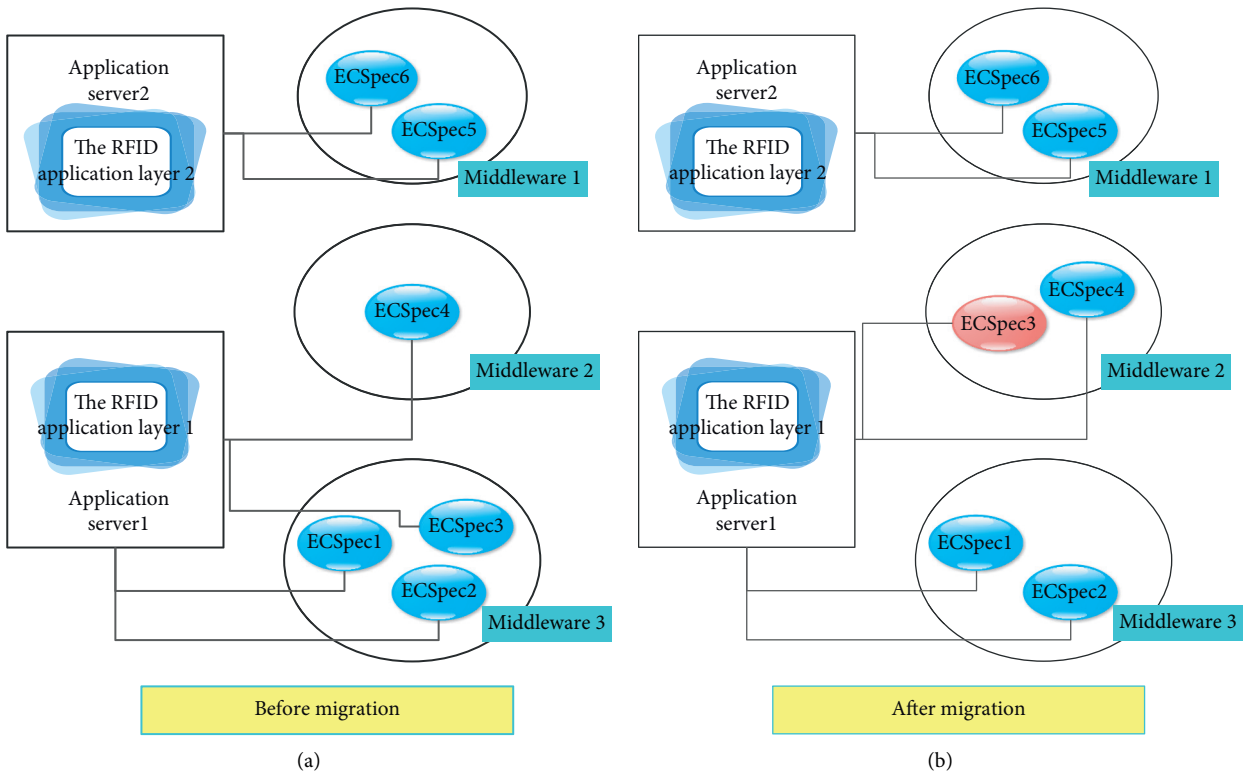

Figure 3: Example diagram of the migration algorithm of ECSpec for RFID middleware load balancing. (a) Before migration. (b) After migration.

ECSpecs distributed among three middleware servers. The connection between the RFID application server and ECSpecs represents the process of ECSpecs distributed by the RFID application.

Suppose we find a large amount of ECSpecs label data, ECSpec1, ECSpec2, and ECSpec3 will cause middleware 1 to overload. In order to reduce the load on middleware 1, we need to move ECSpecs from middleware 1 to other types of low-load middleware. For example, by moving ECSpec3 on middleware 1 to middleware 2 with the lowest load, the load on middleware 1 is reduced, so that middleware 1 is not overloaded. Since ECSpecs directly determines the label data that a large amount of middleware will process, the load status of the entire system caused by ECSpec3 is now balanced by migrating from middleware 1 to middleware 2. This situation is shown in Figure 3(b). The algorithm described above only occurs when the processing capabilities of the middleware components are the same. In the figure, middleware 1, middleware 2, and middleware 3 all have the same ability to process label data.

It is worth noting that the migration of ECSpec3 did not lead to changes in the distribution of RFID applications and the use of ECSpec3. The RFID application does not even notice that the originally allocated ECSpec3 has migrated from one middleware to another. This situation is determined by the properties of ECSpec itself. As described in the previous section, each ECSpec specifies the destination location and this result is transferred by the ECSpec through the Notification component of the URI. Therefore, the result of ECSpec is even still routing migration from the original location to the destination. Therefore, we can safely migrate the ECSpec of the overloaded middleware to the lightest-loaded middleware without worrying about the distribution of RFID applications.

As mentioned above, we balance the system load by migrating the same ECSpec from the overweight middleware to the overweight middleware. In addition, because RFID applications have no dependencies on certain RFID middleware, migration can occur without considering RFID-related applications. Therefore, when we choose to migrate ECSpec, we can ignore the related RFID applications.

Compared with the independence of RFID applications, RFID readers that are closely related to ECSpec can also be considered for migration and distribution of ECSpec. As we discussed in the previous section, each ECSpec has a specification on the logical reader/writer that can read meaningful ECSpec. In other words, ECSpec specifies a specific logical reader/writer that can read tag data. Since the RFID reader is usually connected to an RFID middleware, the reader designated by ECSpec can also be reassigned to another middleware to which ECSpec will move. For example, in Figure 4 showing ECSpec, the configuration of an RFID system and the relationship between the RFID reader and ECSpec are considered. Suppose that we choose to move ECSpec3 to middleware 2; in this case, the related readers R4 and R5 should be moved to middleware 2. At the same time, ECSpec3 will also be migrated to middleware 2. This is because ECSpec3 must be connected to R4 and R5, and each reader can only be connected to one RFID middleware.

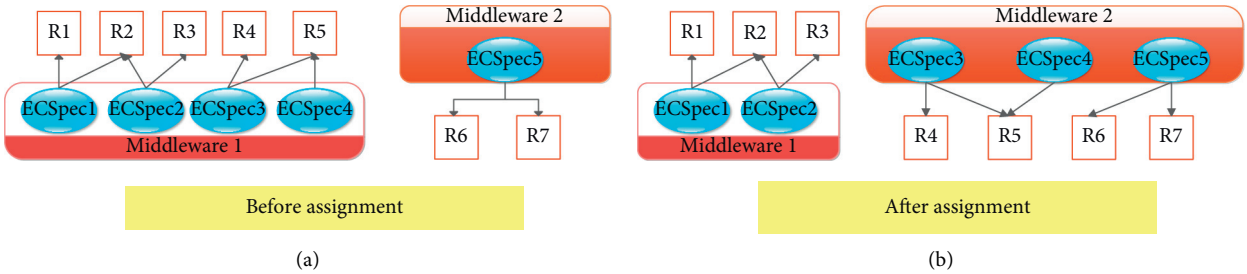

FIGURE 4: Distribution diagram considering the dependency of the reader. (a) Before assignment. (b) After assignment.

Dynamic weighted load algorithm distribution ECSpecs is feasible. According to the above migration and aggregation methods, it can be described as a distributed function to show the intuitiveness of the algorithm. The distribution of RFID middleware ECSpecs can be identified by comparing and calculating distributed functions. The formula is as follows:

$$Y(i) = L(i) \times e_i + R(i) \times e_r + M(i) \times e_m. \quad (1)$$

Among them, $L_{\max}(i)$ represents the maximum and affordable number of effective connections of the reader/writer connected to the i-th RFID middleware. $L(i)$ represents the load status of the i-th RFID middleware server. This can be expressed as follows:

$$L(i) = \frac{L_W(i) \times e_w + L_n(i) \times e_n}{L_{\max}(i)} + L_m(i) \times e_c. \quad (2)$$

$R(i)$ represents the degree of correlation between the i-th RFID middleware and ECSpecs, which is expressed as follows:

$$R(i) = R_W(i) \times e_W + R_n(i) \times e_n. \quad (3)$$

$M(i)$ represents the number of readers that need to be migrated for the i-th RFID middleware, which is expressed as follows:

$$M(i) = M_W(i) \times e_W + M_m(i) \times e_m. \quad (4)$$

$L_W(i)$ represents the number of readers running on the i-th RFID middleware server. $L_n(i)$ represents the number of nonoperational readers on the i-th RFID middleware server. $L_m(i)$ represents the load of the machine on the i-th RFID middleware server. This is expressed as follows:

$$L_m(i) = L_{\text{cpu}}(i) \times e_{\text{cpu}} + L_{\text{mem}}(i) \times e_{\text{mem}}. \quad (5)$$

$L_{\text{cpu}}(i)$ represents the CPU utilization of the i-th RFID middleware server. $L_{\text{mem}}(i)$ represents the memory utilization of the i-th RFID middleware server. Similarly, $R_W(i)$, $R_n(i)$, $M_W(i)$, and $M_n(i)$ are also expressed in the above method. Moreover, $e_i + e_r + e_m = 1$ and $e_i, e_r, e_m, e_w, e_n, e_c, e_{cpu}, e_{mem}$ are corresponding weights.

In order to describe the load status of each node, reader, and middleware cluster, a convenient calculation method is defined, which is expressed as follows:

$$T = \frac{\sum L(i)}{\sum L_{\max}(i)}, \qquad T(i) = \frac{L(i)}{L_{\max}(i)}. \quad (6)$$

In view of a clear understanding of the load balancing under the set dynamic weights, we must first define the load of the three-state nodes of $T(i)$. (A) Light load: this refers to a computing node with a small number of connections and a low machine load of the reader. (B) Heavy load: this refers to a computing node with a large number of connections and a high machine load of the reader. (C) Moderate balance: this refers to a state between heavy load and light load. This state is $|T(i) - T_s| < \Theta$, where T_s is the average and Θ is the deviation. The corresponding balance states $L_{\text{cpu}}(i)$ and $L_{\text{mem}}(i)$ of the utilization rate of the RFID middleware server are similarly expressed as described above. Moreover, when T changes, e_i: e_r: e_m will also change. When T is a light load, then e_i will take a larger value. When T is a heavy load, e_i and e_m will take larger values. In the same way, when $L_{\text{cpu}}(i)$ and $L_{\text{mem}}(i)$ change, e_{cpu} and e_{mem} also change. When $L_{\text{cpu}}(i)$ and $L_{\text{mem}}(i)$ are in a light load state, e_{cpu} and e_{mem} will take smaller values together, and vice versa. In addition, $L_m(i)$ is proportional to e_c. When $T(i)$ changes, e_w: e_n will change accordingly. When $T(i)$ exhibits a heavy load, e_w will take a larger value, and vice versa.

Figure 5 shows the relationship between the components of RLBA. The components MLT and RLT are middleware.

Middleware Load Table (MLT) and Reader Load Table (RLT) provide component management. The IMLTManager interface of MLT and the IRLTManager interface of RLT will perform update and retrieval in each middleware. Therefore, other components can access them by using the corresponding interface.

All nodes begin to perform calculation work. After a period of implementation, the node starts to check whether it is a heavy load node. If the node is overloaded, it will try to distribute its own jobs in the relevant domain. If it is assumed that the heavily loaded node is Lp, there are M nodes in the relevant domain, and these nodes are $L_1, L_2, \ldots\ldots, L_m$. Therefore, the average load is

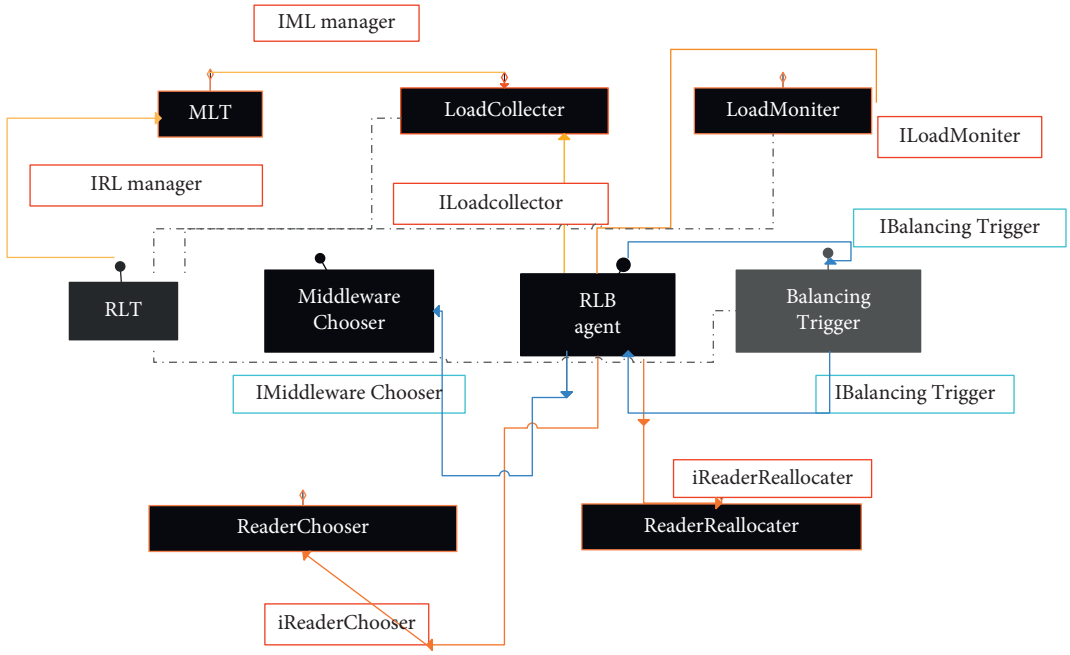

Figure 5: Proxy FD load balancing component diagram.

$$L_{avg} = \frac{1}{m+1}\left(L_p + \sum_{i=1}^{M} L_i\right). \tag{7}$$

In order to achieve uniform distribution, the load can be obtained as Nm by transmitting the overloaded load node to other nodes in the relevant domain. First, a hm value is set. This value is set to move all load nodes to the heaviest load node in the heavy load-related domain. If there is $L_{avg} > L_m$, the expression of N_m is

$$N_m + \left[\left(L_p - L_{avg}\right)\frac{h_m}{\sum_{i=1}^{M} h_i}\right]. \tag{8}$$

Then, the task can be sent to all relevant nodes at this time.

The start strategy at the receiving end is to use a light load start strategy. This strategy requires all other nodes to send tasks to this light node, and other nodes also use this strategy to send tasks to nodes other than the specified node like this light node. This strategy must first set an M value. This M value has the same effect as the M value of the sender startup strategy above, and it is also a critical value for distinguishing between heavy load and light load. The definition of the relevant domain here is the same as the above definition, and the same is true for all nodes to start performing calculation tasks. As time goes by, once a node detects that it is a light node, it also tries to distribute the load to itself in the relevant domain. The formula at this time is the same as formula (7), so that the value of Lavg can be calculated.

In order to achieve uniform distribution, the load can be obtained as N_m by transmitting the overloaded load node to other nodes in the relevant domain. The N_m formula listed here is the same as formula (8), except that the condition here is $L_{avg} < L_m$.

Figure 6 is the architecture diagram of the agent-based RFID middleware. In this figure, the JADE platform is based on JAVA. In the past few years, mobile agent system technology has become an exciting new field of computer science. There are a large number of methods, toolkits, and platforms of different qualities and maturities in this field, including Grasshopper, IBM Aglet, and JADE (Java Agent Development Framework).

Each instance running in the JADE runtime environment is called a container, and this container contains multiple agents. A group of running containers is called a platform. Figure 7 shows the architecture of the JADE platform.

Partial view: once the system starts, this special main container must always be executed on a platform, and through this special container, other containers can be registered. In addition to being able to accept registrations from other containers, the main container is different from other containers in that the main container is loaded with two special agents. AMS (Agent Management System) that can provide naming services (e.g., make sure that each agent on the platform has a unique name) is a representative authoritative platform. DF (Directory Facilitator is a directory service provider) provides a yellow pages service in a specific way. In this way, an agent finds other agents that can meet the needs of the agent in order to obtain the target.

Because the mobile agent's RFID middleware load architecture is built on the agent's RFID middleware architecture, but the mobile agent has a few more functional components that are not available in the agent, and the others are developed based on the agent architecture. Moreover, mobile agents and agents are built on the RLBA architecture, and both can and must run on JADE to some extent. Figure 8 is the architecture class diagram shared by mobile agents and agents.

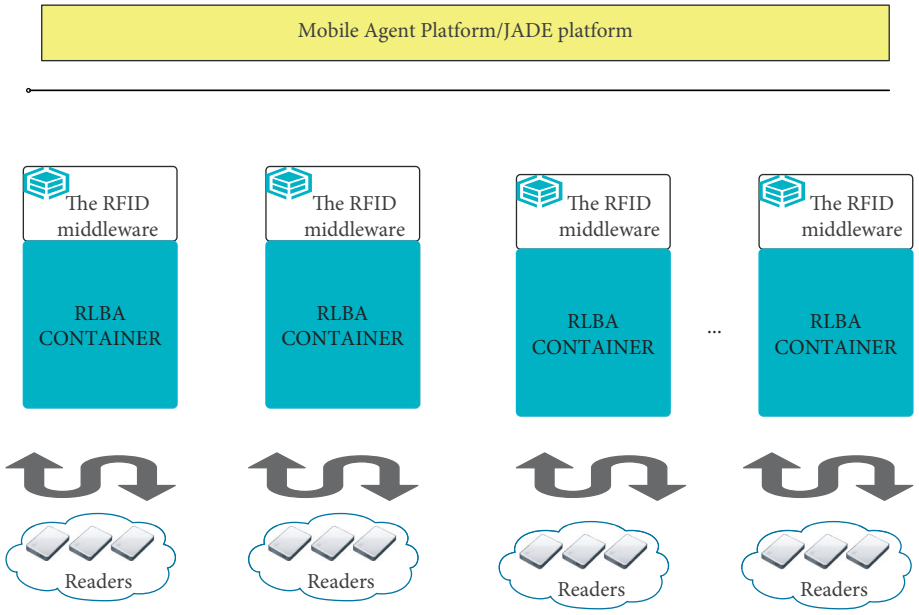

Figure 6: Architecture diagram of agent-based RFID middleware.

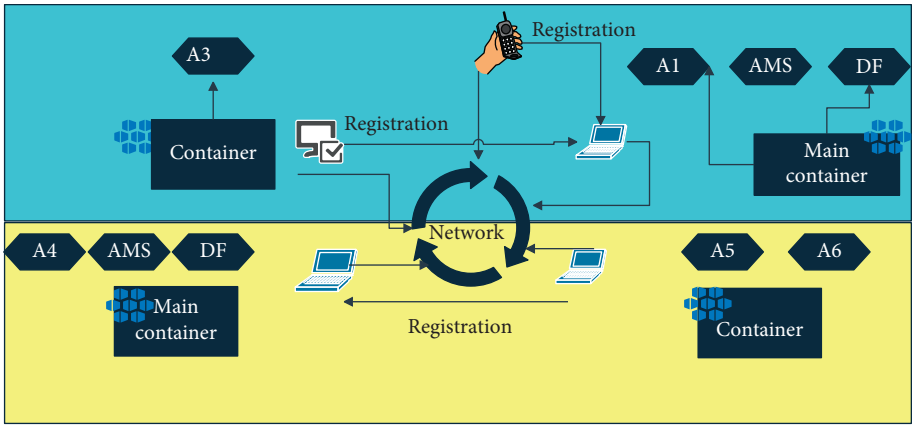

Figure 7: Architecture diagram of JADE platform.

Figure 8 is a class diagram of RLBA, which explains how RLBA and its components perform RFID middleware load balancing. JADE platform provides RLBA extended agent class. The agent class is used to initialize the runtime environment of the agent. GLT and RLT can be used to support RFID middleware load balancing decisions. RLBA is also composed of five components: load monitor, balance trigger, middleware selector, reader/writer distributor, and reader/writer selector. Another important class is NodeCriteria, which provides the definition of the equilibrium metric name, maximum threshold, minimum threshold, and initial type.

4. Intelligent Terminal Container Management System Based on Internet of Things and Big Data Technology

Business research and demand analysis cannot be successful once we need to continue to refine and research. This requires an excellent program developer to have a comprehensive understanding of the overall process of the investigated business and continue to refine it from the shallower to the deeper. We not only need to understand the relationship between terminal companies and various port logistics links but also need to understand the independent work responsibilities of each business department, each job position, and the relationship between each position. Figure 9 shows the business relationship between container terminals and various port logistics links.

Import unloading business refers to the process of unloading containers loaded on arriving ships from the ship to the dock site within a predetermined plan. The specific unloading process mainly involves several aspects shown in Figure 10.

After constructing a terminal container handling intelligent management system based on the Internet of Things and big data technology through the above analysis, the performance of the system is verified. This paper uses

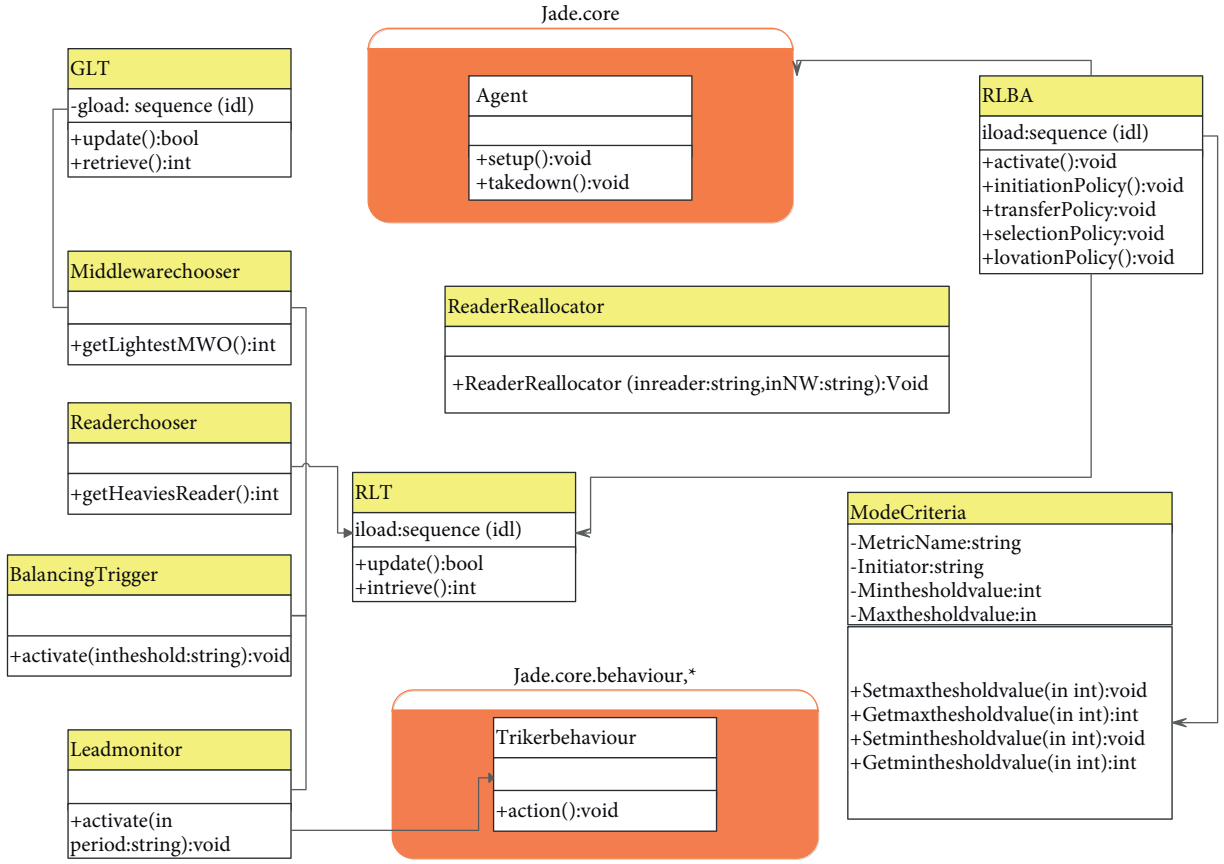

Figure 8: RLBA class diagram.

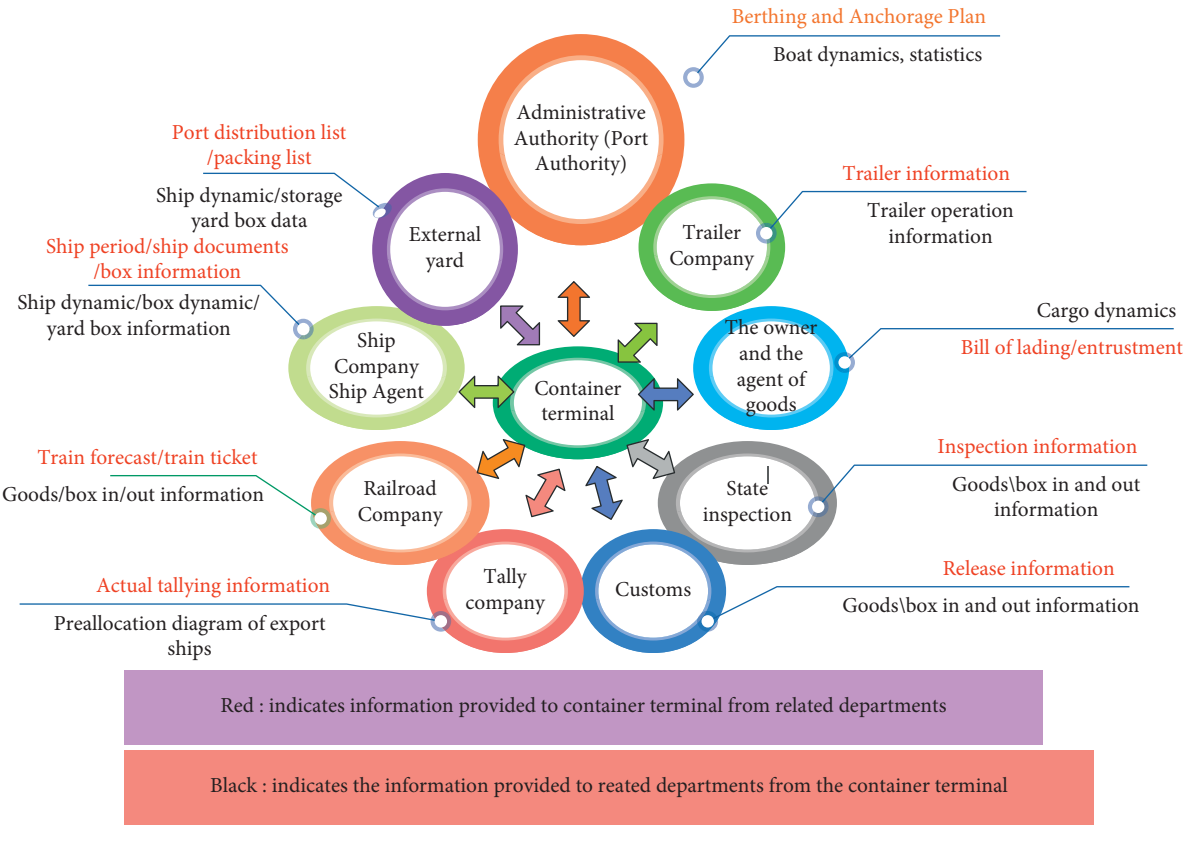

Figure 9: Business relationship between container terminals and various port logistics links.

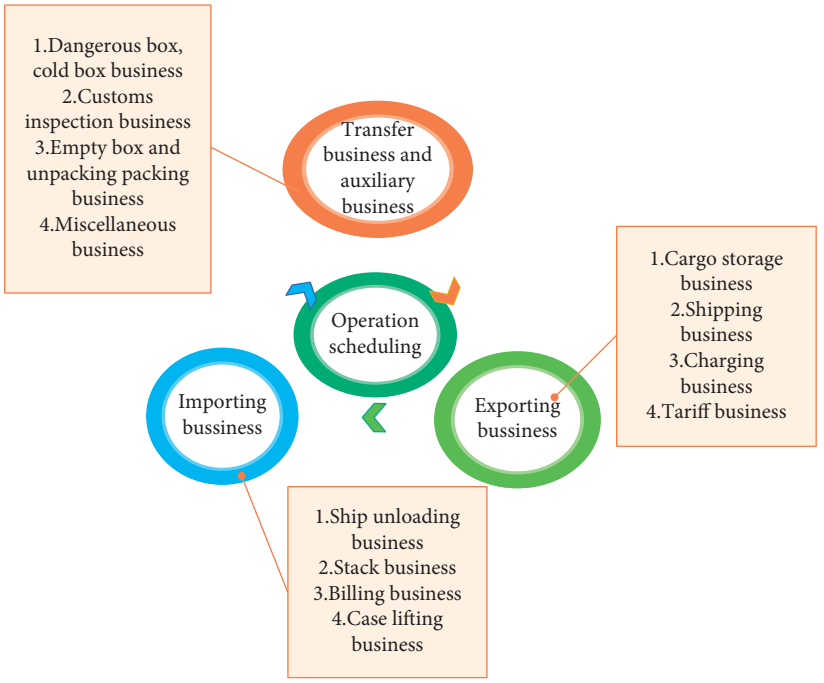

Figure 10: Division of container terminal business.

Table 1: Simulation effect verification of terminal container handling intelligent management system.

Number	Scheduling effect	Loading effect	Management effect
1	79.32	94.37	86.07
2	88.09	90.55	94.23
3	83.52	88.57	89.96
4	89.94	86.76	91.42
5	93.21	87.85	94.97
6	81.70	86.92	96.65
7	93.50	80.08	92.69
8	83.65	83.17	94.01
9	87.49	95.19	90.63
10	83.00	93.90	96.18
11	83.13	87.30	90.08
12	91.18	91.31	95.86
13	86.39	87.87	88.26
14	82.31	89.56	90.98
15	83.73	86.49	95.34
16	83.24	85.93	92.15
17	80.28	95.35	86.47
18	87.41	92.69	94.28
19	80.02	88.56	95.42
20	90.17	88.25	91.05
21	90.76	81.25	94.12
22	80.30	95.53	82.55
23	88.12	86.00	90.58
24	83.59	90.44	88.70
25	92.49	86.31	85.30
26	93.77	95.39	84.77
27	86.69	93.42	86.19
28	83.70	91.09	95.39
29	93.03	88.53	90.34
30	79.59	88.58	84.93
31	80.90	88.54	85.70
32	82.12	88.24	89.50
33	79.14	81.69	91.93

Table 1: Continued.

Number	Scheduling effect	Loading effect	Management effect
34	89.06	86.00	95.81
35	91.95	90.77	93.56
36	79.98	89.92	86.09
37	88.85	84.69	90.04
38	92.35	91.76	86.75

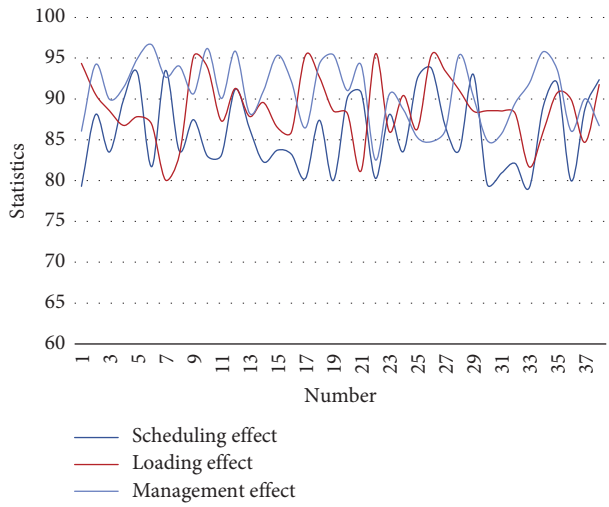

Figure 11: The performance verification results of the terminal container handling management system based on the Internet of Things and big data.

multiple sets of simulation data to count the container dispatching effect, loading effect, and management effect, and the test results shown in Table 1 and Figure 11 are obtained.

From the above research, it can be seen that the terminal container handling management system based on the Internet of Things and big data constructed in this paper basically meets the expected needs, and the system can be used for auxiliary management in subsequent practice.

5. Conclusion

Terminal container handling is a key part of the daily production activities of the port. The overall economic benefits of a container port are closely related to the loading quality of container ships in the port. The loading quality of ships directly affects the ship's port time, maritime navigation safety, and the efficiency of terminal loading and unloading operations, which in turn affects the port's reputation status and operating costs. The current large-scale development of container ships makes the loading and unloading tasks of ships in ports more arduous and the loading process more complicated. How to use the opportunity of the booming development of automated container terminals to integrate informatization, intelligent technology, and advanced management methods to rationally carry out container loading and dispatch, improve terminal service levels and ship transportation efficiency, and improve the competitiveness of the port itself and the transportation efficiency of the shipping industry is an urgent problem to be solved. This paper analyzes the loading and unloading process of containers in the terminal by combining the Internet of Things technology and big data technology and constructs the corresponding intelligent system. The experimental research results show that the terminal container handling management system based on the Internet of Things and big data constructed in this paper basically meets the expected needs, and the system can be used for auxiliary management in subsequent practice.

Data Availability

The labeled dataset used to support the findings of this study is available from the corresponding author upon request.

Conflicts of Interest

The authors declare no conflicts of interest.

Acknowledgments

This study was sponsored by the Wuhan University of Technology.

References

[1] L. Özdamar and M. A. Ertem, "Models, solutions and enabling technologies in humanitarian logistics," *European Journal of Operational Research*, vol. 244, no. 1, pp. 55–65, 2015.

[2] M. Savelsbergh and T. Van Woensel, "50th anniversary invited article-city logistics: challenges and opportunities," *Transportation Science*, vol. 50, no. 2, pp. 579–590, 2016.

[3] S. Tofighi, S. A. Torabi, and S. A. Mansouri, "Humanitarian logistics network design under mixed uncertainty," *European Journal of Operational Research*, vol. 250, no. 1, pp. 239–250, 2016.

[4] D. Cattaruzza, N. Absi, D. Feillet, and J. González-Feliu, "Vehicle routing problems for city logistics," *EURO Journal on Transportation and Logistics*, vol. 6, no. 1, pp. 51–79, 2017.

[5] J. Grabara, M. Kolcun, and S. Kot, "The role of information systems in transport logistics," *International Journal of Educational Research*, vol. 2, no. 2, pp. 1–8, 2014.

[6] E. Taniguchi, R. G. Thompson, and T. Yamada, "Recent trends and innovations in modelling city logistics," *Procedia - Social and Behavioral Sciences*, vol. 125, no. 2014, pp. 4–14, 2014.

[7] J. G. Carlsson and S. Song, "Coordinated logistics with a truck and a drone," *Management Science*, vol. 64, no. 9, pp. 4052–4069, 2018.

[8] R. Puertas, L. Martí, and L. García, "Logistics performance and export competitiveness: European experience," *Empirica*, vol. 41, no. 3, pp. 467–480, 2014.

[9] T. Qu, S. P. Lei, Z. Z. Wang, D. X. Nie, X. Chen, and G. Q. Huang, "IoT-based real-time production logistics synchronization system under smart cloud manufacturing," *International Journal of Advanced Manufacturing Technology*, vol. 84, no. 1-4, pp. 147–164, 2016.

[10] C. Chua, M. Danyluk, D. Cowen, and L. Khalili, "Introduction: turbulent circulation: building a critical engagement with logistics," *Environment and Planning D: Society and Space*, vol. 36, no. 4, pp. 617–629, 2018.

[11] R. Y. Zhong, S. Lan, C. Xu, Q. Dai, and G. Q. Huang, "Visualization of RFID-enabled shopfloor logistics big data in cloud manufacturing," *International Journal of Advanced Manufacturing Technology*, vol. 84, no. 1-4, pp. 5–16, 2016.

[12] Y. Zhang, Z. Guo, J. Lv, and Y. Liu, "A Framework for smart production-logistics systems based on CPS and industrial IoT," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 9, pp. 4019–4032, 2018.

[13] L. Martí, R. Puertas, and L. García, "The importance of the Logistics Performance Index in international trade," *Applied Economics*, vol. 46, no. 24, pp. 2982–2992, 2014.

[14] D. Alem, A. Clark, and A. Moreno, "Stochastic network models for logistics planning in disaster relief," *European Journal of Operational Research*, vol. 255, no. 1, pp. 187–206, 2016.

[15] N. Boysen, S. Emde, M. Hoeck, and M. Kauderer, "Part logistics in the automotive industry: decision problems, literature review and research agenda," *European Journal of Operational Research*, vol. 242, no. 1, pp. 107–120, 2015.

[16] M. Coe, "Missing links: logistics, governance and upgrading in a shifting global economy," *Review of International Political Economy*, vol. 21, no. 1, pp. 224–256, 2014.

[17] S. A. R. Khan, D. Qianli, W. SongBo, K. Zaman, and Y. Zhang, "Environmental logistics performance indicators affecting per capita income and sectoral growth: evidence from a panel of selected global ranked logistics countries," *Environmental Science and Pollution Research*, vol. 24, no. 2, pp. 1518–1531, 2017.

[18] V. Sundquist, L.-E. Gadde, and K. Hulthén, "Reorganizing construction logistics for improved performance," *Construction Management and Economics*, vol. 36, no. 1, pp. 49–65, 2018.

[19] F.-S. Chang, J.-S. Wu, C.-N. Lee, and H.-C. Shen, "Greedy-search-based multi-objective genetic algorithm for emergency logistics scheduling," *Expert Systems with Applications*, vol. 41, no. 6, pp. 2947–2956, 2014.

[20] H. Soleimani and K. Govindan, "Reverse logistics network design and planning utilizing conditional value at risk," *European Journal of Operational Research*, vol. 237, no. 2, pp. 487–497, 2014.

[21] X. Bing, J. M. Bloemhof-Ruwaard, and J. G. A. J. van der Vorst, "Sustainable reverse logistics network design for household plastic waste," *Flexible Services and Manufacturing Journal*, vol. 26, no. 1-2, pp. 119–142, 2014.