18th International Conference on Space Operations, Montreal, Canada, 26 - 30 May 2025.

SpaceOps-2025, ID # 409

Strategic Management of On-Orbit Servicing: Leveraging Operations Research Methods for Enhanced Mission Planning and Scheduling

Adam Abdin

University of Paris-Saclay, CentraleSupélec Engineering School, Department of Industrial Engineering, Gif-sur-Yvette, 91190, France. *adam.abdin@centralesupelec.fr*

Abstract

On-orbit servicing (OOS) is expected to play a growing role in enhancing the resilience, flexibility, and sustainability of space systems. As space infrastructure expands and diversifies, missions involving satellite refueling, repair, debris removal, and life extension will become central to long-term sustainability of the orbital environment. Efficiently planning and managing OOS missions involves making complex decisions across multiple timescales, subject to constraints on resources, operational feasibility, and uncertainty. The domain of operations research (OR) offers a broad set of analytical and computational tools for supporting such decisions, combining mathematical modeling with solution methods for optimization, planning under uncertainty, and systems analysis. This paper aims to provide a structured overview of OR approaches relevant to OOS mission planning and decision-making. It focuses on four broad categories of models: deterministic optimization, stochastic programming, sequential decision-making, and complementary decision analysis tools. These frameworks capture diverse aspects of OOS operations, from short-term scheduling to long-term strategic planning. The aim of this analysis is to synthesize and contextualize representative OR methods for researchers and mission planners working at the intersection of space systems and decision science. While not exhaustive, the paper is intended to support space professionals less familiar with OR techniques, as well as OR researchers seeking to adapt their models to the specific demands of space operations.
Keywords: On-Orbit servicing, Operations Research, Strategic Management, Space Mission Planning.

Acronyms/Abbreviations

AHP	Analytic Hierarchy Process
DP	Dynamic Programming
DQN	Deep Q-Network
GA	Genetic Algorithm
MAUT	Multi-Attribute Utility Theory
MCDA	Multi-Criteria Decision Analysis
MDP	Markov Decision Process
MILP	Mixed-Integer Linear Programming
OOS	On-Orbit Servicing
OR	Operations Research
PPO	Proximal Policy Optimization
PSO	Particle Swarm Optimization
RL	Reinforcement Learning
SAA	Sample Average Approximation
TRPO	Trust Region Policy Optimization

1. Introduction

On-orbit servicing (OOS) is gaining increasing attention as a means to enhance the resilience, flexibility, and sustainability of space systems. Missions such as satellite refueling, inspection, repair, debris removal, and constellation maintenance extend the utility of orbital assets and address growing concerns about orbital congestion and spacecraft lifecycle management [1]. These operations involve a variety of planning tasks: selecting which targets to service, scheduling missions across multiple orbits, optimizing transfer trajectories, coordinating fleets of servicer spacecraft, and assessing long-term strategic value and mission risk. Compared to traditional single-mission space operations, OOS systems must navigate broader decision spaces, evolving objectives, and multiple sources of uncertainty.

To address the complex and evolving nature of mission planning and management for OOS operations, researchers started to utilize a wide range of operations research (OR) tools tailored to different aspects of mission planning and decision support [2]. Optimization methods, including mixed-integer, nonlinear, and heuristic approaches, have been applied to analyze problems such as multi-target spacecraft servicer routing [3; 4; 5], orbital depot placement [6], and constellation servicing across LEO, MEO, and GEO regimes [7]. Several works address uncertainty in servicing operations through stochastic or scenario-based models, including applications to refueling [8], debris removal, and safety-constrained planning [9; 10]. Reinforcement learning and meta-learning techniques are explored for adaptive mission scheduling [11], OOS assistance for collision avoidance [12], and planning active debris removal scheduling [13; 14]. Other studies introduce hybrid frameworks that combine learning-based policies with optimization techniques to multi-rendezvous mission design [15], or integrate dynamic programming with tree search methods for large-scale constellation servicing [16].

These efforts demonstrate that OR techniques are increasingly being adapted to the diverse planning and management challenges expected to be posed by OOS systems. As OOS applications continue to expand in scope and complexity, clarifying how different OR frameworks and modeling paradigms align with specific classes of decisions can help inform both methodological development and practical implementation. These decision problems vary from deterministic orbital routing to planning servicing under uncertain demand, and from short-horizon scheduling to adaptive multi-mission control.

To support this evolving landscape, this paper provides a brief yet structured overview of core OR modeling approaches that can inform OOS mission planning and management. It highlights four broad classes of methods: (1) mixed-integer linear programming for deterministic large-scaly OOS routing and scheduling; (2) two-stage stochastic programming for modeling uncertainty in servicing needs and operational conditions; (3) sequential decision frameworks based on Markov Decision Processes and reinforcement learning for adaptive, multi-step OOS mission planning; and (4) complementary tools such as multi-criteria decision analysis and real options for incorporating qualitative preferences, risk considerations, and strategic flexibility. The discussion aims to clarify how these modeling approaches relate to key classes of OOS decisions, providing a conceptual foundation for researchers and practitioners engaging with this growing field.

2. Modeling On-Orbit Servicing Missions with Mixed-Integer Linear Programming

Mixed-integer linear programming (MILP) is a widely used optimization framework for problems involving both discrete and continuous decisions under linear constraints. MILP offer a structured framework to represent mission logic and operational constraints. This section outlines a general problem structure that illustrates how MILP can represent core elements of OOS planning models.

2.1. General Problem Structure and Formulation

The mathematical structure of an OOS mission optimization problem cast within a MILP framework involves defining decision variables, constraints, and objective functions to describe the elements of space mission planning while accounting for the constraints of the orbital environment. To illustrate how MILP modeling frameworks can be used for OOS mission planning problems, we can define the following sets:

- $\mathscr{S}$: Set of client satellites or space objects requiring servicing
- $\mathscr{D}$: Set of potential locations for supporting infrastructure (e.g., refueling depots)
- $\mathscr{T}$: Set of discrete time steps in the planning horizon $\mathscr{T} = \{t_0, t_1, t_2, \ldots, t_{\max}\}$

The mathematical formulation of a general MILP model can be expressed as:

$$\begin{aligned} \min_{\mathbf{x},\mathbf{y}} \quad & \mathbf{c}^T\mathbf{x} + \mathbf{d}^T\mathbf{y} \\ \text{subject to} \quad & \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{y} \le \mathbf{b} \\ & \mathbf{E}\mathbf{x} + \mathbf{F}\mathbf{y} = \mathbf{e} \\ & \mathbf{x} \in \{0,1\}^n, \quad \mathbf{y} \in \mathbb{R}^m_+ \end{aligned} \tag{1}$$

Where $\mathbf{x}$ represents binary decision variables (e.g., orbital routing of servicing vehicles, servicing decisions), $\mathbf{y}$ represents continuous variables (e.g., propellant use, servicing timing), and the constraints enforce mission requirements and physical limitations. This framework is general to MILP problems, however its adaptation to OOS requires specific accommodations to account for the physics of space operations.

2.2. Time-Expanded Network Representation

A key modeling technique for space mission planning within an MILP framework is the use of time-expanded network representations to transform the continuous-time dynamics of orbital motion and moving targets into a discrete-time network suitable for MILP modeling and optimization [17; 18]. In this approach, both time and space (e.g., orbital positions or operational states) are discretized, creating a network where each node represents a satellite, servicing vehicle, or space infrastructure depot at a specific time step. This yields a static network representation of what is inherently a dynamic problem involving moving objects.

To illustrate this approach, the time-expanded network $G = (N, A)$ is constructed as follows:

1. The node set N consists of tuples (i,t) where $i \in \mathscr{S} \cup \mathscr{D}$ represents a physical location (target satellite or servicing vehicle) and $t \in \mathscr{T}$ represents a specific time step.
2. The arc set A contains elements $((i,t) \rightarrow (j,t'))$ where a feasible transfer from location i at time t to location j at time t' exists according to orbital mechanics.

Each arc $a = ((i,t) \rightarrow (j,t')) \in A$ is characterized by:

- $\Delta V(a)$: The velocity change required for the orbital maneuver.
- $\tau(a) = t' - t$: The time of flight for the transfer.

The key orbital mechanics considerations that determine arc feasibility and cost in the time-expanded network can include: (i) the energy requirements for Hohmann transfers between orbits of different altitudes, (ii) the propellant-intensive plane changes needed for satellites in different orbital inclinations, and (iii) the phasing requirements that create specific time windows when transfers between orbits become possible. These physical constraints fundamentally shape the network structure by limiting when arcs can exist between nodes and by determining the ΔV cost associated with each potential transfer.

The network can also include waiting arcs of the form $((i,t) \rightarrow (i,t+\Delta t))$[1] to allow the servicing spacecraft to remain at a given orbit for some time steps. These have zero ΔV cost but consume time. This is a particularly important consideration when the servicing vehicle needs to wait for favorable transfer windows, which is absent in traditional terrestrial routing problems. This construction effectively "unrolls" the orbital dynamics over time [18], transforming a continuous-time problem with moving targets into a discrete network problem where the time-dependent nature of transfers is explicitly represented.

2.3. MILP Formulation for OOS Mission Planning Optimization

Using the time-expanded network, we can formulate a generic MILP model for OOS mission planning that accounts for the most important mission planning concepts. For simplicity, we present a formulation for a single servicer, though the model can be readily extended to multiple servicers.

Model Formulation: For a set of satellites $\mathscr{S}$ and discrete time steps $\mathscr{T}$, a generic MILP-based formulation for OOS mission planning with a single servicing vehicle can be expressed as:

$$\max \quad \sum_{i\in\mathscr{S}} \sum_{t\in\mathscr{T}} y_{i,t} \tag{2}$$

$$\text{s.t.} \quad y_{i,t} \leq \sum_{a\in\delta^-(i,t)} x_a, \quad \forall i \in \mathscr{S}, t \in \mathscr{T} \tag{3}$$

$$\sum_{t\in\mathscr{T}} y_{i,t} \leq 1, \quad \forall i \in \mathscr{S} \tag{4}$$

$$\sum_{a\in A} \Delta V(a) \cdot x_a \leq F_{\text{initial}} \tag{5}$$

$$\sum_{a\in\delta^-(i,t)} x_a - \sum_{a\in\delta^+(i,t)} x_a = b_{it}, \quad \forall (i,t) \in N \tag{6}$$

$$f_{j,t'} \leq f_{i,t} - \Delta V(a) \cdot x_a + F_{\max}(1 - x_a), \quad \forall a = ((i,t) \rightarrow (j,t')) \in A \tag{7}$$

$$f_{j,t'} \geq f_{i,t} - \Delta V(a) \cdot x_a - F_{\max}(1 - x_a), \quad \forall a = ((i,t) \rightarrow (j,t')) \in A \tag{8}$$

$$f_{i_0,t_0} = F_{\text{initial}} \tag{9}$$

$$0 \leq f_{i,t} \leq F_{\max}, \quad \forall (i,t) \in N \tag{10}$$

For this model, the decision variables are defined as follows: $x_a \in \{0,1\}$ indicates whether arc $a \in A$ is included in the vehicle's trajectory; $y_{i,t} \in \{0,1\}$ indicates whether satellite $i \in \mathscr{S}$ is serviced at time $t \in \mathscr{T}$; and $f_{i,t} \in \mathbb{R}_+$ denotes the remaining propellant at node (i,t).

[1]Unlike transfer arcs $((i,t) \rightarrow (j,t'))$ where $i \neq j$, waiting arcs represent the servicer staying at the same location across time steps. The notation Δt may differ from the global time discretization step if variable time steps are allowed.

Moreover, the key parameters include: $\Delta V(a)$, the velocity increment required for the orbital transfer along arc a; b_{it}, the net flow requirement at node (i,t); $F_{\max}$, the maximum propellant capacity of the servicing vehicle; and F_{initial}, its initial onboard propellant at mission start.

The objective function (2) maximizes the number of satellites serviced during the planning horizon. Constraint (3) ensures that a satellite can only be serviced if the vehicle is present at the corresponding node. Constraint (4) ensures each satellite is serviced at most once. Constraint (5) enforces a mission-wide upper bound on total propellant usage. Constraint (6) maintains flow balance at each node in the time-expanded network, modeling the path continuity of the servicing vehicle. Propellant consumption is captured through the disjunctive constraints (7) and (8), with the initial propellant state specified in constraint (9). Finally, constraint (10) ensures feasible fuel levels throughout the mission.

While this formulation maximizes the number of serviced satellites during a mission, other mission objectives can be considered such as those illustrated in Table (1).

Objective Type	**Mathematical Formulation**	**Description**
Maximize Mission Value	$\max \sum_{i\in\mathscr{S}} \sum_{t\in\mathscr{T}} v_i \cdot y_{i,t}$	Prioritizes servicing high-value satellites when resources are limited
Minimize Propellant Usage	$\min \sum_{a\in A} \Delta V(a) \cdot x_a$	Focuses on fuel efficiency, potentially at the cost of service coverage
Multi-Objective	$\min \alpha \sum_{a\in A} \Delta V(a) \cdot x_a - \beta \sum_{i\in\mathscr{S}} v_i y_{i,t} + \gamma t_{max}$	Balances propellant efficiency, mission value, and time considerations based on mission priorities

Table 1: Alternative objective functions for OOS mission planning: v_i is the value of servicing satellite i, t_{max} is the mission completion time, and α, β, γ are weighting parameters.

In practice, the choice of objective function is closely tied to operational goals and system-level tradeoffs such as maximizing return on investment, minimizing risk, or preserving maneuvering capability for future missions.

2.4. Orbital considerations in constructing feasible network topologies

The distinguishing features of OOS MILP models arise not from the algebraic formulation itself, but from the physics-driven constraints embedded in the construction of the time-expanded network. Unlike terrestrial routing problems, where nodes and arcs can often be defined freely, orbital dynamics impose structural constraints on the network topology that must be enforced during preprocessing.

Key considerations include:

1. Orbital Phasing and Timing Windows: Spacecraft can only transfer between orbits during specific windows when the relative positions of origin and destination allow for efficient maneuvers. This manifests in the network as *missing arcs* between nodes that represent infeasible transfers, creating a sparse network structure where many potential connections are physically impossible.

2. Non-Euclidean Distance Properties: The ΔV expenditure in orbital mechanics differs fundamentally from typical Euclidean distance metrics in three important ways: it is asymmetric ($\Delta V_{i\to j} \neq \Delta V_{j\to i}$), time-varying (costs change with departure time), and non-transitive (adding an intermediate stop, or even waiting in place, can reduce total cost). These properties should be taken into account in modeling and analysis of the optimization output.

3. Orbital Evolution: perturbations such as J2-perturbation naturally changes satellite orbital planes over time without propellant expenditure. The MILP model can capture this through zero-ΔV waiting arcs that allow the servicer to strategically delay transfers until natural orbital evolution creates more favorable conditions.

These orbital considerations are typically handled during the generation of the time-expanded network to ensure that the resulting MILP formulation operates over a feasible and consistent network.

2.5. Application Examples

The MILP formulation can readily accommodate extensions to reflect specific OOS mission planning requirements, including task-specific constraints, operational modes, and servicing objectives. These additions typically involve new decision variables and constraints. Table (2) summarizes several possible OOS mission types and the corresponding modeling adaptations within the MILP framework.

Beyond these, additional operational constraints can be considered-such as communication windows, approach safety requirements, and rendez-vous requirements- can also be added modularly depending on the fidelity required. In principle, most of these extensions can be modeled while maintaining the underlying MILP structure.

Table 2: Adaptation of MILP Framework for Different OOS Mission Types

Mission Type	**Key Extensions to the MILP Framework**
Satellite Refueling	Continuous variables for tracking propellant transfer quantities / Extended resource constraints maintaining mass balance between servicer and clients satellites / Limits on depot capacity or transfer efficiency.
Inspection and Repair	Variables tracking specialized equipment allocation for different repair tasks / Viewing angle constraints / Operation duration constraints based on task complexity.
Debris Removal	Extended constraints for capturing based on object characteristics / Additional ΔV budget for de-orbit maneuvers / Risk-based prioritization in objective function.
Orbital Insertion	Variables for final orbital insertion maneuvers of target satellites from deployment orbits / Phasing constraints for precise positioning within constellations or target orbits / Constraints to enforce ordered deployment of multiple satellites from a single carrier vehicle.

2.6. Solution Methods

MILP-based formulations for OOS mission planning can be solved using a variety of techniques depending on problem scale and structure. This section highlights three broad classes of methods: exact algorithms for moderate-scale instances, decomposition techniques for large and structured problems, and heuristic approaches for computationally challenging or non-convex variants.

Standard solution methods: For moderate-sized instances, standard MILP algorithms—such as branch-and-bound, cutting planes, and branch-and-cut are effective and can be directly implemented using commercial solvers such as Gurobi and CPLEX [19; 20]. These methods are suitable for solving tightly formulated instances, for example, with binary routing and scheduling decisions. They are commonly used in OOS studies involving single- or multi- servicer planning over discrete time horizons [21; 17].

Decomposition techniques: For large-scale scenarios involving extended planning horizons, multiple servicing vehicles, or a dense set of candidate targets, solving the full MILP directly can be computationally prohibitive. In such cases, decomposition-based methods provide tractable alternatives by leveraging problem structure. Benders decomposition separates the problem into a master problem (typically handling binary routing or scheduling decisions) and subproblems (handling continuous resource allocation), which can be solved iteratively and, in some cases, in parallel [22]. Column generation constructs the problem incrementally by generating new variables (e.g., feasible orbital sequences) only when they can improve the solution, allowing for efficient handling of formulations with an exponential number of variables [23]. Additional speedups can be obtained through valid inequalities, which strengthen the MILP relaxation by eliminating infeasible or suboptimal regions of the search space using domain-specific logic [24].

Heuristics and metaheuristics: When problem size, combinatorial complexity, or non-convex features make exact MILP approaches impractical, heuristic and metaheuristic methods offer scalable alternatives for generating high-quality approximate solutions. These methods typically construct or iteratively improve candidate solutions using rule-based strategies or search procedures. Common methods include local search, genetic algorithms (GAs), and particle swarm optimization (PSO), which have been applied to a range of OOS mission planning tasks [25; 26]. Heuristics can also be integrated into hybrid frameworks that embed heuristic guidance within exact algorithms [16], offering a balance between computational efficiency and solution quality for time-sensitive or complex mission planning scenarios.

3. Two-Stage Stochastic Programming for On-Orbit Servicing under Uncertainty

Section (2) presented a deterministic framework for mission planning under known conditions. However, in realistic operational environments, many key parameters—such as servicing needs, satellite availability, and orbital conditions—are inherently uncertain. In some cases it is important to explicitly account for this uncertainty in the OOS planning models. This section illustrates how this can be achieved using a two-stage stochastic programming framework.

3.1. Modeling Uncertainty in OOS Operations

Operational uncertainties in OOS missions can arise from many sources, including stochastic variations in client satellite needs, degradation of on-board systems, uncertain transfer durations, and orbital perturbations. These uncertainties can disrupt planned servicing sequences or compromise mission feasibility if not properly accounted for in the planning phase.

They can be grouped into several broad categories relevant to OOS mission modeling:

- *Service demand uncertainty:* The amount of fuel, time, or specialized equipment required for servicing may vary significantly due to client-specific conditions, degradation profiles, or anomaly-driven task revisions. This can be modeled by a random variable D_i representing demand for satellite i, with expected value $\mathbb{E}[D_i]$ and scenario-dependent realization D_i^ϕ.
- *Operational timing uncertainty:* The duration of orbital maneuvers and servicing tasks can differ from nominal estimates due to navigation delays, environmental conditions, or interactions with client spacecraft. This can be represented as a stochastic task duration $\tau_i = \bar{\tau}_i + \varepsilon_i$, where $\bar{\tau}_i$ is the nominal duration and ε_i is a noise term that can be sampled from a suitable distribution.
- *Orbital parameter uncertainty:* Variations in orbital elements such as inclination, eccentricity, or node position affect transfer feasibility and the associated ΔV costs. These may arise from uncertainties in tracking, maneuver execution errors, or long-term perturbations (e.g., atmospheric drag, third-body effects).
- *System performance uncertainty:* The servicer spacecraft may face reduced thrust, power limitations, or control anomalies, affecting its ability to execute planned maneuvers reliably. For instance, available thrust T could be modeled as $T^\phi = T_0(1-\delta^\phi)$, where δ^ϕ is a scenario-dependent degradation factor. Alternatively, specific impulse I_{sp} may follow a distribution derived from testing data.
- *Satellite failure modeling:* A specific form of demand uncertainty involves probabilistic subsystem failures. For example, the time-to-failure T of a satellite component may follow a Weibull distribution with a probability density function $f(t) = \frac{\beta}{\eta}\left(\frac{t}{\eta}\right)^{\beta-1} e^{-(t/\eta)^\beta}$ and reliability function $R(t) = P(T > t) = 1 - F(t)$, where β and η are the shape and scale parameters of the distribution [27].

3.2. Two-Stage Stochastic Optimization Framework

Among the various methodologies for decision-making under uncertainty, two-stage stochastic programming offers a relatively simple and widely used framework. It separates first-stage (pre-uncertainty) decisions from second-stage (recourse) decisions made after uncertainty is revealed. The objective is to optimize expected mission performance across a range of plausible scenarios [28; 29]. In this approach, first-stage decisions represent the "here-and-now" commitments made prior to uncertainty resolution—such as routing and infrastructure choices. Once the actual mission environment is revealed, second-stage decisions allow scenario-specific operational adjustments.

To formally illustrate this approach, we represent uncertainty through a finite set of scenarios, each encoding a possible realization of the uncertain mission environment. Let Φ denote the finite scenario space, with each scenario $\phi \in \Phi$ assigned a probability p_ϕ such that $\sum_{\phi\in\Phi} p_\phi = 1$. For each scenario, uncertain mission parameters take different realizations that influence the feasibility and cost of decisions. Table (3) summarizes representative decision variables that can be used in OOS mission planning under uncertainty within a two-stage setting. The first-stage illustrate commitments that has to be made prior to uncertainty realization, while the second-stage illustrate recourse actions that can adapt to scenario-specific conditions.

Table 3: Examples of First-Stage and Second-Stage Decisions in Two-Stage OOS Planning

Decision Stage	Example Decision Variables for OOS planning under uncertainty
First-Stage (Strategic)	Infrastructure Deployment ($z_d \in \{0,1\}$): Determines whether infrastructure (e.g., orbital depots) is present or not. Trajectory Selection ($x_a \in \{0,1\}$): Selects maneuver arcs in the time-expanded network to form the servicer route.
Second-Stage (Operational Recourse)	Servicing Execution ($y_i^\phi \in \{0,1\}$): Indicates whether satellite i is serviced in scenario ϕ. Schedule Adjustment ($\Delta t_i^\phi \in \mathbb{R}_+$): Models timing delays due to scenario-dependent task durations. Resource Reallocation ($r_i^\phi \in \mathbb{R}_+$): Allocates fuel or power to match realized demands.

3.2.1. Mathematical Formulation

We now illustrate how to formalize the two-stage stochastic programming model by specifying the objective function and associated constraints that govern first-stage and scenario-dependent second-stage decisions.

First-Stage Problem: Two-stage stochastic optimization minimizes the sum of first-stage costs and the expected second-stage costs over a set of scenarios. This can be formally expressed as shown in Eq. (11)

$$\min_{x\in\mathscr{X}} \left[C_{\text{first}}(x) + \mathbb{E}_{\xi}[Q(x;\xi)]\right] \tag{11}$$

where:

- $C_{\text{first}}(x)$ denotes the deterministic cost associated with first-stage decisions. In OOS problems, this could represent, for example, total propellant cost, the number of maneuvers performed, or any other cost metric linked to routing and planning.
- $\mathbb{E}_{\xi}[Q(x;\xi)] = \sum_{\phi\in\Phi} p_{\phi} \cdot Q(x;\xi^{\phi})$ is the expected second-stage recourse cost, taken over a finite set of uncertainty realizations ξ^{ϕ}, indexed by $\phi \in \Phi$.
- $\mathscr{X}$ denotes the feasible region for first-stage decisions, typically defined by routing, flow conservation, and resource constraints as described in the OOS model presented in Section (2).

Second-Stage Problem: For each scenario $\phi \in \Phi$, the second-stage (recourse) problem is defined as:

$$Q(x;\xi^{\phi}) = \min_{\mathbf{y}^{\phi}} \left\{ C^{\phi}_{\text{recourse}}(\mathbf{y}^{\phi}) \ : \ T^{\phi}(x,\mathbf{y}^{\phi}) \leq h^{\phi} \right\} \tag{12}$$

where:

- $\mathbf{y}^{\phi}$ is the vector of second-stage decision variables under scenario ϕ, including service adjustments, timing modifications, and resource reallocations.
- $C^{\phi}_{\text{recourse}}(\mathbf{y}^{\phi})$ captures the operational cost under scenario ϕ, which may include additional propellant consumption, missed servicing penalties, or delay costs.
- $T^{\phi}(x,\mathbf{y}^{\phi}) \leq h^{\phi}$ represents the set of feasibility constraints that link first-stage decisions (x) to second-stage actions $\mathbf{y}^{\phi}$ under the realized scenario ξ^{ϕ}.

This formulation allows the mission plan to adapt to variations in service demand, operational durations, or satellite availability while optimizing expected performance. The structure of the second-stage problem is typically similar to the deterministic model in Section (2), but with scenario-specific parameters and outcomes.

Linking constraints: A key aspect of the two-stage formulation are the constraints that link first-stage strategic decisions with second-stage recourse actions. For OOS mission planning model these would include (i) resource balance constraints, which ensure that second-stage actions reflect both initial allocations and observed needs; (ii) schedule feasibility constraints, which enforce consistency with planned orbital and temporal structures; and (iii) continuity requirements, which guarantee that committed servicing targets are followed through under each scenario. These linking constraints ensure that post-scenario adjustments remain consistent with the planned mission structure and respect the physical and operational limits of the system.

3.3. Application Examples

As previously discussed, the two-stage stochastic optimization framework can be adapted to various OOS mission types by incorporating uncertainty into key mission parameters. Table (4) illustrates how different OOS stochastic planning problems can be modeled by adjusting the structure of first- and second-stage decisions to accommodate mission-specific sources of uncertainty.

Table 4: Adaptation of Stochastic Framework for Different OOS Mission Types

Mission Type	Stochastic Modeling Representation
Preventive Maintenance	First-stage: Select satellites based on failure probability distributions Second-stage: Adjust servicing actions in response to observed health status or anomaly reports.
Satellite Refueling	First-stage: Plan visitation sequence and depot activation based on nominal demand estimates Second-stage: Adjust delivered fuel to satellites based on realized target needs.
Debris Removal	First-stage: Select debris targets and plan capture sequence Second-stage: Adjust capture strategy and de-orbit maneuvers based on realized tumbling rates, mass estimates, and structural integrity.

These examples illustrate how the two-stage framework can accommodate different sources of mission uncertainty by separating early planning from scenario-specific adjustments. The resulting mission plans would perform well on average across the considered scenarios, providing solutions that are optimized in expectation. While the table highlights three representative cases, the structure is general and can be extended to a wide range of OOS planning problems where uncertainty plays a critical role.

3.4. Solution Methods

Two-stage stochastic programming formulations for OOS planning introduce additional complexity compared to deterministic modeling approaches due to the inclusion of uncertainty and scenario-based decision coupling. The appropriate solution method depends on factors such as the size of the scenario space, problem structure, and the level of accuracy required. Below, we outline several widely used solution approaches that are particularly relevant to OOS mission contexts.

Sample Average Approximation (SAA): A common approach for handling uncertainty in two-stage stochastic optimization is to approximate the expected cost using a finite set of sampled scenarios. In this method, a representative sample $\{\xi^{\phi_1}, \ldots, \xi^{\phi_N}\}$ is drawn from the underlying distribution of the random vector ξ, and the resulting sample average is used in place of the true expectation in the objective function. The problem can then be solved as a deterministic equivalent, and statistical techniques may be used to assess solution quality and convergence as the sample size increases [30]. SAA is especially useful when the true uncertainty distribution is continuous or complex, making full scenario enumeration impractical—such as in early-stage campaign design or exploratory trade studies.

Scenario-Based Decomposition: Another way to address computational tractability in large-scale stochastic programs is through scenario decomposition. The L-shaped method, a classical form of Benders decomposition, separates the problem into a master problem over first-stage variables and independent subproblems for each scenario. Feasibility and optimality cuts from the subproblems iteratively refine the master problem, enabling efficient convergence without solving the full extensive form [31; 29].

Other Decomposition Strategies: Additional techniques such as Progressive Hedging [32] and Lagrangian relaxation [33] offer alternative ways to manage non-anticipativity and coupling constraints. Instead of enforcing shared first-stage decisions directly, these methods iteratively coordinate scenario-specific solutions while gradually reconciling inconsistencies. They are particularly useful in large-scale or distributed settings, such as multi-servicer planning or systems with complex global constraints.

Finally, while two-stage stochastic programming provides a practical foundation for modeling uncertainty in OOS mission planning, extensions such as distributionally robust, multi-stage, or learning-based models may offer greater flexibility for addressing more complex or dynamic settings.

The next section introduces a class of sequential decision-making approaches that naturally accommodate multi-stage planning under uncertainty.

4. Sequential Decision Models for On-Orbit Servicing Operations

4.1. Sequential Nature of OOS Decision Making

Many OOS missions involve a sequence of decisions that must be made over time in the presence of uncertainty. These include not only immediate planning choices, but also adaptive responses to new information, resource updates, and evolving mission conditions. In such settings, the outcome of each decision can influence the state of the system and shape future options, making it important to model the temporal and contingent nature of the planning process [34]. Applications include adaptive servicing schedules, dynamic resource allocation, and evolving infrastructure configurations.

These types of adaptive decision-making problems require models that can capture temporal evolution and the dependence between consecutive decisions over time. Markov Decision Processes (MDPs) offer a mathematically rigorous framework for representing such problems. In an MDP, the system evolves over discrete time steps according to a probabilistic transition model, and decisions are made based on the current system state. The solution is a policy—a mapping from states to actions—that maximizes expected cumulative rewards or minimizes expected costs over a planning horizon [35]. MDPs also provide the foundation for developing autonomous planning systems, particularly through reinforcement learning (RL) techniques, where optimal policies are learned from data via interaction with the environment rather than through complete models of the environment.

4.2. *MDP Mathematical Formulation*

An MDP is defined by the tuple (S, A, P, R, γ), where each component corresponds to a core element of the decision process:

- S is the set of system states;
- A is the set of available actions;
- $P(s'|s,a)$ is the transition probability of reaching state s' from state s after taking action a;
- $R(s,a,s')$ is the immediate reward (or cost) incurred from the state-action transition; and
- $\gamma \in [0,1]$ is a discount factor encoding the relative importance of future outcomes.

Each of these components can be tailored to reflect the structure and dynamics of OOS missions, as illustrated in Table (5).

Table 5: Mapping the MDP Framework to On-Orbit Servicing mission planning contexts

Element	**OOS Context and Examples**
S (State Space)	**Satellite:** Health levels, fuel status, orbital position **Servicer:** Position, remaining resources, capabilities **Infrastructure:** Depot locations, deployment status
A (Action Space)	**Servicing:** Satellite/task selection (repair, refuel, inspect) **Movement:** Orbital transfers, repositioning **Resource:** Propellant allocation, equipment deployment
$P(s'\|s,a)$ (Transitions)	**Satellite Degradation:** Health evolution models **Service Uncertainty:** Success/failure probabilities **Resource Usage:** Deterministic or stochastic consumption
$R(s,a,s')$ (Rewards)	**Service Value:** Benefits from completed operations **Resource Costs:** Penalties for consumption **Risk Penalties:** Discouraging hazardous maneuvers

Applying MDPs in practice requires thoughtful design choices that balance fidelity and tractability. Below we highlight four central considerations.

State Space Design: The state should include all information necessary to predict future system evolution, such as satellite servicing status, servicing vehicle location, available propellant, and mission time. To preserve the Markov property, relevant history (e.g., time since last visit or accumulated delays) may be encoded directly in the state. Orbital positions are often discretized into operational zones to manage dimensionality.

Action Space Complexity: Actions typically involve choices over when, where, and whom to service—leading to a large and combinatorial action space. Structuring actions into high-level decisions (e.g., select next target) followed by lower-level operations (e.g., execute maneuver, perform task) can reduce complexity while preserving operational detail.

Transition Modeling: State transitions are governed by both deterministic elements (e.g., orbital dynamics) and stochastic components (e.g., task success rates, unexpected demand changes). Transition models may be derived from physics-based simulations or estimated from mission data, depending on the availability of system knowledge.

Reward Design: Reward functions encode mission objectives and shape decision policies. Common goals include maximizing completed services, minimizing fuel use, and avoiding mission risk. When objectives conflict, scalarization techniques or constrained formulations can balance trade-offs in a principled way.

4.3. *Application Examples*

Table (6) illustrates representative OOS mission planning scenarios where MDP-based models can be particularly relevant. Each involve multi-step decision-making, where policies must adapt to evolving system states and operational constraints over time.

4.4. *Solution Methods*

A wide range of solution techniques can be used to compute or approximate optimal policies in MDPs, depending on whether the transition model is known and on the size of the state and action spaces. Below, we distinguish between classical dynamic programming approaches for small, model-based problems, and reinforcement learning methods suitable for larger, simulation-based or model-free settings.

Table 6: Examples of Sequential Decision Model Applications in OOS

Application Scenario	**Sequential Modeling Focus**
Constellation Maintenance	Adapt servicing schedules based on evolving satellite needs and failure risks / Learn policies that prioritize servicing based on time-varying state and resource levels / Incorporate degradation models to anticipate future needs.
Campaign-level planning	Coordinate a sequence of servicing missions over long time horizons / Reoptimize routes and service selections in response to changing telemetry, task completion, or system degradation / Balance short-term task value against long-term mission viability.
Adaptive use of OOS infrastructure	Model depot status and location as part of system state / Optimize timing and positioning of infrastructure usage across missions / Develop policies that adapt servicing strategies based on dynamic infrastructure availability and demand realization.

4.4.1. Dynamic Programming Methods

For MDP with tractable state and action spaces, classical dynamic programming (DP) techniques can be applied to compute optimal policies. These include value iteration and policy iteration, which are guaranteed to converge to optimal solutions. Table (7) summarizes the core mechanics of these methods. These methods assume full knowledge of transition probabilities and are most effective when state and action spaces are sufficiently small to enable exact computation.

Table 7: Common Dynamic Programming Approaches. Here, $V^*(s)$ denotes the optimal value function, $V^\pi(s)$ is the value function under policy π, and $\pi^*(s)$ represents the optimal policy.

Method	**Description**
Value Iteration	Iteratively computes the optimal value function using the Bellman equation: $V^*(s) = \max_a \{\sum_{s'} P(s'\|s,a)[R(s,a,s') + \gamma V^*(s')]\}$
Policy Iteration	Alternates between policy evaluation and improvement steps: $\pi^*(s) = \arg\max_a \{\sum_{s'} P(s'\|s,a)[R(s,a,s') + \gamma V^\pi(s')]\}$

4.4.2. Reinforcement Learning (RL) Methods

When transition dynamics are complex, unknown, or difficult to model explicitly, RL methods offer an alternative approach for solving sequential decision problems [36]. These methods enable agents to learn decision policies through repeated interaction with a simulated or real environment, without requiring explicit knowledge of transition probabilities. RL is particularly well suited for training autonomous planning systems, where onboard agents must adapt to uncertain, high-dimensional, or continuously evolving mission environments. This makes RL especially valuable for OOS scenarios involving long-term autonomy, complex operational constraints, or partially known dynamics.

RL techniques for OOS can be broadly categorized into value-based, policy-based, and hybrid actor-critic approaches. Table (8) summarizes the key families of methods, their characteristics, and representative algorithms.

Table 8: Common Reinforcement Learning Approaches. Here, $Q(s,a;\theta)$ denotes a parameterized estimate of the action-value function, $Q^*(s,a)$ is the optimal action-value function, $\pi_\theta(a|s)$ is a parameterized policy, $J(\theta)$ is the expected return under policy π_θ, and ∇_θ denotes the gradient with respect to the policy parameters.

Method Class	**Description**	**Representative Algorithms**
Tabular Methods	Use lookup tables to estimate value functions; suitable for small, discrete state-action spaces.	Q-Learning [37], SARSA [38]
Value Function Approximation	Use function approximators such as neural networks to estimate $Q(s,a;\theta) \approx Q^*(s,a)$ over large or continuous domains.	DQN [39], Double DQN [40], Dueling DQN [41], Rainbow [42]
Policy Gradient Methods	Directly optimize a parameterized policy $\pi_\theta(a\|s)$ by estimating $\nabla_\theta J(\theta) = \mathbb{E}_{\pi_\theta}[\nabla_\theta \log \pi_\theta(a\|s) \cdot Q^{\pi_\theta}(s,a)]$. Particularly effective for continuous action spaces.	PPO [43], TRPO [44], Actor-Critic [45]

5. Complementary Decision Analysis Methods for On-Orbit Servicing Missions

The methodologies discussed in the previous sections offer strong foundations for OOS mission planning and optimal management. Yet, real-world decision-making also involves qualitative trade-offs, stakeholder preferences, market competition, and long-term investment strategies. This section briefly introduces complementary OR methods that extend beyond mathematical modeling and optimization with the aim of offering broader decision support across strategic, tactical, and operational levels.

5.1. Multi-Criteria Decision Analysis (MCDA)

Many OOS planning and management problems involve trade-offs among competing objectives, such as minimizing cost, maximizing mission value, reducing risk, or preserving flexibility, which were highlighted in the optimization models presented in earlier sections. Multi-Criteria Decision Analysis (MCDA) [46] complements these methods by providing structured frameworks to evaluate such trade-offs when preferences are difficult to encode directly in an optimization objective. Techniques such as Multi-Attribute Utility Theory (MAUT) [47] and the Analytic Hierarchy Process (AHP) [48] help structure complex decision problems and quantify stakeholder priorities across economic, technical, and strategic dimensions.

Techniques such as Multi-Attribute Utility Theory (MAUT) [47] and outranking methods (e.g., ELECTRE, PROMETHEE) [49] provide structured approaches for comparing alternatives when objectives conflict or are difficult to reduce to a single metric. While simpler methods such as the Analytic Hierarchy Process (AHP) [48] are also used, outranking methods are often more suitable in technical planning contexts involving both quantitative and qualitative criteria, and where some options may be partially incomparable.

Table (9) summarizes representative evaluation criteria that can be used in MCDA for OOS decision-making, organized by economic, technical, risk, and strategic categories.

Table 9: Representative Criteria in MCDA that could be used for OOS mission planning

Criteria Category	**Examples applicable for OOS**
Economic	Mission cost, return on investment, long-term revenue potential
Technical	Mission duration, maneuver efficiency, system complexity
Risk	Collision probability, failure likelihood, redundancy levels
Strategic	Reusability, market presence, adaptability to future demand

A typical MCDA process involves identifying relevant criteria, assigning weights based on stakeholder preferences, and evaluating each decision alternative against these criteria. Scores can be aggregated using weighted sums, utility functions, or ranking methods to support transparent, preference-informed decision-making.

5.2. Risk Assessment and Portfolio Management Approaches

Risk plays a central role in OOS mission planning, where uncertain operational conditions can significantly impact planning outcomes. Beyond modeling uncertainty within optimization frameworks, additional tools from risk analysis [50] and portfolio theory [51] can support higher-level planning decisions by quantifying exposure and guiding allocation strategies. Portfolio-based approaches treat servicing targets as investments with uncertain returns (e.g., mission value, system health) and risks (e.g., failure likelihood, service cost variability). When planning schedules across multiple satellites or campaigns, such methods can support:

- *Asset allocation:* Prioritize servicing tasks across satellites based on expected benefit and risk contribution.
- *Diversification:* Balance mission portfolios to avoid overcommitment to targets with correlated failure risks or uncertain performance.
- *Risk budgeting:* Constrain cumulative mission exposure (e.g., total fuel at risk, failure probabilities) to remain within operational limits.

Complementary to these resource allocation tools, probabilistic risk assessment techniques—such as fault tree and event tree analysis—can help identify and quantify failure paths that may disrupt scheduling feasibility or mission continuity [50]. Such analysis can support resilient planning strategies.

5.3. Real Options Analysis

Many infrastructure and architectural decisions in OOS mission planning involve long-term commitments under uncertainty—such as whether to deploy additional servicers, activate orbital depots, or reconfigure mission capabilities. Real options analysis provides a framework to quantify the value of flexibility in such decisions [52], treating them as financial-style options that can be exercised when conditions become favorable.

Table (10) summarizes representative option types that can be relevant to the OOS mission planning context, particularly for infrastructure planning and adaptive mission architecture.

Table 10: Illustrative Real Options in OOS Infrastructure Planning

Option Type	**OOS Application**
Option to Defer	Postpone servicer or depot deployment until servicing demand is confirmed
Option to Expand	Add modular servicing capacity as demand grows
Option to Abandon	Repurpose or decommission assets if target constellations are retired
Option to Switch	Alternate between servicing modes (e.g., refueling vs. repair) as mission needs evolve

In practice, real options can be implemented by modeling future decisions as contingent on the resolution of uncertainty. This typically involves identifying decision points (e.g., whether to deploy a servicer), estimating the value of different outcomes under uncertain scenarios, and applying option valuation techniques such as decision trees, binomial lattices, or Monte Carlo simulations. These models can be integrated with stochastic or scenario-based planning frameworks to evaluate the expected value of flexibility and compare alternative strategies [53].

5.4. Methods Integration and Outlook

The complementary methods discussed above can be integrated into layered decision frameworks that address different levels of OOS mission planning. For instance, MCDA can inform strategic prioritization across mission options, whose outputs feed into optimization-based scheduling models (e.g. MILP) at the operational level. Risk-based portfolio approaches and real options analysis can be incorporated to evaluate resource allocation and infrastructure flexibility under uncertainty, complementing deterministic or stochastic planning models.

Future OOS planning frameworks may also incorporate interactions among multiple stakeholders. Game-theoretic models, for example, can help capture competitive dynamics or coordinated investment strategies among service providers and satellite operators. Similarly, queuing models may support real-time service scheduling by modeling task arrival patterns and servicing capacity constraints, offering tools for balancing responsiveness and efficiency in dynamic operational environments.

6. Conclusion

On-orbit servicing (OOS) introduces a wide range of planning and decision-making challenges, from long-term infrastructure development to responsive mission execution in dynamic orbital environments. This paper outlined how core operations research (OR) methodologies can help structure and support these challenges across different levels of complexity and time horizons. The discussion focused on four broad families of models—deterministic optimization, stochastic programming, sequential decision-making, and complementary decision analysis tools—that together offer a diverse and adaptable toolkit for mission planners and system architects. Rather than advancing new algorithms or prescribing detailed solutions, the aim has been to synthesize and contextualize representative techniques from the OR domain, how they align with the distinctive features of OOS mission planning problems. Taken together, these approaches offer a flexible set of tools for the strategic management of OOS systems, enabling planners to better navigate operational and long-term system-level trade-offs.

Looking ahead, a key challenge lies in bridging high-level planning with low-level operational constraints for OOS missions, particularly in modeling proximity operations within mission planning scenarios. The disparity in temporal resolution, combined with limited empirical data from real OOS missions, poses difficulties for model validation and integration. Addressing these gaps will be important to ensure that strategic models remain grounded in operational feasibility.

Acknowledgments

This work was supported by the *Agence Nationale de la Recherche (ANR)*, France, under the project **ANR-23-CE10-0006** (Resilient and Sustainable Planning and Management of Future Space Industry Infrastructure with On-Orbit Servicing), for which Adam Abdin serves as the Principal Investigator.

References

[1] Patrick Rousso, Sanaz Samsam, and Robin Chhabra. A Mission Architecture for On-Orbit Servicing Industrialization. In *2021 IEEE Aerospace Conference (50100)*, pages 1–14, March 2021. ISSN: 1095-323X.

[2] Koki Ho. Space Logistics Modeling and Optimization: Review of the State of the Art. *Journal of Spacecraft and Rockets*, 61(5):1417–1427, 2024.

[3] Hang Woon Lee, David O Williams Rogers, Brycen D Pearl, Hao Chen, and Koki Ho. Deterministic multistage constellation reconfiguration using integer programming and sequential decision-making methods. *Journal of Spacecraft and Rockets*, 62(1):206–222, 2025.

[4] Koki Ho, Yuri Shimane, and Masafumi Isaji. Generalizing space logistics network optimization with integrated machine learning and mathematical programming. *Journal of Spacecraft and Rockets*, pages 1–5, 2024.

[5] Peng Han, Yanning Guo, Chuanjiang Li, Hui Zhi, and Yueyong Lv. Multiple GEO satellites on-orbit repairing mission planning using large neighborhood search-adaptive genetic algorithm. *Advances in Space Research*, 70(2):286–302, July 2022.

[6] Yuri Shimane, Nicholas Gollins, and Koki Ho. Orbital facility location problem for satellite constellation servicing depots. *Journal of Spacecraft and Rockets*, 61(3):808–825, 2024.

[7] Susan E Sorenson and Sarah G Nurre Pinkley. Multi-orbit routing and scheduling of refuellable on-orbit servicing space robots. *Computers & Industrial Engineering*, 176:108852, 2023.

[8] Weikui Liang, Hui Zhi, Peng Han, Guangtao Ran, Guangfu Ma, and Yanning Guo. Geo satellite on-orbit refueling and debris removal hybrid mission planning under uncertainty. *Advances in Space Research*, 74(5):2376–2387, 2024.

[9] Tristan Sarton du Jonchay, Hao Chen, Onalli Gunasekara, and Koki Ho. Framework for modeling and optimization of on-orbit servicing operations under demand uncertainties. *Journal of Spacecraft and Rockets*, 58(4):1157–1173, 2021.

[10] Laura Medioni, Yvan Gary, Myrtille Monclin, Côme Oosterhof, Gaetan Pierre, Tom Semblanet, Perrine Comte, and Kévin Nocentini. Trajectory optimization for multi-target active debris removal missions. *Advances in Space Research*, 72(7):2801–2823, 2023.

[11] Xinyue Hu, Bin Wang, Xin Xiong, Zeyi Sun, Zhe Feng, and Huaiping Jin. On-orbit Service Mission Fast Autonomous Planning Method Based on Meta-reinforcement Learning. In *2024 43rd Chinese Control Conference (CCC)*, pages 2195–2200, July 2024. ISSN: 1934-1768.

[12] Susmitha Patnala and Adam Abdin. On-orbit servicing for spacecraft collision avoidance with autonomous decision making. In *Proceedings of SPAICE2024: The First Joint European Space Agency/IAA Conference on AI in and for Space*, pages 487–492, 2024.

[13] Agni Bandyopadhyay and Günther Waxenegger-Wilfing. Revisiting space mission planning: A reinforcement learning-guided approach for multi-debris rendezvous. In *2024 International Conference on Space Robotics (iSpaRo)*, pages 104–110. IEEE, 2024.

[14] Antoine Poupon, Hugo de Rohan Willner, Pierre Nikitits, and Adam Abdin. Ai-driven risk-aware scheduling for active debris removal missions. In *Proceedings of SPAICE2024: The First Joint European Space Agency/IAA Conference on AI in and for Space*, pages 340–344, 2024.

[15] Antonio López Rivera, Lucrezia Marcovaldi, Jesús Ramírez, Alex Cuenca, and David Bermejo. Design and optimization of multi-rendezvous manoeuvres based on reinforcement learning and convex optimization. *arXiv e-prints*, pages arXiv–2411, 2024.

[16] Ignacio Betti Lhuillier, Andrea Bellome, and Leonard Felicetti. On-orbit Servicing Strategies for Large Constellations via Hybrid Dynamic Programming. In *Proceedings of the 2024 CEAS EuroGNC Conference*, Bristol, UK, June 2024. CEAS-GNC-2024-011.

[17] Koki Ho, Olivier L De Weck, Jeffrey A Hoffman, and Robert Shishko. Dynamic modeling and optimization for space logistics using time-expanded networks. *Acta Astronautica*, 105(2):428–443, 2014.

[18] Max Bannach, Giacomo Acciarini, and Dario Izzo. On the Keplerian TSP and VRP: Benchmarks and Encoding Techniques. In *International Astronautical Congress*, 2024.

[19] Gurobi Optimization, LLC. *Gurobi Optimizer Reference Manual*, 2023.

[20] IBM. *IBM ILOG CPLEX Optimization Studio CPLEX User's Manual*, 2023.

[21] Julia A Tepper, Youssof Fassi, Tristan Sarton du Jonchay, Koki Ho, and Yuri Shimane. Optimization and Modeling of Active Debris Removal using a Time-expanded Network. In *ASCEND 2022*, page 4245. 2022.

[22] R. Rahmaniani, T. G. Crainic, M. Gendreau, and W. Rei. The Benders decomposition algorithm: A literature review. *European Journal of Operational Research*, 259(3):801–817, 2017.

[23] Martin WP Savelsbergh. Branch and price: Integer programming with column generation. In *Encyclopedia of Optimization*, pages 1–5. Springer, 2024.

[24] Laurence A Wolsey. *Integer programming*. John Wiley & Sons, 2020.

[25] Laura Medioni, Yvan Gary, Myrtille Monclin, Côme Oosterhof, Gaetan Pierre, Tom Semblanet, Perrine Comte, and Kévin Nocentini. Trajectory optimization for multi-target Active Debris Removal missions. *Advances in Space Research*, 72(7):2801–2823, 2023.

[26] Jin Haeng Choi, Chandeok Park, and Jinah Lee. Mission planning for active removal of multiple space debris in low Earth orbit. *Advances in Space Research*, 73(9):4800–4812, 2024.

[27] J. F. Castet and J. H. Saleh. Satellite and satellite subsystems reliability: Statistical data analysis and modeling. *Reliability Engineering System Safety*, 95(7):624–640, 2010.

[28] John R Birge and Francois Louveaux. *Introduction to stochastic programming*. Springer Science & Business Media, 2011.

[29] A. Shapiro, D. Dentcheva, and A. Ruszczynski. *Lectures on Stochastic Programming: Modeling and Theory*. Society for Industrial and Applied Mathematics, 3rd edition, 2021.

[30] A. J. Kleywegt, A. Shapiro, and T. Homem-de Mello. The sample average approximation method for stochastic discrete optimization. *SIAM Journal on Optimization*, 12(2):479–502, 2002.

[31] R. Rahmaniani, T. G. Crainic, M. Gendreau, and W. Rei. The Benders decomposition algorithm: A literature review. *European Journal of Operational Research*, 259(3):801–817, 2017.

[32] Fotios Petropoulos, Gilbert Laporte, Emel Aktas, Sibel A Alumur, Claudia Archetti, Hayriye Ayhan, Maria Battarra, Julia A Bennell, Jean-Marie Bourjolly, John E Boylan, et al. Operational Research: methods and applications. *Journal of the Operational Research Society*, 75(3):423–617, 2024.

[33] Arthur M Geoffrion. Lagrangean relaxation for integer programming. In *Approaches to integer programming*, pages 82–114. Springer, 2009.

[34] Hao Chen, Arthur Lapin, Takaya Ukai, Chao Lei, and Koki Ho. Optimization for Large-scale Multi-mission Space Campaign Design by Approximate Dynamic Programming. In *2018 AIAA SPACE and Astronautics Forum and Exposition*, page 5287, 2018.

[35] M. L. Puterman. *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. John Wiley & Sons, 2014.

[36] R. S. Sutton and A. G. Barto. *Reinforcement Learning: An Introduction*. MIT Press, 2nd edition, 2018.

[37] Jesse Clifton and Eric Laber. Q-learning: Theory and applications. *Annual Review of Statistics and Its Application*, 7(1):279–301, 2020.

[38] Jürgen Schmidhuber. Deep learning in neural networks: An overview. *Neural networks*, 61:85–117, 2015.

[39] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, et al. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529–533, 2015.

[40] H. Van Hasselt, A. Guez, and D. Silver. Deep reinforcement learning with double Q-learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 30, 2016.

[41] Ziyu Wang, Tom Schaul, Matteo Hessel, Hado Hasselt, Marc Lanctot, and Nando Freitas. Dueling network architectures for deep reinforcement learning. In *International Conference on Machine Learning*, pages 1995–2003. PMLR, 2016.

[42] M. Hessel, J. Modayil, H. Van Hasselt, T. Schaul, G. Ostrovski, W. Dabney, D. Horgan, B. Piot, M. Azar, and D. Silver. Rainbow: Combining improvements in deep reinforcement learning. In *AAAI Conference on Artificial Intelligence*, 2018.

[43] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

[44] J. Schulman, S. Levine, P. Abbeel, M. Jordan, and P. Moritz. Trust region policy optimization. In *International Conference on Machine Learning*, pages 1889–1897, 2015.

[45] Ivo Grondman, Lucian Busoniu, Gabriel AD Lopes, and Robert Babuska. A survey of actor-critic reinforcement learning: Standard and natural policy gradients. *IEEE Transactions on Systems, Man, and Cybernetics, part C (applications and reviews)*, 42(6):1291–1307, 2012.

[46] Salvatore Greco, Matthias Ehrgott, and José Rui Figueira, editors. *Multiple Criteria Decision Analysis: State of the Art Surveys*, volume 2 of *International Series in Operations Research & Management Science*. Springer, New York, NY, 2 edition, 2016. eBook published: 18 February 2016.

[47] Martin Aruldoss, T Miranda Lakshmi, and V Prasanna Venkatesan. A survey on multi criteria decision making methods and its applications. *American Journal of Information Systems*, 1(1):31–43, 2013.

[48] Thomas L Saaty. Decision making with the analytic hierarchy process. *International journal of services sciences*, 1(1):83–98, 2008.

[49] Denis Bouyssou. *Evaluation and decision models: a critical perspective*, volume 32. Springer Science & Business Media, 2000.

[50] Terje Aven. Risk assessment and risk management: Review of recent advances on their foundation. *European*

journal of operational research, 253(1):1–13, 2016.

[51] Juuso Liesiö, Ahti Salo, Jeffrey M Keisler, and Alec Morton. Portfolio decision analysis: Recent developments and future prospects. *European Journal of Operational Research*, 293(3):811–825, 2021.

[52] Lenos Trigeorgis and Jeffrey J Reuer. Real options theory in strategic management. *Strategic management journal*, 38(1):42–63, 2017.

[53] Michel-Alexandre Cardin. Enabling flexibility in engineering systems: a taxonomy of procedures and a design framework. *Journal of Mechanical Design*, 136(1):011005, 2014.