azengineering

# Blockmodeling and the Estimation of Evolutionary Architectural Growth in Major Defense Acquisition Programs

*LTC Matthew Dabkowski**
*Dr. Ricardo Valerdi*

Department of Systems & Industrial Engineering
College of Engineering
University of Arizona

* mfd1@email.arizona.edu

*Disclaimer: This material is based upon work supported by the Naval Postgraduate School Acquisition Research Program under Grant No. N00244-13-1-0032, and the Office of the Secretary of Defense. The views expressed in written materials or publications, and/or made by speakers, moderators, and presenters, do not necessarily reflect the official policies of the Naval Postgraduate School nor does mention of trade names, commercial practices, or organizations imply endorsement by the U.S. Government.*

# Agenda

- Purpose, Question, and Contribution
- Challenge and Opportunity in Pre-MS A Cost Analysis
- Mapping DoDAF to COSYSMO
- Leveraging SE and Exploiting the SV-3
- Estimating Unforeseen Architectural Growth in MDAPs
  - Microstructure
  - Macrostructure
- Simulating Growth and Estimating Cost
- "Blockmodeling" Beyond Architectural Communities
- Future work
- Questions

**Overarching Purpose**: To transform Model-Based System Engineering (MBSE) artifacts into computational knowledge that can be leveraged early in the system lifecycle when uncertainty is high and confidence is low

↓

**Focused Question**: Can parametric cost estimation, in conjunction with DoD Architecture Framework (DoDAF) models, capture the monetary impact of architectural changes early in the system lifecycle?

↓

**Principal Contribution**: A network science-based algorithm for estimating the cost of unforeseen architectural growth

# Challenge and Opportunity in Pre-MS A Cost Analysis

**We find ourselves in challenging times . . .**

- Sequestration in 2013 + CRs =
  Reduced production +
  Hard modernization decisions +
  ⋯ + **Difficult cost planning**

**. . . and times were already tough . . .**

- 1997-2009: 47 MDAPs had cost overruns of at least 15%/30% over their current/baseline estimates

**. . . especially early in the life cycle . . .**

- ~ 28% of a system's baseline requirements will change

**. . . as late adds carry substantial costs.**

- 2014: 6 of 14 largest cost overruns due to new capabilities

**But there is an appetite for change . . .**

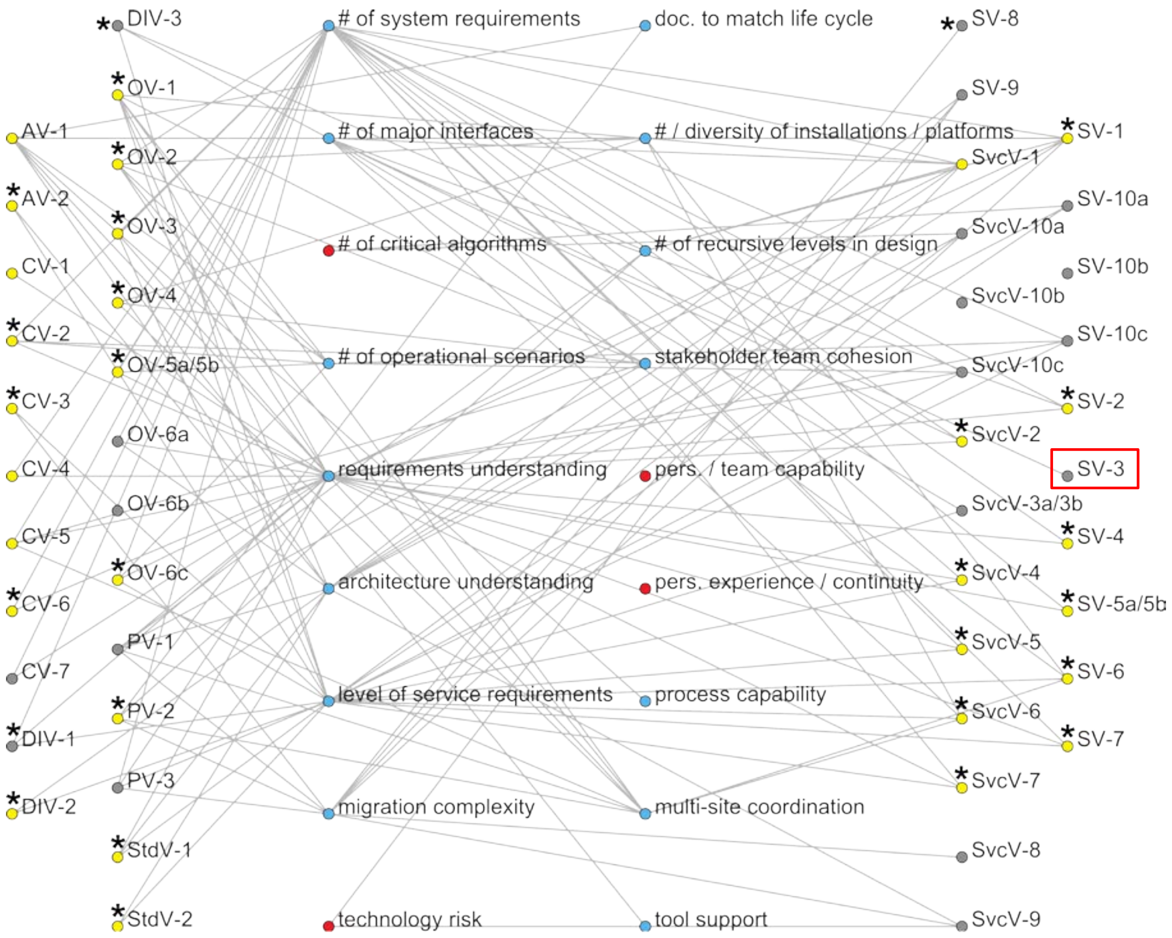
- *WSARA* (2009): Increased the rigor of Pre-MS A cost analysis (baseline shifted from MS B to MS A)
- *DoDI 5000.02* (2013): Mandated a draft CDD, with required DoDAF models, be submitted Pre-MS A

**. . . and this presents an opportunity.**

- DoDAF includes factors that influence system engineering (SE) effort (e.g., interfaces)
- COSYSMO estimates SE effort

***DoDAF's models appear to map to COSYSMO's parameters***

# DoDAF models required Pre-MS A nearly span COSYSMO's drivers*



***78% of COSYSMO's drivers map to DoDAF models submitted early in the life cycle***

Legend

▬ - DoDAF model X is relevant for rating COSYSMO driver Y

🟡 - Model required Pre-MS A (2012-2015)

* - Model required Pre-MS A (2015-)

🔵 - Pre-MS A model(s) maps to driver

🔴 - No Pre-MS A model maps to driver

| | |
|---|---|
| AV | - All (2 models) |
| CV | - Capability (7 models) |
| DIV | - Data and information (3 models) |
| OV | - Operational (9 models) |
| PV | - Project (3 models) |
| StdV | - Standards (2 models) |
| SvcV | - Services (13 models) |
| SV | - Systems (13 models) |

* Valerdi, R., Dabkowski, M., & Dixit, I. (2015). Reliability Improvement of Major Defense Acquisition Program Cost Estimates – Mapping DoDAF to COSYSMO. *Systems Engineering*, 18(5), 530-547. doi:10.1002/sys.21327

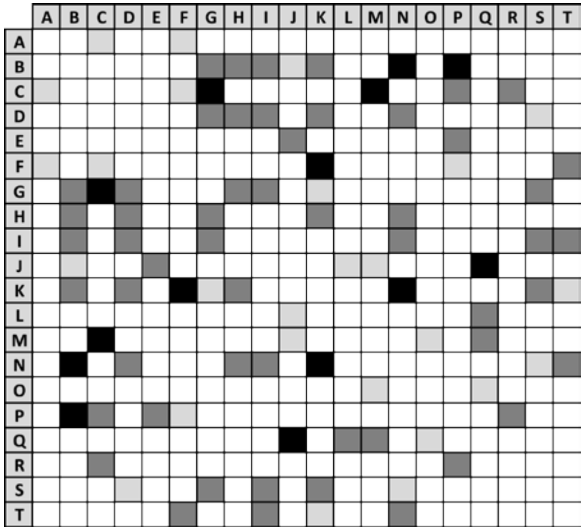
# Leveraging SE and Exploiting the SV-3

- From the 2008 National Research Council report “Pre-Milestone A and Early-Phase Systems Engineering” . . .
  - The “application of SE to decisions made in the pre-Milestone A period is critical to avoiding (or at least minimizing) cost and schedule overruns” (p. 3)
  - 3 of the 6 primary drivers of cost growth addressable by SE are:
    1. Incomplete requirements at MS B,
    2. System complexity (via internal, architectural design), and
    3. External interface complexity (via network-centric operations or “systems of systems” constructs) (pp. 82-85)
- The SV-3 (or Systems-Systems Matrix) provides an abstraction of all 3, as requirements (however incomplete) drive the selection of subsystems (nodes) which are connected by interfaces (edges), both internal and external

***Formally evaluating the SV-3 Pre-MS A and estimating its potential growth holds promise for minimizing cost overruns***

# Hypothetical SV-3

- 20 subsystems with 47 interfaces of varying complexity
- Without loss of generality, assume there are . . .
  - 200 easy, 200 nominal, and 50 difficult requirements
  - 5 difficult critical algorithms
- Using additional $w_{ik}$ and $EM_j$ data,* apply CER to obtain an initial estimate of $PM_{NS}$



| | A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| A | | | □ | | | □ | | | | | | | | | | | | | | |
| B | | | | | | | ▨ | ▨ | ▨ | □ | ▨ | | | ■ | | ■ | | | | |
| C | □ | | | | | □ | ■ | | | | | | ■ | | | ▨ | | ▨ | | |
| D | | | | | | | ▨ | ▨ | ▨ | | ▨ | | | ▨ | | | | | □ | |
| E | | | | | | | | | | ▨ | | | | | | ▨ | | | | |
| F | □ | | □ | | | | | | | | ■ | | | | | □ | | | | ▨ |
| G | | ▨ | ■ | ▨ | | | | ▨ | ▨ | | □ | | | | | | | | ▨ | |
| H | | ▨ | | ▨ | | | ▨ | | | | ▨ | | | ▨ | | | | | | |
| I | | ▨ | | ▨ | | | ▨ | | | | | | | ▨ | | | | | ▨ | ▨ |
| J | | □ | | | ▨ | | | | | | | □ | □ | | | | ■ | | | |
| K | | ▨ | | ▨ | | ■ | □ | ▨ | | | | | | ■ | | | | | ▨ | □ |
| L | | | | | | | | | | □ | | | | | | | ▨ | | | |
| M | | | ■ | | | | | | | □ | | | | | □ | | ▨ | | | |
| N | | ■ | | ▨ | | | | ▨ | ▨ | | ■ | | | | | | | | □ | ▨ |
| O | | | | | | | | | | | | | □ | | | | □ | | | |
| P | | ■ | ▨ | | ▨ | □ | | | | | | | | | | | | ▨ | | |
| Q | | | | | | | | | | ■ | | ▨ | ▨ | | □ | | | | | |
| R | | | ▨ | | | | | | | | | | | | | ▨ | | | | |
| S | | | | □ | | | ▨ | | ▨ | | ▨ | | | □ | | | | | | |
| T | | | | | | ▨ | | | ▨ | | □ | | | ▨ | | | | | | |

SV-3

Interface Complexity

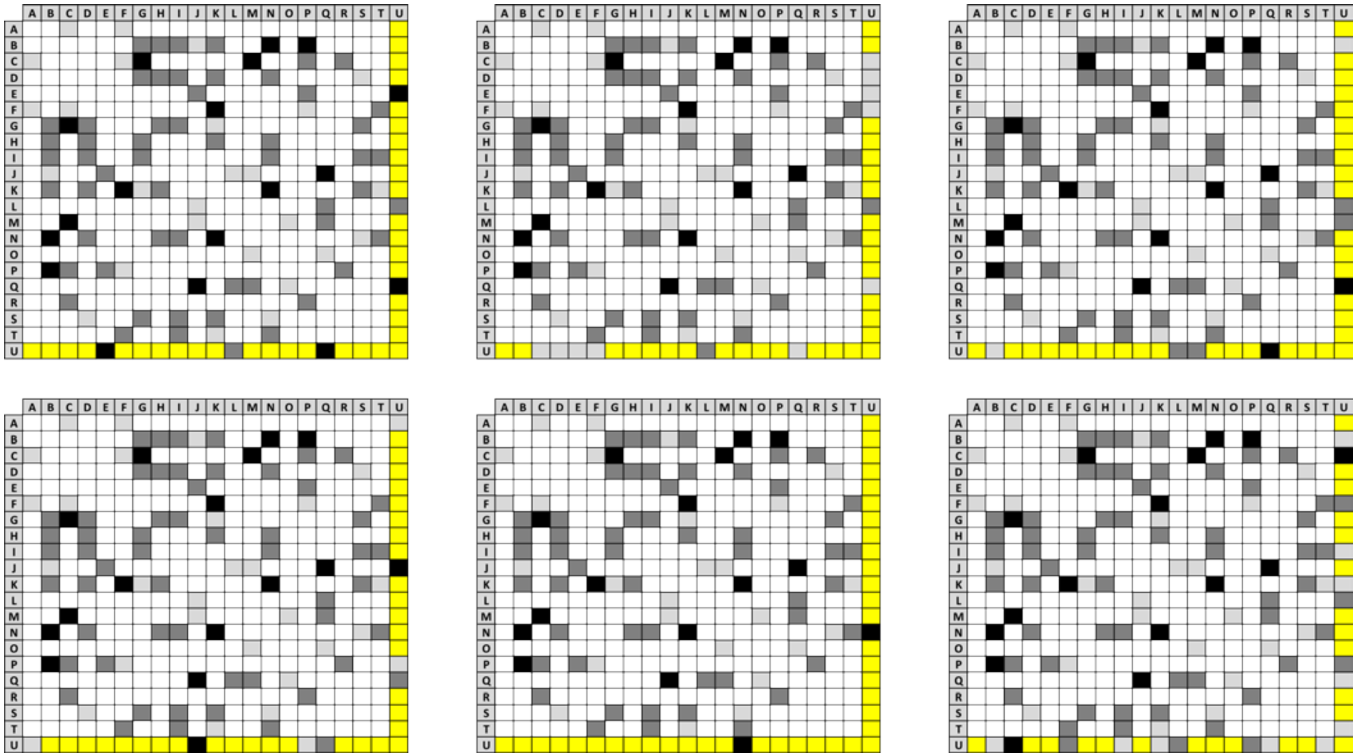
□ = Easy, ▨= Nominal, ■ = Difficult

$$PM_{NS} = 0.25\cdot\left(\underbrace{(0.5\times 200+1.0\times 200+5.0\times 50)}_{\text{requirements}}+\underbrace{(11.5\times 5)}_{\text{algorithms}}+\underbrace{(1.1\times 13+2.8\times 27+6.3\times 7)}_{\text{interfaces}}\right)^{1.06}\cdot 0.89 = 245.27$$

* Valerdi, R. (2008). *The Constructive Systems Engineering Cost Model (COSYSMO): Quantifying the Costs of Systems Engineering Effort in Complex Systems*. Saarbrücken, Germany: VDM Verlag.

# What about inevitable, unforeseen change?

- This is Pre-MS A ⇒ requirements will change



***If we add a new subsystem U to the existing architecture, how will it connect?***

***What will it cost?***

# The analytical task

***Estimate the number of interfaces***
***(by complexity level) U will generate***

(Q1) How many subsystems should **U** connect to (**degree, *m***)?,

(Q2) If **U** connects to *m* subsystems, which *m* subsystems should it connect to (**adjacency**)?, and

(Q3) If **U** connects to a specific set of *m* subsystems, what should the complexity of these interfaces be (**weights**)?

# Network Science – A mechanism for generating unforeseen architectural growth (microstructure)

Fundamental assumption: Current architecture foretells future architecture

*Degree*: Treat degree of **U** ($M_U$) as a random variable with PMF equal to the degree distribution of the existing system ("rich-by-birth") (Dorogovtsev & Mendes, 2003)

| $m$ | 2 | 4 | 5 | 6 | 7 | 8 |
|---|---|---|---|---|---|---|
| $n_m$ | 5 | 3 | 5 | 3 | 3 | 1 |
| $P(M_U = m)$ | 0.25 | 0.15 | 0.25 | 0.15 | 0.15 | 0.05 |

*Adjacency*: Utilize Barabási–Albert (1999) preferential attachment (PA) model, where highly connected subsystems are more likely to interface with **U** ("rich-get-richer")

| System ($i$) | A | B | C | D | E | F | G | H | I | J |
|---|---|---|---|---|---|---|---|---|---|---|
| $d_i$ | 2 | 7 | 6 | 6 | 2 | 5 | 7 | 5 | 6 | 5 |
| $p_i$ | 0.021 | 0.074 | 0.064 | 0.064 | 0.021 | 0.053 | 0.074 | 0.053 | 0.064 | 0.053 |
| **System ($i$)** | **K** | **L** | **M** | **N** | **O** | **P** | **Q** | **R** | **S** | **T** |
| $d_i$ | 8 | 2 | 4 | 7 | 2 | 5 | 4 | 2 | 5 | 4 |
| $p_i$ | 0.085 | 0.021 | 0.043 | 0.074 | 0.021 | 0.053 | 0.043 | 0.021 | 0.053 | 0.043 |

$$p_i = \frac{d_i}{\sum_{j=1}^{N} d_j}$$

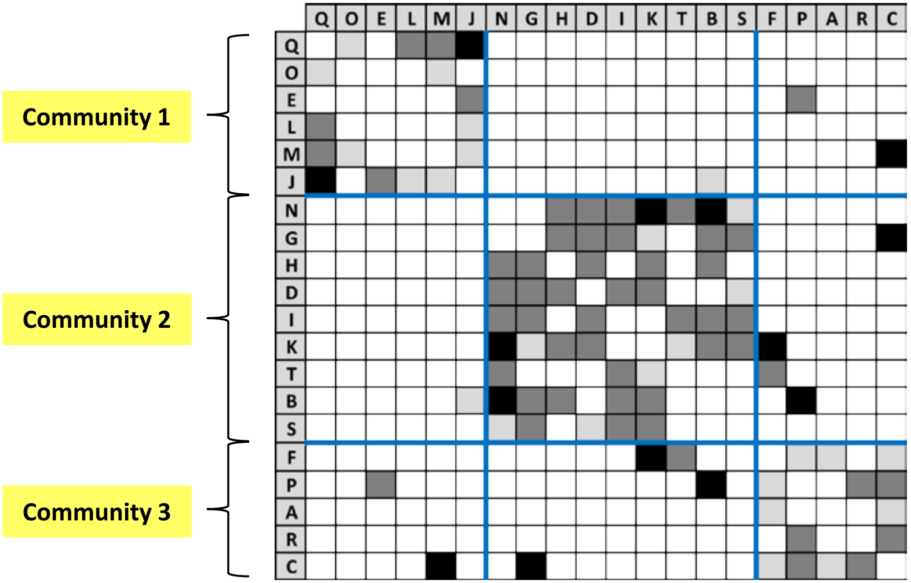
*Weights*: Model complexity of the interface between **U** and subsystem ***i*** ($w_{iU}$) as a random variable, where the pmf for $w_{iU}$ is ***i***'s interface complexity distribution

# Network Science – A mechanism for generating unforeseen architectural growth (macrostructure)

Fundamental assumption: Current architecture foretells future architecture

From *The Art of Systems Architecting*: "The most important aggregation and partitioning heuristics are to **minimize external coupling** and **maximize internal cohesion**"*

*Architectural communities*: Utilize Girvan-Newman (2002) to identify groups of subsystems such that the number of interfaces is sparse between and dense within groups


Community 1
Community 2
Community 3
Q O E L M J N G H D I K T B S F P A R C
Q O E L M J N G H D I K T B S F P A R C

| Intracommunity<br>Community | Δ | Intercommunity<br>Communities | Δ |
|---|---|---|---|
| 1: {Q, O, E, L, M, J} | 0.5333 | 1 and 2 | 0.0095 |
| 2: {N, G, H, D, I, K, T, B, S} | 0.6944 | 1 and 3 | 0.0364 |
| 3: {F, P, A, R, C} | 0.7000 | 2 and 3 | 0.0440 |

* Maier, M., & Rechtin, E. (2000). *The Art of Systems Architecting*. (2nd ed.). New York, NY: CRC Press.

# Simulating Growth and Estimating Cost – Dabkowski et al. (2014)

For a specified number of iterations . . .

*Preprocessing*

1. Initialize the system as the current system

2. Use Girvan-Newman (2002) to identify architectural communities

3. Randomly assign **U** to community *k*

*Intracommunity Growth*

4. Generate a realization for $M_{\mathbf{U},intra}$ given **U** is assigned to community *k* ($m_{intra}$)

5. Connect **U** to $m_{intra}$ subsystems inside community *k* using the BA model

6. For each interface established in (5), assign complexity ($w_{i\mathbf{U},intra}$)

*Intercommunity Growth*

7. Generate a realization for $M_{\mathbf{U},inter}$ given **U** is assigned to community *k* ($m_{inter}$)

8. Connect **U** to $m_{inter}$ communities using the BA model

9. For each interface established in (8), assign complexity ($w_{i\mathbf{U},inter}$)

*Cost Estimation*

10. Estimate cost for augmented system using COSYSMO ($PM_{NS}$*)

11. Calculate additional cost of adding subsystem **U** ($PM_{NS}$* − $PM_{NS}$)

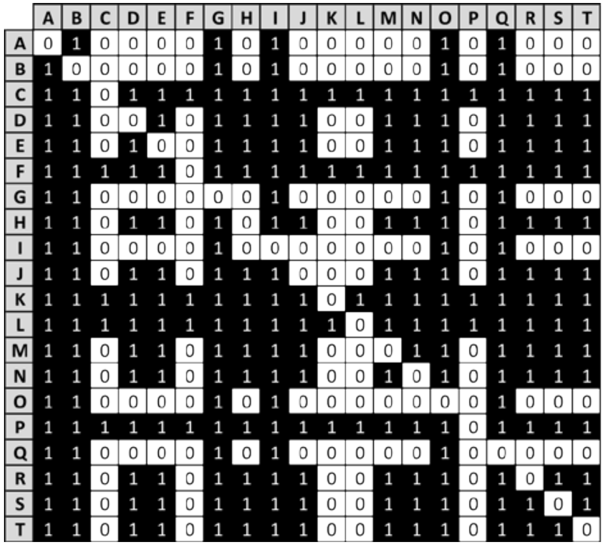
12. Store results and return to (3)

*"Did I build the right model?* ***Is it general enough?"***

# Community detection may miss key macrostructure . . .

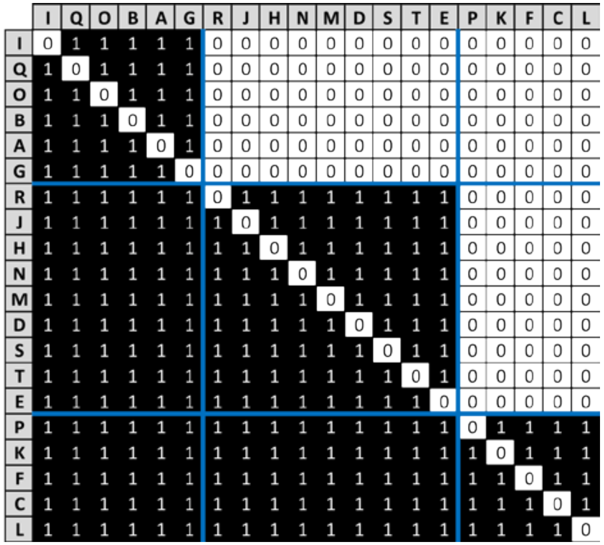
Hypothetical SV-3

Original representation



| | A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| A | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 |
| B | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 |
| C | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| D | 1 | 1 | 0 | 0 | 1 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 1 |
| E | 1 | 1 | 0 | 1 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 1 |
| F | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| G | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 |
| H | 1 | 1 | 0 | 1 | 1 | 0 | 1 | 0 | 1 | 1 | 0 | 0 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 1 |
| I | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 |
| J | 1 | 1 | 0 | 1 | 1 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 1 |
| K | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| L | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| M | 1 | 1 | 0 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 1 | 1 | 1 |
| N | 1 | 1 | 0 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 1 | 1 | 1 |
| O | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| P | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 1 |
| Q | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 |
| R | 1 | 1 | 0 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 1 | 1 | 1 | 0 | 1 | 0 | 1 | 1 |
| S | 1 | 1 | 0 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 1 | 1 | 1 | 0 | 1 | 1 | 0 | 1 |
| T | 1 | 1 | 0 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 0 |

Representation following permutation



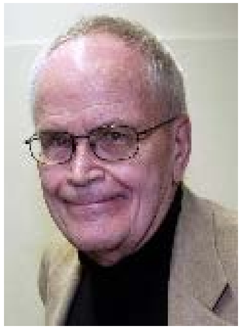
| | I | Q | O | B | A | G | R | J | H | N | M | D | S | T | E | P | K | F | C | L |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| I | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Q | 1 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| O | 1 | 1 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| B | 1 | 1 | 1 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| A | 1 | 1 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| G | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 |
| J | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 |
| H | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 |
| N | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 |
| M | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 |
| D | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 |
| S | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 |
| T | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 |
| E | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| P | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 1 |
| K | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 1 |
| F | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 1 |
| C | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 1 |
| L | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 |

- $N = 20$ subsystems and $E = 251$ *directed* interfaces; relatively dense ($\Delta = 0.661$)
- Girvan-Newman (2002) identifies 6 architectural communities with a modularity of just 0.017

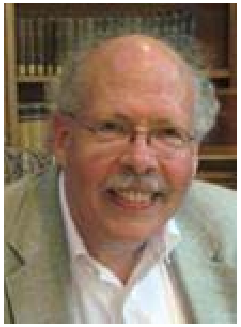
***Girvan-Newman misses the indisputable, hierarchical clustering of subsystems!***

# . . . but blockmodeling does not.

- Blockmodeling
  - Partitions a network consisting of $i = 1, \cdots, N$ objects (i.e., the SV-3) into $k = 1, \cdots, P$ non-overlapping positions, where the positions generally abide the structure represented in a $(P \times P)$ image matrix such that $P \ll N$
  - Developed by computational sociologists at Harvard in the mid-1970's



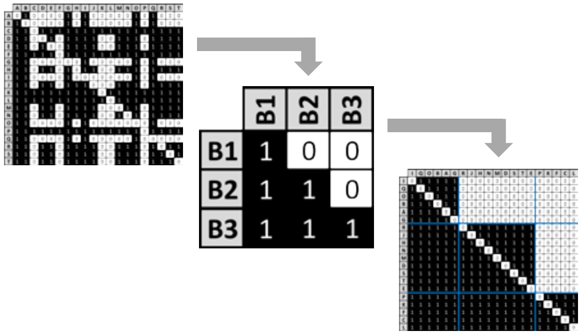
**Harrison White**



**Scott Boorman**

**Ronald Breiger**

**Social structure from multiple networks. I. Blockmodels of roles and positions. *AJS*, 81(4), 730-780.**

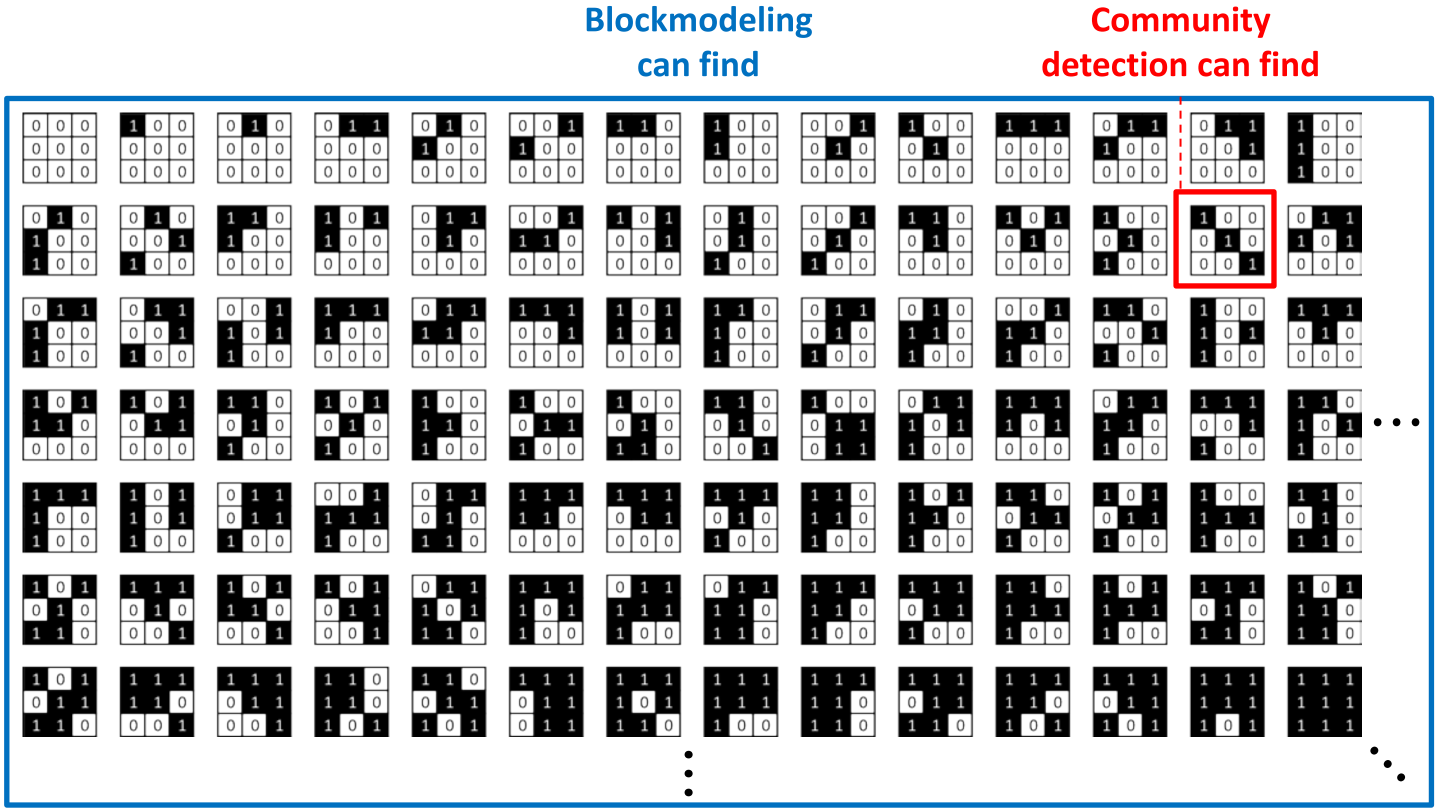


| | B1 | B2 | B3 |
|---|---|---|---|
| B1 | 1 | 0 | 0 |
| B2 | 1 | 1 | 0 |
| B3 | 1 | 1 | 1 |

***Pajek recovers the SV-3's hierarchical structure exactly!***

  - Integrated into popular network analysis software (i.e., Pajek via Doreian, Batagelj, and Ferligoj's (2005) direct approach)

# Blockmodeling is the natural generalization of community detection . . .



***Of the 512 possible $(3 \times 3)$ binary image matrices, community detection can find a partition for 1 – the identity; blockmodeling can accommodate all 512!***

# . . . but blockmodeling is not a panacea.

- Issue #1: Blockmodeling (BM) problems are NP-hard ⇒ time to find globally optimal solutions can *explode* as the # of subsystems/positions ↑

  Consequence #1: BM normally applies heuristics versus exact methods ⇒ better fitting image matrices and partitions may exist

- Issue #2: Exact methods largely confined to *confirmatory* fitting (image matrix is pre-specified) ⇒ exact *exploratory* fitting procedures are lacking

  Consequence #2: An SV-3's macrostructure is not "known in advance" ⇒ available exact BM methods are ill-suited for the task at hand

- Issue #3: Majority of BM heuristics and all exact methods focus on single one-/two-mode networks ⇒ BM multiple relations is an open problem

  Consequence #3: SV-3s are often mixed-mode networks ⇒ new methods are required to accommodate all SV-3s

# WANTED . . . An Efficient Exact Method for Blockmodeling Mixed-Mode SV-3s

Given:

A mixed-mode SV-3

1-mode portion 2-mode portion

| | | Internal Subsystems | | | | | | | | | | External Subsystems | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | I1 | I2 | I3 | I4 | I5 | I6 | I7 | I8 | I9 | I10 | E1 | E2 | E3 | E4 | E5 | E6 | E7 |
| Internal Subsystems | I1 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | I2 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | I3 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | I4 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 |
| | I5 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 |
| | I6 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 |
| | I7 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 |
| | I8 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 |
| | I9 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 1 |
| | I10 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 |

Find:

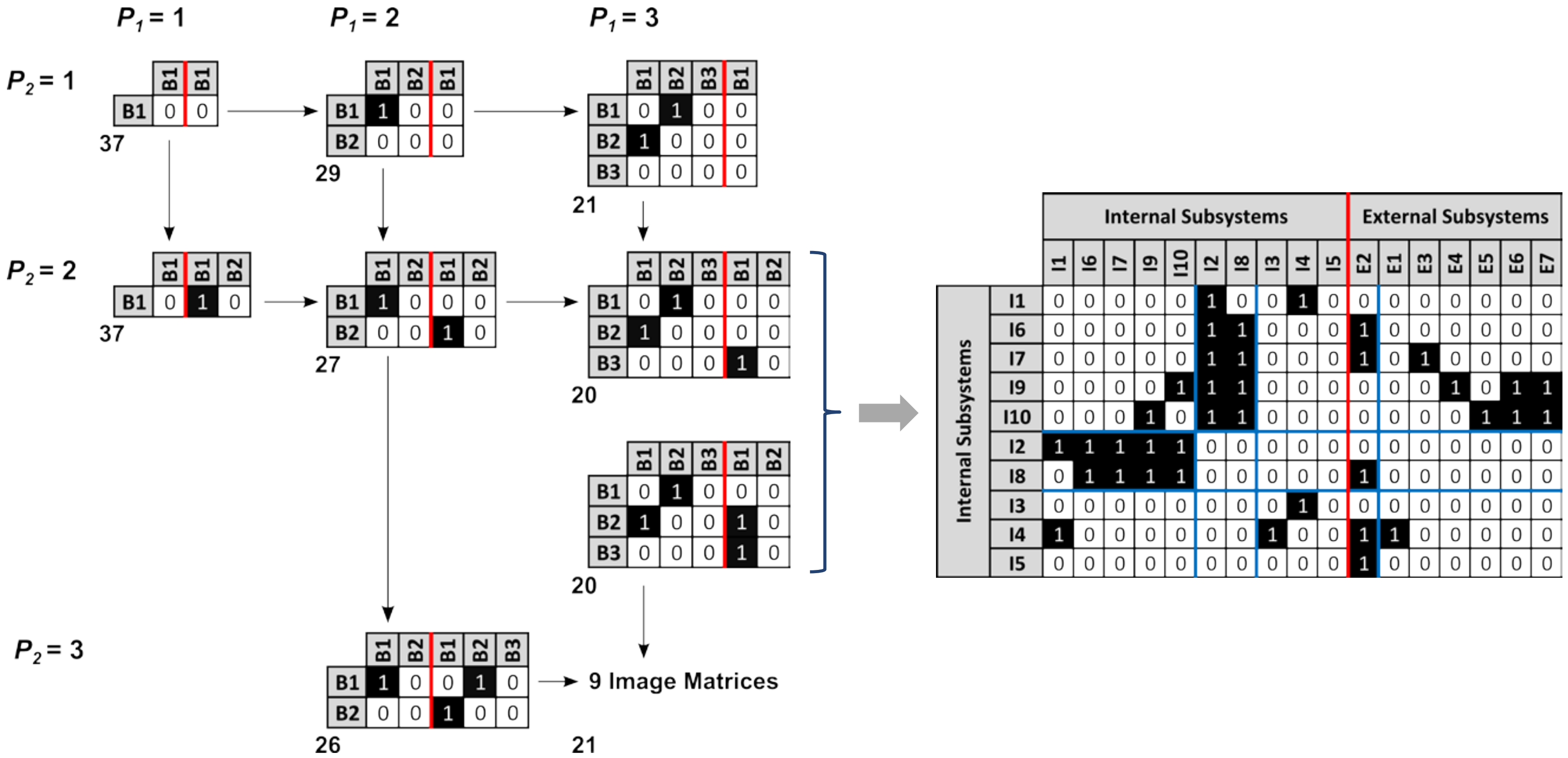
The globally optimal mixed-mode image matrix and corresponding partition with three or fewer internal and external subsystem positions

| | | Internal Subsystems | | | External Subsystems | | |
|---|---|---|---|---|---|---|---|
| | | B1 | B2 | B3 | B1 | B2 | B3 |
| Internal Subsystems | B1 | 0 or 1 | 0 or 1 | 0 or 1 | 0 or 1 | 0 or 1 | 0 or 1 |
| | B2 | 0 or 1 | 0 or 1 | 0 or 1 | 0 or 1 | 0 or 1 | 0 or 1 |
| | B3 | 0 or 1 | 0 or 1 | 0 or 1 | 0 or 1 | 0 or 1 | 0 or 1 |

Idea: Leverage the results in Brusco and Steinley's (2009) paper "Integer programs for one- and two-mode blockmodeling based on prespecified image matrices for structural and regular equivalence"

# Globally Optimal IMs and Partition

- Formulated a series of IPs using C++; solved in IBM's ILOG CPLEX



$P_1 = 1$, $P_2 = 1$ (37)

|  | B1 | B1 |
|---|---|---|
| B1 | 0 | 0 |

$P_1 = 2$, $P_2 = 1$ (29)

|  | B1 | B2 | B1 |
|---|---|---|---|
| B1 | 1 | 0 | 0 |
| B2 | 0 | 0 | 0 |

$P_1 = 3$, $P_2 = 1$ (21)

|  | B1 | B2 | B3 | B1 |
|---|---|---|---|---|
| B1 | 0 | 1 | 0 | 0 |
| B2 | 1 | 0 | 0 | 0 |
| B3 | 0 | 0 | 0 | 0 |

$P_1 = 1$, $P_2 = 2$ (37)

|  | B1 | B1 | B2 |
|---|---|---|---|
| B1 | 0 | 1 | 0 |

$P_1 = 2$, $P_2 = 2$ (27)

|  | B1 | B2 | B1 | B2 |
|---|---|---|---|---|
| B1 | 1 | 0 | 0 | 0 |
| B2 | 0 | 0 | 1 | 0 |

$P_1 = 3$, $P_2 = 2$ (20)

|  | B1 | B2 | B3 | B1 | B2 |
|---|---|---|---|---|---|
| B1 | 0 | 1 | 0 | 0 | 0 |
| B2 | 1 | 0 | 0 | 0 | 0 |
| B3 | 0 | 0 | 0 | 1 | 0 |

(20)

|  | B1 | B2 | B3 | B1 | B2 |
|---|---|---|---|---|---|
| B1 | 0 | 1 | 0 | 0 | 0 |
| B2 | 1 | 0 | 0 | 1 | 0 |
| B3 | 0 | 0 | 0 | 1 | 0 |

$P_1 = 2$, $P_2 = 3$ (26)

|  | B1 | B2 | B1 | B2 | B3 |
|---|---|---|---|---|---|
| B1 | 1 | 0 | 0 | 1 | 0 |
| B2 | 0 | 0 | 1 | 0 | 0 |

→ 9 Image Matrices (21)

|  |  | Internal Subsystems |  |  |  |  |  |  |  |  |  | External Subsystems |  |  |  |  |  |  |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
|  |  | I1 | I6 | I7 | I9 | I10 | I2 | I8 | I3 | I4 | I5 | E2 | E1 | E3 | E4 | E5 | E6 | E7 |
| Internal Subsystems | I1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
|  | I6 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
|  | I7 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 |
|  | I9 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 1 |
|  | I10 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 |
|  | I2 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
|  | I8 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
|  | I3 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
|  | I4 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 |
|  | I5 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |

***Partition of internal subsystems driven by "outside" interfaces***

# Generalizing Dabkowski et al. (2014) via Blockmodeling

For a specified number of iterations . . .

*Preprocessing*

1. Initialize the system as the current system

2. Build an optimal set of $\{P(M=m),\beta\}$ pairs

3. Use Dabkowski-Fan-Breiger (2015; 2016) to identify an optimal $P$-position image matrix and partition of subsystems

*Growth*

4. Randomly select a member from the optimal set of $\{P(M=m),\beta\}$ pairs

5. Generate a realization for the incoming subsystem's (**X**'s) number of interfaces using $P(M=m)$; **if the IM and partition suggest a compelling, underlying architectural structure**, use *Connection Option A*; otherwise, use *Connection Option B*

*Connection Option A (use blockmodel)*

6a. Randomly assign **X** to position *k*,

6b. Model assignment of **X**'s $m$ interfaces to positions as a random $(1 \times P)$ vector $\boldsymbol{C}$, where $\boldsymbol{C}$ follows a Multinomial$(m, \boldsymbol{p})$ distribution and $\boldsymbol{p}$ is the $(1 \times P)$ vector of multinomial probabilities given by

$$\frac{\text{\# interfaces in block } (k,l) \text{ of the partitioned, permuted SV−3}}{\text{\# interfaces in blocks } (k,\bullet) \text{ of the partitioned, permuted SV−3}};$$

generate a feasible realization for $\boldsymbol{C}$

6c. For $l = 1, \cdots, P$, attach **X** to $\boldsymbol{c}_l$ subsystems inside position $l$ using attachment probabilities $p_i = {d_i}^{\beta} / \sum_{j=1}^{N} {d_j}^{\beta}$

6d. For each interface established in (6c), assign complexity ($w_{i\mathbf{X}}$)

*Connection Option B (do not use blockmodel)*

6a. Attach **X** to $m$ subsystems using attachment probabilities $p_i = {d_i}^{\beta} / \sum_{j=1}^{N} {d_j}^{\beta}$

6b. For each interface established in (6a), assign complexity ($w_{i\mathbf{X}}$)

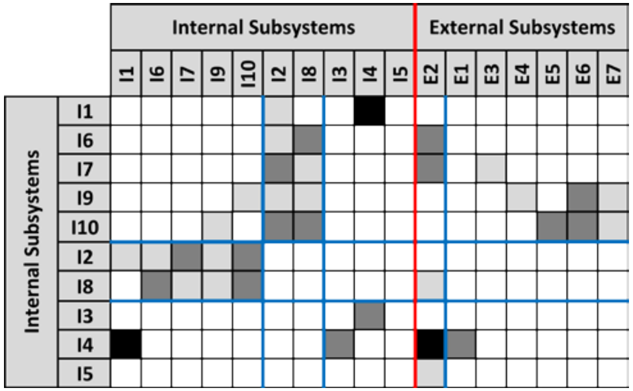
*Cost Estimation (same as Dabkowski et. al (2014), goto Step 4)*

# Estimating the Cost of Architectural Growth

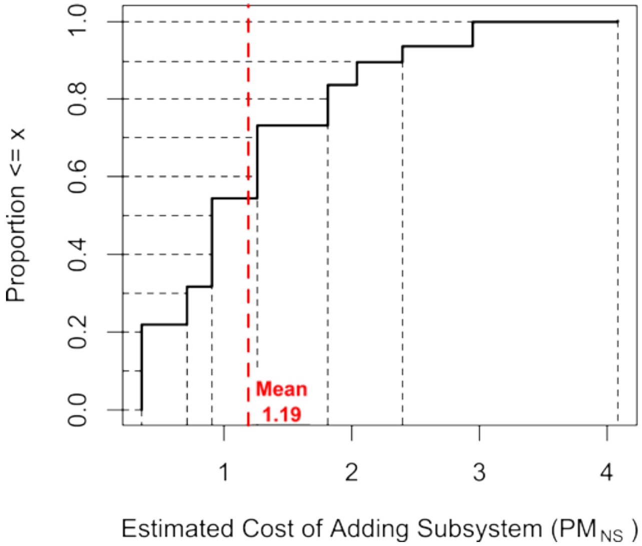
- Assumed the following:
  - $A = 0.25; E = 1.06; \prod_{j=1}^{14} EM_j = 0.89$
  - Requirements: 75 easy, 50 nominal, 10 difficult
  - Internal interfaces: 6 easy, 5 nominal, 1 difficult
  - External interfaces: 6 easy, 6 nominal, 1 difficult

  $\Rightarrow$ 59.24 $PM_{NS}$ of SE effort required
- Coded algorithm in R
- Ran 10,000 iterations

***Expected cost to connect an additional subsystem to the SV-3's internal subsystems is (1.177, 1.206) PM$_{NS}$***



| | | Internal Subsystems | | | | | | | | | | External Subsystems | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | I1 | I6 | I7 | I9 | I10 | I2 | I8 | I3 | I4 | I5 | E2 | E1 | E3 | E4 | E5 | E6 | E7 |
| Internal Subsystems | I1 | | | | | | ░ | | | █ | | | | | | | | |
| | I6 | | | | | | ░ | ▓ | | | | ▓ | | | | | | |
| | I7 | | | | | | ▓ | ░ | | | | ▓ | | ░ | | | | |
| | I9 | | | | | ░ | ░ | ░ | | | | | | | ░ | | ▓ | ░ |
| | I10 | | | | ░ | | ▓ | ▓ | | | | | | | | ▓ | ▓ | ░ |
| | I2 | ░ | ░ | ▓ | ░ | ▓ | | | | | | | | | | | | |
| | I8 | | ▓ | ░ | ░ | ▓ | | | | | | ░ | | | | | | |
| | I3 | | | | | | | | | ▓ | | | | | | | | |
| | I4 | █ | | | | | | | ▓ | | | █ | ▓ | | | | | |
| | I5 | | | | | | | | | | | ░ | | | | | | |



# Future Work

- Gather additional data for further validation and refinement
  - Secure sponsored research to weight SV-3s by interface complexity
  - Work with PMs to obtain multiple snapshots of SV-3s over time
- Explore additional connection options (e.g., model the probability that subsystem **X** is assigned to position *k* as a function of position *k*'s size)
- Modify algorithm to address external architectural growth
- Investigate the evolution of non-DoD architectures (e.g., open-source software architectures, non-militarized space systems, etc.)

# Questions

**CORRESPONDING AUTHOR**:

LTC Matt Dabkowski

matthew.dabkowski@gmail.com

matthew.f.dabkowski.mil@mail.mil

**PRESENTATION TITLE**: Blockmodeling and the Estimation of Evolutionary Architectural Growth in Major Defense Acquisition Programs

# IP formulation

**For all possible mixed-mode image matrices solve . . .**

**Data**

$\boldsymbol{S}^{(1)}$: $(N_1 \times N_1)$ 1-mode portion of the SV-3

$\boldsymbol{S}^{(2)}$: $(N_1 \times N_2)$ 2-mode portion of the SV-3

$\left(\boldsymbol{B}_1^{(r)} | \boldsymbol{B}_2^{(q)}\right)$: Binary $(P_1 \times P_1 | P_1 \times P_2)$ mixed-mode image matrix (IM), where $\boldsymbol{B}_1^{(r)}$ and $\boldsymbol{B}_2^{(q)}$ are the 1- and 2-mode portions

**Indices**

$i, j \in \mathcal{N}_1 \quad m \in \mathcal{N}_2$

$p, l \in \mathcal{P}_1 \quad k \in \mathcal{P}_2$

**Decision variables**

When using image matrix $\left(\boldsymbol{B}_1^{(r)} | \boldsymbol{B}_2^{(q)}\right)$:

$x_{i,p}^{(r,q)} = 1 \Rightarrow$ internal subsystem $i$ assigned to internal position $p$

$y_{m,k}^{(r,q)} = 1 \Rightarrow$ external subsystem $m$ assigned to external position $k$

$$\min \underbrace{\sum_{p,l=1}^{P_1} \sum_{\substack{i,j=1 \\ i \neq j}}^{N_1} w_{i,j,p,l}^{(r,q)} \left(b_{p,l}^{(r)} + s_{i,j}^{(1)} - 2 b_{p,l}^{(r)} s_{i,j}^{(1)}\right)}_{\text{Inconsistencies between one-mode IM and partition of internal subsystems}} + \underbrace{\sum_{i=1}^{N_1} \sum_{m=1}^{N_2} \sum_{p=1}^{P_1} \sum_{k=1}^{P_2} z_{i,m,p,k}^{(r,q)} \left(b_{p,k}^{(q)} + s_{i,m}^{(2)} - 2 b_{p,k}^{(q)} s_{i,m}^{(2)}\right)}_{\text{Inconsistencies between two-mode IM and partitions of internal and external subsystems}}$$

s.t.

$$\sum_{p=1}^{P_1} x_{i,p}^{(r,q)} = 1, \qquad \forall\, i \in \mathcal{N}_1$$

$$\sum_{k=1}^{P_2} y_{m,k}^{(r,q)} = 1, \qquad \forall\, m \in \mathcal{N}_2$$

Each internal / external subsystem assigned to a single internal / external position

$$\sum_{i=1}^{N_1} x_{i,p}^{(r,q)} \geq 1, \qquad \forall\, p \in \mathcal{P}_1$$

$$\sum_{m=1}^{N_2} y_{m,k}^{(r,q)} \geq 1, \qquad \forall\, k \in \mathcal{P}_2$$

No “empty” internal / external positions

$$x_{i,p}^{(r,q)} \in \{0,1\}, \qquad \forall\, i \in \mathcal{N}_1, p \in \mathcal{P}_1$$

$$y_{m,k}^{(r,q)} \in \{0,1\}, \qquad \forall\, m \in \mathcal{N}_2, k \in \mathcal{P}_2$$

Restrict decision variables to be binary

$$\begin{array}{lcll}
w_{i,j,p,l}^{(r,q)} & \leq & x_{i,p}^{(r,q)}, & \forall\, i,j \in \mathcal{N}_1, p,l \in \mathcal{P}_1 \\
w_{i,j,p,l}^{(r,q)} & \leq & x_{j,l}^{(r,q)}, & \forall\, i,j \in \mathcal{N}_1, p,l \in \mathcal{P}_1 \\
w_{i,j,p,l}^{(r,q)} & \geq & x_{i,p}^{(r,q)} + x_{j,l}^{(r,q)} - 1, & \forall\, i,j \in \mathcal{N}_1, p,l \in \mathcal{P}_1 \\
w_{i,j,p,l}^{(r,q)} & \geq & 0, & \forall\, i,j \in \mathcal{N}_1, p,l \in \mathcal{P}_1
\end{array}$$

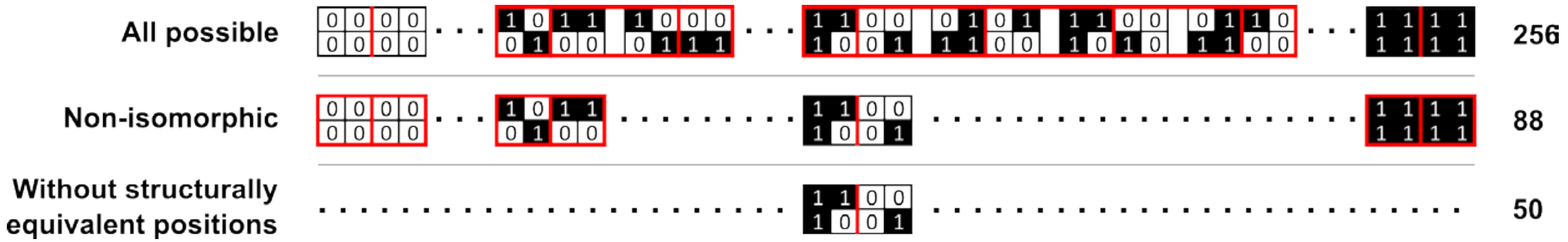
Linearization constraints for $w_{i,j,p,l}^{(r,q)} = x_{i,p}^{(r,q)} x_{j,l}^{(r,q)}$

$$\begin{array}{lcll}
z_{i,m,p,k}^{(r,q)} & \leq & x_{i,p}^{(r,q)}, & \forall\, i \in \mathcal{N}_1, m \in \mathcal{N}_2, p \in \mathcal{P}_1, k \in \mathcal{P}_2 \\
z_{i,m,p,k}^{(r,q)} & \leq & y_{m,k}^{(r,q)}, & \forall\, i \in \mathcal{N}_1, m \in \mathcal{N}_2, p \in \mathcal{P}_1, k \in \mathcal{P}_2 \\
z_{i,m,p,k}^{(r,q)} & \geq & x_{i,p}^{(r,q)} + y_{m,k}^{(r,q)} - 1, & \forall\, i \in \mathcal{N}_1, m \in \mathcal{N}_2, p \in \mathcal{P}_1, k \in \mathcal{P}_2 \\
z_{i,m,p,k}^{(r,q)} & \geq & 0, & \forall\, i \in \mathcal{N}_1, m \in \mathcal{N}_2, p \in \mathcal{P}_1, k \in \mathcal{P}_2.
\end{array}$$

Linearization constraints for $z_{i,m,p,k}^{(r,q)} = x_{i,p}^{(r,q)} y_{m,k}^{(r,q)}$

# Two crucial structural observations

- Observation #1: Some image matrices are created “equal”
- Observation #2: Some positions are created “equal”



| # of positions $(P_1, P_2)$ | All possible IMs | Non-isomorphic IMs | IMs without structurally equivalent positions | Percentage of all possible IMs to fit |
|---|---|---|---|---|
| **(1, 1)** | 4 | 4 | 4 | 100.00% |
| **(1, 2)** | 8 | 6 | 2 | 25.00% |
| **(2, 1)** | 64 | 36 | 32 | 50.00% |
| **(2, 2)** | 256 | 88 | 50 | 19.53% |
| **(2, 3)** | 1024 | 172 | 36 | 3.52% |
| **(3, 1)** | 4096 | 752 | 688 | 16.80% |
| **(3, 2)** | 32768 | 3272 | 2424 | 7.40% |
| **(3, 3)** | 262144 | 10704 | 4912 | 1.87% |
| **Total IMs to fit** | 300364 | 15034 | 8148 | 2.71% |

***Two order of magnitude reduction in the number of IMs to fit***