Multi-Mission Autonomy Beyond Space

A Decision Framework for Avionics Systems

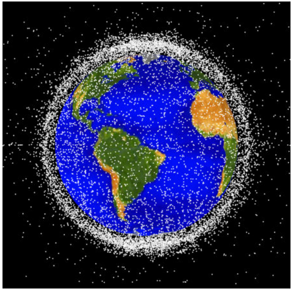
Toward Embedded Mission Autonomy for Avionics-Integrated UxS

Suwon Lee | Department of Future Mobility, Kookmin University
suwon.lee@kookmin.ac.kr | GE Aerospace — KMU Seminar

April 22, 2026

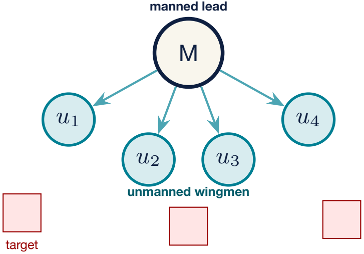
Two Domains, One Problem

Space and aviation share the same planning challenge



Space Operations

congested orbits, limited servicers,
multi-mission demand (debris / service / observation)



Aviation Operations

heterogeneous teams, contested airspace,
diverse task types (ISR / strike / SEAD / EW)

Shared Core Challenge — which targets, in what sequence, under what constraints?

The Orbital Environment

Space domain challenges

- ∼30,000 trackable objects in LEO
- Exponential debris growth (Kessler risk)
- Limited servicer capacity vs. rising demand
- Multiple mission types compete for the same platforms

Critical questions

- *Which* targets to visit?
- *When* and in what *order*?
- Under what *resource* constraints?
- How to maximise overall mission value?

LEO = Low Earth Orbit

background: NASA Orbital Debris Program Office (public domain)

Three Space Mission Types: ADR / OOS / EO

Image: ESA ClearSpace-1 concept

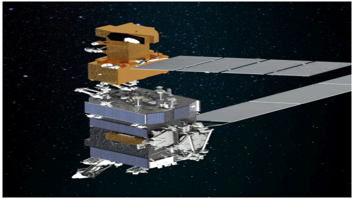


Image: NASA OSAM-1 servicing concept

Image: NASA Landsat program

ADR

Active Debris Removal

Target: Defunct satellites, debris

Value: Risk reduction (**ECI**)

Driver: Safety / sustainability

OOS

On-Orbit Servicing

Target: Serviceable spacecraft

Value: Commercial **NPV**

Driver: Lifetime extension

EO

Earth Observation

Target: Ground scenes, AOIs

Value: Information gain (**PIR**)

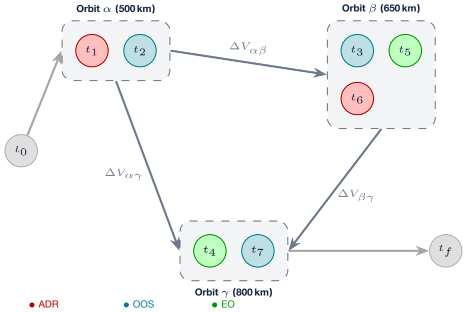
Driver: Timeliness / revisit

ADR = Active Debris Removal OOS = On-Orbit Servicing EO = Earth Observation ECI = Expected Collision Index NPV = Net Present Value PIR = Priority Information Requirement

Unified view: heterogeneous values (ECI/NPV/PIR), shared resource budget (ΔV), shared timing window constraints.

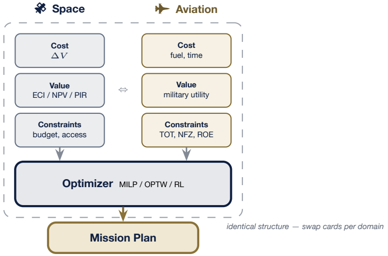
Graph-Based Decision Framework

Two-Layer Hierarchical Graph



Dashed box = orbit cluster (Level 2) ○ = task node (Level 1)

Component Swap = Domain Transfer



Same optimizer, swappable cards: Space (ΔV / ECI / budget) ⇔ Aviation (fuel / utility / TOT).

MILP = Mixed-Integer Linear Programming OPTW = Orienteering Problem with Time Windows RL = Reinforcement Learning

Complexity Design Space: Cost × Value

Objective: $J = \sum_i v(\cdot)\, x_i \; - \; \lambda \sum_{ij} w(\cdot)\, y_{ij}$

Level	Cost $w(\cdot)$	Value $v(\cdot)$
L0	$w(i,j)$ — static	$v(i)$ — static
L1	$w(i,j,t)$ — time	$v(i,t)$ — time
L2	$w(i,j,s_k)$ — state	$v(i,s_k)$ — state

Aviation concretization (L0 ⊂ L1 ⊂ L2):

- **L0**: great-circle distance, fixed fuel burn, static target priority
- **L1**: wind field, daylight window, altitude band, TOT constraint
- **L2**: fuel/weapons/damage state ⇒ SEAD→Strike precedence; history-dependent threat exposure

L2 costs cannot be pre-computed — must be embedded in runtime decision.

both v and w follow the same 3-level hierarchy

Complexity Matrix (Cost × Value)

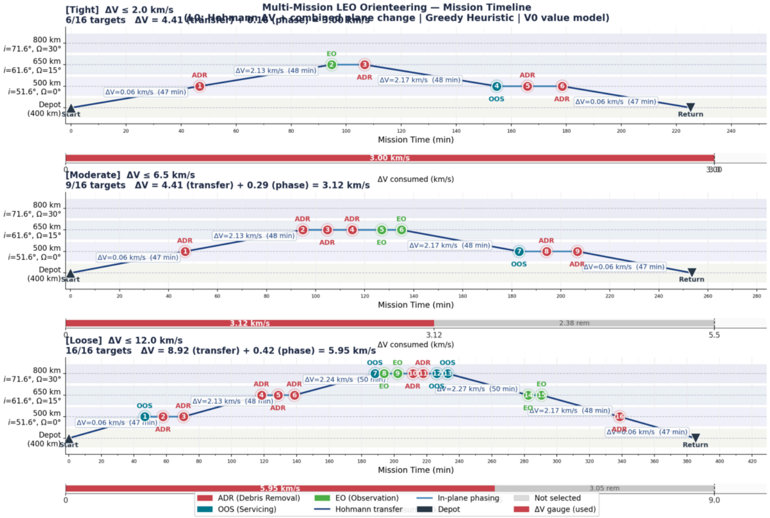
	L0	L1	L2
V0	Classic OP	OPTW	new
V1	TD-OP	**core band**	complex
V2	cond. OP	complex	full dyn.

Most prior work: upper-left (L0, V0) only
This framework: systematically covers the full matrix

OP = Orienteering Problem TD-OP = Time-Dependent OP OPTW = OP with Time Windows
TOT = Time-on-Target SEAD = Suppression of Enemy Air Defenses

Demo: Mission Timeline (Altitude vs. Time)

MILP = Mixed-Integer Linear Programming ADR/ OOS/ EO = see Slide 4 ΔV = velocity change budget



Space $\leftrightarrow$ Aviation: Component-Level Mapping

Concrete per-component mapping across the two domains

Component	Space Domain	Aviation / MUM-T
Graph structure	Orbit clusters $\to$ space objects	**Air zones $\to$ target locations**
Cost Model $w(\cdot)$	ΔV (Hohmann $\to$ J2 $\to$ Tsiolkovsky)	**Fuel (distance $\to$ wind/TOT $\to$ state)**
Value Model $v(\cdot)$	ECI / NPV / PIR	**Military utility / target priority**
Constraint Model	$\Delta V \leq B$, phasing, access window	**Endurance, TOT, weapons, NFZ, ROE**
Optimizer	**Identical** — MILP $\to$ OPTW $\to$ RL $\to$ Hybrid	

Takeaway: only rows 1–4 change across domains; the **Optimizer row is invariant**.

TOT = Time-on-Target NFZ = No-Fly Zone ROE = Rules of Engagement

Aviation Application: MUM-T Scenario

Manned-Unmanned Teaming with Heterogeneous Task Portfolio

MQ-4C Triton
u_1: ISR specialist
long-endurance recon

MQ-9 Reaper
u_2: multi-role
ISR + precision strike

XQ-58A Valkyrie
u_3: loyal wingman
stealth, attritable

Four Aviation Mission Types

mirrors Space ADR/OOS/EO

ISR recon	**Strike** precision	**SEAD** air-def.	**EW** jamming

ADR ↔ Strike   OOS ↔ ISR   EO ↔ SEAD / EW

MUM-T = Manned-Unmanned Teaming   UAV = Unmanned Aerial Vehicle   ISR = Intel./Surveil./Recon.
SEAD = Suppression of Enemy Air Defenses   EW = Electronic Warfare

Capability Matrix $\mathrm{Cap}_{K\times n}$

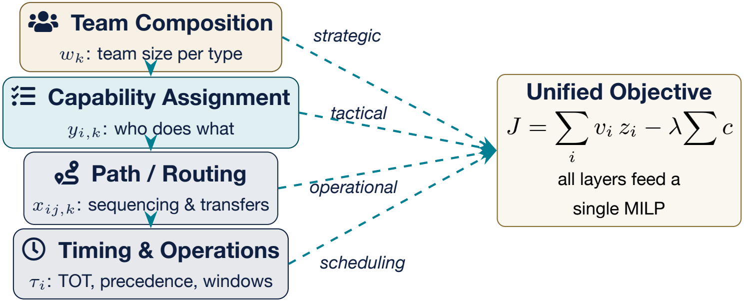
	ISR	Strike	SEAD	EW
u_1	✓			
u_2	✓	✓	✓	✓
u_3		✓	✓	

Key insight: specialist teams often outperform universal platforms under fuel / crew-cost constraints.

Decision Hierarchy: From Portfolio to Timeline

MILP = Mixed-Integer Linear Programming   MUM-T = Manned-Unmanned Teaming

A Single MILP Spans Four Nested Decisions



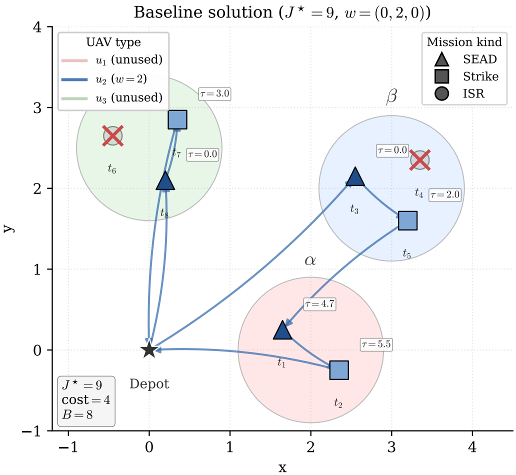
Backward compatible:

- $K=1$ homogeneous $\Rightarrow$ classic OP
- No state $\Rightarrow$ L0 hierarchy collapses

Forward extensible:

- Space mission: k = servicer class, z_i = debris/client
- Aviation: k = UAV type, z_i = target

Demo: Aviation MUM-T Mission Allocation

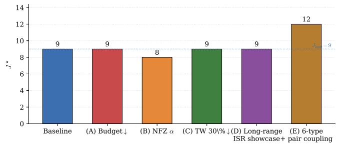


Baseline MILP solution over 3 zones (α, β, γ), 8 missions

Baseline Result

- **2** $\times$ u_2 deployed (multi-role)
- **6 / 8** missions served
- $J^\star = 9$, team cost $= 4$
- all TW / capability constraints met

Robustness across scenarios



Baseline vs. A/B/C/D/E variants

(NFZ, coupling, fleet expansion)

anned Teaming   SEAD = Suppression of Enemy Air Defenses   TW = Time Window

In-House Capabilities at a Glance

MILP = Mixed-Integer Linear Programming   OPTW = Orienteering Problem with Time Windows   RL = Reinforcement Learning

Optimization
MILP / OPTW / RL
task allocation, routing

Systems / Sim
testbed integ.
avionics-in-the-loop

Trajectory
Lambert / Hohmann
wind-aware flight paths

Mission Modeling
cost / value pairs
ISR / Strike / SEAD / EW

Graph Algorithms
2-layer hierarchies
zone ⟶ target clustering

Decision Architecture
nested MILP
MUM-T team–task–timing

Lab Profile

- **Focus**: autonomous multi-mission planning
- **Methods**: graph-based MILP, hierarchical decomp.
- **Space**: LEO transfers, debris / servicing / EO
- **Aviation**: MUM-T task alloc., TOT / NFZ constraints
- **Systems**: sim-in-the-loop prototyping

Decision core portable to **GE Aerospace avionics** — from mission planner to embedded runtime.

Summary

Key Takeaways

1. **Unified framework** for multi-mission planning across domains
2. **Graph-based approach** elegantly captures cost-value symmetry
3. **Hierarchical MILP** integrates team composition, assignment, routing, and timing in one model
4. **In-house capabilities** in MUM-T autonomy and operational constraint modeling

Questions & Discussion

Appendix A: MILP Formulation

Generic Multi-Mission Orienteering Formulation

$$\max \sum_{i \in N} v_i(s_i)\, z_i \;-\; \lambda \sum_{(i,j),k} w_{ij}(s_k)\, x_{ij,k}$$

$$\begin{aligned}
\text{s.t.}\quad & \textstyle\sum_k y_{i,k} = z_i && \text{(assign.)}\\
& \mathrm{Cap}_{c(i),k}\, y_{i,k} \geq y_{i,k} && \text{(capab.)}\\
& \textstyle\sum_j x_{ij,k} = \sum_j x_{ji,k} && \text{(flow)}\\
& \textstyle\sum_{(i,j)} w_{ij}\, x_{ij,k} \leq B_k && \text{(budget)}\\
& \tau_j \geq \tau_i + t_{ij} - M(1 - x_{ij,k}) && \text{(timing)}\\
& \tau_i \in [a_i, b_i], \quad x_{ij,k}, y_{i,k} \in \{0,1\}
\end{aligned}$$

Notation

- N: candidate targets; A: transfer arcs
- $k \in \{1, \ldots, K\}$: platform / servicer type
- $v_i(\cdot)$: value (ECI / NPV / PIR / utility)
- $w_{ij}(\cdot)$: cost (ΔV / fuel / time)
- s_k: state (remaining budget, loadout, damage)
- B_k: per-platform budget
- $[a_i, b_i]$: access / TOT window

Collapses to

- $K=1$, static $w, v \Rightarrow$ classic OP
- Add time window $\Rightarrow$ OPTW
- Add state $s_k \Rightarrow$ full L2 model

OP = Orienteering Problem OPTW = OP with Time Windows TOT = Time-on-Target

Appendix B: Solver Stack & Complexity

Solver stack

- **Modeling**: Python (PuLP, Pyomo)
- **MILP core**: COIN-OR CBC (open)
  - Gurobi / CPLEX for benchmark
- **Heuristics**: ALNS, GRASP for warm start
- **RL layer**: policy over state s_k
  - actor–critic on reduced graph
- **Runtime bindings**: Python $\rightarrow$ C++ for embedded avionics

Complexity & scaling

- Base OP: **NP-hard**
- Exact MILP: $|N| \lesssim 50$ in seconds
- Clustered hierarchy: $|N| \sim 500$ tractable
- State-dependent L2: heuristic / RL required

Scaling strategy

1. Decompose: upper (cluster) + lower (intra)
2. Warm start lower with greedy / ALNS
3. Branch-and-price for large K
4. Rolling-horizon replan on state updates

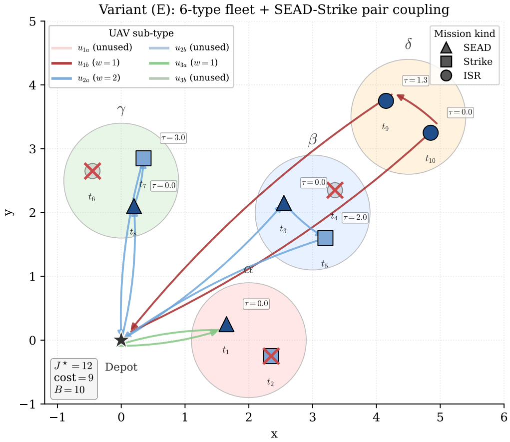
Appendix C: Domain Instantiation

Element	Space (ADR/OOS/EO)	Aviation (MUM-T)
Nodes N	debris, servicing clients, AOIs	ground targets, AOIs
Clusters	orbit-altitude bands	operational zones
Arc cost w_{ij}	ΔV (Hohmann / Lambert)	fuel + time (wind-aware)
Value v_i	ECI, NPV, PIR	military utility, priority
State s_k	ΔV residual, robotic arm use	fuel, weapons, damage, stealth
Budget B_k	total ΔV, propellant mass	endurance, crew duty
Time window	RAAN phasing, solar illum.	TOT, ROE window
Constraints	access, phasing, inclination	NFZ, airspace, precedence

Same MILP skeleton, two instantiations — domain transfer is a matter of swapping w, v, s, B, not rewriting the optimizer.

ECI = Expected Collision Index NPV = Net Present Value PIR = Priority Information Requirement

Appendix D: Aviation Deep-Dive



6-subtype fleet + SEAD–Strike pair coupling

What changed

- Fleet expanded 3 → **6 sub-types** (two generations per family)
- **SEAD–Strike pair coupling** added (fixes half-pair feasibility issue)

Result

- $J^\star = 12$ (**+33%** vs. baseline 9)
- Deploys $u_{1b}\times 1$, $u_{2a}\times 2$, $u_{3a}\times 1$
- **All 3 families active** — premium u_{1b} (long-range ISR) selected
- Team cost rises to 9, but richer capability mix

> Framework extends without re-engineering: add sub-types, tighten constraints, re-solve.

SEAD = Suppression of Enemy Air Defenses TW = Time Window