Advances in Modeling Solar System Internet Structures and their Data Flows

Alan Hylton
NASA Goddard Space Flight Center
alan.g.hylton@nasa.gov

Natalie Tsuei
American University
nt9913a@american.edu

Mark Ronnenberg
Indiana University Bloomington
maronnen@iu.edu

Jihun Hwang
Purdue University
hwang102@purdue.edu

Brendan Mallery
Tufts University
Brendan.mallery@tufts.edu

Jonathan Quartin
University of Colorado Boulder
jonathan.quartin@colorado.edu

Colin Levaunt and Jeremy Quail
University of Vermont
{colin.giles, jeremy.quail}@uvm.edu

***Abstract*— With an ever-increasing presence in space, there is also an increasing burden on existing communications infrastructure. We are heading towards an inflection point where the traditional approach of scheduled, single-path communications for space will no longer be viable. One answer is Delay Tolerant Networking (DTN), which takes the once disparate system of point-to-point links and unifies them in a networked architecture, thereby making communications more scalable. However, much work remains for discovering and harnessing the underlying theory of DTN. For example, in the terrestrial setting the interplay between routing domains is well-understood, however this is not the case in DTNs. In this paper, we build up the fundamental foundations of DTN, with an emphasis on modeling time-varying networks and data flows across them, with examples of cross-domain routing in a DTN.**

A lofty goal of DTN is to enable the so-called Solar System Internet (SSI), which implies a standardized and robust suite of protocols. These protocols include routing across disconnected networks using store, carry, and forward mechanisms, which is necessary due to the disconnections, delays, and mobility intrinsic to space networks. Due to these factors, each of which generalize traditional networking, there is a deep and rich theory of DTNs. Here we build off of past successes to broaden this theory while striving to keep actionable results a goal for future implementations and operations.

The approach includes modeling the unicast, broadcast, and multicast communications using the language of hypergraphs, which capture the geometric properties of such networked communications algebraically. Also inherent to these networks is their time-varying nature, particularly given mobility, and hence we also cultivate modeling techniques that respect this time dependence. This leads us to develop models using tools from category theory and algebraic geometry, which provide a language well-suited to describing synchronization and optimization over such networks. We also introduce and study a novel generalization of curvature applicable to time-evolving networks, which provides quantitative controls on diffusion processes on the network.

Because an interplanetary network would feature links with propagation delays the preclude discovery (feedback) mechanisms, they will always feature a scheduled component. However, it is beneficial to support discovery where possible. While DTNs do not yet have strong definitions for their analogues of autonomous systems or network areas, we show how to join dynamic and schedule-based routing domains, using the language of sheaves, which marks progress towards such definitions. We conclude with a discussion of the progress made, as well as suggestions for future work.

TABLE OF CONTENTS

1. INTRODUCTION

The Internet has enabled an unprecedented scalability in communications, and is developed to support continued scaling. For example, the Internet supports an estimated 10 billion Internet of Things (IoT) devices in 2021, among other devices, and this is expected to grow to at least 25 billion by 2025 [1]. This immense scale does not exist in space, yet not only are more satellites deployed than ever before, but deployment rates are growing quicker than ever. While the many differences in the space and terrestrial environments will support different types of communications, this fundamental goal of networking – to enable scalability – remains constant. The purpose of this paper is to develop modeling tools applicable to the so-called Solar System Internet with a goal of a positive returns to scale.

The differences between networking terrestrially and networking in space can be described as a generalization, that is, assumptions that the Internet relies upon no longer hold. These assumptions include end-to-end connectivity between communicating nodes; indeed in space end-to-end connectivity might never occur. This precludes routing as is known for the Internet. The latencies in space are also different; they can be longer than milliseconds, for example 1.2 seconds to the moon. The variances can also be long, with a range from 33 to 54 minutes to Jupiter. Networking often implies bidirectionality, but more specifically having feedback loops; instead it may be impossible to be reactive as opposed to proactive. As suggested in these examples, mobility also plays a large role – in a word, addressing. The relatively static Internet employs addressing schema; this key ingredient is missing in space.

The space communications community has developed Delay Tolerant Networking (DTN) as a potential answer to the networking-in-space question; see e.g. [2], [3]. Whereas a typical router either forwards or drops incoming message, DTN is a store, carry, and forward approach. By storing and forwarding data, known as *bundles* to DTN, routing can be enabled over temporally disconnected (and hence generalized) networks. Any links between, say, the Earth and the moon and the Earth and Mars will have fundamentally different characteristics (e.g. propagation delays). To unite these disparate links, DTN operates at a high layer as an overlay network. The benefit is that DTN supports the heterogeneity of the underlying space and ground links; this benefit is not free, and brings several well-known challenges. Among these is that there is no addressing scheme in DTN, but rather flat name spaces are used. Previous work has generalized the mathematical modeling approach to this more general setting; see see [4]. The goal of this paper is to advance this theory using tools of mathematics that can both shine some light on addressing and lend themselves to implementation.

A big restriction in routing is there is no rigorous approach to cross-domain routing. One of the benefits of the mathematical approach is that it can define the interfaces between regions while also defining these regions in the first place. The natural starting point for routing in DTNs is with scheduled routing, such as Contact Graph Routing (CGR) [5].

As a brief introduction, CGR uses globally distributed contact schedules to form a graph where the vertices represent contacts and the edges represent storage. Routes are then computed using a shortest-path algorithm, Dijkstra's algorithm, on this graph. There are several drawbacks with CGR: one is that the graph scales with the number of contacts as well as the number of nodes, and hence it becomes computationally expensive for even modest networks. Another drawback is the dependency on global consistency. There are alternatives that begin to address these, such as Delay Tolerant Link State Routing (DTLSR) [6].

Discovery-based approaches have also been researched. A typical link state router will in essence probe a network by flooding it with messages to determine local connectivity, and glues these local observations together in a consistent manner to form a global image of the network. While link state routers drop links when they go down, DTLSR rather modifies probabilities that a link will come back. With or without delay tolerance, the concept of link state routers helps illustrate the local-to-global action of network; consider for example how the local phenomenon of connecting to a network gives rise to the global phenomena of paths (routes) to other devices on the network. In mathematics, the *sheaf* is the formal data structure that takes local observations and either allows one to glue them into global data or explains obstructions to doing so. As such, networks are *sheafy*, and by constructing various sheaves over the graphs that model networks and various routing algorithms, including temporal networks, we can begin to discover their underlying structure.

This paper introduces various temporal graph and graph-like structures (i.e., hypergraphs) for modeling DTNs. We then proceed to define and further motivate sheaves in the context of networks, giving new constructions for routing across network areas. We conclude with future work enabled by our contributions.

2. Graphs and Hypergraphs

Traditionally, networks are modelled using *graphs*, which are collections of node (vertices) connected by lines (edges); graphs naturally exhibit connectivity and enable the modeling of *flow*, among other things. Examples can be seen in Figures 2 and 3. As suggested by these examples, graphs can be extended to include extra data over the vertices and edges, such as color (or any set). A prominent example would be data rates, say in gigabits per second. This makes sense in traditional networks, as full-duplex symmetric links are almost always assumed. In certain applications, such as space, neither bidirectionality nor symmetry of the links can be assumed. To build up graphs as a basic modeling tool, we introduce some of the necessary language used in this section.

Definition 2.1. A **graph** $G = (V_G, E_G)$ consists of a finite vertex set V_G and a finite edge set E_G, consisting of unordered pairs of vertices. If a vertex v lies on an edge e, then we say that v is **incident** to e.

We can then think of a typical wired network as computers/switches/routers (vertices) connected by Ethernet cables (edges). This definition can be generalized to include direction:

Definition 2.2. A **directed graph** (or digraph) $G = (V_G, E_G)$ consists of a finite vertex set V_G and a finite edge set E_G, consisting of ordered pairs of vertices. The edges are then drawn with arrows to indicate the direction.

Wired networks can always be thought of as digraphs, where the edges now represent particular directions of flow. In the case of wireless networks, however, the graph representation fails to capture all of the information about the network. We can still consider point-to-point connections as established at, say, the Network layer, but at the physical layer the connections are better described by a single edge connecting all antennas within range of each other. This observation gives rise to a generalization of graphs known as *hypergraphs*.

Definition 2.3. A **hypergraph** $H = (V_H, E_H)$ consists of a finite vertex set V_H and a finite hyperedge set E_H, consisting of non-empty subsets of V_H. A hypergraph H is called a **clutter** if no hyperedge is contained in any other.

Example 2.4. An example of a hypergraph H is given in Figure 1. This hypergraph has five vertices and three hyperedges.

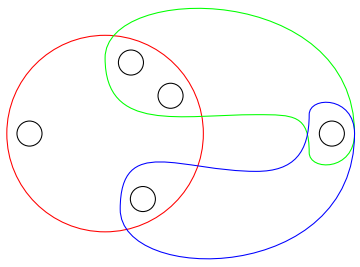


Figure 1. Example hypergraph with five vertices and three hyperedges.

It has been shown that in wireless networks, transmission models can be improved by changing from a graph model to a

hypergraph model [7]; for example, graphs cannot accurately depict the relationships among users/channels or describe the (cumulative) interference between two transmitters [8]. This is evidence that hypergraphs are the *right* structure; and following suit the goal of this paper is to find the right structures for DTNs. Moreover we could also consider a broadcast or multicast domain as single edges in a hypergraph. We hasten to add that hypergraphs are a deeper generalization of graphs than they might first appear, and to bound the problem all hypergraphs in this paper are clutters. In future work, this assumption will be relaxed.

For any hypergraph, there are many associated graphs.

Definition 2.5. Let $H = (V_H, E_H)$ be a hypergraph and s a positive integer. The **s-line graph** $L_s(H)$ is the graph with vertex set E_H and edge set given by

$$\{\{e_i, e_j\} \ : \ |e_i \cap e_j| \geq s\}.$$

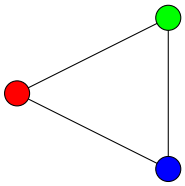


Figure 2. The line graph $L_1(H)$ for the hypergraph in Figure 1.

Example 2.6. The 1-line graph $L_1(H)$ associated to the hypergraph H in Figure 1 is depicted in Figure 2; note the correspondence of the colorings.

Another graph associated to hypergraphs is given below:

Definition 2.7. Given a hypergraph H, we define a **bipartite graph** $B(H)$ with vertex set $V_H \cup E_H$, with two vertices connected by an edge if and only if they correspond to an incident vertex and hyperedge in H.

Example 2.8. The bipartite graph $B(H)$ of the hypergraph H in Figure 1 is illustrated in Figure 3.

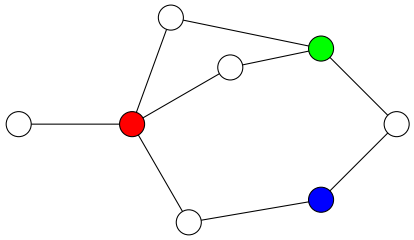


Figure 3. The bipartite graph $B(H)$ associated to the hypergraph in Figure 1.

A **path** in a hypergraph is a sequence

$$v_1, e_1, v_2, e_2, \ldots, e_{n-1}, v_n$$

of vertices v_i and hyperedges e_i such that v_i is incident to e_{i-1} and e_i, and $v_i \neq v_j$, $e_i \neq e_j$ for $i \neq j$. For a positive integer s, an **s-walk** in a hypergraph is a sequence of hyperedges $e_1, \ldots, e_n$ such that $|e_i \cap e_{i+1}| \geq s$ for each i. An s-walk is an **s-path** if in addition $e_i \neq e_j$ for $i \neq j$. Given a hypergraph H, s-walks correspond to walks in the line graph $L_s(H)$, and vice-versa. It is important to note that such graph algorithms as path finding are well-established for hypergraphs.

Hypergraphs can be **directed**. Let H be a hypergraph. We assign a direction to each hyperedge $e \in E_H$ by partitioning e into two sets, a **tail** and a **head**, denoted by $\text{tail}(e)$ and $\text{head}(e)$. We think of e as going "from $\text{tail}(e)$ to $\text{head}(e)$". If H is a directed hypergraph, then the bipartite graph $B(H)$ is naturally directed as well. However, the line graphs $L_s(H)$ are not naturally directed.

Example 2.9. Consider a radio broadcast tower transmitting to cars; in this case, there is a single hyperedge flowing from one source to many destinations.

Edges and hyperedges can have more data associated to them than data rates. An important example is scheduled times of activity, say, between a satellite and a ground station. Perhaps an ideal (hyper)graph model of a network would include data rates, one-way light times, and schedules. From these data one could compute shortest delivery times from a source to a destination; this is precisely how the aforementioned CGR works. Several generalizations of graphs can be employed to minimize the gap between the model and the network. One includes *multigraphs*, where multiple edges are allowed between any two vertices. This approach has been shown to improve computational tractability over CGR [9]. In this paper we consider several approaches to time-varying graphs to capture the temporal dependence of space networks.

3. SHEAVES I: PREVIOUS WORK

As noted in the introduction, networks exhibit local-to-global actions. This is formally captured in the mathematical language of sheaves, and has been the subject of active research, see e.g. [4], [10]. In their full generality sheaves can be intimidating. Fortunately a significantly more approachable specialization of sheaves known as *cellular sheaves* are the objects of study. In this section we provide definitions and an example of how sheaves can capture familiar path finding algorithms.

A sheaf $\mathcal{F}$ is the assignment of data to the edges and vertices of a graph along with mappings between incident edges and vertices. While the definitions are given for graphs, they extend naturally to digraphs. The extension to hypergraphs is less clear, and will be discussed later in this paper.

Definition 3.1. Let $G = (V_G, E_G)$ be a graph. A **sheaf** $\mathcal{F}$ is an assignment of

- a set or algebraic object (e.g. a vector space) to each vertex and edge of G, and

- a mapping $\mathcal{F}(v \rightsquigarrow e)\colon \mathcal{F}(v) \to \mathcal{F}(e)$[1] (of sets, vector spaces, etc.), called a **restriction map**, for all vertices v incident to edges e. Additionally, $\mathcal{F}(h)$ is referred as the **stalk** of $\mathcal{F}$ at h where $h \in V_G \cup E_G$.

The restriction maps show the relationship between the data over the vertices and the edges.

[1]For the sake of simplicity, $v \rightsquigarrow e$ may be interchangeably written as $v \leq e$ throughout this paper.

Definition 3.2. Let $G = (V, E)$ be a graph, let $H \subseteq V \cup E$, and let $\mathcal{F}$ be a sheaf over G. A **section** s is a choice of elements in each vertex and edge of H, and hence consists of a value in each of the stalks, such that for all incident pairs v and e, we have $\mathcal{F}(v \rightsquigarrow e)(s(v)) = s(e)$. If $H = V \cup E$, then s is a **global section**. The space of global section is denoted as $\Gamma(G; \mathcal{F})$.

A section is then consistent information across this set H, and if a global section exists it represents consistency across the graph. This is best illustrated with examples.

Example 3.3. Let $G = (V_G, E_G)$ be a finite directed graph. The **path sheaf** $\mathcal{P}$ over G is a cellular sheaf of sets defined as follows. We designate a source vertex $v_S \in V_G$ and a target vertex $v_T \in V_G$. Then for each vertex $v \in V_G$ we set

$$\mathcal{P}(v) = \begin{cases} \text{Out}(v), & \text{if } v = v_S \\ \text{In}(v), & \text{if } v = v_T \\ (\text{In}(v) \times \text{Out}(v)) \cup \{\bot\}, & \text{otherwise.} \end{cases}$$

For each edge $e \in E_G$ we set

$$\mathcal{P}(e) = \{\bot, \top\}.$$

The restriction maps are defined as follows. At the source vertex we have

$$\mathcal{P}(v_S \leq e)(e') = \begin{cases} \top, & \text{if } e = e' \\ \bot & \text{otherwise.} \end{cases}$$

At the target vertex we have

$$\mathcal{P}(v_T \leq e)(e') = \begin{cases} \top, & \text{if } e = e' \\ \bot, & \text{otherwise.} \end{cases}$$

Finally, for any vertex $v \neq v_S, v_T$, we have

$$\mathcal{P}(v \leq e)((e', f')) = \begin{cases} \top, & \text{if } e = e' \text{ or } e = f' \\ \bot, & \text{otherwise.} \end{cases}$$

Each global section of $\mathcal{P}$ corresponds to either a path in G from v_S to v_T, or the disjoint union of a path in G from v_S to v_T with a disjoint collection of cycles in G which are disjoint from v_S and v_T [11].

Remark 3.4. The definition that we have just given for the path sheaf is taken from [4], and requires the graph G to be directed. The original definition of the path sheaf, given in [11], is for an undirected graph. The two definitions are very similar, and for the sake of simplicity in this paper we will refer to the *path sheaf* for both directed and undirected graphs.

Having a direct relationship between a global section of a sheaf $\mathcal{P}$ and a path from v_S to v_T implies that we can write routing algorithms in a 'sheafy' way, as a form of algorithms that extend local sections to global sections. However, there is no guarantee that a global section found this way is necessarily the desired path, where the desired path is often the shortest path.

Example 3.5. Let $G = (V_G, E_G)$ be a finite directed graph with a weight function $w \colon E_G \to \mathbb{R}^+$ and a designated source node $v_S \in V_G$ and a target node $v_T \in V_G$. We define the **distance path sheaf** $\mathcal{DP}$ as a cellular sheaf on G as follows: For each $v \in V_G$ and $e \in E_G$,

$$\mathcal{DP}(v) = \begin{cases} \text{Out}(v) \times \{0\} & \text{if } v = v_S \\ \text{In}(v) \times \mathbb{R}^+ & \text{if } v = v_T \\ (\text{In}(v) \times \text{Out}(v) \times \mathbb{R}^+) \cup \{\bot\} & \text{otherwise} \end{cases}$$

$$\mathcal{DP}(e) = \mathbb{R}^+ \cup \{\bot\}$$

And the restriction maps for $\mathcal{DP}$ is defined as follows:

$$\mathcal{DP}(v_S \leq e)(e', 0) = \begin{cases} w(e) & \text{if } e = e' \\ \bot & \text{otherwise} \end{cases}$$

$$\mathcal{DP}(v_T \leq e)(e', x) = \begin{cases} x & \text{if } e = e' \\ \bot & \text{otherwise} \end{cases}$$

$$\mathcal{DP}(v \leq e)(e_i, e_o, x) = \begin{cases} x & \text{if } e = e_i \\ x + w(e) & \text{if } e = e_o \\ \bot & \text{otherwise} \end{cases}$$

$$\mathcal{DP}(v \leq e)(\bot) = \bot$$

We describe Dijkstra algorithm using a simple example. Consider the following 'toy' example graph G with four nodes and four edges in Figure 4 below. The graph G in

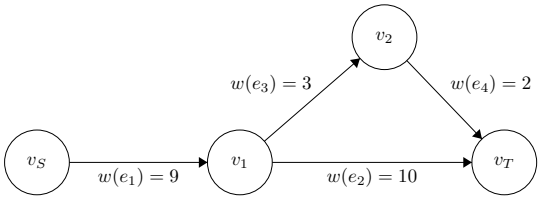


Figure 4. Setup for distance path sheaf example.

Figure 4 has two possible paths from v_S to v_T, namely $p_1 := (v_S, e_1, e_2, v_T)$ and $p_2 := (v_S, e_1, e_3, e_4, v_T)$, with p_2 being the shortest path. This also implies that there must exist only two global sections. Let σ be a global section. First, $\sigma(v_S) = (e_1, 0)$ by definition. We have $\sigma(e_1) = 9$ and so $\sigma(v_1) = (e_1, *, 9)$ where $*$ is a placeholder, unless $\sigma(e_1) = \bot$ which is impossible as σ is a global section and hence must represent a path. This denotes that v_1 is the closest node from v_S so far and the (shortest) distance between v_S and v_1 found so far is 9. From v_1, there are two choices: v_2 via e_3 or v_T via e_2. One global section would have $(\sigma(e_2), \sigma(e_3)) = (10, \bot)$ and the other would have $(\sigma(e_2), \sigma(e_3)) = (\bot, 3)$. Denote each of them as σ_1 and σ_2, i.e.

$$(\sigma_1(e_2), \sigma_1(e_3)) = (10, \bot)$$
$$(\sigma_2(e_2), \sigma_2(e_3)) = (\bot, 3)$$

We first follow σ_1. It results in $\sigma_1(v_1) = (e_1, e_2, 9)$ because it takes e_2 and marks e_3 as inactive (i.e. $\sigma_1(e_3) = \bot$), and as a result $\sigma_1(v_T) = (e_2, 9 + w(e_2)) = (e_2, 19)$. This denotes that the shortest distance from v_S to v_T found so far is $9 + 10 = 19$, and the path is (v_T, e_2, e_1, v_S) written in the reversed direction. Note that the edges saved in each $\sigma(v)$'s serve as the array of previous nodes in the traditional Dijkstra's algorithm. σ_1 is indeed a global section as it represents a path, but it is unknown yet whether it exactly corresponds to the shortest path as v_2 has not been visited

yet (recall that Dijkstra algorithm runs until every node is visited).

By the same reasoning, σ_2 gives us $\sigma_2(v_1) = (e_1, e_3, 9)$, and consequently $\sigma_2(e_4) = 2$, $\sigma_2(v_2) = (e_3, e_4, 9 + w(e_3)) = (e_3, e_4, 12)$, and then $\sigma_2(v_T) = (e_4, 12 + w(e_4)) = (e_4, 14)$. The total distance in $\sigma_2(v_T)$ is clearly less than the distance in $\sigma_1(v_T)$. From this, we conclude that both σ_1 and σ_2 are global sections of $\mathcal{DP}$ over G, yet σ_2 is the one that gives the shortest path of distance 12 represented as $(v_T, e_4, e_3, e_1, v_S)$ in the reverse order. See Figure 5 for the pictorial description of these results.

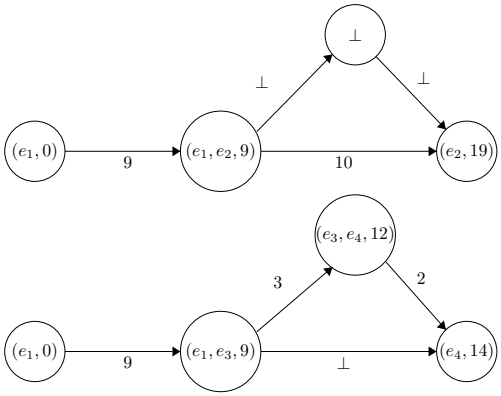


Figure 5. Paths represented as (global) sections σ_1 (top) and σ_2 (bottom) of $\mathcal{DP}$ over G in Figure 4 found using Dijkstra algorithm.

Note that in our example (Figure 4), Dijkstra's algorithm was able to find all possible paths from v_S to v_T as shown in Figure 5. However, in general, Dijkstra is neither required to find all possible paths nor all shortest paths. But the algorithm can be slightly modified to find all possible shortest paths by keeping track of all paths. For finding all possible paths, breadth first search (BFS) or depth first search (DFS) are usually used.

Similar to Dijkstra algorithm, one can write other routing methods using the language of sheaves, such as Bellman-Ford algorithm and Floyd-Warshall algorithm by allowing the weight function to have negative edge values ($w\colon E_G \to \mathbb{R}$), and A* search by augmenting a heuristic function h in the weight function ($w(e) \leftarrow w(e) + h(e)$).

4. Sheaves II: Sheaf Pullbacks

We wish to expand upon section 5 of Sheaves for Routing in DTN [4], where they investigate the gluability of data structures. The language of sheaves motivates us to further study local path finding and how we can stitch together such data via fiber products of sheaves. This is the first step towards using past successes in sheaves to enable routing across network areas by defining these network area interfaces.

In particular, we investigate situations in which the global structure of our network G is not known. This takes the sheaf approach closer to reality, and also represents a departure from CGR which depends on globally distributed and consistent contact times. Suppose, for example, there were networks centered at Mars and around the Earth. Locally, these networks could feature discovery. However, the connections between these networks must be scheduled due to the propagation delays. It is not reasonable to require routers at the Earth to be absolutely aware of the the network state at Mars, but rather the routers should be able to get the message to Mars for local routers to figure out the "last mile" delivery.

Path sheaves

In what follows, we restrict our attention to networks that have a sequential nature, as Figure 6 depicts. While we have investigated sheaf-theoretic constructions that can be used in more general networks, those generalizations appear to lose the refinedness of the data captured by their global sections. We hope that our constructions here can help pave the way toward a full understanding of the utility of sheaves in modeling the transfer of data across arbitrary networks.

Definition 4.1. A **sequential network** is a directed graph G, with designated source node v_s and target node v_t, that admits a cover by subnetworks $G_1, \ldots, G_n$ such that

1. for all i, we have $G_i \cap G_{i+1}$ a set of nodes (no edges)
2. for i, j non-sequential, we have that $G_i \cap G_j$ is empty
3. the source node v_s is in $G_1 \setminus (G_1 \cap G_2)$
4. the target node v_t is in $G_n \setminus (G_{n-1} \cap G_n)$
5. for all i, there are no edges with tail in $G_i \cap G_{i+1}$ and head in G_i
6. for all i, there are no edges with tail in G_{i+1} and head in $G_i \cap G_{i+1}$

Example 4.2. An example sequential network is given in Figure 6. There are three subnetworks given by G_1, G_2, and G_3, with the source and target in G_1 and G_3 respectively. For emphasis, the nodes in the overlapping regions are colored in green whereas the other nodes are in blue.

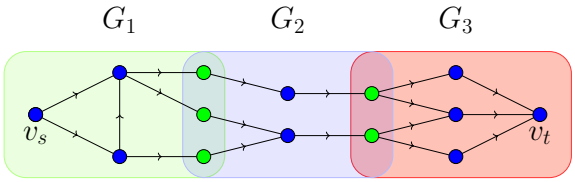


Figure 6. Example sequential network

The intuition here is that n different network areas could perform path finding on their local regions, to then find global paths from v_s to v_t.

But how can we keep track of when our path crosses between the various subnetworks? For this, we modify the definitions of the path sheaf, in order to obtain path sheaves with multiple sources and multiple targets, that can be defined on each subnetwork.

We start by assuming we have some directed graph H, a source set of nodes $S \subset V(H)$, each of which has no incoming edges, and a target set of nodes $T \subset V(H)$, each of which has no outgoing edges. Assume further that S and T are disjoint.

Definition 4.3. A **generalized path sheaf** $\mathcal{GP}$ on H is defined on stalks as follows

$$\mathcal{GP}(v) = \begin{cases} \text{Out}(v) \cup \{\bot\} & \text{if } v \in S \\ \text{In}(v) \cup \{\bot\} & \text{if } v \in T \\ (\text{In}(v) \times \text{Out}(v)) \cup \{\bot\} & \text{otherwise} \end{cases}$$
$$\mathcal{GP}(e) = \{\bot, \top\}$$

Restriction maps are defined exactly as for the path sheaf.

Remark 4.4. As with the original path sheaf, this sheaf has global sections that represent paths from source to target (with possibly disjoint cycles). However, there is now the additional possibility of global sections corresponding to a disjoint union of paths from the source set to the target set. Note that this disjoint union may be empty.

Lemma 4.5. *If a global section of a generalized path sheaf has only a single source node v_s that is not assigned $\bot$, then there is only a single target node that is not assigned $\bot$.*

Proof. Assume we have a global section σ such that for all $v \in T$, we have $\sigma(v) = \bot$. Then the path starting at v_s associated to σ would never terminate. But the graph is finite, so this is impossible. Hence there must be some $w \in T$ where σ is not valued as the off symbol. Now assume there was some other $w' \in T$ that was not off. Then there is some node u at which our path from v_s to T splits into two branches, contradicting the well-definedness of σ at u. □

Remark 4.6. The same is true if we swap the roles of source and target in the above lemma.

In order to glue together these generalized path sheaves across various subnetworks, we define an overlap sheaf on overlaps of subnetworks. For this, assume H is a graph consisting solely of a set of nodes.

Definition 4.7. The **overlap sheaf** $\mathcal{O}$ on H is given by

$$\mathcal{O}(v) = \{\bot, \top\}$$

We now take a sequential network and equip it with copies of the above sheaves to obtain what we call a routable network.

Definition 4.8. A **routable network** consists of the following data:

1. A sequential network $G = G_1 \cup \cdots \cup G_n$, with global source v_s, and target v_t

2. A generalized path sheaf $\mathcal{GP}_1$ on G_1 with source v_s and target set $G_1 \cap G_2$

3. A generalized path sheaf $\mathcal{GP}_n$ on G_n with target v_t and source set $G_{n-1} \cap G_n$

4. For all $1 < i < n$, a generalized path sheaf $\mathcal{GP}_i$ with source set $G_{i-1} \cap G_i$ and target set $G_i \cap G_{i+1}$

5. For all $1 \leq i < n$, an overlap sheaf $\mathcal{O}_{i,i+1}$ on $G_i \cap G_{i+1}$

Given a routable network, the sheaves $\mathcal{GP}_i$ and $\mathcal{GP}_{i+1}$ can be glued together to obtain a sheaf on $G_i \cup G_{i+1}$, namely the fiber product sheaf $\mathcal{GP}_i \times_{\mathcal{O}_{i,i+1}} \mathcal{GP}_{i+1}$.

Remark 4.9. In order to take a fiber product here, we really need sheaves that are defined on the same underlying network. This can be done explicitly by pushing forward our sheaves to G via the inclusions of the respective subnetworks. These push forwards are defined as off everywhere outside of the subnetworks, so we supress the push forward notation for ease of reading.

Remark 4.10. We also need to specify morphisms that this fiber product is taken with respect to.

We first construct the morphism

$$\mathcal{GP}_i \xrightarrow{\phi_i} \mathcal{O}_{i,i+1}$$

It must be trivial away from $G_i \cap G_{i+1}$, since $\mathcal{O}_{i,i+1}$ is valued by $\bot$ on such vertices and edges. For $v \in G_i \cap G_{i+1}$, we define the map in the stalk of v by

$$\phi_i(e) = \top$$
$$\phi_i(\bot) = \bot$$

Similarly, we define the morphism

$$\mathcal{GP}_{i+1} \xrightarrow{\phi_{i+1}} \mathcal{O}_{i,i+1}$$

in the stalk of $v \in G_i \cap G_{i+1}$ to be

$$\phi_{i+1}(e) = \top$$
$$\phi_{i+1}(\bot) = \bot$$

Theorem 4.11. *Let G be a routable network. Then the sheaf*

$$\mathcal{L} := \mathcal{GP}_1 \times_{\mathcal{O}_{1,2}} \cdots \times_{\mathcal{O}_{n-1,n}} \mathcal{GP}_n$$

has global sections in one to one correspondence with the set of paths from v_s to v_t (that have possibly disjoint cycles). In other words, $\mathcal{L}$ is the path sheaf for G.

Proof. For a node $v \in G_i$, that is not contained in G_{i-1} or G_{i+1} we have that

$$\mathcal{L}(v) = \bot \times_\bot \cdots \bot \times_\bot \mathcal{GP}_i(v) \times_\bot \bot \cdots \times_\bot \bot$$

which we identify with $\mathcal{GP}_i(v)$. Similarly, for edges $e \in G_i$, we have that

$$\mathcal{L}(e) = \mathcal{GP}_i(e)$$

For a node in a pairwise overlap $v \in G_i \cap G_{i+1}$, we have that

$$\mathcal{L}(v) = \bot \times_\bot \cdots \bot \times_\bot \mathcal{GP}_i(v) \times_{\mathcal{O}_{i,i+1}} \mathcal{GP}_{i+1}(v) \times_\bot \bot \cdots \times_\bot \bot$$

which we identify with $\mathcal{GP}_i(v) \times_{\mathcal{O}_{i,i+1}(v)} \mathcal{GP}_{i+1}(v)$. This tells us that a global section σ of $\mathcal{L}$ is a global section σ_i of each $\mathcal{GP}_i$, such that for nodes in the overlaps $G_i \cap G_{i+1}$, we have that $\phi_i(\sigma_i) = \phi_{i+1}(\sigma_{i+1})$. In other words, σ_i and σ_{i+1} are in agreement about which nodes in $G_i \cap G_{i+1}$ are on and off.

Now, let's focus on G_1. As was discussed previously, σ_1 must represent a path from v_s to a node of $G_1 \cap G_2$ (with possibly disjoint cycles). Call the path γ_1. On the other hand σ_2 could be a disjoint union of paths from $G_1 \cap G_2$ to $G_2 \cap G_3$. However, agreement of $\mathcal{GP}_1$ and $\mathcal{GP}_2$ on $G_1 \cap G_2$ forces

there to only be a single node of the overlap turned on, and hence σ_2 is a single path (with possibly disjoint cycles). Call the path γ_2. We have that γ_1 and γ_2 connect. Recursively continuing this process, we obtain a path from v_s to v_t (with possibly disjoint cycles).

On the other hand, take a path γ (with possibly disjoint cycles). Restricting to each G_i, we have a datum that fits the bill to be a global section of $\mathcal{GP}_i$, which we call σ_i. Agreement of σ_i and σ_{i+1} in $\mathcal{O}_{i,i+1}$ under the maps ϕ_i and ϕ_{i+1} comes from the fact that the path was connected to begin with, and hence we get a global section of $\mathcal{L}$. □

Distance path sheaves

We now consider weighted sequential networks.

Definition 4.12. A **weighted sequential network** is a sequential network such that all edges are weighted.

These weights can be thought of as some cost of traversing each edge.

We wish to define generalized distance path sheaves on each subnetwork that can be glued together. We start by assuming we have some weighted directed graph H, a source set of nodes $S \subset V(H)$, each of which has no incoming edges, and a target set of nodes $T \subset V(H)$, each of which has no outgoing edges. Further, assume S and T are disjoint. We emphasize that this construction allows for multiple sources and multiple targets.

Definition 4.13. A **generalized distance path sheaf** $\mathcal{GDP}$ on H is defined on stalks as follows

$$\mathcal{GDP}(v) = \begin{cases} \big(\text{Out}(v) \times \mathbb{R}^{\times}\big) \cup \{\perp\} & \text{if } v \in S \\ \big(\text{In}(v) \times \mathbb{R}^{\times}\big) \cup \{\perp\} & \text{if } v \in T \\ \big(\text{In}(v) \times \text{Out}(v) \times \mathbb{R}^{\times}\big) \cup \{\perp\} & \text{otherwise} \end{cases}$$

$$\mathcal{GDP}(e) = \mathbb{R}^{\times} \cup \perp$$

Restriction maps are defined exactly as for the distance path sheaf.

Remark 4.14. For the generalized distance path sheaf, the issue of disjoint cycles is resolved. (See Sheaves for Routing in DTN [4] for why this is the case.)

As in the previous section, we need a notion of overlap sheaf in order to glue such sheaves together. Let H be a graph consisting solely of a set of nodes.

Definition 4.15. The **distance overlap sheaf** $\mathcal{DO}$ on H is given by

$$\mathcal{DO}(v) = \{\perp, \mathbb{R}^{\times}\}$$

Definition 4.16. A **weighted routable network** is a routable network, where the network is weighted, and all path sheaves are replaced by distance path sheaves, and overlap sheaves are replaced by distance overlap sheaves.

In order to glue our sheaves together, we again need to construct morphisms to the overlap sheaves

$$\mathcal{GDP}_i \xrightarrow{\phi_i} \mathcal{DO}_{i,i+1}$$

Now, this map must be trivial away from $G_i \cap G_{i+1}$, since $\mathcal{DO}_{i,i+1}$ is valued by $\perp$ on such vertices and edges. For $v \in G_i \cap G_{i+1}$, we define the map in the stalk of v by

$$\phi_i((e,a)) = a$$
$$\phi_i(\perp) = \perp$$

Similarly, we define the morphism

$$\mathcal{GDP}_{i+1} \xrightarrow{\phi_{i+1}} \mathcal{DO}_{i,i+1}$$

in the stalk of $v \in G_i \cap G_{i+1}$ to be

$$\phi_{i+1}((e,a)) = a$$
$$\phi_{i+1}(\perp) = \perp$$

Theorem 4.17. *Let G be a weighted routable network. Then the sheaf*

$$\mathcal{DL} := \mathcal{GDP}_1 \times_{\mathcal{DO}_{1,2}} \cdots \times_{\mathcal{DO}_{n-1,n}} \mathcal{GDP}_n$$

has global sections in one to one correspondence with the set of paths from v_s to v_t that record total distance traveled at the target node. In other words, $\mathcal{DL}$ is the distance path sheaf for G.

Proof. The argument is nearly identical to the theorem from the previous section, so we only provide a sketch here. We find that a global section σ of $\mathcal{DL}$ is a global section σ_i of each $\mathcal{GDP}_i$, such that for nodes in the overlaps $G_i \cap G_{i+1}$, we have that $\phi_i(\sigma_i) = \phi_{i+1}(\sigma_{i+1})$. In other words, σ_i and σ_{i+1} are in agreement about which nodes in $G_i \cap G_{i+1}$ are not off, and what real values are stored at them.

As before, we start by looking at G_1, for which σ_1 must represent a path from v_s to a node of $G_1 \cap G_2$. Agreement of $\mathcal{GDP}_1$ and $\mathcal{GDP}_2$ on $G_1 \cap G_2$ forces there to only be a single node of the overlap turned on, and so σ_2 must be a single path from source to target in G_2, valued by the same real number at the node where it meets the path coming from σ_1. Recursively continuing this process, we obtain a path from v_s to v_t.

The argument that a path gives rise to a global section follows as in the proof of the theorem in the previous section. □

Carry-over Sheaf

In this section, we consider a special kind of weighted sequential network.

Definition 4.18. A **weighted traceable sequential network** is a weighted sequential network where each node v in a subnetwork G_i can only backtrack to a single node of $G_{i-1} \cap G_i$.

Restricting our attention to such networks allows us to define new sheaves with more refined data in their global sections. More specifically, we construct a sheaf on such a network whose global sections keep track of the costs of traversing each subnetwork separately.

For a weighted traceable sequential network

$$G = G_1 \cup \cdots \cup G_n,$$

we will define a **carry-over sheaf** associated to each subnetwork. However, the i-th carry-over sheaf will be defined on $H_i = G_i \cup \cdots \cup G_n$ (instead of just on G_i). As with the previously defined sheaves, it will have source set $S = G_{i-1} \cap G_i$ and target set $T = G_i \cap G_{i+1}$. But now, since H_i extends beyond the target set, we will have information that is passed further into the network.

Remark 4.19. For G_1, we take $S = \{v_s\}$. For G_n, we take T to be the specified global target set.

Remark 4.20. The construction here will be so that the i-th carry-over sheaf only has information about weights from G_i despite being defined on all of H_i. We are modeling a situation where weights are local, even if unweighted graph information is known more widely across G.

Definition 4.21. A **carry-over sheaf** $\mathcal{C}_i$ on H_i is defined on stalks as follows

$$\mathcal{C}_i(v) = \begin{cases} (\text{Out}(v) \times \mathbb{R}^\times) \cup \{\bot\} & \text{if } v \in S \\ (\text{In}(v) \times \mathbb{R}^\times) \cup \{\bot\} & \text{if } v \in T \\ (\text{In}(v) \times \text{Out}(v) \times \mathbb{R}^\times) \cup \{\bot\} & \text{if } v \in G_i \setminus (S \cup T) \\ \mathbb{R}^\times \cup \{\bot\} & \text{otherwise} \end{cases}$$

$$\mathcal{C}_i(e) = \mathbb{R}^\times \cup \{\bot\}$$

Within G_i, restriction maps are defined exactly as for the generalized distance path sheaf. Outside of G_i, all restriction maps are the identity maps. When restricting from a node of T to an edge outside of G_i, we define

$$\mathcal{C}_i(v < e)((f, a)) = a$$
$$\mathcal{C}_i(v < e)(\bot) = \bot$$

Often data are organized into *tuples*, which are ordered sequences of data. The carry-over sheaf enables a construction where sheaf-gluing produces these tuples. These tuples then represent organized information about the hops through the network areas, that is, they can produce insight into addressing.

Definition 4.22. A **tuply routable network** consists of the following data

1. A weighted traceable sequential network
$$G = G_1 \cup \cdots \cup G_n,$$
with source $v_s \in G_1$, and target set $T \in G_n$
2. For each G_i, a carry-over sheaf $\mathcal{C}_i$ defined on H_i
3. For all $1 \leq i < n$, an overlap sheaf $\mathcal{DO}_{i,i+1}$ on $G_i \cap G_{i+1}$

Again in order to glue our sheaves together, we need to construct morphisms to the distance overlap sheaves

$$\mathcal{C}_i \xrightarrow{\phi_i} \mathcal{DO}_{i,i+1}$$

As before, this map is trivial away from $G_i \cap G_{i+1}$, since $\mathcal{DO}_{i,i+1}$ is value by $\bot$ on such vertices and edges. On $G_i \cap G_{i+1}$ we define the map exactly as in the previous section.

Theorem 4.23. *Let G be a tuply routable network. Then the sheaf*
$$\mathcal{L} := \mathcal{C}_1 \times_{\mathcal{DO}_{1,2}} \cdots \times_{\mathcal{DO}_{n-1,n}} \mathcal{C}_n$$
has global sections in one to one correspondence with the set of paths from v_s to T. Such global sections are valued on exactly one target node $v_t \in T$ with the data $(a_1, a_2, \ldots, a_{n-1}, (e, a_n))$ of an incoming edge e, and the distances a_i travelled on each G_i separately.

Remark 4.24. A proof of this follows in the same vein as the previous two theorems, albeit with some less clean-looking fiber products. For this reason, we find it more instructive to go through an example here rather than a proof.

In figure 7 below, we see an example of a weighted traceable sequential network with three subnetworks highlighted in green, blue, and red respectively.

Just beneath that is an example of a global section of the pullback sheaf $\mathcal{L}$. The purple highlighting indicates the portion of the network that is not off with respect to this global section. The path associated to this global section is (e_1, f_1, f_4, g_2). The target node v_{t_2} is valued with information about distances travelled in each subnetwork.

Remark 4.25. There are also additional nodes and edges turned on throughout the network that are not on the path. The information stored at these edges and nodes is only keeping track of distances traveled in previous subnetworks. We can think of this information as accesible to anyone within reach.

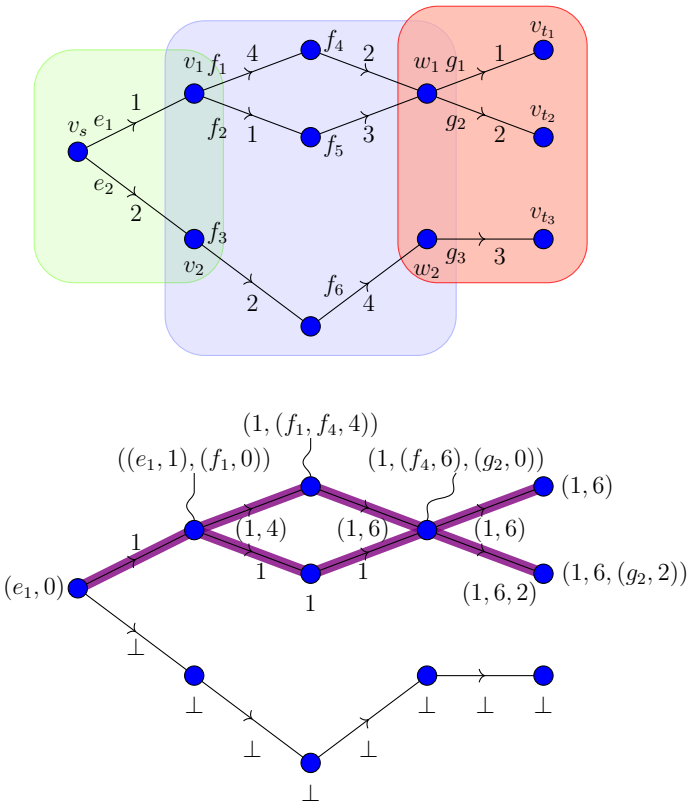


Figure 7. Network with multiple global targets

In figure 8 below, we have another example of a weighted traceable sequential network, this time with only a single global target node. Beneath that is an example of a global section of the pullback sheaf $\mathcal{L}$, with associated path given by $(e_1, e_3, e_5, f_1, f_4)$. At the target node, we have the infor-

mation of total distance traveled on G_1, and total distance traveled on G_2, recorded separately.

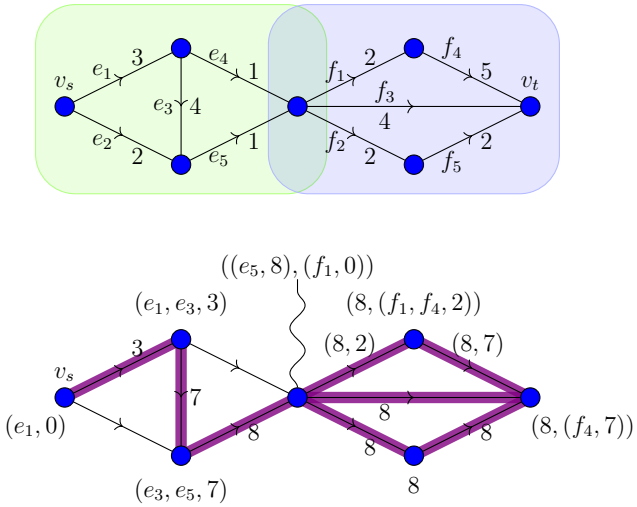


Figure 8. Network with a single global target

We emphasize that in these final examples, the data tracked by the sections could distill structure, such as addresses, with further research.

5. Sheaves III: Higher Machinery

In this section we investigate further applications of cellular sheaves to graphs and hypergraphs. First we explore sheaf constructions for hypergraphs and their associated graphs. In particular, we show how to encode s-walks and paths in a hypergraph as global sections of certain sheaves on the s-line graph and bipartite graph, respectively. Then we explore sheaf cohomology and sheaf Laplacians for sheaves on graphs, with applications to the path sheaf.

Sheaves and Hypergraphs

One way to construct sheaves on a hypergraph H is to instead construct sheaves on its associated graphs $L_s(H)$ and $B(H)$. Recall the definition of the path sheaf $\mathcal{P}$ from before. Choose source and target vertices $v_S, v_T \in V_{L_s(H)}$. Then global sections of the path sheaf $\mathcal{P}$ correspond to paths and unions of paths with cycles in $L_s(H)$. Now, each vertex of $L_s(H)$ corresponds to a hyperedge in H. Let e_S and e_T be hyperedges in H corresponding to v_S and v_T, respectively. Since paths in $L_s(H)$ correspond to sequences of hyperedges in H, we see that global sections of $\mathcal{P}$ over $L_s(H)$ can be interpreted as s-paths in H from e_S to e_T, or unions of such with s-cycles of hyperedges which are disjoint from e_S and e_T.

Suppose H is a directed hypergraph. Then the bipartite graph $B(H)$ is a directed graph. Choose source and target vertices $v_S, v_T \in V_{B(H)}$. Then the path sheaf $\mathcal{P}$ over $B(H)$ records paths in $B(H)$ from v_S to v_T, and unions of such with cycles disjoint from v_S and v_T. A path in $B(H)$ is specified by a sequence of distinct vertices $v_1 = v_S, v_2, \ldots, v_n = v_T$, with v_i adjacent to v_{i+1} for each i. Since $B(H)$ is bipartite, the vertices in this sequence alternate between corresponding to a vertex in H and corresponding to a hyperedge in H.

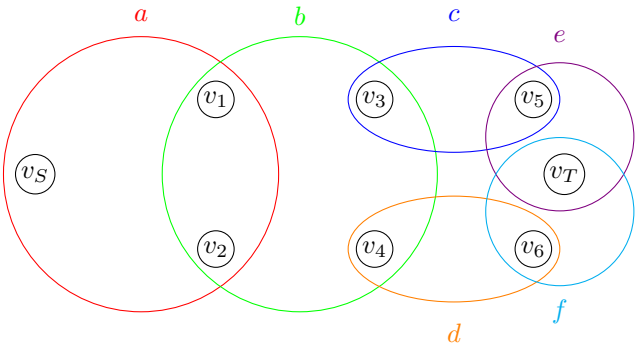


Figure 9. The hypergraph H, referred to in Example 5.1.

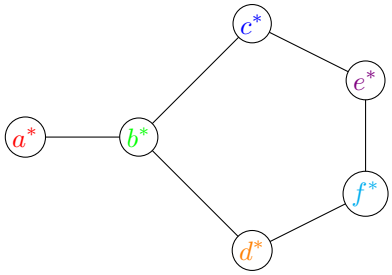


Figure 10. The line graph $L_1(H)$ associated to the hypergraph H in Figure 9.

Therefore a path in $B(H)$ corresponds precisely to a path in H, which is given by an alternating sequence of vertices and hyperedges. So global sections of $\mathcal{P}$ over $B(H)$ correspond to paths in H from v_S to v_T, and unions of such with disjoint cycles.

Example 5.1. Consider the hypergraph H in Figure 9. Suppose we want to encode all s-walks from the hyperedge a to the hyperedge e. We will apply the path sheaf to the s-line graph for H. In Figure 10 the 1-line graph of H is shown. Note that we take $v_S = a^*$ and $v_T = e^*$. Then the path sheaf on the line graph encodes all paths in the line graph from a^* to e^*, and each of these corresponds to a 1-walk in H from a to e.

Now suppose we want to encode paths in H from the vertex v_1 to the vertex v_8. In this case, we apply the path sheaf to the bipartite graph $B(H)$, which is shown in Figure 11.

Global sections of the path sheaf on $B(H)$ correspond to paths in $B(H)$ from $v_S = v_1$ to $v_T = v_8$. Such a path

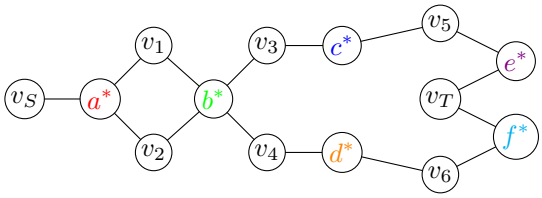


Figure 11. The bipartite graph $B(H)$ associated to the hypergraph H in Figure 9.

can be uniquely specified by a list of distinct vertices, such that successive vertices in the list are adjacent in $B(H)$. For example, consider the path in $B(H)$ given by the sequence $v_1, a^*, v_2, b^*, v_4, c^*, v_6, e^*, v_7$. This corresponds to the sequence $v_1, a, v_2, b, v_4, c, v_6, e, v_7$ in the original hypergraph H, and this sequence uniquely determines a path from v_1 to v_8.

The sheafy constructions for hypergraphs that we have just described involves graphs associated to a hypergraph. The reader may wonder if it is possible to construct sheaves on hypergraphs directly. This is a deep question which lacks a singular answer. One issue is that cellular sheaf constructions depend on the topology of the underlying space. For a hypergraph, there is no canonical choice of topology.

Given a hypergraph H, it is possible to topologize the set $V_H \cup E_H$ such that we can effectively port the path sheaf constructions on $L_s(H)$ or $B(H)$ to H itself. There are potentially many ways to topologize $V_H \cup E_H$, and an interesting research direction is to explore sheaf constructions for different topologies.

On the other hand, it is possible that there is a natural choice of topology and sheaf over a hypergraph, in some suitable sense. Graphs and hypergraphs can be viewed as algebraic varieties [12]. Hence we can view hypergraphs more generally as schemes. A scheme comes equipped with a structure sheaf, so we can get our hands on a sheaf associated to the hypergraph. This sheaf may not have immediate use for routing problems, but it is a first step towards finding such a sheaf.

Sheaf Cohomology and Laplacian

Sheaf Cohomology—Consider a finite directed graph G. Let $\mathcal{F}$ be a sheaf of vector spaces over G. That is, $\mathcal{F}$ is a cellular sheaf on G whose stalks are finite-dimensional vector spaces and whose restriction maps are linear maps. We will assume that all of our vector spaces real.

Given a vertex $v \in V_G$ and edge $e \in E_G$ which is incident to v, we define the **index** $[v : e]$ of v with respect to e by

$$[v : e] = \begin{cases} 1 & \text{if } h(e) = v, \\ -1 & \text{if } t(e) = v. \end{cases}$$

Denote by $C^0(G; \mathcal{F})$ the vector space

$$C^0(G; \mathcal{F}) = \bigoplus_{v \in V_G} \mathcal{F}(v)$$

and by $C^1(G; \mathcal{F})$ the vector space

$$C^1(G; \mathcal{F}) = \bigoplus_{e \in E_G} \mathcal{F}(e).$$

We refer to $C^i(G; \mathcal{F})$ as the space of **i-cochains.** Define a linear map $d : C^0(G; \mathcal{F}) \to C^1(G; \mathcal{F})$ by

$$d|_{\mathcal{F}(v)} = \sum_{e \ni v} [v : e] \mathcal{F}(v \leq e).$$

Definition 5.2. The **sheaf cohomology groups** are defined to be

$$H^0(G; \mathcal{F}) = \ker(d), \quad H^1(G; \mathcal{F}) = \operatorname{coker}(d).$$

It is a basic fact that $H^0(G; \mathcal{F})$ is equal to the set of global sections of $\mathcal{F}$.

Sheaf Laplacians—For sheaves on graphs, sheaf cohomology on its own does not provide much insight. For example, $H^1(G; \mathcal{F})$ is determined by $H^0(G; \mathcal{F})$ by the Rank-Nullity Theorem from linear algebra. Hence $H^1(G; \mathcal{F})$ does not provide any new information. However, we can use the coboundary operator $d : C^0(G; \mathcal{F}) \to C^1(G; \mathcal{F})$ to define the *sheaf Laplacian*, which is a generalization of the well-studied graph Laplacian.

Definition 5.3. Let G be a directed graph and $\mathcal{F}$ a sheaf of vector spaces over G. The **sheaf Laplacian** $\Delta_{\mathcal{F}} : C^0(G; \mathcal{F}) \to C^0(G; \mathcal{F})$ is a linear operator given by the formula $\Delta_{\mathcal{F}} = d^T d$, where d^T denotes the transpose of d. If the sheaf $\mathcal{F}$ is clear from context, we will simply write Δ for the sheaf Laplacian.

The sheaf Laplacian has the property that $\ker(\Delta_{\mathcal{F}}) = H^0(G; \mathcal{F})$, so that the kernel of $\Delta_{\mathcal{F}}$ is equal to the set of global sections of $\mathcal{F}$.

Vector Space-ifying a Sheaf of Sets—Let $\mathbb{F} : \textbf{Set} \to \mathbb{R} - \textbf{Vect}$ denote the functor which takes each set X to the free $\mathbb{R}$-vector space consisting of formal linear combinations of elements of X. The functor $\mathbb{F}$ sends each map of sets $f : X \to Y$ to the linear map $\mathbb{F}(f) : \mathbb{F}(X) \to \mathbb{F}(Y)$ given by

$$\sum_{x \in X} c_x \vec{e}_x \mapsto \sum_{x \in X} c_x \vec{e}_{f(x)}.$$

We can apply $\mathbb{F}$ to a cellular sheaf of sets to obtain a cellular sheaf of vector spaces.

Proposition 5.4. *Let G be a finite directed graph, and let $\mathcal{F} : V_G \cup E_G \to$ **Set** be a cellular sheaf of sets over G. Let $\mathcal{F}(G)$ be the equalizer of the diagram*

$$\prod_{v \in V_G} \mathcal{F}(v) \rightrightarrows \prod_{e \in E_G} \mathcal{F}(e)$$

and let K be the equalizer of the diagram

$$\mathbb{F}\left(\prod_{v \in V} \mathcal{F}(v)\right) \rightrightarrows \mathbb{F}\left(\prod_{e \in E} \mathcal{F}(e)\right)$$

that we get by applying the free functor. Then $\mathbb{F}(\mathcal{F}(G)) \subseteq K$. We note that K can be interpreted as the kernel of the linear map $\mathbb{F}(h) - \mathbb{F}(t)$.

Every linear combination of sections of $\mathcal{F}$ is an element of K. However, there may be elements of K which do not arise as linear combinations of sections of $\mathcal{F}$. For example, we can consider the vector path sheaf $\mathbb{F}(\mathcal{P})$, defined in the next section, for the graph in Figure 13. See Example 5.6 for details.

Path Sheaf

We apply the free functor to the path sheaf, obtaining a cellular sheaf of vector spaces which we denote by $\mathbb{F}(\mathcal{P})$. To simplify computations, we make a slight modification to the definition of $\mathbb{F}(\mathcal{P})$. Namely, we will replace the symbol "$\perp$" with the zero vector in each stalk of $\mathbb{F}(\mathcal{P})$.

Example 5.5. Consider the graph G in Figure 12. A choice of source vertex v_S and target vertex v_T has been made, and the corresponding vector path sheaf $\mathbb{F}(\mathcal{P})$ is shown below.

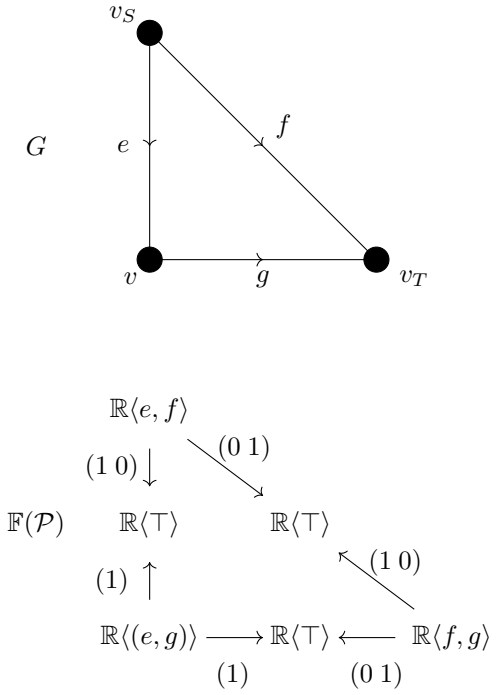


Figure 12. A directed graph G with source v_S and target v_T, along with the corresponding vector path sheaf $\mathbb{F}(\mathcal{P})$.

Each restriction map is a linear map, and is labeled by the corresponding matrix.

We compute the sheaf cohomology of $\mathbb{F}(\mathcal{P})$ for the graph G as follows. The differential d is a linear map

$$d : \mathbb{R}\langle e, f\rangle \oplus \mathbb{R}\langle(e, g)\rangle \oplus \mathbb{R}\langle f, g\rangle \to \mathbb{R}\langle\top\rangle^3$$

and is given by

$$d = \begin{pmatrix} -1 & 0 & 1 & 0 & 0 \\ 0 & -1 & 0 & 1 & 0 \\ 0 & 0 & -1 & 0 & 1 \end{pmatrix}.$$

This is a rank 3 matrix with 2-dimensional kernel. Hence $H^0(G; \mathbb{F}(\mathcal{P})) \cong \mathbb{R}^2$ and $H^1(G; \mathbb{F}(\mathcal{P})) = 0$. A basis for the kernel of d is

$$\left\{ \begin{pmatrix} 0 \\ 1 \\ 0 \\ 1 \\ 0 \end{pmatrix}, \begin{pmatrix} 1 \\ 0 \\ 1 \\ 0 \\ 1 \end{pmatrix} \right\}.$$

Each of these basis vectors corresponds to a path in G from v_S to v_T. The first basis vector corresponds to the path which goes through f, while the second basis vector corresponds to the path through e and g. Note that these are the only two paths between v_S and v_T, so that in this case the dimension of $H^0(G; \mathbb{F}(\mathcal{P}))$ counts the number of paths in G from source to target.

The sheaf Laplacian in this example is given by

$$\Delta = d^T d = \begin{pmatrix} 1 & 0 & -1 & 0 & 0 \\ 0 & 1 & 0 & -1 & 0 \\ -1 & 0 & 2 & 0 & -1 \\ 0 & -1 & 0 & 1 & 0 \\ 0 & 0 & -1 & 0 & 1 \end{pmatrix}.$$

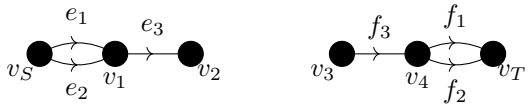


Figure 13. The path sheaf $\mathcal{P}$ over G has no global sections, even though the vector space path sheaf $\mathbb{F}(\mathcal{P})$ admits non-zero global sections.

Example 5.6. The graph G in Figure 13 has two connected components. The source v_S and target v_T are in separate connected components. Furthermore, there are no cycles in G which are disjoint from v_S and v_T. Therefore the path sheaf $\mathcal{P}$ has no global sections at all. The differential d on the vector path sheaf $\mathbb{F}(\mathcal{P})$ is given by

$$d = \begin{pmatrix} -1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \end{pmatrix}.$$

This matrix has a 2-dimensional kernel, with basis

$$\left\{ \begin{pmatrix} -1 \\ 1 \\ -1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ -1 \\ 1 \\ -1 \\ 1 \end{pmatrix} \right\}.$$

Thus $H^0(G; \mathbb{F}(\mathcal{P})) \cong \mathbb{R}^2$. Since $H^0(G; \mathbb{R}(\mathcal{P}))$ is equal to the set of global sections of $\mathbb{F}(\mathcal{P})$, this shows that $\mathbb{F}(\mathcal{P}))$ admits non-zero global sections, even though $\mathcal{P}$ has no global sections at all. This shows that when we convert a sheaf of sets into a sheaf of vector spaces, we may introduce "extra" global sections.

Sheaf Diffusion

The vector space $C^0(G; \mathbb{F}(\mathcal{P}))$ of 0-cochains of the vector path sheaf over a graph G is isomorphic to some finite-dimensional real vector space. As such, we may endow $C^0(G; \mathbb{F}(\mathcal{P}))$ with the usual inner product and its induced metric. Thus we have a notion of distance between two 0-cochains.

The **diffusion equation** is the linear ordinary differential equation

$$\dot{x} = -\Delta x \tag{1}$$

where x is an element of $C^0(G; \mathbb{F}(\mathcal{P}))$. The diffusion equation models $C^0(G; \mathbb{F}(\mathcal{P}))$ as a continuous-time dynamical system. This dynamical system is local in the sense that the value of $\dot{x}$ over a vertex v is only influenced by the values of x over vertices adjacent to v. The following proposition is proved in [13]; see also [14].

Proposition 5.7. *Let G be a digraph and $\mathcal{F}$ be a sheaf of vector spaces on G. The dynamical system modeled by Equation 1 has $H^0(G; \mathcal{F})$ as its space of equilibria. Furthermore, the trajectory of the dynamical system, initialized at*

$x_0 \in C^0(G;\mathcal{F})$, converges to the global section of $\mathcal{F}$ which is nearest to x_0.

When $\mathcal{F} = \mathbb{F}(\mathcal{P})$, Proposition 5.7 gives us a method for finding paths in a digraph G: given any selection $x_0 \in C^0(G;\mathbb{F}(\mathcal{P}))$, we solve Equation 1 with initial condition x_0. The solution is guaranteed to be a global section of $\mathbb{F}(\mathcal{P})$ by Proposition 5.7. One issue with this approach is that, as we have already shown, $\mathbb{F}(\mathcal{P})$ may admit non-trivial global sections which do not correspond to paths. Later we will see that this issue can be circumvented in the context of distributed optimization by introducing constraints. Another method for finding true global sections, utilizing gradient descent, is outlined in [4].

Example 5.8. Consider the graph G and its vector path sheaf shown in Figure 12. Let x_0 be the 0-cochain such that $(x_0)_{v_S} = e$, $(x_0)_v = 0$ and $(x_0)_{v_T} = g$. If we solve Equation 1 with initial condition x_0, we obtain the 0-cochain $\mathbf{x}$ given by $\mathbf{x}_{v_S} = e$, $\mathbf{x}_v = (e, g)$, and $\mathbf{x}_{v_T} = g$. Then $\mathbf{x}$ corresponds to the vector $(1, 0, 1, 0, 1)$, which is an element of $\ker(\Delta) = H^0(G;\mathbb{F}(\mathcal{P}))$. Hence $\mathbf{x}$ uniquely determines a global section of $\mathbb{F}(\mathcal{P})$. By Proposition 5.7, $\mathbf{x}$ minimizes the distance to x_0.

Sheaf diffusion can also be used to extend local cochains to global cochains. Let $U \subset V_G$ be a collection of vertices. The **restricted diffusion equation**, relative to U, is given by

$$\frac{dx}{dt}\Big|_v = \begin{cases} -\Delta x, & \text{if } v \notin U \\ 0, & \text{else} \end{cases}. \tag{2}$$

Solutions to Equation 2, with initial condition $x_0 \in C^0(U;\mathcal{F})$, are called **harmonic extensions** of x_0. Note that a harmonic extensions need not be global sections, and they need not be unique. Indeed, there exists a unique harmonic extension of x_0 if and only if $H^1(G, U;\mathcal{F}) = 0$ [14].

Example 5.9. In the context of vector path sheaves, the restricted diffusion equation can be used to extend a path. For example, we once again consider the graph G in Figure 12. Let $U = \{v_S\}$ and $x_0 \in C^0(U;\mathbb{F}(\mathcal{P})$ given by $(x_0)_{v_S} = e$. We may view x_0 as a partial path from v_S to v_T, consisting of the edge e. In this case $H^1(G, U;\mathbb{F}(\mathcal{P})) = 0$, so that there is a unique harmonic extension $\mathbf{x}$ of x_0 to all of V_G. The harmonic extension $\mathbf{x}$ is given by $\mathbf{x}_{v_S} = (x_0)_S = e$, $\mathbf{x}_v = (e, g)$, and $\mathbf{x}_{v_T} = g$. In this case, $\mathbf{x}$ is also a global section.

On the other hand, suppose $U = \{v\}$ and $x_0 = (e, g)$. Now we have $H^1(G, U;\mathbb{F}(\mathcal{P})) \neq 0$, and in fact there are infinitely many harmonic extensions of x_0. The set of harmonic extensions of x_0 is given by

$$\begin{cases} \mathbf{x}_{v_S} = e + rf \\ \mathbf{x}_v = (e, g) \\ \mathbf{x}_{v_T} = g + rf \end{cases}$$

where $r \in \mathbb{R}$ is arbitrary.

Laplacians and Optimization

Optimization problems over graphs written in terms of graph Laplacian has been an active field of research. Sheaf Laplacians have a benefit over traditional graph Laplacians of allowing an optimization problem to be written in 'sheafy' way. Its potential of being useful for formulating and solving distributed optimization problems has been discussed in papers by Hansen and Ghrist such as [15]. In particular, [15] converts a problem of constrained minimization of an objective function over global sections into a problem of finding saddle points of the Lagrangian function, then find its saddle point using Karush-Kuhn-Tucker (KKT) optimality condition.

Shortest path problem over a graph $G = (V, E)$ can be written as a form of optimization problem. For example, in a dynamic programming form, the problem can be written as solving the following recurrence equation when $i = n = |V|$ and $v = v_T$:

$$\begin{aligned} &\mathrm{OPT}(v, i) \\ &= \min\left(\mathrm{OPT}(v, i-1), \min_{\substack{v' \text{ s.t.} \\ e=(v,v')}} \mathrm{OPT}(v', i-1) + w(e)\right) \end{aligned} \tag{3}$$

where $\mathrm{OPT}(v, i)$ represents the minimum cost of a path from v_S to v using (going through) at most i edges. In fact, this is the motivating building block of the celebrated Bellman-Ford algorithm. This optimization problem then can be translated into a simple linear programming problem

$$\begin{aligned} &\text{minimize} \quad \sum_{e \in E} w(e) x_e \\ &\text{subject to} \quad x_e \geq 0 \quad \forall e \in E \\ &\sum_{u \text{ s.t. } (v,u) \in E} x_{(v,u)} \leq 1 \qquad \forall v \in V \\ &\sum_{\substack{u \text{ s.t.} \\ (v,u) \in E}} x_{(v,u)} - \sum_{\substack{u \text{ s.t.} \\ (u,v) \in E}} x_{(u,v)} = \begin{cases} 1 & \text{if } v = v_S \\ -1 & \text{if } v = v_T \\ 0 & \text{otherwise} \end{cases} \quad \forall v \in V \end{aligned}$$

where $x_e \in \{0, 1\}$ is a boolean indicator variable denoting whether it was chosen ($x_e = 1$) as a part of path or not ($x_e = 0$). The first constraint is clear, the second is about the out-degree of every node on the path being one, and the third is for ensuring that the flow is conserved, and also prevents a creation of loops.

The shortest path problem can be reformulated using the sheaf Laplacian as follows. Consider the vector path sheaf $\mathbb{F}(\mathcal{P})$ over G. For each vertex $v \in V_G$, we compute a local objective function f_v by summing the weights on the outgoing edges incident to v. That is, if $v = v_S$, then

$$f_{v_S}\left(\sum_{e \in \mathrm{Out}(v_S)} \alpha_e e\right) = \sum_{e \in \mathrm{Out}(v_S)} \alpha_e w(e).$$

If $v \neq v_S$ and $v \neq v_T$, then we set

$$\begin{aligned} &f_v\left(\sum_{(e,f) \in \mathrm{In}(v) \times \mathrm{Out}(v)} \alpha_{(e,f)}(e, f)\right) \\ &= \sum_{(e,f) \in \mathrm{In}(v) \times \mathrm{Out}(v)} \alpha_{(e,f)} w(f). \end{aligned}$$

We set $f = \sum_{v \in V_G} f_v$, thereby obtaining a global objective function from the local objective functions.

Our plan is to minimize f over the space of 0-cochains $C^0(G;\mathbb{F}(\mathcal{P}))$. To ensure that we are optimizing over global sections of $\mathbb{F}(\mathcal{P})$, we introduce the constraint $\Delta x = 0$, since $\ker(\Delta)$ is the space of global sections. In addition, since we want to minimize f for *paths* in G, we need to exclude global sections of $\mathbb{F}(\mathcal{P})$ which do not correspond to paths. This is taken care of by adding appropriate constraints. We claim that solutions to the following optimization problem are shortest paths between v_S and v_T. We consider $x \in C^0(G;\mathbb{F}(\mathcal{P}))$, and denote by x_v the value of x over the vertex v.

$$\begin{aligned}
\text{minimize} \quad & f(x) \\
\text{subject to} \quad & \Delta(x) = 0 \\
& x_v \geq 0 \quad \forall v \in V \\
& x \in \mathbb{Z}^{\dim(C^0(G;\mathbb{F}(\mathcal{P})))} \\
& ||x_v||^2 \leq 1 \quad \forall v \in V \\
& ||x_{v_S}||^2 = 1 \\
& ||x_{v_T}||^2 = 1
\end{aligned}$$

When we write $x_v \geq 0$, we mean that each entry of x_v is non-negative (recall that x_v is a vector). The statement $||x_v||^2 \leq 1$ can be interpreted as requiring that each component of the vector x_v is at most 1. This, along with the requirement that x is an integer vector, implies that each x_v has exactly one non-zero entry, which must be 1. In addition, we require that x_{v_S} and v_{v_T} must have a non-zero entry, which is also equal to 1. These constraints eliminate all "extra" sections that may have been introduced by the free functor. This is because all such sections contain negative entries. Our constraints are designed to single out sections of $\mathcal{P}$. Recall that these include disjoint unions of paths and cycles. The linear optimization problem will avoid sections containing cycles. This is because the problem is to minimize $f(x)$, and $f(x)$ sums the weights of the edges that are "turned on" by x. A global section which is the union of a path and a cycle will never minimize f: if we throw out the cycle, then f only sums edge weights along the path, and thus we get a smaller value for f.

We have shown that the shortest path problem can be stated as an optimization problem using sheaf Laplacians. Other problems can also be rephrased using sheaf Laplacians. We briefly discuss this point in the Future Work section.

6. TEMPORAL NETWORKS

The time dependency of any solar system network drives much of the need for the generalizations are given in this and related papers. Essentially, the familiar structures and constructs placed on top of static graphs (and hence static networks) do not cover the temporal case. The approach taken here, however, does; in this section, we formalize a particular path toward modeling temporal networks.

Models of temporal networks in discrete time include sequences of static graphs and time-expanded graphs. Continuous time models of temporal networks include contact graphs as well as edge and vertex-labelled graphs.

Definition 6.1. A **temporal**, or **time-varying**, **(un)directed graph** $T = (V_T, E_T, f, g)$ consists of a vertex set V_T, an edge multiset E_T, an edge labelling $f : E_T \to 2^{\mathbb{R}_{\geq 0}}$, and a vertex labelling $g : V_T \to 2^{\mathbb{R}_{\geq 0}}$, where $2^{\mathbb{R}_{\geq 0}}$ is the power set of $\mathbb{R}_{\geq 0}$.

For every edge $e \in E_T$, the edge label $f(e)$ specifies the times when the edge e is active or available. For every vertex $v \in V_T$, the vertex label $g(v)$ specifies the times when the vertex v is available.

We will first consider a directed temporal graph T for which the vertices are available at all times, and only the edges vary. That is, $\forall v \in V_T, g(v) = \mathbb{R}_{\geq 0}$. We may use a temporal graph of this kind to model a temporal network for which the set of nodes is fixed and the links are time-varying. Let $v_s \in V_T$ be a specified source vertex and $v_t \in V_T \setminus \{v_s\}$ be a specified target vertex. We define a path sheaf $\mathcal{F}$ over T stalk-wise as follows

$$\mathcal{F}(v) = \begin{cases} \{\perp\} \times \text{Out}(v) \times \mathbb{Z}_{\geq 0} \times \{0\} \times \{0\} & \text{if } v = v_s \\ \text{In}(v) \times \{\perp\} \times \{0\} \times \mathbb{Z}_{\geq 0} \times \mathbb{Z}^+ & \text{if } v = v_t \\ \big(\text{In}(v) \times \text{Out}(v) \times \mathbb{Z}_{\geq 0} \times \mathbb{Z}_{\geq 0} \times \mathbb{Z}^+\big) & \\ \cup \{\perp\} & \text{otherwise} \end{cases}$$

$$\mathcal{F}(e) = \big(\mathbb{Z}_{\geq 0} \times \mathbb{Z}^+\big) \cup \{\perp\}$$

Let $x = (e_i, e_o, w, t, d)$ or $\perp$. Then the restriction maps are defined as follows

$$\mathcal{F}(v \leq e)(x) = \begin{cases} (w, 1) & \text{if } e_i = \perp, e = e_o, \\ & \text{and } w \in f(e) \\ (t, d) & \text{if } e = e_i \text{ and } t \in f(e) \\ (t + w, d + 1) & \text{if } e_i \neq \perp, e = e_o, \\ & \text{and } t + w \in f(e) \\ \perp & \text{otherwise} \end{cases}$$

The global sections of $\mathcal{F}$ correspond to temporal paths from the source vertex v_s to the target vertex v_t over the temporal graph T. These in turn correspond to non-self intersecting routes across the temporal network from the source node v_s to the target node v_t.

Remark 6.2. $\mathcal{F}$ is analogous to the distance path sheaf over a static graph. For a given global section of $\mathcal{F}$, if v is vertex that lies on the path that corresponds to the global section, then the data assigned to v, $\mathcal{F}(v)$, includes the number of hops from v_s to v along the path and the time elapsed to arrive at v. Tracking the number of hops ensures that global sections of $\mathcal{F}$ do not include the disjoint union of temporal cycles in T. As with distance path sheaves over static graphs, we expect that $\mathcal{F}$ can be used in sheaf-theoretic path finding algorithms, optimizing for hop distance, elapsed time, or some weighted combination of the two.

The prior model assumed instantaneous transmission across any available link in the temporal network. We now model a temporal network, where the nodes are fixed and the links are time-varying, with non-zero transmission time across links. Let $m \in \mathbb{Z}^+$ be the fixed transmission time for all edges $e \in E_T$ at any time. The edge labelling f is given by

$$\begin{aligned} f : E_T \to \big\{ \{I_\alpha\}_{\alpha \in A \subseteq \mathbb{N}} : a, b \in \mathbb{Z}_{\geq 0}, a < b, I_\alpha = [a, b]; \\ \forall \alpha \neq \beta, I_\alpha \cap I_\beta = \emptyset \big\}. \end{aligned}$$

For v_s, v_t as before, we define a sheaf $\mathcal{F}_2$ over T stalk-wise as follows

$$\mathcal{F}_2(v) = \begin{cases} \{\perp\} \times \text{Out}(v) \times \mathbb{Z}_{\geq 0} \times \{0\} \times \{0\} & \text{if } v = v_s \\ \text{In}(v) \times \{\perp\} \times \{0\} \times \mathbb{Z}^+ \times \mathbb{Z}^+ & \text{if } v = v_t \\ \big(\text{In}(v) \times \text{Out}(v) \times \mathbb{Z}_{\geq 0} \times \mathbb{Z}^+ \times \mathbb{Z}^+\big) & \\ \cup \{\perp\} & \text{otherwise} \end{cases}$$
$$\mathcal{F}_2(e) = \big(\mathbb{Z}_{\geq 0} \times \mathbb{Z}^+\big) \cup \{\perp\}$$

Let $x = (e_i, e_o, w, t, d)$ or $\perp$. Then the restriction maps are defined as follows

$$\mathcal{F}_2(v \leq e)(x) = \begin{cases} (w+m, 1) & \text{if } e_i = \perp, e = e_o, \\ & \quad [w, w+m] \subseteq I_\alpha \in f(e) \\ (t, d) & \text{if } e = e_i, \\ & \quad [t-m, t] \subseteq I_\alpha \in f(e) \\ (t+w+m, d+1) & \text{if } e_i \neq \perp, e = e_o, \\ & \quad [t+w, t+w+m] \\ & \quad \subseteq I_\alpha \in f(e) \\ \perp & \text{otherwise} \end{cases}$$

The global sections of $\mathcal{F}_2$ correspond to temporal paths from v_s to v_t over T that are viable for transmission.

Remark 6.3. In this model, we assume that transmissions across a link must be uninterrupted. If the availability of a link has measure zero, then the link is not represented in the model temporal graph. Furthermore, in this model we prune the availability of temporal network links in their model temporal graph edges as follows: Let l be a link in the temporal network whose availability does not have measure zero and let e_l be the edge in the model temporal graph that represents the link l. If l is available at t and $\nexists a \in \mathbb{R}$ such that l is available for the entire interval $[t-a, t+a]$, then e_l is not available at t.

Transmission time across links in a temporal network need not be uniform. We construct a new model that allows for nodes and links to be time varying, and entails a functions $m : E_T \to \mathbb{Z}^+$ that assigns transmission times to each edge. We will restrict to the cases where for all $e \in E_T$ and $v \in V_T$, $f(e)$ and $g(v)$ are the unions of collection of non-negative integral endpoint intervals, given by

$$\begin{aligned} f : E_T \to &\big\{\{I_{\alpha \in A \subseteq \mathbb{N}}\} : a, b \in \mathbb{Z}_{\geq 0}, a < b, I_\alpha = [a, b]; \\ &\forall \alpha \neq \gamma, I_\alpha \cap I_\gamma = \emptyset\big\} \\ g : V_T \to &\big\{\{I_{\beta \in B \subseteq \mathbb{N}}\} : a, b \in \mathbb{Z}_{\geq 0}, a < b, I_\beta = [a, b]; \\ &\forall \beta \neq \gamma, I_\beta \cap I_\gamma = \emptyset\big\}. \end{aligned}$$

For v_s and v_t as before, we define a sheaf $\mathcal{F}_3$ over T stalk-wise as follows

$$\mathcal{F}_3(v) = \begin{cases} \{\perp\} \times \text{Out}(v) \times \mathbb{Z}_{\geq 0} \times \{0\} & \text{if } v = v_s \\ \text{In}(v) \times \{\perp\} \times \{0\} \times \mathbb{Z}^+ & \text{if } v = v_t \\ \big(\text{In}(v) \times \text{Out}(v) \times \mathbb{Z}_{\geq 0} \times \mathbb{Z}^+\big) & \\ \cup \{\perp\} & \text{otherwise} \end{cases}$$
$$\mathcal{F}_3(e) = \mathbb{Z}^+ \cup \{\perp\}.$$

Let $x = (e_i, e_o, w, t)$ or $\perp$, then the restriction maps are defined as in Equation 4.

$$\mathcal{F}_3(v \leq e)(x) = \begin{cases} w + m(e) & \text{if } e_i = \perp, e = e_o, [0, w] \subseteq I_\beta \in g(v), [w, w+m(e)] \subseteq I_\alpha \in f(e), \\ & \quad w + m(e) \in I_\gamma \in g(\text{Head}(e)) \\ t & \text{if } e = e_i, t \in I_\beta \in g(v), [t-m(e), t] \subseteq I_\alpha \in f(e), t - m(e) \in I_\gamma \in g(\text{Tail}(e)) \\ t + w + m(e) & \text{if } e_i \neq \perp, e = e_o, [t, t+w] \subseteq I_\beta \in g(v), [t+w, t+w+m(e)] \subseteq I_\alpha \in f(e), \\ & \quad t + w + m(e) \in I_\gamma \in g(\text{Head}(e)) \\ \perp & \text{otherwise} \end{cases} \tag{4}$$

The global sections $\mathcal{F}_3$ correspond to temporal paths from v_s to v_t over T, where vertices and edges are time-varying, that are viable for transmission.

Remark 6.4. Just as with the model associated to the sheaf $\mathcal{F}_2$, this model assumes that transmissions across a link must be uninterrupted and prunes link availability in their model edge representation in the same way.

The time-varying capacities of links in a temporal network can be modelled by assigning functions, that take in the time as input and output the link's capacity, to the edges of a temporal graph. Let $f : E_T \to \{\text{step functions over } \mathbb{R}\}$ and $m \in \mathbb{Z}^+$ be the message size. Let v_s be a specified source vertex and v_t a specified target vertex. A **temporal path sheaf** $\mathcal{TPS}$ on T is defined stalk-wise as follows

$$\mathcal{TPS}(v) = \begin{cases} \{\perp\} \times \text{Out}(v) \times \mathbb{Z}_{\geq 0} \times \{0\} \times \{0\} & \text{if } v = v_s \\ \text{In}(v) \times \{\perp\} \times \{0\} \times \mathbb{Z}^+ \times \mathbb{Z}^+ & \text{if } v = v_t \\ \big(\text{In}(v) \times \text{Out}(v) \times \mathbb{Z}_{\geq 0} \times \mathbb{Z}^+ \times \mathbb{Z}^+\big) & \\ \cup \{\perp\} & \text{otherwise} \end{cases}$$
$$\mathcal{TPS}(e) = \big(\{[a, b] : a, b \in \mathbb{Z}_{\geq 0} \text{ and } a < b\} \times \mathbb{Z}^+\big) \cup \{\perp\}$$

Let $x = (e_i, e_o, w, t, d)$ or $\perp$. Then the restriction maps are defined as in Equation 5.

$$\mathcal{TPS}(v \leq e)(x) = \begin{cases} ([w, w+a], 1) & \text{if } e_i = \perp, e = e_o \text{ and } \exists a = \min_{r \in \mathbb{Z}^+} \left\{ m \leq \int_w^{w+a} f(e) d\mu \right\} \\ ([t-a, t], d) & \text{if } e = e_i \text{ and } \exists a = \min_{r \in \mathbb{Z}^+} \left\{ m \leq \int_{t-a}^{t} f(e) d\mu \right\} \\ ([t+w, t+w+a], d+1) & \text{if } e_i \neq \perp, e = e_o, \text{ and } \exists a = \min_{r \in \mathbb{Z}^+} \left\{ m \leq \int_{t+w}^{t+w+a} f(e) d\mu \right\} \\ \perp & \text{otherwise} \end{cases} \tag{5}$$

The global sections of a temporal path sheaf $\mathcal{TPS}$ correspond to temporal paths from v_s to v_t in T that are viable for transmission.

Remark 6.5. This model allows for interruptions in transmission across a link.

Let $S \subset V_T$ be a non-empty source set and $T \subseteq V_T \setminus S$ be a non-empty target set. A **generalized temporal path sheaf** $\mathcal{GTP}$ on T is defined stalk-wise as in Equation 6.

$$\mathcal{GTP}(v) = \begin{cases} (\{\perp\} \times \text{Out}(v) \times \mathbb{Z}_{\geq 0} \times \{0\} \times \{0\}) \cup \left(\text{In}(v) \times \text{Out}(v) \times \mathbb{Z}_{\geq 0} \times \mathbb{Z}^+ \times \mathbb{Z}^+\right) \cup \{\perp\} & \text{if } v \in S \\ \left(\text{In}(v) \times \{\perp\} \times \{0\} \times \mathbb{Z}^+ \times \mathbb{Z}^+\right) \cup \left(\text{In}(v) \times \text{Out}(v) \times \mathbb{Z}_{\geq 0} \times \mathbb{Z}^+ \times \mathbb{Z}^+\right) \cup \{\perp\} & \text{if } v \in T \\ \left(\text{In}(v) \times \text{Out}(v) \times \mathbb{Z}_{\geq 0} \times \mathbb{Z}^+ \times \mathbb{Z}^+\right) \cup \{\perp\} & \text{otherwise} \end{cases} \tag{6}$$
$$\mathcal{GTP}(e) = (\{[a, b] : a, b \in \mathbb{Z}_{\geq 0} \text{ and } a < b\}) \cup \{\perp\}$$

Let $x = (e_i, e_o, w, t, d)$ or $\perp$. Then the restriction maps are defined as in Equation 7.

$$\mathcal{GTP}(v \leq e)(x) = \begin{cases} ([w, w+a], 1) & \text{if } e_i = \perp, e = e_o, \text{ and } \exists a = \min_{r \in \mathbb{Z}^+} \left\{ m \leq \int_w^{w+a} f(e) d\mu \right\} \\ ([t-a, t], d) & \text{if } e = e_i, \text{ and } \exists a = \min_{r \in \mathbb{Z}^+} \left\{ m \leq \int_{t-a}^{t} f(e) d\mu \right\} \\ ([t+w, t+w+a], d+1) & \text{if } e_i \neq \perp, e = e_o, \text{ and } \exists a = \min_{r \in \mathbb{Z}^+} \left\{ m \leq \int_{t+w}^{t+w+a} f(e) d\mu \right\} \\ \perp & \text{otherwise} \end{cases} \tag{7}$$

Global sections of a generalized temporal path sheaf $\mathcal{GTP}$ correspond to the disjoint union of temporal paths from vertices in the source set to vertices in the target set in T.

Remark 6.6. The generalized temporal path sheaf construction is analogous to the generalized distance path sheaf construction over a static graph. If the source set or the target set consist of exactly one vertex, then the global sections of $\mathcal{GTP}$ correspond to temporal paths from the source vertex to vertices in the target set or from vertices in the source set to the target vertex, respectively.

Let $v_s \in V_T$ be a specified source vertex and let $T \subseteq V_T \setminus \{v_s\}$ be a non-empty target set. A **temporal multi-target path sheaf** $\mathcal{TMP}$ on T is defined stalk-wise as in Equation 8

$$\mathcal{TMP}(v) = \begin{cases} \{\perp\} \times \text{Out}(v) \times \mathbb{Z}_{\geq 0} \times \{0\} \times \{0\} & \text{if } v = v_s \\ \left(\text{In}(v) \times \{\perp\} \times \{0\} \times \mathbb{Z}^+ \times \mathbb{Z}^+\right) \cup \left(\text{In}(v) \times \text{Out}(v) \times \mathbb{Z}_{\geq 0} \times \mathbb{Z}^+ \times \mathbb{Z}^+\right) & \text{if } v \in T \\ \left(\text{In}(v) \times \text{Out}(v) \times \mathbb{Z}_{\geq 0} \times \mathbb{Z}^+ \times \mathbb{Z}^+\right) \cup \{\perp\} & \text{otherwise} \end{cases} \tag{8}$$
$$\mathcal{TMP}(e) = (\{[a, b] : a, b \in \mathbb{Z}_{\geq 0} \text{ and } a < b\}) \cup \{\perp\}$$

The restriction maps are defined as for $\mathcal{GTP}$. Global section of a temporal multi-target path sheaf $\mathcal{TMP}$ are temporal paths in T that begin at the source vertex, contain every vertex in the target set, and end at a vertex in the target set.

Temporal Hypergraphs

For networks with broadcast, multicast or anycast capabilities, it may be useful to model them using hypergraphs. When such such networks are time-varying, then they can be modelled using temporal hypergraphs.

Definition 6.7. A **temporal**, or **time-varying**, **(un)directed hypergraph** $H = (V_H, E_H, f, g)$ consists of a vertex set V_H, a hyperedge multiset E_H, a function $f : E_H \to 2^{\mathbb{R}_{\geq 0}}$, and a function $g : V_H \to 2^{\mathbb{R}_{\geq 0}}$. The functions f and g specify, respectively, at what times the hyperedges and vertices are available.

We can extend some of the sheaf constructions over temporal graphs to temporal hypergraphs. We will restrict to the cases where the intervals on which the hyperedges and vertices

are active have non-negative integer endpoints, defined in a similar manner as for temporal graphs. We will also define a function $m : E_H \to \mathbb{Z}^+$, that specifies the transmission time across a hyperedge. Let $T(H) = (V_{T(H)}, E_{T(H)}, \hat{g}, \hat{m})$ be the bipartite graph associated with the hypergraph H, where

$$\hat{g}(v) = \begin{cases} g(v) & \text{if } v \in V_H \subseteq V_{T(H)} \\ f(v) & \text{if } v \in E_H \subset V_{G(H)} \end{cases}$$

and $\hat{m}(e) = m(v)$ where e is incident to v and $v \in E_H \subset V_{T(H)}$.

Let $v_s \in V_H$ be a specified source vertex and $v_t \in V_H$ be a specified target vertex. A **temporal hypergraph path sheaf** $\mathcal{THS}$ on $T(H)$ is defined stalk-wise as in Equation 9.

$$\mathcal{THS}(v) = \begin{cases} \{\perp\} \times \text{Out}(v) \times \mathbb{Z}_{\geq 0} \times \{0\} & \text{if } v = v_s \\ \text{In}(v) \times \{\perp\} \times \{0\} \times \mathbb{Z}^+ & \text{if } v = v_t \\ \big(\text{In}(v) \times \text{Out}(v) \times \mathbb{Z}_{\geq 0} \times \mathbb{Z}^+\big) \cup \{\perp\} & \text{if } v \in V_H \subseteq V_{T(H)} \\ \big(\text{In}(v) \times \text{Out}(v) \times \{0\} \times \mathbb{Z}^+\big) \cup \{\perp\} & \text{if } v \in E_H \subset V_{T(H)} \end{cases} \tag{9}$$
$$\mathcal{THS}(e) = \mathbb{Z}^+ \cup \{\perp\}$$

Let $x = (e_i, e_o, w, t)$ or $\perp$. Then the restriction maps are defined as in Equation 10.

$$\mathcal{THS}(v \leq e)(x) = \begin{cases} w + \hat{m}(e) & \text{if } v = v_s, e = e_o, [0, w] \subseteq I_\beta \in \hat{g}(v), [w, w + \hat{m}(e)] \subseteq I_\gamma \in \hat{g}(\text{Head}(e)) \\ t & \text{if } v \in V_H \subseteq V_{T(H)}, e = e_i, t \in I_\alpha \in \hat{g}(v), [t - \hat{m}(e), t] \subseteq I_\alpha \in \hat{g}(\text{Tail}(e)) \\ t + w + \hat{m}(e) & \text{if } v \in V_H \subseteq V_{T(H)}, e = e_o, [t, t + w] \subseteq I_\alpha \in \hat{g}(v), \\ & \quad [t + w, t + w + \hat{m}(e)] \subseteq I_\alpha \in \hat{g}(\text{Head}(e)) \\ t & \text{if } v \in E_H \subset V_{T(H)}, e = e_i, [t - \hat{m}(e), t] \subseteq I_\alpha \in \hat{g}(v), t - \hat{m}(e) \in \hat{g}(\text{Tail}(e)) \\ t & \text{if } v \in E_H \subset V_{T(H)}, e = e_o, [t - \hat{m}(e), t] \subseteq I_\alpha \in \hat{g}(v), t \in \hat{g}(\text{Head}(e)) \\ \perp & \text{otherwise} \end{cases} \tag{10}$$

Global sections of a temporal hypergraph path sheaf correspond to paths in H from v_s to v_t.

7. Weighted Dowker Complexes

Weighted Dowker Complexes are abstract simplicial complexes that are formed according to a specific binary relation between two sets [16]. Essentially, this creates a "geometric" object out of formal relationships between sets; for example, it can create sets of edges and vertices. Past work with weighted Dowker complexes has been centred around their usage as topological constructions and their topological properties. The topological constructions of weighted Dowker complexes also allow statistical methods to be applied.

We wish to use these tools in order to allow for greater specificity within solar system internet structures. Specifically, we wish to use weighted Dowker complexes to call particular structures within the theoretical models of delay tolerant networks.

In a graph, an edge *must* have either one or two vertices (one if it is a loop), and these vertices must be members of the graph. Worded differently, a graph is a set of sets which are closed under subsets. While this sounds pedantic, it illustrates the more general notion of an abstract simplicial complex.

Definition 7.1. An **abstract simplicial complex**, or **ASC**, is a collection of finite nonempty sets T such that if $N \in T$, then so is every nonempty subset of N.

Typically in an ASC, sets of size 3 are represented as faces, sets of size 2 are represented as edges, and sets of size 1 are represented as vertices.

Example 7.2. The first example of an ASC is given below, and illustrated in Figure 14.

$$\begin{aligned} T = &\{\{A, C, D\}, \\ &\{A, C\}, \{A, D\}, \{B, C\}, \{C, D\}, \\ &\{A\}, \{B\}, \{C\}, \{D\}\}. \end{aligned}$$

Notice that T is closed under taking subsets.

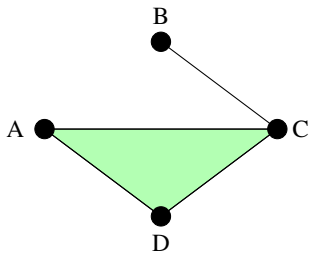


Figure 14. Example abstract simplicial complex

Remark 7.3. If we declare that

$$\begin{aligned} T' = &\{\{A, C, D\}, \\ &\{A, C\}, \{A, D\}, \{B, C\}, \{C, D\}\} \end{aligned}$$

is an ASC, then we can assume that the ASC is *generated* by T' as given. That is, there is a minimal ASC that contains T', which is T as in the example.

Definition 7.4. Let X and Y be sets, and let Z be any relation on X and Y (formally, $Z \subseteq X \times Y$. Then a **Dowker complex** $N(X, Y, Z)$ is an abstract simplicial complex constructed according to the rules of a binary relation between two sets. It is given by the following definition [17]:

$$N(X,Y,Z) = \{[x_{i_0}, \ldots, x_{i_k}] \mid \exists y \in Y \text{such that } (x_{i_j}, y) \in Z \text{ for all } j = 0, \ldots k\}$$

Example 7.5. Let

$$X_1 = [A, B, C, D] \text{ and}$$
$$Y_1 = [1, 2, 3, 4, 5, 6, 7, 8].$$

A relation between X_1 and Y_1 is defined by R_1, as shown by the following matrix:

$$R_1 = \begin{array}{c} \\ A \\ B \\ C \\ D \end{array} \begin{array}{c} \begin{array}{cccccccc} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \end{array} \\ \begin{bmatrix} 1 & 0 & 1 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix} \end{array}$$

This R_1 matrix can then be interpreted as a Dowker complex, in which the matrix rows (X_1) are represented as vertices and the matrix columns (Y_1) denote which vertices are connected. In fact, R_1 gives rise to the same ASC as in Figure 14. To see this, the first column corresponds to the segment joining A and D, and the fifth column corresponds to the face joining A, B, and D. While the edge joining A and C does not appear in the relation explicitly, it must be added for this relation to generate a valid ASC.

Note that in the example, the face ACD has multiplicity 4, namely from columns 5 through 8. A Dowker complex can be extended to retain this information.

Definition 7.6. A **weighted Dowker complex** is a Dowker complex that records the number of times a matrix column is repeated. This number is placed next to the corresponding vertex or edge. If multiple columns form a face on the weighted Dowker complex, the number is placed onto the corresponding face.

Example 7.7. Using the matrix from the previous example, we can begin to note which matrix columns (Y_1) repeat.

We note that where $Y_1 = [1, 3]$, the columns have identical values. They are indicated in bold.

$$R_1 = \begin{bmatrix} \mathbf{1} & 0 & \mathbf{1} & 0 & 1 & 1 & 1 & 1 \\ \mathbf{0} & 1 & \mathbf{0} & 0 & 0 & 0 & 0 & 0 \\ \mathbf{0} & 1 & \mathbf{0} & 1 & 1 & 1 & 1 & 1 \\ \mathbf{1} & 0 & \mathbf{1} & 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$

We note that where $Y_1 = [5, 6, 7, 8]$, the columns also repeat. They are indicated in italics.

$$R_1 = \begin{bmatrix} \mathbf{1} & 0 & \mathbf{1} & 0 & \mathit{1} & \mathit{1} & \mathit{1} & \mathit{1} \\ \mathbf{0} & 1 & \mathbf{0} & 0 & \mathit{0} & \mathit{0} & \mathit{0} & \mathit{0} \\ \mathbf{0} & 1 & \mathbf{0} & 1 & \mathit{1} & \mathit{1} & \mathit{1} & \mathit{1} \\ \mathbf{1} & 0 & \mathbf{1} & 1 & \mathit{1} & \mathit{1} & \mathit{1} & \mathit{1} \end{bmatrix}$$

This gives rise to the weighted Dowker complex in Figure 15.

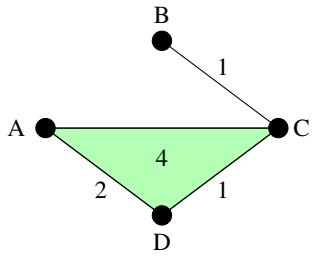


Figure 15. Example weighted abstract simplicial complex

Formally, the total weight is a function t, *t: N(X,Y,Z)* $\to \mathbb{N}$, given by [17]

$$t(\sigma) = \#(y \in Y : (x, y) \in N \text{ for all } x \in \sigma).$$

Definition 7.8. A **dominating set** for a graph $G = (V, E)$ is a subset C of V such that all vertices not in C are adjacent to at least one member of C. The number of vertices in a smallest dominating set for G is known as the dominating number.

Example 7.9. Let G be the graph on vertices $\{1, 2, 3, 4, 5, 6\}$ as shown in Figure 16. A possible dominating set is shown in blue highlighted vertices; the dominating set on the right of the graph shows how considering a dominating set can simplify the graph at hand. In this case, the original graph has three connected components, so the dominating set must have three connected components as well.

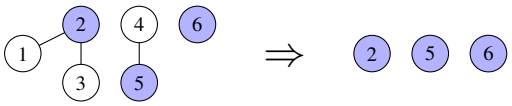


Figure 16. Example dominating set.

Example 7.10. Dominating sets can give rise to weighted Dowker complexes. Firstly, we must convert the dominating set to a binary matrix. To continue the previous example, assume that $X_1 = [2, 5, 6]$ and $Y_1 = [1, 2, 3, 4, 5, 6, (1, 2), (2, 3), (4, 5)]$. Here the ordered pairs represent edges.

$$R_1 = \begin{array}{c} \\ 2 \\ 5 \\ 6 \end{array} \begin{array}{c} \begin{array}{ccccccccc} 1 & 2 & 3 & 4 & 5 & 6 & (1,2) & (2,3) & (4,5) \end{array} \\ \begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix} \end{array}$$

We then can use the resulting R_2 matrix to form a weighted Dowker complex, as shown in Figure 17.

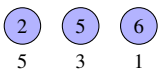


Figure 17. Example weighted Dowker of a dominating set.

Remark 7.11. Weighted Dowker Complexes have the ability to be placed on time-varying graphs and can choose new dominating sets even as the underlying graph changes.

Example 7.12. Figure 18 illustrates a time-varying network with a choice of time-varying dominating sets below the graphs. This shows how the dominating sets can simplify the network dynamics in terms of local connections.

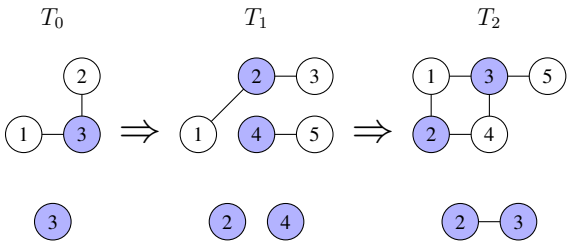


Figure 18. Example time-varying graph with associated time-varying dominating sets shown below.

In a sense, sheaves are for restricting the domains of functions. Their dual, cosheaves, are for gluing objects together [18]. Dowker complexes in particular have been used to form cosheaf constructions [17].

Definition 7.13. Let $G = (V_G, E_G)$ be a graph. A **cosheaf** $\hat{\mathcal{F}}$ is an assignment of

- a set or algebraic object (e.g. a vector space) to each vertex and edge of G, and

- a mapping (of sets, linear, etc.) $\hat{\mathcal{F}}(v \rightsquigarrow e)\colon \hat{\mathcal{F}}(e) \to \hat{\mathcal{F}}(v)$, called an **extension map**, for all vertices v incident to edges e. Additionally, $\hat{\mathcal{F}}(h)$ is referred as the **costalk** of $\mathcal{F}$ at h where $h \in V_G \cup E_G$.

8. Conclusion and Future Work

In this paper we made several contributions to the modeling of space networks with scalability in mind. This includes the ability to attach network areas together, such as scheduled and non-scheduled domains. As such, we can begin to seriously consider the SSI versions of such concepts as *routing domains* or network areas or autonomous systems. We also considered better tools to cover broadcast and multicast scenarios, using hypergraphs, which for example will better capture routing and discovery-style messages. The underlying nature of these time-varying networks was also considered. It was shown that the relationships between nodes and connections in a network can be made geometric via Dowker complexes, enabling the tools of e.g. algebraic topology to be applied. As such, our advances enable research into addressing as well as multi-domain routing for the Solar System Internet. They also open the doors to future research and development projects:

1. We assumed that hypergraphs are clutters. This is not necessarily the case; one can see that a network that features broadcast and multicast as having hyperedges contained in other hyperedges. This generalization requires some results to be recast.
2. Network coding has proven to be beneficial for reducing energy and transmission bandwidth consumption of wireless (ad hoc) networks. One of the most notable result would be a random linear coding (RLC) based broadcast scheme constructed by Fragouli, Widmer, and Le Boudec in [19]. However, such results are often not directly applicable to DTNs due to their peculiar features. We suggest that network coding can be further studied through the lens of sheaf cohomology, pioneered by Ghrist and Hiraoka in [20]. Ghrist and Hiraoka developed a concept of network coding (NC) sheaf, and discovered that the information flows on the network is equivalent to the zeroth cohomology of NC sheaf. Further research studying multi-source/target/cast NC sheaf is required.
3. The types of problems Hansen and Ghrist addressed in their paper [15] did not involve any constraints with inequalities. Hence, we expect that the method suggested by Hansen and Ghrist in can be simplified using Lagrange multipliers instead of KKT optimality condition for some cases. Indeed, if a problem involves inequality constraints, then KKT condition must be used. We also anticipate the study of gradient descent can open a door for lots of other novel optimization methods to join in for sheaf settings, such as the method by Nesterov [21] of approximating a non-smooth functions into smooth functions.
In this paper, we suggested a formulation of shortest path and max-flow-min-cut problem into optimization problems with sheaf Laplacian, as Hansen and Ghrist in [15] did for signal recovery and predictive control problem. We expect that more graph optimization problems (dominating set, graph partitioning, spanning tree, etc.), as well as distributed consensus and synchronization problem can be studied from this direction of approach.
4. With regard to the sheaf pullbacks that were explored in this paper, we hope to further generalize path sheaves beyond sequential networks. Doing so requires care in ensuring that global sections of such a sheaf do not grow in number to a point where they lose their significance to routing problems.
5. While the carry-over sheaf explored in the sheaf pullback section is primitive, we hope to build more robust sheaves that differentiate between data retrieved from each layer of the network.
6. Algebraic Geometry has played a large part in our framing of routing problems. As such, we have some ideas for how routing problems can be re-framed as moduli problems. In particular, we have looked into varieties associated to graphs that arise from LSS ideals [22]. It is our speculation that we could model the loss of edge connectives between vertices, by piecing together the associated LSS varieties in a consistent way. If so, we speculate that there may be such a way to do so for hypergraphs as well.
7. A Python-based sheaf package, PySheaf, can be used to start bridging these results and network simulations to demonstrate the new capability of multi-domain routing in DTNs.
8. Using the tools developed and extended in this paper, the structure (e.g. topology of and nature of continuous maps between) temporal graphs can be determined or even induced.

Acknowledgement

This work was primarily done during the Summer 2022 NASA Space Communications and Navigation (SCaN) Internship program (SIP) at NASA Goddard Space Flight Center (GSFC). Authors are grateful to NASA GSFC and SCaN program for generous support and funding. Authors also thank anonymous reviewers for constructive comments.

REFERENCES

[1] B. Jovanovic, "Internet of Things statistics for 2022 - Taking Things Apart," 05 2022. [Online]. Available: https://dataprot.net/statistics/iot-statistics/

[2] National Aeronautics and Space Administration, "Delay/Disruption Tolerant Networking Overview," 12 2021, Page Editor: Heather Monaghan, NASA Official: Brian Dunbar. [Online]. Available: https://www.nasa.gov/directorates/heo/scan/engineering/technology/delay_disruption_tolerant_networking

[3] L. Torgerson, S. C. Burleigh, H. Weiss, A. J. Hooke, K. Fall, D. V. G. Cerf, K. Scott, and R. C. Durst, "Delay-Tolerant Networking Architecture," RFC 4838, Apr. 2007. [Online]. Available: https://www.rfc-editor.org/info/rfc4838

[4] A. Hylton, R. Short, J. Cleveland, M. Moy, R. Cardona, R. Green, J. Curry, B. Mallery, G. Bainbridge, and Z. Memon, "Sheaf Theoretic Models For Routing In Delay Tolerant Networks," in *IEEE Aerospace Conference (AeroConf 2022)*. Institute of Electrical and Electronics Engineers, 2022. [Online]. Available: https://ntrs.nasa.gov/citations/20220002277

[5] J. A. Fraire, P. Madoery, S. Burleigh, M. Feldmann, J. Finochietto, A. Charif, N. Zergainoh, and R. Velazco, "Assessing Contact Graph Routing Performance and Reliability in Distributed Satellite Constellations," *Journal of Computer Networks and Communications*, vol. 2017, Jul 2017. [Online]. Available: https://doi.org/10.1155/2017/2830542

[6] M. Demmer and K. Fall, "DTLSR: Delay Tolerant Routing for Developing Regions," in *Proceedings of the 2007 Workshop on Networked Systems for Developing Regions*, ser. NSDR '07. New York, NY, USA: Association for Computing Machinery, 2007. [Online]. Available: https://doi.org/10.1145/1326571.1326579

[7] T. Nyasulu and D. H. Crawford, "Comparison of Graph-based and Hypergraph-based Models for Wireless Network Coexistence," in *2021 IEEE International Mediterranean Conference on Communications and Networking (MeditCom)*, 2021, pp. 203–208. [Online]. Available: https://doi.org/10.1109/MeditCom49071.2021.9647587

[8] Q. Li, G. Kim, and R. Negi, "Maximal Scheduling in a Hypergraph Model for Wireless Networks," in *2008 IEEE International Conference on Communications*, 2008, pp. 3853–3857. [Online]. Available: https://doi.org/10.1109/ICC.2008.723

[9] A. Hylton, R. Short, J. Cleveland, O. Freides, Z. Memon, R. Cardona, R. Green, J. Curry, S. Gopalakrishnan, D. V. Dabke *et al.*, "A Survey of Mathematical Structures for Lunar Networks," in *IEEE Aerospace Conference (AeroConf 2022)*. Institute of Electrical and Electronics Engineers, 2022. [Online]. Available: https://ntrs.nasa.gov/citations/20220003566

[10] R. Short, A. Hylton, R. Cardona, R. Green, G. Bainbridge, M. Moy, and J. Cleveland, "Towards Sheaf Theoretic Analyses for Delay Tolerant Networking," in *2021 IEEE Aerospace Conference (Aeroconf 2021)*, 2021, pp. 1–9. [Online]. Available: https://doi.org/10.1109/AERO50100.2021.9438167

[11] M. Moy, R. Cardona, R. Green, J. Cleveland, A. Hylton, and R. Short, "Path Optimization Sheaves," *CoRR*, vol. abs/2012.05974, 2020. [Online]. Available: https://arxiv.org/abs/2012.05974

[12] S. Gharakhloo and V. Welker, "Hypergraph LSS-ideals and Coordinate Sections of Symmetric Tensors," 2022. [Online]. Available: http://arxiv.org/abs/2202.10463

[13] J. Hansen and R. Ghrist, "Toward a Spectral Theory of Cellular Sheaves," *J. Appl. Comput. Topol.*, vol. 3, no. 4, pp. 315–358, 2019. [Online]. Available: https://doi.org/10.1007/s41468-019-00038-7

[14] ——, "Opinion Dynamics on Discourse Sheaves," *SIAM J. Appl. Math.*, vol. 81, no. 5, pp. 2033–2060, 2021. [Online]. Available: https://doi.org/10.1137/20M1341088

[15] ——, "Distributed Optimization with Sheaf Homological Constraints," in *2019 57th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, 2019, pp. 565–571. [Online]. Available: https://doi.org/10.1109/ALLERTON.2019.8919796

[16] C. H. Dowker, "Homology Groups of Relations," *Annals of Mathematics*, vol. 56, no. 1, pp. 84–95, 1952. [Online]. Available: https://doi.org/10.2307/1969768

[17] M. Robinson, "Cosheaf representations of relations and Dowker complexes," *Journal of Applied and Computational Topology*, vol. 6, no. 1, pp. 27–63, Mar 2022. [Online]. Available: https://doi.org/10.1007/s41468-021-00078-y

[18] J. Curry, "Topological Data Analysis and Cosheaves," 2014. [Online]. Available: https://arxiv.org/abs/1411.0613

[19] J. Widmer, C. Fragouli, and J.-Y. Le Boudec, "Efficient Broadcasting Using Network Coding," *IEEE/ACM Transactions on Networking*, vol. 16, no. 2, pp. 450–463, 2008. [Online]. Available: https://doi.org/10.1109/TNET.2007.901080

[20] R. Ghrist and Y. Hiraoka, "Network Codings and Sheaf Cohomology," *IEICE Proceedings Series*, vol. 45, no. A4L-C3, 2011. [Online]. Available: https://doi.org/10.34385/proc.45.A4L-C3

[21] Y. Nesterov, "Smooth minimization of non-smooth functions," *Mathematical programming*, vol. 103, no. 1, pp. 127–152, 2005. [Online]. Available: https://doi.org/10.1007/s10107-004-0552-5

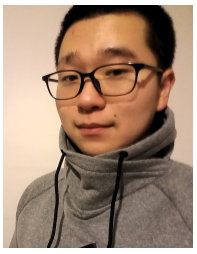
[22] L. Lovász, M. Saks, and A. Schrijver, "Orthogonal representations and connectivity of graphs," *Linear Algebra and its Applications*, vol. 114–115, no. 0024-3795, pp. 439–454, 1989. [Online]. Available: https://doi.org/10.1016/0024-3795(89)90475-8

BIOGRAPHY

Alan Hylton *should probably be designing tube audio circuits, but instead directs Delay Tolerant Networking (DTN) research and development at the NASA Goddard Space Flight Center, where he is humbled to work with his powerful and multidisciplinary team. His formal education is in mathematics from Cleveland State University and Lehigh University, and he considers it his mission to advocate for students. Where possible, he creates venues for mathematicians to work on applied problems, who add an essential diversity to the group.*

Natalie Tsuei *is a current sophomore at American University. She studies computer science and international studies. Her concentrations are in cybersecurity, foreign policy, and national security. After completing her undergraduate degree, Natalie hopes to continue her studies at the graduate level.*

Mark Ronnenberg *is a Ph.D. candidate in mathematics at Indiana University, and plans to graduate in 2023. His research is in the field of low dimensional topology. Thanks to his time at NASA, he is now also interested in applications of pure mathematics to "real world" problems. Prior to coming to Indiana, Mark earned a BA and MA in mathematics from the University of Northern Iowa.*



Jihun Hwang *(Jimmy) is a second year Ph.D. student in computer science at Purdue University. He is mainly interested in studying problems in cryptography through the lens of information theory, analysis of Boolean functions, and computational geometry. However, he ultimately defines himself as an aspiring theoretical computer scientist who likes to discuss any topics in or related to algorithms. Before Purdue, he studied mathematics and computer science at the University of Massachusetts Amherst.*

Brendan Mallery *Brendan is a second year Ph.D. student studying mathematics at Tuts University. His research interests include optimal transportation and the study of Markov chains. Prior to Tufts he obtained a Masters in mathematics at the University at Albany, SUNY and a bachelor's in chemistry and mathematics at Bowdoin College.*

Jonathan Quartin *is a Ph.D. candidate in mathematics at the University of Colorado at Boulder, and plans to graduate in 2023. His graduate work is in Algebraic Geometry, focusing on the geometry of stable map moduli spaces. After graduating, he intends to look for government/industry jobs in the LA area that involve applications of high-level mathematics. Before Boulder, he received a BA in mathematics from the University of California at Berkeley.*

Colin Levaunt *graduated with a Master of Science in Mathematics from the University of Vermont in 2022. And, though, intending to study pure mathematics, as a graduate student, he became more interested in the novel application of advanced mathematical techniques to understanding and solving complex multidisciplinary problems. During the summer of 2022, as an intern at NASA, he had the opportunity to work with a team researching the mathematical foundations of network architectures to enable the future Solar System Internet. He hopes to continue to apply his expertise to challenging and impactful projects.*

Jeremy Quail *is a third year Ph.D. student in mathematics at the University of Vermont. His research is in graph and matroid theory, with a current focus on positroids. Following his experience as a NASA intern in the Summer of 2022, he is interested in continuing to explore applications of high-level math. Jeremy received a BA in mathematics from Queens College and an MS in mathematical sciences from the University of Vermont.*