On Finding Hypercycles in Chemical Reaction Networks

Can Özturan

Department of Computer Eng., Bogazici University, Istanbul, Turkey

Tel: (90)-212-358-7225, e-mail: `ozturaca@boun.edu.tr`

(Revised May 25th, 2007.)

Abstract

Directed hypercycles have recently been used to model chemical reaction networks. We present an NP-completeness proof for the problem of finding a hypercycle in a directed hypergraph. This sheds some light to some open questions posed by Zeigarnik [1] who modelled chemical reactions by directed hypergraphs.

Keywords: directed hypergraphs, hypercycles, chemical reaction networks

1. Introduction

Directed hypergraphs have been used for modelling problems from diverse application domains such as chemical reaction modelling [1], propositional logic, relational databases, urban transit systems [2, 3], parsing [4], secret transfer protocols [5] and bartering [6].

Monomolecular chemical reaction networks have been modelled by directed graphs and have been studied extensively in the literature [7, 8]. Zeigarnik [1] modelled multimolecular reactions by using directed hypergraphs. In [1] Zeigarnik presented open problems about hypercycles and hypercircuits in the context of chemical reaction networks. This paper sheds some light to these questions answering in particular the following problem posed by Zeigarnik: *"Define the criterion that says if a directed hypergraph contains a hypercircuit"*. We show in this paper that the problem of finding out whether a directed hypergraph contains a hypercycle is NP-complete. Hence, any criterion proposed that says whether a directed hypergraph contains a hypercircuit will necessarily require the solution of an NP-complete problem.

Given n molecules (species) and m reactions, a *stoichiometric* $n \times m$ matrix [7] P can be constructed in which rows correspond to molecules and columns correspond to reactions. Each coefficient of this matrix $P_{s,r}$ represents the number of molecules of s produced (if $P_{s,r} > 0$) or consumed (if $P_{s,r} < 0$) or 0 otherwise. Let x be an m-dimensional vector. Suppose that a hypercycle is formed by a set of reactions $C = \{r_1, r_2, \ldots, r_k\}$. Vector x can be used to express this hypercycle by letting $x_r = 1$ for each $r \in C$ and setting $x_r = 0$ otherwise. If x represents a hypercycle, then $Px = 0$, i.e. x will be in the nullspace of P. The problem of solving $Px = 0$ with $x \geq 0$ also arises in the closely related metabolic pathways analysis [9].

In Section 2, we first present directed hypergraph and hypercycle definitions that are used. In Section 3, we establish the complexity of hypercycle existence problem which is the main contribution of this paper.

2. Definitions

A *directed hypergraph* $H(V, E)$ consists of two sets, V and E where V is a set of vertices and E is a set of hyperarcs. Each hyperarc $e = <V_t, V_h>$ is an ordered pair of non-empty disjoint subsets V_t and V_h of V. Here, V_t and V_h are the sets of vertices that appear respectively in the *tail* and *head* of the hyperarc e . An example showing a set of reactions and its directed graph representation is given in Figure 1. The *in-degree(v)* (*out-degree(v)*) of vertex v is defined to be the number of times vertex v appears in the heads (tails) of hyperarcs. The set of vertices that appear in the tail or head of a hyperarc is called a *hypernode.* An $E' \subseteq E$ *induced directed subhypergraph* $H'(V', E')$ of $E(V, E)$ is defined as a directed hypergraph with $V' = (\bigcup_{e \in E'} head(e)) \bigcup (\bigcup_{e \in E'} tail(e))$.

Directed hypergraphs are also known as AND/OR graphs [10, p. 21]. In the *AND/OR graph representation*, a bipartite directed graph is constructed with two types of nodes: AND nodes which represent hyperarcs and OR nodes which represent vertices. Figure 1(c) shows the AND/OR graph representation of the example in Figure 1(b). In the figure, the white nodes represent the AND nodes and the black nodes represent the OR nodes. We will use the notation $G(V_o, V_a, A)$ to represent the AND/OR graph corresponding to a directed hypergraph $H(V, E)$ with $V_o = V$, $V_a = E$ and $A = \{<o, a>: o \in V_o \text{ and } a \in V_a \text{ with } o \in tail(a)\} \bigcup \{<a, o>: o \in V_o \text{ and } a \in V_a \text{ with } o \in head(a)\}$. In this paper, we will call a directed hypergraph *connected* if the underlying undirected graph (each directed edge replaced by an undirected one) of its AND/OR representation is connected.

A *cycle* in a directed graph is a connected subgraph in which for all vertices v in the subgraph, we have $in\text{-}degree(v) = out\text{-}degree(v) = 1$. Depending on the application, a hypercycle in directed hypergraphs has been defined in different ways [2, 3, 5, 11]. These definitions cannot correctly model chemical reactions. Zeigarnik's hypercircuit definition based on requiring all the vertices in hypercircuit to have their in-degree to be equal to their out-degree and which also allows multiple units of elements (i.e. multiple unit vertices) correctly models hypercircuits in chemical

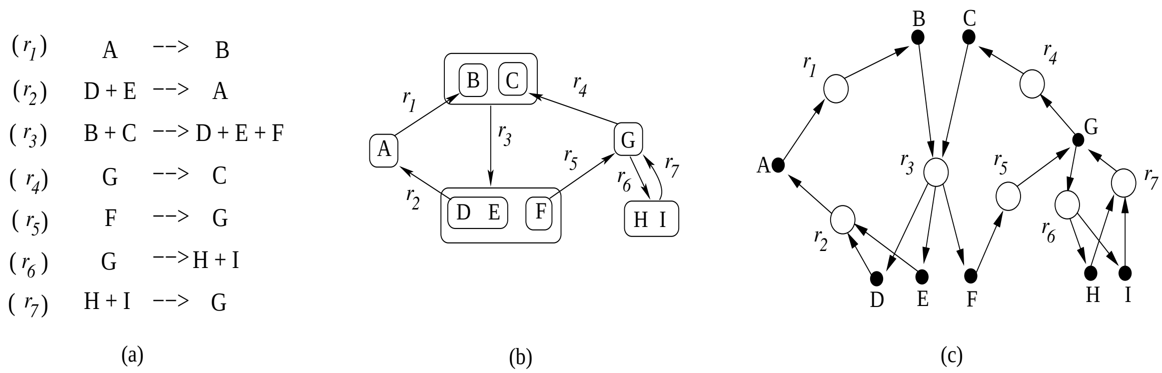

Figure 1: Example reactions (a), its directed hypergraph representation (b) and its AND/OR graph representation (c)

reactions. In this paper, we prove a complexity result for hypercycles and not hypercircuits since the result for hypercycles can be immediately extended to hypercircuits. We formally give our definition of hypercycle as follows: A *hypercycle* is defined as a connected subhypergraph of the directed hypergraph in which for all vertices v in the subhypergraph, we have $\textit{in-degree}(v) = \textit{out-degree}(v) = 1$. For a hypercircuit, we would have similar definition with $\textit{in-degree}(v) = \textit{out-degree}(v)$. In Figure 1, reaction sets $C_1 = \{r_1, r_2, r_3, r_4, r_5\}$ and $C_2 = \{r_6, r_7\}$ are hypercycles.

3. Complexity of the Hypercycle Problem

Given a connected directed hypergraph $H(V, E)$, in order to answer Zeigarnik's open question, we ask the following decision question: Does there exist a hypercycle in $H(V, E)$? We abbreviate this problem as HYC problem. The complexity of this problem is given as follows:

Theorem 1. *HYC problem is NP-complete.*

Proof. Clearly, HYC problem is in NP since we can guess a set of hyperarcs and check in polynomial time whether they form a connected subhypergraph and the vertices appearing in head and tail sets of these hyperarcs have in-degree and out-degree both equal to 1. For proving its NP-hardness, we transform the 3SAT problem [12, p. 48] to the HYC problem. Let $U = \{u_1, u_2, \ldots, u_m\}$ be a set of variables and and $C = \{c_1, c_2, \ldots, c_n\}$ be a set of clauses in conjunctive normal form (CNF) with each clause c_i having three literals. Also let $\bar{U} = \{\bar{u}_1, \bar{u}_2, \ldots, \bar{u}_m\}$ represent the set of complemented variables. We construct a directed hypergraph representing an arbitrary instance of 3SAT as follows:

(i) A hyperarc $<\{C\}, \{c_1, c_2, \ldots, c_n\}>$ is constructed which represents the conjunction of clauses.

(ii) A given clause $c_i = (a + b + c)$ with $a, b, c \in U \cup \bar{U}$ and where $+$ represents the OR function, is true if at least one of the three literals is true. For this to occur, we have 7 possibilities : $(\bar{a} + \bar{b} + c)$,$(\bar{a} + b + \bar{c})$, $(\bar{a} + b + c)$,$(a + \bar{b} + \bar{c})$,$(a + \bar{b} + c)$, $(a + b + \bar{c})$, and $(a + b + c)$. Let $c'_{i,j}$ with $j = 1 \ldots 7$ denote these possibilities. For each clause, we construct 7 hyperarcs: $<\{c_i\}, \{c'_{i,j}\}>$.

(iii) Let L_i ($\bar{L}_i$) be the set of all local literal occurences corresponding to the variable u_i ($\bar{u}_i$) that appear in the heads of hyperarcs constructed in (ii). Let also s_i with $i = 1, \ldots, m$ denote some dummy vertices. A hyperarc $<L_i, \{s_i\}>$ ($<\bar{L}_i, \{s_i\}>$) for each variable u_i (for each negation $\bar{u}_i$) is constructed.

(iv) Finally, a hyperarc $<\{s_1, s_2, \ldots, s_m\}, \{C\}>$ which connects the selected literals to the clauses is constructed.

The above construction takes polynomial time. Note that each hyperarc acts as if ANDing the vertices in the tail and ANDing the vertices in head sets. Each vertex, on the other hand, ORs exclusively (i.e. chooses) just a single incoming hyperarc and a single outgoing hyperarc. As a result, in (i) we AND all clauses. In (ii), we select one configuration which leads to truthness of a clause. In (iii), we combine all local literal configurations and select a value s_i which is either true or false indicated by selecting u_i or $\bar{u}_i$ respectively. The instance of the 3SAT is then satisfiable if there exists a hypercycle in the directed hypergraph constructed.

Conversely, we also show that a hypercycle in the directed hypergraph can be constructed if the corresponding instance of 3SAT is satisfiable. In this case, note that there exists a truth assignment such that at least one literal in each clause is set to true. The assigned values in each clause can be represented by exactly one of the 7 possibilities stated in (ii) above. This means that there is exactly one $<\{c_i\}, \{c'_{i,j}\}>$ hyperarc for each clause c_i, $i = 1, \ldots, n$. The in-degree of each local literal occurrence appearing in the heads of these hyperarcs will be 1 and those that do not appear in the heads of these hyperarcs will be 0. The hyperarc constructed in step (iii) will AND all the local literals with in-degree 1 of the same literal. Depending on whether a variable was assigned true or false, only one of u_i or $\bar{u}_i$ will be selected in accordance with the aformentioned literal. Then, all these will be ANDed into the C vertex in step (iv). Finally, the hyperarc in step (i) will connect C to all the clauses. It is clear from this construction that we end up with a hypercycle, i.e., a connected directed subhypergraph in which for all vertices v in this subhypergraph, we have $in\text{-}degree(v) = out\text{-}degree(v) = 1$. ■

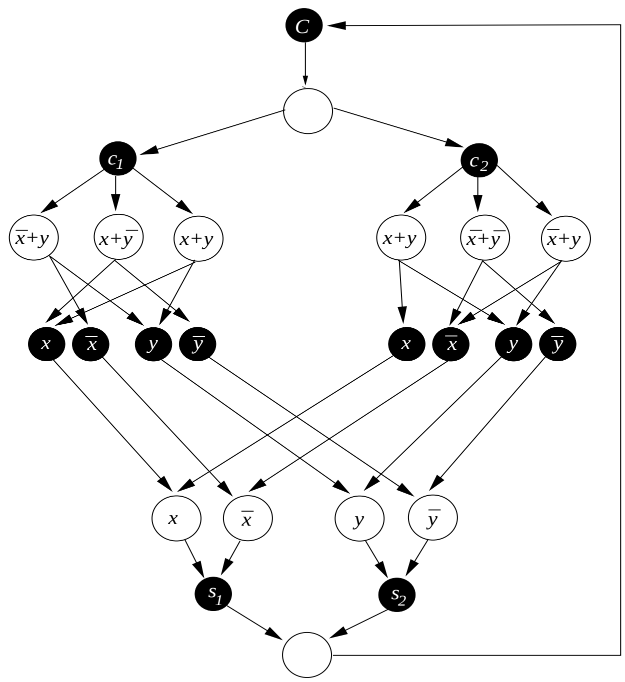

Figure 2: Directed hypergraph for satisfiability of clauses $c_1 = (x + y)$ and $c_2 = (\bar{x} + y)$.

To illustrate the hypergraph construction we prefer to give an example involving clauses with two literals (i.e. polynomially solvable 2SAT example), since 3SAT's 3 literal clause examples looks messy with a lot of nodes in the figure. Construction of the hypergraph in both cases, however, is the same except that hyperarc construction in (ii) now has 3 possibilities instead of 7. The hypergraph constructed is for the satisfiability of clauses $c_1 = (x + y)$ and $c_2 = (\bar{x} + y)$ and is given in Figure 2. Note that here, we use the directed bipartite graph representation (as in Figure 1(c) of the hypergraph - the white nodes represent the hyperarcs and the black nodes represent the vertices). As an example, if we choose $\bar{x}$ and y as incoming hyperarcs to s_1 and s_2 respectively, then a hypercycle can be formed. This corresponds to the values $x = false$ and $y = true$. On the other hand, the choice of x and $\bar{y}$ does not lead to any hypercycle.

We can conclude our paper as follows: In Zeigarnik [1], a *semi-hypergraph* is constructed from a directed hypergraph by removing the directions on hyperarcs. Our NP-completeness proof can be extended immediately to hypercycle in semi-hypergraph problem by removing directions on hyperarcs in our transformation. One of Zeigarnik's main motivation is to come up with a hypercycle formulation that will obey Euler's Formula for the cyclomatic number. Since this formula involves one addition and a subtraction operation (i.e. it is polynomially computed), it would give us yes/no answer about the existence of hypercycles. However, from our NP-completeness result and its extension to semi-hypergraphs, we can conclude that one cannot come up with a general hypercycle formulation (i.e. one that will be in the nullspace of stoichiometric matrix) that can be polynomially computed and that also preserves Euler's formula unless P=NP.

References

[1] A. V. Zeigarnik. On hypercycles and hypercircuits in hypergraphs. In *Discrete Mathematical Chemistry (DIMACS Series in Discrete Mathematics and Theoretical Computer Science)*, vol. 51, pages 377–383. AMS, 2000.

[2] G. Ausiello, R. Giaccio, G. F. Italiano, and U. Nanni. Optimal traversal of directed hypergraphs. Tech. Rep. TR-92-073, The International Comp. Sci. Inst., Berkeley, CA, 1992.

[3] G. Gallo, G. Longo, S. Nguyen, and S. Pallottino. Directed hypergraphs and applications. *Discrete Appl. Math.*, (40):177–201, 1993.

[4] D. Klein and C. Manning. Parsing and hypergraphs. In *Proceedings of the 7th International Workshop on Parsing Technologies (IWPT-2001)*, 2001.

[5] M. Thakur and R. Tripathi. Complexity of linear connectivity problems in directed hypergraphs. In *Proc. Foundations of Software Technology and Theoretical Computer Sci., 24th International Conference, Chennai, India*, pages 481–493, 2004.

[6] C. Ozturan. Resource bartering in data grids. *Scientific Programming*, 12(3):155–168, 2004.

[7] P. M. Gleiss, P. F. Stadler, A. Wagner, and D. A. Fell. Relevant cycles in chemical reaction networks. *Advanced Complex Systems*, 4:207–226, 2001.

[8] O. Temkin, A. Zeigarnik, and D. Bonchev. *Chemical Reaction Networks: A Graph Theoretical Approach.* CRC Press, 1996.

[9] J. A. Papin, N. D. Price, S. J. Wilback, D. A. Fell, and B. O. Palsson. Metabolic pathways in the post-genome era. *Trends in Biochemical Sciences*, 28(5):250–258, 2003.

[10] J. Pearl. *Heuristics.* Addison Wesley, 1984.

[11] Berge C. *Hypergraphs: Combinatorics of Finite Sets.* Elsevier, North Holland, 1989.

[12] M. Garey and D. S. Johnson. *Computers and Intractability, A Guide to the Theory of NP-Completeness.* W. H. Freeman and Company, 1979.