

Article

ACTS: An Ant Colony Based Transmission Scheduling Approach for Cloud Network Collaboration Environment

Ruiying Cheng ¹, Pan Zhang ¹, Lei Xie ¹, Yongqi Ai ¹ and Peng Xu ^{2,*} 

¹ State Grid Information & Telecommunication Branch, Beijing 100761, China; chengruiying@sgcc.com.cn (R.C.); zhang@sgcc.com.cn (P.Z.); xielei@sgcc.com.cn (L.X.); harutoayq@pku.edu.cn (Y.A.)

² State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications, Beijing 100761, China

* Correspondence: xupeng@bupt.edu.cn

Abstract: In traditional cloud computing research, it is often considered that the network resources between edge devices and cloud platform are sufficient, and the symmetry between the upward link from edge devices to the cloud platform and the downward link from cloud platform to edge devices is definite. However, in many application scenarios, the network resources between edge devices and cloud platform might be limited, and the link symmetry may not be guaranteed. To solve this problem, network relay nodes are introduced to realize the data transmission between edge devices and the cloud platform. The environment in which network relay nodes that can cooperate with the cloud platform is called cloud network collaborative environment (CNCE). In CNCE, how to optimize data transmission from edge devices to cloud platform through relay nodes has become one of the most important research topics. In this paper, we focus on the following two influencing factors that previous studies ignored: (1) the multi-link and multi-constraint transmission process; and (2) the timely resource state of the relay node. Taking these factors into consideration, we design a novel data transmission scheduling algorithm, called ant colony based transmission scheduling approach (ACTS). First, we propose a multi-link optimization mechanism to optimize the constraint limits. This mechanism divides the transmission into two links called the downlink relay link and uplink relay link. For the downlink relay link, we use the store-and-forward method for the optimization. For the uplink relay link, we use the min–min method for the optimization. We use the ant colony algorithm for the overall optimization of the two links. Finally, we improve the pheromone update rule of the ant colony algorithm to avoid the algorithm from falling into a local optimum. Extensive experiments demonstrate that our proposed approach has better results in transmission efficiency than other advanced algorithms.

Keywords: resource scheduling; state prediction; store-and-forward; ant colony optimization



Citation: Cheng, R.; Zhang, P.; Xie, L.; Ai, Y.; Xu, P. ACTS: An Ant Colony Based Transmission Scheduling Approach for Cloud Network Collaboration Environment. *Symmetry* **2021**, *13*, 2109. <https://doi.org/10.3390/sym13112109>

Academic Editors: Aviv Gibali and Tomohiro Inagaki

Received: 25 August 2021

Accepted: 19 October 2021

Published: 6 November 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2021 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/>).

1. Introduction

As more and more enterprises adopt cloud platforms as their IT infrastructure, these enterprises also realize that the network quality from edge devices to cloud platforms would greatly affect the efficiency of applications in cloud platforms. In addition to optimizing the network access quality of edge devices and cloud platforms, deploying network resource nodes to relay the data transmission between edge devices and the cloud platform in the network could be a very effective solution as well. The environment in which network resource nodes cooperate with the cloud platform is called the cloud network collaborative environment (CNCE), shown in Figure 1. In CNCE, edge devices mainly focus on data collection, which could generate a large amount of data to transmit to the cloud platform for data processing. The data transmission link from edge devices to cloud platform is composed of two types of links. Links which are used to transmit data from edge devices to relay nodes are downlink relay links (DRL). Links which are used

to transmit data from relay nodes to cloud platform are uplink relay links (URL). How to optimize data transmission from edge devices to cloud platform through relay nodes has become one of the most important research topics in CNCE.

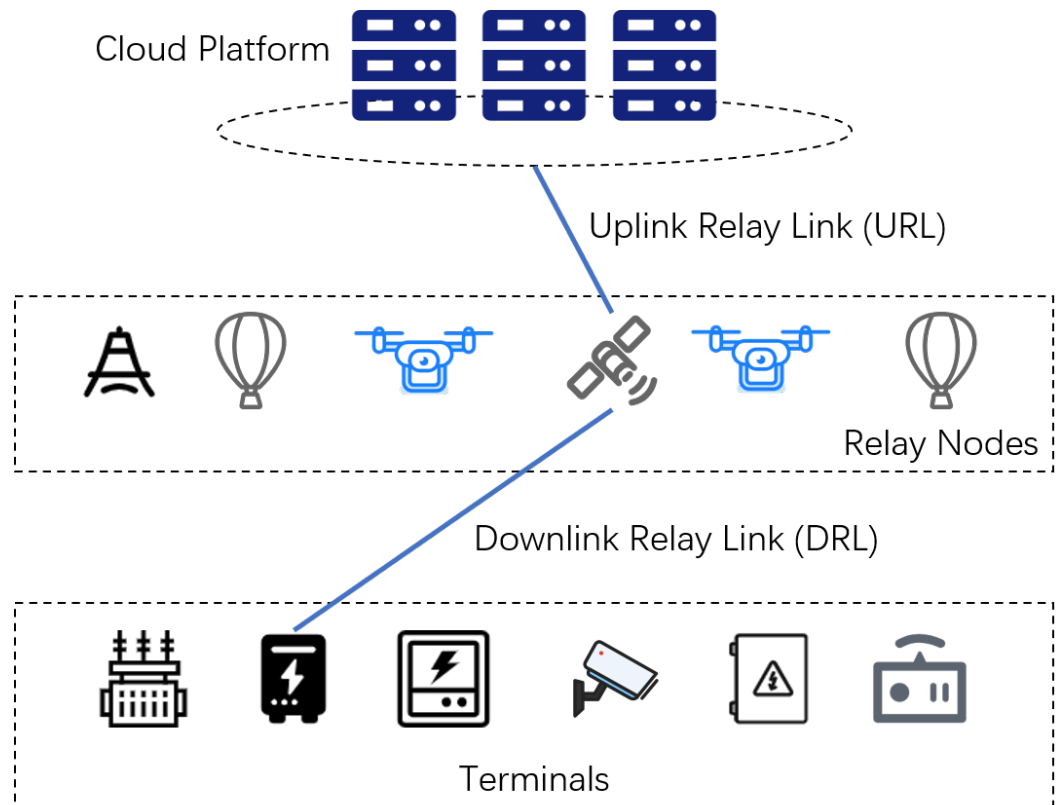


Figure 1. Data transmission scheduling network.

Data transmission scheduling is a large-scale complex combinatorial optimization problem [1]. We need to consider many factors involving resources in the network, multiple relay nodes, multiple tasks and multiple constraints. In the literature [2], the study focuses on overcoming transfer bottlenecks in large data sets. A novel guideline algorithm is given that takes full account of the pipelining, parallelism, and concurrency parameters. The proposed method improves overall network utilization effectively. In this study, it mainly focuses on the overall network scheduling, but our study focuses more on sub-link scheduling in a multi-level network. In Ref. [3], the authors focus on the stream processing problem of edge-end devices under hybrid cloud–edge networks. A highly efficient scheduling strategy to maximize the throughput rate with limited edge resources is proposed, fully considering CPU cost and message size. We take into account another important constraint, the transmission time window of edge devices, so that the algorithm can be applied to more specific industrial areas. The paper mentioned above focuses on the general problem of optimizing data transfer around a network. This paper is targeted on applications which have the following characteristics. The transmission process contains two links. Downlink relay link is the data transmission process from edge devices to relay nodes, and uplink relay link is the data transmission process from relay nodes to the cloud platform. Each link has its own constraints, such as resource constraint or task constraint. The downlink relay link is affected by the environment, and the mutual visibility between edge devices and relay nodes is limited. Data transmission using a downlink relay link is only possible when the transmission network between edge devices and relay nodes is available.

Previous research on this problem mainly focused on optimization on a single transmission link [4–8]. Qing Fang et al. [4] proposed an artificial ant colony algorithm to

optimize the task scheduling of downlink relay links. On this basis, we fully consider the scheduling of uplink and the overall network. Chen Hao et al. [5] designed a data transmission conflict window model for a cluster of satellites and proposed a novel genetic algorithm. In their algorithm, a rote learning operator is designed, which can generate heuristics from past scheduling results and lead the searching direction. Finally, they verified the correctness and practicality of the proposed algorithm through experiments. In the three-layer network model we proposed, we also considered the time window constraint. Zhang Feiyu et al. [6] presented a fast and simple priority scheduling method and demonstrated that it had produced acceptable schedules in most cases. Zhao Jun et al. [7] proposed an optimization algorithm based on the time constraint and period constraint. Liang Z et al. [8] pointed out the use of relay systems to solve the problem of observing large amounts of data from satellites and proposed a scheduling algorithm based on greedy policies for scheduling this scenario. Our research applies the ideas of these three articles to the industrial internet field. In particular, their algorithmic ideas are applied in our sub-link scheduling.

So far, most existing studies—such as [4,5]—focused on various constraints on the individual links, but ignored the optimization of the overall network. Or, studies such as [9] achieved optimization of the overall network, but not enough consideration for individual links' constraints. However, constraints of individual links should be taken into account, and also overall optimization should be pursued. In our scenario, the transmission consists of two links. This paper proposes an overall optimal scheduling algorithm that considers the constraints of each link and the resource status of the relay node, called ant colony based transmission scheduling approach (ACTS). The ant colony algorithm is a probabilistic algorithm used to find optimal paths. Therefore, our new method contains an ant colony algorithm that is responsible for the overall link transmission scheduling. However, it still faces two challenges:

1. How to design an overall optimization model. In the overall scheduling process, we assume two kinds of transmission links. There are resource constraints and time window constraints, which should be respected for each link. We try to achieve the best overall optimization result while satisfying the different constraint limits of each link.
2. How to solve the local optimum problem of ant colony algorithm. It is difficult for the ant colony algorithm to avoid falling into a local optimality. This problem is particularly serious when there are limited numbers of relay nodes.

In order to solve the above two challenges, we propose an effective node state prediction mechanism and redesign the parameters and structure of the ant colony algorithm, with the following details.

1. We propose a multi-link optimization mechanism based on the ant colony algorithm. The transmission is divided into two links. To meet the need, this mechanism uses an ant colony algorithm for the overall optimization. This mechanism also contains the store-and-forward transmission method in downlink relay link and the min–min method in uplink relay link. This mechanism is not only an overall optimization, but also contains optimization methods for each local link.
2. We design a new update rule for the pheromone of the ant colony algorithm. The traditional ant colony algorithm is easy to fall into a local optimum, which is caused by a large number of pheromones gathered in non-optimal paths. To solve this problem, we improve the pheromone update rule.

We conducted experiments to demonstrate the validity of the model. The main contributions are as follows:

1. To achieve efficient data transmission scheduling, we propose the ACTS model for data transmission scheduling. In the ACTS model, we first design the node state prediction mechanism to obtain reliable relay node state information. Then, we design a multi-link optimization mechanism based on the ant colony algorithm as a way to

take into account the respective constraints of each link. Finally, we refine the update rules of the pheromone of the ant colony algorithm to prevent the algorithm from falling into a local optimum.

2. We conducted experiments to demonstrate that our method consistently outperforms the current state-of-the-art methods which support time window constraints.

We organized the paper with following: Section 2 describes our work's scenario and the problems we meet. Section 3 reviews the previous related work. Section 4 describes the problem in resource scheduling. Section 5 describes the detail of our proposed model that combines a prediction mechanism, a multi-link optimization mechanism and a pheromone update rule. Section 6 describes the experimental result that demonstrates the superiority of our model. Finally, Section 8 concludes the whole paper.

2. Background

In traditional cloud computing researches, each device located at the edge of the network acts as a data collection devices, which is mainly responsible for sensing the environment and collecting relevant data. The collected data are sent to the cloud platform for processing. When the communication environment between these edge devices and the cloud platform is excellent, the efficient data transmission ensures the effectiveness of the application located in the cloud platform. However, in many specific application scenarios, the data transmission resources are highly constrained. There may even be periodic or irregular network connection outages. For example, in the harsh natural environment of the plateau and other areas, remote unattended power equipment is affected by the natural environment; only 10:00 to 15:00 in a 24-h day in winter has a stable communication capacity. In specific applications, communication relay nodes are often introduced in order to guarantee the access capability of edge devices. Through the relay node's forwarding capability, the communication quality between the edge devices and the cloud platform is improved. However, how to make full use of relay nodes to optimize the data transmission is worth studying. In this background, achieving efficient data transmission in a resource-constrained network, especially optimizing data transmission through relay nodes, has become one of the most important research topics.

The whole transmission process utilizes two links. The link between the edge device and relay node is called downlink relay link. The link between the relay node and the cloud platform is called uplink relay link. Each link has various transmission constraints, including resource constraints and time window constraint. For resource constraints, we need to know the timely resource status of the relay node to adapt the load balancing strategy. For time window constraints, we assume that data cannot be transmitted between edge devices and relay nodes at all times. Only when the network between edge devices and relay nodes meets some specific conditions can they be visible and send messages to each other. In a real industrial environment, where the individual edge devices may be in a harsh environment, the time windows are inconsistent due to complicating factors, such as weather. In this paper, we construct a simulation environment to simulate this scenario, where we assume that all time windows of edge devices are consistent.

To solve the problem mentioned above, we propose a novel data transmission scheduling algorithm, called ant colony based transmission scheduling approach (ACTS). Then, we propose a multi-link optimization mechanism to solve the constraints of each link. Then, an ant colony algorithm is used for the overall scheduling. Finally, we improve the pheromone update rule of the ant colony algorithm to avoid the algorithm from falling into a local optimum.

3. Related Work

3.1. Resource Scheduling

With the development and popularity of cloud computing, more and more edge devices will collect a large amount of data every day. The collected data will be sent back to the cloud platform in various ways for data analysis and processing. Therefore,

the resource transmission scheduling problem has become a very important research direction, and many researchers have proposed many research methods for different optimization objectives from different perspectives.

In Ref. [4], the study focuses on the transmission scheduling of satellite data. The authors mainly focus on the resource scheduling from the data source to the relay node, and finally propose an artificial bee colony algorithm (ABC) based task scheduling constraint planning model for this scenario, considering various constraints, such as time windows. In Ref. [10], the authors focus on the data scheduling problem from the relay node to the cloud platform and propose a serial scheduling method. First, the request arrangement for this link is determined, and then the scheduling algorithm is proposed based on the obtained request order. In Ref. [11], the study focuses on handover decisions to select the optimal access point in a space-ground integrated network. A handover decision model is developed by using link quality, network quality, load balance and user demand as decision indicators. For Refs. [9,12–14], these studies are based on ant colony algorithm to solve scheduling problems in network environment. They improve the pheromone update rule of the ant colony algorithm by setting different parameters and controlling the constraints to make the algorithm perform better in specific network scenarios.

All the above studies are for a single link or simply treat the transmission network as a whole. However, in a real transmission process, each link has different constraint limits. In this paper, we consider the constraint of downlink relay links and the constraint of uplink relay links and then study the integrated data transmission topology network optimization problem of the overall two links.

By our research, two papers are similar to our work. In Ref. [15], the study proposes a novel three-layer network architecture model and proposes a computational scheme of mutual cooperation between the edge devices. In the cloud computing layer, it provides an effective balanced transmission method to solve the data transmission delay from edge devices to cloud servers. Compared with our research, we implemented both transmission scheduling methods on the three-layer network model. In Ref. [16], the study presents a novel hybrid algorithm to construct a multicast routing tree under bandwidth, delay and delay jitter constraints. The algorithm is based on the ant colony optimization (ACO) algorithm, and devises a memory detection search (MDS) strategy. This is an effective hybrid algorithm.

However, the scenarios to which our theory applies are different. The theories in [15,16] mainly apply to networks in which the links of edge devices and relay nodes are stable, and the relay nodes also have complex topologies. In our work, we focus on solving the transmission problems of edge devices in various extreme environments of the industrial internet field. Extreme environments can cause time window constraints in the transmission process; our algorithm has to be able to adapt to these conditions. Due to the different application scenarios, our relay node interconnection is much simpler.

3.2. Ant Colony Algorithm

The ant colony algorithm is a meta-heuristic algorithm used to find the optimal path. The algorithm has the characteristics of distributed computing, positive information feedback and heuristic search [17].

The basic idea of the ant colony algorithm is that the walking path of an ant represents the feasible solution of the problem to be optimized. All paths of the entire ant colony constitute the solution space of the optimization problem. Ants with shorter paths release more pheromones. As time progresses (which in terms of the execution of the algorithm means that more iterations are performed), the pheromones accumulated on shorter paths gradually increase, and the number of ants who choose the shorter path increase. In the end, the entire ant colony is concentrated on the optimal path under the action of positive feedback. Therefore, it can be regarded as the optimal solution of the optimization problem.

The two key parts of the ant colony algorithm are transition probability and pheromone update rule, which determine the speed and quality of the ant colony algorithm.

3.2.1. Transition Probability

In the environment, at time t , the path where ant m ($m = 1, 2, \dots, M$) moves from point i to point j is determined by the transition probability $p_{ij}^m(t)$. The value of $p_{ij}^m(t)$ of the feasible path near point i is calculated by Equation (1).

$$p_{ij}^m(t) = \begin{cases} \frac{[\tau_{ij}(t)]^\alpha [\eta_{ij}(t)]^\beta}{\sum_{j \in allowed} [\tau_{ij}(t)]^\alpha [\eta_{ij}(t)]^\beta}, & j \notin T \\ 0, & j \in T \end{cases} \quad (1)$$

where $\tau_{ij}(t)$ is the pheromone concentration on the path (i, j) at time t ; T represents the collection of points already traveled. $\eta_{ij}(t)$ is the heuristic information from point i to point j , whose value is $\frac{1}{d_{ij}}$. d_{ij} is the distance from point i to point j . α is the pheromone concentration factor. β is the distance heuristic information strength factor. *allowed* is the set of nodes that can be selected in the next step. Then, the next best path needs to be selected randomly in the next iteration (which in terms of the execution of the algorithm means that more iterations are performed).

3.2.2. Pheromone Update Rule

First, during the ant colony initialization, the pheromones on each path are initialized to the same value according to Equation (2).

$$\tau_{ij}(0) = 1 \quad (2)$$

The pheromone update is divided into the local update and global update. In this paper, the global update is selected, that is, the pheromone update is performed when all ants complete the current iteration. Over time, some of the pheromones on the path will evaporate, and the ant's path will release a certain concentration of pheromones [7,14,17]. After all the ants complete an iteration, the pheromones are updated according to Equation (3).

$$\tau_{ij}(t+1) = (1 - \rho)\tau_{ij}(t) + \rho\Delta\tau_{ij}(t, t+1) \quad (3)$$

where ρ is the pheromone evaporation coefficient. $1 - \rho$ is the pheromone residue factor. $\Delta\tau_{ij}(t, t+1)$ is the pheromone increment on path (i, j) .

$$\Delta\tau_{ij}(t, t+1) = \sum_{m=1}^M \Delta\tau_{ij}^m(t, t+1) \quad (4)$$

In Equation (4), M is the total number of ants. $\Delta\tau_{ij}^m(t, t+1)$ is the pheromone concentration which the m th ant in the path (i, j) releases. There are three pheromone update strategies for its value: ant-quantity model, ant-density model and ant-cycle model [17,18]. The difference between three models is that they have different updating rules of pheromones: the first two models use local information, and the ants update the pheromones on their paths after completing one step. The last model uses the overall information. The ants update the pheromones on their paths after all ants have completed this iteration. Because the last model fully considers the global information, its performance is better than the other two models. In order to improve the global search ability of the ant colony, we use the ant-cycle model. The specific rules for pheromone updates are given in Equation (10) of Section 5.3.

4. Problem Formulation

In CNCE, data transmission is divided into two links: downlink relay link and uplink relay link (recall Figure 1). In the downlink relay link, relay nodes have limited resources and only some specific relay nodes that within a certain distance can communicate with the cloud platform. The edge device need to send data to the relay node that can communicate

with the cloud platform. Meanwhile, the edge device can communicate with the relay node only when specific conditions are met, due to environmental factors. In the uplink relay link, the cloud platform coordinates resources to receive data from relay nodes. The symbol definitions are shown in Table 1.

Table 1. Symbol definition.

Symbol	Definition
$SS = \{ S_{s1}, S_{s2}, \dots, S_{sm} \}$	Collection of source nodes.
$RS = \{ R_{s1}, R_{s2}, \dots, R_{sn} \}$	Collection of relay nodes. $R_{si} = (SC_i, SM_i, SD_i, SN_i)$ represents CPU, memory, hard disk and bandwidth.
$W_i = \{ W_{ij}^k \}$	The k-th visible time window of the i-th source node and the j-th relay node. During time window, data can be transmitted between these two nodes, W_{ij}^k is the length of time window, which is represented by $[W_{ijk}^s, W_{ijk}^e]$.
$GS = \{ G_{s1}, G_{s2}, \dots, G_{sk} \}$	Collection of center cloud platform.
$Task_i = \{ Task_i^j \}$	Data transmission tasks of the i-th node. $Task_i^j = (TP_j, TD_j, TC_j)$ represents task priority, data size and transmission link status (downlink relay link is 1 and uplink relay link is 0).
$V = \{ x_i^j, t_{ij}^s, t_{ij}^e, c_i^j \}$	Scheduling decision variables. x_i^j is bool type, and $x_i^j = 1$ represents $Task_i^j$ can be allocated; t_{ij}^s and t_{ij}^e represent the time of start and end of transmission; c_i^j represents the constraint from i to j.

The specific constraints of CNCE are as follows:

4.1. Downlink Relay Link Constraints

1. Time window constraint

Edge devices are far away and located in mostly harsh climatic environment. Therefore, the edge device and the relay node are visible to each other only within a specific time constraint. Figure 2 shows that the edge device is visible to the relay node R_{s1} and R_{s2} within the time windows $W1 = [T1_{in}, T1_{out}]$ and $W2 = [T2_{in}, T2_{out}]$. The data can be downloaded by R_{s1} in FT_1 or by R_{s2} in FT_2 .

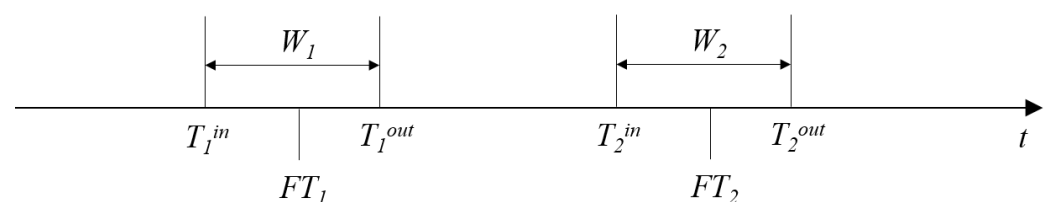


Figure 2. Time window diagram.

For example, suppose that we have a number of terrestrial towers that act as relay nodes for data transmission. However, these towers are located in a very harsh geographical environment with frequent thunderstorms, rain and snow. These factors will make it impossible to transmit data with them at all times, but only during specific time windows when the weather conditions are met. If during a schedule, the edge device cannot transmit with any relay node, tasks will be stored and try to execute in the next schedule.

2. Resource constraint

Each relay node has limited resources, including CPU, memory and bandwidth. The large amount of data generated by edge devices needs to be transmitted. This process requires the CPU, memory and bandwidth of relay nodes. If these resources are not sufficient, then data transfer will be delayed or even blocked, and it is also difficult to guarantee real-time availability. Therefore, it is necessary to ensure that

the relay node has sufficient resources for data transmission. The data transmissions between terminals and relay nodes are competitive; some tasks may not be completed due to insufficient resources by competition. These tasks are stored and participate in the next scheduling. The formalization of the constraint in Equation (5) is as follows:

$$C_2 = \left\{ x \mid \forall Task_i^j \in Task_i, \exists R_{sk} \in RS, sC_k \geq 0 \& sM_k \geq 0 \& tD_j \leq sD_k \& sN_{in_k} \geq 0 \& sN_{out_k} \geq 0, x_i^j = 1 \& c_i^j = 1 \right\} \quad (5)$$

3. Task constraint

Each task is marked as being completed only once and does not participate in subsequent task scheduling.

4.2. Uplink Relay Link Constraints

Relay nodes are generally chosen at locations with good environment. There is no time window constraint due to environmental issues.

1. Resource constraints

We assume that the resource of the cloud platform is sufficient. However, the resources of the relay nodes need to be taken into account, such as CPU, memory and bandwidth. Similar to the description in the downlink relay link constraint, the scarcity of these resources may cause delays or blockages in data transmission. The formalization of the constraint in Equation (6) is as follows:

$$C_4 = \left\{ x \mid \forall Task_i^j \in Task_i, \exists R_{sk} \in RS, sC_k \geq 0 \& sM_k \geq 0 \& tD_j \leq sD_k \& sN_{in_k} \geq 0 \& sN_{out_k} \geq 0, x_i^j = 1 \& c_i^j = 0 \right\} \quad (6)$$

2. Task Constraints

After each transmission task is done, it is marked as completed. That means there are no retransmissions, etc.

5. Methodology

In this paper, we study the resource scheduling problem in the edge node data collection transmission network. This problem is a combinatorial optimization problem. The ant colony algorithm is well suited to solve the combinatorial optimization problem. Therefore, the ant colony algorithm is considered the basis of our model. We propose a new resource scheduling model as in Figure 3 called ant colony based transmission scheduling approach (ACTS). In Figure 3, n represents the times of iterations now; max represents the maximum number of iterations. The prediction results are mainly used to control the task allocation ratio using these states when executing the downlink algorithm during each iteration. It aims at overall optimization, taking into account the constraint on each link. The specific steps are as follows:

1. Initialization of ant colony algorithm parameters. Set the parameters of the ant colony algorithm, including the colony size M , number of iterations K , pheromone enhancement Q , pheromone evaporation factor ρ , and the proportion of pheromone update ants μ . Initialize the task set, with each source node generating a certain number of scheduled tasks.
2. Iterate to find the best path. This is the main part of the algorithm. Each iteration of every ant in the colony achieves the scheduling of all tasks, which will eventually result in a current optimal solution.
 - (a) Randomly generate the starting node for each ant.
 - (b) Optimal selection of paths and selection of suitable nodes based on state transfer probabilities.

- (c) Modify the taboo table by placing the node in the taboo table after selecting the path, using the taboo table to prevent repeated selection of the same node and avoid the loop path [17].
 - (d) Calculate the path distance of individual ants to obtain the optimal path length.
 - (e) Update the pheromones on each path according to the update rules.
 - (f) Iterate incrementally while resetting the route record table used to record the path that each ant assigned to.
3. Output the best path for each generation, and the optimal solution is obtained after several iterations. The overall transmission time is our optimization metric, and the solution with the smallest transmission time is optimal. The overall transmission time can be calculated by Equation (7). The specific calculation procedure is also provided in Algorithm 1.

$$Time_{overall} = \sum_{i=1}^{n_{task}} Time_{downlink}(i) + \sum_{i=1}^{n_{task}} Time_{uplink}(i) \quad (7)$$

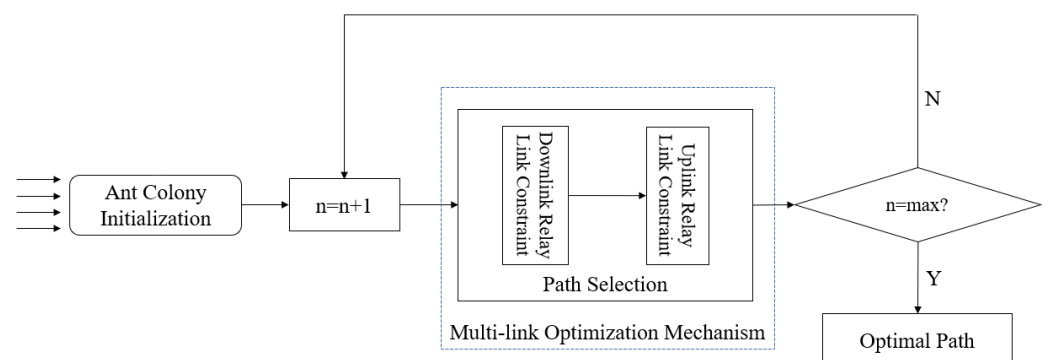


Figure 3. An ant colony optimization model.

In this model, the node state (CPU, memory, storage, etc.) and ant colony algorithm parameters (the colony size M , number of iterations K , pheromone enhancement Q , pheromone evaporation factor ρ , the proportion of pheromone update ants μ) are considered the initialization data of the ant colony. The transmission task is the input of the ant colony algorithm. After several iterations, the optimal path is output, which is the optimal data transmission strategy.

There are two challenges to be faced in order to achieve the ACTS model:

1. The transmission is divided into multiple links. Each link has different constraints. The traditional ant colony algorithm cannot reflect the difference between different links.
2. The ant colony algorithm is easy to fall into the local optimum, especially when the number of the node is small.

Therefore, we propose a multi-link optimization mechanism based on the ant colony algorithm to solve the problem of each link. Then, we try a new pheromone update rule to obtain better result.

They are described in Sections 5.2 and 5.3, respectively.

5.1. Overview of ACTS

Although we have basically described the entire algorithm, there are still some details that need to be clarified. In this section, we provide equations and processes for the details covered above. Then, we give an example to show our algorithm more concretely. Finally, the algorithm pseudo-code is given at the end of this section.

Each ant chooses a path for each iteration, and more specifically, it is the path through the relay nodes that upload the tasks of edge devices to the cloud. Reaching any cloud node

satisfies the problem. Resource and time window constraints are used in the calculation of downlink scheduling times for each iteration. Our algorithm is scheduled in minutes, which means that a group must converge within 60 s. Once the task scheduling path is output by ACTS, tasks are immediately scheduled according to it. At the next scheduling, the state (including the time window) is updated to ensure that the constraints are always up-to-date. If the transmission of any of the tasks is interrupted due to a sudden time window constraint, the tasks are stored and participate in the next scheduling.

There are five concepts that need to be described more specifically:

1. The specific meaning of entities in ACTS
Each ant is assigned a path in an iteration and is updated only before the next iteration. The ant represents an assignment scheme of tasks to relay nodes; the path refers to the selection of relay nodes as paths from the edge nodes to the cloud; and the tasks refer to the transmitted data collections generated by the edge devices.
2. Rules for updating pheromones
The pheromone update rule follows Equation (10). The improvement to the ant colony algorithm is reflected in the fact that only the pheromones of some ant paths are updated. The ant paths' updated pheromones are those with a short task transmission time, which improves the convergence speed and prevents the algorithm from falling into a local optimum.
3. Probability of ant path selection
Random path selection performs after all ants have completed their paths in each iteration. The probability of ant path selection follows Equation (1).
4. Evaluation of the constraints
The time window constraint is used to determine which nodes can participate in the scheduling of a particular task; the resource constraint is used to calculate of the task allocation ratio using the hierarchical analysis matrix.
5. Calculation of transmission time in sub-links
The transmission time can be calculated by the task data size, and the node bandwidth follows Equation (8).

To provide a better overview of the algorithm, a concrete example is given as follows:

Suppose there is a network with the topology shown in Figure 4. Edge devices E_1, E_2, E_3 generated data transmission tasks $T_{E1} = 20$ to C_1 , $T_{E2} = 50$ to C_2 , $T_{E3} = 30$ to C_2 . Under the time window constraints now, E_1 can communicate with R_1 ; E_2 can communicate with R_1, R_2 ; E_3 can only communicate with R_2 . The current node status is as follows: $C_{R1} = 6$ represents the CPU of R_1 , $B_{R1} = 10$ represents the bandwidth of R_1 , $M_{R1} = 10$ represents the memory of R_1 ; $C_{R2} = 3$ represents the CPU of R_2 , $B_{R2} = 5$ represents the bandwidth of R_2 , $M_{R2} = 5$ represents the memory of R_2 .

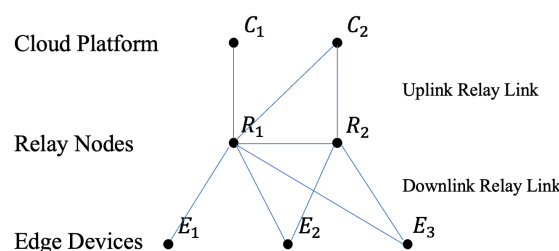


Figure 4. A network topology graph.

In the scheduling process of downlink, using the hierarchical analysis matrix with resource constraints, we can calculate that the tasks of E_2 will be transmitted to R_1 and R_2 in the ratio of 2:1 by the downlink. Due to the time window constraint, the data of E_1 and E_3 can only be transmitted to R_1 and R_2 , respectively. The task T_{E2} of E_2 is split into two parts D_1 and D_2 for transmission, and it is assumed that R_1 is selected for relaying data to

the cloud (this path is determined by the ant colony algorithm as described later). The SAF method (which is described in Section 5.2 below) is used in this process, and the transmit time can be calculated by Equation (8). Suppose that the start time of transmission is T_{start} . For task packet D_1 , the transmission time T_{D1} includes only the time of transmission from E_2 to R_1 and T_{start} . For the task packet D_2 , according to store-and-forward, this part of the data will be transmitted and stored in R_2 , spending time Tr , and then forwarded to R_1 , spending time St . The transmission time for $T_{D2} = T_{start} + Tr + St$. Finally, the total transmission time for T_{E2} in downlink relay link is $MAX\{T_{D1}, T_{D2}\}$.

In the scheduling process of uplink, we use the min-min algorithm for scheduling. First, we list tasks in descending order T_{E2}, T_{E3}, T_{E1} . Next, we assign them in order according to the min-min algorithm. The bandwidth of R_1 is higher than the bandwidth of R_2 , so the task has a shorter transmission time on R_1 . Since they can all be transmitted through R_1 and take the shortest time, initially all three tasks are assigned to R_1 . The current load balance is bad, so we need to reschedule some of the tasks of heavily loaded station R_1 to the less heavily loaded station R_2 . In this scenario, we assume that one task is rescheduled (the last in the descending-order list) T_{E1} . The tasks of heavy station R_1 are scheduled to light station R_2 . We need to schedule the backward tasks to the light station. Finally, the scheduling result is $\{R_1 : T_{E2}, T_{E3}\}, \{R_2 : T_{E1}\}$.

In the ant colony algorithm of the overall network, there are two paths in total. $P_1 = \{E_1 \rightarrow R_1 \rightarrow C_1, E_2 \rightarrow R_1 \rightarrow C_2, E_3 \rightarrow R_2 \rightarrow C_2\}$ and $P_2 = \{E_1 \rightarrow R_1 \rightarrow C_1, E_2 \rightarrow R_2 \rightarrow C_2, E_3 \rightarrow R_2 \rightarrow C_2\}$. There are no loops in either P_1 or P_2 ; we ensure this by the taboo table. For parameters of the ant colony algorithm, we assume that the number of ants is 2; the number of iterations is 5; the pheromone update ratio is 0.5. In the first iteration, the ant M_1 chooses P_1 and ant M_2 chooses P_2 . After the iterations, the calculation result shows that M_1 takes less time. According to the pheromone update rule, both the pheromone on P_1 and P_2 will evaporate, and only M_1 can update the pheromone on P_1 . In the second iteration, according to the random path selection equation, both ants tend to choose P_1 . As the number of iterations increases, the pheromone on P_1 becomes denser and the pheromone on P_2 becomes weaker, thus forming convergence. Finally, the number of iterations reaches the maximum, and the optimal path P_1 is output. P_1 tends to choose the relay node R_1 with a better state. In the set of P , we call P_1 optimal, and follow such a path to perform the tasks.

The overall algorithm of ACTS is described in Algorithm 1.

Algorithm 1 ACTS.

Input: The node state (CPU, memory, storage, etc.), S ; current time window for edge devices, W ; the colony size, M ; number of iterations, K ; pheromone enhancement Q ; pheromone evaporation factor, ρ ; the proportion of pheromone update ants, μ ; transmission tasks for edge devices, TT ; relay nodes, R ; edge devices, E ; total task allocation solution, P ;

Output: Optimal task allocation solution, P_{best} ;

```

1: function DOWNLINKSAF(task allocation solution  $p$ )
2:    $t \leftarrow 0$  /*  $t$  represents the transmission time of downlink */
3:   for each  $e$  in  $E$  do
4:      $R' \leftarrow$  relay nodes that can communicate with  $e$  under time window constraints
        $W_e$ 
5:     Use  $S$  and the final decision matrix allocate task  $TT_e$  to  $R'$ 
6:      $t_1 \leftarrow MAX(\text{time of upload to } R')$ 
7:      $t_2 \leftarrow 0$ 
8:      $r_{main} \leftarrow$  relay node for  $e$  in solution  $p$ 
9:     for each  $r$  in  $R'$  do
10:       $t_{r,r_{main}} \leftarrow$  transmission time of  $r$  to  $r_{main}$ 
11:      if  $t_2 < t_{r,r_{main}}$  then
12:         $t_2 \leftarrow$  transmission time of  $r$  to  $r_{main}$ 

```

```

13:         end if
14:     end for
15:      $t_e \leftarrow t_1 + t_2$ 
16:      $t \leftarrow t + t_e$ 
17: end for
18: return  $t$ 
19: end function
20:
21: function UPLINKMINMIN(task allocation solution  $p$ )
22:      $t \leftarrow 0$  /*  $t$  represents the transmission time of uplink */
23:     for each  $TT_i$  in  $TT$  do
24:          $t_i \leftarrow 0$ 
25:         for each  $r$  in  $R$  do
26:              $t_{i,r} \leftarrow$  Time to transmit task  $TT_i$  using node  $r$ 
27:             if  $t_i > t_{i,r}$  then
28:                  $t_i \leftarrow t_{i,r}$ 
29:                  $r_{best} \leftarrow r$ 
30:             end if
31:         end for
32:          $t_i \leftarrow$  transmission time of  $r_{main}$  to  $r_{best} + t_i$ 
33:          $t \leftarrow t + t_i$ 
34:     end for
35:     return  $t$ 
36: end function
37:
38: function ACTS
39:     for each  $p$  in  $P$  do
40:         Set initial pheromone for initial allocation solution  $p$ 
41:     end for
42:     for each  $m$  in  $M$  do
43:         Randomize a initial allocation solution  $p_m$  for ant  $m$ 
44:         /* taboo table is used to avoid loop path in  $p_m$  */
45:     end for
46:      $n \leftarrow 0$  /*  $n$  is used to record iteration times */
47:     while  $n < K$  do
48:         for each  $m$  in  $M$  do
49:              $time_{downlink} \leftarrow downlinkSAF(p_m)$ 
50:              $time_{uplink} \leftarrow uplinkMinMin(p_m)$ 
51:              $time_m \leftarrow time_{downlink} + time_{uplink}$ 
52:         end for
53:         Sort  $M$  by  $time_m$  in descending sort
54:          $i \leftarrow 0$ 
55:         for each  $m$  in  $M$  do
56:             if  $i <$  pheromone update ants number then
57:                 Update  $p_m$  pheromone by pheromone update rule
58:             end if
59:              $i \leftarrow i + 1$ 
60:         end for
61:         for each  $m$  in  $M$  do
62:             Random select  $p_m$  for ant  $m$  by pheromone concentration
63:             /* taboo table is used to avoid loop path in  $p_m$  */
64:         end for
65:          $n \leftarrow n + 1$ 
66:     end while

```

```

67:    $P_{best} \leftarrow P$  with max pheromone
68:   return  $P_{best}$ 
69: end function

```

5.2. Multi-Link Optimization Mechanism

The transmission process is divided into a downlink relay link and an uplink relay link. The multi-link optimization mechanism refers to different optimizations for each link.

Downlink relay link: In the downlink relay link, the source node transmits data to the target relay node. The communication between nodes is constrained by the time window. There are two transmission methods. The first is that the transmissions carry out in a valid time window between the source and target, called direct transmission (DT), which is proposed by Daewon et al. [19]. In this mode, the frequency of transmissions is extremely limited, and the load pressure on target node is also great. The load of the relay node is different, and many nodes except the target nodes are in a low load state. The second is the store-and-forward method [20], called SAF. SAF is a frequently used method in network transmission. Eventually, the data arrive at the target node by other nodes in a store-and-forward measure, which makes better use of resources and reduces the conflict. We regard a single data transmission task of the source node as the research object, and choose to use the SAF method in the downlink relay link.

In the SAF method, the transmission data are split into several data packets, and different data packets are transmitted by different relay nodes in the store-and-forward measure, which acts as the intermediate node [21]. When these data packets are finally collected on the target node, the relay transmission is complete. D represents the transmission data and D_i represents the i th data packet. The time of D_i from the source node to the target node using the j th relay node is calculated as Equation (8).

$$\begin{aligned}
 Ut_{ij}^{\text{end}} &= T_{ij}^{\text{start}} + Tr_{ij} + St_{ij} \\
 Tr_{ij} &= Task_i / \min\{B_i, B_j\}, St_{ij} = Task_i / B_j
 \end{aligned} \tag{8}$$

where T_{ij} represents the time from the source node to the relay node. St_{ij} represents the time from the relay node to the target node. B_i, B_j represents the bandwidth of node i, j . Both Tr_{ij} and St_{ij} are related to the size of the packet. T_{ij}^{start} indicates the start time of the data packet transmission, which is not only related to the end time of the previous data packet, but also constrained by the time window between the j th relay node. The data packet transmission only starts within a specific time period. For a complete data D , the transmission time is calculated as Equation (9).

$$Ut_j^{\text{end}} = \max\{Ut_{ij}^{\text{end}}\} \tag{9}$$

To obtain the shortest transmission time, we need to consider the segmentation of the data and the selection of the intermediate node at the same time. Constraints include the visible time window with the relay node, the resource state of the relay node, etc. On the one hand, these constraints can directly restrict which nodes can participate in the transmission. On the other hand, they can be used to determine the segmentation of data. This is a multi-attribute decision problem [22], and the attributes of the node state change over time. In this paper, we use an analytic hierarchy method [23] to give an evaluation of each relay node, which will affect the size of the data packets allocated to that node.

The analytic hierarchy diagram is shown as Figure 5. In the method, the multiple attributes of node state are divided into different levels according to the dependency relationship. The lower level is the decomposition of the higher-level attributes. For example, the service quality can be decomposed into two sub-attributes of bandwidth and delay. The sub-attributes in the diagram represent the real-time node state. We use the states of these nodes combined with the decision matrix to evaluate the node state and segment the

data. Our optimization goal is not these properties, but the minimum transmission time. These properties are considered because they have an impact on that goal.

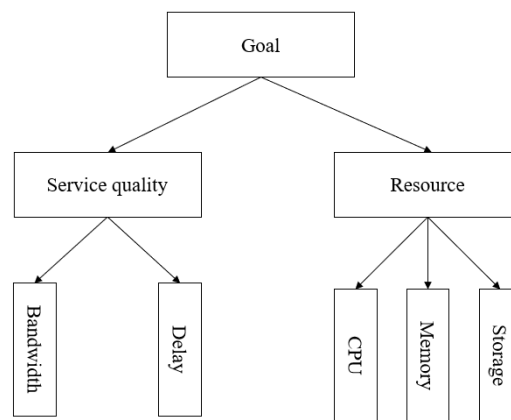


Figure 5. Analytic hierarchy diagram.

Based on the hierarchical structure of attributes, we first construct a multi-attribute decision matrix related to sub-attributes, which could be used to evaluate the node state. We then normalize the decision matrix, and calculate the weight of each decision sub-attribute, and generate the current node state evaluation. When we calculate the upper-level attributes through the sub-attributes, it is necessary to assign weight of sub-attribute. This paper sets the relative weights of the attributes by information entropy. The basic idea is that when different alternative decision-making solutions have a large difference in the value of the same attribute, the attribute has a large impact on the entire decision evaluation. When the difference is small, the attribute has little effect on the decision evaluation. According to the extreme value of information entropy, the value of entropy can just reflect the closeness of different alternatives on the same attribute value. For an attribute, the higher the concentration of its values' distribution, the larger the entropy value is. We finally obtain a decision matrix about level 0. That is, the final decision matrix is generated.

More specifically, here is an example to better understand the use of the hierarchical analysis matrix. Set $A = [a_1, a_2, a_3, a_4, a_5]$ represents the information entropy matrix. The values in the entropy matrix represent bandwidth, delay, CPU, memory, and storage, in that order; they follow the constraint $\sum_{i=1}^5 a_i = 1$. For each relay node, we have the sub-attribute value matrix $R_i = [r_{ib}, r_{id}, r_{ic}, r_{im}, r_{is}]^T$. $r_{ib}, r_{id}, r_{ic}, r_{im}, r_{is}$ represents the bandwidth, delay, CPU, memory, and storage of node i . We use $A \times R_i$ to generate the evaluation of each relay node. The data transmission task is distributed to relay nodes according to the ratio of their evaluation values.

Uplink relay link: In this paper, we propose an improved min–min [24] algorithm to solve the uplink relay link scheduling problem. Min–min is a classical task scheduling algorithm. The main idea of min–min is as follows:

1. For each transmission task in the task set, the minimum times allocated to n stations are calculated. Assume that the task has the shortest time to complete on the k th station, which is $MinTime(i) = MCT(i, k)$. $MCT(i, k)$ represents the minimum complete time for task i to complete transmission on the k th station. There is an array $MinTime$ with m elements.
2. Assume that the a -th element is the smallest in $MinTime$, which corresponds with the b th station; then we assign the a th task to the b th station.
3. Delete the a th task from the task set and update the MCT matrix.

However, there is a big disadvantage of the min–min algorithm, which is that the load balancing performance of the min–min algorithm is bad. To compensate for this disadvantage, to avoid this potential problem, a common method is round robin. However,

min–min is a greedy algorithm that focuses on the execution time of each task. Even after scheduling using the round robin method, if a node always completes the task efficiently, it will still be assigned to multiple tasks. In order to find a better solution, a load-fuzzy classification and local readjustment method is proposed for min–min. Before scheduling, the tasks are sorted according to the number of tasks and the stations are sorted by task processing ability, and then rescheduled. In this way, the problem of load imbalance is weakened to some extent. However, the problem of load imbalance between local resources still exists. In order to solve the problem of local imbalance, fuzzy classification is used. According to the load size of each station, the stations are divided into three categories: heavy load, medium load, and light load. Tasks in heavily loaded stations are rescheduled to stations which are lightly loaded.

As shown in Figure 3, the multi-link optimization mechanism is used in the path selection during the iteration of the ant colony algorithm, including the store-and-forward method and min–min method. It is a good mechanism for overall optimization.

5.3. Pheromone Update Rules

The ant colony algorithm is easy to fall into the local optimum [25], which is caused by the high pheromone concentration on the non-optimal path. It is particularly serious since there are fewer nodes in the scheduling network. In order to prevent the algorithm from falling into a local optimum, the pheromone update rule of the traditional ant colony algorithm is improved.

In this paper, we propose an improved pheromone updating rule. The core idea is to coordinate the convergence speed and global search ability, and resolve the contradiction between the diversity of solutions and the accumulation of the pheromone. The specific description of the new update rule is as follows.

We first sort the ants according to the path length from small to large after all the ants have completed the first iteration, and select the ants sorted in the front part to update the pheromone. The pheromone modification is the same for all ants, which undergoes a pheromone modification (the pheromone is not be updated for the rest of the ants). Pheromone updates are carried out according to Equations (10) and (11):

$$\Delta\tau_{ij}^m(t, t+1) = \begin{cases} \frac{Q}{l_{gbest}} * (l_{worst} - l_{best}) * (1 - \epsilon), rank \leq k \\ 0, rank > k \end{cases} \quad (10)$$

$$k = \mu * M \quad (11)$$

where $\Delta\tau_{ij}^m(t, t+1)$ represents the pheromone increment. l_{gbest} represents the current global optimal solution. l_{worst} and l_{best} represent the local worst and the best solution generated by the current iteration. The units of l_{gbest} , l_{worst} and l_{best} are seconds. ϵ represents the ant colony evolution rate. $\epsilon = k/Kmax$ is the current number of iterations. $Kmax$ is the maximum number of iterations. $rank$ is the sequenced ant number. k represents the number of ants that need to update the pheromones. μ represents the proportion of ants to have updated pheromones. $\mu \in (0, 1)$ and M represent the total number of ants.

However, there is one special case that will still lead our algorithm to a local optimum. When the ants reflecting smaller path lengths are the only ones which have their pheromones updated, the algorithm is trapped in that path. Therefore, we must choose the appropriate value of parameters μ (the proportion of pheromone update ants) and M (the colony size). When the values of μ and M are too small, it is easy to become trapped in the above situation; when the value of μ is too large, the algorithm degenerates into the regular ant colony algorithm. In summary, the choice of values for these two parameters is crucial. These parameters are affected by the number of nodes, so we need to adjust and compare the parameters adequately for environments with different numbers of nodes.

With the accumulation of pheromone on the path, the ants tend to choose the path with the highest pheromone concentration. In order to enhance the optimization ability

of the algorithm in the later iterations and expand the search space, we assume that the max number of iterations of the algorithm into a local optimum is R , which indicates that the algorithm does not show a new optimal solution for R consecutive times, trapping it in a local optimum state. The value of R is related to the scale of the problem and the complexity of the environment. To ensure the versatility of the strategy, the value of R should be as large as possible.

Finally, when the algorithm becomes trapped in a local optimum, the pheromone difference on each path is reduced, and we increase the pheromone evaporation coefficient (as can be seen in Equation (12)), ensuring that the algorithm still has a strong global search ability in the later stage.

$$\tau_{ij}^R = 2 * \tau_{ij}^0, \rho' = 1.5\rho, \text{ if } (N = R) \quad (12)$$

where τ_{ij}^R represents the pheromone concentration in the R th iteration. τ_{ij}^0 represents the initial pheromone concentration.

6. Experimental

In this section, the experimental setup is described in detail in Section 6.1. Next, the experimental results are presented in Section 6.2. Specifically, in order to demonstrate our proposed method's effectiveness, we compare with other baselines to evaluate it as well as the state-of-the-art methods.

6.1. Experimental Setup

Node settings: We have two edge devices (S_{S12}) and five relay nodes (R_{S15}) and four cloud platform receiving devices (G_{S14}). The transmission rate is 900 Mbps, determined by the bandwidth. The operating time of the nodes is from 00:00:00 to 23:59:59. The valid time window between R_{S1} and S_{S1} is specified by default according to the available data and other information in Table 2. The time window between the remaining nodes can be obtained in the same way.

Table 2. Time windows between R_{S1} and S_{S1} .

Number	Start	End
1	04:01	04:58
2	07:15	08:17
3	13:03	14:06
4	16:21	17:20
5	19:35	20:38

Parameter settings: We set the ant colony size to $M = 1000$, number of iterations to $K = 200$, pheromone enhancement to $Q = 2$, the pheromone evaporation coefficient to $\rho = 0.5$, and update the pheromone ant ratio to $\mu = 0.8$. Each source node generates a number of scheduling tasks in a Poisson distribution with a total of 10 groups. For these 10 groups of scheduling tasks, 10 experiments were performed, and the average value was taken as the result in each experiment.

Comparison algorithm: To evaluate the performance of our proposed model, we compare it with several baselines and the state-of-the-art methods, as shown in Table 3, which are DT [19], SAF [20], ABC [4] in the downlink relay link and SOP [26] for the overall link.

Table 3. Comparison methods.

Methods	Description
DT [19]	It directly transmits data by time window between source and target in downlink relay link.
ABC [4]	It uses artificial bee colony algorithm for data transmission in downlink relay link.
SAF [20]	Our proposed approach in downlink relay link. It is based on store and forward.
SOP [26]	It just includes the optimization of each link with overall optimization.
ACTS	Our proposed approach for overall transmission.

6.2. Experimental Results

6.2.1. Experimental Results

To evaluate the performance of our proposed approach, we have 10 experiments. We mark the time of SOP as t_{sop} and the time of ACTS as t_{all} and the relative difference between SOP and ACTS as t_{cut} . t_{cut} is calculated by Equation (13). The experiment results are shown in Table 4.

$$t_{cut} = \frac{t_{sop} - t_{all}}{t_{sop}} \quad (13)$$

Table 4. Experimental results.

Number	Task Amount	Transmission Time in Downlink Relay Link			Transmission Time in Overall Link		$t_{cut}/\%$
		DT/s	ABC/s	SAF/s	SOP/s	ACTS/s	
1	50	4636	3884	3479	7745	5019	35.2
2	98	8572	7536	6771	14,442	10,037	30.5
3	158	14,304	12,233	10,504	23,744	18,401	22.5
4	199	18,590	15,155	13,395	30,265	26,058	13.9
5	269	24,619	20,260	18,301	41,256	34,944	15.3
6	312	28,762	23,593	21,186	46,807	40,114	14.3
7	371	33,433	28,008	24,868	56,137	47,548	15.3
8	391	36,294	29,211	26,571	59,904	49,001	18.2
9	462	42,998	33,914	30,985	71,842	56,611	21.2
10	530	47,461	39,294	35,877	80,626	62,808	22.1

We compare three methods in the downlink relay link, including DT, SAF and ABC. The result is shown as Figure 6. The x-axis shows the task amount. The y-axis shows the time cost by the transmission. The method with the lower cost time performs better for the same task amount. It can be seen from the figure that the DT method is worst, and the ABC method has the second-best effect, and the SAF method is the best, which could transmit all data more efficiently. The result of the DT method may be caused by resource constraints.

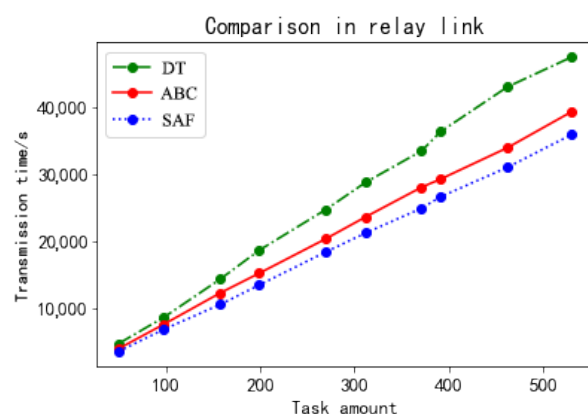


Figure 6. Comparison in downlink relay link.

We compare two methods in the overall link, including SOP and ACTS. The result is shown in Figure 7. The x-axis shows the task amount. The y-axis shows the time cost by the transmission. The performance is evaluated in the same way as for the downlink. The method with the lower cost time performs better for the same task amount. It can be seen from the figure that compared with the traditional step-by-step optimization SOP for different numbers of scheduling tasks, the global optimization ACTS can transmit all data more efficiently.

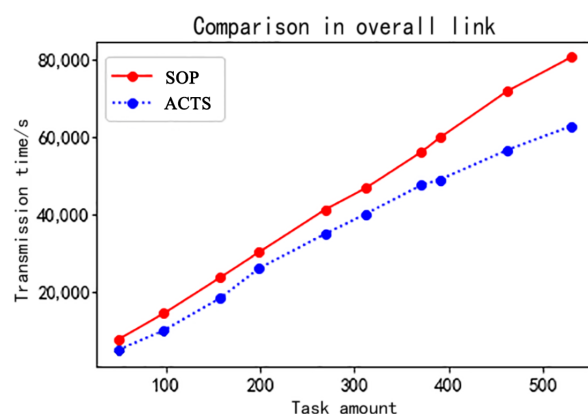


Figure 7. Comparison in overall link.

6.2.2. Effects of Parameters

Task amount: To explore the effect of task amount on our method, ten experiments were performed with different numbers of tasks. From Figure 6, we can know that in the downlink relay link, as the number of tasks increases, the time consumption does not increase proportionally with the number since resource constraints still have a certain effect. The gap between the two methods of DT and SAF increases, which also shows that our proposed method, which uses SAF for downlink scheduling is more suitable for scenarios with large workloads. We use Equation (13) to calculate the time used under a specific task amount to evaluate the performance of methods in the overall link. We can use Equation (13) to calculate the t_{cut} of each case. From Table 4, we can know that from cases 1 to 4, the t_{cut} decreases progressively since the tasks of these groups of cases are small, making the time of SOP method small and the ratio of the shortened time to the SOP method time small with a downward trend. From cases 4 to 10, the t_{cut} gradually increased since the receiving capacity and transmission capacity of the cloud platform remained unchanged, while the number of tasks continued to increase, so the globally optimized result also increased. Then, we can obtain the conclusion that our approach is suitable and has a better performance in nearly all situations.

Algorithm convergence: The experimental convergence effect of each group is similar. Therefore, the fifth experiment is used as an example to show the convergence effect of the algorithm, which is shown as Figure 8. In the figure, the x-axis' value represents the number of iterations. The y-axis' value represents the task processing time. Each point represents the average value of task processing time of all ants during one iteration. The number of iterations of the algorithm is 200, and the total number of ants in a colony is 1000. Each iteration generates 1000 task allocation schemes. After each iteration, a current optimal scheme is selected, and the pheromone concentration of the scheme is increased such that the probability of choosing this solution is high in the next iteration. We also adopt some ants in a random strategy to find a better solution. From Figure 8, we can see that when iterating about 100 times, a global optimal solution appears. We also conduct an experiment with the regular pheromone update rules, and the result is shown in Figure 9. From the figure, we can see that compared with our new algorithm, the regular algorithm falls into the local optimum faster, and cannot obtain the shortest transmission time. Our new algorithm converges to a lower time cost value, which shows that our new algorithm has better convergence.

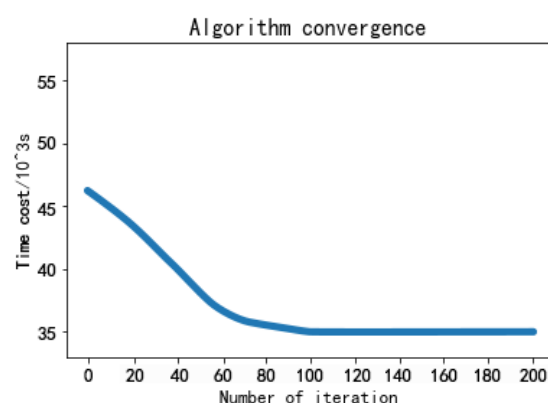


Figure 8. New ant colony algorithm convergence process.

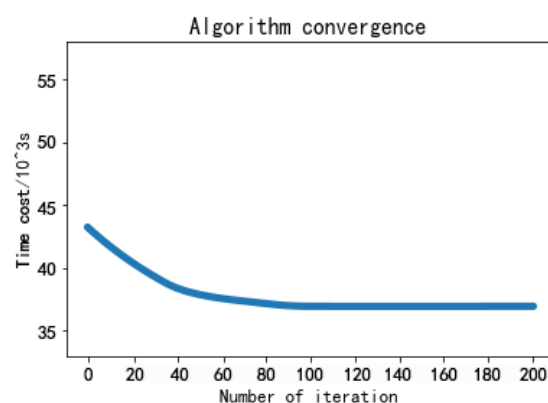


Figure 9. Regular ant colony algorithm convergence process.

7. Discussion

This paper presents an innovative data transmission scheduling optimization algorithm. The algorithm divides the data transmission into two links. While considering the single link optimization, the ant colony algorithm is used to optimize the whole of the two links. The limitation of the ant colony algorithm is that it is easy to fall into the local optimum. Our improved ant colony algorithm modifies the pheromone update rule, which solves the problem of too fast convergence compared with the existing algorithms. The experiment result is closer to the global optimal solution. The existing resource scheduling problems use ant colony algorithms for global optimization, but the optimization on the

downlink relay link and uplink relay link is not considered enough. This method includes SAF and the min–min algorithm in the ant colony algorithm, so the data not only have better path selection, but also have higher transmission efficiency in the downlink relay link and uplink relay link. However, the algorithm still has some shortcomings. The parameters of the algorithm have a certain correlation. The selection of parameters depends more on trial and error. Inappropriate initial parameters will weaken the optimization ability of the algorithm. This experiment is carried out in the simulation environment. Two data collection nodes, five data relay nodes and four data receiving nodes are set. When these initial conditions change, the adaptability of existing parameters to the new scene is not strong enough, and it takes a lot of time to re-adjust parameters to obtain the best optimization ability. In future work, our research direction is to add neural network to the ant colony algorithm, so the algorithm can adaptively adjust parameters through unsupervised learning such that the algorithm is better suitable for different application scenarios.

8. Conclusions

In this paper, we investigate the transmission network between edge devices and the cloud in the industrial internet field. The most important research topic is how to optimize the data transfer from the edge device to the cloud platform through relay nodes. Based on the real industrial background, we focus on how our scheduling algorithm can meet the time window constraints in extreme environments and achieve the shortest possible transmission time in the field of industrial internet.

The network model in this paper contains edge devices, relay nodes and the cloud. Due to the presence of relay nodes, the transmission and scheduling of data are divided into downlink relay links and uplink relay links. For better scheduling of the overall network, we design a novel scheduling algorithm based on the ant colony algorithm, called ACTS. The goal of optimization is the total transmission time. So this algorithm ensures that we are able to select the optimal task transmission path through the relay nodes on the overall network, which has the shortest total transmission time. Edge nodes generate a large number of tasks and the ants represent different task transmission paths through relay nodes in our algorithm. After iteration, the ants converge to the optimal path. In the process of modifying the ant colony algorithm, we also improve the pheromone update rule to avoid the algorithm from falling into a local optimum. The optimal solution is finally arrived at after several iterations.

In addition to the optimization of the overall network, we also focus on the scheduling of sub-links. The most important component of our framework is the multi-link optimization mechanism. The multi-link optimization mechanism is used to solve the constraint problem of each link. The downlink relay link transmission model based on store-and-forward and hierarchical analysis is used to segment the data. The uplink relay link transmission model based on the min–min algorithm and a local readjustment method is proposed to avoid bad load balancing. During each iteration of the ant colony algorithm, the SAF and min–min algorithms are used to evaluate the task transmission time on each sub-link.

Through extensive experimental tests, we build the simulation environment and generate data to simulate the real scenario. Based on the experimental results, we compare our designed method with existing methods and evaluate the effect of parameters. It proves the level of advancement and efficiency of the algorithm we propose.

Author Contributions: Conceptualization, R.C. and P.Z.; methodology, L.X. and P.X.; software, L.X. and Y.A.; validation, P.X.; formal analysis, L.X. and Y.A.; investigation, L.X.; resources, R.C.; data curation, L.X. and Y.A.; writing—original draft preparation, P.X.; writing—review and editing, P.X.; visualization, L.X. and Y.A.; supervision, P.Z.; project administration, P.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declared no potential conflict of interest with respect to the research, authorship, and/or publication of this article.

References

- Manigandan, S.K.; Manjula, S. Hybrid Scheduling Queue Model for Efficient Resource Allocation in Cloud Data Centers. *J. Comput. Theor. Nanosci.* **2018**, *15*, 1038–1043. [\[CrossRef\]](#)
- Yildirim, E.; Kim, J.; Kosar, T. How GridFTP Pipelining, Parallelism and Concurrency Work: A Guide for Optimizing Large Dataset Transfers. In Proceedings of the 2012 SC Companion: High Performance Computing, Networking Storage and Analysis, Salt Lake City, UT, USA, 10–16 November 2012; IEEE Computer Society: Washington, DC, USA, 2012; SCC '12, pp. 506–515. [\[CrossRef\]](#)
- Blamey, B.; Sintorn, I.; Hellander, A.; Toor, S. Resource- and Message Size-Aware Scheduling of Stream Processing at the Edge with application to Realtime Microscopy. *arXiv* **2019**, arXiv:1912.09088.
- Kai, C.H.; Xiao, Y.; Fang, Q. Relay Satellite Scheduling Based on Artificial Bee Colony Algorithm. *J. Electron. Inf. Technol.* **2015**, *37*, 635–640.
- Chen, H.; Zhou, Y.; Du, C.; Li, J. A satellite cluster data transmission scheduling method based on genetic algorithm with rote learning operator. In Proceedings of the 2016 IEEE Congress on Evolutionary Computation (CEC), Vancouver, BC, Canada, 24–29 July 2016; IEEE: Washington, DC, USA, 2016.
- Zhang, F.; Chen, Y.; Chen, Y. Evolving Constructive Heuristics for Agile Earth Observing Satellite Scheduling Problem with Genetic Programming. In Proceedings of the 2018 IEEE Congress on Evolutionary Computation (CEC), Rio de Janeiro, Brazil, 8–13 July 2018; IEEE: Washington, DC, USA, 2018. [\[CrossRef\]](#)
- Zhang, Z.; Hu, F.; Zhang, N. Ant colony algorithm for satellite control resource scheduling problem. *Appl. Intell.* **2018**, *48*, 3295–3305. [\[CrossRef\]](#)
- Liang, Z.; Liu, L.; Jiang, J.W.; Guo, J.; Zhang, Y.; Yan, H. Data relay system data download scheduling algorithm for earth observation satellites. In Proceedings of the 2017 IEEE International Conference on Communication, Networks and Satellite (Comnetsat), Semarang, Indonesia, 5–7 October 2017; IEEE: Washington, DC, USA, 2017. [\[CrossRef\]](#)
- Sreenivasulu, G.; Paramasivam, I. Hybrid optimization algorithm for task scheduling and virtual machine allocation in cloud computing. *Evol. Intell.* **2021**, *14*, 1015–1022. [\[CrossRef\]](#)
- Song, B.; Yao, F.; Chen, Y.; Chen, Y.; Chen, Y.W. A Hybrid Genetic Algorithm for Satellite Image Downlink Scheduling Problem. *Discret. Dyn. Nat. Soc.* **2018**, *2018*, 1531452. [\[CrossRef\]](#)
- Yi, Z.; Jiang, D.; Cao, L.; Du, X. *A Handover Decision Algorithm Based on Evolutionary Game Theory for Space-Ground Integrated Network*; Atlantis Press: Paris, France, 2019.
- He, Y.; Wang, L.; Zhan, Y.; Cao, S.; Luo, X. Dynamic Bandwidth Scheduling Algorithm for Space Applications in FC-AE-1553 Switching Network. In *Asia Communications and Photonics Conference*; Optical Society of America: Washington, DC, USA, 2018; p. Su2A-54.
- Nalini, J.; Khilar, P. Reinforced Ant Colony Optimization for Fault Tolerant Task Allocation in Cloud Environments. *Wirel. Pers. Commun.* **2021**, 1–19. [\[CrossRef\]](#)
- Tamilsenthil, S.; Kangaialmal, A. Chapman Kolmogorov and Jensen Shannon Ant Colony Optimization-Based Resource Efficient Task Scheduling in Cloud. In *Inventive Systems and Control*; Springer: Berlin/Heidelberg, Germany, 2021; pp. 911–926.
- Guangshun, L.; Jiping, W.; Junhua, W.; Jianrong, S. Data Processing Delay Optimization in Mobile Edge Computing. *Wirel. Commun. Mob. Comput.* **2018**, *2018*, 6897523.
- Zhang, X.; Shen, X.; Yu, Z. A Novel Hybrid Ant Colony Optimization for a Multicast Routing Problem. *Algorithms* **2019**, *12*, 18. [\[CrossRef\]](#)
- Shamsudin, H.; Yusof, U.K. Hybrid of Ant Colony Optimization-ANN for User Modeling System. *Adv. Sci. Lett.* **2018**, *24*, 1312–1315. [\[CrossRef\]](#)
- Luo, Z.; Zhao, X.; Xia, H. An ant colony algorithm and simulation for solving minimum MPR sets. *CAAI Trans. Intell. Syst.* **2011**, *6*, 166–171.
- Jung-Hyun, L.; Wang, S.M.; Chung, D.; Hee, K.K.; Jung, O. Multi-satellite control system architecture and mission scheduling optimization. In Proceedings of the IEEE Aerospace Conference, Big Sky, MT, USA, 3–10 March 2012.
- Nandagopal.; Puttaswamy. Lowering Inter-datacenter Bandwidth Costs via Bulk Data Scheduling. In Proceedings of the 2012 12th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (ccgrid 2012), Ottawa, ON, Canada, 13–16 May 2012.
- Peng.; Jian, G.; Liu.; Mingzhou.; Zhang.; Xi.; Ling.; Lin. Hybrid heuristic algorithm for multi-objective scheduling problem. *J. Syst. Eng. Electron.* **2019**, *30*, 327–342.
- Fox, K.; Im, S.; Moseley, B. Energy efficient scheduling of parallelizable jobs. *Theor. Comput. Sci.* **2018**, *726*, 30–40. [\[CrossRef\]](#)
- Jo, S.J.; Na, K.S.; Park, J.E.; Lee, M.S. Decision Making Regarding Key Elements of Korean Disaster Psychiatric Assistance Teams Using the Analytic Hierarchy Process. *Psychiatry Investig.* **2018**, *15*, 663–669. [\[CrossRef\]](#) [\[PubMed\]](#)

-
24. Patel, G.; Mehta, R.; Bhoi, U. Enhanced Load Balanced Min-min Algorithm for Static Meta Task Scheduling in Cloud Computing. *Procedia Comput. Sci.* **2015**, *57*, 545–553. [[CrossRef](#)]
 25. Wu, Y.; Qi, Z.; Jiang, L.; Zhang, C.; Xu, J. A Multi-objective Ant Colony Optimization Algorithm with Local Optimum Avoidance Strategy. In *Machine Learning for Cyber Security*; Springer: Cham, Switzerland, 2020.
 26. Castaing, J. Scheduling Downloads for Multi-Satellite, Multi-Ground Station Missions. *Small Satell. Conf.* **2014**, *8*, 33–45.