Journal of Southeast University (English Edition) Vol. 41, No. 1, pp. 91-100 Mar. 2025 ISSN 1003-7985

Novel two-stage preflow algorithm for solving the maximum flow problem in a network with circles

DANG Yaoguo, HUANG Jinxin, DING Xiaoyu, WANG Junjie

(College of Economics and Management, Nanjing University of Aeronautics and Astronautics, Nanjing 211100, China)

Abstract: The presence of circles in the network maximum flow problem increases the complexity of the preflow algorithm. This study proposes a novel two-stage preflow algorithm to address this issue. First, this study proves that at least one zero-flow arc must be present when the flow of the network reaches its maximum value. This result indicates that the maximum flow of the network will remain constant if a zero-flow arc within a circle is removed; therefore, the maximum flow of each network without circles can be calculated. The first stage involves identifying the zero-flow arc in the circle when the network flow reaches its maximum. The second stage aims to remove the zero-flow arc identified and modified in the first stage, thereby producing a new network without circles. The maximum flow of the original looped network can be obtained by solving the maximum flow of the newly generated acyclic network. Finally, an example is provided to demonstrate the validity and feasibility of this algorithm. This algorithm not only improves computational efficiency but also provides new perspectives and tools for solving similar network optimization problems.

Key words: network with circles; maximum flow; zero-flow arc; two-stage preflow algorithm

DOI: 10. 3969/j. issn. 1003-7985. 2025. 01. 012

The maximum flow problem, a crucial issue in network optimization, has a wide range of engineering and management applications[1]. The goal of the maximum flow problem is to determine the maximum flow through a network while adhering to arc capacity and node equilibrium constraints. This problem is a crucial aspect of network optimization theory and is widely applied to simulate real-life decision scenarios in some fields, such as transportation, biology, medicine, and economics[2-3]. Mirzaei et al. [4] explored the maximum flow network interdiction problem in uncertain environments under information asymmetry conditions. Over half a century, the network maximum flow problem has led to the development of various algorithms[5-6]. Mehryar et al. [7] built a network flow model to solve the reliability allocation problem. Shi et al. [8] evaluated railway traffic based on the Decision Making Trial and Evaluation Laboratory (DEMATEL), analytic hierarchy process (AHP), and analytic network process (ANP) methods. Alipour et al. [9] proposed a new method for solving the maximum flow of a network and verified the effectiveness of the model using instance space analysis. These algorithms are generally categorized into two types, namely feasible flow and preflow advancement algorithms. (1) The feasible flow algorithm was proposed by Ford and Fulkerson in 1956. This algorithm determines whether the network has augmented chains and identifies these chains. The core idea is to use the labeling algorithm to identify extensive chains in the network[10]. In 1989, Ahuja et al. [11] utilized the distance signature concept introduced by Goldberg and Tarjan to construct incremental tracks while retaining distance information from the previous construction through re-signatures. Huang[12] developed a recursive algorithm to generate all feasible flow vectors satisfying at least, with integer-type flow representing undivided demands. (2) Preflow advancement algorithms differ from feasible flow algorithms. The preflow advancement algorithm focuses on the remaining network and works along the edges to facilitate maximum flow advancement until no further progress can be made. In 1974, Karzanov[13] treated the acyclic graph blocking flow as a separate problem and established the preflow concept to address it. Kara et al. [14] introduced a shared-memory parallel push-tagging algorithm that improves the rate of graph coloring on sparse networks. Deutsch et al. [15] developed MineFlow, an open-source C++ library offering efficient and flexible precedence schemes along with a simplified pseudoflow-based solver.

Many scholars have approached the network maximum flow problem from the following perspectives:

(1) Linear programming algorithms. The maximum flow problem is a specific case of linear programming problem. Holzhauser et al. [16] proposed a specialized network simplex algorithm by extending the traditional mini-

Received 2024-08-02, **Revised** 2024-10-08.

Biography: Dang Yaoguo (1964—), male, doctor, professor, iamdangyg@163. com.

Foundation items: The National Natural Science Foundation of China (No. 72001107, 72271120), the Fundamental Research Funds for the Central Universities (No. NS2024047, NP2024106), the China Postdoctoral Science Foundation (No. 2020T130297, 2019M660119).

Citation: DANG Yaoguo, HUANG Jinxin, DING Xiaoyu, et al. Novel two-stage preflow algorithm for solving the maximum flow problem in a network with circles[J]. Journal of Southeast University (English Edition), 2025, 41 (1): 91-100. DOI: 10. 3969/j. issn. 1003-7985. 2025. 01. 012.

mum cost flow. With advancements in computer technology, several scholars have combined machine learning techniques and simplex methods to address the network maximum flow problem[17]. The Weisfeiler-Lehman simplex neural network is built based on a deep learning model with good network stability[18].

(2) Minimum cutoff algorithms. The minimum cutoff and maximum flow problems are paired problems, where the maximum flow in a network corresponds to the minimum cutoff flow. Apaolaza et al. [19] developed a method for the fast computation of minimum cut sets in large networks, known as gMCS. However, these intercept sets do not exist because of potential disconnections between points, which can reduce the efficiency of this algorithm. In practice, first, all possible truncation sets in the original network graph are considered. Then, the minimum of these sets is calculated. Thus, Miraskarshahi et al. [20] developed a minimum coordination support method for the fast enumeration of minimum cut sets in metabolic networks.

(3) Fuzzy solution algorithms. Mathew et al. [21] and Zhu et al. [22] proposed methods for solving directed fuzzy networks to obtain maximum flow. Zhang et al. [23] introduced a parallel maximum flow algorithm to address the high computational complexity associated with the maximum flow algorithm.

(4) Machine learning algorithms. Bertsimas et al. [24] provided an overview of machine learning applications that can be used to address optimization problems. Baycik[25] presented machine-learning-based approaches to address the maximum flow network interdiction problem and efficiently solve large-scale problems. Zhang et al. [26] identified coupled stations in a large-scale public transportation network based on the complex network theory and spatial information embedding. Wang et al. [27] used intelligent optimization algorithms to determine the process of factors affecting air quality in the Yangtze River Economic Belt.

The optimization of the maximum flow in a network is mainly achieved by fully exploiting the characteristics of the network. A ring-free network is a special type of network that can reduce the difficulty and time complexity of solving the maximum flow problem. Simple algorithms for ringless networks are discussed in the studies of Wu et al. [28] and Willson[29], describing the feasible flow and preflow advancement algorithms, respectively. In a ringless network, the absence of backward arcs indicates that reducing the flow on a particular arc does not need to be considered to increase the maximum flow of the network.

Overall, existing models for calculating the maximum flow in networks often overlook the impacts of circles on the network. Two shortcomings must be addressed in accordance with the aforementioned achievements. (1) Circles in the network can increase the computational complexity of the maximum flow problem. Several studies are unable to handle such problems effectively with their algorithms. (2) Numerous existing studies propose models for calculating the maximum flow in a network without circles. However, converting a network with circles into a non-circle network is crucial. This study proposes a novel two-stage preflow algorithm to calculate the maximum flow of a network with circles. The proposed model can convert a network with circles into a non-circle network by determining the zero-flow arc in the circle when it reaches its maximum flow. Then, the maximum flow of the produced non-circle network can be determined using matrix representations.

The inverse research idea is applied to analyze the state of each arc when the network reaches its maximum flow and addresses the aforementioned shortcomings. First, when the network reaches its maximum flow, any circle in the network must contain at least one arc with a minimum possible flow of 0. Using this property, the circle of the original network can be broken to transform it into an acyclic network. The preflow propulsion algorithm is applied to solve the maximum flow of the network after breaking the circle. Then, this property is applied to break the original network and transform it into a loop-free network. Finally, the zero-flow arcs identified within the circles are removed from the original network to ensure a loop-free network, and the maximum flow of the network is solved again. This algorithm divides the solution process into two stages. The first stage breaks the circles and identifies the zero-flow arcs. The second stage removes the zero-flow arcs from the identified circles to produce the corresponding graph for a network without circles from the original network.

1 Theoretical Basis

1.1 Basic concepts

In real life, the term "flow" refers to the movement of matter between different systems. Various systems exhibit flow problems, such as the flow of vehicles in public transportation systems, the flow of water in water supply systems, the flow of cash in financial systems, and the flow of people, information, and materials in military systems. A common feature of these systems is the presence of at least one point of departure and receipt, along with several intermediate points, collectively forming a network of flows.

Some basic concepts, such as feasible flow, augmentation chain, intercept set, and intercept amount, are crucial for understanding the maximum flow problems in networks. These concepts are introduced in detail in the related literature and are not extensively covered in this paper. Several methods, such as the marker method, the shortest expansion path algorithm, the reservation advance method, and the highest marker reservation advance method, can solve the maximum flow problem in networks. Among these methods, the marker method is the most widely used and easiest to understand. The present study uses the marker method to solve the maximum flow problem in networks. The basic idea of the marker method is to incrementally identify the program flow that does not saturate, namely, the expansion chain (the flow on each line is less than the capacity, whereas the reverse flow is not 0). For the program flow, the flow is increased to saturation, and the search for expansion chains is continued until none is detected. The actual network flow (feasible flow) is the maximum flow in the network.

Definition 1 Given a directed network graph $D = (V, E)$, V is one of the top assemblies of network D and E is the set of all edges of network D. At a starting point, V is recorded as v_s. A receiving point, recorded as v_t, is also detected. The rest of the points are referred to as intermediate points. For each arc $e_{ij} \in E$, the corresponding value for $c_{ij}(c_{ij} \geqslant 0) \in \boldsymbol{C}$ exists. e_{ij} is called the capacity of the arc and is usually referred to as a capacity network with transceiver points, recorded as $D = (V, E, \boldsymbol{C})$, which is a function on the set of arcs. $F = \{f_{ij}\}$ records the actual flow rate of e_{ij}.

Definition 2 For a given network $D = (V, E, \boldsymbol{C})$, v_s is the starting point, v_t is the collection, and n is the number of intermediate points. f is a feasible flow in the network if it satisfies the capacity constraints and equilibrium conditions.

(1) Capacity constraints. For each arc, $e_{ij} \in E$ satisfies $0 \leqslant f_{ij} \leqslant c_{ij}$.

(2) Equilibrium conditions. For the intermediate points, a constant outflow equal to the inflow exists, indicating that the following relation is detected for each intermediate point:

$$\sum_{j=1}^{n} f_{ij} - \sum_{j=1}^{n} f_{ji} = 0 \qquad i = 1, 2, \cdots, n;\ i \neq j \tag{1}$$

The following equation is used for the sending and receiving points v_s and v_t:

$$\sum_{i=1}^{n} f_{si} = \sum_{j=1}^{n} f_{jt} = v(f) \tag{2}$$

where $v(f)$ is the network and i is the flow of viable streams.

The maximum flow problem involves finding the feasible flow f. The total traffic is maximized from the originating point to the receiving point of the network $v(f)$, as follows:

$$\begin{aligned} &\max v(f) \\ &\text{s. t.} \\ &0 \leqslant f_{ij} \leqslant c_{ij}, \quad e_{ij} \in E \\ &\sum_{j=1}^{n} f_{ij} - \sum_{k=1}^{n} f_{ki} = \begin{cases} v(f) & i = s \\ 0 & i = 1, 2, \cdots, n \\ -v(f) & i = t \end{cases} \end{aligned} \tag{3}$$

Definition 3 In the network, the arc $f_{ij} = c_{ij}$ is called a saturated arc, the arc $f_{ij} \leqslant c_{ij}$ is called an unsaturated arc, the arc $f_{ij} = 0$ is called a zero-flow arc, and the arc $f_{ij} > 0$ is called a nonzero-flow arc. If μ is a chain from starting point v_s to receiving point v_t and the direction of the chain from v_s to v_t can be defined, then the arc in the same direction as the chain is called the forward arc, recorded as e_{ij}^{+}. The arc in the opposite direction of the chain is called a backward arc, recorded as e_{ij}^{-}. A chain with only forward arcs is called a forward chain.

f is set as the feasible flow, and μ is the chain from v_s to v_t. If μ satisfies the following conditions, then f is an extended chain of feasible flows:

On arc $e_{ij} \in u^{+}$, if $0 \leqslant f_{ij} \leqslant c_{ij}$, then every arc of e_{ij}^{+} is an unsaturated arc.

On arc $e_{ij} \in u^{-}$, if $0 \leqslant f_{ij} \leqslant c_{ij}$, then every arc of e_{ij}^{-} is a nonzero-flow arc.

Definition 4 $D = (V, E, \boldsymbol{C})$ is a directed network graph, and $\mu_{(i, i+1, \cdots, i+k, i)}$ is a directed chain that starts from v_i, subsequently passes through points $v_{i+1}, v_{i+2}, \cdots, v_{i+k}$ and returns to v_i. Such a chain is called a directed circle, recorded as $k_{(i, i+1, \cdots, i+k, i)}$.

Definition 5 $D = (V, E, \boldsymbol{C})$ is a directed network graph, and $k_{(i, i+1, \cdots, i+k, i)}$ is a directed circle that starts from v_i and returns to v_i. When the network reaches its maximum flow, the flows in each arc of the circle are $f_{i,i+1}, f_{i+1,i+2}, \cdots, f_{i+k-1,i+k}, f_{i+k,i}$. Then, the smallest arc flow in the circle is denoted as $f_{\min}$, where

$$f_{\min} = \min \begin{cases} f_{j,j+1} & j = i, i+1, \cdots, i+k-1 \\ f_{j,i} & j = k \end{cases}$$

Different states often exist simultaneously when the network reaches its maximum flow. Suppose $f_{j,j+1}(j = i, i+1, \cdots, i+k)$ is one of the flows of circle $k_{(i, i+1, \cdots, i+k, i)}$ of the arc. The states of the network under the network $f_{j,j+1}(j = i, i+1, \cdots, i+k)$ differ when the network reaches its maximum flow. The smallest of all possible $f_{j,j+1}(j = i, i+1, \cdots, i+k)$ flows is referred to as the minimum possible flow (maximum lower bound flow), regarded as $\min f_{j,j+1}$.

Fig. 1 shows two states when the same network reaches its maximum flow. Based on Definition 5, $k_{(1,2,3,1)}$ is a circle of the network. Thus, $f_{\min} = \min(f_{12}, f_{23}, f_{31}) = 0$, and arc e_{31} is the minimum flow arc. The minimum possible flows for all arcs in the circle are $\min f_{12} = 4$, $\min f_{23} = 1$, and $\min f_{31} = 0$.

1.2 Theorems and corollaries

Theorem 1 $D = (V, E, \boldsymbol{C})$ is a directed network graph, and $k_{(i, i+1, \cdots, i+k, i)}$ is a directed circle that starts from point v_i and returns to point v_i. The first number in parentheses is the capacity, and the second number is the flow. The capacity of each arc is $c_{i,i+1}$, $c_{i+1,i+2}, \cdots,$

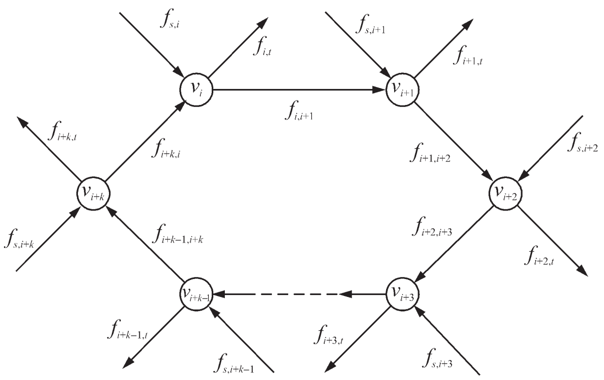

Fig. 1 Schematic of two important concepts

$c_{i+k,i}$. The minimum possible flows for each arc in the circle when the network reaches its maximum flow are $\min f_{i,i+1}$, $\min f_{i+1,i+2}$, $\cdots$, $\min f_{i+k-1,i+k}$, and $\min f_{i+k,i}$. The circle $k_{(i,i+1,\cdots,i+k,i)}$ must be at least one arc with a minimum possible flow equal to 0.

Proof Fig. 2 illustrates the flow diagram of a circle within the network at maximum flow. $f_{i,i+1}$ is the flow from point v_i to point v_{i+1}; f_{si} is the total of the flows of all arcs, except for $f_{i-1,i}$ flowing in the point v_i; f_{it} is the sum of all arcs, except for point $f_{i,i+1}$ flowing from point v_i.

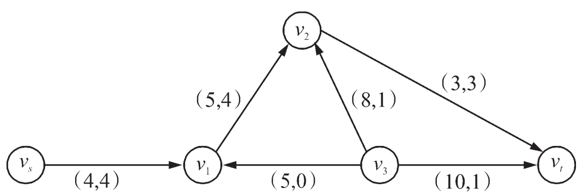

Fig. 2 Flow diagram of a circle within the network diagram

(1) If $f_{\min} = 0$, then a zero-flow arc already exists in the circle at this time, and the minimum possible flow of an arc in the circle is 0.

(2) If $f_{\min} \neq 0$, then the arc with the flow rate $f_{\min}$ in the circle does not have the minimum possible flow rate.

For the circles shown in Fig. 2, the equilibrium equation for each point can be expressed as follows:

$$\begin{cases} f_{s,i} + f_{i+k,i} = f_{i,t} + f_{i,i+1} & m = 0 \\ f_{s,i+m} + f_{i+m-1,i+m} = f_{i+m,t} + f_{i+m,i+m+1} & \\ & m = 1, 2, \cdots, k-1 \\ f_{s,i+k} + f_{i+k-1,i+k} = f_{i+k,t} + f_{i+k,i} & m = k \end{cases} \tag{4}$$

The two sides of Eq. (1) are added separately to yield all points with balanced equations, as follows:

$$\sum_{m=0}^{k} f_{s,i+m} + f_{i+k,i} + \sum_{m=1}^{k-1} f_{i+m-1,i+m} + f_{i+k-1,i+k} = \sum_{m=0}^{k} f_{i+m,t} + f_{i,i+1} + \sum_{m=1}^{k-1} f_{i+m,i+m+1} + f_{i+k,i} \tag{5}$$

Simplifying Eq. (5) yields the following expression:

$$\sum_{m=0}^{k} f_{s,i+m} = \sum_{m=0}^{k} f_{i+m,t} \tag{6}$$

The maximum flow that can impact the network is the sum of the flows from each point to the outside of the circle $\sum_{m=0}^{k} f_{i+m,t}$. Thus, the flow of all arcs in the circle can be simultaneously subtracted from $f_{\min}$, which neither affects the equilibrium conditions nor changes the maximum flow of the network. Therefore, the arc with flow rate $f_{\min}$ is not considered the minimum possible flow rate. Given that the network is already at its maximum flow, no new incremental chains must appear when the traffic to all arcs in the circle is simultaneously subtracted from $f_{\min}$. All arcs flow in the circle minus $f_{\min}$. Thus, the maximum flow of the entire network remains unchanged. If a new incremental chain appears, then the original network has not reached its maximum flow, contradicting the premise that the network has reached its maximum flow. Therefore, at this time, the original flow rate $f_{\min}$ in the circle of this arc is considered the minimum possible flow rate, which is 0. This finding proves the minimum possible flow rate.

Citation 1 As shown in Fig. 2, $D = (V, E, \boldsymbol{C})$ is a directed network graph, $k_{(i,i+1,\cdots,i+k,i)}$ is a directed network that starts from v_i and returns to v_i, points v_i and v_{i+1} are any two points adjacent to each other in the circle, and $e_{i,i+1}$ includes arc links v_i and v_{i+1}. If the flow $f_{i,i+1}$ of arc $e_{i,i+1}$ is the minimum possible quantity $\min f_{i,i+1}$, then this arc is removed from the network when the network reaches its maximum flow state. The maximum flow reduction for the entire network $\min f_{i,i+1}$ and the zero-flow arc in the circle remain unchanged.

Proof Fig. 3 shows the two points v_i, v_{i+1} in the circle when the entire network reaches its maximum flow. $F_{i\mathrm{TI}} = f_{i+k,i} + f_{si}$ is the sum of all arc flows into point v_i. f_{it} is the sum of all remaining arc flows out of point v_i excluding $f_{i,i+1}$. $f_{s,i+1}$ is the sum of all remaining flows into the arc at point v_i excluding $f_{i,i+1}$. $F_{i+1\mathrm{TO}} = f_{i+1,i+2} + f_{i+1,t}$ is the sum of all flows out of point v_{i+1}. $\min f_{i,i+1}$ is the minimum possible flow achieved by arc $e_{i,i+1}$ while guaranteeing the maximum flow in the network. Δf_{it} and $\Delta f_{s,i+1}$ are the possible increase in the flows of f_{it} and $f_{s,i+1}$, respectively.

$f_{i,i+1}$ is considered the smallest possible value of

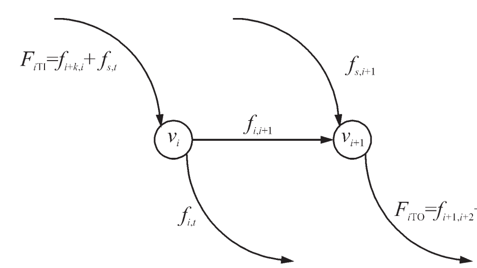

Fig. 3 State diagram of points v_i and v_{i+1} in the circle when the network diagram reaches its maximum flow

$\min f_{i,i+1}$; thus, $\Delta f_{it} = \Delta f_{s,i+1} = 0$. If Δf_{it} and $\Delta f_{s,i+1}$ are not 0, then $f_{i,i+1}$ must be reduced further to increase f_{it} or $f_{s,i+1}$.

(1) If $\min f_{i,i+1} = 0$, then the zero-flow arc in that circle is arc $e_{i,i+1}$. Removing this arc from the network does not affect the entire network.

(2) If $\min f_{i,i+1} \neq 0$, then the arc $e_{i,i+1}$ is removed from the network. F_{iTI} and F_{i+1TO} are reduced by $\min f_{i,i+1}$, and $f_{it}, f_{s,i+1}$ remains unchanged. Given that $\Delta f_{it} = \Delta f_{s,i+1} = 0$, the new extended chain that is not created by the removal of arc $e_{i,i+1}$ from the network does not cause an increase in flow on any other arc in the network, and the maximum flow of the entire network is reduced by $\min f_{i,i+1}$. The zero-flow arc in the circle remains unchanged.

Citation 2 $D = (V, E, \boldsymbol{C})$ is a directed network graph, and $k_{(i,i+1,\cdots,i+k,i)}$ is a directed circle that starts from point v_i and returns to point v_i. The network graph has multiple possible states when the network reaches its maximum flow. In this case, a state where all of the arcs in the circle simultaneously take the minimum possible flow exists.

Proof As shown in Fig. 2, each arc in the circle is connected to four arcs. For example, for arc $e_{i,i+1}$, e_{si}, $e_{i+k,i}$ and e_{it}, $e_{i+1,i+2}$ exist at the input and output, respectively, where e_{si}, e_{it} is the sum of the input and output arcs outside the circle. According to Citation 1, arcs e_{it} and $e_{s,i+1}$ must obtain the maximum possible flow when arc $e_{i,i+1}$ obtains the minimum flow. Moreover, the flow on arc $e_{i+k,i}$, $e_{i+1,i+2}$ remains unaffected. In particular, when an arc in the circle obtains the minimum flow, the front and back arcs in the circle can simultaneously obtain the minimum flow. Similarly, all arcs within the circle can simultaneously obtain the minimum possible flow.

Theorem 2 $D = (V, E, \boldsymbol{C})$ is a directed network graph, $k_{(i,i+1,\cdots,i+k,i)}$ is a directed circle that starts from point v_i and returns to point v_i, and arcs $e_{i,i+1}$, $e_{i+1,i+2}$, $\cdots$, $e_{i+k,i}$ belong to this point. When the network reaches its maximum flow and each arc in the circle obtains the minimum possible flow simultaneously, the original zero-flow arc in the circle remains unchanged after removing any arc from the circle.

Proof According to Citation 2, all arcs within the circle can simultaneously take the minimum possible flow when the network reaches its maximum flow. Moreover, Theorem 1 states that at least one arc in the circle has a minimum possible flow of 0. Therefore, according to Citation 1, when all arcs in the circle are guaranteed to achieve the minimum possible flow, removing any arc from the network at maximum flow does not change the zero-flow arc in the circle.

Corollary 1 $D = (V, E, \boldsymbol{C})$ is a directed network graph, $k_{(i,i+1,\cdots,i+k,i)}$ is a directed circle that starts from point v_i and returns to point v_i, and arcs $e_{i,i+1}$, $e_{i+1,i+2}$, $\cdots$, $e_{i+k,i}$ belong to this point. If the network reaches its maximum flow and all arcs in the circle obtain the minimum possible flow, then removing any arc from the circle in the initial network and removing this arc from the circle of the maximum flow network provide the same maximum flow in the network. Moreover, the zero-flow arc in the circle remains unchanged.

Proof If the arc $e_{i,i+1}$ is removed directly from the initial network, then the maximum flow of the network is solved after removing $e_{i,i+1}$ from Fig. 3, as shown in Fig. 4.

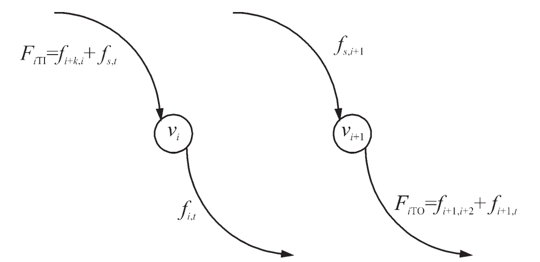

Fig. 4 State diagram of the circle points v_i and v_{i+1} when the network diagram reaches its maximum flow after removing $e_{i,i+1}$

Given that the network has already reached its maximum flow, $\Delta F_{iTI}\Delta f_{it} = 0$ and $\Delta f_{s,i+1}\Delta F_{i+1TO} = 0$. If arc $e_{i,i+1}$ is added to the network that has reached its maximum flow, then the following conditions are satisfied:

(1) If $\Delta F_{iTI}\Delta F_{i+1TO} = 0$, then the maximum flow of the network remains the same after adding $e_{i,i+1}$, $\min f_{i,i+1} = 0$.

(2) If $\Delta F_{iTI}\Delta F_{i+1TO} \neq 0$, then $\Delta f_{it} = \Delta f_{s,i+1} = 0$, and the maximum flow of the network increases by $\min f_{i,i+1}$ after adding $e_{i,i+1}$. Overall, removing $e_{i,i+1}$ from the initial network reduces the maximum flow of the network by $\min f_{i,i+1}$. Combining this condition with Citation 1, the maximum flow of the network is the same. Moreover, the maximum network flow obtained by removing $e_{i,i+1}$ from the initial network is similar to that obtained by removing $e_{i,i+1}$ from the maximum flow network. In addition, the zero-flow arc remains the same.

2 Construction of the Two-Stage Preflow Algorithm

The preflow advancement algorithm for determining the maximum flow of the network starts from the source point and advances the maximum possible flow along the edges of the remaining network until no further advancement can be made. If circles exist in the network, then this algorithm may need to reduce the flow on certain backward arcs to facilitate an increase in the overall maximum flow. Therefore, arcs that are already saturated cannot be ignored. However, if no circles exist in the network, then the saturated arcs in the remaining network can be ignored. The loop-free network is simpler and faster to process than a network with loops. The following algorithm for advancing the network with circles is performed twice to address the maximum flow problem in networks. This algorithm is divided into two stages. The first stage identifies the zero-flow arcs in the circles based on the conclusion of Corollary 1 from Theorem 1. The second stage involves solving the maximum flow for the loop-free network that corresponds to the original network with circles.

According to Citation 1, when advancing to the apex of the circle, the first stage of solving the maximum flow of the network after breaking the circle ensures the minimum possible flow through the advancement process in the circle arc, satisfying $\Delta f_{it} = \Delta f_{s,i+1} = 0$. Priority is given to advancing flow through arcs outside the circle. Once these arcs reach saturation, flow advancement is directed toward the arcs within the circle.

Definition 6 $D = (V, E, \boldsymbol{C})$ is a directed network graph. The capacity of endpoints v_i and v_j is c_{ij}. Thus, the matrix is called the initial capacity matrix, which can be expressed as follows:

$$\boldsymbol{C} = \begin{bmatrix} 0 & c_{s1} & c_{s2} & \cdots & c_{sn} & c_{st} \\ 0 & 0 & c_{13} & \cdots & c_{1n} & c_{1t} \\ 0 & c_{21} & 0 & \cdots & c_{2n} & c_{2t} \\ \vdots & \vdots & \vdots & & \vdots & \vdots \\ 0 & c_{n1} & c_{n2} & \cdots & 0 & c_{nt} \\ 0 & c_{t1} & c_{t2} & \cdots & c_{tn} & 0 \end{bmatrix}$$

Theorem 3 $D = (V, E, \boldsymbol{C})$ is a directed network graph. If no circles exist in network D, then the initial capacity matrix $\boldsymbol{C}$ of any network must be translated into a strictly upper triangular initial capacity matrix $\boldsymbol{C}'$ equal to the maximum flow of the initial network.

Proof Let i, j be the middle two points in the network graph. Then, swapping the labels of points i and j does not affect the maximum flow of the network graph by swapping rows i and j and columns i and j, respectively, in the initial capacity matrix $\boldsymbol{C}$ of the network. The network is a loop-free network. Thus, $c_{ij}c_{ji} = 0$ starts from Column 1 and subsequently searches for $c_{ij} > 0$ and $i > j$ in each column from top to bottom. In the presence of c_{ij}, rows i and j and columns i and j are swapped, respectively, until the last column is adjusted. Then, the entire initial capacity matrix is transformed into the upper triangular initial capacity matrix $\boldsymbol{C}'$.

Theorem 4 $D = (V, E, \boldsymbol{C})$ is a directed network graph without circles. Maximum capacity advancement is made along a certain forward chain from the point of origin to generate a saturated chain. The existing traffic on the arc must not decrease during the subsequent advances.

Proof In the Ford-Fulkerson scalar algorithm, all forward arcs increase the flow with the adjustment of an increasing chain. Moreover, backward arcs decrease the flow. If the initial network is a loop-free network, then Theorem 3 indicates that the initial capacity matrix can be transformed into a strict upper triangular capacity matrix, and all arcs in the network are forward arcs. Thus, no arc decreases the flow during the maximum flow adjustment in the network.

Therefore, Theorem 1 states that all circles are detected in the initial network. Then, an arc in a circle is arbitrarily selected and removed from the network. According to Theorem 3, the initial capacity matrix is transformed into an upper triangular capacity matrix. Finally, Theorem 4 indicates that the network traffic is increased in the upper triangular capacity matrix by traversing it stepwise. In conjunction with Theorem 2, the maximum flow optimization process for the upper triangular capacity matrix must ensure that each arc in the circle achieves the minimum possible flow, as described in Step 4. The specific steps of the maximum flow solution method of the network are as follows:

Phase Ⅰ

Step 1 Find the circle in the initial network. If a circle does not exist, then proceed to Step 2; if a circle exists, then choose any arc to remove and proceed to Step 2.

Step 2 Write the capacity matrix $\boldsymbol{C}$ of the network after breaking the circle and the upper triangular initial capacity matrix $\boldsymbol{C}'$ according to the transformation method in the proof of Theorem 3.

Step 3 Iterate through the numbers in the first row of matrix $\boldsymbol{C}'$ in order, with each number traversed as follows:

① For arc capacity c_{sj}, if $c_{sj} = 0$, then proceed to the traversal of $c_{s,j+1}$. If $c_{sj} \neq 0$, then proceed to row j and start traversing within row $j+1$ starting from $c_{j.j+1}$. If all rows $j+1$ are 0, then proceed to the traversal of $c_{s,j+1}$; otherwise, proceed to Step ②.

② For the first nonzero c_{jk} $(j \leqslant k \leqslant n)$ encountered at row $j+1$, if c_{jk} is the capacity of one of the arcs in the circle, then this arc is skipped; if c_{jk} is not the arc capacity of the circle, then proceed to row $k+1$ to traverse from $c_{k.k+1}$ and repeat Step ② until the last column is traversed in row l. At this point, the series obtained by this round of traversal $\{c_{sj}, c_{j,k}, \cdots, c_{lt}\}$ is removed, and $c_{pq} = \min\{c_{sj}, c_{j,k}, \cdots, c_{lt}\}$ is solved. Then, all corresponding capacities in the series are subtracted from c_{pq}, and the capacity matrix $\boldsymbol{C}'$ is updated and transferred.

③ Then, $c_{pq} = 0$. Continue Step ② from $c_{p,q+1}$. Repeat Step ③. If the expanded chain can be obtained, then the series obtained in this round should be $\{c_{1j} - c_{pq}, c_{jk} - c_{pq}, \cdots, c_{p,q+i}, c_{l't}\}$ $(q < i < t)$. If no augmentation chain exists, then proceed to Step ②. Start from $c_{j,k+1}$ and traverse backward until the first number that is not 0 is encountered. If the end of c_{jt} is traversed, then go back and repeat Step ② for the previously skipped arcs. Once all of the arcs in the circle have been processed, go back to Step ① to process $c_{s,j+1}$. The final capacity matrix $\boldsymbol{C}'$ of the first stage is obtained until the end of the traversal to c_{sn}. Then, proceed to Step 4 (skipping the circle arc ensures that it obtains the minimum possible flow).

Phase Ⅱ

Step 4 Let the network maximum flow matrix $\boldsymbol{A} = \boldsymbol{C}' - \boldsymbol{C}''$ find all of the arcs in the circle in matrix $\boldsymbol{A}$ and identify all of the arcs with a flow rate of 0 in the circle. Remove these arcs in the initial network capacity matrix $\boldsymbol{C}$ to obtain the network maximum flow initial capacity matrix $\boldsymbol{C}_{\max}$ and proceed to Step 5.

Step 5 Repeat Step 2 for the maximum flow initial capacity matrix $\boldsymbol{C}_{\max}$ of the network (given that no more circles exist in the network, skipping the step of the arc in the circle is unnecessary). Based on Step 3, Step 4 yields the network maximum flow matrix $\boldsymbol{A}_{\max}$. The network maximum flow is $f^* = \sum_{i=s}^{n} a_{in}$.

Fig. 5 shows the specific steps of the maximum flow solution method of the network.

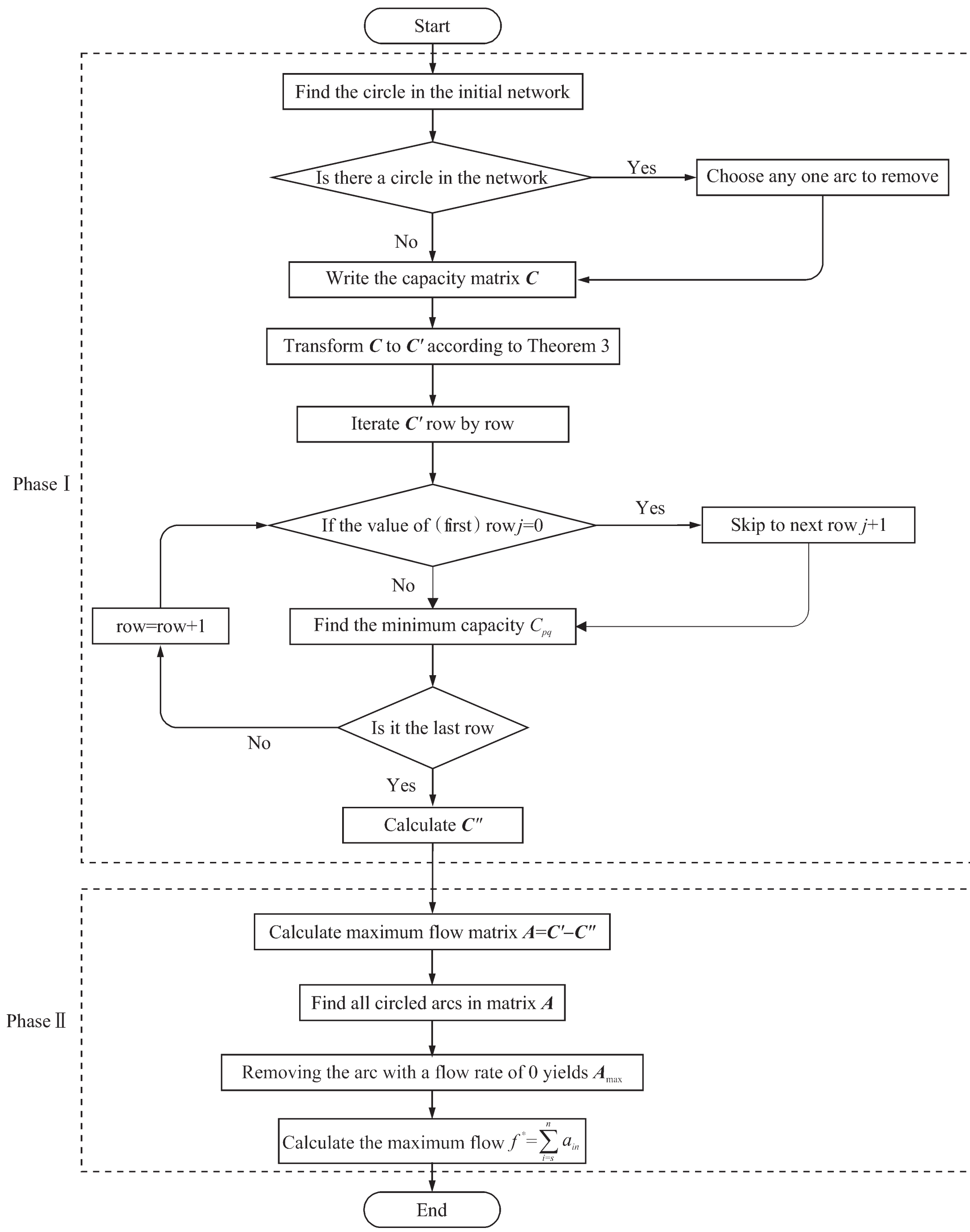

Fig. 5 Flowchart for solving the maximum flow algorithm in a network

3 Case Study and Comparisons with Existing Models

A campus covers an area of approximately 3 000 acres. This area has 40 000 people who rush from the dormitory area to the academic buildings, laboratory buildings, and library in the morning. The maximum capacity of the existing campus road design plan enables the students to rush to the teaching area promptly. The maximum flow of the existing route analysis of the west campus gate to the east court must be calculated. The campus map is shown in Fig. 6.

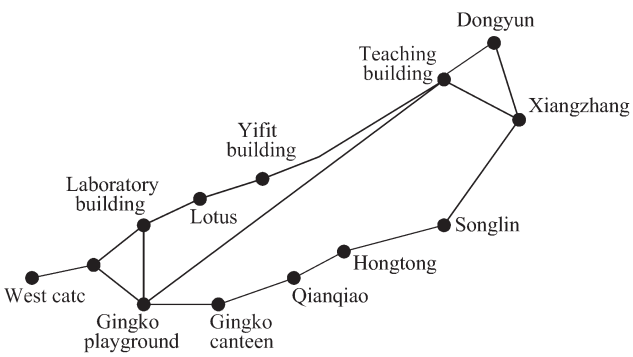

Fig. 6 Diagram of the main traffic routes on the campus

A campus plan is available. A concentrated area is simplified to a point, and the road is simplified to a line by streamlining and marking the road conditions of the campus. Moreover, the distance of each point and the width of the road are measured in accordance with the network map range, and the saturation flow of the road is calculated. The campus transportation network map can be simplified as $D = (V, E, \boldsymbol{C})$, as shown in Fig. 7. The number next to the arc in the figure is the road saturation capacity c_{ij}.

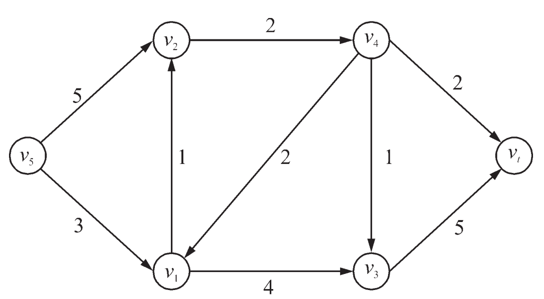

Fig. 7 Campus transportation network map

Step 1 Delete arc e_{24} if circles exist in the network $k_{(1,2,4,1)}$.

Step 2 After the circle is broken, the upper triangular capacity matrix $\boldsymbol{C}'$ is given in accordance with the rank transformation method in the proof of Theorem 3, as follows:

$$\boldsymbol{C}' = \begin{bmatrix} 0 & 0 & 3 & 5 & 0 & 0 \\ 0 & 0 & 2 & 0 & 1 & 2 \\ 0 & 0 & 0 & 1 & 4 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 5 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Step 3 After traversal, the final capacity matrix $\boldsymbol{C}''$ is obtained after breaking the circle, as follows:

$$\boldsymbol{C}'' = \begin{bmatrix} 0 & 0 & 0 & 5 & 0 & 0 \\ 0 & 0 & 2 & 0 & 1 & 2 \\ 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 2 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Step 4 $\boldsymbol{A} = \boldsymbol{C}' - \boldsymbol{C}''$ is obtained. The zero-flow arc in the original network e_{12}, e_{23} is identified. Thus, $\boldsymbol{C}_{\max}$ is obtained as follows:

$$\boldsymbol{A} = \begin{bmatrix} 0 & 0 & 3 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 3 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 3 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}, \boldsymbol{C}_{\max} = \begin{bmatrix} 0 & 3 & 5 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 4 & 0 \\ 0 & 0 & 0 & 2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 2 \\ 0 & 0 & 0 & 0 & 0 & 5 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Step 5 For the preflow propulsion algorithm for $\boldsymbol{C}_{\max}$, $\boldsymbol{A}_{\max}$ is obtained as follows:

$$\boldsymbol{A}_{\max} = \begin{bmatrix} 0 & 3 & 2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 3 & 0 \\ 0 & 0 & 0 & 2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 2 \\ 0 & 0 & 0 & 0 & 0 & 3 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Finally, the traffic of each arc in the network is obtained, and the maximum network flow is $f^* = \sum_{i=s}^{n} a_{in} = 5$. The campus road capacity addresses the needs of all students and teachers to pass by combining the results of the network maximum flow calculation.

Two models are selected for comparison with the proposed model. The compared models are denoted as M1[11], M2[30], and M3[4]. The results are given in Table 1.

Table 1 Comparison with the existing models

Model name	Network with circles
Proposed model	√
M1	×
M2	×
M3	×

Note: The check sign √ means the model has the desired property, whereas mark × indicates that the model does not have the capability.

4 Conclusions

This study showed that the arcs with flow in the network do not form a cycle when the network reaches its maximum flow. The use of the preflow advancement algorithm is further explored to solve the maximum flow in the network. As long as the arcs in the cycle take the minimum possible flow, removing an arc in the cycle does not change the maximum flow of the original network. Based on this idea, a two-stage preflow advancement algorithm is constructed. The first stage identifies the zero-flow arc in the circle at the maximum flow state of the network. The second stage focuses on solving for the maximum flow of the loop-free network after the removal of the zero-flow arc in the circle. The first stage improves the preflow propulsion algorithm to ensure that the minimum possible flow is obtained in the intraloop arc. Compared with the common preflow advancement algorithm, the proposed method does not need to consider the existence of reversed arcs, saving more time than the minimum cut set algorithm and solving the final state matrix of the maximum flow of the network. In addition, choosing different removal arcs or using the arc-flow relationship can help easily determine the zero-flow arc in the circle, and the steps implemented to solve the maximum flow problem can be reduced when the circle in the first stage breaks.

However, when there are numerous circles in the network, the computational complexity of the proposed approach may exponentially grow. The proposed approach can also be expanded to the network with nonconservation flow[12] and acyclic networks[29].

References

[1] CAO B, ZHAO J W, GU Y, et al. Security-aware industrial wireless sensor network deployment optimization [J]. IEEE Transactions on Industrial Informatics, 2020, 16(8): 5309-5316.

[2] ZHANG L, FENG D M, WU G. Vehicle dynamic load recognition method based on LSTM network[J]. Journal of Southeast University (Natural Science Edition), 2023, 53(2): 187-192. (in Chinese)

[3] GUPTA S P, PYAKUREL U, DHAMALA T N. Multi-commodity flow problem on lossy network with partial lane reversals[J]. Annals of Operations Research, 2023, 323(1): 45-63.

[4] MIRZAEI M, MIRZAPOUR AL-E-HASHEM S M J, AKBARPOUR SHIRAZI M. A maximum-flow network interdiction problem in an uncertain environment under information asymmetry condition: Application to smuggling goods [J]. Computers & Industrial Engineering, 2021, 162: 107708.

[5] DING J, WEN C Y, LI G Q, et al. Target controllability in multilayer networks *via* minimum-cost maximum-flow method[J]. IEEE Transactions on Neural Networks and Learning Systems, 2021, 32(5): 1949-1962.

[6] XIE C Y, DONG L. Graph-enhanced neural interactive collaborative filtering[J]. Journal of Southeast University (English Edition), 2022, 38(2): 110-117.

[7] MEHRYAR M, HAFEZALKOTOB A, AZIZI A, et al. Cooperative reliability allocation in network flow problems considering greenhouse gas emissions: Optical fiber networks structure[J]. Journal of Cleaner Production, 2021, 326: 129315.

[8] SHI M Z, LIN Z D. Effectiveness evaluation of railway traffic safety based on the DEMATEL, AHP, and ANP methods[J]. Journal of Southeast University (English Edition), 2023, 39(2): 133-141.

[9] ALIPOUR H, MUÑOZ M A, SMITH-MILES K. Enhanced instance space analysis for the maximum flow problem[J]. European Journal of Operational Research, 2023, 304(2): 411-428.

[10] FORD L R Jr, FULKERSON D R. A simple algorithm for finding maximal network flows and an application to the Hitchcock problem[J]. Canadian Journal of Mathematics, 1957, 9: 210-218.

[11] AHUJA R K, ORLIN J B. A fast and simple algorithm for the maximum flow problem[J]. Operations Research, 1989, 37(5): 748-759.

[12] HUANG D H. A network reliability algorithm for a stochastic flow network with non-conservation flow[J]. Reliability Engineering & System Safety, 2023, 240: 109584.

[13] KARZANOV A. Determining the maximal flow in a network by the method of preflows[J]. Doklady Mathematics, 1974, 15: 434-437.

[14] KARA G, ÖZTURAN C. Algorithm 1002: Graph coloring based parallel push-relabel algorithm for the maximum flow problem[J]. ACM Transactions on Mathematical Software, 2019, 45(4): 1-28.

[15] DEUTSCH M, DAĞDELEN K, JOHNSON T. An open-source program for efficiently computing ultimate pit limits: MineFlow[J]. Natural Resources Research, 2022, 31(3): 1175-1187.

[16] HOLZHAUSER M, KRUMKE S O, THIELEN C. A network simplex method for the budget-constrained minimum cost flow problem[J]. European Journal of Operational Research, 2017, 259(3): 864-872.

[17] LI X L, WANG Y Q, MA G B, et al. Electric load forecasting based on long-short-term-memory network via simplex optimizer during COVID-19[J]. Energy Reports, 2022, 8: 1-12.

[18] HUANG R, FENG M R, CHEN Z J, et al. Exploring network reliability by predicting link status based on simplex neural network[J]. Displays, 2023, 79: 102457.

[19] APAOLAZA I, VALCARCEL L V, PLANES F J. gMCS: Fast computation of genetic minimal cut sets in large networks[J]. Bioinformatics, 2019, 35(3): 535-537.

[20] MIRASKARSHAHI R, ZABETI H, STEPHEN T, et al. MCS_2: Minimal coordinated supports for fast enumeration of minimal cut sets in metabolic networks[J]. Bioinformatics, 2019, 35(14): i615-i623.

[21] MATHEW S, MORDESON J N. Directed fuzzy networks as a model to illicit flows and max flow Min cut theorem[J]. New Mathematics and Natural Computa-

tion, 2017, 13(3): 219-229.
[22] ZHU H X, SHEN J, SU Z G, et al. RBF neural network regression model based on fuzzy observations[J]. Journal of Southeast University (English Edition), 2013, 29 (4): 400-406.
[23] ZHANG R Q, WEI W. A parallel algorithm for the maximum flow problem[J]. International Core Journal of Engineering, 2019, 5: 61-64.
[24] BERTSIMAS D, STELLATO B. The voice of optimization[J]. Machine Learning, 2021, 110(2): 249-277.
[25] BAYCIK N O. Machine learning based approaches to solve the maximum flow network interdiction problem [J]. Computers & Industrial Engineering, 2022, 167: 107873.
[26] ZHANG L, LU J, LEI D. Vulnerability analysis of bus-metro composite network based on complex network and spatial information embedding[J]. Journal of Southeast University (Natural Science Edition), 2019, 49 (4): 773-780. (in Chinese)
[27] WANG J J, CAI Y, FENG Y, et al. Novel grey dynamic trend relational analysis models with different types of accumulation delay effects for time-delay systems[J]. Expert Systems with Applications, 2024, 238: 121661.
[28] WU J S, CHEN N, ZHOU C J, et al. Computing the number of loop-free k-hop paths of networks[J]. IEEE Transactions on Services Computing, 2022, 15(4): 2114-2128.
[29] WILLSON S J. Merging arcs to produce acyclic phylogenetic networks and normal networks[J]. Bulletin of Mathematical Biology, 2022, 84(2): 26.
[30] WU Y, YANG Y L, LIU S Y. The network maximum flow based on the flow matrix[J]. System Engineering, 2007, 25(10): 122-125.

两阶段预流算法构建及其在带环网络最大流中的应用

党耀国, 黄金鑫, 丁孝郁, 王俊杰

(南京航空航天大学经济与管理学院,南京 211100)

摘要: 在处理带环网络的最大流问题时,为了降低算法的复杂度,提出了一种新颖的两阶段预流算法。首先,证明了当网络达到最大流状态时,环中必然存在至少一条最小可能流等于零的弧。在环中,如果每条弧同时获得最小可能流量,在移除任意一条弧之后,环中原始零流弧保持不变。其次,构建了使带环网络转换为无环网络两阶段预流算法:阶段1为当网络达到最大流时标记出环中的零流弧;阶段2为去除在阶段1中找到的零流弧,从而将原本的带环网络转化为无环网络。通过求解新生成的无环网络的最大流,可以得到原始带环网络的最大流解。最后,通过实例验证了该算法的有效性和可行性。该算法不仅提高了计算效率,还为解决类似网络优化问题提供了新的视角和工具。

关键词: 带环网络;最大流;零流弧;两阶段预流算法

中图分类号: O22;O157.5