





A Formal Model for Path Set Attribute Calculation in Network Systems

Giovanni Fiaschi 
Radio New Concept and Algorithms
Ericsson AB
Stockholm, Sweden
Giovanni.Fiaschi@ericsson.com

Carlo Vitucci 
Technology Management
Ericsson AB
Stockholm, Sweden
Carlo.Vitucci@ericsson.com

Thomas Westerback 
Division of Mathematics and Physics
Mälardalen University
Västerås, Sweden
thomas.westerback@mdu.se

Daniel Sundmark 
Computer Science and Software Engineering
Mälardalen University
Västerås, Sweden
daniel.sundmark@mdu.se

Thomas Nolte 
Division of Networked and Embedded Systems
Mälardalen University
Västerås, Sweden
thomas.nolte@mdu.se

Abstract—In graph theory and its practical networking applications, e.g., telecommunications and transportation, the problem of finding paths has particular importance. Selecting paths requires giving scores to the alternative solutions to drive a choice. While previous studies have provided comprehensive evaluation of single-path solutions, the same level of detail is lacking when considering sets of paths. This paper emphasizes that the path characterization strongly depends on the properties under consideration. While property-based characterization is also valid for single paths, it becomes crucial to analyse multiple path sets. From the above consideration, this paper proposes a mathematical approach, defining a functional model that lends itself well to characterizing the path set in its general formulation. The paper shows how the functional model contextualizes specific attributes.

Index Terms—Network Optimization, Path Set Characterization, Routing, Error Probability;

I. INTRODUCTION

A network is a concept often used in several domains: a number of devices or users connected to transfer and exchange data. As a very general definition, in this paper *a network is any collection of objects in which some pairs of these objects are connected by links* [1]. One major application field for the network concept is the telecommunications domain, particularly packet switching, whose purpose is to route packets among devices. Another field of application is transportation, where the purpose is to send vehicles through roads from one place to another.

Routing is the process of choosing a path or a set of paths across one or more networks. Choosing paths implies considering several alternatives, giving a score to each of them, and eventually selecting the highest-scoring alternative. The described selection process is what existing routing algorithms do. Solid solutions exist for single-path routing, where a path is given a cost or length value, which is the sum of

the costs of its component edges. When evaluating sets of multiple paths, scoring criteria are not equally clear. This paper proposes a mathematical model to determine descriptive and representative characteristics of path sets. The model has practical applications in telecommunication systems.

Section II analyses how research has tackled the problem of characterizing network connectivity. Section III recalls essential elements of graph theory and their interpretation for this context. Section IV shows how edge attributes should be composed to calculate equivalent attributes in arbitrary path sets. Section V introduces representations and transformations of path sets to help the property calculation. The full calculation proposal is presented in Section VI and illustrated with notable examples. Section VII analyses the correlation between the paper's proposal and the network polymatroid representation, with an analysis on the discussed attributes. Eventually, Section VIII contains the paper's conclusions and possible future developments.

II. RELATED WORKS

In telecommunications, and more generally in graph theory, analysing the best route is still a research topic. The simplest case, or single-path analysis, is a well-defined problem: finding the shortest route between two points. The “cost” could represent several properties: distance, time, money, and others. Here, algorithms such as Dijkstra [2] and Bellman-Ford [3], [4] have stood the test of time. They focus on calculating the most efficient route, minimizing cost while crossing from one point to another. While most practical approaches focus on additive costs, there exist proposals considering single-path routing generalizations to arbitrary costs [5].

When searching for multiple paths, minimizing the cost alone is not sufficient. Since a single path already secures connectivity between the two points, a cost-based approach would just avoid additional paths. The reason for searching

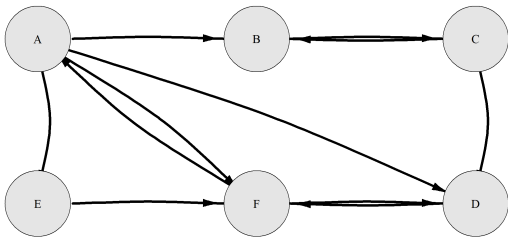


Fig. 1: Generic Graph example

multiple paths is to improve other properties, such as fault resiliency. Previous approaches addressed the problem by adding a diversity score to the set of paths they aimed to find, sometimes implicitly. The diversity concept has several issues. Firstly, there are multiple diversity definitions between two paths. Secondly, extending the diversity concept from a pair of paths to a set of more than two is non-trivial and always risks missing some important properties of the set. Finally, all the definitions of diversity originate from another property, most often the cost, which does not necessarily relate to the property that needs improvement by adding multiple paths; in general, such an approach may mislead the search.

Algorithms solving the K shortest path problem, such as Yen [6] or Eppstein [7] search for multiple paths while minimizing them with a minimum total cost, where the definition of diversity relies implicitly on the presence of at least one different edge, which in most cases is not effective. Others, such as Suurballe [8] and Bhandari [9], focus on finding completely disjoint paths that do not share common features.

Meanwhile, researchers such as Liu [10] and Chondrogianis [11] have explored how to integrate diversity more directly, defining the diversity of a path set as the minimum diversity among any two paths in the set.

Borodin et al. [12] use diversity for high-quality result diversification in information retrieval, database management, and structure localization. Fiaschi et al. [13] apply Borodin's concepts to path search by combining cost and diversity with a parameter tuning their relative importance.

Network graph analysis frequently utilizes the polymatroid mathematical model because it effectively captures the complex constraints of connectivity and interference while simplifying the development of optimization algorithms [14]. Notable examples where polymatroids serve as the primary mathematical model for analysing information flow are the works of Riemensberger et al. [15], Chekuri et al. [16], and Lee et al. [17].

III. ROUTING ELEMENTS

Graphs

Both undirected graphs and directed graphs are considered in graph theory [18]. This paper will focus solely on directed graphs, as they are more applicable to telecommunication and transportation examples [19], where network links are traversed in a specific direction. For an abstract mathematical

representation, a network is represented as a directed graph $\mathcal{G} = (V_g, E_g)$ (see Figure 1), defined by a set V_g of vertices (the network nodes) and a set E_g of edges (the network links), where E_g is a subset of $V_g \times V_g$, the set of ordered pairs of vertices.

Directed Paths

Intuitively, a directed path in a directed graph is an essential concept in applications such as telecommunications and transportation, as it models a route from one network node to another and provides means to convey useful entities between the end nodes, such as information in telecommunications and physical vehicles in transportation. Formally, a directed path from a source vertex s to a destination vertex d , is a sequence of edges $p = \{(v_0, v_1), \dots, (v_{i-1}, v_i), (v_i, v_{i+1}), \dots, (v_{n-1}, v_n)\}$ such that v_0, v_1, \dots, v_n are distinct vertices, $s = v_0$, $d = v_n$ and $(v_{i-1}, v_i) \in E_g$ for $i = 1, 2, \dots, n$. Other equivalent definitions may be found in literature [17], [20], [21].

Connections

In graph theory, a vertex x is connected to a vertex y if a directed path exists from x to y [22], [23]. A vertex may be connected to another vertex in several different ways within a graph. For example, there might be multiple paths between the two vertices.

Definition 1. A path set is a set of paths between the same two vertices.

Definition 2. A connection is the relationship between two vertices together with the attributes characterizing the relationship.

In other words, the definition of a connection refers to the source and destination vertices and its attributes, abstracting from the details of the paths that implement it.

IV. ROUTING ATTRIBUTES

Attributes of a Graph Edge

Edge attributes complete a graph definition, describing some property of the edge. This will help to describe more formally the properties of the connections introduced above. The weight of graph edges can be formally described as a function $w : E_g \rightarrow \mathbb{R}$ (a value of a network link). The meaning of the weight depends on its purpose in specific applications and may represent different concepts. Examples can be delay, administrative cost, capacity, and fault probability.

Attributes of a Directed Path

The weights assigned to the graph edges induce a weight on each directed path. The weight of a path is generally defined as the sum of the weights of its edges. However, for some edge properties it may be convenient to revise the definition according to how the property combines when crossing two or more edges in series.

Different weight meanings require different serial operations. In the following the paper indicates the serial composition of weights with the OP_s operation.

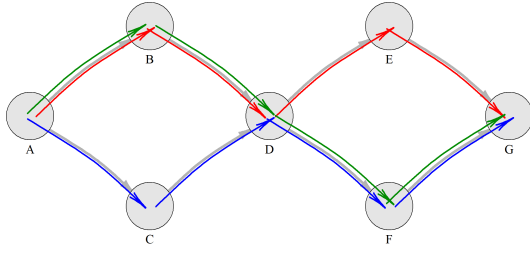


Fig. 2: A graph with three paths

A simple example where the composition is not the sum is the capacity, where the path capacity is limited to the narrowest edge, so that OP_s can be instantiated with the minimum.

Attributes of a Connection Implemented by Disjoint Paths

Setting up one path between two points is essential to allow connections between these points, as when finding a route to travel from one site to another in the transportation application. The path choice determines a specific value of the path attributes, with a meaning on the characteristics of the path, like the time needed to travel from one site to another following that path. If the value of the attribute reachable with a single path is not satisfactory for certain applications, one may want to establish several paths and achieve a change in that value. The composition of attributes using several paths in parallel follows of course a different operation with respect to the serial composition.

Different weight meaning require different parallel operations. In the following, the OP_p operation indicates the parallel composition of weights.

Attributes of a Connection Implemented by Arbitrary Paths

In general, a connection may be implemented by a set partially overlapping paths. In that case, a straightforward combination of serial and parallel operations is inadequate for quantitatively representing the attribute in question. The analysis requires a suitable T transformation of the path set. Different attributes may necessitate different transformations.

V. TOOLS FOR PATH SET REPRESENTATION AND TRANSFORMATION

This chapter presents hypergraphs and their incidence matrices as a convenient tool for representing sets of sets of edges in a graph. A path set is a first example of a set of sets of edges in that a path is a set of edges. The chapter then introduces two transformations of a path set, the union and the cuts, that can also be represented with hypergraphs. Finally it uses vertex-weighted hypergraphs and their r -incidence matrices to incorporate attribute values in the hypergraph representation of path sets and their transformations.

Hypergraphs

Hypergraphs were introduced by C. Berge in the 1960s [24] as a generalization of graphs where edges join any number of vertices.

A (finite) *hypergraph* is the pair $\mathcal{H} = (V_h, E_h)$ where V_h is a finite set of elements called *vertices*, E_h is a finite set of elements called *edges*, and $e \subseteq V_h$ for each element $e \in E_h$. Hypergraphs can be represented using incidence matrices. The *incidence matrix* $B = (b_{ij})_{m \times n}$ of a hypergraph $\mathcal{H} = (V_h, E_h)$ with $V_h = \{v_1, \dots, v_m\}$ and $E_h = \{e_1, \dots, e_n\}$ is defined by

$$b_{ij} = \begin{cases} 1 & \text{if } v_i \in e_j, \\ 0 & \text{otherwise.} \end{cases}$$

A hypergraph can be convenient for describing path sets and their transformations. The next sections explain how to construct hypergraphs for specific representations.

Path Sets

A set of directed paths can be described by a hypergraph, whose vertices correspond to the edges of the graph on which the paths are routed and whose edges correspond to the paths in the set.

For example, Figure 2 illustrates a set of three paths $P = \{p_1, p_2, p_3\}$ (green, red and blue in figure) defined over a graph $\mathcal{G} = (V_g, E_g)$, the three paths connecting vertices A and G .

In the following, the notation $\mathcal{H}_{\text{Id}}(P)$ indicates the hypergraph describing path set P and $\mathbf{R}_{\text{Id}}(P)$ its incidence matrix. In the example of Figure 2, if $\mathcal{H}_{\text{Id}}(P) = (V_h, E_h)$, then $V_h = E_g$, and $E_h = P = \{p_1, p_2, p_3\}$.

Table I.B shows the incidence matrix $\mathbf{R}_{\text{Id}}(P)$ of $\mathcal{H}_{\text{Id}}(P)$.

Union

A very simple transformation of a path set P is the union of all the edges pertaining to its member paths. The union of a path set is a single set of edges, therefore it can be represented by a hypergraph with a single edge.

The notations $\mathcal{H}_{\text{Uni}}(P)$ and $\mathbf{R}_{\text{Uni}}(P)$ will indicate the hypergraph of the union of P and its incidence matrix respectively.

Table I.C shows $\mathbf{R}_{\text{Uni}}(P)$ for the example in Figure 2.

Cuts

A cut is a well-known concept in graph theory: given a partition of the vertices into two disjoint subsets, a graph cut is the set of all edges that have one end vertex in each partition of the cut [25].

Given a path set connecting two vertices S and D , this paper introduces the expression *cut of the path set* for a minimal edge set that, if removed, will make S and D unreachable within the path set. The cut is minimal in the sense that all the edges in the cut must be removed from the path set to make the vertices unreachable.

For example, in Figure 2, the set of edges $B \rightarrow D$ and $F \rightarrow G$ is a cut of the path set, but is not a graph cut, because it is still possible to reach G from A .

The notations $\mathcal{H}_{\text{Cut}}(P)$ and $\mathbf{R}_{\text{Cut}}(P)$ will indicate the hypergraph of the cuts of P and its incidence matrix respectively.

Table I.D shows $\mathbf{R}_{\text{Cut}}(P)$ for the example in Figure 2.

TABLE I: Path set transformation tables

| | A | B | | | C | D | | | | | | | | | | | |
|------|--------------|-----------------------------|----|----|------------------------------|------------------------------|----|----|----|----|----|----|----|----|-----|-----|-----|
| | \mathbf{w} | $\mathbf{R}_{\text{Id}}(P)$ | | | $\mathbf{R}_{\text{Uni}}(P)$ | $\mathbf{R}_{\text{Cut}}(P)$ | | | | | | | | | | | |
| Edge | Weight | P1 | P2 | P3 | Union | C1 | C2 | C3 | C4 | C5 | C6 | C7 | C8 | C9 | C10 | C11 | C12 |
| A→B | w_1 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| A→C | w_2 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| B→D | w_3 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 |
| C→D | w_4 | 0 | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| D→E | w_5 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 |
| D→F | w_6 | 0 | 1 | 1 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 0 |
| E→G | w_7 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 |
| F→G | w_8 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 1 |

Edge Property Vector

A column vector \mathbf{w} encodes the edge properties by associating each edge with a value corresponding to the property of interest. This vector will come handy to distribute the property values over matrices representing a path set or one of its transformations. Table I.A shows a vector $\mathbf{w} = (w_1, \dots, w_8)$ of properties for the graph in Figure 2.

Vertex-Weighted Hypergraphs

This section defines vertex-weighted hypergraphs, which are a generalization of hypergraphs. For vertex-weighted hypergraphs, the paper also introduces r -incidence matrices, which generalize the concept of incidence matrices for hypergraphs.

A *vertex-weighted hypergraph* is a tuple $\mathcal{H} = (V_h, E_h, w)$ where (V_h, E_h) is a hypergraph and $w : V_h \rightarrow \mathbb{R}$ is a weight function which assign a real number $w(v)$ to each vertex $v \in V_h$.

If a hypergraph represents a set of sets of edges in a graph \mathcal{G} , and \mathcal{G} has an associated edge property vector \mathbf{w} , then a vertex-weighted hypergraph can be easily constructed by associating \mathbf{w} to the vertices of the hypergraph, since they correspond to the edges of \mathcal{G} .

For any real number $r \in \mathbb{R}$, let the *r -incidence matrix* $B_r = (b_{ij})_{m \times n}$ of a vertex-weighted hypergraph $\mathcal{H} = (V_h, E_h, w)$ with $V_h = \{v_1, \dots, v_m\}$ and $E_h = \{e_1, \dots, e_n\}$ be defined by

$$b_{ij} = \begin{cases} w(v_i) & \text{if } v_i \in e_j, \\ r & \text{otherwise.} \end{cases}$$

 TABLE II: r -incidence matrices

| | A | | B | | | | | | | | | | | | C | | | | | | | | | | | | |
|------|--------------|--------------|---------------------------------------|----|----|-----|----|----|----|-----|----|-----|-----|-----|---------------------------------------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|---|
| | Cap. | Prob. | $R(0, c, \mathbf{R}_{\text{Cut}}(P))$ | | | | | | | | | | | | $R(1, p, \mathbf{R}_{\text{Cut}}(P))$ | | | | | | | | | | | | |
| Edge | \mathbf{c} | \mathbf{p} | C1 | C2 | C3 | C4 | C5 | C6 | C7 | C8 | C9 | C10 | C11 | C12 | C1 | C2 | C3 | C4 | C5 | C6 | C7 | C8 | C9 | C10 | C11 | C12 | |
| A→B | 25 | 0.0050 | 25 | 25 | 25 | 25 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0.0050 | 0.0050 | 0.0050 | 0.0050 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| A→C | 10 | 0.0075 | 10 | 0 | 0 | 0 | 10 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0.0075 | 1 | 1 | 1 | 0.0075 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| B→D | 25 | 0.0070 | 0 | 0 | 0 | 0 | 25 | 25 | 25 | 25 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 0.0070 | 0.0070 | 0.0070 | 0.0070 | 1 | 1 | 1 | 1 | 1 |
| C→D | 25 | 0.0040 | 0 | 25 | 0 | 0 | 0 | 25 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0.0040 | 1 | 1 | 1 | 0.0040 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| D→E | 10 | 0.0115 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 10 | 10 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0.0115 | 0.0115 | 1 | 1 | 1 |
| D→F | 25 | 0.0045 | 0 | 0 | 25 | 0 | 0 | 0 | 25 | 0 | 25 | 0 | 25 | 0 | 1 | 1 | 0.0045 | 1 | 1 | 1 | 0.0045 | 1 | 0.0045 | 1 | 0.0045 | 1 | 1 |
| E→G | 10 | 0.0105 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 10 | 10 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0.0105 | 0.0105 | 0.0105 | 1 |
| F→G | 100 | 0.0065 | 0 | 0 | 0 | 100 | 0 | 0 | 0 | 100 | 0 | 100 | 0 | 100 | 1 | 1 | 1 | 0.0065 | 1 | 1 | 1 | 0.0065 | 1 | 0.0065 | 1 | 0.0065 | 1 |

Note that the 0-incidence matrix B_0 of a vertex-weighted hypergraph $\mathcal{H} = (V_h, E_h, w)$ with $w(v) = 1$ for all $v \in V$ equals the common incidence matrix B of the hypergraph (V_h, E_h) .

The r -incidence matrices of special interest for this paper are the 0-incidence and 1-incidence matrices.

The notation $B_r = R(r, w, B)$ denotes the r -incidence matrix of a vertex-weighted hypergraph $\mathcal{H} = (V_h, E_h, w)$, where B represents the incidence matrix of the underlying hypergraph (V_h, E_h) without vertex weights.

Table II shows r -incidence matrix examples for hypergraphs representing cut transformation of the path set in Figure 2 with edge property vectors \mathbf{c} (capacity) and \mathbf{p} (unavailability or fault probability), as shown in Table II.A.

VI. CALCULATION OF PROPERTIES

The conclusion of Section IV suggests that, in a specific domain, a mathematical model characterizes a path set independently of the attribute under analysis. The mathematical model represents the intrinsic contributions of the path set:

- an operation that represents the contribution to the characterization due to the serialization of the edges in a path,
- an operation that represents the contribution to the characterization due to the parallelization of the paths in a set of disjoint paths,
- a transformation combing the two operations in an arbitrary path set.

Equation 1 summarizes these three essential elements of the mathematical model, although not necessarily in the order laid down:

$$\Psi(P) = OP_s \circ OP_p \circ T \quad (1)$$

TABLE III: Operations and transformations

| Example | OP_s | OP_p | \mathbf{T} |
|---------------------|---|----------|--------------------------------|
| Delay | + | max | $R(0, w, \mathbf{R}_{Id}(P))$ |
| Administrative cost | + | + | $R(0, w, \mathbf{R}_{Uni}(P))$ |
| Combined capacity | min | + | $R(0, w, \mathbf{R}_{Cut}(P))$ |
| Unavailability | $1 - ((1 - (.)) \times \times (1 - (.)))$ | \times | $R(1, w, \mathbf{R}_{Cut}(P))$ |
| Fault probability | + | \times | $R(1, w, \mathbf{R}_{Cut}(P))$ |

With the tools presented so far, it is now possible to present the complete attribute calculation in specific examples with serial and parallel compositions together with transformations.

Table III shows a summary of five notable examples.

Table IV shows their instantiations in a more direct notation. The serial operations in the given examples are:

- Sum for the delay (edge delays are simply summed)
- Sum for the administrative cost
- Minimum for the capacity (bottleneck edge)
- One's complement to product of the one's complements for the unavailability (because of probability theory [26])
- Sum for the fault probability (as simplification of the unavailability neglecting the higher order components [26])

The parallel operations in the given examples are:

- Max for the delay (assuming to wait for the latest packet)
- Sum for the administrative cost
- Sum for the capacity
- Product for the unavailability and fault probability (because of probability theory [26])

The path set transformations are:

- Identity for the delay, then serial first
- Union for the administrative cost
- Cuts for the capacity, unavailability and fault probability, then parallel first

The r value in the r -incidence matrices shall be the identity element of the operation to be performed first, therefore:

- 0 for delay, administrative cost and capacity (sum first)
- 1 for unavailability and fault probability (product first)

Five examples of path set properties illustrate the application of the method: delay, administrative cost, capacity, unavailability, and fault probability. Table V shows their edge property vectors.

TABLE IV: Property examples instantiations

| | |
|--------------------|--|
| Delay(P) | $\max_{i p_i \in P} \sum_{j e_j \in p_i} w_j$, |
| Cost(P) | $\sum_{i e_i \in \cup P} w_i$, |
| Capacity(P) | $\min_j \sum_i T_{ij}$, where $\mathbf{T} = R(0, w, \mathbf{R}_{Cut}(P))$, |
| UnavailProb(P) | $1 - \prod_j (1 - \prod_i T_{ij})$, where $\mathbf{T} = R(1, w, \mathbf{R}_{Cut}(P))$, |
| FaultProb(P) | $\sum_j \prod_i T_{ij}$, where $\mathbf{T} = R(1, w, \mathbf{R}_{Cut}(P))$. |

Delay

Equation 2 shows the mathematical application example for \mathbf{T} matrix in the delay calculation.

$$Delay(P) = \max \sum_i T_{ij} = \max([340, 230, 225]) = 340\mu s$$

$$\mathbf{T} = R(0, w, \mathbf{R}_{Id}(P)) = R \left(0, \begin{bmatrix} 50 \\ 75 \\ 70 \\ 40 \\ 115 \\ 45 \\ 105 \\ 65 \end{bmatrix}, \begin{bmatrix} 1 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 1 \end{bmatrix} \right) = \begin{bmatrix} 50 & 50 & 0 \\ 0 & 0 & 75 \\ 70 & 70 & 0 \\ 0 & 0 & 40 \\ 115 & 0 & 0 \\ 0 & 45 & 45 \\ 105 & 0 & 0 \\ 0 & 65 & 65 \end{bmatrix} \quad (2)$$

Administrative Cost

Equation 3 shows the numerical value of the administrative cost. $\mathbf{T} = R(0, w, \mathbf{R}_{Uni}(P))$, where Table I.C shows $\mathbf{R}_{Uni}(P)$.

$$Cost(P) = \sum \mathbf{T} = \sum_{1900} ([100, 150, 300, 200, 500, 100, 450, 100]) = (3)$$

Capacity

Equation 4 shows the numerical value of the capacity. $\mathbf{T} = R(0, w, \mathbf{R}_{Cut}(P))$, as described in Table II.B.

$$Capacity(P) = \min \sum_i T_{ij} = \min([35, 50, 50, 125, 35, 50, 50, 125, 35, 110, 35, 110]) = 35Gbps \quad (4)$$

Unavailability and Fault Probability

Equation 5 shows the numerical values of unavailability and fault probability. $\mathbf{T} = R(1, w, \mathbf{R}_{Cut}(P))$, as described in Table II.C.

$$UnavailProb(P) = 1 - \prod_j \left(1 - \prod_i T_{ij} \right) = 0.0005118815 \quad (5)$$

$$FaultProb(P) = \sum_j \prod_i T_{ij} = 0.000512$$

TABLE V: Example edge property vectors

| | Delay (μs) | Admin. cost | Capacity (Gbps) | Fault probability |
|-----|-------------------|-------------|-----------------|-------------------|
| A→B | 50 | 100 | 25 | 0.0050 |
| A→C | 75 | 150 | 10 | 0.0075 |
| B→D | 70 | 300 | 25 | 0.0070 |
| C→D | 40 | 200 | 25 | 0.0040 |
| D→E | 115 | 500 | 10 | 0.0115 |
| D→F | 45 | 100 | 25 | 0.0045 |
| E→G | 105 | 450 | 10 | 0.0105 |
| F→G | 65 | 100 | 100 | 0.0065 |

where $\prod_i T_{ij}$ is:

$$\begin{aligned} & [0.00003750, 0.00002000, 0.00002250, \\ & 0.00003250, 0.00005250, 0.00002800, \\ & 0.00003150, 0.00004550, 0.00005175, \\ & 0.00007475, 0.00004725, 0.00006825] \end{aligned}$$

It is worth noting how the fault probability calculation perfectly matches the HAM method described by Vitucci et al. [27] using a more "empirical" method.

VII. POLYMATROIDS

Matroids, a field in algebraic combinatorics, and the generalization of matroids, polymatroids, are research areas with applications and connections to many other areas in mathematics and computer science. For example, polymatroid theory has proven to be very useful in combinatorial optimization [28]. One of the strengths with matroids and polymatroids is the many equivalent axiom systems from different areas of mathematics that can be used to define these objects. This paper defines matroids and polymatroids using submodular set functions.

Definition 3. Let E be a finite set. A pair $\mathcal{P} = (\rho, E)$ is a (finite) polymatroid on E with set function $\rho : 2^E \rightarrow \mathbb{R}$ if ρ satisfies the three conditions (R1), (R2), and (R3) for all $X, Y \subseteq E$. Note that a matroid is a polymatroid which additionally satisfies the two conditions (R4) and (R5) for all $X \subseteq E$.

$$\begin{aligned} (R1) \quad & \rho(\emptyset) = 0, & (R4) \quad & \rho(X) \in \mathbb{Z}, \\ (R2) \quad & X \subseteq Y \Rightarrow \rho(X) \leq \rho(Y), & (R5) \quad & \rho(X) \leq |X|, \\ (R3) \quad & \rho(X) + \rho(Y) \geq \rho(X \cap Y) + \\ & \rho(X \cup Y). \end{aligned}$$

Submodular and Supermodular Considerations

Polymatroids have proven to be a highly useful mathematical tool in various areas of mathematics and computer science. In this chapter, let E be a set of directed paths from a vertex s to a vertex d in a graph associated with a network. Analysing delay, cost, capacity, unavailability probability, and fault probability of a path set $P \subseteq E$ leads to the definition of set functions $\rho : 2^E \rightarrow \mathbb{R}$. The analysis of these set functions shows that the pair (ρ, E) forms a polymatroid with respect to delay and cost, satisfying conditions (R1)–(R3) in Definition 3. However, for the concepts of capacity, unavailability probability, and fault probability, the situation is different. A set function $\rho : 2^E \rightarrow \mathbb{R}$ is said to be *submodular* if (R3) in Definition 3 holds, i.e., $\rho(X) + \rho(Y) \geq \rho(X \cap Y) + \rho(X \cup Y)$ for all $X, Y \subseteq E$, and ρ is said to be *supermodular* when it satisfies the opposite condition to submodular, i.e., $\rho(X) + \rho(Y) \leq \rho(X \cap Y) + \rho(X \cup Y)$ for all $X, Y \subseteq E$. A supermodular set function ρ satisfying (R1) and (R2) can be transformed into a submodular set function $\rho' : 2^E \rightarrow \mathbb{R}$ that satisfies (R1) and (R2) by defining, for all $X \subseteq E$,

$$\rho'(X) = \rho(E) - \rho(E - X) \quad (6)$$

This transformation ensures that (ρ', E) forms a polymatroid.

This paper examines the concepts of delay, cost, capacity, unavailability, and fault probability from a polymatroidal point of view. Interest in these mathematical structures stems from the capacity of polymatroid theory to provide a valuable framework for future research on path set attribute calculations in network systems.

Definition 4. Let E be a set of directed paths from a vertex s to a vertex d in the graph associated with a network. Furthermore, let w_{Del} , w_{Cost} , w_{Cap} , w_{UPr} , and w_{FPr} be the weight functions on the edges of the graph associated with the concepts delay, cost, capacity, unavailability probability, and fault probability, respectively, on the network. For any subset $P \subseteq E$, let

$$\begin{aligned} (i) \quad & \rho_{\text{Del}}(P) = \text{Delay}(P), \\ (ii) \quad & \rho_{\text{Cost}}(P) = \text{Cost}(P), \\ (iii) \quad & \rho_{\text{Cap}}(P) = \text{Capacity}(P), \\ (iv) \quad & \rho_{\text{UPr}}(P) = \text{UnavailProb}(P), \\ (v) \quad & \rho_{\text{FPr}}(P) = \text{FaultProb}(P), \end{aligned}$$

where $\text{Delay}(P)$, $\text{Cost}(P)$, $\text{Capacity}(P)$, $\text{UnavailProb}(P)$, and $\text{FaultProb}(P)$ are defined as in Table IV with weight functions w_{Del} , w_{Cost} , w_{Cap} , w_{UPr} , and w_{FPr} , respectively.

Delay and Cost

Theorem 1. Let E be a set of directed paths from a vertex s to a vertex d in the graph associated with a network. Then

- (i) (ρ_{Del}, E) is a polymatroid,
- (ii) (ρ_{Cost}, E) is a polymatroid.

Proof. Proving that a set function constitutes a polymatroid requires demonstrating that the set function satisfies conditions (R1)–(R3) in Definition 3. For any path set $A \subseteq E$, let $\mathcal{E}(A)$ denote the set of edges in the path set A .

Let $A, B \subseteq E$.

Statement (i):

(R1): Since the sum of an empty sum equals zero, it follows immediately that $\rho_{\text{Del}}(\emptyset) = 0$.

(R2): Assume that $A \subseteq B$. Then

$$\rho_{\text{Del}}(A) = \max_{a \in A} \rho_{\text{Del}}(a) \leq \max_{b \in B} \rho_{\text{Del}}(b) = \rho_{\text{Del}}(B).$$

(R3): First, observe that

$$\begin{aligned} \rho_{\text{Del}}(A \cup B) &= \max_{e \in A \cup B} \rho_{\text{Del}}(e) \\ &= \max(\max_{a \in A} \rho_{\text{Del}}(a), \max_{b \in B} \rho_{\text{Del}}(b)) \\ &= \max(\rho_{\text{Del}}(A), \rho_{\text{Del}}(B)). \end{aligned}$$

Without loss of generality, assume that $\rho_{\text{Del}}(A \cup B) = \rho_{\text{Del}}(A)$. Then

$$\begin{aligned} \rho_{\text{Del}}(A) + \rho_{\text{Del}}(B) &= \rho_{\text{Del}}(A \cup B) + \rho_{\text{Del}}(B) \\ &\stackrel{(R2)}{\geq} \rho_{\text{Del}}(A \cup B) + \rho_{\text{Del}}(A \cap B). \end{aligned}$$

Statement (ii): Observe that

$$\rho_{\text{Cost}}(A) = \sum_{x \in \mathcal{E}(A)} w_{\text{Cost}}(x).$$

(R1): Since the sum of an empty sum equals zero, it follows immediately that $\rho_{\text{Cost}}(\emptyset) = 0$.

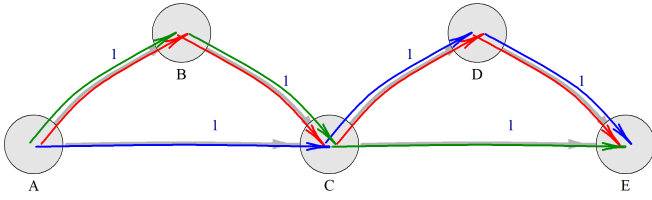


Fig. 3: Capacity example

(R2): Assume that $A \subseteq B$. Then

$$\rho_{\text{Cost}}(A) = \sum_{x \in \mathcal{E}(A)} w_{\text{Cost}}(x) \leq \sum_{y \in \mathcal{E}(B)} w_{\text{Cost}}(y) = \rho_{\text{Cost}}(B).$$

(R3): The following shows that ρ_{Cost} is both submodular and supermodular, i.e., modular,

$$\begin{aligned} \rho_{\text{Cost}}(A) + \rho_{\text{Cost}}(B) &= \\ \sum_{x \in \mathcal{E}(A)} w_{\text{Cost}}(x) + \sum_{y \in \mathcal{E}(B)} w_{\text{Cost}}(y) &= \\ \sum_{x \in \mathcal{E}(A \cup B)} w_{\text{Cost}}(x) + \sum_{y \in \mathcal{E}(A \cap B)} w_{\text{Cost}}(y) &= \\ \rho_{\text{Cost}}(A \cup B) + \rho_{\text{Cost}}(A \cap B). \end{aligned}$$

□

Capacity

The following example shows that ρ_{Cap} is, in general, neither submodular nor supermodular.

Example 1. In Figure 3, let p_1 , p_2 , and p_3 represent the red, green, and blue paths from node a to node e , respectively, and let the capacities of all the edges be 1.

If A is the path set $\{p_1, p_2\}$ and B is the path set $\{p_1, p_3\}$, then

$$\begin{aligned} \rho_{\text{Cap}}(A) + \rho_{\text{Cap}}(B) &= 1 + 1 \\ &< 2 + 1 \\ &= \rho_{\text{Cap}}(A \cup B) + \rho_{\text{Cap}}(A \cap B). \end{aligned}$$

If instead A is the path set $\{p_1\}$ and B is the path set $\{p_2\}$, then

$$\begin{aligned} \rho_{\text{Cap}}(A) + \rho_{\text{Cap}}(B) &= 1 + 1 \\ &> 1 + 0 \\ &= \rho_{\text{Cap}}(A \cup B) + \rho_{\text{Cap}}(A \cap B). \end{aligned}$$

Unavailability and Fault Probability

Both the unavailability $\rho_{\text{UPr}}(P)$ and the Fault probability $\rho_{\text{FPr}}(P)$ do not satisfy the polymatroid property (R1) of Definition 3. If a set is empty, there is no connection available between two vertices, with the natural consequence that $\rho_{\text{UPr}}(\emptyset) = 1$. Following the same principle, if a set is empty there is no working path between two vertices, so even $\rho_{\text{FPr}}(\emptyset) = 1$.

However, for the probability of the complementary events, the Availability function $1 - \rho_{\text{UPr}}(P)$ and the

Serviceability function $1 - \rho_{\text{FPr}}(P)$, would not suffer by the same issue. Investigating about non-decreasing property and submodularity of Availability and Serviceability requires deeper analysis and is considered for further study.

As a summary:

- The associated functions to the concept of delay and cost have the polymatroid properties.
- The associated function to the capacity is neither submodular nor supermodular. This means that the function is not polymatroidial and cannot, in general, be transformed into a polymatroidial set function using equation (6).
- The associated functions to the unavailability and fault probability do not have the polymatroid properties. However, their transformation into a polymatroidial set function need further analysis. For example, by considering the complementary events and associated function Availability and Serviceability.

VIII. CONCLUSIONS AND FUTURE WORKS

Multi-path computation in literature lacks a formal approach to quantify multiple parameters and characterizations of a path set. This paper proposes a paradigm shift. By addressing specific properties of graph edges, it introduces an evaluation of vertex pair connectivity that

- adapts to edge property composition peculiarities, generalizing to non-additive properties, and
- generalizes the computation to path sets and preserves the focus on the property.

The paper contextualizes the proposed solution for five different functional specifications: delay, administrative cost, capacity, unavailability, and fault probability.

The implementation of the functional model was not covered in this work. It may be the subject of a future study, that will also allow the measurement of its complexity, which was not possible to explore in this paper.

Beyond its immediate applications, this solution opens the door to automation in path selection.

By encapsulating functional characteristics in a meta-function, this model lays the groundwork for future algorithms that can dynamically identify optimal path sets, promising to enhance efficiency and significantly reduce manual workload in data analysis.

A polymatroidal analysis of the function model would support an easier evaluation of network configurations. The paper shows the polymatroidal characterization for delay and administrative cost function. Further analysis of Serviceability and Availability is another natural evolution of this paper.

ACKNOWLEDGEMENT

The work presented in this paper is sponsored by Ericsson, Mälardalen University and the Swedish Knowledge Foundation (KKS), via the industrial PhD School ARRAY.

REFERENCES

- [1] D. Easley and J. Kleinberg, *Networks, Crowds, and Markets*, 2010.
- [2] E. W. Dijkstra, "A note on two problems in connexion with graphs," *Numerische Mathematik*, vol. 1, 1959.
- [3] R. Bellman, "On a routing problem," *Quarterly of Applied Mathematics*, vol. 16, 1958.
- [4] L. R. Ford, *Network Flow Theory*. RAND Corporation, 1956.
- [5] J. L. Sobrinho, "An algebraic theory of dynamic network routing," *IEEE/ACM Transactions on Networking*, vol. 13, 2005.
- [6] J. Y. Yen, "Finding the k shortest loopless paths in a network," *Management Science*, vol. 17, 1971.
- [7] D. Eppstein, "Dynamic euclidean minimum spanning trees and extrema of binary functions," *Discrete & Computational Geometry*, vol. 13, 1995.
- [8] J. W. Suurballe, "Disjoint paths in a network," *Networks*, vol. 4, 1974.
- [9] R. Bhandari, "Survivable networks: Algorithms for diverse routing," *Computers & Mathematics with Applications*, vol. 38, 1999.
- [10] H. Liu, C. Jin, B. Yang, and A. Zhou, "Finding top-k shortest paths with diversity," *IEEE Transactions on Knowledge and Data Engineering*, vol. 30, 2018.
- [11] T. Chondrogiannis, P. Bours, J. Gamper, and U. Leser, "Alternative routing: K-shortest paths with limited overlap," in *GIS: Proceedings of the ACM International Symposium on Advances in Geographic Information Systems*, vol. 03-06-November-2015, 2015.
- [12] A. Borodin, A. Jain, H. C. Lee, and Y. Ye, "Max-sum diversification, monotone submodular functions, and dynamic updates," *ACM Transactions on Algorithms*, vol. 13, 2017.
- [13] G. Fiaschi, F. D. Calabrese, and D. Ceccarelli, "Methods and apparatus for transmitting data over multiple paths in a network," 4 2020, patent WIPO PCT/EP2018/078025.
- [14] T. Harks, M. Klimm, and B. Peis, "Resource competition on integral polymatroids," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 8877, 2014.
- [15] M. Riemensberger and W. Utschick, "A polymatroid flow model for network coded multicast in wireless networks," *IEEE Transactions on Information Theory*, vol. 60, 2014.
- [16] C. Chekuri, S. Kannan, A. Raja, and P. Viswanath, "Multicommodity flows and cuts in polymatroidal networks," *SIAM Journal on Computing*, vol. 44, 2015.
- [17] L. G. Latham, "Network flow analysis algorithms," *Ecological Modelling*, vol. 192, 2006.
- [18] A. Sandryhaila and J. M. Moura, "Discrete signal processing on graphs: Frequency analysis," *IEEE Transactions on Signal Processing*, vol. 62, 2014.
- [19] J. Monteiro, G. Robertson, and B. Atkinson, "Networks in transportation-theory," in *Proc. Ctrf 47th Annual Conference*, 2012, pp. 1–21.
- [20] H. Hu, X. Zhai, M. Wang, and G. Hu, "Graph analysis of network flow connectivity behaviors," *Turkish Journal of Electrical Engineering and Computer Sciences*, vol. 27, 2019.
- [21] B. Potter, "Network flow analysis," *Network Security*, vol. 2007, no. 12, pp. 18–19, 2007.
- [22] W. Stallings, *Data and computer communications*, 2007.
- [23] J. Kurose and K. Ross, *Computer networking: A top-down approach, global edition*, 2021.
- [24] C. Berge, *Hypergraphs: Combinatorics of Finite Sets*, ser. North-Holland Mathematical Library. Elsevier Science, 1984.
- [25] R. Diestel, *Graph Theory (Graduate Texts in Mathematics)*, 2000.
- [26] E. Dubrova, *Fault-Tolerant Design*. Springer New York, 2013.
- [27] C. Vitucci, T. Westerbäck, D. Sundmark, H. Forsberg, T. Nolte, and M. Jägemar, "A deductive fault analysis method based on hypergraphs," vol. 58, no. 4, 2024, pp. 378–383, 12th IFAC Symposium on Fault Detection, Supervision and Safety for Technical Processes SAFEPROCESS 2024.
- [28] A. Schrijver, *Combinatorial Optimization*, 2003rd ed., ser. Algorithms and Combinatorics. Berlin, Germany: Springer, Dec. 2002.