

Article

Using k -Mix-Neighborhood Subdigraphs to Compute Canonical Labelings of Digraphs

Jianqiang Hao ^{1,2,*}, Yunzhan Gong ¹, Yawen Wang ¹, Li Tan ² and Jianzhi Sun ²

¹ State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications, 100876 Beijing, China; gongyz@bupt.edu.cn (Y.G.); wangyawen@bupt.edu.cn (Y.W.)

² School of Computer and Information Engineering, Beijing Technology and Business University, 100048 Beijing, China; tanli_913@sina.com (L.T.); sunjz@th.btbu.edu.cn (J.S.)

* Correspondence: Bshjq@vip.163.com; Tel.: +86-10-6898-5704

Academic Editors: Raúl Alcaraz Martínez and Kevin H. Knuth

Received: 18 October 2016; Accepted: 15 February 2017; Published: 22 February 2017

Abstract: This paper presents a novel theory and method to calculate the canonical labelings of digraphs whose definition is entirely different from the traditional definition of *Nauty*. It indicates the mutual relationships that exist between the canonical labeling of a digraph and the canonical labeling of its complement graph. It systematically examines the link between computing the canonical labeling of a digraph and the k -neighborhood and k -mix-neighborhood subdigraphs. To facilitate the presentation, it introduces several concepts including *mix diffusion outdegree sequence* and *entire mix diffusion outdegree sequences*. For each node in a digraph G , it assigns an attribute $m_NearestNode$ to enhance the accuracy of calculating canonical labeling. The four theorems proved here demonstrate how to determine the first nodes added into $MaxQ(G)$. Further, the other two theorems stated below deal with identifying the second nodes added into $MaxQ(G)$. When computing $C_{max}(G)$, if $MaxQ(G)$ already contains the first i vertices u_1, u_2, \dots, u_i , Diffusion Theorem provides a guideline on how to choose the subsequent node of $MaxQ(G)$. Besides, the Mix Diffusion Theorem shows that the selection of the $(i + 1)$ th vertex of $MaxQ(G)$ for computing $C_{max}(G)$ is from the open mix-neighborhood subdigraph $N^{++}(Q)$ of the nodes set $Q = \{u_1, u_2, \dots, u_i\}$. It also offers two theorems to calculate the $C_{max}(G)$ of the disconnected digraphs. The four algorithms implemented in it illustrate how to calculate $MaxQ(G)$ of a digraph. Through software testing, the correctness of our algorithms is preliminarily verified. Our method can be utilized to mine the frequent subdigraph. We also guess that if there exists a vertex $v \in S^+(G)$ satisfying conditions $C_{max}(G - v) \leq C_{max}(G - w)$ for each $w \in S^+(G) \wedge w \neq v$, then $u_1 = v$ for $MaxQ(G)$.

Keywords: canonical labeling; k -mix-neighborhood subdigraph; algorithm; adjacency matrix; mix diffusion degree sequence; entire mix diffusion degree sequences

1. Introduction

A canonical labeling [1–3] of a graph, also called a canonical form [4], a *canonical code* [5], or an *optimum code* [6], is a unique string corresponding to the graph and is lexicographically smallest or largest according to the different definitions used in the studies. Two digraphs are isomorphic if and only if they have the same canonical labelings. Until now, the computation of the canonical labeling as the digraph isomorphism problem remains an unsolved problem in computational complexity theory in the sense that no polynomial-time algorithm exists for calculating the canonical labeling of a digraph. Therefore, the computation of the canonical labeling is *NP*-hard [4,7].

Numerous methods have emerged to calculate the canonical labelings of undirected graphs. However, different methods use distinct definitions of the canonical labeling. Given a graph with n vertices, Huan et al. concatenates the lower triangular entries (including the diagonal entries) of its

adjacency matrix to produce its canonical labeling [8]. Kuramochi and Karypis construct the canonical labeling by concatenating the columns of the upper-triangular portion of its adjacency matrix [9,10]. To create the canonical labeling, He et al. concatenate the rows of its adjacency matrix to form a n^2 binary number [11,12].

Babai and Luks pioneered the establishment of a general group-theoretic approach to compute canonical labeling [4]. However, combinatorial methods have worked well in various special cases. For random graphs, Babai et al. produce a canonical labeling with high probability [4,13]. Arvind et al. propose two corresponding logspace algorithms for partial 2- and 3-Trees [7,14].

Jianqiang Hao establishes a new theoretical framework to compute the canonical labeling $C_{max}(G)$ of undirected graphs by defining a set of concepts useful for classifying graphs [15].

Currently, *Nauty* [1,16–18] is the most popular and practical tools for considering the automorphism group and the *canonical label* of a graph or digraph. On isomorphism testing, *Nauty* is more efficient than Ullmann [19]. It has almost become the industry standard used to calculate the *canonical label*, as well as the automorphism group. For computing the canonical labeling and automorphism group, *Nauty* and [20] use the depth-first search to traverse the potential intermediate nodes in the search tree. The nodes of the search tree generated by *Nauty* are equitable ordered partitions of nodes in G . *Nauty* iteratively refines partitioning nodes until places the nodes that have the same properties into an automorphism orbit. As the partition refinement becomes finer and smaller, it automatically creates the *canonical label*. *Nauty* also requires exponential time to compute the canonical labeling for a given Miyazaki graph [21]. Tener and Deo [22] made improvements for dealing with the problem.

Besides *Nauty*, Bliss [3,23], Traces [2], and Conauto [24] are also state-of-the-art tools for graphs isomorphism testing. Based on backtracking, individualization of vertices, and partition refinement, *Bliss* [3,23] is an efficient canonical labeling tool for handling large and sparse graphs. Katebi et al. [25] can find the symmetry while calculating the canonical labeling. To fix the glitch of *Nauty*, *Traces* [2] uses the strategy of breadth-first search to find the automorphism group and the canonical labeling. Conauto also uses the basic individualization/refinement technique and is very fast for random graphs and several families of hard graphs.

For the improvement of performance, existing algorithms commonly utilize backtracking and orbit partitioning technique to avoid repeatedly visiting the same vertices, as well as manage to reduce the accessed nodes in the search tree. McKay and Piperno provides a comprehensive discussion of the issue [18].

Nauty dominated the field for several decades. As a result, an in-depth study for the canonical labeling has been confined to the theoretical framework of *Nauty*. This means that people just follow the research trajectory of *nauty* to extend and establish further study. When the graph under consideration contains a large number of automorphisms, it is difficult to verify the correctness of the canonical labeling obtained by performing *Nauty* according to the standard of *Nauty*.

Although many algorithms have emerged to calculate the canonical labelings of undirected graphs, to our best knowledge no algorithm other than *Nauty* exists for computing the canonical labelings of digraphs.

Throughout the paper, the canonical labeling $C_{max}(G)$ of a digraph is the lexicographically largest code obtained by concatenating the rows of the associated adjacency matrix (see Definition 7). Our definition of canonical labeling is entirely different from that of *Nauty*.

Although a few algorithms also give the same definition of the canonical labeling as described in Definition 7, their primary purpose is not to study how to construct a canonical labeling of a digraph, but for other intentions such as to mine the frequent subdigraphs. As a result, they can only work for some restricted undirected graph classes. Jianqiang Hao utilizes the $Cen(G)$ to calculate the first node u_1 added into $MaxQ(G)$ of simple undirected graphs [15]. Based on current knowledge and Definition 7, a general algorithm for computing the canonical labelings of digraphs is not available.

This paper focuses on the development of general theory and methods to calculate the canonical labeling $C_{max}(G)$ of digraphs. In the rest of this paper, Section 2 establishes some basic terminology and preliminary information. Section 3 describes the results accompanied by some discussion. Section 4 presents our algorithms for computing the canonical labeling of digraphs. Section 5 displays the implementation of our algorithms and evaluates our approach through many examples. Finally, Section 6 comments on our results and future work.

2. Preliminaries

In this section, we provide a brief review of the fundamental information used throughout the paper. A more comprehensive presentation can be found in most standard textbooks [26,27]. A directed graph (or digraph) $G = \langle V(G), E(G) \rangle$ consists of a nonempty set $V(G)$ of vertices (or nodes) and a set $E(G)$ of directed edges (or arcs). A directed edge associated with the ordered pair (u, v) is said to start at u and end at v . We also say that the vertex u is its tail and the vertex v is its head. Throughout the paper, we denote (u, v) by $u \rightarrow v$. For the edge $u \rightarrow v \in E(G)$, the vertex v is said to be adjacent from the vertex u , the vertex u is said to be adjacent to the vertex v , and the edge $u \rightarrow v$ is said to be incident from the vertex u and incident to the vertex v . The directed distance $\overrightarrow{d}(u, v)$ from vertex u to vertex v in G is the length of the shortest directed path from u to v [28], if any; otherwise $\overrightarrow{d}(u, v) = \infty$.

This paper deals with only finite simple digraphs, which have no loops and no multiple edges. For each $u \in V(G)$, let $d_G^{+-}(u) = |\{v | v \rightarrow u \wedge u \rightarrow v \in E(G)\}|$, $d_G^+(u) = |\{v | u \rightarrow v \in E(G)\}|$, $d_G^-(u) = |\{v | v \rightarrow u \in E(G)\}|$, and $d_G(u) = d_G^+(u) + d_G^-(u)$ denote the *outindegree*, *outdegree*, *indegree*, and *degree* of u respectively, and omit the subscript G when no ambiguity can arise. Denote by $\Delta(G)$, $\Delta^{+-}(G)$, $\Delta^+(G)$, and $\Delta^-(G)$ the *maximum degree*, the *maximum outindegree*, the *maximum outdegree*, and the *maximum indegree* of all vertices of a graph G , respectively.

Denote by $d(G) = (d_G(u_1), d_G(u_2), \dots, d_G(u_n))$ the *degree sequence* of G , by $d_G(V_1) = (d_G(v_1), d_G(v_2), \dots, d_G(v_m))$ the *degree sequence* of a subset $V_1 \subseteq V(G)$ with $v_i \in V_1, i = 1, 2, \dots, m$, and by $d_G(H) = (d_H(w_1), d_H(w_2), \dots, d_H(w_s))$ the *degree sequence* of a subdigraph $H \subseteq G$ with $w_j \in V(H), j = 1, 2, \dots, s$, and omit the subscript G when no ambiguity can arise.

In addition, denote by $d^+(G) = (d_G^+(u_1), d_G^+(u_2), \dots, d_G^+(u_n))$ the *outdegree sequence* of G , by $d_G^+(V_1) = (d_G^+(v_1), d_G^+(v_2), \dots, d_G^+(v_m))$ the *outdegree sequence* of a subset $V_1 \subseteq V(G)$ with $v_i \in V_1, i = 1, 2, \dots, m$, and by $d_G^+(H) = (d_H^+(w_1), d_H^+(w_2), \dots, d_H^+(w_s))$ the *outdegree sequence* of a subdigraph $H \subseteq G$ with $w_j \in V(H), j = 1, 2, \dots, s$, and omit the subscript G when no ambiguity can arise.

Similarly, denote by $d^-(G) = (d_G^-(u_1), d_G^-(u_2), \dots, d_G^-(u_n))$ the *indegree sequence* of G , by $d_G^-(V_1) = (d_G^-(v_1), d_G^-(v_2), \dots, d_G^-(v_m))$ the *indegree sequence* of a subset $V_1 \subseteq V(G)$ with $v_i \in V_1, i = 1, 2, \dots, m$, and by $d_G^-(H) = (d_H^-(w_1), d_H^-(w_2), \dots, d_H^-(w_s))$ the *indegree sequence* of a subdigraph $H \subseteq G$ with $w_j \in V(H), j = 1, 2, \dots, s$, and omit the subscript G when no ambiguity can arise.

Throughout this paper, unless otherwise specified, any given degree sequence is non-increasing. Throughout this paper, let $S(G) = \{u | u \in V(G) \wedge d(u) = \Delta(G)\}$, $S^+(G) = \{u \in V(G) | d^+(u) = \Delta^+(G)\}$, $S^-(G) = \{u \in V(G) | d^-(u) = \Delta^-(G)\}$, and $S^\pm(G) = \{u \in S^+(G) | d^-(u) = \Delta^-(S^+(G))\}$ where $\Delta^-(S^+(G)) = \max\{d^-(u) | u \in S^+(G)\}$.

Definition 1. Let $G = \langle V(G), E(G) \rangle$ be a simple digraph with n nodes. The vertex-induced subdigraph on $V_1 \subseteq V(G)$ of G is the subdigraph with the nodes set V_1 together with any directed edge whose endpoints are both in V_1 , denoted by $G[V_1]$.

Definition 2. Let $G = \langle V(G), E(G) \rangle$ and $H = \langle V(H), E(H) \rangle$ be two digraphs with n nodes. If there exists a bijection $f : V(G) \rightarrow V(H)$ such that $\forall u \rightarrow v \in E(G)$ if and only if $f(u) \rightarrow f(v) \in E(H)$. We say f is an isomorphic map of $G \rightarrow H$. Furthermore, we say that the graph G and H are isomorphic, denoted by $G \cong H$. An isomorphic map f of G onto itself is said to be an automorphism of G .

Definition 3. Let $G = \langle V(G), E(G) \rangle$ be a simple digraph with n nodes v_1, v_2, \dots, v_n . The adjacency matrix $A(G) = (a_{i,j})$ of G is a $n \times n$ -square matrix such that $a_{i,j} = 1$ if there is an edge $v_i \rightarrow v_j \in E(G)$ and $a_{i,j} = 0$ otherwise.

The adjacency matrix for a digraph G does not have to be symmetric, because there may not be an edge $v_i \rightarrow v_j$, when there is an edge from $v_j \rightarrow v_i$. In addition, $a_{i,i} = 0$ for every $v_i \in V(G)$ since G has no loops.

Given two vectors $X = (x_1, x_2, \dots, x_i, \dots, x_m)$ in R^m and $Y = (y_1, y_2, \dots, y_j, \dots, y_n)$ in R^n , the question arises as to how to decide which one is greater. The following conventions apply when comparing two vectors.

Definition 4. Let $X = (x_0, x_1, \dots, x_i, \dots, x_m)$ and $Y = (y_0, y_1, \dots, y_i, \dots, y_n)$ be two vectors in N (the collection of natural numbers) with $m \geq 0$ and $n \geq 0$, respectively. Then, defined the lexicographic order for the two vectors as follows:

1. $X = Y$, if $m = n$ and $x_i = y_i$ for all $0 \leq i \leq m$.
2. $X < Y$ if and only if either of the following is true.
 - (a) $\exists k, 0 \leq k \leq \min(m, n), x_i = y_i$ for $i < k, x_k < y_k$.
 - (b) $x_i = y_i$ for $0 \leq i \leq m$ and $m < n$.

Definition 5. Let $Z_1 = (X_0, X_1, \dots, X_i, \dots, X_m)$ and $Z_2 = (Y_0, Y_1, \dots, Y_j, \dots, Y_n)$ be two vectors with $m \geq 0$ and $n \geq 0$ respectively, with each $X_i, Y_j, i = 0, 1, \dots, m, j = 0, 1, \dots, n$ being a vector in N (the collection of natural numbers). Then, defined the lexicographic order for the two vectors as follows:

1. $Z_1 = Z_2$, if $m = n$ and $X_i = Y_i$ for all $0 \leq i \leq m$.
2. $Z_1 < Z_2$ if and only if either of the following is true.
 - (a) $\exists k, 0 \leq k \leq \min(m, n), X_i = Y_i$ for $i < k, x_k < y_k$.
 - (b) $X_i = Y_i$ for $0 \leq i \leq m$ and $m < n$.

Definition 6. Let $A = (a_{i,j})_{n \times n}$ and $B = (b_{i,j})_{n \times n}$ be two matrices with $a_{i,j}, b_{i,j} = 0, 1$ for $i, j = 1, 2, \dots, n$. Then, defined the lexicographic order for the two matrix as follows:

1. $A = B$, if $a_{i,j} = b_{i,j}$ for all $1 \leq i, j \leq n$.
2. $A < B$, if $\exists i, j, 1 \leq i, j \leq n$ satisfying conditions $a_{k,l} = b_{k,l}$ for all $k \leq i, l \leq j$, and $a_{i,j+1} < b_{i,j+1}$ with $j < n$ or $a_{i+1,1} < b_{i+1,1}$ with $i < n, j = n$.

Given a matrix X , if there is at least one positive element and the remaining elements are 0, we call $X > 0$. Otherwise, if all elements of X are 0, we call $X = 0$.

Definition 7. Let $G = \langle V(G), E(G) \rangle$ be a simple digraph with n nodes whose adjacency matrix is $A(G) = (a_{i,j})_{n \times n}$ (see (1)). To concatenate the rows of $A(G)$ according to the following order $a_{1,1}, a_{1,2}, \dots, a_{1,n}, a_{2,1}, a_{2,2}, \dots, a_{2,n}, \dots, a_{i,1}, a_{i,2}, \dots, a_{i,n}, \dots, a_{n,1}, a_{n,2}, \dots, a_{n,n}$ forms a corresponding binary number $a_{1,1}a_{1,2} \dots, a_{1,n} a_{2,1}a_{2,2} \dots a_{2,n} \dots a_{i,1}a_{i,2} \dots a_{i,n} \dots a_{n,1}a_{n,2} \dots a_{n,n}$, which is called a labeling of G , denoted by $C(G)$.

$$A(G) = \begin{pmatrix} \rightarrow a_{1,1} & \rightarrow a_{1,2} & \rightarrow a_{1,3} & \cdots & \cdots & \cdots & \cdots & \rightarrow a_{1,n} \\ \rightarrow a_{2,1} & \rightarrow a_{2,2} & \rightarrow a_{2,3} & \cdots & \cdots & \cdots & \cdots & \rightarrow a_{2,n} \\ \downarrow \vdots & \cdots & \ddots & \cdots & \cdots & \cdots & \cdots & \downarrow \vdots \\ \downarrow \vdots & \cdots & \cdots & \ddots & \cdots & \cdots & \cdots & \downarrow \vdots \\ \rightarrow a_{i,1} & \rightarrow a_{i,2} & \cdots & \rightarrow a_{i,i-1} & \rightarrow a_{i,i} & \rightarrow a_{i,i+1} & \cdots & \rightarrow a_{i,n} \\ \downarrow \vdots & \cdots & \cdots & \cdots & \cdots & \ddots & \cdots & \downarrow \vdots \\ \downarrow \vdots & \cdots & \cdots & \cdots & \cdots & \cdots & \ddots & \downarrow \vdots \\ \rightarrow a_{n,1} & \rightarrow a_{n,2} & \rightarrow a_{n,3} & \cdots & \cdots & \cdots & \rightarrow a_{n,n-1} & \rightarrow a_{n,n} \end{pmatrix} \quad (1)$$

The first row of $A(G)$ is the *labeling piece* 1 of $C(G)$, denoted by $C^1(G)$. Similarly, the second row is the *labeling piece* 2 of $C(G)$, denoted by $C^2(G)$. \dots . The n th row is the *labeling piece* n of $C(G)$, denoted by $C^n(G)$. It is clear that $C(G) = C^1(G)C^2(G) \cdots C^n(G)$.

A permutation π of the vertices of G is an arrangement of the n vertices without repetition. The number of permutations of the vertices of G is $n!$. Clearly each distinct permutation π of the n vertices of $V(G)$ defines a different adjacency matrix. Given a permutation π , one can obtain a *labeling* $C(G)$ corresponding to π by Definition 7. Denote by $L(G)$ the collection of all *labelings* of G .

For every $C_1(G), C_2(G) \in L(G)$, assume that $C_1(G) = i_1i_2 \cdots i_m, C_2(G) = j_1j_2 \cdots j_n$ with $i_1, i_2, \dots, i_m, j_1, j_2, \dots, j_n = 0$ or 1 . Let $X = (i_1, i_2, \dots, i_m)$ and $Y = (j_1, j_2, \dots, j_n)$. By Definition 4, if $X > Y$, then we call $C_1(G) > C_2(G)$. Otherwise, if $X < Y$, then we call $C_1(G) < C_2(G)$. Otherwise, if $X = Y$, then we call $C_1(G) = C_2(G)$.

It is clear that $(L(G), \leq)$ is a well-ordered set, where \leq denotes the less-than-or-equal-to binary relation on the set $L(G)$ expressed as above. By the well-ordering theorem, it follows that $L(G)$ has a minimum and maximum element, denoted by $C_{min}(G)$ and $C_{max}(G)$ respectively.

The two permutations of the n vertices of G corresponding to $C_{min}(G)$ and $C_{max}(G)$ are the *minimum and maximum node sequence*, denoted by $MinQ(G)$ and $MaxQ(G)$, respectively. Likewise, the two adjacency matrices of G corresponding to $C_{min}(G)$ and $C_{max}(G)$ are the *minimum and maximum canonical label matrix*, denoted by $A_{min}(G)$ and $A_{max}(G)$, individually.

$C^1(G), C^2(G), \dots, C^n(G)$ corresponding to $A_{min}(G)$ are *minimum canonical label piece* $1, 2, \dots, n$ of canonical labeling $C(G)$, denoted by $C^1_{min}(G), C^2_{min}(G), \dots, C^n_{min}(G)$, respectively. Likewise, $C^1(G), C^2(G), \dots, C^n(G)$ corresponding to $A_{max}(G)$ are *maximum canonical label piece* $1, 2, \dots, n$ of canonical labeling $C(G)$, denoted by $C^1_{max}(G), C^2_{max}(G), \dots, C^n_{max}(G)$, respectively.

Based on the above definitions, the following equations hold.

$$C_{min}(G) = C^1_{min}(G)C^2_{min}(G) \cdots C^n_{min}(G) \quad (2)$$

$$C_{max}(G) = C^1_{max}(G)C^2_{max}(G) \cdots C^n_{max}(G) \quad (3)$$

Theorem 1. Let $G = \langle V(G), E(G) \rangle$ and $H = \langle V(H), E(H) \rangle$ be two digraphs with n nodes. Their adjacency matrices are $A(G)$ and $A(H)$ respectively. $G \cong H$ if and only if $C_{max}(G) = C_{max}(H)$.

Definition 8. Let $G = \langle V(G), E(G) \rangle$ be a simple digraph with n nodes. The complement \bar{G} of G is a digraph satisfying the following condition: for all $u, v \in V(G), u \rightarrow v \in E(\bar{G})$ if and only if $u \rightarrow v \notin E(G)$.

Lemma 1. Let $G = \langle V(G), E(G) \rangle$ be a simple digraph with n nodes. $\bar{G} = \langle V(\bar{G}), E(\bar{G}) \rangle$ is the complement digraph of G . We have $C(G) = \bar{C}(\bar{G})$ and $C(\bar{G}) = \overline{C(G)}$.

Proof. The adjacency matrices of G and \bar{G} satisfy the condition

$$A(G) + A(\overline{G}) = J = \begin{pmatrix} 0 & 1 & \cdots & \cdots & 1 \\ 1 & 0 & 1 & \cdots & \vdots \\ \vdots & 1 & 0 & 1 & \vdots \\ \vdots & \vdots & 1 & \ddots & 1 \\ 1 & \cdots & \cdots & 1 & 0 \end{pmatrix}.$$

J is a $n \times n$ matrix of zeros and ones whose main diagonal elements are 0, and all other elements are 1. By $A(G) = J - A(\overline{G})$ and the complement graph \overline{G} , we have $C(G) = \overline{C(\overline{G})}$. Similarly, by $A(\overline{G}) = J - A(G)$, we have $C(\overline{G}) = \overline{C(G)}$ for the complement graph \overline{G} of a graph G . \square

Theorem 2. Let $G = \langle V(G), E(G) \rangle$ be a simple digraph with n nodes. $\overline{G} = \langle V(\overline{G}), E(\overline{G}) \rangle$ is the complement graph of G . We have

$$C_{min}(G) = \overline{C_{max}(\overline{G})} \tag{4}$$

$$C_{max}(G) = \overline{C_{min}(\overline{G})} \tag{5}$$

$$C_{min}(\overline{G}) = \overline{C_{max}(G)} \tag{6}$$

$$C_{max}(\overline{G}) = \overline{C_{min}(G)} \tag{7}$$

Proof. By Lemma 1, it follows that $C(G) = \overline{C(\overline{G})}$. Clearly if the k -bit of $C(\overline{G})$ is 0, the k -bit of $C(G)$ is 1, and vice versa. Therefore, one can easily get the $C_{max}(G)$ by performing a complement operation on $C_{min}(\overline{G})$. Similarly, by Lemma 1, the equality $C(\overline{G}) = \overline{C(G)}$ holds. Clearly if the k -bit of $C(G)$ is 0, the k -bit of $C(\overline{G})$ is 1, and vice versa. Accordingly, one can obtain the $C_{max}(\overline{G})$ of \overline{G} by performing a complement operation on $C_{min}(G)$.

Because $C(K_n)$ is a constant binary number, to minimize $C(G)$ one must maximize $C(\overline{G})$. On the contrary, to maximize $C(G)$, one must maximize $C(\overline{G})$. Similarly, to minimize $C(\overline{G})$ one must maximize $C(G)$. Contrarily to maximize $C(\overline{G})$, one must minimize $C(G)$. From the above analysis, the following equations hold.

$$\begin{aligned} C_{min}(G) &= \overline{C_{max}(\overline{G})} & C_{max}(G) &= \overline{C_{min}(\overline{G})} \\ C_{min}(\overline{G}) &= \overline{C_{max}(G)} & C_{max}(\overline{G}) &= \overline{C_{min}(G)} \end{aligned}$$

\square

By Theorem 2, it can be observed that if one has calculated the $C_{max}(\overline{G})$, one can easily get $C_{min}(G)$. Moreover, the calculation methods of $C_{max}(\overline{G})$ and $C_{max}(G)$ are same.

The paper focuses on the development of efficient methods to calculate $C_{max}(G)$. A *MaxEm digraph* is a digraph with the greatest $C(G)$ that corresponds to a permutation of the vertices.

For every $u \in V(G)$, the number of nodes with outdegree $d_G^+(u)$ is the *outdegree multiplicity* of u , denoted by $dm_G^+(u)$. In addition, unless otherwise specified, throughout this paper, the outdegree sequence is non-increasing.

Definition 9. Let $G = \langle V(G), E(G) \rangle$ be a simple digraph with n nodes. The open neighborhood, in-neighborhood, out-neighborhood, outin-neighborhood, and mix-neighborhood subdigraphs of a vertex u in G are subdigraphs of G defined as

$$\begin{aligned} N^-(u) &= (V(N^-(u)), E(N^-(u))), & N^-(u) &= (V(N^-(u)), E(N^-(u))), \\ N^+(u) &= (V(N^+(u)), E(N^+(u))), & N^{+-}(u) &= (V(N^{+-}(u)), E(N^{+-}(u))), \\ N^{++}(u) &= (V(N^{++}(u)), E(N^{++}(u))) \end{aligned} \tag{8}$$

where

$$V(N(u)) = \{v \in V(G) - u \mid v \rightarrow u \in E(G) \vee u \rightarrow v \in E(G)\}, \tag{9}$$

$$E(N(u)) = \{v \rightarrow w \in E(G) \mid v, w \in V(N(u))\},$$

$$V(N^-(u)) = \{v \in V(G) - u \mid v \rightarrow u \in E(G) \wedge u \rightarrow v \notin E(G)\}, \tag{10}$$

$$E(N^-(u)) = \{v \rightarrow w \in E(G) \mid v, w \in V(N^-(u))\},$$

$$V(N^+(u)) = \{v \in V(G) - u \mid u \rightarrow v \in E(G) \wedge v \rightarrow u \notin E(G)\}, \tag{11}$$

$$E(N^+(u)) = \{v \rightarrow w \in E(G) \mid v, w \in V(N^+(u))\},$$

$$V(N^{+-}(u)) = \{v \in V(G) - u \mid u \rightarrow v \in E(G) \wedge v \rightarrow u \in E(G)\}, \tag{12}$$

$$E(N^{+-}(u)) = \{v \rightarrow w \in E(G) \mid v, w \in V(N^{+-}(u))\}.$$

$$V(N^{++}(u)) = V(N^{+-}(u)) \cup V(N^+(u)), \tag{13}$$

$$E(N^{++}(u)) = \{v \rightarrow w \in E(G) \mid v, w \in V(N^{++}(u))\}.$$

The open k -neighborhood, k -in-neighborhood, k -out-neighborhood, k -outin-neighborhood, and k -mix-neighborhood subdigraphs of u with $k \geq 2$ are subdigraphs of G defined as

$$\begin{aligned} N_k(u) &= (V(N_k(u)), E(N_k(u))), & N_k^-(u) &= (V(N_k^-(u)), E(N_k^-(u))), \\ N_k^+(u) &= (V(N_k^+(u)), E(N_k^+(u))), & N_k^{+-}(u) &= (V(N_k^{+-}(u)), E(N_k^{+-}(u))), \\ N_k^{++}(u) &= (V(N_k^{++}(u)), E(N_k^{++}(u))) \end{aligned}$$

where

$$V(N_k(u)) = \{v \in V(G) - u \mid d(u, v) \leq k \vee d(v, u) \leq k\}, \tag{14}$$

$$E(N_k(u)) = \{v \rightarrow w \in E(G) \mid v, w \in V(N_k(u))\},$$

$$V(N_k^-(u)) = \{v \in V(G) - u \mid d(v, u) \leq k\}, \tag{15}$$

$$E(N_k^-(u)) = \{v \rightarrow w \in E(G) \mid v, w \in V(N_k^-(u))\},$$

$$V(N_k^+(u)) = \{v \in V(G) - u \mid d(u, v) \leq k\}, \tag{16}$$

$$E(N_k^+(u)) = \{v \rightarrow w \in E(G) \mid v, w \in V(N_k^+(u))\},$$

$$V(N_k^{+-}(u)) = \{v \in V(G) - u \mid d(u, v) = d(v, u) \leq k\}, \tag{17}$$

$$E(N_k^{+-}(u)) = \{v \rightarrow w \in E(G) \mid v, w \in V(N_k^{+-}(u))\}.$$

$$V(N_k^{++}(u)) = V(N_k^{+-}(u)) \cup V(N_k^+(u)), \tag{18}$$

$$E(N_k^{++}(u)) = \{v \rightarrow w \in E(G) \mid v, w \in V(N_k^{++}(u))\}.$$

Definition 10. Let $G = \langle V(G), E(G) \rangle$ be a simple digraph with n nodes. The close neighborhood, in-neighborhood, out-neighborhood, outin-neighborhood, and mix-neighborhood subdigraphs of a vertex u in G are subdigraphs of G defined as

$$\begin{aligned} N[u] &= (V(N[u]), E(N[u])), & N^-[u] &= (V(N^-[u]), E(N^-[u])), \\ N^+[u] &= (V(N^+[u]), E(N^+[u])), & N^{+-}[u] &= (V(N^{+-}[u]), E(N^{+-}[u])), \\ N^{++}[u] &= (V(N^{++}[u]), E(N^{++}[u])) \end{aligned}$$

where

$$V(N^-[u]) = V(N^-(u)) \cup \{u\}, \quad E(N^-[u]) = \{v \rightarrow w \in E(G) | v, w \in V(N^-[u])\}, \quad (19)$$

$$V(N^+[u]) = V(N^+(u)) \cup \{u\}, \quad E(N^+[u]) = \{v \rightarrow w \in E(G) | v, w \in V(N^+[u])\}, \quad (20)$$

$$V(N^{+-}[u]) = V(N^{+-}(u)) \cup \{u\}, \quad E(N^{+-}[u]) = \{v \rightarrow w \in E(G) | v, w \in V(N^{+-}[u])\}, \quad (21)$$

$$V(N^{++}[u]) = V(N^{+-}[u]) \cup V(N^+[u]), \quad E(N^{++}[u]) = \{v \rightarrow w \in E(G) | v, w \in V(N^{++}[u])\}. \quad (22)$$

The close k -neighborhood, k -in-neighborhood, k -out-neighborhood, k -outin-neighborhood, and k -mix-neighborhood subdigraphs of u with $k \geq 2$ are subdigraphs of G defined as

$$N_k^-[u] = (V(N_k^-[u]), E(N_k^-[u])), \quad N_k^+[u] = (V(N_k^+[u]), E(N_k^+[u])),$$

$$N_k^{+-}[u] = (V(N_k^{+-}[u]), E(N_k^{+-}[u])), \quad N_k^{++}[u] = (V(N_k^{++}[u]), E(N_k^{++}[u]))$$

where

$$V(N_k^-[u]) = V(N_k^-(u)) \cup \{u\}, \quad E(N_k^-[u]) = \{v \rightarrow w \in E(G) | v, w \in V(N_k^-[u])\}, \quad (24)$$

$$V(N_k^+[u]) = V(N_k^+(u)) \cup \{u\}, \quad E(N_k^+[u]) = \{v \rightarrow w \in E(G) | v, w \in V(N_k^+[u])\}, \quad (25)$$

$$V(N_k^{+-}[u]) = V(N_k^{+-}(u)) \cup \{u\}, \quad E(N_k^{+-}[u]) = \{v \rightarrow w \in E(G) | v, w \in V(N_k^{+-}[u])\}, \quad (26)$$

$$V(N_k^{++}[u]) = V(N_k^{+-}[u]) \cup V(N_k^+[u]), \quad E(N_k^{++}[u]) = \{v \rightarrow w \in E(G) | v, w \in V(N_k^{++}[u])\}. \quad (27)$$

Definition 11. Let $G = \langle V(G), E(G) \rangle$ be a simple digraph with n nodes. The open neighborhood, in-neighborhood, out-neighborhood, outin-neighborhood, and mix-neighborhood subdigraphs of a nodes set $Q \subseteq V(G)$ are subdigraphs of G defined as

$$N^-(Q) = (V(N^-(Q)), E(N^-(Q))), \quad N^+(Q) = (V(N^+(Q)), E(N^+(Q))),$$

$$N^{+-}(Q) = (V(N^{+-}(Q)), E(N^{+-}(Q))), \quad N^{++}(Q) = (V(N^{++}(Q)), E(N^{++}(Q)))$$

where

$$V(N^-(Q)) = \{v \in V(G) - Q \mid \exists u \in Q \wedge (v \rightarrow u \in E(G) \vee u \rightarrow v \in E(G))\}, \quad (29)$$

$$E(N^-(Q)) = \{v \rightarrow w \in E(G) \mid v, w \in V(N^-(Q))\},$$

$$V(N^+(Q)) = \{v \in V(G) - Q \mid \exists u \in Q \wedge v \rightarrow u \in E(G) \wedge u \rightarrow v \notin E(G)\}, \quad (30)$$

$$E(N^+(Q)) = \{v \rightarrow w \in E(G) \mid v, w \in V(N^+(Q))\},$$

$$V(N^{+-}(Q)) = \{v \in V(G) - Q \mid \exists u \in Q \wedge u \rightarrow v \in E(G) \wedge v \rightarrow u \notin E(G)\}, \quad (31)$$

$$E(N^{+-}(Q)) = \{v \rightarrow w \in E(G) \mid v, w \in V(N^{+-}(Q))\},$$

$$V(N^{++}(Q)) = \{v \in V(G) - Q \mid \exists u \in Q \wedge u \rightarrow v \in E(G) \wedge v \rightarrow u \in E(G)\}, \quad (32)$$

$$E(N^{++}(Q)) = \{v \rightarrow w \in E(G) \mid v, w \in V(N^{++}(Q))\},$$

$$V(N^{++}(Q)) = V(N^{+-}(Q)) \cup V(N^+(Q)), \quad (33)$$

$$E(N^{++}(Q)) = \{v \rightarrow w \in E(G) \mid v, w \in V(N^{++}(Q))\}.$$

The open k -neighborhood, k -in-neighborhood, k -out-neighborhood, k -outin-neighborhood, and k -mix-neighborhood subdigraphs of Q with $k \geq 2$ are subdigraphs of G defined as

$$\begin{aligned} N_k^-(Q) &= (V(N_k^-(Q)), E(N_k^-(Q))), & N_k^+(Q) &= (V(N_k^+(Q)), E(N_k^+(Q))), \\ N_k^+(Q) &= (V(N_k^+(Q)), E(N_k^+(Q))), & N_k^{+-}(Q) &= (V(N_k^{+-}(Q)), E(N_k^{+-}(Q))), \\ N_k^{++}(Q) &= (V(N_k^{++}(Q)), E(N_k^{++}(Q))) \end{aligned}$$

where

$$V(N_k^-(Q)) = \{v \in V(G) - Q \mid \exists u \in Q \wedge (d(v, u) \leq k \vee d(u, v) \leq k)\}, \tag{34}$$

$$E(N_k^-(Q)) = \{v \rightarrow w \in E(G) \mid v, w \in V(N_k^-(Q))\},$$

$$V(N_k^+(Q)) = \{v \in V(G) - Q \mid \exists u \in Q \wedge d(v, u) \leq k\}, \tag{35}$$

$$E(N_k^+(Q)) = \{v \rightarrow w \in E(G) \mid v, w \in V(N_k^+(Q))\},$$

$$V(N_k^{+-}(Q)) = \{v \in V(G) - Q \mid \exists u \in Q \wedge d(u, v) \leq k\}, \tag{36}$$

$$E(N_k^{+-}(Q)) = \{v \rightarrow w \in E(G) \mid v, w \in V(N_k^{+-}(Q))\},$$

$$V(N_k^{++}(Q)) = \{v \in V(G) - Q \mid \exists u \in Q \wedge d(u, v) = d(v, u) \leq k\}, \tag{37}$$

$$E(N_k^{++}(Q)) = \{v \rightarrow w \in E(G) \mid v, w \in V(N_k^{++}(Q))\},$$

$$V(N_k^{++}(Q)) = V(N_k^{+-}(Q)) \cup V(N_k^+(Q)), \tag{38}$$

$$E(N_k^{++}(Q)) = \{v \rightarrow w \in E(G) \mid v, w \in V(N_k^{++}(Q))\}.$$

Definition 12. Let $G = \langle V(G), E(G) \rangle$ be a simple digraph with n nodes. The close neighborhood, in-neighborhood, out-neighborhood, outin-neighborhood, and mix-neighborhood subdigraphs of a nodes set $Q \subseteq V(G)$ are subdigraphs of G defined as

$$\begin{aligned} N^-(Q) &= (V(N^-(Q)), E(N^-(Q))), & N^+(Q) &= (V(N^+(Q)), E(N^+(Q))), \\ N^+(Q) &= (V(N^+(Q)), E(N^+(Q))), & N^{+-}(Q) &= (V(N^{+-}(Q)), E(N^{+-}(Q))), \\ N^{++}(Q) &= (V(N^{++}(Q)), E(N^{++}(Q))) \end{aligned}$$

where

$$V(N^-(Q)) = V(N^-(Q)) \cup Q, \quad E(N^-(Q)) = \{v \rightarrow w \in E(G) \mid v, w \in V(N^-(Q))\}, \tag{39}$$

$$V(N^+(Q)) = V(N^+(Q)) \cup Q, \quad E(N^+(Q)) = \{v \rightarrow w \in E(G) \mid v, w \in V(N^+(Q))\}, \tag{40}$$

$$V(N^{+-}(Q)) = V(N^{+-}(Q)) \cup Q, \quad E(N^{+-}(Q)) = \{v \rightarrow w \in E(G) \mid v, w \in V(N^{+-}(Q))\}, \tag{41}$$

$$V(N^{++}(Q)) = V(N^{++}(Q)) \cup Q, \quad E(N^{++}(Q)) = \{v \rightarrow w \in E(G) \mid v, w \in V(N^{++}(Q))\}, \tag{42}$$

$$V(N^{++}(Q)) = V(N^{+-}(Q)) \cup V(N^+(Q)), \quad E(N^{++}(Q)) = \{v \rightarrow w \in E(G) \mid v, w \in V(N^{++}(Q))\}. \tag{43}$$

The close k -neighborhood, k -in-neighborhood, k -out-neighborhood, k -outin-neighborhood, and k -mix-neighborhood subdigraphs of Q with $k \geq 2$ are subdigraphs of G defined as

$$\begin{aligned} N_k^-(Q) &= (V(N_k^-(Q)), E(N_k^-(Q))), & N_k^+(Q) &= (V(N_k^+(Q)), E(N_k^+(Q))), \\ N_k^+(Q) &= (V(N_k^+(Q)), E(N_k^+(Q))), & N_k^{+-}(Q) &= (V(N_k^{+-}(Q)), E(N_k^{+-}(Q))), \\ N_k^{++}(Q) &= (V(N_k^{++}(Q)), E(N_k^{++}(Q))) \end{aligned}$$

where

$$V(N_k [Q]) = V(N_k (Q)) \cup Q, \quad E(N_k [Q]) = \{v \rightarrow w \in E(G) | v, w \in V(N_k [Q])\}, \quad (44)$$

$$V(N_k^- [Q]) = V(N_k^- (Q)) \cup Q, \quad E(N_k^- [Q]) = \{v \rightarrow w \in E(G) | v, w \in V(N_k^- [Q])\}, \quad (45)$$

$$V(N_k^+ [Q]) = V(N_k^+ (Q)) \cup Q, \quad E(N_k^+ [Q]) = \{v \rightarrow w \in E(G) | v, w \in V(N_k^+ [Q])\}, \quad (46)$$

$$V(N_k^{+-} [Q]) = V(N_k^{+-} (Q)) \cup Q, \quad E(N_k^{+-} [Q]) = \{v \rightarrow w \in E(G) | v, w \in V(N_k^{+-} [Q])\}, \quad (47)$$

$$V(N_k^{++} [Q]) = V(N_k^{+-} [Q]) \cup V(N_k^+ [Q]), \quad E(N_k^{++} [Q]) = \{v \rightarrow w \in E(G) | v, w \in V(N_k^{++} [Q])\}. \quad (48)$$

In the following section, unless otherwise specified, each k -neighborhood, k -in-neighborhood, k -out-neighborhood, k -outin-neighborhood, and k -mix-neighborhood subdigraphs are closed.

A digraph is connected if its underlying graph is connected. For $k = 1$, we omit the subscript 1 and let $N^{++}(u) = N_1^{++}(u)$, $N^{++}[u] = N_1^{++}[u]$, $N^{++}(Q) = N_1^{++}(Q)$, and $N^{++}[Q] = N_1^{++}[Q]$.

Definition 13. Let $G = \langle V(G), E(G) \rangle$ be a simple digraph with n nodes. Suppose that $H \subseteq G$ and $u \in V(H)$. We denote by $N_H^k(u)$ the open k -neighborhood subdigraph of u in H , by $N_H^k[u]$ the close k -neighborhood subdigraph of u in H , by $N_H^{k++}(u)$ the open k -mix-neighborhood subdigraph of u in H , and by $N_H^{k++}[u]$ the close k -mix-neighborhood subdigraph of u in H .

For $k = 1$, we omit the superscript 1 and let $N_H(u) = N_H^1(u)$, $N_H[u] = N_H^1[u]$, $N_H^{++}(u) = N_H^{1++}(u)$, and $N_H^{++}[u] = N_H^{1++}[u]$.

Definition 14. Let $G = \langle V(G), E(G) \rangle$ be a simple connected digraph with n nodes. For every $u \in V(G)$, there exists a positive integer k satisfying conditions $N_{k-1}[u] \subset G$ and $N_k[u] = G$. The value of k is called the diffusion radius of u denoted by $\rho_G[u]$, and the subscript G can be omitted when no ambiguity can arise.

By Definition 14, it is clear that $N_{\rho[u]}[u] = G$ for every u in G .

Definition 15. Let $G = \langle V(G), E(G) \rangle$ be a simple connected digraph with n nodes. For every $u \in V(G)$, there exists a positive integer k satisfying conditions $N_{k-1}^{++}[u] \subset G$ and $N_k^{++}[u] = N_{k-1}^{++}[u]$. The value of k is called the mix diffusion radius of u denoted by $\rho_G^{++}[u]$, and the subscript G can be omitted when no ambiguity can arise.

Definition 16. Let $G = \langle V(G), E(G) \rangle$ be a simple digraph and u be a node in G whose close k neighborhood subdigraph is $N_k[u]$ with $k = 1, 2, \dots$. A node in $N[u]$ is a one diffusion node of u . For $k > 1$, a node v in $N_k[u]$ satisfying condition $v \notin N_{k-1}[u]$ is a k diffusion node of u .

Definition 17. Let $G = \langle V(G), E(G) \rangle$ be a simple digraph and u be a node in G whose close k -mix-neighborhood subdigraph is $N_k^{++}[u]$ with $k = 1, 2, \dots$. A node in $N^{++}[u]$ is a one mix diffusion node of u . For $k > 1$, a node v in $N_k^{++}[u]$ satisfying condition $v \notin N_{k-1}^{++}[u]$ is a k mix diffusion node of u .

Every node v in G is assigned an attribute $m_NearestNode$ whose function, described in Section 3.1.2.

Definition 18. Let $G = \langle V(G), E(G) \rangle$ be a simple digraph and u be a node in G whose close k neighborhood subdigraph is $N_k[u]$ with $k = 1, 2, \dots$. Assume that H is a connected component of $N_k[u]$ with $k \geq 1$. Assume that $V_m \subseteq V(H)$ with $m = 0, 1, 2, \dots, t$, where V_0 is in ascending order of attribute $m_NearestNode$ with $v \rightarrow m_NearestNode \geq 1$ for every $v \in V_0$, and V_1, V_2, \dots, V_t contain the $1, 2, \dots, k$ diffusion nodes of u respectively, satisfying conditions $V_0 \cup V_1 \cup \dots \cup V_t = V(H)$ and $V_i \cap V_j = \emptyset$ for $i \neq j, i, j = 0, 1, \dots, t$.

We define $d_{\sigma}^+(H) = (d_{N_k[u]}^+(V_0), \dots, d_{N_k[u]}^+(V_i), \dots, d_{N_k[u]}^+(V_t))$ to be the diffusion outdegree sequence of H where $d_{N_k[u]}^+(V_i)$ with $i = 0, 1, \dots, t$ are the outdegree sequences in descending order induced by all vertices in V_i respectively.

Definition 19. Let $G = \langle V(G), E(G) \rangle$ be a simple digraph and u be a node in G whose close k -mix-neighborhood subdigraph is $N_k^{++}[u]$ with $k = 1, 2, \dots$. Assume that H is a connected component of $N_k^{++}[u]$ with $k \geq 1$. Assume that $V_m \subseteq V(H)$ with $m = 0, 1, 2, \dots, t$, where V_0 is in ascending order of attribute $m_NearestNode$ with $v - > m_NearestNode \geq 1$ for every $v \in V_0$, and V_1, V_2, \dots, V_t contain the $1, 2, \dots, k$ mix diffusion nodes of u respectively, satisfying conditions $V_0 \cup V_1 \cup \dots \cup V_t = V(H)$ and $V_i \cap V_j = \emptyset$ for $i \neq j, i, j = 0, 1, \dots, t$.

We define $d_\tau^+(H) = (d_{N_k^{++}[u]}^+(V_0), \dots, d_{N_k^{++}[u]}^+(V_i), \dots, d_{N_k^{++}[u]}^+(V_t))$ to be the mix diffusion outdegree sequence of H where $d_{N_k^{++}[u]}^+(V_i)$ with $i = 0, 1, \dots, t$ are the outdegree sequences in descending order induced by all vertices in V_i respectively.

Definition 20. Let $G = \langle V(G), E(G) \rangle$ be a simple digraph and u be a node in G whose close k neighborhood subdigraph is $N_k[u]$ with $k = 1, 2, \dots$. Assume that $N_k[u]$ has p connected components H_1, H_2, \dots, H_p with diffusion outdegree sequences $d_\sigma^+(H_1), d_\sigma^+(H_2), \dots, d_\sigma^+(H_p)$ respectively, satisfying conditions $d_\sigma^+(H_1) \geq d_\sigma^+(H_2) \geq \dots \geq d_\sigma^+(H_p)$.

Define $d_G^\sigma[N_k(u)] = (d_\sigma(H_1), d_\sigma(H_2), \dots, d_\sigma(H_p))$ to be the entire diffusion outdegree sequence of $N_k(u)$ pertaining to u in G , and omit the subscript G when no ambiguity can arise.

Definition 21. Let $G = \langle V(G), E(G) \rangle$ be a simple digraph and u be a node in G whose close k -mix neighborhood subdigraph is $N_k^{++}[u]$ with $k = 1, 2, \dots$. Assume that $N_k^{++}[u]$ has p connected components H_1, H_2, \dots, H_p with mix diffusion outdegree sequences $d_\tau^+(H_1), d_\tau^+(H_2), \dots, d_\tau^+(H_p)$ respectively, satisfying conditions $d_\tau^+(H_1) \geq d_\tau^+(H_2) \geq \dots \geq d_\tau^+(H_p)$.

Define $d_G^\tau[N_k^{++}[u]] = (d_\tau^+(H_1), d_\tau^+(H_2), \dots, d_\tau^+(H_p))$ to be the entire mix diffusion outdegree sequence of $N_k^{++}[u]$ pertaining to u in G , and omit the subscript G when no ambiguity can arise.

Definition 22. Let $G = \langle V(G), E(G) \rangle$ be a simple digraph with n nodes. For every $u, v \in V(G)$ with $u \neq v$ and , let $N_1^{++}[u], N_2^{++}[u], \dots, N_{\rho^{++}[u]}^{++}[u]$ be the close $1, 2, \dots, \rho^{++}[u]$ mix-neighborhood subdigraph of u with entire mix diffusion outdegree sequences $d^\tau[N_1^{++}[u]], d^\tau[N_2^{++}[u]], \dots, d^\tau[N_{\rho^{++}[u]}^{++}[u]]$, respectively. Let $N_1^{++}[v], N_2^{++}[v], \dots, N_{\rho^{++}[v]}^{++}[v]$ be the $1, 2, \dots, \rho^{++}[v]$ mix-neighborhood subdigraph of v with entire mix diffusion outdegree sequences $d^\tau[N_1^{++}[v]], d^\tau[N_2^{++}[v]], \dots, d^\tau[N_{\rho^{++}[v]}^{++}[v]]$, respectively. Let $Z_1 = (d^\tau[N_1^{++}[u]], d^\tau[N_2^{++}[u]], \dots, d^\tau[N_{\rho^{++}[u]}^{++}[u]])$ and $Z_2 = (d^\tau[N_1^{++}[v]], d^\tau[N_2^{++}[v]], \dots, d^\tau[N_{\rho^{++}[v]}^{++}[v]])$. If $Z_1 > Z_2$, we write $u \succ v$ with respect to G . Otherwise, if $Z_1 < Z_2$, we write $u \prec v$ with respect to G . Otherwise, if $Z_1 = Z_2$, we write $u \asymp v$ with respect to G , and omit the symbol G when no ambiguity can arise. Denote $u \succ v$ or $u \asymp v$ by $u \succcurlyeq v$ and $u \prec v$ or $u \asymp v$ by $u \preccurlyeq v$.

It is clear that $\succ, \prec, \asymp, \succcurlyeq$ define a binary relation on the set of nodes $V(G)$. By Definition 22, for every $u, v \in V(G)$ with $u \neq v$, one of the following statements is true: (1) $u \succ v$. (2) $u \prec v$. (3) $u \asymp v$.

It can be shown that $(V(G), \succcurlyeq)$ is a well-ordered set, where \succcurlyeq denotes the binary relation $u \succcurlyeq v$ on the set $V(G)$. By the well-ordering theorem, it follows that there exists a maximum and minimum element in $V(G)$, denoted by G_{max}^{\succcurlyeq} and G_{min}^{\succcurlyeq} respectively with $G_{max}^{\succcurlyeq} \in V(G)$ and $G_{min}^{\succcurlyeq} \in V(G)$. The symbol \succcurlyeq can be omitted if no confusion arises. The following Lemmas 2–4 immediately follow from Definition 22.

Lemma 2. Let $G = \langle V(G), E(G) \rangle$ be a simple digraph with n nodes. For every $u, v \in V(G)$ with $u \neq v$, if the symbol \succcurlyeq denotes the binary relation $u \succcurlyeq v$ on the set $V(G)$, then, all of the nodes in G form a single chain L on G : $v_1 \succcurlyeq v_2 \succcurlyeq \dots \succcurlyeq v_i \succcurlyeq \dots \succcurlyeq v_n$ with $v_i \in V(G), i = 1, 2, \dots, n$.

Lemma 3. Let $G = \langle V(G), E(G) \rangle$ be a simple digraph with n nodes. Let $S^+(G) = \{u \in V(G) \mid d^+(u) = \Delta^+(G) = s\}$. For every $u, v \in S^+(G)$ with $u \neq v$, if the symbol \succcurlyeq denotes the binary relation $u \succcurlyeq v$ on the set $S^+(G)$, then, all of the nodes in $S^+(G)$ form a single chain L on G : $v_1 \succcurlyeq v_2 \succcurlyeq \dots \succcurlyeq v_i \succcurlyeq \dots \succcurlyeq v_s$ with $v_i \in S^+(G), i = 1, 2, \dots, s$.

Lemma 4. Let $G = \langle V(G), E(G) \rangle$ be a simple digraph with n nodes. For every $u \in V(G)$, suppose that $N_k^{++}[u]$ is the close k mix-neighborhood subdigraph of u in G . Let $S^+(N_k^{++}[u]) = \{v \in V(N_k^{++}[u]) \mid d^+(v) = \Delta^+(N_k^{++}[u]) = t\}$.

For every $v, w \in S^+(N_k^{++}[u])$ with $v \neq w$, if the symbol \succ denotes the binary relation $v \succ w$ on the set $S^+(N_k^{++}[u])$, then, all of the nodes in $S^+(N_k^{++}[u])$ form a single chain L on $N_k^{++}[u]$: $v_1 \succ v_2 \succ \dots \succ v_i \succ \dots \succ v_t$ with $v_i \in S^+(N_k^{++}[u]), i = 1, 2, \dots, t$.

The conclusions in the following Propositions 1 and 2 are obvious by Definitions 4, 5, and 22.

Proposition 1. Let $G = \langle V(G), E(G) \rangle$ be a simple digraph with n nodes. For every $u, v \in V(G)$ with $u \neq v$ and , let $N_1^{++}[u], N_2^{++}[u], \dots, N_{\rho^{++}[u]}^{++}[u]$ be the close $1, 2, \dots, \rho^{++}[u]$ mix-neighborhood subdigraph of u with entire mix diffusion outdegree sequences $d^\tau[N_1^{++}[u]], d^\tau[N_2^{++}[u]], \dots, d^\tau[N_{\rho^{++}[u]}^{++}[u]]$, respectively. Let $N_1^{++}[v], N_2^{++}[v], \dots, N_{\rho^{++}[v]}^{++}[v]$ be the close $1, 2, \dots, \rho^{++}[v]$ mix-neighborhood subdigraph of v with entire mix diffusion outdegree sequences $d^\tau[N_1^{++}[v]], d^\tau[N_2^{++}[v]], \dots, d^\tau[N_{\rho^{++}[v]}^{++}[v]]$, respectively. Let $Z_1 = (d^\tau[N_1^{++}[u]], d^\tau[N_2^{++}[u]], \dots, d^\tau[N_{\rho^{++}[u]}^{++}[u]]) = (X_1, Y_1)$ with $X_1 = (d^\tau[N_1^{++}[u]], d^\tau[N_2^{++}[u]], \dots, d^\tau[N_k^{++}[u]])$ and $Y_1 = (d^\tau[N_{k+1}^{++}[u]], d^\tau[N_{k+2}^{++}[u]], \dots, d^\tau[N_{\rho^{++}[u]}^{++}[u]])$. Let $Z_2 = (d^\tau[N_1^{++}[v]], d^\tau[N_2^{++}[v]], \dots, d^\tau[N_{\rho^{++}[v]}^{++}[v]]) = (X_2, Y_2)$ with $X_2 = (d^\tau[N_1^{++}[v]], d^\tau[N_2^{++}[v]], \dots, d^\tau[N_k^{++}[v]])$ and $Y_2 = (d^\tau[N_{k+1}^{++}[v]], d^\tau[N_{k+2}^{++}[v]], \dots, d^\tau[N_{\rho^{++}[v]}^{++}[v]])$. If $X_1 = X_2$, then $Y_1 > Y_2$ leads to $Z_1 > Z_2$. Otherwise, $Y_1 < Y_2$ leads to $Z_1 < Z_2$. Otherwise, $Y_1 = Y_2$ leads to $Z_1 = Z_2$. Accordingly, it follows that if $X_1 = X_2$, then $Y_1 > Y_2$ leads to $u \succ v$. Otherwise, $Y_1 < Y_2$ leads to $u \prec v$. Otherwise, $Y_1 = Y_2$ leads to $u \asymp v$.

Proposition 2. Let $G = \langle V(G), E(G) \rangle$ be a simple digraph with n nodes. For every $u, v \in V(G)$ with $u \neq v$ and , let $N_1^{++}[u], N_2^{++}[u], \dots, N_{\rho^{++}[u]}^{++}[u]$ be the close $1, 2, \dots, \rho^{++}[u]$ mix-neighborhood subdigraph of u with entire mix diffusion outdegree sequences $d^\tau[N_1^{++}[u]], d^\tau[N_2^{++}[u]], \dots, d^\tau[N_{\rho^{++}[u]}^{++}[u]]$, respectively. Let $N_1^{++}[v], N_2^{++}[v], \dots, N_{\rho^{++}[v]}^{++}[v]$ be the close $1, 2, \dots, \rho^{++}[v]$ mix-neighborhood subdigraph of v with entire mix diffusion outdegree sequences $d^\tau[N_1^{++}[v]], d^\tau[N_2^{++}[v]], \dots, d^\tau[N_{\rho^{++}[v]}^{++}[v]]$, respectively. If $d^\tau[N_1^{++}[u]] = d^\tau[N_1^{++}[v]], d^\tau[N_2^{++}[u]] = d^\tau[N_2^{++}[v]], \dots, d^\tau[N_{k-1}^{++}[u]] = d^\tau[N_{k-1}^{++}[v]], d^\tau[N_k^{++}[u]] > d^\tau[N_k^{++}[v]]$, then $u \succ v$ with respect to G (see Definition 22).

3. Results and Discussion

Let $G = \langle V(G), E(G) \rangle$ be a simple digraph with n nodes. In the section, we will study how to compute the maximum element $C_{max}(G)$ of the digraph G . Without loss of generality, let $MaxQ(G) = (u_1, u_2, \dots, u_i, \dots, u_n)$. Throughout the paper, our algorithms mentioned use an adjacency list to store the digraph G .

3.1. Compute $C_{max}(G)$ of the Digraph G

In this subsection, we examine how to compute the maximum element $C_{max}(G)$ of a digraph G . What approach should one take to calculate the maximum element $C_{max}(G)$? From the connection between $C_{max}(G)$ and $A_{max}(G)$, any method for calculating $C_{max}(G)$ must first obtain the permutation $MaxQ(G)$ corresponding to the adjacency matrix $A_{max}(G)$.

3.1.1. Compute the First Node u_1 Added into $MaxQ(G)$

In this sub-subsection, we examine how to compute the first vertex u_1 of $MaxQ(G)$. Here, assume that G is a connected digraph of order $n > 1$. Note that to maximize $C(G)$ one must let $a_{1,2} = 1$ (see (1)). $a_{1,2} = 1$ can always be achieved since G is connected with order $n > 1$. Furthermore, to obtain $C_{max}(G)$, one must select u_1 from $S^+(G)$. Only by so doing, there can be more "1"s in the high bits of $C_{max}^1(G)$ such that ensures maximum $C(G)$. Otherwise, $C(G)$ cannot reach the maximum value. From preceding discussion, we get the following result.

Proposition 3. Let $G = \langle V(G), E(G) \rangle$ be a simple digraph with n nodes. Let $S^+(G) = \{u \in V(G) \mid d^+(u) = \Delta^+(G)\}$. Then the selection of u_1 of $MaxQ(G)$ is from $S^+(G)$ for obtaining $C_{max}(G)$.

Proof. Let us assume that $t = \Delta^+(G)$. By (1), it follows that $C_{max}^1(G) = a_{1,1}a_{1,2}a_{1,3} \cdots a_{1,n}$. Since $\Delta^+(G) = t$, it can be shown that $a_{1,1} = 0, a_{1,2} = 1, \dots, a_{1,t+1} = 1$ for $C_{max}^1(G)$. Therefore, the conclusion of Proposition 3 holds. \square

Proposition 4. Let $G = \langle V(G), E(G) \rangle$ be a simple digraph with n nodes. Let $S^+(G) = \{u \in V(G) \mid d^+(u) = \Delta^+(G)\}$. If $|S^+(G)| = 1$ with $v \in S^+(G)$, then $u_1 = v$ for $MaxQ(G)$.

Proof. By Proposition 3, it immediately follows that the selection of u_1 of $MaxQ(G)$ is from $S^+(G)$ for obtaining $C_{max}(G)$. Since $|S^+(G)| = 1$, therefore, $u_1 = v$ for $MaxQ(G)$. \square

Lemma 5. Let $G = \langle V(G), E(G) \rangle$ be a simple digraph with n nodes. Let $S^+(G) = \{u \in V(G) \mid d^+(u) = \Delta^+(G) = s\}$ with $|S^+(G)| > 1$. Let S_1 be the set of all vertices $v \in S^+(G)$ satisfying $|V(N^{+-}(v))| > 0$. Then, $u_1 \in S_1$ for $MaxQ(G)$.

Proof. By Proposition 3, clearly $u_1 \in S^+(G)$ is true. Since G is a simple digraph with n nodes, then $C_{max}^1(G) = a_{1,1}a_{1,2}a_{1,3} \cdots a_{1,n}$ with $a_{1,1} = 0$ and $C_{max}^2(G) = a_{2,1}a_{2,2}a_{2,3} \cdots a_{2,n}$ with $a_{2,2} = 0$ (see (1)). Since $\Delta^+(G) = s$, it follows that $a_{1,2} = a_{1,3} = \dots = a_{1,s+1} = 1$ and $a_{1,s+2} = a_{1,s+3} = \dots = a_{1,n} = 0$ for $C_{max}^1(G)$.

If $u_1 \notin S_1$, one can assert that $u_1 \in S^+(G) - S_1$. As a result, $|V(N^{+-}(u_1))| = 0$ holds by the condition of Lemma 5. Therefore, $C_{max}^2(G) = a_{2,1}a_{2,2}a_{2,3} \cdots a_{2,n}$ with $a_{2,1} = 0$ and $a_{2,2} = 0$.

Otherwise, if $u_1 \in S_1$, then $|V(N^{+-}(u_1))| > 0$ holds by the condition of Lemma 5. Therefore, $C_{max}^2(G) = a_{2,1}a_{2,2}a_{2,3} \cdots a_{2,n}$ with $a_{2,1} = 1$ and $a_{2,2} = 0$.

By comparing the above two results obtained for $C_{max}^2(G)$, we have that $u_1 \in S_1$ holds for $MaxQ(G)$. \square

Theorem 3. Let $G = \langle V(G), E(G) \rangle$ be a simple digraph with n nodes. Let $S^+(G) = \{u \in V(G) \mid d^+(u) = \Delta^+(G) = s\}$ with $|S^+(G)| > 1$. Assume that $v \in S^+(G)$ have the open outin-neighborhood subdigraph $N^{+-}(v)$ with $|V(N^{+-}(v))| > 0$ (see Definition 9). Suppose that there exists a node $v_1 \in V(N^{+-}(v))$ satisfying condition $d_H^+(v_1) = \Delta^+(H)$ where $H = N^{++}(v)$.

For each $w \in S^+(G) \wedge w \neq v$ with $|V(N^{+-}(w))| > 0$, suppose that there exists a node $w_1 \in V(N^{+-}(w))$ satisfying condition $d_M^+(w_1) = \Delta^+(M)$ where $M = N^{++}(w)$. If condition $d_H^+(v_1) > d_M^+(w_1)$ is satisfied, then $u_1 = v$ for $MaxQ(G)$.

Proof. Since G is a simple digraph with n nodes, it follows that $C_{max}^1(G) = a_{1,1}a_{1,2}a_{1,3} \cdots a_{1,n}$ with $a_{1,1} = 0$ and $C_{max}^2(G) = a_{2,1}a_{2,2}a_{2,3} \cdots a_{2,n}$ with $a_{2,2} = 0$ (see (1)). Since $\Delta^+(G) = s$, it follows that $a_{1,2} = a_{1,3} = \dots = a_{1,s+1} = 1$ for $C_{max}^1(G)$.

By the conditions of Theorem 3, clearly $d_H^+(v_1) = \Delta^+(H)$ and $d_M^+(w_1) = \Delta^+(M)$. For simplicity, let us assume that $r = d_H^+(v_1), t = d_M^+(w_1)$.

If conditions $r = d_H^+(v_1) > t = d_M^+(w_1)$ hold for each $w \in S^+(G) \wedge w \neq v$ and let the $u_1 = w$, then at most $C_{max}^2(G) = a_{2,1}a_{2,2}a_{2,3} \cdots a_{2,n}$ with $a_{2,1} = 1, a_{2,2} = 0, a_{2,3} = a_{2,4} = \dots = a_{2,t+2} = 1$, and $a_{2,t+3} = a_{2,t+4} = \dots = a_{2,s+1} = a_{2,s+2} = \dots = a_{2,n} = 0$ since there exists a node $w_1 \in V(N^{+-}(w))$ satisfying condition $d_M^+(w_1) = \Delta^+(M)$ where $M = N^{++}(w)$.

Otherwise, if let the $u_1 = v$, then $C_{max}^2(G) = a_{2,1}a_{2,2}a_{2,3} \cdots a_{2,n}$ with $a_{2,1} = 1, a_{2,2} = 0, a_{2,3} = a_{2,4} = \dots = a_{2,r+2} = 1$, and $a_{2,r+3} = a_{2,r+4} = \dots = a_{2,s+1} = a_{2,s+2} = \dots = a_{2,n} = 0$ since there exists a node $v_1 \in V(N^{+-}(v))$ satisfying condition $d_H^+(v_1) = \Delta^+(H)$ where $H = N^{++}(v)$.

Since $r > t$, Theorem 3 holds by comparing the above two results of $C_{max}^2(G)$ obtained. \square

Theorem 4. Let $G = \langle V(G), E(G) \rangle$ be a simple digraph with n nodes. Let $S^+(G) = \{u \in V(G) \mid d^+(u) = \Delta^+(G) = s\}$ with $|S^+(G)| > 1$. Assume that $v \in S^+(G)$ have the open outin-neighborhood subdigraph $N^{+-}(v)$ with $|V(N^{+-}(v))| > 0$ (see Definition 9). Suppose that there exists a node $v_1 \in V(N^{+-}(v))$ satisfying condition $d_H^+(v_1) = \Delta^+(H)$ where $H = N^{++}(v)$.

For each $w \in S^+(G) \wedge w \neq v$ with $|V(N^{+-}(w))| > 0$, suppose that there exists a node $w_1 \in V(N^{+-}(w))$ satisfying condition $d_M^+(w_1) = \Delta^+(M)$ where $M = N^{++}(w)$. If conditions $d_H^+(v_1) = d_M^+(w_1)$ and $d_G^+(v_1) > d_G^+(w_1)$ hold, then $u_1 = v$ for $\text{MaxQ}(G)$.

Proof. Since G is a simple digraph with n nodes, it follows that $C_{max}^1(G) = a_{1,1}a_{1,2}a_{1,3} \cdots a_{1,n}$ with $a_{1,1} = 0$ and $C_{max}^2(G) = a_{2,1}a_{2,2}a_{2,3} \cdots a_{2,n}$ with $a_{2,2} = 0$ (see (1)). Since $\Delta^+(G) = s$, one can assert that $a_{1,2} = a_{1,3} = \cdots = a_{1,s+1} = 1$ and $a_{1,s+2} = a_{1,s+3} = \cdots = a_{1,n} = 0$ for $C_{max}^1(G)$.

By the conditions of Theorem 4, clearly $d_H^+(v_1) = d_M^+(w_1) = \Delta^+(H)$. For simplicity, let us assume that $t = d_H^+(v_1) = d_M^+(w_1)$ and $l = d_G^+(v_1) > m = d_G^+(w_1)$.

If conditions $d_G^+(v_1) > d_G^+(w_1)$ hold for each $w \in S(G) \wedge w \neq v$ and let the $u_1 = w$, then at most $C_{max}^2(G) = a_{2,1}a_{2,2}a_{2,3} \cdots a_{2,n}$ with $a_{2,1} = 1, a_{2,2} = 0, a_{2,3} = a_{2,4} = \cdots = a_{2,t+2} = 1, a_{2,t+3} = a_{2,t+4} = \cdots = a_{2,s+1} = 0, a_{2,s+2} = a_{2,s+3} = \cdots = a_{2,s+m-t} = 1$, and $a_{2,s+m-t+1} = a_{2,s+m-t+2} = \cdots = a_{2,n} = 0$ since the node $w_1 \in V(N^{+-}(w))$ satisfying condition $d_M^+(w_1) = \Delta^+(M)$ where $M = N^{++}(w)$ (see (1)).

On the contrary, if let the $u_1 = v$, then $C_{max}^2(G) = a_{2,1}a_{2,2}a_{2,3} \cdots a_{2,n}$ with $a_{2,1} = 1, a_{2,2} = 0, a_{2,3} = a_{2,4} = \cdots = a_{2,t+2} = 1, a_{2,t+3} = a_{2,t+4} = \cdots = a_{2,s+1} = 0, a_{2,s+2} = a_{2,s+3} = \cdots = a_{2,s+l-t} = 1, a_{2,s+l-t+1} = a_{2,s+l-t+2} = \cdots = a_{2,n} = 0$ since the node $v_1 \in V(N^{+-}(v))$ satisfying condition $d_H^+(v_1) = \Delta^+(H)$ where $H = N^{++}(v)$ (see (1)).

Since $l > m$, then the binary number $a_{2,s+2}a_{2,s+3} \cdots a_{2,s+l-t} >$ the binary number $a_{2,s+2}a_{2,s+3} \cdots a_{2,s+m-t}$. Therefore, Theorem 4 holds. \square

Theorem 5. Let $G = \langle V(G), E(G) \rangle$ be a simple digraph with n nodes. Let $S^+(G) = \{u \in V(G) \mid d^+(u) = \Delta^+(G) = s\}$ with $|S^+(G)| > 1$. Suppose that $|V(N^{+-}(u))| = 0$ holds for every $u \in S^+(G)$. Let $v \in S^+(G)$ and $v_1 \in V(N^{++}(v))$ satisfying condition $d_H^+(v_1) = \Delta^+(H)$ where $H = N^{++}(v)$. Let $w \in S^+(G) \wedge w \neq v$ and $w_1 \in V(N^{++}(w))$ satisfying condition $d_M^+(w_1) = \Delta^+(M)$ where $M = N^{++}(w)$. If condition $d_H^+(v_1) > d_M^+(w_1)$ holds, then $u_1 = v$ for $\text{MaxQ}(G)$.

Proof. Since G is a simple digraph with n nodes, it follows that $C_{max}^1(G) = a_{1,1}a_{1,2}a_{1,3} \cdots a_{1,n}$ with $a_{1,1} = 0$ and $C_{max}^2(G) = a_{2,1}a_{2,2}a_{2,3} \cdots a_{2,n}$ with $a_{2,2} = 0$ (see (1)). Since $\Delta^+(G) = s$, it follows that $a_{1,2} = a_{1,3} = \cdots = a_{1,s+1} = 1$ for $C_{max}^1(G)$.

By the conditions of Theorem 5, clearly $d_H^+(v_1) = \Delta^+(H)$ and $d_M^+(w_1) = \Delta^+(M)$. For simplicity, let us assume that $r = d_H^+(v_1), t = d_M^+(w_1)$.

If conditions $r = d_H^+(v_1) > t = d_M^+(w_1)$ hold for each $w \in S^+(G) \wedge w \neq v$ and let the $u_1 = w$, then at most $C_{max}^2(G) = a_{2,1}a_{2,2}a_{2,3} \cdots a_{2,n}$ with $a_{2,1} = 0, a_{2,2} = 0, a_{2,3} = a_{2,4} = \cdots = a_{2,t+2} = 1$, and $a_{2,t+3} = a_{2,t+4} = \cdots = a_{2,s+1} = a_{2,s+2} = \cdots = a_{2,n} = 0$ since $|V(N^{+-}(w))| = 0$ holds and $w_1 \in V(N^{++}(w))$ satisfies condition $d_M^+(w_1) = \Delta^+(M)$ where $M = N^{++}(w)$.

Conversely, if let the $u_1 = v$, then $C_{max}^2(G) = a_{2,1}a_{2,2}a_{2,3} \cdots a_{2,n}$ with $a_{2,1} = 0, a_{2,2} = 0, a_{2,3} = a_{2,4} = \cdots = a_{2,r+2} = 1$, and $a_{2,r+3} = a_{2,r+4} = \cdots = a_{2,s+1} = a_{2,s+2} = \cdots = a_{2,n} = 0$ since $|V(N^{+-}(v))| = 0$ holds and $v_1 \in V(N^{++}(v))$ satisfies condition $d_H^+(v_1) = \Delta^+(H)$ where $H = N^{++}(v)$.

Since $r > t$, Theorem 5 holds by comparing the above two results of $C_{max}^2(G)$ obtained. \square

Theorem 6. Let $G = \langle V(G), E(G) \rangle$ be a simple digraph with n nodes. Let $S^+(G) = \{u \in V(G) \mid d^+(u) = \Delta^+(G) = s\}$ with $|S^+(G)| > 1$. Suppose that $|V(N^{+-}(u))| = 0$ holds for every $u \in S^+(G)$. Let $v \in S^+(G)$ and $v_1 \in V(N^{++}(v))$ satisfying condition $d_H^+(v_1) = \Delta^+(H)$ where $H = N^{++}(v)$. Let $w \in S^+(G) \wedge w \neq v$ and $w_1 \in V(N^{++}(w))$ satisfying condition $d_M^+(w_1) = \Delta^+(M)$ where $M = N^{++}(w)$. If conditions $d_H^+(v_1) = d_M^+(w_1)$ and $d_G^+(v_1) > d_G^+(w_1)$ hold, then $u_1 = v$ for $\text{MaxQ}(G)$.

Proof. Since G is a simple digraph with n nodes, it follows that $C_{max}^1(G) = a_{1,1}a_{1,2}a_{1,3} \cdots a_{1,n}$ with $a_{1,1} = 0$ and $C_{max}^2(G) = a_{2,1}a_{2,2}a_{2,3} \cdots a_{2,n}$ with $a_{2,2} = 0$ (see (1)). Since $\Delta^+(G) = s$, one can assert that $a_{1,2} = a_{1,3} = \cdots = a_{1,s+1} = 1$ and $a_{1,s+2} = a_{1,s+3} = \cdots = a_{1,n} = 0$ for $C_{max}^1(G)$.

By the conditions of Theorem 6, clearly $d_H^+(v_1) = d_M^+(w_1) = \Delta^+(H)$. For simplicity, let us assume that $t = d_H^+(v_1) = d_M^+(w_1)$ and $l = d_G^+(v_1) > m = d_G^+(w_1)$.

If conditions $d_H^+(v_1) > d_G^+(w_1)$ hold for each $w \in S(G) \wedge w \neq v$ and let the $u_1 = w$, then at most $C_{max}^2(G) = a_{2,1}a_{2,2}a_{2,3} \cdots a_{2,n}$ with $a_{2,1} = 0, a_{2,2} = 0, a_{2,3} = a_{2,4} = \cdots = a_{2,t+2} = 1, a_{2,t+3} = a_{2,t+4} = \cdots = a_{2,s+1} = 0, a_{2,s+2} = a_{2,s+3} = \cdots = a_{2,s+m-t} = 1$, and $a_{2,s+m-t+1} = a_{2,s+m-t+2} = \cdots = a_{2,n} = 0$ since $|V(N^{+-}(w))| = 0$ holds and $w_1 \in V(N^{++}(w))$ satisfies condition $d_M^+(w_1) = \Delta^+(M)$ where $M = N^{++}(w)$ (see (1)).

Conversely, if let the $u_1 = v$, then $C_{max}^2(G) = a_{2,1}a_{2,2}a_{2,3} \cdots a_{2,n}$ with $a_{2,1} = 0, a_{2,2} = 0, a_{2,3} = a_{2,4} = \cdots = a_{2,t+2} = 1, a_{2,t+3} = a_{2,t+4} = \cdots = a_{2,s+1} = 0, a_{2,s+2} = a_{2,s+3} = \cdots = a_{2,s+l-t} = 1, a_{2,s+l-t+1} = a_{2,s+l-t+2} = \cdots = a_{2,n} = 0$ since $|V(N^{+-}(v))| = 0$ holds and $v_1 \in V(N^{++}(v))$ satisfies condition $d_H^+(v_1) = \Delta^+(H)$ where $H = N^{++}(v)$ (see (1)).

Since $l > m$, then the binary number $a_{2,s+2}a_{2,s+3} \cdots a_{2,s+l-t} >$ the binary number $a_{2,s+2}a_{2,s+3} \cdots a_{2,s+m-t}$. Therefore, Theorem 6 holds. \square

Conjecture 1. Let $G = \langle V(G), E(G) \rangle$ be a simple digraph with n nodes. Let $S^+(G) = \{u \in V(G) \mid d^+(u) = \Delta^+(G) = s\}$ with $|S^+(G)| > 1$. If there exists a vertex $v \in S^+(G)$ satisfying conditions $C_{max}(G - v) \leq C_{max}(G - w)$ for each $w \in S^+(G) \wedge w \neq v$, then $u_1 = v$ for $MaxQ(G)$.

3.1.2. Calculate the Intermediate Vertices Added into $MaxQ(G)$

When our algorithm has computed the first node u_1 of $MaxQ(G)$, how would it determine the subsequent nodes for calculating $C_{max}(G)$? Note that a directed edge of G corresponds to 1 bit of the adjacency matrix $A(G)$. To maximize $C(G)$ by maximizing $C^2(G)$, one must let u_2 belong to the mix-neighborhood subdigraph $N^{++}(u_1)$ so that makes $a_{1,2} = 1$ (see (1)). Otherwise, if $u_2 \notin N^{++}(u_1)$, then $a_{1,2} = 0$ (see (1)) and $C(G) \neq C_{max}(G)$. The following Lemma 6 summarizes the above results.

Lemma 6. Let $G = \langle V(G), E(G) \rangle$ be a simple digraph with n nodes. Let $S^+(G) = \{u \in V(G) \mid d^+(u) = \Delta^+(G) = s\}$. Assume that u_1 is the first node of $MaxQ(G)$ obtained for computing $C_{max}(G)$. By (12) of Definition 9, if condition $|V(N^{+-}(u_1))| \geq 1$ holds, then $u_2 \in V(N^{+-}(u_1))$ for computing the second node of $MaxQ(G)$ so that obtaining $C_{max}(G)$. Further, if $|V(N^{+-}(u_1))| = 1$ with $v \in V(N^{+-}(u_1))$, then $u_2 = v$.

Proof. We now prove the first statement of Lemma 6 by contradiction. Assume by contradiction that $u_2 \notin V(N^{+-}(u_1))$. By Definition 9 and the condition that u_1 is the first node of $MaxQ(G)$, it follows that $u_2 \in V(N^-(u_1)) \cup V(N^+(u_1))$ for obtaining $C_{max}(G)$. By the condition $\Delta^+(G) = s$, clearly $C_{max}^1(G) = a_{1,1}a_{1,2}a_{1,3} \cdots a_{1,n}$ with $a_{1,1} = 0, a_{1,2} = 1, \cdots, a_{1,s+1} = 1, a_{1,s+2} = 0, \cdots, a_{1,n} = 0$ since G is a simple digraph, and $C_{max}^2(G) = a_{2,1}a_{2,2}a_{2,3} \cdots a_{2,n}$. It can be seen that if $u_2 \in V(N^{+-}(u_1))$, then there are $a_{1,2} = 1$ consistent with the requirement of $C_{max}^1(G)$ and $a_{2,1} = 1$ for $C_{max}^2(G)$ (see (1)).

If $u_2 \in V(N^-(u_1))$, by (10) of Definition 9 then $a_{1,2} = 0$ for $C_{max}^1(G)$, leading to a contradiction with the constraint $a_{1,2} = 1$ satisfied by $C_{max}^1(G)$. Otherwise, if $u_2 \in V(N^+(u_1))$, by (11) of Definition 9 then $a_{1,2} = 1$ for $C_{max}^1(G)$ and $a_{2,1} = 0$ for $C_{max}^2(G)$, resulting in a contradiction with the result $a_{2,1} = 1$ obtained for $C_{max}^2(G)$. Therefore, the assumption that $u_2 \notin V(N^{+-}(u_1))$ does not hold. The second claim of Lemma 7 immediately follows from the previous result. \square

Lemma 7. Let $G = \langle V(G), E(G) \rangle$ be a simple digraph with n nodes. Let $S^+(G) = \{u \in V(G) \mid d^+(u) = \Delta^+(G) = s\}$. Assume that u_1 is the first node of $MaxQ(G)$ obtained for computing $C_{max}(G)$. By Definition 9, if conditions $|V(N^{+-}(u_1))| = |V(N^-(u_1))| = 0 \wedge |V(N^+(u_1))| \geq 1$ hold, then $u_2 \in V(N^+(u_1))$ for computing the second node of $MaxQ(G)$ so that getting $C_{max}(G)$.

Proof. Since u_1 is the first node of $MaxQ(G)$ and G is a simple digraph, by $\Delta^+(G) = s$ then $C_{max}^1(G) = a_{1,1}a_{1,2}a_{1,3} \cdots a_{1,n}$ with $a_{1,1} = 0, a_{1,2} = 1, \dots, a_{1,s+1} = 1, a_{1,s+2} = 0, \dots, a_{1,n} = 0$ (see (1)). By the conditions $|V(N^{+-}(u_1))| = |V(N^-(u_1)) - u_1| = 0$, it follows that $a_{2,1} = \dots = a_{n,1} = 0$ for $C_{max}(G)$. Therefore, $u_2 \in V(N^+(u_1))$ for $MaxQ(G)$ due to $|V(N^+(u_1))| \geq 1$ holds. \square

Lemma 8. Let $G = \langle V(G), E(G) \rangle$ be a simple digraph with n nodes. Let $S^+(G) = \{u \in V(G) \mid d^+(u) = \Delta^+(G) = s\}$. Assume that u_1 is the first node of $MaxQ(G)$ obtained for computing $C_{max}(G)$. By Definition 9, if conditions $|V(N^{+-}(u_1))| = 0 \wedge |V(N^-(u_1))| \geq 1 \wedge |V(N^+(u_1))| \geq 1$ hold, then $u_2 \in V(N^+(u_1))$ for computing the second node of $MaxQ(G)$ so that getting $C_{max}(G)$.

Proof. Since u_1 is the first node of $MaxQ(G)$, G is a simple digraph, and $\Delta^+(G) = s$, then $C_{max}^1(G) = a_{1,1}a_{1,2}a_{1,3} \cdots a_{1,n}$ with $a_{1,1} = 0, a_{1,2} = 1, \dots, a_{1,s+1} = 1, a_{1,s+2} = 0, \dots, a_{1,n} = 0$ (see (1)). If $|V(N^{+-}(u_1))| = 0$, clearly $u_2 \notin V(N^{+-}(u_1))$ for $MaxQ(G)$. If $u_2 \in V(N^-(u_1))$, it follows that $a_{1,2} = 0$ for $C_{max}^1(G)$, which is a contradiction with previous result $a_{1,2} = 1$ derived for $C_{max}^1(G)$. Therefore, if $|V(N^+(u_1))| \geq 1$, then $u_2 \in V(N^+(u_1))$. \square

Theorem 7. Let $G = \langle V(G), E(G) \rangle$ be a simple digraph with n nodes. Let $S^+(G) = \{u \in V(G) \mid d^+(u) = \Delta^+(G) = s\}$. By Definition 9, let $H = N^{++}(v)$ be the mix-neighborhood subdigraph of v . For computing $C_{max}(G)$, suppose that $MaxQ(G)$ already contains the first node $u_1 = v$ with $V(N^{+-}(v)) = \{v_1, v_2, \dots, v_t\}$, $t > 1$. If one of the following conditions holds, then $u_2 = v_1$ for $MaxQ(G)$.

1. $d_H^+(v_1) > d_H^+(v_i)$ for $i = 2, \dots, t$.
2. $d_G^+(v_1) > d_G^+(v_i)$ hold for $d_H^+(v_1) = d_H^+(v_i)$ with $i \in \{2, \dots, t\}$.

Proof. (1) By Lemma 6, it follows that if condition $|V(N^{+-}(v))| \geq 1$ holds, then $u_2 \in V(N^{+-}(v))$ for $MaxQ(G)$. By the conditions of Theorem 7, there is $C_{max}^1(G) = a_{1,1}a_{1,2}a_{1,3} \cdots a_{1,n}$ with $a_{1,1} = 0, a_{1,2} = 1, \dots, a_{1,s+1} = 1, a_{1,s+2} = 0, \dots, a_{1,n} = 0$ (see (1)).

Note that $C_{max}^2(G) = a_{2,1}a_{2,2}a_{2,3} \cdots a_{2,n}$ with $a_{2,1} = 1, a_{2,2} = 0$ since $v_1 \in V(N^{+-}(v))$ and G is a simple digraph. For simplicity, let us assume that $r = d_H^+(v_1)$. If $u_2 = v_1$, it can be seen that $a_{2,3} = 1, a_{2,4} = 1, \dots, a_{2,r+2} = 1$ by properly arranging the nodes of $V(H)$ (see (1)). Otherwise, if $u_2 = v_i$ with $i \in \{2, \dots, t\}$, no matter how the vertices in H are ordered such that there is, at least, one 0 among the r entries $a_{2,3}, a_{2,4}, \dots, a_{2,r+2}$ since $d_H^+(v_i) < d_H^+(v_1)$ for $i = 2, \dots, t$ (see (1)). Therefore, the conclusion (1) of Theorem 7 holds.

(2) Observe that $C_{max}^2(G) = a_{2,1}a_{2,2}a_{2,3} \cdots a_{2,n}$ with $a_{2,1} = 1, a_{2,2} = 0$ since $v_1 \in V(N^{+-}(v))$ and G is a simple digraph. For simplicity, let us assume that $j = |V(H)|, r = d_H^+(v_1), l = d_G^+(v_1)$, and $m = \max\{d_G^+(v_i)\}$ with $i \in \{2, \dots, t\}$. Without loss of generality assume $w \in \{v_2, \dots, v_t\}$ with $d_G^+(w) = m$ and $k = l - m$. If $u_2 = v_1$, one can let $a_{2,3} = 1, a_{2,4} = 1, \dots, a_{2,r+2} = 1, a_{2,r+3} = 0, \dots, a_{2,j+2} = 1$ by properly ordering the nodes of $V(H)$ since $r = d_H^+(v_1)$ (see (1)). Since $d_G^+(v_1) > d_G^+(v_i)$ hold for $d_H^+(v_1) = d_H^+(v_i)$ with $i \in \{2, \dots, t\}$, it follows that $k > 0$. Further, one can let $a_{2,j+3} = 1, a_{2,j+4} = 1, \dots, a_{2,j+k+2} = 1, a_{2,j+k+3} = 0, \dots, a_{2,n} = 0$ by properly sorting the nodes of $V(G) - V(H) - v$ since $k > 0$ (see (1)). Otherwise, if $u_2 = w$, no matter how the vertices of $V(G) - V(H) - v$ are ordered such that there is, at least, one 0 among the k entries $a_{2,j+3}, a_{2,j+4}, \dots, a_{2,j+k+2}$ since $m = d_G^+(w) < l = d_H^+(v_1)$ (see (1)). Therefore, the conclusion (2) of Theorem 7 holds. \square

Theorem 8. Let $G = \langle V(G), E(G) \rangle$ be a simple digraph with n nodes. Let $S^+(G) = \{u \in V(G) \mid d^+(u) = \Delta^+(G) = s\}$. For computing $C_{max}(G)$, suppose that $MaxQ(G)$ already contains the first node $u_1 = v$ satisfying conditions $|V(N^{+-}(v))| = 0 \wedge |V(N^-(v))| \geq 1 \wedge |V(N^+(v))| \geq 1$ with $V(N^+(v)) = \{v_1, v_2, \dots, v_t\}$, $t = |V(N^+(v))| > 1$ (see Definition 9). If one of the following conditions holds, then $u_2 = v_1$ for $MaxQ(G)$.

1. $d_{N^+(v)}^+(v_1) > d_{N^+(v)}^+(v_i)$ for $i = 2, \dots, t$.

2. $d_G^+(v_1) > d_G^+(v_i)$ hold for $d_{N^+(v)}^+(v_1) = d_{N^+(v)}^+(v_i)$ with $i \in \{2, \dots, t\}$.

Proof. By Lemma 8, it follows that if conditions $|V(N^{+-}(v))| = 0 \wedge |V(N^-(v))| \geq 1 \wedge |V(N^+(v))| \geq 1$ hold, then $u_2 \in V(N^+(v))$ for $MaxQ(G)$. By the conditions of Theorem 8, there is $C_{max}^1(G) = a_{1,1}a_{1,2}a_{1,3} \dots a_{1,n}$ with $a_{1,1} = 0, a_{1,2} = 1, \dots, a_{1,s+1} = 1, a_{1,s+2} = 0, \dots, a_{1,n} = 0$ (see (1)). Note that $C_{max}^2(G) = a_{2,1}a_{2,2}a_{2,3} \dots a_{2,n}$ with $a_{2,1} = 0, a_{2,2} = 0$ since $|V(N^{+-}(u_1))| = 0$ and G is a simple digraph.

(1) For simplicity, let us assume that $r = d_{N^+(v)}^+(v_1)$. If $u_2 = v_1$, it can be seen that $a_{2,3} = 1, a_{2,4} = 1, \dots, a_{2,r+2} = 1$ by properly arranging the nodes v_1, v_2, \dots, v_t of $V(N^+(v))$ (see (1)). Conversely, if $u_2 = v_i$ with $i \in \{2, \dots, t\}$, no matter how the vertices in $V(N^+(v))$ are ordered such that there is, at least, one 0 among the r entries $a_{2,3}, a_{2,4}, \dots, a_{2,r+2}$ since $d_{N^+(v)}^+(v_i) < d_{N^+(v)}^+(v_1)$ for $i = 2, \dots, t$ (see (1)). Therefore, the conclusion (1) of Theorem 7 holds.

(2) For convenience, let us assume that $j = |V(N^+(v))|, r = d_{N^+(v)}^+(v_1), l = d_G^+(v_1)$, and $m = \max\{d_G^+(v_i)\}$ with $i \in \{2, \dots, t\}$. Without loss of generality assume $w \in \{v_2, \dots, v_t\}$ with $d_G^+(w) = m$ and $k = l - m$.

If $u_2 = v_1$, one can let $a_{2,3} = 1, a_{2,4} = 1, \dots, a_{2,r+2} = 1, a_{2,r+3} = 0, \dots, a_{2,j+2} = 1$ by properly ordering the nodes of $V(N^+(v))$ since $r = d_{N^+(v)}^+(v_1)$ (see (1)). Since $d_G^+(v_1) > d_G^+(v_i)$ hold for $d_{N^+(v)}^+(v_1) = d_{N^+(v)}^+(v_i)$ with $i \in \{2, \dots, t\}$, it follows that $k > 0$. Further, one can let $a_{2,j+3} = 1, a_{2,j+4} = 1, \dots, a_{2,j+k+2} = 1, a_{2,j+k+3} = 0, \dots, a_{2,n} = 0$ by properly sorting the nodes of $V(G) - V(N^+(v)) - v$ since $k > 0$ (see (1)).

Conversely, if $u_2 = w$, no matter how the vertices of $V(G) - V(N^+(v)) - v$ are ordered such that there is, at least, one 0 among the k entries $a_{2,j+3}, a_{2,j+4}, \dots, a_{2,j+k+2}$ since $m = d_G^+(w) < l = d_{N^+(v)}^+(v_1)$ (see (1)). Therefore, the conclusion (2) of Theorem 8 holds. \square

When our algorithm has computed the first i vertices u_1, u_2, \dots, u_i of $MaxQ(G)$, how does it determine the subsequent nodes $u_{i+1}, u_{i+2}, \dots, u_n$ for calculating $C_{max}(G)$? Similar to the above discussion for obtaining u_2 , it can be shown that the selections of the successor vertices $u_{i+1}, u_{i+2}, \dots, u_n$ of $MaxQ(G)$ are from $N(S)$ with $S = \{u_1, u_2, \dots, u_i\}$.

Our algorithm assigns each node in G an attribute called $m_NearestNode$. Once the i th node u_i has been added into $MaxQ(G)$, it writes the index information i of u_i into the attribute domain $m_NearestNode$ of each node $v_j \in N(u_i) = \{v_1, v_2, \dots, v_t\}$ with $j = 1, 2, \dots, t$. If $v_j - > m_NearestNode = 0$, then let $v_j - > m_NearestNode = i$ for each $v_j \in N(u_i)$ with $j = 1, 2, \dots, t$.

Lemma 9. Let $G = \langle V(G), E(G) \rangle$ be a simple digraph with n nodes. Let $S^+(G) = \{u \in V(G) \mid d^+(u) = \Delta^+(G) = s\}$. When computing $C_{max}(G)$, if $MaxQ(G)$ already contains the first node $u_1 = v$ whose open mix-neighborhood subdigraph is $N^{++}(v)$ (see Definition 9), then $u_2, u_3, \dots, u_s, u_{s+1} \in N^{++}(v)$.

Proof. Since the condition $\Delta^+(G) = s$ holds, it can be asserted that $|N^{++}(v)| = s$ by Definition 9. Because G is a simple digraph with n nodes, it follows that $C_{max}^1(G) = a_{1,1}a_{1,2}a_{1,3} \dots a_{1,n}$ with $a_{1,1} = 0, a_{1,2} = a_{1,3} = \dots = a_{1,s+1} = 1, a_{1,s+2} = a_{1,s+3} = \dots = a_{1,n} = 0$ for obtaining $C_{max}(G)$ (see (1)). To ensure $a_{1,2} = a_{1,3} = \dots = a_{1,s+1} = 1$ to maximize $C^1(G)$ (see (1)), the assertion $u_2, u_3, \dots, u_s, u_{s+1} \in N^{++}(v)$ is true. \square

Theorem 9 (Diffusion Theorem of Digraphs). Let $G = \langle V(G), E(G) \rangle$ be a simple connected digraph with n nodes. For computing $C_{max}(G)$, suppose that $MaxQ(G)$ already contains the first m vertices u_1, u_2, \dots, u_m . Let $Q = \{u_1, u_2, \dots, u_m\}$ whose open neighborhood subdigraph is $N(Q)$ (see Definition 11). If $|V(N(Q))| \geq 1$, then the following two conclusions hold.

1. By Definition 11, the selection of the $(m + 1)$ th vertex of $MaxQ(G)$ for computing $C_{max}(G)$ is from the open neighborhood subdigraph $N(Q)$ of the nodes set Q .
2. the vertex-induced subdigraph of the first m vertices is connected.

Proof. (1) We prove by contradiction. If $u_{m+1} \notin V(N(Q))$, without loss of generality let us assume that $\pi_1 = \{u_1, u_2, \dots, u_m, v_1, u_{m+2}, \dots, v_2, \dots, u_n\}$ is a permutation of $V(G)$, satisfying conditions $v_1 = u_{m+1} \notin V(N(Q)), v_2 = u_i \in V(N(Q))$ with $m + 2 \leq i \leq n$.

Further, we may assume that if condition $u_{m+1} \notin V(N(Q))$ holds, the $C(G)$ corresponding to π_1 is the greatest. Assume that the node $v_2 = u_i \in V(N(Q))$ is the node whose index i in π_1 is the smallest index in π_1 than the indexes of other nodes belonging to $V(N(Q))$ in π_1 . This means that no node belonging to $V(N(Q))$ is between u_{m+2} and u_{i-1} of π_1 such that for every node $v \in \{u_{m+2}, u_{m+3}, \dots, u_{i-1}\}, v \notin V(N(Q))$ follows (see Definition 11).

Let $A_1(G)$ be the matrix corresponding to the permutation π_1 . Let $W_1, W_2, W_3,$ and W_4 be the block submatrices of $A_1(G)$ containing the first m rows and the $(m + 1)$ th column, the $(m + 2)$ th to $(i - 1)$ th columns, the i th column, and the $(i + 1)$ th to n th columns, respectively.

Since $v_1 \notin V(N(Q))$, then $W_1 = 0$ holds. From the above result, for every node $v \in \{u_{m+2}, u_{m+3}, \dots, u_{i-1}\}, v \notin V(N(Q))$ holds such that $W_2 = 0$. For $v_2 \in V(N(Q)) = V(N^-(Q)) \cup V(N^{++}(Q))$, if $v_2 \in V(N^-(Q))$, then $W_3 = 0$. Otherwise, if $v_2 \in V(N^{++}(Q))$, then $W_3 > 0$. Therefore, $W_3 \geq 0$.

Similarly, let $X_1, X_2, X_3,$ and X_4 be the block submatrices of $A_1(G)$ formed by the first m columns and the $(m + 1)$ th row, the $(m + 2)$ th to $(i - 1)$ th rows, the i th row, and the $(i + 1)$ th to n th rows, respectively.

Since $v_1 \notin V(N(Q))$, then $X_1 = 0$ holds. For every node $v \in \{u_{m+2}, u_{m+3}, \dots, u_{i-1}\}$, since $v \notin V(N(Q))$ holds, then $X_2 = 0$. For $v_2 \in V(N(Q)) = V(N^-(Q)) \cup V(N^{++}(Q))$, if $v_2 \in V(N^-(Q))$, then $X_3 > 0$ by Definition 11. Otherwise, if $v_2 \in V(N^{++}(Q))$, then $X_3 > 0$. Therefore, $X_3 > 0$.

By merely swapping v_1 and v_2 of π_1 , one can obtain another permutation $\pi_2 = \{u_1, u_2, \dots, u_m, v_2, u_{m+2}, \dots, v_1, \dots, u_n\}$ with $v_1 \notin V(N(Q)), v_2 \in V(N(Q))$.

Similar to $A_1(G)$, let $A_2(G)$ be the matrix corresponding to the permutation π_2 . Let $Y_1, Y_2, Y_3,$ and Y_4 be the block submatrices of $A_2(G)$ containing the first m rows and the $(m + 1)$ th column, the $(m + 2)$ th to $(i - 1)$ th columns, the i th column, and the $(i + 1)$ th to n th columns, respectively.

For $v_2 \in V(N(Q)) = V(N^-(Q)) \cup V(N^{++}(Q))$, if $v_2 \in V(N^-(Q))$, then $Y_1 = 0$. Otherwise, if $v_2 \in V(N^{++}(Q))$, then $Y_1 > 0$. Therefore, $Y_1 \geq 0$. For every node $v \in \{u_{m+2}, u_{m+3}, \dots, u_{i-1}\}$, since $v \notin V(N(Q))$ holds, then $Y_2 = 0$. Since $v_1 \notin V(N(Q))$, then $Y_3 = 0$ holds.

Simila, Let $Z_1, Z_2, Z_3,$ and Z_4 be the block submatrices of $A_2(G)$ formed by the first m columns and the $(m + 1)$ th rows, the $(m + 2)$ th to $(i - 1)$ th rows, the i th row, and the $(i + 1)$ th to n th rows, respectively.

For $v_2 \in V(N(Q)) = V(N^-(Q)) \cup V(N^{++}(Q))$, if $v_2 \in V(N^-(Q))$, then $Z_1 > 0$ by Definition 11. Otherwise, if $v_2 \in V(N^{++}(Q))$, then $Z_1 > 0$. Therefore, $Z_1 > 0$. For every node $v \in \{u_{m+2}, u_{m+3}, \dots, u_{i-1}\}$, since $v \notin V(N(Q))$ holds, then $Z_2 = 0$. Since $v_1 \notin V(N(Q))$, then $Z_3 = 0$ holds.

By Definition 10, observe that $W_4 = Y_4$ since W_4 and Y_4 are both the $m \times (n - i)$ block submatrices defined by the same nodes sequence $u_{i+1}, u_{i+2}, \dots, u_n$, and $X_4 = Z_4$ since X_4 and Z_4 are both the $(n - i) \times m$ block submatrices corresponding to the same nodes sequence $u_{i+1}, u_{i+2}, \dots, u_n$.

Furthermore, by Definition 10 note that $W_2 = Y_2$ since W_2 and Y_2 are both the $m \times (i - m - 2)$ block submatrices generated by the same nodes sequence $u_{m+2}, u_{m+3}, \dots, u_{i-1}$, and $X_2 = Z_2$ since X_2 and Z_2 are both the $(i - m - 2) \times m$ block submatrices corresponding to the same nodes sequence $u_{m+2}, u_{m+3}, \dots, u_{i-1}$.

By Definition 10, it follows from the results discussed above that $W_1 = W_2 = 0, W_3 \geq 0, Y_1 \geq 0, Y_2 = Y_3 = 0, X_1 = X_2 = 0, X_3 > 0, Z_1 > 0, Z_2 = Z_3 = 0$.

Therefore, the new $C(G)$ derived from $A_2(G)$ is greater than the $C(G)$ stemmed from $A_1(G)$ such that brings a contradiction with the previous assumption that $u_{m+1} \notin N(Q)$. This contradiction shows the statement (1) holds.

(2) The result immediately follows from the conclusion (1). \square

Theorem 10 (Mix Diffusion Theorem of Digraphs). *Let $G = \langle V(G), E(G) \rangle$ be a simple connected digraph with n nodes. For computing $C_{max}(G)$, suppose that $MaxQ(G)$ already contains the first m*

vertices u_1, u_2, \dots, u_m . Let $Q = \{u_1, u_2, \dots, u_m\}$ whose open mix-neighborhood subdigraph is $N^{++}(Q)$. If $|V(N^{++}(Q))| \geq 1$, then the following two conclusions hold.

1. By Definition 11, the selection of the $(m + 1)$ th vertex of $\text{Max}Q(G)$ for computing $C_{\max}(G)$ is from the open mix-neighborhood subdigraph $N^{++}(Q)$ of the nodes set Q .
2. the vertex-induced subdigraph of the first m vertices is connected.

Proof. (1) We prove by contradiction. If $u_{m+1} \notin V(N^{++}(Q))$, without loss of generality let us assume that $\pi_1 = \{u_1, u_2, \dots, u_m, v_1, u_{m+2}, \dots, v_2, \dots, u_n\}$ is a permutation of $V(G)$, satisfying conditions $v_1 = u_{m+1} \notin V(N^{++}(Q)), v_2 = u_i \in V(N^{++}(Q))$ with $m + 2 \leq i \leq n$.

Further, we may assume that if condition $u_{m+1} \notin V(N^{++}(Q))$ holds, the $C(G)$ corresponding to π_1 is the greatest. Assume that the node $v_2 = u_i \in V(N^{++}(Q))$ is the node whose index i in π_1 is the smallest index in π_1 than the indexes of other nodes belonging to $V(N^{++}(Q))$ in π_1 . This means that no node belonging to $V(N^{++}(Q))$ is between u_{m+2} and u_{i-1} of π_1 such that for every node $v \in \{u_{m+2}, u_{m+3}, \dots, u_{i-1}\}, v \notin V(N^{++}(Q))$ follows (see Definition 11).

Let $A_1(G)$ be the matrix corresponding to the permutation π_1 . Let W_1, W_2, W_3 , and W_4 be the block submatrices of $A_1(G)$ containing the first m rows and the $(m + 1)$ th column, the $(m + 2)$ th to $(i - 1)$ th columns, the i th column, and the $(i + 1)$ th to n th columns, respectively.

Since $v_1 \notin V(N^{++}(Q))$, then $v_1 \in V(N^-(Q)) \vee v_1 \notin V(N(Q))$. If $v_1 \in V(N^-(Q))$, then $W_1 = 0$ by Definition 11. Otherwise, if $v_1 \notin V(N(Q))$, then $W_1 = 0$. Therefore, $W_1 = 0$ for $v_1 \notin V(N^{++}(Q))$. From the above result, for every node $v \in \{u_{m+2}, u_{m+3}, \dots, u_{i-1}\}, v \notin V(N^{++}(Q))$ holds such that $v \in V(N^-(Q)) \vee v \notin V(N(Q))$. If $v \in V(N^-(Q))$, then $W_2 = 0$ by Definition 11. Otherwise, if $v \notin V(N(Q))$, then $W_2 = 0$. Therefore, $W_2 = 0$. Since $v_2 \in V(N^{++}(Q))$, then $W_3 > 0$.

By merely swapping v_1 and v_2 of π_1 , one can obtain another permutation $\pi_2 = \{u_1, u_2, \dots, u_m, v_2, u_{m+2}, \dots, v_1, \dots, u_n\}$ with $v_1 \notin V(N^{++}(Q)), v_2 \in V(N^{++}(Q))$.

Similar to $A_1(G)$, let $A_2(G)$ be the matrix corresponding to the permutation π_2 . Let Y_1, Y_2, Y_3 , and Y_4 be the block submatrices of $A_2(G)$ containing the first m rows and the $(m + 1)$ th column, the $(m + 2)$ th to $(i - 1)$ th columns, the i th column, and the $(i + 1)$ th to n th columns, respectively.

Clearly $Y_1 > 0$ holds for $v_2 \in V(N^{++}(Q))$. For every node $v \in \{u_{m+2}, u_{m+3}, \dots, u_{i-1}\}$, since $v \notin V(N^{++}(Q))$ holds, then $v \in V(N^-(Q)) \vee v \notin V(N(Q))$. If $v \in V(N^-(Q))$, then $Y_2 = 0$ by Definition 11. Otherwise, if $v \notin V(N(Q))$, then $Y_2 = 0$. Therefore, $Y_2 = 0$. Since $v_1 \notin V(N^{++}(Q))$, then $v_1 \in V(N^-(Q)) \vee v_1 \notin V(N(Q))$. If $v_1 \in V(N^-(Q))$, then $Y_3 = 0$ by Definition 11. Otherwise, if $v_1 \notin V(N(Q))$, then $Y_3 = 0$. Therefore, $Y_3 = 0$.

By Definition 10, observe that $W_4 = Y_4$ since W_4 and Y_4 are both the $m \times (n - i)$ block submatrices corresponding to the same nodes sequence $u_{i+1}, u_{i+2}, \dots, u_n$.

Furthermore, by Definition 10 note that $W_2 = Y_2$ since W_2 and Y_2 are both the $m \times (i - m - 2)$ block submatrices corresponding to the same nodes sequence $u_{m+2}, u_{m+3}, \dots, u_{i-1}$, and $X_2 = Z_2$ since X_2 and Z_2 are both the $(i - m - 2) \times m$ block submatrices corresponding to the same nodes sequence $u_{m+2}, u_{m+3}, \dots, u_{i-1}$.

It follows from the results discussed above that $W_1 = W_2 = 0, W_3 > 0, Y_1 > 0, Y_2 = Y_3 = 0$.

Therefore, the new $C(G)$ derived from $A_2(G)$ is greater than the $C(G)$ stemmed from $A_1(G)$ such that brings a contradiction with the previous assumption that $u_{m+1} \notin N^{++}(Q)$. This contradiction shows the statement (1) holds.

(2) The result immediately follows from the conclusion (1). \square

Corollary 1. Let $G = \langle V(G), E(G) \rangle$ be a simple connected digraph with n nodes. Let $S^+(G) = \{u \in V(G) | d^+(u) = \Delta^+(G) = s\}$. For computing $\text{Max}Q(G)$, suppose that $\text{Max}Q(G)$ already contains the first m vertices u_1, u_2, \dots, u_m . By Definition 11, if $Q = \{u_1, u_2, \dots, u_m\}$ satisfies conditions $|V(N^{++}(Q))| = 0 \wedge |V(N^-(Q))| \geq 1$, then $u_{m+1} \in V(N^-(Q))$.

Proof. By Neighborhood Diffusion Theorem 9, we have that $u_{m+1} \in V(N(Q))$. By Mix Diffusion Theorem 10, it follows that $u_{m+1} \in V(N^{++}(Q))$. Since conditions $|V(N^{++}(Q))| = 0 \wedge |V(N^-(Q))| \geq 1$ hold, the result of Corollary 1 is true. \square

Corollary 2. Let $G = \langle V(G), E(G) \rangle$ be a simple connected digraph with n nodes. For computing $C_{max}(G)$, suppose that $MaxQ(G)$ already contains the first m vertices u_1, u_2, \dots, u_m . If $Q = \{u_1, u_2, \dots, u_m\}$ has open neighborhood subdigraph $N(Q)$ satisfying conditions $|V(N(Q))| \geq 1$. If $|V(N^{++}(Q))| = 0 \wedge |V(N^+(Q))| > 0$, then $u_{m+1} \in N^+(Q)$.

Proof. It follows from Mix Diffusion Theorem 10. \square

3.2. Compute $C_{max}(G)$ for a Disconnected Digraph

Let $G = \langle V(G), E(G) \rangle$ be a simple disconnected digraph with n nodes and p connected components. Suppose that the p connected components are G_1, G_2, \dots, G_p . In this subsection, we study how to compute the maximum element $C_{max}(G)$ of G .

If $\Delta^+(G_1) > \Delta^+(G_2) > \dots > \Delta^+(G_p)$, how does our algorithm work to compute the maximum element $C_{max}(G)$ of G ? Observe that to obtain $C_{max}(G)$ one had to arrange all vertices of each connected component G_i with $i \in \{1, 2, \dots, p\}$ together when constructing the adjacency matrix $A(G)$. The result also follows from the proof of Diffusion Theorem of Digraphs 9.

First, we analyze the features of the adjacency matrix $A(G)$. When constructing the adjacency matrix $A(G)$, we arrange all vertices of each connected component G_i with $i \in \{1, 2, \dots, p\}$ together. As a result, the adjacency matrix $A(G)$ is a block matrix, each block of which corresponds to a connected component. Next, we study the relationship between $C(G)$ and $A(G)$. Furthermore, we show how to solve the $C_{max}(G_1)$ of the adjacency matrix $A(G)$.

Lemma 10. Let $G = \langle V(G), E(G) \rangle$ be a simple disconnected digraph that have two disjoint connected components $G_1 = \langle V(G_1), E(G_1) \rangle$ and $G_2 = \langle V(G_2), E(G_2) \rangle$ with k and l nodes respectively. Suppose that

$$\begin{aligned} C_{max}(G_1) &= C_{max}^1(G_1)C_{max}^2(G_1) \cdots C_{max}^{k-1}(G_1), \\ C_{max}(G_2) &= C_{max}^1(G_2)C_{max}^2(G_2) \cdots C_{max}^{l-1}(G_2). \end{aligned}$$

If $\Delta^+(G_1) > \Delta^+(G_2)$, then $C_{max}(G)$ satisfies the following equality:

$$C_{max}(G) = C_{max}^1(G)C_{max}^2(G) \cdots C_{max}^{k-1}(G)C_{max}^k(G)C_{max}^{k+2}(G) \cdots C_{max}^{k+l-1}(G), \tag{49}$$

where

$$\begin{aligned} C_{max}^1(G) &= C_{max}^1(G_1) \overbrace{00 \cdots 0}^l, & C_{max}^2(G) &= C_{max}^2(G_1) \overbrace{00 \cdots 0}^l & \dots & \dots & \dots & \dots \\ C_{max}^{k-1}(G) &= C_{max}^{k-1}(G_1) \overbrace{00 \cdots 0}^l, & C_{max}^k(G) &= \overbrace{00 \cdots 0}^l, & & & C_{max}^{k+1}(G) &= C_{max}^1(G_2), \\ C_{max}^{k+2}(G) &= C_{max}^2(G_2), & \dots & \dots & & & C_{max}^{k+l-1}(G) &= C_{max}^{l-1}(G_2). \end{aligned}$$

Proof. If $\Delta^+(G_1) > \Delta^+(G_2)$ holds, then $\Delta^+(G) = \Delta^+(G_1)$. By Proposition 3, it follows that to obtain $C_{max}(G)$, one must choose the vertex with the maximum outdegree from G_1 as the first vertex u_1 of $MaxQ(G)$.

By Diffusion Theorem of Digraphs 9, the subsequent $k - 1$ vertices added into $MaxQ(G)$ must be taken from G_1 . Similarly, by Proposition 3, it follows that to obtain $C_{max}(G)$, one must choose the $(k + 1)$ th vertex from G_2 with the maximum outdegree as the $(k + 1)$ th vertex u_{k+1} of $MaxQ(G)$.

By Diffusion Theorem of Digraphs 9, the next $l - 1$ vertices added into $MaxQ(G)$ must be from G_2 . Carefully examining (1), it is not difficult to find that (49) holds. \square

Note that to ensure the maximization of $C_{max}(G)$, one must add l 0 after $C_{max}^1(G_1), C_{max}^2(G_1), \dots, C_{max}^{k-1}(G_1)$ respectively so that let $C_{max}^k(G)$ be equal l 0.

Theorem 11. Let $G = \langle V(G), E(G) \rangle$ be a simple disconnected digraph with n nodes and p connected components. Suppose that the p connected components are G_1, G_2, \dots, G_p with $|V(G_1)| = n_1, |V(G_2)| = n_2, \dots, |V(G_p)| = n_p$. If $C_{max}(G_1) > C_{max}(G_2) > \dots > C_{max}(G_p)$, then $C_{max}(G)$ satisfies the following equality:

$$\begin{aligned}
 C_{max}(G) = & C_{max}^1(G_1) \overbrace{0 \dots 0}^{n-n_1} \dots C_{max}^{n_1-1}(G_1) \overbrace{0 \dots 0}^{n-n_1} \overbrace{0 \dots 0}^{n-n_1} \\
 & C_{max}^1(G_2) \overbrace{0 \dots 0}^{n-n_1-n_2} \dots C_{max}^{n_2-1}(G_2) \overbrace{0 \dots 0}^{n-n_1-n_2} \overbrace{0 \dots 0}^{n-n_1-n_2} \\
 & \vdots \\
 & C_{max}^1(G_{p-1}) \overbrace{0 \dots 0}^{n_p} \dots C_{max}^{n_{p-1}-1}(G_{p-1}) \overbrace{0 \dots 0}^{n_p} \overbrace{0 \dots 0}^{n_p} \\
 & C_{max}^1(G_p) \dots C_{max}^{n_p-1}(G_p).
 \end{aligned} \tag{50}$$

Proof. We prove (50) by induction on the number p of branches. By the preceding definition, we have $C_{max}(G) = C_{max}^1(G)C_{max}^2(G) \dots C_{max}^{n-1}(G)$ for $p = 1$. Thus, (50) in Theorem 11 holds for $p = 1$. By Lemma 10, (50) holds for $p = 2$.

By induction, suppose that (50) holds for $p = k$. In the following, we prove that the equality (50) also holds for $p = k + 1$. We can now treat the front k branch digraphs as the digraph H . Therefore, (50) also holds for the digraph H .

$$\begin{aligned}
 C_{max}(H) = & C_{max}^1(G_1) \overbrace{0 \dots 0}^{n-n_{k+1}-n_1} \dots C_{max}^{n_1-1}(G_1) \overbrace{0 \dots 0}^{n-n_{k+1}-n_1} \overbrace{0 \dots 0}^{n-n_{k+1}-n_1} \\
 & C_{max}^1(G_2) \overbrace{0 \dots 0}^{n-n_{k+1}-n_1-n_2} \dots C_{max}^{n_2-1}(G_2) \overbrace{0 \dots 0}^{n-n_{k+1}-n_1-n_2} \overbrace{0 \dots 0}^{n-n_{k+1}-n_1-n_2} \\
 & \vdots \\
 & C_{max}^1(G_{k-1}) \overbrace{0 \dots 0}^{n_k} \dots C_{max}^{n_{k-1}-1}(G_{k-1}) \overbrace{0 \dots 0}^{n_k} \overbrace{0 \dots 0}^{n_k} \\
 & C_{max}^1(G_k) C_{max}^2(G_k) \dots C_{max}^{n_k-1}(G_k),
 \end{aligned} \tag{51}$$

where

$$\begin{aligned}
 C_{max}^1(H) &= C_{max}^1(G_1) \overbrace{0 \dots 0}^{n-n_{k+1}-n_1}, \\
 &\vdots \\
 C_{max}^{n_1-1}(H) &= C_{max}^{n_1-1}(G_1) \overbrace{0 \dots 0}^{n-n_{k+1}-n_1}, \\
 C_{max}^{n_1}(H) &= \overbrace{0 \dots 0}^{n-n_{k+1}-n_1},
 \end{aligned}$$

$$\begin{aligned}
 C_{max}^{n_1+1}(H) &= C_{max}^1(G_2) \overbrace{0 \cdots 0}^{n-n_{k+1}-n_1-n_2}, \\
 &\vdots \\
 C_{max}^{n_1+n_2-1}(H) &= C_{max}^{n_2-1}(G_2) \overbrace{0 \cdots 0}^{n-n_{k+1}-n_1-n_2}, \\
 C_{max}^{n_1+n_2}(H) &= \overbrace{0 \cdots 0}^{n-n_{k+1}-n_1-n_2}, \\
 &\vdots \\
 C_{max}^{n-n_{k+1}-n_k-n_{k-1}+1}(H) &= C_{max}^1(G_{k-1}) \overbrace{0 \cdots 0}^{n_k}, \\
 &\vdots \\
 C_{max}^{n-n_{k+1}-n_k-1}(H) &= C_{max}^{n_{k-1}-1}(G_{k-1}) \overbrace{0 \cdots 0}^{n_k},
 \end{aligned}$$

By Lemma 10, we have

$$\begin{aligned}
 C_{max}(G) &= \\
 &C_{max}^1(H) \overbrace{0 \cdots 0}^{n_{k+1}} C_{max}^2(H) \overbrace{0 \cdots 0}^{n_{k+1}} \cdots C_{max}^{n-n_{k+1}-1}(H) \overbrace{0 \cdots 0}^{n_{k+1}} \\
 &\overbrace{0 \cdots 0}^{n_{k+1}} C_{max}^1(G_{k+1}) C_{max}^2(G_{k+1}) \cdots C_{max}^{n_{k+1}-1}(G_{k+1}).
 \end{aligned} \tag{52}$$

By (51), substituting $C_{max}^1(H)$ to $C_{max}^{n-n_{k+1}-1}(H)$ into (52), we obtain

$$\begin{aligned}
 C_{max}(G) &= \\
 &C_{max}^1(G_1) \overbrace{0 \cdots 0}^{n-n_{k+1}-n_1} \overbrace{0 \cdots 0}^{n_{k+1}} C_{max}^2(G_1) \overbrace{0 \cdots 0}^{n-n_{k+1}-n_1} \overbrace{0 \cdots 0}^{n_{k+1}} \cdots C_{max}^{n_k-1}(G_k) \\
 &\overbrace{0 \cdots 0}^{n_{k+1}} \overbrace{0 \cdots 0}^{n_{k+1}} C_{max}^1(G_{k+1}) C_{max}^2(G_{k+1}) \cdots C_{max}^{n_{k+1}-1}(G_{k+1}).
 \end{aligned} \tag{53}$$

Thus, we have

$$\begin{aligned}
 C_{max}(G) &= C_{max}^1(G_1) \overbrace{0 \cdots 0}^{n-n_1} \cdots C_{max}^{n_1-1}(G_1) \overbrace{0 \cdots 0}^{n-n_1} \overbrace{0 \cdots 0}^{n-n_1} \\
 &C_{max}^1(G_2) \overbrace{0 \cdots 0}^{n-n_1-n_2} \cdots C_{max}^{n_2-1}(G_2) \overbrace{0 \cdots 0}^{n-n_1-n_2} \overbrace{0 \cdots 0}^{n-n_1-n_2} \\
 &\vdots \\
 &C_{max}^1(G_k) \overbrace{0 \cdots 0}^{n_{k+1}} \cdots C_{max}^{n_k-1}(G_k) \overbrace{0 \cdots 0}^{n_{k+1}} \overbrace{0 \cdots 0}^{n_{k+1}} \\
 &C_{max}^1(G_{k+1}) \cdots C_{max}^{n_{k+1}-1}(G_{k+1}).
 \end{aligned} \tag{54}$$

Thus, the equality (50) holds for $p = k + 1$. □

By Theorem 11, it can be seen that one must first calculate $C_{max}(G_i)$ of each branch for $i = 1, 2, \dots, p$ for obtaining $C_{max}(G)$, respectively. Furthermore, one substitutes $C_{max}(G_i)$ into (50) sequentially to obtain $C_{max}(G)$ of a simple disconnected digraph G .

If the above conditions are not satisfied, how does one calculate $C_{max}(G)$ of a disconnected undirected G ? Based on an analysis of the preceding results, establish the following Theorem 12 that is more general than Lemma 10.

Theorem 12. Let $G = \langle V(G), E(G) \rangle$ be a simple disconnected digraph that have two disjoint connected components $G_1 = \langle V(G_1), E(G_1) \rangle$ and $G_2 = \langle V(G_2), E(G_2) \rangle$ with k and l nodes respectively. Let $S^+(G_1) = \{u \in V(G_1) \mid d^+(u) = \Delta^+(G_1)\}$ and $S^+(G_2) = \{u \in V(G_2) \mid d^+(u) = \Delta^+(G_2)\}$ satisfying condition $\Delta^+(G_1) = \Delta^+(G_2)$. Suppose that

$$C_{max}(G_1) = C_{max}^1(G_1)C_{max}^2(G_1) \cdots C_{max}^{k-1}(G_1), \tag{55}$$

$$C_{max}(G_2) = C_{max}^1(G_2)C_{max}^2(G_2) \cdots C_{max}^{l-1}(G_2), \tag{56}$$

If there exists a node $u \in S^+(G_1)$ satisfying condition $|N^{+-}(u)| > 0 \wedge \Delta^+(N^{++}(u)) > \Delta^+(N^{++}(v))$ for $\forall v \in S_2$, then $C_{max}(G)$ satisfies the following equality:

$$C_{max}(G) = C_{max}^1(G)C_{max}^2(G) \cdots C_{max}^{k-1}(G)C_{max}^k(G)C_{max}^{k+2}(G) \cdots C_{max}^{k+l-1}(G), \tag{57}$$

where

$$\begin{aligned} C_{max}^1(G) &= C_{max}^1(G_1) \overbrace{00 \cdots 0}^l, & C_{max}^2(G) &= C_{max}^2(G_1) \overbrace{00 \cdots 0}^l & \dots & \dots & \dots & \dots & \dots & \dots \\ C_{max}^{k-1}(G) &= C_{max}^{k-1}(G_1) \overbrace{00 \cdots 0}^l, & C_{max}^k(G) &= \overbrace{00 \cdots 0}^l, & & & C_{max}^{k+1}(G) &= C_{max}^1(G_2), & & \\ C_{max}^{k+2}(G) &= C_{max}^2(G_2), & & \dots & & & C_{max}^{k+l-1}(G) &= C_{max}^{l-1}(G_2). & & \end{aligned}$$

Proof. By the condition of Theorem 12, $\Delta^+(G) = \max\{\Delta^+(G_1), \Delta^+(G_2)\} = \Delta^+(G_1) = \Delta^+(G_2)$ since $\Delta^+(G_1) = \Delta^+(G_2)$ holds.

If there exists a node $u \in S^+(G_1)$ satisfying condition $|N^{+-}(u)| > 0 \wedge \Delta^+(N^{++}(u)) > \Delta^+(N^{++}(v))$ for $\forall v \in S_2$, by Theorem 3, one must choose the first vertex u_1 added into $MaxQ(G)$ from G_1 so that obtain $C_{max}(G)$.

By Diffusion Theorem of Digraphs 9, one must select u_2, u_3, \dots, u_k into $MaxQ(G)$ from G_1 to obtain $C_{max}(G)$. In addition, one must choose the subsequent l nodes into $MaxQ(G)$ from G_2 . By (55) and (56), it follows that (58) holds.

$$C_{max}(G) = C_{max}^1(G)C_{max}^2(G) \cdots C_{max}^{k-1}(G)C_{max}^k(G)C_{max}^{k+2}(G) \cdots C_{max}^{k+l-1}(G), \tag{58}$$

where

$$\begin{aligned} C_{max}^1(G) &= C_{max}^1(G_1) \overbrace{00 \cdots 0}^l, & C_{max}^2(G) &= C_{max}^2(G_1) \overbrace{00 \cdots 0}^l, & & & & & & \dots & \dots & \dots & \dots & \dots & \dots \\ C_{max}^{k-1}(G) &= C_{max}^{k-1}(G_1) \overbrace{00 \cdots 0}^l, & C_{max}^k(G) &= \overbrace{00 \cdots 0}^l, & & & C_{max}^{k+1}(G) &= C_{max}^1(G_2), & & & & & & & \\ C_{max}^{k+2}(G) &= C_{max}^2(G_2), & & \dots & & & C_{max}^{k+l-1}(G) &= C_{max}^{l-1}(G_2). & & & & & & & \end{aligned}$$

Note that to ensure the maximization of $C_{max}(G)$, l 0 must be added after $C_{max}^1(G_1), C_{max}^2(G_1), \dots, C_{max}^{k-1}(G_1)$ respectively and let $C_{max}^k(G)$ be equal l 0. \square

4. Our Algorithms for Computing the Canonical Labelings of Digraphs

In the section, based on the results of the previous sections, we present our algorithms for computing canonical labelings of digraphs. We display the major steps required for calculating the maximum element $C_{max}(G)$ of G . When our algorithm has calculated the node u_1 of $MaxQ(G)$, then, it constructs the close mix-neighborhood subdigraph $N^{++}[u_1]$ of the set $S_1 = \{u_1\}$ (see Figure 1a), from which it picks a few vertices into $MaxQ(G)$. For clarity of presentation, we call this process PROGRESS 1. Then again, it builds the close mix-neighborhood subdigraph $N^{++}[S_2]$ of the nodes set $S_2 = \{u_1, u_2\}$ (see Figure 1b), from which it picks a few vertices into $MaxQ(G)$. We call this process

PROGRESS 2. ... Then again, it builds the close mix-neighborhood subdigraph $N^{++}[S_r]$ of the nodes set $S_r = \{u_1, u_2, \dots, u_r\}$ (see Figure 1c,d), from which it picks a few vertices into $MaxQ(G)$. We call this process PROGRESS r . This process continues until it puts all vertices in G into $MaxQ(G)$ (see Figure 1e,f).

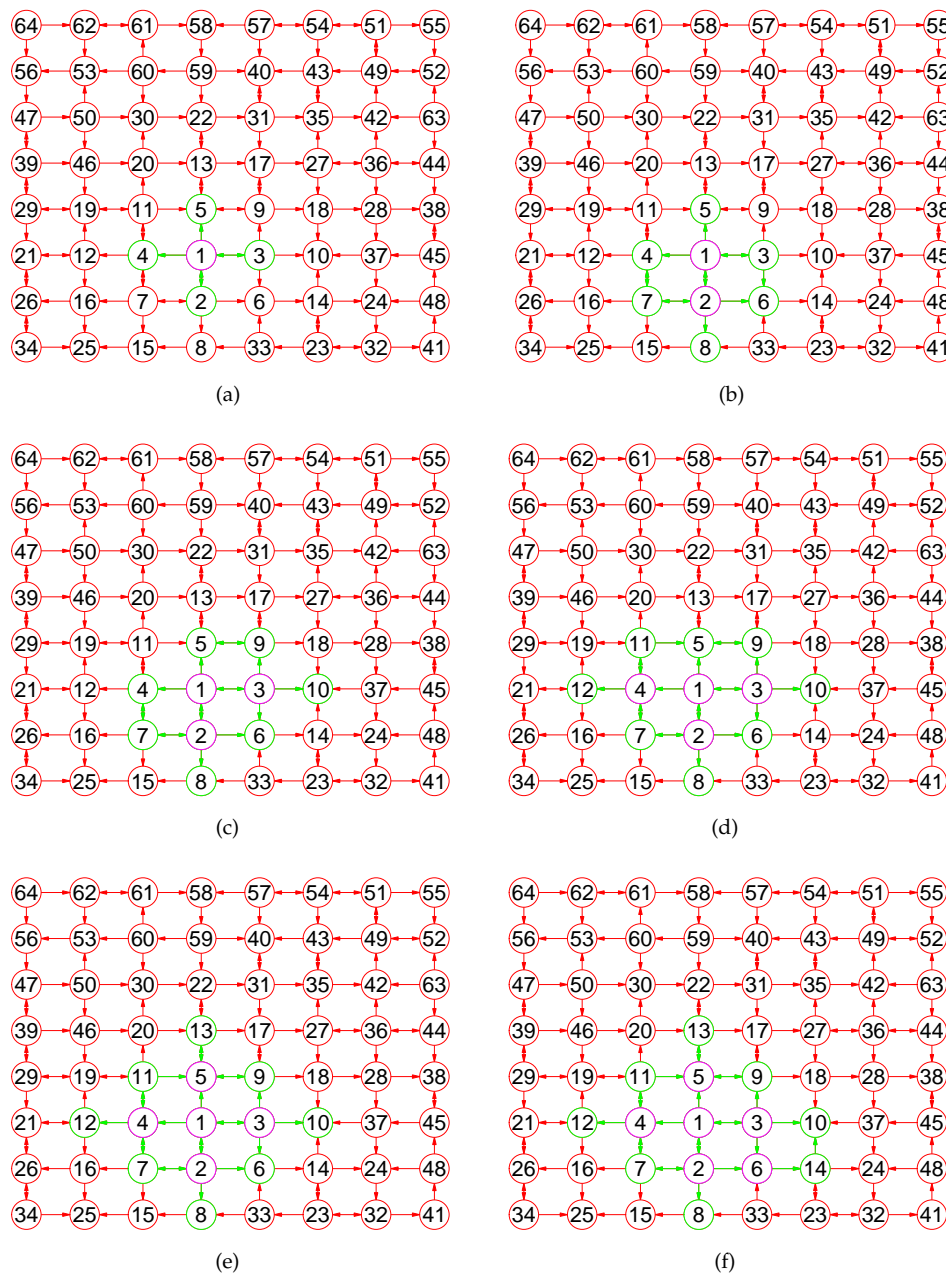


Figure 1. Each close *mix*-neighborhood subdigraph of different nodes sets of an 8×8 grid digraph $G_{8,8}$ consists of pink and green nodes and edges. (a) The close mix-neighborhood subdigraph $N^{++}[1]$; (b) $N^{++}[1,2]$; (c) $N^{++}[1,2,3]$; (d) $N^{++}[1,2,3,4]$; (e) $N^{++}[1,2,3,4,5]$; (f) $N^{++}[1,2,3,4,5,6]$.

For PROGRESS 1, after calculating $N^{++}[S_1]$, by Lemma 2, our algorithm arranges all nodes of $N^{++}[S_1]$ into a single chain L_1 (see Algorithm 1). For simplify, let $V_1 = V(N^{++}[S_1])$. If there are two nodes $v_i, v_j \in L_1$ satisfying condition $v_i \succ v_j$ with respect to $N^{++}[S_1]$, then it continues to determine whether $v_i \succ v_j$ or $v_i \prec v_j$ with respect to G . If $v_i \succ v_j$, it rearranges v_i in front of the v_j in L_1 . Otherwise, it rearranges v_i in back of the v_j in L_1 .

For each *PROGRESS* $r, r = 2, 3, \dots$, when calculating $N^{++}[S_r]$, our algorithm continues to calculate the close-in-neighborhood subdigraph $N^-[S_r]$ and let $N^{++}[S_r] = N^-[S_r]$ if condition $N^{++}[S_r] = \emptyset$ holds for some r .

For each *PROGRESS* $r, r = 2, 3, \dots$, after calculating $N^{++}[S_r]$, our algorithm in turn computes $V_2 = V(N^{++}(S_{r-1})) \cap V(N^{++}(u_r))$ (see Figure 2d), $V_3 = V(N^{++}(S_{r-1})) - V(N^{++}(u_r))$ (see Figure 2e), $V_4 = V(N^{++}(u_r)) - V(N^{++}(S_{r-1}))$ (see Figure 2f), and the outdegree sequences $d_{N^{++}[S_r]}^+(V_2), d_{N^{++}[S_r]}^+(V_3), d_{N^{++}[S_r]}^+(V_4)$ in decreasing order respectively, where $S_{r-1} = \{u_1, u_2, \dots, u_{r-1}\}$. It can be shown that $V_2 \cup V_3 \cup V_4 = V(N^{++}(S_r))$ and $V_i \cap V_j = \emptyset$ for $i \neq j, i, j = 2, 3, 4$. By Lemma 2, it arranges all nodes of V_i into a single chain L_i (see Algorithm 1) with $i = 2, 3, 4$, respectively.

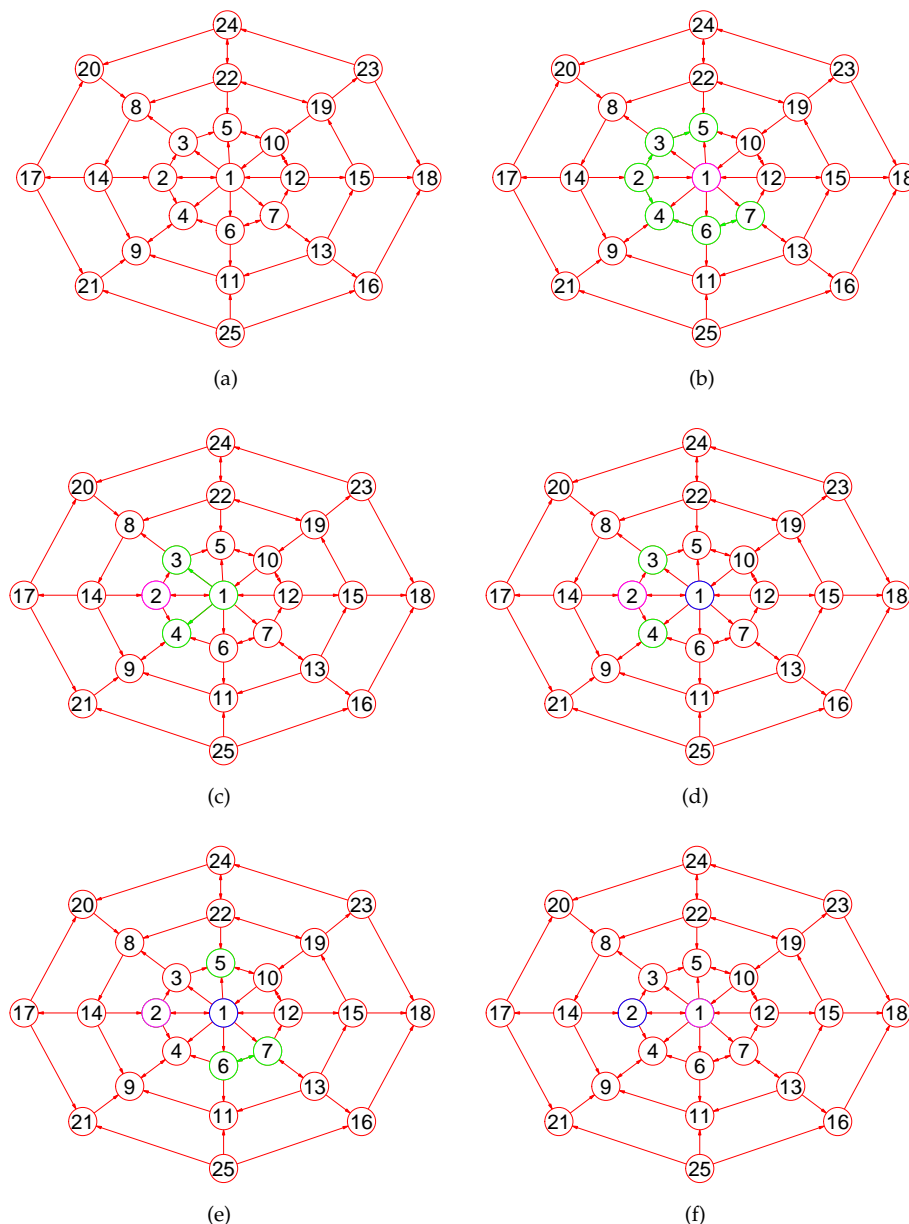


Figure 2. A wheel graph G , two open mix-neighborhood subdigraphs $N^{++}(1)$ and $N^{++}(2)$, and the three relevant nodes sets generated by the boolean operations of $N^{++}(1)$ and $N^{++}(2)$. (a) A wheel digraph G ; (b) The open mix-neighborhood subdigraph $N^{++}(1)$; (c) The open mix-neighborhood subdigraph $N^{++}(2)$; (d) $V(N^{++}(1)) \cap V(N^{++}(2)) = \{3, 4\}$; (e) $V(N^{++}(1)) - V(N^{++}(2)) = \{5, 6, 7\}$; (f) $V(N^{++}(2)) - V(N^{++}(1)) = \emptyset$.

Algorithm 1: Arrange all nodes of V_i into a single chain L_i for a close mix-neighborhood subdigraph $N^{++}[S_r]$ with $i = 1, 2, 3, 4$, respectively where $V_1 = V(N^{++}[S_1])$, $V_2 = V(N^{++}(S_{r-1})) \cap V(N^{++}(u_r))$, $V_3 = V(N^{++}(S_{r-1})) - V(N^{++}(u_r))$, and $V_4 = V(N^{++}(u_r)) - V(N^{++}(S_{r-1}))$.

Input : 1. An simple connected digraph G with n nodes and the close mix-neighborhood subdigraph $N^{++}[S_r] = \langle V(N^{++}[S_r]), E(N^{++}[S_r]) \rangle$ of a nodes set $S_r \subseteq V(G)$.

2. A nodes set $V_i = \{v_1, v_2, \dots, v_t\}$ stored in an array *NodeArray*.

Output: A list L used to store and rank all nodes of the chain L_i .

```

1  $l \leftarrow 1; j \leftarrow 1; flag \leftarrow 0;$ 
2 Initialize local variables NodeMax, Node;
3 Compute the outdegree sequence  $d_{N^{++}[S_r]}^+(V_i) = (d_{N^{++}[S_r]}^+(v_1), d_{N^{++}[S_r]}^+(v_2), \dots, d_{N^{++}[S_r]}^+(v_t))$  in
   descending order;
4 for ( $l \leftarrow 0$  to  $t - 2$ ) {
5    $NodeMax \leftarrow NodeArray[l];$ 
6   for ( $j \leftarrow l + 1$  to  $t - 1$ ) {
7      $Node \leftarrow NodeArray[j];$ 
8      $flag \leftarrow 0;$ 
9     Compare_Two_Nodes( $N^{++}[S_r]$ , NodeMax, Node, flag); see Algorithm 2;
10    if ( $flag == 0$ ) then //  $NodeMax \succ Node$  with regard to  $N^{++}[S_r]$ 
11      go back to the beginning of the for-loop;
12    else if ( $flag == 1$ ) then //  $NodeMax \prec Node$  with regard to  $N^{++}[S_r]$ 
13       $NodeMax \leftarrow Node;$ 
14    else if ( $flag == 2$ ) then //  $NodeMax \asymp Node$  with regard to  $N^{++}[S_r]$ 
15       $flag \leftarrow 0;$ 
16      Compare_Two_Nodes( $G$ , NodeMax, Node, flag); see Algorithm 2;
17      if ( $flag == 0$  or  $2$ ) then //  $NodeMax \succcurlyeq Node$  with regard to  $G$ 
18        go back to the beginning of the for-loop;
19      else if ( $flag == 1$ ) then //  $NodeMax \prec Node$  with regard to  $G$ 
20         $NodeMax \leftarrow Node;$ 
21    Add the NodeMax to the end of the list  $L$ ;

```

Algorithm 2: Compare the entire mix diffusion outdegree sequences $d_H^{\tau}[N_1^{++}[v]]$, $d_H^{\tau}[N_2^{++}[v]]$, \dots , $d_H^{\tau}[N_{\rho[v]}^{++}[v]]$ and $d_H^{\tau}[N_1^{++}[w]]$, $d_H^{\tau}[N_2^{++}[w]]$, \dots , $d_H^{\tau}[N_{\rho[w]}^{++}[w]]$ of two nodes v and w in H .

```

1 void Compare_Two_Nodes(Graph H, CNode v, CNode w, int &flag)
2 {
3   r ← 0; k ← 0; flag ← 0;
4   while (r ≤ ρ++[v] and r ≤ ρ++[w]){ // r ≤ mix diffusion radius ρ++[v] and
   ρ++[w]
5     r ← r + 1; k ← 1;
6     Compute the close r mix-neighborhood subdigraph Nr++[v] of v in H;
7     Compute the p connected components H1, H2, ⋯, Hp of Nr++[v];
8     Calculate the entire mix diffusion outdegree sequencedHτ[Nr++[v]] = (dτ+(H1), dτ+(H2), ⋯,
   dτ+(Hp)), satisfying conditions dτ+(H1) ≥ dτ+(H2) ≥ ⋯ ≥ dτ+(Hp);
9     Compute the close r mix-neighborhood subdigraph Nr++[w] of w in H;
10    Compute the l connected components J1, J2, ⋯, Jl of Nr++[w];
11    Calculate the entire mix diffusion outdegree sequencedHτ[Nr++[w]] = (dτ+(J1), dτ+(J2), ⋯,
   dτ+(Jl)), satisfying conditions dτ+(J1) ≥ dτ+(J2) ≥ ⋯ ≥ dτ+(Jl);
12    while (k ≤ p and k ≤ l){
13      flag ← 0;
14      // Compare two mix diffusion outdegree sequences (see Algorithm 3)
   Compare_Two_Mix_Diffusion_OutDegree_Sequences(H, Nr++[v], Nr++[w], dτ+(Hk),
   dτ+(Jk), flag);
15      if (flag == 0 or flag == 1)then // v ≻ w or v ≺ w with regard to H
16        return;
17      k ← k + 1;
18    if (k ≤ p and k > l)then
19      flag ← 0; return; // v ≻ w with regard to H;
20    else if (k > p and k ≤ l)then
21      flag ← 1; return; // v ≺ w with regard to H;
22    if (r ≤ ρ++[v] and r > ρ++[w])then
23      flag ← 0; return; // v ≻ w with regard to H;
24    else if (r > ρ++[v] and r ≤ ρ++[w])then
25      flag ← 1; return; // v ≺ w with regard to H;
26    else
27      flag ← 2; return; // v ≍ w with regard to H;
28 }

```

Algorithm 3: Compare two *mix diffusion outdegree sequences* $d_{\tau}^{+}(H_k)$ and $d_{\tau}^{+}(J_k)$ of two nodes v and w in H .

```

1 void Compare_Two_Mix_Diffusion_OutDegree_Sequences(Digraph H, Digraph  $N_r^{++}[v]$ ,
  Digraph  $N_r^{++}[w]$ , QueueArray  $d_{\tau}^{+}(H_k)$ , QueueArray  $d_{\tau}^{+}(J_k)$ , int &flag)
2 {
3    $d_1 \leftarrow 0; d_2 \leftarrow 0; m \leftarrow 0; l \leftarrow 1;$ 
4   Let mix diffusion outdegree sequence  $d_{\tau}^{+}(H_k) = (d_{N_r^{++}[v]}^{+}(V_0), d_{N_r^{++}[v]}^{+}(V_1), \dots, d_{N_r^{++}[v]}^{+}(V_s));$ 
  // Have  $V_0 \vee V_1 \vee \dots \vee V_s = V(H_k)$  and  $V_i \cap V_j = \emptyset$  for  $i \neq j, i, j = 0, 1, \dots, s;$ 
5   Let mix diffusion outdegree sequence  $d_{\tau}^{+}(J_k) = (d_{N_r^{++}[w]}^{+}(U_0), d_{N_r^{++}[w]}^{+}(U_1), \dots, d_{N_r^{++}[w]}^{+}(U_t));$ 
  // Have  $U_0 \vee U_1 \vee \dots \vee U_t = V(J_k)$  and  $U_i \cap U_j = \emptyset$  for  $i \neq j, i, j = 0, 1, \dots, t;$ 
6   while ( $m \leq s$  and  $m \leq t$ ) { //  $m \leq s$  and  $m \leq t$ 
7     Let the outdegree sequence  $d_{N_r[v]}^{+}(V_m) = (d_{N_r[v]}^{+}(a_1), d_{N_r[v]}^{+}(a_2), \dots, d_{N_r[v]}^{+}(a_j), \dots,$ 
   $d_{N_r[v]}^{+}(a_{\lambda}))$  in descending order; // Have  $V_m = \{a_1, a_2, \dots, a_{\lambda}\};$ 
8     Let the outdegree sequence  $d_{N_r[w]}^{+}(U_m) = (d_{N_r[w]}^{+}(b_1), d_{N_r[w]}^{+}(b_2), \dots, d_{N_r[w]}^{+}(b_j), \dots,$ 
   $d_{N_r[w]}^{+}(b_{\mu}))$  in descending order; // Have  $U_m = \{b_1, b_2, \dots, b_{\mu}\};$ 
9     while ( $l \leq \lambda$  and  $l \leq \mu$ ) {
10       $d_1 \leftarrow d_{N_r[v]}^{+}(a_l); d_2 \leftarrow d_{N_r[w]}^{+}(b_l);$ 
11      // Compare  $d_1$  and  $d_2$  (see Algorithm 4);
12      Compare_ $d_1\_and\_d_2(m, d_1, d_2, flag);$ 
13      if ( $flag == 0$  or  $flag == 1$ ) then
14        return; //  $d_{\tau}^{+}(H_k) \neq d_{\tau}^{+}(J_k)$  with regard to  $H;$ 
15       $l \leftarrow l + 1;$ 
16    if ( $l \leq \lambda$  and  $l > \mu$ ) then
17       $flag \leftarrow 0;$  return; //  $d_{\tau}^{+}(H_k) > d_{\tau}^{+}(J_k)$  with regard to  $H;$ 
18    else if ( $l > \lambda$  and  $l \leq \mu$ ) then
19       $flag \leftarrow 1;$  return; //  $d_{\tau}^{+}(H_k) < d_{\tau}^{+}(J_k)$  with regard to  $H;$ 
20     $m \leftarrow m + 1; l \leftarrow 1;$ 
21    if ( $m \leq s$  and  $m > t$ ) then
22       $flag \leftarrow 0;$  return; //  $d_{\tau}^{+}(H_k) > d_{\tau}^{+}(J_k)$  with regard to  $H;$ 
23    else if ( $m > s$  and  $m \leq t$ ) then
24       $flag \leftarrow 1;$  return; //  $d_{\tau}^{+}(H_k) < d_{\tau}^{+}(J_k)$  with regard to  $H;$ 
25    else
26       $flag \leftarrow 2;$  return; //  $d_{\tau}^{+}(H_k) == d_{\tau}^{+}(J_k)$  with regard to  $H;$ 
27  }

```

Algorithm 4: Compare d_1 and d_2 .

```

1 void Compare_d1_and_d2(int m, int d1, int d2, int &flag)
2 {
3     flag ← 0;
4     if (m == 0)then
5         if (d1 < d2)then
6             | flag ← 0; return;
7         else if (d1 > d2)then
8             | flag ← 1; return;
9     else
10        if (d1 > d2)then
11            | flag ← 0; return;
12        else if (d1 < d2)then
13            | flag ← 1; return;
14    flag ← 2; return;
15 }

```

Next, our algorithm sequentially performs the following steps for the nodes of L_i with $i = 2, 3, 4$:

1. Starting from the head of L_2 , our algorithm successively determines whether each node $u \in L_2$ satisfies the *outdegree multiplicity* condition $dm_{N^{++}(S_r)}^+(u) = 1$. If the number of vertices satisfying condition $dm_{N^{++}(S_r)}^+(u) = 1$ is less than 2 in L_2 , it puts u into $MaxQ(G)$. If there are two nodes $v_i, v_j \in L_2$ satisfying condition $v_i \asymp v_j$ with respect to $N^{++}[S_r]$, then it continues to determine whether $v_i \succ v_j$ or $v_i \prec v_j$ with respect to G . If $v_i \succ v_j$, it rearranges the v_i in front of the v_j in L_2 (see Algorithm 1). Otherwise, it rearranges the v_i in back of the v_j in L_2 (see Algorithm 1).
2. Except the nodes added into $MaxQ(G)$, it uses a queue Q to store the intermediate nodes to be added to $MaxQ(G)$. After performing Step 1, it sequentially determines whether or not each node $u \in L_2$ is in Q . If u is in Q and the number of nodes added into $MaxQ(G)$ is less than 2 in the preceding procedures, it puts u into $MaxQ(G)$ and simultaneously deletes u from Q . Otherwise, it inputs u into Q .
3. For L_3 , if the number of vertices of L_2 added into $MaxQ(G)$ is 0 and the number of vertices satisfying condition $dm_{N^{++}(S_r)}^+(u) = 1$ with $u \in L_3$ is less than 2, it puts u into $MaxQ(G)$. The remaining processing steps are the same as for L_2 .
4. For L_4 , if the number of vertices of L_2 and L_3 added into $MaxQ(G)$ is 0 and the number of vertices satisfying the condition $dm_{N^{++}(S_r)}^+(u) = 1$ with $u \in L_4$ is less than 2, it puts u into $MaxQ(G)$. The remaining processing steps are the same as for L_2 .

Our algorithm uses an array $MaxQ$ to store the nodes of $MaxQ(G)$ and an array Q to keep the nodes to be added to $MaxQ(G)$ temporarily.

Experiments demonstrate that our approach is a novel way by which one can accurately calculate *MaxEm digraphs* (defined in Section 2) for many types of digraphs. Figures 3–21 produced by our software show the correctness of our software for calculating *MaxEm digraphs* of these digraph classes aforementioned.

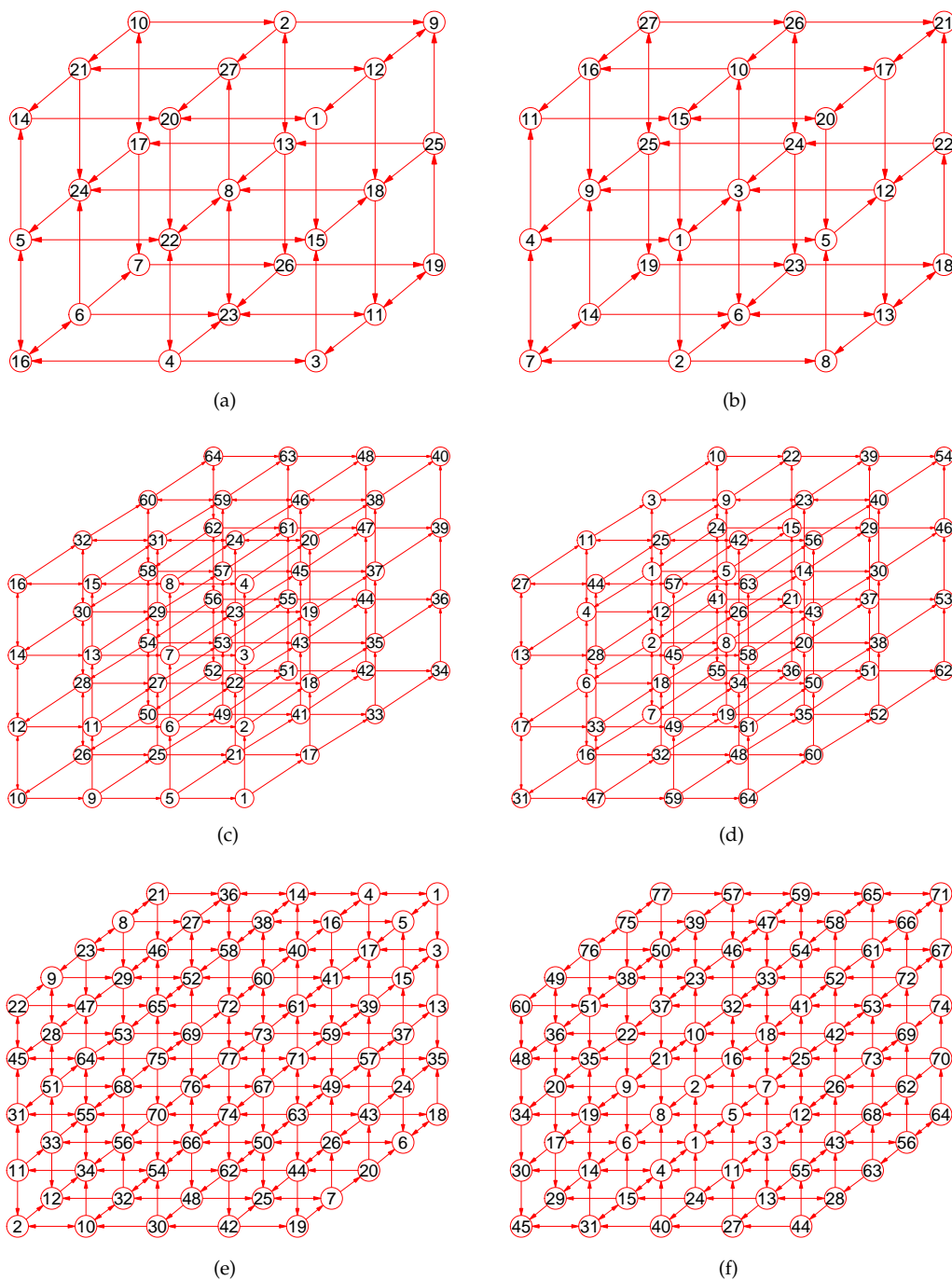


Figure 3. The *MaxEm* digraphs of three digraphs $G_{3,3,3}$, $G_{4,4,4}$, and G_1 . (a) The $3 \times 3 \times 3$ grid digraph $G_{3,3,3}$ with 27 nodes and 54 directed edges; (b) The *MaxEm* digraph of $G_{3,3,3}$; (c) The $4 \times 4 \times 4$ grid digraph $G_{4,4,4}$ with 64 nodes and 144 directed edges; (d) The *MaxEm* digraph of $G_{4,4,4}$; (e) A digraph G_1 with 77 nodes and 196 directed edges; (f) The *MaxEm* digraph of G_1 .

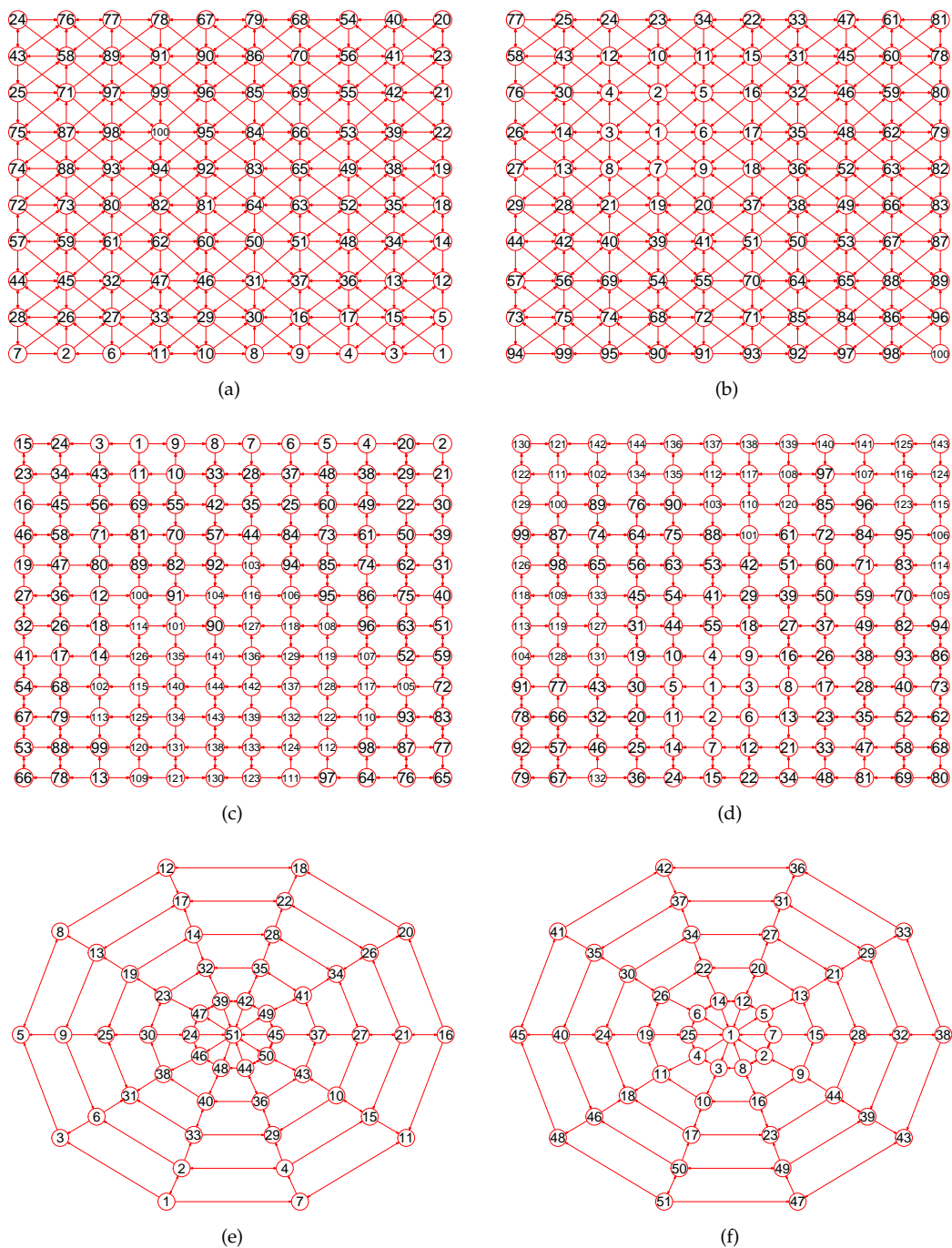


Figure 4. The $MaxEm$ digraphs of three digraphs G_2 , $G_{12,12}$, and G_3 . (a) A 10×10 king digraph G_2 with 100 vertices and 342 directed edges ; (b) The $MaxEm$ digraph of G_2 ; (c) A 12×12 grid digraph $G_{12,12}$ with 144 nodes and 264 edges; (d) The $MaxEm$ digraph of $G_{12,12}$; (e) A wheel digraph G_3 with 51 nodes and 100 directed edges; (f) The $MaxEm$ digraph of G_3 .

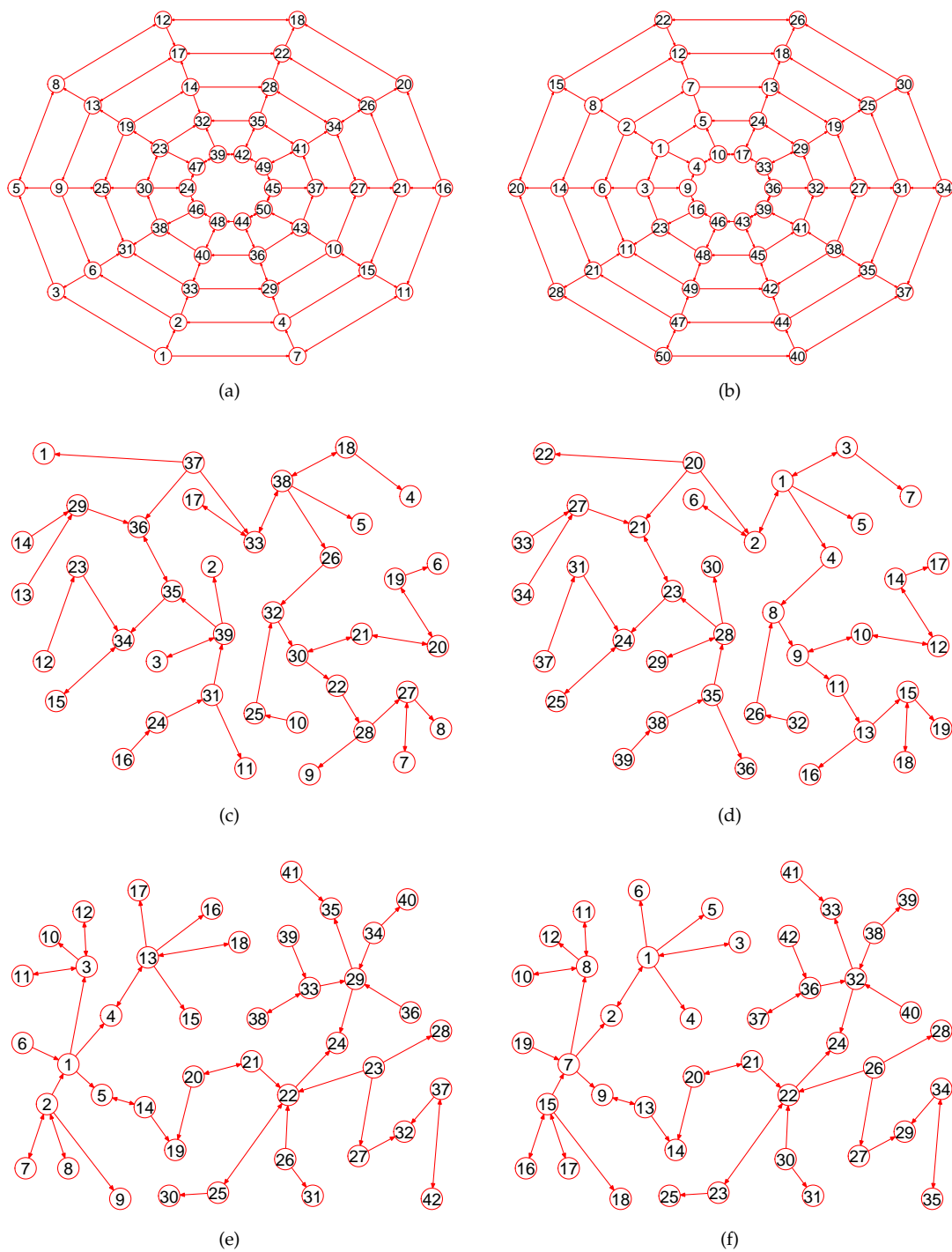


Figure 5. The *MaxEm* digraphs of G_4 , T_1 , and T_2 . (a) A digraph G_4 with 50 nodes and 90 directed edges; (b) The *MaxEm* digraph of G_4 ; (c) A directed tree T_1 with 39 nodes and 38 directed edges; (d) The *MaxEm* digraph of T_1 ; (e) A directed tree T_2 with 42 nodes and 41 directed edges; (f) The *MaxEm* digraph of T_2 .

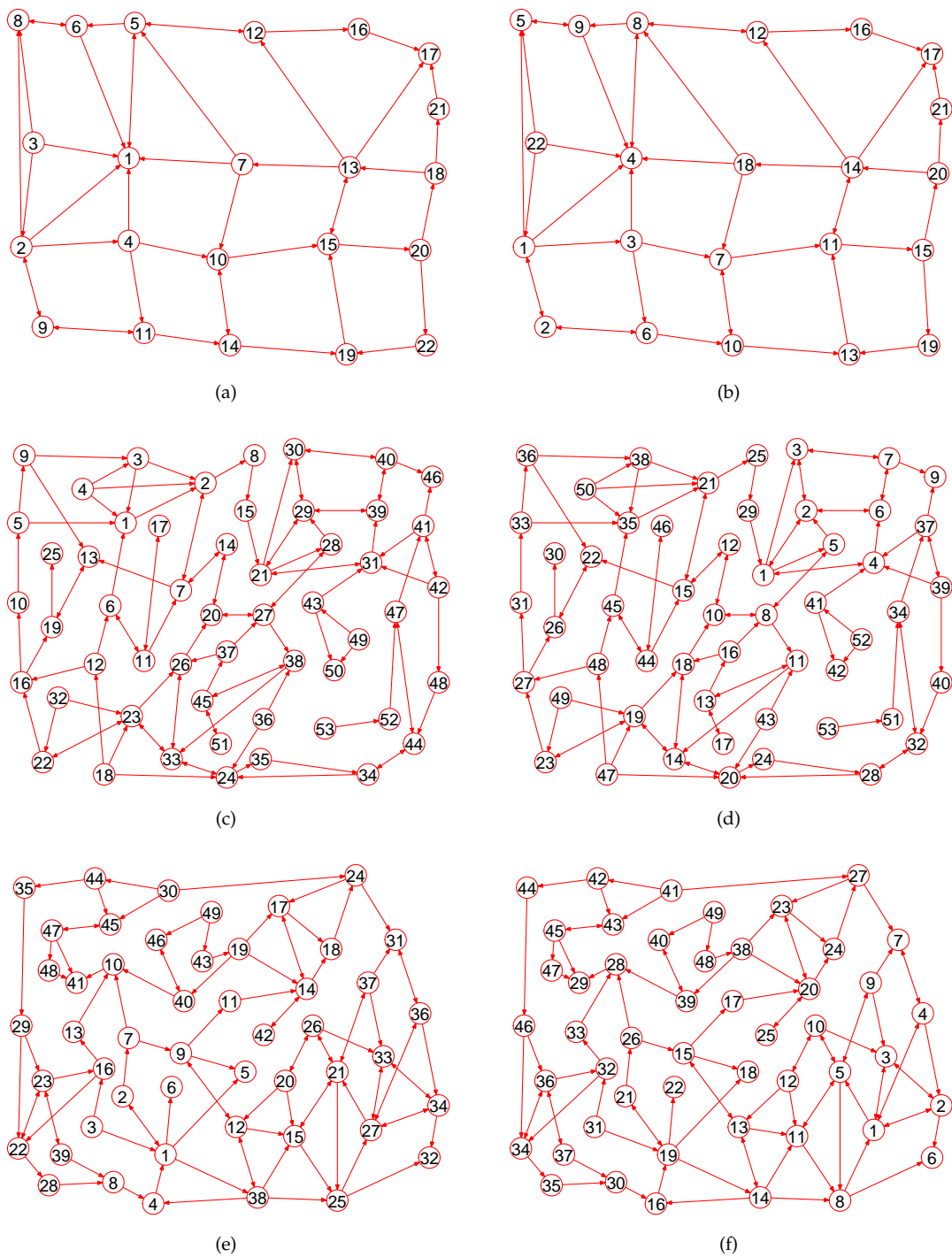


Figure 6. The *MaxEm* digraphs of three digraphs G_5 , G_6 , and G_7 . (a) A digraph G_5 with 22 nodes and 37 directed edges; (b) The *MaxEm* digraph of G_5 ; (c) A digraph G_6 with 53 nodes and 80 directed edges; (d) The *MaxEm* digraph of G_6 ; (e) A graph G_7 with 49 nodes and 78 directed edges; (f) The *MaxEm* digraph of G_7 .

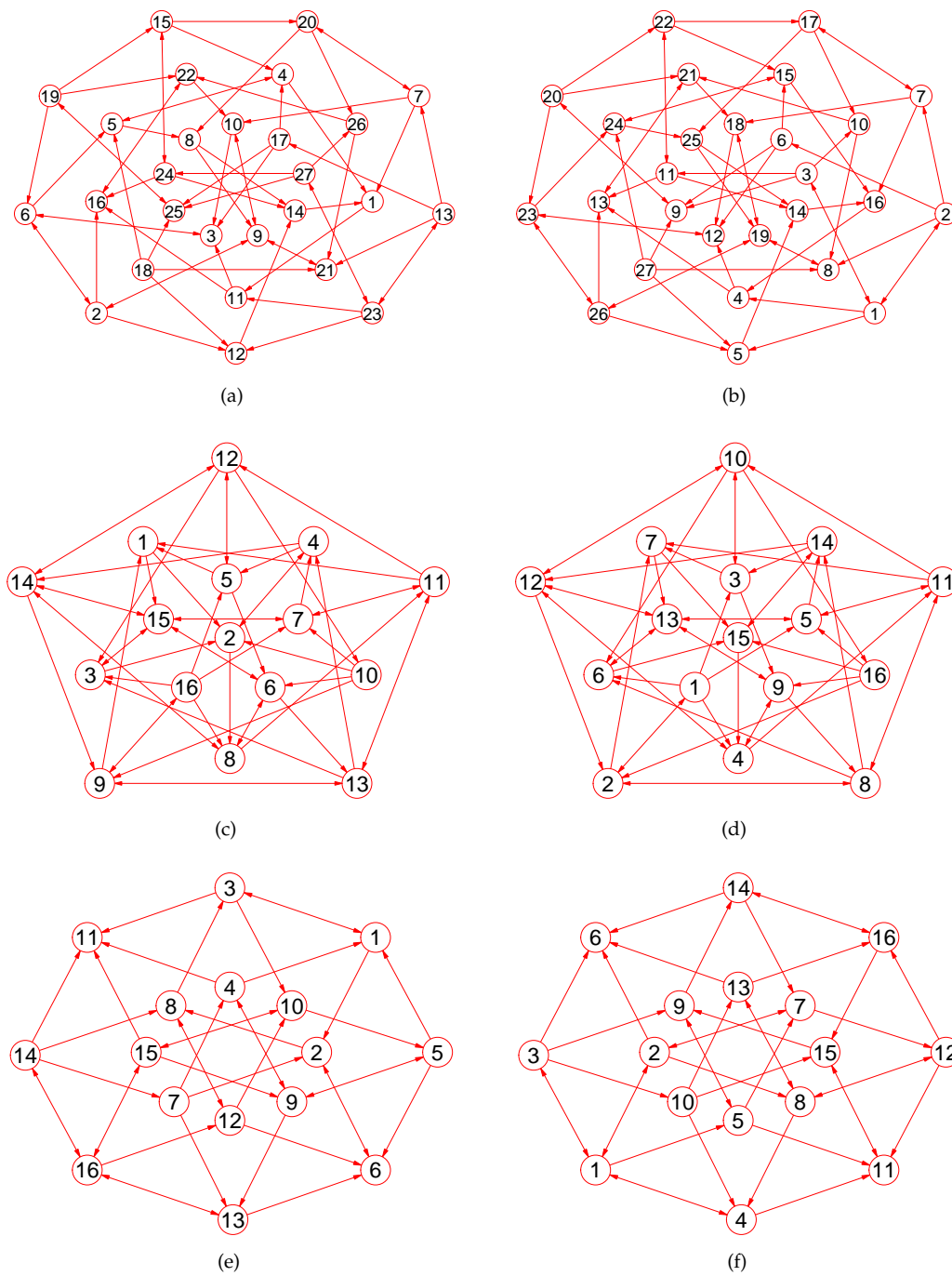


Figure 7. The *MaxEm* digraphs of three digraphs G_8 , G_9 , and G_{10} . (a) The Doyle digraph G_8 with 27 nodes and 54 directed edges; (b) The *MaxEm* digraph of G_8 ; (c) The Clebsch digraph G_9 with 16 nodes and 40 directed edges; (d) The *MaxEm* digraph of G_9 ; (e) The 4-hypercube digraph G_{10} with 16 nodes and 32 directed edges; (f) The *MaxEm* digraph of G_{10} .

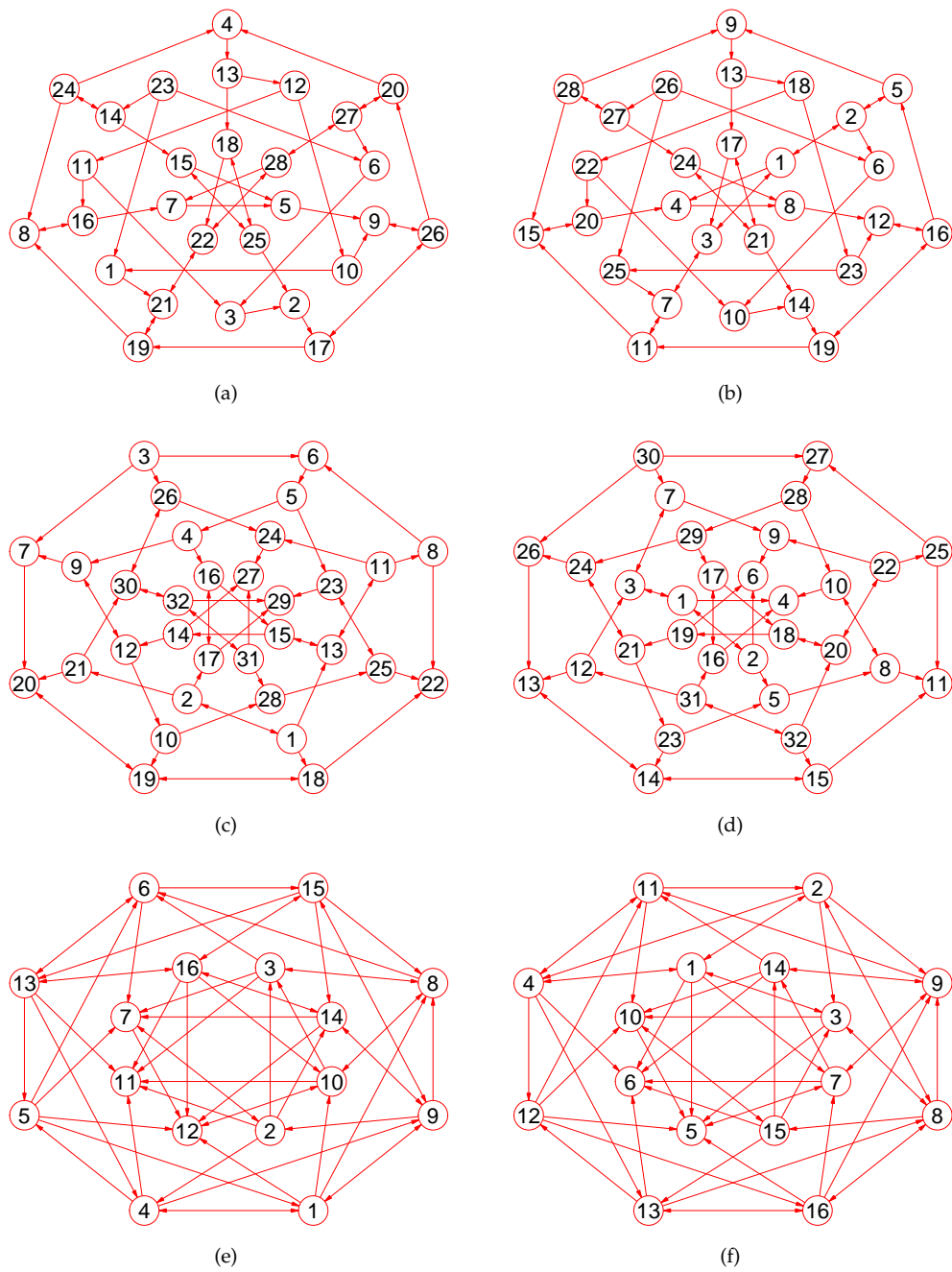


Figure 8. The *MaxEm* digraphs of three digraphs G_{11} , G_{12} , and G_{13} . (a) The coxeter digraph G_{11} with 28 nodes and 42 directed edges; (b) The *MaxEm* digraph of G_{11} ; (c) The Dyck digraph G_{12} with 32 vertices and 48 directed edges; (d) The *MaxEm* digraph of G_{12} ; (e) A Shrikhande digraph G_{13} with 16 vertices and 48 directed edges; (f) The *MaxEm* digraph of G_{13} .

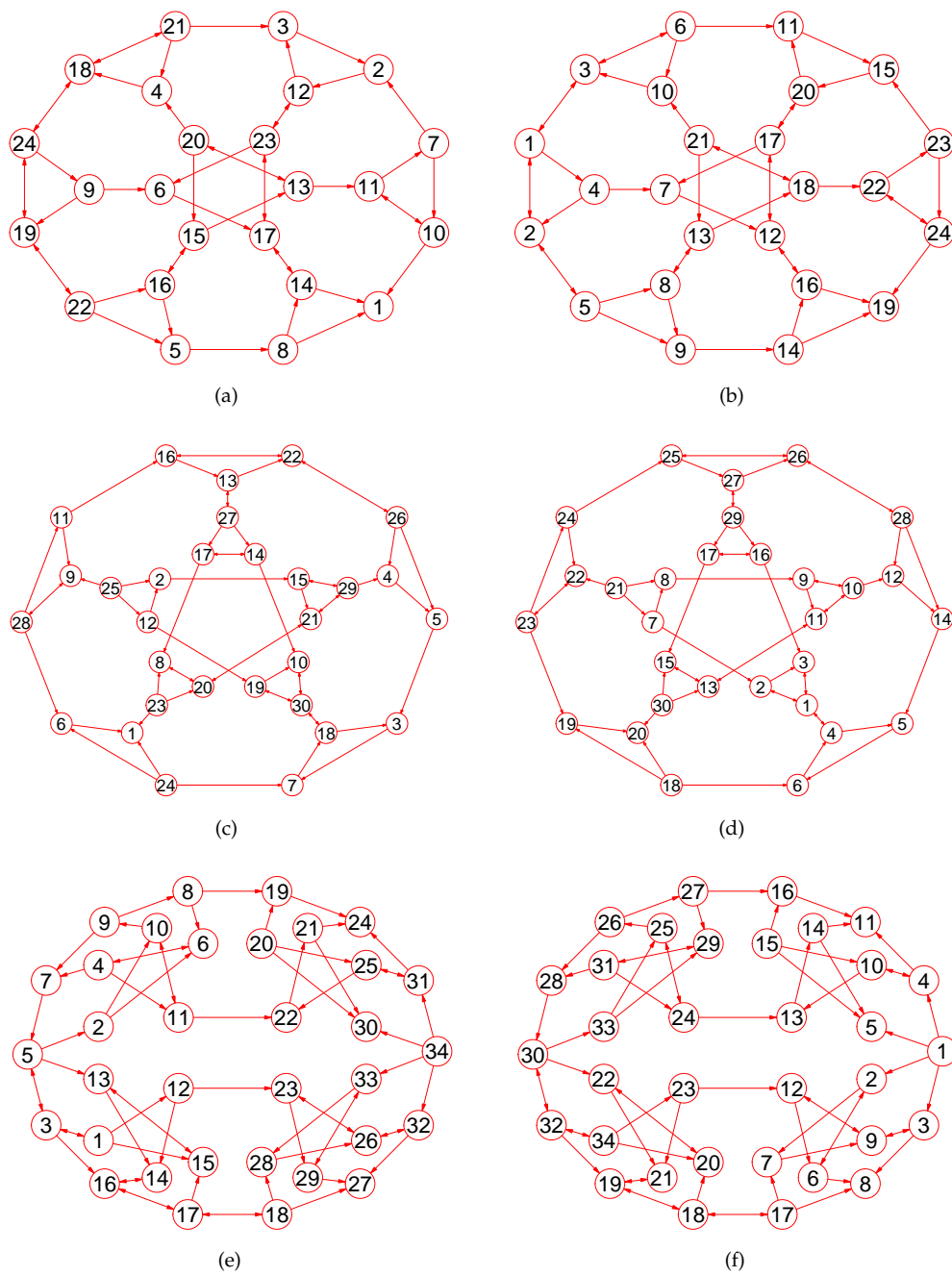


Figure 9. The *MaxEm* digraphs of three digraphs G_{14} , G_{15} , and G_{16} . (a) The 6th order cube-connected cycle digraph G_{14} with 24 vertices and 36 directed edges; (b) The *MaxEm* digraph of G_{14} ; (c) A triangle-replaced digraph G_{15} with 30 nodes and 45 directed edges; (d) The *MaxEm* digraph of G_{15} ; (e) The Thomassen digraph G_{16} with 34 vertices and 52 directed edges; (f) The *MaxEm* digraph of G_{16} .

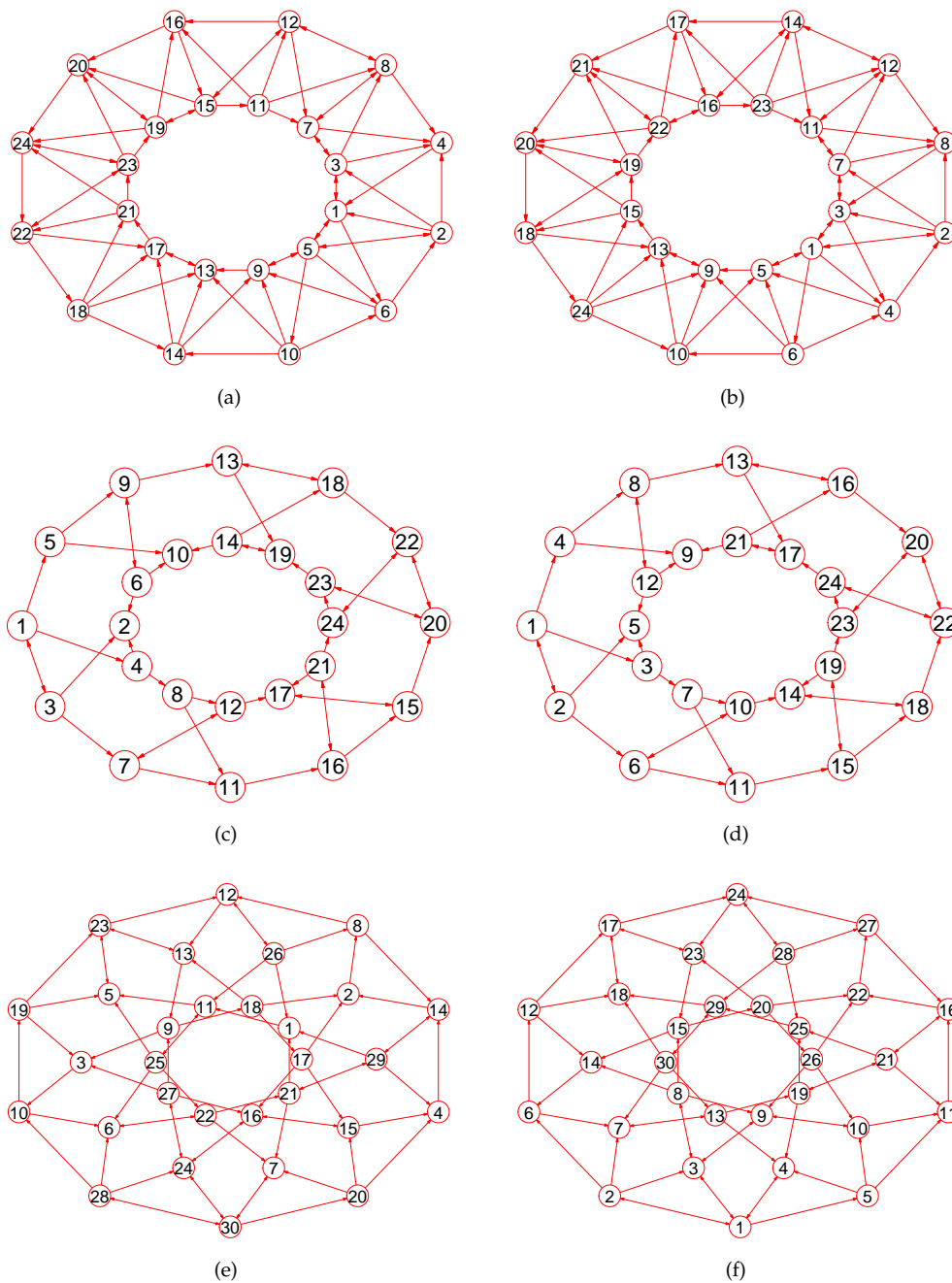


Figure 10. The *MaxEm* digraphs of three digraphs G_{17} , G_{18} , and G_{19} . (a) The musical digraph G_{17} with 24 nodes and 60 directed edges; (b) The *MaxEm* digraph of G_{17} ; (c) The 12-crossed prism digraph G_{18} with 24 nodes and 36 directed edges; (d) The *MaxEm* digraph of G_{18} ; (e) The Icosidodecahedral digraph G_{19} with 30 nodes and 60 directed edges; (f) The *MaxEm* digraph of G_{19} .

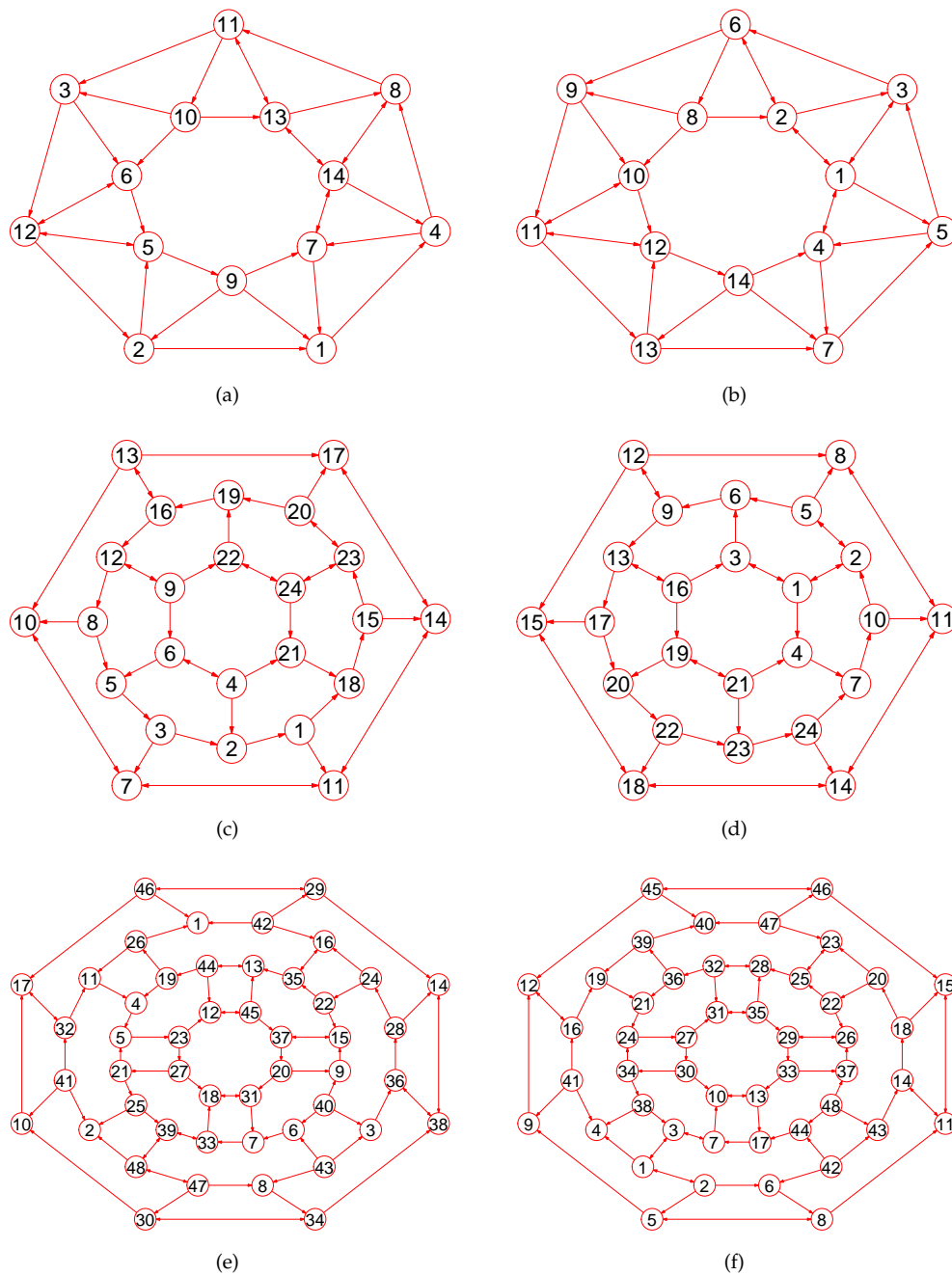


Figure 11. The *MaxEm* digraphs of three digraphs G_{20} , G_{21} , and G_{22} . (a) The 7-antiprism digraph G_{20} with 14 vertices and 28 edges; (b) The *MaxEm* digraph of G_{20} ; (c) A fullerene digraph G_{21} with 24 vertices and 36 directed edges; (d) The *MaxEm* digraph of G_{21} ; (e) The great rhombicuboctahedron digraph G_{22} with 48 vertices and 72 directed edges; (f) The *MaxEm* digraph of G_{22} .

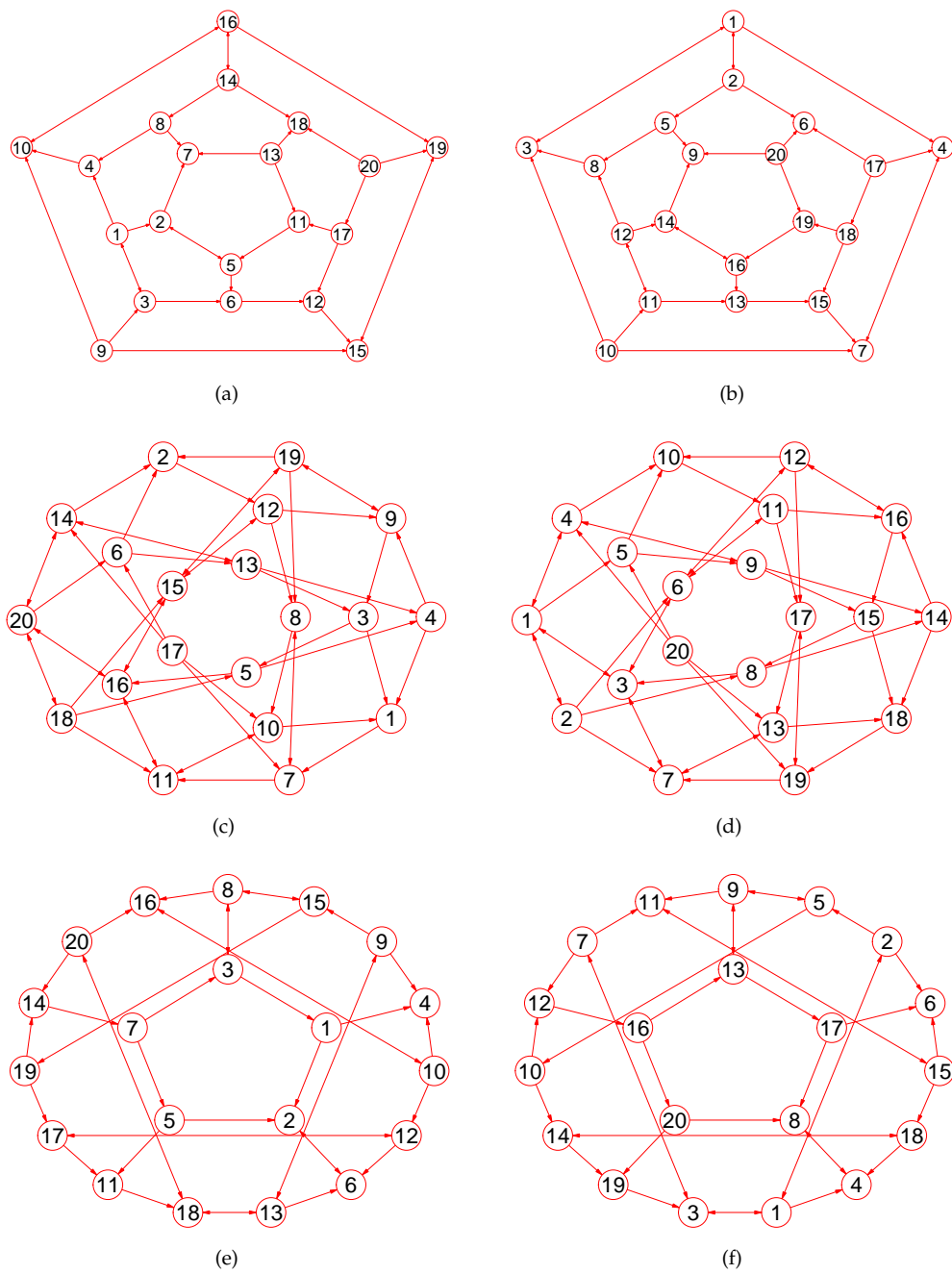


Figure 12. The *MaxEm* digraphs of three digraphs G_{23} , G_{24} , and G_{25} . (a) A Hamiltonian digraph G_{23} with 20 nodes and 30 directed edges; (b) The *MaxEm* digraph of G_{23} ; (c) The Folkman digraph G_{24} with 20 nodes and 40 directed edges; (d) The *MaxEm* digraph of G_{24} ; (e) The snark digraph G_{25} with 20 vertices and 30 directed edges; (f) The *MaxEm* digraph of G_{25} .

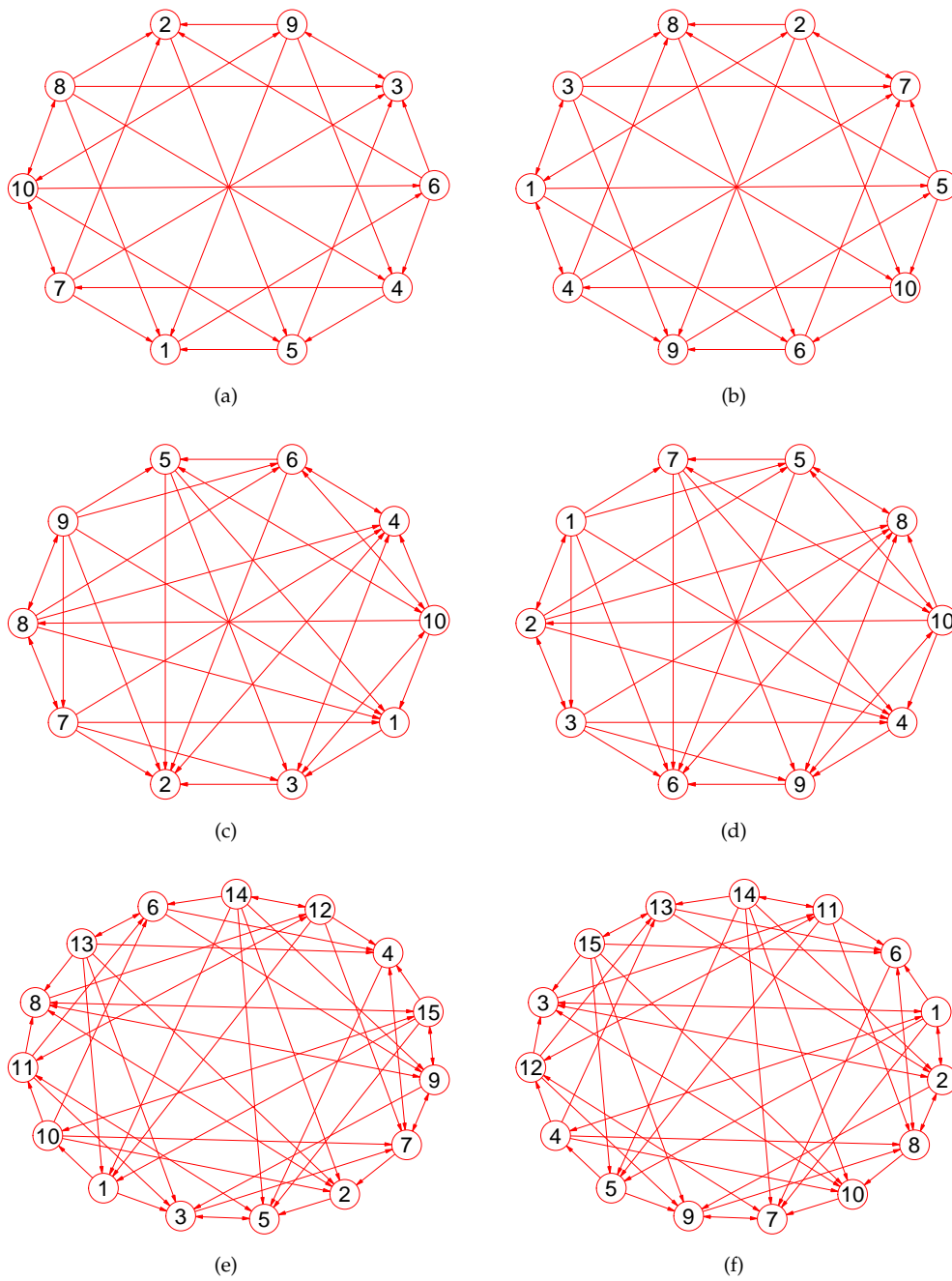


Figure 13. The *MaxEm* digraphs of three digraphs $K_{(5,5)}$, G_{26} , and G_{27} . (a) The complete bipartite digraph $K_{(5,5)}$ with 10 nodes and 25 directed edges; (b) The *MaxEm* digraph of $K_{(5,5)}$; (c) The triangular digraph G_{26} with 10 nodes and 30 directed edges; (d) The *MaxEm* digraph of G_{26} ; (e) A generalized quadrangle digraph G_{27} with 15 nodes and 45 directed edges; (f) The *MaxEm* digraph of G_{27} .

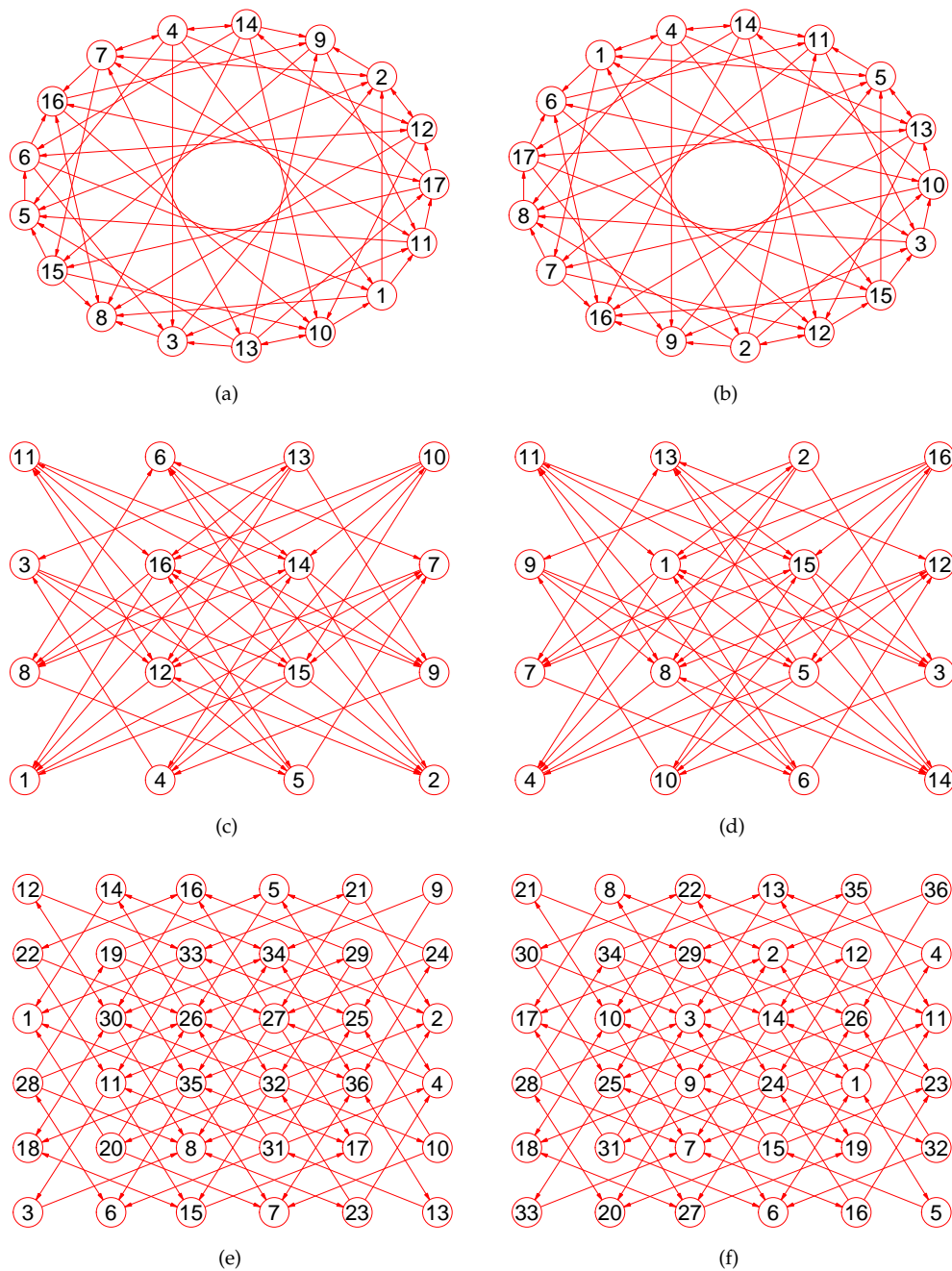


Figure 14. The *MaxEm* digraphs of three digraphs G_{28} , G_{29} , and G_{30} . (a) The 6-Andrásfai digraph G_{28} with 17 nodes and 51 edges; (b) The *MaxEm* digraph of G_{28} ; (c) The 4-dimensional Keller digraph G_{29} with 16 nodes and 46 directed edges; (d) The *MaxEm* digraph of G_{29} ; (e) The 6×6 knight digraph G_{30} with 36 vertices and 80 directed edges; (f) The *MaxEm* digraph of G_{30} .

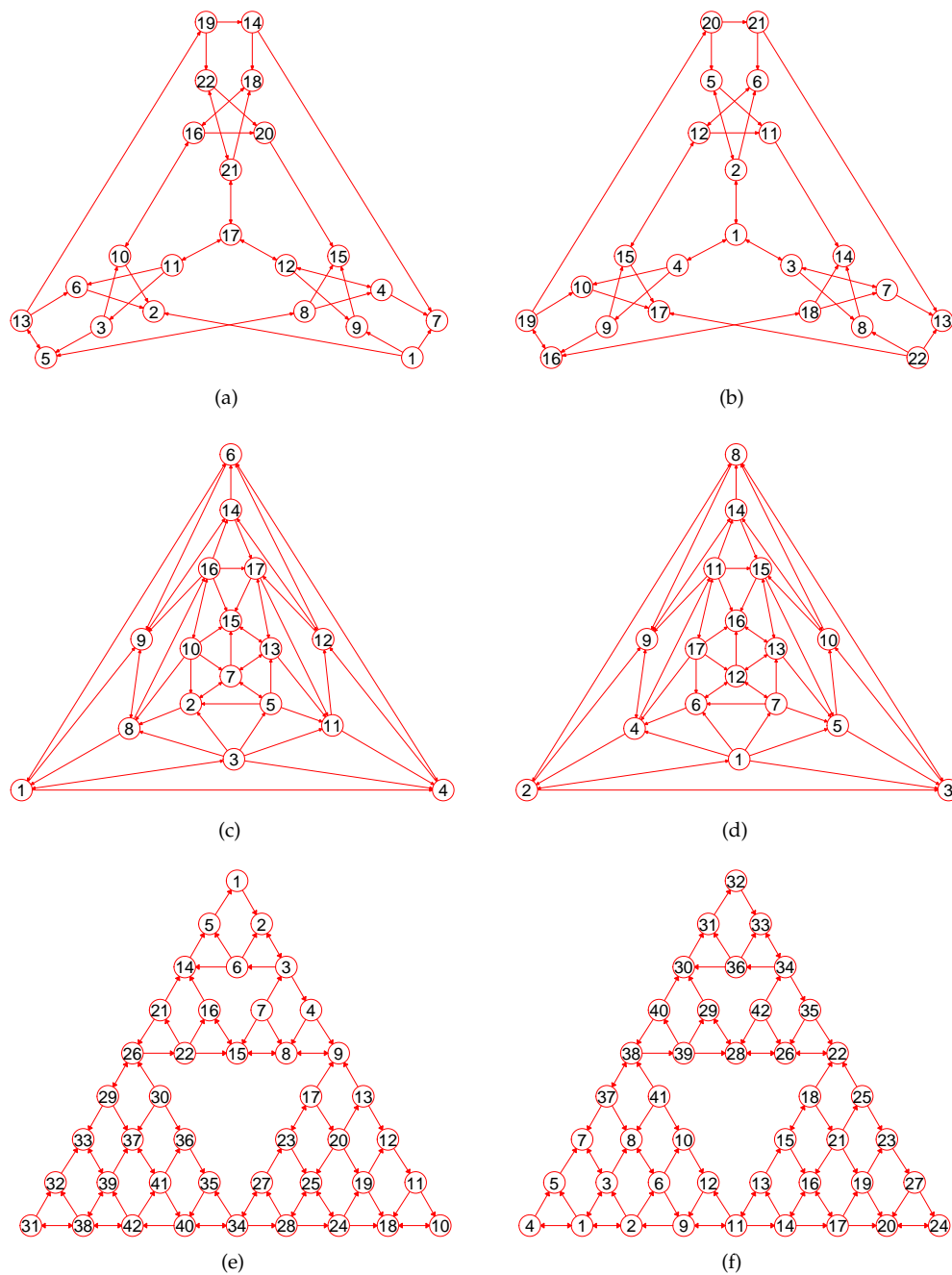


Figure 15. The *MaxEm* digraphs of three digraphs G_{31} , G_{32} , and G_{33} . (a) The Loupequine snarks digraph G_{31} with 22 nodes and 33 directed edges; (b) The *MaxEm* digraph of G_{31} ; (c) The Errera digraph G_{32} with 17 nodes and 45 directed edges; (d) The *MaxEm* digraph of G_{32} ; (e) The Sierpinski sieve digraph G_{33} with 42 nodes and 72 directed edges; (f) The *MaxEm* digraph of G_{33} .

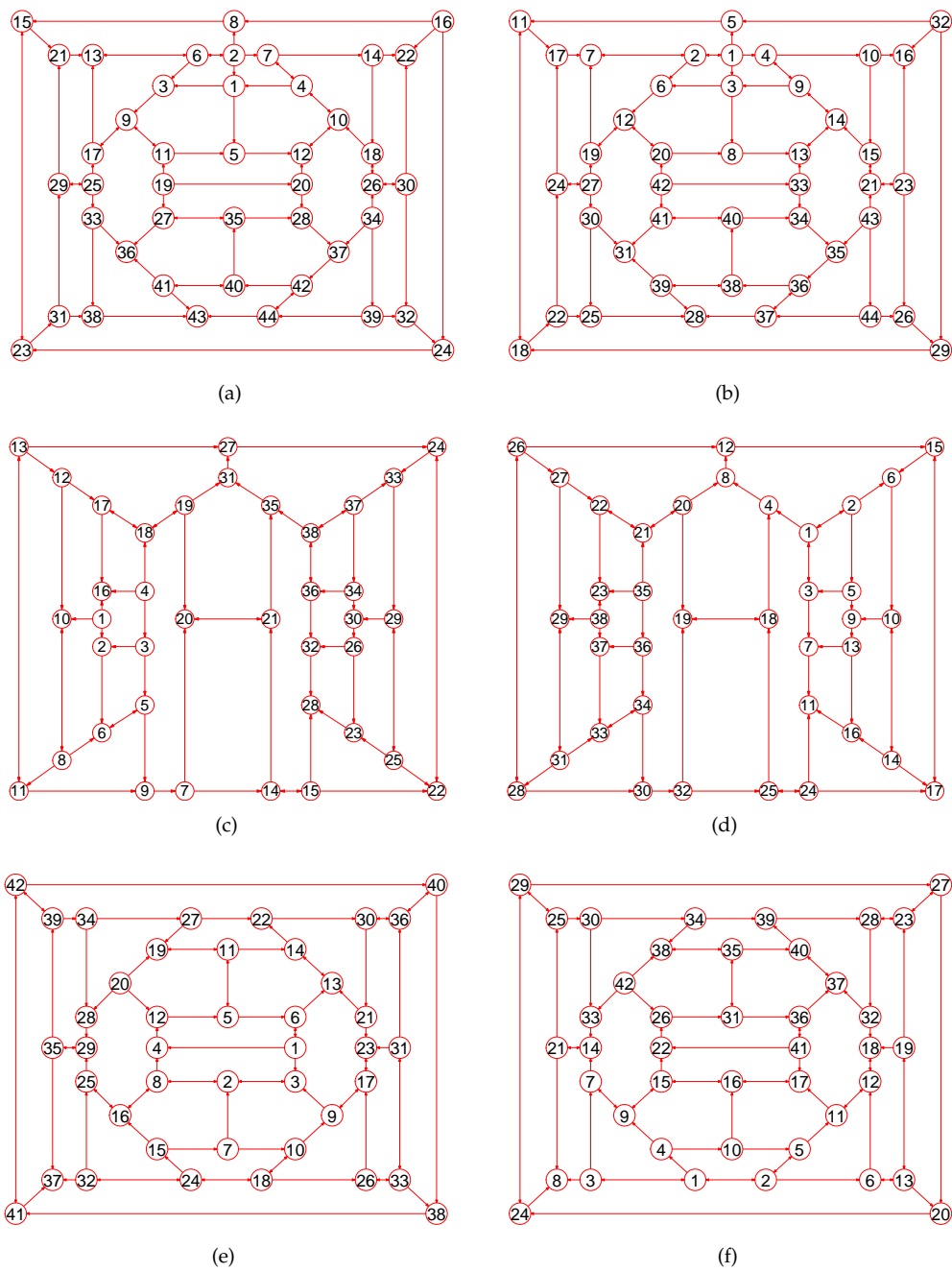


Figure 16. The *MaxEm* digraphs of three digraphs G_{34} , G_{35} , and G_{36} . (a) The Grinberg digraph G_{34} with 44 nodes and 67 directed edges; (b) The *MaxEm* digraph of G_{34} ; (c) A digraph G_{35} with 38 nodes and 57 directed edges; (d) The *MaxEm* digraph of G_{35} ; (e) The Grinberg digraph G_{36} with 42 vertices and 63 directed edges; (f) The *MaxEm* digraph of G_{36} .

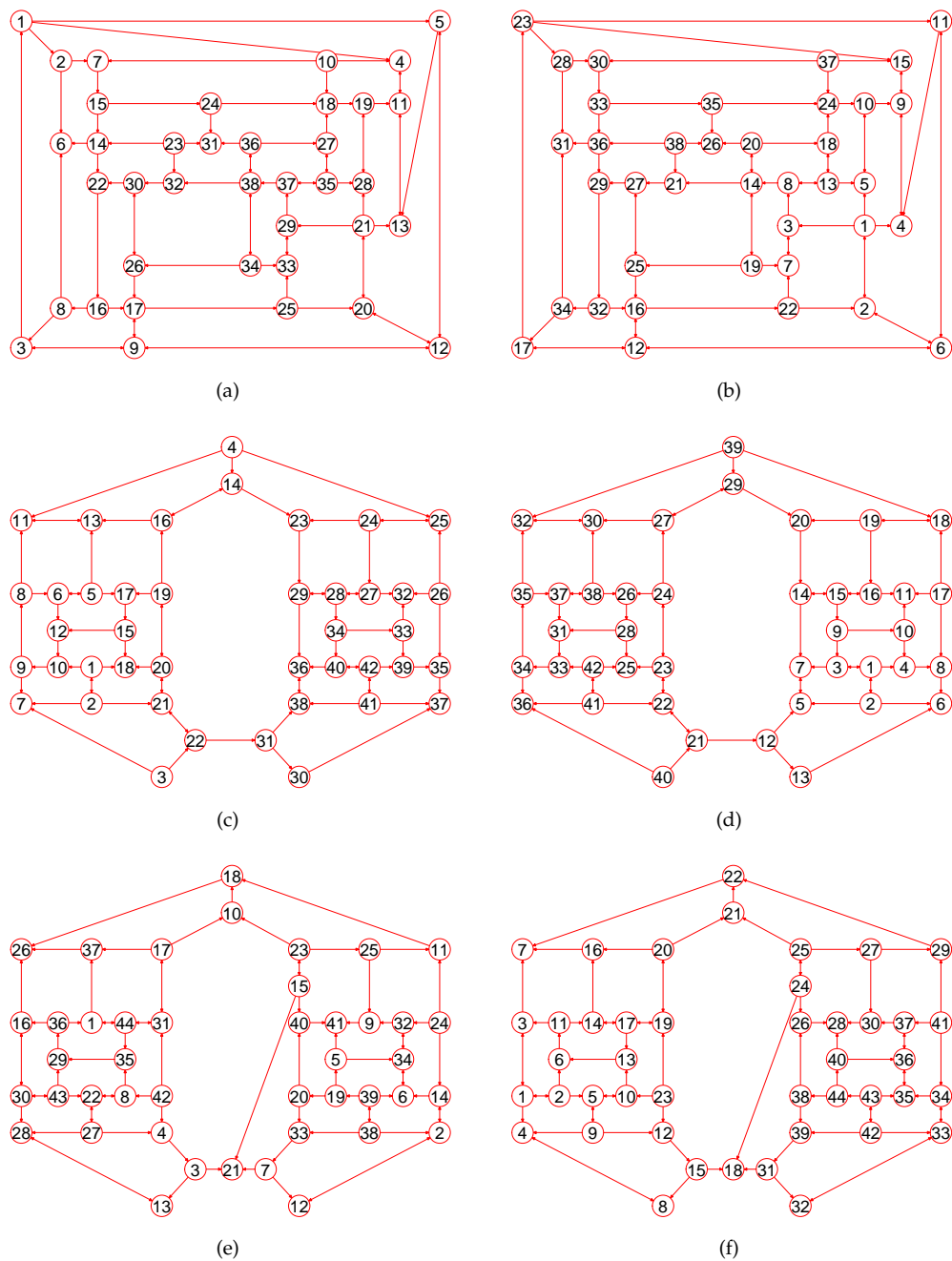


Figure 17. The *MaxEm* digraphs of three digraphs G_{37} , G_{38} , and G_{39} . (a) A pentagonal icositetrahedral digraph G_{37} with 38 nodes and 60 directed edges; (b) The *MaxEm* digraph of G_{37} ; (c) The Faulkner-Younger digraph G_{38} with 42 vertices and 62 directed edges; (d) The *MaxEm* digraph of G_{38} ; (e) The Faulkner-Younger digraph G_{39} with 44 nodes and 65 directed edges; (f) The *MaxEm* digraph of G_{39} .

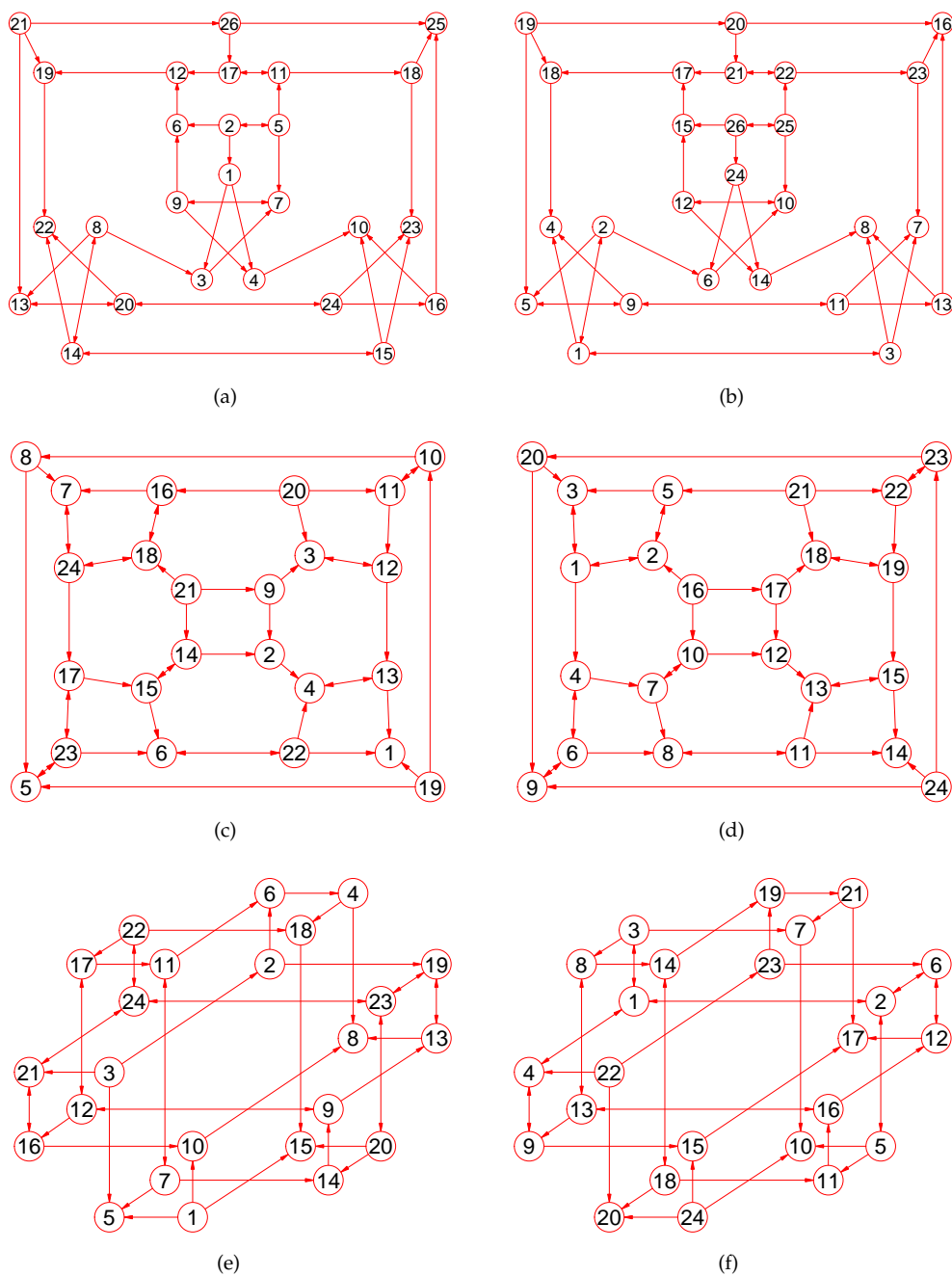


Figure 18. The *MaxEm* digraphs of three digraphs G_{40} , G_{41} , and G_{42} . (a) The Celmins Swart snarks digraph G_{40} with 26 vertices and 39 directed edges; (b) The *MaxEm* digraph of G_{40} ; (c) The truncated octahedron digraph G_{41} with 24 nodes and 36 directed edges; (d) The *MaxEm* digraph of G_{41} ; (e) The Nauru digraph G_{42} with 24 nodes and 36 directed edges; (f) The *MaxEm* digraph of G_{42} .

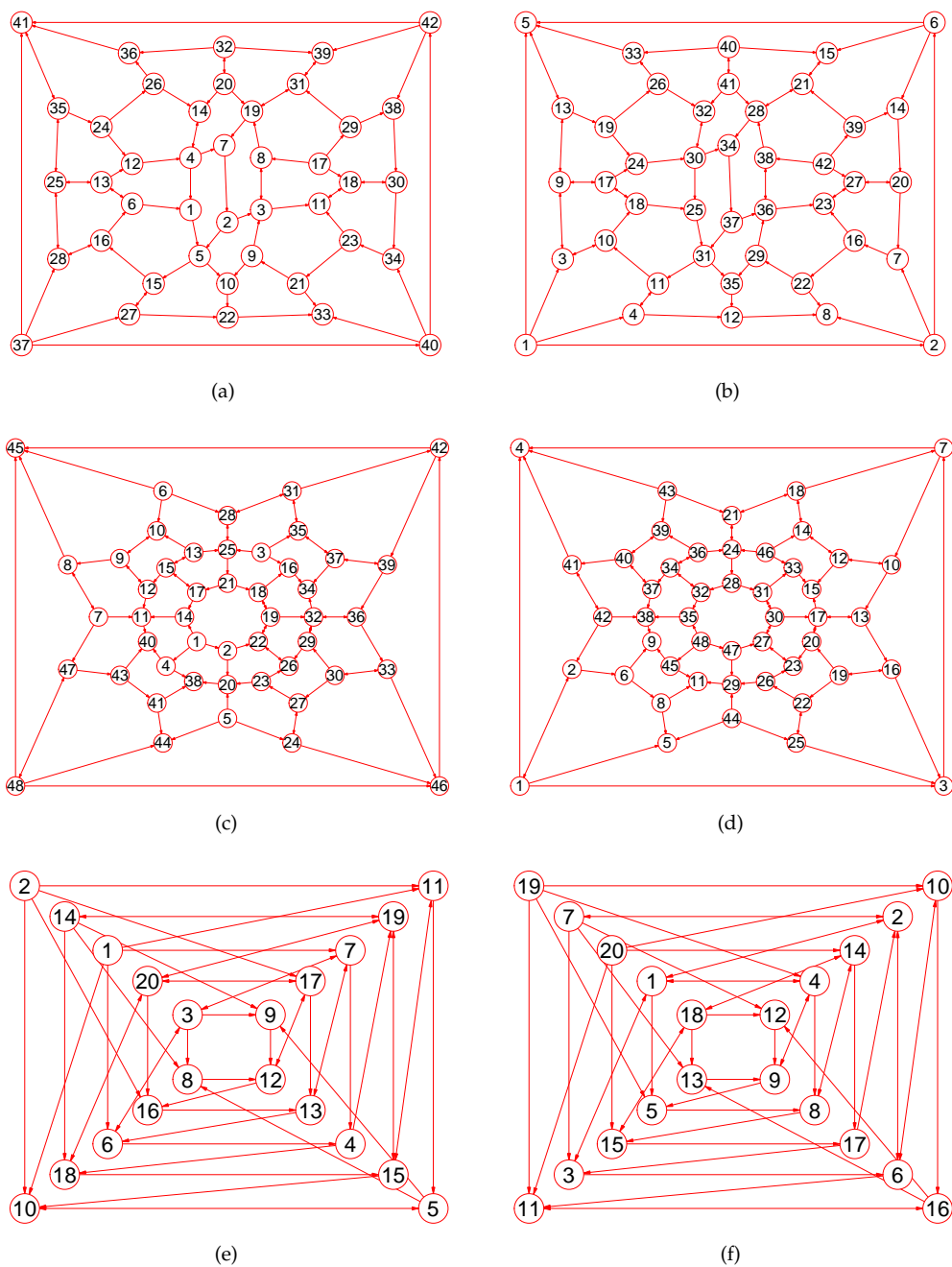


Figure 19. The *MaxEm* digraphs of three digraphs G_{43} , G_{44} , and G_{45} . (a) The Wiener-Araya digraph G_{43} with 42 nodes and 67 directed edges; (b) The *MaxEm* digraph of G_{43} ; (c) The Zamfirescu digraph G_{44} with 48 nodes and 76 directed edges; (d) The *MaxEm* digraph of G_{44} ; (e) The Folkman digraph G_{45} with 20 nodes and 40 directed edges; (f) The *MaxEm* digraph of G_{45} .

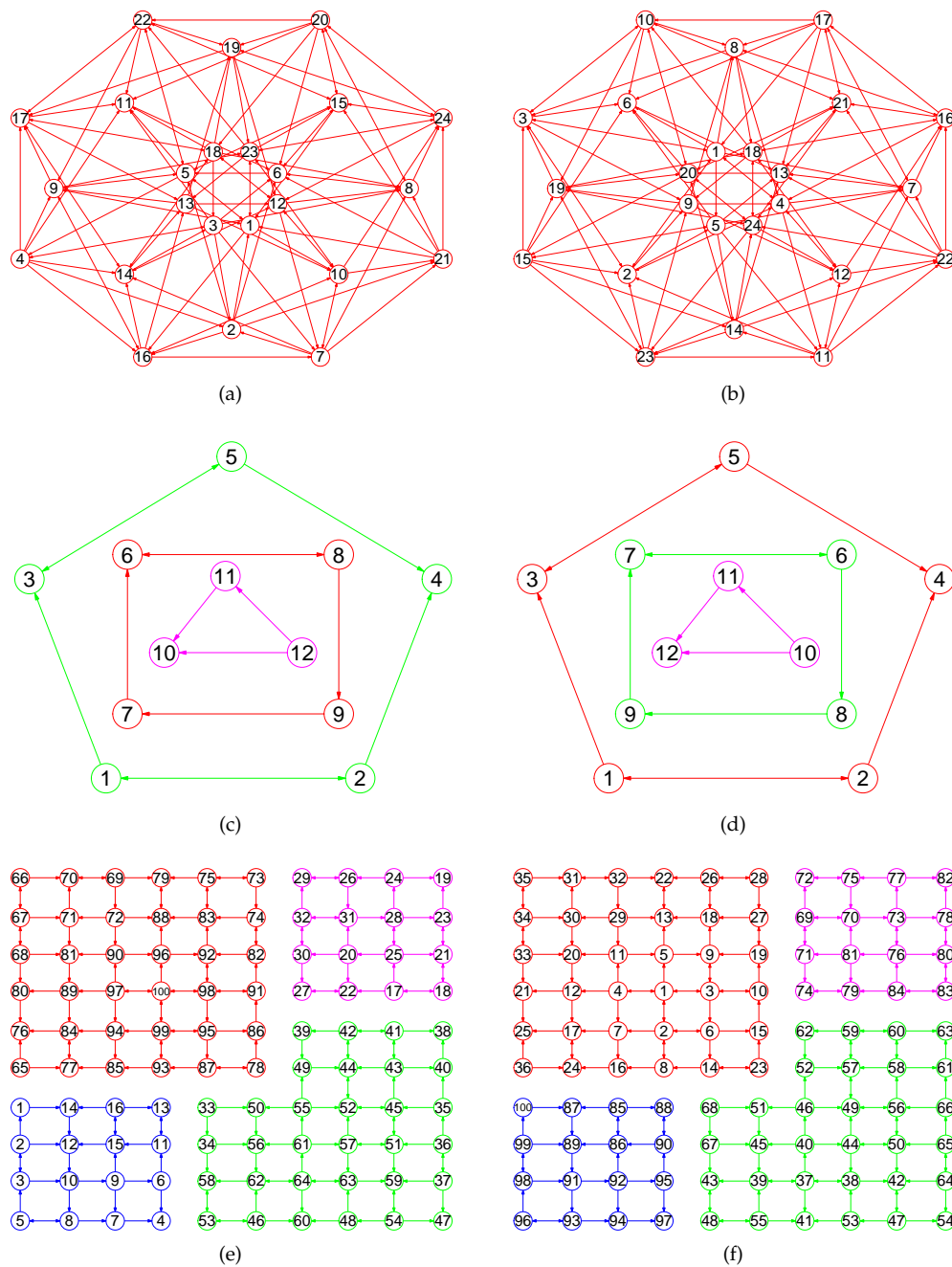


Figure 20. The *MaxEm* digraphs of three digraphs G_{46} , G_{47} , and G_{48} . (a) The 24-cell digraph G_{46} with 24 nodes and 94 directed edges; (b) The *MaxEm* digraph of G_{46} ; (c) A disconnected graph G_{47} with 12 nodes and 12 directed edges; (d) The *MaxEm* digraph of G_{47} ; (e) A disconnected digraph G_{48} that has four connected components and a total of 100 nodes and 160 directed edges; (f) The *MaxEm* digraph of G_{48} .

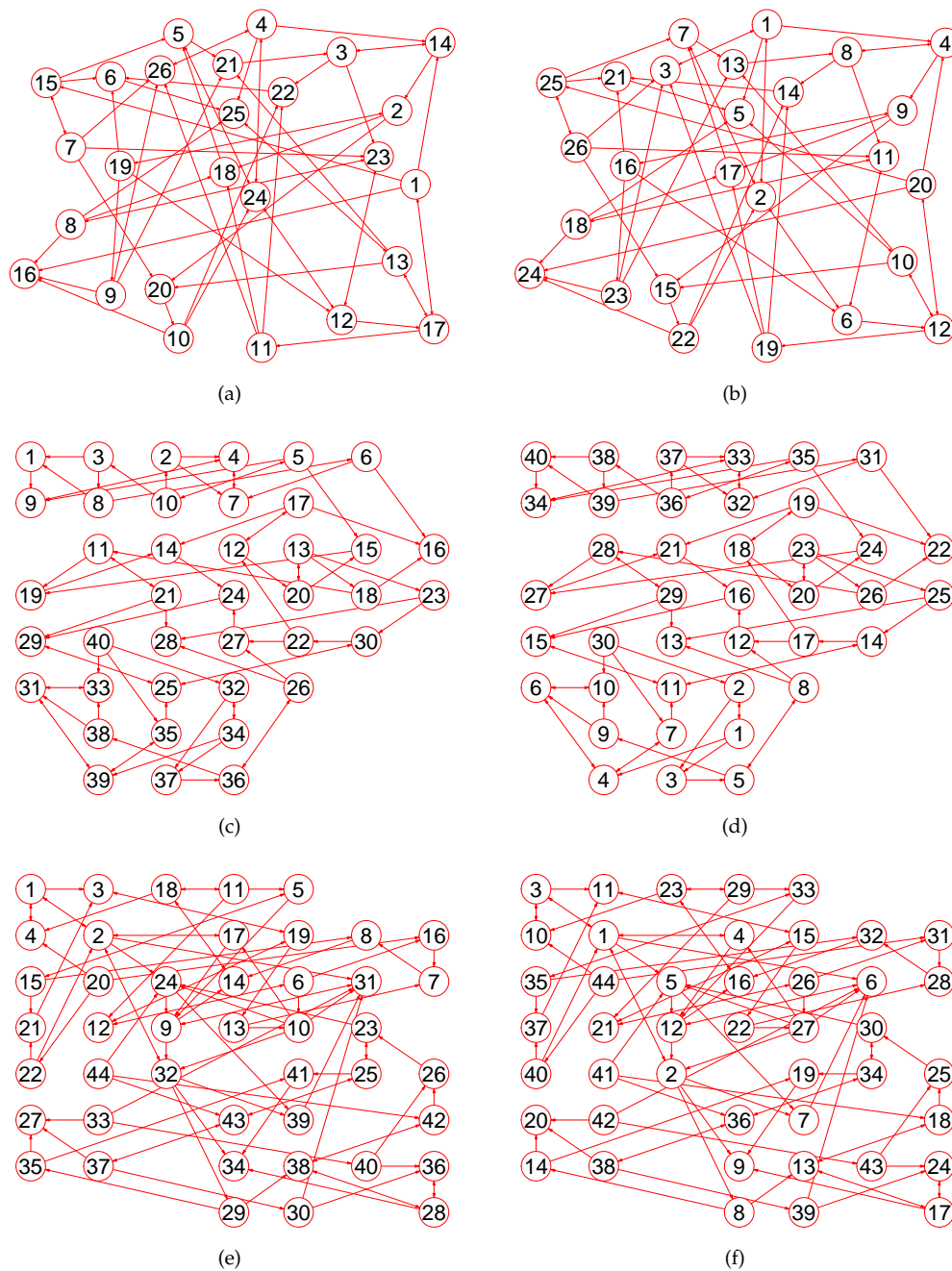


Figure 21. The *MaxEm* digraphs of three digraphs G_{49} , G_{50} , and G_{51} . (a) The projective plane digraph G_{49} with 26 nodes and 52 directed edges; (b) The *MaxEm* digraph of G_{49} ; (c) The Miyazaki digraph G_{50} with 40 nodes and 60 directed edges; (d) The *MaxEm* digraph of G_{50} ; (e) The Cubic Hypohamiltonian digraph G_{51} with 44 nodes and 75 directed edges; (f) The *MaxEm* digraph of G_{51} .

5. Software Implementation

Applying the principles described in the preceding sections, we have developed a set of software tools called GraphLabel 1.0 for computing canonical labelings of Digraphs. Our development environment includes an Intel(R) Core(TM)2 Quad CPU Q6600 @2.40GHz with 4.00 GB of RAM. The operating system is Microsoft Windows 8.1 Professional Edition. The graphics card is an NVIDIA GeForce 9800 GT. The display resolution is 1024 × 768 × 32 bits (RGB). The internal hard drive is 500 GB. The programming environment is the Microsoft Visual C++ 2012.

The software adopts object-oriented technology to design several relevant classes including the classes *CNode*, *CNodeNeighbor*, *CEdge*, *CEdgeNeighbor*, *CGraph*, and so on. A detailed description of the software functions is outside the scope of this article. We will explain it in another paper. All the figures presented in this paper are produced by using our software system.

We selected a digraph set to test the accuracy of our algorithms. Using our own software platform, we randomly produced a large number of digraphs as the test cases, including Figures 3e, 5c,e and 6. To increase the breadth and depth of our testing, we also select many test cases from the library of benchmarks [29] and online library [30] including Figures 3a,c, 4, 5a and 7–21.

We apply our algorithms to as many types of digraphs as possible. These digraphs shown in the article are just a small part of tested digraphs due to the limited length of the article. Each digraph displayed in the paper includes both the original digraph and the resulting digraph to compare entirely.

6. Conclusions and Future Work

In summary, we obtain the following conclusions: By Theorems 2–12, the paper has established a relatively complete theoretical system for calculating the *MaxEm* digraphs of digraphs. Algorithms 1–4 are novel and can accurately calculate *MaxEm* digraphs for many types of digraphs (see Figures 3–21). Algorithms 1–4 are also available for simple disconnected digraphs. For each node in a digraph *G*, the introduction of the attribute *m_NearestNode* improves the accuracy of calculating canonical labeling. Through software testing, the correctness of our algorithms is preliminarily verified. Our method can be utilized to mine the frequent subdigraph. Besides, it offers Conjecture 1.

Of course, there are still many places we need to improve, including to prove the conjectures proposed by us, enhance our software system, and use more test cases to test our procedures. In particular, we need to strengthen our algorithms so that it can calculate the canonical labeling for more types of digraphs. In future studies, we will extend our approach to mine the frequent subdigraphs and calculate the canonical labelings of weighted graphs and digraphs.

Acknowledgments: The work described in this paper was supported by Key Project of the National Natural Science Foundation of China (No. 91318301), National Natural Science Foundation of China (No. 61202080), China Postdoctoral Science Foundation (No. 2015M581032).

Author Contributions: Jianqiang Hao and Yunzhan Gong wrote the paper; Jianqiang Hao, Yawen Wang, Li Tan and Jianzhi Sun developed the software; All authors conceived and designed the experiments; Jianqiang Hao, Li Tan, and Jianzhi Sun performed the experiments; Jianqiang Hao, Yawen Wang, and Li Tan analyzed the data. All authors have read and approved the final manuscript.

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

References

1. McKay, B. Computing automorphisms and canonical labellings of graphs In *Combinatorial Mathematics; Lecture Notes in Mathematics*; Springer: Berlin/Heidelberg, Germany, 1978; Volume 686, pp. 223–232.
2. Piperno, A. Search space contraction in canonical labeling of graphs. *arXiv* **2008**, arXiv:0804.4881.
3. Junttila, T.; Kaski, P. Engineering an Efficient Canonical Labeling Tool for Large and Sparse Graphs. In Proceedings of the Ninth Workshop on Algorithm Engineering and Experiments and the Fourth Workshop on Analytic Algorithmics and Combinatorics, New Orleans, LA, USA, 6 January 2007; SIAM: Philadelphia, PA, USA, 2007; pp. 135–149.
4. Babai, L.; Luks, E.M. Canonical Labeling of Graphs. In *Proceedings of the Fifteenth Annual ACM Symposium on Theory of Computing*; ACM: New York, NY, USA, 1983; pp. 171–183.
5. Ivanciuc, O. Canonical Numbering and Constitutional Symmetry. In *Handbook of Chemoinformatics*; Wiley-VCH Verlag GmbH: Weinheim, Germany, 2008; pp. 139–160.
6. Shah, Y.J.; Davida, G.I.; McCarthy, M.K. Optimum Features and Graph Isomorphism. *IEEE Trans. Syst. Man Cybern.* **1974**, *3*, 313–319.
7. Arvind, V.; Das, B.; Köbler, J. A Logspace Algorithm for Partial 2-Tree Canonization. In *Computer Science—Theory and Applications*; Springer: Berlin/Heidelberg, Germany, 2008; Volume 5010, pp. 40–51.

8. Huan, J.; Wang, W.; Prins, J. *Efficient Mining of Frequent Subgraphs in the Presence of Isomorphism*; IEEE Computer Society: Washington, DC, USA, 2003; pp. 549–552.
9. Kuramochi, M.; Karypis, G. Finding Frequent Patterns in a Large Sparse Graph. *Data Min. Knowl. Discov.* **2005**, *11*, 243–271.
10. Kuramochi, M.; Karypis, G. An efficient algorithm for discovering frequent subgraphs. *IEEE Trans. Knowl. Data Eng.* **2004**, *16*, 1038–1051.
11. He, P.R.; Zhang, W.J.; Li, Q. Some further development on the eigensystem approach for graph isomorphism detection. *J. Frankl. Inst. Eng. Appl. Math.* **2005**, *342*, 657–673.
12. Kashani, Z.; Ahrabian, H.; Elahi, E.; Nowzari-Dalini, A.; Ansari, E.; Asadi, S.; Mohammadi, S.; Schreiber, F.; Masoudi-Nejad, A. Kavosh: A new algorithm for finding network motifs. *BMC Bioinform.* **2009**, *10*, 318.
13. Babai, L.; Kucera, L. Canonical labelling of graphs in linear average time. In Proceedings of the 20th Annual Symposium on Foundations of Computer Science, San Juan, Puerto Rico, 29–31 October 1979; pp. 39–46.
14. Arnborg, S.; Proskurowski, A. Canonical Representations of Partial 2- and 3-Trees. In *Proceedings of the 2nd Scandinavian Workshop on Algorithm Theory*; Springer: Berlin/Heidelberg, Germany, 1990; Volume 477, pp. 197–214.
15. Hao, J.; Gong, Y.; Tan, L.; Duan, D. Apply Partition Tree to Compute Canonical Labelings of Graphs. *Int. J. Grid Distrib. Comput.* **2016**, *9*, 241–263.
16. McKay, B.D. Practical Graph Isomorphism. *Congr. Numer.* **1981**, *30*, 45–87.
17. McKay, B.D. Isomorph-Free Exhaustive Generation. *J. Algorithms* **1998**, *26*, 306–324.
18. McKay, B.D.; Piperno, A. Practical graph isomorphism, II. *J. Symb. Comput.* **2014**, *60*, 94–112.
19. Ullmann, J.R. An Algorithm for Subgraph Isomorphism. *J. ACM* **1976**, *23*, 31–42.
20. Yan, X.; Han, J. gSpan: Graph-based substructure pattern mining. In Proceedings of the 2002 IEEE International Conference on Data Mining (ICDM 2002), Maebashi City, Japan, 9–12 December 2002; pp. 721–724.
21. Miyazaki, T. The complexity of McKay’s canonical labeling algorithm. In *Groups and Computation II*; American Mathematical Society: Providence, RI, USA, 1997; Volume 28, pp. 239–256.
22. Tener, G.; Deo, N. Efficient isomorphism of miyazaki graphs. *Algorithms* **2008**, *5*, 7.
23. Junttila, T.; Kaski, P. Conflict Propagation and Component Recursion for Canonical Labeling. In *Theory and Practice of Algorithms in (Computer) Systems*; Lecture Notes in Computer Science; Springer: Berlin/Heidelberg, Germany, 2011; Volume 6595, pp. 151–162.
24. López-Presa, J.L.; Anta, A.F.; Chiroque, L.N. Conauto-2.0: Fast Isomorphism Testing and Automorphism Group Computation. *arXiv* **2011**, arXiv:1108.1060.
25. Katebi, H.; Sakallah, K.; Markov, I. Graph Symmetry Detection and Canonical Labeling: Differences and Synergies. In *Proceedings Turing-100; EPIC*: Manchester, UK, 2012; Volume 10, pp. 181–195.
26. Bang-Jensen, J.; Gutin, G.Z. *Digraphs: Theory, Algorithms and Applications*, 2nd ed.; Springer: Berlin/Heidelberg, Germany, 2008.
27. Bollobás, B. *Modern Graph Theory*; Springer: Berlin/Heidelberg, Germany, 2013; Volume 184.
28. Chartrand, G.; Tian, S. Distance in digraphs. *Comput. Math. Appl.* **1997**, *34*, 15–23.
29. ALENEX 2007 Submission: Source Code, Benchmark Instances, and Summary Results. Available online: <http://www.tcs.hut.fi/Software/benchmarks/ALENEX-2007/> (accessed on 17 February 2017).
30. Weisstein, E.W. Simple Directed Graph. From MathWorld—A Wolfram Web Resource. Available online: <http://mathworld.wolfram.com/SimpleDirectedGraph.html> (accessed on 18 February 2017).

