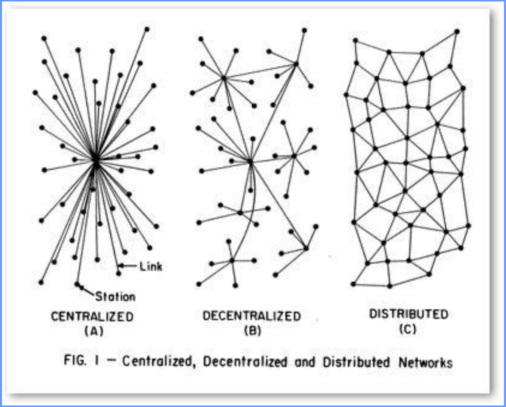
Compositional Design and Tasking of Networks


Link
Station
CENTRALIZED
(A)
DECENTRALIZED
(B)
DISTRIBUTED
(C)
FIG. 1 — Centralized, Decentralized and Distributed Networks

John D. Foley, Metron, Inc.

NIST, 15 March 2018

In theory, the construction of network operads is:

• Encode your favorite kind of network as a lax symmetric monoidal functor $F\colon S(C) \to \mathrm{Mon}$

• Apply the symmetric monoidal Grothendieck construction to get the symmetric monoidal category $(\int F, \otimes_F)$

• Let O_F be the endomorphism operad of $\int F$

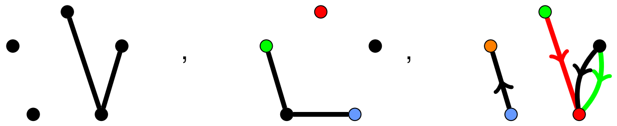
Theorem (Baez, Foley, Moeller, Pollard)

The composite functor

$$\mathrm{NetMod} \xrightarrow{\int} \mathrm{SMCat} \xrightarrow{\mathbf{op}(-)} \mathrm{Operads}$$

constructs a network operad O_F for each network model F.

In practice, you *have* a kind of network you wish to compose:



The challenge is to extract the essential features of composition, driven by a target application.

In our framework, you must answer two questions:

- How are networks placed side-by-side?

- How do networks change as you overlay edges?

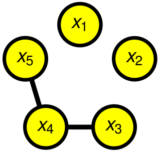
By building these features into a network model, an operad that provides formal instructions compose networks is obtained.

For the Complex Adaptive System Composition and Design Environment (CASCADE) project, an early application was **range-limited communication networks**.

For this application:

• Each vertex in a network has an attribute: a location x in $\mathbb{R}^2$.

• Edges in a network are constrained: two way communication between vertices is possible only if the distance between their attributes is less than L.



$d(x_i, x_{i+1}) < L$: Edge ok.

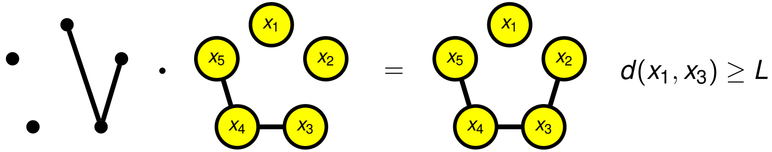
$d(x_i, x_{i+2}) \geq L$: No edge.

Q: How are range-limited communication networks placed side-by-side?

A: Take the disjoint union. This simple answer often works!

Q: How do these networks change as you overlay edges?

A: Comm links are added only if vertices are within range limits.



This means that simple graphs are sufficient to give *instructions* to compose range-limited communication networks.

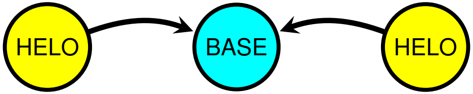
For CASCADE, our challenge is to **design and task** a search and rescue (SAR) System of System (SoS) to effectively rescue isolated personel (IP).

In our November scenario, design networks have

• Vertex colors encoding entity types: bases, platforms, . . .

• State attributes for vertices: locations, speed, range,...

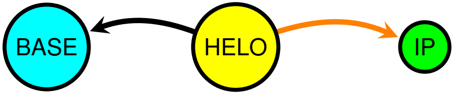
• Directed edges assign assets to bases, constrained by base capacity and runway length.



After a design network is selected based on the anticipated need for SAR, tasking occurs in response to IP events.

Design networks are extended[1] to task networks that also represent platforms given SAR tasks with a start time:

- A new vertex color for IP event.
- Edge colors for task relationships: recover, escort, . . .



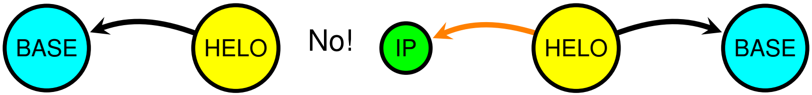
As in range-limited networks, some task constraints use state attributes from the *algebra* of task networks–e.g. Helicopter 1 at Base 2 does not that the range to recover IP group 3.

[1]Using an appropriate map of algebras

More fundamental constraints on design and tasking instructions are expressible in the *operad*–e.g.

• One base per platform, one task at a time.

• Asset must share a base in order to team.



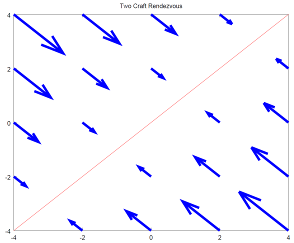
• This illustrates that more advanced network operads can deconflict composition instructions, in an *order dependent* way.

• Overlaying networks also allows structures or behaviors on a single network to be combined directly.

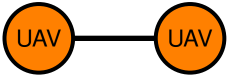
Suppose we want to coordinate the behavior of 2 UAVs with a single location attribute.

One simple cooperative behavior matches states via

$$\dot{x}_1(t) = x_2(t) - x_1(t)$$
$$\dot{x}_2(t) = x_1(t) - x_2(t)$$



Simple graphs on two vertices can encode:

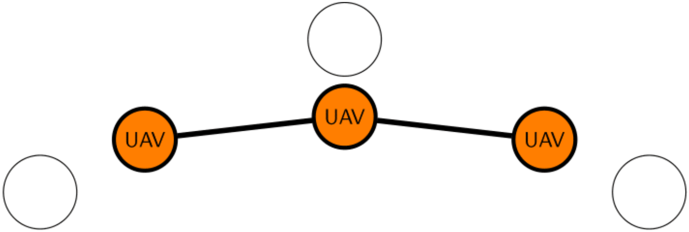
working together 

or not

with cooperation as above and non-cooperation simply static.

To coordinate the behavior of *n* UAVs, we can overlay these pairwise interactions via point-wise addition of vector fields:

$$\dot{x}_i(t) = \sum_{j-i\in g} x_j(t) - x_i(t)$$



Known results give global stability for any *connected* network. Further techniques are available to encode dynamics for formation control, sensor coverage, etc.

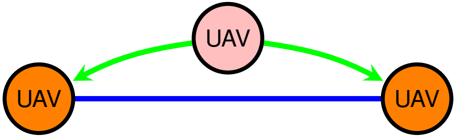
Of course, there are many other ways to ‘overlay dynamics’ with network operads.

For example, simple graphs can act by overlaying edges via:

$$\dot{x}_i(t) = \sum_{i \neq j} f_{I(j-i\in g)}(x_j(t) - x_i(t)), \quad f_k(-y) = -f_k(y), \quad k \in \{0, 1\}$$

or even other notions of ‘sum’–e.g. to enforce a max speed.

Current CASCADE work is combining decentralized control of formations–with ‘leader’ UAVs–with decentralized pursuit of coverage of an anticipate area of operations for SAR.



For more see:

• John Baez, John Foley, Joseph Moeller and Blake Pollard, Network models: arXiv:1711.00037.

• John Baez, Complex adaptive system design, *Azimuth*.

• Mehran Mesbahi and Magnus Egerstedt, *Graph Theoretic Methods for Multiagent Networks*, Chapter 7.