Complexity of Linear Connectivity Problems in Directed Hypergraphs*

Mayur Thakur[1***] and Rahul Tripathi[2]

[1] Dept. of Computer Science, University of Missouri–Rolla, Rolla, MO 65409, USA.
thakurk@umr.edu
[2] Dept. of Computer Science, University of Rochester, Rochester, NY 14627, USA.
rahult@cs.rochester.edu

Abstract. We introduce a notion of linear hyperconnection (formally denoted L-*hyperpath*) between nodes in a directed hypergraph and relate this notion to existing notions of hyperpaths in directed hypergraphs. We observe that many interesting questions in problem domains such as secret transfer protocols, routing in packet filtered networks, and propositional satisfiability are basically questions about existence of L-hyperpaths or about cyclomatic number of directed hypergraphs w.r.t. L-hypercycles (the minimum number of hyperedges that need to be deleted to make a directed hypergraph free of L-hypercycles). We prove that the L-hyperpath existence problem, the cyclomatic number problem, the minimum cyclomatic set problem, and the minimal cyclomatic set problem are each complete for a different level (respectively, NP, Σ_2^p, Π_2^p, and DP) of the polynomial hierarchy.

1 Introduction

Roughly speaking, a directed hypergraph is a generalization of directed graphs in which each directed hyperedge is allowed to have multiple source (tail) nodes and multiple destination (head) nodes. Thus, a (simple) directed edge is a hyperedge with exactly one tail node and exactly one head node. Directed hypergraphs have been used to model a wide variety of problems in propositional logic [6, 20], relational databases [3, 11, 28], urban transportation planning [10, 18], chemical reaction mechanisms [26, 30], Petri nets [2, 23], operations research [10], and probabilistic parsing [16]. They have been introduced under different names such as "And-Or graphs" and "FD-graphs."

Ausiello, D'Atri, and Saccá [3] introduced the notion of directed hypergraphs, though they called them "FD-graphs," where FD stands for "functional dependency," and used them to represent dependencies among attributes in relational databases. They presented efficient algorithms for several problems related to transitive closure, minimization, etc., of attributes in a database. Gallo et al. [10]

*** Work done in part while affiliated with the Department of Computer Science at the University of Rochester, Rochester, NY 14627, USA.

* Supported in part by grants NSF-INT-9815095 and NSF-CCF-0426761.

first formalized the basic notions related to directed hypergraphs like connectivity, paths, and cuts, and showed applications of directed hypergraphs to problems such as functional dependency in relational database theory and route planning in urban transport system design. More recently, a unified view of deterministic and probabilistic parsing has been obtained by showing them to be equivalent to traversals in directed hypergraphs [16].

The notion of *connection* in a directed graph is simple and can be recursively defined as follows. Each node is connected to itself and a node x is connected to another node y, if there exists a node z such that there is a connection from node x to node z and there is an edge from node z to node y. But there does not seem to be one common intuitive notion for a *hyperconnection* (i.e., connection in directed hypergraphs). In fact, different notions of hyperpaths and hypercycles in directed hypergraphs have been defined in the literature [7, 8, 10, 19, 30] based on varying intuitive notions of hyperconnection in problem domains.

In this paper, first we review these notions of hyperpaths and hypercycles in directed hypergraphs. We describe our notion of linear hyperconnection, and define L-hyperpath and L-hypercycle. We identify domains such as secret transfer protocols, packet filtered networks, and propositional satisfiability, where the notion of linear hyperconnection can be used to capture key problems. We study the complexity of basic computational problems in directed hypergraphs: finding whether hyperpaths of certain a type exist between two given nodes, finding whether hyperpaths of a certain type with a given bound on certain measure exist between two given nodes, etc.

The cyclomatic number of a (simple) graph is the minimum number of edges that need to be removed to make the graph acyclic. Intuitively speaking, the cyclomatic number of a graph measures the degree of cyclicity of a graph. We can define the cyclomatic number of a directed hypergraph analogously to that in simple graphs. That is, the cyclomatic number of a directed hypergraph w.r.t. L-hypercycles is the minimum number of hyperedges that need to be removed such that the resulting hypergraph does not contain any L-hypercycle. We study the computational complexity of problems related to cyclomatic number of directed hypergraphs w.r.t. L-hypercycles: computing the cyclomatic number of a directed hypergraph, finding whether a given set of hyperedges forms a cyclomatic set, finding whether these hyperedges form a minimal cyclomatic set, etc. We prove that many of these fundamental problems are computationally hard (under standard complexity-theoretic assumptions). We prove that these problems are each complete for a different level of the polynomial hierarchy (see Definition 7 for precise definitions of these problems): CYCLOMATIC-NUMBER is Σ_2^p-complete, MIN-CYCLOMATIC-SET is Π_2^p-complete, and MINIMAL-CYCLOMATIC-SET is DP-complete.

It is interesting to compare the complexity of connectivity problems on directed graphs and directed hypergraphs. In particular, consider the complexity of the following *path existence* problems:

- Is there a path between two given vertices in a directed graph? This problem is known to be NL-complete.

- Is there a B-hyperpath between two given vertices in a directed hypergraph? This problem is in P [10].
- Is there an L-hyperpath between two given vertices in a directed hypergraph? In this paper, we show that this problem is NP-complete.

Compare also the complexity of the following *cyclomatic number* problems:

- Is there a set of k edges whose deletion makes a directed graph free of cycles? This problem is NP-complete [14].
- Is there a set of k hyperedges whose deletion makes a directed hypergraph free of L-hypercycles? In this paper, we show that this problem is Σ_2^p-complete.

(Due to space limitations, proofs of most results have been omitted. Please see the full version [27] for omitted proofs and for details of how L-hyperpaths in directed hypergraphs can be used to model practical problems.)

2 Notions of Hyperpaths in Directed Hypergraphs

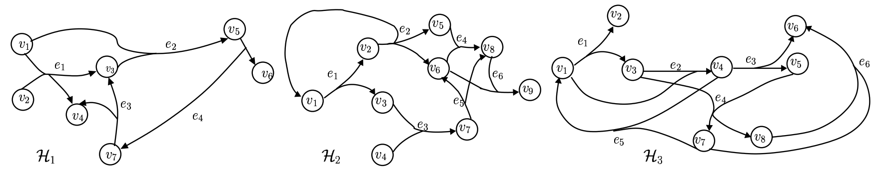
Definition 1 ([10]). *A* directed hypergraph $\mathcal{H}$ *is a tuple* (V, E)*, where* V *is a finite set and* $E \subseteq 2^V \times 2^V$ *such that, for every* $e = (T(e), H(e)) \in E$, $T(e) \neq \emptyset$, $H(e) \neq \emptyset$, *and* $T(e) \cap H(e) = \emptyset$. *For every integer* $k \geq 1$, *a* k-directed hypergraph $\mathcal{H}$ *is a directed hypergraph in which, for every* $e \in E(\mathcal{H})$, $|T(e)| \leq k$ *and* $|H(e)| \leq k$.

A B-hyperedge *(*F-hyperedge*) is a hyperedge* $e = (T(e), H(e))$ *such that* $|H(e)| = 1$ *(respectively,* $|T(e)| = 1$*). A* B-hypergraph *(*F-hypergraph*) is a hypergraph* $\mathcal{H}$ *such that each hyperedge in* $\mathcal{H}$ *is a B-hyperedge (respectively, F-hyperedge). A directed hypergraph* $\mathcal{H}' = (V', E')$ *is a* subhypergraph *of* $\mathcal{H}$ *if* $V' \subseteq V$, $E' \subseteq E$.

Let $e = (T(e), H(e))$ be a hyperedge in some directed hypergraph $\mathcal{H}$. Then, $T(e)$ is known as the *tail* of e and $H(e)$ is known as the *head* of e. The size of representing a directed hypergraph $\mathcal{H}$ is taken to be $|V(\mathcal{H})| + |E(\mathcal{H})|$ unless another representation scheme is explicitly mentioned as in, for example, Section 6. Given a directed hypergraph $\mathcal{H} = (V, E)$, its symmetric image $\overline{\mathcal{H}}$ is a directed hypergraph defined as follows: $V(\overline{\mathcal{H}}) = V(\mathcal{H})$ and $E(\overline{\mathcal{H}}) = \{(H, T) \mid (T, H) \in E(\mathcal{H})\}$.

Definition 2 ([10]). *Let* $\mathcal{H} = (V, E)$ *be a directed hypergraph.*

1. *A* simple path Π_{st} *from* $s \in V(\mathcal{H})$ *to* $t \in V(\mathcal{H})$ *in* $\mathcal{H}$ *is a sequence* $(v_1, e_1, v_2, e_2, \ldots, v_k, e_k, v_{k+1})$ *consisting of distinct vertices and hyperedges such that* $s = v_1$, $t = v_{k+1}$, *and for every* $1 \leq i \leq k$, $v_i \in T(e_i)$ *and* $v_{i+1} \in H(e_i)$. *If, in addition,* $t \in T(e_1)$ *then* Π_{st} *is a* simple cycle. *A simple path is* cycle-free *if it does not contain any subpath that is a simple cycle.*
2. *A* B*-hyperpath from* $s \in V(\mathcal{H})$ *to* $t \in V(\mathcal{H})$ *in* $\mathcal{H}$ *can be defined in terms of a notion of* B-connection *in directed hypergraphs. The recursive definition of* B*-connection to a node* s *is as follows: (i) a node* s *is* B*-connected to itself, and (ii) if there is a hyperedge* e *such that all the nodes in* $T(e)$ *are* B*-connected to* s*, then every node in* $H(e)$ *is* B*-connected to* s*. A* B-hyperpath



(a) (b) (c)

Fig. 1. (a) $\Pi_{v_1v_4} = (v_1, e_2, v_5, e_4, v_7, e_3, v_4)$ is a simple path and $\Pi_{v_1v_3} = (v_1, e_2, v_5, e_4, v_7, e_3, v_3)$ is a simple cycle in $\mathcal{H}_1$. (b) Directed hypergraph $\mathcal{G}_1$ with $V(\mathcal{G}_1) = \{v_1, v_2, v_3, v_5, v_6, v_8, v_9\}$ and $E(\mathcal{G}_1) = \{e_1, e_2, e_4, e_6\}$ is a B-hyperpath from v_1 to v_9 in $\mathcal{H}_2$, directed hypergraph $\mathcal{G}_2$ with $V(\mathcal{G}_2) = \{v_3, v_4, v_6, v_7, v_8, v_9\}$ and $E(\mathcal{G}_2) = \{e_3, e_5, e_6\}$ is an F-hyperpath from v_3 to v_9 in $\mathcal{H}_2$, and directed hypergraph $\mathcal{G}_3$ with $V(\mathcal{G}_3) = \{v_6, v_7, v_8, v_9\}$ and $E(\mathcal{G}_3) = \{e_5, e_6\}$ is a BF-hyperpath from v_7 to v_9 in $\mathcal{H}_2$. (c) A directed hypergraph $\mathcal{H}_3$ with an L-hypercycle. In all the figures, arrows on a hyperedge point to the vertices in the head of the hyperedge.

from s to t in $\mathcal{H}$ is a minimal (with respect to deletion of vertices and hyperedges) subhypergraph of $\mathcal{H}$ where t is B-connected to s. (See also [7, 8] and [5] for equivalent, but alternative, characterizations of B-hyperpaths.)

3. *An F-*hyperpath *Π_{st} from $s \in V(\mathcal{H})$ to $t \in V(\mathcal{H})$ in $\mathcal{H}$ is a subhypergraph of $\mathcal{H}$ such that $\overline{\Pi}_{st}$ is a B-hyperpath from t to s in $\overline{\mathcal{H}}$.*
4. *A BF-*hyperpath *is a hypergraph that is both a B-hyperpath and an F-hyperpath.*

Even though the notions of B- and F-hyperpath capture problems in several different problem domains (see, e.g., [10, 12]), there are other problem domains for which these definitions do not seem to be the right one. We mention three such problem domains in Section 3.

3 L-Hyperpath

3.1 Definition and Relationship with Other Notions of Hyperpaths

The notions of hyperpaths (B-, F-, and BF-hyperpaths) defined by Gallo et al. [10] differ from the notion of a (directed) path in a (simple) directed graph in that, roughly speaking, the hyperpaths are not required to be "linear." By that we mean that while a path in a directed graph is an alternating sequence of vertices and edges, a (B-, F-, or BF-) hyperpath may not have this form. Although the definition of a simple path (Definition 2, part 1) requires linearity, that definition is too weak to capture the expressiveness of directed hypergraphs in the following sense: Given any directed hypergraph $\mathcal{H}$ and $u, v \in V(\mathcal{H})$, there is a simple path from u to v in $\mathcal{H}$ if and only if there is a simple path from u to v in a directed graph G with $V(G) = V(\mathcal{H})$ and $E(G) = \{(u, v) \mid (\exists e \in E(\mathcal{H}))[u \in T(e) \text{ and } v \in H(e)]\}$.

In this section, we introduce a notion of linear hyperpath, called L-hyperpath, and relate this notion of L-hyperpath with previously studied notions of directed hyperpaths.

Definition 3. *An L-hyperpath Π_{st} from s to t in a directed hypergraph $\mathcal{H} = (V, E)$ is a sequence $(v_1, e_1, v_2, e_2, \ldots, v_k, e_k, v_{k+1})$ consisting of distinct vertices and hyperedges such that $s = v_1$, $t = v_{k+1}$, for every $1 \le i \le k$, $v_i \in T(e_i)$ and $v_{i+1} \in H(e_i)$, and for every $1 \le i \le k$, $T(e_i) \subseteq \{s\} \cup H(e_1) \cup \ldots \cup H(e_{i-1})$. If, in addition, $t \in T(e_1)(= \{s\})$ then Π_{st} is an L-hypercycle in $\mathcal{H}$.*

L-hyperpaths inherit the linearity property from simple paths and the restricted B-connection property from B-hyperpaths. L-hyperpaths may alternatively be expressed in terms of directed hypergraphs as follows. For any L-hyperpath $\Pi = (v_1, e_1, v_2, e_2, \ldots, v_k, e_k, v_{k+1})$, let $\mathcal{H}_\Pi$ be defined as the subhypergraph of $\mathcal{H}$ such that, $V(\mathcal{H}_\Pi) = \{v_1\} \cup H(e_1) \cup H(e_2) \cup \ldots \cup H(e_k)$ and $E(\mathcal{H}_\Pi) = \{e_1, e_2, \ldots, e_k\}$. We say that $\mathcal{H}_\Pi$ is the *hypergraph representation* of Π. In Figure 1(c), $\Pi_1 = (v_1, e_1, v_3, e_2, v_4, e_3, v_5, e_4, v_7)$ is an L-hyperpath and $\Pi_2 = (v_1, e_1, v_3, e_2, v_4, e_3, v_5, e_4, v_7, e_5, v_1)$ is an L-hypercycle in $\mathcal{H}_3$. Also, note that there is no L-hyperpath from v_4 to v_1 in $\mathcal{H}_3$ and that the hypergraph representation of Π_1 is $\mathcal{H}' = (V(\mathcal{H}_3), \{e_1, e_2, e_3, e_4\})$.

Theorem 4. *Let $\mathcal{H}$ be a B-hypergraph, $\mathcal{G}$ be a subhypergraph of $\mathcal{H}$, and $s, t \in V(\mathcal{G})$. Then, the following holds: $\mathcal{G}$ is the hypergraph representation of an L-hyperpath Π_{st} from s to t if and only if $\mathcal{G}$ is a minimal (w.r.t. deletion of vertices and hyperedges) subhypergraph of $\mathcal{H}$ such that t is B-connected to s and there is a simple cycle-free path from s to t that consists of all the hyperedges of $\mathcal{G}$.*

The study of L-hyperpaths is interesting from a theoretical point of view because, as argued earlier, the notion of L-hyperpaths is a restriction of the notion of simple paths and the notion of B-hyperpaths. The study of cyclomatic number of hypergraphs is of fundamental significance (see [9, 1]) and so it is interesting to investigate the complexity of computing the cyclomatic number (in the L-hypercycle notion) of directed hypergraphs. On the practical side, we show in the full version [27] of this paper that many interesting questions in problem domains such as secret transfer protocols, routing in packet filtered networks, and propositional satisfiability can be modeled using the notion of L-hyperconnection. The linearity constraint of L-hyperpaths turns out to be crucial in correctly modeling problems in these domains.

4 Computational Problems on Directed Hyperpaths

Many applications of graphs require one to associate a cost (or, weight) on the edges of the graph. The cost of a path in a graph is then defined to be the sum of cost of edges in the path. In contrast, since the structure of a hyperpath is more complicated, a number of measures on hyperpaths in a directed hypergraph are defined and studied in the literature [4, 7, 8, 15, 17, 18, 24]. We observe that the measures defined for previously studied notions of directed hyperpaths are applicable also for L-hyperpaths if the hypergraph representation of an L-hyperpath is considered in the definition. This indicates that the notion of L-hyperpaths is robust and it suggests that L-hyperpaths may be used to model a variety of problems that require these measures on hyperpaths.

For any $X \in \{B, F, L\}$ and for any measure function μ_X on X-hyperpaths of $\mathcal{H}$, we define the following decision problems related to directed hypergraphs:

1. X-HYPERPATH $= \{\langle \mathcal{H}, s, t \rangle \mid \mathcal{H}$ is a directed hypergraph that contains an X-hyperpath Π_{st} from s to $t\}$.
2. μ_X-OPT-HYPERPATH $= \{\langle \mathcal{H}, s, t, k \rangle \mid \mathcal{H}$ is a directed weighted hypergraph that contains an X-hyperpath Π_{st} from s to t such that $\mu_X(\Pi_{st}) \leq k\}$.

Gallo et al. [10] showed that both B-HYPERPATH and F-HYPERPATH are solvable in polynomial time. Ausiello et al. [7] and Italiano and Nanni [15] proved that μ_B-OPT-HYPERPATH is NP-complete when μ_B is one of the following measure functions: (a) number of hyperedges, (b) cost, (c) size. Ausiello et al. [7] and Ausiello, Italiano, and Nanni [8] proved that μ_B-OPT-HYPERPATH is solvable in polynomial time when μ_B is the rank of a B-hyperpath (see [7, 8] for the definition of rank). Theorem 5 states our result related to L-hyperpaths.

Theorem 5. *For every $k \geq 2$,* L-HYPERPATH *is* NP*-complete when restricted to k-directed B-hypergraphs.*

It follows from the proof of Theorem 5 that, for every $k \geq 2$ and for any polynomial-time computable measure μ_L on L-hyperpaths, the problems L-HYPERCYCLE (given a directed hypergraph $\mathcal{H}$, does $\mathcal{H}$ contain an L-hypercycle?) and μ_L-OPT-HYPERPATH are NP-complete when restricted to k-directed B-hypergraphs. A simple observation shows that the problems L-HYPERPATH and μ_L-HYPERPATH are in P when restricted to F-hypergraphs. Thus, roughly speaking, the intrinsic hardness of these problems is due to the presence of B-hyperedges. In contrast, we find that the proofs of Theorems 8 and 9 seem to require both B- and F-hyperedges in the construction. It will be interesting to see whether proofs of Theorems 8 and 9 can be carried out without requiring F-hyperedges in the construction.

5 Cyclomatic Number of a Directed Hypergraph

The cyclomatic number of a hypergraph is the minimum number of hyperedges that need to be deleted so that the resulting hypergraph has no hypercycle. For a connected graph $G = (V, E)$, the cyclomatic number is given by $|E| - |V| + 1$. The cyclomatic number of any undirected hypergraph is also efficiently computable [1]. For directed hypergraphs, the notion of cyclomatic number can be defined as follows.

Definition 6. *Given a directed hypergraph $\mathcal{H} = (V, E)$, the* cyclomatic number *of $\mathcal{H}$ with respect to L-hypercycles is the following:* $\min\{k \in \mathbb{N} \mid (\exists B \subseteq E)[|B| = k$ *and there are no L-hypercycles in* $(V, E - B)]\}$.

In this section, we study the complexity of several decision problems related to the abovementioned definition of cyclomatic number of a directed hypergraph.

Definition 7. *1.* CYCLOMATIC-SET $= \{\langle \mathcal{H}, B\rangle \mid \mathcal{H} = (V, E)$ *is a directed hypergraph and* $B \subseteq E$ *such that* $\mathcal{H}' = (V, E - B)$ *has no L-hypercycle* $\}$.
2. CYCLOMATIC-NUMBER $= \{\langle \mathcal{H}, k\rangle \mid \mathcal{H} = (V, E)$ *is a directed hypergraph such that there exists a set* $B \subseteq E$, $|B| \leq k$ *and* $\langle \mathcal{H}, B\rangle \in$ CYCLOMATIC-SET$\}$.
3. MIN-CYCLOMATIC-SET $= \{\langle \mathcal{H}, B\rangle \mid \langle \mathcal{H}, B\rangle \in$ CYCLOMATIC-SET *and, for each* B' *such that* $|B'| < |B|$, $\langle \mathcal{H}, B'\rangle \notin$ CYCLOMATIC-SET$\}$.
4. MINIMAL-CYCLOMATIC-SET $= \{\langle \mathcal{H}, B\rangle \mid \langle \mathcal{H}, B\rangle \in$ CYCLOMATIC-SET *and, for each* B' *such that* $B' \subsetneq B$, $\langle \mathcal{H}, B'\rangle \notin$ CYCLOMATIC-SET$\}$.

Clearly CYCLOMATIC-SET is coNP-complete, since for any directed hypergraph $\mathcal{H}$, $\langle \mathcal{H}\rangle \in$ L-HYPERCYCLE $\iff \langle \mathcal{H}, \emptyset\rangle \notin$ CYCLOMATIC-SET. The completeness results for remaining problems is stated below.

Theorem 8. *For every* $k \geq 2$, CYCLOMATIC-NUMBER *is* Σ_2^p*-complete when restricted to* k*-directed hypergraphs.*

Proof Sketch: It is clear that CYCLOMATIC-NUMBER is in Σ_2^p. We give a polynomial-time many-one reduction σ from $\mathrm{QSAT}_2(F)$, a problem known to be Σ_2^p-complete (see [21]), to CYCLOMATIC-NUMBER. An instance $\langle X, Y, \phi\rangle$ is in $\mathrm{QSAT}_2(F)$ if and only if ϕ is a boolean formula on disjoint sets X and Y of variables and there exists a truth-value assignment α for X such that for all truth-value assignments β for Y, it holds that $\phi(\alpha, \beta) = \text{False}$.

Let $\langle X, Y, \phi\rangle$ be an instance of $\mathrm{QSAT}_2(F)$, where $X = \{x_1, x_2, \ldots, x_m\}$ and $Y = \{y_1, y_2, \ldots, y_n\}$ are disjoint sets of variables. Without loss of generality, we assume that each variable appears in any clause at most once. Let the clauses of ϕ be $\phi_1, \phi_2, \ldots \phi_s$, and, for $i \in \{1, \ldots, s\}$, let p_i denote the number of occurrences of variables from Y in ϕ_i. Also, for each $1 \leq i \leq s$ and $1 \leq j \leq p_i$, we use $y_{v(i,j)}$ to denote the j-th variable in ϕ_i that belongs to Y. For each $i \leq n$, let n_i denote the number of occurrences of y_i (i.e., as y_i or $\overline{y_i}$) in ϕ. On input $\langle X, Y, \phi\rangle$, σ outputs $\langle \mathcal{H}, m\rangle$ where $\mathcal{H}$ is a directed hypergraph whose construction we describe below. The construction of $\mathcal{H}$ uses four kinds of gadgets. These are selector, k-divider for $k \geq 1$, k-chooser for $k \geq 0$, and switch as shown in Figures 2 and 3.

$\mathcal{H}$ consists of (1) s choosers, $C_1, C_2, \ldots, C_s$, where C_i is a p_i-chooser corresponding to clause ϕ_i, $1 \leq i \leq s$, (2) n dividers, $D_1, D_2, \ldots, D_n$, where D_i is an n_i-divider corresponding to variable y_i, $1 \leq i \leq n$, (3) m selectors, $L_1, L_2, \ldots, L_m$, where L_i corresponds to variable x_i, $1 \leq i \leq m$, and (4) $\sum_{i=1}^{s} p_i$ switches, $S_{1,1}, S_{1,2}, \ldots, S_{1,p_1}, S_{2,1}, \ldots, S_{2,p_2}, \ldots, S_{s,p_s}$ where $S_{i,j}$, $1 \leq i \leq s$ and $1 \leq j \leq p_i$, corresponds to the j-th literal in ϕ_i that belongs to Y. Note that if for some i, $p_i = 0$, i.e., if clause ϕ_i does not have any variable from Y, then there is no switch corresponding to clause ϕ_i.

We use $h(k, j)$ to denote the sum of the number of occurrences of y_j in clauses $\phi_1, \ldots, \phi_k$ with $h(0, j) = 0$ for each j. For each switch S_{ℓ_1,ℓ_2}, where $1 \leq \ell_1 \leq s$ and $1 \leq \ell_2 \leq p_{\ell_1}$, let $\mathrm{succ}(S_{\ell_1,\ell_2})$ be the switch succeeding S_{ℓ_1,ℓ_2} in the ordering $S_{1,1}, S_{1,2}, \ldots, S_{1,p_1}, S_{2,1}, \ldots, S_{2,p_2}, \ldots, S_{s,p_s}$ if $\ell_1 \neq s$ and $\ell_2 \neq p_s$, and is undefined if $\ell_1 = s$ and $\ell_2 = p_s$. For each label x and for each gadget y, we use the shorthand $\mathrm{vertex}_{\mathcal{H}}(x, y)$ ($\mathrm{edge}_{\mathcal{H}}(x, y)$) to denote "the vertex labeled x in gadget y in $\mathcal{H}$" (respectively, "the hyperedge labeled x in gadget y in $\mathcal{H}$").

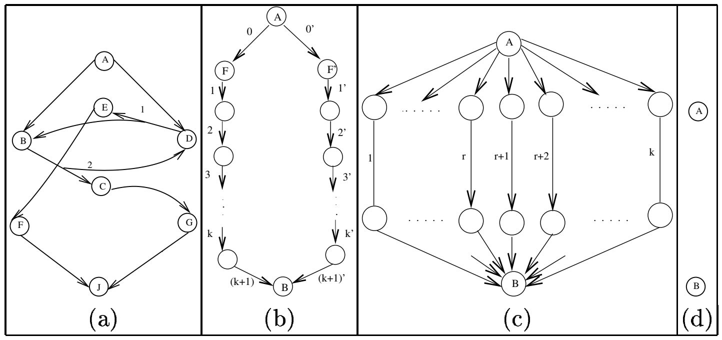


Fig. 2. Gadgets used in the reduction from $\text{QSAT}_2(F)$ to CYCLOMATIC-NUMBER. (a) A selector. (b) A k-divider, where $k \geq 1$. (c) A k-chooser, where $k \geq 1$. (d) A 0-chooser.

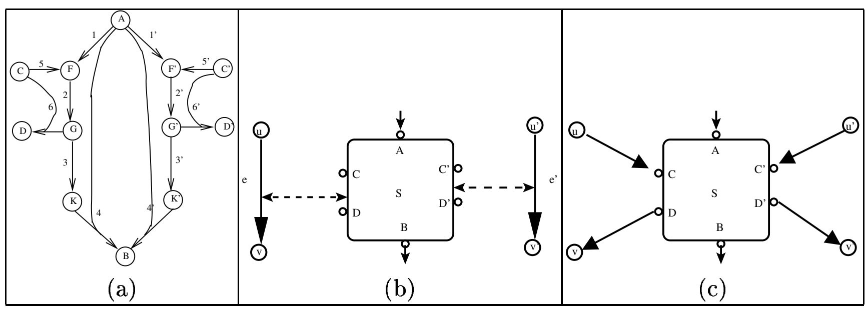


Fig. 3. (a) A switch. (b) Schematic representation of placing a switch S (shown as a rectangular box in the figure) between edges $e = (\{u\}, \{v\})$ and $e' = (\{u'\}, \{v'\})$. (c) The actual placement of a switch S between edges e and e': the edges e and e' are deleted and new edges $(\{u\}, \{C\})$, $(\{D\}, \{v\})$, $(\{u'\}, \{C'\})$ and $(\{D'\}, \{v'\})$ are added.

The vertices of $\mathcal{H}$ consist exactly of the vertices of the above gadgets. The gadgets are connected by hyperedges that are described as follows.

1. (Place a switch between the edge in chooser C_i corresponding to an occurrence of a variable y_j and an edge in divider D_j.) For each $1 \leq i \leq s$, and for each $1 \leq j \leq p_i$, if $y_{v(i,j)}$ occurs as $y_{v(i,j)}$ in ϕ_i, then place switch $S_{i,j}$ between edge j of p_i-chooser C_i and edge $h(i, v(i,j))$ of divider $D_{v(i,j)}$. Otherwise, that is if $y_{v(i,j)}$ occurs as $\overline{y_{v(i,j)}}$ in ϕ_i, then place switch $S_{i,j}$ between edge j of p_i-chooser C_i and edge $h(i, v(i,j))'$ of divider $D_{v(i,j)}$.
2. (Connect the choosers in series.) For each $1 \leq i < s$, connect $\text{vertex}_{\mathcal{H}}(B, C_i)$ to $\text{vertex}_{\mathcal{H}}(A, C_{i+1})$ with a simple directed edge.
3. (Connect the dividers in series.) For each $1 \leq j < n$, connect $\text{vertex}_{\mathcal{H}}(B, D_j)$ to $\text{vertex}_{\mathcal{H}}(A, D_{j+1})$ with a simple directed edge.
4. (Connect the selectors in series.) For each $1 \leq k < m$, connect $\text{vertex}_{\mathcal{H}}(J, L_k)$ to $\text{vertex}_{\mathcal{H}}(A, L_{k+1})$ with a simple directed edge.

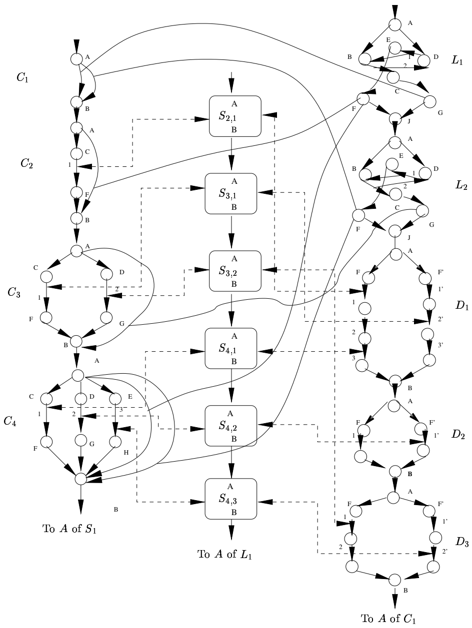


Fig. 4. Hyperedges used to connect gadgets when $\phi = (x_1 \vee \overline{x_2}) \wedge (\overline{x_1} \vee y_1) \wedge (x_2 \vee \overline{y_1} \vee y_3) \wedge (\overline{x_1} \vee \overline{x_2} \vee y_1 \vee \overline{y_2} \vee \overline{y_3})$. C_1, C_2, C_3, and C_4 are 0-chooser, 1-chooser, 2-chooser, and 3-chooser, respectively. D_1, D_2, and D_3 are 3-divider, 1-divider, and 2-divider, respectively. $S_{2,1}, \ldots, S_{4,3}$ are switches and L_1 and L_2 are selectors in $\mathcal{H}$. The hyperedges connecting selectors to choosers are bypass hyperedges.

5. (Connect the switches in series.) For each ℓ_1, ℓ_2, where $1 \le \ell_1 \le s$ and $1 \le \ell_2 \le p_{\ell_1}$, if $\text{succ}(S_{\ell_1,\ell_2})$ is defined then connect $\text{vertex}_{\mathcal{H}}(B, S_{\ell_1,\ell_2})$ to $\text{vertex}_{\mathcal{H}}(A, \text{succ}(S_{\ell_1,\ell_2}))$ with a simple directed edge.
6. (Connect choosers, switches and selectors.) If ϕ does not contain any variable from Y, then connect $\text{vertex}_{\mathcal{H}}(B, C_s)$ to $\text{vertex}_{\mathcal{H}}(A, L_1)$ with a simple directed edge. Otherwise, i.e., if there is at least one clause ϕ_i with an occurrence of a Y variable, do the following. Let S_{ℓ_1,ℓ_2} be the first switch and $S_{\ell'_1,\ell'_2}$ be the last switch (note: switches are ordered in the lexicographic ordering of pairs (i, j) denoting switches $S_{i,j}$) such that $1 \le \ell_1 \le s$, $1 \le \ell'_1 \le s$, $1 \le \ell_2 \le p_{\ell_1}$ and $1 \le \ell'_2 \le p_{\ell'_1}$. Connect $\text{vertex}_{\mathcal{H}}(B, C_s)$ to $\text{vertex}_{\mathcal{H}}(A, S_{\ell_1,\ell_2})$ and connect $\text{vertex}_{\mathcal{H}}(B, S_{\ell'_1,\ell'_2})$ to $\text{vertex}_{\mathcal{H}}(A, L_1)$ with simple directed edges.
7. (Connect dividers and choosers.) Connect $\text{vertex}_{\mathcal{H}}(B, D_n)$ to $\text{vertex}_{\mathcal{H}}(A, C_1)$ with a simple directed edge.
8. (Connect selectors and dividers.) Connect $\text{vertex}_{\mathcal{H}}(J, L_m)$ to $\text{vertex}_{\mathcal{H}}(A, D_1)$ with a simple directed edge.
9. **Bypass hyperedges:** (Connect the selector L_i corresponding to a variable x_i with the chooser C_j.) If a variable x_i occurs as x_i in ϕ_j, then add a hyperedge $(\{\text{vertex}_{\mathcal{H}}(G, L_i), \text{vertex}_{\mathcal{H}}(A, C_j)\}, \{\text{vertex}_{\mathcal{H}}(B, C_j)\})$. Otherwise, if x_i occurs as $\overline{x_i}$ in ϕ_j then add a hyperedge $(\{\text{vertex}_{\mathcal{H}}(F, L_i), \text{vertex}_{\mathcal{H}}(A, C_j)\}, \{\text{vertex}_{\mathcal{H}}(B, C_j)\})$.

We show in [27] that if $\langle X, Y, \phi \rangle \in \mathrm{QSAT}_2(F)$ then the cyclomatic number of $\mathcal{H}$ is m and if $\langle X, Y, \phi \rangle \notin \mathrm{QSAT}_2(F)$ then the cyclomatic number of $\mathcal{H}$ is $m+1$. ∎

Via constructions based on the gadgets used in the proof of Theorem 8, we can show the following results.

Theorem 9. *For every $k \geq 2$,* MIN-CYCLOMATIC-SET *is Π_2^p-complete when restricted to k-directed hypergraphs.*

Theorem 10. *For every $k \geq 2$,* MINIMAL-CYCLOMATIC-SET *is* DP*-complete when restricted to k-directed B-hypergraphs.*

6 Succinct Representations of Directed Hypergraphs

It is to be noted that a hypergraph on n vertices may have $\Theta(3^n)$ hyperedges in the worst case. In contrast, the number of edges in a (simple) graph is $O(n^2)$. From an implementation perspective, any representation that stores information for individual hyperedges of a hypergraph is impractical for hypergraphs with a large number of hyperedges. Thus, alternative ways to represent hypergraphs must be explored. Several graphs occurring in practice, such as the graphs that model VLSI circuits, have a highly organized structure and can be described in a succinct way by a circuit or a boolean formula. Galperin and Wigderson [13] showed that trivial graph properties, (e.g., the existence of a triangle) become NP-complete, and Papadimitriou and Yannakakis [22] showed that graph properties that are ordinarily NP-complete become NEXP-complete when the graph is succinctly described by a circuit. In this section, we investigate the computational complexity of the L-hyperpath existence problem when directed hypergraphs are represented in an exponentially succinct way.

Definition 11. *1. A* succinct representation *of a directed hypergraph $\mathcal{H}(V, E)$, where $V = \{1, \ldots, n\}$, is a boolean circuit $C_\mathcal{H}$ with $2n$ input gates and an output gate such that, for each $e \subseteq 2^V \times 2^V$, $e \in E$ if and only if $C_\mathcal{H}(x, y)$ outputs 1, where $x = \chi_{T(e)}(1) \ldots \chi_{T(e)}(n)$ and $y = \chi_{H(e)}(1) \ldots \chi_{H(e)}(n)$.*

2. A succinct representation *of a k-directed hypergraph $\mathcal{H}(V, E)$, where $V = \{1, \ldots, n\}$, is a boolean circuit $C_\mathcal{H}$ with $2k\lceil\log(n+1)\rceil$ input gates and an output gate, where $0^{\lceil\log(n+1)\rceil}$ is the encoding of a dummy node not in $\mathcal{H}$ and for each $1 \leq i \leq n$, $bin(i)$—the binary representation of integer i in $\lceil\log(n+1)\rceil$ bits—is the encoding of node i in $\mathcal{H}$. Furthermore, for each $e = (\{i_1, \ldots, i_{\ell_1}\}, \{j_1, \ldots, j_{\ell_2}\}) \subseteq 2^V \times 2^V$, $i_1 < \ldots < i_{\ell_1}$ and $j_1 < \ldots < j_{\ell_2}$, $e \in E$ iff $C_\mathcal{H}(x, y)$ outputs 1, where $x = 0^{(n-\ell_1)\lceil\log(n+1)\rceil} bin(i_{\ell_1}) \ldots bin(i_2) bin(i_1)$ and $y = 0^{(n-\ell_2)\lceil\log(n+1)\rceil} bin(j_{\ell_2}) \ldots bin(j_2) bin(j_1)$.*

Definition 12. *1.* SUCCINCT-LHYPERPATH $= \{\langle C, u, v\rangle \mid C$ *succinctly represents a directed hypergraph $\mathcal{H}_C$ and* $\langle \mathcal{H}_C, u, v\rangle \in$ L-HYPERPATH$\}$.

2. k-SUCCINCT-LHYPERPATH $= \{\langle C, u, v\rangle \mid C$ *succinctly represents a k-directed hypergraph $\mathcal{H}_C$ and* $\langle \mathcal{H}_C, u, v\rangle \in$ L-HYPERPATH$\}$.

Wagner [29] (see also [22]) showed that even for simple subclasses of graphs—directed trees, directed acyclic graphs, directed forests, and undirected forests—the reachability problem for each class with succinct input representation is PSPACE-complete. Tantau [25] showed that the reachability problem for succinctly represented (strong) tournaments is Π_2^p-complete. Using the proof of Theorem 5, it can be easily shown that SUCCINCT-LHYPERPATH is NP-complete and, for every $k \geq 2$, k-SUCCINCT-LHYPERPATH is NEXP-complete.

We leave open the exact complexity of the L-hyperpath existence problem with succinct input representation for particular subclasses of directed hypergraphs.

7 Open Problems

Theorem 5 proves that L-HYPERPATH in NP-complete even when restricted to 2-directed hypergraphs with only B-hyperedges. However, Theorem 8 only shows that the CYCLOMATIC-NUMBER problem is Σ_2^p-complete for 2-directed hypergraphs. In fact, the construction uses both B- and F-hyperedges. It is interesting to analyze the complexity of the CYCLOMATIC-NUMBER problem restricted to directed hypergraphs with only B-hyperedges.

Acharya [1] showed a connection between the cyclomatic number and the planarity of an undirected hypergraph. It will be interesting to find connections between the cyclomatic number of directed hypergraphs w.r.t. L-hypercycles and notions in the theory of directed hypergraphs.

In this paper, we mentioned three problem domains where L-hyperpaths can be used to model the problem. It will be interesting to find more domains where L-hyperpaths can be used to model interesting problems.

Acknowledgment We thank Edith Hemaspaandra, Lane Hemaspaandra, Christopher Homan, Proshanto Mukherji, Srinivasan Parathasarathy, Len Schubert, and Holger Spakowski for helpful discussions and insightful comments. We are grateful to anonymous referees for helpful comments.

References

1. B. Acharya. On the cyclomatic number of a hypergraph. *Discrete Mathematics*, 27:111–116, 1979.
2. P. Alimonti, E. Feuerstein, and U. Nanni. Linear time algorithms for liveness and boundedness in conflict-free Petri nets. In *Proceedings, 1st Latin American Symposium on Theoretical Informatics*, pages 1–14. Springer-Verlag *Lecture Notes in Computer Science #583*, 1992.
3. G. Ausiello, A. D'Atri, and D. Saccá. Graph algorithms for functional dependency manipulation. *Journal of the ACM*, 30:752–766, 1983.
4. G. Ausiello, A. D'Atri, and D. Saccá. Minimal representation of directed hypergraphs. *SIAM Journal on Computing*, 15:418–431, 1986.
5. G. Ausiello, P. Franciosa, and D. Frigioni. Directed hypergraphs: Problems, algorithmic results, and a novel decremental approach. In *Italian Conference on TCS*, pages 312–328. Springer-Verlag *LNCS #2202*, 2001.

6. G. Ausiello and R. Giaccio. On-line algorithms for satisfiability formulae with uncertainty. *Theoretical Computer Science*, 171:3–24, 1997.
7. G. Ausiello, R. Giaccio, G. Italiano, and U. Nanni. Optimal traversal of directed hypergraphs. Manuscript, 1997.
8. G. Ausiello, G. Italiano, and U. Nanni. Hypergraph traversal revisited: Cost measures and dynamic algorithms. In *Proceedings of the 23rd International Symposium on MFCS*, pages 1–16. Springer-Verlag *LNCS #1450*, 1998.
9. C. Berge. *Graphs and Hypergraphs.* North-Holland, 1973.
10. G. Gallo, G. Longo, S. Pallottino, and S. Nguyen. Directed hypergraphs and applications. *Discrete Applied Mathematics*, 42:177–201, 1993.
11. G. Gallo and G. Rago. A hypergraph approach to logical inference for datalog formulae. Technical Report 28/90, Dip. di Informatica, Univ. of Pisa, Italy, 1990.
12. G. Gallo and M. Scutella. Directed hypergraphs as a modelling paradigm. Technical Report TR-99-02, Dipartimento di Informatica, February 1999.
13. H. Galperin and A. Wigderson. Succinct representations of graphs. *Information and Control*, 56(3):183–198, March 1983.
14. M. Garey and D. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness.* W. H. Freeman and Company, 1979.
15. G. Italiano and U. Nanni. On line maintainenance of minimal directed hypergraphs. In *3rd Italian Conf. on Theoretical Computer Science*, pages 335–349. World Scientific Co., 1989.
16. D. Klein and C. Manning. Parsing and hypergraphs. In *Proceedings of the 7th International Workshop on Parsing Technologies (IWPT-2001)*, 2001.
17. D. Knuth. A generalization of Dijkstra's algorithm. *Information Processing Letters*, 6(1):1–5, 1977.
18. S. Nguyen and S. Pallottino. Hyperpaths and shortest hyperpaths. *Combinatorial Optimization*, 1403:258–271, 1989.
19. L. Nielsen, D. Pretolani, and K. Andersen. A remark on the definition of a B-hyperpath. Technical report, Department of Operations Research, University of Aarhus, 2001.
20. N. Nilson. *Principles of Artificial Intelligence.* Springer Verlag, 1982.
21. C. Papadimitriou. *Computational Complexity.* Addison-Wesley, 1994.
22. C. Papadimitriou and M. Yannakakis. A note on succinct representations of graphs. *Information and Control*, 71(3):181–185, December 1986.
23. C. Petri. Communication with automata. Technical Report Supplement 1 to Tech. Report RADC-TR-65-377,1, Univ. of Bonn, 1962.
24. G. Ramalingam and T. Reps. An incremental algorithm for a generalization of the Shortest Path problem. *Journal of Algorithms*, 21:267–305, 1996.
25. T. Tantau. A note on the complexity of the reachability problem for tournaments. In *ECCCTR: Electronic Colloquium on Computational Complexity*, 2001.
26. O. Temkin, A. Zeigarnik, and D. Bonchev. *Chemical Reaction Networks: A Graph-Theoretical Approach.* CRC Press, 1996.
27. M. Thakur and R. Tripathi. Complexity of linear connectivity problems in directed hypergraphs. Technical Report TR814, Department of Computer Science, University of Rochester, September 2003.
28. J. Ullman. *Principles of Database Systems.* Computer Science Press, 1982.
29. K. Wagner. The complexity of combinatorial problems with succinct input representations. *Acta Informatica*, 23:325–356, 1986.
30. A. Zeigarnik. On hypercycles and hypercircuits in hypergraphs. *DIMACS Series in Discrete Mathematics and Theoretical Computer Science*, 51, 2000.