Network Energy-Saving Adjustment Routing Under Changing Demands: Models and Algorithms

MENGJIA ZHANG1, WENGUO YANG1, SUIXIANG GAO1, AND WEI XU2
1School of Mathematical Sciences, University of Chinese Academy of Sciences, Beijing 100049, China
2Ali Travel Network Technology (Beijing) Co., Ltd., Beijing 100102, China
Corresponding author: Wenguo Yang (yangwg@ucas.ac.cn)

This work was supported by the National Natural Science Foundation of China under Grant 11571015.

ABSTRACT Routing is key to ensuring the normal operation of a network. The rapid development of networks and the continuous expansion in scale have placed high demands on network energy-saving routing. In this study, we described the problem of network energy-saving adjustment routing under the condition of point-to-point demand changes. The influences of point-to-point demand changes on routing scheme adjustment and network energy consumption are considered, and optimal adjustment of the routing scheme when the demand in the network changes from one moment to another is studied. A 0-1 nonlinear integer programming routing adjustment model is established to adjust the costs and minimize network routing energy consumption. Furthermore, it is proven that the problem is NP-hard, therefore two algorithms are designed: a heuristic algorithm based on the depth-first search and a heuristic algorithm based on k-shortest paths. The validity of the routing adjustment model and the algorithm is verified for both a small network and a medium-to-large size network.

INDEX TERMS Network routing, network energy consumption, adjusting costs, demand changes.

I. INTRODUCTION

In recent years, the demand for networks has been increasing with the rapid development of the Internet, contributing to continuous expansion and so network energy consumption has come to the foreground [1], [2]. Information and Communication Technology (ICT) accounts for between 2% and 10% of global energy consumption and according to reports, will continue to grow at a rate of approximately 16% [3], [4]. In additions, carbon emissions caused by ICT accounts for approximately 2% of the global total and is predicted to double by 2020 [5], [6]. Among the available technologies, various communication devices in wired networks, such as routers, account for 22% of the total ICT energy consumption [7]. Since routers are the main equipment for networks, designing a proper energy management scheme for routers has become a focus of current scholarly interest [8]–[10].

To support the maximum amount of user access and to guarantee the Quality of Service (QoS) while supporting traffic, most existing networks have been over-provisioned. Studies have shown that the energy consumption of network devices and components, such as integrated chassis and line-cards, are not substantially affected by network loads [11]. Most network devices operate all day, and a large number of low-utilization network resources can consume substantial energy, necessitating network energy conservation. Router line-cards in networks can quickly switch from working mode to sleep mode [12]. Therefore, when network loads are low, devices or components with low utilization can be put into sleep mode, which will improve the utilization of resources in the network and reduce its total energy consumption.

In addition to network energy-saving routing, it is also necessary to consider the effects of different costs, such as network adjustment costs when the state of the network changes at certain moments. When requests change, the corresponding transmission path will also change. Therefore, how to reconfigure networks (that is, redistributing solutions) under certain requirements is also studied in this paper.

The key to minimizing the costs of network routing adjustments is to render changes in the request transmission path as small as possible. This process considers both the network...
adjustment costs and the fluctuations caused by changing the network routing. Therefore, it is crucial to consider the costs of network routing adjustments. This is also why this paper studies network energy-saving adjustment routing under request changes. Most of the existing literature on network energy consumption focus on given networks of requests, and consider only the minimization of network energy consumption to obtain a better request transmission method. However, in reality, when network usage transitions from one state to another, the network needs to be adjusted, incurring additional adjustment costs.

As to the transmission path of a request at the next moment in the network, it is obviously inappropriate to consider only the network energy consumption and adjustment costs incurred when the network adjusts at the next moment. In this case, the optimization goal is no longer just the minimization of energy consumption but a combination of network routing energy consumption and adjustment costs. Apparently, it is necessary to consider both two dimensions. This paper considers how to reasonably transmit new requests when the requests change in the network.

The main contributions of this paper are as follows:

1) A new focus of network energy-saving adjustment routing when network demand changes is proposed. When demands change in the network, considering energy-saving network routing and adjustment costs, we establish a 0-1 nonlinear integer programming model for energy-saving network adjustment routing. Based on the relationship between the 1-norm and linear programming, it is transformed into a mixed integer linear programming model.

2) The complexity result of the routing energy-saving adjustment problem is given. The theoretical analysis of the model proves that the problem is NP-hard.

3) Two efficient and practical solution algorithms are designed: a heuristic algorithm based on the depth-first search and another heuristic algorithm based on the k-shortest paths. The values of parameter k are experimentally analyzed, and the appropriate values are given. The effectiveness of the algorithm is verified for a small network and a medium-to-large network.

The rest of the paper is organized as follows; Section 2 addresses energy saving and path reconstruction and Section 3 describes the problem studied in this paper. By abstracting the network route into a directed weighted graph and establishing an extended network, we propose a 0-1 nonlinear integer programming model for the extended network, and a mixed integer linear programming model after transformation. Section 4 provides the theoretical analysis and proves that the problem is NP-hard. Section 5 details the two designed heuristic algorithms. Section 6 provides the experimental results. Finally, Section 7 provides conclusions.

II. RELATED WORK

Network routing optimization focuses in finding the best transmission path in network routing for requests given in the network. The two prerequisites are: first, the network we focus on is a given one; second, we know whether one or more constraints (such as demand and delay) are to be met. The network routing model can be represented by an undirected weighted graph \(G(V, E)\), where \(V\) represents a set of nodes and \(E\) represents a set of links. In this paper, our focus is on establishing the optimal transmission method under the constraint that the demand and delay are met when the request changes.

Gupta and Singh were the first to propose the issue of energy consumption for the Internet [13]. They discussed the impacts of energy-savings on network protocols by means of network device sleep and proposed two sleep modes: link-level uncoordinated sleep and component level coordinated sleep. Chabarek et al. [14] assumed that each request has an alternative path to save energy in the network by transferring traffic on the network to a subset of alternative paths. These authors established a hybrid integer programming model based on those known alternative paths, which were solved using CPLEX. Chiaraviglio et al. [15] turned the ISP's integrated chassis energy-saving problem into the Minimum Cost Multi-commodity Flow Problem (MCMF). When the demand between source and sink nodes is known, the energy consumption of routers and links in the network is minimized, and the heuristic algorithm is applied to solve the problem. Wang et al. [16] refined the internal structure of the router and the energy consumption proportion of each component, adjusted the goal of network energy conservation, and established a mathematical model for energy-saving optimization to minimize the total energy consumption of the integrated chassis and the line-cards. A heuristic algorithm is used to solve the problem.

Despite a lack of literature on network routing adjustment issues, some studies on path adjustments have been conducted. Clausen et al. [17] outlined the model for aircraft and crew scheduling problems. Visentini et al. [18] reviewed the recovery methods of vehicle scheduling in transportation services. Spliet et al. [19] studied the redistribution path problem in order to obtain a new solution under the constraint of satisfying some capacity so as to ensure the smallest possible sum of the costs of the scheme and the costs of deviating from the original plan. Eglese and Zambirinis [20] reviewed the application of disruption management in vehicle routing and scheduling for road freight transport. Keskin et al. [21] studied the Electric Vehicle Routing Problem with Time Windows (EVRPTW) and established a mixed integer linear programming model by minimizing vehicle costs, driver costs, energy costs and late penalties. A mathematical algorithm combining the meta-heuristic algorithm and mixed integer linear programming for adaptive large neighborhood search is proposed.

This paper considers the energy consumption of network routing and its adjustment cost, with a focus on how to
enable reasonable transmission of new demand changes in the network so as to minimize the weighted sum of network energy consumption and network adjustment costs.

III. PROBLEM FORMULATION

A. ASSUMPTIONS

Most of the energy consumption of working routers comes from the integrated chassis and line-cards. Starting from the internal structure of the router, this paper adopts the following assumptions concerning the problem description.

1) A router has numerous line-cards, each with multiple ports. Each port can be connected to any port of any line-card of any router to become a physical link. Only one physical link can be connected to each port.

2) A physical link with a capacity of \( C \) includes two links in both directions with a capacity of \( C \).

3) Any request can be entered and output from any port of any line-card. The incoming and outgoing ports of the line-card may be different.

4) If the traffic on the link passes, the corresponding port is used. The line-card cannot be shut down even with only one port on the line-card in use, and can only be shut down with no traffic on the line-card.

5) If the line-card is used, the corresponding integrated chassis must be used. Only when all the line-cards on the integrated chassis are turned off can the integrated chassis be turned off.

6) The traffic is not divertible, that is, a request can only be transmitted along one path.

B. PROBLEM DESCRIPTION

The router network is abstracted into a directed weighted graph \( G(N,A) \). Here \( N \) represents a set of router nodes, and \( A \) represents the set of arcs connecting routers. Each node in the set consumes energy, and each arc has its parameters such as its capacity \( C \) and delay \( \text{del} \). At time \( t \), under the condition that the existing request in the network satisfies the demand and delay constraints, the goal is to obtain an optimal solution that simultaneously minimize the energy consumption and establish a set of transmission paths for the request. When the request changes during period \( t+1 \), the changed request in the network needs to redetermine the transmission path. At this time, when redistributing the changed request path, we need to consider the adjustment costs caused by changes in transmission paths from time \( t \) to time \( t+1 \). The performance difference in network energy consumption at time \( t+1 \) (i.e., the difference between the actual network energy consumption at time \( t+1 \) and the energy consumption when the target is only energy consumption) so that the weighted sum of adjustment costs and the performance of the network energy consumption is minimized. The following are detailed descriptions of the problems using symbols.

In time period \( t \), there are \( D \) requests represented by \( \text{LSPs} = \{(s_i(t), e_i(t), d_i(t), \text{delay}_i(t)), i = 1, \cdots, D\} \), which are routed in the network, where \( s_i(t), e_i(t), d_i(t), \text{and delay}_i(t) \) represent the origin, destination, demand, and delay of the request \( i \), respectively. With the objective of minimizing energy consumption, the optimal transmission path \( P_i^*(t) \) for each request \( i \), the transmission paths of \( D \) requests \( P^*(t) = \{P_1^*(t), P_2^*(t), \cdots, P_D^*(t)\} \), the corresponding optimal solution \( X^*(t) \) and the optimal value \( f(X^*(t)) \) can be obtained. The request has changed during time period \( t+1 \), and the new requests are represented as \( \text{LSPs} = \{(s_i(t+1), e_i(t+1), d_i(t+1), \text{delay}_i(t+1)), i = 1, \cdots, D\} \). Each request has a new transmission path \( P_i(t+1) \) in the network. This path is equal to transmitting the new request after time period \( t+1 \) under the premise that the connection and the use of the integrated chassis, the line-card and the port in period \( t \) are known. Therefore, the sum of the performance of the network energy consumption and the adjustment costs to be minimized, since the dimensions are consistent.

C. EXPAND NETWORK

Since the original network cannot represent the detailed connection situation inside a router, this paper uses the ideas from Wang et al. [16] to construct the network in order to better establish the model of network energy-saving adjustment routing. That is, we consider each line-card of the router as a node and get the expanded network \( G' = (N', A') \) of the original network \( G = (N, A) \) as follows.

For each node \( v \) of the original network, we copy it \( n_v \) times, where \( n_v \) is the number of line-cards of router \( v \). We connect each copy node to \( v \) and obtain a star graph \( \text{Star}(v) \) with link costs of 0, capacity of \( \infty \) and time delays of 0. The center of the star represents the integrated chassis of router \( v \), and the leaves denote line-cards. By linking stars according to the connections of the line-cards in the original network and ensuring the capacity, costs and time delays of the original network are unchanged, we can obtain the expanded network \( G' = (N', A') \). We use \( A0 \) to denote the inside of the star graph, which is also the set of the integrated chassis and line-cards links. Here \( A1 \) denotes the outer side of the star graph, which is the set of the line-cards and line-cards links and \( N1 \) denotes all the duplicate nodes, which is the set of line-card. Then, we have \( |N'| = |N| + |N1| \) and \( |A'| = |A1| + |A0| \). To clearly show the topology of the expanded network, Figure 1 and 2 are provided as examples.

![FIGURE 1. Router network structure and internal topology.](image)

![FIGURE 2. The expanded network structure diagram of Figure 1.](image)
that assume that each router has two line-cards. Table 1 shows the notations used in this paper.

### D. LINEAR INTEGER PROGRAMMING ROUTING FORMULATION

To more clearly describe the network routing adjustment and energy saving problem when the request changes, this paper establishes a Linear Integer Programming Routing (LIPR) model that first minimizes the network energy consumption problem. The model is adapted from the research on network energy-saving routing in reference [16].

In the LIPR model, the goal is to minimize the energy consumption of the integrated chassis and the line-cards. The first and third constraints are the sleep conditions of the integrated chassis and line-cards, respectively. The second constraint is the flow conservation constraint, which also reflects the non-divertible flow. The fourth constraint is the capacity constraint. The fifth constraint is the delay constraint. The sixth constraint is an acyclic constraint. The last three constraints represent the 0-1 value of variables.

### E. NONLINEAR INTEGER PROGRAMMING ADJUSTMENT ROUTING FORMULATION

Note that the above LIPR problem’s decision variable is \( X \), and the corresponding optimal solution is \( X^* \). Then, the decision variable at time \( t+1 \) is \( X(t+1) \). Thus, \( X(t+1) = \{ y_j(t+1), x_v(t+1), x_u(t+1) \} \) if \( v, u \) \( \in \mathcal{N}_1 \), \( u, v \) \( \in \mathcal{N}_1 \), \( (u, v) \in \mathcal{A}_1 \), where \( \mathcal{N}_1 \) is the set of nodes, \( \mathcal{N}_1 \) is the set of line-cards, and \( \mathcal{A}_1 \) is the set of links.

Considering the adjustment costs of the network when the change is requested at time \( t+1 \) and the performance of the energy consumption at this time, we have the following Nonlinear Integer Programming Adjustment Routing (NLIPAR) formulation.

In the NLIPAR problem, the objective function \( M \) represents the adjustment costs that are required to adjust the network nodes or links. Here \( \beta \in [0, 1] \) is a proportional coefficient, and \( \alpha_1 \) and \( \alpha_2 \) are coefficients used to normalize the two types of targets, namely, the adjustment costs and performance differences.

In constraints (3) and (7), constraint (3) can be turned into (3’): \( x_{uv}(t+1) \leq \frac{1}{2}x_{uv}(t+1) + x_{uv}(t+1), \forall (u, v) \in \mathcal{A}_1 \). Then, the number of constraints is reduced from \( D[A1] \) to \( 2|A1| \). Since \( f^*_{u,v} \) is the optimal value of the LIPR problem when the request is \( D(t+1), D(t+1) \), \( f^*_{u,v} \) is the optimal value of the LIPR problem when the request is \( D(t+1), f^*_{u,v} \) is the optimal value of the LIPR problem when the request is \( D(t+1), D(t+1) \), \( f^*_{u,v} \) is the optimal value of the LIPR problem when the request is \( D(t+1), D(t+1) \). Because the performance difference at time \( t+1 \) is considered, the acyclic constraint has no effect on the optimal value of the considered problem. Similarly, using the method from reference [22], the acyclic constraint (6) can be removed. In addition, given a network, the costs of adjusting the links and points of the network may be different. For a more general form, this paper changes the adjustment costs \( M \) \( ||X(t+1) - X^*(t)|| \) to \( ||W(X(t+1) - X^*(t))|| \). Through the above simplification, the NLIPAR1 model is obtained, where \( ||W(X(t+1) - X^*(t))|| \) is specifically as follows.

Suppose the adjustment costs of the integrated chassis are \( w_1 \), the adjustment costs of the line-cards are \( w_2 \), and the adjustment cost of the links of line-cards are \( w_3 \). Then \( ||W(X(t+1) - X^*(t))|| = \sum_{j \in \mathcal{N}_1} y_j(t+1) - y^*_j(t+1) \) and \( \sum_{v \in \mathcal{N}_1} w_2 |x_v(t+1) - x^*_v(t)| \) and \( \sum_{(u, v) \in \mathcal{A}_1} w_3 |x_{uv}(t+1) - x^*_{uv}(t)| \).

Since the nonlinear programming \( \min_{x \in \mathbb{R}^m} \| x \| \) can be converted to linear programming, we have the following equivalent forms:

\[
\min_{y, z \in \mathbb{R}^m} \quad y + z
\]
\[
\text{s.t.} \quad y - z = x
\]
\[
\quad y \geq 0
\]
\[
\quad z \geq 0
\]

Here, \( y \in \mathbb{R}^m; z \in \mathbb{R}^m \); \( y \) takes all positive elements of \( x \), and the rest are 0; \( z \) takes all negative elements of \( x \), and the rest are 0. For example, \( x = [1, 2, 4, 0, 0, 0, 0] \), and then \( y = [1, 2, 4, 0, 0, 0, 0] \).

Based on the above discussion, the network adjustment costs can be converted to a linear form.

Note: \( \{y_j(t+1), j \in \mathcal{N}_1\} \) as \( \{y_1(t+1), y_2(t+1), \cdots, y_{|\mathcal{N}_1|}(t+1)\} \); \( \{y^*_j(t), j \in \mathcal{N}_1\} \) as \( \{y^*_1(t), y^*_2(t), \cdots, y^*_{|\mathcal{N}_1|}(t)\} \); \( x_1(t+1) \), \( x_2(t+1) \), \( \cdots, x_{|\mathcal{N}_1|}(t+1) \), \( x^*_1(t) \), \( x^*_2(t) \), \( \cdots, x^*_{|\mathcal{N}_1|}(t) \) \( \in \mathcal{N}_1 \) as \( \{x^*_1(t), x^*_2(t), \cdots, x^*_{|\mathcal{N}_1|}(t)\} \); \( \{x_v(t+1), (u, v) \in \mathcal{A}_1\} \) as \( \{x^*_v(t), x^*_v(t), \cdots, x^*_{|\mathcal{N}_1|}(t)\} \); \( \{x^*_{uv}(t), (u, v) \in \mathcal{A}_1\} \) as \( \{x^*_uv(t), x^*_uv(t), \cdots, x^*_{|\mathcal{N}_1|}(t)\} \). Let:

\[
P_k = \begin{cases} y_k(t+1) - y^*_k(t), \\ k = 1, 2, \cdots, |\mathcal{N}_1|; \\ x_{k-N}(t+1) - x^*_{k-N}(t), \\ k = |\mathcal{N}_1| + 1, \cdots, |\mathcal{N}_1| + |\mathcal{N}_1|; \\ x_{k-N}(t+1) - x^*_{k-N}(t), \\ k = |\mathcal{N}_1| + |\mathcal{N}_1| + 1, \cdots, |\mathcal{N}_1| + |\mathcal{N}_1| + |\mathcal{A}_1|. \\
\end{cases}
\]
\[
\begin{align*}
\min & \quad \sum_{j \in N} PC_y y_j + \sum_{v \in N_1} PC_x x_v \\
\text{s.t.} & \quad \frac{1}{|N_1|} \sum_{v \in \text{star}(j)} x_v \leq y_j, \quad \forall j \in N \\
& \quad \sum_{v \in N^+(u)} x^i_{uv} - \sum_{v \in N^-(u)} x^i_{uv} = \begin{cases} 
1, & u = s_i \\
-1, & u = e_i, \quad \forall u \in N', i = 1, \ldots, D \\
0, & \text{else}
\end{cases} \\
& \quad x^i_{uv} \leq \frac{1}{2} (x_u + x_v), \quad \forall (u, v) \in A_1, i = 1, \ldots, D \\
& \quad \sum_{i=1}^D x^i_{uv} d_i \leq C_{uv}, \quad \forall (u, v) \in A_1 \\
& \quad \sum_{(u, v) \in A_1} x^i_{uv} \text{delay}_{uv} \leq \text{delay}_i, \quad i = 1, \ldots, D \\
& \quad \sum_{v:(u, v) \in A_1} x^i_{uv} \leq 1, \quad \forall u \in N \setminus \{s_i, e_i\}, i = 1, \ldots, D \\
& \quad y_j \in \{0, 1\}, \quad \forall j \in N \\
& \quad x_v \in \{0, 1\}, \quad \forall v \in N_1 \\
& \quad x^i_{uv} \in \{0, 1\}, \quad \forall (u, v) \in A_1, i = 1, \ldots, D
\end{align*}
\]

\[
\begin{align*}
\min & \quad (1 - \beta) \alpha_1 M \|X(t + 1) - X^*(t)\|_1 + \beta \alpha_2 |f(X(t + 1), D(t + 1)) - f(X^*(t + 1), D(t + 1))| \\
\text{s.t.} & \quad \frac{1}{|N_1|} \sum_{v \in \text{star}(j)} x_v(t + 1) \leq y_j(t + 1), \quad \forall j \in N \\
& \quad \sum_{v \in N^+(u)} x^i_{uv}(t + 1) - \sum_{v \in N^-(u)} x^i_{uv}(t + 1) = \begin{cases} 
1, & u = s_i(t + 1) \\
-1, & u = e_i(t + 1), \quad \forall u \in N', i = 1, \ldots, D \\
0, & \text{else}
\end{cases} \\
& \quad x^i_{uv}(t + 1) \leq \frac{1}{2} (x_u(t + 1) + x_v(t + 1)), \quad \forall (u, v) \in A_1, i = 1, \ldots, D \\
& \quad \sum_{i=1}^D x^i_{uv}(t + 1) d_i(t + 1) \leq C_{uv}, \quad \forall (u, v) \in A_1 \\
& \quad \sum_{(u, v) \in A_1} x^i_{uv}(t + 1) \text{delay}_{uv}(t + 1) \leq \text{delay}_i(t + 1), \quad i = 1, \ldots, D \\
& \quad \sum_{v:(u, v) \in A_1} x^i_{uv}(t + 1) \leq 1, \quad \forall u \in N \setminus \{s_i(t + 1), e_i(t + 1)\}, i = 1, \ldots, D \\
& \quad \frac{1}{D} \sum_{i=1}^D x^i_{uv}(t + 1) \leq x_{uv}(t + 1), \quad \forall (u, v) \in A_1 \\
& \quad y_j(t + 1) \in \{0, 1\}, \quad \forall j \in N \\
& \quad x_v(t + 1) \in \{0, 1\}, \quad \forall v \in N_1 \\
& \quad x^i_{uv}(t + 1) \in \{0, 1\}, \quad \forall (u, v) \in A', i = 1, \ldots, D \\
& \quad x_{uv}(t + 1) \in \{0, 1\}, \quad \forall (u, v) \in A_1
\end{align*}
\]
Then, the adjusting cost function:
\[ \| W(X(t+1) - X^*(t)) \|_1 \]
is equivalent to formulation LP.

Note: \( g(X(t + 1)) = \sum_{k=1}^{[|N|]} w_1(p_k + q_k) + \sum_{k=[|N|]+1}^{[|N|]+[|N|]+[|A|]} w_2(p_k + q_k) + \sum_{k=[|N|]+[|N|]+[|A|]}^{[|N|]+[|N|]+[|A|]} w_3(p_k + q_k) \). The NLIPAR1 model is equivalent to the following Mixed Integer Linear Programming Adjustable Routing (MILPAR) model:

\[
\begin{align*}
\min & \quad \alpha_1 f(X(t + 1)) \\
\text{s.t.} & \quad \text{(1) - (2) (3') (4) - (16)} \\
\end{align*}
\]

In the MILPAR model, the number of independent variables is \( 2(|N|) + |N1| + |A1| + (1 + m + 2mD)|N| + (1 + 2D)|A| = 2(|N| + m|N| + |A|) + (1 + m + 2mD)|N| + (1 + 2D)|A| = 3(|N| + m|N| + |A|) + (1 + m + 2mD)|N| + (1 + 2D)|A| \), and the number of constraints is \( |N| + |N1| + |A1| + (1 + m)D + 4mD + 4m^2|N|^2 \). If \( m > |N| \), the number of independent variables is \( O(m^3|N|) \), and the number of constraints is \( O(m^3|N|) \). If \( m > |N| \), the number of independent variables is \( O(m^3) \), and the number of constraints is \( O(m^3|N|) \). If \( m > |N| \), the number of independent variables is \( O(m^3|N|) \), and the number of constraints is \( O(m^3|N|) \).

And the number of constraints is \( O(|D|) \). If \( D > |N| \), the number of independent variables is \( O(|D|^2|N|^2) \), and the number of constraints is \( O(|D|^2|N|^2) \). If \( D > m \) and \( D > |N| \), the number of independent variables is \( O(|D|^4) \), and the number of constraints is \( O(|D|^4) \).

\[ \begin{align*}
\min & \quad \alpha_2 f(X(t + 1)), D(t + 1) - f^*_t(t+1) \\
\text{s.t.} & \quad \text{(1) - (2) (3') (4) - (16)} \\
\end{align*} \]

This function is equivalent to

\[ \begin{align*}
\min & \quad \alpha_2 f(X(t + 1)), D(t + 1) - f^*_t(t+1) \\
\text{s.t.} & \quad \text{(1) - (2) (3') (4) - (16)} \\
\end{align*} \]

In the LIPR1 problem, different requests \( i = 1, \ldots, D \) are regarded as different commodities \( i = 1, \ldots, D \) in the Multi-Commodity Flow problem, and the non-divertible constraint is regarded as fixed and indivisible for each commodity. The capacity constraint in the LIPR1 problem is equivalent to the capacity limitation in the Multi-Commodity Flow problem, and the demand of each request is considered to be the amount of transportation of each item. In addition, transportation costs of items are considered to be 0. When the energy consumption of the integrated chassis is recorded as the fixed costs of the local construction costs, and the energy consumption of the line-card is recorded as the variable costs of the local construction costs, then the energy consumption of the integrated chassis and line-cards is minimized. The energy consumption is equivalent to the construction costs.
in places where the transportation of goods is minimized. Therefore, we reduce the Multi-Commodity Flow problem with capacity limitations to the LIPIR1 problem under polynomial time. Since the Multi-Commodity Flow problem with the capacity limitation is NP-hard, the LIPIR1 problem is an NP-hard problem. Thus, NLIPAR1 is NP-hard.

For ease of description, we note the following:

\[ g_1(X(t + 1)) = (1 - \beta)\alpha_1 g(X(t + 1)) = (1 - \beta)\alpha_1 \|W(X(t + 1) - X^*(t))\|_1, \]
\[ f_1(X(t + 1)) = \beta\alpha_2 f(X(t + 1), D(t(1)) - f^*_1), \]
and
\[ h(X(t + 1)) = g_1(X(t + 1)) + f_1(X(t + 1)). \]

Note that \( X^{**}(t + 1) \) is the optimal solution to the model NLIPAR1. Also note that \( Paths = \{P_1, P_2, \ldots, P_D\} \) are the transmission paths of the D requests corresponding to any feasible solution \( X(t + 1) \) of this problem.

**Definition 1 (1-Neighborhood):** For any feasible solution \( X(t + 1) \) of the problem, we have the corresponding transmission paths \( P_1, P_2, \ldots, P_D \). If only one transmission path is changed at a time, and the other \( D - 1 \) paths remain unchanged, all the feasible solutions consisting of new \( D \) paths are called the 1-neighborhood of \( X(t + 1) \).

**Definition 2 (The Descent Direction of the Function \( h(X(t + 1)) \)):** Let \( X^{**}(t + 1) \) be any feasible solution in the 1-neighborhood of \( X(t + 1) \), and let \( \Delta g_1 = g_1(X^{**}(t + 1)) - g_1(X(t + 1)) \) and \( \Delta f_1 = f_1(X^{**}(t + 1)) - f_1(X(t + 1)) \). If \( \Delta g_1 > \Delta f_1 \), then, in \( X^{**}(t + 1) \) and \( X(t + 1) \), there are corresponding paths of \( Paths^{**} \) and \( Paths^* \), respectively, and the request with a different path is a descent direction of \( h(X(t + 1)) \).

**Theorem 2:** For the optimal solution \( X^{**}(t + 1) \) of the function values corresponding to all feasible solutions in the 1-neighborhood will not be smaller than the values corresponding to \( X^{**}(t + 1) \).

**Proof:** Suppose that the optimal paths of the \( D \) requests corresponding to the optimal solution \( X^{**}(t + 1) \) is \( Paths^{**} = \{P_1^{**}, P_2^{**}, \ldots, P_D^{**}\} \), a feasible solution in any 1-neighborhood can be recorded as \( X^{**}(t + 1) \), and the corresponding paths are \( Paths^* = \{P_1^*, P_2^*, \ldots, P_D^*\} \), that is, only one path is changed.

Since the global optimal solution must be a local optimal solution, any feasible solution \( X(t + 1) \) in the 1-neighborhood of the optimal solution \( X^{**}(t + 1) \) must have \( h(X^{**}(t + 1)) \geq h(X(t + 1)) \), that is, \( h(Paths^{**}) \geq h(Paths^*) \).

Based on Theorem 2, the nature of the problem that is studied in this paper is analyzed. It can be seen from the above analysis that \( X(t + 1) \) is the optimal solution of \( f(X(t + 1), D(t + 1)) \), and \( X^*(t + 1) \) is also the optimal solution of \( f_1(X(t + 1)) \). If \( X^*(t + 1) \) is also the optimal solution of \( g_1(X(t + 1)) \), since \( h(X(t + 1)) = g_1(X(t + 1)) + f_1(X(t + 1)) \), then \( X^*(t + 1) \) is the optimal solution of \( h(X(t + 1)) \).

Otherwise, \( X^*(t + 1) \) is not the optimal solution of \( g_1(X(t + 1)) \). Then, at this time, the value of \( g_1(X^*(t + 1)) \) may be large, and the solution \( X^*(t + 1) \) needs to be improved. When improving the solution \( X^*(t + 1) \), a descent direction of the function \( h(X(t + 1)) \) is found. In addition, in the 1-neighborhood of \( X^*(t + 1) \), the feasible solution that improves \( h(X(t + 1)) \) is the most a new solution of \( h(X(t + 1)) \).

Then, the same analysis is performed for the feasible solution of the new solution iteratively.

Based on Theorem 2, two heuristic algorithms for solving this problem are designed.

**V. ALGORITHMS**

The optimal solution \( X^*(t + 1) \) of the LIPIR problem at time \( t + 1 \) is also a feasible solution to the NLIPAR1 problem. Using Theorem 2, the optimal solution exists in the 1-neighborhood of the feasible solution. Since the feasible solution \( X^*(t + 1) \) only considers the network energy consumption without considering the cost of network adjustment, we start from the perspective of the solution with the least energy consumption to find the improved solution in the 1-neighborhood. If a feasible solution that satisfies demand and delay constraints can be found and adjustment costs are less than the initial adjustment costs, the energy consumption of the new feasible solution will be calculated. When the reduced value of the adjustment costs of the new feasible solution is more than the value that is added by the energy consumption, the new feasible solution will replace the original feasible solution and iterate until no better feasible solution can be found. Therefore, it finds an optimal solution \( X^{**}(t + 1) \) for this problem.

**A. HEURISTIC ALGORITHM BASED ON DEPTH-FIRST SEARCH**

**Step 1:** Given the initial feasible solution \( Paths = \{Path(X^*(t + 1)), h_{best} = h(Paths) \), num = 1, and precision \( \epsilon \), then we can obtain \( g_1(Paths) \), \( f_1(Paths) \), and \( h(Paths) \).

**Step 2:** If \( num \leq n \), go to Step 3; otherwise, stop, and then record \( h_{best} \) and \( Paths \).

**Step 3:** Calculate \( i^* = \arg\min_{\forall i \leq D} g(Paths - P_i) \).

**Step 4:** Reassign the path \( P_{i^*} \) for \( i^* \). The depth-first algorithm is used to find the request \( i^* \), which satisfies the delay threshold, demand requirements, and \( \epsilon \). Then, we obtain \( P_{i^*} = \arg\max_{\forall P \in \text{Paths}} g_1(Paths - g_1((Paths - P_{i^*}) \cup P) - f_1((Paths - P_{i^*}) \cup P) - f_1(Paths)) \), which is the path reassignment for request \( i^* \).

**Step 6:** Note \( Paths = (Paths - P_{i^*}) \cup P_{i^*} \), and calculate \( h(Paths) \) and \( h_{best} = h(Paths) \), and then record \( h(Paths) \) and \( h_{best} \). Otherwise, update \( Paths \) and \( h_{best} \). Then, let \( Paths = Paths_{p}, h_{best} = h(Paths_{p}), \) and \( num = num + 1 \), and return to Step 2.

In step(2), \( n \) is the maximum number of iterations. Step(3) calculates the request \( i^* \) with the greatest change in the network adjustment costs. Step(4) is a process of reallocating the path for \( i^* \). In the remaining network \( G_{i^*} \), the capacity of each link is \( C_{uv} = C_{uv} - (Paths - P_{i^*}) \) and it occupies the traffic.
of \( uv \). Then, the depth-first search method is used to find all feasible solutions in the current 1-neighborhood.

B. HEURISTIC ALGORITHM BASED ON K-SHORTEST PATHS

Algorithm 1 differs from Algorithm 2 as different methods are used to find a new feasible solution for the 1-neighborhood. Algorithm 1 uses the depth search to find all feasible paths from the origin to the destination. Algorithm 2 uses the k-shortest paths [23] [24] idea to find k feasible paths from the origin to the destination. \( k \) is an adjustable parameter, referring to the number of alternative candidate paths. Therefore, the heuristic algorithm based on the k-shortest paths is less complex than the heuristic algorithm based on the depth-first search.

Step1: Given the initial feasible solution \( \text{Paths} \doteq \text{Path}(X^*(t + 1)) \), denote \( h_{best} = h(\text{Paths}) \), \( num = 1 \) and precision \( \epsilon \). Then, we can formulate \( g_1(\text{Paths}), f_1(\text{Paths}) \) and \( h(\text{Paths}) \).

Step2: If \( num \leq n \), go to Step 3; otherwise, stop, and then record \( h_{best} \) and \( \text{Paths} \).

Step3: Calculate \( i^* = \arg\min_{1 \leq i \leq D} g(\text{Paths} - P_i) \).

Step4: Reassign the path \( P^{i}_{r} \) for \( i^* \). It would be better for \( i^* \) to use the opening links in \( X^*(t) \). The weights of each link opened in the network of \( X^*(t) \) are set as 0, and those that have not been opened are set as 1. Then, the \( k \) paths of request \( t^{i*} \) are obtained by the k-shortest path algorithm for \( p_1, p_2, \ldots, p_k \).

Step5: \( \forall P \in \{p_1, p_2, \ldots, p_k\} \). For those paths \( P \) where \( g((\text{Paths} - P_r) \cup P) < g(\text{Paths}) \), calculate \( g_1((\text{Paths} - P_r) \cup P), f_1((\text{Paths} - P_r) \cup P) \) and \( g_1((\text{Paths} - P_r) \cup P) - f_1((\text{Paths} - P_r) \cup P) - f_1((\text{Paths} - P_r) \cup P) \). Then, we obtain \( P^{*}_{r} = \arg\min_{P \in P} g_1((\text{Paths} - P_r) \cup P) - f_1((\text{Paths} - P_r) \cup P) - f_1((\text{Paths} - P_r) \cup P) \), which is the path reassignment for request \( t^{i*} \).

Step6: Note \( \text{Paths'} = (\text{Paths} - P_r) \cup P^{*}_{r} \) and calculate \( h(\text{Paths'}) \) and \( \Delta h = h_{best} - h(\text{Paths'}) \). If \( \Delta h < \epsilon \), stop, and then record \( h(\text{Paths'}) \) and \( \text{Paths'} \). Otherwise, update \( h_{best} \) and \( \text{Paths} \). Let \( \text{Paths} = \text{Paths'} \), \( h_{best} = h(\text{Paths'}) \) and \( num = num + 1 \), and return to Step 2.

Step(4) is a new network that is constructed according to the network usage at time \( t \) and the available capacity of the network at time \( t + 1 \). Using the idea of the k-shortest paths in the new network, it can find \( k \) feasible paths for request \( t^i \).

VI. EXPERIMENTAL EVALUATION

This paper uses C# to generate the network topology and request LSPs and simulates the effectiveness of the designed algorithm on three different sized synthetic networks. With the assumption that each integrated chassis has 8 line-cards, the number of links in the composite network topology is randomly assigned between 5 and 10 and each request demand is a random number between 0 and 5. This paper adopts the designed heuristic algorithms using the operating environment of Microsoft Visual Studio by C#.

To verify the effectiveness of the proposed algorithm, the solutions that are obtained by the depth-first search-based heuristic algorithm and the k-shortest-path based heuristic algorithm are compared with the initial solution \( h_0 \).

Suppose that the energy consumption of the line-cards is the same and is a random number between 5 and 10, the energy consumption of the different integrated chassis is the same and is a random number between 30 and 50, the adjustment costs of the integrated chassis are 10, the adjustment costs of the line-cards are 5, the link adjustment costs are 2, and all routers can send and receive requests. The simulation is aimed at minimizing the sum of the network energy consumption and adjustment costs.

The synthetic network sizes are 20, 50, 80, and 100, respectively. This paper takes a 50-node network as an example. Figure 3 shows the expanded network. The example contains 400 line-cards and 10 requests.

FIGURE 3. 50-node expanded network.

Table 2 shows the optimal values, initial values and optimal percentages of initial values for algorithm 1 when \( \beta \) takes on different values in the extended network with 10 requests and 50 nodes.

| \( \beta \) | Initial Value \( h_0 \) | Optimal Value \( h_{best}^1 \) | \( \frac{h_0 - h_{best}^1}{h_0} \) |
|---|---|---|---|
| 0 | 4560.000 | 1200.000 | 97.39% |
| 0.1 | 4104.000 | 1306.668 | 96.82% |
| 0.2 | 3648.000 | 1413.336 | 96.13% |
| 0.3 | 3192.000 | 1520.004 | 95.24% |
| 0.4 | 2736.000 | 1626.727 | 94.95% |
| 0.5 | 2280.000 | 1733.400 | 92.40% |
| 0.6 | 1824.000 | 1840.008 | 89.91% |
| 0.7 | 1368.000 | 1946.676 | 85.77% |
| 0.8 | 912.000 | 2053.344 | 77.48% |
| 0.9 | 456.000 | 2160.012 | 52.63% |

It can be seen from Table 2 that different \( \beta \) values result in different initial values \( h_0 \) and optimal values \( h_{best}^1 \). When \( \beta = 0 \), the objective function value \( h \) only considers the network adjustment costs. Since the initial solution is only the optimal solution when considering energy consumption, the network adjustment costs are the largest at this time. That is, the initial value \( h_0 \) is the largest. As \( \beta \) gradually increases, which means that the proportion of the network adjustment costs is gradually decreasing, the initial value \( h_0 \) is gradually decreasing. For the optimal value \( h_{best}^1 \), since \( \beta = 0 \), the objective function \( h \) is equivalent to considering only the adjustment costs.
of the network, and the difference in the energy consumption is not considered. Moreover, Algorithm 1 is more concerned with the adjustment costs of the network. Therefore, Algorithm 1 works well, since it provides the optimal results. With the gradual increase of $\beta$, the proportion of network adjustment costs gradually decreases. The optimization effect of Algorithm 1 is gradually weakened, but overall, Algorithm 1 has a good performance on the initial solution’s optimization improvement. Since most of the literatures focus on energy consumption, few studies have combined routing adjustment costs with energy consumption. However, the adjustment costs sometimes will be very large, so this paper studies the combination of the two. Since the adjustment costs account for a larger proportion, the value of $\beta$ is less than 0.5 in more cases. In those cases, the optimization effect of Algorithm 1 on the initial solution is over 94%, and so Algorithm 1 is very effective on solving this problem.

For the sake of simplicity, in the following simulation results presented use $\beta = 0.3$ as an example. Table 3 shows the optimal value $h_1^{\text{best}}$, the running time and the optimized ratio to the initial value that is obtained from simulations using Algorithm 1 with different network sizes.

As seen from Table 3, in general, as the network scale increases, the running time to find the optimal value required for Algorithm 1 becomes longer. In the Table 3, the running time of the network which is with 80 integrated chassis, 640 line-cards and 20 requests, is longer than the one which includes 100 integrated chassis, 800 line-cards and 20 requests. Since the network is randomly generated, the number of links is random, and the initial point of the request is also random. When the link of the network and the request generation are better, the algorithm will run very fast, so this makes sense. But in the worst case, the larger the network, the longer it takes to run.

Since Algorithm 1 is designed based on depth-first search, its complexity is high. When the value of parameter $k$ in Algorithm 2 is appropriate, the complexity of Algorithm 2 is much lower than that of Algorithm 1. Table 4 below shows the simulation results for different $k$ values in Algorithm 2 for medium-to-large size networks with 50 integrated chassis, 400 line-cards, 10 requests and small-scale networks with 100 integrated chassis, 800 line-cards and 50 requests.

It can be seen from Table 4 that when considering the network optimal solution and running time, it is more suitable to use Algorithm 1 for small size networks and Algorithm 2 for medium-to-large size networks. For large-scale networks, the depth-first search based on Algorithm 1 is very large, thus it takes more time and memory. After the program runs for 60 hours, there is still no feasible solution obtained. Considering the initial solution corresponding to different network sizes in Table 3 and the optimal solutions that are obtained by different methods and different $k$ values, it can be seen that the running time of Algorithm 2 based on $k$-shortest paths increases as $k$ increases. However, when $k > 5$, as $k$ gradually increases further, the improvement of the optimal solution is not significant, although the running time has increased a lot, as far the given network size is concerned, it is most suitable when $k = 5$. If a better feasible solution would be obtained, $k = 1$ should be a great choice. Because the running time is short, and a better solution than the initial solution can be found. Therefore, according to different concerns, different algorithms and different parameters can be selected for experiments to achieve the desired results. Table 5 is the optimal values of the corresponding network and data in Table 3 that are calculated by Algorithm 2 when $k = 5$. From the comparison of the running time of Algorithm 1 and Algorithm 2 in Table 4, it can be seen that medium-to-large size networks are more suitable for Algorithm 2. Table 5 only gives the calculation results of Algorithm 2 and table 4 gives the running time comparison, and Table 5 is no longer shown.
VII. CONCLUSION

This paper studies how to transmit new requests when demands change in the network so that network energy consumption and adjustment costs are minimized. The expanded network is designed by analyzing the internal structure and the connection of the router. In an expanded network, the goal is to minimize the sum of network adjustment costs and network energy consumption which can be achieved by switching more idle devices to sleep mode through rerouting and aggregating network traffic. In this paper, the 0-1 integer programming model, which minimizes network adjustment costs and the energy consumption sum for the integrated chassis and line-cards in the expanded network, is presented and it is proven that this problem is NP-hard. Therefore, following optimality analysis, two heuristic algorithms are designed, the heuristic algorithm based on the depth-first search and the heuristic algorithm based on the k-shortest paths, and the effectiveness of the algorithms is verified using a synthetic network.