SYSTEMS ENGINEERING
Research Center

CISA
Center for Integrated Systems in Aerospace

Purdue System of Systems Analytic Workbench [RT-155]

Sponsor: DASD(SE)

By
Dr. Karen Marais

8th Annual SERC Sponsor Research Review
November 17, 2016
20 F Street NW Conference Center
20 F Street, NW
Washington, DC

www.sercuarc.org

Motivation for Research

- SoS Architectures are highly complex, with many interdependencies across diverse constituent systems
- Difficult to know how and when to add/remove/integrate systems or connections
  - Too big for one analyst
  - Too many contingencies and choices for simple tools
  - Too many stakeholders for top-down management

Pain Points	Question
SoS Authority	*What are effective collaboration patterns in systems of systems?*
Leadership	*What are the roles and characteristics of effective SoS leadership?*
Constituent Systems	*What are effective approaches to integrating constituent systems into a SoS?*
Autonomy, Interdependencies & Emergence	*How can SE provide methods and tools for addressing the complexities of SoS interdependencies and emergent behaviors?*
Capabilities & Requirements	*How can SE address SoS capabilities and requirements?*
Testing, Validation & Learning	*How can SE approach the challenges of SoS testing, including incremental validation and continuous learning in SoS?*
SoS Principles	*What are the key SoS thinking principles, skills and supporting examples?*

Survey identified seven 'pain points' raising a set of SoS SE questions

From: "Systems of Systems Pain Points", Dr. Judith Dahmann, INCOSE Webinar Series on Systems of Systems, 22-FEB, 2013

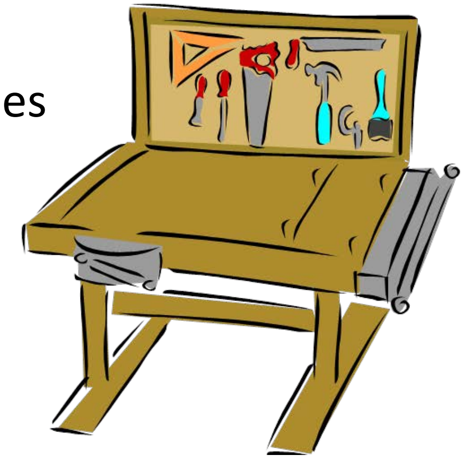
Can an organized set of Methods, Processes and Tools (MPTs), presented in a user-friendly way, solve these problems?

SERC RT-108/134/155 Projects have been pursuing this question

Vision: A *Useful* SoS Analytic Workbench

- Rational
  - Relegate complexities to methods
  - Delegate decision-making to users
- Open
  - Accommodates insertion of new SoS analytic methods (from Purdue or others)

- Interoperable
  - Outcomes produced in form suitable for additional SoSE phases
  - 'Domain agnostic', cross platform operations
  - Address uncertainty in data/simulation outcomes

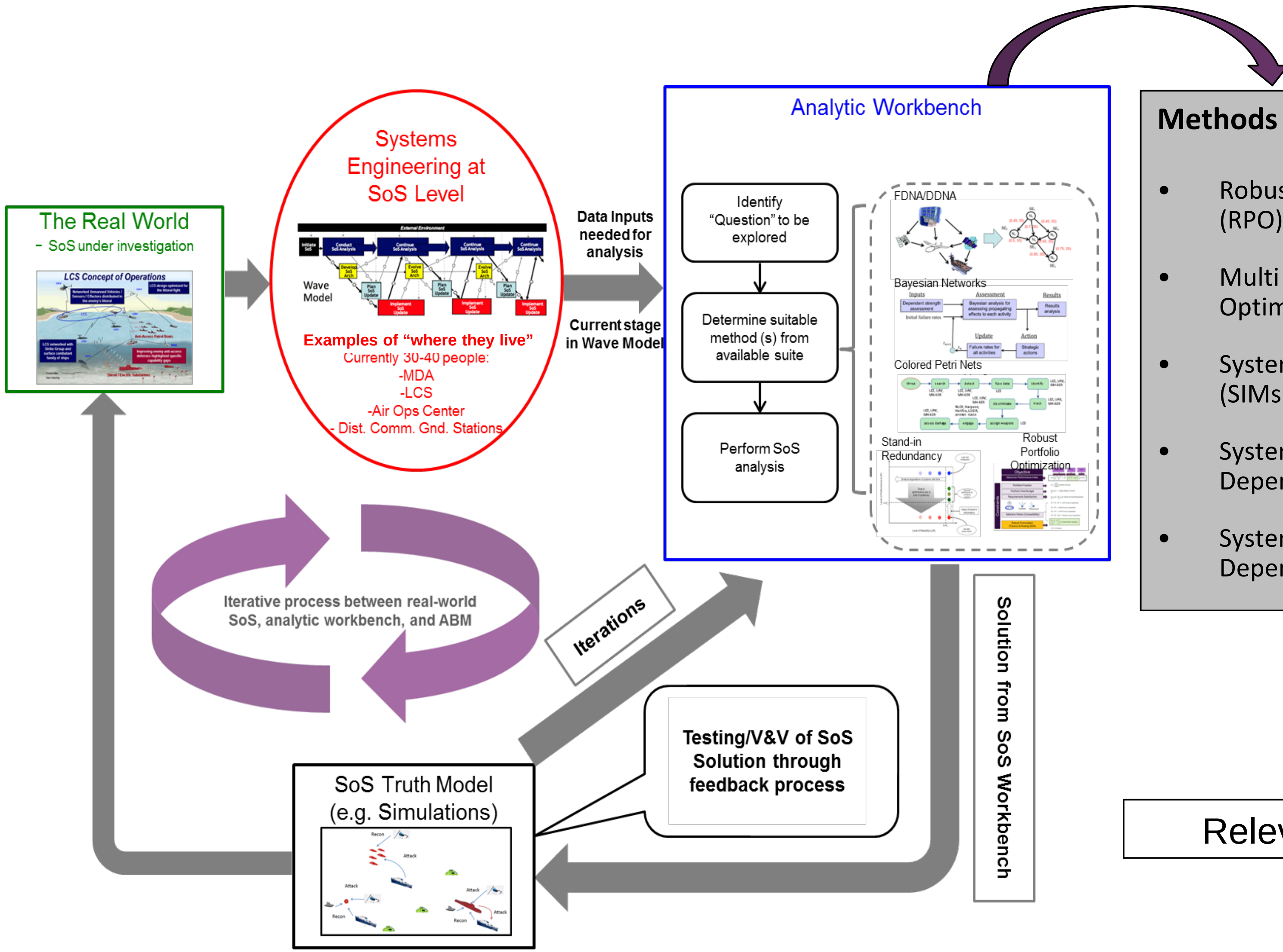


- Useable
  - (Scalability) ➔ reasonable scaling of computational need to problem sizes
  - (Ease of Use) ➔ Users can translate problem to inputs required by relevant methods and tools

SYSTEMS ENGINEERING
Research Center

Concept: SoS Analytic Workbench

CISA
Center for Integrated Systems in Aerospace

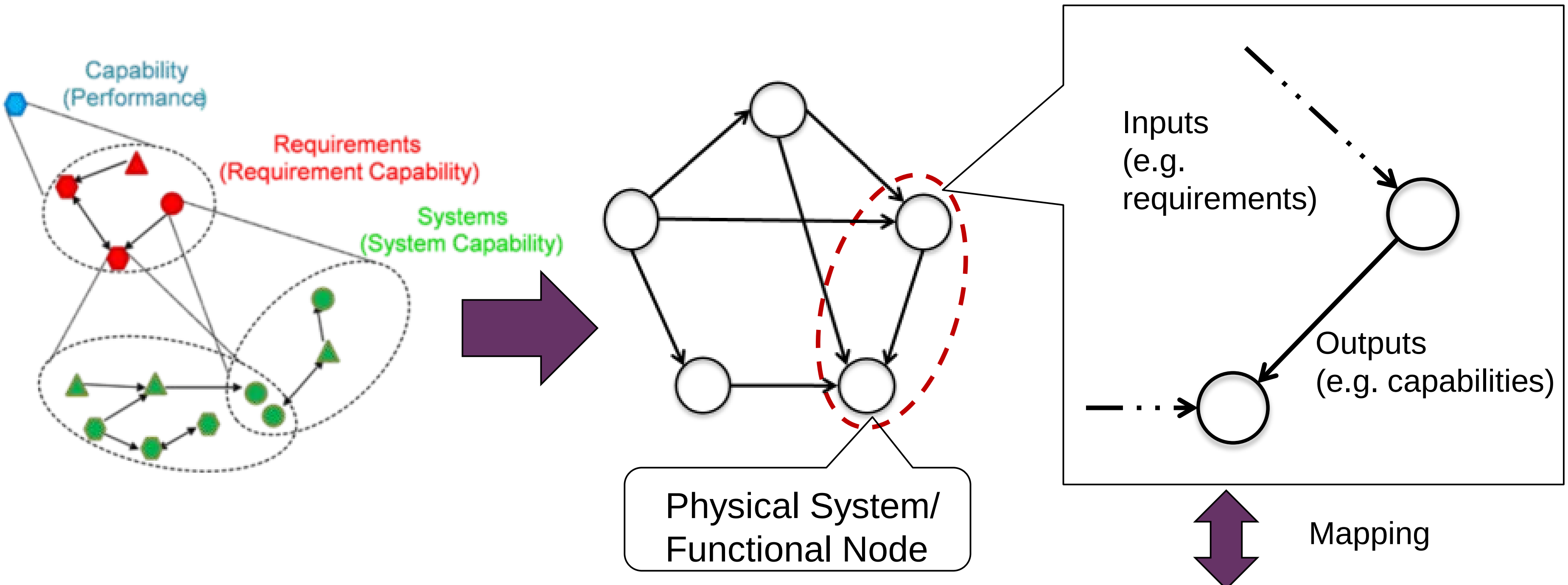


Methods in Toolset:

- Robust Portfolio Optimization (RPO)
- Multi STakeholder Dynamic Optimization (MuSTDO)
- System Importance Measures (SIMs)
- Systems Operational Dependency Analysis (SODA)
- Systems Developmental Dependency Analysis (SDDA)

Relevant Input Data

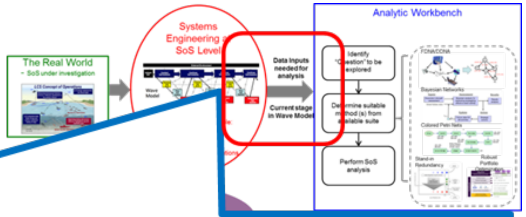
Graph-basis Data Model / Representation



- Translate SoS problem into network topology with hierarchy (nodes, links, inputs, outputs)
- Map data and description to equivalent network representation

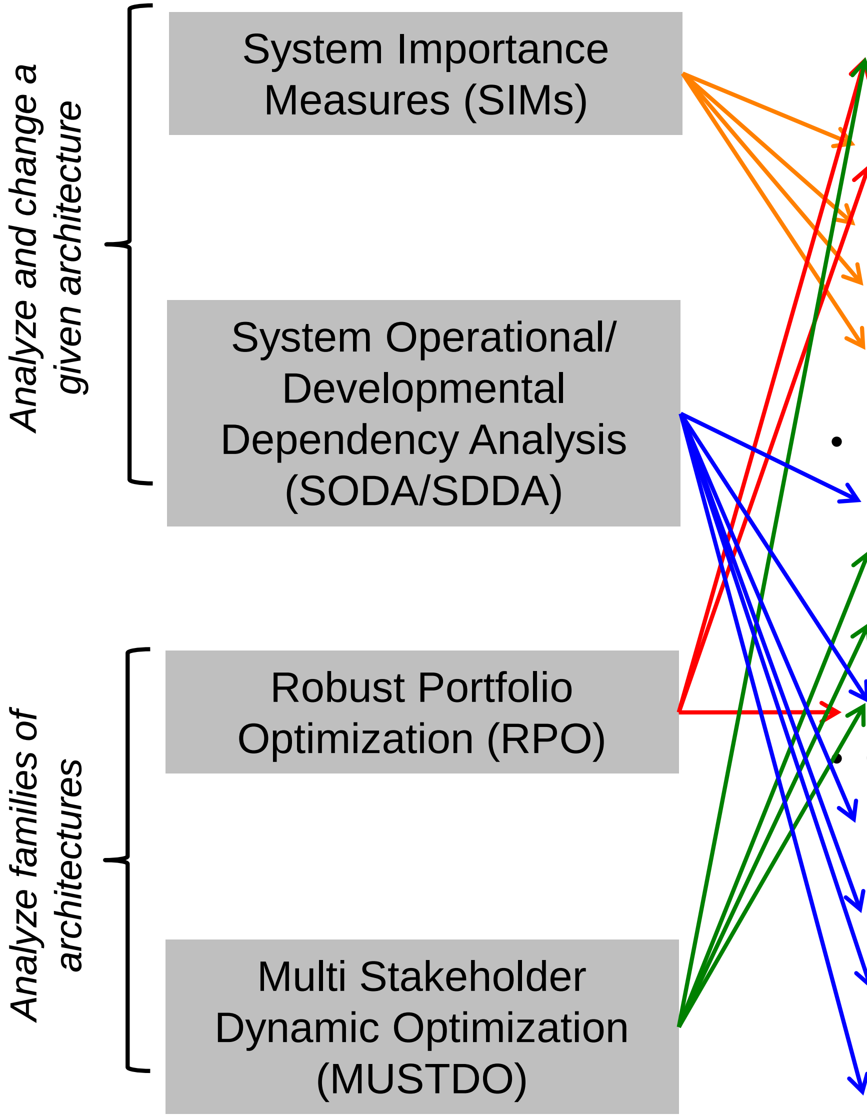
OV – Operation Flow
SV – Service Flow
PV – Project Flow
…
Simulation/Actual data

Archetypal Questions in SoS development and operation



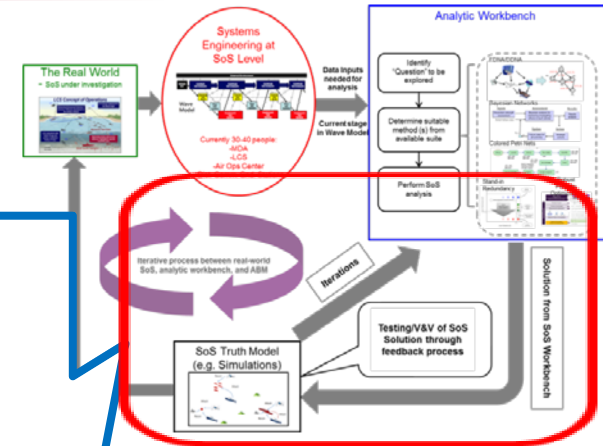
- **SoS Capability / Resilience**
  1. What combination of systems gives the desired aggregate SoS capabilities?
  2. What changes to which systems offer the most (performance, resilience, etc.) leverage?
  3. Which systems are critical to SoS performance? SoS risks?
  4. Which parts of the SoS have excess or inadequate resilience?
  5. Which design principles can improve SoS robustness and resilience?
- **Development**
  6. How do/should partial capabilities evolve over time?
  7. How do we optimize multi-stage acquisitions in SoS development?
  8. How do we coordinate planning between local and SoS-level stakeholders?
  9. How do (desired and undesired) changes in system properties affect SoS development?
- **Critical System Characterization**
  10. What is the impact of partial/total system failures during operations?
  11. What is the impact of partial/total failure of a system during development?
  12. What are the most critical systems in a given operational (or developmental) network?
  13. What is the impact of development delays in an interdependent network?

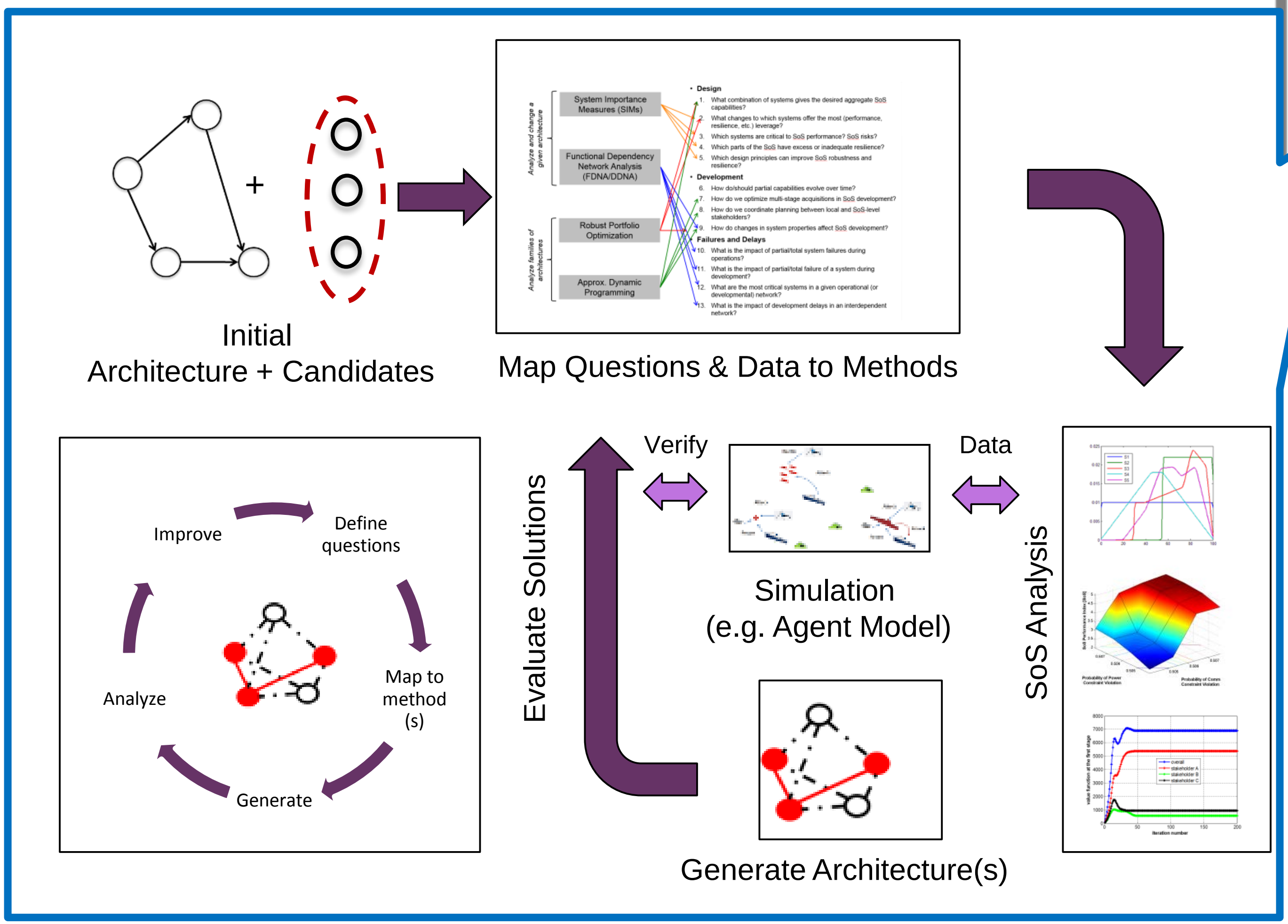
Addressing Archetypal Questions



- **Capability**
  1. What combination of systems gives the desired aggregate SoS capabilities?
  2. What changes to which systems offer the most (performance, resilience, etc.) leverage?
  3. Which systems are critical to SoS performance? SoS risks?
  4. Which parts of the SoS have excess or inadequate resilience?
  5. Which design principles can improve SoS robustness and resilience?
- **Development**
  6. How do/should partial capabilities evolve over time?
  7. How do we optimize multi-stage acquisitions in SoS development?
  8. How do we coordinate planning between local and SoS-level stakeholders?
  9. How do changes in system properties affect SoS development?
- **Critical System ID**
  10. What is the impact of partial/total system failures during operations?
  11. What is the impact of partial/total failure of a system during development?
  12. What are the most critical systems in a given operational (or developmental) network?
  13. What is the impact of development delays in an interdependent network?

Analysis and Verification

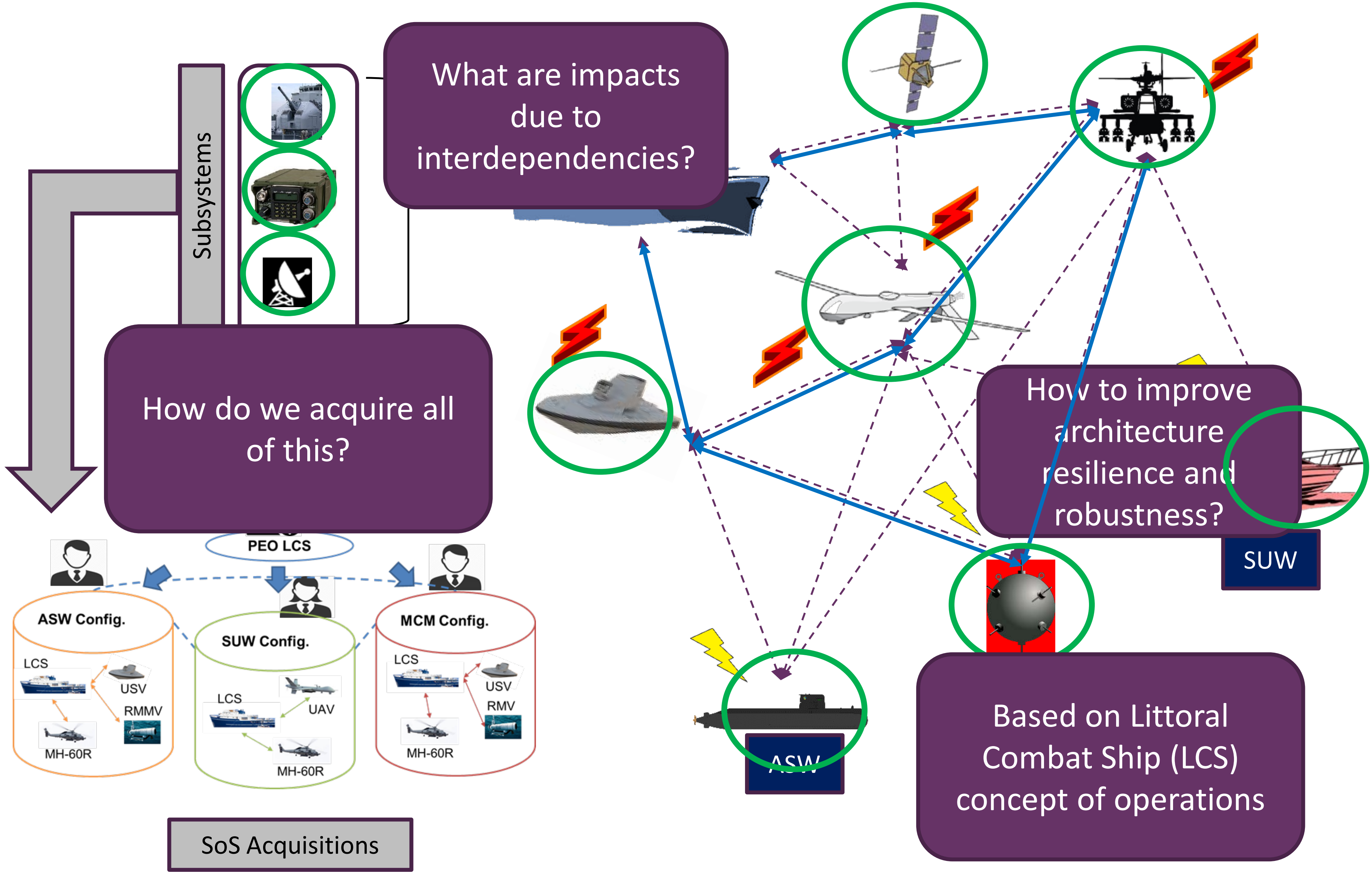




Iterative process to improve architecture

Use of simulation as a "truth model" and/or as data generator

Concept Naval Warfare Scenario



Systems Operational Dependency Analysis (SODA)

Capturing the impact of complex technical dependencies

- Which systems are critical to SoS performance? SoS risks?
- Which design principles can improve SoS robustness and resilience?
- What is the impact of partial/total system failures during operations?
- What are the most critical systems in a given operational network?

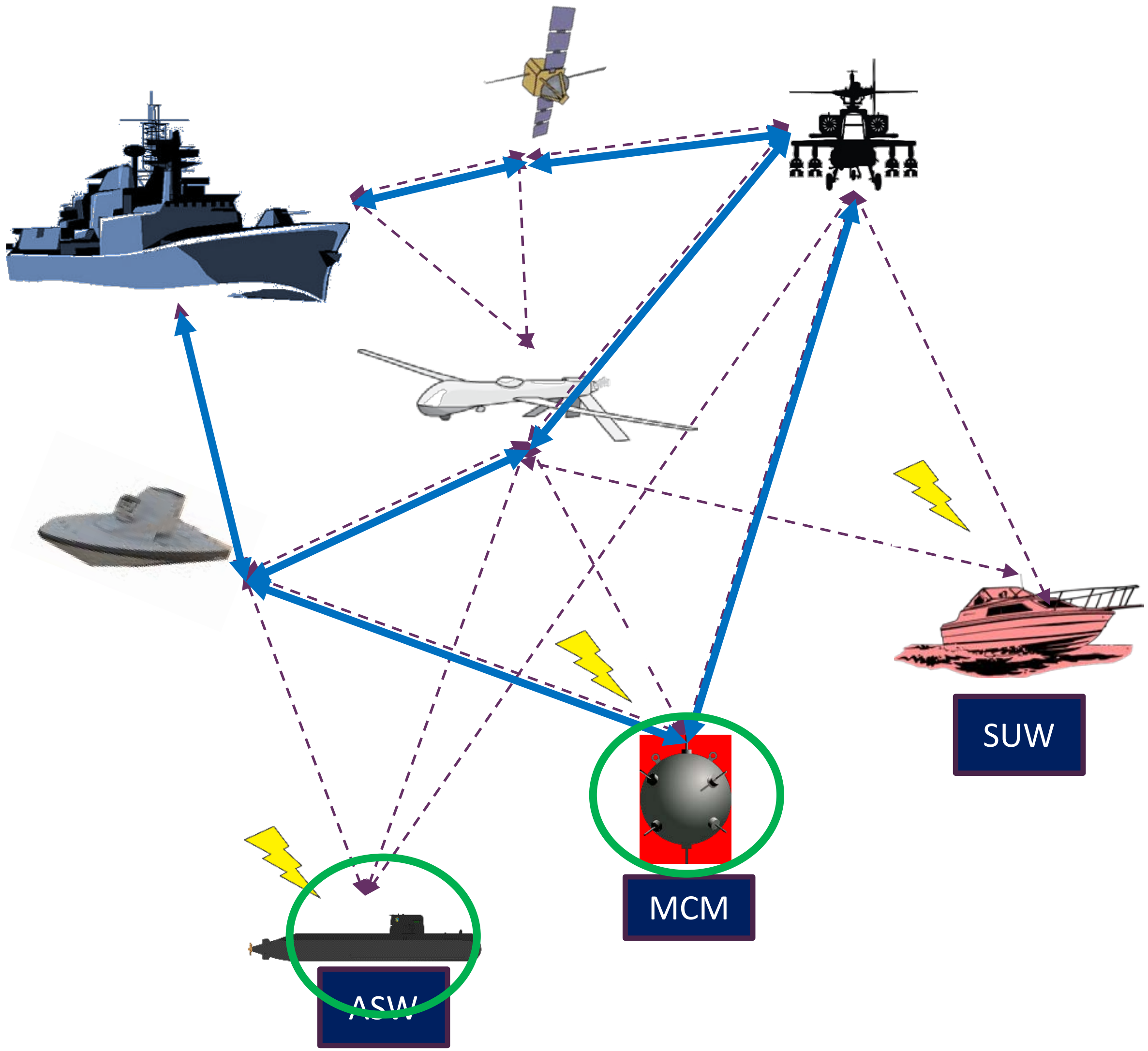
Systems Developmental Dependency Analysis (SDDA)

Capturing the impact of developmental schedule dependencies and stakeholder decisions

- How do/should partial capabilities evolve over time?
- What is the impact of partial/total failure of a system during development?
- What are the most critical systems in a given operational (or developmental) network?
- What is the impact of development delays in an interdependent network?

SODA/SDDA

What are impacts due to interdependencies?

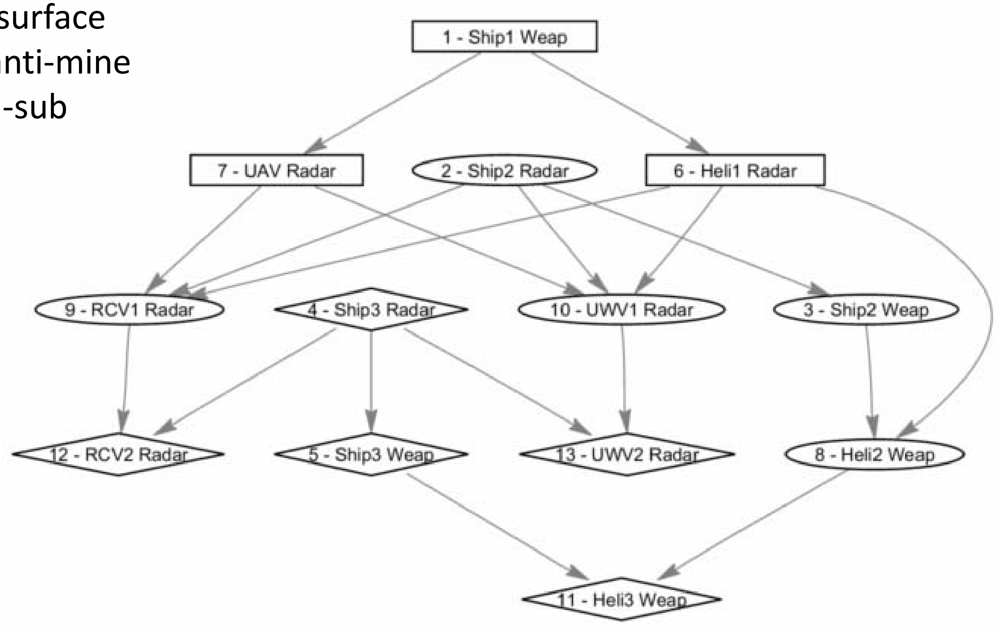


The Naval Warfare Scenario for SDDA

Three architectures with different developmental dependencies

- **Architecture A: ships first, then surface systems, followed by anti-submarine and anti-mine**

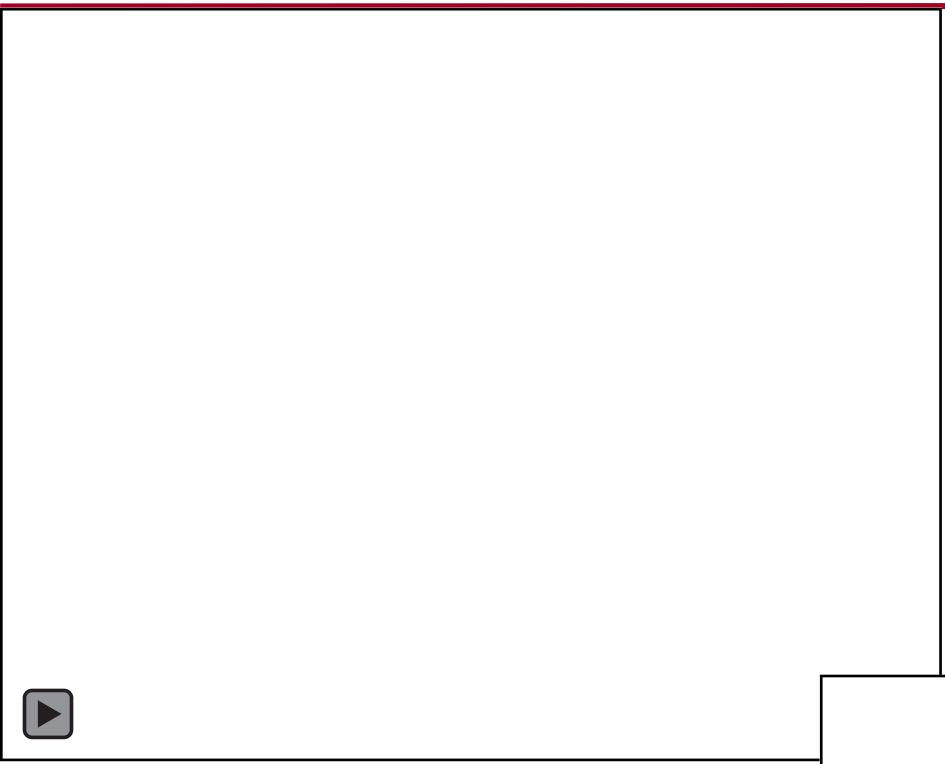
Rectangles: surface
Diamonds: anti-mine
Ellipses: anti-sub



Architecture A

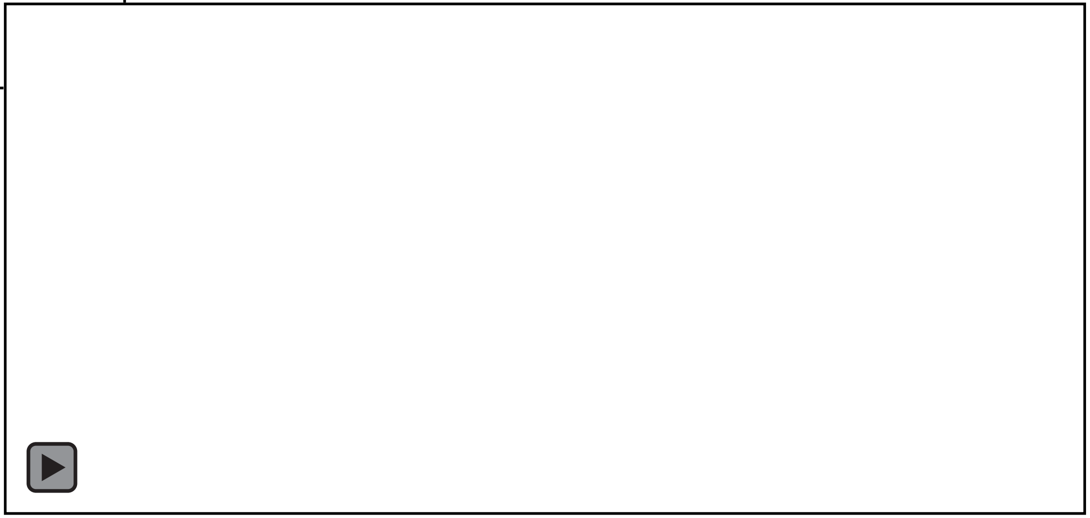
- Architecture B: surface and anti-mine systems first, followed by anti-submarine
- Architecture C: surface systems independently. Completion of anti-mine depends on anti-submarine sys

SDDA results
1. Development schedule

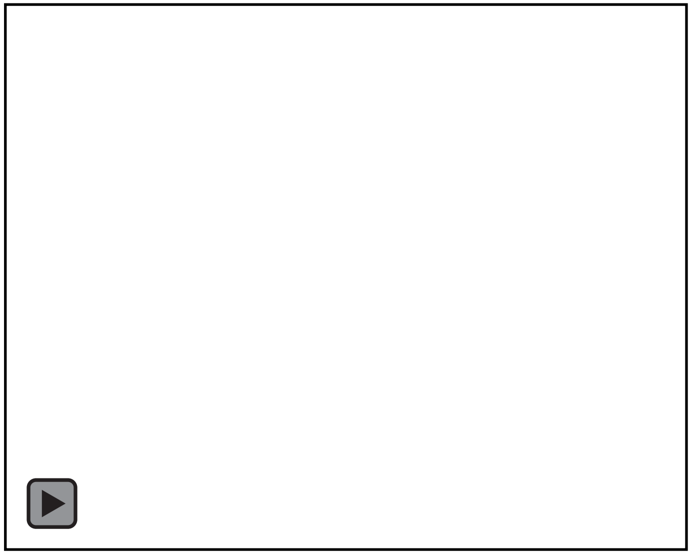


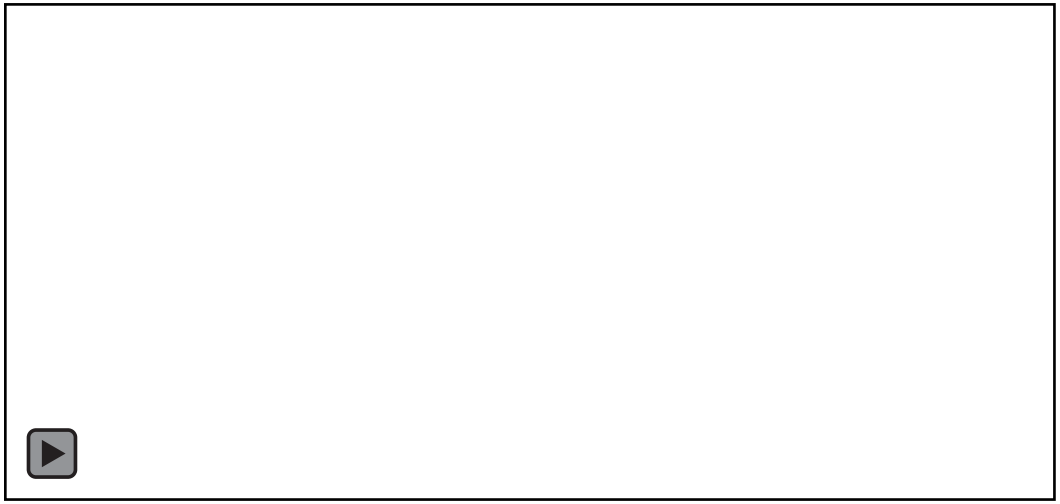
Architecture A

- **Uncertainties vary, but generally decrease over time**
- **Each system is deployed on the appropriate field when its development is completed**

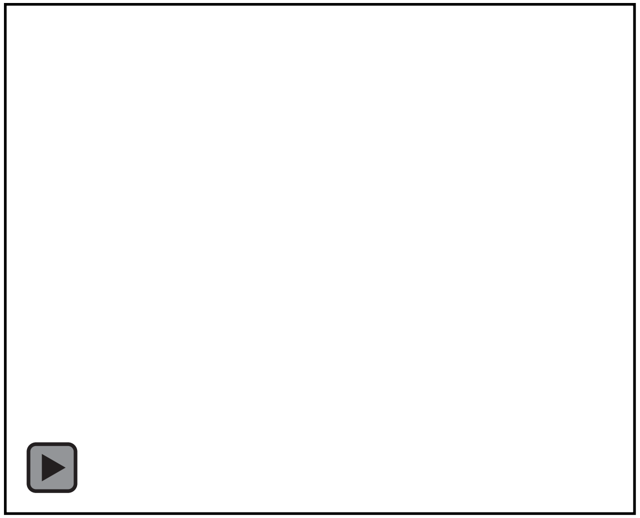


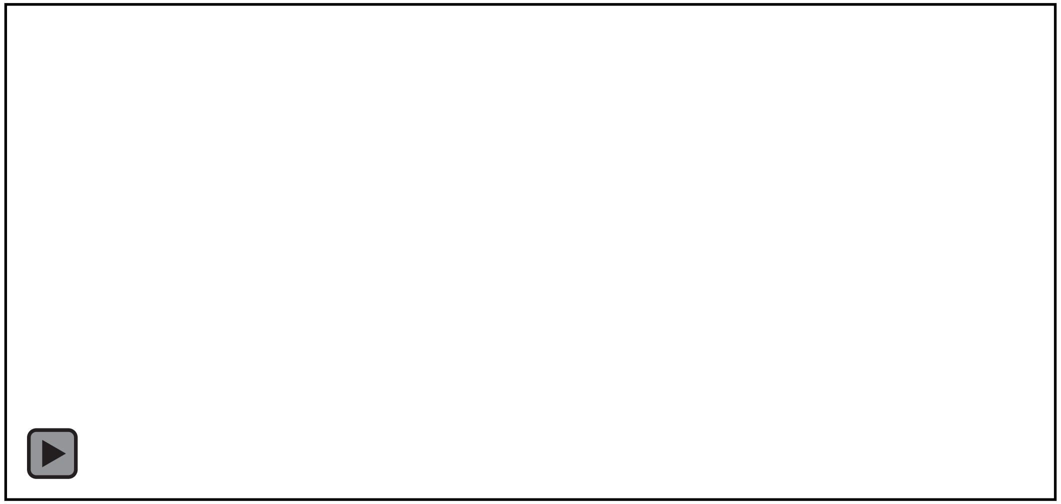
SDDA results
2. Comparison of different architectures





Architecture B





Architecture C

SDDA results
3. Criticality in development schedule

- Initial delay is the median expected delay for each system (same in all architectures)
- Final delay is the delay on the overall completion time of the entire architecture, due to initial delay in one system
- Delays can be partially/fully absorbed (green/transparent), reflect entirely on the final delay (orange), or cause final delays higher than the initial delay (red)

Combined application of SODA and SDDA
1. Partial capabilities over time

- Architecture A fast in achieving partial capabilities in anti-sub systems
- Architecture A fastest to achieve full capabilities in anti-sub systems

- Architecture B fastest to achieve most full capabilities
- Architecture B slower to achieve partial capability in anti-sub systems

- Architecture C fast in achieving partial capabilities in anti-sub systems

- «Jumps» in capabilities when new systems are deployed
- Earlier capabilities must be traded-off with reduced flexibility
- Different reaction to delays. Some systems not affected by delays (e.g. anti-mine in B)

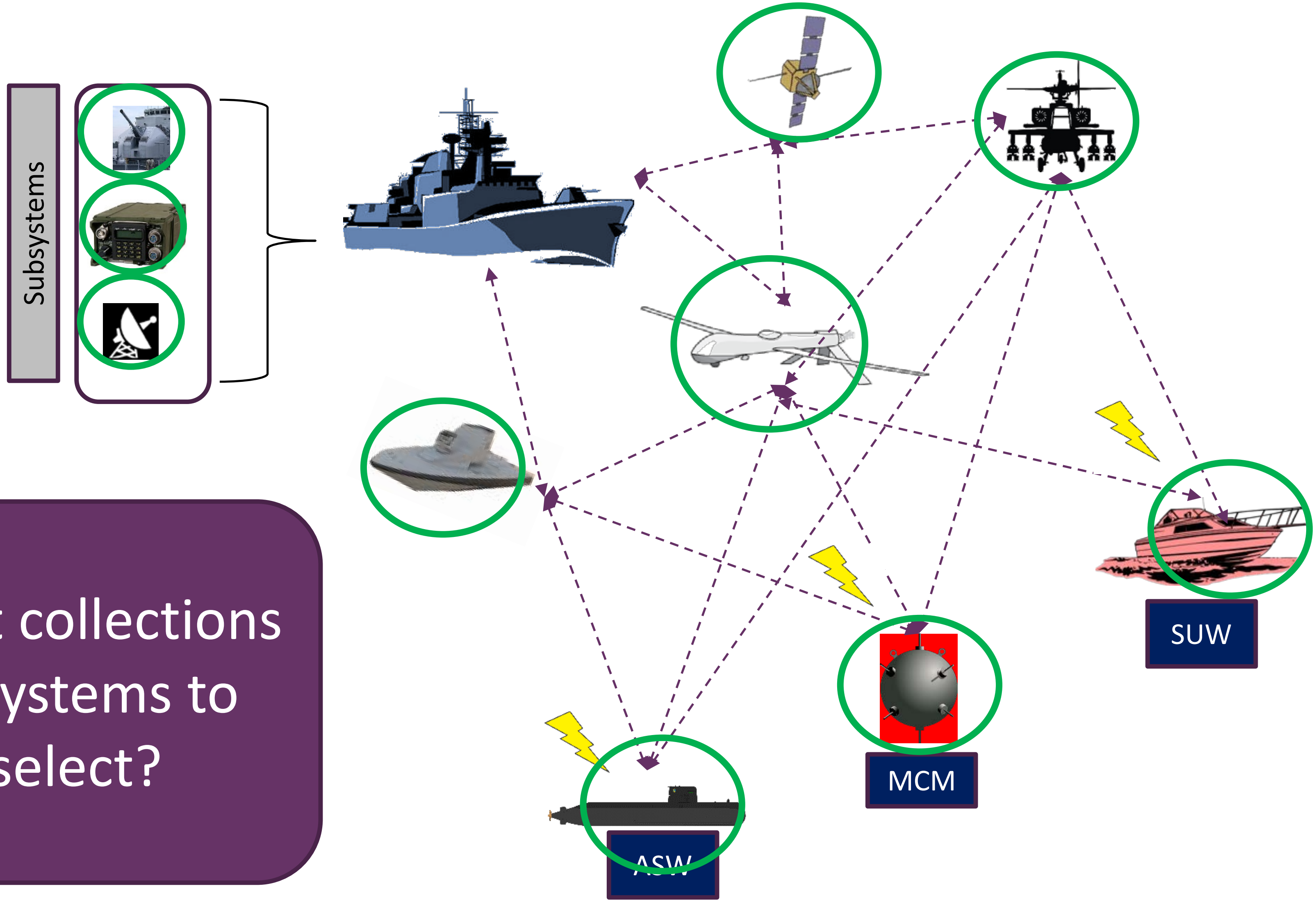
Further results with SODA and SDDA

- Criticality of systems in the operational domain
- Robustness and resilience to failures. Comparison of alternate architectures
- Impact of managerial decisions in development stage (trade-off between risk, time and cost).
  - In architecture A, reviewing schedule is often useful in case of wrong initial decision

Robust Portfolio Optimization (RPO)

Finding good collections of systems to develop

Robust Portfolio Optimization

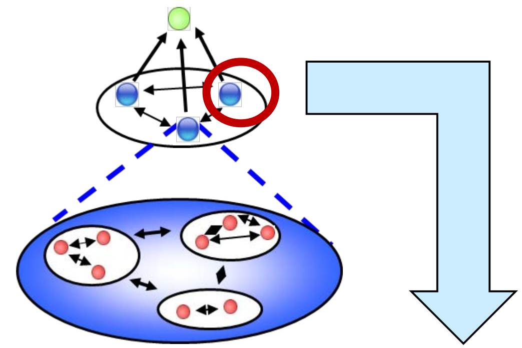


What collections of systems to select?

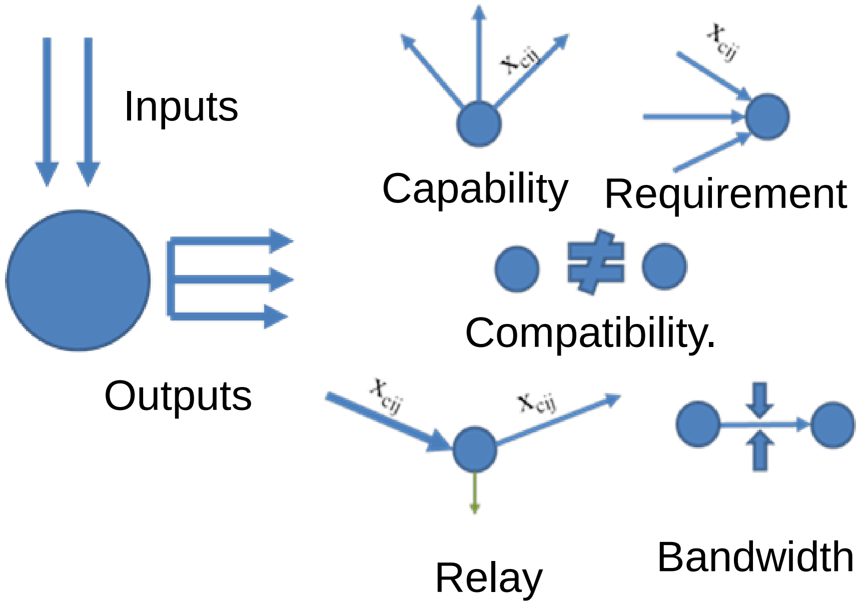
Robust Portfolio Optimization

- Treat SoS as portfolio of systems

- Model individual systems as nodes
    - Functional & Physical representation



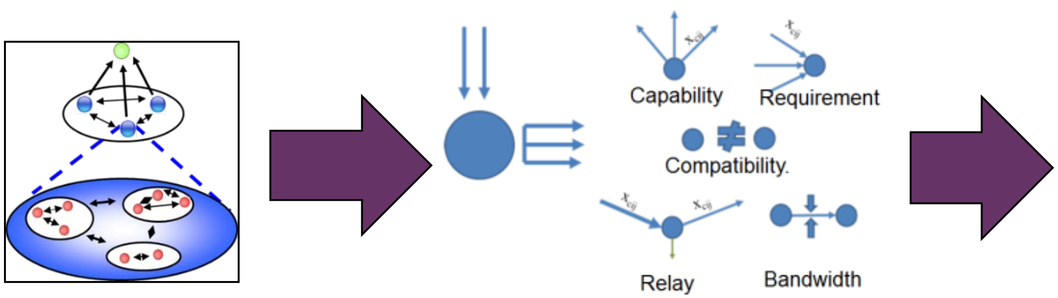
- Rules for node connectivity
    - Compatibility between nodes
    - Bandwidth of linkages
    - Supply (Capability)
    - Demand (Requirements)
    - Relay capability

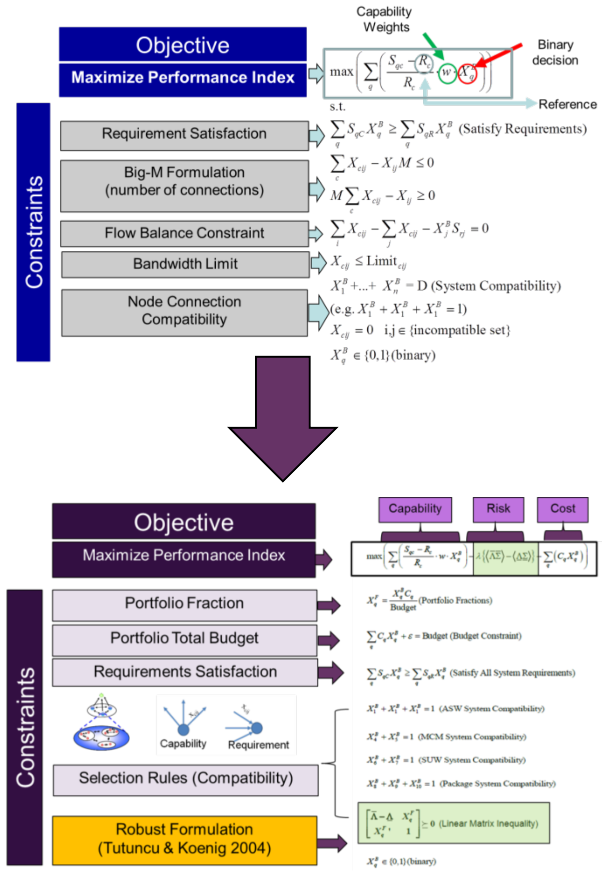


- Represent as mathematical programming problem

Robust Portfolio Optimization

Decision support approach from financial engineering/operations research to identify portfolios of systems by leveraging performance against risk under uncertainties





- Represent behaviors as connectivity constraints
- Employ robust optimization techniques to deal with data uncertainty
- Computationally efficient tools to solve even for very large number of nodes

Robustification to include data uncertainties

SYSTEMS ENGINEERING Research Center

NWS Communications Layer Analysis

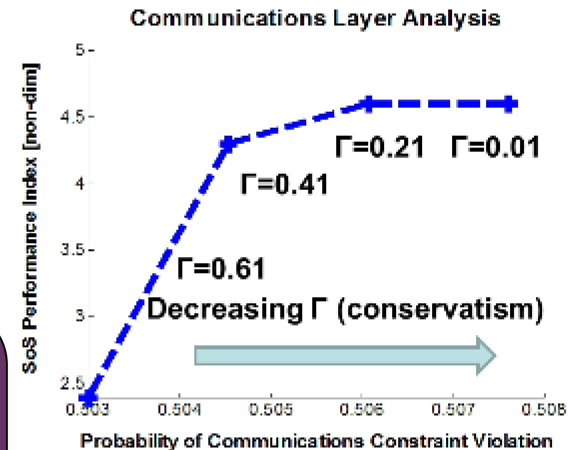
CISA Center for Integrated Systems in Aerospace

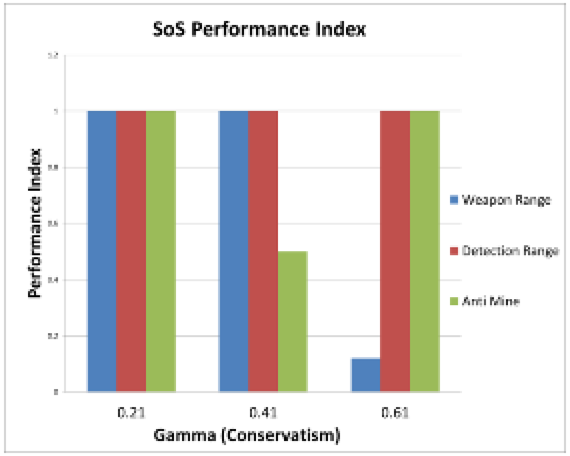
- Build in robustness for communications layer subject to uncertainties in performance
- Robustness of ‘requirements for communications capability being met’

Systems	Available System Packages	Gamma (Level of Conservatism) 0.01	0.21	0.41	0.61
ASW	Variable Depth	-	-	-	-
	Multi Fcn Tow	x	x	x	x
	Lightweight Tow	-	-	-	-
MCN	RAMCS II	-	-	-	-
	ALMDS (MH-60)	x	x	-	-
SUW	N-LOS Missiles	x	x	x	-
	Griffin Missiles	-	-	-	x
Seaframe	Package 1	-	-	-	-
	Package 2	-	-	-	-
	Package 3	x	x	x	x
Comm.	System 1	-	-	-	-
	System 2	x	x	x	x
	System 3	-	-	-	-
	System 4	-	-	-	x
	System 5	-	-	-	-
	System 6	x	x	x	-

Portfolios of systems at prescribed conservatism

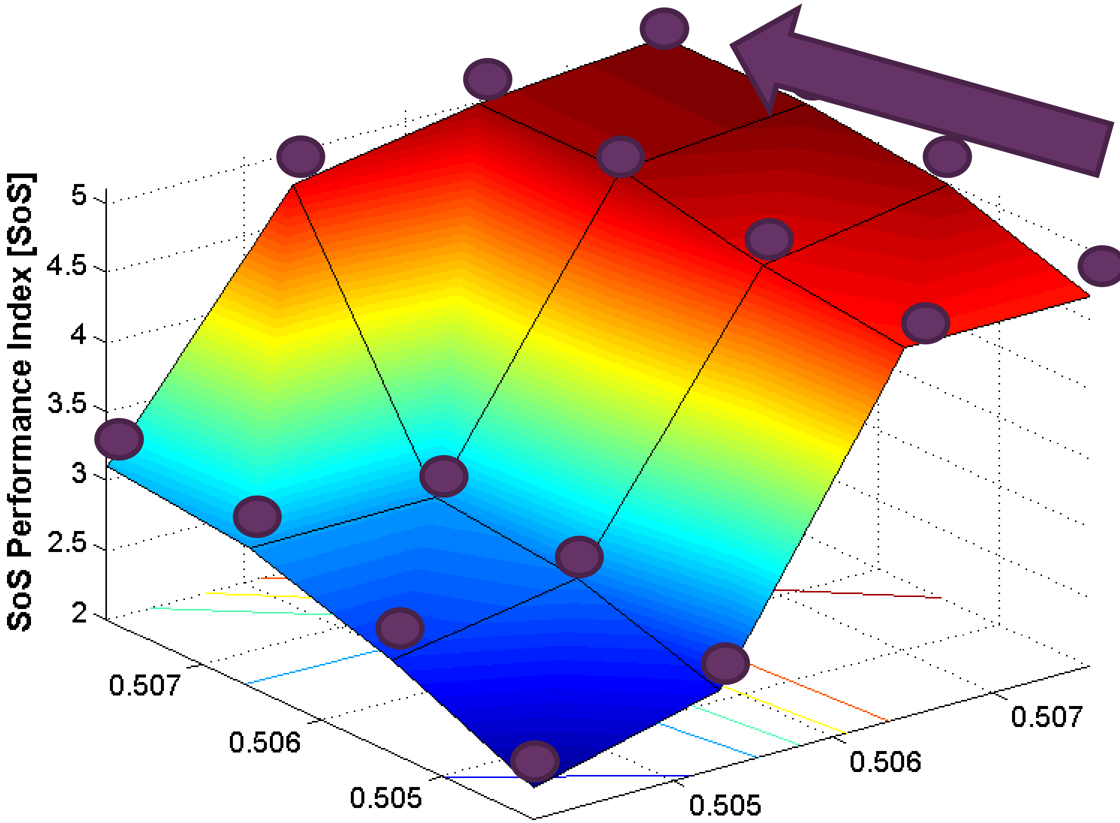
Trade SoS Performance for Communication Conservatism (e.g. against cyber-attack)





Multi-Metrics: Power & Comm. Layer Analysis

- Build in robustness for communications and power layer simultaneously
- Robustness to constraint violation of ‘requirements for communications and power generation capability being met’ → Tradespace analysis



Each point is a collection of systems

Probabilistic guarantees on constraint violation for multiple dimensions

Trade Communication Conservatism Against other metrics (e.g. Power Layer)

SYSTEMS ENGINEERING
Research Center

CISA
Center for Integrated Systems in Aerospace

Systems Importance Measures (SIMs)

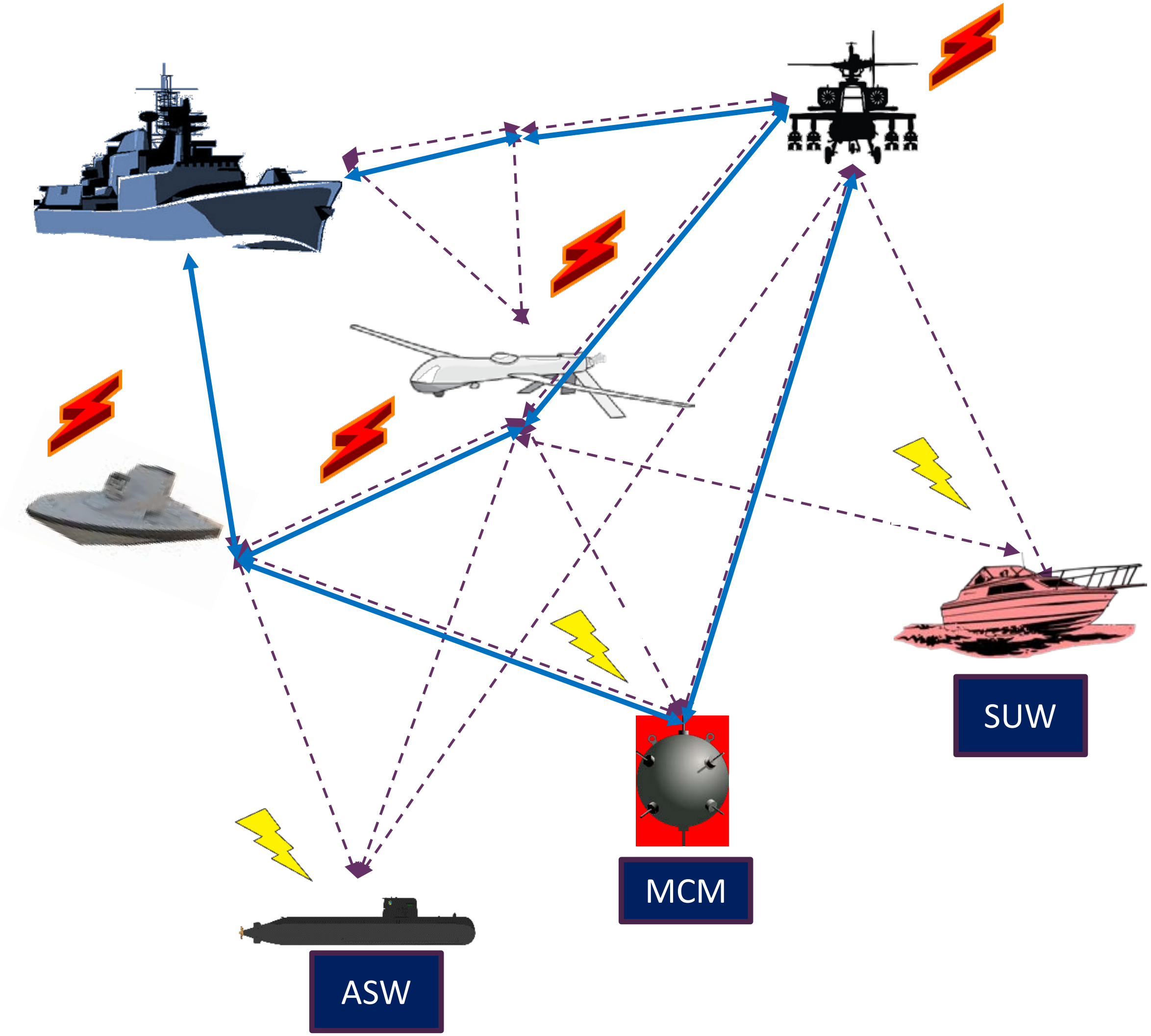
How to strategically build resilience into an architecture

SYSTEMS ENGINEERING Research Center

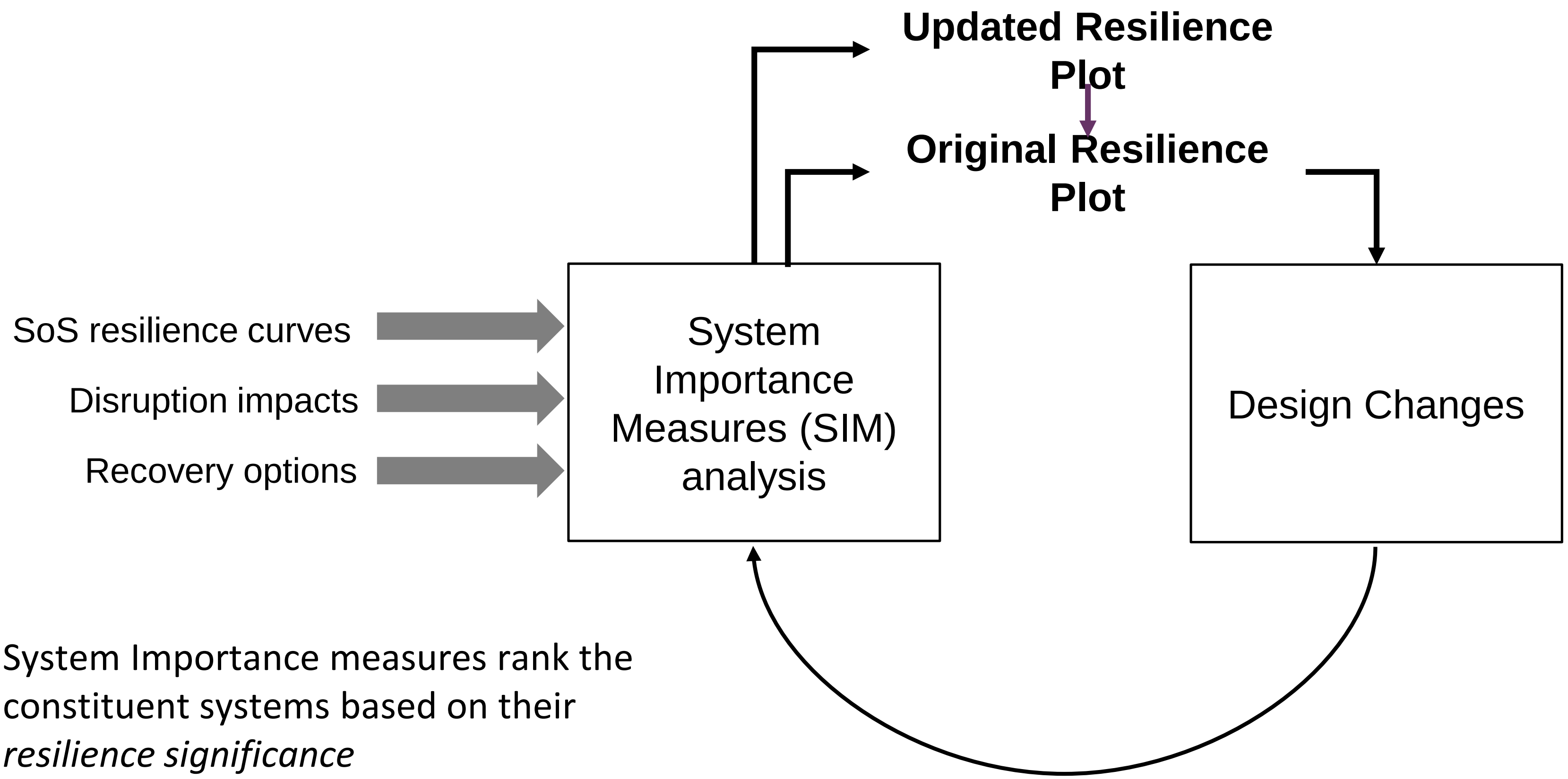
SIM within NWS

CISA Center for Integrated Systems in Aerospace

How to improve architecture resilience and robustness?

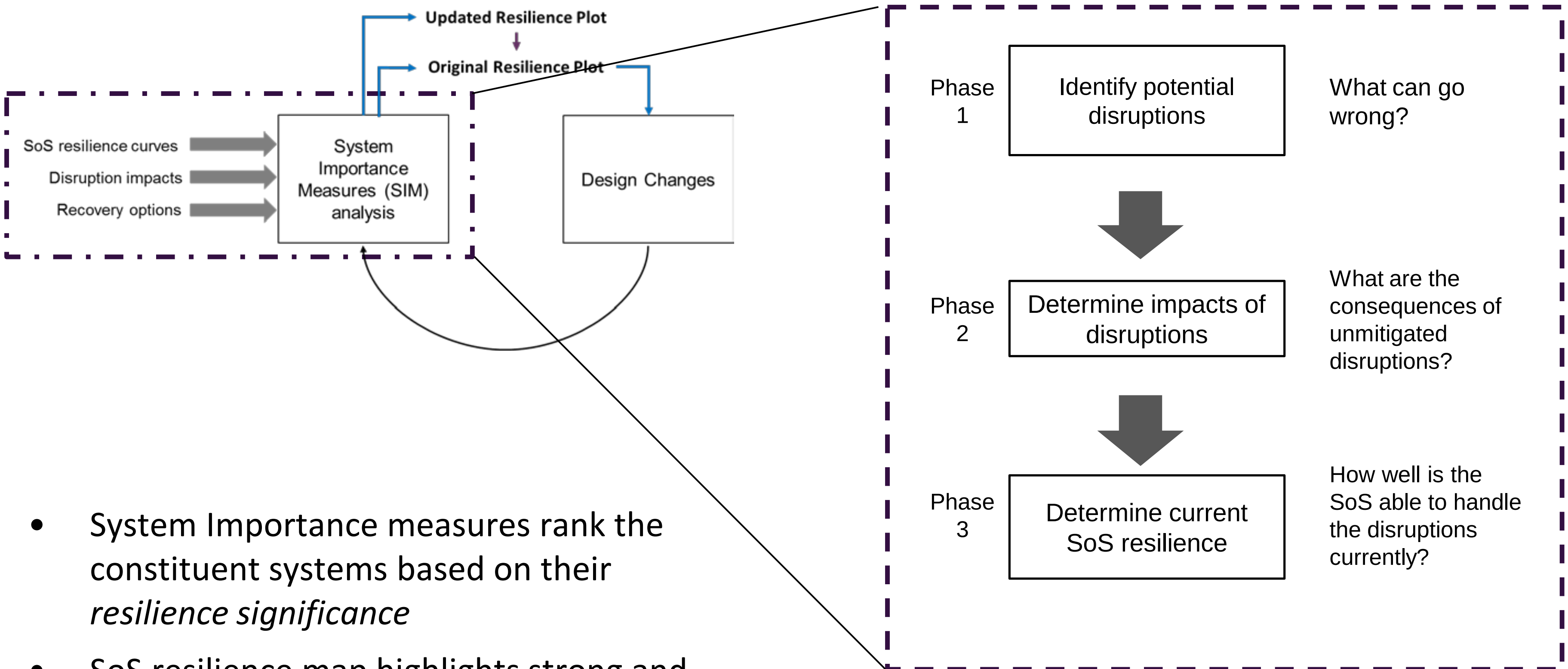


SIM Resilience Design: 4 Phases of SIM Analysis



- System Importance measures rank the constituent systems based on their *resilience significance*
- SoS resilience plot highlights strong and weak points
- Iteratively use design principles to update SoS until desired resilience is achieved

SIM Resilience Design: 4 Phases of SIM Analysis



- System Importance measures rank the constituent systems based on their *resilience significance*
- SoS resilience map highlights strong and weak points
- Iteratively use design principles to update SoS until desired resilience is achieved

Three System Importance Measures

System Disruption Importance (SDI):

- What is the impact of an unmitigated disruption on the SoS?

System Disruption Conditional Importance (SDCI):

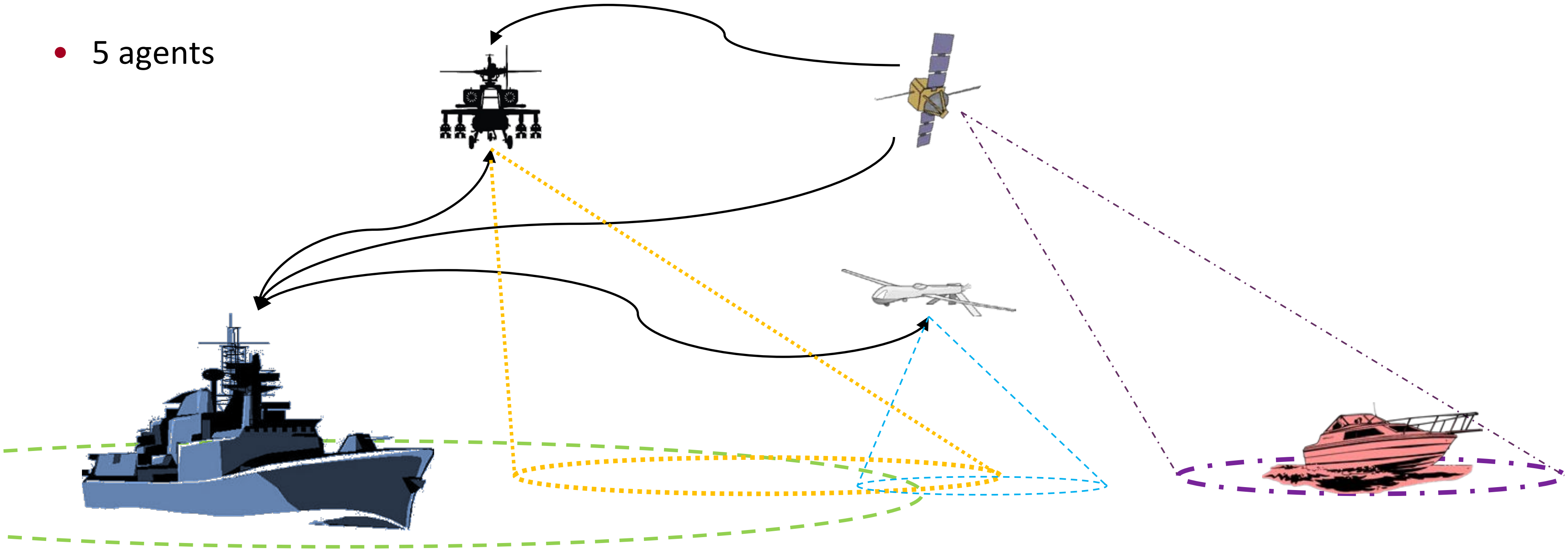
- How important is a disruption given that its impact is mitigated?

System Disruption Mitigation Importance (SDMI):

- How effective is a mitigation measure?

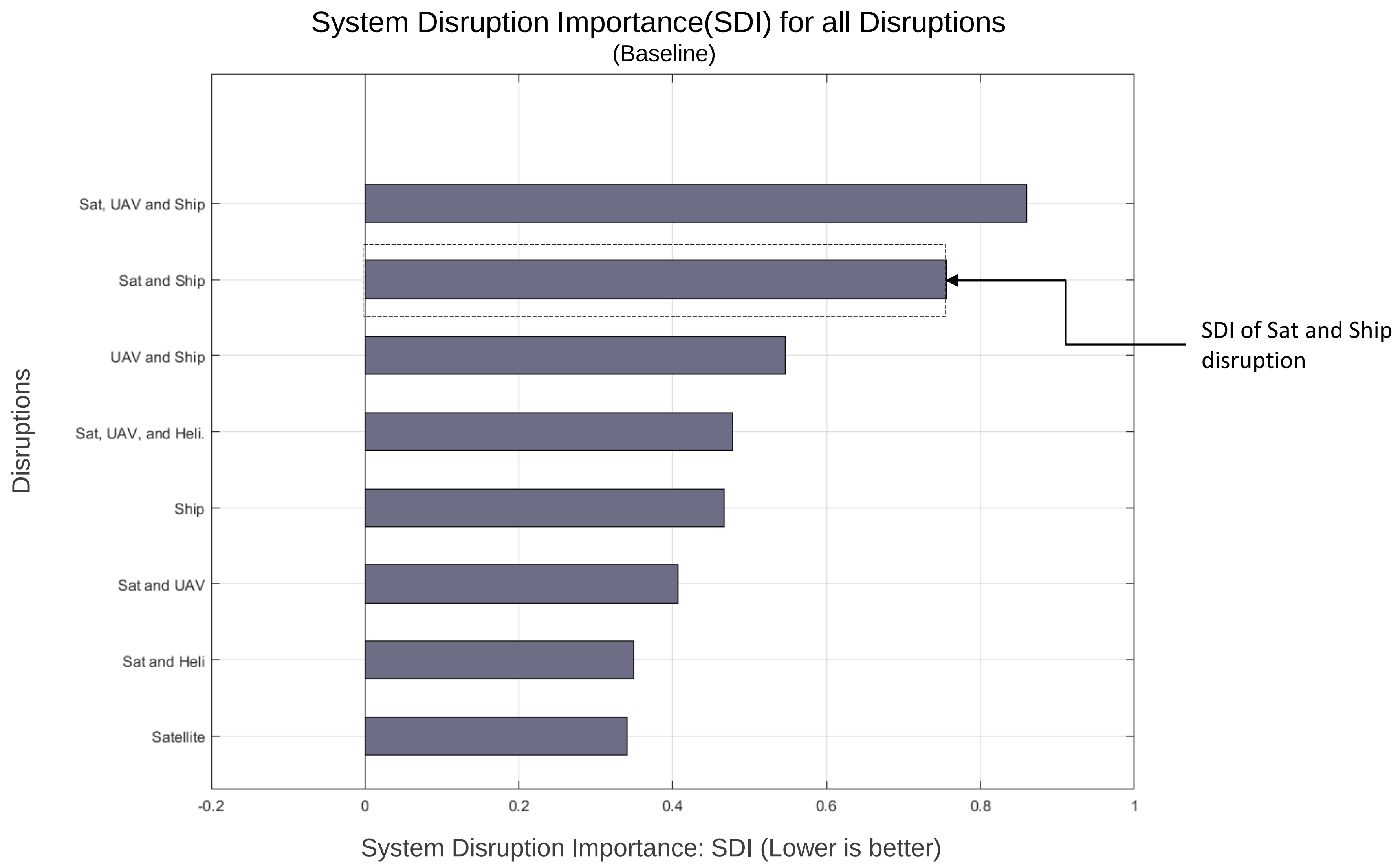
Our Naval Warfare Scenario: ABM simulation

- 5 agents



- Goal of the system is to destroy the enemy ship within the mission time
  - System Performance is defined as the percentage of successful missions
- Consider resilience of the system for a set of disruptions and mitigations

Without mitigations, the system is not resilient to disruptions

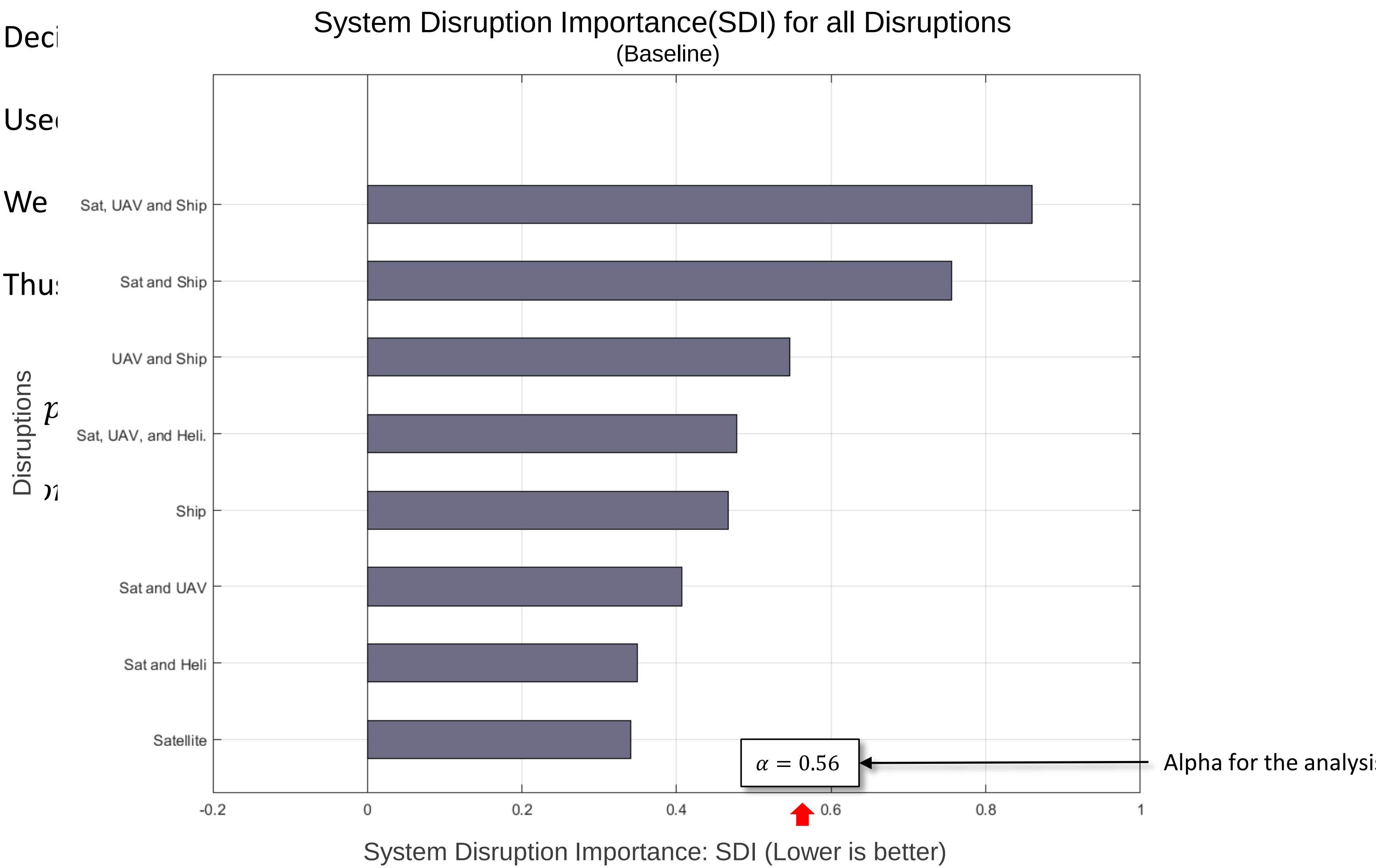


SYSTEMS ENGINEERING Research Center

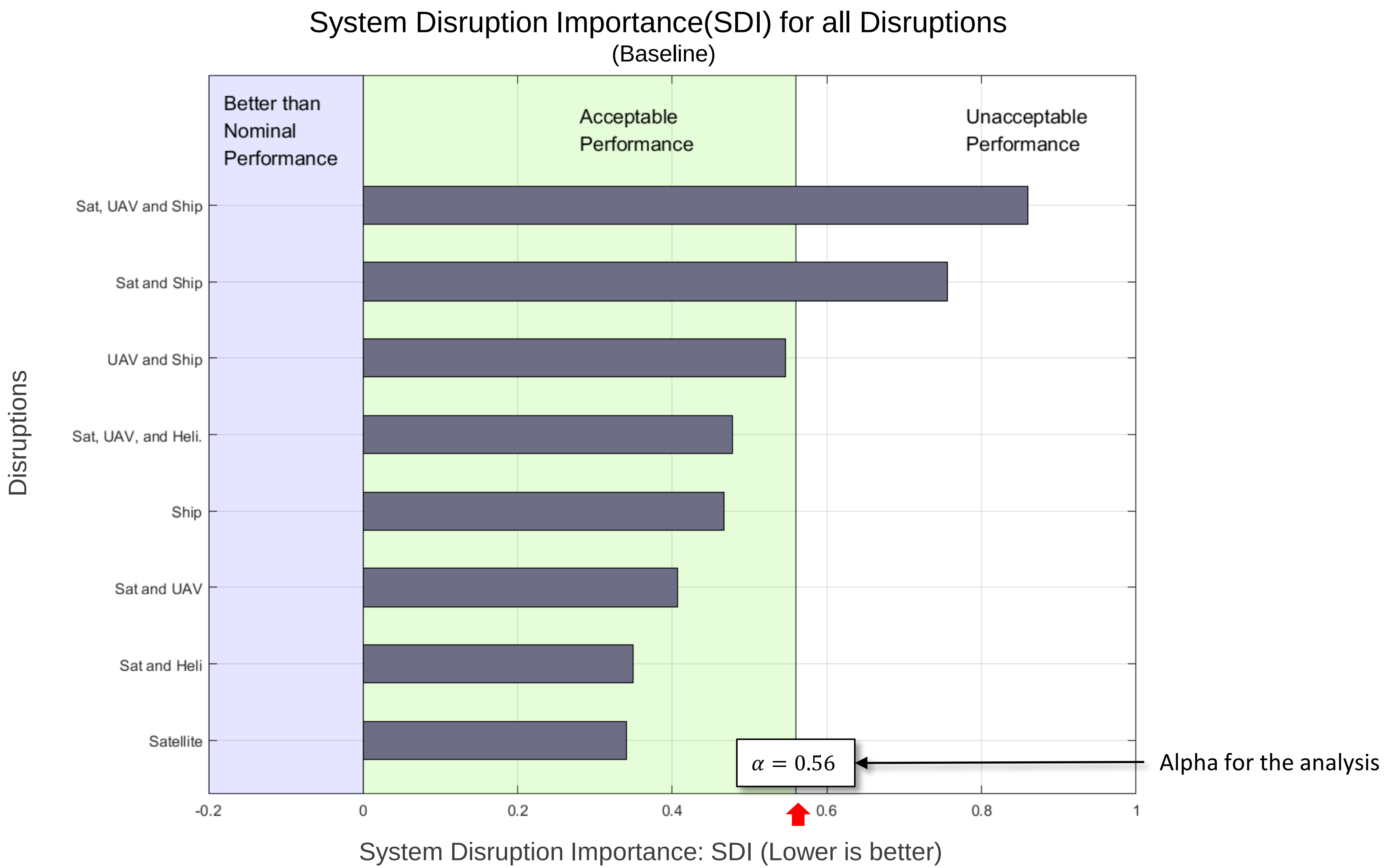
Use decision threshold (α) to identify critical disruptions

CISA Center for Integrated Systems in Aerospace

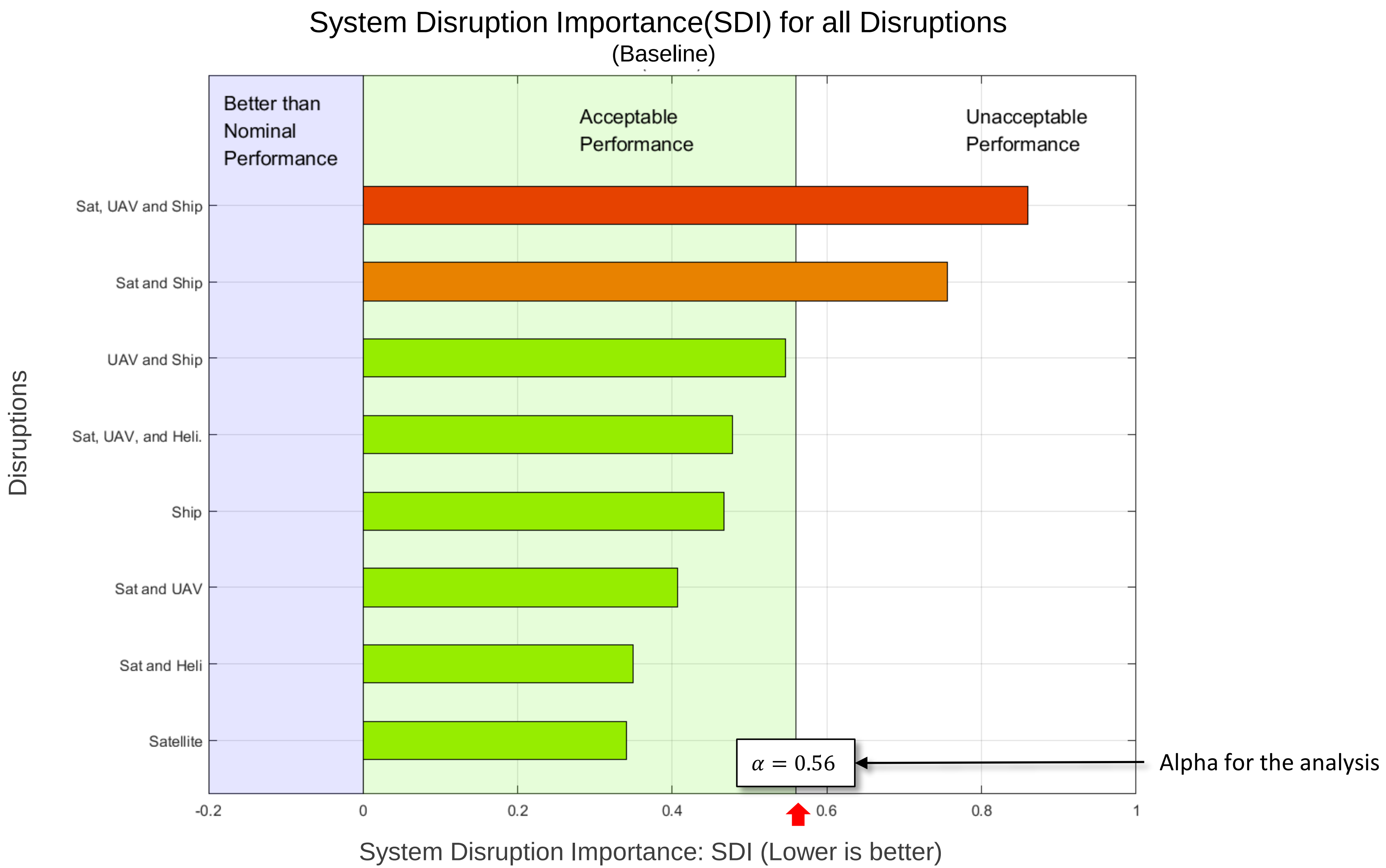
- Deci
- Use
- We
- Thu
- p
- n



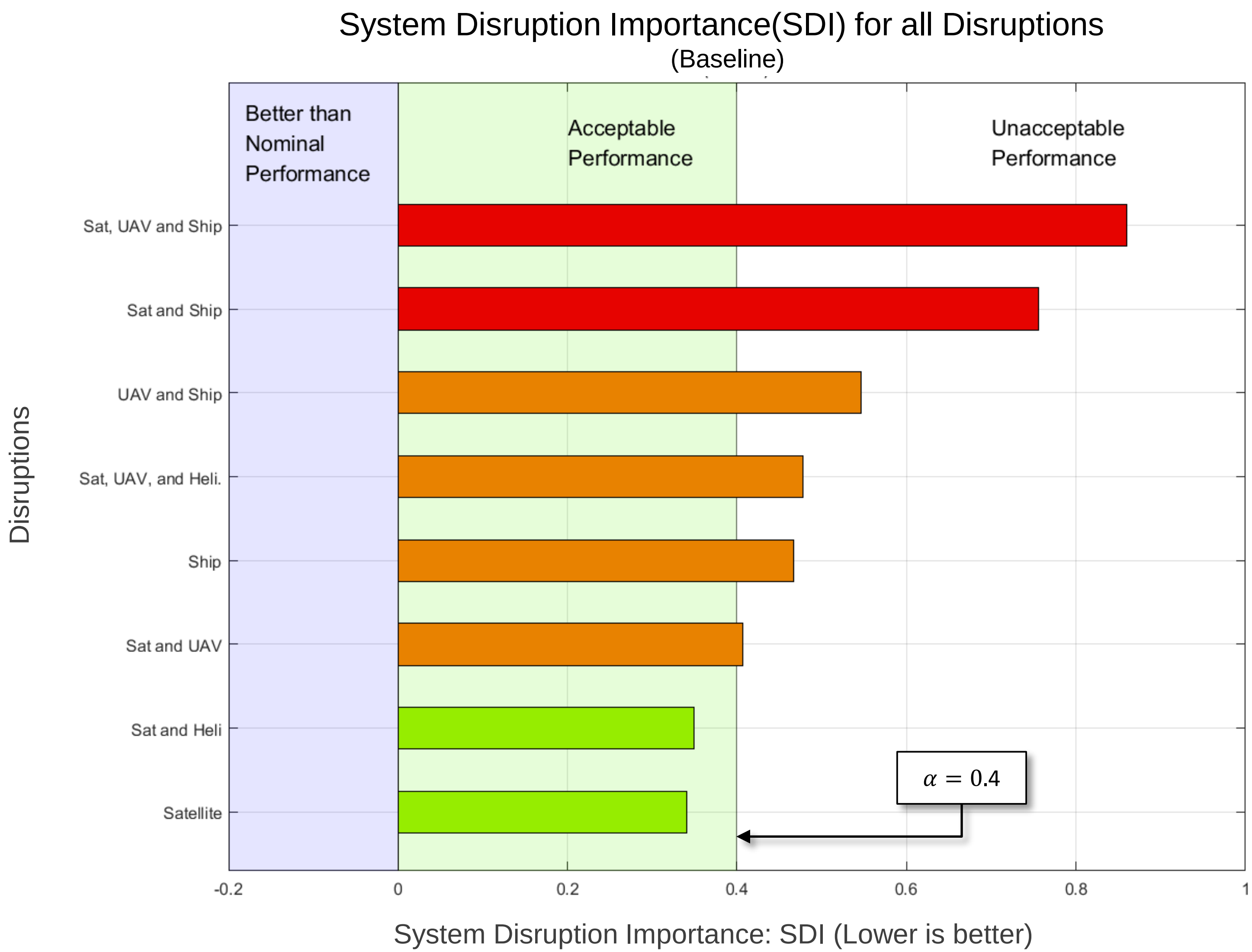
Decision threshold divides the graph into three zones



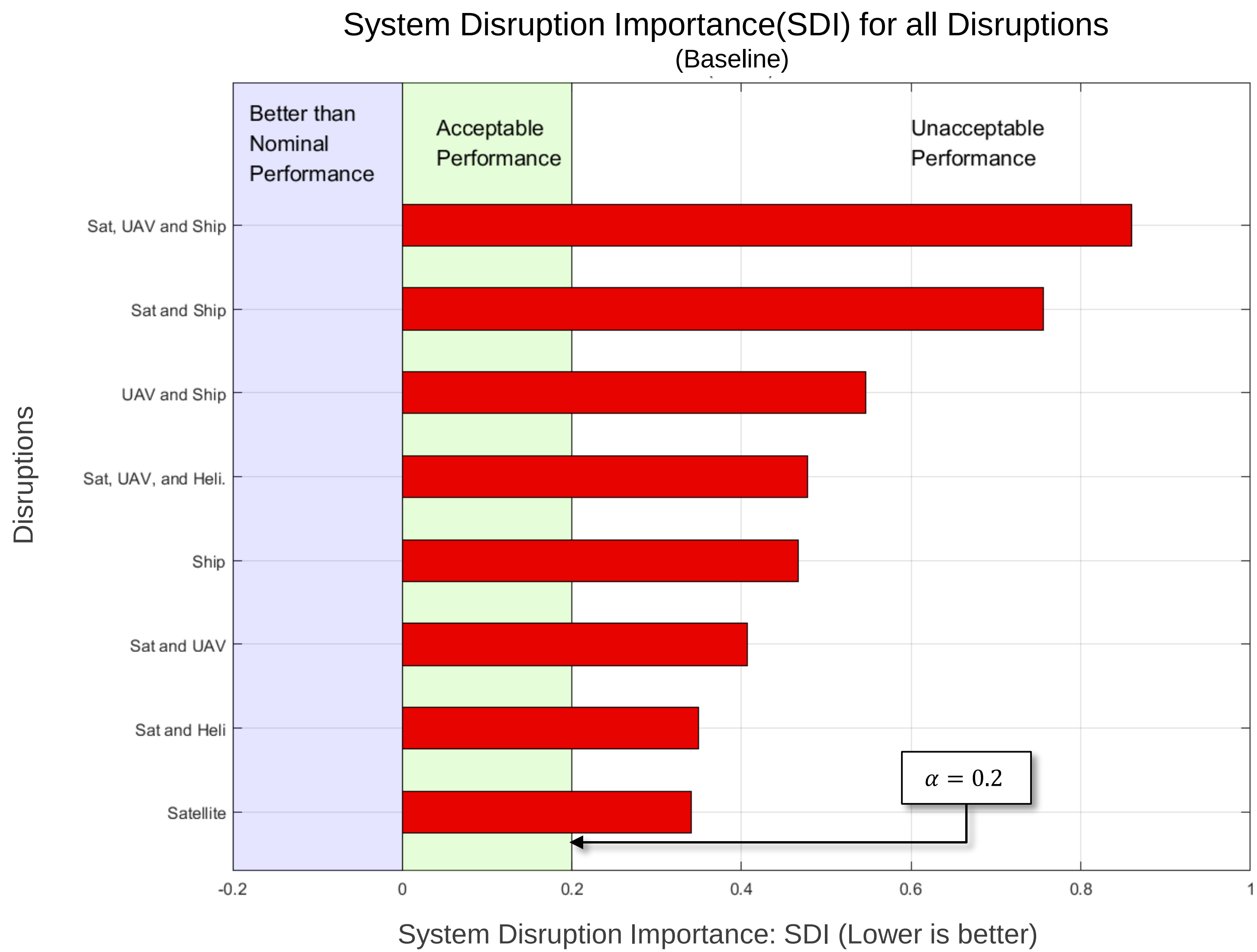
Color code disruptions based on α—red disruptions are poorly mitigated



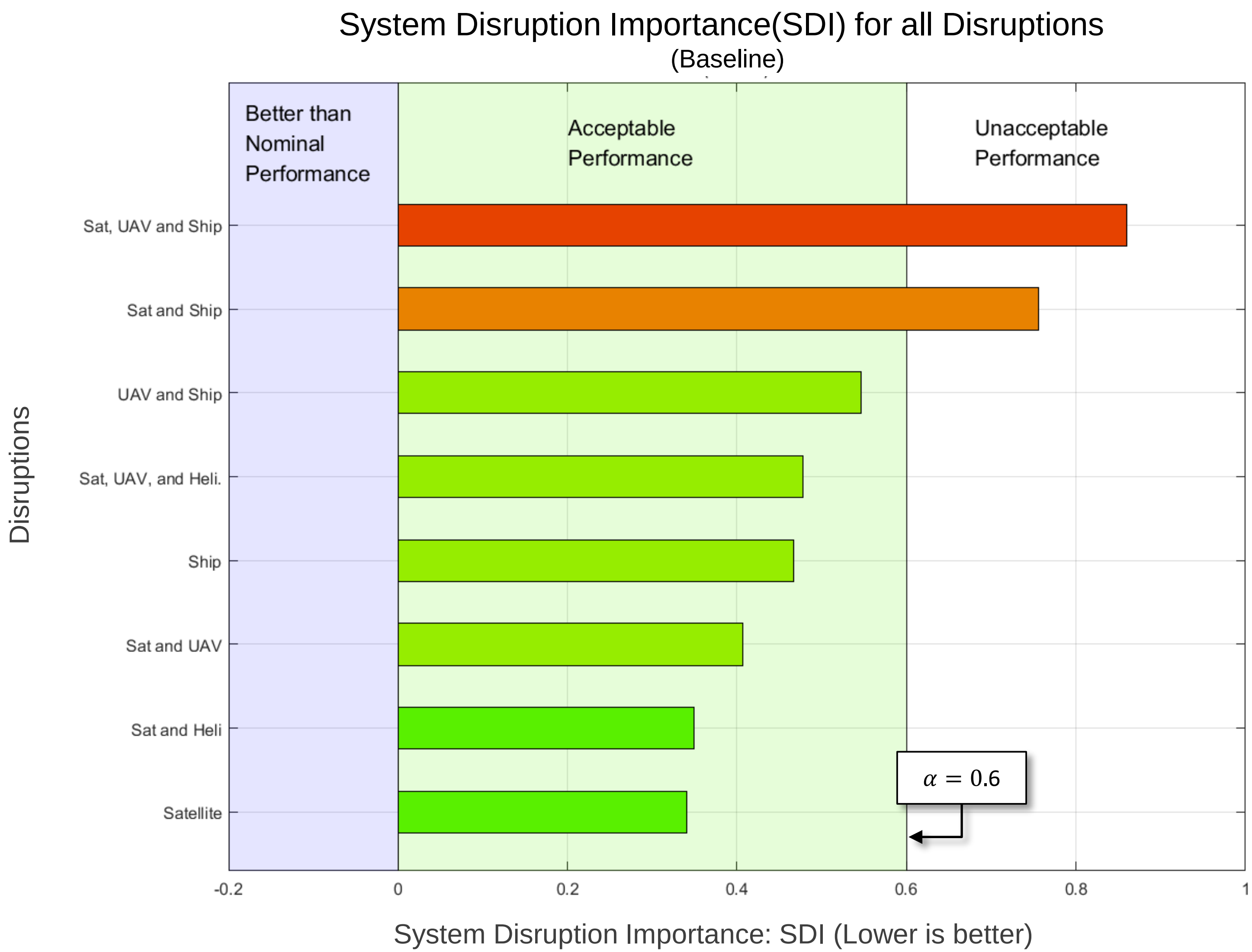
Decrease α to reflect risk



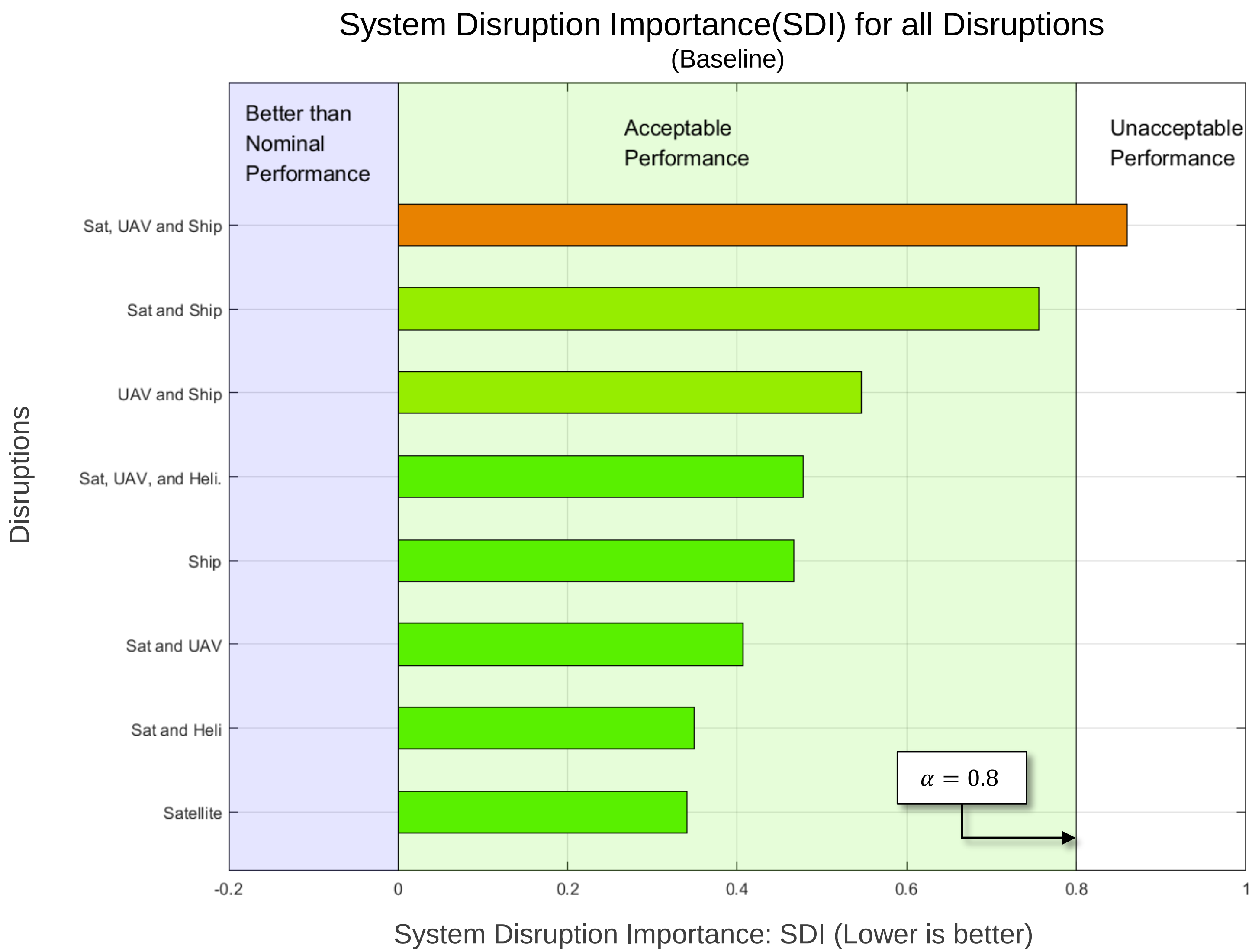
Decrease α to reflect risk aversion



Increase α to reflect risk tolerance



Increase α to reflect risk tolerance

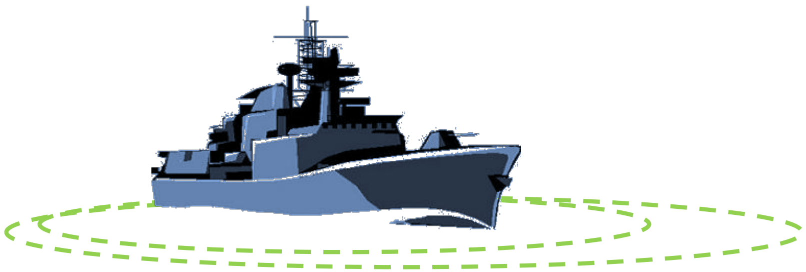


SYSTEMS ENGINEERING Research Center

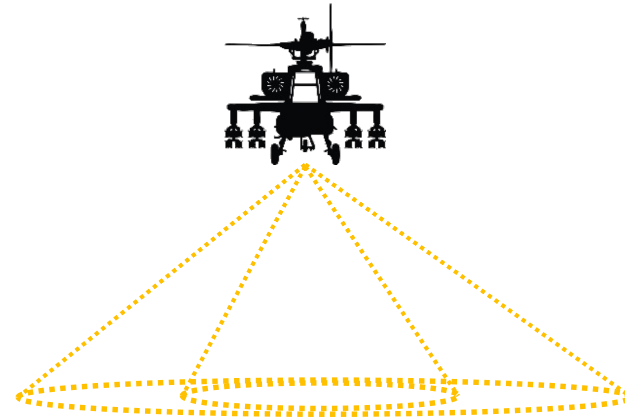
Consider four mitigations

CISA Center for Integrated Systems in Aerospace

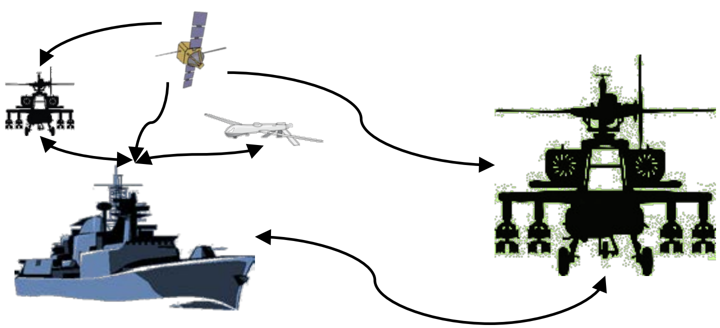
Mitigation 1: Increase Ship Radar Range



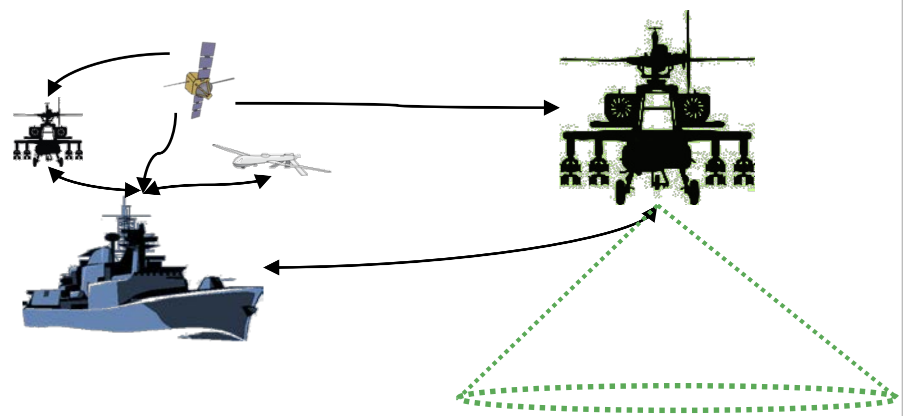
Mitigation 2: Increase Helicopter Weapon Range



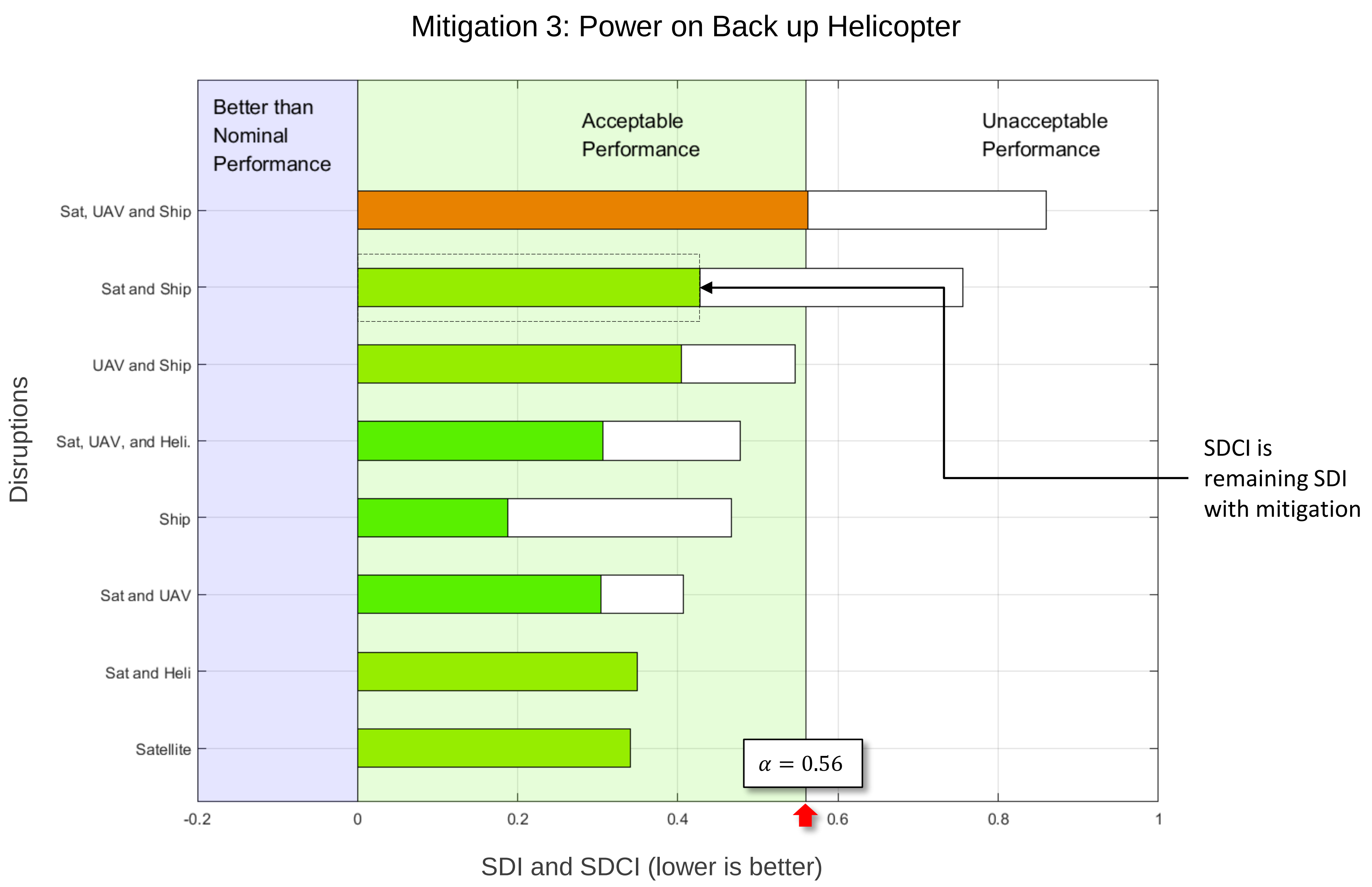
Mitigation 3: Add Backup Helicopter



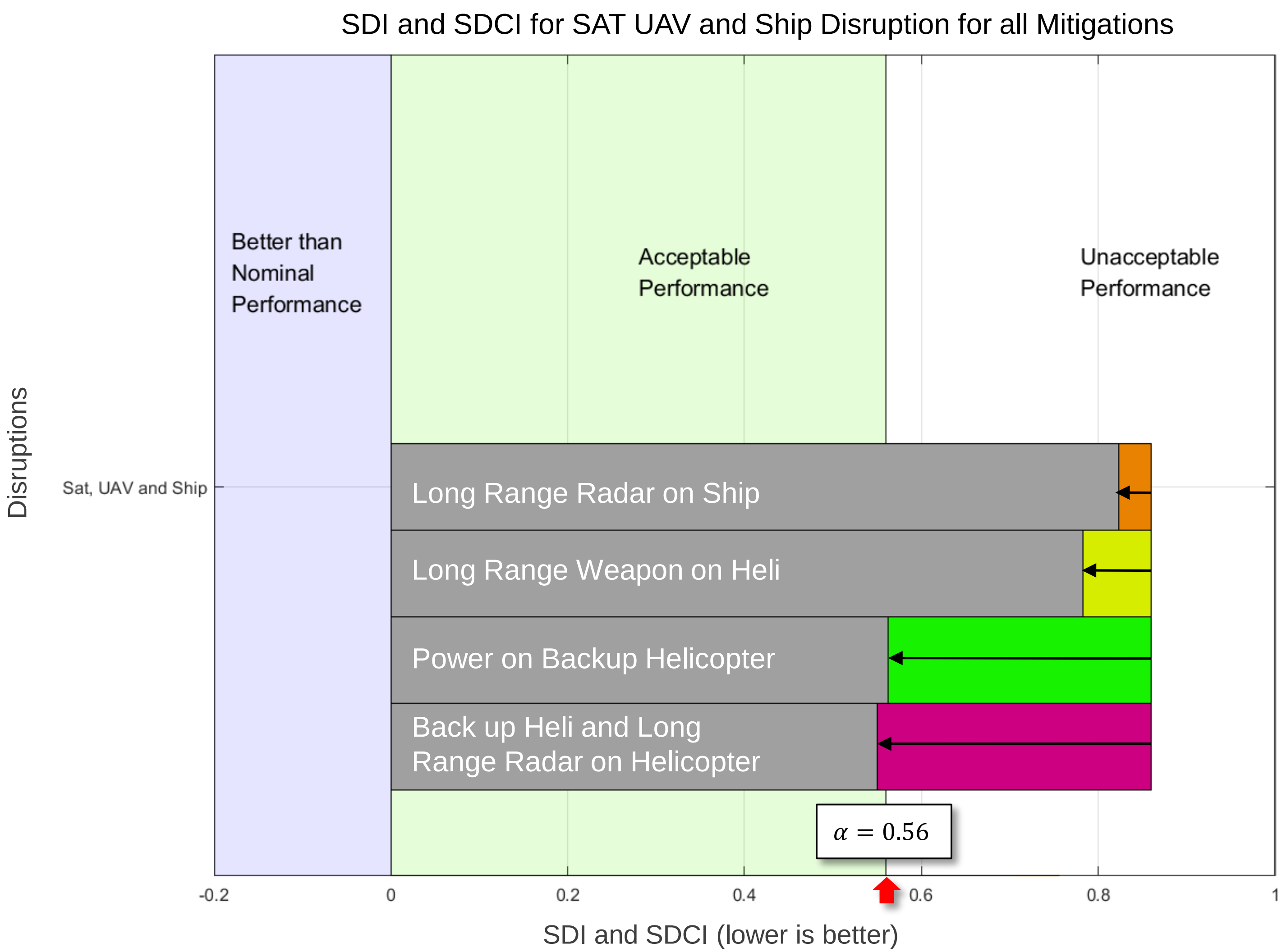
Mitigation 4: Backup Helicopter with Long Range Radar



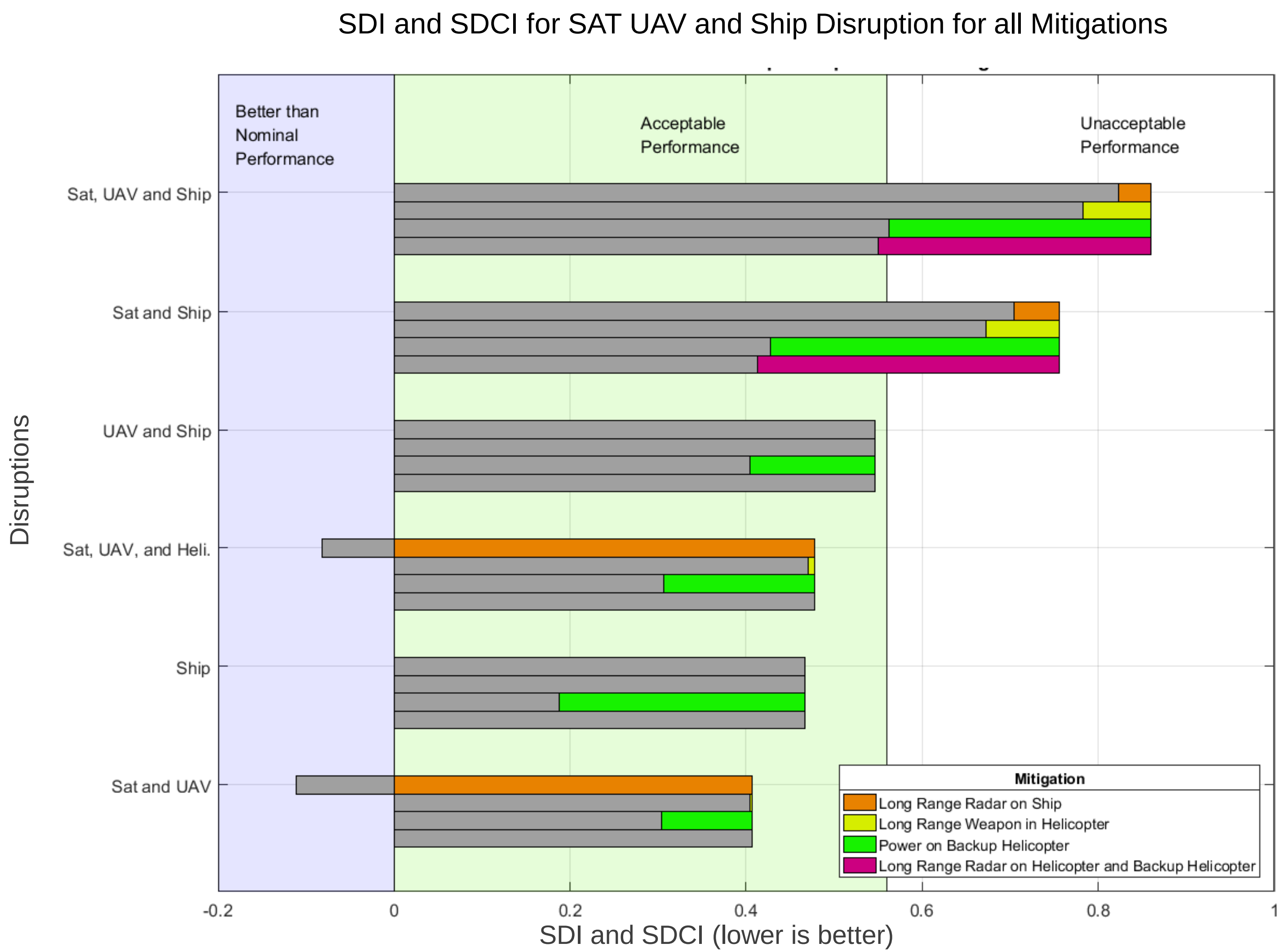
Mitigation reduces effective disruption importance, moving more disruptions into the green zone



Effect of different mitigations on a particular disruption



Resilience Map for all Mitigations



Long Range Radar and Backup Helicopter have best impact on resilience

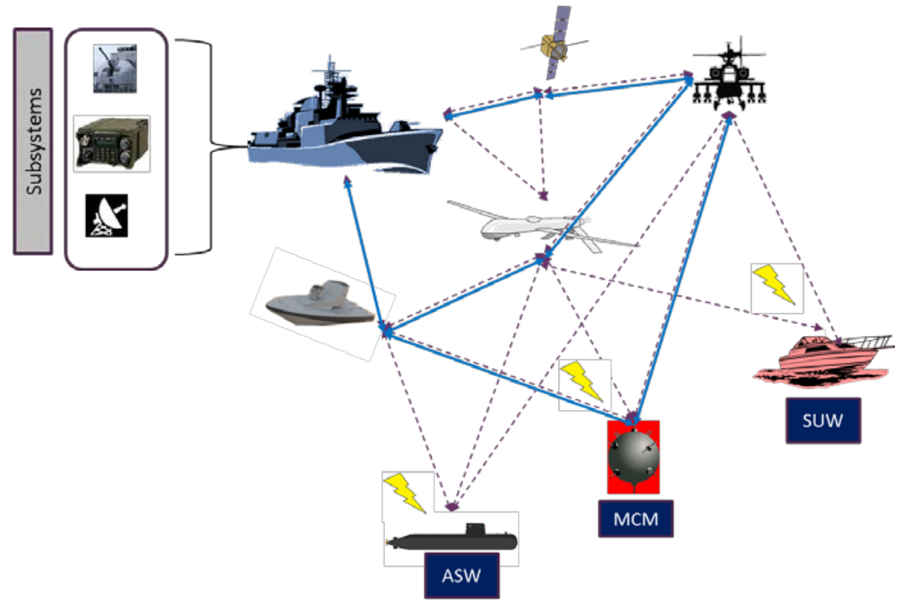
Multi-Stakeholder Dynamic Optimization (MUSTDO)

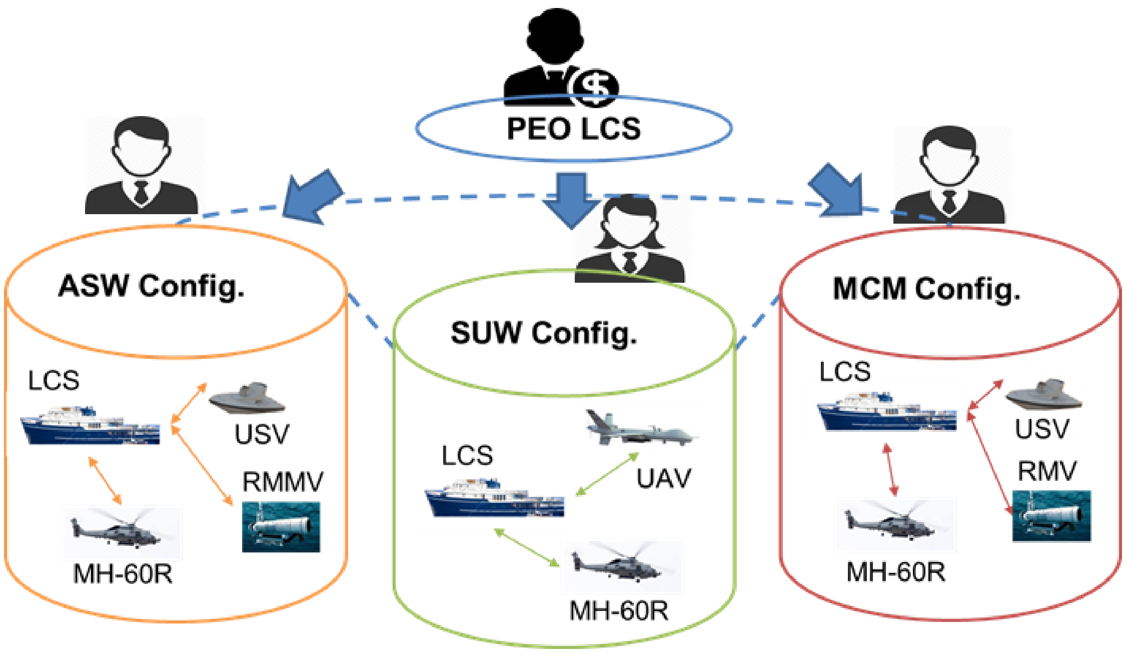
Dynamically contracting across an enterprise

- What combination of systems gives the desired aggregate SoS capabilities?
- How do we optimize multi-stage acquisitions in SoS development?
- How do we coordinate planning between local and SoS-level stakeholders?
- How do changes in system properties affect SoS development?

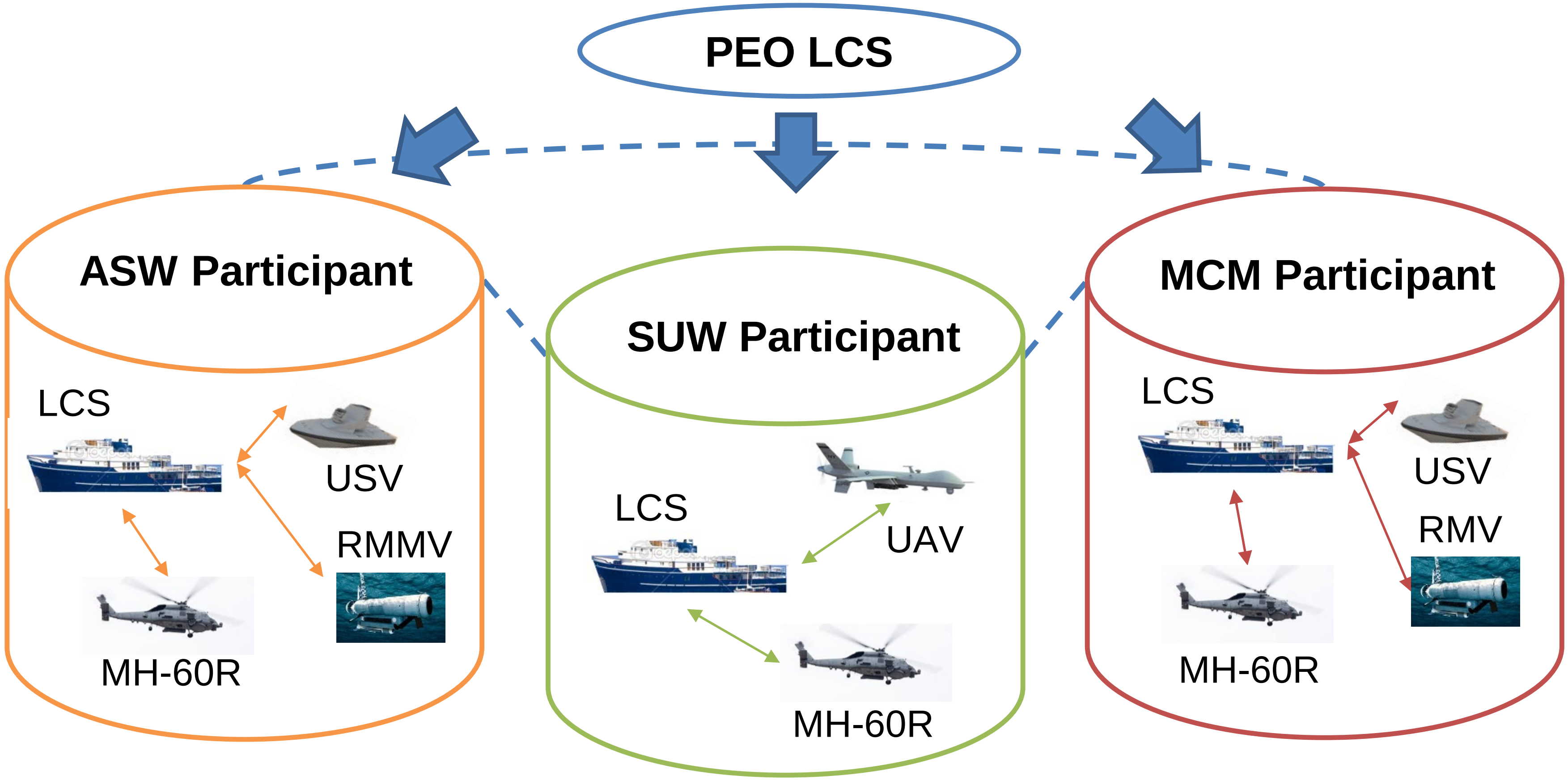
Multi-Stakeholder Dynamic Optimization (MUSTDO)

How do we acquire all of this?





Adaptation of naval warfare scenario

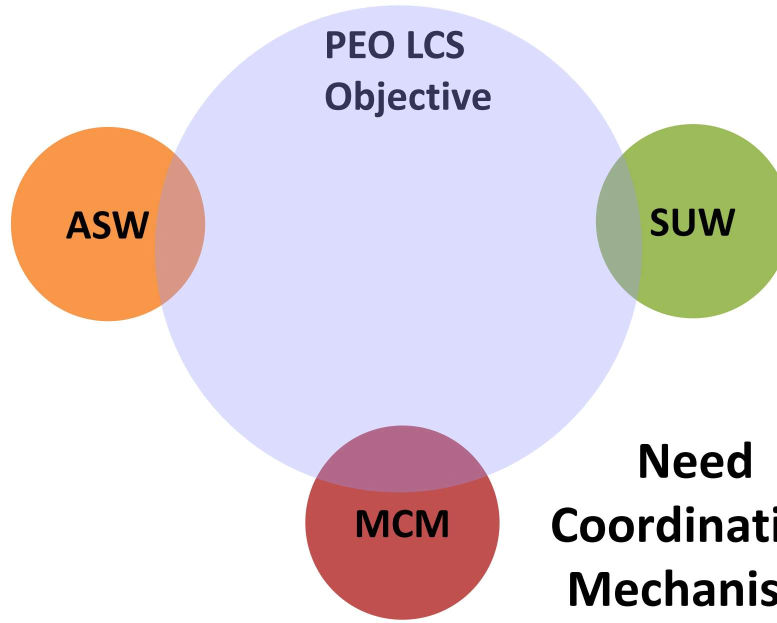


(**PEO:** Program Executive Offices; **LCS:** Littoral Combat Ship; **ASW:** Anti-Submarine Warfare; **SUW:** Surface Warfare; **MCM:** Mine Countermeasures; **MH-60R:** Multi-Mission Helicopter; **USV:** Unmanned Surface Vehicle; **RMMV:** Remote Multi-Mission Vehicle; **UAV:** Unmanned Aerial Vehicle; **RMV:** Remote Minehunting Vehicle)

What is the problem?

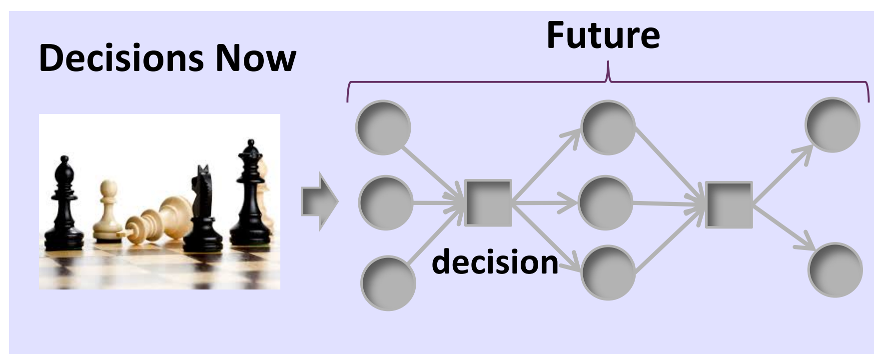
PEO LCS (SoS Manager)

SoS Participants pursue individual objectives



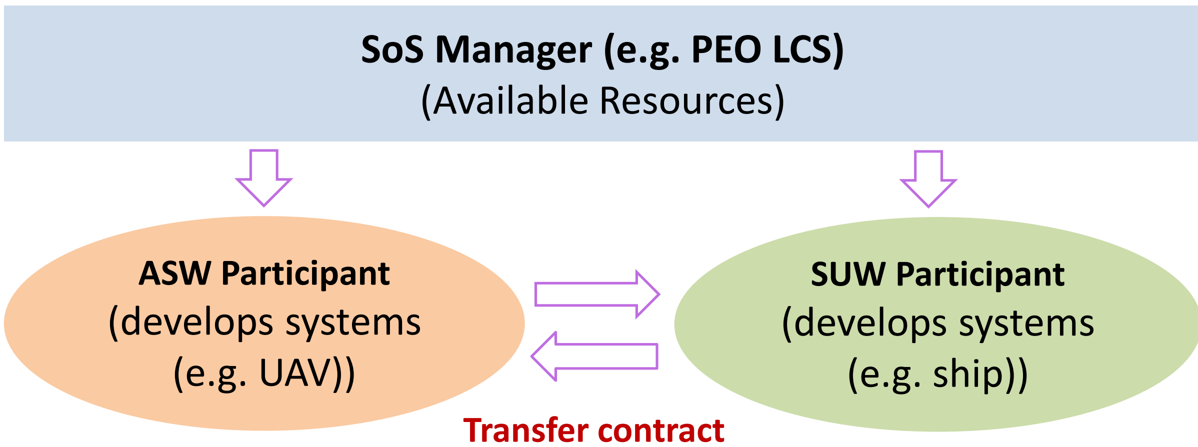
Need Coordination Mechanism

Long-term capability is not sufficiently considered



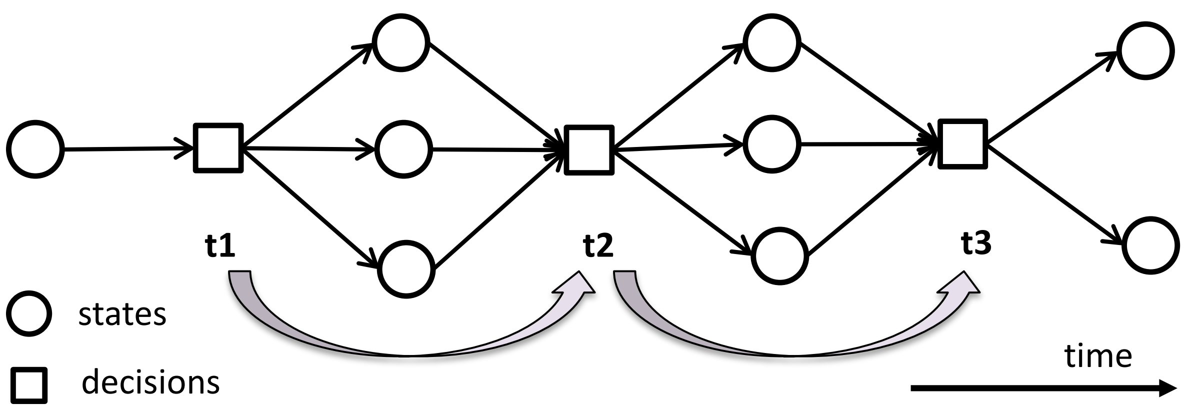
Need to Think Ahead

What is the solution approach?



Transfer Contract

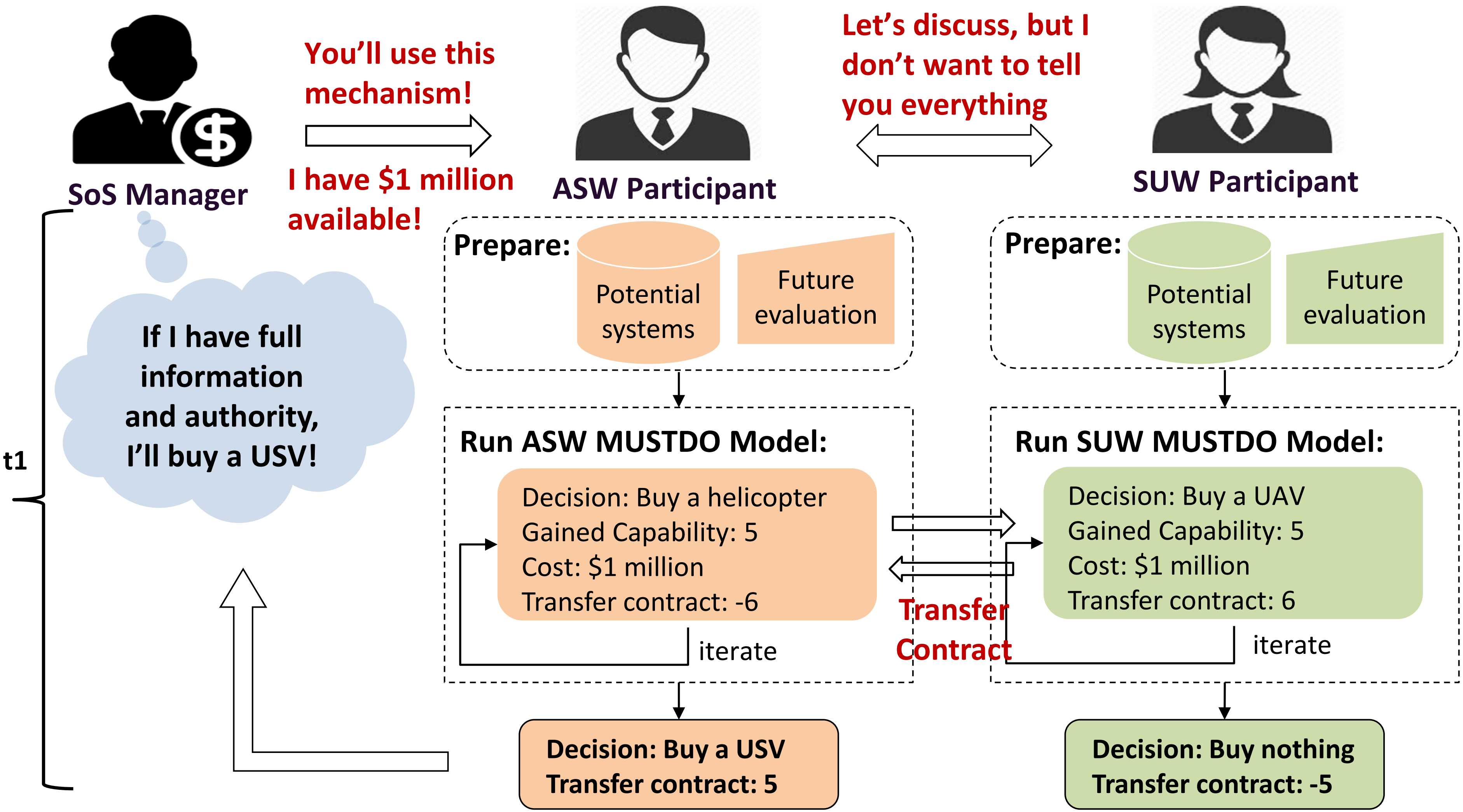
- Compensation for consuming the shared resources
- Interpretation: partial capability (technology, knowledge, etc.); monetary value



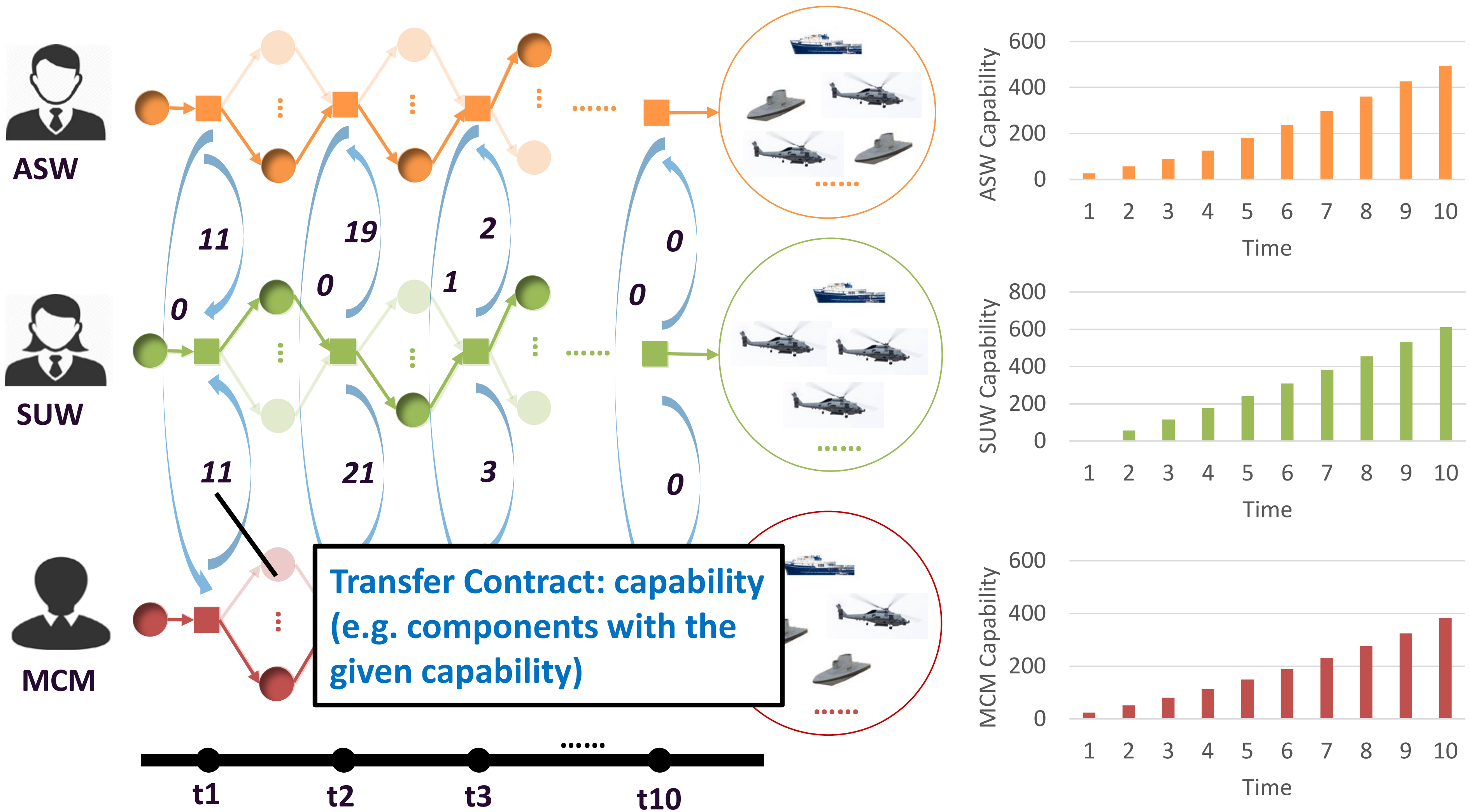
Approximate value functions

- Capture potential future values
- Associate with transfer contract

How to implement the method?



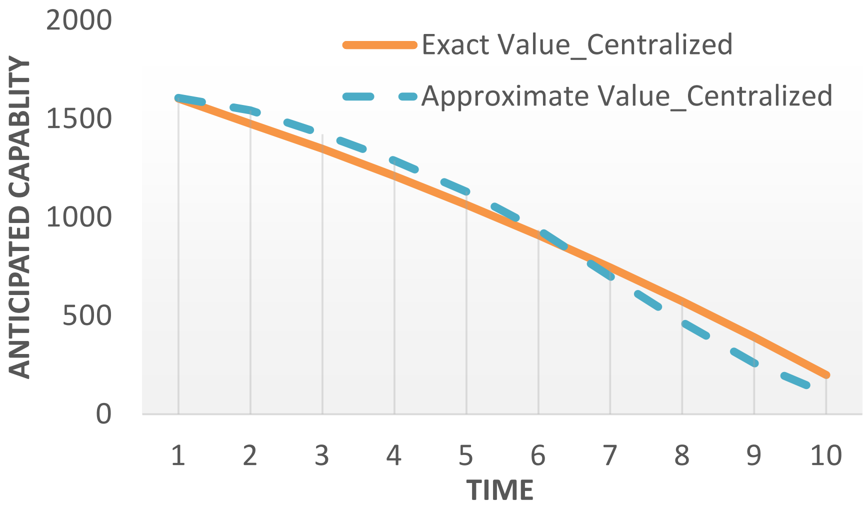
MUSTDO suggests multi-stage decisions, transfer contracts, and capabilities



MUSTDO ensures participants' and SoS manager's objectives match

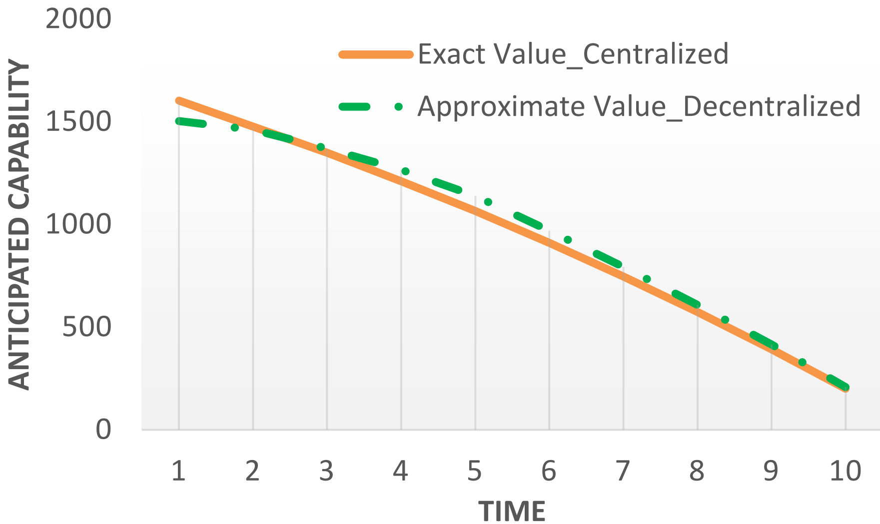
SYSTEMS ENGINEERING Research Center

CISA Center for Integrated Systems in Aerospace



Effectiveness of the approximation

- "Centralized": A benchmark case assuming that SoS manager has absolute authority
- "Approximate Value": Obtained by using approximate dynamic programming



Effectiveness of the MUSTDO mechanism

- "Approximate Value_Decentralized": Aggregated approximate value from ASW, SUW, and MCM participants

Contribution of MUSTDO to SoS development

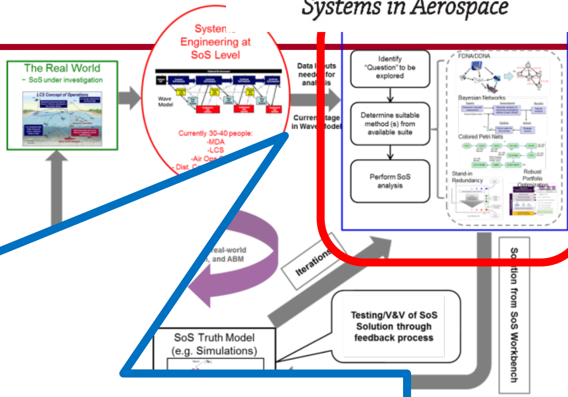
- A structured framework for SoS participants and SoS managers to plan, communicate, and negotiate with each other more effectively
  - Helps SoS managers and participants select the best architecture under uncertainty over a time period for a given budget
  - Helps decision makers to understand how they affect each other and cooperate to achieve more efficient solutions without sharing full information

RT-155 Current Engagement & Future Directions

- **Current:** Pilot experimentation, and transitions to collaborators and research partners
  - Naval Surface Warfare Center Dahlgren Division (NSWCDD)
    - Currently active and ongoing work to transition use of SODA/SDDA and SIMs toolset for use at NSWCDD
    - Concept applications successfully transitioned
  - MITRE Corporation
    - Ongoing and active work exchange on transitioning SODA/SDDA and RPO toolset for use to conduct internal case analyses
    - Future projected collaborations to deepen development and use of toolset
- **Future:** Expanding partner list and refinement based on feedback from current pilot applications on collaborator side [upcoming with Johns Hopkins APL]
- Transition strategy for software tools to be shared with broader DoD community

Backup slides

Inputs for SoS Analysis



Examining Current SoS AWB Methods

Translate user input into parameters of SoS AWB and data requirement

	User Input	AWB Parameter
FDNA/DDNA	Time to detect enemy / % of enemies detected	Operability
	Probability of radar node detecting an enemy	Self Effectiveness (SE)
	Scaled loss of operability when input missing	Strength of Dependency (SOD)
	Effects of total loss of input	Criticality of Dependency (COD)
Robust Portfolio Optimization/ ADP	Effective range of radar	System Capabilities
	Power req. of radar	System Requirements
	Types of compatible power supplies	System Compatibilities
System Importance Measures	Probability of radar loss	System Disruption Importance (SDI)
		System Recoverability Importance (SRI)

Backup slides - MUSTDO

SYSTEMS ENGINEERING Research Center

Motivation and objective

Causes of acquisition failures → Potential solutions → Proposed method

Causes of acquisition failures

- **Authority conflicts**
- **Lack of structured control**
- **Misalignment of objectives among the systems**
- **Evolutionary nature**
- **Requirement creep**
- **Emergent behaviors**
- **Unstable budget**
- **……**

Potential solutions

- **Process and workflow formalization (systems or SoS engineering)**
- **Simulation tools (e.g. agent-based simulation)**
- **Quality based analysis (e.g. resilience, flexibility, robustness)**
- **Computational tools (e.g. optimization)**
- **……**

Proposed method

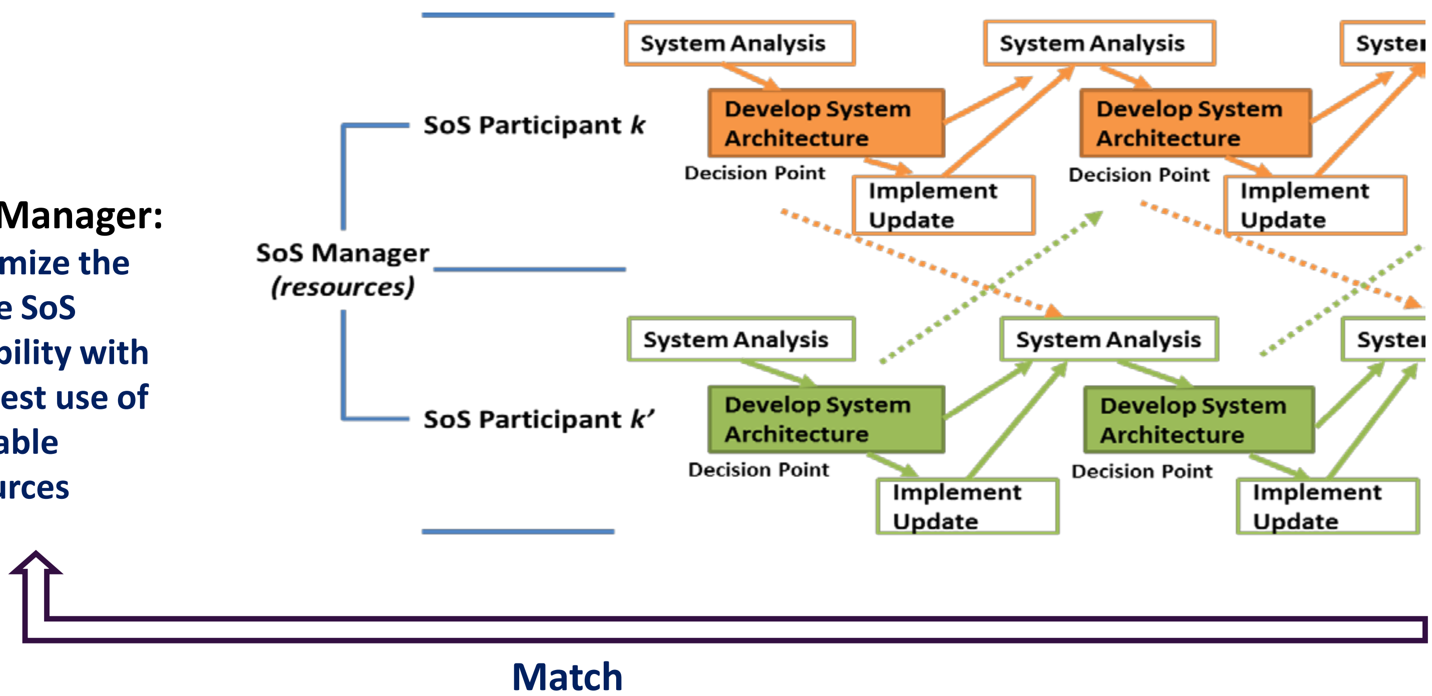
Multi-Stakeholder Dynamic Optimization (MuSTDO)

- **Support architecture selection under uncertainty**
- **Support coordination of resource conflict between stakeholders on both current and future capability**

Research objective

Develop the mathematical formulation and solution approach that generates an approximately optimal set of multi-stage architectural decisions for SoS managers with limited collaboration between conflicted and independent SoS participants

SoS Manager:
Maximize the entire SoS capability with the best use of available resources



SoS Participants:
Maximize its own capability under the mechanism and resources provided by SoS manager

Solution approach: basic mathematical formulation

	SoS Manager	SoS Participant
Objective Function	$\max E\{\sum_{t=1}^{T} \boldsymbol{cap}_t \cdot \boldsymbol{x}_t\}$	$\max E\{\sum_{t=1}^{T}(\boldsymbol{cap}_t^k \cdot \boldsymbol{x}_t^k + TC_t^k)\}$
Constraints: Budget Limits	$\boldsymbol{cost}_t \cdot \boldsymbol{x}_t \leq \boldsymbol{b}_t$	$\sum_{k \in \mathcal{K}} \boldsymbol{cost}_t^k \cdot \boldsymbol{x}_t^k \leq \boldsymbol{b}_t$
Constraints: Binary Decisions	$x_{i,t} \in \{0,1\},\ i \in \mathcal{I}$	$x_{i,t}^k \in \{0,1\},\ i \in \mathcal{I}^k$
Constraints: Compatibility	$x_{i,t} + x_{j,t} = 1,\ i \neq j, i, j \in \mathcal{I}$	$x_{i,t}^k + x_{j,t}^k = 1,\ i \neq j, i, j \in \mathcal{I}^k$
Constraints: Transition Function	$\boldsymbol{b}_{t+1} = \boldsymbol{b}_t - \boldsymbol{cost}_t \cdot \boldsymbol{x}_t$ $\boldsymbol{cap}_{t+1} = \boldsymbol{cap}_t + \widehat{\boldsymbol{cap}}_{t+1}$	$\boldsymbol{b}_{t+1} = \boldsymbol{b}_t - \sum_{k \in \mathcal{K}} \boldsymbol{cost}_t^k \cdot \boldsymbol{x}_t^k$ $\boldsymbol{cap}_{t+1}^k = \boldsymbol{cap}_t^k + \widehat{\boldsymbol{cap}}_{t+1}^k$

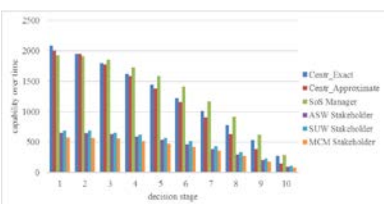
SYSTEMS ENGINEERING Research Center

Solution approach: detailed workflow of the method

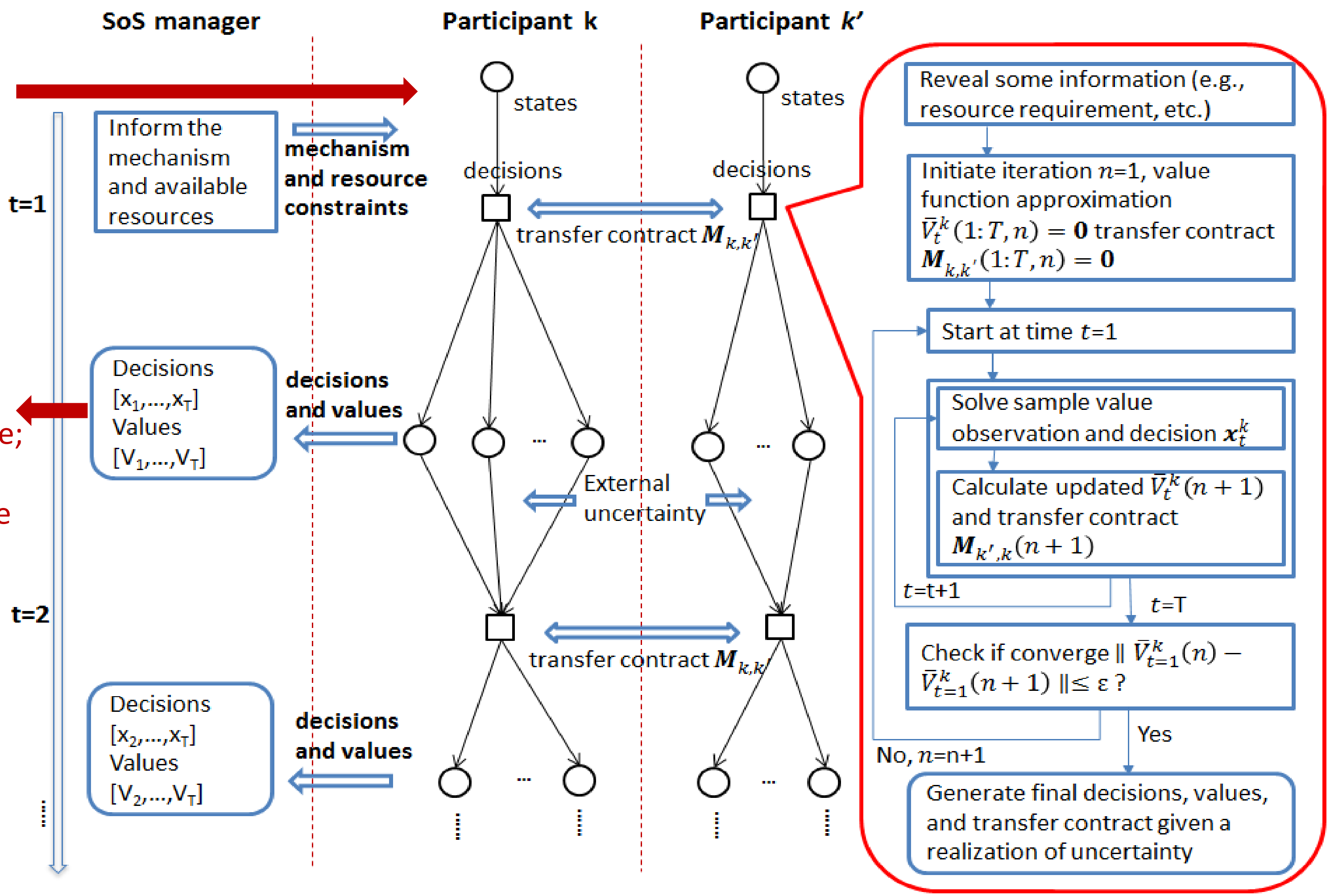
Input:
system capability;
resource requirement;
available resources

Participants	Systems	Capability	Resource Requirement
ASW	MH-60R	50	25,000,000
	USV	30	630,000
SUW	MH-60R	65	30,000,000
	UAV	35	15,000,000

Output:
gained capability over time;
decisions over time;
transfer contract over time



Time	(From) ASW		SUW	
	(To) SUW	MCM	ASW	MCM
t=1	12.82	8.89	0	0
t=2	0.73	0.53	0	0
t=3	13.60	9.43	17.59	17.92
t=4	0.71	0.53	28.30	25.04
t=5	12.93	8.96	16.75	17.06

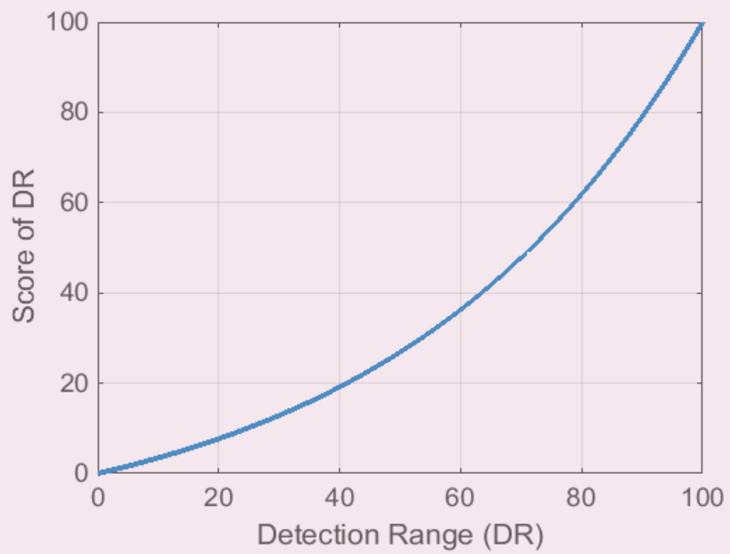


Preparation: obtain capability index

Identify key attributes → Obtain scoring function → Calculate capability index

$$v(DR) = 100 \cdot \frac{1-\exp(DR/\rho)}{1-\exp((DR_{max}-DR_{min})/\rho)}$$

- Detection range
- Probability of kill



additive value model:

$$cap(l) = \sum_{m=1}^{n} \omega_m v_m(l_m)$$

cap: capability index

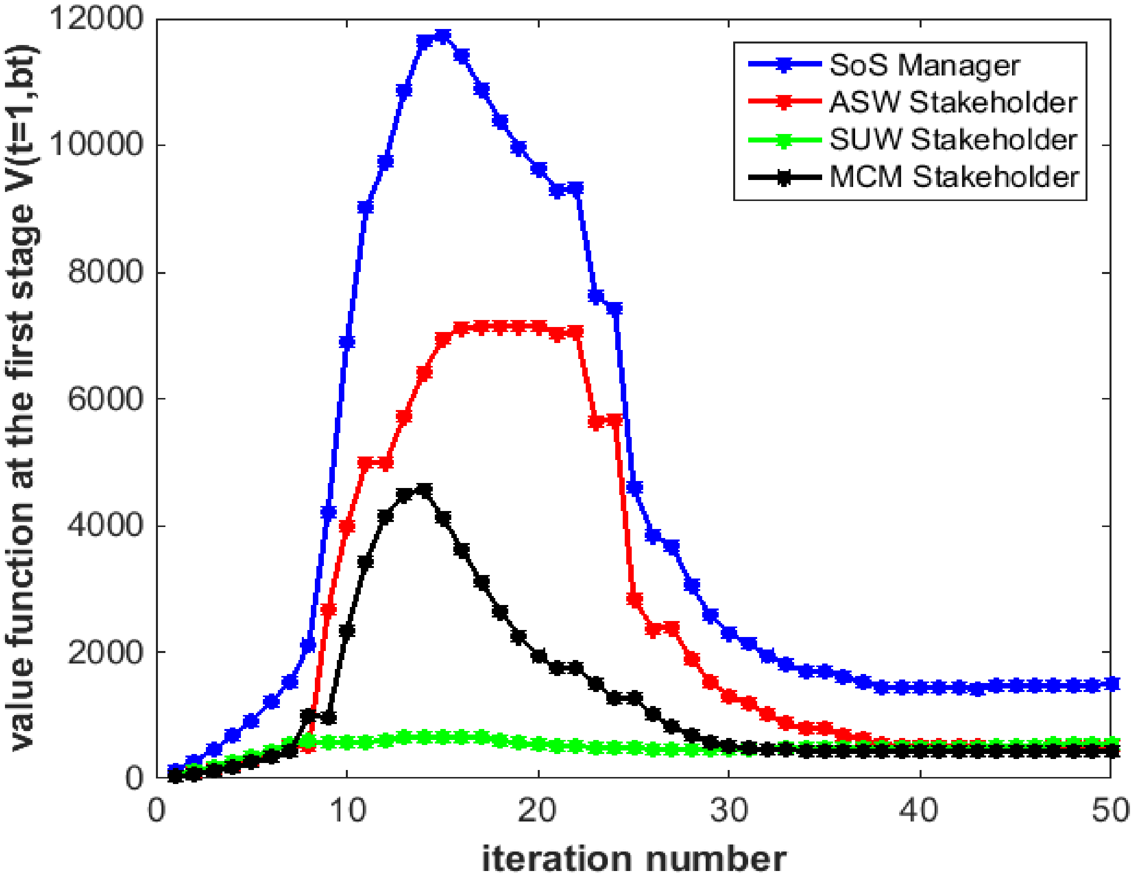
m: measure of merit

ω: weight

v: value at the scoring function

l: level of the measure of merit

Deterministic experiments: convergence



- Convergence criteria: ||approximate value (n+1) – approximate value (n) ||<= ε

- Convergence: the value function at the first stage gets converged in around 50 iterations

Backup slides - SODA

Systems Operational Dependency Analysis (SODA)

Convenient parametric model of complex systems and SoS behavior

- **Insight into causes of observed behavior** without need for complete simulation
  → Asking "Why?" rather than Just "How?"
- **Intuitive parameters** of one-to-one dependencies used to model cascading effects
  - Trade-off details for ease of use, intuitiveness, fast analysis
- **Information for high level, early design/architecting decisions**
  - identification of criticalities
  - flawed vs. promising architectures in early design

Improvement of previous parametric input/output models

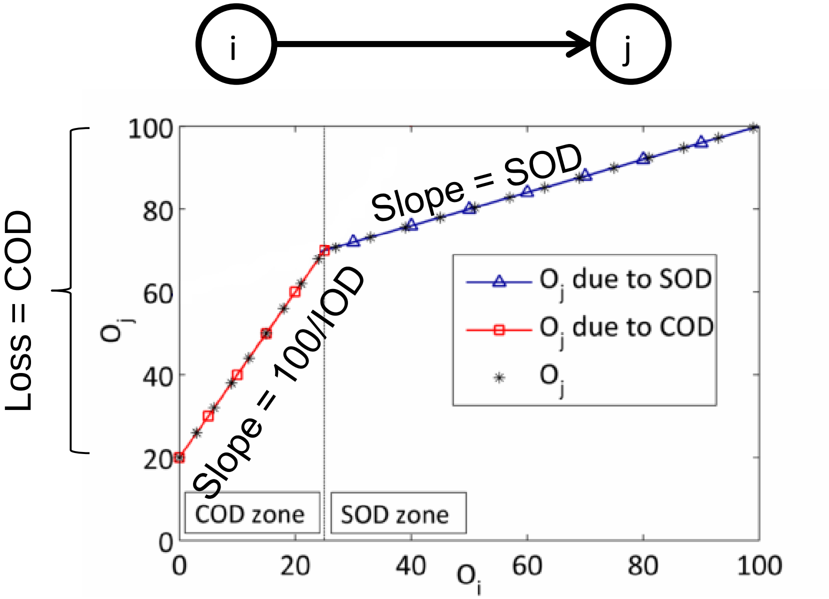
Leontief model
linear economic model

Input/Output model (Haimes)
linear 2-parameters model for infrastructures

FDNA (Garvey and Pinto)
2-parameters dependency model for capabilities portfolio

SODA
3-parameters (and internal status) dependency model for system analysis and architecting

SODA: input/output model

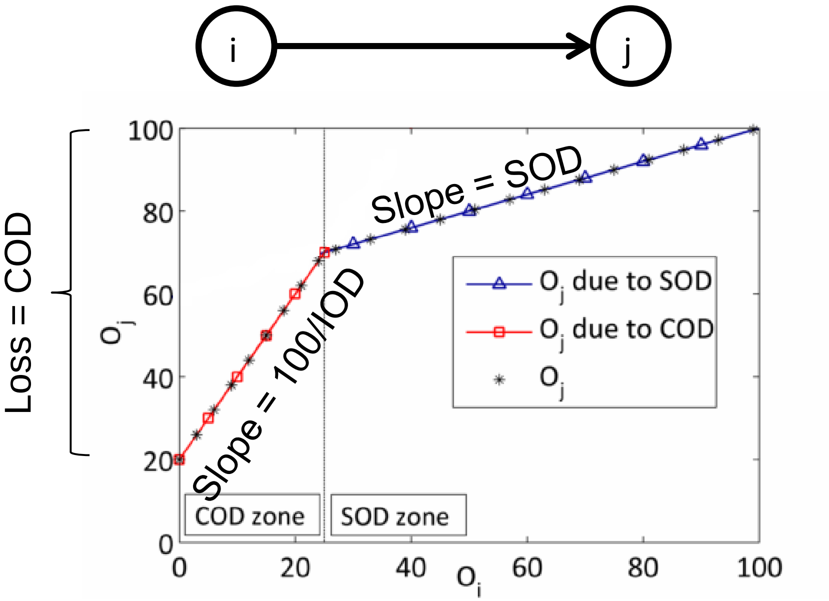


Operability of node j in function of the operability of node i (Se_j = 100)

- SODA computes the operability O of nodes, based on:
  - **Self-Effectiveness (SE),** i.e. the internal status
  - **Strength of Dependency (SOD),** i.e. how much of a system's operability depends on the feeder systems.
  - **Criticality of Dependency (COD),** i.e. loss in operability when the feeders fail completely.
  - **Impact of Dependency (IOD),** i.e. how wide is the "COD zone".

- Values might be assessed through ABM simulation, historical/experimental data, or expert opinion
- The use of one-to-one dependencies and intuitive parameters make this model convenient in case of complex systems

SODA: input/output model



Operability of node j in function of the operability of node i (Se_j = 100)

Root nodes:

$$O_i = SE_i$$

Dependent nodes:

$$O_j = \min\left(O_j^S, O_j^C\right)$$

Term dependending on SOD:

$$O_j^S = \frac{1}{n}\sum_{i=1}^{n} O_{ij}^S$$

$$O_{ij}^S = \alpha_{ij} O_i + \left(1 - \alpha_{ij}\right) SE_j$$

Term depending on COD:

$$O_j^C = \min\left(O_{1j}^C, O_{2j}^C, \cdots, O_{nj}^C\right)$$

$$O_{ij}^C = \left(100 - \beta_{ij}\right) W_i^{\lambda} + \frac{100}{\gamma_{ij}} O_i$$

SYSTEMS ENGINEERING
Research Center

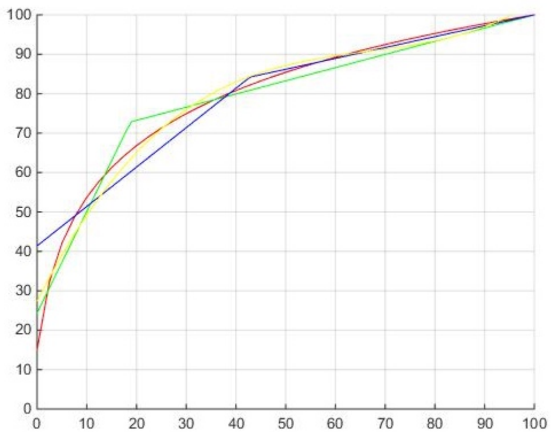
SODA: parameters

Parametric regression model

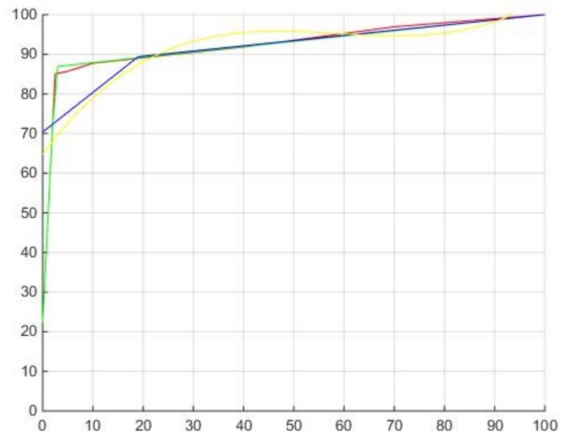
$$\min_{\alpha_{ij},\beta_{ij},\gamma_{ij}} \sum_{a=1}^{k} \| [O_1^a(\alpha,\beta,\gamma) \cdots O_n^a(\alpha,\beta,\gamma)] - [D_1^a \cdots D_n^a] \|^2$$

$$s.t.\quad \begin{aligned} 0 \le \alpha_{ij} \le 1 \\ 0 \le \beta_{ij} \le 100 \\ 0 \le \gamma_{ij} \le 100 \end{aligned}$$

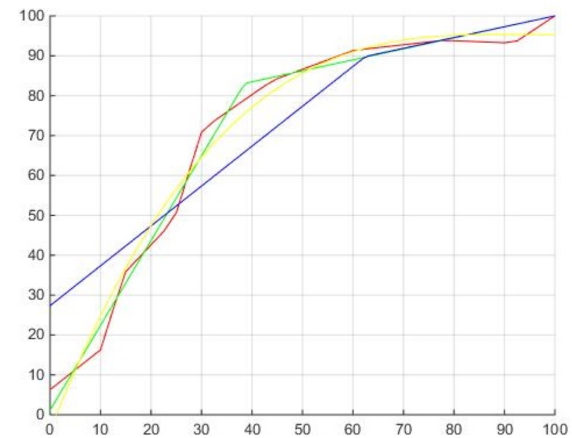
Comparison to FDNA and Response Surface Methodology



Modeling of logarithmic I/O.
RMS error in FDNA: 5.08.
RMS error in SODA: 2.99



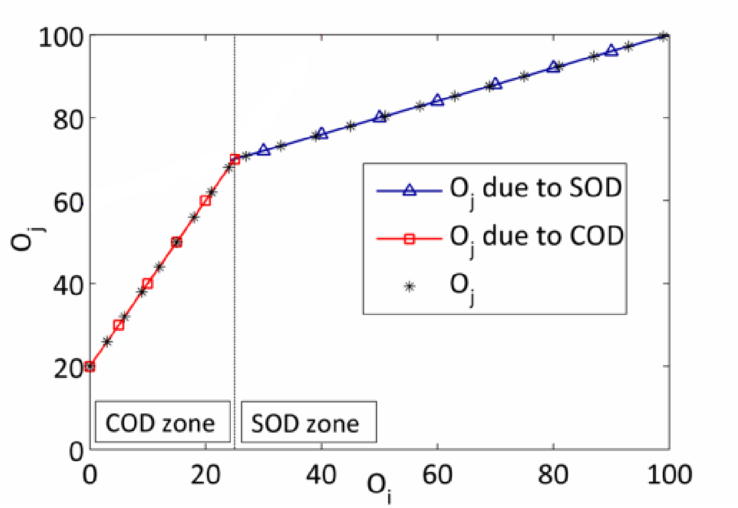
Modeling of step-like I/O.
RMS error in FDNA: 8.16.
RMS error in SODA: 0.51



Modeling of user-input I/O.
RMS error in FDNA: 10.08.
RMS error in SODA: 2.55.

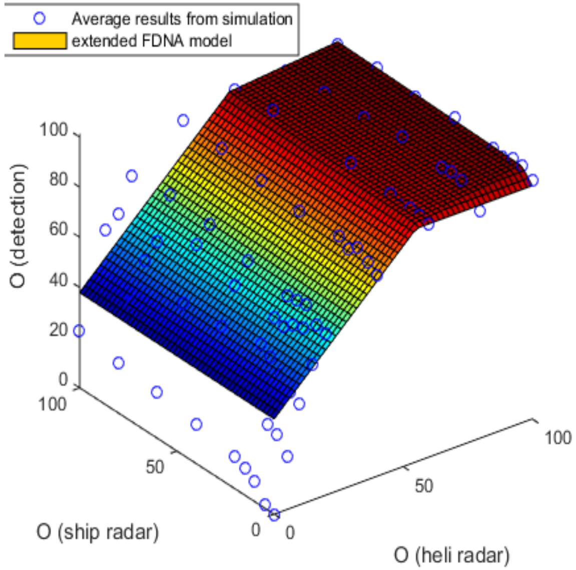
Actual I/O function — SODA
Response Surface — FDNA

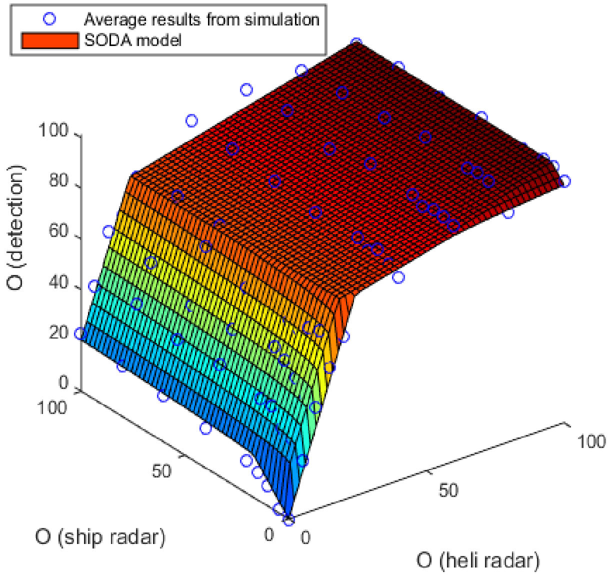
SODA: multiple dependencies



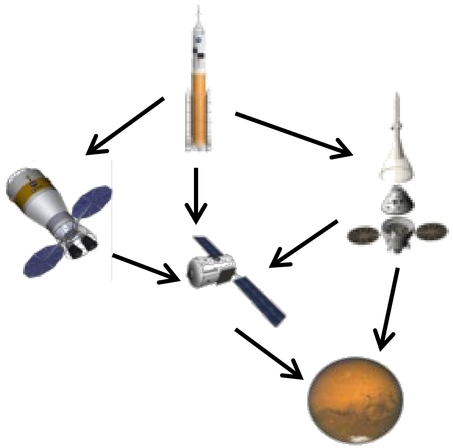
Single dependency input/output SODA model

Multiple dependency data fitting

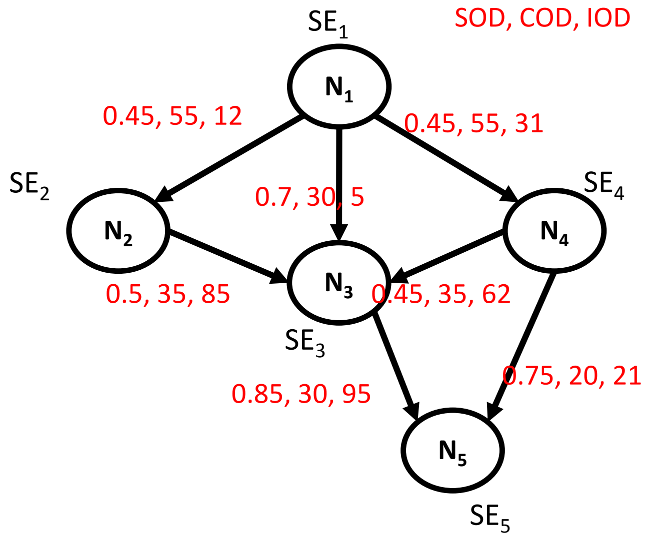




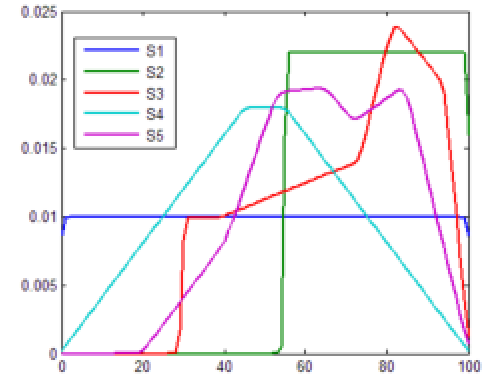
SODA: analytical process



Operational Dependencies in a complex system

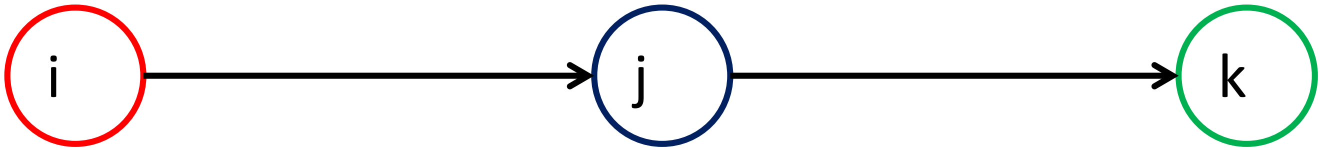


SODA representation: operability depends on **internal status** (Self-Effectiveness SE), and **Strength, Criticality, and Impact of Dependency (SOD, COD, IOD)**

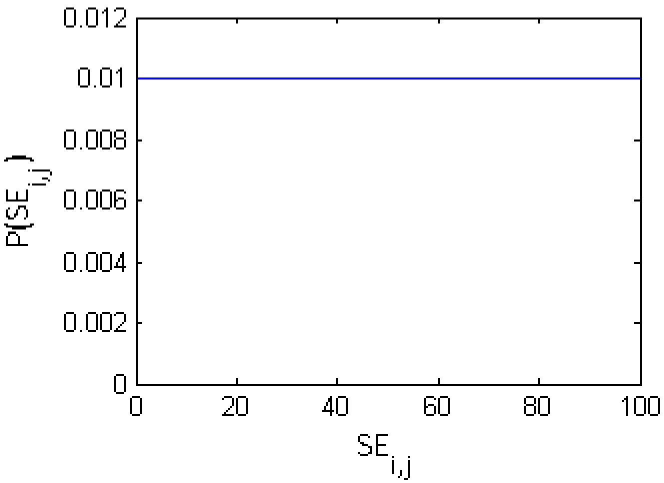


Effects of failures in 2 systems: probability of the operability of interest

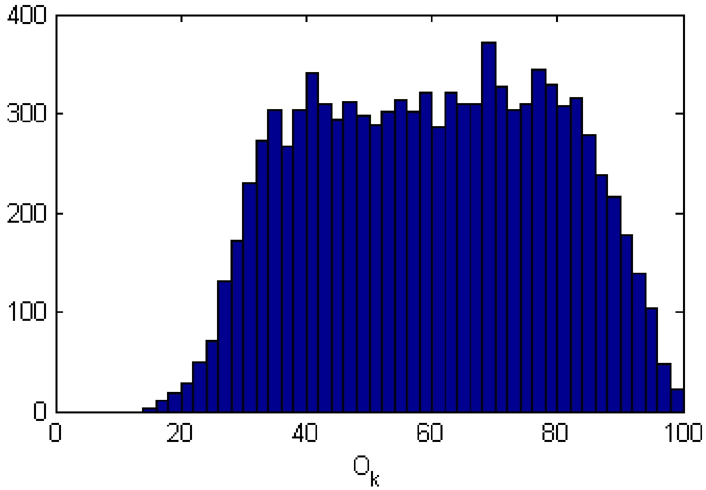
How to "use" SODA parameters



Parameters of dependency unknown. Observed behavior (when the probability distribution of the Self-Effectiveness of systems i and j is uniform between 0 and 100, and system k is working at maximum Self-Effectiveness):



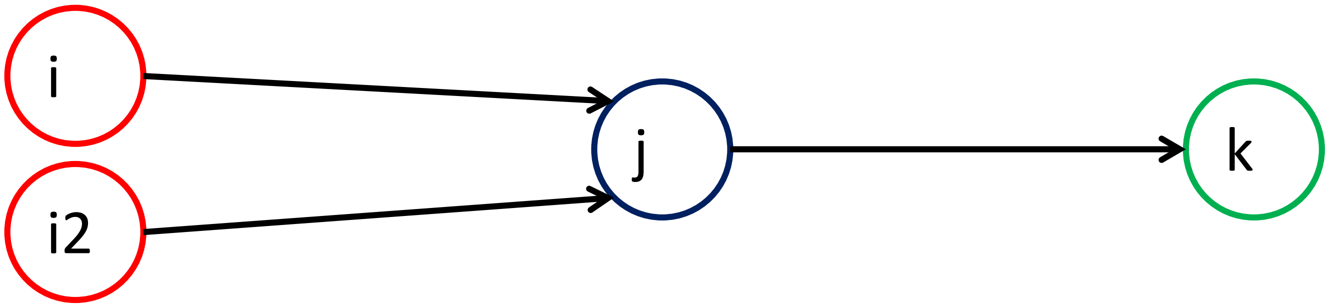
Probability distribution of the Self-Effectiveness of systems i and j



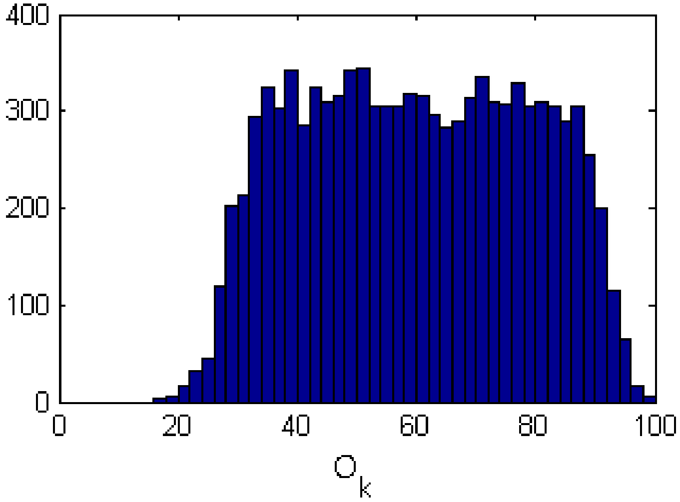
Histogram of instances having a given operability of system k (10000 total instances)

$E(O_k)$=60.1 $\sigma(O_k)$=19.1

How to "use" SODA parameters: Redundant system i2



System i is supported by a redundant system (with uniform probability distribution of the Self-Effectiveness)

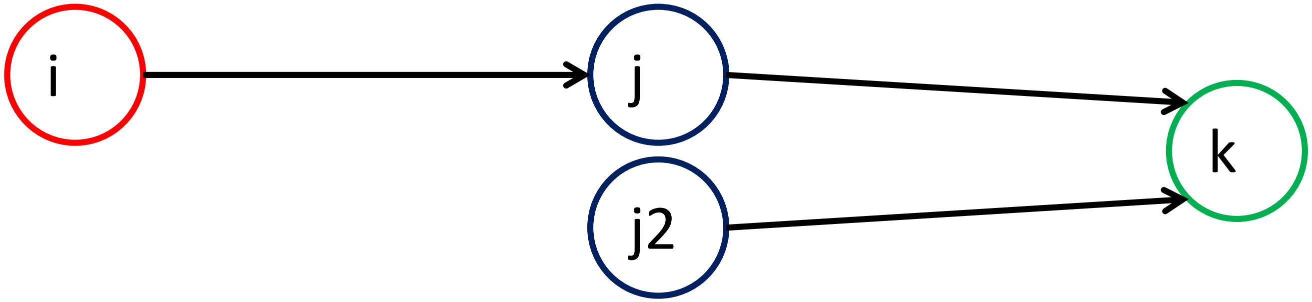


No substantial change

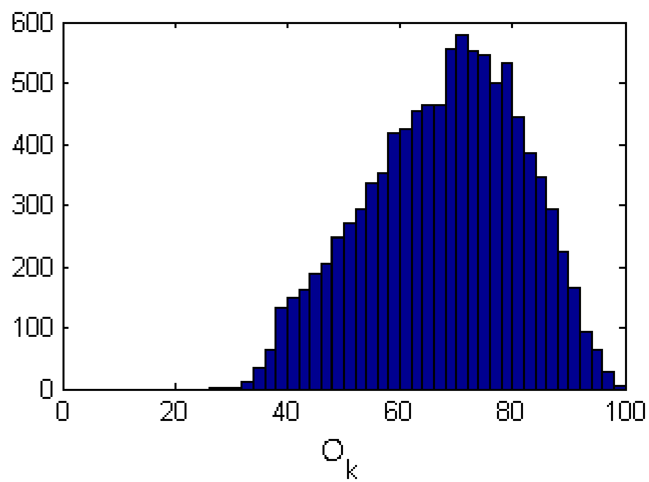
Histogram of instances having a given operability of system k (10000 total instances)

$E(O_k)$=59.9 $\sigma(O_k)$=18.8

How to "use" SODA parameters: Redundant independent system j2



System j is supported by a redundant system (with uniform probability distribution of the Self-Effectiveness)

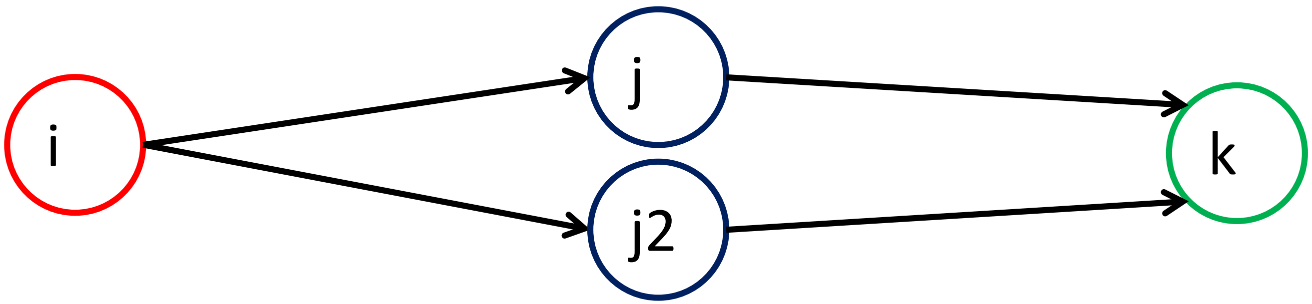


Improvement

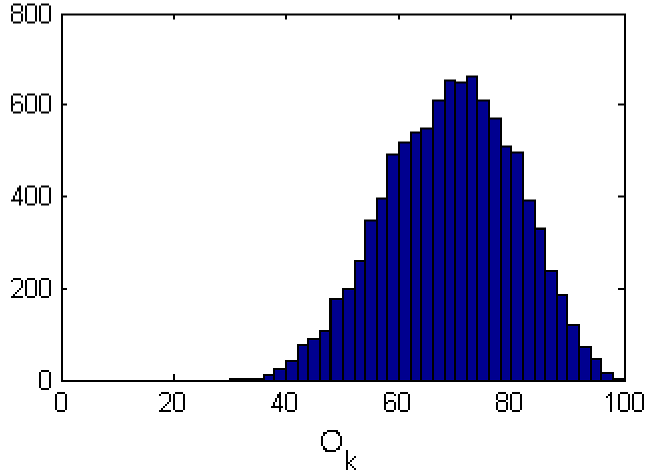
Histogram of instances having a given operability of system k (10000 total instances)

$E(O_k)$=67.9 $\sigma(O_k)$=13.7

How to "use" SODA parameters: redundant dependent system j2



System j is supported by a redundant system (with uniform probability distribution of the Self-Effectiveness)

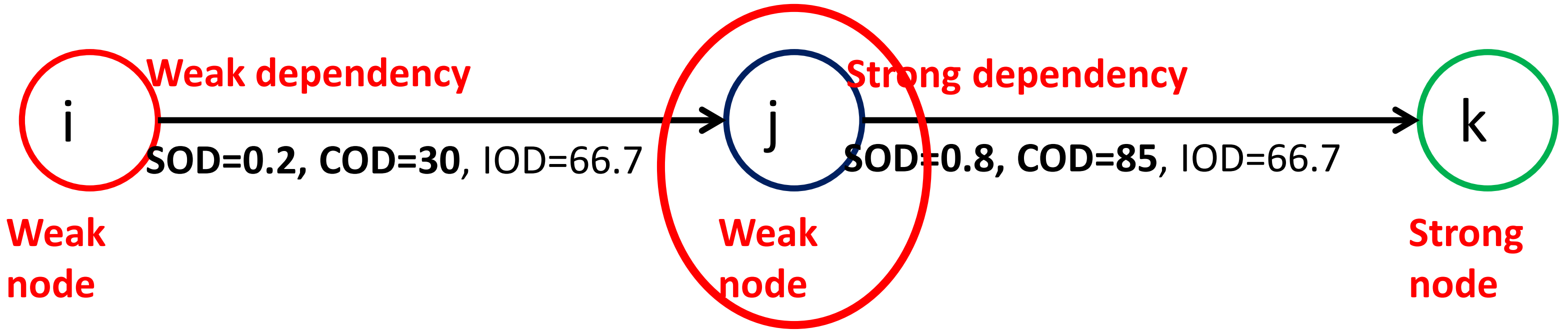


Slight further improvement

Histogram of instances having a given operability of system k (10000 total instances)

$E(O_k)$=69.2 $\sigma(O_k)$=11.5

How to "use" SODA parameters: knowing the parameters



Parameters of dependency **known**. (the probability distribution of the Self-Effectiveness of systems i and j is uniform between 0 and 100, and system k is working at max Self-Effectiveness)

The weakness of node j, its weak dependency from node i, and its strong influence to node k suggest that node j is **critical**. Therefore, one of the possible improvements that can be implemented is giving some redundancy to this node (this confirms the simulated results)

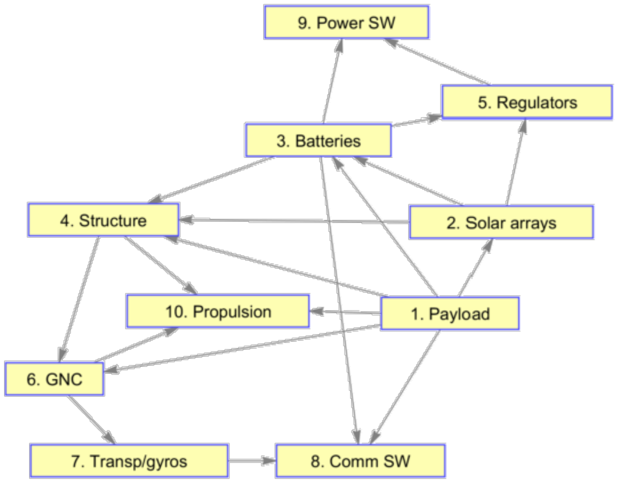
The parameters also suggest that improving the robustness of node i won't have a big impact, since its influence on node j is limited. Instead, other improvements may involve increasing the robustness of node j, or decreasing the dependency of node k from node j, if possible.

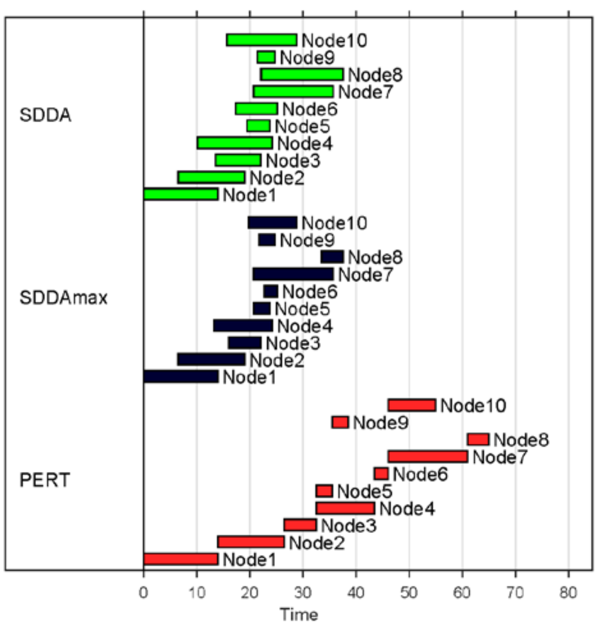
SYSTEMS ENGINEERING
Research Center

Backup slides - SDDA

SYSTEMS ENGINEERING
Research Center

Systems Developmental Dependency Analysis (SDDA)





Parametric model of developmental dependencies

- **Models parallel development and partial overlapping**
- **Scheduling and rescheduling based on delays and risks**
- **Educated decision for scheduling policies**
- Gives information on the effect of stakeholder decisions
- Trade-off between development time, partial capabilities, flexibility

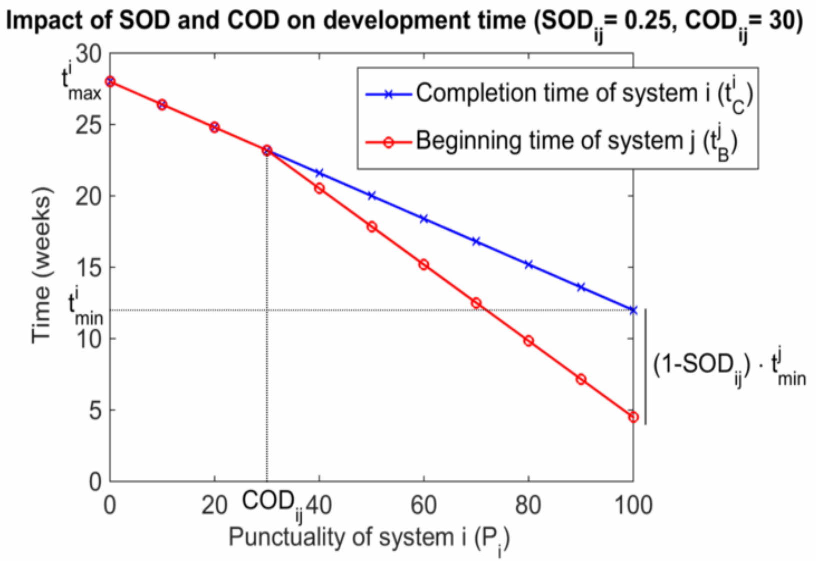
SDDA: partial dependency model

Parameters:

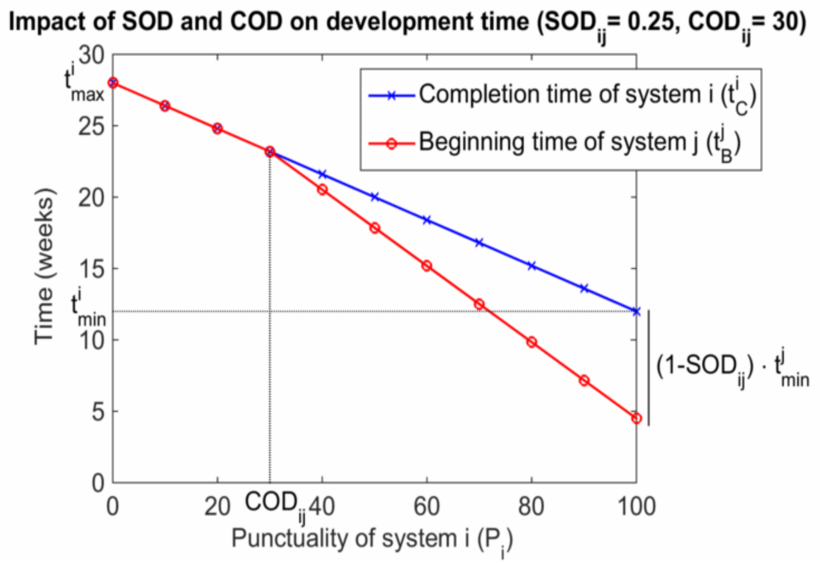
Strength of Dependency (SOD): how much is the amount of early development of the receiver system j that cannot be executed before a feeder system i is fully developed (the less the SOD, the less the development of system j depend on that of system i).

Criticality of Dependency (COD): what is the minimum operability of feeder system i that allows for early start in development of system j.

Parameters can come from historical data, expert judgment, or evaluation of amount of information required for development



SDDA: input/output model



Root nodes (beginning and completion time):

$$t_B^i = 0$$

$$t_C^i = t_{min}^i + \left(1 - \frac{P_i}{100}\right)\left(t_{max}^i - t_{min}^i\right)$$

Dependent nodes (development time):

$$t_D^j = t_{min}^j + \left(1 - \frac{P_j}{100}\right)\left(t_{max}^j - t_{min}^j\right)$$

Dependent nodes (beginning time):

${}^i t_B^j = t_C^i$ (if below criticality)

$${}^i t_B^j = t_C^i - t_{min}^j \left(1 - \alpha_{ij}\right) \frac{\left(P_i - \beta_{ij}\right)}{\left(100 - \beta_{ij}\right)}$$

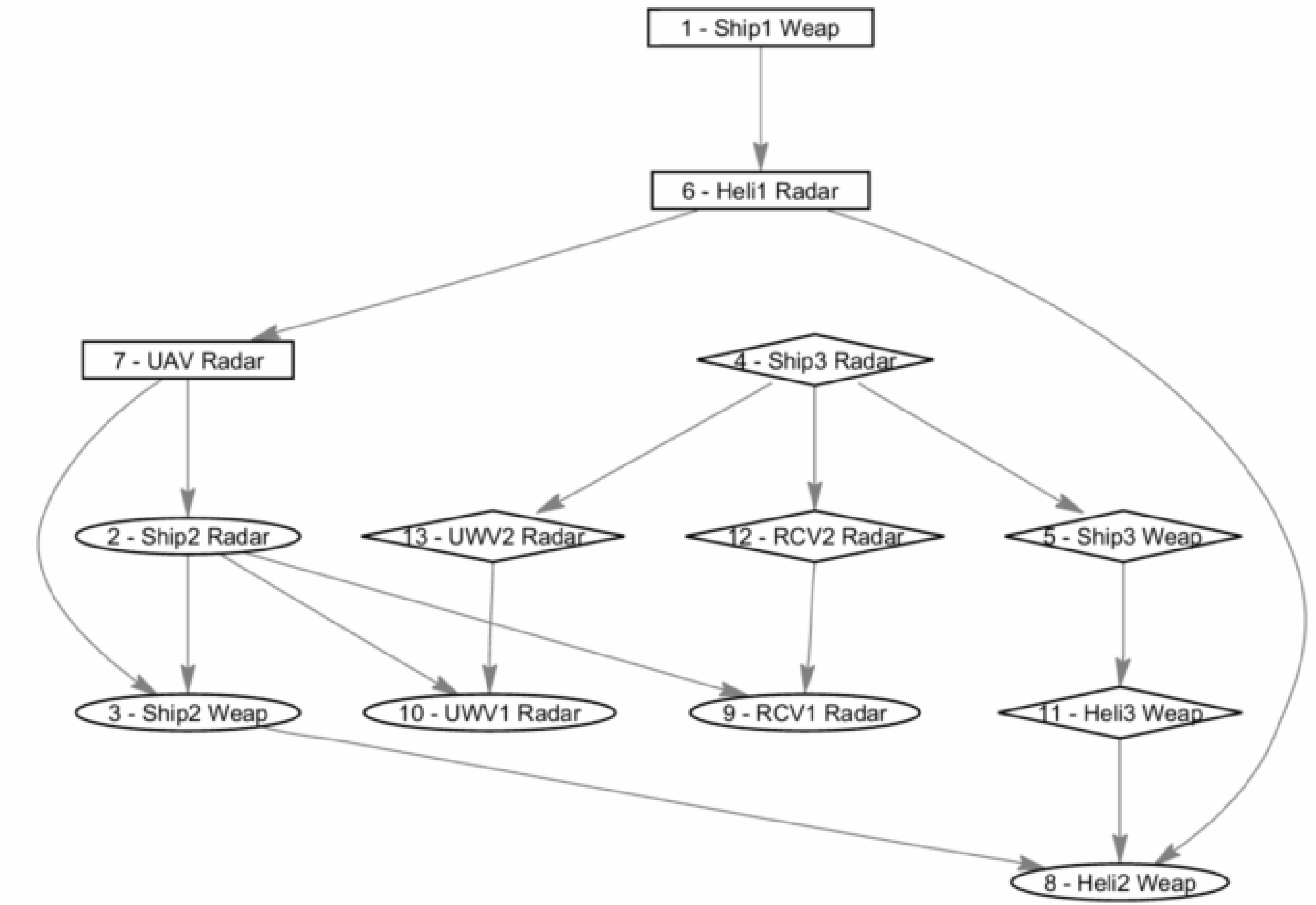
$$t_B^j = \frac{1}{n} \sum_{k=1}^{n} {}^k t_B^j$$

Dependent nodes (completion time):

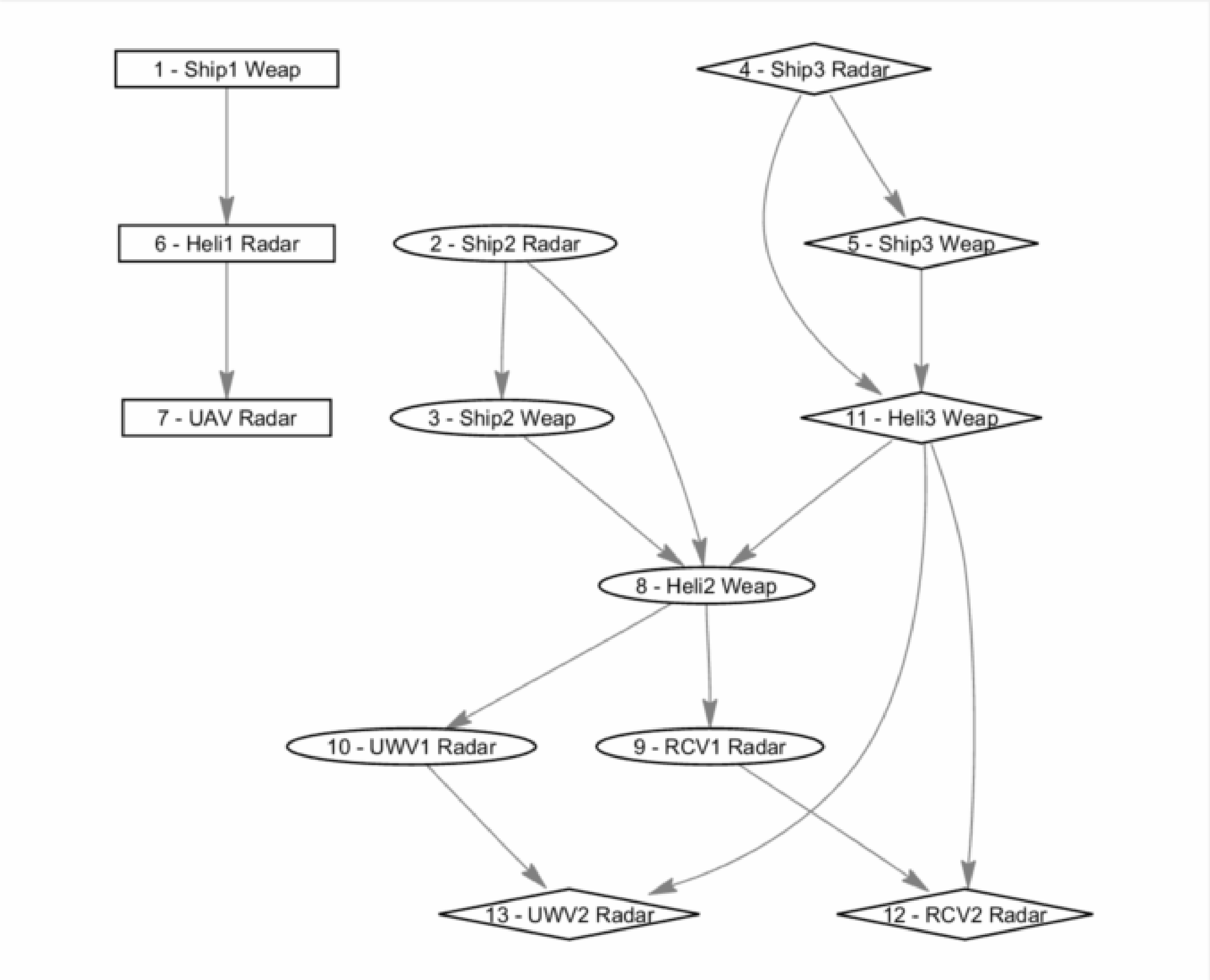
$${}^i t_C^j = \max\left(t_B^j + t_D^j, t_C^i + \alpha_{ij} t_{min}^j\right)$$

$$t_C^j = \max_n {}^n t_C^j$$

Architecture B of the NWS



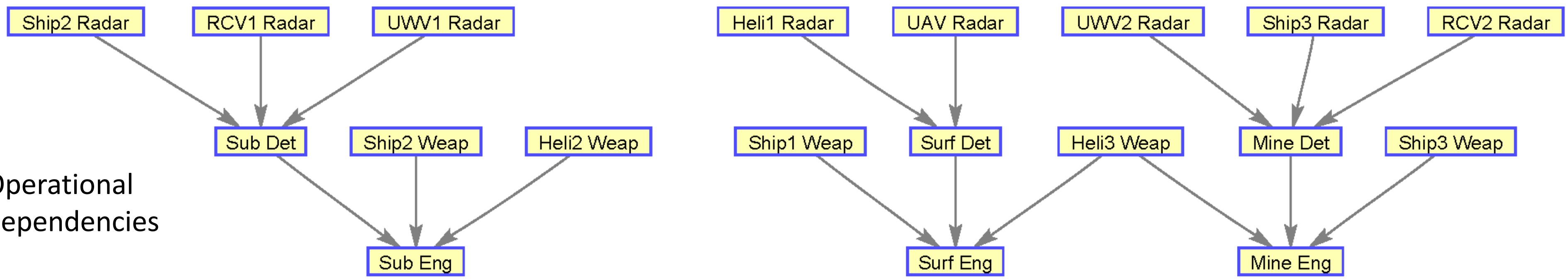
Architecture C of the NWS

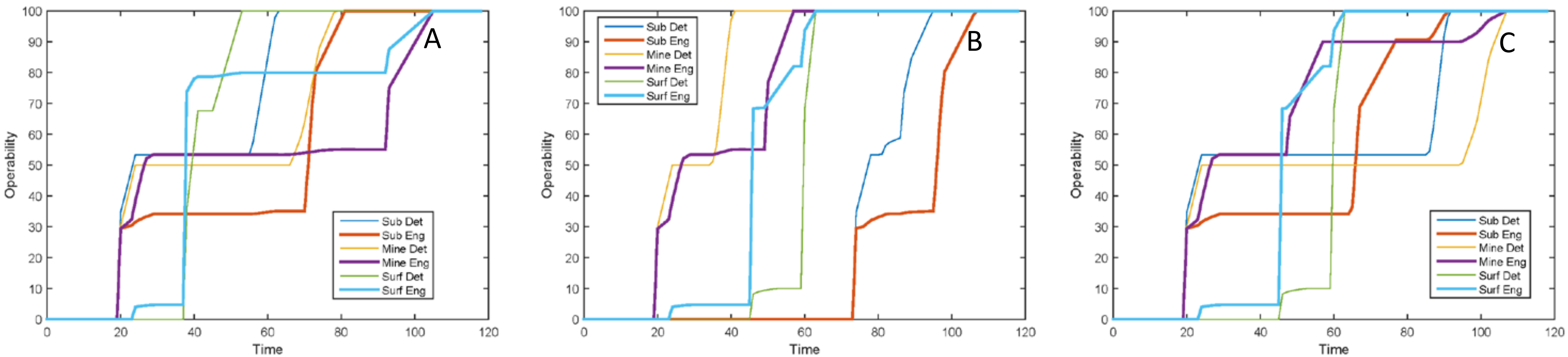


SYSTEMS ENGINEERING
Research Center

Operational dependencies in NWS

Operational dependencies





- A fastest to achieve partial and full capabilities in anti-sub systems
- B slower to achieve 50% capability in anti-sub systems, but fastest to achieve most full capabilities
- C as fast as B in development of surface systems, but slower in anti-mine
- **Different reaction to delays. Sometimes systems not affected by delays (e.g. anti-mine in B)**

Further results with SODA and SDDA

- Criticality of systems in the operational domain
- Robustness and resilience to failures. Comparison of alternate architectures
- Impact of managerial decisions in development stage (trade-off between risk, development time and cost)

Decision for t_B^i	Actual P_i	Review at $t=30$	Informed $E(t_C^{13})$	Actual $E(t_C^{13})$	% late inst.	Avg delay	% early inst.	Avg gain	min $\sum_{i=1}^{13} t_D^i$	Actual $\sum_{i=1}^{13} t_D^i$
Expected value	80	No	119.6	119.5	46.6	1.351	53.4	1.391	409.6	481.8
		Yes	119.6	119.6	44.5	1.063	55.5	1.198	409.6	481.8
	70	No	131.8	121.5	0	-	100	10.304	428.4	505.5
		Yes	131.8	125.4	0	-	100	6.173	428.4	489.1
	90	No	112.7	117.6	100	4.942	0	-	390.8	459.5
		Yes	112.7	117.2	100	4.454	0	-	390.8	479.6
10^{th} percentile (late start)	80	No	119.6	124.6	98.8	5.003	1.2	0.728	409.6	473.3
		Yes	119.6	121.8	92.7	2.29	7.3	0.607	409.6	484.2
	70	No	131.8	126.6	1.6	0.79	98.4	5.291	428.4	496.8
		Yes	131.8	128	0.7	0.463	99.3	3.609	428.4	492.2
	90	No	112.7	122.7	100	10.111	0	-	390.8	451
		Yes	112.7	119.1	100	6.372	0	-	390.8	481.7
90^{th} percentile (early start)	80	No	119.6	117	4.7	0.534	95.3	2.769	409.6	493.7
		Yes	119.6	117.5	5.3	0.562	94.7	2.512	409.6	479.3
	70	No	131.8	119.1	0	-	100	12.658	428.4	517.5
		Yes	131.8	122.8	0	-	100	8.83	428.4	485.9
	90	No	112.7	115.2	96.3	2.683	3.7	0.521	390.8	471.3
		Yes	112.7	115.3	95.6	2.642	4.4	0.489	390.8	477.4

SYSTEMS ENGINEERING
Research Center

Outcomes of SERC research – SODA and SDDA

- This research funded in part by the US Department of Defense through the Systems Engineering Research Center (SERC) RT-36, RT-44, RT-108, RT-134, RT-155
- Pilot applications and interaction with practitioners at the DoD, the MITRE corporation, the US Naval Surface Warfare Center Dahlgren Division, SANDIA National Laboratory, NASA Advanced Concept Office at Marshall Space Flight Center (MSFC)
- Research project with NASA MSFC through Jacobs Engineering carried on in 2015
- **Research project ongoing with NASA MSFC since July 2016**
- **Guariniello, Cesare, and Daniel DeLaurentis. "Supporting design via the System Operational Dependency Analysis methodology." *Research in Engineering Design* (2016): 1-17, DOI: 10.1007/s00163-016-0229-0**
- **Paper "Systems Developmental Dependency Analysis for Schedule and Decision Support" ready for submission to Design Science Journal**
- Journal paper about SODA application to cybersecurity in work.
- Journal paper about combined use of SODA and Robust Portfolio Optimization in work
- **Eight peer-reviewed conference papers at the Conference on Systems Engineering Research, the International Astronautical Congress, the AIAA Space Conference**

Backup slides - SIMs

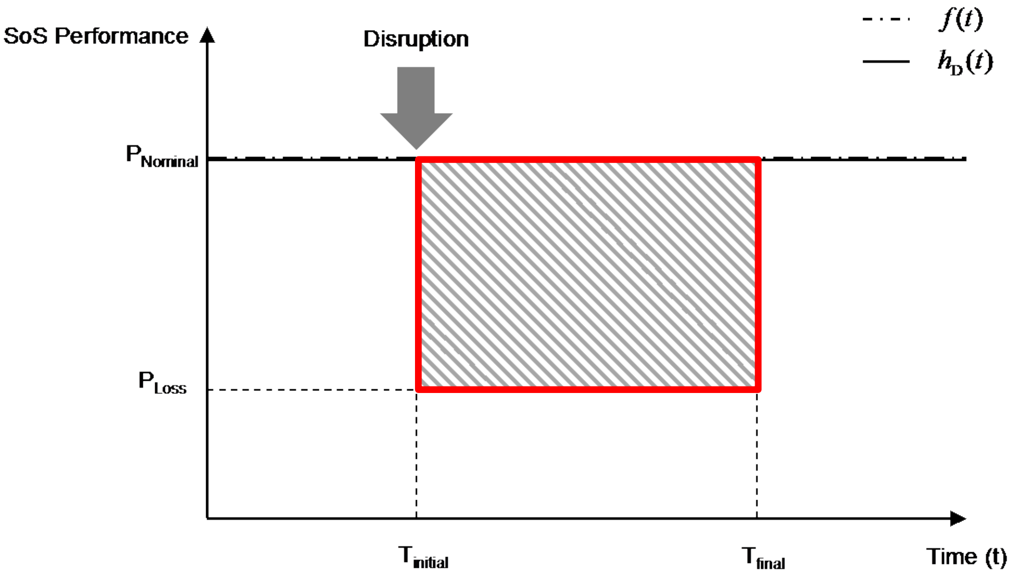
System Disruption Importance

- What is the impact of an unmitigated disruption on the SoS?

$$\text{Impact}_D = \int_{T_{\text{initial}}}^{T_{\text{final}}} f(t) - h_D(t)$$

- How important is an unmitigated disruption relative to other disruptions?

$$SDI_D = \frac{\text{Impact}_D}{\text{Worst-case SoS impact}}$$

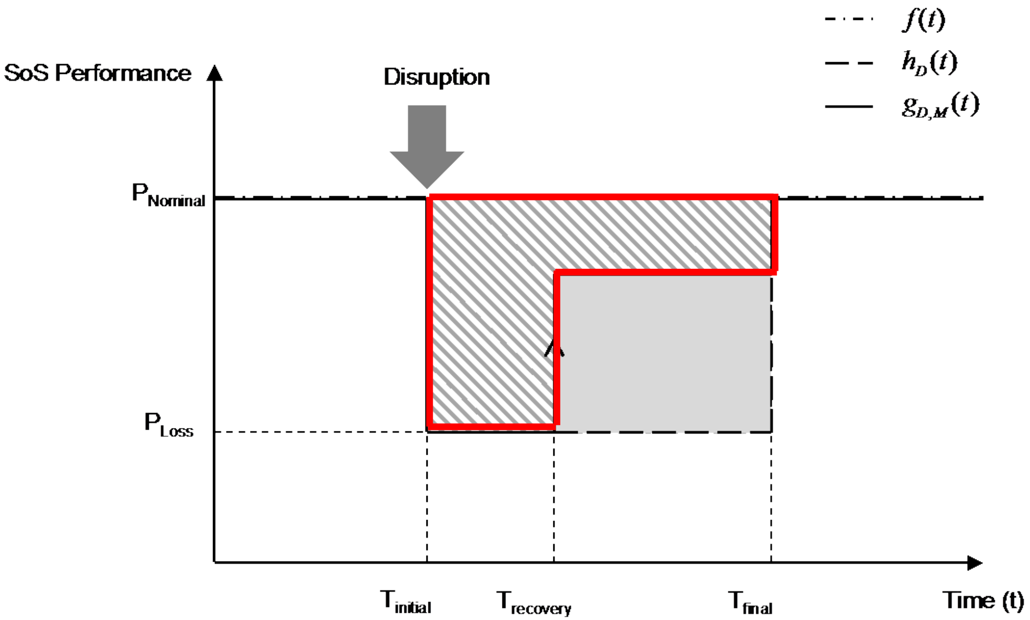


System Disruption Conditional Importance:

- How important is a disruption given that its impact is mitigated?

$$SDCI_{D,M} = \frac{\int_{T_{\text{initial}}}^{T_{\text{final}}} f(t) - g_{D,M}(t)}{\text{Worst-case SoS impact}}$$

- When mitigation is not possible, $SDCI_{D,M}$ is undefined

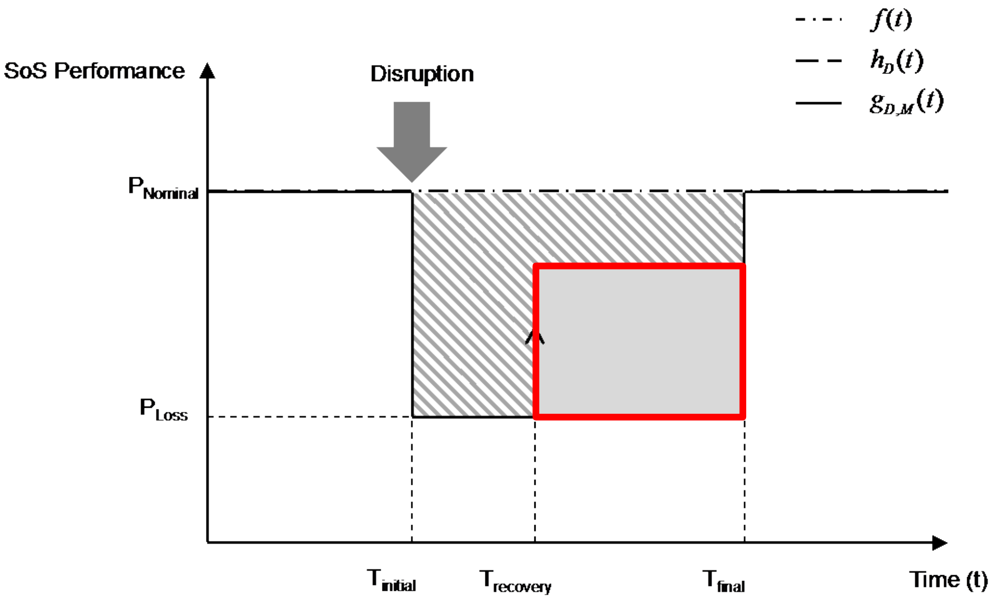


System Disruption Mitigation Importance:

- How effective is a mitigation measure?
- When mitigation is not possible, $SDMI_{D,M}$ is undefined

$$SDMI_{D,M} = \frac{\int_{T_{initial}}^{T_{final}} g_{D,M}(t) - h_D(t)}{\text{Worst-case SoS impact}}$$

$$\boxed{SDMI_{D,M} = SDI_D - SDCI_{D,M}}$$



RPO Backup Slides

General Optimization Problem

Objective

Maximize Performance Index

$$\max \left(\sum_q \left(\frac{S_{qc} - R_c}{R_c} \cdot w \cdot X_q^B \right) \right)$$

Capability Weights

Binary decision

Reference

s.t.

Constraints

Requirement Satisfaction

$$\sum_q S_{qC} X_q^B \geq \sum_q S_{qR} X_q^B \text{ (Satisfy Requirements)}$$

Big-M Formulation (number of connections)

$$\sum_c X_{cij} - X_{ij} M \leq 0$$

$$M \sum_c X_{cij} - X_{ij} \geq 0$$

Flow Balance Constraint

$$\sum_i X_{cij} - \sum_j X_{cij} - X_j^B S_{rj} = 0$$

Bandwidth Limit

$$X_{cij} \leq \text{Limit}_{cij}$$

Node Connection Compatibility

$$X_1^B + \ldots + X_n^B = \text{D (System Compatibility)}$$

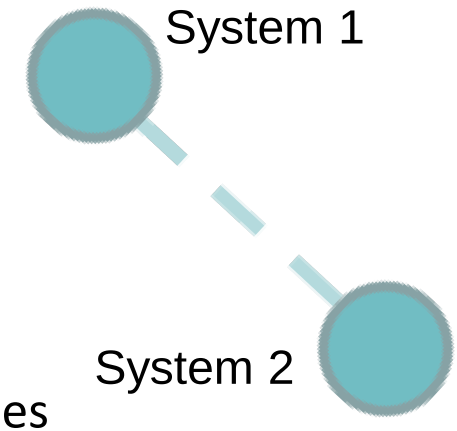
$$\text{(e.g. } X_1^B + X_1^B + X_1^B = 1\text{)}$$

$$X_{cij} = 0 \quad \text{i,j} \in \{\text{incompatible set}\}$$

$$\in \{0,1\} \text{ (binary)}$$

Dealing with Uncertainty

- Entities
  - —**System Capability**: Actual performance of system individually and as a whole SoS entity
  - —**System Interdependence:** Interdependencies between systems and effects on translation of capability uncertainties

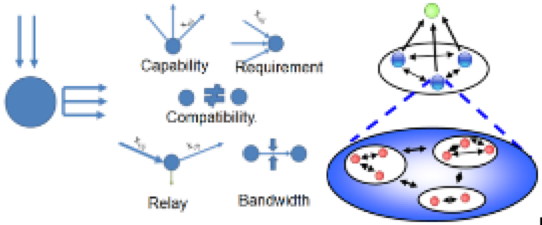


- Addressing data uncertainty in portfolio selection
  - Uncertainties in node (system) performance and connections (links)
  - Capture variation in performance at each node as **uncertainty sets.**
  - Variations/uncertainty bounds from ABM simulation or design choice.

SYSTEMS ENGINEERING Research Center

Robust Operational Constraints

- Use Bertsimas-Sim approach to uncertain (data uncertainty) constraints
- Benefits: Linear Programming approach, constraint violation control with probabilistic guarantees, extends to discrete optimization



Constraint Rules for Connectivity & Operations

$$
\begin{aligned}
&\sum_i X_{cij} \geq X_j^B S_{rj} \\
&\sum_i X_{cij} \geq X_j^B S_{rj} \\
&X_1 + \cdots + X_n = 0 \\
&\sum_c X_{cij} - X_{ij} M \leq 0 \\
&M \sum_c X_{cij} - X_{ij} \geq 0 \\
&\sum_i X_{cij} - \sum_j X_{cij} - X_j^B S_{rj} = 0
\end{aligned}
$$

$$[A]\{X_q\} \leq \{b\}$$

Adjust conservatism Γ_i term to control probability of **constraint violation**

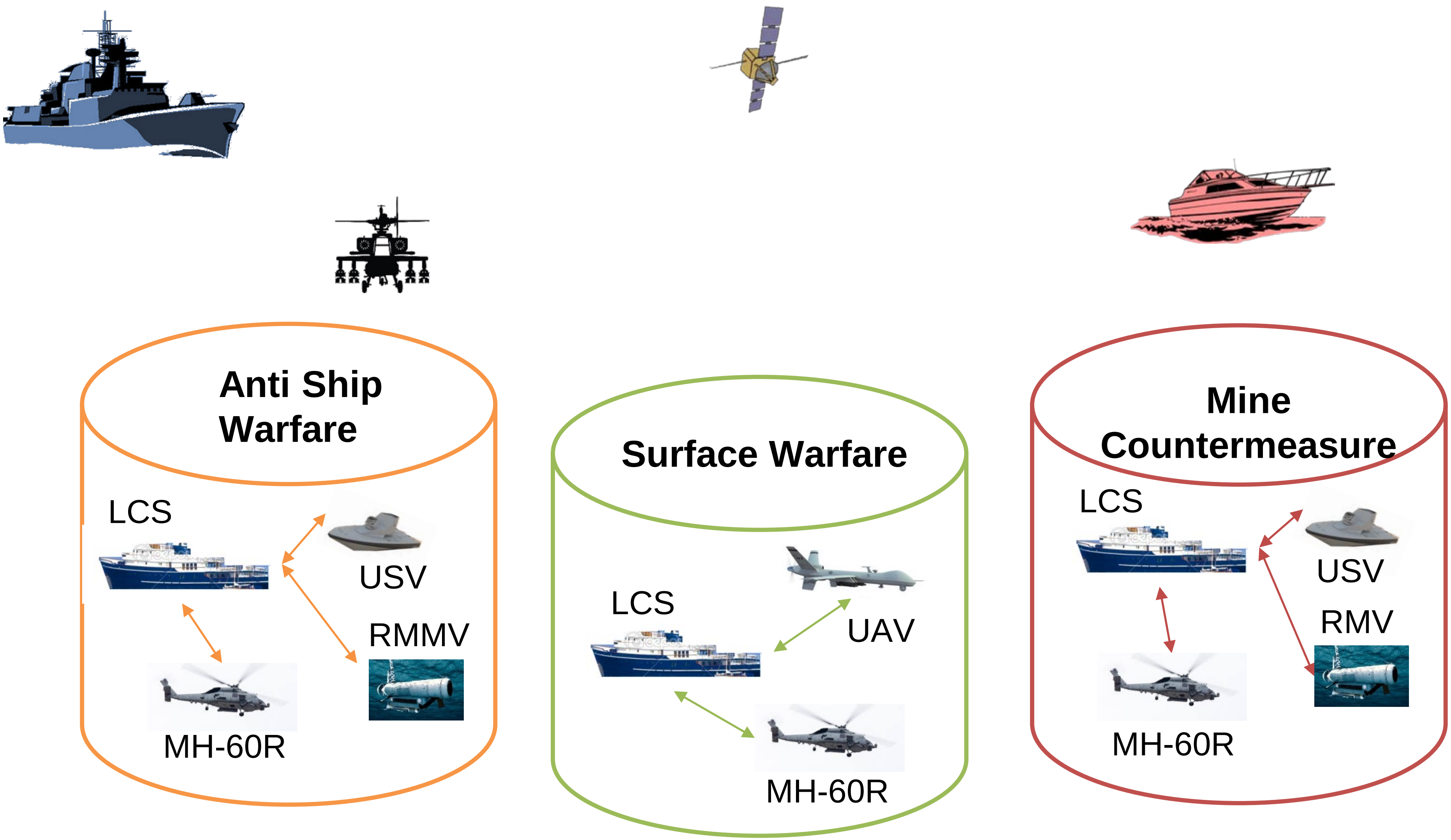
$$\sum_q S_{qC} X_q^B + \max\left\{\hat{S}_{qC} y_j + \left(\Gamma_i - \lfloor \Gamma_i \rfloor \hat{S}_{it_i} y_t\right)\right\} \leq b_i$$

$$-y_j \leq X_q^B \leq y_j$$

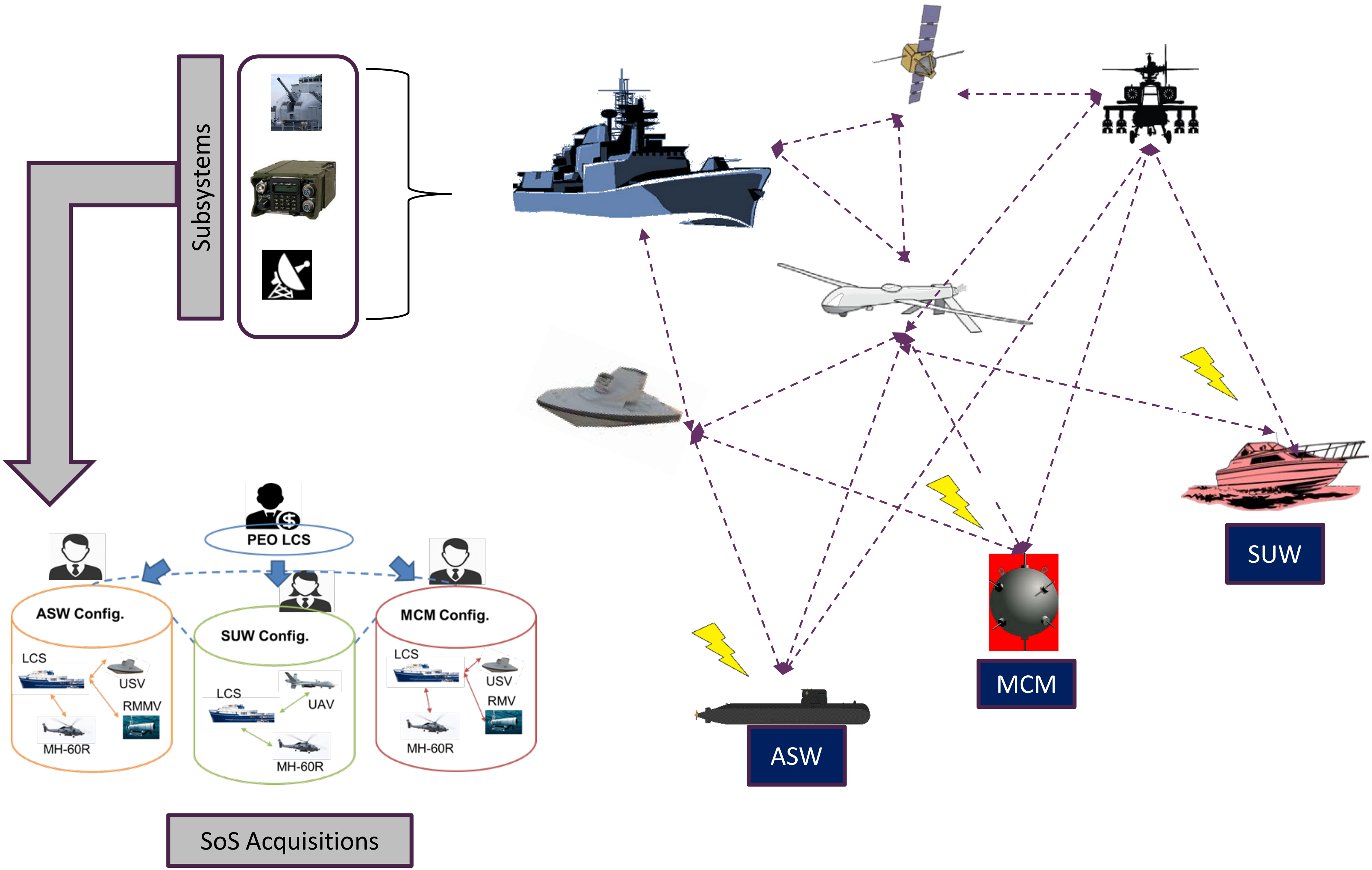
$$y \geq 0$$

Conservatism Added
(This can be converted to an LP == easy to solve even for large problems)

Concept Naval Scenario



Concept Naval Warfare Scenario

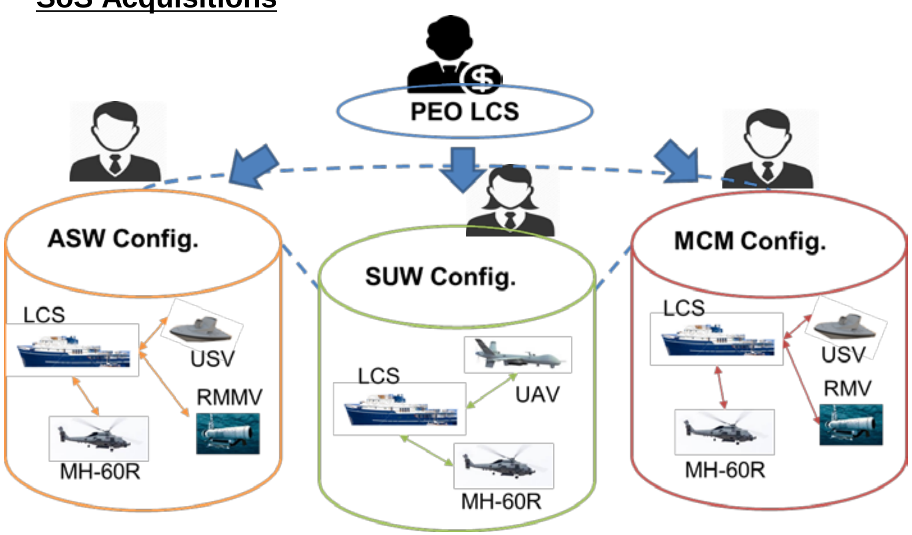


Concept Naval Warfare Scenario

Military Systems (Assets)

Mission Scenarios

SoS Acquisitions



Assessing Solutions (Cost, Performance, Risk, Resilience etc.)