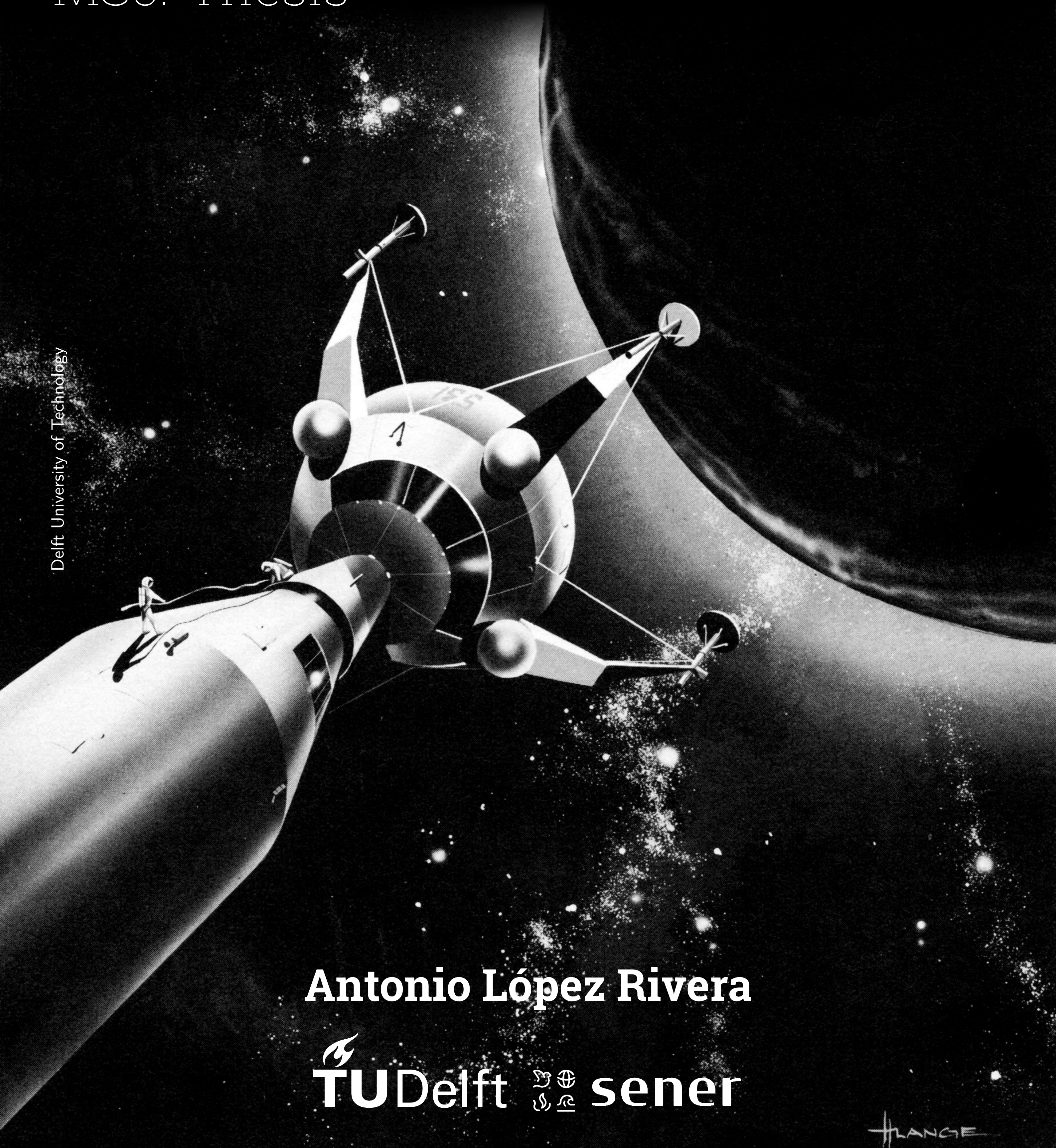
Reinforcement Learning for Multi-Rendezvous Mission Design

MSc. Thesis

Delft University of Technology



Antonio López Rivera

TU Delft sener

This page is intentionally left blank.

Reinforcement Learning for Multi-Rendezvous Mission Design

by

Antonio López Rivera

To obtain the degree of Master of Science
at the Delft University of Technology
to be defended publicly on Thursday, October the 31st, 2024

Supervisor:	Marc Naeije	TU Delft Aerospace Engineering
Co-supervisor:	Jesús Ramírez	SENER Aerospace & Defence
Project Duration:	February, 2024 - October, 2024	
Faculty:	Faculty of Aerospace Engineering, Delft	

Cover: Harry H.K. Lange, Conquering the Sun's Empire, © 1963
Style: Own work, based on the TU Delft house style

This page is intentionally left blank.

To my family.

Abstract

The design of multi-target rendezvous trajectories, which see a spacecraft approaching a sequence of objects in orbit as efficiently (by some metric) as possible, is a challenging problem of critical importance for Active Debris Removal (ADR), On-Orbit Servicing (OOS) and cis-Lunar logistics more widely. This thesis investigates two primary challenges in space Vehicle Routing Problems (VRPs): the application of Neural Combinatorial Optimization (NCO) methods for ADR missions and the integration of verifiable trajectory optimization techniques for OTV payload deployment.

The first research focus assesses the efficacy of NCO methods in designing multi-target rendezvous trajectories for ADR missions. An Attention-based routing policy, comprising a Graph Attention Network and a Pointer Network, was developed and trained using Reinforcement Learning (RL) algorithms, including REINFORCE, Advantage Actor-Critic (A2C), and Proximal Policy Optimization (PPO). Through hyperparameter analysis utilizing ANOVA, embedding dimension and the number of encoder layers were identified as critical factors influencing model performance. The trained policy was evaluated on scenarios involving 10, 30, and 50 transfers based on the Iridium 33 debris cloud. In missions with 10 transfers, the NCO policy achieved a mean optimality gap of 32%, outperforming the Dynamic RAAN Walk (DRW) heuristic in both mission cost and runtime. However, performance degraded in more complex scenarios with 30 and 50 transfers, indicating limited generalization beyond the training conditions. Grid search hyperparameter optimization revealed that while model performance improves with increased complexity, gains are marginal, and larger training datasets enhance convergence speed with only slight improvements in final performance. These findings demonstrate that NCO methods are effective for ADR missions with a limited number of targets but face scalability and generalization challenges in more complex scenarios.

The second research focus involves the design and optimization of multi-rendezvous trajectories for the UARX Space OSSIE mission using a modular framework that integrates Heuristic Combinatorial Optimization (HCO) with Sequential Convex Programming (SCP). This framework successfully determined optimal target sequences and generated near fuel-optimal trajectories for OSSIE, a translational and mass-dynamic payload delivery platform. An Attention-based routing policy trained with RL was integrated into the combinatorial optimization process, enhancing the efficiency of mission planning. Applied to the OSSIE mission, the framework effectively explored the mission design space, optimizing 5000 mission scenarios and affirming the vehicle's capability to fulfill advertised services. The modularity of the framework ensures adaptability to mission-specific constraints and facilitates future extensions, such as the incorporation of low-thrust propulsion profiles.

Overall, this thesis confirms that NCO methods are applicable and effective for specific instances of space VRPs, particularly in optimizing ADR missions with a limited number of targets and in near-static mission scenarios where RAAN convergence is not required. The integration of verifiable trajectory optimization techniques with advanced routing policies presents a viable approach for efficient and adaptable mission planning. However, scalability and generalization remain challenges that necessitate further research. Recommendations include refining NCO model architectures to enhance scalability and generalization, exploring hybrid approaches that combine NCO with traditional heuristics, and developing automated machine learning frameworks to optimize model performance and robustness.

The project successfully achieved its primary objectives: developing and implementing heuristic and neural combinatorial optimization solvers for space VRPs, designing a modular trajectory optimization framework, and conducting comprehensive mission analyses for the OSSIE OTV. In doing so it has increased the mission design capabilities for space logistics missions at SENER Aerospace & Defence, as well as a provided a strong foundation for future research and development aimed at addressing the increasing complexities of space operations.

Preface

I would like to extend my deepest gratitude to Marc Naeije for his support and for his guidance through this project, for giving me the freedom to explore widely and deeply —even at my peril— and for his continued concern for my well being. Second, but not in importance, I would like to thank Jesús Ramírez for his unwavering commitment and for our thoughtful discussions. To both I owe a great stake of who I am as an engineer today.

I would like to extend my thanks to Lucrezia Marcovaldi, Álex Cuenca and David Bermejo for their comradery, their advice and their commitment to the project, to Miguel Martínez de Bujo for taking me under his wing when I first arrived at SENER, and to Mercedes Ruíz for granting me the opportunity to conduct this research at SENER Aerospace & Defence.

This thesis concludes my time in Delft 6 years after I first arrived to study Aerospace Engineering. I was a different person, and it was a different world. In Delft I met some of the people I'll be indebted to for the rest of my life. Thank you. Thank you Victoria, for always being there. Thank you to Román Chiva, to Francesco Branca, to Alejandro Montoya, to Martín Camus, to Marta Bermúdez and Anibal Pastinante, to Neil Richez and Silvio Peressini, to Daniel Stutman, and to you —you know who you are—. What I'd do without you girls, I don't know.

And lastly, I would like to thank my family. I truly would not be here without you. I owe you everything, I will be forever touched that you trusted a fool like me to embark on this adventure, and I will be there for you always. Gracias Tere. Gracias Yago. Gracias mamá. Y gracias, papá.

Gracias José, gracias Ángel, gracias Picu, y gracias, Toti. Gracias Riveras y Fernández, gracias López y Peláez. Y Alicia, Álvaro y Alejo: esto es para vosotros.

Per aspera, ad astra.

Nomenclature

Abbreviations

Abbreviation	Meaning
A2C	Advantage Actor-Critic
A3C	Asynchronous Advantage Actor-Critic
ADR	Active Debris Removal
ANOVA	Analysis of Variance
AOP	Argument of Perigee
CO	Combinatorial Optimization
DNN	Deep Neural Network
DRW	Dynamic RAAN Walk
ECI	Earth-Centered Inertial frame
FYS	Fisher-Yates Shuffle
GAT	Graph Attention Network
GNN	Graph Neural Network
GPU	Graphics Processing Unit
IPC	Impulsive Plane Change
JIT	Just-In-Time compilation
KS	Knuth's Algorithm for Uniform Permutation Sampling using Sobol Points
LFC	Lyapunov Feedback Control
LEO	Low Earth Orbit
LVLH	Local-Vertical Local-Horizontal frame
MEE	Modified Equinoctial Elements
MDP	Markov Decision Process
MEO	Medium Earth Orbit
MHT	Multiple Hohmann Transfer
MINLP	Mixed-Integer Nonlinear Programming
ML	Machine Learning
NCO	Neural Combinatorial Optimization
NN	Nearest-Neighbour Search
NIC	Nodal Inclination Change maneuver

Continued on next page

Abbreviation	Meaning
OOS	On-Orbit Servicing
OSSIE	UARX Orbit Solutions to Simplify Injection and Exploration, UARX Space OTV
OTV	Orbital Transfer Vehicle
PN	Pointer Network
PPO	Proximal Policy Optimization
Q-Law	Lyapunov Feedback Control Law based on the proximity quotient Q
REINFORCE	Monte Carlo Policy Gradient Algorithm
RL	Reinforcement Learning
RQ-Law	Rendezvous Q-Law
RSW	Radial, Along-track (S), Cross-track (W) frame
SCP	Sequential Convex Programming
RCS	Radar Cross-Section
SHP	Sobol Hypercube Permutations
SMA	Semi-major Axis
SP	Sobol Permutations
STSP	Spacecraft Traveling Salesman Problem
TOF	Time of Flight
TSP	Traveling Salesman Problem
VRP	Vehicle Routing Problem

Mathematical Notation

Notation	Meaning
r	Scalar variable
$\mathbf{r}$	Vector variable
$\|\mathbf{r}\|$	Euclidean norm of vector $\mathbf{r}$
$\overline{r}$	Mean value of r
$r \sim P$	r is sampled from distribution P
$\mathbb{E}[X]$	Expectation of random variable X
∇Q	Gradient of function Q
∇_θ	Gradient with respect to θ
∇_ϕ	Gradient with respect to ϕ
$\dot{Q}$	Time derivative of Q
$\lceil x \rceil$	Ceiling function of x
$\min(a, b)$	Minimum of a and b
$\vert x \vert$	Absolute value of x

Continued on next page

Notation	Meaning
$\text{argsort}(z)$	Indices that would sort vector z
$\text{clip}(x, a, b)$	Clipping function limiting x to the interval $[a, b]$
$\sin(x), \cos(x), \tan^{-1}(x)$	Trigonometric functions
$B(p, q)$	Beta function
$F_j^{-1}(x)$	Inverse cumulative distribution function
$\mathbb{R}^n$	n-dimensional real space
$\mathbb{S}^{d-1}$	Unit sphere in $\mathbb{R}^d$
$\mathfrak{S}^n$	Symmetric group of permutations of n elements
$\pi_\theta(a_t\|s_t)$	Policy probability of action a_t given state s_t under parameters θ
$\pi_{\theta_{\text{old}}}(a_t\|s_t)$	Policy with old parameters θ_{old}
$r_t(\theta)$	Probability ratio at time t
$r^{(i)}(\theta)$	Probability ratio for instance i
τ	Trajectory (sequence of states, actions, rewards)

Latin Symbols

Symbol	Meaning	Units/Value
a	Semi-major axis	m
a_T	Target semi-major axis	m
A	Reference area; advantage function	m^2; –
A_t	Advantage at time t	–
$A^{(i)}$	Advantage for instance i	–
a_t	Action at time t	–
$a^{(i)}$	Solution for instance i	–
$a_{J_{2,r}}, a_{J_{2,\theta}}, a_{J_{2,\phi}}$	Components of acceleration due to J_2 perturbation	$\text{m}\,\text{s}^{-2}$
c	Constant or parameter	–
C_D	Drag coefficient	–
d	Dimension	–
e	Eccentricity	–
e_T	Target eccentricity	–
f, g, h, k	Equinoctial elements	–
f_{burn}	Burn frequency	s^{-1}
G	Universal gravitational constant	$6.674\,30\text{e}{-11}\,\text{m}^3\,\text{kg}^{-1}\,\text{s}^{-2}$
G_t	Return at time t	–
$G^{(i)}$	Return for instance i	–
g_0	Standard Earth gravity	$9.806\,65\,\text{m}\,\text{s}^{-2}$

Continued on next page

Symbol	Meaning	Units/Value
i	Index; inclination	–; rad
i_1, i_2	Inclinations of initial and target orbits	rad
I_{sp}	Specific Impulse	–
J_2	Second zonal harmonic coefficient	1.0826e−3
j	Index	–
k	Number of burns; scaling parameter	–
k_d, k_c	Number of burns for departure and circularization	–
k_{NIC}	Number of burns for NIC	–
k_{IPC}	Number of burns for IPC	–
L	True longitude; loss function	rad; –
L_t	Clipped objective at time t	–
$L^{(i)}$	Clipped objective for instance i	–
L_{policy}	Policy loss	–
L_{value}	Value function loss	–
L_0, L_1	Initial and target true longitude	rad
m	Spacecraft mass	kg
m_0	Initial mass	kg
m_f	Fuel mass	kg
m_1, m_2	Masses of bodies 1 and 2	kg
n	Orbital mean motion	s^{-1}
$\hat{\mathbf{e}}_r, \hat{\mathbf{e}}_\theta, \hat{\mathbf{e}}_\phi$	Basis vectors in LVLH frame	–
$\hat{\mathbf{u}}$	Thrust direction unit vector	–
$\overline{P}_{\text{MHT}}$	Average orbital period during MHT	s
p	Semi-latus rectum	m
P	Orbital period	s
Π	Set of permutations	–
r	Radial distance; scalar variable	m; –
r_0	Radius of initial orbit	m
r_1	Radius of target orbit	m
r_p	Periapsis radius	m
$r_{p_{\text{min}}}$	Minimum periapsis radius	m
$R(x^{(i)}, a^{(i)})$	Reward function for instance i	–
s^2	Variable in Gauss Variational Equations	–
s_t	State at time t	–
T	Thrust magnitude; time horizon	N; –
T_{burn}	Burn time	s
T_{cooldown}	Cooldown time	s

Continued on next page

Symbol	Meaning	Units/Value
t	Time index	–
t_0	Start time	s
$\mathcal{T}$	Best time-to-go	s
v	Variable in Gauss Variational Equations; velocity	–; $\mathrm{m\,s^{-1}}$
v_{eq}	Exhaust velocity	$\mathrm{m\,s^{-1}}$
V_0	Orbital velocity	$\mathrm{m\,s^{-1}}$
$V_\phi(s_t)$	Estimated value function at state s_t	–
$V_\phi(x^{(i)})$	Estimated value function for instance $x^{(i)}$	–
w	Variable in Gauss Variational Equations	–
x	Vector in $\mathbb{R}^n$; problem instance	–
$x^{(i)}$	Problem instance i	–
$\mathbf{a}_D$	Acceleration due to drag	$\mathrm{m\,s^{-2}}$
$\mathbf{a}_{J_2}$	Acceleration due to J_2 perturbation	$\mathrm{m\,s^{-2}}$
$\mathbf{F}_{1,2}$	Gravitational force between bodies 1 and 2	N
$\mathbf{r}_{1,2}$	Position vector from body 1 to body 2	m
$r^{(i)}(\theta)$	Probability ratio for instance i	–
$r_t(\theta)$	Probability ratio at time t	–
$\mathbf{v}$	Velocity vector	$\mathrm{m\,s^{-1}}$
$\dot{m}_f$	Mass flow rate	$\mathrm{kg\,s^{-1}}$
$W_x, W_p, W_{\text{œ}}, W_L, W_{\text{scl}}$	Weights in Q-law and RQ-law	–
œ	Modified Equinoctial Elements (a, f, g, h, k)	–
œ_T	Target orbital elements	–
$\text{œ}_{T,\text{aug}}$	Augmented target elements	–

Greek Symbols

Symbol	Meaning	Units/Value
α	Learning rate for policy	–
β	Learning rate for value function	–
γ	Relative inclination	rad
δ	Small change or variation	–
ϵ	Clipping parameter	–
μ	Earth's gravitational parameter	3.986e14 $\mathrm{m^3\,s^{-2}}$
ξ	Ratio of orbit radii (r_1/r_0)	–
ρ	Atmospheric density	$\mathrm{kg\,m^{-3}}$
θ	Policy parameters; true anomaly	–; rad

Continued on next page

Symbol	Meaning	Units/Value
θ_{old}	Previous policy parameters	–
ϕ	Value function parameters; spherical coordinates	–
$\pi_\theta(a_t \mid s_t)$	Policy probability of action a_t given state s_t under parameters θ	–
$\pi_{\theta_{\text{old}}}(a_t \mid s_t)$	Policy with old parameters θ_{old}	–
σ	True anomaly	rad
τ	Trajectory (sequence of states, actions, rewards)	–
ω	Argument of Perigee	rad
Ω	Right Ascension of the Ascending Node	rad
Ω_1, Ω_2	RAAN of initial and target orbits	rad
Ψ	Transformation matrix in MEEs	–
Δi	Change in inclination	rad
ΔV	Delta-V (change in velocity)	$\text{m}\,\text{s}^{-1}$
ΔV_{MHT}	Total delta-V for MHT	$\text{m}\,\text{s}^{-1}$
ΔV_d	Departure delta-V	$\text{m}\,\text{s}^{-1}$
ΔV_c	Circularization delta-V	$\text{m}\,\text{s}^{-1}$
ΔV_{NIC}	Delta-V for NIC	$\text{m}\,\text{s}^{-1}$
ΔV_{IPC}	Delta-V for IPC	$\text{m}\,\text{s}^{-1}$
$\Delta_r, \Delta_t, \Delta_n$	Perturbing accelerations in radial, tangential, normal directions	$\text{m}\,\text{s}^{-2}$
$\Delta L_{[-\pi,\pi]}$	Difference in true longitude wrapped to $[-\pi, \pi]$	rad

Subscripts

Subscript	Meaning
t	At time step t
(i)	For instance i
r	Radial component
n	Normal component
θ	Tangential component
ϕ	Normal component (in LVLH frame)
I	Inertial frame
s	Spacecraft
C	Central body (Earth)
old	Previous parameter value
0	Initial value
T	Target value
MHT	Related to Multiple Hohmann Transfer

Continued on next page

Subscript	Meaning
NIC	Related to Nodal Inclination Change
IPC	Related to Impulsive Plane Change
burn	Burn time
cooldown	Cooldown time
ECI	Earth-Centered Inertial frame
MEE	Modified Equinoctial Elements frame
D	Related to drag
J_2	Related to J_2 perturbation

Reinforcement Learning Jargon Terms

Term	Symbol	Definition
Monte Carlo Policy Gradient Method		An approach that estimates policy gradients using complete sampled trajectories from the environment without requiring knowledge of the environment's dynamics [1].
Gradient Ascent		An optimization technique that adjusts parameters in the direction of the positive gradient to maximize a function.
Policy Parameters	θ	The set of parameters that define the policy π_θ.
Policy	$\pi_\theta(a\|s)$	A mapping from states s to a probability distribution over actions a, parameterized by θ.
Trajectory	τ	A sequence of states, actions, and rewards observed when an agent interacts with the environment: $\tau = \{s_0, a_0, r_0, \ldots, s_T, a_T, r_T\}$.
State	s_t	The representation of the environment at time step t.
Action	a_t	The decision made by the agent at time step t.
Reward	r_t	The immediate scalar feedback received after taking action a_t in state s_t.
Return	G_t	The cumulative future reward from time t onwards, defined as $G_t = \sum_{k=t}^{T} r_k$.
Expected Return	$\mathbb{E}[G_t\|s_t]$	The expected cumulative reward obtained by following a policy π_θ from a given state s_t, defined as $\mathbb{E}[G_t\|s_t] = \mathbb{E}\left[\sum_{k=t}^{T} r_k \middle\| s_t\right]$.
Gradient of the Log-Likelihood	$\nabla_\theta \log \pi_\theta(a_t\|s_t)$	The derivative of the log-probability of taking action a_t in state s_t with respect to the policy parameters θ.
Policy Gradient	$\nabla_\theta J(\theta)$	The gradient of the expected return with respect to the policy parameters θ.

Continued on next page

Term	Symbol	Definition
Policy Rollout		The process of generating trajectories by following the policy in the environment.
Convergence		The point at which the policy parameters θ stabilize and further updates become negligible.
Variance		A measure of dispersion in the estimates of the policy gradient, affecting the stability and speed of learning.
Baseline	$V_\phi(s_t)$	A function subtracted from the return G_t to reduce the variance of the policy gradient estimate without introducing bias [1].
Actor		The component of the algorithm that represents the policy π_θ, responsible for selecting actions.
Critic		The component that estimates the value function $V_\phi(s)$, providing feedback to the actor.
Value Function	$V_\phi(s)$	A function estimating the expected return from state s, parameterized by ϕ.
Advantage Function	A_t	A measure of how much better or worse an action a_t is compared to the critic's estimated value of state s_t, defined as $A_t = G_t - V_\phi(s_t)$.
Minimize the Squared Error		The process of updating the value function parameters ϕ to reduce the difference between estimated and actual returns.
Variance Reduction		Techniques used to decrease the variance in policy gradient estimates, improving learning stability.
Surrogate Objective Function		An objective function that approximates the true objective but is designed to be more tractable or stable during optimization.
Clipping Mechanism	ϵ	A method to limit the change in the probability ratios during policy updates to within a specified range, controlled by the clipping parameter ϵ.
Probability Ratio	$r_t(\theta)$	The ratio of the probabilities of an action under the new and old policies, defined as $r_t(\theta) = \frac{\pi_\theta(a_t\|s_t)}{\pi_{\theta_{\text{old}}}(a_t\|s_t)}$.
Clipped Surrogate Objective	L_t	An objective function that includes the minimum of the unclipped and clipped policy objectives to prevent large deviations in policy updates.
Exploration and Exploitation		The trade-off between trying new actions to discover their effects (exploration) and using known actions that yield high rewards (exploitation).

List of Figures

List of Tables

List of Algorithms

Contents

1 Introduction

The design of multi-target rendezvous manoeuvres, which see a spacecraft approaching a sequence of objects in orbit as efficiently (by some metric) as possible, has seen a considerable surge in interest in recent years for the purposes of Active Debris Removal (ADR) missions [2]–[7] to tackle the space debris problem [8], [9], as well as On-Orbit Servicing (OOS) missions [4], [10], [11] and advanced space logistics concepts [12].

1.1 The Space Traveling Salesman Problem

The problem of designing such trajectories, known as the Space Traveling Salesman Problem (STSP), is an example of a Mixed Integer Non-Linear Programming (MINLP) problem with factorial complexity over the number of targets. MINLP problems are notoriously difficult to approach. An optimal solution to the STSP consists of the optimal sequence in which to visit a set of targets and the optimal (by some metric) transfer trajectory between each target in the optimal sequence. The STSP is conceptually related to the classical Traveling Salesman Problem (TSP), with the added complexities inherent to the space environment: notably, a 6-dimensional non-Euclidean state space, mass dynamics, spacecraft propulsion constraints, and the progressive drift of the states of orbiting bodies due to secular perturbations, chiefly J_2 for Earth-orbiting spacecraft.

Formally, the STSP is the problem of finding a minimum weight path (if the spacecraft must end the tour back at its initial state, a Hamiltonian path) in a complete weighted graph $G \coloneqq \{\mathcal{V}(t), \mathbf{W}(\pi)\}$, where $\mathcal{V}(t)$ is the set of graph vertexes (targets, the state of which drifts over time) and $\mathbf{W}(\pi) \coloneqq \mathcal{V} \times \mathcal{V} \to \mathbb{R}^+$ is a map that associates an edge weight (a transfer cost) to each ordered vertex pair [2], and may depend on the sequence π in which the targets are visited. One such case is when payload mass is a large percentage of the spacecraft's wet mass, and thus deployment sequence has a non-negligible impact on fuel consumption. A standard approach to solve the STSP is Benders decomposition [4]–[7], where the MINLP problem is divided into a higher-level Combinatorial Optimization (CO) problem and a lower-level trajectory optimization problem. A transfer cost estimator is then used to calculate the cumulative cost of tours in the CO problem. Transfer cost estimators may be database-dependent [13], [14], database-independent (analytical), or learning-based [15].

State-of-the-art CO methods fall in two camps: exact methods and heuristic methods, which are less costly and can produce near-optimal results, but cannot offer optimality guarantees whatsoever [2], [16]. Exact methods based on tree searches are the norm for highly complex, large STSP variants; all winning submissions of the Global Trajectory Optimization Competitions have made use of tree search approaches [2], [13], [17]. Heuristic optimization methods however are an attractive option to solve smaller STSP instances (up to hundreds of targets [2]) due to their capacity to achieve near-optimal results with lower computational cost [2], and are widely applied in literature to tackle multi-rendezvous mission design [2]–[5], [7]. Heuristic optimization methods have also been successfully applied to complex STSP instances where the cost of exact approaches is unfeasible [13]. High quality approximate solutions are highly desirable for the heuristic optimization process. As the complexity of the generalized Vehicle Routing Problem (VRPs) increases, high quality approximate solutions become more difficult to obtain [18]. This bodes ill for the field of space logistics, as the complexity of space VRPs beyond the STSP is bound to increase over time: this will happen as Low Earth Orbit (LEO) and Medium Earth Orbit (MEO) become more congested, the in-space manufacturing and servicing industries rise, and space logistics operations become more complex [19]. It becomes pressing to ask whether learning-based methods from the field of CO could be applied in the space domain.

Machine Learning (ML) approaches for spacecraft trajectory design have seen a surge of interest in recent years [20], with strong results achieved both for trajectory cost estimation [15] and spacecraft guidance [21]. Neural Combinatorial Optimization (NCO) uses Deep Neural Networks (DNNs) to automate the problem-solving process, mostly under the Reinforcement Learning (RL) paradigm, as supervised learning is often unfeasible for large or theoretically hard problems. NCO offers the attractive prospect of alleviating the scaling issues of exact approaches, while removing the need for handcrafted heuristics, which often require significant domain-specific adjustments [18]. This is more the case as the realism of VRPs increases to match real operational conditions —with more constraints, more complicated and dynamic environments, possibly multiple agents. NCO has shown promising performance on various CO problems [18], especially when coupled with advanced policy search procedures [16].

1.1.1 Space Debris and Active Space Debris Removal

The present work focuses on the design of multi-rendezvous trajectories for ADR missions. This section starts with a historical overview of the space debris problem, as well as assessments of the current situation of space debris and recommendations from authoritative sources. The origin, evolution and impact of the most critical fragmentation events of the last 20 years are discussed afterwards. The section ends with a discussion of the ADR mission concept considered in this work.

Historical Overview and Space Debris Remediation

Since the launch of Sputnik-1 in 1957, Earth orbit has increasingly accumulated space debris, posing significant risks to operational satellites and space exploration. The first recorded catastrophic fragmentation event occurred in 1961, generating over 300 trackable debris pieces [22]. Over subsequent decades, more than 200 such events have contributed to a current population of approximately 34,000 trackable fragments larger than 10 cm in LEO [23].

Kessler and Cour-Palais [24] hypothesized in 1978 a scenario, now known as Kessler Syndrome, where the density of objects in LEO is high enough that collisions between objects could cause a cascade, exponentially increasing the number of debris and rendering space activities in certain orbital ranges unfeasible. NASA studies have suggested that some regions of LEO may have already reached this critical density, implying that even without new launches, the debris population could continue to grow due to ongoing collisions [25].

Awareness of the space debris problem has grown rapidly in recent years, both at the technical and policy levels [19], [26]. International agreements and guidelines, such as the *United Nations Space Debris Mitigation Guidelines of the Committee on the Peaceful Uses of Outer Space* [27] and the 25-year rule, accepted by both the United States National Air and Space Administration (NASA) and the European Space Agency (ESA), whereby defunct satellites must be de-orbited within 25 years after mission completion [28]. The ESA *Space Debris Mitigation Requirements* [29], in effect since November 2023[1], further reduced the maximum disposal phase in low-Earth orbit from 25 to 5 years. Adherence to these however is not globally enforced. The mitigation of existing debris is largely a problem to be solved at the economic, technology, testing, political and legal level [28].

ADR has emerged as a crucial strategy to mitigate this threat. Shan, Guo, and Gill [30] provided a comprehensive review of various ADR methods, including robotic arms, nets, harpoons, and electrodynamic tethers. Prioritizing the removal of large, massive debris objects—such as defunct satellites and spent rocket upper stages in densely populated orbital regions—is essential[9], [31]–[33], and is the chief priority in ADR mission development up to the present day.

ESA has been at the forefront of ADR mission development. The e.Deorbit programme [34] aimed to study the removal of large ESA-owned objects from orbit, involving extensive feasibility studies and technology development [34]. In 2018 the RemoveDEBRIS mission [35], funded by the European Commission, became the first operative ADR mission, demonstrating various cost-effective key technologies for ADR including net and harpoon capture using miniaturized systems designed for the capture of large uncompliant objects[36]. As of the present day, the

[1] https://esoc.esa.int/new-space-debris-mitigation-policy-and-requirements-effect

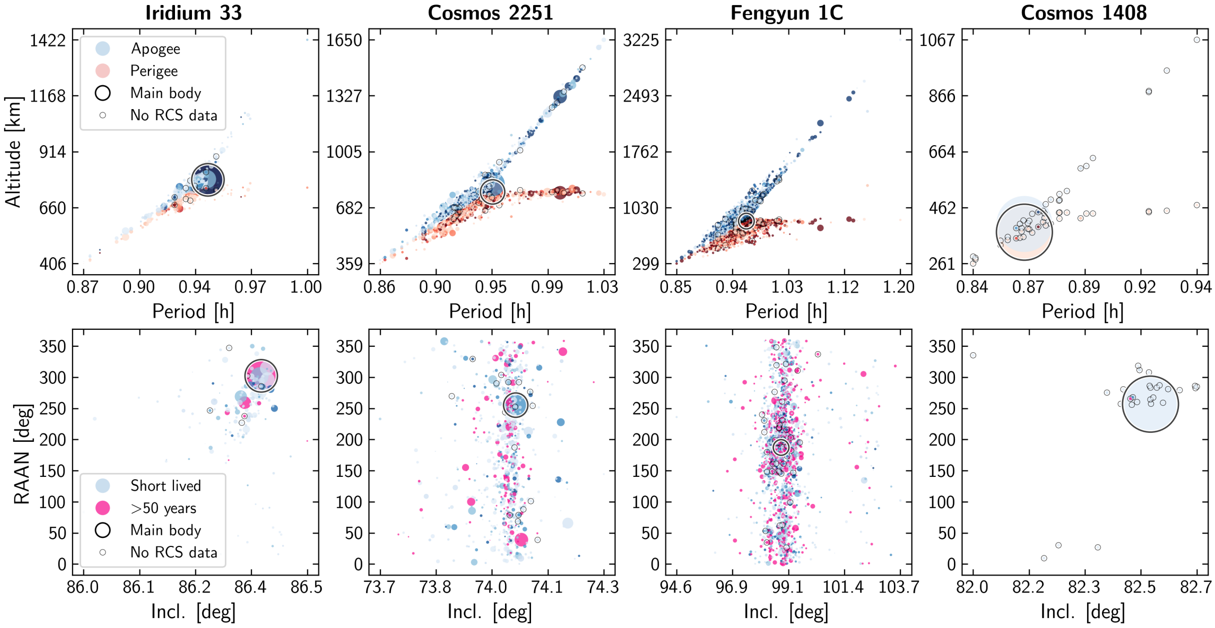


Figure 1.1: Gabbard diagram of the Iridium 33, Cosmos 2251, Fengyun 1C and Cosmos 1408 debris clouds as of September 2024. Point size proportional to RCS. Color intensity in top row proportional to lifetime before natural decay. Data obtained from Celestrak as of September 2024. Own work.

ESA ClearSpace-1[2] mission aims to de-orbit ESA-owned objects with mass greater than 100 kg by 2025 [37]. These efforts underscore the current emphasis on removing large, massive debris objects to mitigate the risk of catastrophic collisions.

Recent economic analysis from the NASA Orbital Debris Program Office Phase 2 report highlights the cost-effectiveness of certain debris remediation methods [19]. The report indicates that remediation approaches targeting small debris removal and just-in-time collision avoidance can yield net benefits within a decade. Specifically, methods like removing 1–10 cm debris or nudging large debris to prevent collisions may provide substantial economic returns by reducing collision risks and associated costs for satellite operators. These conditions herald opportunity for efficient multi-rendezvous ADR missions, and further on ADR constellations, to mitigate the space debris problem. This study aims to investigate the viability of NCO methods for the design of such missions.

Critical Fragmentation Events: Origin, Evolution and Impact

On February 10, 2009, the operational Iridium 33 satellite collided with the defunct Russian satellite Cosmos 2251 at an altitude of approximately 790 km, resulting in the first recorded collision between two intact satellites in orbit [38]. The collision generated over 2,000 pieces of trackable debris, significantly increasing collision risks for operational spacecraft in LEO [38]. The debris cloud, dispersed along the original orbits of the satellites, further congested the already crowded 700–800 km altitude region [28]. Subsequent space debris generation events, notably the Chinese Fengyun-1C anti-satellite test in 2007 [39], which produced over 3,000 pieces of debris, and the Russian Cosmos 1408 destruction in 2021 [40], have exacerbated debris congestion in LEO.

The Gabbard diagram [41] is a plot which shows the distribution of a cloud of orbiting objects across the altitude-period and Right Ascension of the Ascending Node (RAAN)-inclination axes. Gabbard diagrams are useful to intuitively assess the evolution of a cloud over time and the cost of an ADR mission targeting it, and are widely used to characterize fragmentation events [2], [41]. The Gabbard diagrams of the four debris clouds —Iridium 33, Cosmos 2252, Fengyun 1C and Cosmos 1408— as of September 2024 can be seen in Fig. 1.1. Up-to-date

[2] https://www.esa.int/Space_Safety/ClearSpace-1

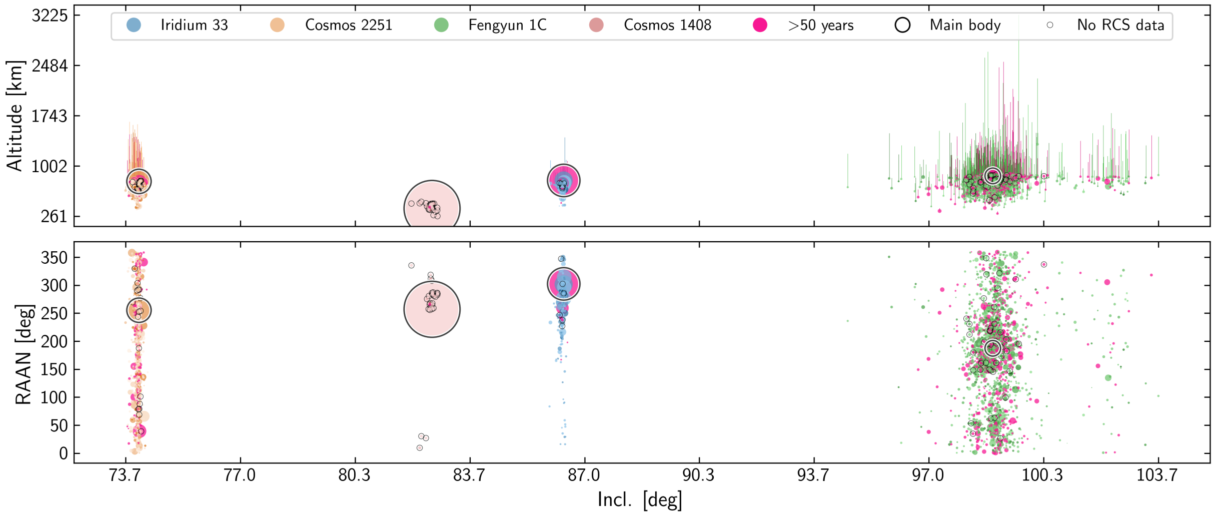


Figure 1.2: Joint altitude-RAAN-inclination diagram of the Iridium 33, Cosmos 2251, Fengyun 1C and Cosmos 1408 debris clouds as of September 2024. Point size proportional to RCS. Color intensity in top row proportional to lifetime before natural decay. Color trails, top: perigee to apogee altitude. Data obtained from Celestrak as of September 2024. Own work.

satellite tracking data is obtained from CelesTrak[3]. The Gabbard diagrams in Fig. 1.1 display two more important pieces of information: the Radar Cross-Section (RCS) of the debris, and the expected natural decay time of debris. Decay time data is obtained from the ESA Database and Information System Characterising Objects in Space (DISCOS)[4].

In Fig. 1.2 the altitude and RAAN distributions of the four clouds are displayed together as a function of inclination. This offers a mission designer's view of the problem: a map relating the most important orbital parameters for mission design in LEO to the likelihood of collisions with debris from the Iridium 33, Cosmos 2252, Fengyun 1C and Cosmos 1408 clouds.

Debris clouds evolve over time due various perturbations. The most important perturbations are atmospheric drag, which causes debris to shed altitude over time and eventually decay, and the gravity gradient perturbation. As can be seen in Tab. 1.1 decay has considerably reduced the size of the clouds since their fragmentation events. Cosmos 1408, the lowest of the four clouds —see Fig. 1.2, is an extreme example of natural decay due to drag with over 96% of all fragments having decayed between 2021 and the present day in 2024. The Fengyun 1C cloud will outlast all of them. Up to 1000 pieces of debris will remain in orbit by the year 2100 according to ESA estimates. The secular impact of the gravity gradient perturbation causes a linear drift in the Argument of Perigee (AOP) and RAAN of the orbit drift over time. This effect is fast (for context, SSO orbits see a RAAN drift of 360° per year) and has a profound impact on rendezvous cost, as the RAAN gap to be closed is the greatest contributor to plane change ΔV for high inclination orbits [2].

The combination of the altitude-period and RAAN-inclination distributions in Fig. 1.1 is helpful to gain intuition about the evolution of the clouds over time. Observe how as of 2024 the range of RAAN values of the Cosmos 2251 as well as the evenness of the distribution of debris over RAAN are both remarkably large. This is caused by the low inclination of the cloud, which causes RAAN drift to accelerate, and the large range of semi-major axes and eccentricities in the cloud, which cause variations in the rates of drift of different pieces of debris within the cloud. The Cosmos 1408 and Iridium 33 clouds have higher inclinations as well as lower variances in semi-major axis and eccentricity, leading to considerably more dense clouds along RAAN. The Fengyun 1C cloud towers

[3] https://celestrak.org
[4] https://discosweb.esoc.esa.int

Table 1.1: Number of debris fragments and RCS in [m^2] of the Iridium 33, Cosmos 2252, Fengyun 1C and Cosmos 1408 debris clouds at the moment of fragmentation event (**T0**), as of the present day in 2024, and estimates for the year 2050 and 2100. Estimates are obtained from the ESA DISCOS database.

	Iridium 33		**Cosmos 2251**		**Fengyun 1C**		**Cosmos 1408**	
	Count	RCS	Count	RCS	Count	RCS	Count	RCS
T0	631	20,64	1626	34,81	3043	55,29	1801	7,51
2024	193	10,24	831	21,13	2192	42,96	68	7,50
2050	19	3,61	207	7,63	903	20,65	0	0,00
2100	5	3,01	76	3,38	435	11,04	0	0,00

above all others both in physical terms —the ranges of RAAN and inclination, the ranges of semi-major axis and eccentricity, the number of active debris and the lifetime of active debris— as well as damage to space operations, as the 96-98° inclination range is critical for many spacecraft in LEO that rely on Sun-Synchronous Orbits (SSO) for their operations. As the cost of rendezvous is primarily driven by plane change cost [2], ADR missions are bound to require extreme amounts of ΔV.

These events underscore the economic impact of space debris proliferation and emphasize the urgent need for remediation strategies to safeguard space assets and ensure sustainable orbital use.

Active Debris Removal Mission Concept

This study considers an ADR mission targeting the Iridium 33 debris cloud. The characteristics of the spacecraft considered through the rest of this work follow in Tab. 1.2. The propulsion system is based on the specifications of existing Gridded Ion Thruster designs for small spacecraft, using the same power to thrust ratio of 21 kW N^{-1} of the MiXi GIT propulsion system developed at UCLA [42] (for more information refer to O'Reilly's extensive review of electric propulsion systems for small spacecraft [43]). Spacecraft structural, fuel and payload mass are only indicative of a spacecraft fit for this type of mission; payload mass is sized to 10-30 active de-orbiting payloads [30], [35].

Table 1.2: Debris chaser spacecraft specifications.

Wet mass	Fuel mass	Payload mass	Max a_T	Min a_T	I_{sp}	ΔV budget	Max thrust	Max power
1200 kg	450 kg	500 kg	3e—4 m s^{-2}	1e—5 m s^{-2}	3000 s	6.00 km s^{-1}	0.36 N	7.55 kW

1.1.2 Sequential Payload Deployment to Low Earth Orbit

We demonstrate its capabilities by solving the multi-rendezvous trajectory optimization problem of UARX Space's OSSIE[5] OTV: a translational and mass-dynamic multi-satellite deployment problem in LEO.

UARX Space OSSIE OTV

To demonstrate the capabilities of the proposed framework, the UARX Space Orbit Solutions to Simplify Injection and Exploration OTV, known as OSSIE, will be used as a case study. Depicted in Fig. 1.3, OSSIE is a modular payload delivery platform with LEO, MEO and cis-lunar capability. In this paper, we consider a nominal mission profile aiming to deliver 4 PocketQubes, 8 CubeSats, and 1 small satellite to LEO.

The propulsion system used for this mission consists of 4 parallel Dawn Aerospace B20 bi-propellant (nitrous oxide and propene) thrusters[6] (specifications in Tab. 1.3). As of the time of writing, the duty-cycle constraints of the thruster cluster constrain manoeuver design to multiple-revolution transfers, with up to two impulses per orbit in LEO. Attitude control is performed using Dawn Aerospace B1 thrusters and magnetorquers. The vehicle

[5] https://www.uarx.com/projects/ossie.php

[6] https://www.dawnaerospace.com/green-propulsion

must retain attitude control authority at all times while ensuring correct pointing for thrusting as well as all other operational modes. This poses a challenge for fuel consumption, as the B1 thrusters are the main attitude control actuators and share fuel with the B20 thrusters used to perform orbital transfers.

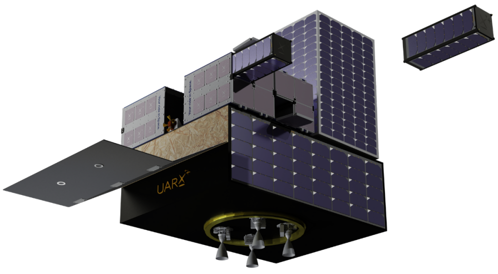


Figure 1.3: UARX Space OSSIE OTV. Credit: UARX Space.

Table 1.3: Specifications of the OSSIE OTV and Dawn Aerospace B20 thrusters.

OSSIE OTV		**Dawn Aerospace B20**	
Specification	Value	Specification	Value
Wet mass	235 kg	Specific impulse	277 s
Payload mass	80 kg	Peak thrust	12.6 N
Fuel mass	35 kg	Minimum impulse bit	1 N s
Cargo capacity	48 U		

The specifications of the current configuration of OSSIE follow in Tab. 1.3. OSSIE has a wet mass of approximately 235 kg, of which close to 50% is shed through the mission —either deployed or consumed propellant; this means that the mass deployment sequence may have a considerable impact on the fuel required to complete the mission. OSSIE is deployed to a nominal insertion orbit in LEO, which constraints transfer sequence design, and, furthermore, must conduct a decommissioning manoeuvre to ensure it decays within 5 years of EOL as per the ESA Space Debris Mitigation Requirements [29].

Optimization Problem Statement

Under the previous considerations, the OSSIE trajectory optimization problem results in a highly tailored multi-rendezvous trajectory optimization problem. The goal of the problem is to construct a fuel-optimal optimal trajectory which reaches all payload destination orbits. The number of payloads will be variable in each mission and up to 13, considering the advertised carrying capacity of OSSIE of 8 CubeSats, 4 PocketCubes and 1 small satellite. A solution to the problem must minimize fuel consumption considering:

- The impact of the relevant perturbations on multi-revolution impulsive manoeuvres.
- The impact of mass deployment sequence on propellant consumption.
- Insertion and decommissioning orbit constraints.
- Ensuring the spacecraft retains attitude control authority through the manoeuvre.

1.2 Research Questions

RQ.1 *Can NCO methods be used to learn effective routing policies for space VRPs?*

RQ.2 *What is a suitable approach for the design of multi-rendezvous trajectories with strict feasibility guarantees?*

RESEARCH SUBQUESTIONS

1. Can NCO methods be used to improve the performance of state-of-the-art solvers for space VRPs?
   (a) What is a suitable NCO architecture to learn routing policies for space VRPs?
   (b) What are the optimal training procedures for NCO agents designed to solve the STSP?
   (c) Can NCO methods yield improved performance with respect to state-of-the-art HCO algorithms for the STSP?
   (d) Can NCO methods be leveraged to improve the performance and speed of HCO algorithms?
2. How can trajectory optimization methods with feasibility guarantees, specifically Sequential Convex Programming (SCP), be efficiently integrated in the multi-rendezvous trajectory optimization process?

1.3 Project Goals

G.1 *To implement and optimize a multi-rendezvous trajectory optimization solver based on HCO methods.*

G.2 *To implement and optimize a multi-rendezvous trajectory optimization solver based on NCO.*

G.3 *To design a multi-rendezvous trajectory optimization software architecture covering the end-to-end guidance design process from mission specification to trajectory refinement and verification for use in on-board GNC, and to apply it to obtain optimal multi-rendezvous trajectories for the OSSIE OTV.*

G.4 *To analyze the nominal mission scenario designed for the OSSIE OTV and assess the capability of the OSSIE OTV to achieve that scenario considering the full envelope of feasible customer demands (advertised services).*

PROJECT SUBGOALS

1. To implement a near-optimal and efficient LEO-to-LEO LTTO algorithm
   (a) To assess LTTO approaches in literature and choose an optimal approach for the conceptual design of low-thrust multi-rendezvous missions
   (b) To implement a 6-element targeting low-thrust guidance policy
   (c) To implement an optimal simulator to generate trajectories for use in the conceptual design of low-thrust ADR missions in LEO
   (d) To implement an accurate transfer cost estimation method to be used in CO
2. To implement a near-optimal LEO-to-LEO impulsive trajectory optimization algorithm, capable of considering the mission objectives and constraints of the OSSIE OTV
   (a) To assess impulsive trajectory design approaches in literature and choose an optimal approach for the OSSIE OTV
   (b) To implement a 3-element targeting impulsive guidance policy
   (c) To implement an optimal simulator to generate feasible trajectories for the OSSIE OTV fit for use as warm starts in SCP
   (d) To implement an accurate transfer cost estimation method to be used in CO
3. To implement a state-of-the-art space VRP solver using HCO
   (a) To model the dynamic environment of space VRPs in LEO
   (b) To implement a population sampling algorithm capable of leveraging approximate solutions obtained with other methods
   (c) To determine the optimal HCO method for the ADR STSP and OSSIE mission design cases
4. To implement a state-of-the-art space VRP solver using NCO

   (a) To assess NCO approaches in literature and select an optimal policy for the solution of space VRPs
   (b) To assess the performance of various RL algorithms to train space VRP routing policies, and find an optimal choice
   (c) To implement a state-of-the-art hyperparameter optimization process
   (d) To assess the performance of NCO routing policies in comparison with HCO

5. To investigate the viability of the nominal mission scenarios designed for the OSSIE OTV
   (a) To implement a highly performant space VRP solver tailored for the OSSIE OTV
   (b) To implement a large-scale Monte Carlo analysis tool capable of efficiently generating and solving arbitrary mission scenarios
   (c) To conduct a statistical analysis of the performance requirements of nominal mission scenarios, and assess the capacity of the OSSIE OTV to satisfy all nominal mission scenarios

1.4 Thesis Focus and Structure

The present work aims to answer two fundamental research questions. The first has to do with the applicability and effectiveness of NCO approaches for the design of multi-target rendezvous trajectories. The second has to do with the design of trajectories with feasibility guarantees for OSSIE. Two research lines were drawn in this work with the goal of giving a definitive answer to each research question: a study of the design of Active Debris Removal missions in LEO using NCO, and the design of a guidance system to generate multi-rendezvous for the OSSIE OTV with correctness guarantees. At the core of both pieces of research is a solution framework for multi-rendezvous problems capable of integrating NCO policies. As a result this thesis consists of two different research papers, and accompanying material.

The accompanying material in this thesis consists of two chapters and four appendices, and has two aims. The first aim is to provide a unified and coherent discussion of topics which are transversal in both papers, and of topics which are not common to both papers but complementary, e.g. trajectory design. The second aim is to provide greater depth in the discussion of selected topics, when such depth would not be warranted in a paper limited to a maximum number of pages.

The background —common and complementary— of the research presented in this thesis is presented in Chapter 2. Orbital mechanics are discussed in Section 2.1, followed by impulsive trajectory design in Section 2.2 and low-thrust trajectory design in Section 2.3. Combinatorial Optimization follows. Section 2.4 discusses HCOmbinatorial Optimization. Lastly, Section 2.5 discusses Neural Combinatorial Optimization.

Chapter 3 deals with verification and validation, the design of the experiments conducted in this work and sensitivity analysis. The purpose of this chapter is to provide the reader with a unified discussion of the measures taken to ensure the correctness of all methods implemented in this work, in Section 3.1; to ensure the robustness of all experiments conducted in this work, in Section 3.2; and to study the sensitivity of NCO policy performance on the hyperparameters of the model and training process, in Section 3.3.

Chapter 4 contains the paper titled **Neural Combinatorial Optimization for Multi-Rendezvous Mission Design**. This paper aims to answer Research Question 1: *Can NCO methods be used to learn effective routing policies for space VRPs?*

Chapter 5 contains the paper titled **Design And Optimization Of Multi-rendezvous Maneuvres Based On Reinforcement Learning And Convex Optimization**, presented at the 2024 International Astronautical Congress in Milan, Italy. This paper aims to answer Research Question 2: *What is a suitable approach for the design of multi-rendezvous trajectories with strict feasibility guarantees?*

Chapter 6 concludes the thesis outlining the main results of this thesis. Section 6.1 assesses the outcomes of the thesis, in two parts. Firstly, the two research questions are reviewed in light of the research done. Then, a thorough analysis of compliance with project goals is presented.

The conclusion is followed by the bibliography and four appendices. Appendix A deals with the statistical modelling of space debris clouds, vital for the creation of realistic environments to train NCO policies for ADR missions. Appendix B contains an in-depth discussion of statistical models for permutations, which lay at the foundation of the HCO framework presented in this work. Appendix C discusses the architecture of the software developed for this research. Lastly, the planning and execution of the project is discussed in Appendix D.

2 Background

This chapter establishes the fundamental theoretical background for the two lines of research conducted in this work. This comprises physical models, discussed in Section 2.1, trajectory design methods, discussed in Section 2.2 and Section 2.3 and heuristic and neural combinatorial optimizatation methods, discussed in Section 2.4 and Section 2.5 respectively. This chapter is supported by Appendix A and Section B.2, dedicated to statistical models for space debris clouds and permutations.

2.1 Orbital Mechanics

This section introduces the reference frames (Section 2.1.1), translational physical models (Section 2.1.2) and orbit propagation models (Section 2.1.3) used in this work.

2.1.1 Reference Frames

Earth-Centered Inertial reference frame

The Earth-Centered Inertial (ECI) reference frame is defined with its origin at the Earth's center of mass and remains fixed with respect to the celestial sphere [44]. The basis vectors are oriented as follows:

- $\mathbf{x}_I$ points towards the March equinox, also known as the First Point of Aries, where the Earth's equatorial plane intersects the ecliptic plane.
- $\mathbf{y}_I$ lies in the equatorial plane, orthogonal to $\mathbf{x}_I$, completing the right-handed coordinate system.
- $\mathbf{z}_I$ aligns with the Earth's rotation axis, pointing towards the North Celestial Pole.

Local-Vertical Local-Horizontal reference frame

The Local-Vertical Local-Horizontal (LVLH) reference frame is a non-inertial, rotating coordinate system centered at the spacecraft [45]. This reference frame is also known as the RSW frame and CSN frame in literature. We make use of the Hintz convention with the radial component pointing away from the center of gravity of the Earth [45], [46]. The basis vectors are defined as follows:

- $\hat{\mathbf{e}}_r$ points away the center of gravity of the Earth, aligned with the negative position vector $-\mathbf{r}$ (Local Vertical).
- $\hat{\mathbf{e}}_\phi$ is oriented opposite to the orbit angular momentum vector $\mathbf{h}$.
- $\hat{\mathbf{e}}_\theta$ forms a right-handed coordinate system, and is tangential to the instantaneous velocity of the spacecraft.

The LVLH and ECI frames is visualized in Fig. 2.1. The transformation of a vector $\hat{\mathbf{u}}_{\text{ECI}}$ defined in the ECI frame to the LVLH frame $\hat{\mathbf{u}}_{\text{MEE}}$ is defined in Eq. 2.1.

$$\hat{\mathbf{u}}_{\text{ECI}} = [\hat{\mathbf{e}}_r \quad \hat{\mathbf{e}}_\theta \quad \hat{\mathbf{e}}_\phi]\hat{\mathbf{u}}_{\text{MEE}} \tag{2.1a}$$

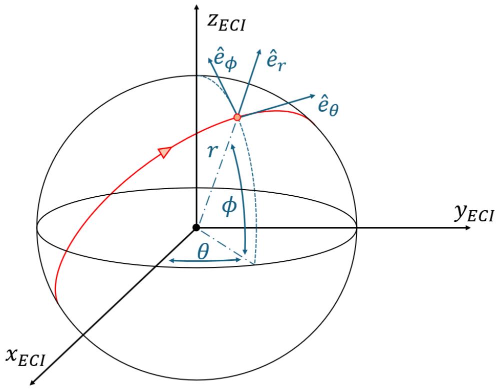


Figure 2.1: ECI and LVLH reference frames.

$$\hat{\mathbf{e}}_r = \frac{\mathbf{r}}{\|\mathbf{r}\|}; \quad \hat{\mathbf{e}}_\phi = \frac{\mathbf{r}\times\mathbf{v}}{\|\mathbf{r}\times\mathbf{v}\|}; \quad \hat{\mathbf{e}}_\theta = \hat{\mathbf{e}}_\phi \times \hat{\mathbf{e}}_r \tag{2.1b}$$

2.1.2 Environment Model

Point-Mass Gravity Field Model

The simplest possible gravity field model is the point mass approximation [47]. Newton's law of gravitation for two point masses is given by:

$$(\mathbf{F}_1)_2 = \frac{Gm_1m_2}{\|\mathbf{r_{1,2}}\|^3}\mathbf{r_{1,2}} \tag{2.2}$$

where $(\mathbf{F}_1)_2$ is the force exerted on body 1 due to the gravitational pull of body 2, $\mathbf{r_{1,2}}$ is the vector from the center of gravity of body 1 to the center of gravity of body 2, G is the universal gravitational constant, and m_1 and m_2 are the masses of the bodies. Assuming that spacecraft mass is negligible with respect of that of the central body yields the following expression for the gravitational acceleration of a spacecraft s orbiting a central body C:

$$(\mathbf{a}_s)_C = -G\frac{m_E}{\|\mathbf{r_{C,s}}\|^3}\mathbf{r_{C,s}} \tag{2.3}$$

where $\mathbf{r_{C,s}}$ is the vector from the center of gravity central body to that of the spacecraft. As the only problems considered in this work are two-body problems in Earth orbit, the convention $\mathbf{a_k}$ will be used to refer to the acceleration exerted on a spacecraft due to a factor $\mathbf{k}$, e.g. a_{J2} will be used to refer to the acceleration exerted on a spacecraft corresponding to the $J2$ zonal harmonic coefficient. This perturbation in particular is discussed next.

Natural Perturbations

Natural perturbations are those that stem from differences between the model of the space environment and real physical phenomena. Two natural perturbations must be considered for the design of near-Earth orbits: atmospheric drag, and gravity gradient perturbations. The perturbing acceleration from atmospheric drag is modelled by Eq. 2.4,

$$\mathbf{a}_D = -C_D\frac{1}{2}\rho\frac{A}{m}\mathbf{v}\|\mathbf{v}\| \tag{2.4}$$

where C_D is the drag coefficient related to the reference area A of the spacecraft, m is the mass of the spacecraft, ρ stands for atmospheric pressure and $\mathbf{v}$ is the velocity of the satellite relative to the rotating atmosphere of the Earth [47]. Drag is a non-conservative force which constantly drains the orbital energy of an orbiting body, causing orbital decay. Drag is the most important natural perturbations for orbits at an altitude of 200 km, and can be neglected for orbits with altitudes above 1000 km [47].

The gravity gradient perturbation is the natural perturbation with the greatest impact for trajectory design in the operational regimes considered in this work: the location of the Iridium 33 debris cloud at an altitude of approximately 800 km, and the nominal altitude envelope of the OSSIE OTV centered at 600 km. The gravity gradient perturbation is caused by the difference between the real gravity field of the Earth and the ideal point-mass model of the Earth's gravity field. The greatest contribution to the gravity gradient perturbation comes from the Earth oblateness gravity potential distortion [2], [48], characterized by the second zonal harmonic coefficient or J_2. The instantaneous acceleration components due to the J_2 perturbation follow in Eq. 2.5, expressed in the LVLH frame.

$$\begin{aligned} a_{J_{2,r}} &= -\frac{3\mu J_2 R_e^2}{2r^4}\left(1-\frac{12v^2}{s^4}\right) \\ a_{J_{2,\theta}} &= -\frac{12\mu J_2 R_e^2}{r^4}\left(\frac{v(h\cos L + k\sin L)}{s^4}\right) \\ a_{J_{2,\phi}} &= -\frac{6\mu J_2 R_e^2}{r^4}\left(\frac{v(1-h^2-k^2)}{s^4}\right) \end{aligned} \tag{2.5}$$

The J_2 perturbation has a secular impact on the RAAN and the AOP over a full orbit [47], modelled by Eq. 2.6,

$$\begin{aligned} \frac{\mathrm{d}\Omega}{\mathrm{d}t} &= -\frac{3}{2}J_2\left(\frac{R_e}{p}\right)n\cos i; \\ \frac{\mathrm{d}\omega}{\mathrm{d}t} &= -\frac{3}{4}J_2\left(\frac{R_e}{p}\right)n(5\cos^2 i - 1)); \end{aligned} \tag{2.6}$$

where $n = \sqrt{\mu/a^3}$ is the mean motion of the orbiting body.

Thrust Acceleration

The thrust acceleration $\mathbf{a}_T$ applied by the spacecraft in the RWS frame is defined in Eq. 2.7, where $\hat{\mathbf{u}}$ is the direction of application of thrust.

$$\mathbf{a}_T = \frac{T}{m}\hat{\mathbf{u}} \tag{2.7}$$

2.1.3 Orbit Propagation Model

The translational state of spacecraft is propagated using the Modified Equinoctial Elements (MEEs) described by Hintz [45] including the retrograde factor I, which are nonsingular for all eccentricities and inclinations. A visualization of the physical meaning of each MEE follows in Fig. 2.2. The conversion from classical Keplerian elements [45] to MEEs follows in Eq. 2.8:

$$\begin{aligned} p &= a(1-e^2); \\ f &= e\cos(\omega+\Omega); \\ g &= e\sin(\omega+\Omega); \\ h &= \tan(i/2)\sin(\Omega); \\ k &= \tan(i/2)\cos(\Omega); \\ L &= \theta + I\Omega + \omega; \end{aligned} \tag{2.8}$$

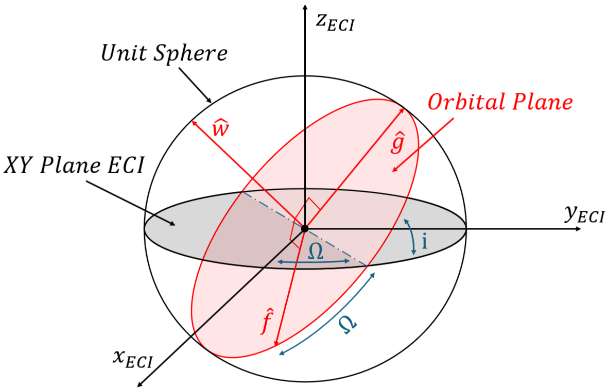


Figure 2.2: Modified Equinoctial Elements (MEE) with respect to orbital plane.

where *a* stands for the semimajor axis, *e* for orbital eccentricity, *i* for orbital inclination, Ω for RAAN, ω for AOP, and θ the true anomaly.

The Gauss Variational Equations for MEEs [45], in Eq. 2.9, are used to model the time evolution of the spacecraft's state:

$$\begin{aligned}
\frac{\mathrm{d}p}{\mathrm{d}t} &= \frac{2p}{w}\sqrt{\frac{p}{\mu}}\Delta_t; \\
\frac{\mathrm{d}f}{\mathrm{d}t} &= \sqrt{\frac{p}{\mu}}\left\{\Delta_r \sin(L) + \frac{(w+1)\cos(L)+f}{w}\Delta_t - g\frac{v}{w}\Delta_n\right\}; \\
\frac{\mathrm{d}g}{\mathrm{d}t} &= \sqrt{\frac{p}{\mu}}\left\{-\Delta_r \cos(L) + \frac{(w+1)\sin(L)+g}{w}\Delta_t - f\frac{v}{w}\Delta_n\right\}; \\
\frac{\mathrm{d}h}{\mathrm{d}t} &= \sqrt{\frac{p}{\mu}}\frac{s^2}{2w}\cos(L)\Delta_n; \\
\frac{\mathrm{d}k}{\mathrm{d}t} &= \sqrt{\frac{p}{\mu}}\frac{s^2}{2w}\sin(L)\Delta_n; \\
\frac{\mathrm{d}L}{\mathrm{d}t} &= \sqrt{\mu p}\left(\frac{w}{p}\right)^2 + \sqrt{\frac{p}{\mu}}\frac{v}{w}\Delta_n;
\end{aligned} \tag{2.9}$$

where s^2, v and w are defined as follows,

$$\begin{aligned}
s^2 &= 1 + h^2 + k^2; \\
v &= h\sin(L) - k\cos(L); \\
w &= 1 + f\cos(L) + g\sin(L);
\end{aligned} \tag{2.10}$$

and Δ_r, Δ_t and Δ_n are perturbing accelerations in the radial, tangential, and normal directions of the spacecraft's LVLH frame $\hat{e}$ depicted in Fig. 2.1. The spacecraft's mass is propagated according to Eq. 2.11, assuming constant I_{sp} through a single burn.

$$\frac{\mathrm{d}m}{\mathrm{d}t} = \frac{T}{v_{\text{eq}}}; \tag{2.11}$$

2.2 Impulsive Trajectory Design

This section introduces the impulsive trajectory design policies used for the UARX Space OSSIE OTV. OSSIE sports a propulsion system consisting of four Dawn Aerospace B20 thrusters in parallel, with a duty cycle reminiscent of an impulsive propulsion system but with a fairly low maximum thrust of 45 [N]. This means OSSIE must perform multi-revolution transfers applying many, weak impulsive burns to reach its targets. OSSIE is required to achieve 3-element targeting: semi-major axis, inclination and phase. Hohmann transfers are commonly used to transfer between coplanar circular orbits with varying radii. Nodal Inclination Change (NIC) maneuvers are a standard approach used to adjust orbital inclination without altering RAAN [5], [47]. This section discusses the design of multi-revolution Hohmann transfers in Section 2.2.1, multi-revolution NIC manoeuvres in Section 2.2.2, multi-revolution IPC manoeuvres in Section 2.2.3, and sequential manoeuvres targeting both semi-major axis and plane changes, by combining MHT, NIC and IPC manoeuvres Section 2.2.4.

2.2.1 Multiple Hohmann Transfer manoeuvres

A Hohmann transfer is the optimal manoeuvre to transfer between two coplanar circular orbits of different radii. It is a two-impulse manoeuvre consisting of an initial burn, either raising apogee or lowering perigee —if the transfer aims to lower the Semi-Major Axis (SMA) of the initial orbit— and a circularization burn when the apogee (or perigee) of the transfer manoeuvre is reached. For OSSIE, the ΔV required to perform a direct Hohmann transfer may not be achievable. This limitation is dealt with by performing Multiple Hohmann Transfers (MHT) instead. The MHT splits the departure and circularization burns into many consecutive insertion and circularization smaller burns, affordable by the maximum ΔV that can be injected at a time with the employed thrusters.

Required ΔV, fuel mass and number of burns

The ΔV required by the MHT is equal to the one of a direct Hohmann Transfer achieving the same altitude change. The total magnitude can be computed as follows in Eq. 2.12, where ΔV_d and ΔV_c stand for departure and circularization ΔV, $V_0 = \sqrt{\mu/r_0}$ is the orbital velocity at the departure orbit and $\xi = r_1/r_0$ is the ratio of target to departure orbit [47].

$$\Delta V_{\text{MHT}} = \Delta V_d + \Delta V_c \tag{2.12a}$$

$$\Delta V_d = V_0 \left| \sqrt{\frac{2n}{\xi + 1}} - 1 \right| \tag{2.12b}$$

$$\Delta V_c = V_0 \sqrt{\frac{1}{\xi}} \left| \sqrt{\frac{2n}{\xi + 1}} - 1 \right| \tag{2.12c}$$

The fuel mass required to perform the MHT is calculated as follows in Eq. 2.13, assuming constant I_{sp} through the manoeuvre; the number k of burns required to perform a manoeuvre is obtained as the ceil fraction of required fuel mass over mass flow $\dot{m}_f = T/(I_{sp}g_0)$ (see Eq. 2.13). The mass flow is assumed constant, as T and I_{sp} are assumed constant through the transfer.

$$m_f = m_0 \left[1 - \exp\left(-\frac{\Delta V}{I_{sp} g_0} \right) \right]; \quad k = \left\lceil \frac{m_f}{\dot{m}_f} \right\rceil \tag{2.13}$$

Time Of Flight

The total TOF of the MHT follows in Eq. 2.14,

$$\text{TOF}_{\text{MHT}} = \min(1, f_{\text{burn}}) \cdot (k_d + k_c) \cdot \overline{P}_{\text{MHT}} \tag{2.14}$$

where k_d and k_c are the number of burns required to perform the departure and circularization legs of the manoeuvre respectively, calculated using Eq. 2.13, f_{burn} stands for the burn frequency, and $\overline{P}$ stands for the average orbital period through the transfer. The burn frequency is the maximum number of burns per second that the spacecraft can perform, defined in Eq. 2.15,

$$f_{\text{burn}} = (T_{\text{burn}} + T_{\text{cooldown}})^{-1} \tag{2.15}$$

where T_{burn} is the burn time of the spacecraft's propulsion system and T_{cooldown} is the required cooldown time after one burn. The average orbital period through the transfer, $\overline{P}$, can be derived to be Eq. 2.16:

$$\overline{P}_{\text{MHT}} = \frac{4\pi}{5\sqrt{\mu}\,(r_2 - r_1)}\left(r_2^{\frac{5}{2}} - {r_1}^{\frac{5}{2}}\right) \tag{2.16}$$

Phasing

The spacecraft ends the MHT at true longitude $L_0 + \pi$. To avoid phasing manoeuvres after reaching the required orbit, in order to acquire the target true longitude, the MHT start epoch is selected so that $L_0(t_0) = L_1(t_0 + \text{TOF}_{\text{MHT}}) - \pi$, where TOF is the estimated time of flight. This correction is computed after the MHT calculation, thus TOF$_{\text{MHT}}$ is known.

2.2.2 Nodal Inclination Change manoeuvres

An NIC manoeuvre modifies the inclination of an orbit without affecting its RAAN. Thrust is applied impulsively as the spacecraft crosses the ascending or descending node, such that the resulting velocity vector is that of the orbit with the desired inclination. A multiple NIC manoeuvre consists of splitting the impulsive ΔV that must be applied into multiple burns, up to twice per orbit, as the spacecraft crosses the orbit ascending and descending nodes.

Required ΔV, fuel mass and number of burns

The ΔV required for an NIC follows in Eq. 2.17,

$$\Delta V_{\text{NIC}} = 2V_0 \sin\left(\frac{\Delta i}{2}\right) \tag{2.17}$$

where m_f stands for fuel mass, and k_{NIC} is the number of burns required for to perform the NIC, which is calculated using Eq. 2.13. As the ΔV depends only on the change in inclination and orbital velocity (constant through the manoeuvre), the cost of a multiple NIC is the same as the cost of a single-burn NIC [47].

Time Of Flight

The TOF of a multiple NIC is calculated using Eq. 2.18,

$$\text{TOF}_{\text{NIC}} = \min(2, f_{\text{burn}}) \cdot (k_{\text{NIC}}) \cdot P \tag{2.18}$$

where $P = 2\pi\sqrt{r^3/\mu}$ is the constant orbital period.

2.2.3 Impulsive Plane Change Manoeuvres

An Impulsive Plane Change (IPC) manoeuvre modifies the inclination and RAAN of an orbit simultaneously. This is done by applying thrust at the intersection of the original and target orbit [2], [5]. A multiple IPC manoeuvre consists of splitting the impulsive ΔV that must be applied into multiple burns, up to twice per orbit, as the spacecraft's orbit intersects the target orbit. Critically this assumes circular orbits with equal radius (otherwise it is not a given that there will be any intersections between the two orbits at which to apply thrust).

Required ΔV, fuel mass and number of burns

The ΔV required for an IPC follows in Eq. 2.19a, where γ is the relative inclination, defined in Eq. 2.19b [2], [5]. As the ΔV depends only on the change in relative inclination and orbital velocity (constant through the manoeuvre), the cost of a multiple IPC is the same as the cost of a single-burn IPC. The fuel mass m_f and number of burns k_{IPC} required for to perform the IPC is calculated using Eq. 2.13.

$$\Delta V_{\text{IPC}} = 2V_0 \sin\left(\frac{\gamma}{2}\right) \tag{2.19a}$$

$$\gamma = \arccos(\cos i_1 \cos i_2 + \sin i_1 \sin i_2[\cos\Omega_1 \cos\Omega_2 + \sin\Omega_1 \sin\Omega_2]) \tag{2.19b}$$

Time Of Flight

The TOF of a multiple IPC is calculated with Eq. 2.20, where $P = 2\pi\sqrt{r^3/\mu}$ is the constant orbital period.

$$\text{TOF}_{\text{IPC}} = \min(2, f_{\text{burn}}) \cdot (k_{\text{IPC}}) \cdot P \tag{2.20}$$

2.2.4 Sequential impulsive manoeuvres

From Eq. 2.17 it is of paramount importance to lower orbital velocity before performing an NIC. This is the case for IPCs as well, see Eq. 2.19. OSSIE will often have to reach targets with different semi-major axes and inclinations than its current ones: the most common case is orbit acquisition for payloads which require a particular orbital altitude to carry out their activity as well as a sun-synchronous orbit —commonly the case for Earth observation and mapping satellites. Algorithm 1 is used to plan sequential MHT and plane change manoeuvres, either NIC or IPC, by conducting the plane change when the semi-major axis is highest, such that the ΔV required for the plane change leg is minimized.

Algorithm 1: Sequential MHT-NIC/IPC Transfer

```
if $r_2 > r_1$ then                                        // Orbit raising
    Step 1: MHT
    Compute coast time to $L_0$
    Compute $\Delta V_{\text{MHT}}$
    Compute time of flight $t_{\text{MHT}}$
    Step 2: Plane change
    Compute coast time to closest node
    Compute $\Delta V_{\text{NIC/IPC}}$
    Compute time of flight $t_{\text{NIC/IPC}}$
else                                                       // Orbit lowering
    Step 1: Plane change
    Compute coast time to closest node
    Compute $\Delta V_{\text{NIC/IPC}}$
    Compute time of flight $t_{\text{NIC/IPC}}$
    Step 2: MHT
    Compute coast time to $L_0$
    Compute $\Delta V_{\text{MHT}}$
    Compute time of flight $t_{\text{MHT}}$
```

2.3 Low-Thrust Trajectory Design

Low-Thrust Trajectory Optimization (LTTO) is a critical aspect of mission planning for spacecraft employing electric propulsion systems. Traditionally, these optimization problems are formulated as Optimal Control Problems

(OCPs), suitable for continuous spacecraft dynamics with real variables and parameters [49]. However, electric propulsion systems exhibit hybrid dynamical behavior due to discrete operational modes—thrusting and coasting—and mission design variables [50]. In these cases LTTO is tackled as a Hybrid Optimal Control Problem [50]. Solving HOCPs is challenging due to the interconnection of continuous and discrete dynamics, rendering methods for Continuous Optimal Control Problems (COCPs) and purely discrete optimization unsuitable [51]. The Hybrid Minimum Principle (HMP) provides first-order necessary conditions for HOCPs by generalizing Pontryagin's Maximum Principle, but it requires the sequence of discrete events to be specified by the user, converting the HOCP into a multi-point boundary value problem solvable by indirect methods [52]. Dynamic programming has been extended to HOCPs, but convergence to the true value function is not generally guaranteed [53]. Consequently, HOCPs are typically solved using direct methods formulated as MINLP, which are NP-hard to solve [50]. To reduce computational time, hybrid optimization schemes with nested loops are employed, where the inner loop solves continuous variables using gradient-based solvers, and the outer loop handles discrete variables using heuristic algorithms. Other methods include branch and bound, branch and cut, outer approximation, and generalized Benders decomposition [50].

2.3.1 Approaches

Morante et al. [50] identify two types of LTTO approaches: analytical and numerical. Analytical methods provide closed-form or semi-analytical solutions for specific LTTO scenarios by simplifying assumptions such as constant thrust direction or specific boundary conditions [54], [55]. Analytical methods are seldom feasible for most LTTO problems, hence the numerical approach is the norm for low-thrust mission design [50]. Numerical methods are classified in direct methods, indirect methods and dynamic programming [50]. Each method may then be solved using either gradient-based, heuristic or hybrid algorithms [50].

Direct methods, such as direct collocation, transcribe the OCP into a nonlinear programming problem, allowing for the incorporation of complex constraints and mission requirements [56]. Direct methods are the most common LTTO approach, but they are computationally costly and rely on initial guesses of the final trajectory [50]. Less accurate but faster direct approaches have been developed, notably the Sims-Flanagan Transcription (SFT) scheme [50]. An important family of direct methods are those that derive control laws from predefined guidance schemes, which yield suboptimal solutions but are faster [50]. Indirect methods [49], [57] derive the necessary conditions for optimality using Pontryagin's Maximum Principle but are particularly sensitive to initial guesses and challenging for problems with path constraints [49]. The homotopy method is often used to generate progressively better initial guesses [58]. Dynamic programming methods solve optimal control problems by decomposing them into simpler subproblems, solving each recursively, and utilizing the principle of optimality to construct the overall solution [59]. Whiffen [60] developed the Static/Dynamic optimal Control (SDC) algorithm, a form of Differential Dynamic Programming (DDP) implemented in the software Mystic, which is considered state-of-the-art and was used in designing missions like NASA's DAWN [61]. Computational cost limits trajectories optimized by Mystic to 250 revolutions [50]. Other approaches trade-off accuracy in favor of faster generation [50]. Examples are Hybrid Differential Dynamic Programming (HDDP) [62], [63], and the DDP approach proposed by Aziz et al. [64] making use of a Sundman transformation, capable of optimizing geocentric trajectories of up to 2000 revolutions. Despite their robustness and flexibility, dynamic programming methods are computationally intensive, making them less practical for high-dimensional problems and conceptual design [50].

2.3.2 Lyapunov Control

Of the family of direct Low-Thrust Trajectory Optimization (LTTO) methods [49], [50] that make use of predefined control laws [50], [65], Lyapunov Control (LC) methods are notable for being both fast and able to generate reasonable estimates of optimal planetocentric trajectories [50]. Since their introduction by Ilgen in 1993 [66], LC methods have been extended to incorporate a variety of mission constraints [50], [67], [68]. A notable advantage of LC laws is that they naturally drive the spacecraft to the desired final state, removing the need to include boundary conditions on the final state [50]. The Q-Law in particular, introduced and refined by Petropoulos [67], [69], has been widely used for preliminary mission design [50] as well as to generate initial guesses for high-fidelity tools, notably JPL's Mystic [50], [70], [71]. In 2023 Narayanaswamy et al. [46] introduced the Rendezvous Q-Law

(RQ-Law), capable of dynamic six-element targeting by means of a semi-major axis augmentation scheme.

Q-Law

The Q-Law, originally introduced by Petropoulos [69], is a Lyapunov feedback control law for low-thrust trajectory optimization based on the “proximity quotient” Q, a candidate Lyapunov function that approximates the best quadratic time-to-go [69]. MEE formulations of the Q-Law were later developed by Petropoulos [67] and Varga [68]. Q is defined in Eq. 2.21,

$$\mathsf{Q}(\oe, \oe_T, W_x) = (1 + W_p P) \sum_{\oe} S_{\oe} W_{\oe} \left(\frac{\oe - \oe_T}{\dot{\oe}_{xx}} \right)^2, \quad \oe = a, f, g, h, k. \tag{2.21}$$

where the periapsis penalty P is defined in Eq. 2.22 and the element scaling factors $S_{\oe}$ are defined in Eq. 2.23.

$$P = \exp\left(k \left(1 - \frac{r_p}{r_{p_{\min}}} \right) \right) \tag{2.22}$$

$$S_{\oe} = \begin{cases} \left(1 + \left(\frac{|a - a_T|}{m a_T} \right)^n \right)^{\frac{1}{r}} & \oe = a, \\ 1 & \oe = f, g, h, k. \end{cases} \tag{2.23}$$

The feedback control law is derived such that $\dot{Q}$ is negative definite, and follows in Eq. 2.24. This follows from applying the chain rule $\dot{Q} = \nabla Q \dot{\boldsymbol{\oe}}$ and observing that $\dot{\boldsymbol{\oe}} = \Psi \mathbf{u}$, where Ψ stands for the Gauss variational equations in MEEs (Eq. 2.9).

$$\mathbf{u} = -\Psi^\top \nabla Q \tag{2.24}$$

The original law has 11 scalar parameters:

- W_P: periapsis penalty weight in Eq. 2.21
- $W_{\oe}$: element weights (for elements a, f, g, h and k) in Eq. 2.21
- $r_{p,\text{min}}$: minimum periapse radius $r_{p,\text{min}}$ in Eq. 2.22
- k: minimum periapse penalty parameter parameter in Eq. 2.22
- m, n and r: semi-major axis scaling parameters in Eq. 2.23.

Rendezvous Q-Law

The Rendezvous Q-Law (RQ-Law), proposed by Narayanaswamy [46], builds on the work of Lantukh et al. [72] extending the Q-Law to enable dynamic six-element targeting by means of a semi-major axis augmentation scheme (Eq. 2.25). The scheme is designed to induce an error in the semi-major axis related to the difference between the current and desired true longitude. This scheme adds two parameters to the feedback control law: the true longitude weight W_L and W_{scl}, which determines the slope of the induced error. The scheme becomes active after reaching 5-element convergence, splitting the manoeuvre in two phases: orbit acquisition and longitude acquisition, or phasing.

$$\oe_{T,\text{aug}} = \begin{cases} a_T + \frac{2W_L}{\pi} \left(a_T - \frac{r_{p,\text{min}}}{1 - \sqrt{f_C^2 + g_C^2}} \right) \tan^{-1}(W_{\text{scl}} \Delta L_{[-\pi,\pi]}), & \oe = a \\ \oe_T, & \oe \in \{f, g, h, k\} \end{cases} \tag{2.25}$$

Perturbations and Constraints

The most relevant perturbations for planetocentric trajectories are gravity field distortions, chiefly J_2 in the case of the Earth [13], [68]. Perturbing terms are seldom included in the formulation of the Q-Law, as the feedback control tends to waste fuel to counter-act the perturbing accelerations instead of taking advantage of them [67], [68]: instead, the parameters of the Q-Law are optimized to obtain near-optimal results under the effect of perturbations, using either conventional [68] or ML-based methods [73]. Eclipse and duty cycle constraints are also important for electric propulsion spacecraft [68].

Minimizing trajectory generation time is highly desirable, as a very large number of trajectories must be generated to train the NCO policy and assess the viability of NCO for space VRPs. The dynamicity of orbiting debris however, chiefly driven by the secular impact of the J_2 perturbation, is critical to the complexity of the STSP, and cannot be neglected [2]. To balance performance and realism this study considers unperturbed transfers between orbiting targets, while propagating the cloud according to the secular impact of the J_2 perturbation [13], [17].

Parameters, Weights and Tolerances

Tab. 2.1 lists the values of the Q-Law and RQ-Law parameters used. Tab. 2.2 lists the element weights and convergence tolerances used. The values of the parameters and weights are those recommended by Varga et al. [68] in the case of the Q-Law and by Narayanaswamy et al. [46] in the case of the RQ-Law. Note the presence of a relaxed tolerance. The relaxed tolerance is used to end the manoeuvre once it has been met 10 times, if convergence has not been achieved by that point. This is to avoid convergence issues, especially in the case of the RQ-Law [46]. Tolerances are set so that at worst (that is, in the case the relaxed tolerances are met 10 times) the Cartesian position error of an RQ-Law transfer between two average debris objects in the Iridium 33 cloud (average altitude of 700 km) will be of approximately 100 km, which is considered a reasonable distance at which to switch to rendezvous guidance strategies (such as relative navigation guidance schemes).

Table 2.1: Q-Law and RQ-Law parameter values.

k	m	n	r	b	W_p	W_l	W_{scl}
100.0	3.0	4.0	2.0	0.01	1.0	0.0594	3.6230

Table 2.2: Element weights and convergence tolerances.

Element	$W_{\text{œ}}$	$W_{\text{œ,phasing}}$	**Tolerance**	**Relaxed Tolerance**
a	1	10	1×10^{2} m	1×10^{3} m
e	1	1	1×10^{-3} deg	1×10^{-2} deg
i	1	1	1×10^{-3} deg	1×10^{-2} deg
Ω	1	1	1×10^{-3} deg	1×10^{-2} deg
ω	1	1	1×10^{-3} deg	1×10^{-2} deg
θ	—	—	1×10^{-1} deg	1 deg

Integrator Selection

The Tudat Space[1] astrodynamics library [74] is used to implement the simulator. Tudat supports a wide variety of integrators, including fixed and variable step Runge-Kutta (RK) integrators, Bulirsch-Stoer (BS) extrapolation integrators and variable-order Adams-Bashfort-Moulton (ABM) integrators. For further information about integrator families and particular integratrators refer to the Tudatpy documentation[2], Press et al. [75] and Montenbruck and Gill [44].

[1] https://docs.tudat.space/en/latest/

[2] https://docs.tudat.space/en/latest/_src_user_guide/state_propagation/propagation_setup/integration_setup.html

Table 2.3: Integrators available in the Tudat Space astrodynamics library. V.S.: variable-step.

Name	**V.S.**	**Description**
Forward Euler Method		First-order explicit method
Runge-Kutta 4th Order Method		Classic 4th-order Runge-Kutta method
Explicit Midpoint Method		Second-order explicit midpoint method
Explicit Trapezoid Rule		Second-order explicit trapezoid rule (Heun's method)
Ralston's Method (2nd Order)		Second-order Ralston's method
Runge-Kutta 3rd Order Method		Third-order Runge-Kutta method
Ralston's Method (3rd Order)		Third-order Ralston's method
SSPRK3 Method		Third-order Strong Stability Preserving Runge-Kutta method
Ralston's Method (4th Order)		Fourth-order Ralston's method
Runge-Kutta 3/8 Rule		Fourth-order Runge-Kutta 3/8-rule method
Heun-Euler Method	✗	Second-order Heun's method with embedded first-order Euler method
Runge-Kutta-Fehlberg 1(2) Method	✗	Second-order RKF method with embedded first-order method
Runge-Kutta-Fehlberg 4(5) Method	✗	Fifth-order RKF method with embedded fourth-order method
Runge-Kutta-Fehlberg 5(6) Method	✗	Sixth-order RKF method with embedded fifth-order method
Runge-Kutta-Fehlberg 7(8) Method	✗	Eighth-order RKF method with embedded seventh-order method
Dormand-Prince 8(7) Method	✗	Eighth-order Dormand-Prince method with embedded seventh-order method
Runge-Kutta-Fehlberg 8(9) Method	✗	Ninth-order RKF method with embedded eighth-order method
Runge-Kutta-Verner 8(9) Method	✗	Ninth-order Runge-Kutta-Verner method with embedded eighth-order method
Runge-Kutta-Feagin 10(8) Method	✗	Tenth-order RKF method with embedded eighth-order method
Runge-Kutta-Feagin 12(10) Method	✗	Twelfth-order RKF method with embedded tenth-order method
Runge-Kutta-Feagin 14(12) Method	✗	Fourteenth-order RKF method with embedded twelfth-order method
Bulirsch-Stoer Integrator	✗	Extrapolation integrator using Bulirsch-Stoer sequence
Deufelhard Integrator	✗	Extrapolation integrator using Deufelhard sequence
Adams-Bashforth-Moulton Integrator	✗	Variable order and step size predictor-corrector integrator of orders 6–12

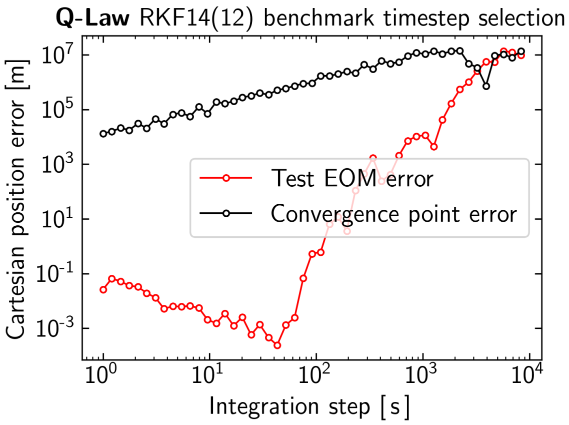


Figure 2.3: Benchmark step size selection. Test EOM error: error as of the last epoch of the tested (higher step size) trajectory. Convergence point error: distance between the last point of the tested and benchmark (lower step size) trajectories.

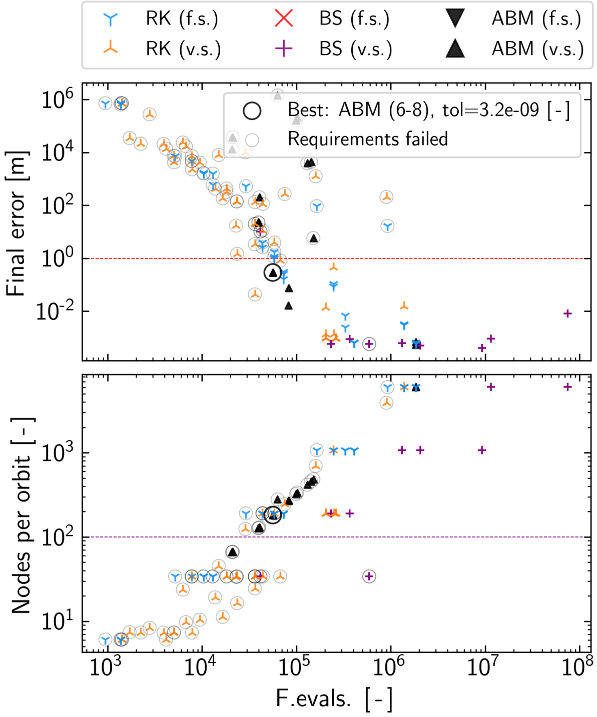


Figure 2.4: Integrator selection summary. Red line, top: final integration error requirement of 1 meter. Purple line, bottom: output density requirement of 100 nodes per orbit.

A trade-off considering all integrators available in Tudat was conducted to determine the best integrator for this use case. The complete list of Tudat integrators can be seen in Tab. 2.3. The selection of numerical integrator was based on the performance in simulating the average transfer in the RAAN walk [2] that traverses the Iridium 33 debris cloud.

A benchmark had to be constructed to measure the performance of the integrators. Benchmark selection was performed using the Q-Law, as Q-Law convergence is considerably more robust than RQ-Law phasing convergence [46]. The Runge-Kutta-Feagin 14(12) (RKF1412) integrator was chosen to generate the benchmark trajectory. The benchmark timestep was chosen as the largest time-step before the integration error of the benchmark becomes driven by numerical rounding error. An array of candidate benchmark timesteps **t** was created containing log-spaced timesteps between 10,000 [s] and 1 [s]. For each pair of timesteps $\mathbf{t}_i$ and $\mathbf{t}_{i+1}$ —where $\mathbf{t}_i > \mathbf{t}_{i+1}$—, a benchmark integration error was obtained as the difference in Cartesian position between the final state of the spacecraft in the $\mathbf{t}_i$ propagation, and the state of the spacecraft in the $\mathbf{t}_{i+1}$ at the final epoch of the $\mathbf{t}_i$ propagation. The latter value is obtained by interpolating the reference $\mathbf{t}_{i+1}$ propagation; Runge's phenomenon is avoided by discarding the last 5 timesteps of the interpolated $\mathbf{t}_{i+1}$ propagation. This error decreases as $\mathbf{t}_i$ and $\mathbf{t}_{i+1}$ become smaller until plateauing, meaning that further reduction of the benchmark's timestep does not improve integration error anymore. The result can be seen in Fig. 2.3. A benchmark step size of 4.5 [s] was selected. Note the rather large convergence point error seen in Fig. 2.3, it indicates that agreement about the convergence point is much harder to achieve than rounding-error level integration error. This is considerably dependent on the convergence tolerances, and is of no concern for benchmark selection.

Selection criteria are a maximum final Cartesian position error of 1 meter and a minimum node density of 100 nodes per orbit to ensure feasible and realistic interpolation of trajectories. Each integrator was evaluated under

Table 2.4: Performance of the 15 best integrators. Note how the 6 best integrators perform very similarly. Any of these would be fairly good choices for this use case.

Settings	Step Size Control	F. Evals [-]	Final Error [m]	Density [n/o]	CPU Time [s]
ABM 6-8	tol $= 3.2 \times 10^{-9}$	5.625×10^{4}	2.873×10^{-1}	1.820×10^{2}	2.148
ABM 6-9	tol $= 3.2 \times 10^{-9}$	5.627×10^{4}	2.857×10^{-1}	1.820×10^{2}	2.140
Ralston 3	dt $= 32\,\mathrm{s}$	5.840×10^{4}	9.768×10^{-1}	1.909×10^{2}	1.710
RK4	dt $= 32\,\mathrm{s}$	7.300×10^{4}	2.560×10^{-1}	1.909×10^{2}	2.810
RK4 3/8	dt $= 32\,\mathrm{s}$	7.300×10^{4}	1.624×10^{-1}	1.909×10^{2}	2.139
Ralston 4	dt $= 32\,\mathrm{s}$	7.300×10^{4}	3.079×10^{-1}	1.909×10^{2}	2.138
ABM 6-9	tol $= 1 \times 10^{-10}$	8.248×10^{4}	1.610×10^{-2}	2.678×10^{2}	3.171
ABM 6-8	tol $= 1 \times 10^{-10}$	8.345×10^{4}	7.553×10^{-2}	2.709×10^{2}	3.217
RKF4(5) higher	dt $= 32\,\mathrm{s}$	1.022×10^{5}	2.825×10^{-1}	1.909×10^{2}	3.593
RKF4(5) lower	dt $= 32\,\mathrm{s}$	1.022×10^{5}	2.870×10^{-1}	1.909×10^{2}	2.992
RKF5(6) higher	dt $= 32\,\mathrm{s}$	1.314×10^{5}	1.878×10^{-2}	1.909×10^{2}	3.837
RKF5(6) lower	dt $= 32\,\mathrm{s}$	1.314×10^{5}	1.944×10^{-2}	1.909×10^{2}	3.861
DP8(7) higher	dt $= 32\,\mathrm{s}$	2.044×10^{5}	1.318×10^{-3}	1.909×10^{2}	6.120
DP8(7) lower	dt $= 32\,\mathrm{s}$	2.044×10^{5}	1.374×10^{-2}	1.909×10^{2}	6.080
RKF7(8) lower	dt $= 32\,\mathrm{s}$	2.044×10^{5}	9.511×10^{-4}	1.909×10^{2}	5.822

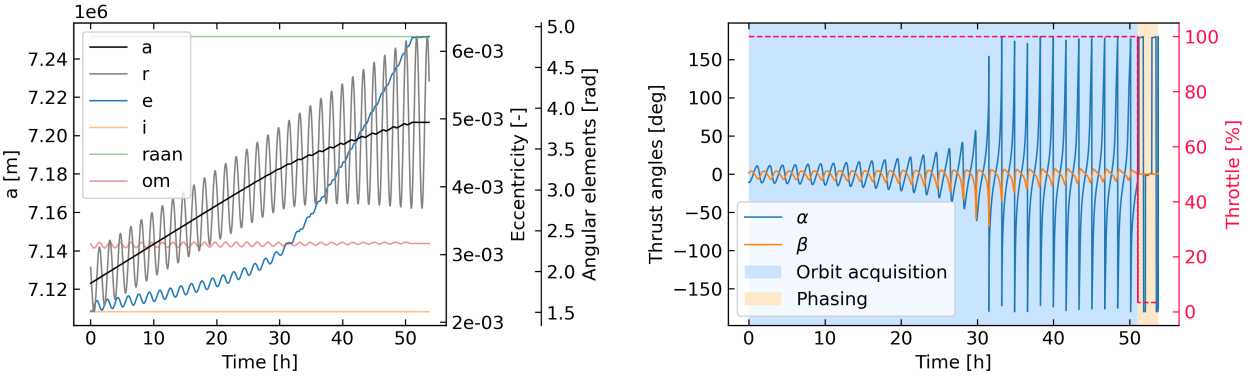
both fixed and variable step sizes. Fixed-step integrators were tested with five timesteps ranging from 1 second to 1000 seconds. Variable-step integrators were assessed using five tolerance levels (global and relative tolerances set equally) from 1×10^{-10} to 1×10^{-6}. Variable-order integrators were examined with lower orders 6 and 7, and higher orders 6, 8, and 9. Extrapolation integrators were evaluated using step counts from 1 to 10.

Integrators meeting the criteria were ranked based on the number of function evaluations, final error, node density, and CPU time, in that order. The performance of the top 5 integrators from each family is visualized in Fig. 2.4. The top 15 integrators are summarized in Tab. 2.4. The most performant integrator is the ABM integrator with variable orders 6-8 and variable step size, using a tolerance of 3.2×10^{-9}. Observe however that the 6 best integrators in the table perform very similarly. Any of the 6 best integrators in Tab. 2.4 would be a good choice for this use case. The ABM integrator is used for the rest of this work based on the lower number of function evaluations.

Fig. 2.5 shows the average transfer in the Iridium 33 RAAN walk. The transfer in question consists of raise of SMA of 84 km, a change in eccentricity of 4 1e—3, a change in inclination of 1 1e—3°, a change in RAAN of 2 79e—2°, and changes in AOP and True Anomaly (TA) of approximate 1.4°. The total manoeuvre time is of 55 hours, of which 51 are spent in the orbit acquisition leg and the phasing leg lasting under slightly under 4 hours. This will be a relevant consideration for transfer time estimation, which is the next topic of discussion. Lastly, Fig. 2.6 shows extreme transfers scenarios in the Iridium 33 and Fengyun 1C debris clouds, with the spacecraft traveling from the center-of-RCS of the cloud to the center-of-RCS of the cloud plus 3 times the standard deviation of each element in the cloud.

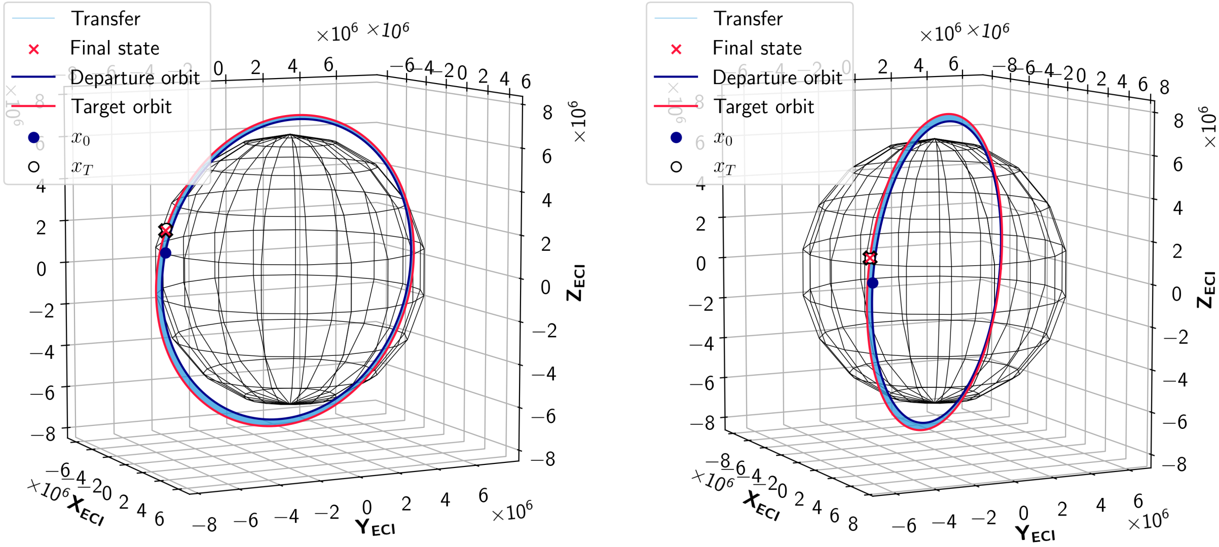
Transfer Cost Estimation

The capacity to quickly estimate the duration and cost (in ΔV or fuel mass) of low-thrust transfers *without the need to propagate* is highly attractive for NCO, as the training process requires estimating the cost of many (millions of) transfers. Fast, approximate transfer cost estimation methods are commonly used in conventional in STSP practice, especially for complex problems [13], [17]. Transfer cost estimation approaches (both impulsive and low-thrust) may be either analytical [5], [17], [76], which rely on simplifying assumptions to obtain closed-form expressions of transfer cost, or numerical, which rely on the pre-computation of many transfers, which are later



(a) (b)

Figure 2.5: Average transfer in the Iridium 33 debris cloud. (a) Keplerian element history through the transfer. (b) Control history; α and β are the in-ecliptic and out-of-ecliptic thrust angles with respect to the orbital velocity unit vector $\hat{\mathbf{e}}_\theta$ [46].



(a) (b)

Figure 2.6: Extreme 6-element rendezvous transfers in the Iridium 33 and Fengyun 1C debris clouds using the RQ-Law. Origin: cloud center-of-RCS. Target: cloud centroid plus 3 times the standard deviation of elements in the cloud. (a) Iridium 33. (b) Fengyun 1C.

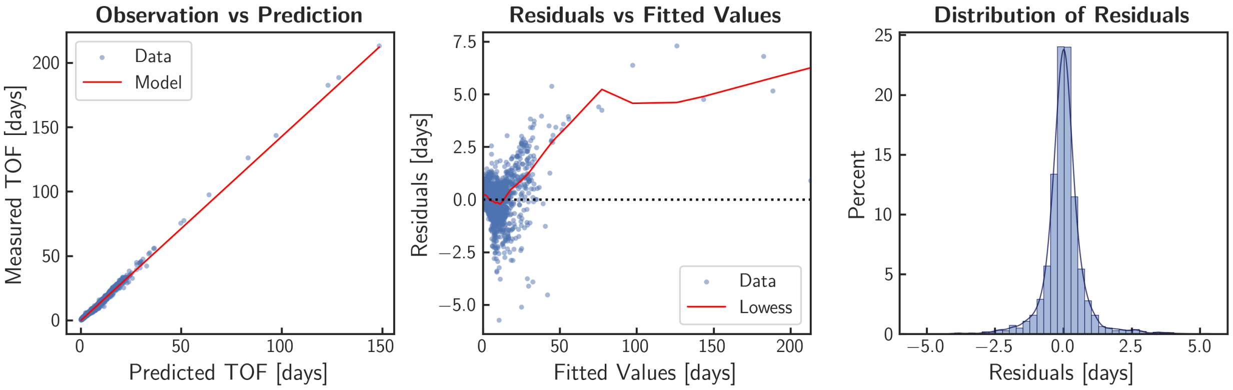


Figure 2.7: Estimation of RQ-Law TOFs using the Q proximity quotient. Histogram bin width (right side): 6 hours.

used to infer transfer costs in the optimization process: numerical approaches may either database-dependent [13], often relying on transfer window constraints, or based on multivariate regression. Deep Learning (DL) has been particularly successful for the latter [15].

A linear model based on the best time-to-go $\mathcal{T}$ (Eq. 2.26) is used to estimate the duration of RQ-Law transfers. The model is defined in Eq. 2.27. $\mathcal{T}$ follows from the definition of the proximity quotient Q, which approximates the best quadratic time-to-go [69].

$$\mathcal{T} = \sqrt{\mathrm{Q}} \tag{2.26}$$

The model in Eq. 2.27 is obtained by linear regression of measured TOFs using the RQ-Law with respect to predicted TOFs using the best time-to-go. The analysis was done considering all RAAN walk [2] transfers through the Iridium 33, Cosmos 2251 and Fengyun 1-C debris clouds. Both Q-Law and RQ-Law transfers were considered. The clouds are considered static through the tour to make the target dataset independent of the transfer strategy: static RAAN walk transfers are considered representative of possible transfers in LEO, and so valid for analysis. No instantaneous perturbations nor coasting phases are considered in these transfers, as discussed previously.

Tab. 2.5 reports the results of the linear regression analysis for both the Q-Law and RQ-Law. A strong positive correlation between the best time-to-go $\mathcal{T}$ and the measured TOF exists for both Q-Law and RQ-Law transfers (Pearson $r > 0.99$), indicating a strong linear relationship [77]. Fig. 2.7 shows predicted and observed RQ-Law TOFs and summarizes the performance of the linear model. Observe the strongly linear relationship of measured TOFs as a function of predicted TOFs, as well as the highly concentrated and highly symmetric distribution of residuals with mean at 0.

A linear model was fit for each strategy. Outlier TOFs, outliers outside of the 3σ range, were excluded. In both cases the linear model explains over 99% of the variance in observations ($R^2 > 0.99$), demonstrating an excellent fit [78]; the mean estimation error ε is close to 0 as well. Distribution similarity is verified using the Kolmogorov-Smirnov (KS) test [79]. In both cases the KS p-values indicate that there is no statistically significant difference between the estimated and observed TOF distributions.

Table 2.5: Goodness-of-fit analysis of the TOF models. ε: estimation error.

	Pearson r	**Slope**	**Intercept [h]**	**R-squared**	**KS statistic**	**KS p-value**	$\bar{\varepsilon}$ **[min]**	σ_ε **[min]**
Q-law	0.9972	1.4329	-15.9	0.995	2.55e-02	0.28	1.5	45.0
RQ-law	0.9970	1.4309	-9.72	0.994	2.31e-02	0.39	1.5	46.5

$$\text{TOF} = 1.4309\mathcal{T} + C; \quad C = -9.72 \text{ [hours]} \tag{2.27}$$

The required fuel mass $m_{f,\text{req}}$ is calculated by multiplying the estimated TOF by the constant fuel mass flow $\dot{m}$ (see Eq. 2.11). ΔV cost is estimated using Eq. 2.28, which assumes refuelling takes place when the spacecraft's fuel mass M_f is spent. The same approach is used to calculate the cumulative ΔV cost of complete tours. Critically, this model is suitable for the aforementioned debris clouds, using the specified RQ-Law parameters and tolerances. Application to other cases should follow careful analysis and verification.

$$\Delta V = \begin{cases} v_{\text{eq}} \log\left(\frac{m_0}{m_0 - m_{f,\text{req}}}\right) & m_{f,\text{req}} \leq M_f \\ n\Delta V_{\text{tank}} + v_{\text{eq}} \log\left(\frac{m_0}{m_0 - m_{\text{rem}}}\right), \quad \text{where} \quad \begin{cases} \Delta V_{\text{tank}} & = v_{\text{eq}} \log\left(\frac{m_0}{m_0 - M_f}\right) \\ n & = \left\lfloor \frac{m_{f,\text{req}}}{m_{f,\text{req}}} \right\rfloor \\ m_{\text{rem}} & = m_{f,\text{req}} - nM_f \end{cases} & \text{else.} \end{cases} \tag{2.28}$$

2.4 Heuristic Combinatorial Optimization

A modular STSP solver based on population-based Heuristic Combinatorial Optimization (HCO) is used to obtain near-optimal solutions. The choice for HCO is motivated by the widespread use of HCO methods to solve STSPs in literature [2]–[5], [7] and the availability of highly performant, open-source heuristic multi-objective optimization libraries such as `pygmo`[3] [80] and `pymoo`[4] [81] greatly eases the application, benchmarking and selection of diverse HCO algorithms for specific problem variants. The combinatorial optimization component is implemented using the `pygmo`, a parallel multi-objective global optimization library based on the Archipelago meta-heuristic [80].

2.4.1 Meta-heuristic

Combinatorial optimization meta-heuristics are algorithms that efficiently explore large discrete search spaces to find optimal or near-optimal solutions to combinatorial problems [82]. `pygmo` implements the Archipelago meta-heuristic [80]. The Archipelago algorithm evolves sub-populations, or "islands", concurrently, starting from different initial populations and possibly using different evolutionary strategies. Periodic migrations of individuals between islands promote diversity and prevent premature convergence, improving the algorithm's ability to avoid local optima and thoroughly explore the search space. This parallel and cooperative framework accelerates the optimization process and improves the robustness and quality of the solutions obtained [83].

2.4.2 Population Sampling

High quality population sampling is crucial for heuristic global optimization algorithms which are sensitive to initialization conditions [84]. Examples are population-based algorithms and algorithms with memory mechanisms such as Simulated Annealing [85]. Tab. 2.7 indicates which of the `pygmo` global optimizers which is sensitive to initialization conditions.

Population sampling in the case of CO problems involves sampling permutations $\sigma \in \mathfrak{S}^n$ of length n. Uniform permutation sampling is used to cover the search space as widely as possible. Sobol permutations [84] are used to obtain high quality uniform samples of $\mathfrak{S}^n$. Critically, advanced approaches and hand-crafted heuristics are often used in CO to determine near-optimal solutions prior to HCO [2], [13]. Distance-based permutation sampling is used to leverage known approximate solutions, using the Mallows Model [86]–[88] under the Hamming distance, which is defined as the number of positions at which two sequences differ [88], [89]. Empirical results show that a combination of both approaches is best to balance the exploration of the search space and the exploitation of known approximate solutions.

[3] `https://esa.github.io/pygmo2/`

[4] `https://pymoo.org/`

2.4.3 Permutation Encoding

Random keys permutation encoding [90] is selected for its versatility for optimizing complex discrete optimization problems across various domains [91], [92]. Permutations are encoded using vectors of continuous values, or random keys, in the $[0, 1)$ range. A permutation $\sigma \in \mathfrak{S}^n$ is encoded by generating a vector of random keys $\mathbf{x} \in [0, 1)^n$, sorting it, and permuting it by σ. Decoding is done by applying the standard `argsort` operation to $\mathbf{x}$, which takes an array of numerical values s and returns a permutation σ containing the index of each value of s in the sorted array s^*, such that the permutation of s by σ results in the sorted array s^*.

2.4.4 Approximate Solutions

Active Debris Removal

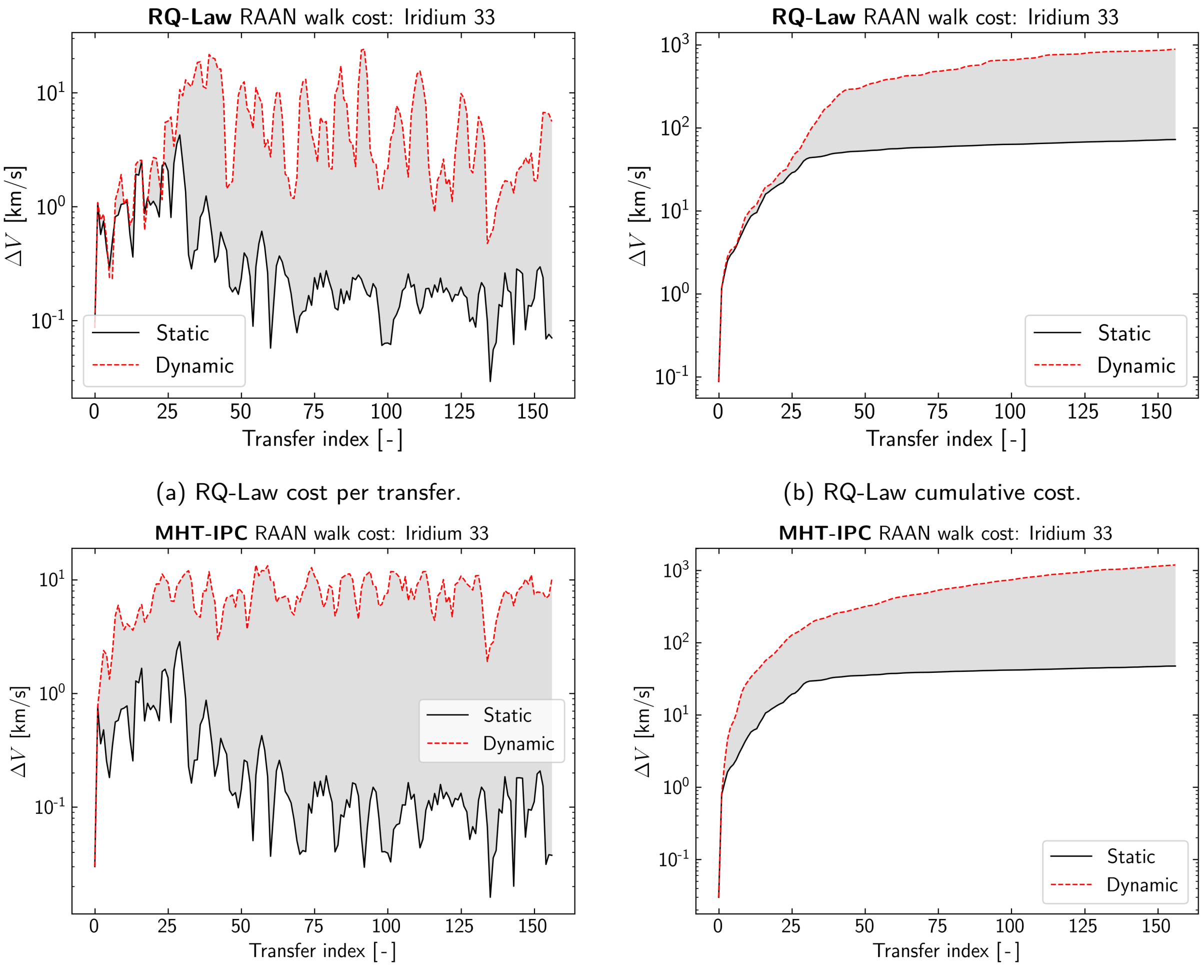
Izzo et al. [2] showed that the optimal solution of the STSP closely resembles a monotonically increasing RAAN walk. This result is intuitive as transfer cost is primarily driven by plane change cost, and plane change cost (Eq. 2.19) is primarily driven by the RAAN gap that must be closed [2], [5] for orbits with relatively high inclination. Most Earth orbiting spacecraft [93] and all the debris clouds under consideration (Fig. 1.1). The RAAN walk holds only for static orbiting targets however, and ADR missions in LEO are an example of a highly dynamic perturbed STSP due to the RAAN drift induced by the J_2 perturbation [2]. Fig. 2.8 shows the impact of RAAN drift on the cost of the RAAN walk through the Iridium 33 debris cloud, using the RQ-Law and MHT-NIC with a spacecraft with equal impulse (ΔV output over time): the increase in cost is dramatic. Furthermore, the optimal static and dynamic tours are not related, with their Spearman rank correlation quickly decreasing over time [2], as RAAN drift rates are independent of the original ranking (Eq. 2.6). Observe how the increase in tour cost is greater when using MHT-IPC manoeuvres, as transfers are considerably slower, and thus RAAN drift demerits the RAAN walk faster.

Two tree search approaches and one STSP routing heuristic are considered to obtain approximate solutions. The tree search approaches considered are Beam Search (BS) and Nearest-Neighbor (NN) search [94], [95], both considering transfer cost using a given transfer strategy. Fig. 2.9a and Fig. 2.9b show the cumulative cost of the RAAN walk, NN and BS tours through the Iridium 33 cloud, under RAAN drift, for the RQ-Law and MHT-IPC transfer strategies respectively. A detailed definition of the BS algorithm follows in Algorithm 3. The key concept to keep in mind is that the algorithm consists of a sequential expansion-pruning process with a fixed beam width w. At each step w active paths exist, each with a number m of available locations to be visited. The expansion step consists of calculating the cost of all possible transfers, from the last node of each active path to all available nodes, e.g. for $w = 8$ active paths and $m = 4$ nodes left to visit, the cost of all possible $w \cdot m = 32$ transfers is estimated. This results in 32 active paths. The pruning step consists of ranking these 32 paths by cost, and keeping the top w paths only. This results in a fixed number of active paths w which are kept for the next iteration, now with 3 targets left to visit. NN is effectively a BS of beam width equal to 1.

The STSP routing heuristic considered is the Dynamic RAAN Walk (DRW). The DRW is defined as a greedy search over a nearest-RAAN target ranking policy, which is performed sequentially in the dynamic STSP environment until all targets have been visited. Algorithm 2 describes the process in pseudocode. The benefit of using a heuristic over a search is runtime, as can be seen in Tab. 2.6. Despite its simplicity the DRW performs surprisingly well, as can be seen in Fig. 2.9a.

Table 2.6: Time required to generate an approximate solution for the Iridium 33 STSP of 167 transfers.

	RAAN walk	DRW	NN	BS ($w = 20$)
Runtime	0,2 ms	80,9 ms	2,1 s	79,3 s



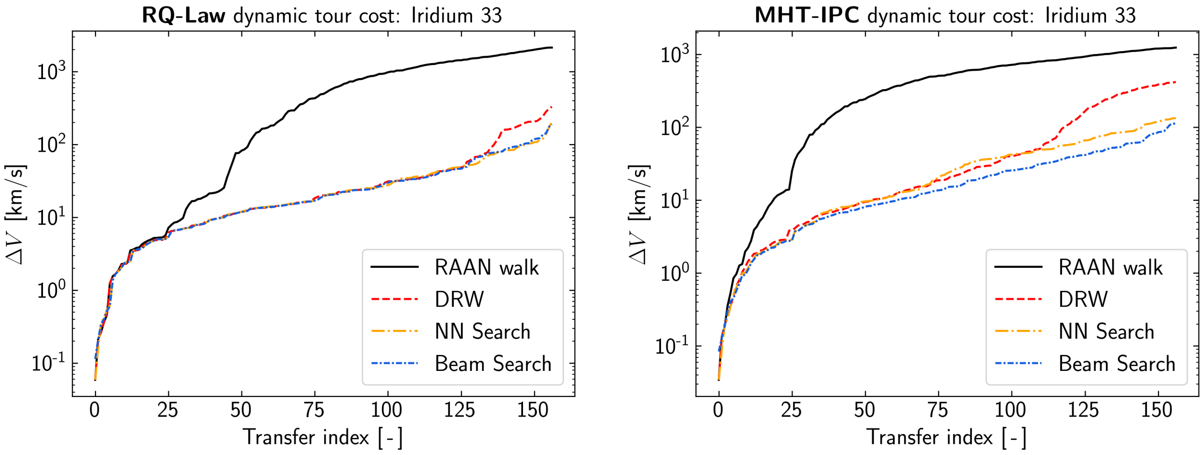
(a) RQ-Law cost per transfer.

(b) RQ-Law cumulative cost.

(c) MHT-IPC cost per transfer.

(d) MHT-IPC cumulative cost.

Figure 2.8: Impact of RAAN drift on the cost of the RAAN walk through the Iridium 33 debris cloud, using RQ-Law and MHT-IPC transfers.



(a) RQ-Law.

(b) MHT-IPC.

Figure 2.9: Cumulative cost of tours traversing the Iridium 33 debris cloud as a function of transfer index, under RAAN drift.

Algorithm 2: Dynamic RAAN Walk (DRW) for STSP Routing

```
Data: Starting state $x_0$, Set of targets $\mathbf{X} = \{x_1, x_2, \ldots, x_N\}$
Result: Optimal tour $p$, Total Delta V $c$
$p \leftarrow [x_0]$
$U \leftarrow \mathbf{X}$
$c \leftarrow 0$
while $U \neq \emptyset$ do
    Rank $U$ based on nearest RAAN to $p[-1]$
    $x_{\text{next}} \leftarrow$ first target in ranked $U$
    $\Delta V \leftarrow \text{EstimateDeltaV}(p[-1], x_{\text{next}})$
    $p \leftarrow p + [x_{\text{next}}]$
    $c \leftarrow c + \Delta V$
    $U \leftarrow U \setminus \{x_{\text{next}}\}$
return $p$, $c$
```

Beam Search is the superior approach, though choosing an optimal beam width is challenging, and the cost of the search quickly grows as beam width is increased. A Beam Search with a beam width of 2 is used to generate approximate solutions. Fig. 2.9 shows considerable improvement from using NN and BS for the Iridium 33 walk, for both RQ-Law and MHT-IPC transfers. BS yields the best improvement out of the three, especially for the latter.

Orbital Insertion

The OSSIE OTV STSP variant is concerned with changes in semi-major axis, inclination and phase. The primary cost driver in this case are inclination changes. Four hand-crafted heuristics are considered to provide candidate solutions: ascending and descending inclination and payload mass walks.

2.4.5 Heuristic Optimization Algorithm

Optimizer selection is conducted by trading off the all global heuristic optimizers available in pygmo based on their performance on the STSP variant at hand. All algorithms listed in Tab. 2.7 are considered in the trade-off. For

Algorithm 3: Beam Search in a dynamic graph (state transitions cause a change in the global state).

```
Input: $w$, $x_0$, $\mathbf{X} = \{x_1, x_2, \dots, x_N\}$, $k(x_i, x_j)$, $f(\mathbf{X}, T)$
Output: Optimal tour $p$
$B \leftarrow \{(0, [x_0], [0], [0])\}$          // Initialize min-heap with total cost, tour, TOFs and $\Delta V$s
for $t = 1$ to $N$ do
    $B' \leftarrow \emptyset$                     // Initialize min-heap $B'$ with size limit $w$
    for $(c, p, \mathbf{T}, □\mathbf{V})$ in $B$ do                                    // Expansion
        $\mathbf{X}' \leftarrow f(\mathbf{X}, \sum \mathbf{T})$                    // Propagate global state
        for $x_j \notin p$ do
            $p' \leftarrow p + [x'_j]$                                             // Update path
            $\mathbf{T}_j, □\mathbf{V}_j \leftarrow k(x_s, x_j)$      // Compute TOF and $\Delta V$ of current transfer
            $c' \leftarrow \sum □\mathbf{V}$                                // Compute tour $\Delta V$ so far
            $B' \leftarrow (c', p', \mathbf{T}, □\mathbf{V})$                          // Append node to heap
    while $|B'| > w$ do                                                            // Pruning to $w$
        Remove element with maximal $c'$ from $B'$          // Prune based on cumulative $\Delta V$
    $B \leftarrow B'$
Return $p$ from $B$ with minimal $c$
```

further information about specific algorithms refer to the pygmo capabilities page[5] and optimizer documentation pages. Critically, optimizers of the family of Evolutionary Strategies depend on internal sampling processes and are thus not sensitive to initial populations: in general terms other optimizers are preferrable if candidate solutions can be leveraged.

Heuristic Optimizer Selection for the ADR STSP

The choice of heuristic global optimizer for the ADR STSP was determined by a trade-off of all pygmo global optimizers on a reduced 50-transfer Iridium 33 ADR STSP. A 16-island archipelago is used with 80 individuals per island. 5 evolutionary cycles are carried out, each consisting of 500 generations of isolated evolution in each island followed by a migratory process between islands. Two variants of the optimization process are considered: one with uniformly sampled populations over all islands, and another with 8 of the 16 initial populations sampled around approximate Beam Search solutions using the Mallows Model under the Hamming distance. To ensure the trade-off is robust, the performance of each optimizer is evaluated by considering 10 optimizations using different random seeds.

Fig. 2.10 summarizes the performance of all the considered algorithms. The figure shows, for each optimizer, the performance of the best individual in the algorithm as a function of generation. This is shown for both the optimization starting from a uniformly sampled initial population, and the optimization starting from a population sampled in the proximity of the candidate solution obtained using the BS algorithm defined previously. Observe how the improvement that comes from sampling initial populations around approximate solutions is large across final performance, consistency, and convergence speed. The Simple Genetic Algorithm[6] shows both superior performance and consistency than any other option, consistently reaching the best tour found even when starting from a uniformly sampled population.

Heuristic Optimizer Selection for the OSSIE STSP

A similar process was followed to determine the choice of heuristic global optimizer for the OSSIE STSP. A 13-transfer mission scenario —the maximum number of transfers OSSIE is capable in its nominal mission scenario— was used to measure integrator performance. A 16-island archipelago is used with 80 individuals per island. 2 evolutionary cycles are carried out, each consisting of 50 generations of isolated evolution in each island followed

[5] https://esa.github.io/pygmo2/overview.html

[6] https://esa.github.io/pygmo2/.../sga

Table 2.7: Global optimizers available through the pygmo optimization library. SIC: sensitive to initial conditions.

Name	Codename	MO	SIC	Notes
Extended Ant Colony Optimization	GACO		✓	Utilizes pheromone trails for solution generation
Differential Evolution	DE		✓	Can initialize with a user-provided population
Self-adaptive DE	jDE and iDE		✓	Evolves from the initial population
Self-adaptive DE	pDE		✓	Evolves from the initial population
Grey Wolf Optimizer	GWO			Initializes with a wolf pack representing the population
Improved Harmony Search	IHS	✓	✓	Generates harmony memory independently
Particle Swarm Optimization	PSO		✓	Particles are initialized based on the provided population
Particle Swarm Optimization Generational	GPSO		✓	Particles are initialized based on the provided population
(N+1)-ES Simple Evolutionary Algorithm	SEA		✓	Employs internal sampling mechanisms
Simple Genetic Algorithm	SGA		✓	Directly utilizes the initial population
Corana's Simulated Annealing	SA		✓	Starts from a single initial guess, not a population
Artificial Bee Colony	ABC		✓	Initializes bee positions based on the provided population
Covariance Matrix Adaptation Evolution Strategy	CMA-ES			Can use an initial mean and population
Exponential Evolution Strategies	xNES			Uses its own internal sampling process
Non-dominated Sorting GA	NSGA2	✓	✓	Utilizes the initial population for multi-objective sorting
Multi-Objective EA with Decomposition	MOEA/D	✓	✓	Relies on the initial population for decomposed subproblems
Generational Multi-Objective EA with Decomposition	GMOEA/D	✓	✓	Uses the initial population in generational cycles
Multi-objective Hypervolume-based ACO	MHACO	✓	✓	Generates solutions based on hypervolume criteria
Non-dominated Sorting PSO	NSPSO	✓	✓	Initializes particles based on the provided population

by a migratory process between islands.

As previously optimizations with uniformly sampled populations and populations sampled around candidate solutions are performed. To ensure the trade-off is robust, the performance of each optimizer is evaluated considering 100 randomized 13-transfer STSP instances based on the nominal mission scenario of the OSSIE OTV.

Fig. 2.11 summarizes the performance of all the considered algorithms. The figure is equivalent to Fig. 2.10 in all but content. Observe how, in contrast with the 50-transfer ADR case, most optimizers are able to find the global optimum in this case. Generational Particle Swarm Optimization[7] is the most consistent algorithm in this case. The Exponential Evolutionary Strategies optimizer[8] is also a good option, displaying the fastest convergence.

Notably, the improvement that comes from sampling initial populations around approximate solutions is not large for this problem. The candidate solutions do not provide good approximations of optimal tours. This is caused by the highly localized nature of the targets, which are constrained to narrow semi-major axis and inclination bands, together with the constraints imposed on the initial and final stops of the tour —the nominal insertion orbit and nominal decomissioning orbit—, which results in an open-loop STSP the optimal path of which is hard to assess a priori.

Conclusions

Three fundamental conclusions are drawn from the study of HCO methods applied to the ADR and OSSIE OTV STSPs:

- Firstly, HCO is a robust and performant approach to solve small STSP instances up to 20 targets.
- Secondly, the robust performance achieved for smaller problems does not extend to larger and more dynamic STSPs, and neither problems with more convoluted constraints such as acceptance windows.
- Lastly, high quality approximate solutions are highly desirable for difficult STSP instances.

Larger problems with more convoluted constraints are of particular concern for the developing space industry

[7] https://esa.github.io/pygmo2/.../pso_gen

[8] https://esa.github.io/pygmo2/.../xnes

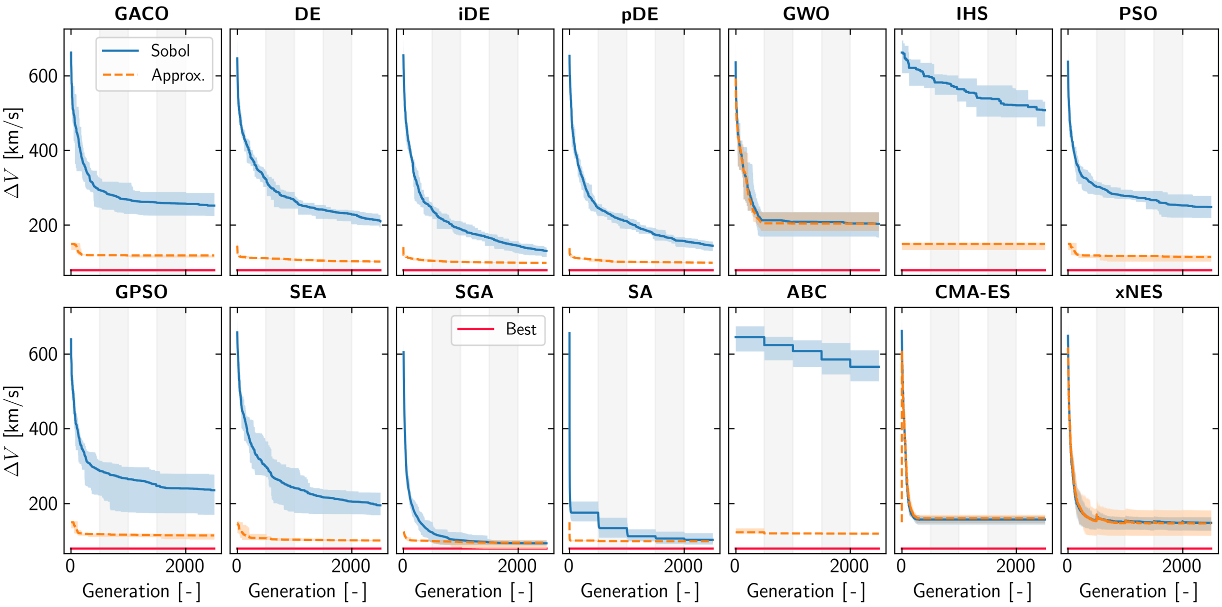


Figure 2.10: Performance of all pygmo optimizers on the dynamic Iridium 33 ADR STSP over 100 optimizations. Red line: best cost obtained across all cases, achieved with SGA. Grey bands: distinct evolutionary periods, 500 generations each. Blue: uniformly sampled initial populations. Orange: initial populations sampled around approximate solutions. Shaded area: best to worst cost achieved. Solid lines: mean cost.

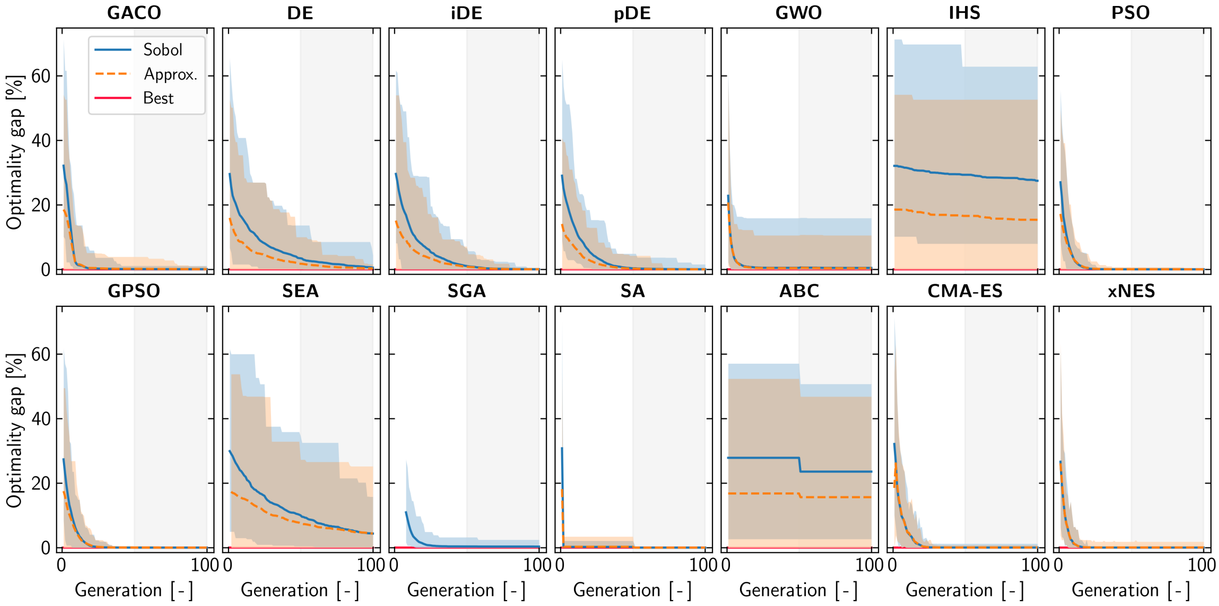


Figure 2.11: Performance of all pygmo optimizers on 100 different 13-transfer OSSIE STSP instances. Performance is represented by the optimality gap as different instances have different optimal costs. Red line: optimum, on average 26.72 [kg] of fuel mass spent with a standard deviation of 2.25 [kg]. Grey bands: distinct evolutionary periods, 50 generations each. Blue: uniformly sampled initial populations. Orange: initial populations sampled around approximate solutions. Shaded area: best to worst losses (tour costs) achieved. Solid lines: mean loss.

[19]. Problem variants considering multiple agents and constraints such as acceptance windows, split deliveries and others [18], which are common in operations research, are expected to grow in importance for spacecraft operations and logistics over time [19] as LEO and MEO become more congested, in-space manufacturing and servicing rise, and space logistics operations become as a consequence more complex [19].

The promise of NCO methods is to automate the process of discovering high quality heuristics for highly complex CO problems [16], [18]. It becomes pressing to ask whether these methods could be applied in the domain of space logistics as well.

The two problems studied so far provide a well-suited test set to assess the applicability and performance of NCO approaches for space logistics problems. The OSSIE STSP is a smaller, less demanding STSP which can be used to assess the feasibility of applying NCO in the space environment, and for which optimal solutions can be readily obtained to assess the performance of NCO solvers. The ADR STSP on the other hand is a far more challenging problem where NCO methods could yield real operational advantage in tandem with HCO approches.

2.5 Neural Combinatorial Optimization

NCO uses DNNs to automate the process of determining heuristics to solve CO problems. RL is the dominant paradigm for NCO, as supervised learning is often unfeasible for large or theoretically hard problems. NCO offers the attractive prospect of alleviating the scaling issues of exact approaches, while removing the need for handcrafted heuristics [18], and has shown promising performance on various CO problems [18], and has been shown to achieve high quality results, especially when coupled with advanced policy search procedures [16].

The `RL4CO`[9] NCO library is used to implement and train the STSP routing policy. `RL4CO` is a benchmark library for NCO based on PyTorch [96] with standardized, modular, and highly performant implementations of various environments, policies and RL algorithms, covering the entire NCO pipeline [18].

2.5.1 Environment

The state of the targets is described using Keplerian elements (which is practical and common practice when analyzing debris clouds [2]). The global state is propagated through the sequential decision-making process. This is done by estimating the transfer times, and using a secular perturbation model to propagate the state of the cloud. The Q-based TOF estimation method defined in Eq. 2.27 is used to estimate transfer times. The secular perturbation model used for LEO multi-rendezvous missions is the J_2 secular perturbation model defined in Eq. 2.6.

Active Debris Removal

The capacity to automatically generate large amounts of realistic Active Space Debris removal scenarios is fundamental for training. These scenarios are generated using statistical models of real debris clouds. The models are obtained by fitting cloud observations with the parametric statistical models that best match the observations. All 19 parametric statistical models available in PyTorch[10] [96] are considered. Goodness of fit is assessed using the Kolmogorov-Smirnov (KS) test [79]. The result is a composite model of the translational state and other properties of a debris cloud, comprising 6 or more parametric models: one for each Keplerian element, and more for other measurements such as radar-cross section if relevant. Fig. 2.12 shows the model generated for the Iridium 33 cloud. In this work the environment is limited to the translational state of the cloud.

The range of values which may be sampled by each model is limited to the range of observed values. This is achieved with inverse transform sampling [97] when the Inverse Cumulative Distribution Function (ICDF) of the parametric model is defined and implemented in PyTorch, and with vectorized rejection sampling [98] if the ICDF is not available.

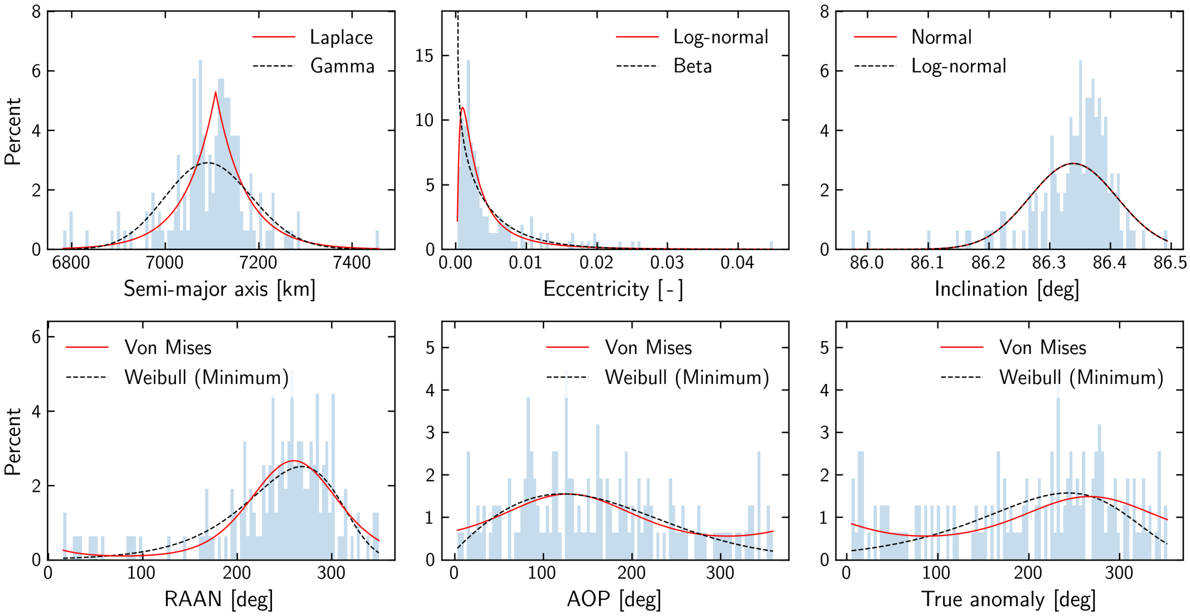


Figure 2.12: Statistical model of the translational state of the Iridium 33 debris cloud. Histograms: observations. Curve, red: parametric model that best matches the observations, where goodness of fit is measured using the KS statistic. Curve, black: second-best parametric model.

Table 2.8: Top: OSSIE insertion and decommissioning orbits. Bottom: statistical model describing expected payload Keplerian states. In commercial operations the target state of each payload is specified by the client. SSO stands for SSO inclination variance.

	Orbit	a [km]	e [-]	i [deg]	Ω [deg]	ω [deg]	θ [deg]	Mass [kg]
OSSIE	Insertion	500	0	97	158	0	0	
	Decommissioning	250	0	-	-	-	-	
Payload	Nominal	500	0	97	158	0	0	Variable
	Spread	50	0	SSO	180	180	180	15%
	Distribution	Uniform	Exponential	Uniform	Uniform	Uniform	Uniform	Exponential

OSSIE STSP

The generation of realistic mission scenarios for the OSSIE STSP is considerably simpler, as payload states are limited to the advertised payload delivery range of the OSSIE OTV. Tab. 2.8 describes the nominal mission scenario for OSSIE and the statistical model used to generate the target Keplerian states for a given payload. OSSIE payloads are highly likely to be destined for SSO orbits with altitude priority, but not necessarily. A worst case scenario is assumed, uniformly sampling payload inclinations in the entire SSO inclination range, defined as the range from $i_{\text{SSO}}(a_{\text{min}})$ to $i_{\text{SSO}}(a_{\text{max}})$, where $i_{\text{SSO}}(a)$ is the inclination required to achieve an SSO orbit (RAAN drift of 360°per year, see Eq. 2.6) at a given a. Payload masses are sampled from an exponential distribution, assuming a worst case scenario where payload masses are never below their nominal value: 6 kg for CubeSats, 1.5 kg for PocketQubes and 25 kg for small satellites. Payloads may be released individually or at once.

OSSIE payloads may be released individually or bundled with other payloads. This variable is both client-dependent and so highly unpredictable, and greatly impactful for mission cost as it determines the number of

[9] https://rl4.co
[10] https://pytorch.org/docs/stable/distributions.html

transfers that must be carried out. We model this by uniformly sampling number of bundles between 2 (all payloads in two bundles) and the total number of payloads, which is 13 for OSSIE (every payload launched to a distinct target orbit); payloads are assigned to each bundle uniformly at random.

2.5.2 Policy

An autoregressive Attention-based policy[11] is used to learn the routing problem. First introduced by Kool et al. [99], the policy encodes the input graph using a Graph Attention Network (GAT), and decodes the solution using a Pointer Network. Kool et al. train this policy using the REINFORCE RL algorithm to achieve considerably better performence than other learned heuristics [99]. Fraçois et al. find this policy to be a highly efficient learning component in their comprehensive analysis of learning performance in NCO methods [16].

2.5.3 Policy Search

Policy search strategies determine how actions are selected based on the learned policy, balancing exploration and exploitation to find optimal or near-optimal solutions. Policy search is fundamental for the performance of NCO algorithms [16]. We consider two policy search strategies implemented in `RL4CO`: greedy search and sampling.

Greedy Search

Greedy search is the simplest policy search strategy, where at each decision step, the action with the highest probability (or highest estimated reward) is selected deterministically. This approach is computationally efficient but may lead to suboptimal solutions due to its myopic nature, potentially getting trapped in local optima.

Algorithm 4: Greedy Search Strategy

```
Data: Trained policy $\pi_\theta$, initial state $s_0$
Result: Solution path
$s \leftarrow s_0$
$path \leftarrow []$
while not terminal state do
    Select action $a \leftarrow \arg\max_a \pi_\theta(a|s)$
    Append $a$ to $path$
    Transition to next state $s \leftarrow f(s, a)$
return $path$
```

Sampling Search

Sampling search introduces stochasticity by selecting actions based on the probability distribution defined by the policy. Instead of always choosing the most probable action, actions are sampled according to their probabilities, allowing for greater exploration of the action space. This strategy can escape local optima and explore diverse solution paths but may require more computational resources and time to converge to high-quality solutions.

Beam Search

Beam search strikes a balance between greedy and sampling strategies by maintaining a fixed number of the most promising partial solutions (beams) at each step. At each decision point, all possible extensions of the current beams are considered, and the top k beams with the highest cumulative probabilities are retained for the next step. This approach enhances exploration while controlling computational complexity, leading to better solution quality compared to greedy search without incurring the full cost of exhaustive search.

[11] `https://rl4.co/docs/.../AttentionModelPolicy`

Algorithm 5: Sampling Search Strategy

```
Data: Trained policy $\pi_\theta$, initial state $s_0$
Result: Solution path
$s \leftarrow s_0$
$path \leftarrow []$
while not terminal state do
    Sample action $a \sim \pi_\theta(\cdot|s)$
    Append $a$ to $path$
    Transition to next state $s \leftarrow f(s, a)$
return $path$
```

Algorithm 6: Beam Search Strategy [94], [95]

```
Data: Trained policy $\pi_\theta$, initial state $s_0$, beam width $k$
Result: Best solution path
$B \leftarrow \{(s_0, [])\}$
while not all beams have terminal states do
    $B' \leftarrow []$
    foreach beam $(s, path)$ in $B$ do
        foreach action $a$ in available_actions(s) do
            $p_a \leftarrow \pi_\theta(a|s)$
            $s' \leftarrow f(s, a)$
            $path' \leftarrow path \cup \{a\}$
            $B' \mathrel{+}= \{(s', path', p_a)\}$
    Sort $B'$ by cumulative probability in descending order
    $B \leftarrow$ top $k$ beams from $B'$
$path \leftarrow$ beam with the highest cumulative probability
return $path$
```

2.5.4 Reinforcement Learning Algorithms

Three RL algorithms are considered to train the routing policy: REINFORCE, Advantage Actor-Critic, and Proximal Policy Optimization.

Definition of Important Terms

RL literature makes use of a significant amount of non-trivial jargon terms. Please refer to Tab. 6 for the definition of all jargon terms that will be used in the following sections.

Stochastic Policy Gradient: The REINFORCE Algorithm

The REINFORCE algorithm[12], introduced by Williams in 1992 [100], is a Monte Carlo policy gradient method used to optimize stochastic policies. It is based on the principle of gradient ascent, aiming to maximize the expected return of a policy by adjusting the policy parameters, denoted by θ. The algorithm operates by iteratively generating full trajectories of interaction with the environment and using these to compute an unbiased estimate of the policy gradient. The algorithm does not require explicit knowledge of the environment's dynamics, instead relying on the experiences gathered through policy rollouts. A summary of the algorithm in pseudocode can be seen in Algorithm 7.

The process begins with the initialization of the policy parameters θ and a learning rate α. A trajectory $\tau = \{s_0, a_0, r_0, \ldots, s_T, a_T, r_T\}$ is sampled by following the current policy π_θ, where s_t represents the state at time step t, a_t is the action taken at that state, and r_t is the immediate reward received. This sequence of state-action-reward tuples is used to compute the return G_t for each timestep t, defined as:

$$G_t = \sum_{k=t}^{T} r_k \tag{2.29}$$

This return serves as the total cumulative reward from time step t to the end of the episode. For each timestep in the trajectory, the gradient of the log-likelihood of the taken action under the current policy, $\nabla_\theta \log \pi_\theta(a_t|s_t)$, is computed. This quantity measures how the likelihood of taking action a_t under the policy changes with respect to the policy parameters. The product of this gradient with the return G_t gives an estimate of how changes to the policy would impact the cumulative reward:

$$\nabla_\theta J(\theta) \approx \nabla_\theta \log \pi_\theta(a_t|s_t) \cdot G_t \tag{2.30}$$

The policy parameters θ are then updated via gradient ascent:

$$\theta \leftarrow \theta + \alpha \nabla_\theta \log \pi_\theta(a_t|s_t) \cdot G_t \tag{2.31}$$

This process is repeated for each timestep in the trajectory. The policy parameters are adjusted in the direction that increases the likelihood of actions leading to higher cumulative returns. The process continues, with new trajectories being sampled, until the policy converges, defined by negligible updates to θ over iterations.

Although REINFORCE provides an unbiased estimate of the policy gradient, the use of complete trajectories can introduce high variance into the gradient estimates, which can slow down convergence in practice. Various techniques, such as baselines or variance reduction methods, can be introduced to mitigate this issue, though they are not inherently part of the algorithm.

[12] REINFORCE stands for: Monte Carlo REward Increment = Nonnegative Factor × Offset Reinforcement × Characteristic Eligibility

Algorithm 7: REINFORCE Algorithm [100]

```
Data: Initial policy parameters $\theta$, learning rate $\alpha$
Result: Optimized policy parameters $\theta$
while not converged do
    Sample a trajectory $\tau = \{s_0, a_0, r_0, s_1, a_1, r_1, \ldots, s_T\}$ using policy $\pi_\theta$
    Compute the return $G_t = \sum_{k=t}^{T} r_k$ for each timestep $t$
    for each timestep $t$ in $\tau$ do
        Compute the gradient estimate $\nabla_\theta \log \pi_\theta(a_t|s_t) \cdot G_t$
        Update parameters: $\theta \leftarrow \theta + \alpha \nabla_\theta \log \pi_\theta(a_t|s_t) \cdot G_t$
```

Advantage Actor-Critic

The Advantage Actor-Critic (A2C) algorithm refines the classical REINFORCE method by incorporating a learned value function as a baseline to reduce the variance of the policy gradient estimates. This approach, initially formulated by Konda and Tsitsiklis [101], is designed to optimize both a policy and a value function simultaneously, referred to as the actor and critic, respectively. The actor represents the policy π_θ, parametrized by θ, while the critic estimates the state value function $V_\phi(s)$, parametrized by ϕ. The critic's role is to provide an estimate of the expected return from a given state, which serves as a baseline to stabilize the updates made to the actor. By subtracting the value function from the total return, A2C computes an advantage function that quantifies how much better or worse a specific action was compared to the expected outcome, thereby improving learning efficiency. A summary of the algorithm in pseudocode can be seen in Algorithm 8.

The algorithm begins by initializing the policy parameters θ, the value function parameters ϕ, and the respective learning rates α (for the policy) and β (for the value function). The interaction with the environment proceeds by sampling a trajectory $\tau = \{s_0, a_0, r_0, \ldots, s_T\}$ from the current policy π_θ. At each timestep t, the total return G_t, representing the cumulative reward from time t until the end of the trajectory, is computed as:

$$G_t = \sum_{k=t}^{T} r_k \tag{2.32}$$

Next, the critic estimates the value function at time t, denoted as $V_\phi(s_t)$. The advantage A_t at timestep t is then computed by subtracting the estimated value from the total return:

$$A_t = G_t - V_\phi(s_t) \tag{2.33}$$

This advantage quantifies the relative quality of the action taken compared to the critic's baseline estimate of the state's value. The policy parameters θ are updated by performing gradient ascent, using the gradient of the log-likelihood of the action taken under the current policy, scaled by the computed advantage:

$$\theta \leftarrow \theta + \alpha \nabla_\theta \log \pi_\theta(a_t|s_t) \cdot A_t \tag{2.34}$$

Simultaneously, the value function parameters ϕ are updated using the difference between the return and the value function estimate, which serves as the error signal for the critic:

$$\phi \leftarrow \phi + \beta(G_t - V_\phi(s_t))\nabla_\phi V_\phi(s_t) \tag{2.35}$$

This update step minimizes the squared error between the estimated and actual returns, allowing the critic to improve its estimation of the value function. The process repeats, iteratively updating the policy and value function parameters, until convergence, which occurs when the changes in the policy parameters become negligible.

A2C benefits from the reduced variance in gradient estimates due to the use of the value function as a baseline. However, it still relies on complete trajectories for updates, similar to REINFORCE, and is not inherently parallelized, as is its asynchronous counterpart A3C [102].

Algorithm 8: Advantage Actor-Critic Algorithm [101], [102]

```
Data: Initial policy parameters $\theta$, value function parameters $\phi$, learning rates $\alpha$, $\beta$
Result: Optimized policy parameters $\theta$ and value function parameters $\phi$
while not converged do
    Sample a trajectory $\tau = \{s_0, a_0, r_0, s_1, a_1, r_1, \dots, s_T\}$ using policy $\pi_\theta$
    for each timestep $t$ in $\tau$ do
        Compute the return $G_t = \sum_{k=t}^{T} r_k$
        Estimate the value $V_\phi(s_t)$
        Compute the advantage $A_t = G_t - V_\phi(s_t)$
        Update policy: $\theta \leftarrow \theta + \alpha \nabla_\theta \log \pi_\theta(a_t|s_t) \cdot A_t$
        Update value function: $\phi \leftarrow \phi + \beta(G_t - V_\phi(s_t)) \nabla_\phi V_\phi(s_t)$
```

Proximal Policy Optimization

Proximal Policy Optimization (PPO), proposed by Schulman et al. [103], addresses the instability and large policy update deviations found in traditional policy gradient methods like A2C. PPO introduces a surrogate objective function that constrains the magnitude of policy updates through a clipping mechanism, ensuring more gradual and stable learning. By enforcing limits on the size of the update steps, PPO effectively balances exploration and exploitation during training, making it suitable for a wide range of reinforcement learning applications. A summary of the algorithm in pseudocode can be seen in Algorithm 9.

The algorithm begins by initializing the policy parameters θ, the value function parameters ϕ, the learning rates α (for the policy) and β (for the value function), and the clipping parameter ϵ, which controls the extent to which policy updates are clipped. The training process involves iteratively sampling a batch of trajectories $\{\tau_i\}$ from the environment using the current policy π_θ. For each trajectory τ_i, and at each timestep t, the following steps are performed:

1. **Return Calculation:** The total return G_t at timestep t, defined as the sum of future rewards from time t until the end of the trajectory, is computed as:

$$G_t = \sum_{k=t}^{T} r_k$$

2. **Value Estimation:** The critic estimates the value of the current state s_t using the value function $V_\phi(s_t)$.

3. **Advantage Calculation:** The advantage A_t is then computed as the difference between the actual return and the estimated value:

$$A_t = G_t - V_\phi(s_t)$$

   This advantage function measures how much better the action a_t taken at state s_t performed compared to the expected value, thereby informing policy improvement.

4. **Probability Ratio:** The probability ratio $r_t(\theta)$ between the current policy and the previous policy $\pi_{\theta_{\text{old}}}$ is calculated:

$$r_t(\theta) = \frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_{\text{old}}}(a_t|s_t)}$$

   This ratio indicates how much the new policy diverges from the old one.

5. **Clipped Surrogate Objective:** To prevent excessively large policy updates, PPO uses a clipped surrogate objective function. This objective limits the change in the probability ratio by clipping $r_t(\theta)$ to lie within the range $[1-\epsilon, 1+\epsilon]$. The clipped objective is computed as:

$$L_t = \text{min}(r_t(\theta)A_t, \text{clip}(r_t(\theta), 1-\epsilon, 1+\epsilon)A_t)$$

   The objective ensures that updates only happen when the advantage is sufficiently large, and constrains the update if the ratio $r_t(\theta)$ deviates too much from 1.

6. **Accumulating Losses:** The policy loss is accumulated over all timesteps in the trajectory:

$$L_{\text{policy}} += -L_t$$

   The value loss, which penalizes the squared error between the estimated value and the return, is also accumulated:

$$L_{\text{value}} += (G_t - V_\phi(s_t))^2$$

7. **Parameter Updates:** After accumulating the losses over the entire batch of trajectories, the policy parameters are updated by performing gradient descent on the policy loss:

$$\theta \leftarrow \theta - \alpha \nabla_\theta L_{\text{policy}}$$

   Similarly, the value function parameters are updated based on the value loss:

$$\phi \leftarrow \phi - \beta \nabla_\phi L_{\text{value}}$$

This process repeats iteratively, with new batches of trajectories being sampled, until the policy and value function parameters converge.

PPO's clipped objective ensures more stable training by preventing overly large policy updates, and it has proven to be effective in various domains, from combinatorial optimization to control tasks in robotics.

Proximal Policy Optimization Variant for Autoregressive Policies

PPO is modified by Kool et al. [104] for use with autoregressive policies in combinatorial optimization tasks. In contrast to classical PPO, which treats actions independently in a Markov Decision Process (MDP), this variant models the entire autoregressive decoding process as a single decision step, capturing the sequential dependencies between actions. A summary of the algorithm in pseudocode can be seen in Algorithm 10.

This approach aligns with the Attention Model framework, streamlining training and solution generation in tasks with sequential decision-making. However, treating the decoding process as a single-step MDP introduces certain trade-offs:

- **Approximation Bias**: Temporal dependencies between actions are not explicitly modeled, which can lead to approximation bias.
- **Entropy Estimation**: Computing policy entropy over the full solution space is intractable, requiring Monte Carlo sampling and reducing entropy estimate accuracy.
- **Reduced Gradient Information**: The lack of multi-step transitions limits gradient information, potentially slowing policy improvement.

Algorithm 9: Proximal Policy Optimization Algorithm [103]

```
Data: Initial policy parameters $\theta$, value function parameters $\phi$, learning rates $\alpha$, $\beta$, clipping parameter $\epsilon$
Result: Optimized policy parameters $\theta$ and value function parameters $\phi$
while not converged do
    Sample a batch of trajectories $\{\tau_i\}$ using current policy $\pi_\theta$
    for each trajectory $\tau_i$ do
        for each timestep $t$ in $\tau_i$ do
            Compute the return $G_t = \sum_{k=t}^{T} r_k$
            Estimate the value $V_\phi(s_t)$
            Compute the advantage $A_t = G_t - V_\phi(s_t)$
            Compute the probability ratio $r_t(\theta) = \frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_{\text{old}}}(a_t|s_t)}$
            Compute the clipped objective: $L_t = \text{min}(r_t(\theta)A_t, \text{clip}(r_t(\theta), 1-\epsilon, 1+\epsilon)A_t)$
            Accumulate policy loss: $L_{\text{policy}} \mathrel{+}= -L_t$
            Accumulate value loss: $L_{\text{value}} \mathrel{+}= (G_t - V_\phi(s_t))^2$
    Update policy parameters: $\theta \leftarrow \theta - \alpha \nabla_\theta L_{\text{policy}}$
    Update value function parameters: $\phi \leftarrow \phi - \beta \nabla_\phi L_{\text{value}}$
```

Algorithm 10: PPO Variant for Autoregressive Policies [104]

```
Data: Initial policy parameters $\theta$, value function parameters $\phi$, learning rates $\alpha$, $\beta$, clipping parameter $\epsilon$
Result: Optimized policy parameters $\theta$ and value function parameters $\phi$
while not converged do
    Sample a batch of solutions $\{a^{(i)}\}$ for problem instances $\{x^{(i)}\}$ using policy $\pi_\theta$
    for each solution $a^{(i)}$ for instance $x^{(i)}$ do
        Compute the return $G^{(i)} = R(x^{(i)}, a^{(i)})$
        Estimate the value $V_\phi(x^{(i)})$
        Compute the advantage $A^{(i)} = G^{(i)} - V_\phi(x^{(i)})$
        Compute the probability ratio $r^{(i)}(\theta) = \frac{\pi_\theta(a^{(i)}|x^{(i)})}{\pi_{\theta_{\text{old}}}(a^{(i)}|x^{(i)})}$
        Compute the clipped objective: $L^{(i)} = \text{min}(r^{(i)}(\theta)A^{(i)}, \text{clip}(r^{(i)}(\theta), 1-\epsilon, 1+\epsilon)A^{(i)})$
        Accumulate policy loss: $L_{\text{policy}} \mathrel{+}= -L^{(i)}$
        Accumulate value loss: $L_{\text{value}} \mathrel{+}= (G^{(i)} - V_\phi(x^{(i)}))^2$
    Update policy parameters: $\theta \leftarrow \theta - \alpha \nabla_\theta L_{\text{policy}}$
    Update value function parameters: $\phi \leftarrow \phi - \beta \nabla_\phi L_{\text{value}}$
```

3 Verification, Validation and Robustness

The purpose of this chapter is to provide a unified discussion of verification, validation and robustness in the present work. The focus of this chapter are the four main modules developed in this work: the Impulsive Trajectory Design module, the Low Thrust Trajectory Design module, the Heuristic Combinatorial Optimization module and the Neural Combinatorial Optimization module. Refer to Appendix C for a detailed discussion of software architecture.

Section 3.1 discusses the verification and validation —where applicable— of all methods implemented in this work. The methods and procedures used to ensure experimental robustness across this research are presented in Section 3.2. Lastly, Section 3.3 discusses the sensitivity analysis methodology used to determine the impact of hyperparameters on the performance of Neural Combinatorial Optimization models.

3.1 Verification & Validation

Verification and validation are the most critical aspect of scientific software design. Without correctness guarantees for each method involved in a piece of research there can be no confidence in the results produced. Furthermore, the validation of the impulsive multi-rendezvous manoeuvres produced for the OSSIE OTV is of critical importance to answer Research Question 2: *What is a suitable approach for the design of multi-rendezvous trajectories with strict feasibility guarantees?*

Verification was of primary concern from the start of development. The development process used through this work adhered to the principles of test-driven development as outlined by Beck [105]. Refer to Appendix C for further details on the architecture of the software and the software development process. Down to its core, TDD can be summarized as: develop by testing. This research required the implementation of a large variety of compatible modules and methods, requiring a wide range of component and integration tests. The result is four large test sets verifying the four main modules developed in this work.

This section presents the verification test campaign of each of the four main modules. Section 3.1.1 discusses the verification of the Impulsive Trajectory Design module. Section 3.1.3 discusses the verification of the Low Thrust Trajectory Design module. Section 3.1.4 discusses the verification of the Heuristic Combinatorial Optimization module. Lastly, Section 3.1.5 discusses the verification of the Neural Combinatorial Optimization module.

3.1.1 Impulsive Trajectory Design

The experimental campaign for the Impulsive Trajectory Design (IPD) module focused on three fundamental areas: analytical trajectory design tests, sequential trajectory tests, and perturbation tests. The summary of all conducted tests is presented in Tab. 3.1

Test Targets and Outcomes

———— Analytical Trajectory Design Tests

Analytical trajectory design tests were conducted to verify the accuracy of Multiple Hohmann Transfer (IPD.T1) and inclination change (IPD.T2) implementations. These tests ensured that the analytical methods correctly compute MHT trajectories and accurately model inclination changes. The outcomes confirmed that the analytical implementations produced trajectories consistent with theoretical expectations, demonstrating the correctness of the trajectory design methods.

———— Sequential Trajectory Tests

Sequential trajectory tests involved validating the analytical sequence of impulsive maneuvers (IPD.T3). The objective was to ensure that the sequence of maneuvers generates trajectories that align with expected analytical predictions. The tests successfully demonstrated that the analytical sequences produce accurate trajectory outcomes, confirming the reliability of the maneuver sequencing processes.

———— Perturbation Tests

Perturbation tests aimed to verify that the analytical method can generate transfer trajectories considering perturbations, specifically by propagating them under perturbations analytically using the secular RAAN drift equation (IPD.T4). These tests evaluated whether the analytical implementation of the secular J_2 perturbation accurately accounts for the effects of perturbative forces in trajectory generation. The results indicated that the analytical implementation of secular J_2 perturbations matched the expected theoretical results, confirming the correct incorporation of perturbative effects into the trajectory design methods.

Conclusion

The experimental campaign for the IPD module successfully verified the functionality and performance of the four core analytical impulsive trajectory design methods: MHT, NIC, IPC and sequential impulsive trajectory design. All tests passed, confirming that the module operates as intended. These verifications establish the reliability of the IPD module's trajectory generation capabilities, providing a solid foundation for its integration into mission planning systems.

Table 3.1: Impulsive trajectory design test summary table. Pass criterions are to be read as follows. Successful completion: completion of the test with no errors; no assertions are made. Consistency (different types): assertion test; test result matches expected value. Inspection: manual inspection of test results; applied when the design of assertion tests is difficult.

Code	Component	Test rationale	Test criterion	Status
IPD.T1	**MHT**	Verify the analytical MHT trajectory design method	Analytical MHT results match expected theoretical trajectories	✓
IPD.T2	**NIC**	Verify the analytical NIC trajectory design method	Analytical NIC results match expected theoretical trajectories	✓
IPD.T3	**Sequential guidance policy**	Verify that the analytical sequence of maneuvers produces correct outcomes	Analytical sequential impulsive manoeuvre results match expected theoretical trajectories	✓
IPD.T4	**Secular perturbation model**	Verify that $J2$ perturbations are correctly accounted for	Analytical implementation of secular $J2$ perturbation matches expected theoretical results	✓

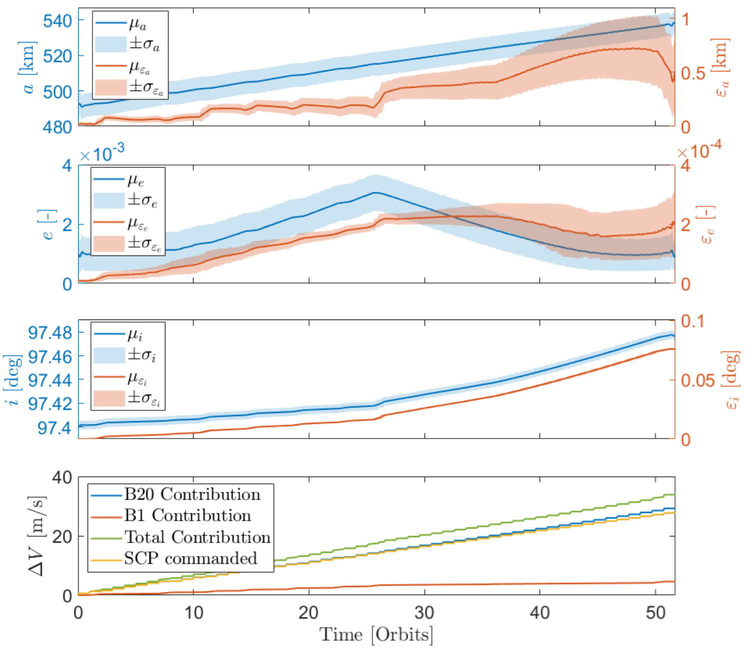


Figure 3.1: OSSIE FES validation results for a coplanar manoeuvre raising semi-major axis by 50 km. Top 3 panels: moving averages, 30 minute window.

3.1.2 Re-Optimization and Validation of Impulsive Trajectories

The SENER OSSIE Functional Engineering Simulator (FS) is the high-fidelity simulation environment serves as the core framework for modelling, testing and validating the guidance, navigation and control systems of the spacecraft. Matlab Simulink[106] is used to handle simulation, while Matlab[107] is used for pre- and post-processing tasks.

Translational dynamics are simulated using the Cowell propagator. Five natural perturbations are considered in the acceleration model: non-spherical Earth effects accounting for zonal harmonics of up to degree 6, atmospheric drag (density obtained from the U.S. Standard Atmosphere model), third-body perturbations from the Sun and Moon and solar radiation pressure. Spacecraft mass is propagated using a constant I_{sp} mass propagator.
Attitude dynamics are modeled using Euler's equation, which incorporates the effects of external torques: as OSSIE makes no use of internal momentum devices, internal torques are not considered in this analysis as they have a negligible magnitude in the actual spacecraft configuration. The key external torques considered are gravity gradient (modeled using the J2 gravitational coefficient), magnetic torques (perturbation by means of residual magnetic dIPDle moment), aerodynamic drag, and solar radiation pressure. Other potential torque sources, such as control system torques by means of the reaction control system, are modeled according to the precise thrust curves provided by the actuator constructed models.
The simulation is integrated using a fixed-step 4th order Runge-Kutta integrator running at 200 Hz (timestep of 5 ms), enabling accurate system characterization while maintaining sufficient simulation speed, to perform consequent testing campaigns.

The results of the preliminary validation of co-planar transfers in the FES are presented in Fig. 3.1: the first upper three plots depict the evolution in time of the orbital parameters during the simulation (in blue), demonstrating that the optimized trajectory was correctly followed, as contrasted by the error with respect to the SCP (in orange). The test shows that the achieved error range is compliant with client requirements. The last subplot presents a comparison between the ΔV consumption achieved by the FES and the SCP: the additional ΔV requested by GNC results from pointing correction maneuvres, to allign the B20 thrusters with the thrusting direction computed in the optimization framework.

The validation of non-coplanar and sequential transfers is an ongoing task taking place at SENER. No other validation results are available for inclusion in this work as of the time of writing. The validation results presented and preliminary results from in-house experimentation look promising. Research Question 2 receives a positive answer.

3.1.3 Low Thrust Trajectory Design

The testing campaign comprised sixteen distinct tests, each targeting specific functionalities of the LTTO module. The summary of these tests is presented in Tab. 3.2.

Verification Strategy

The verification process for the LTTO module focused on ensuring the correct and reliable operation of the Lyapunov feedback control guidance policies, specifically the Q-Law and RQ-Law. Given the complexity inherent in the analytical formulation of these guidance laws, a two-pronged verification approach was adopted:

1. **Custom Propagator Verification**: The Q-Law and RQ-Law were initially verified using a custom propagator. This approach provided enhanced control over each step of the trajectory propagation process, facilitating a detailed assessment of the guidance policies' implementation.

2. **Tudat Integration Verification**: Subsequently, the guidance policies were integrated into the Tudat simulation environment. Propagation results obtained from the custom propagator were compared against those from Tudat to ensure coherence and consistency between the two systems. Prior to verifying the guidance policies, both the custom propagator and the integrators were independently validated against Tudat to establish baseline accuracy.

This verification strategy ensured a comprehensive evaluation of the LTTO module's functionality, addressing both internal implementation accuracy and external integration compatibility.

Test Targets and Outcomes

——— Orbital Element Transformations (LTTO.T1)

Test Rationale: To ensure that orbital element transformations introduce minimal error.

Outcome: Transformation consistency was achieved, as all relevant tests completed without discrepancies.

——— Propagator Verification

- **Cowell Propagator (LTTO.T2):** The custom Cowell propagator was verified against Tudat using constant axial force trajectories with the Euler integrator. The Cartesian position error remained below 1e-7 meters after 30 hours, meeting the pass criterion.

- **MEE Propagator (LTTO.T3):** Similarly, the custom MEE propagator was validated against Tudat under the same conditions, achieving the required accuracy.

- **MEE Multitype Propagator (LTTO.T4):** The multitype MEE propagator was tested using Q-Law trajectories with the Euler integrator, maintaining Cartesian position errors below 1e-7 meters after 30 hours.

- **MEE with Propagator (LTTO.T5) and MEE with Multitype Propagator (LTTO.T6):** Both configurations were successfully verified against Tudat, adhering to the stringent error thresholds.

——— Integrators Verification

- **Integrators (LTTO.T7 to LTTO.T9):** The MEE propagator, along with RK4 and RK56 integrators, were rigorously tested against Tudat. All integrators maintained Cartesian position errors below 1e-7 meters after 30 hours. Additionally, inner step state propagation within custom integrators was verified, ensuring accurate state transitions.

Guidance Policy Integration

- **Q-Law and RQ-Law in Tudat (LTTO.T10 and LTTO.T11):** Both guidance policies were correctly implemented and successfully integrated with Tudat for trajectory simulations. The tests concluded with successful completions and inspections.
- **Simulator Test (LTTO.T12):** The MEE multitype propagator, along with RK4 and RK56 integrators, was validated using a mean Q-Law transfer over the Iridium 33 cloud. The Cartesian position error was maintained below 1e-5 meters after 50 hours, satisfying the pass criterion.
- **Guidance Policy Inspections (LTTO.T13 and LTTO.T14):** Inspections of the Q-Law and RQ-Law implementations using the custom propagator and integrator confirmed the correct functioning of the ADR trajectories and guidance policies.

Decoupled Guidance Policy Verification

- **Decoupled RQ-Law (LTTO.T15 and LTTO.T16):** Both decoupled implementations of the Q-Law were verified using custom propagators and within the Tudat propagator framework. The tests were successfully completed and inspected, ensuring the decoupled guidance policies operate as intended.

Conclusion

The LTTO module's test campaign successfully fulfilled all defined pass criteria across sixteen distinct tests. The comprehensive verification process demonstrated the module's robust implementation and reliable performance in trajectory generation, orbital element transformations, and thoroughly verified the Q-Law and RQ-Law guidance policies using both custom and Tudat propagators. Test outcomes verify the robust implementation and reliable performance of the module's functionalities. Detailed discussions and in-depth analyses of these findings are documented in subsequent sections of this work.

Table 3.2: Low thrust trajectory design test summary table.

Code	Component	Test rationale	Pass criterion	Status
LTTO.T1	**Orbital element transformations**	Ensure that orbital element transformations introduce minimal error	Transformation consistency	✓
LTTO.T2	**Cowell propagator**	Verify custom Cowell propagator against Tudat using constant axial force trajectories (Euler integrator)	Cartesian position error below 1e-7 meters after 30 hours	✓
LTTO.T3	**MEE propagator**	Verify Custom MEE propagator against Tudat using constant axial force trajectories (Euler integrator)	Cartesian position error below 1e-7 meters after 30 hours	✓
LTTO.T4	**MEE multitype propagator**	Verify multitype MEE propagator using Q-Law trajectories (Euler integrator)	Cartesian position error below 1e-7 meters after 30 hours	✓
LTTO.T5	**MEE with a propagator**	Verify MEE with a propagator against Tudat using constant axial force trajectories (Euler integrator)	Cartesian position error below 1e-7 meters after 30 hours	✓
LTTO.T6	**MEE with a multitype propagator**	Verify multitype MEE with a propagator against Tudat using constant axial force trajectories (Euler integrator)	Cartesian position error below 1e-7 meters after 30 hours	✓

Continued on next page

Table 3.2: Low thrust trajectory design test summary table. (Continued)

Code	Component	Test rationale	Pass criterion	Status
LTTO.T7	**Integrators**	Verify MEE propagator and RK4 and RK56 integrators against Tudat using constant axial force trajectories (Euler integrator)	Cartesian position error below 1e-7 meters after 30 hours	✓
LTTO.T8	**Integrators**	Verify MEE multitype propagator and RK4 and RK56 integrators against Tudat using constant axial force trajectories	Cartesian position error below 1e-7 meters after 30 hours	✓
LTTO.T9	**Integrators**	Verify inner step state propagation in custom integrators	Cartesian position error below 1e-7 meters after 30 hours	✓
LTTO.T10	**Q-Law Tudat guidance policy**	Verify Q-Law is correctly implemented and integrates with Tudat software for trajectory simulations	Successful completion, inspection	✓
LTTO.T11	**RQ-Law Tudat guidance policy**	Verify RQ-Law is correctly implemented and integrates with Tudat software for trajectory simulations	Successful completion, inspection	✓
LTTO.T12	**Simulator**	Verify MEE multitype propagator and RK4 and RK56 integrators against Tudat using mean Q-Law transfer over Iridium 33 cloud	Cartesian position error below 1e-5 meters after 50 hours	✓
LTTO.T13	**Q-Law guidance policy**	Inspect Q-Law ADR trajectories using custom propagator and integrator	Successful completion, inspection	✓
LTTO.T14	**RQ-Law guidance policy**	Inspect RQ-Law implementation using custom propagator	Successful completion, inspection	✓
LTTO.T15	**Decoupled RQ-Law guidance policy**	Verify decoupled implementation of Q-Law using custom propagator	Successful completion, inspection	✓
LTTO.T16	**Decoupled RQ-Law Tudat guidance policy**	Verify decoupled implementation of Q-Law in Tudat propagator	Successful completion, inspection	✓

3.1.4 Heuristic Combinatorial Optimization

The experimental campaign for the Heuristic Combinatorial Optimization (HCO) module focused on 15 components. The aim of this campaign was to verify the correct implementation of all components, and the correct integration of all components, resulting in a considerable collection of tests. The summary of all tests conducted to verify the HCO module is presented in Tab. 3.3.

Test Targets and Outcomes

——— Archipelago

Function: Manages multiple evolutionary populations that evolve in parallel and periodically exchange individuals, enhancing genetic diversity and exploration of the solution space.

Tests:

- **HCO.T1 (Archipelago HCO meta-heuristic):** Verified the functionality of the archipelago framework. The framework executes without errors and properly manages populations..

——— Population

Function: Handles the creation, encoding, and decoding of solution populations for optimization algorithms.

Tests:

- **HCO.T2 (Creation, encoding and decoding of populations):** Verified population handling. Populations are created, encoded, and decoded correctly without errors..
- **HCO.T3 (Encoding scheme correctness):** Ensured encoding consistency. Transformations between encodings are accurate..
- **HCO.T7 (Archipelago populations with sampling methods):** Verified that archipelago populations are correctly sampled using both uniform and distance-based methods. Populations are diverse and consistent with sampling methods..

——— Permutation Sampling

Function: Generates initial populations by sampling permutations of possible solutions, aiding in thorough exploration of the solution space.

Tests:

- **HCO.T5 (Uniform permutation sampling):** Verified uniform sampling. Permutations are sampled uniformly without bias..
- **HCO.T6 (Distance-based permutation sampling):** Verified distance-based sampling. Sampling prioritizes permutations based on the Hamming distance..

——— Traveling Salesman Problem (TSP)

Function: Implements the classical TSP as a baseline for route optimization.

Tests:

- **HCO.T4 (Classical TSP problem class):** Verified classical TSP implementation. Solves standard TSP instances accurately..

——— Postprocessing

Function: Processes and logs the results of optimization runs, including evolutionary histories.

Tests:

- **HCO.T8 (STSP evolution logging):** Ensured accurate logging. Evolutionary data is logged correctly..
- **HCO.T9 (Sparse Evolutionary Histories Test):** Confirmed history consistency. Sparse records match dense records..

——— Visualization

Function: Provides graphical representations of optimization processes and results.

Tests:

- **HCO.T10 (STSP Local Evolution Visualization Test):** Verified island-level evolution visuals. Visualizations are accurate, complete, and effectively communicate results..
- **HCO.T11 (STSP Evolution Visualization Test):** Verified global evolution visuals. Visualizations are accurate, complete, and effectively communicate results..
- **HCO.T12 (STSP Evolution Visualization with Initial Guesses Test):** Verified visualization of initial guess impact on optimization performance. Visualizations correctly incorporate initial guesses and convey impact effectively..
- **HCO.T13 (Gabbard Diagram Test):** Verified Gabbard diagrams. Diagrams correctly represent trajectory parameters and effectively convey relevant information about each debris cloud..

——— STSP

Function: Extends the classical TSP to the STSP, adapting it for space mission planning where orbital mechanics are considered.

Tests:

- **HCO.T14 (STSP global optimization):** Verified correctness of the generalized STSP problem class and assessed global optimization performance. Algorithms find optimal solutions efficiently..

——— INTEGER STSP

Function: Implements the STSP using integer solution encoding, for the purpose of testing integer global optimization algorithms.

Tests:

- ✗ **HCO.T15 (Integer STSP global optimization):** Verified correctness of the generalized Integer STSP problem class and assessed global optimization performance. Negative outcome: the integer optimization algorithms in `pygmo` could not achieve any progress on this variant of the problem. This reinforced the already strong choice of using random-keys permutation encoding [91], [92] to embed decision vectors into a continuous space. Refer to Appendix B for further details on permutation encoding and sampling.

——— OPTIMIZERS

Function: Incorporates various optimization algorithms to solve the STSP and its variants.

Tests:

- **HCO.T16 (STSP global optimization with Simulated Annealing):** Assessed Simulated Annealing performance. Converges to high-quality solutions..
- **HCO.T17 (STSP local optimization):** Tested local optimization capability. Succeeds in locally improving solutions..
- **HCO.T18 (STSP Monotonic Basin Hopping for local optimization):** Evaluated Basin Hopping effectiveness. Escapes local optima, finds superior solutions, but overall less efficient than global optimizers..

——— RAAN WALK AND INITIAL GUESSES

Function: Manages the progression of the Right Ascension of the Ascending Node (RAAN) in orbital trajectories and utilizes initial guesses to enhance optimization.

Tests:

- **HCO.T19 (RAAN walk test):** Verified RAAN walk optimality. RAAN walk is a near-optimal policy for static STSP as expected [2]..

- **HCO.T20 (STSP Initial Guesses Test):** Verified initial guess effectiveness. Initial guesses improve optimization metrics..

Low-Thrust Trajectory Optimization (LTTO) Module

Function: Generates trajectories using low-thrust propulsion models, specifically the Q-Law and RQ-Law, essential for efficient mission planning.

Tests:

- **HCO.T21 (Q-Law ADR trajectory generation):** Verified Q-Law ADR trajectory generation. Trajectories meet accuracy and efficiency criteria..
- **HCO.T22 (RQ-Law ADR trajectory generation):** Verified RQ-Law ADR trajectory generation. Trajectories meet accuracy and efficiency criteria..

Trajectory Estimation

Function: Estimates costs and parameters of various orbital transfer trajectories, including low-thrust and impulsive maneuvers.

Tests:

- **HCO.T23 (MHT trajectory estimation):** Verified MHT transfer cost estimation. Estimations align with theoretical models..
- **HCO.T24 (MHT-IPC trajectory estimation):** Verified MHT-IPC transfer cost estimation. Accurate cost and parameter assessments..
- **HCO.T25 (MHT-NIC trajectory estimation):** Verified MHT-NIC transfer cost estimation. Estimations are accurate and consistent..
- **HCO.T26 (RQ-Law trajectory estimation):** Verified RQ-Law transfer cost estimation. Estimations match theoretical models..
- **HCO.T27 (MHT-IPC optimized trajectory estimation):** Verified just-in-time (JIT) compiled MHT-IPC transfer cost estimation. Optimized estimations show improved metrics..
- **HCO.T28 (Q-Law optimized trajectory estimation):** Verified JIT-compiled Q-Law transfer cost estimation. Optimized estimations meet performance metrics..
- **HCO.T29 (MHT-IPC dynamic STSP tour cost estimation):** Verified JIT-compiled dynamic MHT-IPC tour cost estimation. Cost estimations reflect mission requirements accurately..
- **HCO.T30 (Q-Law dynamic STSP tour cost estimation):** Verified JIT-compiled dynamic Q-Law tour cost estimation. Cost estimations are accurate and consistent..

Heuristic Search Methods

Function: Implements heuristic methods like Nearest Neighbor and Beam Search to construct initial solutions and improve optimization performance.

Tests:

- **HCO.T31 (Dynamic Q-Law STSP Nearest Neighbor search):** Tested nearest neighbor heuristic. Produces high-quality initial solutions..
- **HCO.T32 (MHT-IPC Q-Law STSP Nearest Neighbor search):** Assessed heuristic with MHT-IPC. Generates effective initial solutions..
- **HCO.T33 (Dynamic Q-Law STSP Beam Search):** Verified beam search implementation. Explores multiple paths, improves solution quality..

——— Generalized STSP

Function: Addresses complex versions of the STSP, incorporating factors like variable payload masses, dynamic environments, and different propulsion models.

Tests:

- **HCO.T34 (Static Q-Law STSP global optimization):** Tested static global optimization. Identifies optimal solutions correctly..
- **HCO.T35 (Dynamic Q-Law STSP global optimization using initial guesses):** Verified dynamic optimization with initial guesses. Improved convergence and solution quality..
- **HCO.T36 (Static MHT-IPC STSP global optimization):** Assessed static MHT-IPC optimization. Finds optimal solutions aligning with requirements..
- **HCO.T37 (Dynamic MHT-IPC STSP global optimization using initial guesses):** Tested dynamic MHT-IPC optimization with initial guesses. Enhanced performance metrics achieved..
- **HCO.T38 (Mass-varying payload generation):** Ensured variable payload handling. Payloads generated accurately and consistently..
- **HCO.T39 (Mass-varying tour cost estimation):** Verified cost estimation with varying payloads. Cost estimations accurately reflect payload variations..
- **HCO.T40 (OSSIE STSP global optimization):** Tested OSSIE STSP optimization performance. Solves complex scenarios meeting all criteria..

——— Active Debris Removal Coasting Transfer Policy (ADR CTP)

Function: Implements coasting strategies in low-thrust trajectory planning to optimize fuel consumption and mission duration in ADR missions.

Tests:

- **HCO.T41 (RQ-Law coasting strategy for the dynamic STSP):** Verified RQ-Law coasting strategy. Optimizes fuel consumption and mission duration correctly..

Conclusion

The experimental campaign for the HCO module successfully verified the functionality and performance of its key components. All tests, except HCO.T15, passed, confirming that the module operates as intended. Test HCO.T15 failed due to the inability of integer optimization methods to make progress on the problem variant, reinforcing the choice of using random-keys permutation encoding. These verifications establish the reliability of the HCO module's optimization capabilities, providing a solid foundation for its integration into mission planning systems.

Table 3.3: Heuristic Combinatorial Optimization test summary table.

Code	Component	Test rationale	Test criterion	Status
HCO.T1	**Archipelago**	Verify archipelago framework functionality	Executes without errors, proper population management	✓
HCO.T2	**Population**	Verify population handling	Correct creation, encoding, decoding without errors	✓
HCO.T3	**Population**	Ensure encoding consistency	Accurate transformations between encodings	✓

Continued on next page

Table 3.3: Heuristic Combinatorial Optimization test summary table. (Continued)

Code	Component	Test rationale	Test criterion	Status
HCO.T4	**TSP**	Verify classical TSP implementation	Solves standard TSP instances accurately	✓
HCO.T5	**Permutation Sampling**	Verify uniform sampling	Permutations sampled uniformly without bias	✓
HCO.T6	**Permutation Sampling**	Verify distance-based sampling	Sampling prioritizes permutations based on the Hamming distance	✓
HCO.T7	**Population**	Verify that archipelago populations are correctly sampled using both uniform and distance-based methods	Populations are diverse and consistent with sampling methods	✓
HCO.T8	**Postprocessing**	Ensure accurate logging	Evolutionary data logged correctly	✓
HCO.T9	**Postprocessing**	Confirm history consistency	Sparse records match dense records	✓
HCO.T10	**Visualization**	Verify island-level evolution visuals	Island-level visualizations are accurate, complete and effectively communicate results	✓
HCO.T11	**Visualization**	Verify global evolution visuals	Global optimization visualizations are accurate, complete and effectively communicate results	✓
HCO.T12	**Visualization**	Verify visualization of initial guess impact on optimization performance	Visualizations correctly incorporate initial guesses and convey impact effectively	✓
HCO.T13	**Visualization**	Verify Gabbard diagrams	Diagrams correctly represent trajectory parameters, effectively convey all relevant information about each debris cloud	✓
HCO.T14	**STSP**	Verify correctness of generalized STSP problem class, and assess global optimization performance	Algorithms find optimal solutions efficiently	✓
HCO.T15	**STSP**	Verify correctness of generalized Integer STSP problem class, and assess global optimization performance	Integer methods do not progress, confirming encoding choice	✗
HCO.T16	**Optimizers**	Assess Simulated Annealing performance	Converges to high-quality solutions	✓
HCO.T17	**Optimizers**	Test local optimization capability	Succeed in locally improving solutions	✓
HCO.T18	**Optimizers**	Evaluate Basin Hopping effectiveness	Escapes local optima, finds superior solutions, overall less efficient than global optimizers	✓

Continued on next page

Table 3.3: Heuristic Combinatorial Optimization test summary table. (Continued)

Code	Component	Test rationale	Test criterion	Status
HCO.T19	**STSP**	Verify RAAN walk optimality	RAAN walk is near-optimal policy for static STSP as expected [2]	✓
HCO.T20	**STSP**	Verify initial guess effectiveness	Initial guesses improve optimization metrics	✓
HCO.T21	**LTTO Module**	Verify Q-Law ADR trajectory generation	Trajectories meet accuracy and efficiency criteria	✓
HCO.T22	**LTTO Module**	Verify RQ-Law ADR trajectory generation	Trajectories meet accuracy and efficiency criteria	✓
HCO.T23	**Trajectory Estimation**	Verify MHT transfer cost estimation	Estimations align with theoretical models	✓
HCO.T24	**Trajectory Estimation**	Verify MHT-IPC transfer cost estimation	Accurate cost and parameter assessments	✓
HCO.T25	**Trajectory Estimation**	Verify MHT-NIC transfer cost estimation	Estimations are accurate and consistent	✓
HCO.T26	**Trajectory Estimation**	Verify RQ-Law transfer cost estimation	Estimations match theoretical models	✓
HCO.T27	**Trajectory Estimation**	Verify JIT-compiled MHT-IPC transfer cost estimation	Optimized estimations show improved metrics	✓
HCO.T28	**Trajectory Estimation**	Verify JIT-compiled Q-Law transfer cost estimation	Optimized estimations meet performance metrics	✓
HCO.T29	**Trajectory Estimation**	Verify JIT-compiled dynamic MHT-IPC tour cost estimation	Cost estimations reflect mission requirements accurately	✓
HCO.T30	**Trajectory Estimation**	Verify JIT-compiled dynamic Q-Law tour cost estimation	Cost estimations are accurate and consistent	✓
HCO.T31	**Nearest Neighbor Search**	Test nearest neighbor heuristic	Produces high-quality initial solutions	✓
HCO.T32	**Nearest Neighbor Search**	Assess heuristic with MHT-IPC	Generates effective initial solutions	✓
HCO.T33	**Beam Search**	Verify beam search implementation	Explores multiple paths, improves solution quality	✓
HCO.T34	**Generalized STSP**	Test static global optimization	Identifies optimal solutions correctly	✓
HCO.T35	**Generalized STSP**	Verify dynamic optimization with initial guesses	Improved convergence and solution quality	✓
HCO.T36	**Generalized STSP**	Assess static MHT-IPC optimization	Finds optimal solutions aligning with requirements	✓
HCO.T37	**Generalized STSP**	Test dynamic MHT-IPC optimization with guesses	Enhanced performance metrics achieved	✓

Continued on next page

Table 3.3: Heuristic Combinatorial Optimization test summary table. (Continued)

Code	Component	Test rationale	Test criterion	Status
HCO.T38	**Generalized STSP**	Ensure variable payload handling	Payloads generated accurately and consistently	✓
HCO.T39	**Generalized STSP**	Verify cost estimation with varying payloads	Cost estimations accurately reflect payload variations	✓
HCO.T40	**Generalized STSP**	Test OSSIE STSP optimization performance	Solves complex scenarios meeting all criteria	✓
HCO.T41	**ADR CTP**	Verify RQ-Law coasting strategy	Optimizes fuel consumption and mission duration correctly	✓

3.1.5 Neural Combinatorial Optimization

The experimental campaign for the NCO module focused on 6 key areas: the default NCO loop using REINFORCE, logging, environment, Advantage Actor-Critic, Proximal Policy Optimization and hyperparemter optimization. Test-driven development was of critical importance for the development of the NCO module, enabling the quick buildup of a reliable NCO codebase thanks to progressive verification. The summary of all tests conducted to verify the HCO module is presented in Tab. 3.4.

Test Targets and Outcomes

———— Neural Combinatorial Optimization

Function: Develops and trains neural network policies using reinforcement learning algorithms to solve combinatorial optimization problems specific to spacecraft routing in ADR missions.

Tests:

- **NCO.T1 (Verify simplest NCO setup on the classical TSP):** Tested the NCO module on the classical Traveling Salesman Problem to ensure that the routing policy can be learned correctly.
- **NCO.T8 (Test default NCO setup for static ADR scenarios using the relative inclination cost function):** Evaluated the NCO setup on static ADR scenarios using the relative inclination cost function. Observed consistent validation loss reduction over training.
- **NCO.T9 (Test default NCO setup for static ADR scenarios using MHT maneuvers):** Assessed the NCO module using MHT maneuvers in static ADR scenarios. Consistent validation loss reduction was achieved.
- **NCO.T10 (Test default NCO setup for static ADR scenarios using NIC maneuvers):** Tested the module with NIC maneuvers. Training showed consistent validation loss reduction.
- **NCO.T11 (Test default NCO setup for static ADR scenarios using IPC maneuvers):** Evaluated the NCO setup using IPC maneuvers. Consistent reduction in validation loss was observed.
- **NCO.T12 (Test default NCO setup for static, fuel-optimal ADR scenarios using MHT-IPC maneuvers):** Tested combined MHT and IPC maneuvers in static, fuel-optimal ADR scenarios. Training resulted in consistent validation loss reduction.
- **NCO.T13 (Test default NCO setup for static, fuel-optimal ADR scenarios using IPC maneuvers):** Assessed the module using IPC maneuvers in fuel-optimal scenarios. Observed consistent validation loss reduction.
- **NCO.T14 (Test default NCO setup for static ADR scenarios using MHT-IPC maneuvers):** Evaluated the module with MHT-IPC maneuvers in static scenarios. Consistent validation loss reduction was achieved.
- **NCO.T15 (Test default NCO setup for static ADR scenarios using Q-Law maneuvers):** Tested the NCO setup using Q-Law maneuvers. Training showed consistent reduction in validation loss.

- **NCO.T16 (Test default NCO setup for dynamic ADR scenarios using IPC maneuvers):** Evaluated the module in dynamic ADR scenarios with IPC maneuvers. Consistent validation loss reduction was observed.
- **NCO.T17 (Test default NCO setup for dynamic, fuel-optimal ADR scenarios using MHT-NIC maneuvers):** Assessed the module using MHT-NIC maneuvers in dynamic, fuel-optimal scenarios. Training resulted in consistent validation loss reduction.
- **NCO.T18 (Test default NCO setup for dynamic ADR scenarios using Q-Law maneuvers):** Tested the NCO setup with Q-Law maneuvers in dynamic scenarios. Consistent reduction in validation loss was achieved.

Logging

Function: Manages the recording and storage of training logs and geometric data during the training process, facilitating monitoring and analysis of training progress.

Tests:

- **NCO.T2 (Verify logging functionality):** Verified that logs are correctly recorded during training sessions.
- **NCO.T7 (Verify logging functionality for generalized STSPs):** Ensured that geometric data is logged accurately for generalized Spacecraft Traveling Salesman Problems (STSPs).

Environment

Function: Generates realistic ADR mission scenarios for training, based on statistical models derived from real-world debris clouds, and implements cost functions for different transfer maneuvers.

Tests:

- **NCO.T3 (Verify generation of realistic ADR mission scenarios based on the Iridium 33 debris cloud):** Confirmed that scenarios are generated according to statistical models from the Iridium 33 debris cloud.
- **NCO.T4 (Verify vectorized cost functions and cumulative cost functions against HCO cost functions):** Verified that vectorized cost functions for impulsive and low-thrust transfers are correctly implemented and that static and dynamic cumulative cost functions align with those used in the HCO module.

Advantage Actor-Critic

Function: Implements the Advantage Actor-Critic reinforcement learning algorithm for training the neural routing policy, balancing exploration and exploitation during training.

Tests:

- **NCO.T5 (Test A2C algorithm to solve fuel-optimal ADR mission scenarios):** Evaluated the A2C algorithm on fuel-optimal ADR mission scenarios. The algorithm converged and performed as expected.

Proximal Policy Optimization

Function: Implements the Proximal Policy Optimization reinforcement learning algorithm for training the neural routing policy, aiming to improve training stability and performance.

Tests:

- **NCO.T6 (Test PPO algorithm to solve fuel-optimal ADR mission scenarios):** Tested the PPO algorithm on fuel-optimal ADR mission scenarios. The algorithm did not perform as well as expected. Potential causes for this could be the default settings of the PPO algorithm in `RL4CO`, or less likely a more fundamental inadequacy of the method for highly dynamic Vehicle Routing Problems (VRP).

Hyperparameter Optimization

Function: Applies various techniques to identify the main effects of hyperparameters in the training process and enhance model performance.

Tests:

- **NCO.T19 (Verify ANOVA experiment design with Taguchi L27 array on a mock system identification test case):** Verified that ANOVA results are correctly computed and that the main effects of all parameters in the test are accurately identified.

Conclusion

The experimental campaign for the NCO module successfully verified the functionality and performance of its key components. All tests passed, confirming that the module operates as intended. These verifications establish the reliability of the NCO module for the training of NCO policy for space vehicle routing problems, providing a solid foundation for its integration into mission planning systems.

Table 3.4: NCO test summary table.

Code	Component	Test rationale	Test criterion	Status
NCO.T1	**NCO**	Verify simplest NCO setup on the classical TSP	TSP routing policy learned correctly	✓
NCO.T2	**Logging**	Verify logging functionality	Logs are correctly recorded	✓
NCO.T3	**Environment**	Verify generation of realistic ADR mission scenarios based on the Iridium 33 debris cloud	Scenarios are generated as per statistical models	✓
NCO.T4	**Environment**	Verify vectorized cost functions for impulsive and low thrust transfers, and verify static and dynamic cumulative cost functions against HCO cost functions	Successfully verified cost calculations for JGCD scenarios	✓
NCO.T5	**A2C**	Test A2C algorithm to solve fuel-optimal ADR mission scenarios	A2C algorithm converges and performs as expected	✓
NCO.T6	**PPO**	Test PPO algorithm to solve fuel-optimal ADR mission scenarios	PPO algorithm converges, performs worse than expected	✓
NCO.T7	**Logging**	Verify logging functionality for generalized STSP's	Geometric data is logged accurately	✓
NCO.T8	**NCO**	Test default NCO setup for static ADR scenarios using the relative inclination cost function	Consistent validation loss reduction over training	✓
NCO.T9	**NCO**	Test default NCO setup for static ADR scenarios using MHT manoeuvres	Consistent validation loss reduction over training	✓
NCO.T10	**NCO**	Test default NCO setup for static ADR scenarios using NIC manoeuvres	Consistent validation loss reduction over training	✓
NCO.T11	**NCO**	Test default NCO setup for static ADR scenarios using IPC manoeuvres	Consistent validation loss reduction over training	✓
NCO.T12	**NCO**	Test default NCO setup for static, fuel-optimal ADR scenarios using MHT-IPC manoeuvres	Consistent validation loss reduction over training	✓

Continued on next page

Table 3.4: NCO test summary table. (Continued)

Code	Component	Test rationale	Test criterion	Status
NCO.T13	**NCO**	Test default NCO setup for static, fuel-optimal ADR scenarios using IPC manoeuvres	Consistent validation loss reduction over training	✓
NCO.T14	**NCO**	Test default NCO setup for static ADR scenarios using MHT-IPC manoeuvres	Consistent validation loss reduction over training	✓
NCO.T15	**NCO**	Test default NCO setup for static ADR scenarios using Q-Law manoeuvres	Consistent validation loss reduction over training	✓
NCO.T16	**NCO**	Test default NCO setup for dynamic ADR scenarios using IPC manoeuvres	Consistent validation loss reduction over training	✓
NCO.T17	**NCO**	Test default NCO setup for dynamic fuel-optimal ADR scenarios using MHT-NIC manoeuvres	Consistent validation loss reduction over training	✓
NCO.T18	**NCO**	Test default NCO setup for dynamic ADR scenarios using Q-Law manoeuvres	Consistent validation loss reduction over training	✓
NCO.T19	**Hyperparameter optimization**	Verify ANOVA experiment design with Taguchi L27 array on a mock system identification test case	ANOVA results are correctly computed, main effects of all parameters in the test are correctly identified	✓

3.2 Robust Experiment Design

Second only to verification and validation in importance, the design of robust and reproducible experiments and the correct and thorough analysis of experimental results are fundamental for the integrity and value of scientific research.

This section aims to discuss the experiment design and analysis approach used in this work. The results of the experimental campaigns that are discussed in these sections are the results that have been presented so far in this work, and those that are yet to be discussed. The purpose of this section is not to give a detailed overview of the results obtained, as that is the scope of the thesis itself. Instead, this section aims to provide the reader with a fine-grained reference to the design of all experiments carried out in this work.

A set of fundamental general experiment design maxims applied throughout this work are presented in Section 3.2.1. This is followed a detailed report of the experimental campaigns conducted with each of the four main modules developed in this research: Impulsive Trajectory Design in Section 3.2.2, Low Thrust Trajectory Design in Section 3.2.3, Heuristic Combinatorial Optimization in Section 3.2.4, and lastly Neural Combinatorial Optimization in Section 3.2.5.

3.2.1 General Considerations

Random Number Generation

- All random number generation processes are controlled by set seeds to ensure reproducibility.

Experimental Robustness

- All data gathering experiments are repeated using different random seeds between 5 and 100 times, depending on the cost of the experiment, to ensure the robustness of gathered data.

Storage and Post-processing of Results

- All experimental results are stored using human-readable filenames encoding experiment information.
- All NCO training runs are thoroughly logged using either Weights and Biases[1] (WandB) or TensorBoard[2].

Consistent Experimental Environment

- All experiments are conducted in a controlled environment with consistent hardware and software configurations to eliminate variability due to external factors [108].

Version Control

- All code and experiment configurations are version-controlled using Git[3] [109] to ensure reproducibility and traceability.

Data Validation

- Experimental data is validated before use to prevent errors due to corrupted or invalid data [110].

Statistical Analysis

- Appropriate statistical methods are used to analyze results, and statistical tests are applied to determine significance [77].

Documentation

- All experiment settings, parameters, and procedures are thoroughly documented to facilitate reproducibility [111].

Automation

- Experiments are automated where possible to reduce human error and increase efficiency [112].

Control Experiments

- Control experiments are conducted to establish baseline performance and validate the experimental setup [113].

Data Integrity and Backup

- Experimental data is backed up across the servers and machines used in this research. Critical experimental data is version-controlled [114].

Sample Size and Statistical Power

- Adequate sample sizes are used to ensure statistical power and reliability of results [77]. This is particularly important for the the ANOVA analyses conducted to assess hyperparameter impact on NCO model performance. Refer to Section 3.3 for an in-depth discussion of this topic.

3.2.2 Impulsive Trajectory Design

Analysis Summary and Key Outcomes

The experimental test campaign for the Impulsive Trajectory Design (IPD) module aimed to generate realistic Multiple Hohmann Transfer (IPO.A1), Node Inclination Change (IPO.A2), and sequential MHT-NIC (IPO.A3)

[1] `https://wandb.ai/site`

[2] `https://www.tensorflow.org/tensorboard`

[3] `https://git-scm.com/`

trajectories for the OSSIE OTV under J_2 perturbations. These trajectories were then used as warm starts for the trajectory re-optimization process using SCP. Tab. 3.5 summarizes the experiments conducted. In each case, the perturbed trajectories were successfully used as warm starts for the SCP trajectory re-optimization process. These results confirm the IPO module's ability to generate accurate perturbed trajectories, supporting its integration into mission planning workflows.

Table 3.5: Impulsive trajectory design experimental campaign summary table.

Code	Area	Analysis Target	Outcome
IPO.A1	**MHT**	Produce realistic MHT trajectories for the OSSIE OTV under $J2$	Perturbed MHT trajectories succesfully used as warm starts for the SCP trajectory re-optimization process
IPO.A2	**NIC**	Produce realistic NIC trajectories for the OSSIE OTV under $J2$	Perturbed NIC trajectories succesfully used as warm starts for the SCP trajectory re-optimization process
IPO.A3	**MHT-NIC**	Produce realistic sequential MHT-NIC trajectories for the OSSIE OTV under $J2$	Perturbed MHT-NIC trajectories succesfully used as warm starts for the SCP trajectory re-optimization process

3.2.3 Low Thrust Trajectory Design

Analysis Summary and Key Outcomes

The experimental campaign for the Low Thrust Trajectory Optimization (LTTO) module encompassed ten experiments across four primary research areas. The summary of these experiments is presented in Tab. 3.6.

———— Benchmark Selection

Experiment **LTTO.A1** focused on determining the appropriate integrator step size for benchmarking purposes. The outcome of this experiment established an adequate step size that ensures reliable integration results for subsequent analyses.

———— Integrator Selection

Experiments **LTTO.A2** and **LTTO.A3** addressed the selection of optimal integrators for the Q-Law guidance policy within the Tudat framework. **LTTO.A2** involved trial testing of the Q-Law guidance policy with various integrators, resulting in successful trials across different integrator configurations. **LTTO.A3** conducted a trade-off analysis of all available integrators in Tudat to identify the most performant integrator-setting combinations for generating ADR Q-Law trajectories. This analysis led to the identification and selection of the optimal integrator settings.

———— OSSIE Mission Design

A series of experiments (**LTTO.A4** to **LTTO.A8**) were conducted to design missions for the OSSIE scenario using both Q-Law and RQ-Law guidance policies integrated with Tudat. **LTTO.A4**, **LTTO.A5**, **LTTO.A6**, and **LTTO.A7** successfully designed missions using RQ-Law and Q-Law, both integrated and standalone within Tudat. Additionally, **LTTO.A8** focused on decoupled mission design using Q-Law with Tudat, resulting in the successful implementation of decoupled Q-Law mission designs.

———— ADR Mission Design

Experiments **LTTO.A9** and **LTTO.A10** concentrated on mission design for the ADR scenario using the RQ-Law guidance policy within Tudat. Both experiments successfully designed missions employing RQ-Law, demonstrating the capability of the LTTO module to support ADR mission planning effectively.

Table 3.6: Low thrust trajectory design experimental campaign summary table.

Code	Area	Analysis Target	Outcome
LTTO.A1	**Benchmark selection**	Determining the benchmark integrator step size	Determined adequate benchmark integration step size
LTTO.A2	**Integrator selection**	Trial testing of Q-Law guidance policy with different integrators	Successfully trialed Q-Law guidance policy with various integrators
LTTO.A3	**Integrator selection**	Trade-off all available integrators in Tudat to find the optimal integrator for generating ADR Q-Law trajectories	Identified list of highly performant integrator-setting combinations, selected optimal integrator
LTTO.A4	**OSSIE mission design**	Mission design using RQ-Law integrated with Tudat for OSSIE scenarios	Successfully designed missions using RQ-Law with Tudat
LTTO.A5	**OSSIE mission design**	Mission design using RQ-Law for OSSIE scenarios	Successfully designed missions using RQ-Law
LTTO.A6	**OSSIE mission design**	Mission design using Q-Law integrated with Tudat for OSSIE scenarios	Successfully designed missions using Q-Law with Tudat
LTTO.A7	**OSSIE mission design**	Mission design using Q-Law for OSSIE scenarios	Successfully designed missions using Q-Law
LTTO.A8	**OSSIE mission design**	Decoupled mission design using Q-Law with Tudat for OSSIE scenarios	Successfully implemented decoupled Q-Law mission designs with Tudat
LTTO.A9	**ADR mission design**	Mission design using RQ-Law with Tudat for ADR scenarios	Successfully designed missions using RQ-Law with Tudat for ADR
LTTO.A10	**ADR mission design**	Mission design using RQ-Law for ADR scenarios	Successfully designed missions using RQ-Law for ADR

3.2.4 Heuristic Combinatorial Optimization

This section summarizes the experimental campaign conducted on the Heuristic Combinatorial Optimization (HCO) module, focusing on various strategies and methodologies applied to space mission planning and optimization problems. The primary objectives were to evaluate different sampling methods, optimization strategies, trajectory estimations, and their impacts on mission design, particularly in the context of space debris modeling and active debris removal missions.

Analysis Summary and Key Outcomes

The experimental campaign explored several key areas:

———— Permutation Sampling

The impact of different permutation sampling methods on optimization performance was assessed. Uniform sampling served as a baseline, while distance-based sampling demonstrated improved solution diversity, leading to enhanced optimization results.

———— Space Debris Modeling and Study

Statistical analyses were conducted to understand the characteristics and evolution of space debris clouds. Models predicting future debris population trends were developed, and Gabbard diagrams were created to visualize orbital parameters, providing valuable insights for long-term space mission planning.

Approximate Solutions Visualization

Visualization techniques were employed to illustrate the progression of the Right Ascension of the Ascending Node (RAAN) in dynamic scenarios. These visualizations aided in understanding RAAN adjustments using Q-Law and impulsive maneuvers, contributing to better mission planning strategies.

Optimizer Trade-Off Studies

Various optimization strategies were evaluated, including local and global optimization methods, meta-heuristic algorithms, and the use of initial guesses. The studies revealed that initial guesses significantly enhance optimization efficiency and that combining strategies can lead to improved results.

Trajectory Estimation

Regression models for Time of Flight (TOF) and the Q parameter were developed using RQ-Law and Q-Law with Tudat software. These models facilitate accurate trajectory parameter predictions, essential for effective mission planning and fuel estimation.

Perturbation Impact Studies

The effects of dynamic trajectories, RAAN walks, beam search algorithms, and impulsive maneuvers on mission planning were analyzed. Findings indicated that dynamic strategies and initial guesses significantly improve optimization performance, and beam search algorithms enhance optimization in dynamic scenarios.

Optimization Performance

The influence of initial guesses and different sampling methods on global optimization was examined. High-quality initial guesses were found to substantially improve optimization results, and the choice of sampler was identified as a critical factor affecting optimization effectiveness.

OSSIE and ADR Mission Design

For OSSIE (Optimization with Spacecraft and Space-based Infrastructure Evolution) and ADR missions, dynamic global optimization strategies were evaluated. The use of Multiple Hohmann Transfers without Inclination Change (MHT-NIC), impulsive maneuvers, and RQ-Law trajectories demonstrated benefits in mission design. Initial guesses further enhanced optimization performance, and fuel consumption estimates were accurately provided.

Reinforcement Learning

Optimality gaps in reinforcement learning (RL) training datasets were analyzed for both OSSIE and ADR missions. Quantifying these gaps provided insights for refining RL models and improving their accuracy in mission planning applications.

Coasting Transfer Policies

The impact of coasting transfer policies on mission efficiency was evaluated. Implementing coasting strategies was found to offer benefits in fuel savings and mission duration, contributing to more efficient ADR missions.

Table 3.7: Heuristic Combinatorial Optimization experimental campaign summary table.

Code	Area	Analysis Target	Outcome
HCO.A1	**Permutation sampling**	Evaluating uniform sampling in optimization	Assessed impact of uniform sampling on solution quality
HCO.A2	**Permutation sampling**	Analyzing the impact of distance-based sampling on optimization	Demonstrated advantages of distance-based sampling in solution diversity

Continued on next page

Table 3.7: Heuristic Combinatorial Optimization experimental campaign summary table. (Continued)

Code	Area	Analysis Target	Outcome
HCO.A3	**Space debris modelling**	Providing a statistical summary of debris cloud characteristics	Presented key statistics of debris cloud for analysis
HCO.A4	**Space debris study**	Summarizing statistics of debris cloud with static RAAN walk	Generated statistical data on debris cloud behavior with static RAAN
HCO.A5	**Space debris study**	Analyzing the evolution of space debris over time	Modeled and predicted debris population changes
HCO.A6	**Space debris study**	Creating Gabbard diagrams for debris analysis	Produced diagrams illustrating debris orbital characteristics
HCO.A7	**Approximate solutions**	Visualizing dynamic RAAN walk using Q-Law	Created visual representations of RAAN progression
HCO.A8	**Approximate solutions**	Visualizing dynamic RAAN walk with impulsive maneuvers	Provided visual insights into impulsive RAAN changes
HCO.A9	**Optimizer trade-off**	Rapid evaluation of global optimization strategies	Quickly identified effective global optimizers
HCO.A10	**Optimizer trade-off**	Rapid evaluation of local optimization strategies	Provided quick insights into local optimization performance
HCO.A11	**Optimizer trade-off**	Assessing trade-offs in local optimization methods	Identified strengths and weaknesses of local optimization approaches
HCO.A12	**Optimizer trade-off**	Evaluating local optimization with initial guesses	Showed initial guesses improve local optimization efficiency
HCO.A13	**Optimizer trade-off**	Testing local optimization on constrained problems	Demonstrated local optimization effectiveness under constraints
HCO.A14	**Optimizer trade-off**	Analyzing combined local optimization strategies	Found that combining strategies enhances optimization results
HCO.A15	**Optimizer trade-off**	Evaluating global optimization techniques	Assessed the performance of various global optimizers
HCO.A16	**Optimizer trade-off**	Impact of initial guesses on global optimization	Confirmed initial guesses improve global optimization efficiency
HCO.A17	**Optimizer trade-off**	Analyzing combined global optimization strategies	Improved results using combined optimization approaches
HCO.A18	**Optimizer trade-off**	Evaluating different meta-heuristic algorithms	Compared performance of various meta-heuristics
HCO.A19	**Trajectory estimation**	Time of Flight and Q parameter regression analysis using RQ-Law with Tudat for ADR	Derived regression models for TOF-Q relationships
HCO.A20	**Trajectory estimation**	Time of Flight and Q parameter regression analysis using Q-Law with Tudat for ADR	Developed regression models for Q-Law trajectories
HCO.A21	**Perturbation impact**	Assessing the impact of using dynamic RQ-Law trajectories	Demonstrated benefits in trajectory optimization

Continued on next page

Table 3.7: Heuristic Combinatorial Optimization experimental campaign summary table. (Continued)

Code	Area	Analysis Target	Outcome
HCO.A22	**Perturbation impact**	Evaluating the impact of dynamic RAAN walk using Q-Law	Showed improved mission planning with dynamic RAAN walk
HCO.A23	**Perturbation impact**	Assessing impact of dynamic RAAN walk with impulsive maneuvers	Analyzed effectiveness of impulsive RAAN adjustments
HCO.A24	**Perturbation impact**	Evaluating beam search in dynamic RAAN walk using RQ-Law	Found beam search enhances RAAN optimization
HCO.A25	**Perturbation impact**	Beam search in dynamic RAAN walk using MHT with Impulsive Plane Changes	Improved optimization with beam search and MHT-IPC
HCO.A26	**Perturbation impact**	Assessing the impact of dynamic Q-Law trajectories	Confirmed advantages in mission planning
HCO.A27	**Perturbation impact**	Evaluating the impact of dynamic impulsive maneuvers	Demonstrated effectiveness in trajectory optimization
HCO.A28	**Perturbation impact**	Analyzing fuel mass consumption in dynamic impulsive maneuvers	Provided insights into fuel efficiency
HCO.A29	**Perturbation impact**	Impact of initial guesses in dynamic global optimization with Q-Law	Improved optimization results with initial guesses
HCO.A30	**Optimization performance**	Assessing the impact of initial guesses on global optimization	Found that good initial guesses significantly improve optimization results
HCO.A31	**Optimization performance**	Assessing initial guesses in dynamic global optimization with impulsive maneuvers	Confirmed initial guesses enhance optimization in dynamic impulsive scenarios
HCO.A32	**Optimization performance**	Assessing the impact of different samplers in global optimization	Determined the influence of sampler choice on optimization
HCO.A33	**Optimizer trade-off**	Evaluating dynamic global optimization with initial guesses using impulsive maneuvers	Identified performance benefits of using initial guesses in dynamic optimization
HCO.A34	**Optimizer trade-off**	Evaluating dynamic global optimization using impulsive maneuvers	Assessed the effectiveness of impulsive maneuvers in dynamic global optimization
HCO.A35	**Optimizer trade-off**	Evaluating combined strategies in dynamic global optimization with impulsive maneuvers	Showed improved optimization results with combined strategies
HCO.A36	**Optimizer trade-off**	Evaluating dynamic meta-heuristics with impulsive maneuvers	Showed effectiveness of meta-heuristics in dynamic impulsive scenarios
HCO.A37	**Optimizer trade-off**	Evaluating dynamic meta-heuristics using Q-Law	Showed Q-Law's effectiveness in dynamic meta-heuristics
HCO.A38	**OSSIE mission design**	Analyzing OSSIE dynamic global optimization with MHT without Inclination Change	Demonstrated benefits in OSSIE scenarios

Continued on next page

Table 3.7: Heuristic Combinatorial Optimization experimental campaign summary table. (Continued)

Code	Area	Analysis Target	Outcome
HCO.A39	**OSSIE mission design**	Impact of initial guesses in OSSIE dynamic global optimization using MHT-NIC	Confirmed initial guesses improve optimization
HCO.A40	**OSSIE mission design**	Combined strategies in OSSIE dynamic global optimization with MHT-NIC	Showed improved results with combined approaches
HCO.A41	**ADR mission design**	Evaluating combined strategies in dynamic global optimization using RQ-Law for ADR	Improved optimization outcomes with RQ-Law
HCO.A42	**ADR mission design**	Evaluating dynamic global optimization using RQ-Law for ADR	Assessed RQ-Law effectiveness in optimization
HCO.A43	**ADR mission design**	Impact of initial guesses in dynamic global optimization with RQ-Law for ADR	Improved optimization with initial guesses
HCO.A44	**OSSIE mission design**	Verifying mass changes during OSSIE tours	Confirmed correctness of mass change calculations in OSSIE tours
HCO.A45	**OSSIE mission design**	Generating mission scenarios for OSSIE	Successfully generated mission scenarios for analysis
HCO.A46	**OSSIE mission design**	Analyzing optimality gap in RL training datasets for OSSIE	Quantified optimality gaps to improve reinforcement learning training
HCO.A47	**OSSIE mission design**	Nominal OSSIE dynamic global optimization using impulsive maneuvers without Inclination Change	Evaluated nominal performance in OSSIE missions
HCO.A48	**OSSIE mission design**	OSSIE dynamic global optimization with Impulsive Plane Changes	Analyzed performance with IPC maneuvers
HCO.A49	**OSSIE mission design**	Estimating fuel mass consumption in OSSIE missions	Provided accurate fuel consumption estimates
HCO.A50	**OSSIE mission design**	Analyzing optimal tours in OSSIE using dynamic global optimization	Identified optimal tour strategies for OSSIE missions
HCO.A51	**OSSIE mission design**	Performing Monte Carlo analysis on OSSIE scenarios	Generated statistical insights from Monte Carlo simulations
HCO.A52	**ADR mission design**	Generating trajectories using Q-Law with Tudat for ADR	Successfully generated RQ-Law ADR trajectories through the Iridium33 cloud
HCO.A53	**RL**	Analyzing optimality gap in RL training datasets for ADR	Quantified gaps to enhance reinforcement learning training
HCO.A54	**ADR CTP**	Evaluating the impact of coasting transfer policies	Identified benefits of coasting strategies

3.2.5 Neural Combinatorial Optimization

This section describes the experimental test campaign conducted to study the performance of NCO approaches. The primary objective of this campaign was to evaluate the performance and reliability of various NCO algorithms and their configurations in optimizing mission scenarios for the OSSIE and ADR STSP mission scenarios. The experiments aimed to assess the effectiveness of different RL strategies in training policies that achieve fuel-optimal trajectories, as well as to determine the best-performing algorithms through comprehensive measurements of algorithm performance, the impact of various hyperparameters on model performance, and hyperparameter optimization.

Analysis Summary and Key Outcomes

———— Performance Measurement

The performance measurement area focused on exporting validation datasets for OSSIE and ADR mission scenarios (NCO.A1, NCO.A2). The objective was to ensure that the datasets accurately represent the mission parameters required for subsequent policy training and evaluation. The outcomes confirmed the successful export of datasets for OSSIE scenarios (NCO.A1) and JGCD scenarios (NCO.A2), providing a robust foundation for validating the NCO module's performance across different mission types.

———— OSSIE Mission Design

In the OSSIE mission design area, experiments evaluated various RL algorithms, including REINFORCE, A2C, and PPO, for training fuel-optimal policies in the OSSIE STSP (NCO.A3, NCO.A4, NCO.A5, NCO.A6). The rationale was to determine the most effective RL algorithm based on policy performance and training efficiency. The results demonstrated that REINFORCE, A2C, and PPO successfully trained NCO policies for the OSSIE STSP (NCO.A3, NCO.A4, NCO.A5). Furthermore, a trade-off analysis (NCO.A6) measured the performance of all RL algorithms, identifying the best-performing algorithm by evaluating policy performance and training time.

———— Low Thrust OSSIE Mission Design

The low thrust OSSIE mission design area extended the evaluation of RL algorithms by incorporating the Q-Law guidance policy (NCO.A7, NCO.A8, NCO.A9). These experiments aimed to assess the effectiveness of REINFORCE, A2C, and PPO in training NCO policies specifically tailored for Q-Law guided OSSIE STSP scenarios. The outcomes confirmed that all three RL algorithms successfully trained NCO policies for the Q-Law OSSIE STSP (NCO.A7, NCO.A8, NCO.A9), demonstrating the versatility and robustness of the NCO module in handling different guidance policies.

———— ADR Mission Design

In the ADR mission design area, the focus was on evaluating RL algorithms for the Iridium 33 ADR scenario (NCO.A10, NCO.A11, NCO.A12, NCO.A13, NCO.A14). The experiments aimed to determine the efficacy of REINFORCE, A2C, and PPO in training fuel-optimal policies and to identify the most suitable algorithm through performance trade-offs. REINFORCE and A2C successfully trained NCO policies for the Iridium 33 ADR STSP, with results logged using WandB and TensorBoard (NCO.A10, NCO.A11, NCO.A12). However, PPO (NCO.A13) did not outperform A2C and REINFORCE, indicating a need for further refinement. A comprehensive trade-off analysis (NCO.A14) evaluated the performance of all RL algorithms, determining that A2C offered the best balance between policy performance and training efficiency for the ADR scenarios.

———— Hyperparameter Optimization

The hyperparameter optimization area addressed the fine-tuning of the A2C algorithm to enhance its performance in the Iridium 33 ADR STSP scenario (NCO.A15, NCO.A16). Conducting ANOVA statistical analysis (NCO.A15) successfully identified the most significant parameters influencing A2C performance. Additionally, grid search hyperparameter optimization (NCO.A16) was performed, resulting in the determination of optimal hyperparameter settings for A2C in JGCD scenarios. These optimizations contributed to improved training outcomes and policy effectiveness.

Table 3.8: NCO experimental campaign summary table.

Code	Area	Analysis Target	Outcome
NCO.A1	**Performance measurement**	Export validation datasets of OSSIE mission scenarios	Successfully exported datasets for OSSIE scenarios
NCO.A2	**Performance measurement**	Export validation datasets of ADR mission scenarios	Successfully exported datasets for JGCD scenarios
NCO.A3	**OSSIE mission design**	Evaluate REINFORCE for the fuel-optimal OSSIE STSP	REINFORCE successfully trains NCO policy for OSSIE STSP
NCO.A4	**OSSIE mission design**	Evaluate A2C for the fuel-optimal OSSIE STSP	A2C successfully trains NCO policy for OSSIE STSP
NCO.A5	**OSSIE mission design**	Evaluate PPO for the fuel-optimal OSSIE STSP	PPO successfully trains NCO policy for OSSIE STSP
NCO.A6	**OSSIE mission design**	Trade-off REINFORCE, A2C and PPO for the fuel-optimal OSSIE STSP	Measured performance of all RL algorithms on OSSIE STSP scenario and determined best RL algorithm based on policy performance and training time
NCO.A7	**Low Thrust OSSIE mission design**	Evaluate REINFORCE for the fuel-optimal OSSIE STSP using the Q-Law	REINFORCE successfully trains NCO policy for the Q-Law OSSIE STSP
NCO.A8	**Low Thrust OSSIE mission design**	Evaluate A2C for the fuel-optimal OSSIE STSP using the Q-Law	A2C successfully trains NCO policy for the Q-Law OSSIE STSP
NCO.A9	**Low Thrust OSSIE mission design**	Evaluate PPO for the fuel-optimal OSSIE STSP using the Q-Law	PPO successfully trains NCO policy for the Q-Law OSSIE STSP
NCO.A10	**ADR mission design**	Evaluate REINFORCE for the Iridium 33 ADR scenario; results logged with WandB	REINFORCE successfully trains NCO policy for the Iridium 33 ADR STSP
NCO.A11	**ADR mission design**	Evaluate A2C for the Iridium 33 ADR scenario; results logged with WandB	A2C successfully trains NCO policy for the Iridium 33 ADR STSP
NCO.A12	**ADR mission design**	Evaluate A2C for the Iridium 33 ADR scenario; results logged with TensorBoard	A2C successfully trains NCO policy for the Iridium 33 ADR STSP. Locally logged results useful for fast-paced experimentation with smaller training datasets, batch sizes
NCO.A13	**ADR mission design**	Evaluate PPO for the Iridium 33 ADR scenario; results logged with WandB	✗ PPO successfully trains NCO policy for the Iridium 33 ADR STSP, yet considerably underperforms A2C and does not outperform REINFORCE in a determinant way

Continued on next page

Table 3.8: NCO experimental campaign summary table. (Continued)

Code	Area	Analysis Target	Outcome
NCO.A14	**ADR mission design**	Trade-off REINFORCE, A2C and PPO for the Iridium 33 ADR STSP	Measured performance of all RL algorithms on Iridium 33 ADR STSP scenario and determined best RL algorithm based on policy performance and training time
NCO.A15	**Hyperparameter optimization**	Conduct ANOVA statistical analysis on A2C performance in the Iridium 33 ADR STSP scenario	Successfully identified the most significant parameters for A2C performance in the Iridium 33 ADR STSP scenario
NCO.A16	**Hyperparameter optimization**	Perform grid search hyperparameter optimization for A2C in the Iridium 33 ADR STSP scenario	Successfully performed grid search hyperparameter optimization for A2C in JGCD scenarios

3.3 Sensitivity Analysis: Hyperparameter Impact Determination with ANOVA

This section deals with sensitivity analysis. Sensitivity analysis was critical to study the impact of the various hyperparameters on the performance of the NCO policy used in this work, to choose a reduced set of hyperparameters to optimize, and to study the behavior of policy performance as a function of these key parameters. Section 3.3.1 discusses the analysis performed to efficiently determine the most relevant hyperparameters for policy performance. Section 3.3.2 discusses the full factorial ANOVA conducted to find an optimal set of key hyperparameters, and study the behavior of policy performance as a function of model complexity.

3.3.1 Fractional Factorial ANOVA with the Taguchi L27 Design

Optimizing the performance of NCO models involves tuning a multitude of hyperparameters. Determining the statistical relevance of these hyperparameters is essential to identify which ones significantly influence model performance and to prioritize them for further optimization [115]. ANOVA serves as a robust statistical procedure to assess the significance of multiple factors simultaneously [116].

Orthogonal arrays, particularly Taguchi factorial designs, facilitate efficient experimentation by systematically varying hyperparameters across predefined levels while minimizing the number of required experimental runs [117]–[119]. The Taguchi L27 orthogonal array, also known as L_{27}-$A3^{13\text{-}10}$ fractional factorial design [79], is specifically designed to evaluate up to 13 factors at three levels, making it suitable for comprehensive hyperparameter analysis with limited resources [79]. Table 3.9 presents the Taguchi L27 orthogonal array utilized in this study.

A linear ANOVA was conducted using the `statsmodels` library [120] to evaluate the main effects of 13 hyperparameters on the model's performance. The 13 chosen hyperparameters, their function, and the rationale for choosing them can be seen in Tab. 3.10. The choice of a linear model is justified by the initial focus on identifying straightforward, additive relationships between hyperparameters and performance metrics, with the goal of identifying the most impactful hyperparameters for further optimization.

Tab. 3.11 presents the ANOVA results for the linear effects of each hyperparameter. The Sum of Squares (Sum Sq), degrees of freedom (df), F-statistic (F), and p-values (PR(>F)) are reported for each factor. ANOVA relies on two fundamental assumptions: normality of residuals, and homogeneity of variances across all factor levels —the property known as homoscedasticity [79], [116]. Residual normality was verified by the Shapiro-Wilk test [79], [121] ($p = 0.57$) and the Anderson-Darling test [79] ($p = 0.27$). Homoscedasticity was verified by visual inspection of the residuals. Furthermore as will be seen, the impacts estimated by the ANOVA match those expected from domain expertise. The ANOVA is thus considered valid to diagnose the main effects of the 13

Table 3.9: Taguchi L27 orthogonal array, also known as L_{27}-$A3^{13\text{-}10}$ fractional factorial design [79].

Run	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**	**17**	**18**	**19**	**20**	**21**	**22**	**23**	**24**	**25**	**26**	**27**
X1	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	2	2	2	2	2	2	2	2	2
X2	0	0	0	1	1	1	2	2	2	0	0	0	1	1	1	2	2	2	0	0	0	1	1	1	2	2	1
X3	0	0	0	1	1	1	2	2	2	1	1	1	0	0	0	1	1	1	2	2	2	0	0	0	1	1	2
X4	0	0	0	1	1	1	2	2	2	2	2	2	0	0	0	1	1	1	0	0	0	2	2	2	1	1	1
X5	0	1	2	0	1	2	0	1	2	0	1	2	0	1	2	0	1	2	0	1	2	0	1	2	0	1	0
X6	0	1	2	0	1	2	0	1	2	1	2	0	2	0	1	1	0	0	2	0	1	2	0	1	0	1	0
X7	0	1	2	0	1	2	0	1	2	2	0	1	1	0	0	2	1	1	1	2	0	2	1	0	2	0	1
X8	0	1	2	1	2	0	2	1	0	0	1	2	0	1	2	1	0	2	2	1	2	1	2	1	2	1	2
X9	0	1	2	1	2	0	2	1	0	1	2	0	2	1	0	2	1	0	0	1	2	1	2	1	1	2	2
X10	0	1	2	1	2	0	1	2	0	3	0	1	1	2	0	2	2	1	1	2	0	2	1	0	1	0	1
X11	0	1	2	2	0	1	1	2	0	1	2	0	2	0	1	1	0	1	1	0	2	2	1	3	1	2	1
X12	0	1	2	2	0	1	2	3	0	2	1	2	1	3	0	0	1	2	3	2	1	1	3	1	2	1	2
X13	0	1	2	2	0	1	2	3	0	3	2	1	0	1	2	3	2	1	1	3	2	1	2	1	3	2	1

hyperparameters considered on policy performance.

The ANOVA results indicate that embedding dimension and number of encoder layers are the only hyperparameters with statistically significant effects on model performance, with p-values of 0,005776 and 0,048680, respectively. embedding dimension exhibits the highest Sum of Squares (3,46E+09), suggesting it has the most substantial impact. number of encoder layers follows with a Sum of Squares of 1,50E+09, indicating a meaningful but slightly lesser influence. All other hyperparameters do not show significant linear effects, as their p-values exceed the conventional threshold of $0,05$. The linear model accounts for approximately 69,87% of the total observed variance ($R^2 = 69,87$%). A considerable portion of variability (30.13%) is not explained by the linear effects of the analyzed hyperparameters, indicating the presence of higher order effects which are not modelled.

The significant effect of embedding dimension suggests that increasing this hyperparameter enhances the model's capacity to capture and represent input features effectively, thereby improving performance. The significance of the number of encoder layers implies that adding more layers may contribute to deeper feature extraction and more complex graph representations, although its impact is less pronounced compared to embedding dimension. The non-significant effects of other hyperparameters indicate that, within the tested ranges, their individual linear contributions to model performance are minimal. However, it is possible that these hyperparameters may interact with each other or exhibit non-linear relationships that are not captured in the current linear ANOVA model. The R^2 value of 69,87% indicates a moderately strong fit. While the linear model explains a substantial portion of the variance, there remains considerable unexplained variability, potentially due to unmodeled factors or complex relationships among hyperparameters.

The ANOVA analysis identifies embedding dimension as the most statistically significant hyperparameter influencing model performance, followed by a marginal effect from the number of encoder layers. These findings suggest that increasing the embedding dimension is desirable to enhance feature representation capabilities, and that optimizing the number of encoder layers can potentially improve the depth and quality of feature extraction, albeit with a less substantial impact. Embedding dimension and number of encoder layers were chosen for hyperparameter optimization, discussed next.

Table 3.10: Policy architecture and training hyperparameters considered in the 3-level ANOVA.

Hyperparameter	Component	Function	Rationale	ANOVA Levels
Embedding Dimension	Embedding Layer	Size of node embeddings representing input features.	Influences the capacity to capture feature representations.	64, 128, 256
Number of Encoder Layers	GAT	Number of layers in the GAT, affecting depth and representation learning.	Determines the depth of feature extraction and complexity of graph representations.	2, 3, 4
Number of Attention Heads	GAT	Number of parallel attention mechanisms per GAT layer.	Enhances the model's ability to focus on different parts of the graph simultaneously.	4, 8, 16
Feedforward Hidden Size	GAT	Size of the hidden layer in GAT's feedforward network.	Affects the model's capacity and computational complexity.	256, 512, 1024
Dropout Rate	GAT	Probability of dropping units during training to prevent overfitting.	Helps in regularizing the model and improving generalization.	0.1, 0.3, 0.5
Temperature	PN	Scales logits before softmax to control randomness in action selection.	Balances exploration and exploitation during policy generation.	0.5, 1.0, 2.0
Tanh Clipping	PN	Limits the output of the tanh activation to prevent extreme values.	Ensures numerical stability by preventing large activation values.	0, 10, 20
Actor Learning Rate	Actor Optimizer	Learning rate for the actor (policy) network optimizer (e.g., Adam).	Influences the speed and stability of policy updates.	1e-5, 1e-4, 1e-3
Weight Decay	Actor Optimizer	Regularization parameter to prevent overfitting by penalizing large weights.	Controls the model's generalization and prevents overfitting.	0, 1e-4, 1e-3
Gradient Clipping Value	Actor Optimizer	Maximum allowed value for gradients during backpropagation to prevent exploding gradients.	Ensures training stability by avoiding excessively large gradients.	0.5, 1.0, 2.0
Critic Learning Rate	Critic Optimizer	Learning rate for the critic network optimizer (e.g., Adam).	Affects the stability and speed of value estimation updates.	1e-5, 1e-4, 1e-3
Reward Scaling	REINFORCE Baseline	Scales the reward signal to stabilize training and improve gradient estimates.	Enhances training stability by normalizing reward magnitudes.	1, 10, 100
Critic Hidden Dimension	Critic Network	Size of the hidden layers within the critic network.	Influences the critic's capacity to accurately estimate value functions.	128, 256, 512

Table 3.11: ANOVA results. Dashed line separates statistically significant factors ($p < 0.05$) from non-significant factors.

Hyperparameter	Sum Sq	df	F	PR(>F)	R^2
Embedding Dimension	3,46E+09	1	10,9	**0,005776**	25,20%
Number of Encoder Layers	1,50E+09	1	4,73	**0,04868**	10,96%
Feedforward Hidden Size	7,82E+08	1	2,46	0,140893	5,70%
Weight Decay	7,28E+08	1	2,29	0,154256	5,30%
Number of Attention Heads	7,03E+08	1	2,21	0,160724	5,13%
Actor Learning Rate	6,66E+08	1	2,1	0,171374	4,86%
Critic Hidden Dimension	6,18E+08	1	1,94	0,18656	4,51%
Gradient Clipping Value	5,55E+08	1	1,75	0,209273	4,04%
Dropout Rate	3,24E+08	1	1,02	0,331401	2,36%
Tanh Clipping	2,11E+08	1	0,66	0,429799	1,54%
Critic Learning Rate	3,58E+07	1	0,11	0,742551	0,26%
Temperature	1,58E+06	1	0,00	0,944904	0,01%
Reward Scaling	4,52E+05	1	0,00	0,970506	0,00%
ANOVA model	9,58E+09	13	-	-	**69,87%**
Residuals	4,13E+09	13	-	-	**30,13%**

3.3.2 Full Factorial ANOVA

A full factorial (grid search) approach was conducted to optimize the embedding dimension and number of encoder layers, considering four levels. The levels considered for each hyperparameter are those previously presented in Tab. 3.10. Fig. 3.4 shows the optimality gaps obtained by each of the models in the grid search, using BS to decode the policies. No statistically significant improvement is observed in comparison with the default parameters. Fig. 3.3 shows the combined learning curve all grid search runs. Fig. 3.2 shows the Pareto front of the grid search. An interesting feature in Fig. 3.4 is the underperformance of models in the diagonal. The amount of trainable parameters of the models considered can be seen in Fig. 3.5.

Table 3.12: Results of the full factorial ANOVA of embedding dimension (ED) and number of encoder layers (NL). The ANOVA model includes linear, quadratic and interaction terms.

Term	Sum Sq	df	F	PR(>F)	R^2
ED	1,57E+09	1.0	97,805333	**0,002199**	66,30%
NL	7,73E+07	1.0	4,831624	0,115377	3,28%
ED^2	4,72E+08	1.0	29,455497	**0,012275**	19,97%
NL^2	3,77E+07	1.0	2,352144	0,222656	1,59%
ED*NL	1,61E+08	1.0	10,067658	**0,050366**	6,82%
ANOVA model	2,31E+09	3.0	-	-	97,97%
Residual	4,80E+07	3.0	-	-	2,03%

Figure 3.4: Optimality gaps obtained for each run in the grid search, using beam search to decode the policy.

		EL		
		2	3	4
	128	53,68%	40,85%	45,95%
EB	256	74,54%	59,46%	39,03%
	512	41,87%	36,79%	66,04%

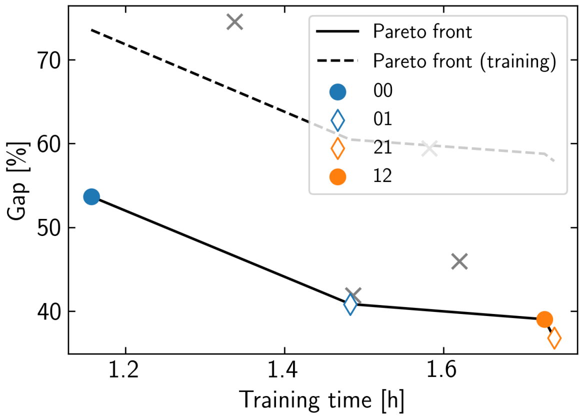


Figure 3.2: Pareto front of the full factorial design, using beam search to decode the learned policies.

Figure 3.5: Number of trainable parameters for each configuration considered in the grid search.

		EL		
		2	3	4
	128	**0.51M**	×1,4	×1,8
ED	256	×3,0	×**4,0**	×5,0
	512	×9,8	×12,8	×**15,9**

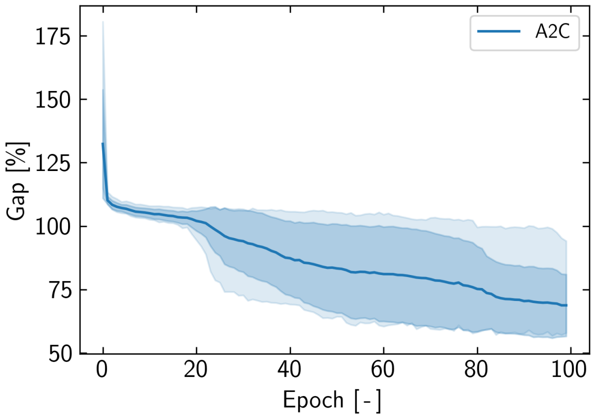


Figure 3.3: Grid search learning curve. During training the policy is decoded using greedy search.

A second ANOVA was performed on the results of the grid search. This time quadratic and interaction effects were included in the ANOVA model. The results follow in Tab. 3.12. The Shapiro-Wilk test ($p = 0.9004$) and Anderson-Darling test ($p = 0.153$) again confirm that the residuals are normally distributed, and homoscedasticity is visually verified, confirming the validity of the analysis. The ANOVA model accounts for 97.97% of the observed variance, indicating a strong fit. The full factorial ANOVA confirms that both embedding dimension and number of layers have significant effects on performance. Embedding dimension shows a strong main effect ($F = 97.81$, $p = 0.0022$) and a significant quadratic term ($F = 29.46$, $p = 0.0123$). The interaction between embedding dimension and number of layers is marginally significant ($F = 10.07$, $p = 0.0504$), indicating that their combined influence affects performance. These findings confirm the significant linear effects identified by the L27 fractional ANOVA and indicate the presence of additional non-linear relationships.

The results indicate that network architecture is the principal driver of model performance, but model performance increases asymptotically slowly with the number of parameters.

3.3.3 Impact of Training Dataset Size on Policy Performance

The amount of data provided to the model for training is key for its performance [18]. This is especially the case for attention-based ML models [122]–[124]. Fig. 3.6 shows the effect of increasing the size of the training dataset from 1M to 3M tours of 10 transfers. Observe the large increase in convergence speed. The final performance improves as well. Notwithstanding, model performance eventually plateaus to a validation optimality gap in training of approximately 50%. This confirms that the architecture of the ML model is the factor limiting further learning.

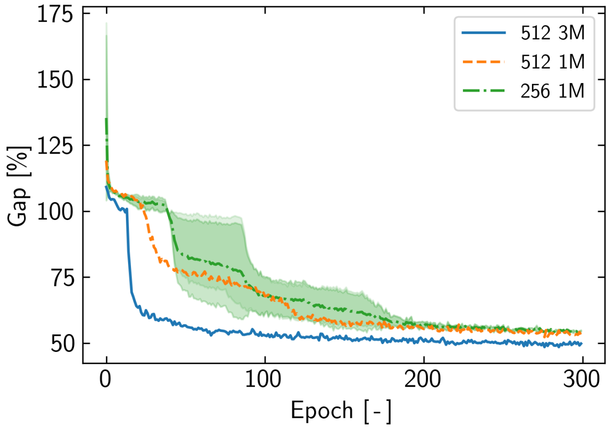


Figure 3.6: Final learning curves. Note the impact of increasing training dataset size from 1M to 3M. Notwithstanding, the capacity of the model to learn asymptotically approaches a similar limit at 50% optimality gap.

3.3.4 Final Performance and Generalization to Larger Routing Problems

The final policy was trained on 3 million tours consisting of 10 transfers. To evaluate the capacity of the policy to generalize to larger problems, the trained policy was employed to plan scenarios with 10, 30, and 50 transfers. BS was used to search the trained policy. Tab. 3.13 presents the final performance results of the trained NCO policy compared to the DRW heuristic across scenarios with 10, 30, and 50 transfers. Each case consisted in the planning of 1000 missions. The metrics include the mean and standard deviation of ΔV (change in velocity), the optimality gap percentage, and the evaluation time in milliseconds. Optimality gaps are obtained by comparison with the solution obtained using HCO.

The trained NCO policy exhibits better performance in the 10-node scenario than the DRW heuristic. However, as the number of nodes increases to 30 and 50, the performance of the NCO policy greatly deteriorates. This indicates a limitation in the policy's ability to generalize to mission scenarios with a higher number of transfers: the learned policy is not generally applicable for the design of ADR missions.

In terms of computational efficiency, the NCO policy outperforms DRW across the board. The difference is large for small mission scenarios, but becomes less significant for 30 and 50 transfer scenarios.

Table 3.13: Policy performance compared to the DRW STSP heuristic for STSPs with 10, 30 and 50 targets. Performance measured on test datasets of 1000 missions.

Number of nodes		**10**		**30**		**50**	
Heuristic		**AM BS**	**DRW**	**AM BS**	**DRW**	**AM BS**	**DRW**
ΔV [km/s]	μ	**106,4**	122,1	573,5	**208,1**	1041,8	**261,2**
	σ	16,2	31,4	83,9	46,2	68,2	52,6
Optimality gap [%]	μ	**32,6**	50,5	616,0	**50,4**	555,4	**63,9**
	σ	25,5	36,8	132,7	37,2	97,4	38,3
Evaluation time [ms]	μ	**99,2**	2494,9	**3592,0**	8534,5	**8630,0**	16147,1

4 Neural Combinatorial Optimization for Multi-Rendezvous Mission Design

Neural Combinatorial Optimization for Multi-Rendezvous Mission Design

Antonio López Rivera
Delft University of Technology, Delft, Kluyverweg 1, 2629 HS Delft, The Netherlands

Optimal solutions to space vehicle routing problems are essential for space logistics activity such as Active Debris Removal (ADR), which addresses the growing threat of space debris. This research investigates the applicability and effectiveness of Neural Combinatorial Optimization (NCO) methods in designing low-thrust multi-target rendezvous trajectories for ADR missions. An attention-based routing policy, comprising a graph attention network and a pointer network, was trained using REINFORCE, Advantage Actor-Critic, and Proximal Policy Optimization. Hyperparameter analysis with ANOVA identified embedding dimension and encoder layers as the critical factors influencing model performance. The trained policy was evaluated on scenarios with 10, 30, and 50 transfers based on the Iridium 33 debris cloud. In missions with 10 transfers, the NCO policy achieved a mean optimality gap of 32%, outperforming the Dynamic RAAN Walk heuristic. However, performance degraded in scenarios with 30 and 50 transfers, indicating limited generalization beyond training conditions. Grid search hyperparameter optimization revealed that model performance improves asymptotically with model complexity, while larger training datasets enhanced convergence speed with marginal gains in final performance. This research demonstrates that NCO methods can be effective for ADR missions with a limited number of targets, but face scalability and generalization challenges in more complex scenarios.

Nomenclature

Abbreviations

A2C	Advantage Actor-Critic
A3C	Asynchronous Advantage Actor-Critic
ADR	Active Debris Removal
ANOVA	Analysis of Variance
AOP	Argument of Perigee

CO	Combinatorial Optimization
DNN	Deep Neural Network
DRW	Dynamic RAAN Walk
ECI	Earth-Centered Inertial frame
FYS	Fisher-Yates Shuffle
GAT	Graph Attention Network
GNN	Graph Neural Network
GPU	Graphics Processing Unit
KS	Knuth's Algorithm for Uniform Permutation Sampling using Sobol Points
LFC	Lyapunov Feedback Control
LEO	Low Earth Orbit
LVLH	Local-Vertical Local-Horizontal frame
MEE	Modified Equinoctial Elements
MDP	Markov Decision Process
MEO	Medium Earth Orbit
MINLP	Mixed-Integer Nonlinear Programming
ML	Machine Learning
NCO	Neural Combinatorial Optimization
NN	Nearest-Neighbour Search
OOS	On-Orbit Servicing
PN	Pointer Network
PPO	Proximal Policy Optimization
Q-Law	Lyapunov Feedback Control Law based on the proximity quotient Q
REINFORCE	Monte Carlo Policy Gradient Algorithm
RL	Reinforcement Learning
RQ-Law	Rendezvous Q-Law
RSW	Radial, Along-track (S), Cross-track (W) frame

SCP	Sequential Convex Programming
RCS	Radar Cross-Section
SHP	Sobol Hypercube Permutations
SMA	Semi-major Axis
SP	Sobol Permutations
STSP	Spacecraft Traveling Salesman Problem
TOF	Time of Flight
TSP	Traveling Salesman Problem
VRP	Vehicle Routing Problem

Mathematical Notation

r	Scalar variable
$\mathbf{r}$	Vector variable
$\|\|\mathbf{r}\|\|$	Euclidean norm of vector $\mathbf{r}$
$\overline{r}$	Mean value of r
$r \sim P$	r is sampled from distribution P
$\dot{Q}$	Time derivative of Q
$\lceil x \rceil$	Ceiling function of x
$\min(a, b)$	Minimum of a and b
$\|x\|$	Absolute value of x
$\text{argsort}(z)$	Indices that would sort vector z
$\sin(x), \cos(x), \tan^{-1}(x)$	Trigonometric functions
$B(p, q)$	Beta function
$F_j^{-1}(x)$	Inverse cumulative distribution function
$\mathbb{R}^n$	n-dimensional real space
$\mathbb{S}^{d-1}$	Unit sphere in $\mathbb{R}^d$

$\mathfrak{S}^n$	Symmetric group of permutations of n elements

Latin Symbols

a	Semi-major axis	m
a_T	Target semi-major axis	m
A	Reference area; advantage function	m^2; –
A_t	Advantage at time t	–
$A^{(i)}$	Advantage for instance i	–
a_t	Action at time t	–
$a^{(i)}$	Solution for instance i	–
$a_{J_{2,r}}, a_{J_{2,\theta}}, a_{J_{2,\phi}}$	Components of acceleration due to J_2 perturbation	m s^{-2}
c	Constant or parameter	–
C_D	Drag coefficient	–
d	Dimension	–
e	Eccentricity	–
e_T	Target eccentricity	–
f, g, h, k	Equinoctial elements	–
f_{burn}	Burn frequency	s^{-1}
G	Universal gravitational constant	6.674 30e−11 m^3 kg^{-1} s^{-2}
G_t	Return at time t	–
$G^{(i)}$	Return for instance i	–
g_0	Standard Earth gravity	9.806 65 m s^{-2}
i	Index; inclination	–; rad
i_1, i_2	Inclinations of initial and target orbits	rad
I_{sp}	Specific Impulse	–
J_2	Second zonal harmonic coefficient	1.0826e−3

j	Index	–
k	Number of burns; scaling parameter	–
k_d, k_c	Number of burns for departure and circularization	–
L	True longitude; loss function	rad; –
L_t	Clipped objective at time t	–
$L^{(i)}$	Clipped objective for instance i	–
L_{policy}	Policy loss	–
L_{value}	Value function loss	–
L_0, L_1	Initial and target true longitude	rad
m	Spacecraft mass	kg
m_0	Initial mass	kg
m_f	Fuel mass	kg
m_1, m_2	Masses of bodies 1 and 2	kg
n	Orbital mean motion	s^{-1}
$\hat{\mathbf{e}}_r, \hat{\mathbf{e}}_\theta, \hat{\mathbf{e}}_\phi$	Basis vectors in LVLH frame	–
$\hat{\mathbf{u}}$	Thrust direction unit vector	–
p	Semi-latus rectum	m
P	Orbital period	s
Π	Set of permutations	–
r	Radial distance; scalar variable	m; –
r_0	Radius of initial orbit	m
r_1	Radius of target orbit	m
r_p	Periapsis radius	m
$r_{p_{\text{min}}}$	Minimum periapsis radius	m
$R(x^{(i)}, a^{(i)})$	Reward function for instance i	–
s^2	Variable in Gauss Variational Equations	–
s_t	State at time t	–

T	Thrust magnitude; time horizon	N; –
T_{burn}	Burn time	s
T_{cooldown}	Cooldown time	s
t	Time index	–
t_0	Start time	s
$\mathcal{T}$	Best time-to-go	s
v	Variable in Gauss Variational Equations; velocity	–; m s^{-1}
v_{eq}	Exhaust velocity	m s^{-1}
V_0	Orbital velocity	m s^{-1}
$V_\phi(s_t)$	Estimated value function at state s_t	–
$V_\phi(x^{(i)})$	Estimated value function for instance $x^{(i)}$	–
w	Variable in Gauss Variational Equations	–
x	Vector in $\mathbb{R}^n$; problem instance	–
$x^{(i)}$	Problem instance i	–
$\mathbf{a}_{J_2}$	Acceleration due to J_2 perturbation	m s^{-2}
$\mathbf{F}_{1,2}$	Gravitational force between bodies 1 and 2	N
$\mathbf{r}_{1,2}$	Position vector from body 1 to body 2	m
$r^{(i)}(\theta)$	Probability ratio for instance i	–
$r_t(\theta)$	Probability ratio at time t	–
$\mathbf{v}$	Velocity vector	m s^{-1}
$\dot{m}_f$	Mass flow rate	kg s^{-1}
$W_x, W_p, W_{\text{œ}}, W_L, W_{\text{scl}}$	Weights in Q-law and RQ-law	–
œ	Modified Equinoctial Elements (a, f, g, h, k)	–
œ_T	Target orbital elements	–
$\text{œ}_{T,\text{aug}}$	Augmented target elements	–

Greek Symbols

α	Learning rate for policy	–
β	Learning rate for value function	–
γ	Relative inclination	rad
δ	Small change or variation	–
ϵ	Clipping parameter	–
μ	Earth's gravitational parameter	3.986e14 $\mathrm{m}^3\,\mathrm{s}^{-2}$
ξ	Ratio of orbit radii (r_1/r_0)	–
ρ	Atmospheric density	$\mathrm{kg\,m}^{-3}$
θ	Policy parameters; true anomaly	–; rad
θ_{old}	Previous policy parameters	–
ϕ	Value function parameters; spherical coordinates	–
$\pi_\theta(a_t\|s_t)$	Policy probability of action a_t given state s_t under parameters θ	–
$\pi_{\theta_{\text{old}}}(a_t\|s_t)$	Policy with old parameters θ_{old}	–
σ	True anomaly	rad
τ	Trajectory (sequence of states, actions, rewards)	–
ω	Argument of Perigee	rad
Ω	Right Ascension of the Ascending Node	rad
Ω_1, Ω_2	RAAN of initial and target orbits	rad
Ψ	Transformation matrix in MEEs	–
Δi	Change in inclination	rad
ΔV	Delta-V (change in velocity)	$\mathrm{m\,s}^{-1}$
$\Delta_r, \Delta_t, \Delta_n$	Perturbing accelerations in radial, tangential, normal directions	$\mathrm{m\,s}^{-2}$
$\Delta L_{[-\pi,\pi]}$	Difference in true longitude wrapped to $[-\pi, \pi]$	rad

Subscripts

t	At time step t
(i)	For instance i
r	Radial component
n	Normal component
θ	Tangential component
ϕ	Normal component (in LVLH frame)
I	Inertial frame
s	Spacecraft
C	Central body (Earth)
old	Previous parameter value
0	Initial value
T	Target value
ECI	Earth-Centered Inertial frame
MEE	Modified Equinoctial Elements frame
D	Related to drag
J_2	Related to J_2 perturbation

Introduction

The design of multi-target rendezvous manoeuvres, which see a spacecraft approaching a sequence of objects in orbit as efficiently (by some metric) as possible, has seen a considerable surge in interest in recent years for the purposes of Active Debris Removal (ADR) missions [1–6] to tackle the space debris problem [7, 8], as well as On-Orbit Servicing (OOR) missions [3, 9, 10] and advanced space logistics concepts [11].

The problem of designing such trajectories, known as the Space Traveling Salesman Problem (STSP), is an example of a Mixed Integer Non-Linear Programming (MINLP) problem with factorial complexity over the number of targets. MINLP problems are notoriously difficult to approach. An optimal solution to the STSP consists of the optimal sequence in which to visit a set of targets and the optimal (by some metric) transfer trajectory between each target in the optimal sequence. The STSP is conceptually related to the classical Traveling Salesman Problem (TSP), with the added complexities inherent to the space environment: notably, a 6-dimensional non-Euclidean state space, mass dynamics, spacecraft propulsion constraints, and the progressive drift of the states of orbiting bodies due to secular perturbations, chiefly J_2 for Earth-orbiting spacecraft.

Formally, the STSP is the problem of finding a minimum weight path (if the spacecraft must end the tour back at its initial state, a Hamiltonian path) in a complete weighted graph $G \coloneqq \{\mathcal{V}(t), \mathbf{W}(\pi)\}$, where $\mathcal{V}(t)$ is the set of graph vertexes (targets, the state of which drifts over time) and $\mathbf{W}(\pi) \coloneqq \mathcal{V} \times \mathcal{V} \to \mathbb{R}^+$ is a map that associates an edge weight (a transfer cost) to each ordered vertex pair [1], and may dependent on the sequence π in which the targets are visited. One such case is when payload mass is a large percentage of the spacecraft's wet mass, and thus deployment sequence has a non-negligible impact on fuel consumption. A standard approach to solve the STSP is Benders decomposition [3–6], where the MINLP problem is divided into a higher-level Combinatorial Optimization (CO) problem and a lower-level trajectory optimization problem. A transfer cost estimator is then used to calculate the cumulative cost of tours in the CO problem. Transfer cost estimators may be database-dependent [12, 13], database-independent (analytical), or learning-based [14].

State-of-the-art CO methods fall in two camps: exact methods and heuristic methods, which are less costly and can produce near-optimal results, but cannot offer optimality guarantees whatsoever [1, 15]. Exact methods based on tree searches [16, 17] are the norm for highly complex, large STSP variants; all winning submissions of the Global Trajectory Optimization Competitions have made use of tree search approaches [1, 12, 18]. Heuristic optimization methods however are an attractive option to solve smaller STSP instances (up to hundreds of targets [1]) due to their capacity to achieve near-optimal results with lower computational cost [1], and are widely applied in literature to tackle multi-rendezvous mission design [1–4, 6]. Heuristic optimization methods have also been successfully applied to complex STSP instances where the cost of exact approaches is unfeasible [12]. High quality approximate solutions are highly desirable for the heuristic optimization process. As the complexity of the generalized Vehicle Routing Problem (VRPs) increases, high quality approximate solutions become more difficult to obtain [19]. This bodes ill for the field of space logistics, as the complexity of space VRPs beyond the STSP is bound to increase over time: this will happen as LEO and MEO become more congested, the in-space manufacturing and servicing industries rise, and space logistics operations become more complex [20]. It becomes pressing to ask whether learning-based methods from the field of CO could be applied in the space domain.

Machine Learning (ML) approaches for spacecraft trajectory design have seen a surge of interest in recent years [21], with strong results achieved both for trajectory cost estimation [14] and spacecraft guidance [22]. Neural Combinatorial Optimization (NCO) uses Deep Neural Networks (DNNs) to automate the problem-solving process, mostly under the RL paradigm, as supervised learning is often unfeasible for large or theoretically hard problems. NCO offers the attractive prospect of alleviating the scaling issues of exact approaches, while removing the need for handcrafted heuristics, which often require significant domain-specific adjustments [19]. NCO has shown promising performance on various CO problems [19], especially when coupled with advanced policy search procedures [15].

The present work aims to assess the applicability and effectiveness of NCO approaches for the design of multi-target rendezvous trajectories. It is of interest to consider both the standalone effectiveness of NCO approaches to solve the STSP, as well as the performance of NCO approaches combined with conventional methods.

I. Background and Preliminaries

This section will introduce the necessary background for the discussion and assessment of NCO methods for space VRPs. A brief historical overview of the space debris crisis is presented in Section I.A along with the case study considered in this work: an ADR mission targeting the Iridium 33 debris cloud. Orbital dynamics are discussed in Section I.B, followed by Low-Thrust Trajectory Optimization in Section I.C. Heuristic combinatorial optimization is discussed last in Section I.D.

A. Space Debris and Space Debris Remediation

The present work focuses on the design of multi-rendezvous trajectories for ADR missions. Since the first recorded catastrophic fragmentation event in 1961 [23], more than 200 such events have contributed to a population of over 34,000 trackable fragments larger than 10 cm in Low Earth Orbit (LEO) [24]. The rapid increase of debris density led to the Kessler Syndrome hypothesis in 1978 [25]: recent studies indicate critical debris density may have been reached as of the present day [26, 27]. Awareness of the problem has grown rapidly in recent years [20, 27]. Efforts to mitigate space debris so far have been strongly biased towards prevention [28–30]; compliance with measures is not globally enforced however. Active Debris Removal (ADR) has emerged as a crucial means to mitigate the threat from existing debris [31]. The removal of large uncompliant objects from orbit has been the primary focus of ADR mission design up to the present day [8, 32–38]. Against this trend, the 2024 NASA OTPS Phase 2 report [20] finds that de-orbiting 1–10 cm debris to prevent collisions may yield substantial economic returns by reducing collision risks and associated costs for satellite operators. These conditions herald opportunity for efficient multi-rendezvous ADR missions and constellations to mitigate the threat from existing debris. This study aims to investigate the viability of NCO methods for the design of such missions.

The case study considered in this work is an ADR mission targeting the Iridium 33 debris cloud [29, 39]. Iridium 33 is one of the most widely studied clouds in LEO, other notable clouds being the Cosmos 2242 [39], Fengyun 1C [40] and Cosmos 1408 [41] clouds. The Gabbard diagram [42] of the four clouds can be seen in Fig. 1. Radar Cross-Section (RCS) and expected decay time data is present as well. A tabulated summary of the four clouds follows in Table 6. The Fengyun 1C cloud will outlast all other clouds: up to 1000 pieces of debris will remain in orbit by the year 2100 according to ESA estimates. Up-to-date satellite tracking data is obtained from CelesTrak[*], and decay time data is obtained from the ESA Database and Information System Characterising Objects in Space (DISCOS) database[†].

[*] https://celestrak.org

[†] https://discosweb.esoc.esa.int

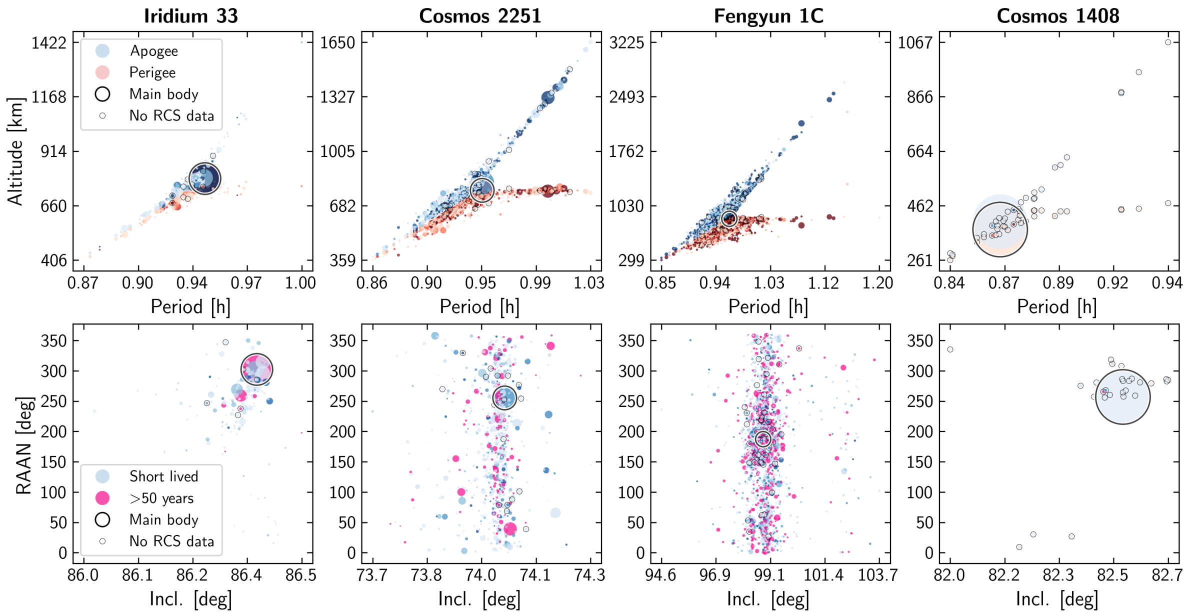


Fig. 1 Gabbard diagram of the Iridium 33, Cosmos 2251, Fengyun 1C and Cosmos 1408 debris clouds as of September 2024. Point size proportional to RCS. Color intensity in top row proportional to lifetime before natural decay. Data obtained from Celestrak as of September 2024. Own work.

Table 6 Number of debris fragments and RCS in [m^2] of the Iridium 33, Cosmos 2252, Fengyun 1C and Cosmos 1408 debris clouds at the moment of fragmentation event (T0), as of the present day in 2024, and estimates for the year 2050 and 2100. Estimates are obtained from the ESA DISCOS database.

	Iridium 33		**Cosmos 2251**		**Fengyun 1C**		**Cosmos 1408**	
	Count	RCS	Count	RCS	Count	RCS	Count	RCS
T0	631	20,64	1626	34,81	3043	55,29	1801	7,51
2024	193	10,24	831	21,13	2192	42,96	68	7,50
2050	19	3,61	207	7,63	903	20,65	0	0,00
2100	5	3,01	76	3,38	435	11,04	0	0,00

Fig. 2 displays the altitude and RAAN distributions of the four clouds together as a function of inclination. This offers a mission designer's view of the problem: a map relating the most important orbital parameters for mission design in LEO to the likelihood of collisions with debris from the Iridium 33, Cosmos 2252, Fengyun 1C and Cosmos 1408 clouds. Observe the large range of RAAN values in the three main clouds —Iridium 33, Cosmos 2252 and Fengyun 1C—. The Fengyun 1C cloud stands out in stark contrast both in terms of physical properties —the ranges of RAAN and inclination, the ranges of semi-major axis and eccentricity, the number of active debris and the lifetime of active debris— as well as damage to space operations, as the 96°-98° inclination range is critical for Sun-Synchronous Orbits (SSO) in LEO. As the cost of redezvous is primarily driven by plane change cost [1], ADR missions are bound to require extreme amounts of ΔV.

Electrical propulsion is the only viable means for very high ΔV missions [43, 44]. An electrical propulsion ADR

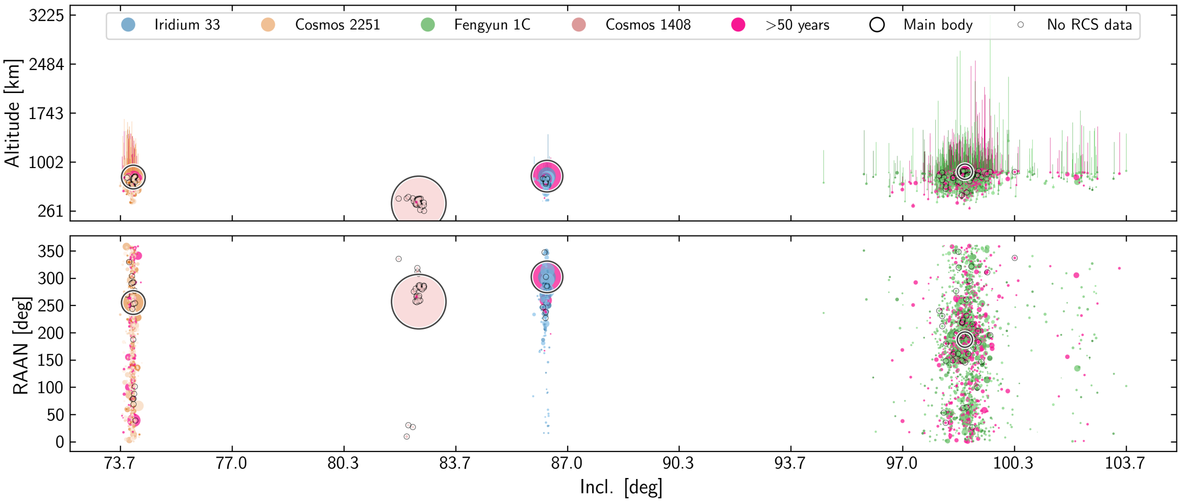


Fig. 2 Joint altitude-RAAN-inclination diagram of the Iridium 33, Cosmos 2251, Fengyun 1C and Cosmos 1408 debris clouds as of September 2024. Point size proportional to RCS. Color intensity in top row proportional to lifetime before natural decay. Color trails, top: perigee to apogee altitude. Data obtained from Celestrak as of September 2024. Own work.

spacecraft concept will be considered in this work; specifications follow in Table 7. The propulsion system is based on the specifications of existing Gridded Ion Thruster designs for small spacecraft, using the same power to thrust ratio of 21 kW N^{-1} as the MiXi GIT propulsion system developed at UCLA [45] (for more information refer to O'Reilly's extensive review of electric propulsion systems for small spacecraft [44]). Spacecraft structural, fuel and payload mass are only indicative of a spacecraft fit for this type of mission; payload mass is sized to 10-30 active de-orbiting payloads [31, 38].

Table 7 Debris chaser spacecraft specifications.

Wet mass	Fuel mass	Payload mass	Max a_T	Min a_T	I_{sp}	Delta V budget	Max thrust	Max power
1200 kg	450 kg	500 kg	3e–4 m s^{-2}	1e–5 m s^{-2}	3000 s	6.00 km s^{-1}	0.36 N	7.55 kW

B. Dynamics

The state of the spacecraft is propagated using the MEEs described by Hintz [46] including the retrograde factor I, which are nonsingular for all eccentricities and inclinations. The conversion from classical Keplerian elements [46] to MEEs follows in Eq. 1:

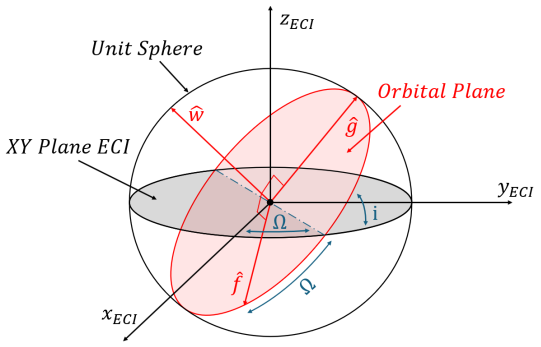


Fig. 3 Modified Equinoctial Elements (MEE) with respect to orbital plane.

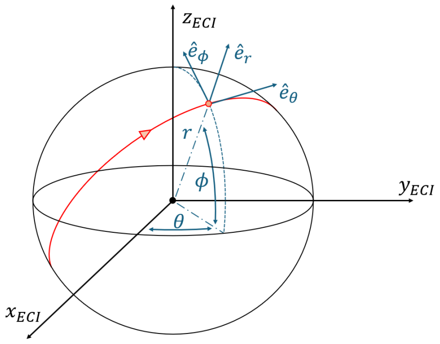


Fig. 4 ECI and LVLH reference frames.

$$
\begin{aligned}
&p = a(1-e^2);\\
&f = e\cos(\omega+\Omega);\\
&g = e\sin(\omega+\Omega);\\
&h = \tan(i/2)\sin(\Omega);\\
&k = \tan(i/2)\cos(\Omega);\\
&L = \theta + I\Omega + \omega;
\end{aligned}
\tag{1}
$$

The Gauss Variational Equations for MEEs [46], in Eq. 2, are used to model the time evolution of the spacecraft's state:

$$
\begin{aligned}
\frac{\mathrm{d}p}{\mathrm{d}t} &= \frac{2p}{w}\sqrt{\frac{p}{\mu}}\Delta_t;\\
\frac{\mathrm{d}f}{\mathrm{d}t} &= \sqrt{\frac{p}{\mu}}\left\{\Delta_r\sin(L) + \frac{(w+1)\cos(L)+f}{w}\Delta_t - g\frac{v}{w}\Delta_n\right\};\\
\frac{\mathrm{d}g}{\mathrm{d}t} &= \sqrt{\frac{p}{\mu}}\left\{-\Delta_r\cos(L) + \frac{(w+1)\sin(L)+g}{w}\Delta_t - f\frac{v}{w}\Delta_n\right\};\\
\frac{\mathrm{d}h}{\mathrm{d}t} &= \sqrt{\frac{p}{\mu}}\frac{s^2}{2w}\cos(L)\Delta_n;\\
\frac{\mathrm{d}k}{\mathrm{d}t} &= \sqrt{\frac{p}{\mu}}\frac{s^2}{2w}\sin(L)\Delta_n;\\
\frac{\mathrm{d}L}{\mathrm{d}t} &= \sqrt{\mu p}\left(\frac{w}{p}\right)^2 + \sqrt{\frac{p}{\mu}}\frac{v}{w}\Delta_n;
\end{aligned}
\tag{2}
$$

where s^2, v and w are defined as follows,

$$\begin{aligned} s^2 &= 1 + h^2 + k^2; \\ v &= h\sin(L) - k\cos(L); \\ w &= 1 + f\cos(L) + g\sin(L); \end{aligned} \tag{3}$$

and Δ_r, Δ_t and Δ_n are perturbing accelerations in the radial, tangential, and normal directions of the spacecraft's LVLH frame $\hat{e}$ depicted in Fig. 4. The unit thrust vector in the ECI frame, $\hat{\mathbf{u}}_{\mathrm{ECI}}$, is related to the unit thrust vector in the LVLH frame $\hat{\mathbf{u}}_{\mathrm{MEE}}$ by Eq. 4.

$$\hat{\mathbf{u}}_{\mathrm{ECI}} = [\hat{\mathbf{e}}_r \quad \hat{\mathbf{e}}_\theta \quad \hat{\mathbf{e}}_\phi]\hat{\mathbf{u}}_{\mathrm{MEE}} \tag{4a}$$

$$\hat{\mathbf{e}}_r = \frac{\mathbf{r}}{||\mathbf{r}||}; \quad \hat{\mathbf{e}}_\phi = \frac{\mathbf{r}\times\mathbf{v}}{||\mathbf{r}\times\mathbf{v}||}; \quad \hat{\mathbf{e}}_\theta = \hat{\mathbf{e}}_\phi \times \hat{\mathbf{e}}_r \tag{4b}$$

Then, the thrust acceleration $\mathbf{a}_T$ applied by the spacecraft in the RWS frame is defined in Eq. 5,

$$\mathbf{a}_T = \frac{T}{m}\hat{\mathbf{u}} \tag{5}$$

where $\hat{\mathbf{u}}$ is the direction of application of thrust. The spacecraft's mass is propagated assuming constant specific impulse (I_{sp}) through the burn. The change in mass over time is defined by Eq. 6:

$$\frac{\mathrm{d}m}{\mathrm{d}t} = \frac{T}{v_{\mathrm{eq}}}; \tag{6}$$

where T is the applied thrust and $v_{\mathrm{eq}} = I_{sp}g_0$ is the equivalent exit velocity of the engine.

C. Low-Thrust Trajectory Optimization

Of the family of direct Low-Thrust Trajectory Optimization (LTTO) methods [43, 47] that make use of predefined control laws [43, 48], Lyapunov Control (LC) methods are notable for being both fast and able to generate reasonable estimates of optimal planetocentric trajectories [43]. Since their introduction by Ilgen in 1993 [49], LC methods have been extended to incorporate a variety of mission constraints [43, 50, 51]. A notable advantage of LC laws is that they naturally drive the spacecraft to the desired final state, removing the need to include boundary conditions on the final state [43]. The Q-Law in particular, introduced and refined by Petropoulos [50, 52], has been widely used for preliminary mission design [43] as well as to generate initial guesses for high-fidelity tools, notably JPL's Mystic [43, 53, 54]. In 2023 Narayanaswamy et al. [55] introduced the Rendezvous Q-law (RQ-law), capable of dynamic six-element targeting by means of a semi-major axis augmentation scheme.

1. Q-Law

The Q-law, originally introduced by Petropoulos [52], is a Lyapunov feedback control law for low-thrust trajectory optimization based on the "proximity quotient" Q, a candidate Lyapunov function that approximates the best quadratic time-to-go [52]. MEE formulations of the Q-Law were later developed by Petropoulos [50] and Varga [51]. Q is defined in Eq. 7,

$$\mathrm{Q}(\text{œ}, \text{œ}_T, W_x) = (1 + W_p P) \sum_{\text{œ}} S_{\text{œ}} W_{\text{œ}} \left(\frac{\text{œ} - \text{œ}_T}{\dot{\text{œ}}_{xx}} \right)^2, \quad \text{œ} = a, f, g, h, k. \tag{7}$$

where the periapsis penalty P is defined in Eq. 8 and the element scaling factors $S_{\text{œ}}$ are defined in Eq. 9.

$$P = \exp\left(k \left(1 - \frac{r_p}{r_{p_{\min}}} \right) \right) \tag{8}$$

$$S_{\text{œ}} = \begin{cases} \left(1 + \left(\frac{|a - a_T|}{m a_T} \right)^n \right)^{\frac{1}{r}} & \text{œ} = a, \\ 1 & \text{œ} = f, g, h, k. \end{cases} \tag{9}$$

The feedback control law is derived such that $\dot{Q}$ is negative definite, and follows in Eq. 10. This follows from applying the chain rule $\dot{Q} = \nabla Q \dot{\boldsymbol{\text{œ}}}$ and observing that $\dot{\boldsymbol{\text{œ}}} = \Psi \mathbf{u}$, where Ψ stands for the Gauss variational equations in MEEs (Eq. 2).

$$\mathbf{u} = -\Psi^{\top} \nabla Q \tag{10}$$

2. Rendezvous Q-Law

The Rendezvous Q-law (RQ-law), proposed by Narayanaswamy [55], builds on the work of Lantukh et al. [56] extending the Q-law to enable dynamic six-element targeting by means of a semi-major axis augmentation scheme (Eq. 11). The scheme is designed to induce an error in the semi-major axis related to the difference between the current and desired true longitude. The scheme becomes active after reaching 5-element convergence, splitting the manoeuvre in two phases: orbit acquisition and longitude acquisition, or phasing.

$$\text{œ}_{T,\text{aug}} = \begin{cases} a_T + \frac{2W_L}{\pi} \left(a_T - \frac{r_{p,\min}}{1 - \sqrt{f_C^2 + g_C^2}} \right) \tan^{-1}(W_{\text{scl}} \Delta L_{[-\pi,\pi]}), & \text{œ} = a \\ \text{œ}_T, & \text{œ} \in \{f, g, h, k\} \end{cases} \tag{11}$$

3. Perturbations and Constraints

The most relevant perturbations for planetocentric trajectories are gravity field distortions, chiefly J_2 in the case of the Earth [12, 51]. Perturbing terms are seldom included in the formulation of the Q-Law, as the feedback control tends to waste fuel to counter-act the perturbing accelerations instead of taking advantage of them [50, 51]: instead, the parameters of the Q-Law are optimized to obtain near-optimal results under the effect of perturbations, using either conventional [51] or ML-based methods [57]. Eclipse and duty cycle constraints are also important for electric propulsion spacecraft [51].

Minimizing trajectory generation time is highly desirable, as a very large number of trajectories must be generated to train the NCO policy and assess the viability of NCO for space VRPs. The dynamicity of orbiting debris however, chiefly driven by the secular impact of the J_2 perturbation, is critical to the complexity of the STSP, and cannot be neglected [1]. To balance performance and realism this study considers unperturbed transfers between orbiting targets, while propagating the cloud according to the secular impact of the J_2 perturbation [12, 18]. The secular impact of J_2 on the Right Ascension of the Ascending Node (RAAN) and Argument Of Perigee (AOP) or orbiting debris is described by Eq. 12 [58], where $n = \sqrt{\mu/a^3}$ is the mean motion of the orbiting body.

$$\begin{aligned}
\frac{\mathrm{d}\Omega}{\mathrm{d}t} &= -\frac{3}{2}J_2\left(\frac{R_e}{p}\right)n\cos i; \\
\frac{\mathrm{d}\omega}{\mathrm{d}t} &= -\frac{3}{4}J_2\left(\frac{R_e}{p}\right)n(5\cos^2 i - 1));
\end{aligned} \tag{12}$$

4. Simulation

The Tudat Space[‡] astrodynamics library [59] is used to implement the simulator. Integration is performed using an Adams-Bashforth-Moulton integrator of orders 6-8 and variable step size (using global and relative tolerance of 3.2×10^{-9}). The choice of integrator results from a trade-off of all Tudat integrators considering computational cost, accuracy and output density. The original Q-Law has eleven parameters (five being element weights), and the RQ-Law adds two parameters more (one being an element weight). Table 8 lists the values of the Q-Law and RQ-Law parameters used. Table 9 lists the element weights and convergence tolerances used. The values of the parameters and weights are those recommended by Varga et al. [51] in the case of the Q-Law and by Narayanaswamy et al. [55] in the case of the RQ-Law.

Fig. 5 shows the average transfer in the RAAN walk [1] across the Iridium 33 cloud. The transfer consists of raise of SMA of 84 km, a change in eccentricity of 4 1e−3, a change in inclination of 1 1e−3°, a change in RAAN of 2 79e−2°, and changes in AOP and True Anomaly (TA) of approximate 1.4°. The total manoeuvre time is of 55 hours, of which 51 are spent in the orbit acquisition leg and the phasing leg lasting under slightly under 4 hours. This will be a relevant

[‡] `https://docs.tudat.space/en/latest/`

Table 8 Q-Law and RQ-Law parameter values.

k	m	n	r	b	W_p	W_l	W_{scl}
100.0	3.0	4.0	2.0	0.01	1.0	0.0594	3.6230

Table 9 Element weights and convergence tolerances.

Element	$W_{\text{œ}}$	$W_{\text{œ,phasing}}$	**Tolerance**	**Relaxed Tolerance**
a	1	10	1×10^{2} m	1×10^{3} m
e	1	1	1×10^{-3} deg	1×10^{-2} deg
i	1	1	1×10^{-3} deg	1×10^{-2} deg
Ω	1	1	1×10^{-3} deg	1×10^{-2} deg
ω	1	1	1×10^{-3} deg	1×10^{-2} deg
θ	—	—	1×10^{-1} deg	1 deg

consideration for transfer time estimation, which is the next topic of discussion. Lastly, Fig. 6 shows extreme transfers scenarios in the Iridium 33 and Fengyun 1C debris clouds, with the spacecraft traveling from the center-of-RCS of the cloud to the center-of-RCS of the cloud plus 3 times the standard deviation of each element in the cloud.

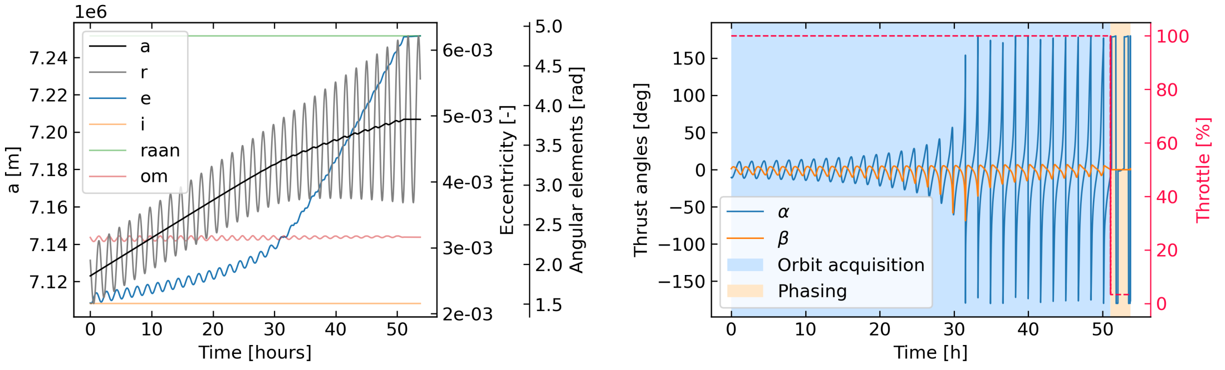
5. Transfer Cost Estimation

The capacity to quickly estimate the duration and cost (in ΔV or fuel mass) of low-thrust transfers *without the need to propagate* is highly attractive for NCO, as the training process requires estimating the cost of many (millions of) transfers. Fast, approximate transfer cost estimation methods are commonly used to solve the higher level combinatorial problem in STSPs, especially for complex problems [12, 18]. Transfer cost estimation approaches (both impulsive and low-thrust) may be either analytical [4, 18, 60], which rely on simplifying assumptions to obtain closed-form expressions of transfer cost, or numerical, which rely on the pre-computation of many transfers, which are later used to infer transfer costs in the optimization process: numerical approaches may either database-dependent [12], often relying on transfer window constraints, or based on multivariate regression. DL has been particularly successful for the latter [14].

A linear model based on the best time-to-go $\mathcal{T}$ (Eq. 13) is used to estimate the duration of RQ-Law transfers. The model is defined in Eq. 14. $\mathcal{T}$ follows from the definition of the proximity quotient Q, which approximates the best quadratic time-to-go [52].

$$\mathcal{T} = \sqrt{\mathrm{Q}} \tag{13}$$

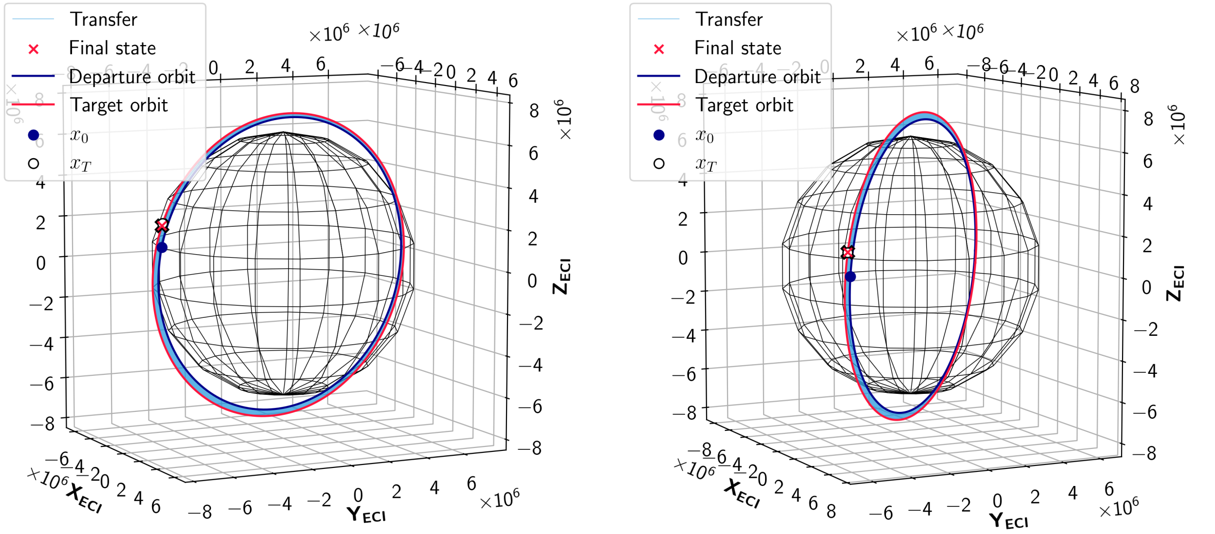
The model in Eq. 14 is obtained by linear regression of measured TOFs using the RQ-Law with respect to predicted TOFs using the best time-to-go. The analysis was done considering all RAAN walk [1] transfers through the Iridium 33, Cosmos 2251 and Fengyun 1-C debris clouds. Both Q-Law and RQ-Law transfers were considered. The clouds



(a)

(b)

Fig. 5 Average transfer in the Iridium 33 debris cloud. (a) Keplerian element history through the transfer. (b) Control history; α and β are the in-ecliptic and out-of-ecliptic thrust angles with respect to $\hat{\mathbf{e}}_\theta$ (Eq. 4) [55].



(a)

(b)

Fig. 6 Extreme 6-element rendezvous transfers in the Iridium 33 and Fengyun 1C debris clouds using the RQ-Law. Origin: cloud centroid. Target: cloud centroid plus 3 times the standard deviation of elements in the cloud. (a) Iridium 33. (b) Fengyun 1C.

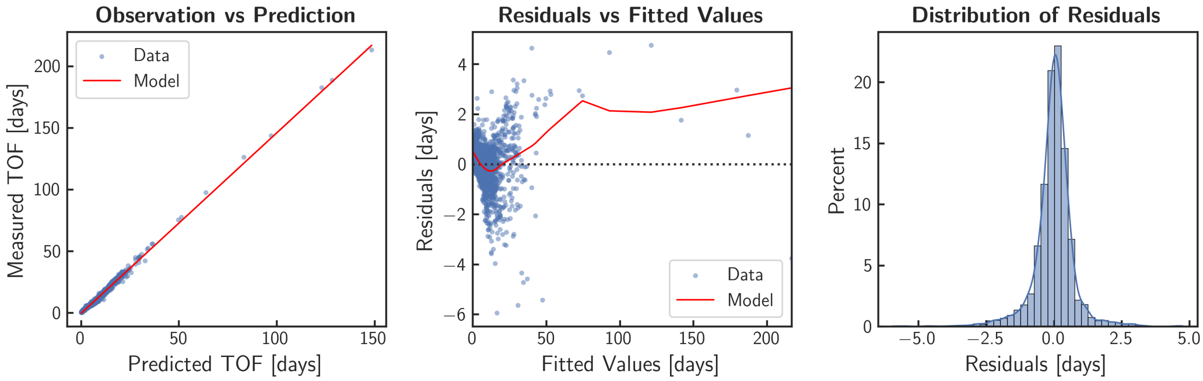


Fig. 7 Estimation of RQ-Law TOFs using the Q proximity quotient. Histogram bin width (right side): 6 hours.

are considered static through the tour to make the target dataset independent of the transfer strategy: static RAAN walk transfers are considered representative of possible transfers in LEO, and so valid for analysis. No instantaneous perturbations nor coasting phases are considered in these transfers, as discussed in Section I.C.3. Fig. 7 shows predicted and observed RQ-Law TOFs and summarizes the performance of the linear model. Table 10 reports the results of the linear regression analysis for both the Q-Law and RQ-Law. A strong positive correlation between the best time-to-go $\mathcal{T}$ and the measured TOF exists for both Q-Law and RQ-Law transfers (Pearson $r > 0.99$), indicating a strong linear relationship [61]. A linear model was fit for each strategy. Outlier TOFs, outliers outside of the 3σ range, were excluded. In both cases the linear model explains over 99% of the variance in observations ($R^2 > 0.99$), demonstrating an excellent fit [62]; the mean estimation error ε is close to 0 as well. Distribution similarity is verified using the Kolmogorov-Smirnov (KS) test [63]. In both cases the KS p-values indicate that there is no statistically significant difference between the estimated and observed TOF distributions.

Table 10 Goodness-of-fit analysis of the TOF models. ε: estimation error.

	Pearson r	Slope	Intercept [h]	R-squared	KS statistic	KS p-value	$\bar{\varepsilon}$ [min]	σ_ε [min]
Q-law	0.9972	1.4329	-15.9	0.995	2.55e-02	0.28	1.5	45.0
RQ-law	0.9970	1.4309	-9.72	0.994	2.31e-02	0.39	1.5	46.5

$$\text{TOF} = \max(1.4309\mathcal{T} + C, \mathcal{T}); \quad C = -9.72 \text{ [hours]} \tag{14}$$

The required fuel mass $m_{f,\text{req}}$ is calculated by multiplying the estimated TOF by the constant fuel mass flow $\dot{m}$ (see Eq. 6). ΔV cost is estimated using Eq. 15, which assumes refuelling takes place when the spacecraft's fuel mass M_f is spent. The same approach is used to calculate the cumulative ΔV cost of complete tours. Critically, this model is suitable for the aforementioned debris clouds, using the specified RQ-Law parameters and tolerances. Application to other cases should follow careful analysis and verification.

$$\Delta V = \begin{cases} v_{\text{eq}} \log\left(\frac{m_0}{m_0 - m_{f,\text{req}}}\right) & m_{f,\text{req}} \leq M_f \\ n\Delta V_{\text{tank}} + v_{\text{eq}} \log\left(\frac{m_0}{m_0 - m_{\text{rem}}}\right), \quad \text{where} \begin{cases} \Delta V_{\text{tank}} & = v_{\text{eq}} \log\left(\frac{m_0}{m_0 - M_f}\right) \\ n & = \left\lfloor \frac{m_{f,\text{req}}}{m_{f,\text{req}}} \right\rfloor \\ m_{\text{rem}} & = m_{f,\text{req}} - nM_f \end{cases} & \text{else.} \end{cases} \tag{15}$$

D. Heuristic Combinatorial Optimization

A modular STSP solver based on population-based heuristic optimization is used to obtain near-optimal solutions. The choice for heuristic optimization is motivated by the widespread use of heuristic optimization methods to solve STSPs in literature [1–4, 6] and the availability of highly performant, open-source heuristic multi-objective optimization libraries such as `pygmo`[§] [64] and `pymoo`[¶] [65], which greatly eases the benchmarking and selection of diverse heuristic optimization algorithms for specific problem variants. The combinatorial optimization component is implemented using `pygmo`, a parallel multi-objective global optimization library based on the Archipelago meta-heuristic [64].

1. Meta-heuristic

Combinatorial optimization meta-heuristics are algorithms that efficiently explore large discrete search spaces to find optimal or near-optimal solutions to combinatorial problems [66]. `pygmo` implements the Archipelago meta-heuristic [64]. The Archipelago algorithm evolves sub-populations, or "islands", concurrently, starting from different initial populations and possibly using different evolutionary strategies. Periodic migrations of individuals between islands promote diversity and prevent premature convergence, improving the algorithm's ability to avoid local optima and thoroughly explore the search space. This parallel and cooperative framework accelerates the optimization process and improves the robustness and quality of the solutions obtained [67].

2. Population Sampling

High quality population sampling is crucial for heuristic global optimization algorithms which are sensitive to initialization conditions [68]. Examples are population-based algorithms and algorithms with memory mechanisms such as Simulated Annealing [69].

Population sampling in the case of CO problems involves sampling permutations $\sigma \in \mathfrak{S}^n$ of length n. Uniform permutation sampling is used to cover the search space as widely as possible. Relevant algorithms are the Fisher-

[§] `https://esa.github.io/pygmo2/`

[¶] `https://pymoo.org/`

Yates Shuffle (FYS) [70], Knuth's algorithm using Sobol points (KS) [68, 71] and Sobol Permutations (SP) [68]. A fourth algorithm is proposed: uniformly sampling the $[0, 1)^n$ hypercube using Sobol points and applying an `argsort` operation to obtain uniformly sampled permutations of $\mathfrak{S}^n$. This algorithm was found to yield more uniform samples than FYS and KS while being orders of magnitude faster than SP.

Notably, advanced approaches and hand-crafted heuristics are often used in CO to determine near-optimal solutions prior to heuristic optimization [1, 12]. Distance-based permutation sampling is used to leverage known approximate solutions, using the Mallows Model [72–74] under the Hamming distance. Empirical results show that a combination of both approaches is best to balance the exploration of the search space and the exploitation of known approximate solutions.

3. Permutation Encoding

Random keys permutation encoding [75] is selected for its versatility for optimizing complex discrete optimization problems across various domains [76, 77]. Permutations are encoded using vectors of continuous values, or random keys, in the $[0, 1)$ range. A permutation $\sigma \in \mathfrak{S}^n$ is encoded by generating a vector of random keys $\mathbf{x} \in [0, 1)^n$, sorting it, and permuting it by σ. Decoding is done by applying an `argsort` operation to $\mathbf{x}$.

4. Approximate Solutions

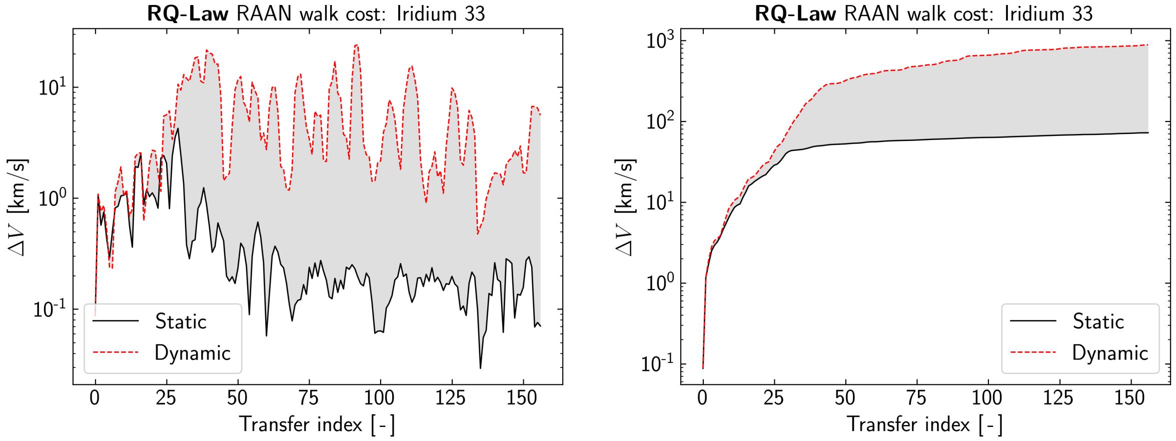
Izzo et al. [1] showed that the optimal solution of the STSP closely resembles a monotonically increasing RAAN walk. This result is intuitive as transfer cost is primarily driven by plane change cost, and plane change cost (Eq. 16) is primarily driven by the RAAN gap that must be closed [1, 4] for orbits with relatively high inclination. Most Earth orbiting spacecraft [78] and all the debris clouds under consideration (Fig. 1).

$$\Delta V_\gamma = 2V_0 \sin\left(\frac{\gamma}{2}\right) \tag{16a}$$

$$\gamma = \arccos(\cos i_1 \cos i_2 + \sin i_1 \sin i_2 [\cos\Omega_1 \cos\Omega_2 + \sin\Omega_1 \sin\Omega_2]) \tag{16b}$$

The RAAN walk holds only for static orbiting targets however, and ADR missions in LEO are an example of a highly dynamic perturbed STSP due to the RAAN drift induced by the J_2 perturbation [1]. Fig. 8 shows the impact of RAAN drift on the cost of the RAAN walk through the Iridium 33 debris cloud: the increase in cost is dramatic. Furthermore, the optimal static and dynamic tours are not related, with their Spearman rank correlation quickly decreasing over time [1], as RAAN drift rates are independent of the original ranking (Eq. 12).

Two tree search approaches and one STSP routing heuristic are considered to obtain approximate solutions. The tree search approaches considered are Beam Search (BS) and Nearest-Neighbor (NN) search [79, 80], both considering



(a) Cost per transfer.

(b) Cumulative cost.

Fig. 8 Impact of RAAN drift on the cost of the RAAN walk through the Iridium 33 debris cloud.

transfer cost using a given transfer strategy, in this case RQ-Law transfers. Fig. 9 shows the cumulative cost of the RAAN walk, NN and BS tours through the Iridium 33 cloud, under RAAN drift. BS is the superior approach, though choosing an optimal beam width is challenging, and the cost of the search quickly grows as beam width is increased. A BS with a beam width of 2 is used to generate approximate solutions.

The STSP routing heuristic considered is the Dynamic RAAN Walk (DRW). The DRW is defined as a greedy search over a nearest-RAAN target ranking policy, which is performed sequentially in the dynamic STSP environment until all targets have been visited. The benefit of using a heuristic over a search is runtime, as can be seen in Table 11. Despite its simplicity the DRW performs surprisingly well, as can be seen in Fig. 9.

Table 11 Time required to generate an approximate solution for the Iridium 33 STSP of 167 transfers.

	RAAN walk	DRW	NN	BS (w = 20)
Runtime	0,2 ms	80,9 ms	2,1 s	79,3 s

5. Heuristic Optimization Algorithm

Optimizer selection is conducted by trading off the performance of all global heuristic optimizers available in `pygmo` (refer to the `pygmo` capabilities page[‖]) on the STSP variant at hand. Critically, optimizers of the family of Evolutionary Strategies depend on internal sampling processes and are thus not sensitive to initial populations: in general terms other optimizers are preferrable if candidate solutions can be leveraged.

The choice of heuristic global optimizer is determined by a trade-off of all `pygmo` global optimizers on a reduced 50-transfer Iridium 33 ADR STSP. A 16-island archipelago is used with 80 individuals per island. 5 evolutionary cycles

[‖] `https://esa.github.io/pygmo2/overview.html`

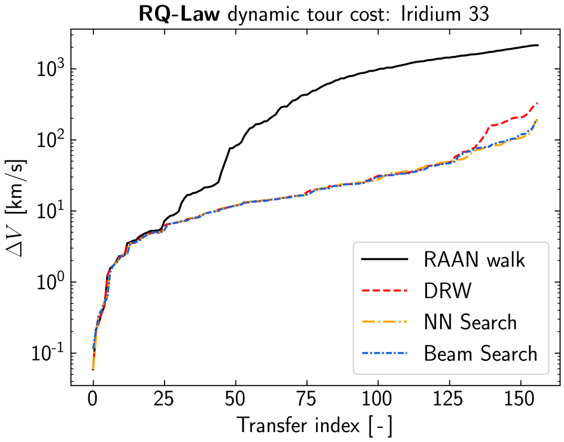


Fig. 9 Cumulative cost of tours traversing the Iridium 33 debris cloud as a function of transfer index, under RAAN drift. BS beam width: 20.

are carried out, each consisting of 500 generations of isolated evolution in each island followed by a migratory process between islands. Two variants of the optimization process are considered: one with uniformly sampled populations over all islands, and another with 8 of the 16 initial populations sampled around approximate BS solutions using the Mallows Model under the Hamming distance, which is defined as the number of positions at which two sequences differ [74, 81]. To ensure the trade-off is robust, the performance of each optimizer is evaluated considering 10 optimizations using different random seeds.

Fig. 10 summarizes the performance of all the considered algorithms. The figure shows, for each optimizer, the performance of the best individual in the algorithm as a function of generation. This is shown for both the optimization starting from a uniformly sampled initial population, and the optimization starting from a population sampled in the proximity of the candidate solution obtained using the BS algorithm defined previously. Observe how the improvement that comes from sampling initial populations around approximate solutions is large across final performance, consistency and convergence speed. The Simple Genetic Algorithm[**] shows both superior performance and consistency than any other option, consistently reaching the best tour found even when starting from a uniformly sampled population.

High quality approximate solutions are very desirable. As the complexity of VRPs increases high quality approximate solutions become more difficult to obtain however. This bodes ill in the field of space logistics, as the complexity of space VRPs is bound to increase as LEO and MEO become more congested, the in-space manufacturing and servicing industries rise, and space logistics operations become more complex [20]. The promise of NCO methods is to automate the process of discovering high quality heuristics for highly complex CO problems [15, 19]. It becomes pressing to ask whether these methods could be applied in the space domain as well.

[**] `https://esa.github.io/pygmo2/.../sga`

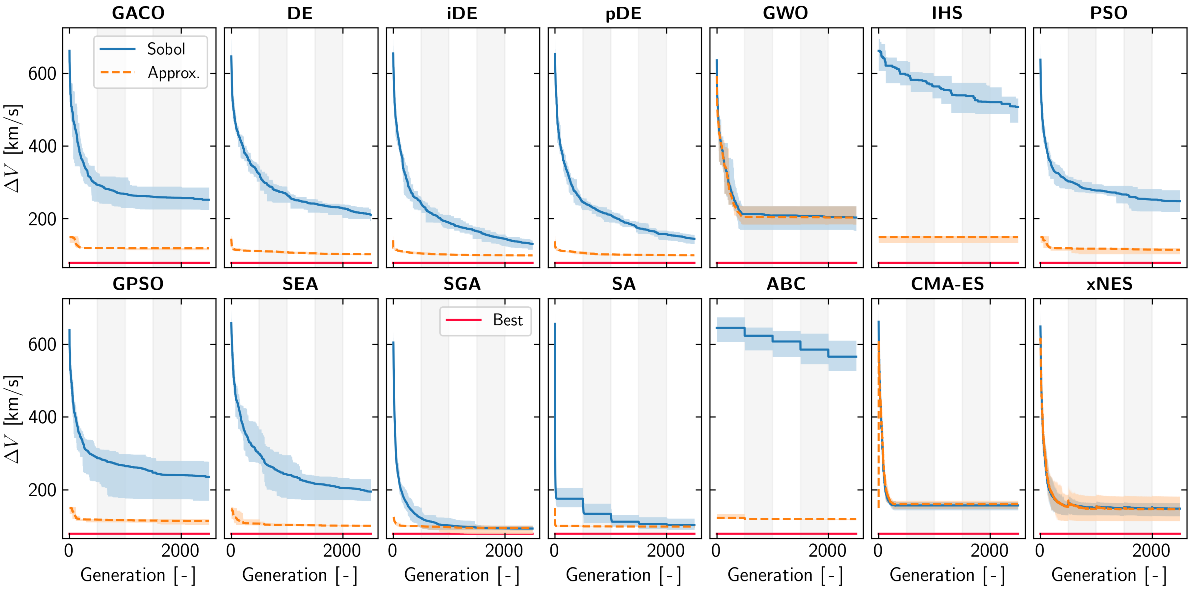


Fig. 10 Global optimization performance summary. Red line: best cost obtained across all cases, achieved with SGA. Grey bands: distinct evolutionary periods, 500 generations each. Blue: uniformly sampled initial populations. Orange: initial populations sampled around approximate solutions. Shaded area: best to worst losses (tour costs) achieved. Solid lines: mean loss.

II. Neural Combinatorial Optimization for Space VRPs

NCO uses DNNs to automate the process of determining heuristics to solve CO problems. RL is the dominant paradigm for NCO, as supervised learning is often unfeasible for large or theoretically hard problems. NCO offers the attractive prospect of alleviating the scaling issues of exact approaches, while removing the need for handcrafted heuristics [19], and has shown promising performance on various CO problems [19], and has been shown to achieve high quality results, especially when coupled with advanced policy search procedures [15].

The RL4CO[††] NCO library is used to implement and train the STSP routing policy. RL4CO is a benchmark library for NCO based on PyTorch [82] with standardized, modular, and highly performant implementations of various environments, policies and RL algorithms, covering the entire NCO pipeline [19].

This section discusses the implementation of an STSP environment for NCO in Section II.A, the NCO routing policy in Section II.B, training algorithms in Section II.D and policy search procedures in Section II.C.

A. Environment

The state of the targets is described using Keplerian elements (which is practical and common practice when analyzing debris clouds [1]). The agent is trained in realistic Active Space Debris removal scenarios. These scenarios are generated using statistical models of real debris clouds. The models are obtained by fitting cloud observations with

[††] https://rl4.co

the parametric statistical models that best match the observations. All 19 parametric statistical models available in PyTorch[‡‡] [82] are considered. Goodness of fit is assessed using the Kolmogorov-Smirnov (KS) test [63]. The result is a composite model of the translational state and other properties of a debris cloud, comprising 6 or more parametric models: one for each Keplerian element, and more for other measurements such as radar-cross section if relevant. Fig. 11 shows the model generated for the Iridium 33 cloud. In this work the environment is limited to the translational state of the cloud.

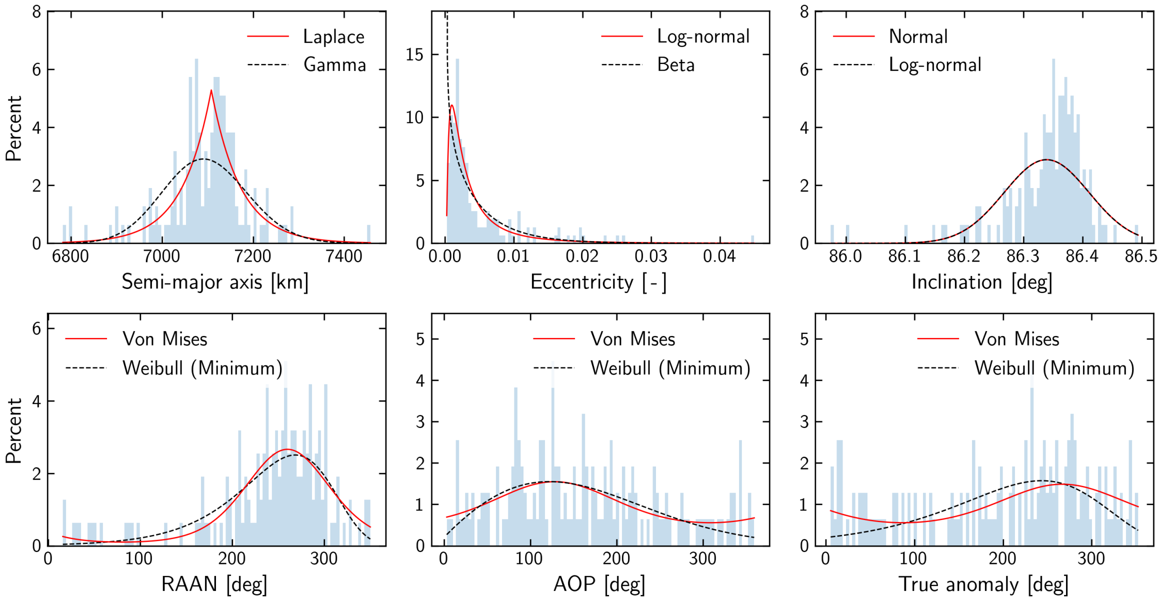


Fig. 11 Statistical model of the translational state of the Iridium 33 debris cloud. Histograms: observations. Curve, red: parametric model that best matches the observations, where goodness of fit is measured using the KS statistic. Curve, black: second-best parametric model.

The range of values which may be sampled by each model is limited to the range of observed values. This is achieved with inverse transform sampling [83] when the Inverse Cumulative Distribution Function (ICDF) of the parametric model is defined and implemented in PyTorch, and with vectorized rejection sampling [84] if the ICDF is not available.

The global state is propagated through the sequential decision-making process. This is done by estimating the transfer times, and using a secular perturbation model to propagate the state of the cloud. The Q-based TOF estimation method defined in Eq. 14 is used to estimate transfer times. The secular perturbation model used for LEO multi-rendezvous missions is the J_2 secular perturbation model defined in Eq. 12.

[‡‡] `https://pytorch.org/docs/stable/distributions.html`

B. Policy

An autoregressive Attention-based policy[§§] is used to learn the routing problem. First introduced by Kool et al. [85], the policy encodes the input graph using a Graph Attention Network (GAT), and decodes the solution using a Pointer Network. Kool et al. train this policy using the REINFORCE RL algorithm to achieve considerably better performence than other learned heuristics [85]. Fraçois et al. find this policy to be a highly efficient learning component in their comprehensive analysis of learning performance in NCO methods [15].

C. Policy Search

Policy search strategies determine how actions are selected based on the learned policy, balancing exploration and exploitation to find optimal or near-optimal solutions. Policy search is fundamental to improve the performance of NCO algorithms [15]. Three policy search strategies are considered: greedy search, sampling search and beam search.

1. Greedy Search

Greedy search is the simplest policy search strategy, where at each decision step, the action with the highest probability (or highest estimated reward) is selected deterministically. This approach is computationally efficient but may lead to suboptimal solutions due to its myopic nature, potentially getting trapped in local optima.

2. Stochastic Search

Stochastic policy search, also known as sampling search, introduces randomness in the policy search by selecting actions based on the probability distribution defined by the learned policy. Instead of always choosing the most probable action, actions are sampled according to their probabilities, allowing for greater exploration of the action space. This strategy can escape local optima and explore diverse solution paths but may require more computational resources and time to converge to high-quality solutions.

3. Beam Search

Beam search (BS) strikes a balance between greedy and sampling strategies by maintaining a fixed number of the most promising partial solutions (beams) at each step. At each decision point, all possible extensions of the current beams are considered, and the top k beams with the highest cumulative probabilities are retained for the next step. This approach enhances exploration while controlling computational complexity, leading to better solution quality compared to greedy search without incurring the full cost of exhaustive search.

[§§] `https://rl4.co/docs/.../AttentionModelPolicy`

D. Reinforcement Learning Algorithms

Three RL algorithms are considered to train the routing policy: REINFORCE, Advantage Actor-Critic, and Proximal Policy Optimization. This section consists of a brief introduction of each method, followed by an RL algorithm trade-off based on the training cost and policy performance achieved in a 10-transfer ADR STSP scenario based on the Iridium 33 debris cloud.

1. Overview

Stochastic Policy Gradient The REINFORCE algorithm, introduced by Williams [86], is a stochastic policy gradient method that uses Monte Carlo sampling to compute an unbiased estimate of the policy gradient. Full trajectories are sampled, and the return for each trajectory is used to update policy parameters via gradient ascent to maximize expected reward. REINFORCE suffers from high variance in gradient estimates, slowing convergence.

Advantage Actor-Critic Advantage Actor-Critic (A2C), introduced by Konda and Tsitsiklis [87] and extended by Mnih et al. [88], reduces variance in policy gradient estimates by incorporating a baseline. The baseline, given by the value function estimated by a critic, is subtracted from the return to compute an advantage, stabilizing policy updates. A2C concurrently optimizes the actor (policy) and critic (value function) in a single process, improving learning efficiency.

Proximal Policy Optimization for Autoregressive Policies Proximal Policy Optimization (PPO), proposed by Schulman et al. [89], addresses the instability of A2C by introducing a clipped objective function that limits the magnitude of policy updates. This improves the stability of training, making PPO widely applicable in reinforcement learning tasks.

The variant of PPO by Kool et al. [90], used here, modifies PPO for autoregressive policies, which generate solutions sequentially, where each action depends on prior actions. Kool's variant treats the entire autoregressive process as a single decision step, reducing the complexity of the Markov Decision Process (MDP). While effective in capturing sequential dependencies, this approach can introduce approximation bias and reduce gradient information due to its single-step treatment of the decoding process [90].

2. Algorithm Selection

The three RL algorithms under consideration were traded-off on a 10-transfer scenario. Training was repeated 5 times using different random seeds to ensure the trade-off was robust. To establish an intuitive scale for policy performance, the validation dataset was extracted and optimized using the HCO module. Model validation performance is expressed in terms of optimality gaps with respect to the cost of the tours optimized with HCO. Table 12 lists the most relevant training settings and policy architecture hyperparameters. With the exception of of the embedding size of

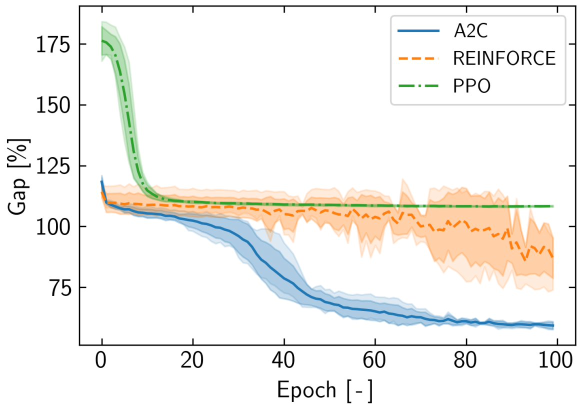


Fig. 12 Learning curve for each algorithm, represented by the validation reward. Shaded areas: 1 standard deviation range, min-max range.

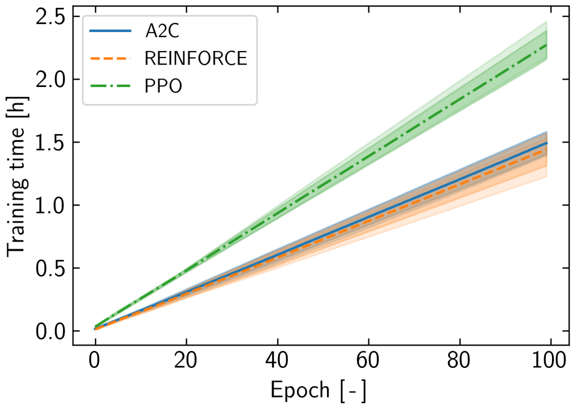


Fig. 13 Training time for each algorithm. Shaded areas: 1 standard deviation range, min-max range.

Table 12 Training and policy architecture settings used to compare RL algorithm performance.

Training dataset	Batch size	Optimizer	Learning rate	Epochs	Embedding size	Encoder layers	Att. Heads
1M scenarios	32768	Adam	1,00E-04	100	256	3	8

256, all RL algorithm and policy architecture hyperparameters used at this stage were the defaults in `RL4CO`. Training was conducted for 100 epochs on NVIDIA L40 GPU systems (16 vCPUs, 250GB RAM). The training runs of each algorithm were conducted simultaneously on different machines to avoid polluting training time measurements.

The experimental results, summarized in Table 13 and illustrated in Fig. 12 and Fig. 13, indicate that A2C consistently outperforms REINFORCE and the modified PPO across multiple metrics. Specifically, A2C achieves the lowest mean ΔV and optimality gap, with values of 110. km/s and 29.0%, respectively, using the beam search policy search strategy. In contrast, REINFORCE and PPO exhibit higher mean ΔV values of 206.4 km/s and 169.4 km/s, and optimality gaps of 156.7% and 110.6%, respectively.

Training times for REINFORCE and A2C are comparable, averaging around 1.5 hours, while PPO requires significantly more time at approximately 2.27 hours. Evaluation times remain similar across all algorithms, with minor variations attributable to the search strategies employed.

These observations suggest that A2C not only produces better-performing policies but also does so with training efficiency similar to REINFORCE and superior to PPO. The lower standard deviations in ΔV and optimality gap for A2C indicate more stable and reliable policy learning.

Despite PPO's effectiveness in various reinforcement learning tasks [89], its modified version for autoregressive policies does not demonstrate the expected performance gains in this context. The reduced performance of PPO may stem from approximation biases introduced by treating the entire decoding process as a single-step MDP and challenges in entropy estimation, as discussed in Section II.D.1. Additionally, the loss of detailed gradient information due to the

Table 13 RL training performance using REINFORCE, A2C and PPO. Bold: best result. Multiple results highlighted if best cannot be decided considering mean and observed variance.

		REINFORCE			**A2C**			**PPO**		
Search strategy		Greedy	Stoch.	BS	Greedy	Stoch.	BS	Greedy	Stoch.	BS
ΔV [km/s]	μ	206,4	212,7	172,6	129,6	130,1	**111,0**	169,4	178,4	154,6
	σ	39,7	39,7	29,6	28,1	28,4	19,7	36,6	36,4	30,0
Optimality gap [%]	μ	156,7	164,6	114,7	61,3	61,9	**38,3**	110,6	121,7	92,1
	σ	56,0	57,2	43,1	39,1	39,2	29,0	50,28	50,1	41,6
Training time [h]	μ	-	**1,44**	-	-	**1,49**	-	-	2,27	-
	σ	-	0,13	-	-	0,09	-	-	0,11	-
Evaluation time [ms]	μ	**36,1**	43,7	86,5	**37,9**	43,8	87,1	**35,3**	43,5	86,6
	σ	4,5	11,1	6,5	7,42	11,7	6,58	4,8	12,9	11,0

simplified MDP formulation could hinder effective policy updates.

Based on these results, A2C is selected as the reinforcement learning algorithm for training the routing policy in the ADR STSP problem. Its superior performance and efficient training make it the most suitable choice among the algorithms evaluated.

E. Hyperparameter Impact Determination with ANOVA

Optimizing the performance of neural combinatorial optimization (NCO) models involves tuning a multitude of hyperparameters. Determining the statistical relevance of these hyperparameters is essential to identify which ones significantly influence model performance and to prioritize them for further optimization [91]. Analysis of Variance (ANOVA) serves as a robust statistical procedure to assess the significance of multiple factors simultaneously [92].

Orthogonal arrays, particularly Taguchi factorial designs, facilitate efficient experimentation by systematically varying hyperparameters across predefined levels while minimizing the number of required experimental runs [93–95]. The Taguchi L27 orthogonal array, also known as L_{27}-A$3^{13\text{-}10}$ fractional factorial design [63], is specifically designed to evaluate up to 13 factors at three levels, making it suitable for comprehensive hyperparameter analysis with limited resources [63]. Table 14 presents the Taguchi L27 orthogonal array utilized in this study.

A linear ANOVA was conducted using the `statsmodels` library [96] to evaluate the main effects of 13 hyperparameters on the model's performance. The 13 chosen hyperparameters, their function, and the rationale for choosing them can be seen in Table 15. The choice of a linear model is justified by the initial focus on identifying straightforward, additive relationships between hyperparameters and performance metrics, with the goal of identifying the most impactful hyperparameters for further optimization.

Table 16 presents the ANOVA results for the linear effects of each hyperparameter. The Sum of Squares (Sum Sq), degrees of freedom (df), F-statistic (F), and p-values (PR(>F)) are reported for each factor. ANOVA relies on two fundamental assumptions: normality of residuals, and homogeneity of variances across all factor levels —the property

Table 14 Taguchi L27 orthogonal array, also known as L_{27}-A$3^{13\text{-}10}$ fractional factorial design [63].

Run	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27
X1	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	2	2	2	2	2	2	2	2	2
X2	0	0	0	1	1	1	2	2	2	0	0	0	1	1	1	2	2	2	0	0	0	1	1	1	2	2	1
X3	0	0	0	1	1	1	2	2	2	1	1	1	0	0	0	1	1	1	2	2	2	0	0	0	1	1	2
X4	0	0	0	1	1	1	2	2	2	2	2	2	0	0	0	1	1	1	0	0	0	2	2	2	1	1	1
X5	0	1	2	0	1	2	0	1	2	0	1	2	0	1	2	0	1	2	0	1	2	0	1	2	0	1	0
X6	0	1	2	0	1	2	0	1	2	1	2	0	2	0	1	1	0	0	2	0	1	2	0	1	0	1	0
X7	0	1	2	0	1	2	0	1	2	2	0	1	1	0	0	2	1	1	1	2	0	2	1	0	2	0	1
X8	0	1	2	1	2	0	2	1	0	0	1	2	0	1	2	1	0	2	2	1	2	1	2	1	2	1	2
X9	0	1	2	1	2	0	2	1	0	1	2	0	2	1	0	2	1	0	0	1	2	1	2	1	1	2	2
X10	0	1	2	1	2	0	1	2	0	3	0	1	1	2	0	2	2	1	1	2	0	2	1	0	1	0	1
X11	0	1	2	2	0	1	1	2	0	1	2	0	2	0	1	1	0	1	1	0	2	2	1	3	1	2	1
X12	0	1	2	2	0	1	2	3	0	2	1	2	1	3	0	0	1	2	3	2	1	1	3	1	2	1	2
X13	0	1	2	2	0	1	2	3	0	3	2	1	0	1	2	3	2	1	1	3	2	1	2	1	3	2	1

known as homoscedasticity [63, 92]. Residual normality was verified by the Shapiro-Wilk test [63, 97] ($p = 0.57$) and the Anderson-Darling test [63] ($p = 0.27$). Homoscedasticity was verified by visual inspection of the residuals. Furthermore as will be seen, the impacts estimated by the ANOVA match those expected from domain expertise. The ANOVA is thus considered valid to diagnose the main effects of the 13 hyperparameters considered on policy performance.

The ANOVA results indicate that embedding dimension and number of encoder layers are the only hyperparameters with statistically significant effects on model performance, with p-values of 0,005776 and 0,048680, respectively. embedding dimension exhibits the highest Sum of Squares (3,46E+09), suggesting it has the most substantial impact. number of encoder layers follows with a Sum of Squares of 1,50E+09, indicating a meaningful but slightly lesser influence. All other hyperparameters do not show significant linear effects, as their p-values exceed the conventional threshold of $0,05$. The linear model accounts for approximately 69,87% of the total observed variance ($R^2 = 69,87\%$). A considerable portion of variability (30.13%) is not explained by the linear effects of the analyzed hyperparameters, indicating the presence of higher order effects which are not modelled.

The significant effect of embedding dimension suggests that increasing this hyperparameter enhances the model's capacity to capture and represent input features effectively, thereby improving performance. The significance of the number of encoder layers implies that adding more layers may contribute to deeper feature extraction and more complex graph representations, although its impact is less pronounced compared to embedding dimension. The non-significant effects of other hyperparameters indicate that, within the tested ranges, their individual linear contributions to model performance are minimal. However, it is possible that these hyperparameters may interact with each other or exhibit non-linear relationships that are not captured in the current linear ANOVA model. The R^2 value of 69,87% indicates a moderately strong fit. While the linear model explains a substantial portion of the variance, there remains considerable unexplained variability, potentially due to unmodeled factors or complex relationships among hyperparameters.

The ANOVA analysis identifies embedding dimension as the most statistically significant hyperparameter influenc-

Table 15 Policy architecture and training hyperparameters considered in the 3-level ANOVA.

Hyperparameter	Component	Function	Rationale	Levels
Embedding Dimension	Embedding Layer	Size of node embeddings representing input features.	Influences the capacity to capture feature representations.	• 64 • 128 • 256
Number of Encoder Layers	GAT	Number of layers in the GAT, affecting depth and representation learning.	Determines the depth of feature extraction and complexity of graph representations.	• 2 • 3 • 4
Number of Attention Heads	GAT	Number of parallel attention mechanisms per GAT layer.	Enhances the model's ability to focus on different parts of the graph simultaneously.	• 4 • 8 • 16
Feedforward Hidden Size	GAT	Size of the hidden layer in GAT's feedforward network.	Affects the model's capacity and computational complexity.	• 256 • 512 • 1024
Dropout Rate	GAT	Probability of dropping units during training to prevent overfitting.	Helps in regularizing the model and improving generalization.	• 0.1 • 0.3 • 0.5
Temperature	PN	Scales logits before softmax to control randomness in action selection.	Balances exploration and exploitation during policy generation.	• 0.5 • 1.0 • 2.0
Tanh Clipping	PN	Limits the output of the tanh activation to prevent extreme values.	Ensures numerical stability by preventing large activation values.	• 0 • 10 • 20
Actor Learning Rate	Actor Optimizer	Learning rate for the actor (policy) network optimizer (Adam).	Influences the speed and stability of policy updates.	• 1e-5 • 1e-4 • 1e-3
Weight Decay	Actor Optimizer	Regularization parameter to prevent overfitting by penalizing large weights.	Controls the model's generalization and prevents overfitting.	• 0 • 1e-4 • 1e-3
Gradient Clipping Value	Actor Optimizer	Maximum allowed value for gradients during backpropagation to prevent exploding gradients.	Ensures training stability by avoiding excessively large gradients.	• 0.5 • 1.0 • 2.0
Critic Learning Rate	Critic Optimizer	Learning rate for the critic network optimizer (Adam).	Affects the stability and speed of value estimation updates.	• 1e-5 • 1e-4 • 1e-3
Reward Scaling	Baseline	Scales the reward signal to stabilize training and improve gradient estimates.	Enhances training stability by normalizing reward magnitudes.	• 1 • 10 • 100
Critic Hidden Dimension	Critic Network	Size of the hidden layers within the critic network.	Influences the critic's capacity to accurately estimate value functions.	• 128 • 256 • 512

Table 16 ANOVA results. Dashed line separates statistically significant factors ($p < 0.05$) from non-significant factors.

Hyperparameter	Sum Sq	df	F	PR(>F)	R^2
Embedding Dimension	3,46E+09	1	10,9	**0,005776**	25,20%
Number of Encoder Layers	1,50E+09	1	4,73	**0,04868**	10,96%
Feedforward Hidden Size	7,82E+08	1	2,46	0,140893	5,70%
Weight Decay	7,28E+08	1	2,29	0,154256	5,30%
Number of Attention Heads	7,03E+08	1	2,21	0,160724	5,13%
Actor Learning Rate	6,66E+08	1	2,1	0,171374	4,86%
Critic Hidden Dimension	6,18E+08	1	1,94	0,18656	4,51%
Gradient Clipping Value	5,55E+08	1	1,75	0,209273	4,04%
Dropout Rate	3,24E+08	1	1,02	0,331401	2,36%
Tanh Clipping	2,11E+08	1	0,66	0,429799	1,54%
Critic Learning Rate	3,58E+07	1	0,11	0,742551	0,26%
Temperature	1,58E+06	1	0,00	0,944904	0,01%
Reward Scaling	4,52E+05	1	0,00	0,970506	0,00%
ANOVA model	9,58E+09	13	-	-	**69,87%**
Residuals	4,13E+09	13	-	-	**30,13%**

ing model performance, followed by a marginal effect from the number of encoder layers. These findings suggest that increasing the embedding dimension is desirable to enhance feature representation capabilities, and that optimizing the number of encoder layers can potentially improve the depth and quality of feature extraction, albeit with a less substantial impact. Embedding dimension and number of encoder layers were chosen for hyperparameter optimization, discussed next.

F. Hyperparameter Optimization

A full factorial (grid search) approach was conducted to optimize the embedding dimension and number of encoder layers, considering four levels. The levels considered for each hyperparameter are those previously presented in Table 15. Fig. 16 shows the optimality gaps obtained by each of the models in the grid search, using BS to decode the policies. No statistically significant improvement is observed in comparison with the RL algorithm trade-off results in Table 13. Fig. 15 shows the combined learning curve all grid search runs. Fig. 14 shows the Pareto front of the grid search. An interesting feature in Fig. 16 is the underperformance of models in the diagonal. The amount of trainable parameters of the models considered can be seen in Fig. 17.

A second ANOVA was performed on the results of the grid search. This time quadratic and interaction effects were included in the ANOVA model. The results follow in Table 17. The Shapiro-Wilk test ($p = 0.9004$) and Anderson-Darling test ($p = 0.153$) again confirm that the residuals are normally distributed, and homoscedasticity is visually verified, confirming the validity of the analysis. The ANOVA model accounts for 97.97% of the observed variance,

Table 17 Results of the full factorial ANOVA of embedding dimension (ED) and number of encoder layers (NL). The ANOVA model includes linear, quadratic and interaction terms.

Term	Sum Sq	df	F	PR(>F)	R^2
ED	1,57E+09	1.0	97,805333	**0,002199**	66,30%
NL	7,73E+07	1.0	4,831624	0,115377	3,28%
ED^2	4,72E+08	1.0	29,455497	**0,012275**	19,97%
NL^2	3,77E+07	1.0	2,352144	0,222656	1,59%
ED*NL	1,61E+08	1.0	10,067658	**0,050366**	6,82%
ANOVA model	2,31E+09	3.0	-	-	97,97%
Residual	4,80E+07	3.0	-	-	2,03%

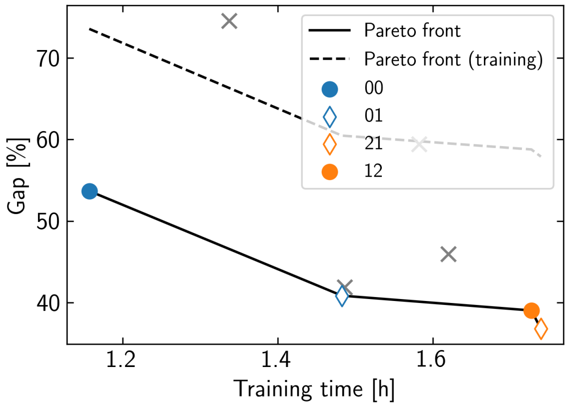


Fig. 14 Pareto front of the full factorial design, using beam search to decode the learned policies.

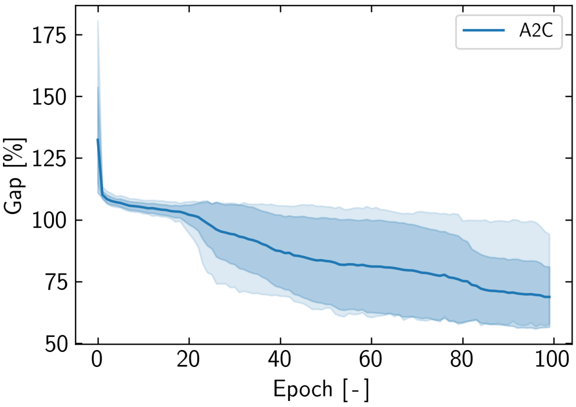


Fig. 15 Grid search learning curve. During training the policy is decoded using greedy search.

indicating a strong fit. The full factorial ANOVA confirms that both embedding dimension and number of layers have significant effects on performance. Embedding dimension shows a strong main effect ($F = 97.81$, $p = 0.0022$) and a significant quadratic term ($F = 29.46$, $p = 0.0123$). The interaction between embedding dimension and number of layers is marginally significant ($F = 10.07$, $p = 0.0504$), indicating that their combined influence affects performance. These findings confirm the significant linear effects identified by the L27 fractional ANOVA and indicate the presence of additional non-linear relationships.

The results indicate that network architecture is the principal driver of model performance, but model performance increases asymptotically slowly with the number of parameters.

G. Impact of Training Dataset Size on Policy Performance

The amount of data provided to the model for training is key for its performance [19]. This is especially the case for attention-based ML models [98–100]. Fig. 18 shows the effect of increasing the size of the training dataset from 1M to 3M tours of 10 transfers. Observe the large increase in convergence speed. The final performance improves as well. Notwithstanding, model performance eventually plateaus to a validation optimality gap in training of approximately

Fig. 16 Optimality gaps obtained for each run in the grid search, using beam search to decode the policy.

		EL		
		2	3	4
EB	128	53,68%	40,85%	45,95%
	256	74,54%	59,46%	39,03%
	512	41,87%	36,79%	66,04%

Fig. 17 Number of trainable parameters for each configuration considered in the grid search.

		EL		
		2	3	4
ED	128	**0.51M**	×1,4	×1,8
	256	×3,0	×**4,0**	×5,0
	512	×9,8	×12,8	×**15,9**

50%. This confirms that the architecture of the ML model is the factor limiting further learning.

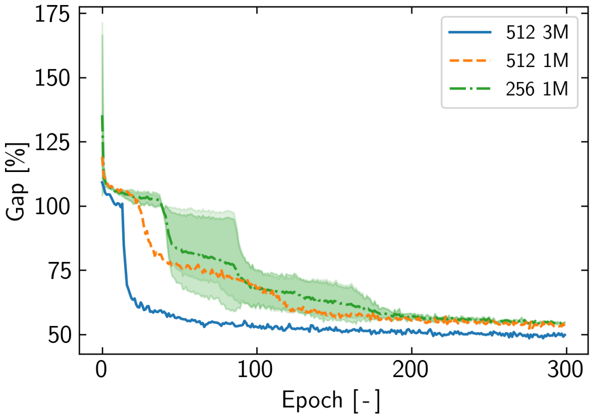


Fig. 18 Final learning curves. Note the impact of increasing training dataset size from 1M to 3M. Notwithstanding, the capacity of the model to learn asymptotically approaches a similar limit at 50% optimality gap.

H. Final Performance and Generalization to Larger Routing Problems

The final policy was trained on 3 million tours consisting of 10 transfers. To evaluate the capacity of the policy to generalize to larger problems, the trained policy was employed to plan scenarios with 10, 30, and 50 transfers. BS was used to search the trained policy. Table 18 presents the final performance results of the trained NCO policy compared to the DRW heuristic across scenarios with 10, 30, and 50 transfers. Each case consisted in the planning of 1000 missions. The metrics include the mean and standard deviation of ΔV (change in velocity), the optimality gap percentage, and the evaluation time in milliseconds. Optimality gaps are obtained by comparison with the solution obtained using HCO.

The trained NCO policy exhibits better performance in the 10-node scenario than the DRW heuristic. However, as the number of nodes increases to 30 and 50, the performance of the NCO policy greatly deteriorates. This indicates a limitation in the policy's ability to generalize to mission scenarios with a higher number of transfers: the learned policy is not generally applicable for the design of ADR missions.

In terms of computational efficiency, the NCO policy outperforms DRW across the board. The difference is large for small mission scenarios, but becomes less significant for 30 and 50 transfer scenarios.

Table 18 Policy performance compared to the DRW STSP heuristic for STSPs with 10, 30 and 50 targets. Performance measured on test datasets of 1000 missions.

Number of nodes		**10**		**30**		**50**	
Heuristic		**AM BS**	**DRW**	**AM BS**	**DRW**	**AM BS**	**DRW**
ΔV [km/s]	μ	**106,4**	122,1	573,5	**208,1**	1041,8	**261,2**
	σ	16,2	31,4	83,9	46,2	68,2	52,6
Optimality gap [%]	μ	**32,6**	50,5	616,0	**50,4**	555,4	**63,9**
	σ	25,5	36,8	132,7	37,2	97,4	38,3
Evaluation time [ms]	μ	**99,2**	2494,9	**3592,0**	8534,5	**8630,0**	16147,1

III. Conclusion

This study evaluated the applicability and effectiveness of Neural Combinatorial Optimization (NCO) methods for space Vehicle Routing Problems (VRPs), focusing on the Active Debris Removal (ADR) Space Traveling Salesman Problem (STSP) using the Iridium 33 debris cloud as a case study.

A statistical model of the Iridium 33 space debris cloud was created to enable the generation of millions of realistic ADR scenarios for the training of NCO policies. An electric propulsion ADR spacecraft concept is used in this work. Lyapunov Feedback Control was used to generate low-thrust trajectories. In particular, the Rendezvous Q-Law (RQ-Law) Lyapunov Feedback Control low thrust guidance policy is used to 6-element rendezvous transfer trajectories between targets. An efficient transfer cost estimator based on the best quadratic time to go was designed and verified. Finally, a generalized STSP environment model was implemented considering the secular $J2$ perturbation on Right Ascension of the Ascending Node (RAAN) and Argument Of Perigee (AOP).

An Attention-based routing policy, integrating a Graph Attention Network (GAT) and a Pointer Network (PN), was implemented and trained using Reinforcement Learning (RL) algorithms, specifically REINFORCE, Advantage Actor-Critic (A2C), and Proximal Policy Optimization (PPO). Among these, A2C demonstrated the best performance. Hyperparameter analysis through ANOVA identified embedding dimension and the number of encoder layers as critical factors influencing model efficacy. Subsequent grid search hyperparameter optimization revealed that while increasing model complexity marginally enhances performance, larger training datasets significantly improve convergence speed with minimal gains in final performance, converging to a 50% optimality gap under greedy policy search.

The trained NCO policy achieved a mean optimality gap of 32% over 1000 missions with 10 transfers, outperforming Heuristic Combinatorial Optimization (HCO) methods and the Dynamic RAAN Walk (DRW) heuristic in both mission cost and runtime. This indicates that NCO methods are effective for ADR missions with a limited number of targets, offering efficient and viable routing solutions. However, the policy's performance declined in more complex scenarios involving 30 and 50 transfers, highlighting limitations in generalizing beyond the conditions encountered during training.

These findings underscore the potential of NCO methods in optimizing ADR missions but also emphasize the need for advancements to address scalability and generalization challenges. Future research should focus on refining NCO model architectures to better handle dynamic VRPs, and to achieve generalization to unseen numbers of targets. The development of RL training procedures exposing policies to problems with variable numbers of nodes while training should be a priority for future research. Alternative machine learning approaches, such as Deep Reinforcement Learning algorithms that incorporate tree search strategies like Monte Carlo Tree Search (MCTS), are an attractive option to improve the performance and scalability of policies. Leveraging data augmentation and transfer learning techniques may further improve policy robustness and effectiveness in larger-scale missions.

In conclusion, this research demonstrates that NCO methods hold promise for learning effective and efficient routing policies for space VRPs, particularly in ADR scenarios with a constrained number of targets. However, to realize their full potential, further development of policy models is required. The integration of NCO with established optimization techniques offers a viable pathway for enhancing mission planning capabilities, paving the way for more sophisticated and scalable solutions for mission planning and autonomy in space logistics.

Acknowledgments

This research would not have been possible without the support and the guidance of Marc Naeije from the Delft University of Technology and Jesús Ramírez from SENER Aerospace & Defence. Additionally, the thoughtful conversations with Lucrezia Marcovaldi, Álex Cuenca and David Bermejo at SENER Aerospace & Defence greatly improved the final result of this work. Lastly, the author would like to extend the deepest gratitude to their family, and to all those that were there all along. Victoria Catalán, Román Chiva, Francesco Branca, Alejandro Montoya and Martin Camús, Silvio Peressini, Neil Richez, and you —you know who you are—.

References

[1] Izzo, D., Getzner, I., Hennes, D., and Simões, L., "Evolving Solutions to TSP Variants for Active Space Debris Removal: Genetic and Evolutionary Computation Conference (GECCO)," *Proceedings of the 17th annual conference on Genetic and evolutionary computation (GECCO 2015)*, 2015, pp. 1207–1214. https://doi.org/10.1145/2739480.2754727, publisher: ACM Press.

[2] Ricciardi, L., and Vasile, M., *Solving multi-objective dynamic travelling salesman problems by relaxation*, 2019. https://doi.org/10.1145/3319619.3326837, pages: 2007.

[3] Federici, L., Zavoli, A., and Colasurdo, G., "Evolutionary Optimization of Multirendezvous Impulsive Trajectories," *International Journal of Aerospace Engineering*, Vol. 2021, 2021, pp. 1–19. https://doi.org/10.1155/2021/9921555.

[4] Medioni, L., Gary, Y., Monclin, M., Oosterhof, C., Pierre, G., Semblanet, T., Comte, P., and Nocentini, K., "Trajectory optimization for multi-target Active Debris Removal missions," *Advances in Space Research*, Vol. 72, No. 7, 2023, pp. 2801–2823. https://doi.org/10.1016/j.asr.2022.12.013, URL https://www.sciencedirect.com/science/article/pii/S027311772201105X.

[5] Barea, A., Urrutxua, H., and Cadarso, L., "Large-scale object selection and trajectory planning for multi-target space debris removal missions," *Acta Astronautica*, Vol. 170, 2020, pp. 289–301. https://doi.org/10.1016/j.actaastro.2020.01.032, URL https://www.sciencedirect.com/science/article/pii/S0094576520300436.

[6] Narayanaswamy, S., Wu, B., Ludivig, P., Soboczenski, F., Venkataramani, K., and Damaren, C. J., "Low-thrust rendezvous trajectory generation for multi-target active space debris removal using the RQ-Law," *Advances in Space Research*, Vol. 71, No. 10, 2023, pp. 4276–4287. https://doi.org/10.1016/j.asr.2022.12.049, URL https://www.sciencedirect.com/science/article/pii/S0273117722011656.

[7] Mark, C. P., and Kamath, S., "Review of Active Space Debris Removal Methods," *Space Policy*, Vol. 47, 2019, pp. 194–206. https://doi.org/10.1016/j.spacepol.2018.12.005, URL https://www.sciencedirect.com/science/article/pii/S0265964618300110.

[8] Bonnal, C., Ruault, J.-M., and Desjean, M.-C., "Active debris removal: Recent progress and current trends," *Acta Astronautica*, Vol. 85, 2013, pp. 51–60. https://doi.org/10.1016/j.actaastro.2012.11.009, URL https://www.sciencedirect.com/science/article/pii/S0094576512004602.

[9] Sellmaier, F., Boge, T., Spurmann, J., Gully, S., Rupp, T., and Huber, F., "On-Orbit Servicing Missions: Challenges and Solutions for Spacecraft Operations," *SpaceOps 2010 Conference*, American Institute of Aeronautics and Astronautics, Huntsville, Alabama, 2010. https://doi.org/10.2514/6.2010-2159, URL https://arc.aiaa.org/doi/10.2514/6.2010-2159.

[10] Sarton du Jonchay, T., Chen, H., Isaji, M., Shimane, Y., and Ho, K., "On-Orbit Servicing Optimization Framework with High- and Low-Thrust Propulsion Tradeoff," *Journal of Spacecraft and Rockets*, Vol. 59, No. 1, 2022, pp. 33–48. https://doi.org/10.2514/1.A35094, URL https://doi.org/10.2514/1.A35094, publisher: American Institute of Aeronautics and Astronautics _eprint: https://doi.org/10.2514/1.A35094.

[11] Sorenson, S. E., and Pinkley, S. G. N., "Multi-orbit routing and scheduling of refuellable on-orbit servicing space robots," *Computers & Industrial Engineering*, Vol. 176, 2023, p. 108852. https://doi.org/10.1016/j.cie.2022.108852, URL https://www.sciencedirect.com/science/article/pii/S0360835222008403.

[12] Petropoulos, A., Grebow, D., Jones, D., Lantoine, G., Nicholas, A., Roa, J., Senent, J., Stuart, J., Arora, N., Pavlak, T., Lam, T., McElrath, T., Roncoli, R., Garza, D., Bradley, N., Landau, D., Tarzi, Z., Laipert, F., Bonfiglio, E., and Sims, J., "GTOC9: Methods and Results from the Jet Propulsion Laboratory Team," 2017.

[13] Lu, S., Zhang, Y., Zhang, J., Cai, H., Qi, R., Li, X., Qi, Y., Xiao, Q., Shen, A., Zhang, T., Zhang, K., Ye, J., and Tian, Z., "GTOC 11: Results found at Beijing Institute of Technology and China Academy of Space Technology," *Acta Astronautica*, Vol. 202, 2023, pp. 876–888. https://doi.org/10.1016/j.actaastro.2022.07.039, URL https://www.sciencedirect.com/science/article/pii/S0094576522003885.

[14] Li, H., Chen, S., Izzo, D., and Baoyin, H., "Deep networks as approximators of optimal low-thrust and multi-impulse cost

in multitarget missions," *Acta Astronautica*, Vol. 166, 2020, pp. 469–481. https://doi.org/10.1016/j.actaastro.2019.09.023, URL https://ui.adsabs.harvard.edu/abs/2020AcAau.166..469L, aDS Bibcode: 2020AcAau.166..469L.

[15] Francois, A., Cappart, Q., and Rousseau, L.-M., "How to Evaluate Machine Learning Approaches for Combinatorial Optimization: Application to the Travelling Salesman Problem," 2019. URL https://arxiv.org/abs/1909.13121.

[16] Russel, S., and Norvig, P., *Artificial Intelligence: A Modern Approach*, 4th ed., Vol. 175, 2020.

[17] Cormen, T. H., Leiserson, C. E., Rivest, R. L., and Stein, C., *Introduction to Algorithms*, 4th ed., MIT Press, 2009.

[18] Hallmann, M., Schlotterer, M., Heidecker, A., Sagliano, M., Fumenti, F., Maiwald, V., and Schwarz, R., *GTOC 9, Multiple Space Debris Rendezvous Trajectory Design in the J2 environment*, 2017.

[19] Berto, F., Hua, C., Park, J., Luttmann, L., Ma, Y., Bu, F., Wang, J., Ye, H., Kim, M., Choi, S., Zepeda, N. G., Hottung, A., Zhou, J., Bi, J., Hu, Y., Liu, F., Kim, H., Son, J., Kim, H., Angioni, D., Kool, W., Cao, Z., Zhang, Q., Kim, J., Zhang, J., Shin, K., Wu, C., Ahn, S., Song, G., Kwon, C., Tierney, K., Xie, L., and Park, J., "RL4CO: an Extensive Reinforcement Learning for Combinatorial Optimization Benchmark," , Jun. 2024. https://doi.org/10.48550/arXiv.2306.17100, URL http://arxiv.org/abs/2306.17100, arXiv:2306.17100 [cs].

[20] Locke, J., Colvin, T., J., Ratliff, L., Abdulhamid, A., and Samples, C., "Cost and Benefit Analysis of Mitigating, Tracking, and Remediating Orbital Debris," Tech. Rep. 20240003484, NASA Office of Technology, Policy, and Strategy, NASA Headquarters 300 E Street SW Washington, DC 20024, 2024. URL https://www.nasa.gov/wp-content/uploads/2024/05/2024-otps-cba-of-orbital-debris-phase-2-plus-svgs-v3-tjc-tagged.pdf?emrc=224cd2.

[21] Izzo, D., Märtens, M., and Pan, B., "A survey on artificial intelligence trends in spacecraft guidance dynamics and control," *Astrodynamics*, Vol. 3, No. 4, 2019, pp. 287–299. https://doi.org/10.1007/s42064-018-0053-6, URL https://doi.org/10.1007/s42064-018-0053-6.

[22] Izzo, D., Ozturk, E., and Märtens, M., *Interplanetary Transfers via Deep Representations of the Optimal Policy and/or of the Value Function*, 2019. https://doi.org/10.48550/arXiv.1904.08809.

[23] Klinkrad, H., *Space Debris*, Springer Praxis Books, Springer Berlin Heidelberg, 2006. https://doi.org/10.1007/3-540-37674-7, URL http://link.springer.com/10.1007/3-540-37674-7.

[24] "ESA'S Annual Space Environment Report," LOG GEN-DB-LOG-00288-OPS-SD, ESA ESOC Space Debris Office, Robert-Bosch-Strasse 5 D-64293 Darmstadt Germany, Jul. 2024. URL https://www.sdo.esoc.esa.int/environment_report/Space_Environment_Report_latest.pdf.

[25] Kessler, D. J., and Cour-Palais, B. G., "Collision frequency of artificial satellites: The creation of a debris belt," *Journal of Geophysical Research*, Vol. 83, No. A6, 1978, pp. 2637–2646. https://doi.org/10.1029/JA083iA06p02637, URL https://ui.adsabs.harvard.edu/abs/1978JGR....83.2637K/abstract.

[26] Liou, J. C., and Johnson, N. L., "Instability of the present LEO satellite populations," *Advances in Space Research*, Vol. 41, No. 7, 2008, pp. 1046–1053. https://doi.org/10.1016/j.asr.2007.04.081, URL https://www.sciencedirect.com/science/article/pii/S0273117707004097.

[27] Hall, L., "The History of Space Debris," *Space Traffic Management Conference*, 2014. URL https://commons.erau.edu/stm/2014/thursday/19.

[28] for Outer Space Affairs, U. N. O., "Space Debris Mitigation Guidelines of the Committee on the Peaceful Uses of Outer Space," , Jan. 2010. URL https://www.unoosa.org/pdf/publications/st_space_49E.pdf.

[29] for the Assessment of NASA's Orbital Debris Programs, C., *Limiting Future Collision Risk to Spacecraft: An Assessment of NASA's Meteoroid and Orbital Debris Programs*, National Academies Press, Washington, D.C., 2011. https://doi.org/10.17226/13244, URL http://www.nap.edu/catalog/13244.

[30] Group, E. S. D. M. W., "ESA Space Debris Mitigation Requirements," , Oct. 2023. URL https://technology.esa.int/upload/media/DGHKMZ_6542582e18e33.pdf.

[31] Shan, M., Guo, J., and Gill, E., "Review and comparison of active space debris capturing and removal methods," *Progress in Aerospace Sciences*, Vol. 80, 2016, pp. 18–32. https://doi.org/10.1016/j.paerosci.2015.11.001, URL https://www.sciencedirect.com/science/article/pii/S0376042115300221.

[32] Wiedemann, C., Krag, H., Bendisch, J., and Sdunnus, H., "Analyzing costs of space debris mitigation methods," *Advances in Space Research*, Vol. 34, No. 5, 2004, pp. 1241–1245. https://doi.org/10.1016/j.asr.2003.10.041, URL https://www.sciencedirect.com/science/article/pii/S0273117704001048.

[33] Borelli, G., Trisolini, M., Massari, M., and Colombo, C., *A comprehensive ranking framework for active debris removal missions candidates*, 2021.

[34] Saunders, C., Forshaw, J., Lappas, V., Wade, D., Iron, D., Hughes, N., Greves, D., and Biesbroek, R., "Business and economic considerations for service oriented active debris removal missions," *Proceedings of the International Astronautical Congress, IAC*, Vol. 3, 2014, pp. 1729–1743.

[35] Forshaw, J. L., Aglietti, G. S., Fellowes, S., Salmon, T., Retat, I., Hall, A., Chabot, T., Pisseloup, A., Tye, D., Bernal, C., Chaumette, F., Pollini, A., and Steyn, W. H., "The active space debris removal mission RemoveDebris. Part 1: From concept to launch," *Acta Astronautica*, Vol. 168, 2020, pp. 293–309. https://doi.org/10.1016/j.actaastro.2019.09.002, URL https://www.sciencedirect.com/science/article/pii/S0094576519312512.

[36] Biesbroek, R., Innocenti, L., Wolahan, A., and Serrano, S. M., "E.DEORBIT – ESA'S ACTIVE DEBRIS REMOVAL MISSION," ESA Space Debris Office, Darmstadt, Germany, 2017. URL https://conference.sdo.esoc.esa.int/proceedings/sdc7/paper/1053/SDC7-paper1053.pdf.

[37] Biesbroek, R., Aziz, S., Wolahan, A., Cipolla, S., Richard-Noca, M., and Piguet, L., "THE CLEARSPACE-1 MISSION: ESA AND CLEARSPACE TEAM UP TO REMOVE DEBRIS," ESA Space Debris Office, Darmstadt, Germany, 2021. URL https://conference.sdo.esoc.esa.int/proceedings/sdc8/paper/320/SDC8-paper320.pdf.

[38] Forshaw, J. L., Aglietti, G. S., Navarathinam, N., Kadhem, H., Salmon, T., Pisseloup, A., Joffre, E., Chabot, T., Retat, I., Axthelm, R., Barraclough, S., Ratcliffe, A., Bernal, C., Chaumette, F., Pollini, A., and Steyn, W. H., "RemoveDEBRIS: An in-orbit active debris removal demonstration mission," *Acta Astronautica*, Vol. 127, 2016, pp. 448–463. https://doi.org/10.1016/j.actaastro.2016.06.018, URL https://www.sciencedirect.com/science/article/pii/S009457651530117X.

[39] Kelso, T., "Analysis of the Iridium 33Cosmos 2251 Collision," , ???? URL https://www.researchgate.net/publication/242543407_Analysis_of_the_Iridium_33Cosmos_2251_Collision.

[40] Johnson, N. L., Stansbery, E., Liou, J.-C., Horstman, M., Stokely, C., and Whitlock, D., "The characteristics and consequences of the break-up of the Fengyun-1C spacecraft," *Acta Astronautica*, Vol. 63, No. 1-4, 2008, pp. 128–135. https://doi.org/10.1016/j.actaastro.2007.12.044, URL https://linkinghub.elsevier.com/retrieve/pii/S0094576507003281.

[41] Pardini, C., and Anselmo, L., "The short-term effects of the Cosmos 1408 fragmentation on neighboring inhabited space stations and large constellations," *Acta Astronautica*, Vol. 210, 2023, pp. 465–473. https://doi.org/10.1016/j.actaastro.2023.02.043, URL https://www.sciencedirect.com/science/article/pii/S0094576523001078.

[42] Johnson, N. L., Gabbard, J. R., Devere, G. T., and Johnson, E. E., "History of on-orbit satellite fragmentations," Tech. Rep. NAS 1.26:171814, Aug. 1984. URL https://ntrs.nasa.gov/citations/19840026390, nTRS Author Affiliations: Teledyne Brown Engineering NTRS Document ID: 19840026390 NTRS Research Center: Legacy CDMS (CDMS).

[43] Morante, D., Sanjurjo Rivo, M., and Soler, M., "A Survey on Low-Thrust Trajectory Optimization Approaches," *Aerospace*, Vol. 8, No. 3, 2021. https://doi.org/10.3390/aerospace8030088, URL https://www.mdpi.com/2226-4310/8/3/88.

[44] O'Reilly, D., Herdrich, G., and Kavanagh, D. F., "Electric Propulsion Methods for Small Satellites: A Review," *Aerospace*, Vol. 8, No. 1, 2021, p. 22. https://doi.org/10.3390/aerospace8010022, URL https://www.mdpi.com/2226-4310/8/1/22, number: 1 Publisher: Multidisciplinary Digital Publishing Institute.

[45] Conversano, R. W., and Wirz, R. E., "Mission Capability Assessment of CubeSats Using a Miniature Ion Thruster," *Journal of Spacecraft and Rockets*, Vol. 50, No. 5, 2013, pp. 1035–1046. https://doi.org/10.2514/1.A32435, URL https://doi.org/10.2514/1.A32435, publisher: American Institute of Aeronautics and Astronautics _eprint: https://doi.org/10.2514/1.A32435.

[46] Hintz, G., "Survey of Orbit Element Sets," *Journal of Guidance Control and Dynamics - J GUID CONTROL DYNAM*, Vol. 31, 2008, pp. 785–790. https://doi.org/10.2514/1.32237.

[47] Betts, J. T., "Survey of Numerical Methods for Trajectory Optimization," *Journal of Guidance, Control, and Dynamics*, Vol. 21, No. 2, 1998, pp. 193–207. https://doi.org/10.2514/2.4231, URL https://arc.aiaa.org/doi/10.2514/2.4231, publisher: American Institute of Aeronautics and Astronautics.

[48] Falck, R., Sjauw, W., and Smith, D., *Comparison of Low-Thrust Control Laws for Applications in Planetocentric Space*, 2014. https://doi.org/10.2514/6.2014-3714, journal Abbreviation: 50th AIAA/ASME/SAE/ASEE Joint Propulsion Conference 2014 Publication Title: 50th AIAA/ASME/SAE/ASEE Joint Propulsion Conference 2014.

[49] Ilgen, M., "Low thrust otv guidance using Lyapunov optimal feedback control techniques." *Advances in the Astronautical Sciences*, Vol. 85, No. Pt 2, 1993, pp. 1527–1545. URL https://jglobal.jst.go.jp/en/detail?JGLOBAL_ID=200902144132473360.

[50] Petropoulos, A., "Refinements to the Q-law for low-thrust orbit transfers," Vol. 120, 2005, pp. 963–982.

[51] Varga, G. I., and Perez, J. M. S., "Many-Revolution Low-Thrust Orbit Transfer Computation Using Equinoctial Q-Law Including J2 and Eclipse Effects," *ICATT*, 2016. URL https://indico.esa.int/event/111/contributions/346/attachments/336/377/Final_Paper_ICATT.pdf.

[52] Petropoulos, A., "Low-Thrust Orbit Transfers Using Candidate Lyapunov Functions with a Mechanism for Coasting," *AIAA/AAS Astrodynamics Specialist Conference and Exhibit*, Guidance, Navigation, and Control and Co-located Conferences, American Institute of Aeronautics and Astronautics, 2004. https://doi.org/10.2514/6.2004-5089, URL https://arc.aiaa.org/doi/10.2514/6.2004-5089.

[53] Lee, S., Von Allmen, P., Fink, W., Petropoulos, A., and Terrile, R., "Design and optimization of low-thrust orbit transfers," *IEEE Aerospace Conference Proceedings*, 2005.

[54] Shannon, J., Ozimek, M., Atchison, J., and Hartzell, C., "Q-Law Aided Direct Trajectory Optimization of Many-Revolution Low-Thrust Transfers," *Journal of Spacecraft and Rockets*, Vol. 57, 2020, pp. 1–11. https://doi.org/10.2514/1.A34586.

[55] Narayanaswamy, S., and Damaren, C. J., "Equinoctial Lyapunov Control Law for Low-Thrust Rendezvous," *Journal of Guidance, Control, and Dynamics*, Vol. 46, No. 4, 2023, pp. 781–795. https://doi.org/10.2514/1.G006662, URL https://doi.org/10.2514/1.G006662, publisher: American Institute of Aeronautics and Astronautics _eprint: https://doi.org/10.2514/1.G006662.

[56] Lantukh, D., Ranieri, C., Diprinzio, M., and Edelman, P., *Enhanced Q-Law Lyapunov Control for Low-Thrust Transfer and Rendezvous Design*, 2017.

[57] Holt, H., Armellin, R., Baresi, N., Hashida, Y., Turconi, A., Scorsoglio, A., and Furfaro, R., "Optimal Q-laws via reinforcement learning with guaranteed stability," *Acta Astronautica*, Vol. 187, 2021, pp. 511–528. https://doi.org/10.1016/j.actaastro.2021.07.010, URL https://www.sciencedirect.com/science/article/pii/S0094576521003684.

[58] Wakker, K., *Fundamentals of Astrodynamics*, TU Delft Library, 2015. URL https://repository.tudelft.nl/islandora/object/uuid%3A3fc91471-8e47-4215-af43-718740e6694e.

[59] Dirkx, D., Fayolle, M., Garrett, G., Avillez, M., Cowan, K., Cowan, S., Encarnacao, J., Fortuny Lombrana, C., Gaffarel, J., Hener, J., Hu, X., van Nistelrooij, M., Oggionni, F., and Plumaris, M., "The open-source astrodynamics Tudatpy software - overview for planetary mission design and science analysis," *European Planetary Science Congress*, 2022, pp. EPSC2022–253. https://doi.org/10.5194/epsc2022-253, URL https://ui.adsabs.harvard.edu/abs/2022EPSC...16..253D/abstract.

[60] Hon, J., "Optimization Framework for Low-Thrust Active Debris Removal Missions with Multiple Selected Targets," AMOS, 2022. URL https://amostech.com/TechnicalPapers/2022/Poster/Hon.pdf.

[61] Cohen, J., *Statistical Power Analysis for the Behavioral Sciences*, 2$^{\text{nd}}$ ed., Routledge, New York, 1988. https://doi.org/10.4324/9780203771587.

[62] Draper, N. R., and Smith, H., *Applied Regression Analysis*, John Wiley & Sons, 1998. Google-Books-ID: d6NsDwAAQBAJ.

[63] Guthrie, W. F., and Heckert, N. A., "NIST/SEMATECH Engineering Statistics Handbook," *NIST*, 2016. URL https://www.itl.nist.gov/div898/handbook/, last Modified: 2016-04-27T15:32-04:00.

[64] Biscani, F., and Izzo, D., "A parallel global multiobjective framework for optimization: pagmo," *Journal of Open Source Software*, Vol. 5, No. 53, 2020, p. 2338. https://doi.org/10.21105/joss.02338, URL https://joss.theoj.org/papers/10.21105/joss.02338.

[65] Blank, J., and Deb, K., "Pymoo: Multi-Objective Optimization in Python," *IEEE Access*, Vol. 8, 2020, pp. 89497–89509. https://doi.org/10.1109/ACCESS.2020.2990567, URL https://ieeexplore.ieee.org/document/9078759, conference Name: IEEE Access.

[66] Blum, C., and Roli, A., "Metaheuristics in Combinatorial Optimization: Overview and Conceptual Comparison," *ACM Comput. Surv.*, Vol. 35, 2001, pp. 268–308. https://doi.org/10.1145/937503.937505.

[67] Coello, C., Veldhuizen, D., and Lamont, G., *Evolutionary Algorithms for Solving Multi-Objective Problems Second Edition*, 2007. https://doi.org/10.1007/978-0-387-36797-2, journal Abbreviation: Kluwer Academic Publication Title: Kluwer Academic.

[68] Mitchell, R., Cooper, J., Frank, E., and Holmes, G., "Sampling Permutations for Shapley Value Estimation," *Journal of Machine Learning Research*, Vol. 23, No. 43, 2022, pp. 1–46. URL http://jmlr.org/papers/v23/21-0439.html.

[69] Kirkpatrick, S., Gelatt, C. D., and Vecchi, M. P., "Optimization by Simulated Annealing," *Science*, Vol. 220, No. 4598, 1983, pp. 671–680. https://doi.org/10.1126/science.220.4598.671, URL https://www.science.org/doi/10.1126/science.220.4598.671, publisher: American Association for the Advancement of Science.

[70] Eberl, M., "Fisher-Yates shuffle," *Arch. Formal Proofs*, 2016. URL https://www.semanticscholar.org/paper/Fisher-Yates-shuffle-Eberl/5e24ffebdc35e8e11af823505cbd5c6d5407f23e.

[71] Knuth, D., *The Art of Computer Programming*, 3$^{\text{rd}}$ ed., Vol. Volume 2: Seminu- merical Algorithms., Addison-Wesley Longman Publishing Co., Inc., USA, 1997. URL https://www-cs-faculty.stanford.edu/~knuth/taocp.html.

[72] Mallows, C. L., "Non-Null Ranking Models. I," *Biometrika*, Vol. 44, No. 1/2, 1957, pp. 114–130. https://doi.org/10.2307/2333244, URL https://www.jstor.org/stable/2333244, publisher: [Oxford University Press, Biometrika Trust].

[73] Diaconis, P., "Group Representations in Probability and Statistics," *Lecture Notes-Monograph Series*, Vol. 11, 1988, pp. i–192. URL https://www.jstor.org/stable/4355560, publisher: Institute of Mathematical Statistics.

[74] Irurozki, E., “Sampling and learning distance-based probability models for permutation spaces,” Ph.D. thesis, Universidad del País Vasco - Euskal Herriko Unibertsitatea, 2014. URL https://dialnet.unirioja.es/servlet/tesis?codigo=213087.

[75] Bean, J. C., “Genetic Algorithms and Random Keys for Sequencing and Optimization,” *ORSA Journal on Computing*, Vol. 6, No. 2, 1994, pp. 154–160. https://doi.org/10.1287/ijoc.6.2.154, URL https://pubsonline.informs.org/doi/abs/10.1287/ijoc.6.2.154, publisher: ORSA.

[76] Krömer, P., Uher, V., and Snášel, V., “Novel Random Key Encoding Schemes for the Differential Evolution of Permutation Problems,” *IEEE Transactions on Evolutionary Computation*, Vol. 26, No. 1, 2022, pp. 43–57. https://doi.org/10.1109/TEVC.2021.3087802, URL https://ieeexplore.ieee.org/document/9449662, conference Name: IEEE Transactions on Evolutionary Computation.

[77] Londe, M. A., Pessoa, L. S., Andrade, C. E., and Resende, M. G. C., “Biased random-key genetic algorithms: A review,” *European Journal of Operational Research*, 2024. https://doi.org/10.1016/j.ejor.2024.03.030, URL https://www.sciencedirect.com/science/article/pii/S0377221724002303.

[78] Boley, A., and Byers, M., “Satellite mega-constellations create risks in Low Earth Orbit, the atmosphere and on Earth,” *Scientific Reports*, Vol. 11, 2021, p. 10642. https://doi.org/10.1038/s41598-021-89909-7.

[79] Freitag, M., and Al-Onaizan, Y., “Beam Search Strategies for Neural Machine Translation,” *Proceedings of the First Workshop on Neural Machine Translation*, edited by T. Luong, A. Birch, G. Neubig, and A. Finch, Association for Computational Linguistics, Vancouver, 2017, pp. 56–60. https://doi.org/10.18653/v1/W17-3207, URL https://aclanthology.org/W17-3207.

[80] Lowerre, B., “The HARPY speech recognition system,” 1976. URL https://www.semanticscholar.org/paper/The-HARPY-speech-recognition-system-Lowerre/bdb3f20fe41bb95f6bc9d162e827de8db3f952d7.

[81] Waggener, B., and Waggener, W. N., *Pulse Code Modulation Techniques*, Springer Science & Business Media, 1995. Google-Books-ID: 8l_o6kI3760C.

[82] Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Köpf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., and Chintala, S., “PyTorch: an imperative style, high-performance deep learning library,” *Proceedings of the 33rd International Conference on Neural Information Processing Systems*, 721, Curran Associates Inc., Red Hook, NY, USA, 2019, pp. 8026–8037.

[83] von Neumann, J., “Various Techniques Used in Connection With Random Digits,” *Journal of Research of the National Bureau of Standards*, 1951. URL https://mcnp.lanl.gov/pdf_files/InBook_Computing_1961_Neumann_JohnVonNeumannCollectedWorks_VariousTechniquesUsedinConnectionwithRandomDigits.pdf.

[84] Hastings, W. K., “Monte Carlo sampling methods using Markov chains and their applications,” *Biometrika*, , No. 57, 1970, p. 97. URL https://probability.ca/hastings/hastings.pdf.

[85] Kool, W., van Hoof, H., and Welling, M., "Attention, Learn to Solve Routing Problems!" , Feb. 2019. https://doi.org/10.48550/arXiv.1803.08475, URL http://arxiv.org/abs/1803.08475, arXiv:1803.08475 [cs, stat].

[86] Williams, R. J., "Simple statistical gradient-following algorithms for connectionist reinforcement learning," *Machine Learning*, Vol. 8, No. 3, 1992, pp. 229–256. https://doi.org/10.1007/BF00992696, URL https://doi.org/10.1007/BF00992696.

[87] Konda, V., and Tsitsiklis, J., "Actor-Critic Algorithms," *Advances in Neural Information Processing Systems*, Vol. 12, MIT Press, 1999. URL https://proceedings.neurips.cc/paper_files/paper/1999/hash/6449f44a102fde848669bdd9eb6b76fa-Abstract.html.

[88] Mnih, V., Badia, A. P., Mirza, M., Graves, A., Lillicrap, T. P., Harley, T., Silver, D., and Kavukcuoglu, K., "Asynchronous Methods for Deep Reinforcement Learning," , Jun. 2016. https://doi.org/10.48550/arXiv.1602.01783, URL http://arxiv.org/abs/1602.01783, arXiv:1602.01783 [cs].

[89] Schulman, J., Wolski, F., Dhariwal, P., Radford, A., and Klimov, O., "Proximal Policy Optimization Algorithms," , Aug. 2017. https://doi.org/10.48550/arXiv.1707.06347, URL http://arxiv.org/abs/1707.06347, arXiv:1707.06347 [cs].

[90] Kool, W., Hoof, H. v., and Welling, M., "Attention, Learn to Solve Routing Problems!" 2018. URL https://openreview.net/forum?id=ByxBFsRqYm.

[91] Hastie, T., Tibshirani, R., and Friedman, J., *The Elements of Statistical Learning*, Springer Series in Statistics, Springer, New York, NY, 2009. https://doi.org/10.1007/978-0-387-84858-7, URL http://link.springer.com/10.1007/978-0-387-84858-7.

[92] Kutner, M. H. (ed.), *Applied linear statistical models*, 5th ed., McGraw-Hill/Irwin series Operations and decision sciences, McGraw-Hill Irwin, Boston, Mass., 2005.

[93] Taguchi, G., and Konishi, S., *Orthogonal Arrays and Linear Graphs: Tools for Quality Engineering*, American Supplier Institute, 1987. Google-Books-ID: _5BRnQAACAAJ.

[94] Taguchi, G., *Taguchi Methods: Design of Experiments*, ASI Press, 1993. Google-Books-ID: veNTAAAAMAAJ.

[95] Maghsoodloo, S., Ozdemir, G., Jordan, V., and Huang, C.-H., "Strengths and limitations of Taguchi's contributions to quality, manufacturing, and process engineering," *Journal of Manufacturing Systems*, Vol. 23, No. 2, 2004, pp. 73–126. https://doi.org/10.1016/S0278-6125(05)00004-X, URL https://www.sciencedirect.com/science/article/pii/S027861250500004X.

[96] Seabold, S., and Perktold, J., "Statsmodels: Econometric and Statistical Modeling with Python," *Proceedings of the 9th Python in Science Conference*, edited by S. v. d. Walt and J. Millman, 2010, pp. 92 – 96. https://doi.org/10.25080/Majora-92bf1922-011, URL https://proceedings.scipy.org/articles/Majora-92bf1922-011.

[97] Shapiro, S. S., and Wilk, M. B., "An analysis of variance test for normality (complete samples)†," *Biometrika*, Vol. 52, No. 3-4, 1965, pp. 591–611. https://doi.org/10.1093/biomet/52.3-4.591, URL https://doi.org/10.1093/biomet/52.3-4.591.

[98] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., and Polosukhin, I., "Attention Is All You Need," , Aug. 2023. https://doi.org/10.48550/arXiv.1706.03762, URL http://arxiv.org/abs/1706.03762, arXiv:1706.03762.

[99] Kaplan, J., McCandlish, S., Henighan, T., Brown, T. B., Chess, B., Child, R., Gray, S., Radford, A., Wu, J., and Amodei, D., "Scaling Laws for Neural Language Models," , Jan. 2020. https://doi.org/10.48550/arXiv.2001.08361, URL http://arxiv.org/abs/2001.08361, arXiv:2001.08361.

[100] Hoffmann, J., Borgeaud, S., Mensch, A., Buchatskaya, E., Cai, T., Rutherford, E., Casas, D. d. L., Hendricks, L. A., Welbl, J., Clark, A., Hennigan, T., Noland, E., Millican, K., Driessche, G. v. d., Damoc, B., Guy, A., Osindero, S., Simonyan, K., Elsen, E., Rae, J. W., Vinyals, O., and Sifre, L., "Training Compute-Optimal Large Language Models," , Mar. 2022. https://doi.org/10.48550/arXiv.2203.15556, URL http://arxiv.org/abs/2203.15556, arXiv:2203.15556.

5 Design And Optimization Of Multi-Rendezvous Maneuvres Based On Reinforcement Learning And Convex Optimization

75th International Astronautical Congress (IAC), Milan, Italy, 14-18 October 2024.

IAC–24–87909

Design And Optimization Of Multi-rendezvous Maneuvres Based On Reinforcement Learning And Convex Optimization

Antonio López Rivera[†a*], **Lucrezia Marcovaldi**[b], **Jesús Ramírez**[c], **Alex Cuenca**[d], **David Bermejo**[e]

[a] *Delft University of Technology, The Netherlands*
[b] *SENER Aerospace & Defence, Spain*
[c] *SENER Aerospace & Defence, Spain*
[d] *SENER Aerospace & Defence, Spain*
[e] *SENER Aerospace & Defence, Spain*
[*] *Corresponding author*
[†] *Main author*

Abstract

Optimizing space vehicle routing is crucial for critical applications such as on-orbit servicing, constellation deployment, and space debris de-orbiting. Multi-target Rendezvous presents a significant challenge in this domain. This problem involves determining the optimal sequence in which to visit a set of targets, and the corresponding optimal trajectories: this results in a demanding NP-hard problem. We introduce a framework for the design and refinement of multi-rendezvous trajectories based on heuristic combinatorial optimization and Sequential Convex Programming. Our framework is both highly modular and capable of leveraging candidate solutions obtained with advanced approaches and handcrafted heuristics. We demonstrate this flexibility by integrating an Attention-based routing policy trained with Reinforcement Learning to improve the performance of the combinatorial optimization process. We show that Reinforcement Learning approaches for combinatorial optimization can be effectively applied to spacecraft routing problems. We apply the proposed framework to the UARX Space OSSIE mission: we are able to thoroughly explore the mission design space, finding optimal tours and trajectories for a wide variety of mission scenarios.

Nomenclature

Symbol	Meaning	Value/Units
μ	Earth gravitational constant	3.986e5 $\mathrm{km^3\,s^{-2}}$
g_0	Standard Earth gravity	9.806 65 $\mathrm{m\,s^{-2}}$
R_e	Earth mean equatorial radius	6378.14 km
J_2	Second Earth zonal harmonic	1.0826e–3
I_{sp}	Specific impulse	s
a	Semi-major axis	m
p	Semi-latus rectum	m
e	Eccentricity	–
i	Inclination	rad
Ω	RAAN	rad
ω	AOP	rad
σ	True Anomaly	rad
L	True Longitude	rad
n	Orbital mean motion	$\mathrm{s^{-1}}$

Acronyms/Abbreviations

Term	Acronym
Mixed-Integer Nonlinear Programming	(MINLP)
Traveling Salesman Problem	(TSP)
Vehicle Routing Problem	(VRP)
Operations Research	(OR)
Space Traveling Salesman Problem	(STSP)
Orbit Transfer Vehicle	(OTV)
Modified Equinoctial Elements	(MEEs)
Multiple Hohmann Transfer	(MHT)
Nodal Inclination Change manoeuvre	(NIC)
Semi-major Axis	(SMA)
Right Ascension of Ascending Node	(RAAN)
Argument Of Perigee	(AOP)
Time Of Flight	(TOF)
Sequential Convex Programming	(SCP)
SENER Optimization Toolbox	(SOTB)
Machine Learning	(ML)
Reinforcement Learning	(RL)
Neural Combinatorial Optimization	(NCO)
Deep Neural Network	(DNN)
Graph Attention Network	(GAT)
Advantage Actor-Critic	(A2C)
Proximal Policy Optimization	(PPO)

1. Introduction

The present work introduces a general optimization framework for multiple-rendezvous manoeuvres, which see a spacecraft approaching a sequence of objects in orbit as efficiently as possible. An optimal solution to the multi-target rendezvous trajectory optimization problem or STSP consists of the optimal sequence in which to visit a set of targets and the optimal transfer trajectory between each target in the optimal sequence. The STSP is an NP-hard MINLP problem with factorial complexity over the number of targets. The STSP bears resemblance to the classical TSP, albeit with the added complexities inherent to the space environment, notably a 6-dimensional state space, mass dynamics, propulsion constraints, and the change over time of target states due to secular perturbations, chiefly J_2 for Earth-orbiting spacecraft.

Formally, the STSP is the problem of finding a minimum weight path (if the spacecraft must end the tour back at its initial state, a Hamiltonian path) in a complete weighted graph $G := \{\mathcal{V}(t), \mathbf{W}(\pi)\}$, where $\mathcal{V}(t)$ is the set of graph vertexes (targets, the state of which drifts over time) and $\mathbf{W}(\pi) := \mathcal{V} \times \mathcal{V} \to \mathbb{R}^+$ is a map that associates an edge weight (a transfer cost) to each ordered vertex pair[1], and may dependent on the sequence π in which the targets are visited. One such case is when payload mass is a large percentage of the spacecraft's wet mass, and thus deployment sequence has a non-negligible impact on fuel consumption. The STSP is an example of a MINLP problem, which are notoriously difficult to approach, and has seen a considerable surge in interest in active debris removal missions[1–6] to tackle the space debris problem[7, 8], as well as on-orbit servicing missions[3, 9, 10] and advanced space logistics concepts[11]. A standard approach to solve MINLP problems is Benders decomposition[5, 12], where the MINLP problem is divided into a higher-level combinatorial optimization problem and a lower-level trajectory optimization problem; the higher level combinatorial problem is then solved exactly by applying Benders optimality cuts. Decomposition approaches using heuristic optimization are common as well[3, 4, 6]. In both approaches a transfer cost estimator is used to calculate the cumulative cost of tours in the combinatorial optimization problem. Transfer cost estimators may be database-dependent[13, 14], database-independent (analytical), or learning-based[15].

State-of-the-art combinatorial optimization methods fall in two camps: exact methods and heuristic methods, which are less costly and can produce near-optimal results, but cannot offer optimality guarantees whatsoever[1, 16]. Exact methods based on tree searches are the norm for highly complex, large STSP variants; all winning submissions of the Global Trajectory Optimization Competitions have made use of tree search approaches[1, 13, 17]. Heuristic optimization methods however are an attractive option to solve smaller STSP instances (up to hundreds of targets[1]) due to their capacity to achieve near-optimal results with lower computational cost[1], and are widely applied in literature to tackle multi-rendezvous mission design[1–4, 6]. Heuristic optimization methods have also been successfully applied to complex STSP instances where the cost of exact approaches is unfeasible[13]. Furthermore, the availability of highly performant, open-source heuristic multi-objective optimization libraries such as `pygmo`[1][18] and `pymoo`[2][19] greatly eases the application, benchmarking and selection of diverse heuristic optimization algorithms for specific problem variants. We thus opt for population-based heuristic optimization to develop the combinatorial optimization component of our solver.

Convex optimization is leveraged to tailor optimal manoeuvres to the specific vehicle requirements, such as the specifics of the propulsion system. The applications of convex optimization in the field of aerospace have grown in importance in the recent years. SOCP, in particular, is a common choice when nonlinear constraints are involved in the formulation of the OCP at the base of the orbit transfer optimization[20, 21]. These methods however are vulnerable to convergence issues: this has been tackled in literature by using SCP solvers provided with an initial guess close to the optimal solution, and by keeping propagation and accumulation errors due to the iterations of the algorithm low; Foust et al.[22] and Ramírez and Hewing[23] propose the implementation of an SCP solver in combination with an 4th-order Runge-Kutta, to integrate in the OCP an accurate model of the dynamics and reduce error accumulation. Discretization is crucial in determining the ability of the solver to find an optimal solution, re-adapting the dynamics and constraints of the OCP as functions of a finite number of parameters. Topputo et al.[24] develop an SCP solver based on mesh refinement to obtain a quasi-optimal warm starting, demonstrating the robustness of the algorithm and its efficiency to generate warm starts for the SCP solver. SCP solvers based on interior-point methods, in particular, allow to significantly reduce computational cost, making convex programming algorithms suitable also for demanding applications, such as on-line guidance[23, 25].

This work presents a modular and adaptable STSP optimization framework based on Benders decomposition

[1] `https://esa.github.io/pygmo2/`

[2] `https://pymoo.org/`

and heuristic combinatorial optimization, capable of solving highly tailored versions of the STSP. We demonstrate its capabilities by solving the multi-rendezvous trajectory optimization problem of UARX Space's OSSIE[3] OTV: a translational and mass-dynamic multi-satellite deployment problem in LEO.

The paper is organized as follows: Section 2 describes the operational profile and performance envelope of the OSSIE OTV. Section 3 defines the models used for the orbital dynamics and perturbations. Section 4 proceeds to define the trajectory design and trajectory cost estimation methods used for the UARX Space OSSIE OTV. Section 4 presents the Bolza formulation of the MINLP, the architecture of our optimization framework, introduces the heuristic combinatorial optimization methodology, and defines the SCP algorithm together with the OCP and the solver tailoring based on SOTB. Experimental results are discussed in Section 7. Subsection 7.1 defines a model to generate randomized mission scenarios for OSSIE. Subsection 7.2 and Subsection 7.3 discuss the results obtained from heuristic and neural combinatorial optimization. A statistical mission feasibility analysis follows in Subsection 7.4. Subsection 7.5 presents the trajectory optimization results from SCP. The results section concludes with the formal verification of the obtained trajectories in a high-fidelity simulator in Subsection 7.6. Lastly, Section 8 lays out our main conclusions and recommendations for future research.

2. Mission Profile

To demonstrate the capabilities of the proposed framework, the UARX Space Orbit Solutions to Simplify Injection and Exploration OTV, known as OSSIE, will be used as a case study. Depicted in Fig. 1, OSSIE is a modular payload delivery platform with LEO, MEO and cis-lunar capability. In this paper, we consider a nominal mission profile aiming to deliver 4 PocketQubes, 8 CubeSats, and 1 small satellite to LEO.

The propulsion system used for this mission consists of 4 parallel Dawn Aerospace B20 bi-propellant (nitrous oxide and propene) thrusters[4] (specifications in Table 1). As of the time of writing, the duty-cycle constraints of the thruster cluster constrain manoeuver design to multiple-revolution transfers, with up to two impulses per orbit in LEO.

The specifications of the current configuration of OSSIE follow in Table 1. OSSIE has a wet mass of approximately 235 kg, of which close to 50% is shed through the mission —either deployed or consumed propellant; this means that the mass deployment sequence may have a considerable impact on the fuel required to complete the mission. OSSIE is deployed to a nominal insertion orbit in LEO, which constraints transfer sequence design, and must conduct a decommissioning manoeuvre to ensure it decays within 5 years of EOL as per the ESA Space Debris Mitigation Requirements [26].

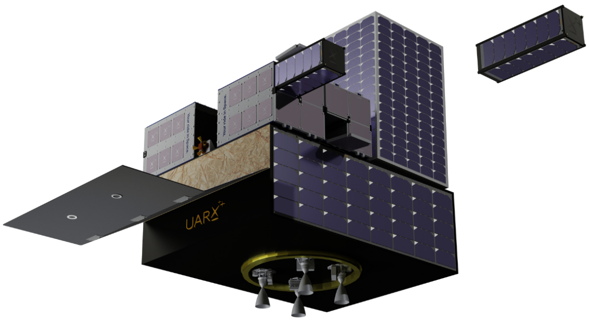


Fig. 1. UARX Space OSSIE OTV. Credit: UARX Space.

Table 1. Specifications of the OSSIE OTV and Dawn Aerospace B20 thrusters.

OSSIE	Value	Dawn Aerospace B20	Value
Wet mass	235 kg	Specific impulse	277 s
Payload mass	80 kg	Peak thrust	12.6 N
Fuel mass	35 kg	Minimum impulse bit	1 N s
Cargo capacity	48 U		

Under the previous considerations, the OSSIE trajectory optimization problem results in a highly tailored multi-rendezvous trajectory optimization problem, which aims to minimize fuel consumption while accounting for:

- Multi-revolution impulsive manoeuvres in LEO under perturbations.
- The impact of mass deployment sequence on propellant consumption.
- Insertion and decommissioning orbit constraints.

3. Environment Model

To setup the optimization space, the spacecraft state is modelled using the MEEs described by Hintz[27], including the retrograde factor I, which are nonsingular for all

[3] https://www.uarx.com/projects/ossie.php

[4] https://www.dawnaerospace.com/green-propulsion

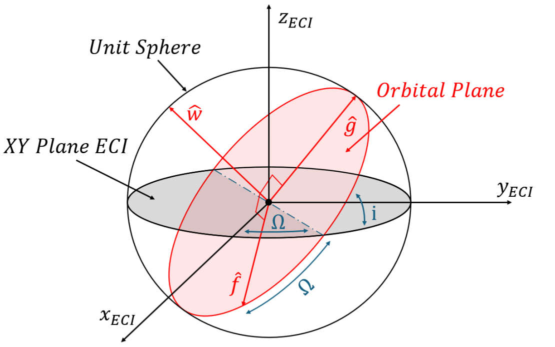


Fig. 2. Modified Equinoctial Elements (MEE) with respect to orbital plane.

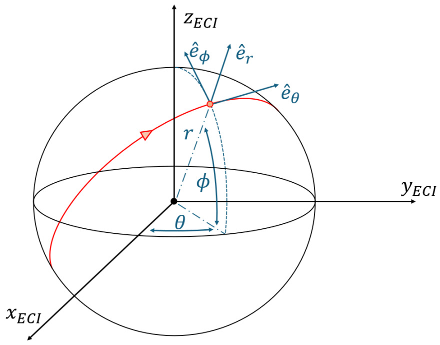


Fig. 3. ECI and LVLH reference frames.

eccentricities and inclinations. The conversion from classical elements to MEEs follows in Eq. 1:

$$
\begin{aligned}
p &= a(1-e^2);\\
f &= e\cos(\omega+\Omega);\\
g &= e\sin(\omega+\Omega);\\
h &= \tan(i/2)\sin(\Omega);\\
k &= \tan(i/2)\cos(\Omega);\\
L &= \theta + I\Omega + \omega;
\end{aligned}
\tag{1}
$$

The Gauss Variational Equations for MEEs[27] in Eq. 2, are used to model the time evolution of the spacecraft's translational state:

$$
\begin{aligned}
\frac{\mathrm{d}p}{\mathrm{d}t} &= \frac{2p}{w}\sqrt{\frac{p}{\mu}}\Delta_t;\\
\frac{\mathrm{d}f}{\mathrm{d}t} &= \sqrt{\frac{p}{\mu}}\Bigg\{\Delta_r\sin(L);\\
&\qquad + \frac{(w+1)\cos(L)+f}{w}\Delta_t - g\frac{v}{w}\Delta_n\Bigg\};\\
\frac{\mathrm{d}g}{\mathrm{d}t} &= \sqrt{\frac{p}{\mu}}\Bigg\{-\Delta_r\cos(L);\\
&\qquad + \frac{(w+1)\sin(L)+g}{w}\Delta_t - f\frac{v}{w}\Delta_n\Bigg\};\\
\frac{\mathrm{d}h}{\mathrm{d}t} &= \sqrt{\frac{p}{\mu}}\frac{s^2}{2w}\cos(L)\Delta_n;\\
\frac{\mathrm{d}k}{\mathrm{d}t} &= \sqrt{\frac{p}{\mu}}\frac{s^2}{2w}\sin(L)\Delta_n;\\
\frac{\mathrm{d}L}{\mathrm{d}t} &= \sqrt{\mu p}\left(\frac{w}{p}\right)^2 + \sqrt{\frac{p}{\mu}}\frac{v}{w}\Delta_n;
\end{aligned}
\tag{2}
$$

where s^2, v and w are defined as follows,

$$
\begin{aligned}
s^2 &= 1 + h^2 + k^2;\\
v &= h\sin(L) - k\cos(L);\\
w &= 1 + f\cos(L) + g\sin(L);
\end{aligned}
\tag{3}
$$

and Δ_r, Δ_t and Δ_n are perturbing accelerations in the radial, tangential, and normal directions of the spacecraft's LVLH frame. The LVLH and ECI frames are depicted in Fig. 3. The unit thrust vector in the ECI frame, $\hat{\mathbf{u}}_{\mathrm{ECI}}$, is related to the unit thrust vector in the LVLH frame $\hat{\mathbf{u}}_{\mathrm{MEE}}$ by Eq. 4.

$$
\hat{\mathbf{u}}_{\mathrm{ECI}} = [\hat{\mathbf{e}}_r \quad \hat{\mathbf{e}}_\theta \quad \hat{\mathbf{e}}_\phi]\hat{\mathbf{u}}_{\mathrm{MEE}} \tag{4a}
$$

$$
\hat{\mathbf{e}}_r = \frac{\mathbf{r}}{||\mathbf{r}||}; \quad \hat{\mathbf{e}}_\phi = \frac{\mathbf{r}\times\mathbf{v}}{||\mathbf{r}\times\mathbf{v}||}; \quad \hat{\mathbf{e}}_\theta = \hat{\mathbf{e}}_\phi \times \hat{\mathbf{e}}_r \tag{4b}
$$

Then, the thrust acceleration $\mathbf{a}_T$ applied by the spacecraft in the RWS frame is defined in Eq. 5, where $\hat{\mathbf{u}}$ is the direction of application of thrust. The spacecraft's mass is propagated according to Eq. 6, assuming constant I_{sp} through a single burn.

$$
\mathbf{a}_T = \frac{T}{m}\hat{\mathbf{u}} \tag{5}
$$

$$
\frac{\mathrm{d}m}{\mathrm{d}t} = \frac{T}{I_{sp}g_0}; \tag{6}
$$

With regards to perturbations, the greatest impact for trajectory design in the case of near-Earth orbits comes from the Earth oblateness gravity potential distortion[28], characterized by the second zonal harmonic coefficient or J_2. The instantaneous acceleration components due to J_2 are included in the environmental model, which in the LVLH frame follow in Eq. 7.

$$\begin{aligned}\Delta_{J_{2,r}} &= -\frac{3\mu J_2 R_e^2}{2r^4}\left(1-\frac{12v^2}{s^4}\right)\\ \Delta_{J_{2,\theta}} &= -\frac{12\mu J_2 R_e^2}{r^4}\left(\frac{v(h\cos L+k\sin L)}{s^4}\right)\\ \Delta_{J_{2,\phi}} &= -\frac{6\mu J_2 R_e^2}{r^4}\left(\frac{v(1-h^2-k^2)}{s^4}\right)\end{aligned} \tag{7}$$

The J_2 perturbing acceleration causes a secular perturbation on RAAN and AOP over a full orbit[29], modelled by Eq. 8, where $n = \sqrt{\mu/a^3}$ is the mean motion of the orbiting body.

$$\begin{aligned}\frac{\mathrm{d}\Omega}{\mathrm{d}t} &= -\frac{3}{2}J_2\left(\frac{R_e}{p}\right)n\cos i;\\ \frac{\mathrm{d}\omega}{\mathrm{d}t} &= -\frac{3}{4}J_2\left(\frac{R_e}{p}\right)n(5\cos^2 i-1));\end{aligned} \tag{8}$$

4. Guidance Policies and Transfer Cost Estimation

As discussed in Section 2, OSSIE is constrained to multi-revolution impulsive manoeuvres. For the payload deployment mission scenario under consideration, OSSIE must achieve high accuracy SMA, inclination and phase convergence: RAAN targeting is not of interest to the current mission, as RAAN drift is very large for LEO orbits. This section presents the orbit guidance policies used to achieve insertion and the analytical models used to estimate their cost in ΔV, fuel mass and time of flight. MHT manoeuvres are discussed in Subsection 4.1, NIC manoeuvres in Subsection 4.2, and sequential MHT-NIC manoeuvres in Subsection 4.3. Lastly, Subsection 4.4 discusses the impact of the J_2 perturbation on the manoeuvres.

4.1 Multiple Hohmann Transfer manoeuvres

A Hohmann transfer is the optimal manoeuvre to transfer between two coplanar circular orbits of different radii. It is a two-impulse manoeuvre consisting of an initial burn, either raising apogee or lowering perigee (depending on objective),and a circularization burn when the apogee (or perigee) of the transfer manoeuvre is reached. For OSSIE, the ΔV required to perform a direct Hohmann transfer may not be achievable. Instead, a MHT approach is used. The MHT splits the departure and circularization burns into many consecutive insertion and circularization smaller burns, affordable by the maximum ΔV that can be injected at a time with the employed thrusters.

4.1.1 Required ΔV, fuel mass and number of burns

The ΔV required by the MHT is equal to the one of a direct Hohmann Transfer achieving the same altitude change. The total magnitude can be computed as follows in Eq. 9, where ΔV_d and ΔV_c stand for departure and circularization ΔV, $V_0 = \sqrt{\mu/r_0}$ is the orbital velocity at the departure orbit and $\xi = r_1/r_0$ is the ratio of target to departure orbit[29].

$$\Delta V_{\text{MHT}} = \Delta V_d + \Delta V_c \tag{9a}$$

$$\Delta V_d = V_0\left|\sqrt{\frac{2n}{\xi+1}}-1\right| \tag{9b}$$

$$\Delta V_c = V_0\sqrt{\frac{1}{\xi}}\left|\sqrt{\frac{2n}{\xi+1}}-1\right| \tag{9c}$$

The fuel mass required to perform the MHT is calculated as follows in Eq. 10, assuming constant I_{sp} through the manoeuvre; the number k of burns required to perform a manoeuvre is obtained as the ceil fraction of required fuel mass over mass flow $\dot{m}_f = T/(I_{sp}g_0)$ (see Eq. 10). The mass flow is assumed constant, as T and I_{sp} are assumed constant through the transfer.

$$m_f = m_0\left[1-\exp\left(-\frac{\Delta V}{I_{sp}g_0}\right)\right]; \quad k = \left\lceil\frac{m_f}{\dot{m}_f}\right\rceil \tag{10}$$

4.1.2 Time Of Flight

The total TOF of the MHT follows in Eq. 11, where the burn frequency f_{burn} is defined in Eq. 12, $\overline{P}$ stands for the average orbital period through the transfer defined in Eq. 13, and k_d and k_c are the number departure and circularization burns respectively, defined in Eq. 10.

$$\text{TOF}_{\text{MHT}} = \min(1, f_{\text{burn}})\cdot(k_d+k_c)\cdot\overline{P}_{\text{MHT}} \tag{11}$$

$$f_{\text{burn}} = (T_{\text{burn}}+T_{\text{cooldown}})^{-1} \tag{12}$$

$$\overline{P}_{\text{MHT}} = \frac{4\pi}{5\sqrt{\mu}\,(r_2-r_1)}\left(r_2^{\frac{5}{2}}-r_1^{\frac{5}{2}}\right) \tag{13}$$

4.1.3 Phasing

The spacecraft ends the MHT at true longitude $L_0 + \pi$. To avoid phasing manoeuvres after reaching the required orbit, in order to acquire the target true longitude, the MHT start epoch is selected so that $L_0(t_0) = L_1(t_0 + \text{TOF}_{\text{MHT}}) - \pi$, where TOF is the estimated time of flight. This correction is computed after the MHT calculation, thus TOF_{MHT} is known.

4.2 Nodal Inclination Change manoeuvres

An NIC manoeuvre modifies the inclination of an orbit without affecting its RAAN. Thrust is applied impulsively as the spacecraft crosses the ascending or descending node, such that the resulting velocity vector is that of the orbit with the desired inclination. A multiple NIC manoeuvre consists of splitting the impulsive ΔV that must be applied into multiple burns, up to twice per orbit, as the spacecraft crosses the orbit ascending and descending nodes.

4.2.1 Required ΔV, fuel mass and number of burns

The ΔV required for a multiple NIC is the same as for a single burn and follows in Eq. 14. The fuel mass m_f and number of burns k_{NIC} required for to perform the NIC is calculated using Eq. 10[29].

$$\Delta V_{\text{NIC}} = 2V_0 \sin\left(\frac{\Delta i}{2}\right) \tag{14}$$

4.2.2 Time Of Flight

The TOF of a multiple NIC is calculated with Eq. 15, where $P = 2\pi\sqrt{r^3/\mu}$ is the constant orbital period.

$$\text{TOF}_{\text{NIC}} = \min(2, f_{\text{burn}}) \cdot (k_{\text{NIC}}) \cdot P \tag{15}$$

4.3 Sequential MHT-NIC manoeuvres

From Eq. 14 it is of paramount importance to lower orbital velocity before performing an NIC. OSSIE will often have to reach targets with different semi-major axes and inclinations than its current ones: the most common case is orbit acquisition for payloads which require a particular orbital altitude to carry out their activity as well as a sun-synchronous orbit —commonly the case for Earth observation and mapping satellites. Algorithm 1 resolves sequential MHT and NIC manoeuvres by conducting the NIC when the semi-major axis is highest, such that the ΔV required for the NIC leg is minimized.

Algorithm 1: Sequential MHT-NIC Transfer

```
if $r_2 > r_1$ then                    // Orbit raising
    Step 1: MHT
    Compute coast time to $L_0$
    Compute $\Delta V_{\text{MHT}}$
    Compute time of flight $t_{\text{MHT}}$
    Step 2: NIC
    Compute coast time to closest node
    Compute $\Delta V_{\text{NIC}}$
    Compute time of flight $t_{\text{NIC}}$
else                                   // Orbit lowering
    Step 1: NIC
    Compute coast time to closest node
    Compute $\Delta V_{\text{NIC}}$
    Compute time of flight $t_{\text{NIC}}$
    Step 2: MHT
    Compute coast time to $L_0$
    Compute $\Delta V_{\text{MHT}}$
    Compute time of flight $t_{\text{MHT}}$
end
```

4.4 J_2 perturbation

RAAN and AOP drift according to Eq. 8 for the duration of the manoeuvre. In the case of the MHT this only impacts the phasing coasting leg due to the drift of AOP. The NIC manoeuvre is unaffected.

5. Trajectory Optimization

This section presents the optimization framework proposed for solving the problem introduced in Section 2 subject to dynamics of Section 3 and manoeuvres as described in Section 4. In Subsection 5.1 we present a modular solver architecture based on integer-nonlinear decomposition and heuristic combinatorial optimization, capable of integrating diverse trajectory estimation methods, heuristic optimization algorithms, and trajectory refinement procedures. Subsection 5.2 discusses the combinatorial optimization components of the solver. Subsection 5.3 discusses trajectory generation, and Subsection 5.4 presents the SCP trajectory re-optimization block and specific tailoring to the OSSIE mission design case.

5.1 Solver Architecture

Our aim is to design a highly adaptable solver capable of generating multi-rendezvous trajectories for spacecraft guidance with strict feasibility guarantees, and to use it to design trajectories for the OSSIE OTV. We achieve this using a 3-stage solver architecture (Fig. 4) consisting of target sequence optimization, trajectory generation and re-optimization, and trajectory verification. Each component is implemented modularly using standardized interfaces, resulting in a framework that is highly adaptable to new mission design scenarios, while benefiting from high component reusability.

5.2 Sequence Optimization Problem

We make use of heuristic, population-based combinatorial optimization to optimize target sequences. Knowledge of near-optimal solutions, obtained by any means, is leveraged by initializing populations using distance-based permutation sampling. Later in Section 6 we demonstrate this versatility by integrating an attention-based STSP routing model trained using RL to greatly improve convergence to the global optimum.

5.2.1 Population Sampling and Encoding

High quality population sampling is crucial for population-based global combinatorial optimization[30]. We make use of uniform permutation sampling to cover the search space as widely as possible. The combinatorial optimization space is vast however, and advanced methods and hand-crafted heuristics are often used to determine near-optimal solutions prior to heuristic optimization[1, 13]. When these are available we make use of distance-based permutation sampling to spawn initial populations in the vicinity of candidate solutions, greatly helping the search space exploration process. We find that a combination of both approaches is optimal to balance exploration and exploitation of the search space.

Uniform Permutation Sampling We uniformly sample the permutation group $\mathfrak{S}^n$ by drawing Sobol points in the $[0, 1)^n$ hypercube and applying an `argsort` operation. This algorithm was found to yield superior samples than the Fisher-Yates shuffle[31] and Knuth's algorithm[30], while being orders of magnitude faster than advanced approaches such as Sobol Permutations[30].

Distance-based Permutation Sampling The Mallows model[32] (and related methods), first introduced in 1957[33], is the most relevant distance-based statistical model for permutations[34]. Permutations are exponentially less probable as they stray further from a central permutation σ_0. Refer to Irurozki, 2014[34] for a comprehensive definition of the model.

Population Encoding Random keys permutation encoding[35] is selected for its versatility for optimizing complex discrete optimization problems across various domains[36, 37]. Permutations are encoded using vectors of continuous values, or random keys, in the $[0, 1)$ range. A permutation $\sigma \in \mathfrak{S}^n$ is encoded by generating a vector of random keys $\mathbf{x} \in [0, 1)^n$, sorting it, and permuting it by σ. Decoding is done by applying an `argsort` operation to $\mathbf{x}$.

5.2.2 Combinatorial optimization

The generalized STSP to be solved is described in Eq. 16, where T is the set of all targets and c_π is the cumulative cost function of a tour. The cumulative cost function sequentially estimates transfer cost and time of flight and propagates the environment according to a perturbation model (see inputs in Fig. 4 for reference). Transfer cost estimation and environment propagation are implemented modularly, resulting in a highly flexible modelling process and high module reusability. In the case of OSSIE this is the J_2 secular perturbation model in Eq. 8. Departure state h and a decommissioning orbit d are implemented as optional search space constraints.

$$\begin{aligned} \min_{\pi} \quad & \sum_{k=0}^{n-1} c_{\pi(k),\pi(k+1)} \\ \text{s.t.} \quad & \pi(0) = h, \\ & \pi(n) = \begin{cases} h & \text{(Hamiltonian cycle)} \\ d & \text{(Decom.)} \end{cases}, \\ & \{\pi(k)\}_{k=1}^{n-1}, = T \setminus \{h\}, \\ & \pi(k) \in T \cup \{d\}, \quad k \in 0, 1, \dots, n \end{aligned} \tag{16}$$

The combinatorial optimization component is implemented using the `pygmo` [5] parallel multi-objective global optimization library, which is based on the underlying PAGMO C++ library[18]. This choice allows us to quickly evaluate the performance of a wide variety of global and local optimization algorithms, and design optimal solvers for particular mission design scenarios such as that of OSSIE.

Meta-heuristic Combinatorial optimization meta-heuristics are sophisticated algorithmic frameworks designed to efficiently explore large and complex discrete search spaces to identify optimal or near-optimal solutions for combinatorial problems. We make use of the Archipelago meta-heuristic implemented in `pygmo`. The Archipelago meta-heuristic is based on the concurrent

[5] `https://esa.github.io/pygmo2/overview.html`

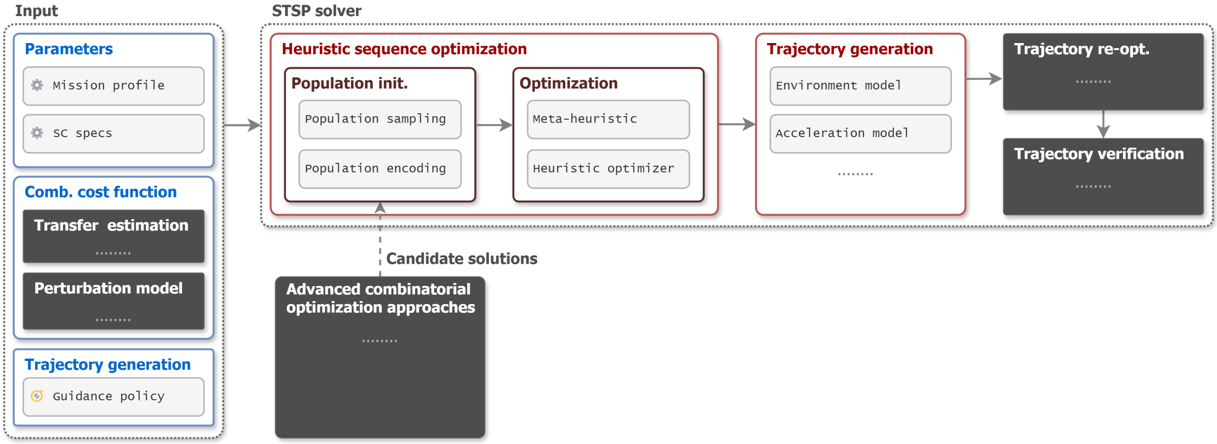


Fig. 4. STSP solver architecture. In black: complex components (integrated using standardized interfaces) the internal structure of which is out of the scope of this diagram.

evolution of sub-populations, or ”islands,” starting from distinct initial populations, and possibly using distinct evolutionary strategies. Periodic migrations of individuals between islands promote diversity and prevent premature convergence, enhancing the algorithm's ability to escape local optima and explore diverse regions of the search space. This parallel and cooperative framework not only accelerates the optimization process but also improves the robustness and quality of the solutions obtained.

Heuristic Optimization Algorithm Optimizer selection is conducted by trading off the performance of all global heuristic optimizers available in pygmo (refer to the pygmo capabilities page[6]) on the STSP variant at hand. Critically, optimizers of the family of Evolutionary Strategies depend on internal sampling processes and are thus not sensitive to initial populations: in general terms other optimizers are preferrable if candidate solutions can be leveraged (this is not the case for OSSIE, as will be shown in Section 7).

5.3 Trajectory Generation

Trajectories are generated by propagating the state of the spacecraft under a guidance policy. The guidance policy is implemented as a state machine that determines when and how to apply thrust through the manoeuvre, following the guidance policy API of the Tudat Space[7] astrodynamics library[38]. This provides ample flexibility for the implementation of the simulator itself. In the case of OSSIE we generate trajectories using a simple analytical propagator, considering impulsive orbital changes and secular J_2 perturbations (see Subsection 4.4). In more complex scenarios we make use of Tudat to implement and refine the simulator used to generate spacecraft trajectories.

5.4 Trajectory re-optimization with Sequential Convex Programming

The final block of the optimization chain is in charge of adapting the ideal transfer maneuvers to spacecraft feasible trajectories. To account for actuation constraints, by introducing the state vector:

$$x = [p, f, g, h, k, L, m]^T \tag{17}$$

as well as the control vector:

$$u = [u_r, u_\theta, u_\phi]^T \tag{18}$$

the following OCP is formulated for each transfer arc:

$$\begin{aligned} \min_{X \in \mathbb{R}^{n_x}, U \in \mathbb{R}^{n_u}} \quad & \frac{1}{2}(x_N - x_{ref})^T P (x_N - x_{ref}) \\ & + \frac{1}{2} U^T R U \\ \text{s.t.} \quad & x_{i+1} = f(x_i, u_i) \quad i \in \{0, \dots, N-1\} \\ & \|u_i\| \leq T_{\max_i} \quad i \in \{0, \dots, N-1\} \\ & x_0 = \hat{x_0} \end{aligned} \tag{19}$$

where N is the length of the optimization horizon, i.e., the number of stages; $n_x = 7(N+1)$ and $n_u = 3N$ the total number of state and control optimization variables, respectively; so that $X = \text{col}(x_0, \cdots, x_N)$ and

[6] https://esa.github.io/pygmo2/overview.html
[7] https://docs.tudat.space/en/latest/

$U = \text{col}(u_0, \cdots, u_{N-1})$; $f(x_i, u_i)$ the discretized non-linear orbital dynamics, $\hat{x_0}$ the initial state and being x_{ref} the reference trajectory, i.e., the optimized arc obtained from the combinatorial problem.

The Euclidean norm of the control input $|| \cdot ||$ is constrained to a maximum thrust $T_{\max_i}$, mission and potentially time-dependent. The penalty of the thrust action is weighted by $R \in \mathbb{R}^{n_u \times n_u}$, $R \succcurlyeq 0$, against the final error $(x_N - x_{ref})$, weighted by matrix $P \in \mathbb{R}^{7\times7}$, $P \succcurlyeq 0$. It is highlighted that here the penalty over the final state error is a relaxation of the hard constraint $x(t_N) = x_N$ with x_N the final state of each arc of trajectory from the combinatorial problem, to avoid unfeasibilities when adapting to spacecraft dynamics and constraints. The TOF between $\hat{x_0}$ and x_N is then used to adapt the optimization horizon. Note that the number of arcs simultaneously optimized determines the authority given to this layer of the framework to modify the transfers, being the case with a single arc focused on replicating the Hohmann arc, autonomously adapting it to the spacecraft constraints.

Finally, note that the OCP in Eq. 19 does not include attitude constraints, e.g., for guaranteeing that the thrust vector can be oriented in time to perform the required impulses. Certain smoothness of the attitude can be argued in the implicit effect of the control cost regularization, yet the formulation is currently under development for explicit attitude constraints inclusion.

5.4.1 Solution with SCP

The resultant generic OCP in Eq. 19 is convex except for the non-linear dynamics implicit in $f(x_i, u_i)$. To solve it, an SCP solver is used from the SOTB[8] library. The algorithm implemented follows a classical SCP logic, solving guess-based convexified iterations of the original problem until a convergence criterion is matched. Dynamic trust regions are employed to control the iteration progress and avoid divergence or solution chattering due to non-linear effects. We refer to [23] for further details on the solver.

To reduce errors arising from numerical discretization and propagation during the iterations of the optimization solver, the same MEEs formulation of Section 3 are adopted in the SCP with a proper discretization step for the horizon length and integrator selection. Furthermore, the optimization variables and the dynamics model have been normalized, i.e., the state variables have been scaled according to the MEE parametrization of the latter point of each optimization horizon, and the control input sequence has been scaled according to the maximum value of the thrust. This normalization moves the optimum region to similar orders of magnitude in states and control, providing a smoother behavior of the SCP internal IPM and smaller linearization errors, aside from easing the algorithm tuning.

5.4.2 OSSIE actuation constraints

Addressing now the particularities of the study case, OSSIE OTV presents a limitation in the thruster actuation that impedes the continuous firing for more than ΔT_{on} seconds, requiring after a minimum off-time for cool-down. To comply with the maximum firing time of the OSSIE thruster actuators, $T_{\max_i}$ shall be set so that after the maximum firing time the constraint is set to zero until the minimum off-time is reached. To simplify the problem and avoid introducing the time as a constrained optimization variable, the input TOF between consecutive ΔV from the combinatorial problem is used to compute $T_{\max_i}$ as sketched in Fig. 5, i.e., forcing $T_{\max_i} = 0$ for transfer periods which are already larger than the minimum off-time.

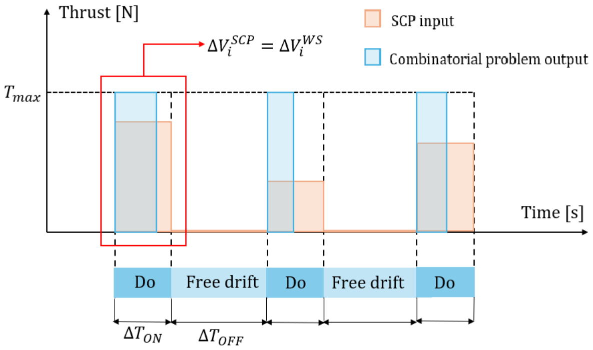


Fig. 5. On-time constraints scheme and warm-starting procedure

Note that this decision enforces the actuation windows based on the combinatorial solution and Hohmann transfers. For multiple arcs optimization, this might create a source of suboptimality. Nonetheless, for the single arc optimization, this does not generate a problem because maintaining the TOF is an objective thus the action window coincides with the perigee or apogee, i.e., the most efficient times to act. This simplification results into a pragmatic solution because few simultaneous arcs optimization is sufficient in the considered mission.

[8] SOTB is an independent in-house toolbox developed in MATLAB devoted for real-time embedded applications. Within available solvers, SCP algorithms focused on nonlinear numerical optimal control powered by Interior Point Method (IPM) solvers are ideal for the developed application.

5.4.3 Multi-impulsive warm starting

Warm starting can significantly reduce the number of required iterations in an SCP algorithm. For the mission case, an efficient initialization is created with the solution of the combinatorial problem. The departure and complete ΔV time sequence are used to automatically adapt the discretization scheme of the trajectory arc(s) optimization and on-time constraints, based on the number of consecutive arcs to be simultaneously optimized (parameter). Then, ΔVs are translated to forces realizing equivalent total pulse but uniformly distributed in ΔT_{on}. Fig. 5 illustrates the warm start creation process. This approach generally situates the initial guess close to the optimum, significantly reducing the required convexification iterations.

6. Reinforcement Learning for Combinatorial Optimization

ML approaches for spacecraft trajectory design have seen a surge of interest in recent years[39], with strong results achieved both for trajectory cost estimation[15] and spacecraft guidance[40]. NCO uses DNNs to automate the problem-solving process, mostly under the RL paradigm, as supervised learning is often unfeasible for large or theoretically hard problems. NCO offers the attractive prospect of alleviating the scaling issues of exact approaches, while removing the need for handcrafted heuristics, which often require significant domain-specific adjustments[41]. NCO has shown promising performance on various combinatorial optimization problems[41], especially when coupled with advanced policy search procedures[16].

We make use of `RL4CO`[9] to implement and train the STSP routing policy. `RL4CO` is a benchmark library for NCO with standardized, modular, and highly performant implementations of various environments, policies and RL algorithms, covering the entire NCO pipeline[41].

6.1 Environment

The space environment is modelled in MEEs. The translational state is extended with the mass of the vehicle and targets. The resulting 7-dimensional state space is embedded using a feedforward NN into a 128-dimensional space in which the policy operates. The global state is propagated through the sequential decision-making process. Environment propagation is implemented by means of a transfer TOF estimation method and perturbation model: in the case of OSSIE these are the sequential MHT-NIC trajectory estimator, and secular J_2 perturbation model defined in Section 4.

6.2 Policy

We make use of an autoregressive attention model[10] first introduced by Kool et al.[42], which encodes the input graph using a GAT, and decodes the solution using a Pointer Network. This policy trained using the REINFORCE RL algorithm has been demonstrated to perform considerably better than other learned heuristics[42], as well as to have a highly efficient learning component[16].

6.3 Reinforcement Learning Algorithms

We make use of three RL algorithms implemented in `RL4CO` to train the policy: REINFORCE, Advantage Actor-Critic, and Proximal Policy Optimization.

Stochastic Policy Gradient The REINFORCE algorithm, introduced by Williams[43], is a stochastic policy gradient method which uses Monte Carlo sampling to estimate policy gradient. By generating complete trajectories and using the returns from these trajectories, REINFORCE provides an unbiased estimate of the gradient. It updates the policy parameters by maximizing the expected reward through gradient ascent. However, REINFORCE often suffers from high variance in gradient estimates, which can hinder convergence.

Advantage Actor-Critic A2C methods enhance REINFORCE by incorporating a baseline to reduce the variance of gradient estimates. Introduced by Konda and Tsitsiklis[44] and popularized in the asynchronous framework by Mnih et al.[45], A2C simultaneously learns a policy (actor) and a value function (critic). The critic estimates the value function, which serves as a baseline to compute the advantage, thereby stabilizing and accelerating training.

Proximal Policy Optimization PPO, introduced by Schulman et al.[46], addresses some of the limitations of A2C by ensuring that policy updates do not deviate excessively from the current policy. PPO employs a clipped objective function to constrain the policy updates, balancing exploration and exploitation more effectively. This leads to more stable and reliable training, making PPO a popular choice for various RL applications, including combinatorial optimization tasks.

[9] `https://rl4.co`

[10] `https://rl4.co/docs/.../AttentionModelPolicy`

Table 2. Top: OSSIE insertion and decommissioning orbits. Bottom: statistical model describing expected payload Keplerian states. In commercial operations the target state of each payload is specified by the client. SSO stands for SSO inclination variance.

	Orbit	a [km]	e [-]	i [deg]	Ω [deg]	ω [deg]	θ [deg]	Mass [kg]
OSSIE	Insertion	500	0	97	158	0	0	
	Decommissioning	250	0	-	-	-	-	
Payload	Nominal	500	0	97	158	0	0	Variable
	Spread	50	0	SSO	180	180	180	15%
	Distribution	Uniform	Exponential	Uniform	Uniform	Uniform	Uniform	Exponential

6.4 Policy Search

Policy search strategies determine how actions are selected based on the learned policy, balancing exploration and exploitation to find optimal or near-optimal solutions. Policy search is fundamental for the performance of NCO algorithms[16]. We consider three policy search strategies implemented in `RL4CO`: greedy search, sampling and beam search.

7. Results

This section presents empirical results obtained for the OSSIE OTV trajectory optimization problem. Subsection 7.1 describes the mission scenarios for OSSIE and presents a statistical model used to randomly generate realistic mission scenarios. In Subsection 7.2 we report and analyze heuristic optimization performance. In Subsection 7.3 we analyze the performance of the RL Attention-based routing policy for this STSP variant.In Subsection 7.4 presents a study of OSSIE's mission design envelope based on a Monte Carlo campaign spanning all feasible mission modes. Subsection 7.5 presents the trajectory re-optimization results obtained using Sequential Convex Programming. Lastly, Subsection 7.6 discusses the verification of the generated trajectories on the OSSIE Functional Engineering Simulator developed at SENER.

7.1 Mission Modelling

Table 2 describes the nominal mission scenario for OSSIE and the statistical model used to generate the target Keplerian states for a given payload. OSSIE payloads are highly likely to be destined for SSO orbits with altitude priority, but not necessarily. A worst case scenario is assumed, uniformly sampling payload inclinations in the entire SSO inclination range, defined as the range from $i_{\text{SSO}}(a_{\text{min}})$ to $i_{\text{SSO}}(a_{\text{max}})$, where $i_{\text{SSO}}(a)$ is the inclination required to achieve an SSO orbit (RAAN drift of 360°per year, see Eq. 8) at a given a. Payload masses are sampled from an exponential distribution, assuming a worst case scenario where payload masses are never below their nominal value: 6 kg for CubeSats, 1.5 kg for PocketQubes and 25 kg for small satellites. Payloads may be released individually or at once.

OSSIE payloads may be released individually or bundled with other payloads. This variable is both client-dependent and so highly unpredictable, and greatly impactful for mission cost as it determines the number of transfers that must be carried out. We model this by uniformly sampling number of bundles between 2 (all payloads in two bundles) and the total number of payloads, which is 13 for OSSIE (every payload launched to a distinct target orbit); payloads are assigned to each bundle uniformly at random.

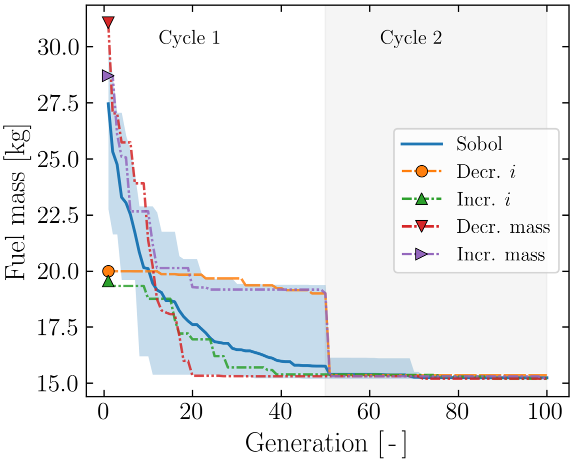


Fig. 6. Best fuel mass achieved as a function of generation (13-transfer mission, GPSO). Shaded regions: evolutionary cycles.

Table 3. Policy test performance and training time for the three RL algorithms considered. BS: Beam Search.

RL algorithm	REINFORCE			A2C			PPO		
Search strategy	Greedy	Stoch.	BS	Greedy	Stoch.	BS	Greedy	Stoch.	BS
Fuel mass [kg]	21.65	22.58	**20.67**	24.31	25.51	22.93	22.25	22.98	21.14
Optimality gap [%]	7.86	12.53	**3.02**	21.17	27.15	14.28	10.86	14.52	5.35
Training time [min]	15.1	-	-	**9.8**	-	-	30.6	-	-

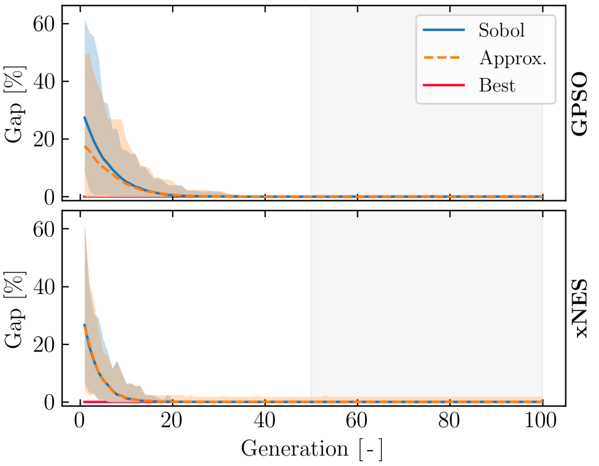


Fig. 7. Optimization of 100 randomized 13-transfer OSSIE mission scenarios using GPSO and xNES. Best: on average 26.72 [kg] of fuel mass spent with a standard deviation of 2.25 [kg]. Shaded regions: evolutionary cycles.

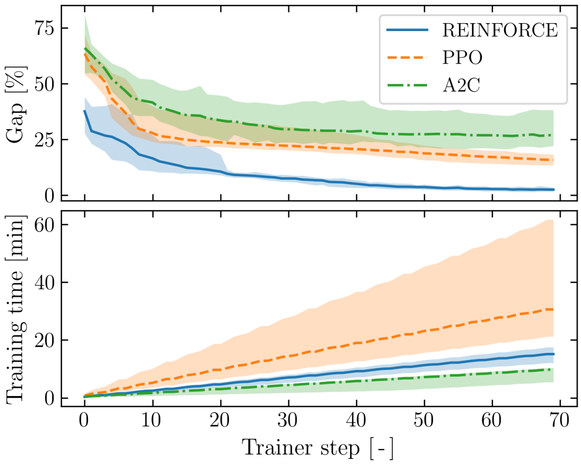


Fig. 8. Training validation reward curve (expressed as a mean optimality gap) with REINFORCE, A2C and PPO.

7.2 Heuristic Combinatorial Optimization

We find that GPSO[11] and xNES[12] perform best in the case of OSSIE. Fig. 6 shows the evolutionary curve of a realistic 13-transfer mission scenario for OSSIE using GPSO, which is sensitive to initial populations. Fig. 7 shows the evolutionary curve of 100 randomized 13-transfer mission scenarios.

Four hand-crafted heuristics are used to provide candidate solutions: ascending and descending inclination and payload mass walks. The increasing inclination walk is a fairly good heuristic in Fig. 6, but this fails to replicate over more cases, with optimizations leveraging the four heuristics performing very similarly to those starting with uniformly sampled populations in Fig. 7. xNES outperforms GPSO in this case, exhibiting more uniform convergence. This dynamic only increases with problem size: improved heuristics are highly desirable, and more so as problem size increases, as is likely to be the case through the operational life of OSSIE.

7.3 Reinforcement Learning

We train the Attention policy on 100,000 10-transfer mission scenarios using a batch size of 5096, for 50 epochs. The policy is tested on 1000 mission scenarios. Near-optimal solutions for all scenarios are obtained a priori using heuristic optimization, and used as a benchmark to calculate optimality gaps. Training is repeated 10 times with each algorithm using different random seeds.

Table 3 reports mean policy training times and performances obtained with REINFORCE, A2C and PPO. Fig. 8 shows the corresponding learning curves. REINFORCE yields the best training out of the three RL algorithms, with a mean optimality with respect to the heuristic solutions of 3.02% using Beam Search. As expected[16] Beam Search improves the final result compared to the greedy and sampling searches. The policy's performance

[11] `https://esa.github.io/pygmo2/.../pso_gen`
[12] `https://esa.github.io/pygmo2/.../xnes`

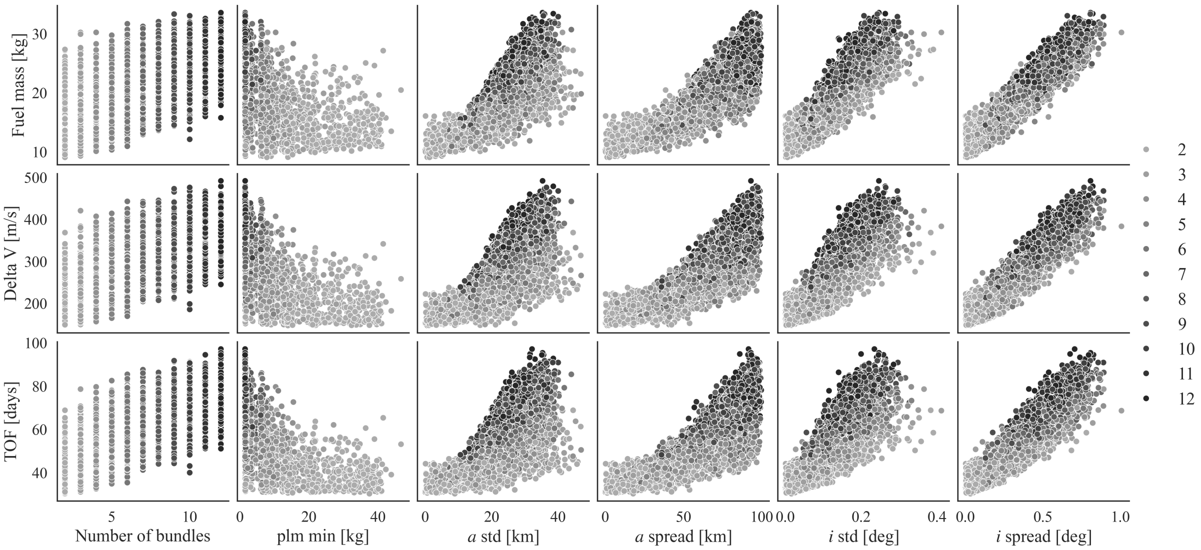


Fig. 9. Cost of 5000 optimized mission scenarios. Fuel mass, ΔV and TOF are plotted against, from left to right: number of bundles, minimum payload mass, semi-major axis standard deviation and range, and inclination standard deviation and range. Observe the strong correlations between number of bundles and cost, and between inclination spread and cost.

is superior to that of the hand-crafted heuristics considered for OSSIE (Fig. 7).

Having a reliably performant learned heuristic is considerably beneficial for combinatorial optimization (see Fig. 6); in the sections that follow we use GACO with initial populations determined by the policy to optimize OSSIE mission scenarios.

7.4 Mission Analysis

We proceed to perform a Monte Carlo analysis covering all feasible mission scenarios based on the nominal payload list of OSSIE of 8 CubeSats, 4 PocketQubes and 1 small satellite. 5000 mission scenarios are solved and reported.

Fig. 9 shows tour cost in terms of fuel mass consumption, ΔV required and TOF as a function of number of bundles, minimum payload mass, semi-major axis standard deviation and range, and inclination standard deviation and range. As expected, number of bundles and inclination range is the greatest driver of mission cost. Fig. 10 desegregates fuel consumption by number of bundles, which is clearly the greatest driver of mission cost. In all cases OSSIE is, on average, capable of fulfilling its mission and decommissioning afterwards.

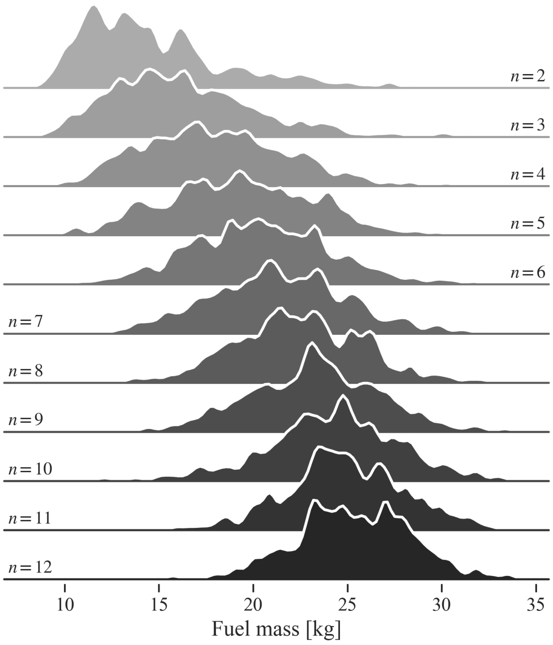


Fig. 10. Fuel consumption as a function of number of bundles.

Table 4. Summary of OSSIE scenarios under analysis: 1) coplanar transfer, 2) noncoplanar transfer, 3) inclination change (0.25°), 4) large inclination change (1°).

Test Case	TOF [hr]	a_{t_i} [km]	a_{t_f} [km]	i_{t_i} [deg]	i_{t_f} [deg]
Case 1	163.45	6950	7000	97.3964	97.3964
Case 2	263.93	6950	7000	97.2714	97.5214
Case 3	99.57	7000	7000	97.2714	97.5214
Case 4	192.87	7000	7000	96.8964	97.8964

Table 5. Comparison between the SCP manoeuvre arrival orbit and the desired target orbit.

Test Case	TOF [hr]	ΔV_{SCP} [m/s]	ΔV_{combin} [m/s]	ΔV_{reduct} [%]	Δa_{target} [km]	Δe_{target} [deg]	Δi_{target} [-]
Case 1	163.37	29.59	27.09	9.20	1.54	-0.0007	0.0007
Case 2	262.43	29.91	32.71	-8.53	0.06	0.0116	0.0537
Case 3	99.47	32.71	32.71	0.00	1.57	0.0167	0.0040
Case 4	192.87	132.42	132.72	0.22	0.05	0.0052	0.0144

7.5 SCP

To validate the SCP algorithm for trajectory re-optimization and adaptation to spacecraft constraints, a selection of orbit transfers of the previous section are re-optimized. Table 4 gather the test cases and Table 5 gathers the errors computed as SCP result with respect to target and the change in ΔV with respect to combinatorial problem solution.

The results show that the SCP manages to adapt the transfer manoeuvres to the spacecraft constraints with minimum impact over injection errors or ΔV. Note that the SCP is tuned to prioritize the mission objectives and encourage that output results meet the orbit injection accuracy requirements ($\Delta a \leq 10[km], \Delta i \leq 0.1°$). This might result in an increment of the overall ΔV cost if compared to the combinatorial solution, as seen in cases 1 and 3 of Table 5, even if control cost is minimized.

For the non-coplanar scenarios, i.e., change of height followed by change of altitude, the SCP manages to reduce the propellant consumption of the 8.5357% if compared with the combinatorial approximation. Note, however, that complete eccentricity elimination is not achieved: the latter can be solved, if required, by orbit circularization procedures, which would hinder propellant cost reductions gained by the SCP.

In general, we have observed that the current tuning benefits from a division of transfer arcs without mixing inclination and altitude change objectives simultaneously. This might be driven by the warm starting that assumes such specific procedure. Other warm starting strategies or SCP tunings are currently being explored for further propellant minimization by combination of objective changes. A 3D overview of the optimized transfer maneuvre is shown in Figure 11.

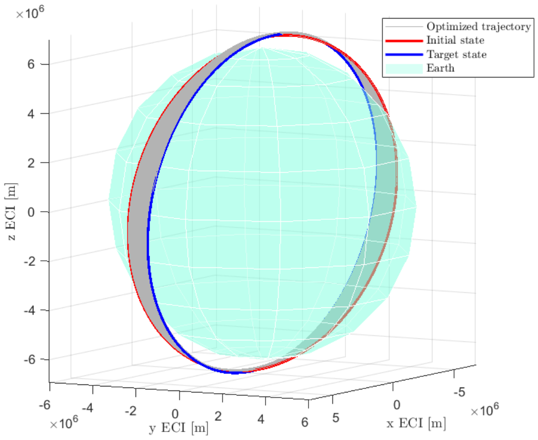


Fig. 11. SCP-tailored noncoplanar maneuver (corresponding to the Test Case 2 in Table 4)

7.6 Verification in High-fidelity Simulator

The SENER OSSIE FES high-fidelity simulation environment serves as the core framework for modelling, testing and verifying. Matlab Simulink[47] is used to handle simulation, while Matlab[48] is used for pre- and post-processing tasks.

Translational dynamics are simulated using the Cowell propagator. Five natural perturbations are considered in the acceleration model: non-spherical Earth effects accounting for zonal harmonics of up to degree 6, atmospheric drag (density obtained from the U.S. Standard Atmosphere model), third-body perturbations from the Sun and Moon and solar radiation pressure. Spacecraft mass is propagated using a constant I_{sp} mass propagator.

Attitude dynamics are modeled using Euler's equation, which incorporates the effects of external torques: as OSSIE makes no use of internal momentum devices, internal torques are not considered in this analysis as they have a negligible magnitude in the actual spacecraft configuration. The key external torques considered are gravity gradient (modeled using the J2 gravitational coefficient), magnetic torques (perturbation by means of residual magnetic dipole moment), aerodynamic drag, and solar radiation pressure. Other potential torque sources, such as control system torques by means of the reaction control system, are modeled according to the precise thrust curves provided by the actuator constructed models.

The simulation is integrated using a fixed-step 4th order Runge-Kutta integrator running at 200 Hz (timestep of 5 ms), enabling accurate system characterization while maintaining sufficient simulation speed, to perform consequent testing campaigns.

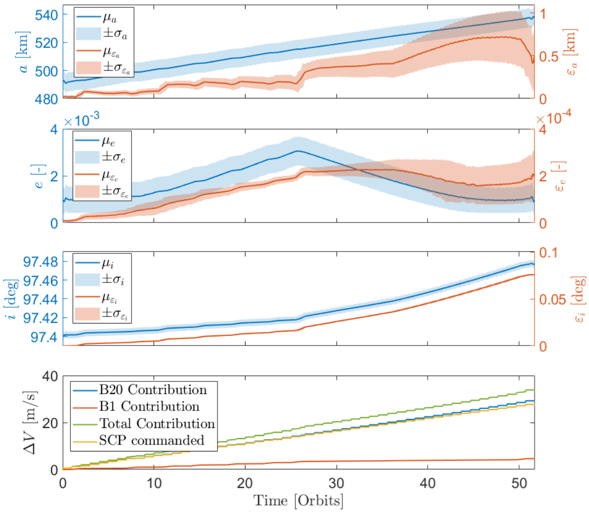


Fig. 12. Preliminary verification results of case 1. Top 3 panels: moving averages, 30 minute window.

The results of the preliminary verification of case 1 in Table 4 in the FES are presented in Fig. 12: the first upper three plots depict the evolution in time of the orbital parameters during the simulation (in blue), demonstrating that the optimized trajectory was correctly followed, as contrasted by the error with respect to the SCP (in orange). The test shows that the achieved error range is compliant with client requirements. The last subplot presents a comparison between the ΔV consumption achieved by the FES and the SCP: the additional ΔV requested by GNC results from pointing correction maneuvres, to allign the B20 thrusters with the thrusting direction computed in the optimization framework.

8. Conclusion

We have presented a framework for addressing the Multi-target Rendezvous problem, applied to the UARX Space OSSIE mission. The framework determines optimal sequences for visiting multiple targets and generates near fuel-optimal trajectories. By integrating an Attention-based routing policy trained with Reinforcement Learning, the combinatorial optimization process is enhanced, demonstrating the effectiveness of RL in spacecraft routing. The framework accommodates the propulsion requirements of the OSSIE system, ensuring high modularity and adaptability to mission-specific constraints. This tool provides feasible and optimal guidance with minimal design effort and allows for future extensions, including the incorporation of more complex thrust profiles such as low-thrust scenarios. The proposed approach advances space vehicle routing methodologies and supports further development using reinforcement learning and advanced optimization techniques.

Acknowledgements

This work is the result of a collaborative effort between the Delft University of Technology and SENER Aerospace & Defence. We would like to thank Marc Naeije from TU Delft for his endorsement of the project and his keen supervision and advice. Lastly, we would like to extends our thanks to the entire team at UARX Space for giving us the opportunity to be a part of the OSSIE mission. Per aspera ad astra.

References

[1] D. Izzo, I. Getzner, D. Hennes, and L. Simões, “Evolving Solutions to TSP Variants for Active Space Debris Removal: Genetic and Evolutionary Computation Conference (GECCO),” *Proceedings of the 17th annual conference on Genetic and evolutionary computation (GECCO 2015)*, S. Silva, Ed., pp. 1207–1214, 2015, Publisher: ACM Press, ISSN: 9781450334723. DOI: 10.1145/2739480.2754727.

[2] L. Ricciardi and M. Vasile, *Solving multi-objective dynamic travelling salesman problems by relaxation*. Jul. 2019, Pages: 2007. doi: 10.1145/3319619.3326837.

[3] L. Federici, A. Zavoli, and G. Colasurdo, "Evolutionary Optimization of Multirendezvous Impulsive Trajectories," *International Journal of Aerospace Engineering*, vol. 2021, pp. 1–19, May 2021. doi: 10.1155/2021/9921555.

[4] L. Medioni *et al.*, "Trajectory optimization for multi-target Active Debris Removal missions," *Advances in Space Research*, Space Environment Management and Space Sustainability, vol. 72, no. 7, pp. 2801–2823, Oct. 2023, issn: 0273-1177. doi: 10.1016/j.asr.2022.12.013. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S027311772201105X (visited on 09/18/2024).

[5] A. Barea, H. Urrutxua, and L. Cadarso, "Large-scale object selection and trajectory planning for multi-target space debris removal missions," *Acta Astronautica*, vol. 170, pp. 289–301, May 2020, issn: 0094-5765. doi: 10.1016/j.actaastro.2020.01.032. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0094576520300436 (visited on 09/18/2024).

[6] S. Narayanaswamy, B. Wu, P. Ludivig, F. Soboczenski, K. Venkataramani, and C. J. Damaren, "Low-thrust rendezvous trajectory generation for multi-target active space debris removal using the RQ-Law," *Advances in Space Research*, vol. 71, no. 10, pp. 4276–4287, May 2023, issn: 0273-1177. doi: 10.1016/j.asr.2022.12.049. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0273117722011656 (visited on 09/22/2024).

[7] C. P. Mark and S. Kamath, "Review of Active Space Debris Removal Methods," *Space Policy*, vol. 47, pp. 194–206, Feb. 2019, issn: 0265-9646. doi: 10.1016/j.spacepol.2018.12.005. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0265964618300110 (visited on 09/18/2024).

[8] C. Bonnal, J.-M. Ruault, and M.-C. Desjean, "Active debris removal: Recent progress and current trends," *Acta Astronautica*, vol. 85, pp. 51–60, Apr. 2013, issn: 0094-5765. doi: 10.1016/j.actaastro.2012.11.009. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0094576512004602 (visited on 09/18/2024).

[9] F. Sellmaier, T. Boge, J. Spurmann, S. Gully, T. Rupp, and F. Huber, "On-Orbit Servicing Missions: Challenges and Solutions for Spacecraft Operations," en, in *SpaceOps 2010 Conference*, Huntsville, Alabama: American Institute of Aeronautics and Astronautics, Apr. 2010, isbn: 978-1-62410-164-9. doi: 10.2514/6.2010-2159. [Online]. Available: https://arc.aiaa.org/doi/10.2514/6.2010-2159 (visited on 09/18/2024).

[10] T. Sarton du Jonchay, H. Chen, M. Isaji, Y. Shimane, and K. Ho, "On-Orbit Servicing Optimization Framework with High- and Low-Thrust Propulsion Tradeoff," *Journal of Spacecraft and Rockets*, vol. 59, no. 1, pp. 33–48, 2022, Publisher: American Institute of Aeronautics and Astronautics _eprint: https://doi.org/10.2514/1.A35094, issn: 0022-4650. doi: 10.2514/1.A35094. [Online]. Available: https://doi.org/10.2514/1.A35094 (visited on 09/18/2024).

[11] S. E. Sorenson and S. G. N. Pinkley, "Multi-orbit routing and scheduling of refuellable on-orbit servicing space robots," *Computers & Industrial Engineering*, vol. 176, p. 108 852, Feb. 2023, issn: 0360-8352. doi: 10.1016/j.cie.2022.108852. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0360835222008403 (visited on 09/18/2024).

[12] D. Morante, M. Sanjurjo Rivo, and M. Soler, "A Survey on Low-Thrust Trajectory Optimization Approaches," *Aerospace*, vol. 8, no. 3, 2021, issn: 2226-4310. doi: 10.3390/aerospace8030088. [Online]. Available: https://www.mdpi.com/2226-4310/8/3/88.

[13] A. Petropoulos *et al.*, "GTOC9: Methods and Results from the Jet Propulsion Laboratory Team," May 2017.

[14] S. Lu *et al.*, "GTOC 11: Results found at Beijing Institute of Technology and China Academy of Space Technology," *Acta Astronautica*, vol. 202, pp. 876–888, Jan. 2023, issn: 0094-5765. doi: 10.1016/j.actaastro.2022.07.039. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0094576522003885 (visited on 09/18/2024).

[15] H. Li, S. Chen, D. Izzo, and H. Baoyin, "Deep networks as approximators of optimal low-thrust and multi-impulse cost in multitarget missions," *Acta Astronautica*, vol. 166, pp. 469–481, Jan. 2020,

ADS Bibcode: 2020AcAau.166..469L, ISSN: 0094-5765. DOI: 10.1016/j.actaastro.2019.09.023. [Online]. Available: https://ui.adsabs.harvard.edu/abs/2020AcAau.166..469L (visited on 09/18/2024).

[16] A. Francois, Q. Cappart, and L.-M. Rousseau, "How to Evaluate Machine Learning Approaches for Combinatorial Optimization: Application to the Travelling Salesman Problem," Sep. 2019. [Online]. Available: https://arxiv.org/abs/1909.13121.

[17] M. Hallmann *et al.*, *GTOC 9, Multiple Space Debris Rendezvous Trajectory Design in the J2 environment*. Jun. 2017.

[18] F. Biscani and D. Izzo, "A parallel global multiobjective framework for optimization: Pagmo," en, *Journal of Open Source Software*, vol. 5, no. 53, p. 2338, Sep. 2020, ISSN: 2475-9066. DOI: 10.21105/joss.02338. [Online]. Available: https://joss.theoj.org/papers/10.21105/joss.02338 (visited on 09/18/2024).

[19] J. Blank and K. Deb, "Pymoo: Multi-Objective Optimization in Python," *IEEE Access*, vol. 8, pp. 89 497–89 509, 2020, Conference Name: IEEE Access, ISSN: 2169-3536. DOI: 10.1109/ACCESS.2020.2990567. [Online]. Available: https://ieeexplore.ieee.org/document/9078759 (visited on 09/18/2024).

[20] P. L. X. Liu, "Solving non-convex optimal control problems by convex optimization," *Journal of Guidance, Control, and Dynamics*, vol. 37, no. 3, pp. 750–765, 2014.

[21] S. B. A. Domahidi E. Chu, "Ecos: An socp solver for embedded systems," *European Control Conference (ECC)*, 2013.

[22] F. Y. H. R. Foust S. Chung, "Optimal guidance and control with nonlinear dynamics using sequential convex programming," *Journal of Guidance Control & Dynamics*, vol. 43, no. 4, pp. 633–644, 2020.

[23] L. H. J. Ramirez, "Sequential convex programming for optimal line of sight steering in agile missions," 9^{th} *European Conference for Aeronautics and Aerospace Sciences (EUCASS)*, 2022.

[24] F. T. C. Hofmann, "Rapid low-thrust trajectory optimization in deep space based on convex programming," *Journal of Guidance, Control, and Dynamics*, 2021.

[25] S. J. W. J. Nocedal, *Numerical Optimization*. New York, NY, USA: Springer, 2^{nd} edition, 2006.

[26] E. S. D. M. W. Group, *ESA Space Debris Mitigation Requirements*, English, Oct. 2023. [Online]. Available: https://technology.esa.int/upload/media/DGHKMZ_6542582e18e33.pdf (visited on 09/23/2024).

[27] G. Hintz, "Survey of Orbit Element Sets," *Journal of Guidance Control and Dynamics - J GUID CONTROL DYNAM*, vol. 31, pp. 785–790, May 2008. DOI: 10.2514/1.32237.

[28] K. T. Alfriend, S. R. Vadali, P. Gurfil, J. P. How, and L. S. Breger, "Chapter 2 - Fundamental Astrodynamics," in *Spacecraft Formation Flying*, K. T. Alfriend, S. R. Vadali, P. Gurfil, J. P. How, and L. S. Breger, Eds., Oxford: Butterworth-Heinemann, Jan. 2010, pp. 13–38, ISBN: 978-0-7506-8533-7. DOI: 10.1016/B978-0-7506-8533-7.00207-4. [Online]. Available: https://www.sciencedirect.com/science/article/pii/B9780750685337002074 (visited on 09/18/2024).

[29] K. Wakker, *Fundamentals of Astrodynamics*. TU Delft Library, Jan. 2015, ISBN: 978-94-6186-419-2. [Online]. Available: https://repository.tudelft.nl/islandora/object/uuid%5C%3A3fc91471-8e47-4215-af43-718740e6694e.

[30] R. Mitchell, J. Cooper, E. Frank, and G. Holmes, "Sampling Permutations for Shapley Value Estimation," *Journal of Machine Learning Research*, vol. 23, no. 43, pp. 1–46, 2022, ISSN: 1533-7928. [Online]. Available: http://jmlr.org/papers/v23/21-0439.html (visited on 09/19/2024).

[31] M. Eberl, "Fisher-Yates shuffle," *Arch. Formal Proofs*, 2016. [Online]. Available: https://www.semanticscholar.org/paper/Fisher-Yates-shuffle-Eberl/5e24ffebdc35e8e11af823505cbd5c6d5407f23e (visited on 09/19/2024).

[32] P. Diaconis, "Group Representations in Probability and Statistics," *Lecture Notes-Monograph Series*, vol. 11, pp. i–192, 1988, Publisher: Institute of Mathematical Statistics, ISSN: 0749-2170. [Online]. Available: https://www.jstor.org/stable/4355560 (visited on 09/19/2024).

[33] C. L. Mallows, "Non-Null Ranking Models. I," *Biometrika*, vol. 44, no. 1/2, pp. 114–130, 1957, Publisher: [Oxford University Press, Biometrika Trust], ISSN: 0006-3444. DOI: 10.2307/2333244. [Online]. Available: https://www.jstor.org/stable/2333244 (visited on 09/19/2024).

[34] E. Irurozki, "Sampling and learning distance-based probability models for permutation spaces," es, Ph.D. dissertation, Universidad del País Vasco - Euskal Herriko Unibertsitatea, 2014. [Online]. Available: `https://dialnet.unirioja.es/servlet/tesis?codigo=213087` (visited on 09/19/2024).

[35] J. C. Bean, "Genetic Algorithms and Random Keys for Sequencing and Optimization," *ORSA Journal on Computing*, vol. 6, no. 2, pp. 154–160, May 1994, Publisher: ORSA, ISSN: 0899-1499. DOI: `10.1287/ijoc.6.2.154`. [Online]. Available: `https://pubsonline.informs.org/doi/abs/10.1287/ijoc.6.2.154` (visited on 09/19/2024).

[36] P. Krömer, V. Uher, and V. Snášel, "Novel Random Key Encoding Schemes for the Differential Evolution of Permutation Problems," *IEEE Transactions on Evolutionary Computation*, vol. 26, no. 1, pp. 43–57, Feb. 2022, Conference Name: IEEE Transactions on Evolutionary Computation, ISSN: 1941-0026. DOI: `10.1109/TEVC.2021.3087802`. [Online]. Available: `https://ieeexplore.ieee.org/document/9449662` (visited on 09/19/2024).

[37] M. A. Londe, L. S. Pessoa, C. E. Andrade, and M. G. C. Resende, "Biased random-key genetic algorithms: A review," *European Journal of Operational Research*, Mar. 2024, ISSN: 0377-2217. DOI: `10.1016/j.ejor.2024.03.030`. [Online]. Available: `https://www.sciencedirect.com/science/article/pii/S0377221724002303` (visited on 09/20/2024).

[38] D. Dirkx *et al.*, "The open-source astrodynamics Tudatpy software - overview for planetary mission design and science analysis," en, *European Planetary Science Congress*, EPSC2022–253, Sep. 2022. DOI: `10.5194/epsc2022-253`. [Online]. Available: `https://ui.adsabs.harvard.edu/abs/2022EPSC...16..253D/abstract` (visited on 09/19/2024).

[39] D. Izzo, M. Märtens, and B. Pan, "A survey on artificial intelligence trends in spacecraft guidance dynamics and control," en, *Astrodynamics*, vol. 3, no. 4, pp. 287–299, Dec. 2019, ISSN: 2522-0098. DOI: `10.1007/s42064-018-0053-6`. [Online]. Available: `https://doi.org/10.1007/s42064-018-0053-6` (visited on 09/18/2024).

[40] D. Izzo, E. Ozturk, and M. Märtens, *Interplanetary Transfers via Deep Representations of the Optimal Policy and/or of the Value Function*. Apr. 2019. DOI: `10.48550/arXiv.1904.08809`.

[41] F. Berto *et al.*, *RL4CO: An Extensive Reinforcement Learning for Combinatorial Optimization Benchmark*, arXiv:2306.17100 [cs], Jun. 2024. DOI: `10.48550/arXiv.2306.17100`. [Online]. Available: `http://arxiv.org/abs/2306.17100` (visited on 09/18/2024).

[42] W. Kool, H. van Hoof, and M. Welling, *Attention, Learn to Solve Routing Problems!* arXiv:1803.08475 [cs, stat], Feb. 2019. DOI: `10.48550/arXiv.1803.08475`. [Online]. Available: `http://arxiv.org/abs/1803.08475` (visited on 09/20/2024).

[43] R. J. Williams, "Simple statistical gradient-following algorithms for connectionist reinforcement learning," en, *Machine Learning*, vol. 8, no. 3, pp. 229–256, May 1992, ISSN: 1573-0565. DOI: `10.1007/BF00992696`. [Online]. Available: `https://doi.org/10.1007/BF00992696` (visited on 09/20/2024).

[44] V. Konda and J. Tsitsiklis, "Actor-Critic Algorithms," in *Advances in Neural Information Processing Systems*, vol. 12, MIT Press, 1999. [Online]. Available: `https://proceedings.neurips.cc/paper_files/paper/1999/hash/6449f44a102fde848669bdd9eb6b76fa-Abstract.html` (visited on 09/20/2024).

[45] V. Mnih *et al.*, *Asynchronous Methods for Deep Reinforcement Learning*, arXiv:1602.01783 [cs], Jun. 2016. DOI: `10.48550/arXiv.1602.01783`. [Online]. Available: `http://arxiv.org/abs/1602.01783` (visited on 09/20/2024).

[46] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, *Proximal Policy Optimization Algorithms*, arXiv:1707.06347 [cs], Aug. 2017. DOI: `10.48550/arXiv.1707.06347`. [Online]. Available: `http://arxiv.org/abs/1707.06347` (visited on 09/20/2024).

[47] T. M. Inc., *Simulation and Model-Based Design*, Natick, Massachusetts, United States, 2023. [Online]. Available: `https://mathworks.com/products/simulink.html`.

[48] T. M. Inc., *MATLAB version: 23.2.0 (R2023b)*, Natick, Massachusetts, United States, 2023. [Online]. Available: `https://www.mathworks.com`.

6 Conclusions and Recommendations

The aim of the present work was to address two challenges in the design of multi-target rendezvous missions: the application of Neural Combinatorial Optimization (NCO) methods for Active Debris Removal (ADR) missions and the integration of verifiable trajectory optimization techniques for the design of Orbit Transfer Vehicle (OTV) payload deployment multi-rendezvous missions. The main conclusions from this work are laid out in Section 6.1. A second result of this work is a set of recommendations for the design of multi-rendezvous missions, and for the further research of Neural Combinatorial Optimization approaches to space Vehicle Routing Problems (VRPs). These recommendations are offered in Section 6.2. The work is wrapped with a final assessment of the research questions that motivated it, in Section 6.3, and an assessment of compliance with the stated goals of the project, in Section 6.4.

6.1 Conclusions

In evaluating NCO methods for ADR, an Attention-based routing policy incorporating a Graph Attention Network (GAT) and a Pointer Network (PN) was implemented. The policy was trained using Reinforcement Learning (RL) algorithms, specifically REINFORCE, Advantage Actor-Critic (A2C), and Proximal Policy Optimization (PPO). The A2C algorithm demonstrated superior performance. Hyperparameter analysis identified embedding dimension and the number of encoder layers as critical factors influencing model performance. The trained policy effectively solved missions with 10 transfers, achieving an optimalit gap of 32% compared to Heuristic Combinatorial Optimization (HCO) solutions, and outperforming the Dynamic RAAN Walk (DRW) heuristic both in terms of mission cost and the time required to generate tours. However, performance did not translate to missions with 30 and 50 transfers, indicating limited generalization capabilities beyond the training conditions.

Regarding trajectory optimization for OTV payload deployment, a modular framework combining HCO, trajectory re-optimization with Sequential Convex Programming (SCP), and trajectory verification and validation was designed and implemented. Applied to the UARX Space OSSIE mission, this framework successfully determined optimal target sequences and generated near fuel-optimal trajectories for OSSIE. An Attention-based routing policy was trained on the OSSIE STSP scenario. In this near-static case —as RAAN is not targeted— the policy performed considerably better than handcrafted heuristics, achieving a 3% optimality gap with respect to the optimum. This demonstrates the feasibility of applying NCO methods in simpler —less dynamic— spacecraft VRPs, such as payload deployment scenarios. The framework's modularity ensured adaptability to mission-specific constraints and facilitated future extensions, such as incorporating low-thrust propulsion profiles.

Overall, the research confirms that NCO methods are applicable and can be effective to learn routing policies for space VRPs, particularly for short ADR missions with a limited number of targets, and for near-static mission scenarios where RAAN convergence is not required. Furthermore, NCO policies can be efficiently evaluated to generate tours and select individual targets to visit, offering interesting possibilities for vehicle autonomy. However, their scalability and generalization to more complex missions with a higher number of transfers remain challenges. The integration of verifiable trajectory optimization techniques with advanced routing policies presents a viable approach for efficient and adaptable mission planning, although further advancements are necessary to enhance performance in larger-scale scenarios.

6.2 Recommendations

This section offers recommendations for the design of multi-rendezous missions, and for the development of NCO methods, and ML methods more generally, for space VRPs.

6.2.1 Recommendations for Mission Design

Active Debris Removal in the Spaceship Era

Schedule to perform its first Moon landing in September 2026 and possibly entering service in late 2024, Spaceship[1] will be the world's most powerful launch vehicle ever developed, capable of carrying up to 150 metric tonnes fully reusable and 250 metric tonnes expendable. This enormous carrying capacity will upend the economy of spacecraft design, and will likely cause a continental drift away from mass-conscious spacecraft development. With large scale in-orbit refueling already in the horizon, Spaceship will bring missions that are completely unfeasible today from the realm of imagination —and academic research— to that of real engineering possibility.

ADR and OTV missions are in a good position under this new space launch paradigm. Extremely high-tonnage, monolithic launches pose a boon for the fine-grained orbit insertion services offered by OTVs. As regards the design of ADR missions, the possibility of in-orbit refueling and component re-stocking offers large opportunity. We recommend research into the economic viability of persistent ADR operations in Low Earth Orbit using multiple spacecraft, considering in-orbit refueling and in-orbit re-stocking of debris de-orbiting payloads. As regards the role of NCO algorithms, we believe that NCO algorithms will become more applicable and useful —not less— as uncertainty and multi-agent interactions become more relevant in ADR mission design.

6.2.2 Methods and Future Research

Neural Combinatorial Optimization Methods for Space Vehicle Routing Problems

As far as NCO methods are concerned, future research should focus on refining NCO model architectures to better handle dynamic VRPs, and to achieve generalization to unseen numbers of targets. Recently developed models aiming to establish foundational models for CO, such as RouteFinder by Berto et al. [125], are attractive research targets. We recommend research into methods better suited to capture the dynamicity of the problem [126], as well as ways to model multiple-spacecraft ADR mission scenarios with path constraints.

Alternative Machine Learning Approaches

In terms of alternative approaches, we recommend research into Deep RL algorithms incorporating tree search strategies, such as Monte Carlo Tree Search (MCTS) [127].

Automatic Design of Model Architectures and Reinforcement Learning Algorithms

The automatic design of ML models and optimal RL algorithms, known as AutoML and AutoRL[2], is critical to improve model performance and robustness. Most commonly this process consists in the optimization of the hyperparameters that define ML model architectures and RL algorithms. Parker and Holder conducted a thorough review of the field in 2022 [128]. Random search and grid search approaches are commonly used for hyperparameter optimization in literature [15], [128]. Bayesian optimization has seen wide use in the field to improve over the former naïve approaches [128]. Hyperband is a successful AutoML algorithm based on tree searches and a resource allocation pruning strategy [129]. Hyperband is widely available through the Optuna [130], Ray Tune [131] and Keras [132] among other libraries. It has been succesfully applied for hyperparameter optimization both in ML and RL settings [133]. State-of-the art approaches combine Bayesian optimization and tree search approaches. This concept was introduced by Falkner et al. with the BOHB[3] [134] algorithm. The development of this family of algorithms is a highly active research area [135]. As of the time of writing the

[1] https://www.spacex.com/vehicles/starship/

[2] https://autorl.org/

[3] https://www.automl.org/blog_bohb/

DEHB[4] [136] and SMAC3[5] [137] algorithms are the considered state-of-art. Advanced algorithms are often used in distributed optimization frameworks to improve optimization performance, such as Ray Tune[6] [131].

6.3 Final Assessment of Research Questions

Code	Question	Answer	References
RQ.1	*Can NCO methods be used to learn effective routing policies for space VRPs?*	NCO methods can be applied to space VRPs. NCO can learn heuristics which are more efficient and more effective than the considered alternatives. However, NCO policies do are found to fail in generalizing to missions with different numbers of targets than those seen in training.	Chapter 4
RQ.2	*What is a suitable approach for the design of multi-rendezvous trajectories with strict feasibility guarantees?*	We propose a 3-stage STSP trajectory optimization framework integrating Combinatorial Optimization, trajectory re-optimization and trajectory verification and validation. This framework is tested to design and verify impulsive trajectories for the OSSIE OTV payload deployment STSP variant.	Chapter 5
SQ.1	*Can NCO methods be used to improve the performance of state-of-the-art solvers for space VRPs?*	NCO methods do not at the moment outperform state-of-the-art Combinatorial Optimization methods. NCO methods can be used effectively to generate approximate solutions to speed up the HCO process. The Attention NCO policy studied in this work, however, does not generalize to mission scenarios with different numbers of nodes than those seen in training. NCO policies with robust generalization capabilities are required before NCO approaches can be recommended for integration in practical space VRP solvers.	Chapter 4
SQ.1.a	*What is a suitable NCO architecture to learn routing policies for space VRPs?*	The architectures championed by current literature consist of Graph Neural Networks (GNNs) trained using Reinforcement Learning. The Attention based GNNs introduced by Kool, et al., consisting of a GAT and a PN, is used in this work.	Chapter 4
SQ.1.b	*What are the optimal training procedures for NCO agents designed to solve the STSP?*	Advantage Actor-Critic (A2C) yields best results for highly dynamic STSPs, outperforming REINFORCE and Proximal Policy Optimization (PPO). In the case of the near-static OSSIE payload deployment STSP, REINFORCE yields the best results. A2C is recommended for future practitioners. Further research into improving the performance of PPO is warranted as well, as the algorithm has a record of very strong performance which belies the results observed in this work.	Chapter 4

Continued on next page

[4] https://automl.github.io/DEHB/latest/
[5] https://automl.github.io/SMAC3/v2.1.0/
[6] https://docs.ray.io/en/latest/tune/index.html

Code	Question	Answer	References
SQ.1.c	*Can NCO methods yield improved performance with respect to state-of-the-art HCO algorithms for the STSP?*	NCO methods do not at the moment outperform state-of-the-art Combinatorial Optimization methods. The Attention NCO policy considered in this work approaches the performance of Heuristic Combinatorial Optimization methods only for the short sequence, relatively unperturbed target deployment STSP of the OSSIE OTV.	Chapter 4
SQ.1.d	*Can NCO methods be leveraged to improve the performance and speed of HCO algorithms?*	NCO methods can be used effectively to generate approximate solutions to speed up the HCO process. The Attention NCO policy studied in this work, however, does not generalize to mission scenarios with different numbers of nodes than those seen in training. NCO policies with robust generalization capabilities are required before NCO approaches can be recommended for integration in practical space VRP solvers.	Chapter 4
SQ.2	*How can trajectory optimization methods with feasibility guarantees, specifically Sequential Convex Programming (SCP), be efficiently integrated in the multi-rendezvous trajectory optimization process?*	Trajectory re-optimization methods, and in particular SCP, are integrated in the STSP solution framework by defining a data flow architecture from mission specification and approximate STSP solutions to trajectory verification. The architecture is implemented through a set of data interfaces, allowing the integration of modules implemented in different languages. The STSP solver is successfully applied to design multi-rendezvous trajectories for the OSSIE OTV.	Chapter 5

6.4 Final Assessment of Compliance with Project Objectives

Code	Context	Goal	Achieved	Executive summary	References
G.1	Heuristic Combinatorial Optimization		✓	Implemented and HCO solver for multi-rendezvous mission design, applied it for ADR and OTV mission design. Researched and traded-off solution encoding schemes, population sampling methods and meta-heuristics to design baseline solver. Determined optimal global heuristic optimization algorithms for the OSSIE and ADR STSP mission scenarios.	Section 2.4

Continued on next page

Code	Context	Goal	Achieved	Executive summary	References
G.2	Neural Combinatorial Optimization		✓	Implemented a NCO solver for space vehicle routing problems. Demonstrated capability to learn routing policies for 3-element payload delivery STSP's and 6-element ADR STSP's. Researched hyperparameter impact on NCO policy using ANOVA. Applied results to perform hyperparameter optimization of NCO policy. Studied NCO model performance ADR STSP scenarios based on the Iridium 33 debris cloud with 10, 30 and 50 transfers.	Section 2.5
G.3	Software Architecture Design		✓	Designed and implemented a 3-stage, modular STSP solver architecture integrating combinatorial optimization, trajectory re-optimization and trajectory verification and validation for use in real missions. Implemented distance-based permutation sampling to leverage candidate solutions from advanced combinatorial optimization methods, enabling integration of heuristic, tree-search and NCO based approaches into STSP solver.	Appendix C
G.4	Mission Analysis of the OSSIE OTV		✓	Implemented OSSIE mission scenario model accounting for: variable payload target states, masses and bundling. Optimized 5000 mission scenarios based on the specifications of the first OSSIE satellite deployment mission. Positively assessed capability of vehicle to fullfil advertised services.	Chapter 5
SG.1	LTTO	LTTO Algorithm	✓	Implemented and thoroughly verified Q-Law and RQ-Law low-thrust transfer guidance policies.	Section 2.3
SG.1.a	LTTO	LTTO Approach Assessment	✓	Assessed literature to select an optimal LTTO approach for conceptual design. Selected Q-Law Lyapunov Feedback Control due to proven use for preliminary mission design, speed of trajectory generation.	Section 2.3
SG.1.b	LTTO	Guidance Policy Implementation	✓	Implemented Q-Law (5-element targeting) and RQ-Law (6-element targeting) low-thrust guidance policy. Implemented policies for both custom and Tudat propagators.	Section 2.3

Continued on next page

Code	Context	Goal	Achieved	Executive summary	References
SG.1.c	LTTO	Optimal Simulator	✓	Selected benchmark optimizer for LTTO. Determined optimal benchmark integration step. Selected optimal integrator for LTTO based on computational cost of integration and output density.	Section 2.3
SG.1.d	LTTO	Cost Estimation Method	✓	Developed Q-Law and RQ-Law transfer cost estimation methods for use in CO based on the best quadratic time of flight. Verified transfer cost estimation method against simulated transfer time measurements.	Section 2.3
SG.2	IPD	Impulsive Trajectory Algorithm	✓	Implemented analytical 3-element targeting impulsive trajectory design algorithm for the OSSIE OTV under the $J2$ perturbation.	Section 2.2
SG.2.a	IPD	Impulsive Approach Assessment	✓	Assessed literature to select an optimal IPD approach for the OSSIE mission. Selected MHT and NIC manoeuvres.	Section 2.2
SG.2.b	IPD	Guidance Policy Implementation	✓	Implemented sequential guidance policy for 3-element targeting with impulsive manoeuvres, chaining MHT and NIC manoeuvres.	Section 2.2
SG.2.c	IPD	Feasible Trajectory Simulator	✓	Implemented an analytical simulator considering the $J2$ perturbation. Used the simulator to generate feasible trajectories used as warm starts for SCP.	Section 2.2
SG.2.d	IPD	Cost Estimation Method	✓	Developed impulsive transfer cost estimation methods for use in CO based on analytical transfer cost derivations. Verified cost estimation methods against simulated transfer time measurements.	Section 2.2
SG.3	HCO	Space VRP Heuristic Solver	✓	Assessed literature to verify choice for HCO. Implemented HCO solver for generalized space VRPs using `pygmo`.	Section 2.4
SG.3.a	HCO	Dynamic Environment Modeling	✓	Created a generalized STSP environment model considering RAAN and AOP drift due to the $J2$ perturbation. Created spacecraft modelling module and used it to model OSSIE, the concept ADR spacecraft, and more. Implemented STSP cost functions for low-thrust and impulsive spacecraft.	Section 2.4

Continued on next page

Code	Context	Goal	Achieved	Executive summary	References
SG.3.b	HCO	Population Sampling Algorithm	✓	Implemented and evaluated multiple uniform permutation sampling algorithms, selecting Sobol Hypercube Permutations. Implemented Mallows Model for permutations to perform distance-based permutation sampling based on the Hamming distance.	Section 2.4
SG.3.c	HCO	Heuristic CO Method Selection	✓	Identified the optimal heuristic global optimizers for the ADR and OSSIE STSPs among all global optimizers available in pygmo. Demonstrated superior performance against local optimization under the following meta-heuristics: shotgun-and-scatter and Monotonic Basing Hopping.	Section 2.4
SG.4	NCO	Space VRP NCO Solver	✓	Implemented NCO solver for generalized space VRPs based using RL4CO.	Section 2.5
SG.4.a	NCO	NCO Approach Assessment	✓	Assessed literature to select optimal NCO approach. Selected attention GNN routing policy.	Section 2.5
SG.4.b	NCO	RL Algorithm Performance	✓	Assessed performance of three RL algorithms to train OSSIE and ADR STSP policies: REINFORCE, A2C and PPO. Found REINFORCE to be best for the OSSIE STSP. Found A2C to be best for ADR STSP.	Section 2.5
SG.4.c	NCO	Hyperparameter Optimization	✓	Performed 3-level ANOVA to determine hyperparameters driving performance. Performed 3-level grid search of key hyperparameters. Found that model architecture and size of training dataset drives model performance.	Section 3.3
SG.4.d	NCO	Performance Comparison	✓	Assessed performance of NCO methods compared to HCO methods for ADR STSP mission scenarios with 10, 30 and 50 transfers.	Section 3.3
SG.5	OSSIE	Mission Viability Analysis	✓	Analyzed feasibility of nominal mission scenarios. Positively assessed capability of vehicle to fulfill advertised services.	Chapter 5
SG.5.a	OSSIE	Mission Modelling	✓	Implemented OSSIE mission scenario model accounting for: variable payload target states, masses and bundling.	Chapter 5

Continued on next page

Code	Context	Goal	Achieved	Executive summary	References
SG.5.b	OSSIE	Statistical Performance Analysis	✓	Optimized 5000 mission scenarios based on the specifications of the first OSSIE satellite deployment mission. Positively assessed capability of vehicle to fullfil advertised services.	Chapter 5

Bibliography

[1] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*, 2nd. 2018. [Online]. Available: http://incompleteideas.net/book/the-book.html.

[2] D. Izzo, I. Getzner, D. Hennes, and L. Simões, "Evolving solutions to TSP variants for active space debris removal: Genetic and evolutionary computation conference (GECCO)," *Proceedings of the 17th annual conference on Genetic and evolutionary computation (GECCO 2015)*, S. Silva, Ed., pp. 1207–1214, 2015, Publisher: ACM Press, ISSN: 9781450334723. DOI: 10.1145/2739480.2754727.

[3] L. Ricciardi and M. Vasile, *Solving multi-objective dynamic travelling salesman problems by relaxation*. Jul. 13, 2019, 1999 pp., Pages: 2007. DOI: 10.1145/3319619.3326837.

[4] L. Federici, A. Zavoli, and G. Colasurdo, "Evolutionary optimization of multirendezvous impulsive trajectories," *International Journal of Aerospace Engineering*, vol. 2021, pp. 1–19, May 27, 2021. DOI: 10.1155/2021/9921555.

[5] L. Medioni, Y. Gary, M. Monclin, *et al.*, "Trajectory optimization for multi-target active debris removal missions," *Advances in Space Research*, Space Environment Management and Space Sustainability, vol. 72, no. 7, pp. 2801–2823, Oct. 1, 2023, ISSN: 0273-1177. DOI: 10.1016/j.asr.2022.12.013. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S027311772201105X (visited on 09/18/2024).

[6] A. Barea, H. Urrutxua, and L. Cadarso, "Large-scale object selection and trajectory planning for multi-target space debris removal missions," *Acta Astronautica*, vol. 170, pp. 289–301, May 1, 2020, ISSN: 0094-5765. DOI: 10.1016/j.actaastro.2020.01.032. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0094576520300436 (visited on 09/18/2024).

[7] S. Narayanaswamy, B. Wu, P. Ludivig, F. Soboczenski, K. Venkataramani, and C. J. Damaren, "Low-thrust rendezvous trajectory generation for multi-target active space debris removal using the RQ-law," *Advances in Space Research*, vol. 71, no. 10, pp. 4276–4287, May 15, 2023, ISSN: 0273-1177. DOI: 10.1016/j.asr.2022.12.049. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0273117722011656 (visited on 09/22/2024).

[8] C. P. Mark and S. Kamath, "Review of active space debris removal methods," *Space Policy*, vol. 47, pp. 194–206, Feb. 1, 2019, ISSN: 0265-9646. DOI: 10.1016/j.spacepol.2018.12.005. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0265964618300110 (visited on 09/18/2024).

[9] C. Bonnal, J.-M. Ruault, and M.-C. Desjean, "Active debris removal: Recent progress and current trends," *Acta Astronautica*, vol. 85, pp. 51–60, Apr. 1, 2013, ISSN: 0094-5765. DOI: 10.1016/j.actaastro.2012.11.009. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0094576512004602 (visited on 09/18/2024).

[10] F. Sellmaier, T. Boge, J. Spurmann, S. Gully, T. Rupp, and F. Huber, "On-orbit servicing missions: Challenges and solutions for spacecraft operations," in *SpaceOps 2010 Conference*, Huntsville, Alabama: American Institute of Aeronautics and Astronautics, Apr. 25, 2010, ISBN: 978-1-62410-164-9. DOI: 10.2514/6.2010-2159. [Online]. Available: https://arc.aiaa.org/doi/10.2514/6.2010-2159 (visited on 09/18/2024).

[11] T. Sarton du Jonchay, H. Chen, M. Isaji, Y. Shimane, and K. Ho, "On-orbit servicing optimization framework with high- and low-thrust propulsion tradeoff," *Journal of Spacecraft and Rockets*, vol. 59, no. 1, pp. 33–48, 2022, Publisher: American Institute of Aeronautics and Astronautics _eprint: https://doi.org/10.2514/1.A3509 ISSN: 0022-4650. DOI: 10.2514/1.A35094. [Online]. Available: https://doi.org/10.2514/1.A35094 (visited on 09/18/2024).

[12] S. E. Sorenson and S. G. N. Pinkley, "Multi-orbit routing and scheduling of refuellable on-orbit servicing space robots," *Computers & Industrial Engineering*, vol. 176, p. 108 852, Feb. 1, 2023, ISSN: 0360-8352. DOI: 10.1016/j.cie.2022.108852. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0360835222008403 (visited on 09/18/2024).

[13] A. Petropoulos, D. Grebow, D. Jones, *et al.*, "GTOC9: Methods and results from the jet propulsion laboratory team," May 1, 2017.

[14] S. Lu, Y. Zhang, J. Zhang, *et al.*, "GTOC 11: Results found at beijing institute of technology and china academy of space technology," *Acta Astronautica*, vol. 202, pp. 876–888, Jan. 1, 2023, ISSN: 0094-5765. DOI: 10.1016/j.actaastro.2022.07.039. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0094576522003885 (visited on 09/18/2024).

[15] H. Li, S. Chen, D. Izzo, and H. Baoyin, "Deep networks as approximators of optimal low-thrust and multi-impulse cost in multitarget missions," *Acta Astronautica*, vol. 166, pp. 469–481, Jan. 1, 2020, ADS Bibcode: 2020AcAau.166..469L, ISSN: 0094-5765. DOI: 10.1016/j.actaastro.2019.09.023. [Online]. Available: https://ui.adsabs.harvard.edu/abs/2020AcAau.166..469L (visited on 09/18/2024).

[16] A. Francois, Q. Cappart, and L.-M. Rousseau, "How to evaluate machine learning approaches for combinatorial optimization: Application to the travelling salesman problem," Sep. 2019. [Online]. Available: https://arxiv.org/abs/1909.13121.

[17] M. Hallmann, M. Schlotterer, A. Heidecker, *et al.*, *GTOC 9, Multiple Space Debris Rendezvous Trajectory Design in the J2 environment*. Jun. 1, 2017.

[18] F. Berto, C. Hua, J. Park, *et al.*, *RL4co: An extensive reinforcement learning for combinatorial optimization benchmark*, Jun. 21, 2024. DOI: 10.48550/arXiv.2306.17100. arXiv: 2306.17100[cs]. [Online]. Available: http://arxiv.org/abs/2306.17100 (visited on 09/18/2024).

[19] J. Locke, T. Colvin J., L. Ratliff, A. Abdulhamid, and C. Samples, "Cost and benefit analysis of mitigating, tracking, and remediating orbital debris," NASA Office of Technology, Policy, and Strategy, NASA Headquarters 300 E Street SW Washington, DC 20024, 20240003484, 2024, p. 152. [Online]. Available: https://www.nasa.gov/wp-content/uploads/2024/05/2024-otps-cba-of-orbital-debris-phase-2-plus-svgs-v3-tjc-tagged.pdf?emrc=224cd2.

[20] D. Izzo, M. Märtens, and B. Pan, "A survey on artificial intelligence trends in spacecraft guidance dynamics and control," *Astrodynamics*, vol. 3, no. 4, pp. 287–299, Dec. 1, 2019, ISSN: 2522-0098. DOI: 10.1007/s42064-018-0053-6. [Online]. Available: https://doi.org/10.1007/s42064-018-0053-6 (visited on 09/18/2024).

[21] D. Izzo, E. Ozturk, and M. Märtens, *Interplanetary Transfers via Deep Representations of the Optimal Policy and/or of the Value Function*. Apr. 18, 2019. DOI: 10.48550/arXiv.1904.08809.

[22] H. Klinkrad, *Space Debris* (Springer Praxis Books). Springer Berlin Heidelberg, 2006, ISBN: 978-3-540-25448-5. DOI: 10.1007/3-540-37674-7. [Online]. Available: http://link.springer.com/10.1007/3-540-37674-7 (visited on 09/21/2024).

[23] "ESA's annual space environment report," ESA ESOC Space Debris Office, Robert-Bosch-Strasse 5 D-64293 Darmstadt Germany, LOG GEN-DB-LOG-00288-OPS-SD, Jul. 19, 2024, p. 121. [Online]. Available: https://www.sdo.esoc.esa.int/environment_report/Space_Environment_Report_latest.pdf (visited on 09/21/2024).

[24] D. J. Kessler and B. G. Cour-Palais, "Collision frequency of artificial satellites: The creation of a debris belt," *Journal of Geophysical Research*, vol. 83, pp. 2637–2646, A6 Jun. 1978. DOI: 10.1029/JA083iA06p02637. [Online]. Available: https://ui.adsabs.harvard.edu/abs/1978JGR....83.2637K/abstract (visited on 09/21/2024).

[25] J. .-. Liou and N. L. Johnson, "Instability of the present LEO satellite populations," *Advances in Space Research*, vol. 41, no. 7, pp. 1046–1053, Jan. 1, 2008, ISSN: 0273-1177. DOI: 10.1016/j.asr.2007.04.081. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0273117707004097 (visited on 09/21/2024).

[26] L. Hall, "The history of space debris," *Space Traffic Management Conference*, Nov. 6, 2014. [Online]. Available: https://commons.erau.edu/stm/2014/thursday/19.

[27] U. N. O. for Outer Space Affairs, *Space debris mitigation guidelines of the committee on the peaceful uses of outer space*, Jan. 2010. [Online]. Available: https://www.unoosa.org/pdf/publications/st_space_49E.pdf.

[28] C. for the Assessment of NASA's Orbital Debris Programs, *Limiting Future Collision Risk to Spacecraft: An Assessment of NASA's Meteoroid and Orbital Debris Programs*. Washington, D.C.: National Academies Press, Nov. 16, 2011, ISBN: 978-0-309-21974-7. DOI: 10.17226/13244. [Online]. Available: http://www.nap.edu/catalog/13244 (visited on 09/23/2024).

[29] E. S. D. M. W. Group, *ESA space debris mitigation requirements*, Oct. 30, 2023. [Online]. Available: https://technology.esa.int/upload/media/DGHKMZ_6542582e18e33.pdf (visited on 09/23/2024).

[30] M. Shan, J. Guo, and E. Gill, "Review and comparison of active space debris capturing and removal methods," *Progress in Aerospace Sciences*, vol. 80, pp. 18–32, Jan. 1, 2016, ISSN: 0376-0421. DOI: 10.1016/j.paerosci.2015.11.001. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0376042115300221 (visited on 09/21/2024).

[31] C. Wiedemann, H. Krag, J. Bendisch, and H. Sdunnus, "Analyzing costs of space debris mitigation methods," *Advances in Space Research*, Space Debris, vol. 34, no. 5, pp. 1241–1245, Jan. 1, 2004, ISSN: 0273-1177. DOI: 10.1016/j.asr.2003.10.041. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0273117704001048 (visited on 09/21/2024).

[32] G. Borelli, M. Trisolini, M. Massari, and C. Colombo, *A comprehensive ranking framework for active debris removal missions candidates*. Apr. 29, 2021.

[33] C. Saunders, J. Forshaw, V. Lappas, *et al.*, "Business and economic considerations for service oriented active debris removal missions," *Proceedings of the International Astronautical Congress, IAC*, vol. 3, pp. 1729–1743, Jan. 1, 2014.

[34] R. Biesbroek, L. Innocenti, A. Wolahan, and S. M. Serrano, "E.DEORBIT – ESA's ACTIVE DEBRIS REMOVAL MISSION," presented at the 7th European Conference on Space Debris, Darmstadt, Germany: ESA Space Debris Office, Apr. 18, 2017. [Online]. Available: https://conference.sdo.esoc.esa.int/proceedings/sdc7/paper/1053/SDC7-paper1053.pdf.

[35] J. L. Forshaw, G. S. Aglietti, N. Navarathinam, *et al.*, "RemoveDEBRIS: An in-orbit active debris removal demonstration mission," *Acta Astronautica*, vol. 127, pp. 448–463, Oct. 1, 2016, ISSN: 0094-5765. DOI: 10.1016/j.actaastro.2016.06.018. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S009457651530117X (visited on 09/23/2024).

[36] J. L. Forshaw, G. S. Aglietti, S. Fellowes, *et al.*, "The active space debris removal mission RemoveDebris. part 1: From concept to launch," *Acta Astronautica*, vol. 168, pp. 293–309, Mar. 1, 2020, ISSN: 0094-5765. DOI: 10.1016/j.actaastro.2019.09.002. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0094576519312512 (visited on 09/23/2024).

[37] R. Biesbroek, S. Aziz, A. Wolahan, S. Cipolla, M. Richard-Noca, and L. Piguet, "THE CLEARSPACE-1 MISSION: ESA AND CLEARSPACE TEAM UP TO REMOVE DEBRIS," presented at the 8th European Conference on Space Debris, Darmstadt, Germany: ESA Space Debris Office, Apr. 20, 2021. [Online]. Available: https://conference.sdo.esoc.esa.int/proceedings/sdc8/paper/320/SDC8-paper320.pdf.

[38] T. Kelso. "Analysis of the iridium 33cosmos 2251 collision." (), [Online]. Available: https://www.researchgate.net/publication/242543407_Analysis_of_the_Iridium_33Cosmos_2251_Collision (visited on 09/24/2024).

[39] N. L. Johnson, E. Stansbery, J.-C. Liou, M. Horstman, C. Stokely, and D. Whitlock, "The characteristics and consequences of the break-up of the fengyun-1c spacecraft," *Acta Astronautica*, vol. 63, no. 1, pp. 128–135, Jul. 2008, ISSN: 00945765. DOI: 10.1016/j.actaastro.2007.12.044. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0094576507003281 (visited on 09/23/2024).

[40] C. Pardini and L. Anselmo, "The short-term effects of the cosmos 1408 fragmentation on neighboring inhabited space stations and large constellations," *Acta Astronautica*, vol. 210, pp. 465–473, Sep. 1, 2023, ISSN: 0094-5765. DOI: 10.1016/j.actaastro.2023.02.043. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0094576523001078 (visited on 09/23/2024).

[41] N. L. Johnson, J. R. Gabbard, G. T. Devere, and E. E. Johnson, "History of on-orbit satellite fragmentations," NAS 1.26:171814, Aug. 1, 1984, NTRS Author Affiliations: Teledyne Brown Engineering NTRS Document ID: 19840026390 NTRS Research Center: Legacy CDMS (CDMS). [Online]. Available: https://ntrs.nasa.gov/citations/19840026390 (visited on 10/19/2024).

[42] R. W. Conversano and R. E. Wirz, "Mission capability assessment of CubeSats using a miniature ion thruster," *Journal of Spacecraft and Rockets*, vol. 50, no. 5, pp. 1035–1046, 2013, Publisher: American Institute of Aeronautics and Astronautics _eprint: https://doi.org/10.2514/1.A32435, ISSN: 0022-4650. DOI: 10.2514/1.A32435. [Online]. Available: https://doi.org/10.2514/1.A32435 (visited on 09/24/2024).

[43] D. O'Reilly, G. Herdrich, and D. F. Kavanagh, "Electric propulsion methods for small satellites: A review," *Aerospace*, vol. 8, no. 1, p. 22, Jan. 2021, Number: 1 Publisher: Multidisciplinary Digital Publishing Institute, ISSN: 2226-4310. DOI: 10.3390/aerospace8010022. [Online]. Available: https://www.mdpi.com/2226-4310/8/1/22 (visited on 09/24/2024).

[44] O. Montenbruck and E. Gill, *Satellite Orbits*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2000, ISBN: 978-3-540-67280-7 978-3-642-58351-3. DOI: 10.1007/978-3-642-58351-3. [Online]. Available: http://link.springer.com/10.1007/978-3-642-58351-3 (visited on 10/03/2024).

[45] G. Hintz, "Survey of orbit element sets," *Journal of Guidance Control and Dynamics - J GUID CONTROL DYNAM*, vol. 31, pp. 785–790, May 1, 2008. DOI: 10.2514/1.32237.

[46] S. Narayanaswamy and C. J. Damaren, "Equinoctial lyapunov control law for low-thrust rendezvous," *Journal of Guidance, Control, and Dynamics*, vol. 46, no. 4, pp. 781–795, 2023, Publisher: American Institute of Aeronautics and Astronautics _eprint: https://doi.org/10.2514/1.G006662, ISSN: 0731-5090. DOI: 10.2514/1.G006662. [Online]. Available: https://doi.org/10.2514/1.G006662 (visited on 09/22/2024).

[47] K. Wakker, *Fundamentals of Astrodynamics*. TU Delft Library, Jan. 2015, ISBN: 978-94-6186-419-2. [Online]. Available: https://repository.tudelft.nl/islandora/object/uuid%5C%3A3fc91471-8e47-4215-af43-718740e6694e.

[48] K. T. Alfriend, S. R. Vadali, P. Gurfil, J. P. How, and L. S. Breger, "Chapter 2 - fundamental astrodynamics," in *Spacecraft Formation Flying*, K. T. Alfriend, S. R. Vadali, P. Gurfil, J. P. How, and L. S. Breger, Eds., Oxford: Butterworth-Heinemann, Jan. 1, 2010, pp. 13–38, ISBN: 978-0-7506-8533-7. DOI: 10.1016/B978-0-7506-8533-7.00207-4. [Online]. Available: https://www.sciencedirect.com/science/article/pii/B9780750685337002074 (visited on 09/18/2024).

[49] J. T. Betts, "Survey of numerical methods for trajectory optimization," *Journal of Guidance, Control, and Dynamics*, vol. 21, no. 2, pp. 193–207, Mar. 1998, Publisher: American Institute of Aeronautics and Astronautics, ISSN: 0731-5090. DOI: 10.2514/2.4231. [Online]. Available: https://arc.aiaa.org/doi/10.2514/2.4231 (visited on 09/24/2024).

[50] D. Morante, M. Sanjurjo Rivo, and M. Soler, "A survey on low-thrust trajectory optimization approaches," *Aerospace*, vol. 8, no. 3, 2021, ISSN: 2226-4310. DOI: 10.3390/aerospace8030088. [Online]. Available: https://www.mdpi.com/2226-4310/8/3/88.

[51] I. M. Ross and C. N. D'Souza, "Hybrid optimal control framework for mission planning," *Journal of Guidance, Control, and Dynamics*, vol. 28, no. 4, pp. 686–697, Jul. 2005, Publisher: American Institute of Aeronautics and Astronautics, ISSN: 0731-5090. DOI: 10.2514/1.8285. [Online]. Available: https://arc.aiaa.org/doi/10.2514/1.8285 (visited on 09/26/2024).

[52] B. Passenberg, *Theory and Algorithms for Indirect Methods in Optimal Control of Hybrid Systems*. Technischen Universitat Munchen, Jan. 19, 2012, 247 pp.

[53] M. Rungger and O. Stursberg, "Continuity of the value function for exit time optimal control problems of hybrid systems," presented at the Proceedings of the IEEE Conference on Decision and Control, Dec. 1, 2010, pp. 4210–4215. DOI: 10.1109/CDC.2010.5718054.

[54] H. S. Tsien, "Take-off from satellite orbit," *Journal of the American Rocket Society*, vol. 23, no. 4, pp. 233–236, Jul. 1953, Publisher: American Institute of Aeronautics and Astronautics. DOI: 10.2514/8.4599. [Online]. Available: https://arc.aiaa.org/doi/10.2514/8.4599 (visited on 09/26/2024).

[55] R. H. Battin, "An introduction to the mathematics and methods of astrodynamics, revised edition," Reston, VA: American Institute of Aeronautics and Astronautics, Inc., Jan. 1, 1999, ISBN: 978-1-56347-342-5 978-1-60086-154-3. DOI: 10.2514/4.861543. [Online]. Available: https://arc.aiaa.org/doi/book/10.2514/4.861543 (visited on 09/26/2024).

[56] B. Conway, *Spacecraft Trajectory Optimization*. The Edinburgh Building, Cambridge CB2 8RU, UK: Cambridge University Press, 2010, 312 pp., ISBN: 978-1-107-65382-5.

[57] A. Rao, "A survey of numerical methods for optimal control," *Advances in the Astronautical Sciences*, vol. 135, Jan. 1, 2010.

[58] T. Haberkorn, P. Martinon, and J. Gergaud, "Low thrust minimum-fuel orbital transfer: A homotopic approach," *Journal of Guidance, Control, and Dynamics*, vol. 27, no. 6, pp. 1046–1060, Nov. 2004, Publisher: American Institute of Aeronautics and Astronautics, ISSN: 0731-5090. DOI: 10.2514/1.4022. [Online]. Available: https://arc.aiaa.org/doi/10.2514/1.4022 (visited on 09/26/2024).

[59] R. Bellman, *Dynamic programming*. Princeton, NJ: Princeton Univ. Pr, 1984, 339 pp., ISBN: 978-0-691-07951-6.

[60] G. Whiffen, *Mystic: Implementation of the Static Dynamic Optimal Control Algorithm for High-Fidelity, Low-Thrust Trajectory Design*. Aug. 21, 2006, ISBN: 978-1-62410-048-2. DOI: 10.2514/6.2006-6741.

[61] J. A. Sims and G. J. Whiffen, *Application of a novel optimal control algorithm to low-thrust trajectory optimization*, Feb. 14, 2001. [Online]. Available: https://ntrs.nasa.gov/citations/20210002619 (visited on 09/25/2024).

[62] G. Lantoine and R. P. Russell, "A hybrid differential dynamic programming algorithm for constrained optimal control problems. part 2: Application," *Journal of Optimization Theory and Applications*, vol. 154, no. 2, pp. 418–442, Aug. 1, 2012, ISSN: 1573-2878. DOI: 10.1007/s10957-012-0038-1. [Online]. Available: https://doi.org/10.1007/s10957-012-0038-1 (visited on 09/26/2024).

[63] G. Lantoine and R. P. Russell, "A hybrid differential dynamic programming algorithm for constrained optimal control problems. part 1: Theory," *Journal of Optimization Theory and Applications*, vol. 154, no. 2, pp. 382–417, Aug. 1, 2012, ISSN: 1573-2878. DOI: 10.1007/s10957-012-0039-0. [Online]. Available: https://doi.org/10.1007/s10957-012-0039-0 (visited on 09/26/2024).

[64] J. Aziz, J. Parker, D. Scheeres, and J. Englander, "Low-thrust many-revolution trajectory optimization via differential dynamic programming and a sundman transformation," *The Journal of the Astronautical Sciences*, vol. 65, Jan. 4, 2018. DOI: 10.1007/s40295-017-0122-8.

[65] R. Falck, W. Sjauw, and D. Smith, *Comparison of Low-Thrust Control Laws for Applications in Planetocentric Space*. Jul. 28, 2014, Journal Abbreviation: 50th AIAA/ASME/SAE/ASEE Joint Propulsion Conference 2014 Publication Title: 50th AIAA/ASME/SAE/ASEE Joint Propulsion Conference 2014, ISBN: 978-1-62410-303-2. DOI: 10.2514/6.2014-3714.

[66] M. Ilgen, “Low thrust otv guidance using lyapunov optimal feedback control techniques.,” *Advances in the Astronautical Sciences*, vol. 85, pp. 1527–1545, Pt 2 1993, ISSN: 0065-3438. [Online]. Available: https://jglobal.jst.go.jp/en/detail?JGLOBAL_ID=200902144132473360 (visited on 09/28/2024).

[67] A. Petropoulos, “Refinements to the q-law for low-thrust orbit transfers,” vol. 120, pp. 963–982, Jan. 1, 2005.

[68] G. I. Varga and J. M. S. Perez, “Many-revolution low-thrust orbit transfer computation using equinoctial q-law including j2 and eclipse effects,” *ICATT*, 2016. [Online]. Available: https://indico.esa.int/event/111/contributions/346/attachments/336/377/Final_Paper_ICATT.pdf.

[69] A. Petropoulos, “Low-thrust orbit transfers using candidate lyapunov functions with a mechanism for coasting,” in *AIAA/AAS Astrodynamics Specialist Conference and Exhibit*, ser. Guidance, Navigation, and Control and Co-located Conferences, American Institute of Aeronautics and Astronautics, Aug. 16, 2004. DOI: 10.2514/6.2004-5089. [Online]. Available: https://arc.aiaa.org/doi/10.2514/6.2004-5089 (visited on 09/22/2024).

[70] S. Lee, P. Von Allmen, W. Fink, A. Petropoulos, and R. Terrile, “Design and optimization of low-thrust orbit transfers,” *IEEE Aerospace Conference Proceedings*, Jan. 1, 2005.

[71] J. Shannon, M. Ozimek, J. Atchison, and C. Hartzell, “Q-law aided direct trajectory optimization of many-revolution low-thrust transfers,” *Journal of Spacecraft and Rockets*, vol. 57, pp. 1–11, May 11, 2020. DOI: 10.2514/1.A34586.

[72] D. Lantukh, C. Ranieri, M. Diprinzio, and P. Edelman, *Enhanced Q-Law Lyapunov Control for Low-Thrust Transfer and Rendezvous Design*. Aug. 24, 2017.

[73] H. Holt, R. Armellin, N. Baresi, *et al.*, “Optimal q-laws via reinforcement learning with guaranteed stability,” *Acta Astronautica*, vol. 187, pp. 511–528, Oct. 1, 2021, ISSN: 0094-5765. DOI: 10.1016/j.actaastro.2021.07.010. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0094576521003684 (visited on 09/28/2024).

[74] D. Dirkx, M. Fayolle, G. Garrett, *et al.*, “The open-source astrodynamics tudatpy software - overview for planetary mission design and science analysis,” *European Planetary Science Congress*, EPSC2022–253, Sep. 2022. DOI: 10.5194/epsc2022-253. [Online]. Available: https://ui.adsabs.harvard.edu/abs/2022EPSC...16..253D/abstract (visited on 09/19/2024).

[75] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, *Numerical Recipes: The Art of Scientific Computing*, 3rd edition. Cambridge: Cambridge University Press, Sep. 10, 2007, 1256 pp., ISBN: 978-0-521-88068-8.

[76] J. Hon, “Optimization framework for low-thrust active debris removal missions with multiple selected targets,” M. Reza Emami, Ed., AMOS, 2022. [Online]. Available: https://amostech.com/TechnicalPapers/2022/Poster/Hon.pdf.

[77] J. Cohen, *Statistical Power Analysis for the Behavioral Sciences*, 2nd ed. New York: Routledge, Jul. 1, 1988, 567 pp., ISBN: 978-0-203-77158-7. DOI: 10.4324/9780203771587.

[78] N. R. Draper and H. Smith, *Applied Regression Analysis*. John Wiley & Sons, Apr. 23, 1998, 736 pp., Google-Books-ID: d6NsDwAAQBAJ, ISBN: 978-0-471-17082-2.

[79] W. F. Guthrie and N. A. Heckert, “NIST/SEMATECH engineering statistics handbook,” *NIST*, 2016, Last Modified: 2016-04-27T15:32-04:00. [Online]. Available: https://www.itl.nist.gov/div898/handbook/ (visited on 09/24/2024).

[80] F. Biscani and D. Izzo, “A parallel global multiobjective framework for optimization: Pagmo,” *Journal of Open Source Software*, vol. 5, no. 53, p. 2338, Sep. 13, 2020, ISSN: 2475-9066. DOI: 10.21105/joss.02338. [Online]. Available: https://joss.theoj.org/papers/10.21105/joss.02338 (visited on 09/18/2024).

[81] J. Blank and K. Deb, “Pymoo: Multi-objective optimization in python,” *IEEE Access*, vol. 8, pp. 89 497–89 509, 2020, Conference Name: IEEE Access, ISSN: 2169-3536. DOI: 10.1109/ACCESS.2020.2990567. [Online]. Available: https://ieeexplore.ieee.org/document/9078759 (visited on 09/18/2024).

[82] C. Blum and A. Roli, “Metaheuristics in combinatorial optimization: Overview and conceptual comparison,” *ACM Comput. Surv.*, vol. 35, pp. 268–308, Jan. 1, 2001. DOI: 10.1145/937503.937505.

[83] C. Coello, D. Veldhuizen, and G. Lamont, *Evolutionary Algorithms for Solving Multi-Objective Problems Second Edition*. Jan. 1, 2007, Journal Abbreviation: Kluwer Academic Publication Title: Kluwer Academic, ISBN: 978-0-387-33254-3. DOI: 10.1007/978-0-387-36797-2.

[84] R. Mitchell, J. Cooper, E. Frank, and G. Holmes, “Sampling permutations for shapley value estimation,” *Journal of Machine Learning Research*, vol. 23, no. 43, pp. 1–46, 2022, ISSN: 1533-7928. [Online]. Available: http://jmlr.org/papers/v23/21-0439.html (visited on 09/19/2024).

[85] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, “Optimization by simulated annealing,” *Science*, vol. 220, no. 4598, pp. 671–680, May 13, 1983, Publisher: American Association for the Advancement of Science. DOI: 10.1126/science.220.4598.671. [Online]. Available: https://www.science.org/doi/10.1126/science.220.4598.671 (visited on 09/29/2024).

[86] C. L. Mallows, “Non-null ranking models. i,” *Biometrika*, vol. 44, no. 1, pp. 114–130, 1957, Publisher: [Oxford University Press, Biometrika Trust], ISSN: 0006-3444. DOI: 10.2307/2333244. [Online]. Available: https://www.jstor.org/stable/2333244 (visited on 09/19/2024).

[87] P. Diaconis, “Group representations in probability and statistics,” *Lecture Notes-Monograph Series*, vol. 11, pp. i–192, 1988, Publisher: Institute of Mathematical Statistics, ISSN: 0749-2170. [Online]. Available: https://www.jstor.org/stable/4355560 (visited on 09/19/2024).

[88] E. Irurozki, “Sampling and learning distance-based probability models for permutation spaces,” Ph.D. dissertation, Universidad del País Vasco - Euskal Herriko Unibertsitatea, 2014. [Online]. Available: https://dialnet.unirioja.es/servlet/tesis?codigo=213087 (visited on 09/19/2024).

[89] B. Waggener and W. N. Waggener, *Pulse Code Modulation Techniques*. Springer Science & Business Media, 1995, 388 pp., Google-Books-ID: 8l_o6kI3760C, ISBN: 978-0-442-01436-0.

[90] J. C. Bean, “Genetic algorithms and random keys for sequencing and optimization,” *ORSA Journal on Computing*, vol. 6, no. 2, pp. 154–160, May 1994, Publisher: ORSA, ISSN: 0899-1499. DOI: 10.1287/ijoc.6.2.154. [Online]. Available: https://pubsonline.informs.org/doi/abs/10.1287/ijoc.6.2.154 (visited on 09/19/2024).

[91] P. Krömer, V. Uher, and V. Snášel, “Novel random key encoding schemes for the differential evolution of permutation problems,” *IEEE Transactions on Evolutionary Computation*, vol. 26, no. 1, pp. 43–57, Feb. 2022, Conference Name: IEEE Transactions on Evolutionary Computation, ISSN: 1941-0026. DOI: 10.1109/TEVC.2021.3087802. [Online]. Available: https://ieeexplore.ieee.org/document/9449662 (visited on 09/19/2024).

[92] M. A. Londe, L. S. Pessoa, C. E. Andrade, and M. G. C. Resende, “Biased random-key genetic algorithms: A review,” *European Journal of Operational Research*, Mar. 26, 2024, ISSN: 0377-2217. DOI: 10.1016/j.ejor.2024.03.030. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0377221724002303 (visited on 09/20/2024).

[93] A. Boley and M. Byers, “Satellite mega-constellations create risks in low earth orbit, the atmosphere and on earth,” *Scientific Reports*, vol. 11, p. 10 642, May 20, 2021. DOI: 10.1038/s41598-021-89909-7.

[94] M. Freitag and Y. Al-Onaizan, “Beam search strategies for neural machine translation,” in *Proceedings of the First Workshop on Neural Machine Translation*, T. Luong, A. Birch, G. Neubig, and A. Finch, Eds., Vancouver: Association for Computational Linguistics, Aug. 2017, pp. 56–60. DOI: 10.18653/v1/W17-3207. [Online]. Available: https://aclanthology.org/W17-3207 (visited on 09/30/2024).

[95] B. Lowerre, “The HARPY speech recognition system,” 1976. [Online]. Available: https://www.semanticscholar.org/paper/The-HARPY-speech-recognition-system-Lowerre/bdb3f20fe41bb95f6bc9d162e827de8db3f952d7 (visited on 09/30/2024).

[96] A. Paszke, S. Gross, F. Massa, *et al.*, “PyTorch: An imperative style, high-performance deep learning library,” in *Proceedings of the 33rd International Conference on Neural Information Processing Systems*, 721, Red Hook, NY, USA: Curran Associates Inc., 2019, pp. 8026–8037. (visited on 09/24/2024).

[97] J. von Neumann, "Various techniques used in connection with random digits," *Journal of Research of the National Bureau of Standards*, Applied Math Series, 1951. [Online]. Available: https://mcnp.lanl.gov/pdf_files/InBook_Computing_1961_Neumann_JohnVonNeumannCollectedWorks_VariousTechniquesUsedinConnectionwithRandomDigits.pdf (visited on 09/24/2024).

[98] W. K. Hastings, "Monte carlo sampling methods using markov chains and their applications," *Biometrika*, no. 57, p. 97, 1970. [Online]. Available: https://probability.ca/hastings/hastings.pdf.

[99] W. Kool, H. van Hoof, and M. Welling, *Attention, learn to solve routing problems!* Feb. 7, 2019. DOI: 10.48550/arXiv.1803.08475. arXiv: 1803.08475[cs,stat]. [Online]. Available: http://arxiv.org/abs/1803.08475 (visited on 09/20/2024).

[100] R. J. Williams, "Simple statistical gradient-following algorithms for connectionist reinforcement learning," *Machine Learning*, vol. 8, no. 3, pp. 229–256, May 1, 1992, ISSN: 1573-0565. DOI: 10.1007/BF00992696. [Online]. Available: https://doi.org/10.1007/BF00992696 (visited on 09/20/2024).

[101] V. Konda and J. Tsitsiklis, "Actor-critic algorithms," in *Advances in Neural Information Processing Systems*, vol. 12, MIT Press, 1999. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/1999/hash/6449f44a102fde848669bdd9eb6b76fa-Abstract.html (visited on 09/20/2024).

[102] V. Mnih, A. P. Badia, M. Mirza, *et al.*, *Asynchronous methods for deep reinforcement learning*, Jun. 16, 2016. DOI: 10.48550/arXiv.1602.01783. arXiv: 1602.01783[cs]. [Online]. Available: http://arxiv.org/abs/1602.01783 (visited on 09/20/2024).

[103] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, *Proximal policy optimization algorithms*, Aug. 28, 2017. DOI: 10.48550/arXiv.1707.06347. arXiv: 1707.06347[cs]. [Online]. Available: http://arxiv.org/abs/1707.06347 (visited on 09/20/2024).

[104] W. Kool, H. v. Hoof, and M. Welling, "Attention, learn to solve routing problems!," presented at the International Conference on Learning Representations, Sep. 27, 2018. [Online]. Available: https://openreview.net/forum?id=ByxBFsRqYm (visited on 07/24/2024).

[105] K. Beck, *Test-driven Development: By Example*. Addison-Wesley Professional, 2003, 241 pp., Google-Books-ID: CUlsAQAAQBAJ, ISBN: 978-0-321-14653-3.

[106] T. M. Inc., *Simulation and model-based design*, version 23.2, Natick, Massachusetts, United States, 2023. [Online]. Available: https://mathworks.com/products/simulink.html.

[107] T. M. Inc., *MATLAB version: 23.2.0 (r2023b)*, version 23.2.0 (R2023b), Natick, Massachusetts, United States, 2023. [Online]. Available: https://www.mathworks.com.

[108] W. R. Shadish, T. D. Cook, and D. D. Campbell, *Experimental and Quasi-Experimental Designs for Generalized Causal Inference*. Boston, MA: Houghton Mifflin Company, 2002, 623 pp.

[109] S. Chacon and B. Straub, *Pro Git*, 2nd edition. Apress, Nov. 18, 2014, 1061 pp.

[110] C. Batini and M. Scanapiecca, *Data Quality* (Data-Centric Systems and Applications). Springer Berlin Heidelberg, 2006, ISBN: 978-3-540-33172-8. DOI: 10.1007/3-540-33173-5. [Online]. Available: http://link.springer.com/10.1007/3-540-33173-5 (visited on 10/22/2024).

[111] B. A. Nosek, G. Alter, G. C. Banks, *et al.*, "Promoting an open research culture: Author guidelines for journals could help to promote transparency, openness, and reproducibility," *Science*, vol. 348, no. 6242, pp. 1422–1425, 2015, Place: US Publisher: American Assn for the Advancement of Science, ISSN: 1095-9203. DOI: 10.1126/science.aab2374.

[112] M. Fewster and D. Graham, *Software Test Automation: Effective Use of Test Execution Tools*. Addison-Wesley, 1999, 596 pp., Google-Books-ID: in7gTCu5SE8C, ISBN: 978-0-201-33140-0.

[113] R. A. Fisher, *The design of experiments* (The design of experiments). Oxford, England: Oliver & Boyd, 1935, xi, 251, Pages: xi, 251.

[114] W. K. Michener, "Ten simple rules for creating a good data management plan," *PLOS Computational Biology*, vol. 11, no. 10, e1004525, Oct. 22, 2015, Publisher: Public Library of Science, ISSN: 1553-7358. DOI: 10.1371/journal.pcbi.1004525. [Online]. Available: https://journals.plos.org/ploscompbiol/article?id=10.1371/journal.pcbi.1004525 (visited on 10/22/2024).

[115] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning* (Springer Series in Statistics). New York, NY: Springer, 2009, ISBN: 978-0-387-84857-0 978-0-387-84858-7. DOI: 10.1007/978-0-387-84858-7. [Online]. Available: http://link.springer.com/10.1007/978-0-387-84858-7 (visited on 10/20/2024).

[116] M. H. Kutner, Ed., *Applied linear statistical models*, 5. ed, McGraw-Hill/Irwin series Operations and decision sciences, Boston, Mass.: McGraw-Hill Irwin, 2005, 1396 pp., ISBN: 978-0-07-238688-2.

[117] G. Taguchi and S. Konishi, *Orthogonal Arrays and Linear Graphs: Tools for Quality Engineering*. American Supplier Institute, 1987, 72 pp., Google-Books-ID: _5BRnQAACAAJ, ISBN: 978-0-941243-01-8.

[118] G. Taguchi, *Taguchi Methods: Design of Experiments*. ASI Press, 1993, 366 pp., Google-Books-ID: veNTAAAAMAAJ, ISBN: 978-0-941243-18-6.

[119] S. Maghsoodloo, G. Ozdemir, V. Jordan, and C.-H. Huang, "Strengths and limitations of taguchi's contributions to quality, manufacturing, and process engineering," *Journal of Manufacturing Systems*, vol. 23, no. 2, pp. 73–126, Jan. 1, 2004, ISSN: 0278-6125. DOI: 10.1016/S0278-6125(05)00004-X. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S027861250500004X (visited on 10/18/2024).

[120] S. Seabold and J. Perktold, "Statsmodels: Econometric and statistical modeling with python," in *Proceedings of the 9th Python in Science Conference*, S. v. d. Walt and J. Millman, Eds., 2010, pp. 92–96. DOI: 10.25080/Majora-92bf1922-011. [Online]. Available: https://proceedings.scipy.org/articles/Majora-92bf1922-011.

[121] S. S. Shapiro and M. B. Wilk, "An analysis of variance test for normality (complete samples)†," *Biometrika*, vol. 52, no. 3, pp. 591–611, Dec. 1, 1965, ISSN: 0006-3444. DOI: 10.1093/biomet/52.3-4.591. [Online]. Available: https://doi.org/10.1093/biomet/52.3-4.591 (visited on 10/22/2024).

[122] A. Vaswani, N. Shazeer, N. Parmar, *et al.*, *Attention is all you need*, Aug. 2, 2023. DOI: 10.48550/arXiv.1706.03762. arXiv: 1706.03762. [Online]. Available: http://arxiv.org/abs/1706.03762 (visited on 10/24/2024).

[123] J. Kaplan, S. McCandlish, T. Henighan, *et al.*, *Scaling laws for neural language models*, Jan. 23, 2020. DOI: 10.48550/arXiv.2001.08361. arXiv: 2001.08361. [Online]. Available: http://arxiv.org/abs/2001.08361 (visited on 10/24/2024).

[124] J. Hoffmann, S. Borgeaud, A. Mensch, *et al.*, *Training compute-optimal large language models*, Mar. 29, 2022. DOI: 10.48550/arXiv.2203.15556. arXiv: 2203.15556. [Online]. Available: http://arxiv.org/abs/2203.15556 (visited on 10/24/2024).

[125] F. Berto, C. Hua, N. G. Zepeda, *et al.*, *RouteFinder: Towards foundation models for vehicle routing problems*, Oct. 9, 2024. DOI: 10.48550/arXiv.2406.15007. arXiv: 2406.15007. [Online]. Available: http://arxiv.org/abs/2406.15007 (visited on 10/21/2024).

[126] M. Raissi, P. Perdikaris, and G. E. Karniadakis, "Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations," *Journal of Computational Physics*, vol. 378, pp. 686–707, Feb. 1, 2019, ISSN: 0021-9991. DOI: 10.1016/j.jcp.2018.10.045. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0021999118307125 (visited on 10/24/2024).

[127] D. Silver, T. Hubert, J. Schrittwieser, *et al.*, *Mastering chess and shogi by self-play with a general reinforcement learning algorithm*, _eprint: 1712.01815, 2017. [Online]. Available: https://arxiv.org/pdf/1712.01815.pdf.

[128] J. Parker-Holder, R. Rajan, X. Song, *et al.*, "Automated reinforcement learning (AutoRL): A survey and open problems," *Journal of Artificial Intelligence Research*, vol. 74, pp. 517–568, Jun. 1, 2022, ISSN: 1076-9757. DOI: 10.1613/jair.1.13596. arXiv: 2201.03916[cs]. [Online]. Available: http://arxiv.org/abs/2201.03916 (visited on 10/07/2024).

[129] L. Li, K. Jamieson, G. DeSalvo, A. Rostamizadeh, and A. Talwalkar, *Hyperband: A novel bandit-based approach to hyperparameter optimization*, Jun. 18, 2018. DOI: 10.48550/arXiv.1603.06560. arXiv: 1603.06560[cs,stat]. [Online]. Available: http://arxiv.org/abs/1603.06560 (visited on 10/07/2024).

[130] T. Akiba, S. Sano, T. Yanase, T. Ohta, and M. Koyama, "Optuna: A next-generation hyperparameter optimization framework," in *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, ser. KDD '19, New York, NY, USA: Association for Computing Machinery, Jul. 25, 2019, pp. 2623–2631, ISBN: 978-1-4503-6201-6. DOI: 10.1145/3292500.3330701. [Online]. Available: https://doi.org/10.1145/3292500.3330701 (visited on 10/20/2024).

[131] R. Liaw, E. Liang, R. Nishihara, P. Moritz, J. E. Gonzalez, and I. Stoica, *Tune: A research platform for distributed model selection and training*, Jul. 13, 2018. DOI: 10.48550/arXiv.1807.05118. arXiv: 1807.05118. [Online]. Available: http://arxiv.org/abs/1807.05118 (visited on 10/20/2024).

[132] F. Chollet, *Keras*, 2015. [Online]. Available: https://keras.io.

[133] T. Eimer, M. Lindauer, and R. Raileanu, "Hyperparameters in reinforcement learning and how to tune them," *arXiv.org*, Jun. 2, 2023. [Online]. Available: https://arxiv.org/abs/2306.01324v1 (visited on 10/07/2024).

[134] S. Falkner, A. Klein, and F. Hutter, *BOHB: Robust and efficient hyperparameter optimization at scale*, Jul. 4, 2018. DOI: 10.48550/arXiv.1807.01774. arXiv: 1807.01774[cs,stat]. [Online]. Available: http://arxiv.org/abs/1807.01774 (visited on 10/07/2024).

[135] K. Eggensperger, P. Müller, N. Mallik, *et al.*, *HPOBench: A collection of reproducible multi-fidelity benchmark problems for HPO*, Oct. 6, 2022. DOI: 10.48550/arXiv.2109.06716. arXiv: 2109.06716[cs]. [Online]. Available: http://arxiv.org/abs/2109.06716 (visited on 10/07/2024).

[136] N. Awad, N. Mallik, and F. Hutter, "DEHB: Evolutionary hyperband for scalable, robust and efficient hyperparameter optimization," *arXiv.org*, May 20, 2021. [Online]. Available: https://arxiv.org/abs/2105.09821v2 (visited on 10/07/2024).

[137] M. Lindauer, K. Eggensperger, M. Feurer, *et al.*, "SMAC3: A versatile bayesian optimization package for hyperparameter optimization," *Journal of Machine Learning Research*, vol. 23, no. 54, pp. 1–9, 2022, ISSN: 1533-7928. [Online]. Available: http://jmlr.org/papers/v23/21-0888.html (visited on 10/07/2024).

[138] P. Virtanen, R. Gommers, T. E. Oliphant, *et al.*, "SciPy 1.0: Fundamental algorithms for scientific computing in python," *Nature Methods*, vol. 17, no. 3, pp. 261–272, Mar. 2, 2020, ISSN: 1548-7091, 1548-7105. DOI: 10.1038/s41592-019-0686-2. [Online]. Available: https://www.nature.com/articles/s41592-019-0686-2 (visited on 09/24/2024).

[139] L. Devroye, *Non-Uniform Random Variate Generation*. New York, NY: Springer, 1986, ISBN: 978-1-4613-8645-2 978-1-4613-8643-8. DOI: 10.1007/978-1-4613-8643-8. [Online]. Available: http://link.springer.com/10.1007/978-1-4613-8643-8 (visited on 09/24/2024).

[140] M. Eberl, "Fisher-yates shuffle," *Arch. Formal Proofs*, 2016. [Online]. Available: https://www.semanticscholar.org/paper/Fisher-Yates-shuffle-Eberl/5e24ffebdc35e8e11af823505cbd5c6d5407f23e (visited on 09/19/2024).

[141] D. Knuth, *The Art of Computer Programming*, 3rd Ed. USA: Addison-Wesley Longman Publishing Co., Inc., 1997, vol. Volume 2: Seminu- merical Algorithms. ISBN: 0-201-89684-2. [Online]. Available: https://www-cs-faculty.stanford.edu/~knuth/taocp.html (visited on 09/30/2024).

[142] L. E. Blumenson, "A derivation of n-dimensional spherical coordinates," *The American Mathematical Monthly*, vol. 67, no. 1, pp. 63–66, 1960, Publisher: [Taylor & Francis, Ltd., Mathematical Association of America], ISSN: 0002-9890. DOI: 10.2307/2308932. [Online]. Available: https://www.jstor.org/stable/2308932 (visited on 09/30/2024).

[143] G. W. Snecdecor and W. G. Cochran, *Statistical Methods, 8th Edition | Wiley*, 8th. Wiley-Blackwell, 1989, 524 pp., ISBN: 978-0-8138-1561-9. (visited on 09/30/2024).

[144] M. L. McHugh, "The chi-square test of independence," *Biochemia Medica*, vol. 23, no. 2, pp. 143–149, 2013, ISSN: 1330-0962. DOI: 10.11613/bm.2013.018.

[145] H. L. Gantt, *Work, Wages, and Profits*. Engineering Magazine Company, 1913, 340 pp., Google-Books-ID: p2ANAAAAYAAJ.

[146] D. J. Anderson, *Kanban: Successful Evolutionary Change for Your Technology Business*. Blue Hole Press, 2010, 262 pp., Google-Books-ID: RJ0VUkfUWZkC, ISBN: 978-0-9845214-0-1.

A Statistical Modelling of Space Debris Clouds

The generation of realistic synthetic datasets is fundamental for training Deep Learning (DL) policies using RL. A statistical modelling procedure is required to create realistic models of space debris clouds. This section presents the statistical modelling procedure used, as well as the procedure used to impose bounds on the resulting statistical models, such as to ensure synthetic datasets are realistic.

A.1 Statistical Modelling Procedure

An optimal (in terms of goodness-of-fit) statistical model of the translational state of a debris cloud is required to generate realistic synthetic debris datasets. These datasets are fundamental for the training of routing policies using RL. The state of the cloud is described using Keplerian elements (which is practical and common practice when analyzing debris clouds [2]). The model must be implemented using the PyTorch[1] library [96] DL library.

The process begins with the characterization of the Keplerian orbital elements of the debris objects in orbit. This involves fitting multiple parametric statistical distributions to each orbital element in the debris cloud dataset. The fitting process is conducted using the SciPy[2] Python library [138], as it provides a better interface for parametric model fitting than PyTorch. Candidate distributions encompass all distributions available both in both SciPy and PyTorch. The candidate distributions can be seen in Tab. A.1.

The goodness-of-fit for each fitted distribution is assessed using the Kolmogorov-Smirnov (KS) statistic, which quantifies the maximum discrepancy between the empirical Cumulative Distribution Function (CDF) of the observed data and the theoretical CDF of the proposed model[3] [79]. The distribution that minimizes the KS statistic for each orbital element is selected as the optimal model. The result is an aggregate model of the translational state of the cloud suitable for generating training data for RL algorithms.

Fig. A.1 shows the statistical model created for the Iridium 33 cloud. The circular Von Mises distribution is chosen for RAAN, AOP and true anomaly.

[1] https://pytorch.org

[2] https://scipy.org

[3] https://www.itl.nist.gov/div898/handbook/eda/section3/eda35g.htm

Table A.1: Single-variable distributions available in SciPy and PyTorch respectively. HT: Heavy-tailed.

Name	**HT**	**SciPy**	**PyTorch**	**Description**
Beta		`beta`	`Beta`	Skewed, used in Bayesian statistics
Cauchy	✗	`cauchy`	`Cauchy`	Symmetric, heavy-tailed
Chi-squared		`chi2`	`Chi2`	Skewed, used in hypothesis testing
Exponential		`expon`	`Exponential`	Skewed, modeling time between events
Fisher-Snedecor	✗	`f`	`FisherSnedecor`	Skewed, used in ANOVA and variance analysis
Gamma		`gamma`	`Gamma`	Skewed, modeling waiting times
Gumbel		`gumbel_<r,l>`	`Gumbel`	Skewed, used in extreme value theory
Half-Cauchy	✗	`halfcauchy`	`HalfCauchy`	Skewed, used as priors in Bayesian statistics
Laplace		`laplace`	`Laplace`	Symmetric, used in signal processing
Log-normal		`lognorm`	`LogNormal`	Skewed, modeling multiplicative processes
Normal		`norm`	`Normal`	Symmetric, widely used in statistics
Pareto	✗	`pareto`	`Pareto`	Skewed, modeling wealth distribution
Uniform		`uniform`	`Uniform`	Symmetric, modeling equal probability across range
Von Mises		`vonmises`	`VonMises`	Symmetric, used for circular data
Weibull		`weibull_<min,max>`	`Weibull`	Skewed, used in reliability and survival analysis
Inverse Gamma	✗	`invgamma`	`Inversegamma`	Skewed, used in Bayesian statistics as priors
Student's t	✗	`t`	`StudentT`	Symmetric, robust to outliers in regression

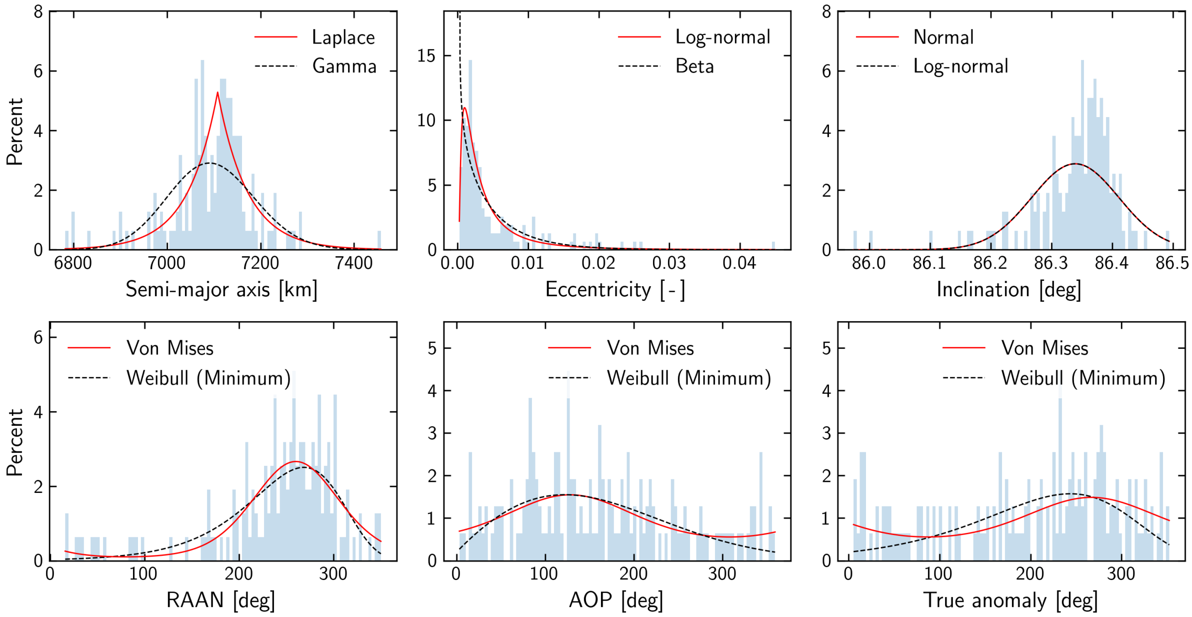


Figure A.1: Statistical model of the Iridium 33 debris cloud.

A.2 Imposing Bounds on Statistical Models of Debris Clouds

The statistical models resulting from the procedure may make use of heavy-tailed distributions, which may generate values very far from the mean of the distribution. To ensure the synthetic datasets created with space debris cloud models are realistic it is of vital importance that the models be bounded to feasible debris states. A bounding procedure is necessary, which must not impact the statistical properties of the models. A combination of Inverse Transform Truncated Sampling (ITTS) and Vectorized Rejection Sampling (VRS) is used to impose bounds on cloud models. Element values are bounded to the extremal values found in the real dataset. ITS and VRS are discussed next, followed by the combined algorithm (Algorithm 13).

A.2.1 Inverse Transform Truncated Sampling

Inverse Transform Sampling is a fundamental method for generating samples at random from any probability distribution given its CDF [97]. Formally, given a target distribution with CDF $F_X(x)$, ITS operates by drawing a uniform random variable $U \sim \mathcal{U}(0,1)$ and computing the sample $X = F_X^{-1}(U)$. This technique leverages the property that $F_X(X) \sim \mathcal{U}(0,1)$, ensuring that the generated samples X adhere to the desired distribution [139]. ITTS consists in imposing sampling bounds $[a, b]$ on an arbitrary distribution, by sampling U uniformly within the CDF range $[F_X(a), F_X(b)]$, and applying the Inverse CDF (ICDF) to obtain X.

Algorithm 11: Inverse Transform Truncated Sampling

```
Input: CDF $F_X(x)$, bounds $a, b$, sample shape $k$
Output: Tensor of truncated samples $X$
$U \sim \mathcal{U}(F_X(a), F_X(b))$ with shape $k$
$X \leftarrow F_X^{-1}(U)$
return $X$
```

A.2.2 Vectorized Rejection Sampling

Rejection Sampling is a sampling technique that imposes bounds on sample values without altering the statistical properties of the underlying distribution [98]. This is achieved by repeatedly generating candidate samples from the target distribution and rejecting those that fall outside the specified range $[a, b]$, continuing until n valid samples are obtained [98]. VRS (Algorithm 12) parallelizes the traditional rejection sampling method, making it highly suitable for GPU-accelerated environments. VRS involves initializing a sample tensor of shape k filled with `NaN`s, and iteratively: replacing `NaN` entries with new samples, and overwriting samples outside $[a, b]$ with `NaN`. The procedure continues for as long as `NaN` entries exist in X. The approach replaces loops with batch boolean, sampling and allocation operations.

Algorithm 12: Vectorized Rejection Sampling

```
Input: Distribution $D$, bounds $a, b$, sample shape $k$
Output: Tensor of Tensor of truncated samples $X$
$X \leftarrow$ tensor of shape $k$ filled with NaN
while NaN in $X$ do
    $m \leftarrow$ number of NaN entries in $X$
    $S \sim D$ with size $m$
    $X[X == \text{NaN}] \leftarrow S$
    $X[X < a$ or $X > b] \leftarrow$ NaN
return $X$
```

A.2.3 Hybrid Truncated Distribution Sampling

Effective sampling from truncated distributions within PyTorch necessitates the availability and implementation of the inverse CDF (ICDF) of the target distribution. When the ICDF is defined, ITTS (Algorithm 11) offers a direct and efficient sampling pathway. In the absence of an ICDF, VRS (Algorithm 12) provides a robust alternative. Algorithm 13 dynamically selects the appropriate sampling strategy based on the availability of the `icdf`, prioritizing ITTS for distributions with an implemented `icdf` and defaulting to VRS otherwise.

Algorithm 13: Hybrid Truncated Distribution Sampling

```
Input: Distribution $D$, bounds $a, b$, sample shape $k$
Output: Tensor of truncated samples $X$
if $D$ has as icdf method then
    $X \leftarrow$ ITTS$(D, a, b, k)$                    // Algorithm 11
else
    $X \leftarrow$ VRS$(D, a, b, k)$                     // Algorithm 12
return $X$
```

B Statistical Models for Permutations

High quality population sampling is crucial for heuristic global optimization algorithms which are sensitive to initialization conditions [84]. Examples are population-based algorithms and algorithms with memory mechanisms such as Simulated Annealing [85].

Population sampling in the case of CO problems involves sampling permutations $\sigma \in \mathfrak{S}^n$ of length n. Uniform permutation sampling is used to cover the search space as widely as possible. Relevant algorithms are the Fisher-Yates Shuffle (FYS) [140], Knuth's algorithm using Sobol points (KS) [84], [141] and Sobol Permutations (SP) [84]. A fourth algorithm is proposed: uniformly sampling the $[0, 1)^n$ hypercube using Sobol points and applying an `argsort` operation to obtain uniformly sampled permutations of $\mathfrak{S}^n$. This algorithm is found to yield more uniform samples than FYS and KS while being orders of magnitude faster than SP. Notably, advanced approaches and hand-crafted heuristics are often used in CO to determine near-optimal solutions prior to heuristic optimization [2], [13]. Distance-based permutation sampling is used to leverage known approximate solutions, using the Mallows Model [86]–[88] under the Hamming distance. Empirical results show that a combination of both approaches is best to balance the exploration of the search space and the exploitation of known approximate solutions.

This appendix defines and compares the statistical models for permutations considered in this work. Uniform statistical models for permutations are discussed first in Section B.1, and distance-based statistical models for permutations are discussed in Section B.2.

B.1 Uniform Models

This section introduces the four uniform permutation sampling models considered in this work. FYS is discussed in Section B.1.1, SP in Section B.1.3, KS in Section B.1.2 and SHP in Section B.1.4. Lastly, Section B.1.5 discusses empirical results and the choice of uniform permutation sampling algorithm used throughout this work.

B.1.1 Fisher-Yates Shuffle

The FYS (Algorithm 14) generates uniformly random permutations of a finite sequence in linear time by iteratively swapping each element with a randomly selected element from its current position to the end of the sequence [140].

B.1.2 Knuth's Algorithm

The `argsort` operation is effectively a map between a Euclidean space $\mathbb{R}^n$ and a permutation group $\mathfrak{S}^n$. This transformation has radial symmetry, breaking up the unit hypersphere into Voronoi cells of equal area. Knuth's algorithm leverages this connection to obtain uniformly sampled permutations. The algorithm is summarized in Algorithm 15. Our implementation makes use of Sobol points to obtain uniformly sampled points in $\mathbb{R}^{n-1}$. The

Algorithm 14: Fisher-Yates Uniform Permutation Sampling

```
Input: Permutation $A$ of length $n$
Output: Uniformly randomized permutation
for $i \leftarrow \{2, ..., n\}$ do
    $j \leftarrow$ Integer sampled u.a.r. in $[1, i]$
    Swap $A[i]$ and $A[j]$
return $A$
```

projection matrix $\hat{U}$ is defined as follows in Eq. B.1 [84]:

$$\hat{U} = \begin{bmatrix} \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} & 0 & 0 & \cdots & 0 \\ \frac{1}{\sqrt{6}} & \frac{1}{\sqrt{6}} & -\frac{2}{\sqrt{6}} & 0 & \cdots & 0 \\ \frac{1}{\sqrt{12}} & \frac{1}{\sqrt{12}} & \frac{1}{\sqrt{12}} & -\frac{3}{\sqrt{12}} & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ \frac{1}{\sqrt{(d-1)d}} & \frac{1}{\sqrt{(d-1)d}} & \frac{1}{\sqrt{(d-1)d}} & \cdots & \frac{1}{\sqrt{(d-1)d}} & -\frac{(d-1)}{\sqrt{(d-1)d}} \end{bmatrix} \tag{B.1}$$

The algorithm relies on the following geometric argument: the Euclidean distances between the vertexes of the $\mathfrak{S}^n$ permutohedron are related to permutation distances, ergo points uniformly sampled in Euclidean space will result in uniformly distant permutations under the Kendall-τ, Spearman distances, and other permutation distances.

Algorithm 15: Knuth's algorithm for uniformly sampling permutations.

```
Input: $m$, $n$
Output: $\Pi$
for $i \leftarrow 1$ to $m$ do
    $x \leftarrow$ SobolPoint$(i, n, d)$                // $x \in (-1, 1)^{d-1}$
    $y \leftarrow \frac{x}{\|x\|}$
    $z \leftarrow \hat{U}^T y$                          // $z \in \mathbb{R}^n$
    $\Pi_i \leftarrow$ argsort$(z)$
return $\Pi$
```

B.1.3 Sobol Permutations

Mitchell et al. [84] leverage the relationship between the $\mathbb{S}^{d-1}$ hypersphere and $\mathfrak{S}^n$ used by Knuth [141] to obtain uniform permutation samples with superior statistical properties. The hypersphere $\mathbb{S}^{d-2} = \{x \in \mathbb{R}^{d-1} \colon \|x\| = 1\}$ that inscribes the $\mathfrak{S}^n$ permutahedron is used as a continuous relaxation of $\mathfrak{S}^n$. Mitchell et al. then use Sobol points evenly spaced over the surface of $\mathbb{S}^{d-2}$ to obtain high quality uniform permutation samples. The procedure is described in Algorithm 16. Sobol points are sampled uniformly from the hypercube $[0, 1)^{d-2}$, transformed to uniformly distributed Blumenson generalized spherical coordinates [142] by means of an inverse CDF, projected to the hyperplane inscribing the hypersphere, and transformed to permutations by applying an `argsort`.

Finding the inverse CDF of the Blumenson generalized spherical coordinates means solving a set of $d - 2$

Algorithm 16: Uniform permutation sampling using Sobol permutations, adapted from Mitchell et al. [84].

```
Input: $m$, $n$
Output: $\Pi$
for $i \leftarrow 1$ to $m$ do
    $x \leftarrow \texttt{SobolPoint}(i, n, d)$                // $x \in [0, 1)^{d-2}$
    $\phi \leftarrow \mathbf{0}$
    for $j \leftarrow 1$ to $d - 2$ do
        $\phi_j \leftarrow F_j^{-1}(x_j)$                      // Inverse CDF
    $y \leftarrow \texttt{PolarToCartesian}(1, \phi)$           // $y \in \mathbb{R}^{d-1}$
    $z \leftarrow \hat{U}^T y$                                   // $z \in \mathbb{R}^n$
    $\Pi_i \leftarrow \text{argsort}(z)$
return $\Pi$
```

transcendental equations, as the PDF of the generalized spherical coordinates is:

$$f(\phi_j) = \begin{cases} \dfrac{1}{2\pi} & j = d-2 \\ \dfrac{1}{B(\frac{d-j-1}{2}, \frac{1}{2})} \sin^{d-j-2}(\phi_j) & 1 \leq j < d-2 \end{cases} \tag{B.2}$$

where B is the beta function [142]. Mitchell et al. propose using root finding methods to obtain the inverse CDF. This comes at great computational cost for even moderately long sequences and sample sizes. Eliminating the transformation to generalized spherical coordinates means that the explicit discrepancy convergence rates of Sobol points do not translate to $\mathfrak{S}^n$ [84], hampering the quality of samples, and so is not an option.

B.1.4 Directly Sampling the Unit Hypercube in $\mathbb{R}^n$

Both Knuth's algorithm and SP rely on hyperspheres, as the Euclidean distance between points in the hypersphere that inscribes a permutohedron $\mathfrak{S}^n$ are closely related to various distances metrics between permutations of $\mathfrak{S}^n$. We propose an alternative: directly sampling the unit hypercube in $\mathbb{R}^n$ and applying an `argsort` operation to obtain permutations of $\mathfrak{S}^n$. We make use of Sobol points to sample the $\mathbb{R}^n$ hypercube. The algorithm will be referred to as Sobol Hypercube Permutations (SHP).

The algorithm relies on the following argument of probability: as the `argsort` operation bins $\mathbb{R}^n$ into Voronoi cells of equal volume, an algorithm that samples $\mathbb{R}^n$ with uniform probability density should sample all permutations of $\mathfrak{S}^n$ with equal likelihood. The algorithm is described in Algorithm 17.

Algorithm 17: Uniform permutation sampling using Sobol points in the $[0, 1)^n$ hypercube.

```
Input: $m$, $n$
Output: $\Pi$
for $i \leftarrow 1$ to $m$ do
    $x \leftarrow \texttt{SobolPoint}(i, n, d)$                // $x \in [0, 1)^d$
    $y \leftarrow \frac{x}{\|x\|}$
    $\Pi_i \leftarrow \text{argsort}(y)$
return $\Pi$
```

B.1.5 Sample Quality and Optimization Performance

Sample quality is determined by its uniformity. Sample uniformity is evaluated using the Chi-Squared test [143], [144]. Uniformity is evaluated considering the frequency with which each number $x \in [0, n]$ appears in each index of the sampled permutations of $\mathfrak{S}^n$. The expected distribution is uniform: at index 0 of all permutations sampled from $\mathfrak{S}^n$ we expect to find every number $x \in [0, n]$ with equal probability.

The null hypothesis of the Chi-Squared test is that the distribution underlying the observations is uniform, and that the difference between the observed and expected frequencies is due to random chance. The Chi-Squared test p-value is the probability that the null hypothesis in this case is true: that is, that the discrepancies between expected and observed frequencies is due to random chance. A small p-value (< 0.05) indicates that it is unlikely that the underlying distribution is the expected uniform distribution.

Fig. B.1 shows the uniformity of samples of 1000 permutations of $\mathfrak{S}^{100}$ obtained using FYS, Knuth's algorithm using Sobol points, SP and SHP. The computation time of the three algorithms was of approximately 1 [ms] for FYS, Knuth's algorithm and SHP, and 150 [s] for SP. The uniformity of the samples produced by the four algorithms is comparable, while SHP, Knuth's algorithm and FYS are orders of magnitude faster than SP. SP is discarded due to its large computational cost and marginal benefit.

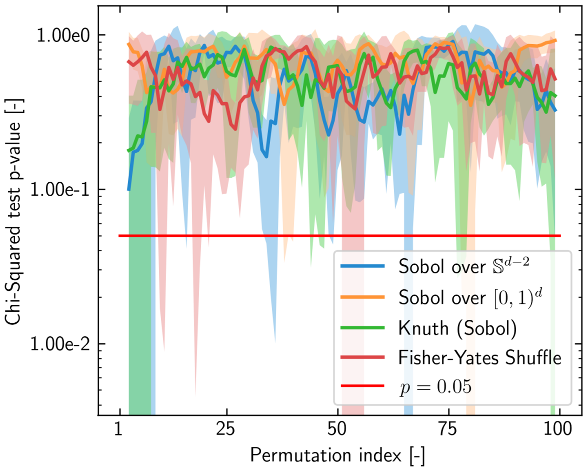


Figure B.1: Performance comparison of FYS (red), Knuth's algorithm using Sobol points (green), SP and SHP (ours, orange). 1000 samples of $\mathfrak{S}^{100}$.

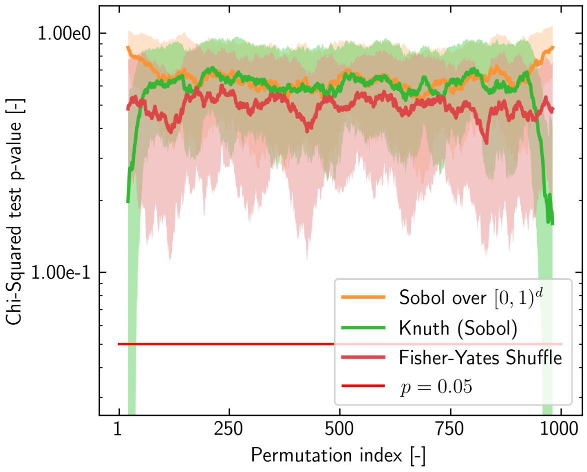


Figure B.2: Performance comparison of FYS, Knuth's algorithm using Sobol points and SHP. 10,000 samples of $\mathfrak{S}^{1000}$.

The performance of Knuth's algorithm, SHP and FYS was evaluated for sampling larger batches of longer sequences. Fig. B.2 shows the uniformity of samples of 10,000 permutations of $\mathfrak{S}^{1000}$. At similar runtimes SHP delivers superior uniformity, especially at the extremes. FYS has inferior performance across the board. Knuth's algorithm behaves well, but sample uniformity quickly degrades close to the extremes. This was to a certain degree expected from the remarks of Mitchell et al. [84].

Empirical results showed no statistically relevant change in optimization performance from sampling initial populations using FYS as opposed to SHP. SHP is used for uniform permutation sampling in this work due to its high speed and the superior statistical properties of the permutation samples obtained with it. That being said, FYS is a simple and cheap alternative to obtain uniformly sampled permutations for combinatorial optimization.

B.2 Distance-based Models

The Mallows model [87] (and variants), first introduced in 1957 [86], is the most relevant distance-based statistical model for permutations [88]. It assigns permutations a probability $p(\sigma)$ based on their proximity to a central permutation σ_0, as measured by some distance metric, and a dispersion parameter θ. The Mallows model is defined in Eq. B.3. Permutations are exponentially less probable as they stray further from σ_0.

$$p(\sigma) = \frac{\exp(-\theta d(\sigma, \sigma_0))}{\psi(\theta)} \tag{B.3}$$

The normalization constant $\psi(\theta) = \sum_\sigma \exp(-\theta d(\sigma))$ can be expressed as the sum in Eq. B.4, where $S_h(n, d)$ is the number of permutations of length n at a given distance d from the central permutation:

$$\psi(\theta) = \sum_{d=0}^{n} S_h(n, d) \exp(-\theta d) \tag{B.4}$$

Choosing a permutation distance metric relevant for the problem at hand is thus vital. We make use of the Hamming distance in this case, defined in Eq. B.5. The Hamming distance for rankings is a metric that quantifies the dissimilarity between two permutations by counting the number of positions at which the corresponding elements differ. The number of permutations $S_h(n, d)$ of length n at a Hamming distance d of any other permutation is defined in Eq. B.6, where S_d (Eq. B.7) is the number of *derangements* (permutations with no fixed point) of length d [88].

$$d_H(\pi, \sigma) = \sum_{i=1}^{n} \mathbb{I}(\pi(i) \neq \sigma(i)) \tag{B.5}$$

$$S_h(n, d) = \binom{n}{d} S(d) \tag{B.6}$$

$$S(d) = \begin{cases} 1 & \text{if } d = 0, \\ 0 & \text{if } d = 1, \\ (d-1)[S(d-1) + S(d-2)] & \text{otherwise.} \end{cases} \tag{B.7}$$

Algorithm 18: Sampling Permutations via Mallows Model

```
Input: Number of samples $m$, permutation length $n$, dispersion parameter $\theta$, central permutation $\sigma_0$
Output: sample $\in \mathbb{S}_n^m$
for $d \leftarrow 0$ to $n$ do
    $S_h \leftarrow \binom{n}{d} \cdot S(d)$
    $p(d) \leftarrow S_h e^{-\theta d}$
$\psi(\theta) \leftarrow \sum_{d=0}^{n} p(d)$
$\forall\, d, p(d) \leftarrow \frac{p(d)}{\psi(\theta)}$
$\{d_i\}_{i=1}^{m} \leftarrow \text{Multinomial}(m, \{p(d)\}_{d=0}^{n})$
for $i \leftarrow 1$ to $m$ do
    Select $d_i$ indices $F \subseteq \{1, \dots, n\}$ uniformly at random
    Permute $F$ as a derangement $\pi$
    $\sigma_i \leftarrow \sigma_0$
    $\sigma_i[F] \leftarrow \pi$
    sample$[i] \leftarrow \sigma_i$
return sample
```

C Software Architecture and Development

This chapter discusses the architecture and development process of the software developed in this work. Section C.1 discusses the architecture of the modular STSP solver developed. Section C.2 describes its four principal modules: the Impulsive Trajectory Design module, the Low Thrust Trajectory Design module, the Heuristic Combinatorial Optimization module and the Neural Combinatorial Optimization module. Section C.3 describes the software development process used to implement all software produced in this project, and concludes the chapter with a summary of the code base as of the time of writing.

C.1 Software Architecture

The architecture of the software developed in this work is based on a modular STSP solver concept, described in Fig. C.1. The final aim of this architecture is to enable the construction of tailorable, modular STSP solvers capable of offering feasibility guarantees for the generated trajectories.

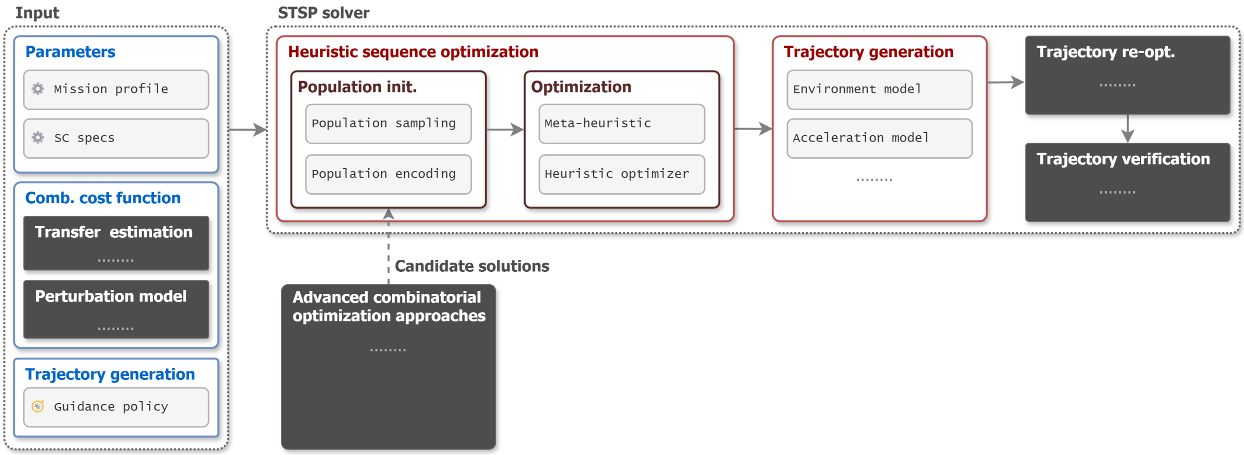


Figure C.1: STSP solver architecture. In black: complex components (integrated using standardized interfaces) the internal structure of which is out of the scope of this diagram.

This final aim is achieved using a 3-stage architecture: target sequence optimization, trajectory generation and re-optimization, and trajectory verification and validation. The architecture itself consists of standardized interfaces between each component. The integration of the different components may be done at the language/module level, or at execution level if the modules are implemented using different languages or programs. In the case of OSSIE the SCP module and the OSSIE Functional Engineering Simulator module, which are implemented in MATLAB. Interfacing these modules with the combinatorial optimization modules was possible thanks to the standardized data interfaces defined. The result is a framework that is highly adaptable to new mission design scenarios, while benefiting from high component reusability.

C.2 Overview of Main Modules

C.2.1 Impulsive Trajectory Optimization

The purpose of the Impulsive Trajectory Optimization module is to generate coplanar and out-of-plane impulsive transfer trajectories under the $J2$ perturbation, considering spacecraft propulsion system requirements. The module was used for the design impulsive ADR maneuvers, to generate trajectories for the OSSIE mission, and to verify impulsive trajectory estimation methods.

C.2.2 Low-Thrust Trajectory Optimization

The purpose of the Low-Thrust Trajectory Optimization module is to generate low-thrust rendezvous maneuvers targeting all six orbital elements using the Q-Law and RQ-Law Lyapunov Feedback Control guidance laws. The module was used to generate trajectories for ADR mission scenarios, to design warm starts for the OSSIE OTV based on Lyapunov Feedback Control, and to validate low-thrust transfer cost estimation methods.

C.2.3 Heuristic Combinatorial Optimization

The purpose of the Heuristic Combinatorial Optimization module is to solve generalized combinatorial optimization problems, specifically multi-rendezvous spacecraft routing problems, using heuristic optimization techniques. The module can leverage advanced solutions to accelerate the heuristic optimization process and includes candidate solution generation methods based on heuristics and tree search methods. The module was used to solve the OSSIE and ADR mission design Spacecraft Traveling Salesman Problems (STSPs).

C.2.4 Neural Combinatorial Optimization

The purpose of the NCO module is to train ML policies for space vehicle routing problems, and specifically NCO policies based on Graph Neural Networks (GNNs) trained using RL. The module was used to train attention-based NCO policies for the OSSIE and ADR STSPs and evaluate the applicability and performance of NCO methods to solve various STSPs.

C.3 Software Development Approach and Codebase Summary

The development process employed adheres to principles of test-driven development (TDD) as outlined by Beck [105]. The methodology is structured as follows:

1. **Module Creation**: Initiate by developing a Python module corresponding to the desired functionality, such as low-thrust trajectory design.

2. **Test Directory Setup**: Establish a `tests` directory. For each new functionality, a specific test file is authored to verify the respective component.

3. **Architectural Planning**: Prior to implementation, the necessary architecture is determined. Ensuring uniform interfaces across all modular components is critical, particularly for impulsive and low-thrust trajectory estimation and generation modules.

4. **Development Workflow**: Implementation proceeds based on the test files, utilizing the Python debugger to examine internal mechanics thoroughly.

5. **Integration and Deployment**: Upon completion of functionality, both the test file and the implementation are committed to the version control system.

C.3.1 Analysis Process

The analysis process diverges from testing by focusing on data processing and management. This involves:

- **Analysis Directory**: Each repository contains an `analysis` directory with scripts designed to perform single, reliable, and reproducible analyses, producing data and visualizations.
- **Granular Extensions**: When extending analyses, tasks are bifurcated to minimize logical complexity and enhance reproducibility.

C.3.2 Software Tools

Software tools integrated into the development include:

- **Version Control**: Git[1] [109] was used for version control.
- **Development Environment**: Microsoft Visual Studio Code (VSCode)[2] was used for development.
- **Computational Resources**: CPU-intensive workloads were executed on the TU Delft Thales server, while Graphics Processing Unit (GPU)-intensive workloads are run using Paperspace[3] and RunPod[4]. RunPod was superior both in terms of ease of development (SSH'ing to the running compute nodes) as well as GPU availability, and is recommended by the author
- **DL Training Analysis**: Weights and Biases[5] (WandB) and TensorBoard[6] were used to monitor NCO training. WandB relies on cloud storage and TensorBoard on local storage. WandB enabled concurrent training on multiple machines reporting to a centralized WandB project, while TensorBoard was overall more reliable and comfortable to use. The ability to concurrently train and analyze multiple runs was too valuable however. WandB was the platform of choice for most of this work.

This structured approach ensures that development is systematically guided by tests, facilitating robust and maintainable code, while the analysis framework supports reproducible research practices.

C.3.3 Codebase Summary

Tab. C.1 summarizes the structure and magnitude of the codebase produced in this work. The summary was obtained using `cloc`[7]. The codebase consits of the following modules:

- **Spacecraft Modelling**: various models of impulsive and low-thrust spacecraft, as well as statistical modelling methods for OTV missions
- **Space Debris Dataset Construction**: methods and scripts to obtain space debris data using Celestrak[8] and the ESA DISCOS (Database and Information System Characterising Objects in Space) database[9]
- **Impulsive Trajectory Design**: implementation, tests and analysis of impulsive trajectory design methods
- **Low Thrust Trajectory Design**: implementation, tests and analysis of Lyapunov Control low-thrust trajectory optimization methods

[1] https://git-scm.com

[2] https://code.visualstudio.com

[3] https://www.paperspace.com

[4] https://www.runpod.io

[5] https://wandb.ai/site

[6] https://www.tensorflow.org/tensorboard

[7] https://github.com/AlDanial/cloc

[8] https://celestrak.org

[9] https://discosweb.esoc.esa.int

- **Neural Combinatorial Optimization**: implementation, tests and analysis of neural combinatorial optimization methods. Implemented in this module: vectorized space debris cloud models, vectorized cost functions, `RL4CO` environments among others

- **Heuristic Combinatorial Optimization**: implementation, tests and analysis of heuristic combinatorial optimization methods. Implemented in this module: permutation sampling methods, statistical modelling procedures for space debris clouds, population sampling and distribution for archipelagos, various approximate solution algorithms, cost functions, `pygmo` problem classes and `pygmo` implementation code among others

Table C.1: Codebase summary.

Module	**Files**	**Blank**	**Comment**	**Code**
Spacecraft Modelling	13	126	141	466
Space Debris Dataset Construction	3	90	146	168
Impulsive Trajectory Design	47	1,029	1,156	2,940
↼ Spacecraft Modelling				
Low Thrust Trajectory Design	91	3,605	4,083	10,907
↼ Spacecraft Modelling				
Heuristic Combinatorial Optimization	295	9,313	12,276	31,938
↼ Spacecraft Modelling				
↼ Space Debris Dataset Construction				
↼ Impulsive Trajectory Design				
↼ Low Thrust Trajectory Design				
Neural Combinatorial Optimization	115	2,705	2,785	10,551
↼ Spacecraft Modelling				
↼ Space Debris Dataset Construction				
Total (unique)	406	11,774	14,914	40,594

D Planning and Management of the Project

The completion of the present work involved the research and development of a fairly large amount of independent and interdependent components and modular interfaces, amounting to over 40,000 lines of Python code when adding implementations, tests and different performance analyses (see Tab. C.1). Furthermore, a considerable amount of development happened along two parallel avenues: impulsive multi-rendezvous trajectory design for the OSSIE OTV, resulting in the paper presented at the 2024 International Astronautics Conference in Milan, and low-thrust multi-rendezvous trajectory design for ADR. Both avenues required separate research, implementation and testing under different mission requirements and different deadlines, while ensuring coherence with the research direction and objectives of the thesis.

The project involved interaction between 3 developing stakeholders: the SENER Aerospace & Defence on-board navigation and control development team working on OSSIE, the UARX Space OSSIE propulsion team (and by proxy the Dawn Aerospace test engineering team in New Zealand), and the author. As in any complex project contingencies took place and plans had to be continuously assessed and adapted to ensure the success of the project. Critically, the long-term project plan was assessed and re-drawn after the Midterm Review, leading to an extension of the duration of the thesis from the original allotted time of 28 weeks to a thesis duration of 37 weeks. 37 weeks was chosen as it is the maximum nominal duration of a full-time MSc thesis project as per the TU Delft graduation guidelines after 2022[1]. All original research goals and project objectives were met by the end of the project.

The author was responsible for the definition of the research direction, the definition of all project objectives with the exception of top-level requirements related to the OSSIE OTV mission, and planning and managing both the long-term and short-term development of the project. The author was also responsible for all external consulting engaged through the project, with the exception of the Midterm Review. The aim of this appendix is to give insight into how the project was planned, how the development of the project was managed, and how the entire process was informed by stakeholders and external consultants to ensure the successful completion of the project. Section D.1 is concerned with the planning of the project and Section D.2 with its management.

D.1 Project Planning

Project planning was divided into a long-term plan and short-term sprints. The key priorities for project planning were to ensure that the planning of short-term tasks was informed by long-term priorities, and to make progress measurable and relatable to long-term goals so as to assess the long-term status of the project on-the-go.

[1] https://www.tudelft.nl/studenten/lr-studentenportal/onderwijs/master/thesis

D.1.1 Long-Term Planning

Planning started by establishing a research direction attractive to TU Delft and a set of project goals attractive for SENER Aerospace & Defence. This resulted in the creation of a long-term project plan and project proposal which was presented to Marc Naeije. The process carried out to establish the research direction, construct an initial project plan was the following:

1. **Literature Review and Research Question Identification**
   A literature review was conducted to survey existing work in the field, leading to the formulation of two relevant research questions and a set of project objectives. With exception of the project objectives related to the OSSIE OTV mission, all research questions and project objectives were independently determined by the author.

2. **High-Level Project Plan Development**
   The high-level project plan was developed by:
   (a) Identifying the necessary research and development tasks to create the space Vehicle Routing Problem (VRP) solver.
   (b) Determining the sequential dependencies among the top-level project components.
   (c) Estimating the duration of each top-level task.
   (d) Creating a Gantt chart to illustrate the development timeline [145].

3. **Project Proposal Preparation and Review**
   The project proposal, including the literature review and development plan, was written by the author. It was reviewed by Jesús Ramírez, supervisor at SENER Aerospace & Defence, and subsequently approved by Marc Naeije, the primary project supervisor.

4. **Initial Project Duration Consideration**
   The original plan estimated a duration of 28 weeks, considerably below the TU Delft MSc Thesis program's thesis project duration of 32-37 weeks.

D.1.2 Short-Term Planning

Weekly and day-to-day planning focused on implementing the high-level project plan through detailed, actionable steps:

1. **Task Identification**
   Specific low-level development tasks were identified for each top-level project component.

2. **Task Estimation**
   The research and development time for each low-level task was estimated based on complexity and size. Estimated completion times were often difficult to estimate accurately.

3. **Priority Setting**
   Tasks were prioritized based on their importance to the project's success and their position within the critical development path.

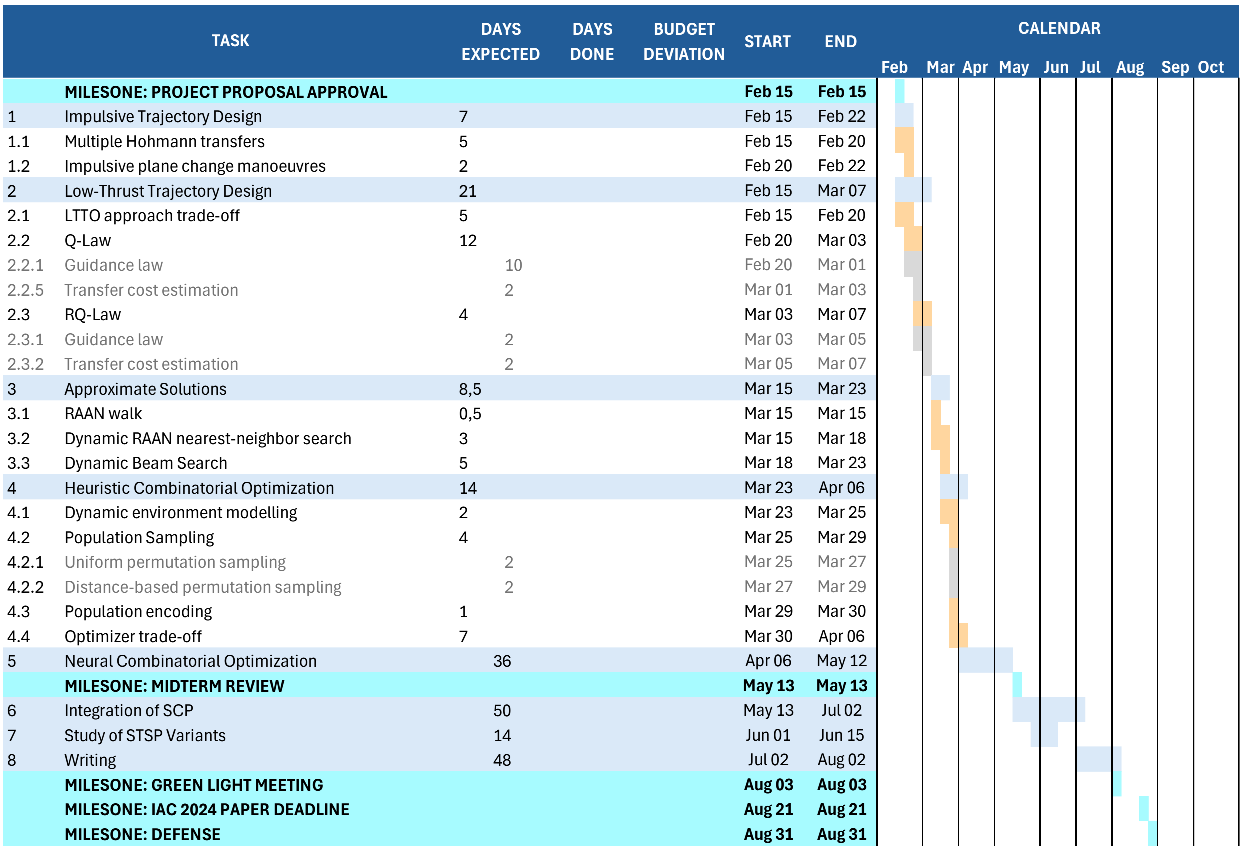
4. **Sprint Allocation**
   Development tasks were assigned to weekly sprints to facilitate iterative progress.

5. **Progress Tracking**
   A Kanban board was utilized to monitor and communicate task statuses [146].

This process was conducted recursively in the case of complex lower level tasks.

D.1.3 Initial Project Plan

The initial project plan can be seen in Fig. D.1. While theoretical understanding existed of the modules that would be required, development uncertainty was large for development blocks farther on in the future, especially for NCO, for which extensive research into DL frameworks was required. The main priority in the initial phases of the project was to start development of blocks with high certainty as early as possible. Early development blocks were planned down to the level of weekly objectives to estimate development time as accurately as possible. This allowed an assessment of the available development time for later blocks. The development time of such blocks was then estimated based on expected complexity.



	TASK	DAYS EXPECTED	DAYS DONE	BUDGET DEVIATION	START	END
	MILESONE: PROJECT PROPOSAL APPROVAL				**Feb 15**	**Feb 15**
1	Impulsive Trajectory Design	7			Feb 15	Feb 22
1.1	Multiple Hohmann transfers	5			Feb 15	Feb 20
1.2	Impulsive plane change manoeuvres	2			Feb 20	Feb 22
2	Low-Thrust Trajectory Design	21			Feb 15	Mar 07
2.1	LTTO approach trade-off	5			Feb 15	Feb 20
2.2	Q-Law	12			Feb 20	Mar 03
2.2.1	Guidance law	10			Feb 20	Mar 01
2.2.5	Transfer cost estimation	2			Mar 01	Mar 03
2.3	RQ-Law	4			Mar 03	Mar 07
2.3.1	Guidance law	2			Mar 03	Mar 05
2.3.2	Transfer cost estimation	2			Mar 05	Mar 07
3	Approximate Solutions	8,5			Mar 15	Mar 23
3.1	RAAN walk	0,5			Mar 15	Mar 15
3.2	Dynamic RAAN nearest-neighbor search	3			Mar 15	Mar 18
3.3	Dynamic Beam Search	5			Mar 18	Mar 23
4	Heuristic Combinatorial Optimization	14			Mar 23	Apr 06
4.1	Dynamic environment modelling	2			Mar 23	Mar 25
4.2	Population Sampling	4			Mar 25	Mar 29
4.2.1	Uniform permutation sampling	2			Mar 25	Mar 27
4.2.2	Distance-based permutation sampling	2			Mar 27	Mar 29
4.3	Population encoding	1			Mar 29	Mar 30
4.4	Optimizer trade-off	7			Mar 30	Apr 06
5	Neural Combinatorial Optimization	36			Apr 06	May 12
	MILESONE: MIDTERM REVIEW				**May 13**	**May 13**
6	Integration of SCP	50			May 13	Jul 02
7	Study of STSP Variants	14			Jun 01	Jun 15
8	Writing	48			Jul 02	Aug 02
	MILESONE: GREEN LIGHT MEETING				**Aug 03**	**Aug 03**
	MILESONE: IAC 2024 PAPER DEADLINE				**Aug 21**	**Aug 21**
	MILESONE: DEFENSE				**Aug 31**	**Aug 31**

Figure D.1: Gantt chart of the original top level plan.

D.2 Project Management

Project management involved overseeing the project's execution in accordance with the established plan. It included communicating the project's status, assessing progress with key stakeholders, and re-planning when necessary.

D.2.1 Communication with Stakeholders and External Consultants

Effective communication was essential to ensure all parties were privy to the state of the project and maintaining alignment of all relevant stakeholders with respect to development priorities and project management decisions.

Regular Communication

1. **Weekly Supervisor Meetings**
   Weekly meetings were conducted with Marc Naeije and Jesús Ramírez to discuss progress updates and evaluate the project's current state. These meetings were essential for managing short-term development activities.

2. **Monthly Industrial Presentations**
   Monthly presentations were delivered to the SENER Aerospace & Defence GNC group to report on the project's progress, specifically regarding the multi-rendezvous trajectory optimization problem of the OSSIE OTV.

Irregular Communication

1. **External Consultant Engagement**
   Communications were maintained with external consultants—professors and industry professionals specializing in astrodynamics, statistics, and related fields—to obtain expert insights and validate methodological decisions. The external consultants included:

   - **Javier Roa Vicens (SpaceX)**: Regularization techniques, specifically Sundman transformations.
   - **José Antonio López Ortí (Universidad Jaume I)**: Generalized Sundman anomalies and regularization methods.
   - **Dario Izzo (ESA Advanced Concepts Team)**: STSP research interests and learning algorithms.
   - **Ekhine Irurozki (Universidad del País Vasco, France Telecom)**: Statistical models for permutations.
   - **Dominic Dirkx (TU Delft)**: Regularization vs. variable step integration and verification of the Lyapunov control module.
   - **Steve Gehly (TU Delft)**: Hybrid ML-tree search approaches for STSP and sensor allocation problems.
   - **Regino Criado Herrero (Universidad Rey Juan Carlos)**: Statistics for permutation groups.
   - **Erik-Jan van Kampen (TU Delft)**: RL applications to dynamic STSPs and performance measurement of ML policies.
   - **Máximo Fernández (SENER Aerospace & Defence)**: ML for regression and Kolmogorov-Arnold Networks.

2. **Private Management Meetings**
   Private meetings were held with upper management at SENER Aerospace & Defence to evaluate the feasibility of integrating project developments into the company's commercial offerings in on-board GNC and mission planning markets.

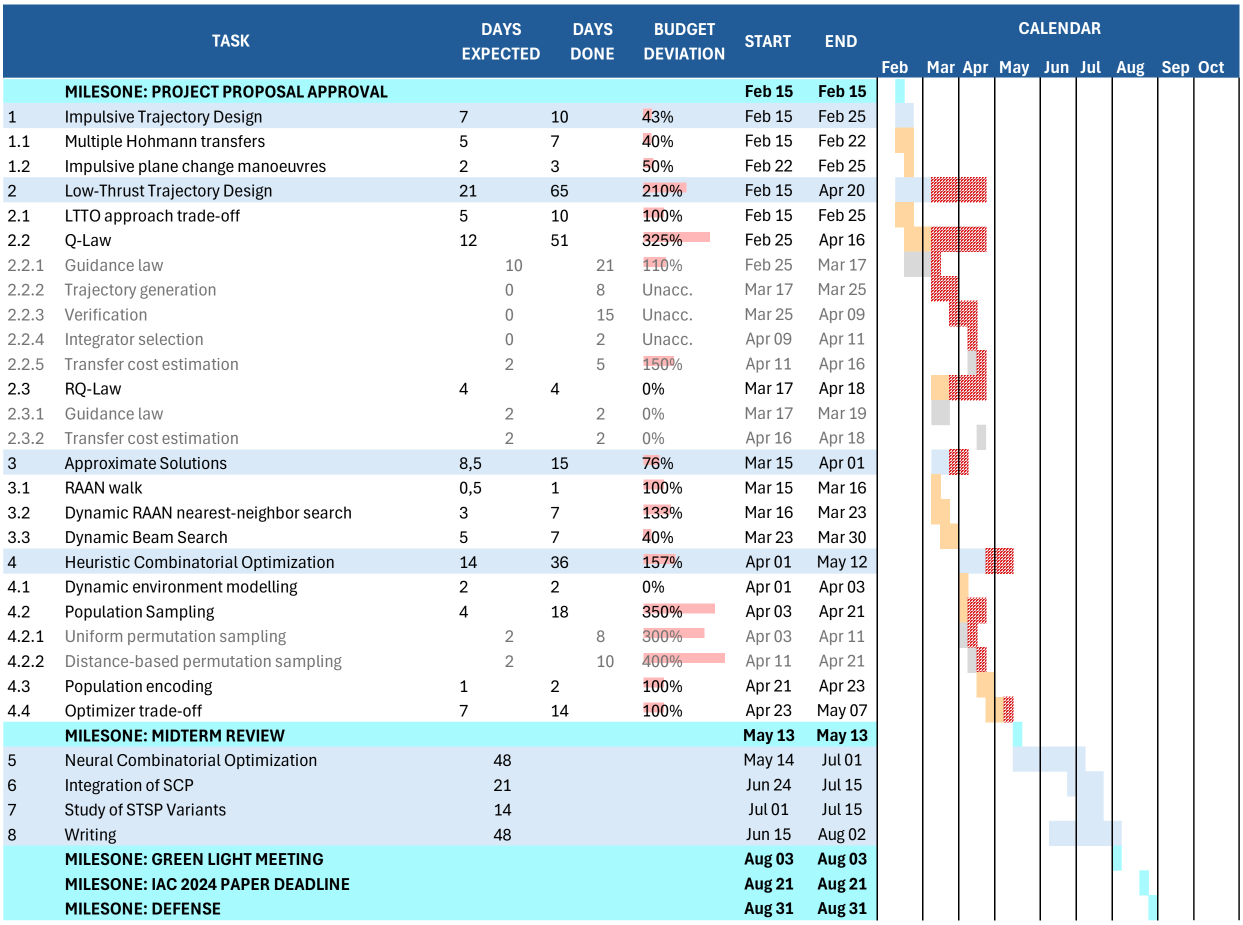
3. **Midterm Review**
   A Midterm Review was conducted in Delft on May 13, 2024, consisting of a formal presentation of the state of the project in relation to its final objectives and project plan, followed by a discussion. Present were TU Delft faculty members Erwin Mooij, Dominic Dirkx, Marc Naeije, and SENER Aerospace & Defence supervisor Jesús Ramírez. The Midterm Review served as a checkpoint to assess the health of the project and make necessary adjustments to the project plan.

D.2.2 Project Assessment and Re-planning

Regular assessments were conducted to monitor progress, evaluate alignment with objectives, and identify necessary adjustments. These assessments were categorized into short-term and long-term evaluations, guiding re-planning strategies and facilitating effective communication with stakeholders.

Short-Term Assessment and Re-planning

Weekly assessments reviewed immediate project progress, completed tasks, identified challenges, and adjusted priorities. Short-term tasks were re-planned in weekly sessions to address progress and emerging issues, ensuring resource allocation to high-priority tasks.



	TASK	DAYS EXPECTED	DAYS DONE	BUDGET DEVIATION	START	END
	MILESONE: PROJECT PROPOSAL APPROVAL				**Feb 15**	**Feb 15**
1	Impulsive Trajectory Design	7	10	43%	Feb 15	Feb 25
1.1	Multiple Hohmann transfers	5	7	40%	Feb 15	Feb 22
1.2	Impulsive plane change manoeuvres	2	3	50%	Feb 22	Feb 25
2	Low-Thrust Trajectory Design	21	65	210%	Feb 15	Apr 20
2.1	LTTO approach trade-off	5	10	100%	Feb 15	Feb 25
2.2	Q-Law	12	51	325%	Feb 25	Apr 16
2.2.1	Guidance law	10	21	110%	Feb 25	Mar 17
2.2.2	Trajectory generation	0	8	Unacc.	Mar 17	Mar 25
2.2.3	Verification	0	15	Unacc.	Mar 25	Apr 09
2.2.4	Integrator selection	0	2	Unacc.	Apr 09	Apr 11
2.2.5	Transfer cost estimation	2	5	150%	Apr 11	Apr 16
2.3	RQ-Law	4	4	0%	Mar 17	Apr 18
2.3.1	Guidance law	2	2	0%	Mar 17	Mar 19
2.3.2	Transfer cost estimation	2	2	0%	Apr 16	Apr 18
3	Approximate Solutions	8,5	15	76%	Mar 15	Apr 01
3.1	RAAN walk	0,5	1	100%	Mar 15	Mar 16
3.2	Dynamic RAAN nearest-neighbor search	3	7	133%	Mar 16	Mar 23
3.3	Dynamic Beam Search	5	7	40%	Mar 23	Mar 30
4	Heuristic Combinatorial Optimization	14	36	157%	Apr 01	May 12
4.1	Dynamic environment modelling	2	2	0%	Apr 01	Apr 03
4.2	Population Sampling	4	18	350%	Apr 03	Apr 21
4.2.1	Uniform permutation sampling	2	8	300%	Apr 03	Apr 11
4.2.2	Distance-based permutation sampling	2	10	400%	Apr 11	Apr 21
4.3	Population encoding	1	2	100%	Apr 21	Apr 23
4.4	Optimizer trade-off	7	14	100%	Apr 23	May 07
	MILESONE: MIDTERM REVIEW				**May 13**	**May 13**
5	Neural Combinatorial Optimization	48			May 14	Jul 01
6	Integration of SCP	21			Jun 24	Jul 15
7	Study of STSP Variants	14			Jul 01	Jul 15
8	Writing	48			Jun 15	Aug 02
	MILESONE: GREEN LIGHT MEETING				**Aug 03**	**Aug 03**
	MILESONE: IAC 2024 PAPER DEADLINE				**Aug 21**	**Aug 21**
	MILESONE: DEFENSE				**Aug 31**	**Aug 31**

Figure D.2: Gantt chart illustrating the state of the project at the time of the Midterm.

Long-Term Assessment and Re-planning

Periodic evaluations of the project's direction and development priorities were conducted with external consultants to ensure research objectives remained attainable. The Midterm Review on May 13, 2024 was used as a pivot to evaluate and re-draw the project plan as needed.

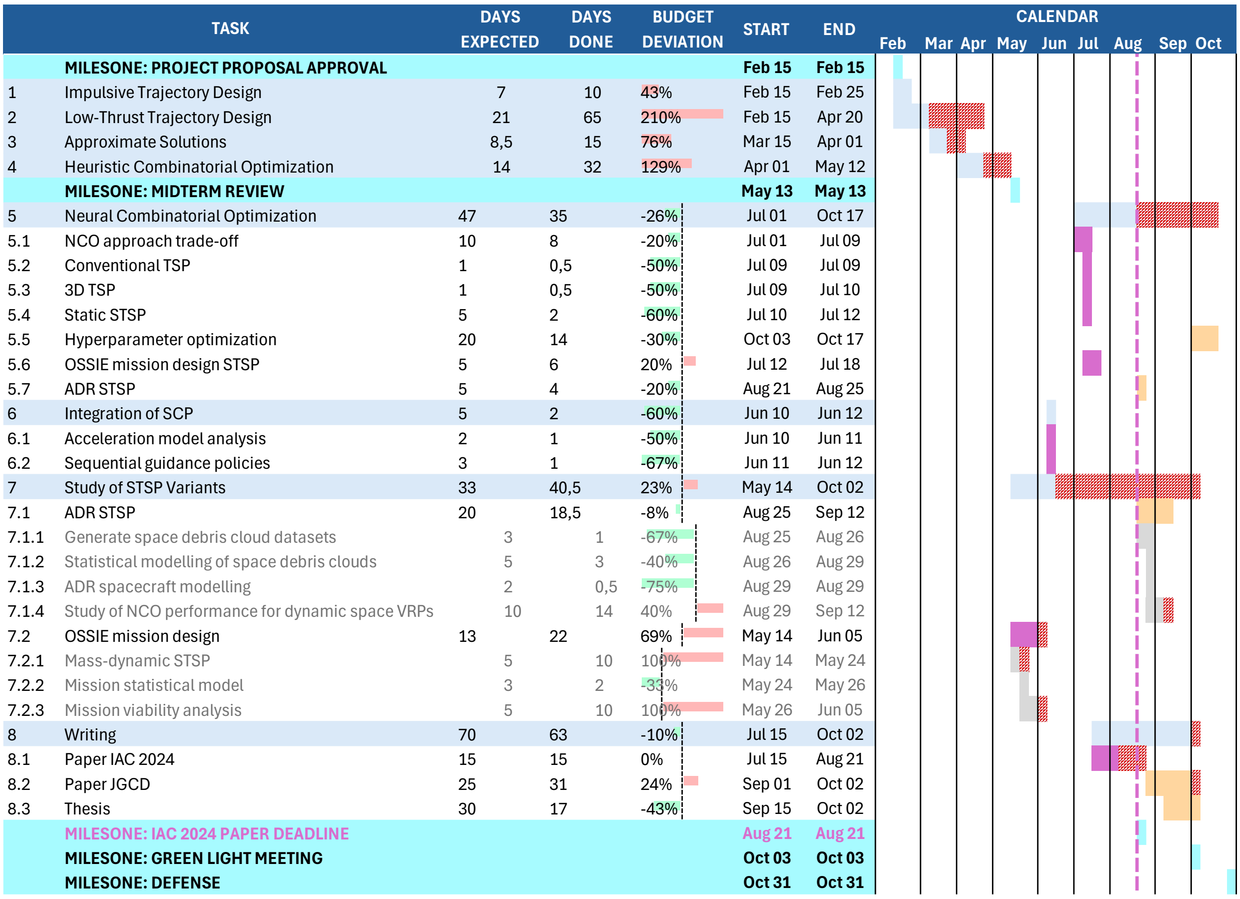
Fig. D.2 illustrates the state of the project at the time of the Midterm. The project suffered from considerable delay. Two major sources of uncertainty had negatively impacted development time estimates: the operational capabilities of the UARX OSSIE OTV, which were only clarified by UARX Space and Dawn Aerospace mid-way through the project, definitively settling for impulsive manoeuvres, and the requirements of the SCP solver. The SCP solver was initially considered capable of generating trajectories without requiring warm-starts. Thus, the allocated development time for impulsive and low-thrust trajectory design did not consider the need to generate and verify realistic trajectories. This caused a major underestimation of the development time required for low-thrust trajectory design, which had cascaded to later development blocks by the time of the Midterm.

Updated Project Plan

The project plan was re-drawn after the Midterm Review with two priorities in mind:

1. Ensuring all tasks required for the 2024 IAC were complete by August the 21st

2. Ensuring all remaining work was completed by October the 3rd, namely all work related to the ADR paper



	TASK	DAYS EXPECTED	DAYS DONE	BUDGET DEVIATION	START	END
	MILESONE: PROJECT PROPOSAL APPROVAL				**Feb 15**	**Feb 15**
1	Impulsive Trajectory Design	7	10	43%	Feb 15	Feb 25
2	Low-Thrust Trajectory Design	21	65	210%	Feb 15	Apr 20
3	Approximate Solutions	8,5	15	76%	Mar 15	Apr 01
4	Heuristic Combinatorial Optimization	14	32	129%	Apr 01	May 12
	MILESONE: MIDTERM REVIEW				**May 13**	**May 13**
5	Neural Combinatorial Optimization	47	35	-26%	Jul 01	Oct 17
5.1	NCO approach trade-off	10	8	-20%	Jul 01	Jul 09
5.2	Conventional TSP	1	0,5	-50%	Jul 09	Jul 09
5.3	3D TSP	1	0,5	-50%	Jul 09	Jul 10
5.4	Static STSP	5	2	-60%	Jul 10	Jul 12
5.5	Hyperparameter optimization	20	14	-30%	Oct 03	Oct 17
5.6	OSSIE mission design STSP	5	6	20%	Jul 12	Jul 18
5.7	ADR STSP	5	4	-20%	Aug 21	Aug 25
6	Integration of SCP	5	2	-60%	Jun 10	Jun 12
6.1	Acceleration model analysis	2	1	-50%	Jun 10	Jun 11
6.2	Sequential guidance policies	3	1	-67%	Jun 11	Jun 12
7	Study of STSP Variants	33	40,5	23%	May 14	Oct 02
7.1	ADR STSP	20	18,5	-8%	Aug 25	Sep 12
7.1.1	Generate space debris cloud datasets	3	1	-67%	Aug 25	Aug 26
7.1.2	Statistical modelling of space debris clouds	5	3	-40%	Aug 26	Aug 29
7.1.3	ADR spacecraft modelling	2	0,5	-75%	Aug 29	Aug 29
7.1.4	Study of NCO performance for dynamic space VRPs	10	14	40%	Aug 29	Sep 12
7.2	OSSIE mission design	13	22	69%	May 14	Jun 05
7.2.1	Mass-dynamic STSP	5	10	100%	May 14	May 24
7.2.2	Mission statistical model	3	2	-33%	May 24	May 26
7.2.3	Mission viability analysis	5	10	100%	May 26	Jun 05
8	Writing	70	63	-10%	Jul 15	Oct 02
8.1	Paper IAC 2024	15	15	0%	Jul 15	Aug 21
8.2	Paper JGCD	25	31	24%	Sep 01	Oct 02
8.3	Thesis	30	17	-43%	Sep 15	Oct 02
	MILESONE: IAC 2024 PAPER DEADLINE				**Aug 21**	**Aug 21**
	MILESONE: GREEN LIGHT MEETING				**Oct 03**	**Oct 03**
	MILESONE: DEFENSE				**Oct 31**	**Oct 31**

Figure D.3: Gantt chart of the updated top level plan as of October 2024. Purple: items prioritized to achieve the IAC 2024 submission deadline. Yellow: items de-prioritized until after the IAC 2024 submission deadline. Vertical purple line, right: IAC 2024 submission deadline.

and the MSc thesis itself

The final project plan is summarized in Fig. D.3. Each outstanding development block was thoroughly analyzed and subdivided to the level of weekly objectives. The development time of low-level development blocks was then estimated. Development uncertainty was considerably lower by this point: this manifested in much more accurate development time estimates of lower-level blocks. After this analysis had been completed, items which were vital to achieve the IAC 2024 submission deadline were prioritized. Items which were not were de-prioritized and scheduled for after the IAC 2024 submission deadline.

Progress under the re-drawn project plan was swift and suffered no major contingencies. All original research goals and project objectives had been met by the end of the project.

D.3 Work Breakdown Structure

The following is a comprehensive work breakdown structure of the major development blocks of the project, listing for each: the realized development effort in hours, its dependencies, the expected output of the block, and all identified sub-items and sub-sub-items. More lower level items were identified and completed. Such items are out of the scope of this general overview.

1. **Impulsive Trajectory Design**
   - **Time required for completion**: 13.5 days
   - **Requirements to start the block**: Project plan approval
   - **Expected output**: Python module capable of estimating and generating realistic multi-revolution impulsive trajectories satisfying spacecraft constraints and considering relevant perturbations

1.1 **Multiple Hohmann Transfer**

1.1.1 **Transfer estimation methods**: Analytical transfer estimation methods [1 day]
1.1.2 **Transfer generation**: Implement transfer generation [3 days]
1.1.3 **Verification**: Test transfer methods [3 days]

1.2 **Impulsive Plane Change Manoeuvres**

1.2.1 **Transfer estimation methods**: Estimation techniques [1 day]
1.2.2 **Transfer generation**: Implement plane change [1 day]
1.2.3 **Verification**: Test plane changes [1 day]

2. **Low-Thrust Trajectory Design**
   - **Time required for completion**: 66 days
   - **Requirements to start the block**: Project plan approval
   - **Expected output**: Python module capable of estimating and generating multi-revolution low-thrust trajectories

2.1 **LTTO Approach Trade-Off**: Evaluate LTTO approaches and select an approach for the conceptual design of ADR missions [10 days]

2.2 **Q Law**

2.2.1 **Guidance law**: Implement Q law guidance [15 days]
2.2.2 **Trajectory generation**: Generate low-thrust trajectories [15 days]
2.2.3 **Verification**: Test Q law trajectories [15 days]
2.2.4 **Integrator selection**: Choose numerical integrators [2 days]
2.2.5 **Transfer cost estimation**: Estimate transfer costs [5 days]

2.3 **RQ Law**

2.3.1 **Guidance law**: Implement RQ law guidance [2 days]
2.3.2 **Transfer cost estimation**: Estimate transfer costs [2 days]

3. **Approximate Solutions**
   - **Time required for completion**: 4.5 days
   - **Requirements to start the block**: 1, 2
   - **Expected output**: Python module capable of generating high quality approximate solutions for space VRPs

3.1 **RAAN Walk**: Implement RAAN walk [0.5 days]

3.2 **Dynamic RAAN NN Search**: Implement dynamic RAAN nearest neighbor search [3 days]

3.3 **Dynamic Beam Search Based on Transfer Cost**: Implement beam search based on transfer cost [1 day]

4. **Heuristic Combinatorial Optimization**

   - **Time required for completion**: 29 days
   - **Requirements to start the block**: 1, 2, 3
   - **Expected output**: Python implementation of a modular heuristic combinatorial optimization solver for space VRPs using `pygmo`

   4.1 **Dynamic Environment Modelling**: Design and implement a modular solution to estimate tour cost propagating the global state as dictated by a secular perturbation model [2 days]

   4.2 **Population Sampling**

   4.2.1 **Uniform permutation sampling**: Research, implement and trade-off state-of-the-art uniform permutation sampling algorithms [6 days]

   4.2.2 **Distance-based permutation sampling**: Research, implement and trade-off state-of-the-art distance-based permutation sampling algorithms [10 days]

   4.3 **Population Encoding**: Research and select a permutation encoding scheme [1 day]

   4.4 **Optimizer Trade-Off**: Write an analysis tool to robustly measure the performance of all `pygmo` optimizers for a given space VRP [10 days]

5. **Neural Combinatorial Optimization**

   - **Time required for completion**: 32 days
   - **Requirements to start the block**: 1, 2, 3, 4
   - **Expected output**: Python implementation of a Neural Combinatorial Optimization solver for space VRPs

   5.1 **NCO Approach Trade-Off**: Evaluate NCO approaches from literature and choose a solver architecture and DL/NCO framework [10 days]

   5.2 **Conventional TSP**: Implement an NCO solver for the conventional TSP [2 days]

   5.3 **3D TSP**: Implement an NCO solver for the 3-dimensional TSP [1 day]

   5.4 **Static STSP**: Implement an NCO solver for the static STSP [5 days]

   5.5 **Hyperparameter Optimization**: Implement a state-of-the-art hyperparameter optimization method and apply it to tune the NCO solver for specific problem instances [5 days]

   5.6 **OSSIE Mission Design STSP**: Implement an NCO solver the mass-dynamic OSSIE OTV multi-rendezvous trajectory optimization problem [5 days]

   5.7 **ADR STSP**: Implement an NCO solver for the highly dynamics ADR STSP [5 days]

6. **Integration of SCP**

   - **Time required for completion**: 5 days
   - **Requirements to start the block**: 1, 2
   - **Expected output**: Integration of SCP methods into the project framework.

   6.1 **Acceleration Model Analysis**: Determine perturbations to account for [2 days]

   6.2 **Sequential Impulsive and Low-Thrust Policies**: **Sequential impulsive and low-thrust policies**: Implement sequential impulsive and low-thrust guidance policies [3 days]

7. **Study of STSP Variants**

- **Time required for completion**: 33 days
- **Requirements to start the block**: 1, 2, 3, 4, 5
- **Expected output**: Analysis of various STSP variants.

7.1 **ADR STSP**

7.1.1 **Generate space debris cloud datasets**: Obtain space debris cloud data from Celestrak and generate up-to-date datasets for the Iridium 33, Cosmos 2251, Fengyun 1C and Cosmos 1408 space debris clouds [3 days]

7.1.2 **Statistical modelling of space debris clouds**: Implement a statistical modelling procedure for space debris clouds and use it to create a realistic statistical model of the Iridium 33 debris cloud [5 days]

7.1.3 **ADR spacecraft modelling**: Create a realistic model of a spacecraft for ADR missions in LEO [2 days]

7.1.4 **Study NCO performance for dynamic space VRPs**: Solve the Iridum 33 ADR problem using NCO and assess the performance of NCO compared to that of heuristic CO methods [10 days]

7.2 **OSSIE Mission Design**

7.2.1 **Mass-dynamic STSP**: Implement an STSP variant accounting for mass changes due the capture or release of payloads [5 days]

7.2.2 **Mission statistical model**: Construct a statistical model to generate realistic mission scenarios for OSSIE, considering varying payload target states, payload masses, and payload bundling [3 days]

7.2.3 **Mission viability analysis**: perform a cost analysis of all possible mission scenarios for the OSSIE OTV derived from the nominal payload BOM of the vehicle, and define the mission design envelope of the OSSIE OTV [5]

8. **Writing**

- **Time required for completion**: 55 days
- **Requirements to start the block**: Concurrent
- **Expected output**: Final thesis and related publications.

8.1 **Paper IAC 2024**: Prepare and submit paper to IAC 2024 [15 days]

8.2 **Journal paper**: Prepare and submit paper to journal [20 days]

8.3 **Thesis**: Write the thesis document [20 days]