JOHNS HOPKINS
APPLIED PHYSICS LABORATORY

Quantifying Schedule Uncertainty in Acquisition

A Bayesian Network Approach

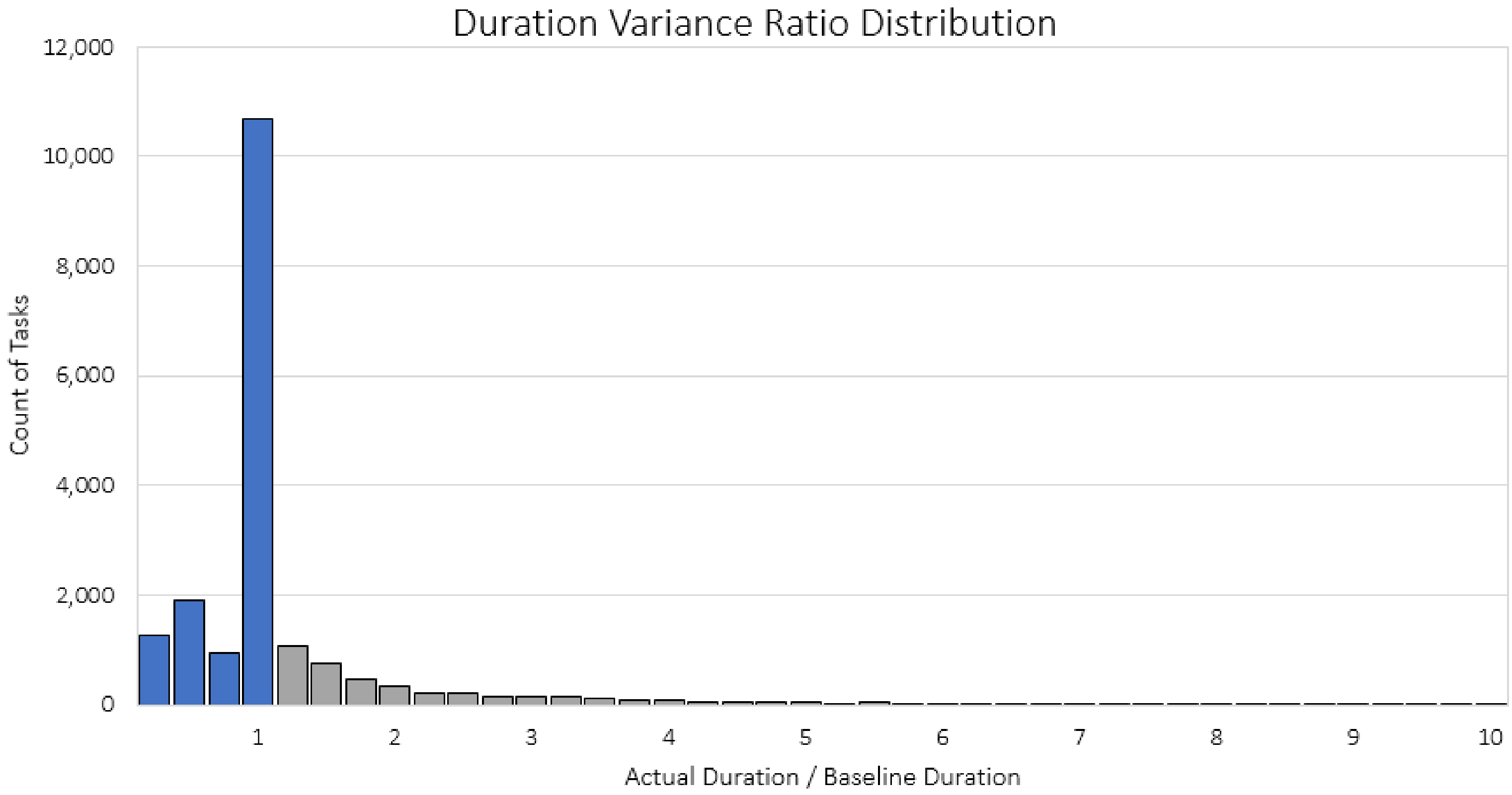
Joshua Hamilton
Affordability Analyst/Data Scientist
Johns Hopkins University – Applied Physics Lab

Outline

- Problem Statement
- Background
  - Traditional Methods
  - Previous Work
- Research Objectives
- Bayesian Reasoning: The Basics
- Model Structure
- Methodology
- Case Studies
- Conclusion
- Future Work

Problem Statement

- In many major DoD acquisition programs, most tasks appear to perform well against their individual baselines
- Yet, programs still face **persistent and significant schedule delays**
- Fundamental question: **If tasks are on time, why are programs late?**



Background

- Major Development Programs Routinely Encounter Acquisition Problems
- GAO 2020 Findings
  - 2020 GAO Report[1] noted 85 major defense acquisition programs (MDAPs) cumulatively overran their budget baseline by **$638 billion**
  - 2024 GAO Report[2] reported that average time for weapon systems to reach initial operating capability (IOC) increased to 3 years, from 8 to 11 years

85
MDAPs analyzed

$638B
Total cost growth

3 years
Avg. IOC delay

- Underlying Challenge
  - Modern programs are complex and highly interdependent
  - Traditional tools assume **task-level independence** and **fixed critical paths**
  - These methods miss **cascading risks** and **non-linear delay accumulation**
- Why This Matters
  - Delays impact warfighter readiness, cost efficiency, and strategic agility
  - Understanding how and why delays accumulate is critical for future program success

We need better tools to quantify how delay propagates across dependent tasks

1. GAO-20-439; 2. GAO-24-106831

Traditional Methods Overview

Method	Strength	Assumptions	Key Limitation
Critical Path Method (CPM)	Clear critical path identification	• Deterministic durations • No uncertainty	Ignores uncertainty entirely
Program Evaluation Review Technique (PERT)	Adds probabilistic task durations	• Task independence • Beta distribution	Does not account for alternate critical paths or dependencies
Monte Carlo Simulation (MCS)	Generates probabilistic outcomes	• Independent task duration inputs	Fails to model causal relationships between tasks
Critical Chain Project Management (CCPM)	Focuses on buffers to manage slippage	• Task overestimation • Single buffer concept	Ignores structural causes of delay propagation

Traditional methods are not equipped to model interdependent delays or update dynamically as new data emerges

Previous Work: Fragile Task Identification

What We Did

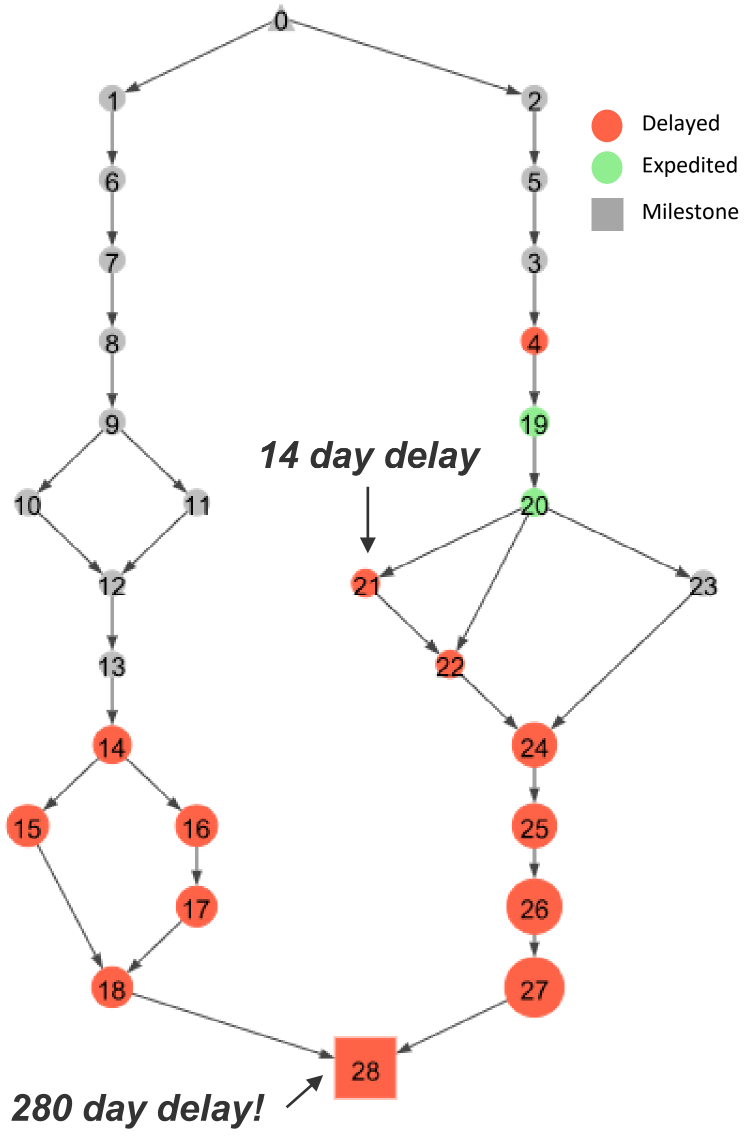
- Applied network analysis to a major acquisition program schedule
- Mapped task dependencies to display network structure
- Identified statistically significant metrics (e.g., in-degree, out-degree, reachability) that signal fragile tasks

What We Found

- Tasks with **high in-degree** (many predecessors) and **high out-degree** (many successors) are **structurally fragile**
- Structural fragility = Tasks with delays lead to a subsequent cascade of increasing delays across multiple generations of successors

Missing Link

- Previous work identified *where* delay risk exists
- Next step: Quantify *how much* delay fragile tasks introduce



Research Objective

✅ Objectives

- Estimate the **magnitude of delay** introduced by high-risk fragile tasks
- Generate **probabilistic forecasts** of milestone and program-level completion dates
- Enable **dynamic updates to delay** forecasts as new task status data becomes available

Key Research Questions

⚠️ **Risk Drivers**

Which tasks pose the greatest risk of cascading delay?

🎯 **Delay Magnitude**

How much could a fragile task delay a milestone – or the whole program?

📈 **Forecast Accuracy**

How can we produce more reliable and dynamic schedule forecasts than traditional tools?

Bayesian Reasoning – A Primer

- Bayes' Theorem

$$P(H \mid E) = \frac{P(E \mid H) * P(H)}{P(E)} \quad \text{where}$$

H: Hypothesis (e.g., a sub-contractor delivers on time)

E: Evidence (e.g., a task is late)

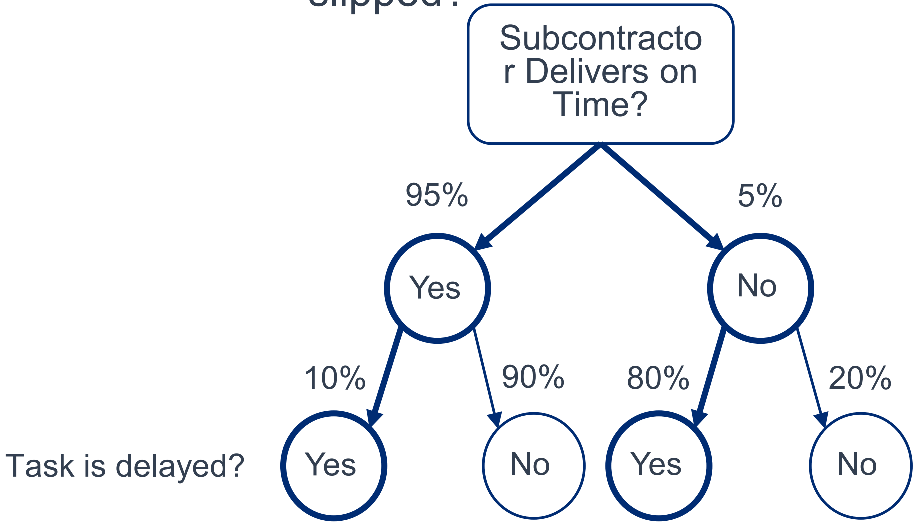
P(H): Prior belief before seeing evidence

P(E): Probability of witnessing the new evidence under all hypotheses

P(E | H): Likelihood of seeing evidence if the hypothesis is true

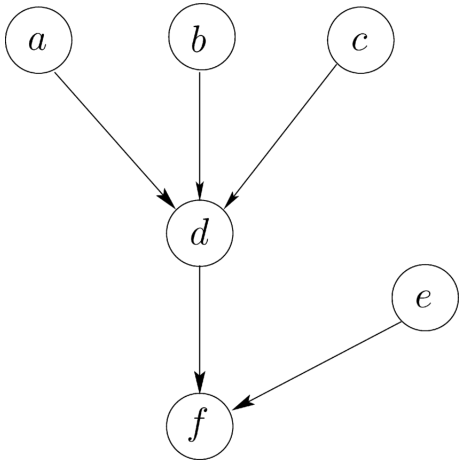
P(H | E): Updated belief after observing the evidence (Posterior)

- Bayesian Inference
  - Used to **revise beliefs** based on new observations
  - Example: A subcontractor usually delivers on time, but a delay occurs. What is the chance the sub actually slipped?



- $S =$ Delivers on time $\quad \overline{S} =$ Failed to deliver on time
- $D =$ Task is delayed $\quad \overline{D} =$ Task is not delayed
- $P(S) = 0.95 \rightarrow P(\overline{S}) = 0.05$
- $P(D \mid S) = 0.10 \rightarrow P(\overline{D} \mid S) = 0.90$
- $P(D \mid \overline{S}) = 0.80 \rightarrow P(\overline{D} \mid \overline{S}) = 0.20$
- $P(D) = P(S) * P(D \mid S) + P(\overline{S}) * P(D \mid \overline{S}) = 0.135$
- $P(S \mid D) = \frac{P(D \mid S) * P(S)}{P(D)} = \frac{0.10 * 0.95}{0.135} = 0.704$
- Posterior: Updated belief = ~70% chance the sub delivers on time

Why Bayesian Networks?



🔍 BNs Employ Several Desirable Statistically Modeling Attributes

- **Visual Structure:** Maps dependencies across tasks
- **Probabilistic Logic:** Handles uncertainty explicitly
- **Causal Modeling:** Models how delays propagate from task to task

🔄 Dynamic & Scalable

- BNs scale Bayes' logic across many variables
- Can be **updated in real-time** as new information becomes available
- Helps predict future delays and the underlying drivers of the forecasted delays, so that proactive mitigations can be administered

VS Why BNs Outperform Traditional Methods

- Handles interdependencies unlike CPM or PERT
- Incorporates uncertainty, unlike CPM
- Learns from new data, unlike MCS or CCPM

BN Forecasts Support Tactical to Strategic Planning

Schedule Level	WBS/OBS Level	What It Tells You	Example Output
Task	WBS Level 4+, Work Packages	Likelihood and timing of individual task delays	"Task A has a 40% chance of exceeding planned duration"
Milestone	WBS Level 3, Control Accounts	Forecasted delay and confidence levels	"Milestone X has a 70% chance of being > 15 days late"
Program	WBS Level 1-2, Program Office	Overall delivery timeline distribution	"The program has an 80% chance of completing by FY30 Q3"

Bayesian Networks generate actionable forecasts aligned to the WBS and OBS – from daily task execution to program delivery

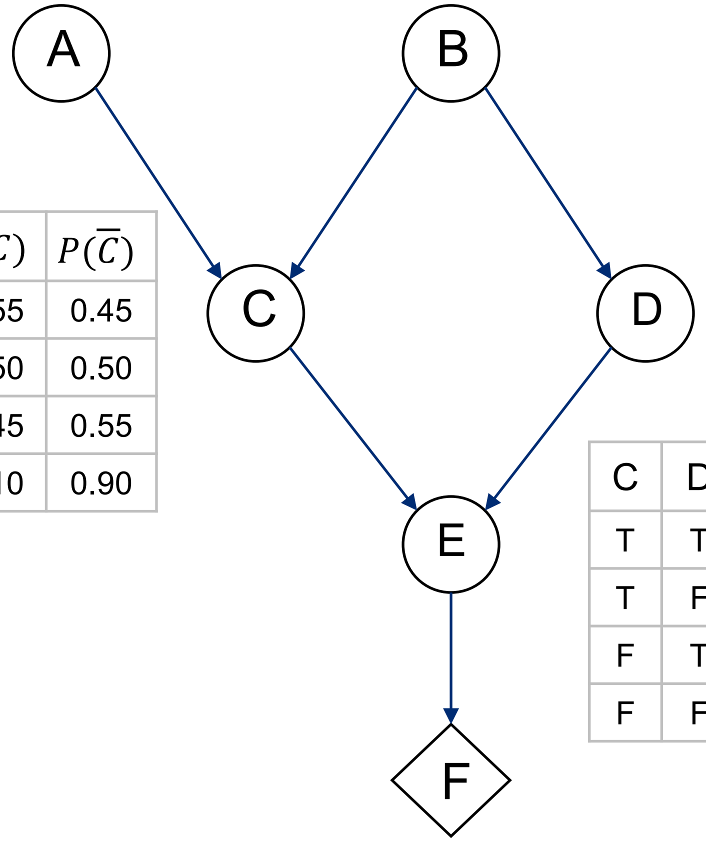
BN Model Structure

- Bayesian Network in Schedule Context
  - A graphical model representing the program's task schedule
  - Nodes = Activities or Milestones
  - Edges = Tasks relationships (e.g., task C is a successor to task A)
  - Each node has a Conditional Probability Table (CPT) that reflects how its duration depends on its parent nodes

- What It Models
  - Task duration uncertainty using probability distributions (triangular, beta, etc.)
  - Dependency-driven delay propagation through the network
  - Criticality determined by how a node's delay affects successors

$P(A)$	$P(\bar{A})$
0.20	0.80

$P(B)$	$P(\bar{B})$
0.40	0.60



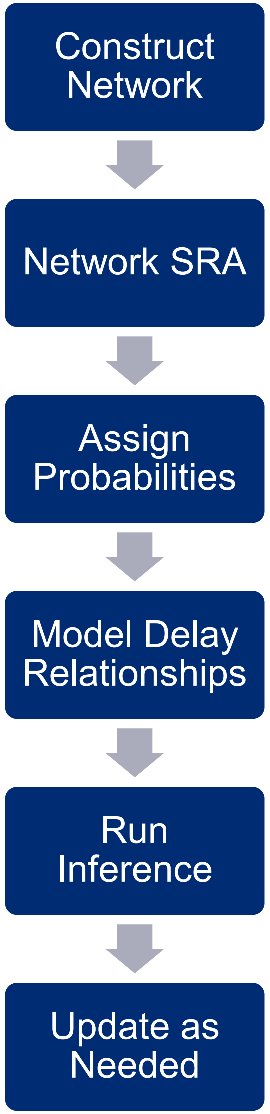
A	B	$P(C)$	$P(\bar{C})$
T	T	0.55	0.45
T	F	0.50	0.50
F	T	0.45	0.55
F	F	0.10	0.90

B	$P(D)$	$P(\bar{D})$
T	0.60	0.40
F	0.55	0.45

C	D	$P(E)$	$P(\bar{E})$
T	T	0.70	0.30
T	F	0.50	0.50
F	T	0.45	0.55
F	F	0.20	0.80

E	$P(F)$	$P(\bar{F})$
T	0.70	0.30
F	0.20	0.80

Methodology



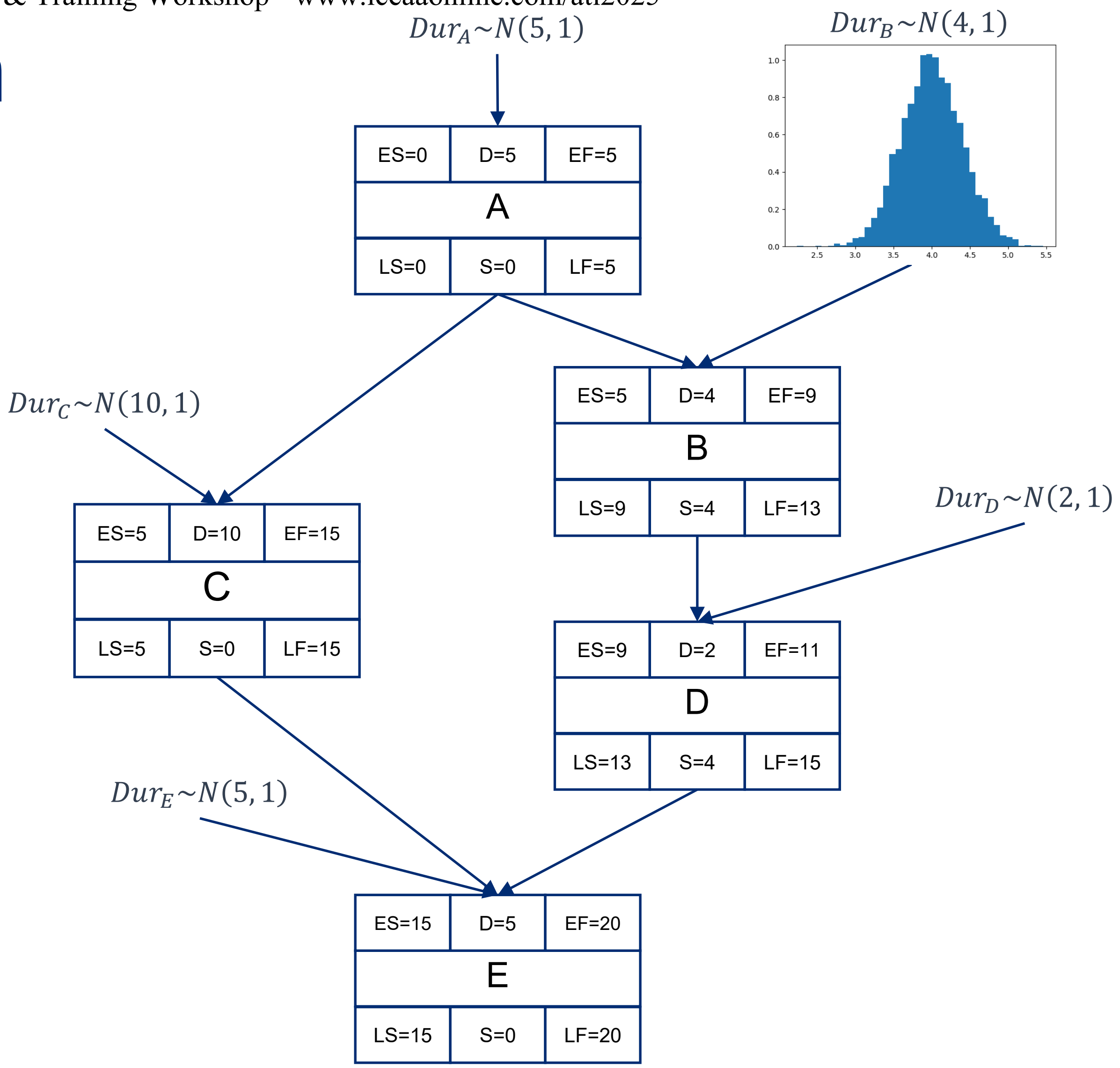
1. Construct the Network
   - Define the nodes (tasks/milestones)
   - Establish predecessor-successor relationships (edges)
2. Networking Schedule Risk Assessment (SRA)
   - Select and filter tasks or milestones with high potential for schedule delay, using network analysis metrics (e.g., in-degree, out-degree)
3. Assign Probability Distributions
   - Define task durations using probability distributions (e.g., triangular, beta, Weibull, etc.)
   - Distributions are based on historical data or subject matter expert judgement
   - Additionally, probability distributions account for high risk tasks (higher variance)
4. Model Conditional Delay Relationships
   - Run Monte Carlo simulation using CPM to generate task or milestone delay distributions
   - Use these outcomes to model how delays propagate through the network
5. Run Inference
   - Forward reasoning: estimate the likelihood of delay at milestones or program completion
   - Backward reasoning: trace likely causes of observed or forecasted delay
6. Update Dynamically
   - As new status data becomes available, update task durations to actuals and adjust distributions
   - Re-run inference to generate updated predictions

Case Study #1 – Definition

Simple Project[3]

- 5 Activities (A, B, C, D, E)
- Durations:
  - A = 5 weeks
  - B = 4 weeks
  - C = 10 weeks
  - D = 2 weeks
  - E = 5 weeks

Node	Description	Node Probability Table
D	Duration	$N(\mu, \sigma)$
ES	Early Start	$Max[EF_j \mid j\ predecessors]$
EF	Early Finish	$ES + D$
LS	Late Start	$LF - D$
LF	Late Finish	$Min[LS_j \mid j\ successors]$

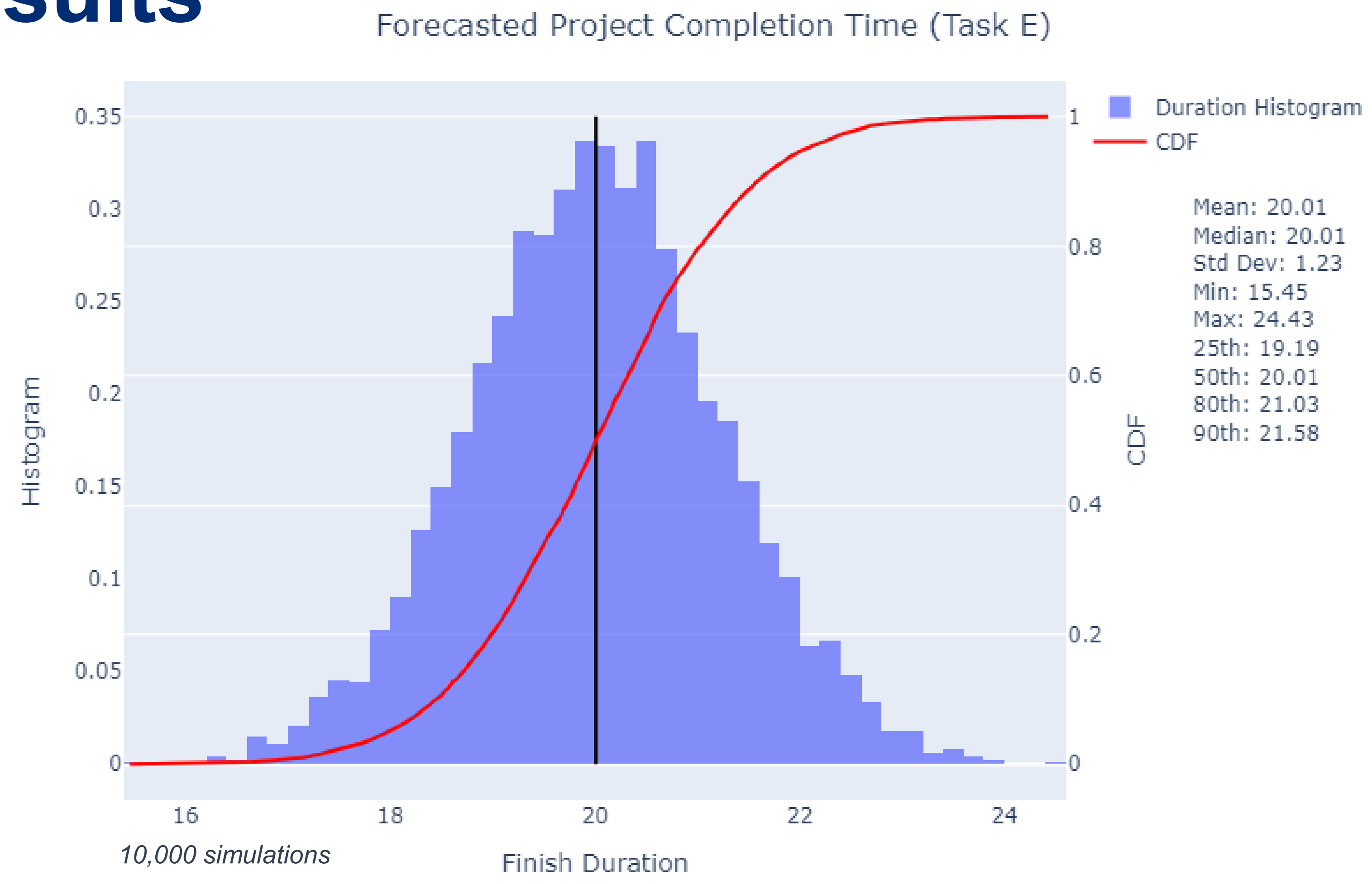


3. Khodakarami, V. (2009)

Case Study #1 – Results

- Key Results
  - Mean Duration: 20.01 weeks
  - Standard Deviation: 1.23 weeks
  - 90% Confidence Interval: [15.45, 21.58]
  - 80^{th} Percentile: 21.03 weeks

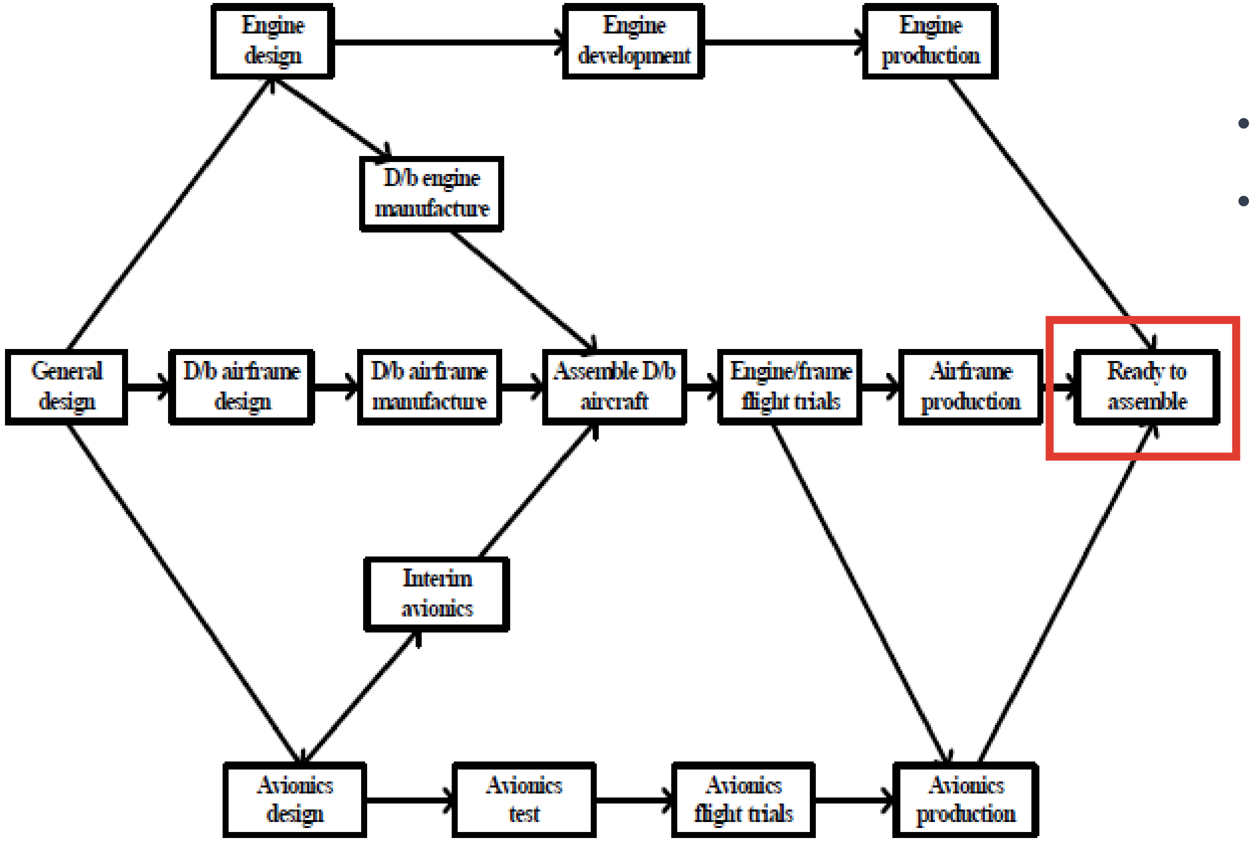
- Recommended Actions
  - Add contingency time to Task E
  - Communicate potential delay range to stakeholders
  - Reevaluate schedule and project assumptions



10,000 simulations

There's a 10% chance the project extends beyond 21.6 weeks – a risk window that deterministic CPM alone would not reveal

Case Study #2 – Aircraft Development Program



- Derived from a UK Ministry of Defense (MoD) aircraft development program with durations in months[4]
- 14 interdependent activities (airframe, engine, avionics)
- Objective: Assess schedule risk for Ready to Assemble milestone

Activity	Distribution	Triangular Distribution		
		Min	Mode	Max
General Design	Triangular	4	10	21
Engine Design	Triangular	21	32	55
Avionics Design	Triangular	1	17	19
D/b Airframe Design	Triangular	5	15	32
D/b Engine Manufacture	Triangular	7	9	11
Interim Avionics	Triangular	7	14	27
D/b Airframe Manufacture	Triangular	8	11	17
Assemble D/b Aircraft	Triangular	3	5	10
Engine Development	Triangular	20	23	40
Engine Production	Triangular	12	13	14
Avionics Test	Gamma	mean = 10, mode = 5		
Avionics Flight Trials	Discrete	Relative Probability 1:2:1:1 of 4, 5, 6, 24		
Engine/Frame Flight Trials	Discrete	Relative Probability 1:2:2:1:0.5 of 5, 6, 7, 8, 13		
Airframe Production	Triangular	12	14	18
Avionics Production	Triangular	14	16	24

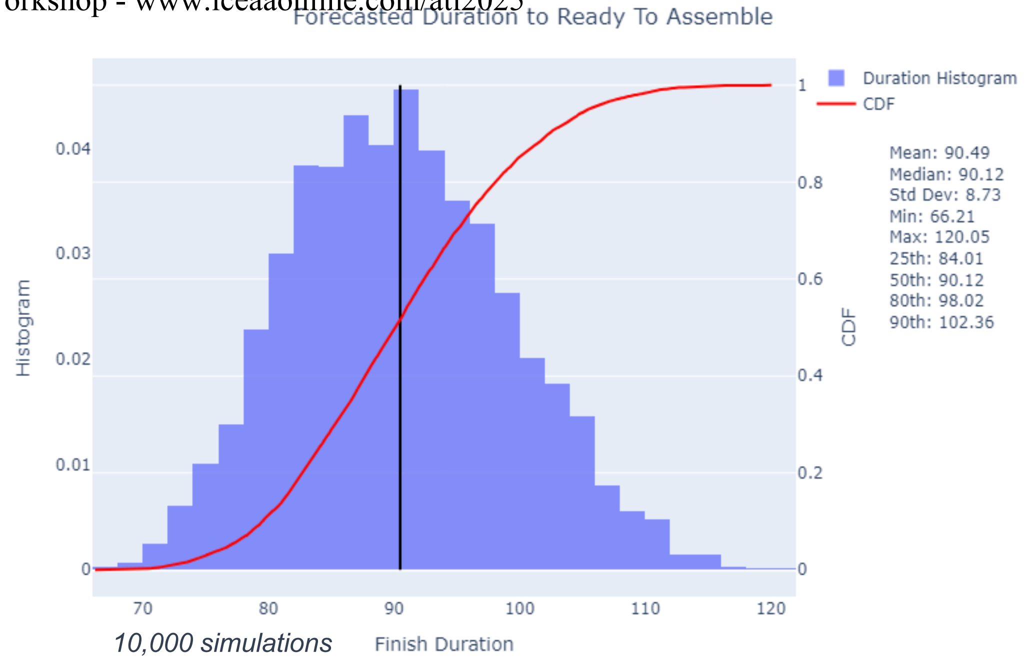
4. Bowers, J. (1994) & Williams, T. (2004)

Case Study #2 – Delay Risk & Critical Path Drivers for Ready to Assemble

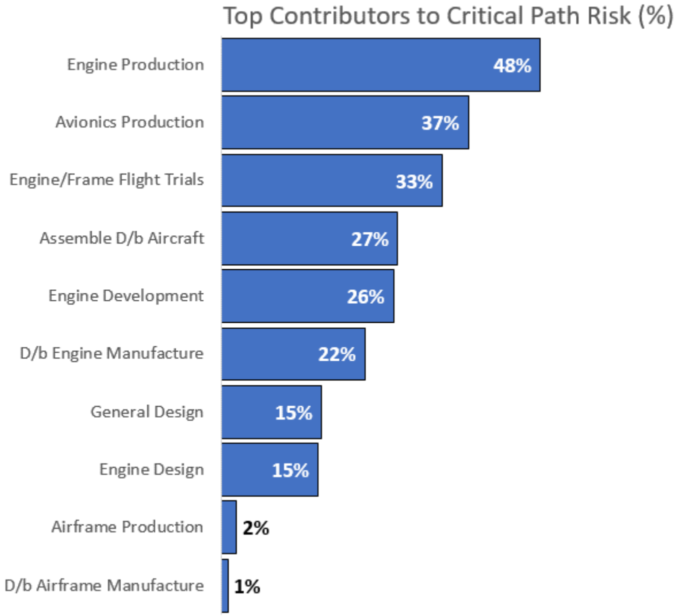
- Key Results for Ready to Assemble
  - Mean Duration: 90.49 months
  - Standard Deviation: 8.73 months
  - 90% Confidence Interval: [66.21, 102.36]
  - 80th Percentile: 98.02 months

- Recommended Actions
  - Prioritize monitoring of high-frequency drivers: Engine & Avionics Production
  - Share confidence intervals with leadership to support contingency planning

Multiple high-frequency critical paths across engine and avionics subsystems drive upper-tail schedule risk — a dynamic risk profile that deterministic CPM cannot capture



10,000 simulations



Key Takeaways

- BN + Monte Carlo + CPM = a dynamic, risk-aware forecasting method
  - Models uncertainty in task durations
  - Captures interdependencies and cascading risk
  - Produces probability-based forecasts at task, milestone, and program levels

- From the Case Studies:
  - Even simplistic networks hide delay risks not visible by CPM
  - Complex programs often have **multiple risk contributors** – not just a single “critical path”

- Traditional Tools
  - BN models reveal **long-tail risks** and **structural delay propagation** that deterministic tools overlook
  - However, BN models are far more complex and require more time, effort and software considerations than traditional approaches

This approach supports better planning, earlier intervention, and more informed schedule risk decisions

Future Work

🛠 Enhancing the Model

- Incorporate both **known risks** (e.g., supplier delays, resource allocations, technical complexity) and **unknown risks** into task duration distributions
- Explore additional distribution types (e.g., beta-PERT, log-normal) to better match real-world uncertainty
- Utilize schedule risk assessments for insight into task durations and risks
- Introduce correlation structures between tasks (shared risks or resources, schedule constraints)

📈 Advancing Forecast Utility

- Generate early warning indicators for milestone slippage
- Develop decision-support tools for resource allocation or replanning
- Integrate BN outputs with broader program data streams (cost, performance)

Future development will focus on realism, scalability, and integration – turning insight into impact

JOHNS HOPKINS
APPLIED PHYSICS LABORATORY