Transforming Multiplex Networks to Other Forms

James Q. Sherrell
U.S. Military Academy at West Point, NY, USA

Abstract

Multiplex networks are useful for modeling certain scenarios such as transportation between logistics hubs with various, hub-specific transport mediums, a chain of decision-making processes by various entities within an organization, and more. However, this is not the only type of network useful for modeling these scenarios.

Using various techniques, we demonstrate how to convert a multiplex network into both a multidimensional and simple network. We demonstrate the results on a synthetic, multiplex communication sub-network. In addition to our resultant networks, we briefly demonstrate how network model choice affects certain metrics of interest.

Background

Multilayer networks contain granular details about the subjects they model. The notion of multilayer networks encompasses many different types of networks with differing relationship natures for both inter- and intra-layer edges. Furthermore, multilayer networks are not always the only means to model some subjects. Different network models enable us to observe subjects in different lights.

A *simple* network consists of vertices and edges. We do not permit multiple edges between any pair of vertices.

A *multidimensional* network consists of vertices, edges, and labels ascribed to the edges to differentiate the nature of the connected nodes' relationships.

A *multiplex* network consists of multiple layers. Within each layer, we permit simple networks. Inter-layer edges (i.e., *coupling edges*) connect counterpart nodes. *Counterpart nodes* are representations of vertices from a fundamental vertex set within each layer. A vertex within the fundamental vertex set is a *super-vertex*.

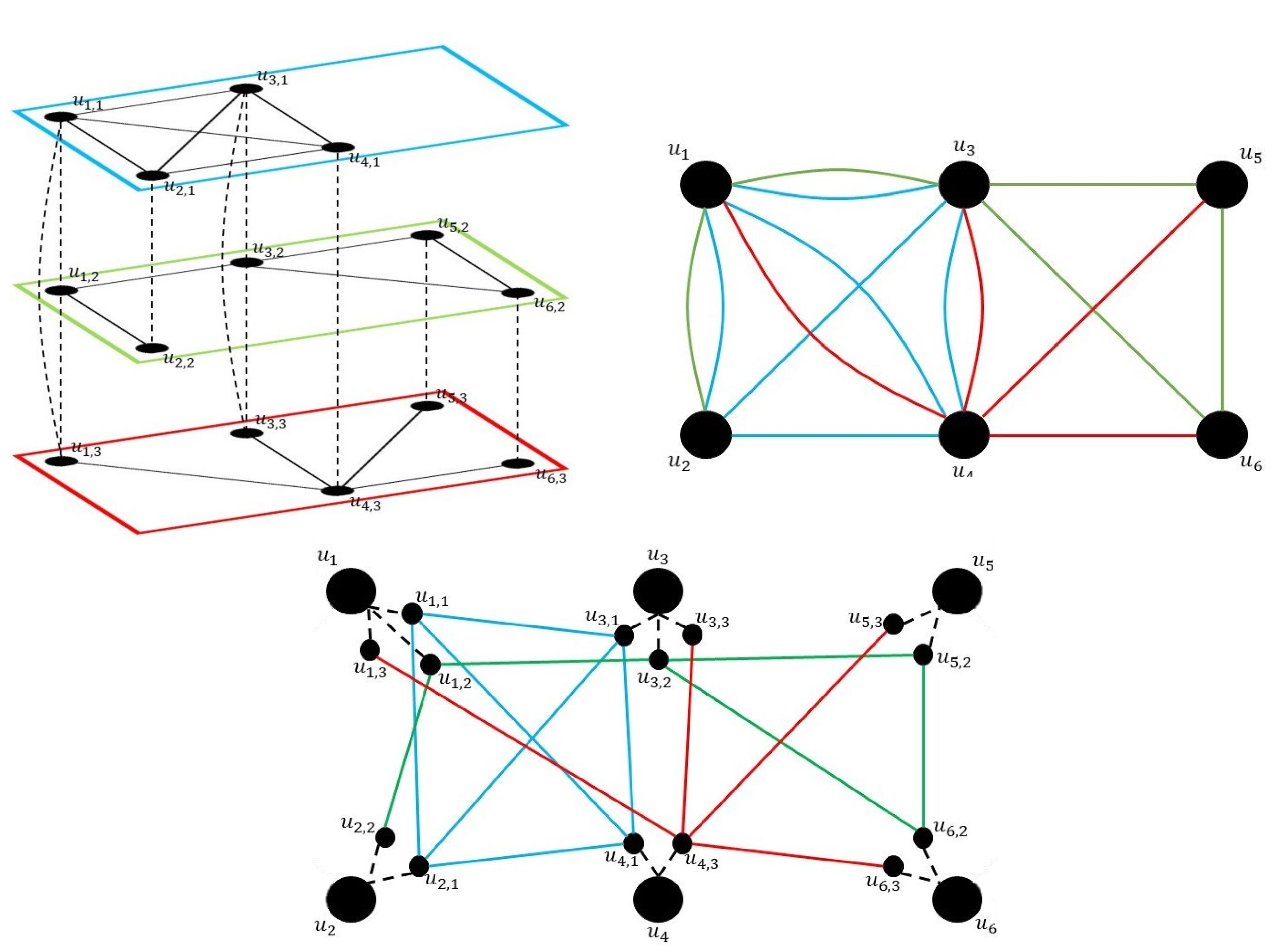


Figure 1: Examples of a subject modelled as a (top-left) multiplex, (top-right) multi-dimensional, and (bottom) simple network.

Various set operations (e.g., union and deletion) convert one type of network to another. From the network perspective, these operations appear to aggregate and divide vertices and edges. We develop algorithms to demonstrate these transformative processes.

Methodology

Given a multiplex network, we employ Algorithm (1) to create a multidimensional network. Algorithm (2), which incorporates Algorithms (3) and (4), creates a simple network from the multidimensional network.

Algorithm 1: Creating G^{md} from G^{mp}

Data: $G^{mp} = (V^{mp}, E^{mp}, L^{mp})$
Result: $G^{md} = (V^{md}, E^{md})$

Instantiate empty set E^{md}

let $V^{mp} = W \times L^{mp}$
$V^{md} = W$

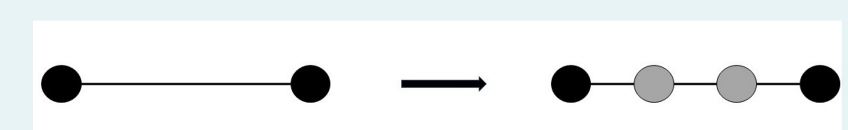
foreach $L \in L^{mp}$
  let $W' \subseteq W$ be maximally independent where $W' \times L = \vec{1}$

  foreach distinct $u', v' \in W'$
    $e' = (u', v')$ with label L
    $E^{md} = E^{md} \cup \{e'\}$

Algorithm 3: Perform edge subdivision twice

Data: $G^{sm} = (V^{sm}, E^{sm})$, $(u, v) \in E^{sm}$
Result: Two new vertices in V^{sm} and one edge of E^{sm} replaced with three

$V^{sm} = V^{sm} \cup \{x, y\}$ where $x, y \notin V^{sm}$
$E^{sm} = (E^{sm} - \{(u, v)\}) \cup \{(u, x), (x, y), (y, v)\}$
Label $(u, x), (x, y), \text{and } (y, v)$ with layer label of (u, v)



Algorithm 2: Creating G^{sm} from G^{md}

Data: $G^{md} = (V^{md}, E^{md})$ with edge layer labels
Result: $G^{sm} = (V^{sm}, E^{sm})$

Instantiate empty set W

$G^{sm} = copy(G^{md})$

foreach $(u, v) \in E^{sm}$
  subdivide (u, v) twice (Algorithm 3)
  $W = W \cup \{u\}$ if $u \notin W$
  $W = W \cup \{v\}$ if $v \notin W$

foreach $w \in W$
  $P = \{(w, n) \in E^{sm}\ \forall n \in N(w)\}$
  partition P by layer label ($P = P_1 \cup \cdots \cup P_k$)

  foreach P_i partition
    merge P_i (Algorithm 4)

Algorithm 4: Merge a partition

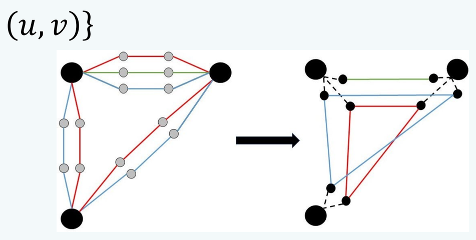
Data: $G^{sm} = (V^{sm}, E^{sm})$, edge set partition P, super-vertex w
Result: All vertices of $N(w)$ mapped to a single vertex. Each edge incident to these vertices mapped to edges incident with the new vertex.

Instantiate empty sets X_{rem} and Y_{rem}

$V^{sm} = V^{sm} \cup \{k\}$ where $k \notin V^{sm}$
$Y_{add} = \{(w, k)\}$

foreach $(w, u) \in P$
  foreach $v \in N(u)$ where $v \neq w$
    $Y_{rem} = Y_{rem} \cup \{(w, u), (u, v)\}$
    $X_{rem} = X_{rem} \cup \{u\}$
    $Y_{add} = Y_{add} \cup \{(k, v)\}$

$V^{sm} = V^{sm} - X_{rem}$
$E^{sm} = (E^{sm} - Y_{rem}) \cup Y_{add}$



Results

Figure 2: (top-left) multiplex network, $|V^{mp}| = 66$ and $|E^{mp}| = 142$
(top-right) multi-dimensional network, $|V^{md}| = 32$ and $|E^{mp}| = 91$
(bottom) simple network, $|V^{sm}| = 99$ and $|E^{sm}| = 170$

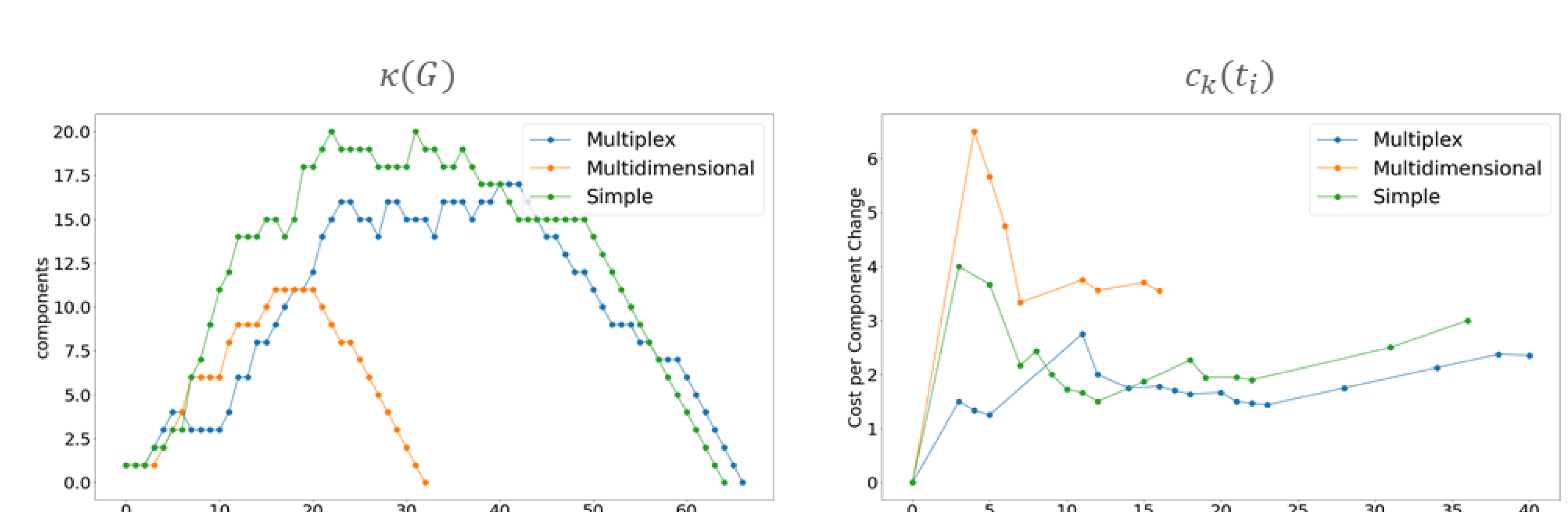


Figure 3: (left) number of components in each model for sequential node deletions. (right) accumulated cost to increase network fragmentation.