BACHELOR

# Minimum reaction cuts in metabolic networks

Cardon, S.

*Award date:*
2009

# Minimum reaction cuts in metabolic networks
Bachelor eindproject van

Stefan Cardon, 498327

begeleiding: J.C.M. Keijsper
tweede beoordeling: C.A.J. Hurkens

August 12, 2009

# Contents



# 1 Introduction

In the cell of an organism a chemical reaction can take place that is unwanted, i.e. it could be bad for the organism. Stopping or blocking this reaction then naturally becomes a interesting problem. The mathematical formulation that is given later on is proven to be NP-hard. This paper will look in to the relation between reaction cuts and directed hypergraph cuts. This is motivated by the fact that finding cuts in graphs is easy. In the second section of this paper cuts in directed hypergraphs will be studied, in order to discus their relation with reaction cuts in the third section.

In this section metabolic networks will be defined and some properties explained. A mathematical model will be given in terms of directed hypergraphs which will be defined along the way. Then Reaction Cuts in metabolic networks can be treated. The last part of this section is about how the rest of the paper is going to discuss the usefulness of directed hypergraphs as a modeling tool for Reaction Cuts in metabolic networks.

In the cells of organisms cellular compounds interact in a vast number of ways. Such an interaction is called a reaction. The compounds needed for a reaction are called the input compounds of the reaction. The compounds that appear after the reaction are the output or result compounds. A set of these reactions is called a metabolic network if they are connected. Here connected means that they have compounds in common. The compounds that are not output of one of the reactions in the network are called the input compounds of the metabolic network, similarly output compounds of a metabolic network are the compounds that are no input compounds of a reaction in the network.

In many applications networks are modeled by graphs. This is also a natural way to model Metabolic Networks. In order to be able to make a distinction between input and output compounds it is useful to use a directed graph. Because a reaction can have multiple input and output compounds it would be nice to have a kind of edge that is able to connect more than two compounds. A directed hypergraph is a tool that has such properties, so a formal definition will be given and then it will be explained how this will be used to model a Metabolic Network.

**Definition** A *directed hypergraph* is a pair $H = (V, E)$, where $V = \{v_1, v_2, \ldots, v_n\}$ is the set of vertices (or nodes) and $E = \{E_1, E_2, \ldots, E_m\}$ the hyperedges where each $E_i$ is an ordered pair $E_i = (X, Y)$ of possibly empty disjoint subsets of vertices; $X$ is called the tail and $Y$ the head of $E_i$.

Now the hypergraph model of a Metabolic Network can be described. The compounds will be seen as the vertices of the graph and the hyperedges will describe the reactions where the tail of a hyperedge consists of the input compounds and the head of the output compounds. A reversible reaction will be modeled by two hyperdedges, one for each direction of the reaction. A directed hypergraph can be described by its vertex-hyperedge incidence matrix. In this matrix a zero occurs when a vertex is not contained in a edge, a minus one appears when a vertex is part of the tail of an edge and a one means that a vertex is in the head of the edge. In biochemical terms a particular extension of this incidence matrix is called the stoichiometric matrix. An example of a hypergraph and its stoichiometric matrix are given in Figure 1 and Table 1.

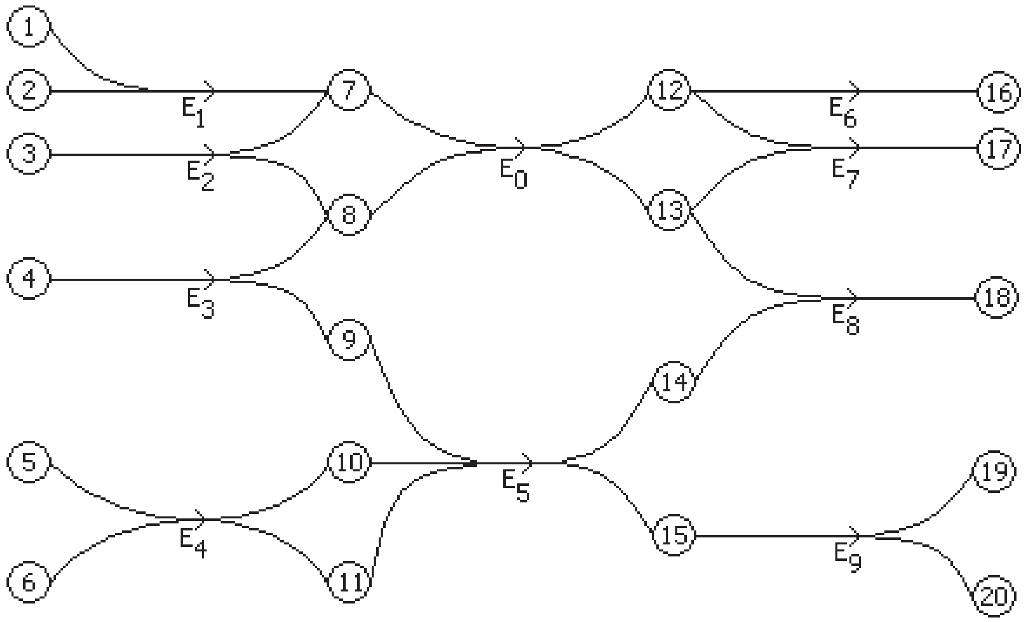



Figure 1: A directed hypergraph

In the *stoichiometric matrix* there are reactions $D_i$ given that are not in the figure. These are dummy reactions that represent the entering of the cell by input compounds and the exiting of the output compounds. These dummy reactions are needed for the definition of modes and elementary modes.

| | $E_0$ | $E_1$ | $E_2$ | $E_3$ | $E_4$ | $E_5$ | $E_6$ | $E_7$ | $E_8$ | $E_9$ | $D_1$ | $\dots$ | $D_6$ | $D_{16}$ | $\dots$ | $D_{20}$ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 0 | -1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | | 0 | 0 | | 0 |
| 2 | 0 | -1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | | 0 | 0 | | 0 |
| 3 | 0 | 0 | -1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | | 0 | 0 | | 0 |
| 4 | 0 | 0 | 0 | -1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | | 0 | 0 | | 0 |
| 5 | 0 | 0 | 0 | 0 | -1 | 0 | 0 | 0 | 0 | 0 | 0 | | 0 | 0 | | 0 |
| 6 | 0 | 0 | 0 | 0 | -1 | 0 | 0 | 0 | 0 | 0 | 0 | | 1 | 0 | | 0 |
| 7 | -1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | | 0 | 0 | | 0 |
| 8 | -1 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | | 0 | 0 | | 0 |
| 9 | 0 | 0 | 0 | 1 | 0 | -1 | 0 | 0 | 0 | 0 | 0 | | 0 | 0 | | 0 |
| 10 | 0 | 0 | 0 | 0 | 1 | -1 | 0 | 0 | 0 | 0 | 0 | | 0 | 0 | | 0 |
| 11 | 0 | 0 | 0 | 0 | 1 | -1 | 0 | 0 | 0 | 0 | 0 | | 0 | 0 | | 0 |
| 12 | 1 | 0 | 0 | 0 | 0 | 0 | -1 | -1 | 0 | 0 | 0 | | 0 | 0 | | 0 |
| 13 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | -1 | -1 | 0 | 0 | | 0 | 0 | | 0 |
| 14 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | -1 | 0 | 0 | | 0 | 0 | | 0 |
| 15 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | -1 | 0 | | 0 | 0 | | 0 |
| 16 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | | 0 | -1 | | 0 |
| 17 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | | 0 | 0 | | 0 |
| 18 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | | 0 | 0 | | 0 |
| 19 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | | 0 | 0 | | 0 |
| 20 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | | 0 | 0 | | -1 |

Table 1: The stoichiometric matrix

A set of reactions is called *a reaction cut* for the objective reaction $r_0$ if blocking them causes the objective reaction not to take place. This notion will be defined exactly in a little while, modes and elementary modes in particular need to be defined first.

**Definition** Given the stoichiometric matrix $S$ of a metabolic network, a *mode* is a vector $v$ for which:

- $Sv = 0$
- $v_j \geq 0$ if j is a irreversible reaction.

An *elementary mode* then is a mode $v \neq 0$ for which it holds:

- There exist no mode $w \neq 0$ such that the nonzero elements of $w$ are contained in the set of nonzero elements of $v$

To be able to associate reactions with a mode, the *support* of a (elementary) mode is defined as the set of reactions corresponding to the nonzero entries of the mode.

In the metabolic network depicted in Figure 1 there are two elementary modes. The support of these modes is shown in Figure 2.

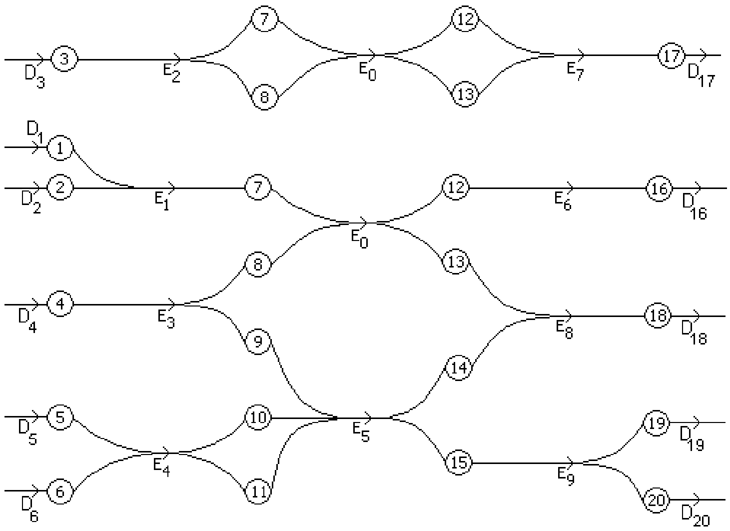



Figure 2: The support of two elementary modes

If a reaction is excluded from an elementary mode it is no longer a mode and we call the mode *blocked*. Then a reaction in that mode can only take place if it is also in an other elementary mode. This leads to the following definition for a reaction cut.

**Definition** A *Reaction Cut* for objective reaction $r_0$ is set of reactions containing at least one reaction of the support of every elementary mode containing $r_0$.

A *trivial* Reaction Cut of course is $\{r_0\}$. This might be uninteresting for several reasons. A reason could be the inability to exclude this reaction, this is influenced not only by the input compounds but also by enzymes. An other reason for this trivial cut not to be interesting is that there are more then one objective reactions. There could be even a worse situation namely you are only able to control the input compounds of the network, which would mean you are only able to cut the dummy reactions. These possibilities will be considered in the modeling of a reaction cut with a directed hypergraph minimal cut.

**Definition** A *Minimum Reaction Cut* for objective reaction $r_0$ is a nontrivial Reaction Cut excluding $r_0$ consisting of no more reactions then any other nontrivial Reaction Cut excluding $r_0$.

In the network given in Figure 1 with $E_0$ the objective reaction a minimum reaction cut is for instance given by $\{E_1, E_7\}$. The problem of finding a minimum reaction cut given a metabolic network and an objective reaction is NP-hard which is a result of [Acuña et al.].

# 2 Min Cut in directed hypergraphs

This section will treat the concept of minimum cut in directed hypergraphs. This is not as straight forward as in ordinary graphs. There are several ways to define Cuts in a hypergraph. Two of these will be given first and then treated separately because the hardness of finding a minimum cut differs for these different definitions.

It is not exactly true that there are multiple definitions for cuts in directed hypergraphs, more precisely the differences appear in the cutsets. A cut can be defined exactly in the same way as in ordinary graphs and undirected hypergraphs.

**Definition** A *cut* $C = (S, T)$ is a partition of the vertices $V$ of a directed hypergraph $H = (V, E)$.

Because $T = V \setminus S$ a cut can also be given by a single set of vertices $S$, this will be done in some contexts. For an ordinary graph the cut set $\delta(S)$ of $S$ would be defined as the edges $(u, v)$ with $u \in S$ and $v \in T$, for an undirected hypergraph it would be the set of arcs that have at least one vertex in $S$ and in $T$. Now for directed hypergraphs there are some choices. Do all heads of an arc need to be in $T$ and all tails in $S$ or just one? Also does the cut set consist of only arcs from $S$ to $T$ or do the others contribute as well? With the following definitions these possible choices will be made more clear and more formal.

**Definition** The *All tails in, all heads out Cut* $C = (S, T)$ is a cut as above, with as cut set

$$\delta_{all}(S) := \{E = (X, Y) | X \subseteq S, Y \subseteq V \setminus S\}$$

.

**Definition** The *A tail in, a head out Cut* $C = (S, T)$ is a cut as above, with as cut set

$$\delta_{a,a}(S) := \{E = (X, Y) | X \cap S \neq \varnothing, Y \cap V \setminus S \neq \varnothing\}$$

.

Both these definition can be made bidirectional, so that they include both arcs from $S$ to $T$ and arcs from $T$ to $S$. Because the *bidirectional all tails in, all heads out* version will not be treated in this paper a formal definition will only be given for the other variant.

**Definition** The *Bidirectional a tail in, a head out Cut* $C = (S, T)$ is a cut as above, with as cutset

$$\delta_{bi,a,a}(S) := (E = (X, Y) | (X \cap S \neq \varnothing \wedge Y \cap V \setminus S \neq \varnothing) \vee (X \cap V \setminus S \neq \varnothing \wedge Y \cap S \neq \varnothing)$$

.

The min Cut problem on directed hypergraphs now is to find a cut for which the summed weight of all arcs in the cutset is not larger than that of any other cutset, where the weight of an arc is given by $w : E \to \mathbb{R}$ the weight function of the arcs.

Often we are only interested in finding a minimum cut of a special type, i.e. an $s - t$-cut is a cut $C = (S, T)$ such that $s \in S$ and $t \in T$.

In the next two subsections the hardness of finding a minimum cut is discussed for several of the cases above. In the first of these two subsections the *All tails in, all heads out* definition will be treated. For this case it will be shown that the corresponding weight function is not submodular and a result of [Gallo et al.] for a special case will be stated. In the second subsection the *A tail in, a head out* case will be treated, for this case it will be shown that the corresponding weight function is submodular. And for the *bidirectional* version of this case an algorithm will be given.

## 2.1 All tails in, all heads out

In this section some aspects of finding a minimum cut in the *All tails in, all heads out* case are treated. It will be shown that the corresponding weight function is not submodular, this will be done in the first part of this subsection. In the second part a result of [Gallo et al.] will be mentioned. This result shows that finding a minimum $s-t$-cut is NP-hard.

### 2.1.1 Submodularity

Submodularity will be introduced in this part and it will be shown that the weight function corresponding to the cut definition in this section is not submodular. It is interesting to look at submodular functions because they can be minimized in polynomial time. For instance [Schrijver] gives an algorithm to do so.

Let $2^N$ denote the set of all subsets of $N$.

**Definition** A function $w : 2^N \to \mathbb{R}$ is *submodular* if for all subsets $A, B \subseteq N$

$$w(A \cap B) + w(A \cup B) \leq w(A) + w(B)$$

In the min cut problem as defined above the weight of the cutset is to be minimized, the weight-function $w_{all} : 2^N \to \mathbb{R}$ that gives the weight of cutset of the *All tails in, all heads out* definition can be defined as follows given a weightfunction $w : E \to \mathbb{R}$ on the edges.

$$w_{all}(A) := w(\delta_{all}(A)) = \sum_{e \in \delta_{all}(A)} w(e)$$

To show this weightfunction $w_{all}$ is not submodular it suffices to consider a hypergraph $H = (V, E)$ such that $V = \{a, b, v\}$ and $E$ consists only of one arc with tails $a$ and $b$ and $v$ as head. Now $A, B \subset V$ are such that $A = \{a\}$ and $B = \{b\}$. Then the arc contributes to $w(A \cup B)$ not to $w(A)$ and $w(B)$. This example is shown in figure 3.

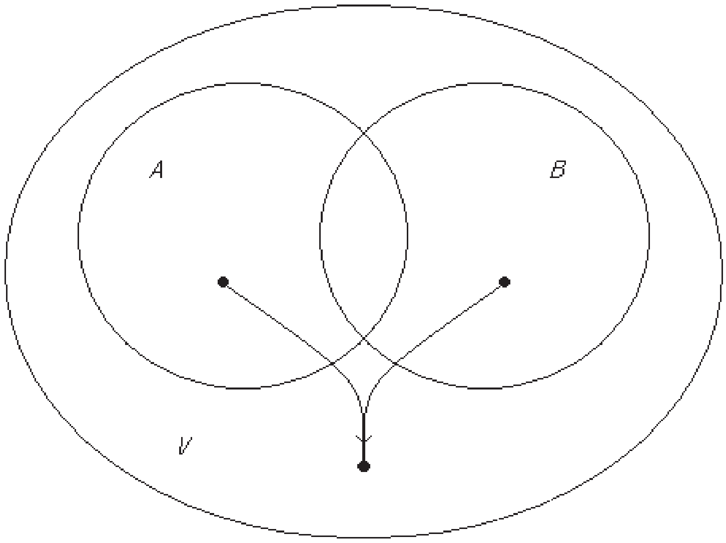



Figure 3: An arc that contributes $w(A \cup B)$ but not to $w(A)$ and $w(B)$

#### 2.1.2 A special case

There exists a special case of the *All tails in, all heads out* min $s-t$-cut which is known to be NP-hard. This special case is when all arcs have only one head, so $Y$ is consists only of a single point, this is called a *B-graph.* This is a result of [Gallo et al.], they proved this by reducing Max-Horn-Sat to Min $s-t$-Cut in B-graphs. Max Horn Sat is a special case of the satisfiability problem that is known to be NP-hard. Hence the general *all tails in, all heads out* $s-t$ case is NP-hard.

### 2.2 A tail in, a head out

In this section the *A tail in, a head out* case will be discussed. It will be shown that the corresponding weightfunction is submodular for the one directional case. In the second part of this section a algorithm will be given for the bidirectional case. This algorithm will be an extension of an algorithm for an ordinary graph given by Stoer and Wagner in [StWa].

#### 2.2.1 Submodularity

Similar to section 2.1.1 a weightfunction $w_{a,a} : 2^N \to \mathbb{R}$ corresponding to the cut definition in this section can be formulated. The weightfunction in the *A tail in, a head out* case is:

$$w_{a,a}(A) := w(\delta_{a,a}(A)) = \sum_{e \in \delta_{a,a}(A)} w(e)$$

To prove this weightfunction is submodular it will be shown that the hyperarcs which contribute to $W(A \cap B) + W(A \cup B)$ contribute at least as much to $W(A) + W(B)$. This will be done by case distinction in table 2. Because the one directional case is considered here it is not useful to know if a arc has tails anywhere except in $A \setminus B$, $B \setminus A$ or $A \cap B$, because that does not infect their contribution to the sets. The same holds for the heads in $A \setminus B$,$B \setminus A$ and $V \setminus (A \cup B)$.

### 2.2.2 Algorithm

In this section a polynomial algorithm for the bidirectional *a tail in, a head out* case will be given. The algorithm extends both the algorithm of Stoer an Wagner [StWa] for ordinary graphs and the one for undirected hypergraps by Klimmek and Wagner [KliWa].

The algorithm consists of $|V|-1$ phases, called minimum cut phases. They have as input a directed hypergraph $H$, a weightfunction on the edges $w$ and a starting vertex $a$. A phase outputs a cut of the phase, the lightest of these cuts of the phase is the minimum cut. A phase looks like:

**MinimumCutPhase**$(H, w, a)$
$A \leftarrow a$
*while* $A \neq V$
  add to $A$ the most tightly connected vertex
store the cut-of-the-phase and shrink $H$ by merging the two vertices added last

This minimum cut phase needs a little more explanation and formal definitions of *the most tightly connected vertex*, *cut-of-the-phase* and *merging two vertices*.

A subset $A$ of the hypergraph's vertices grows starting with an arbitrary single vertex until $A$ is equal to $V$. In each step the vertex outside A *most tightly connected* is added to $A$ until $A$ contains al vertices. Formally a vertex $z \in V \setminus A$ such that $w(A, z) = max\{w(A, y)|y \in V \setminus A\}$ is the *most tightly connected* vertex, where $w(A, y)$ is the sum of the weights of the set of arcs $(A, y)$. The set $(A, y)$ contains the arcs with a tail in $A$ and a head $y$ and the arcs with a head in $A$ and a tail $y$. At the end of a phase the vertices added last are *merged*, i.e., the two vertices are replaced by one new one. The hyperedges containing only the two merged vertices are removed. The cut $C(\{s\}, V \setminus \{s\})$ where $s$ is the vertex added last (before the merge) is called the *cut-of-the-phase*. The lightest of these cuts-of-the-phase is the desired minimum cut. The algorithm can be described as follows.

**MinimumCut**$(G, w, a)$
*while* $|V| > 1$
  **MinimumCutPhase**$(G, w, a)$
  *if* the cut-of-the-phase is lighter than the current minimum cut
    *then* store the cut-of-the-phase as the current minimum cut.

The algorithm is given and explained but not yet proven to be correct, this will be done in the last part of this section. The core of the proof of correctness is the following lemma.

**Lemma 2.1** *Each cut-of-the-phase is a minimum s-t cut in the current hypergraph, where s and t are the two vertices added last.*

**Proof** The run of MinimumCutPhase orders the vertices of the current hypergraph linearly, starting with $a$ and ending with $s$ and $t$, where $t$ is added last to $A$ and $s$ is the vertex added just before that. Now we look at an arbitrary $s-t$ cut $C = (S, T)$ of the current hypergraph and show, that it is at least as heavy as the cut-of-the-phase.

We call a vertex $v \neq a$ *active* (with respect to $C$) when $v$ and the vertex added just before $v$ are in two different parts of $C$. Let $w(C)$ be a shorter notation for $w(\delta_{bi,a,a}(S))$, $A_v$ the set of all vertices

| $T(e)$ | | | $H(e)$ | | | Contributes to | | | |
|---|---|---|---|---|---|---|---|---|---|
| $A \setminus B$ | $B \setminus A$ | $A\cap$ B | $A \setminus B$ | $B \setminus A$ | $V \setminus (A \cup B)$ | $W(A \cap B)$ | $W(A \cup B)$ | $W(A)$ | $W(B)$ |
| x | | | x | | | | | | |
| x | | | | x | | | | x | |
| x | | | | | x | | x | x | |
| x | | | x | x | | | | x | |
| x | | | x | | x | | x | x | |
| x | | | | x | x | | x | x | |
| x | | | x | x | x | | x | x | |
| | x | | x | | | | | | x |
| | x | | | x | | | | | |
| | x | | | | x | | x | | x |
| | x | | x | x | | | | | x |
| | x | | x | | x | | x | | x |
| | x | | | x | x | | x | | x |
| | x | | x | x | x | | x | | x |
| | | x | x | | | x | | | x |
| | | x | | x | | x | | x | |
| | | x | | | x | x | x | x | x |
| | | x | x | x | | x | | x | x |
| | | x | x | | x | x | x | x | x |
| | | x | | x | x | x | x | x | x |
| | | x | x | x | x | x | x | x | x |
| x | x | | x | | | | | | x |
| x | x | | | x | | | | x | |
| x | x | | | | x | | x | x | x |
| x | x | | x | x | | | | x | x |
| x | x | | x | | x | | x | x | x |
| x | x | | | x | x | | x | x | x |
| x | x | | x | x | x | | x | x | x |
| x | | x | x | | | x | | | x |
| x | | x | | x | | x | | x | |
| x | | x | | | x | x | x | x | x |
| x | | x | x | x | | x | | x | x |
| x | | x | x | | x | x | x | x | x |
| x | | x | | x | x | x | x | x | x |
| x | | x | x | x | x | x | x | x | x |
| | x | x | x | | | x | | | x |
| | x | x | | x | | x | | x | |
| | x | x | | | x | x | x | x | x |
| | x | x | x | x | | x | | x | x |
| | x | x | x | | x | x | x | x | x |
| | x | x | | x | x | x | x | x | x |
| | x | x | x | x | x | x | x | x | x |
| x | x | x | x | | | x | | | x |
| x | x | x | | x | | x | | x | |
| x | x | x | | | x | x | x | x | x |
| x | x | x | x | x | | x | | x | x |
| x | x | x | x | | x | x | x | x | x |
| x | x | x | | x | x | x | x | x | x |
| x | x | x | x | x | x | x | x | x | x |

Table 2:

added before $v$ (excluding $v$), $C_v = (S \cap (A_v \cup \{v\}), T \cap ((A_v \cup \{v\}))$ is called the cut *induced* on $A_v \cup \{v\}$ by $C = (S, T)$, and $w(C_v)$ hence is short for $w(\delta_{bi,a,a}(S \cap (A_v \cup \{v\})))$ We show for every active vertex $v$

$$w(A_v, v) \leq w(C_v)$$

by induction on the number of active vertices: For the first active vertex the inequality is satisfied with equality. Let the inequality hold for all active vertices up to the active vertex $v$, and let $u$ be the next active vertex that is added. Then we have

$$w(A_u, u) = w(A_v, u) + w((A_u \setminus A_v, u) \setminus (A_v, u))$$

Now, $w(A_v, u) \leq w(A_v, v)$ as $v$ was chosen as the vertex most tightly connected to $A_v$. By induction $w(A_v, v) \leq w(C_v)$. All hyperedges in $(A_u \setminus A_v, u) \setminus (A_v, u)$ contribute to $w(C_u)$ but not to $w(C_v)$. So

$$w(A_u, u) \leq w(C_v) + w((A_u \setminus A_v, u) \setminus (A_v, u)) \leq w(C_u)$$

As t is an active vertex with respect to $C$ we can conclude that $w(A_t, t) \leq w(C_t)$ which says exactly that the cut-of-the-phase is at most as heavy as $C = C_t$. □

Using the following theorem of [StWa] that gives a simple case distinction, it is shown that the smallest of these cuts-of-the-phase is indeed the minimum cut we are looking for.

**Theorem 2.2** *Let $s$ and $t$ be two vertices of a directed hypergraph $H$. Let $H/\{s,t\}$ be the graph obtained by identifying $s$ and $t$. Then a minimum cut of $H$ can be obtained from taking the smaller of a minimum $s - t$-cut of $H$ and a minimum cut of $H/\{s,t\}$, in the bidirectional* all tails in, all heads out *case.*

**Proof** If there exists no minimum cut separating $s$ and $t$ then identifying $s$ and $t$ does not affect any minimum cut because there is no arc for which a head or tail moves from one part of the minimum cut to the other.

It can be noticed that this theorem does not hold in the *all tails in, all heads out* case. This is can be shown by a counterexample. Consider a directed hypergraph $H = (V, E)$ with $V = \{s, t, u\}$ containing an arc with tail $s$ and heads $t$ and $u$, and an arc with just tail $s$ and head $t$. Both arcs have weight one. Clearly $c = (\{s, t\}, \{u\})$ is minimum cut, it has weight zero. The minimum $s - t$-cut is $C = (\{s, u\}, \{t\})$ with weight one, and the minimum cut in $H/\{s,t\}$ is $C = (\{s\}, \{u\})$ with also weight one. Hence neither the minimum $s - t$-cut nor the minimum cut in $H/\{s,t\}$ is the minimum cut in the hypergraph. Therefor the algorithm does not work for the *all tails in all heads out* definition.

# 3 The relations between directed hypergraph cuts and reaction cuts

In this section it is discussed which if any of the in the previous section defined cuts are reaction cuts and if so what the minimum cut of a hypergraph tells about the minimum reaction cut. In order to ensure that the objective reaction is blocked some modeling is needed. For instance $s-t$-cut sets can be used where $s$ and $t$ depend on the objective reaction and/or the weightfunction depends on the objective reaction and or other information about a practical situation. The first two models that will be discussed in this section will be using a $s-t$-cut sets and the last one uses the weightfunction. First the result of the previous section need to be checked for directed hypergraph $s-t$-cuts.

The first result of the previous section need not be checked because the $s-t$ variant for the *all tails in, all heads out* case is NP-hard. This was shown in the section 2.1.2 via a special case. For the *a tail in, a head out case* case the submodularity of the corresponding weightfunction was proven with the case distinction in table 2. This still holds for the $s-t$ case if the ground set $2^N$ of the weightfunction $w_{a,a} : 2^N \to \mathbb{R}$ is restricted to all subsets that contain $s$ and do not contain $t$, i.e. all subsets that are $s-t$-cuts. The algorithm from 2.2.2 is not useful to find minimum bidirectional $s-t$-cuts.

## 3.1 All tails in, all heads out $s-t$-cut

For this definition of a cut no efficient algorithm is known but it is still going to be discussed how it relates to a reaction cut. For $s$ the point is taken that is connected to the input compounds with dummy reactions, $t$ similarly is the point connected to the output compounds. The weight function is taken nonnegative but is otherwise not specified. With a simple example it can be shown that a *all tails in all heads out* $s-t$-cut set is not necessarily a reaction cut.

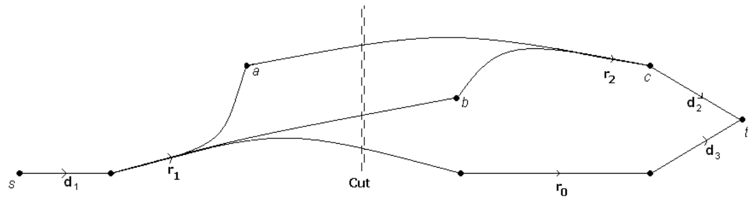



Figure 4: Example of all tails in all heads out cut that is not a reaction cut

Consider the metabolic network in Figure 4 consisting of only one elementary mode, with the objective reaction $r_0$ to be stopped. It can be seen that the *all tails in, all heads out* cut set corresponding to the cut indicated in the figure does not contain any of the reactions therefore $r_0$ is not stopped and the cut is not a reaction cut.

To show a reaction cut is not necessarily a *all tails in, all heads out* $s-t$-cut Figure 5 can be used. Clearly $\{r_2\}$ is still a reaction cut, for $\{r_2\}$ to be the cut set of a *all tails in, all heads out* $s-t$-cut $S$ needs to contain at least $s$, $a$ and $b$. If $S$ only contains $s$, $a$ and $b$ then would $r_4$ also be in the cut set, to prevent this $S$ should also contain the output compound of $r_4$. Then the cut set would contain $r_2$ and $d_3$ to prevent this $t$ should be added to $S$ but then it would no longer a $s-t$-cut. Hence a reaction cut is not always a *all tails in all heads out* cut set.

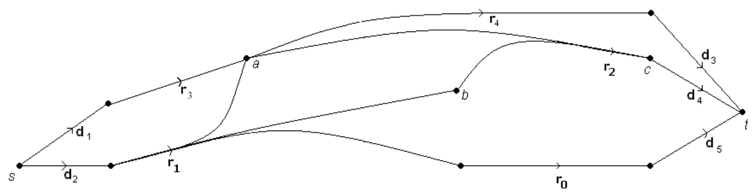



Figure 5: $\{r_2\}$ is an example of a reaction cut that is not a all tails out all heads in cut

### 3.2 A tail in, a head out $s - t$-cut

It is clear that in contrast with the other definition of a cut no reaction that *passes* the cut survives. Hence no elementary mode that *passes* the cut survives. And thus if all input compounds of the metabolic network are on one side of the cut and the output compounds on the other this clearly is a reaction cut. Hence if $s$ is connected to the input compounds of the network and $t$ is connected to the output compounds, with dummy reactions then any $s - t$-cut is a reaction cut. The weight of the arcs corresponding with reactions you can block is set to one, the weight of the arcs that cannot be blocked is set to infinity.

It is established that a *a tail in, a head out* cut in this case corresponds to a reaction cut, but what does the minimum directed hypergraph cut say about the minimum reaction cut. The example in figure 6 gives a minimum $s - t$-cut that is not a minimum reaction cut.

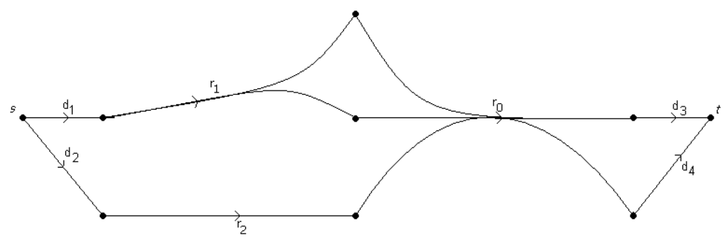



Figure 6: $\{r_1, r_2\}$ is an example of s minimum cut that is not a minimum reaction cut

Here the objective reaction $r_0$ is the only one that cannot be blocked so clearly $\{r_1, r_2\}$ forms *a tail in a head out* cut of minimum weight, but just one of the two is enough for a reaction cut.

It can also be seen that the fact wether input compounds can or cannot be controlled does not change this example, if the impossibility is modeled by infinite weight for the dummy reactions connected with input compounds. If $s$ and $t$ where only connected to the input or output compounds of the elementary modes containing $r_0$ this example still holds because this example contains only one elementary mode and this elementary mode contains $r_0$.

Next we consider a variation where all the input compounds of a elementary mode containing the objective reaction are connected with $s$ by a single hyperarc, for $t$ we do the same with the output compounds of the elementary modes. We take the weights of the arcs one except for the one objective reaction which is infinite. This way a minimum $s - t$-cut would give a reaction cut not larger then the number of elementary modes containing the objective reaction. But still it need not be optimal as can be seen in Figure 7 where $\{r_2\}$ is a minimum reaction cut and $\{d_1, d_2\}$ is minimum $s - t$-cut. But if in this model a minimum cut is found smaller then the number of elementary modes containing $r_0$ it could be an interesting reaction cut. Then if one of the dummy reactions is in the cut you are still free to choose any reaction from the corresponding elementary mode.

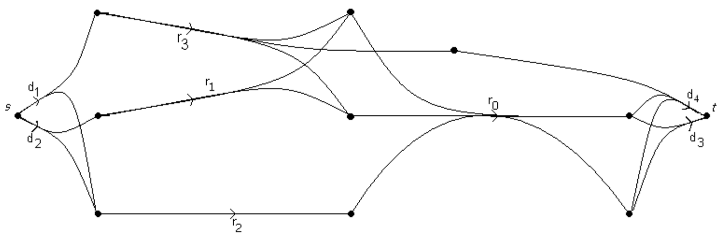



Figure 7: $\{d_1, d_2\}$ is an example of a minimum $s-t$-cut that is not a minimum reaction cut in the case the elementary modes are connected to $s$ and $t$ by a single hyperarc.

### 3.2.1 The input case

In the introduction it was already mentioned that there could be a case where you can only control the input and output compounds and no reaction at all. In this section this case will be modeled. This will be done by setting the weight on all the reactions to infinity except the dummy reactions they are set to one, where there is a dummy reaction for every input and output compound. Then the minimum cut algorithm from section 2.2.2 would give as a cut set all dummy reactions connected with $s$ or with $t$. Which is a reaction cut but not a very intelligent one, and surely no minimum one as a single input or output compound of every elementary mode would suffice.

# 4 Conclusion

There are two topics treated in this paper: minimum cuts in directed hypergraphs and the relation between these cuts and minimum reaction cuts. For minimum cut in directed hypergraphs there several definitions possible. For one a polynomial time algorithm is given.

For the *all tails in, all heads out* no polynomial time algorithm was found. The strategy of finding a minimum cut by minimizing a submodular function was tried, this strategy was not successful because the weightfunction was shown to be not submodular. There is a result of [Gallo et al.] that finding a minimum $s-t$-cut of this type is NP-hard.

In the one directional and the $s-t$ variants of the *a tail in, a head out* case the weightfunction was proven to be submodular, hence the strategy of finding a minimum cut by applying an algorithm for minimizing a submodular function would work for this case. Further more for the bidirectional case an algorithm is given to find the minimum cut in a direct way. This algorithm does not work for $s-t$-cuts in directed hypergraphs.

For the relation between directed hypergraph cuts and reaction cuts the main result is that any *a tail in, a head out* $s-t$-cut is a reaction cut, although the minimum weight directed hypergraph cut does not give any information about the minimum weight reaction cut.

Hence directed hypergraph minimum cuts are so far not a powerful tool to model minimum reaction cuts but this in not surprising for the polynomial cases of hypergraph minimum cut, as [Acuña et al.] showed a minimum reaction cut is a NP-hard problem. They can be used to find reaction cuts but the minimum $s-t$ cuts are not always minimum reaction cuts. More research can be done on directed hypergraph minimum cuts there maybe more helpful definitions, especially $s-t$-cuts could be studied. This could be done together with flows.

## References


[Acuña et al.] V. Acuña, F. Chierichetti, V Lacroix, A Marchetti-Spaccamela, M-F. Sagot, L Stougie, Modes and cuts in metabolic networks: Complexity and algorithms, Bio Systems 95, 51-60, jan 2009.

[Gallo et al.] G. Gallo, C. Gentile, D. Pretolani, and G. Rago. Max Horn SAT and the minimum cut problem in directed hypergraphs. Mathematical Programming 80, 213-237, 1998.

[KliWa] R. Klimmek, F. Wagner, A Simple Hypergraph Min Cut Algorithm, Technical Report, b 96-02, Inst. Of Computer Science, Freie Universitat Berlin, 1996.

[Schrijver] A. Schrijver, A combinatorial algorithm minimizing submodular functions in strongly polynomial time, Journal of Combinatorial Theory, Series B 80(2), 346-355, 2000.

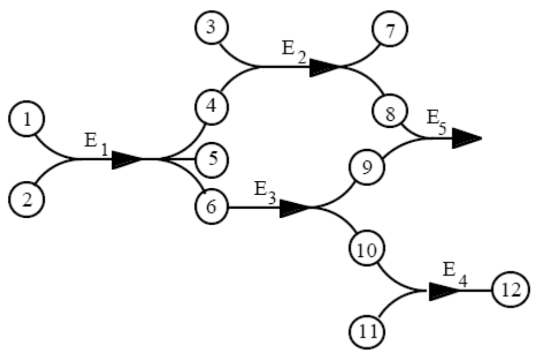
[StWa] M. Stoer, F. Wagner, A Simple Min Cut Algorithm, Proceedings of the 2nd Annual European Symposium on Algorithms, Lecture Notes in Computer Science 855, 141-147, 1994.

# Metabolische netwerken modelleren

Stefan Cardon, Technische Wiskunde, **Technische Universiteit Eindhoven** 2009

Metabolische netwerken zijn netwerken van chemische reacties in een cel van een organisme. In zo'n metabolisch netwerk kan een reactie voorkomen waarvan je liever niet hebt dat deze plaats vind, deze veroorzaakt bijvoorbeeld een ziekte. Deze reactie zelf kan je niet uitzetten wel sommige andere in het netwerk, door bijvoorbeeld de invloed van enzymen op deze reacties. Het vinden van de reacties die dit voor elkaar krijgen heet het *ReactieSnede* probleem. Natuurlijk is het interessant om dit met zo min mogelijk reacties voor elkaar te krijgen dit heet het *Minimum Reactie Snede* probleem.

Dit probleem Minimum Reactie Snede is een moeilijk probleem*, mijn bachelor eindproject ging over het oplossen of benaderen hiervan met behulp van een gerichte hypergraaf als model. Met een model wordt in de wiskunde een vaak iets vereenvoudigde weergave van het probleem in wiskundige termen bedoeld. Hieronder zal geschetst worden wat gerichte hypergrafen zijn en hoe en of ze van pas kunnen komen bij het zoeken naar een Minimum Reactie Snede.




Hiernaast is een gerichte hypergraaf afgebeeld bestaande uit punten en pijlen. In een normale graaf zou je alleen lijnen hebben tussen tweetallen punten. Het gericht zijn maakt van een lijn een pijl, dus hij gaat dan van een punt naar een ander punt en niet terug. De aanduiding hyper geeft aan dat een pijl van meerdere punten kan komen en naar meerdere punten kan gaan.

In de theorie over grafen kent men ook Snedes. Bij zo'n snede worden de punten in twee groepen verdeeld en kijkt men naar de pijlen die van de ene naar de andere groep gaan. Een verdeling waarbij er de minste pijlen van de ene naar de andere groep gaan heet een *Minimum Snede*. In een hypergraaf kan zo'n snede nog op verschillende manieren worden opgevat. Moeten alle voeten van de pijlen in een groep zitten en de uiteinden in de andere of mag er ook nog een uiteinde in de groep van de voeten zitten. Voor het vinden van sommige van deze snedes zijn wel snelle* methodes bekend.

Een van de snedes in gerichte hypergrafen is inderdaad een Reactie Snede, alleen is helaas een minimum snede in de graaf niet altijd een Minimum Reactie Snede. Een minimum snede in een hypergraaf kan dus meer reacties bevatten dan een Minimum Reactie Snede. Om te weten hoeveel meer en of er niet nog andere snedes in hypergrafen zijn die de Minimum Reactie Snede geven of beter benaderen is nog meer onderzoek nodig.

* Dit zijn vereenvoudigde begrippen uit de complexiteits theorie, bijvoorbeeld op http://nl.wikipedia.org/wiki/Complexiteitstheorie is daar meer over te lezen.