Cyclic Linking Network

Chung Chan

***Abstract*—A general network link model is formulated, unifying the previous directed cyclic graphical network, linear deterministic network and layered linking network. It provides a seamless extension of Menger's theorem that the network can be decomposed into disjoint augmenting paths up to the min-cut value even in the presence of cycles and interference. This is obtained by developing new concepts for linking systems, which also lead to polynomial-time algorithms that compute the shortest path, maximum flow and optimal path decomposition.**

***Index Terms*—Menger's theorem, max-flow min-cut, linking system, matroid, cyclic network, linear deterministic network**

I. INTRODUCTION

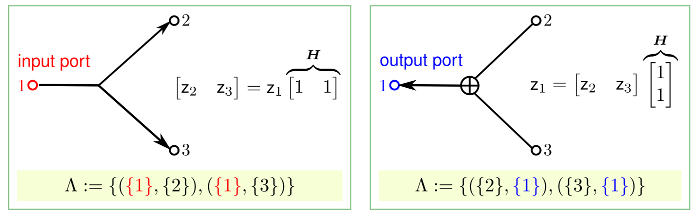
In graph theory, Menger's theorem states that the size of the minimum edge-cut (min-cut) separating two nodes is equal to the maximum number of edge-disjoint paths between them. The paths can be obtained efficiently by computing the maximum flow (max-flow) using the Ford-Fulkerson algorithm. The theory was applied to information flow in network coding [2], where each edge is an independent channel with unit capacity. Independent information can be routed along different disjoint paths, and this attains the maximum data rate because the min-cut value is equal to the cut-set bound.

Unlike commodity flow, however, information flow naturally calls for a more general network link model. A linear deterministic network was proposed in [3] to capture the broadcast and interference of signals in wireless networks. A layered linking network was proposed in [4] to abstract away the linear algebra by linking systems [5, 6], and reveal the essential combinatorial structure for efficient construction of a routing solution. Unfortunately, these results do not generalize Menger's theorem because the network had to be converted first to a layered model with no cycles by time-expansion [2], and this does not give an efficient time-invariant routing solution. Indeed, the difficulty of extending Menger's theorem directly to cyclic linear network motivated a seemingly different approach in [7] though linear system theory. However, the combinatorial structure fundamental to information flow was not revealed.

The purpose of this work is to extend Menger's theorem to a general cyclic linking network model. This is inspired by a combinatorial notion of flow identified in [8] from the study of cyclic linear networks. The theory applies therein to give an efficient construction of linear time-invariant network codes. More details can be found in [1, 8].

Chung Chan (cchan@inc.cuhk.edu.hk, chungc@alum.mit.edu) is with the Institute of Network Coding, the Chinese University of Hong Kong.

This work was supported by a grant from the University Grants Committee of the Hong Kong Special Administrative Region, China (Project No. AoE/E-02/08). The author would like to thank Kenneth Shum, Qifu Sun and Raymond Yeung for helpful comments and discussions.



(a) Broadcast link (b) Interference link

Fig. 1. Examples of linked pairs

II. PRELIMINARIES

We first review the basic ideas of linking systems in [5, 6].

Definition 1 (Linking system) A *linking system* is a triple (X, Y, Λ) where X and Y are finite ground sets and $\emptyset \neq \Lambda \subseteq 2^X \times 2^Y$ is a set of *linked pairs* satisfying

$$\forall\, (X', Y') \in \Lambda : \quad |X'| = |Y'|, \tag{1a}$$
$$\forall\, (X', Y') \in \Lambda,\ X'' \subseteq X',\ \exists Y'' \subseteq Y' : (X'', Y'') \in \Lambda, \tag{1b}$$
$$\forall\, (X', Y') \in \Lambda,\ Y'' \subseteq Y',\ \exists X'' \subseteq X' : (X'', Y'') \in \Lambda, \tag{1c}$$
$$\forall\, (X'_1, Y'_1), (X'_2, Y'_2) \in \Lambda,\ \exists (X', Y') \in \Lambda :$$
$$X'_1 \subseteq X' \subseteq X'_1 \cup X'_2 \quad \text{and} \quad Y'_2 \subseteq Y' \subseteq Y'_1 \cup Y'_2. \tag{1d}$$

The *linking function* $\lambda : 2^X \times 2^Y \mapsto \mathbb{N}$ is defined as

$$\lambda(P, Q) := \max\{|X'| : (X', Y') \in \Lambda \cap (2^P \times 2^Q)\} \tag{2}$$

for $P \subseteq X$ and $Q \subseteq Y$. □

The linking system is a generalization of the matrix as follows.

Example 1 (Non-singular submatrices) Given a matrix $\boldsymbol{H}$, let X and Y be the sets of row and column indices respectively, and Λ be the set of $(X', Y') \in 2^X \times 2^Y$ such that the submatrix $\boldsymbol{H}[X', Y']$ with rows and columns indexed by X' and Y' respectively is non-singular, i.e. $\det(\boldsymbol{H}) \neq 0$ and $|X'| = |Y'|$. Then, it can be shown that (X, Y, Λ) is a linking system with the linking function $\lambda(P, Q) = \operatorname{rank}(\boldsymbol{H}[P, Q])$. *It is said to be representable by* $\boldsymbol{H}$. Some examples are shown in Fig. 1. □

Example 2 (Matchings in bipartite graph) Consider a matrix $\boldsymbol{G}$ in which the non-zero entries are distinct indeterminates, i.e. algebraically independent. Let $G = (X, Y, E)$ be a bipartite graph with edge set $E \subseteq X \times Y$ such that $(x, y) \in E$ iff the corresponding entry $\boldsymbol{H}[x, y] \neq 0$. Then, *the linking system for* $\boldsymbol{G}$ *can be represented by the bipartite graph* G because $\det(\boldsymbol{G}[X', Y'])$ is a non-zero polynomial in the indeterminates iff there is a perfect matching from X' to Y' in G [9]. The linking function $\lambda(P, Q)$ is the size of the maximum matching from P to Q. □

λ in the above examples can be computed efficiently from their representations even though the size of Λ can be exponential in $|X|$ and $|Y|$. It is therefore useful to define a linking system from a linking function as follows.

Definition 2 A linking system can be defined alternatively by a linking function $\lambda : X \times Y \mapsto \mathbb{N}$ satisfying

$$\begin{aligned} &\forall\, P'' \subseteq P' \subseteq X,\, Q'' \subseteq Q' \subseteq Y : \\ &\quad \lambda(P'', Q'') \leq \lambda(P', Q') \leq \min\{|P'|, |Q'|\} \qquad \textbf{(3a)} \\ &\forall\, (P_1, Q_1), (P_2, Q_2) \in 2^X \times 2^Y : \\ &\textstyle\sum_i \lambda(P_i, Q_i) \geq \lambda\left(\bigcup_i P_i, \bigcap_i Q_i\right) + \lambda\left(\bigcap_i P_i, \bigcup_i Q_i\right). \qquad \textbf{(3b)} \end{aligned}$$

The set of linked pairs is given by

$$\Lambda := \{(X', Y') \in 2^X \times 2^Y : \lambda(X', Y') = |X'| = |Y'|\}, \quad \textbf{(4)}$$

which can be shown to satisfy (1). □

The linking system can also be defined from the matroid and therefore inherits the structure well-known for polynomial-time algorithms. Like matroids, there are operations on linking systems that return a new linking system as follows.

Definition 3 (Restriction) The *restriction* of a linking system $L := (X, Y, \Lambda)$ to a pair $(\tilde{X}, \tilde{Y})$ of finite ground sets is

$$L[\tilde{X}, \tilde{Y}] := (\tilde{X}, \tilde{Y}, \Lambda \cap (2^{\tilde{X}} \times 2^{\tilde{Y}})). \quad \textbf{(5)}$$

It is a linking system with the linking function equal to

$$\lambda_{L[\tilde{X}, \tilde{Y}]}(P, Q) := \lambda(P \cap X, Q \cap Y) \quad \textbf{(6)}$$

for $P \subseteq \tilde{X}$ and $Q \subseteq \tilde{Y}$. n.b. $\tilde{X}$ and $\tilde{Y}$ may contain elements outside X and Y respectively for the *trivial extension*. □

Definition 4 (Union) The *union* of the linking systems $L_i := (X_i, Y_i, \Lambda_i)$ for $i = 1, 2$ is $L_1 \vee L_2 := (\bigcup_i X_i, \bigcup_i Y_i, \Lambda')$ with

$$\Lambda' := \{(\textstyle\bigcup_i X'_i, \bigcup_i Y'_i) : (X'_i, Y'_i) \in \Lambda_i, \emptyset = \bigcap_i X'_i = \bigcap_i Y'_i\}. \textbf{(7)}$$

It is a linking system with the linking function equal to

$$\begin{aligned} \lambda_{L_1 \vee L_2}(P, Q) := \min_{\hat{X} \subseteq P, \hat{Y} \subseteq Q} \sum_i \lambda_i(\hat{X} \cap X_i, \hat{Y} \cap Y_i) \\ + |X \setminus \hat{X}| + |Y \setminus \hat{Y}| \quad \textbf{(8)} \end{aligned}$$

where λ_i is the linking function of L_i [9]. □

A product operation is also defined in [5] and used to derive the max-flow min-cut result for layered linking networks [4]. However, for cyclic networks without the layered structure, the technique does not apply and so we need new concepts.

III. Network Link Model

Definition 5 (Linking network) A linking network is a linking system $L := (Z, Z, \Lambda)$, or simply (Z, Λ), where Z is called the set of *ports* and Λ characterizes the interconnection. □

In contrast with the matroidal undirected network in [10], linking network is directed in the sense that $(X', Y') \in \Lambda$ may not imply $(Y', X') \in \Lambda$. Fig. 1 gives some examples of the linking network with $Z := \{1, 2, 3\}$. They are also examples of the acyclic and layered networks defined below.

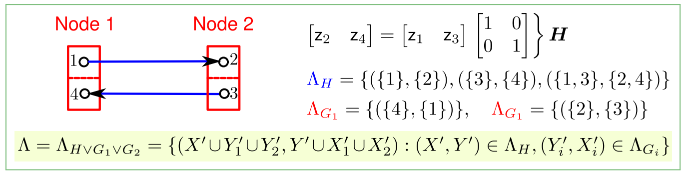


Fig. 2. Cycle

Definition 6 A linking network (Z, Λ) is *acyclic* if Z can be ordered as $(p_i : 1 \leq i \leq |Z|)$ such that $(\{p_k\}, \{p_j\}) \notin \Lambda$ if $j \leq k$. In particular, it is *layered* if Z can be partitioned as $(Z_i : 1 \leq i \leq l)$ such that the network is a union of some linking systems $L_i := (Z_i, Z_{i+1}, \Lambda_i)$ for $1 \leq i < l$. □

This covers the layered linking network in [4]. For the examples in Fig. 1, ports 2 and 3 belong to one layer and port 1 belongs to another. A cyclic network is shown in Fig. 2 and an acyclic but non-layered network is given in Fig. 4(a).

To apply the linking network model to information flow over channels, we consider an input-output model where ports are partitioned into inputs and outputs of a finite set V of nodes. e.g. port 1 can be regarded as an input port in Fig. 1(a) and an output port in Fig. 1(b). More generally, each node $i \in V$ has a finite set X_i of input ports and a finite set Y_i of output ports. There is a channel from the input ports to the output ports modeled by a linking system

$$H := (\textstyle\bigcup_{i \in V} X_i, \bigcup_{i \in V} Y_i, \Lambda_H). \quad \textbf{(9)}$$

Let $G_i := (Y_i, X_i, \Lambda_{G_i})$ be a linking system representable by the complete bipartite graph from Y_i to X_i for $i \in V$, i.e.

$$\Lambda_{G_i} := \{(Y'_i, X'_i) \in 2^{Y_i} \times 2^{X_i} : |Y'_i| = |X'_i|\}. \quad \textbf{(10)}$$

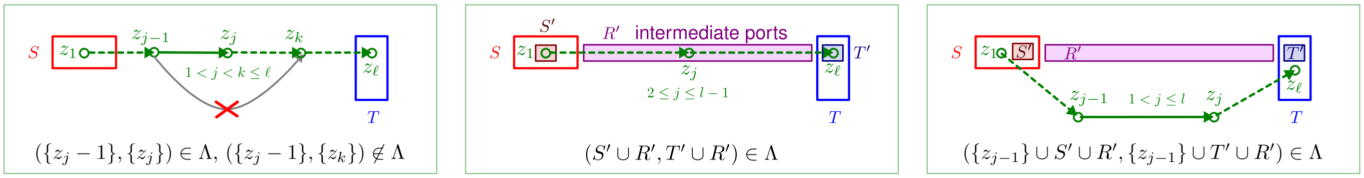
As in [4], this captures how node i may possibly encode its inputs from its outputs since every subset Y'_i of output ports can be matched to any subset of input ports X'_i. The overall input-output linking network is defined as follows.

Definition 7 (IO model) The linking network $L := (Z, \Lambda)$ of the above *input-output (IO) model* is defined as the union

$$L := \left(H \vee \textstyle\bigvee_{i \in V} G_i\right)[Z, Z] \quad \textbf{(11)}$$

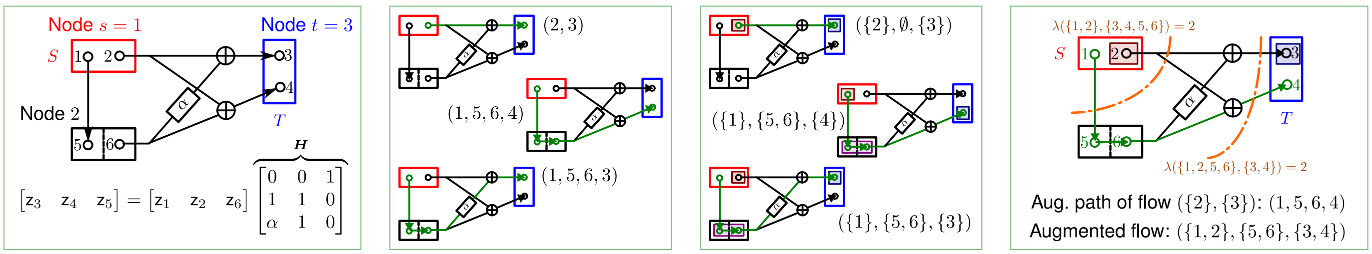
where $Z := \bigcup_{i \in V}(X_i \cup Y_i)$ regarding X_i's and Y_i's as disjoint sets. The network is said to be *linear* if H is representable by a matrix. In particular, it is said to be *graphical* if the matrix can be a permutation matrix, i.e. each $(\{x\}, \{y\}) \in \Lambda$ can be represented by an independent edge in a directed graph. □

The linear linking network covers the linear deterministic network [3, 8], with H in (9) represented by the transfer matrix of the network, and G_i represented by the coding matrix of user $i \in V$. Fig. 2 is an example of a graphical network, where $V = \{1, 2\}$, $X_1 = \{1\}$, $Y_1 = \{4\}$, $X_2 = \{3\}$ and $Y_2 = \{2\}$. H can be represented by the identity matrix with rows and columns indexed by $\{1, 3\}$ and $\{2, 4\}$ respectively. It can also be represented by two independent edges, namely $(1, 2)$ and $(3, 4)$. The linear network in Fig. 4(a) is not graphical because the transfer matrix is not a permutation matrix.



(a) (S,T) path (b) (S,T) flow (c) Disjoint augmenting path of an (S,T) flow

Fig. 3. General notions of paths and flows in linking networks



(a) A linear network: $\alpha \notin \{0,1\}$ (b) (s,t) paths (c) (s,t) flows of size 1 (d) (s,t) flow of size 2

Fig. 4. Paths and flows of a linear linking networks

IV. Shortest Path

A path is defined as a sequence of linked ports as follows.

Definition 8 (Path) Let $L := (Z, \Lambda)$ be a linking network. For disjoint subsets S and T of Z, an (S,T) path is a *minimal* sequence $(z_1, \dots, z_l)$ of $l \geq 2$ ports such that

$$z_1 \in S, \quad z_l \in T \quad \text{and} \quad (\{z_j\}, \{z_{j+1}\}) \in \Lambda \tag{12}$$

for all $j \in \{1, \dots, l-1\}$. □

Minimality of the sequence implies that there is no shortcut around any ports in the path as illustrated in Fig. 3(a).

Proposition 1 *For any (S,T) path $(z_1, \dots, z_l)$,*

$$z_j \notin S \cup T \text{ and } (\{z_{j-1}\}, \{z_k\}) \notin \Lambda \text{ if } 1 < j < k \leq l. \tag{13}$$

In particular, all the ports in a path must be distinct. □

For the input-output model, a path alternates between the input and output ports and visits a sequence of distinct nodes.

Definition 9 Consider the input-output linking network in (11). For $s \in V$, $t \in V \setminus \{s\}$, an (X_s, Y_t) path or simply an (s,t) path is a minimal sequence $(x_1, y_1, \dots, x_l, y_l)$ of $2l \geq 2$ alternating input and output ports with $x_1 \in X_s$, $y_l \in Y_t$, and

$$\forall\, k \in \{1, \dots, l\}, \qquad (\{x_k\}, \{y_k\}) \in \Lambda_H, \qquad \text{and}$$
$$\forall\, j \in \{1, \dots, l-1\}, \;\; \exists i \in V, \quad y_j \in Y_i, x_{j+1} \in X_i.$$

We say node i is visited at step j if $y_j \in Y_i$ or $x_{j+1} \in X_i$. □

Minimality implies that a path is acyclic as usual.

Proposition 2 *A path visits each node in V at most once.* □

For graphical networks, since every $(\{x_j\}, \{y_j\}) \in \Lambda_H$ is an edge, an (s,t) path corresponds to a sequence of edges going from s to t through a sequence of distinct intermediate nodes. For linear networks, a path supports a unit of information flow

Function SP(L, S, T)
% return a shortest (S,T) path of L or NULL if none exists.
Initialization:
  Set $Q \leftarrow$ Ports(L), $\lambda \leftarrow$ LinkingFunction(L).
  For $z \in S$, **set** $d(z) \leftarrow 1$ and $p(z) \leftarrow (z)$.
  For $z \in Q \setminus S$, **set** $d(z) \leftarrow \infty$ and $p(z) \leftarrow ()$.
Loop:
  Set $v \leftarrow \arg\min_{z \in Q} d(z)$ and $Q \leftarrow Q \setminus \{v\}$.
  If $d(v) = \infty$, **return** NULL. % T is not reachable from S.
  If $v \in T$, **return** $p(v)$. % shortest path discovered.
  For $z \in Q$ such that $\lambda(\{v\}, \{z\}) > 0$,
    If $d(z) > d(v) + 1$, % it is shorter to reach z through v.
      set $d(z) \leftarrow d(v) + 1$ and $p(z) \leftarrow p(v) \circ (z)$.
  Goto Loop.

Algorithm 1. Construct a shortest path.

as follows. Consider the network in Fig. 4(b). $(2,3)$ is a path means that node 1 can write a message u to $\mathsf{z}_2 \leftarrow \mathsf{u}$ so that node 3 can observe it from $\mathsf{z}_3 = \mathsf{z}_2 = \mathsf{u}$ with all other input ports unused, i.e. $\mathsf{z}_1, \mathsf{z}_6 \leftarrow 0$. Similarly, $(1,5,6,4)$ is a path means that node 1 can write a message u to $\mathsf{z}_1 \leftarrow \mathsf{u}$ so that node 2 can observe it from $\mathsf{z}_5 = \mathsf{z}_1 = \mathsf{u}$. Node 2 can then write it to $\mathsf{z}_6 \leftarrow \mathsf{z}_5 = \mathsf{u}$ so that node 3 can observe it from $\mathsf{z}_4 = \mathsf{z}_6 = \mathsf{u}$ with the port $\mathsf{z}_2 \leftarrow 0$ unused.

Similar to Dijkstra's algorithm, the shortest path can be computed efficiently by dynamic programming in Algorithm 1. The correctness follows from the usual tail-optimality below.

Proposition 3 *If $(z_1, \dots, z_l)$ is a shortest (S,T) path, then $(z_1, \dots, z_k)$ is a shortest $(S, \{z_k\})$ path for $2 \leq k \leq l$.* □

The condition Algorithm 1 returns NULL can be stated in terms of the min-cut value defined as follows.

Definition 10 (Cut) For linking network $L := (Z, \Lambda)$,

$$\min_{Z'} \lambda(Z', Z \setminus Z') \quad \text{s.t.} \quad Z' \subseteq Z : Z' \supseteq S, Z \setminus Z' \supseteq T \tag{14}$$

is called the minimum (S,T) cut value of L. □

```
Function MF(L,S,T)
% return a maximum (S,T) flow of L.
Initialization:
 Set (S',R',T') ← (∅,∅,∅).
Loop:
 Set f ← (S',R',T'), p ←SP(L_f, S \ S', T \ T').
 If p =NULL, return f.      % no more augmenting paths.
 Set l ←length(p), (z_j : 1 ≤ j ≤ l) ← p.
 Set S'' ← {z_1}, T'' ← {z_l}, R'' ← {z_j : 2 ≤ j ≤ l − 1}.
 Set (S',R',T') ← (S' ∪ S'', R' ∪ R'' \ (R' ∩ R''), T' ∪ T'').
 Goto Loop.
```

Algorithm 2. Construct a maximum flow.

Proposition 4 *There is no (S,T) path over L iff the minimum (S,T) cut value of L is zero.* □

V. Maximum Flow

Inspired by information flow over cyclic linear networks [8], we define a flow in linking networks as follows.

Definition 11 (Flow) An (S,T) flow of $L := (Z,\Lambda)$ is a triple (S',R',T') satisfying

$$S' \subseteq S, \quad T' \subseteq T \quad \text{and} \quad (S' \cup R', T' \cup R') \in \Lambda \tag{15}$$

as illustrated in Fig. 3(b). The size of the flow is $|S'| = |T'|$. □

A path gives rise to a flow of size 1 as expected.

Proposition 5 *An (S,T) path $(z_1,\dots,z_l)$ gives rise to an (S,T) flow $(\{z_1\},\{z_j : 2 \le j \le l-1\},\{z_l\})$ of size 1.* □

A path can also be extracted from a flow of size larger than 1, with SP($L[Z',Z']$,S,T) in Algorithm 1 where $Z' := S' \cup R' \cup T'$.

Proposition 6 *An (S,T) flow (S',R',T') of size at least 1 contains an (S,T) path with all its ports from $S' \cup R' \cup T'$.* □

For the network in Fig. 4(a), each flow of size 1 in Fig. 4(c) corresponds to a path in Fig. 4(b). $(\{1,2\},\{5,6\},\{3,4\})$ is a flow of size 2, and is indeed the max-flow. Similar to the Ford-Fulkerson algorithm, the max-flow can be constructed efficiently in Algorithm 2 by successively finding an augmenting path in the residual network defined below.

Definition 12 (Residual network) The residual network of $L := (Z,\Lambda)$ with respect to f is defined as $L_f := (Z',\Lambda')$ with $Z' := Z \setminus (S' \cup T')$ and $(P,Q) \in \Lambda' \subseteq 2^{Z'} \times 2^{Z'}$ iff

$$(S' \cup R' \cup P \setminus (Q \cap R'), T' \cup R' \cup Q \setminus (P \cap R')) \in \Lambda \tag{16}$$

□

Proposition 7 *L_f is a linking network with linking function equal to $\lambda_{L_f}(P,Q)$ defined as*

$$\begin{aligned}&\lambda(S' \cup R' \cup P \setminus (Q \cap R'), T' \cup R' \cup Q \setminus (P \cap R'))\\ &+ |Q| - |T' \cup R' \cup Q \setminus (P \cap R')|\end{aligned} \tag{17}$$

for $P,Q \subseteq Z'$, where λ is the linking function of L. □

In Algorithm 2, (S'',R'',T'') is a flow of L_f by Proposition 5. It can be argued using (16) that the updated value of (S',R',T') at the end of each loop is a strictly larger (S,T) flow of L. By Proposition 4, the condition Algorithm 2 returns f can be stated in terms of the min-cut value as follows.

Proposition 8 *L_f has no $(S \setminus S', T \setminus T')$ path iff the minimum (S,T) cut value of L is the size of the flow $f := (S',R',T')$.* □

It follows from (1b) and (1c) that the max-flow size is upper bounded by the min-cut value, and so we have the following.

Theorem 1 (Max-flow min-cut) *The maximum size of an (S,T) flow over $L := (Z,\Lambda)$ with linking function λ is equal to the minimum value of an (S,T) cut (14).* □

VI. Disjoint Paths

For the network in Fig. 4(a), $(\{1,2\},\{5,6\},\{3,4\})$ is a flow of size 2. To see it as a union of two disjoint paths as shown in Fig. 4(d), we define the following operation.

Definition 13 (Contraction) The *contraction* of a linking system $L := (X,Y,\Lambda)$ by $(\tilde{X},\tilde{Y}) \in \Lambda$ is defined as $L/(\tilde{X},\tilde{Y}) := (X \setminus \tilde{X}, Y \setminus \tilde{Y}, \Lambda')$ with

$$\Lambda' := \left\{(X',Y') \in 2^{X \setminus \tilde{X}} \times 2^{Y \setminus \tilde{Y}} : (X' \cup \tilde{X}, Y' \cup \tilde{Y}) \in \Lambda\right\}.$$ □

Proposition 9 *$L/(\tilde{X},\tilde{Y})$ defined above is a linking system with its linking function equal to*

$$\lambda_{L/(\tilde{X},\tilde{Y})}(P,Q) := \lambda(P \cup \tilde{X}, Q \cup \tilde{Y}) - |\tilde{X}| \tag{18}$$

for $P \subseteq X \setminus \tilde{X}$ and $Q \subseteq Y \setminus \tilde{Y}$. □

A disjoint path that augments a flow to a larger flow can be defined as follows.

Definition 14 (Disjoint augmenting path) The contraction L/f of a linking network $L := (Z,\Lambda)$ by an (S,T) flow $f := (S',R',T')$ is defined as

$$L/(X',Y')[Z',Z'] \text{ with } \begin{cases}(X',Y') = (S' \cup R', T' \cup R')\\ Z' = S' \cup R' \cup T').\end{cases} \tag{19}$$

A disjoint augmenting path of f is an $(S \setminus S', T \setminus T')$ path of L/f. More explicitly, it is a minimal sequence $(z_1,\dots,z_l)$ of $l \ge 2$ ports from Z' such that

$$z_1 \in S \setminus S',\ z_l \in T \setminus T',\ (\{z_j\} \cup X', \{z_{j+1}\} \cup Y') \in \Lambda \tag{20}$$

for all $j \in \{1,\dots,l-1\}$. This is illustrated in Fig. 3(c). □

The contracted network is indeed a restriction of the residual network in Definition 12 to the ports not in the flow.

Proposition 10 *$L/f = L_f[Z',Z']$ where $f := (S',R',T')$ is a flow of L and $Z' := Z/(S' \cup R' \cup T')$.* □

Thus, a path in L/f is a path in L_f that augments f.

Proposition 11 *For an augmenting path $(z_1,\dots,z_l)$ of an (S,T) flow (S',R',T'),*

$$(\{z_1\} \cup S', \{z_j : 2 \le j \le l-1\} \cup R', \{z_l\} \cup T')$$

is a larger (S,T) flow augmented with the path. □

Function PD(L, f, S, T)
% return a maximal sequence of disjoint augmenting paths
% contained in the (S,T) flow f of L.
Initialization:
Set $(S',R',T') \leftarrow f$, $Z' \leftarrow S' \cup R' \cup T'$, $L' \leftarrow L[Z', Z']$.
Set $j \leftarrow 0$, $f_j \leftarrow (S_j, R_j, T_j) \leftarrow (\emptyset, \emptyset, \emptyset)$.
Loop:
If $j = |S'|$, **return** $(p_{j'} : 1 \leq j' \leq j)$. % no more paths.
Set $j \leftarrow j+1$, $p_j \leftarrow$SP$(L'/f_{j-1}, S \setminus S_{j-1}, T \setminus T_{j-1})$.
Set $l \leftarrow$length(p_j), $(z_{j'} : 1 \leq j' \leq l) \leftarrow p_j$.
Set $S'' \leftarrow \{z_1\}$, $T'' \leftarrow \{z_l\}$, $R'' \leftarrow \{z_{j'} : 2 \leq j' \leq l-1\}$.
Set $f_j \leftarrow (S_j, R_j, T_j) \leftarrow (S_{j-1} \cup S'', R_{j-1} \cup R'', T_{j-1} \cup T'')$.
Goto Loop.

Algorithm 3. Decompose a flow into disjoint augmenting paths.

e.g., in Fig. 4(d), the path $(1,5,6,4)$ augments the flow $(\{2\}, \emptyset, \{3\})$ to the larger flow $(\{1,2\}, \{5,6\}, \{3,4\})$. It satisfies (20) because Λ contains $(\{1,2\},\{5,3\})$, $(\{5,2\},\{6,3\})$ and $(\{6,2\},\{4,3\})$. It supports an additional unit of information flow as follows. Node 1 write the first message u_1 to $\mathsf{z}_2 \leftarrow \mathsf{u}_1$ and the second message to $\mathsf{z}_1 = \mathsf{u}_2$. Node 2 observes $\mathsf{z}_5 = \mathsf{z}_1 = \mathsf{u}_2$ and write it to $\mathsf{z}_6 \leftarrow \mathsf{z}_5 = \mathsf{u}_2$. Node 3 observes

$$\begin{bmatrix} \mathsf{z}_3 & \mathsf{z}_4 \end{bmatrix} = \begin{bmatrix} \mathsf{z}_2 & \mathsf{z}_6 \end{bmatrix} \boldsymbol{H}[\{2,6\},\{3,4\}] = \begin{bmatrix} \mathsf{u}_1 & \mathsf{u}_2 \end{bmatrix} \begin{bmatrix} 1 & \alpha \\ 1 & 1 \end{bmatrix}.$$

Both messages can be recovered because $(\{6,2\},\{4,3\}) \in \Lambda$, i.e. $\boldsymbol{H}[\{2,6\},\{3,4\}]$ is invertible.

Algorithm 3 decomposes an (S,T) flow $f := (S',R',T')$ of L into a sequence of disjoint augmenting paths illustrated in Fig. 5(a). $f_{j-1} := (S_{j-1}, R_{j-1}, T_{j-1})$ is a subflow of f, i.e. $(S_{j-1}, R_{j-1}, T_{j-1}) \in 2^{S'} \times 2^{R'} \times 2^{T'}$ is a flow of L. It follows from definition (20) that $(S' \setminus S_{j-1}, R' \setminus R_{j-1}, T' \setminus T_{j-1})$ is an $(S \setminus S_{j-1}, T \setminus T_{j-1})$ flow of $L \setminus f_{j-1}$, and so p_j is a disjoint augmenting path of f_{j-1}, for j up to size $|S'|$ of f.

Theorem 2 (Menger-type) *An (S,T) flow (S',R',T') contains a sequence $(p_j : 1 \leq j \leq k)$ of $k = |S'|$ augmenting paths, where p_1 is an (S,T) path and p_j for $1 < j \leq k$ is an augmenting path of the (S,T) flow composed of the previous paths $p_{j'}$ for $1 \leq j' \leq j-1$.* □

The above theorem generalizes Menger's theorem seamlessly. e.g., the gammoid linking system defined in [5] by disjoint paths over graphs is covered as a special case of the following.

Theorem 3 *The set of (S',T') such that (S',R',T') is an (S,T) flow for some R' forms a linking system.* □

VII. Max-flow Min-cut Theorem for IO Model

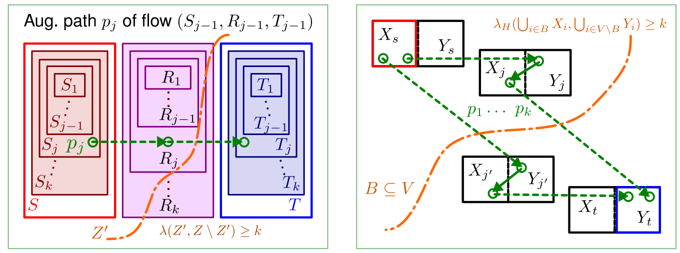
For the IO model in (11), the min-cut expression can be further simplified as follows and illustrated in Fig. 5(b).

Theorem 4 *The maximum size of an (s,t) flow in (11) is*

$$\min_{B} \lambda_H \left(\bigcup_{i \in B} X_i, \bigcup_{i \in V \setminus B} Y_i \right) \text{ s.t. } B \subseteq V : s \in B \not\ni t, \quad \textbf{(21)}$$

which is called the minimum value of an (s,t) cut. □

e.g., in Fig. 4(d), the min-cut value is equal to 2, which is the size of the max-flow. The simplified min-cut expression (21) can be derived from (14) by the following result.



(a) Linking network (b) IO model

Fig. 5. Menger-type and max-flow min-cut theorem

Lemma 1 *Consider a linking network $L := (Z, \Lambda)$, disjoint sets S and T of Z, and a linking system G representable by a complete bipartite graph from $\hat{Y} \subseteq Z \setminus S$ to $\hat{X} \subseteq Z \setminus T$. If $L = L' \vee G$ for some linking system $L' = (Z \setminus \hat{Y}, Z \setminus \hat{X}, \Lambda')$, then the min-cut value (14) is not increased by imposing the additional constraint on Z' that*

$$Z' \overset{(a)}{\supseteq} (\hat{X} \cup \hat{Y}) \setminus T \quad \text{or} \quad Z \setminus Z' \overset{(b)}{\supseteq} (\hat{X} \cup \hat{Y}) \setminus S. \qquad \textbf{(22)}$$

Furthermore, (a) can be imposed if $S \supseteq \hat{X}$ while (b) can be imposed if $T \supseteq \hat{Y}$. □

This can be proved using (8). For the IO model in (11), we can apply the above result with $S = X_s$, $T = Y_t$, $G := G_i$ repeatedly for each $i \in V$. It follows that Z' can contain Y_s and exclude X_t entirely without increasing the min-cut value in (14). For $i \in V \setminus \{s,t\}$, it is also admissible to have Z' either include or exclude $X_i \cup Y_i$, which gives (21).

VIII. Conclusion

We have formulated a cyclic linking network model that extends Menger's theorem with the general notions of paths and flows. Efficient algorithms are given to construct the shortest path, maximum flow and optimal path decomposition. The new concepts developed for linking systems may potentially apply to other problems on information flow.

References

[1] C. Chan, publications. http://goo.gl/4YZLT.
[2] R. Ahlswede, N. Cai, S.-Y. Li, and R. Yeung, "Network information flow," *IEEE Transactions on Information Theory*, vol. 46, no. 4, pp. 1204 –1216, Jul. 2000.
[3] A. Avestimehr, S. Diggavi, and D. Tse, "Wireless network information flow: A deterministic approach," *IEEE Transactions on Information Theory*, vol. 57, no. 4, pp. 1872–1905, 2011.
[4] M. Goemans, S. Iwata, and R. Zenklusen, "An algorithmic framework for wireless information flow," in *47th Annual Allerton Conference on Communication, Control, and Computing*, Sep. 2009, pp. 294–300.
[5] A. Schrijver, "Matroids and linking systems," *Journal of Combinatorial Theory, Series B*, vol. 26, no. 3, pp. 349 – 369, 1979.
[6] J. P. Kung, "Bimatroids and invariants," *Advances in Mathematics*, vol. 30, no. 3, pp. 238 – 249, 1978.
[7] S. Y. Park and A. Sahai, "An algebraic mincut-maxflow theorem," in *2011 IEEE International Symposium on Information Theory Proceedings (ISIT)*, Aug. 2011, pp. 608–612.
[8] C. Chan, K. W. Shum, and Q. T. Sun, "Combinatorial flow over cyclic linear network," see [1].
[9] K. Murota, *Matrices and Matroids for Systems Analysis*, ser. Algorithms and Combinatorics. Dordrecht: Springer, 2010.
[10] C. Chan, "Matroidal undirected network," in *2012 IEEE International Symposium on Information Theory Proceedings (ISIT)*, Cambridge, MA, Jul. 2012.