Neural combinatorial optimization for multi-rendezvous mission design

López Rivera, Antonio; Naeije, Marc

DOI
10.1016/j.asr.2025.03.050

Publication date
2025

Document Version
Final published version

Published in
Advances in Space Research

Citation (APA)
López Rivera, A., & Naeije, M. (2025). Neural combinatorial optimization for multi-rendezvous mission design. *Advances in Space Research*, *75*(10), 7306-7326. https://doi.org/10.1016/j.asr.2025.03.050

Available online at www.sciencedirect.com

ScienceDirect

Advances in Space Research 75 (2025) 7306–7326

ADVANCES IN SPACE RESEARCH
(a COSPAR publication)

www.elsevier.com/locate/asr

Neural combinatorial optimization for multi-rendezvous mission design

Antonio López Rivera[a,b,c], Marc Naeije[a,*]

[a] Delft University of Technology, Kluiverweg 1, Delft 2629 HS, Zuid Holland, the Netherlands
[b] Sener Aerospace & Defence, AOCS-GNC, Calle Severo Ochoa 4, Tres Cantos 28760, Madrid, Spain
[c] Present address: The Exploration Company, Behringstraße 6, Planegg 82152, Munich, Germany

Received 11 November 2024; received in revised form 9 March 2025; accepted 21 March 2025
Available online 26 March 2025

Abstract

Optimal solutions to spacecraft 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 effectiveness of Neural Combinatorial Optimization (NCO) methods for the autonomous planning of low-thrust, multi-target ADR missions, an instance of the Space Traveling Salesman Problem (STSP). An autoregressive, attention-based routing policy was trained to solve 10-transfer ADR routing problems using REINFORCE, Advantage Actor-Critic, and Proximal Policy Optimization. A hyperparameter sensitivity analysis identified embedding dimension and the number of encoder layers as the critical factors influencing model performance, while an ablation study found the attention-based encoder to be the most critical architectural component of the policy. The trained policy was evaluated on 10-, 30-, and 50-transfer scenarios based on the Iridium 33 debris cloud, comparing its performance to a baseline provided by a novel ADR STSP routing heuristic (Dynamic RAAN Walk, DRW) and near-optimal benchmarks obtained via Heuristic Combinatorial Optimization (HCO). In missions with 10 transfers, the NCO policy achieved a mean optimality gap of 32%, outperforming DRW. However, performance degraded significantly in scenarios with 30 and 50 transfers, suggesting limited generalization to larger problems. A hyperparameter search further revealed that the performance of the NCO model considered in this work improves asymptotically with its size. Exposure to greater numbers of training scenarios did not yield significant performance gains. This work demonstrates that NCO methods can be effective for the autonomous planning of ADR missions with a limited number of targets, but face scalability and generalization challenges in more complex scenarios.

Keywords: Active debris removal; Trajectory optimization; Reinforcement learning; Graph attention networks; Debris mitigation

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[1]) missions (Izzo et al., 2015; Ricciardi and Vasile, 2019; Federici et al., 2021; Medioni et al., 2023; Barea et al., 2020; Narayanaswamy et al., 2023) to tackle the space debris problem (Mark and Kamath, 2019; Bonnal et al., 2013), as well as On-Orbit Servicing (OOS) missions (Sellmaier et al., 2010; Jonchay et al., 2022; Federici et al., 2021) and advanced space logistics concepts (Sorenson and Pinkley, 2023).

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

[*] Corresponding author.
E-mail addresses: antonio.lopez@exploration.space (A. López Rivera), m.c.naeije@tudelft.nl (M. Naeije).

[1] See nomenclature at end of paper.

https://doi.org/10.1016/j.asr.2025.03.050
0273-1177/© 2025 The Author(s). Published by Elsevier B.V. on behalf of COSPAR.

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 transfer trajectory between each target in the optimal sequence, where optimality is defined by some metric. 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 $G := \{\mathscr{V}(t), \mathbf{W}(\pi)\}$, where $\mathscr{V}(t)$ is the set of graph vertexes (targets, the state of which drifts over time) and $\mathbf{W}(\pi) := \mathscr{V} \times \mathscr{V} \to \mathbb{R}^+$ is a map that associates an edge weight (a transfer cost) to each ordered vertex pair (Izzo et al., 2015), and may be 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 decomposition (Barea et al., 2020; Federici et al., 2021; Medioni et al., 2023; Narayanaswamy et al., 2023), 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 (Petropoulos et al., 2017; Lu et al., 2023), database-independent (analytical), or learning-based (Li et al., 2020).

State-of-the-art CO methods fall in two camps: exact methods and Heuristic Combinatorial Optimization (HCO) methods, which are less costly and can produce near-optimal results, but cannot offer optimality guarantees whatsoever (Izzo et al., 2015; François et al., 2019). Exact methods based on tree searches (Russel and Norvig, 2020; Cormen et al., 2009) are the norm for highly complex and large STSP variants; all winning submissions to the Global Trajectory Optimization Competitions have used tree search approaches (Izzo et al., 2015; Petropoulos et al., 2017; Hallmann et al., 2017). However, HCO methods are an attractive option to solve smaller STSP instances (up to hundreds of targets, see Izzo et al. (2015)) due to their capacity to achieve near-optimal results with lower computational cost (Izzo et al., 2015), and are widely applied in the literature to tackle multi-rendezvous mission design (Izzo et al., 2015; Narayanaswamy et al., 2023; Medioni et al., 2023; Federici et al., 2021; Ricciardi and Vasile, 2019). HCO methods have also been successfully applied to complex STSP instances where the cost of exact approaches is unfeasible (Petropoulos et al., 2017). High-quality approximate solutions are highly desirable for the HCO process. As the complexity of the generalized Vehicle Routing Problem (VRP) increases, high-quality approximate solutions become more difficult to obtain (Berto et al., 2024b). This bodes ill for the field of space logistics, as the complexity of space VRPs 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 (Locke et al., 2024).

Machine Learning (ML) approaches for spacecraft trajectory design have seen a surge of interest in recent years (Izzo et al., 2019a), with strong results achieved both for spacecraft guidance (Izzo et al., 2019b; Yang et al., 2024) and transfer cost estimation (Li et al., 2020). Neural Combinatorial Optimization (NCO) consists in the training and use of 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 eliminating the need for handcrafted heuristics, which often require significant domain-specific adjustments (Berto et al., 2024b). NCO has shown promising performance on various CO problems (Berto et al., 2024b; Berto et al., 2024c), including multi-agent VRPs (Berto et al., 2024a) and spacecraft sensor allocation problems (Jacquet et al., 2024), especially when coupled with advanced policy search procedures (François et al., 2019). It becomes pressing to ask whether learning-based methods from the field of CO could be applied in the spacecraft routing domain.

The present work aims to assess the applicability and effectiveness of NCO approaches for the design of multi-target rendezvous trajectories. To do so, an NCO policy is designed, trained, and refined to solve realistic ADR STSP scenarios based on the Iridium 33 space debris cloud. The performance of the NCO policy is then benchmarked against near-optimal solutions obtained using HCO, and compared to that of a highly performant hand-crafted STSP routing heuristic.

The paper is structured as follows. Section 2 discusses orbital mechanics, trajectory design, the mathematical formulation of the ADR STSP, and its solution using HCO. Section 3 contains a historical overview of the space debris problem and introduces a novel approach for the statistical modelling of space debris clouds, which is key for the generation of realistic ADR STSP scenarios for RL. Lastly, Section 4 discusses the design, training and refinement of an NCO policy capable of solving the Iridium 33 ADR STSP, and an analysis of its performance and generalization capabilities. Section 5 concludes the paper with a summary of key findings and recommendations for future research.

2. Background

This section will introduce the necessary background for the discussion and assessment of NCO methods for space VRPs. Orbital dynamics are discussed in Section 2.1, followed by Low-Thrust Trajectory Optimization in Section 2.3. The generalized formulation of the STSP, including the definition of the cost function, is given in Section 2.4. A novel routing heuristic for the dynamic STSP is introduced in Section 2.5, which will be used as a baseline against which to compare NCO policy performance. Lastly, Section 2.6 discusses an HCO solver used to obtain near-optimal solutions for the STSP.

2.1. Dynamics

The state of the spacecraft is propagated using the Modified Equinoctial Elements (MEE) described by Hintz (2008) including the retrograde factor I, which are nonsingular for all eccentricities and inclinations. The MEEs are related to the classical Keplerian elements (Hintz, 2008) by Eq. (1):

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

The Gauss Variational Equations (GVE) for MEE (Hintz, 2008) are used to model the time evolution of the spacecraft's state. The GVE follow in Eq. (2),

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

where s^2, v and w are defined in Eq. (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. 1b.

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

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).

$$\begin{aligned} &\hat{\mathbf{u}}_{\mathrm{ECI}} = \begin{bmatrix}\hat{\mathbf{e}}_r & \hat{\mathbf{e}}_\theta & \hat{\mathbf{e}}_\phi\end{bmatrix}\hat{\mathbf{u}}_{\mathrm{MEE}} \\ &\hat{\mathbf{e}}_r = \frac{\mathbf{r}}{\|\mathbf{r}\|}; \quad \hat{\mathbf{e}}_\phi = \frac{\mathbf{r}\times\mathbf{v}}{\|\mathbf{r}\times\mathbf{v}\|}; \quad \hat{\mathbf{e}}_\theta = \hat{\mathbf{e}}_\phi \times \hat{\mathbf{e}}_r \end{aligned} \tag{4}$$

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

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

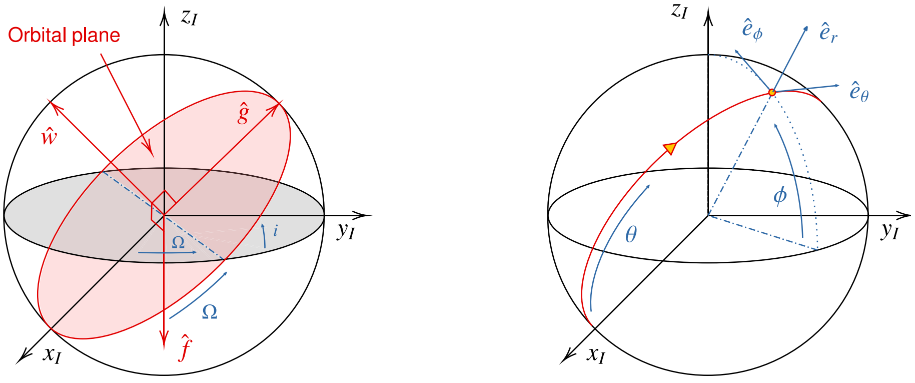
where $\hat{\mathbf{u}}$ is the direction of application of thrust. The spacecraft's mass is propagated assuming constant specific impulse (I_{sp}) through the burn. The change in mass over time is defined by Eq. (6):

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

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

2.2. Active debris removal spacecraft characteristics

An electrical propulsion ADR spacecraft concept is considered in this work; specifications follow in Table 1. The propulsion system is based on the specifications of existing Gridded Ion Thruster (GIT) designs for small spacecraft (O'Reilly et al., 2021; Conversano and Wirz, 2013). Spacecraft structural, fuel and payload mass are indicative of a spacecraft fit for this type of mission; payload mass is sized to 10–30 active de-orbiting payloads (Shan et al., 2016; Forshaw et al., 2016).



(a) MEE with respect to orbital plane.

(b) ECI and LVLH reference frames.

Fig. 1. Illustration of the MEE and the LVLH reference frame.

Table 1
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
1200kg	450kg	500kg	3e–4ms^{-2}	1e–5ms^{-2}	3000s	6,00kms^{-1}	0,36N	7.55kW

2.3. Low-thrust trajectory optimization

A Lyapunov Control (LC) guidance law is used in this work to generate low-thrust trajectories. LC methods are direct Low-Thrust Trajectory Optimization (LTTO) methods (Betts, 1998; Morante et al., 2021) that make use of predefined control laws (Morante et al., 2021; Falck et al., 2014), which are derived from Lyapunov functions. LC methods are notable for being both fast and able to generate reasonable estimates of optimal planetocentric trajectories (Morante et al., 2021).

2.3.1. Q-law

The Q-law, originally introduced by Petropoulos (Petropoulos, 2004), is a LC guidance law for low-thrust trajectory optimization based on the "proximity quotient" Q, a candidate Lyapunov function that approximates the best quadratic time-to-go (Petropoulos, 2004). MEE formulations of the Q-Law were later developed by Petropoulos (Petropoulos, 2005) and Varga (Varga and Perez, 2016). The proximity quotient Q is defined in Eq. (7):

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

where $\dot{\oe}_{xx}$ is the maximum rate of change of each orbital element (Petropoulos, 2004), the periapsis penalty P is defined in Eq. (8) and the element scaling factors $S_{\oe}$ are defined in Eq. (9).

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

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

The feedback control law is derived such that $\dot{\mathrm{Q}}$ is negative semi-definite, and follows in Eq. (10). This follows from applying the chain rule $\dot{\mathrm{Q}} = \nabla^\top \mathrm{Q}\dot{\oe}$ and observing that $\dot{\oe} = \Psi\mathbf{u}$, where Ψ stands for the state-space input matrix derived from the GVEs in MEE (Eq. (2)).

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

2.3.2. Rendezvous Q-law

The Rendezvous Q-law (RQ-law), proposed by Narayanaswamy and Damaren (2023), extends the Q-law to enable dynamic six-element targeting by means of a semi-major axis augmentation scheme (Eq. (11)). The scheme becomes active after reaching 5-element convergence, splitting the manoeuvre in two phases: orbit acquisition, in which the standard Q-Law is used to achieve 5-element convergence, and phasing, in which the semi-major axis augmentation scheme is activated to achieve the desired true longitude.

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

2.3.3. Perturbations

The most relevant perturbation for the design of the ADR trajectories considered in this work are gravity field distortions, in particular the Earth oblateness gravity field distortion parametrized by the J_2 zonal coefficient (Varga and Perez, 2016; Petropoulos et al., 2017). Drag is neglected as the Iridium 33 cloud is located at an altitude of approximately 800 km, but should be considered for ADR missions targeting clouds at lower altitudes (Wakker, 2015).

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 (Izzo et al., 2015). The secular impact of J_2 on the Right Ascension of the Ascending Node (RAAN) and Argument Of Perigee (AOP) of orbiting debris is described by Eq. (12) (Wakker, 2015), where $n = \sqrt{\mu/a^3}$ is the mean motion of the orbiting body.

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

To balance computational efficiency and accuracy, this study adopts a hybrid scheme that accounts for J_2 for longer timescales, and omits it for shorter timescales. Specifically, for each pair of consecutive debris targets, the transfer time is computed using an unperturbed (central-gravity) model. Upon arrival, the orbits of all debris objects are propagated under J_2 for the estimated transfer duration. This approach is justified by the small relative RAAN precession between consecutive debris over a transfer arc (order of days), meaning the unperturbed Q-Law transfer can be considered representative even in the presence of oblateness effects. Similar decoupled methods have previously been used to tackle large combinatorial trajectory design problems (Petropoulos et al., 2017; Hallmann et al., 2017).

Table 2
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 3
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

2.3.4. Simulation

The Tudat Space[2] astrodynamics library (Dirkx et al., 2022) 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}). Table 2 lists the values of the Q-Law and RQ-Law parameters used. Table 3 lists the element weights and convergence tolerances used.

Fig. 2 shows the average transfer in the RAAN walk (Izzo et al., 2015) 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 h, of which approximately 51 are spent in the orbit acquisition leg and 4 in the phasing leg. The brevity of the phasing leg is characteristic of RQ-Law transfers, and will be an important consideration for transfer time estimation, which will be discussed next. Fig. 3 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 a fictional target located at the center-of-RCS of the cloud plus 3 times the standard deviation of each element in the cloud.

2.3.5. 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 (Petropoulos et al., 2017; Hallmann et al., 2017). Transfer cost estimation approaches (both impulsive and low-thrust) may be either analytical (Hallmann et al., 2017; Hon, 2022; Medioni et al., 2023), 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 (Petropoulos et al., 2017), often relying on transfer window constraints, or based on multivariate regression. DL has been particularly successful for the latter (Li et al., 2020).

A linear model based on the best time-to-go $\mathscr{T}$ (Eq. (13)) is used to estimate the duration of RQ-Law transfers. The model is defined in Eq. (14). $\mathscr{T}$ follows from the definition of the proximity quotient Q, which approximates the best quadratic time-to-go (Petropoulos, 2004).

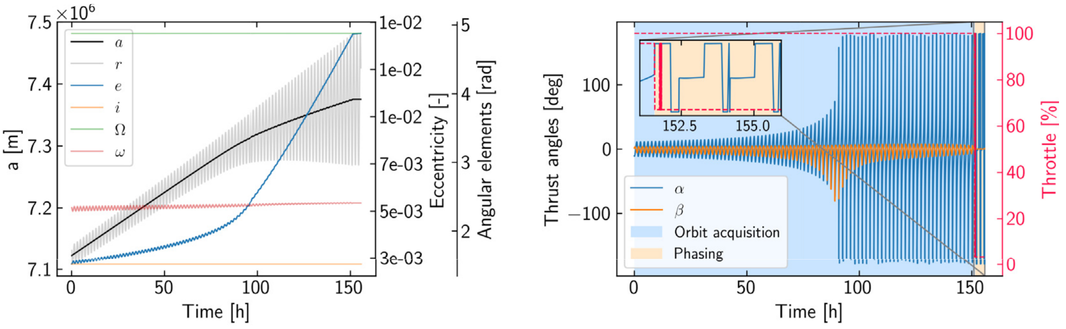
$$\mathscr{T} = \sqrt{\text{Q}} \tag{13}$$

The model in Eq. (14) is obtained by linear regression, aiming to predict measured TOFs as a function of the best time-to-go. The regression was performed considering all RAAN walk (Izzo et al., 2015) transfers through the Iridium 33, Cosmos 2251 and Fengyun 1-C debris clouds. Models for both Q-Law and RQ-Law transfers were constructed. 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 in Eq. (2.3.3). Fig. 4 shows predicted and observed RQ-Law TOFs and summarizes the performance of the linear model. Table 4 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 $\mathscr{T}$ and the measured TOF exists for both Q-Law and RQ-Law transfers (Pearson $r > 0.99$), indicating a strong linear relationship (Cohen, 1988). 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 (Draper and Smith, 1998); the mean estimation error ε is close to 0 as well. The Kolmogorov–Smirnov (KS) test (Guthrie and Heckert, 2016) was used to verify the similarity of the distributions of measured TOFs, and TOFs predicted by the models. In both cases the KS p-values indicate that there is no statistically significant difference between the estimated and observed TOF distributions.

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

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.

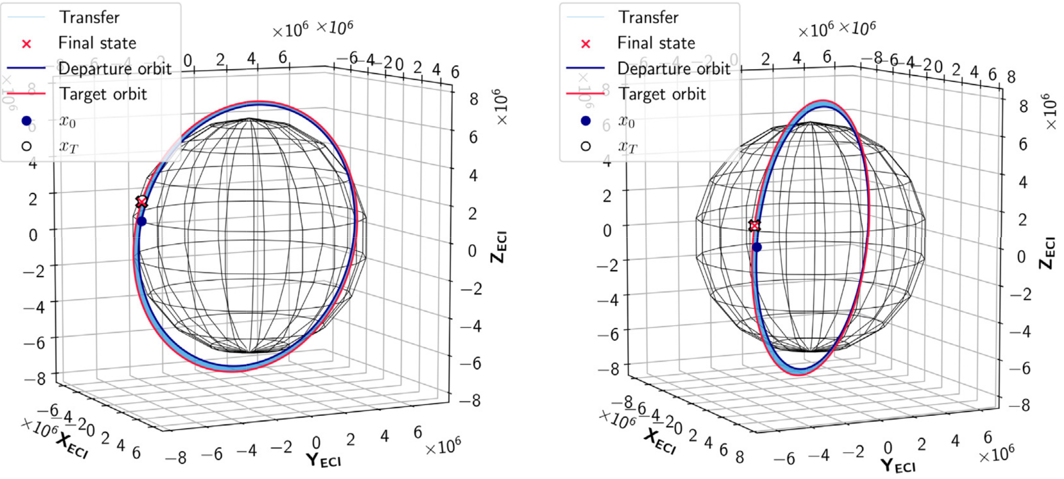
[2] https://docs.tudat.space/



(a)

(b)

Fig. 2. Average transfer in the Iridium 33 debris cloud. (a) Keplerian element history through the transfer. (b) Control history; α and β are the in-plane and out-of-plane (ecliptic plane) thrust angles with respect to $\hat{\mathbf{e}}_\theta$ (Eq. (4); notation from Narayanaswamy and Damaren (2023)).



(a)

(b)

Fig. 3. 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 each element of the cloud. (a) Iridium 33. (b) Fengyun 1C.

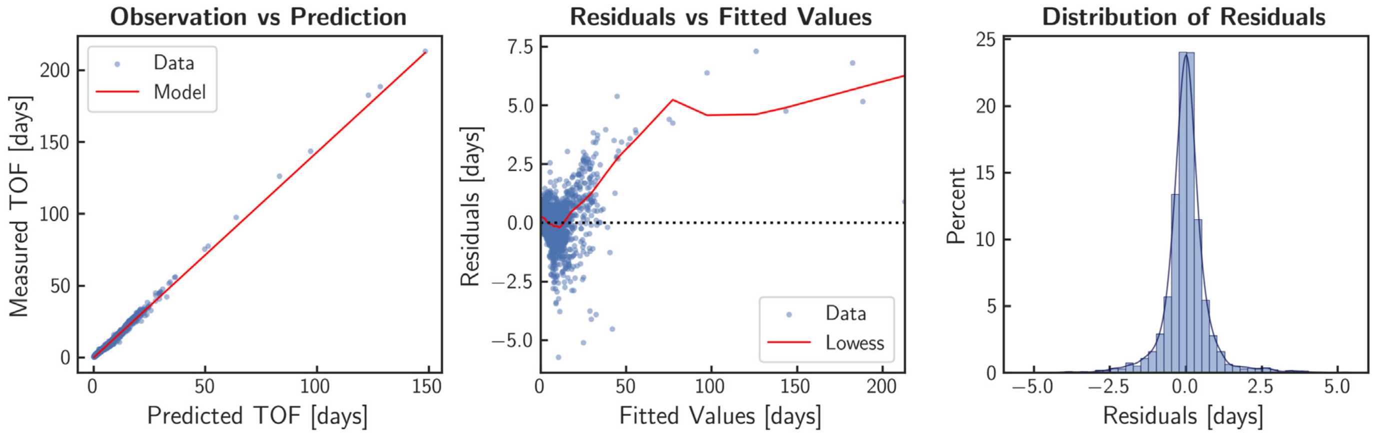


Fig. 4. Estimation of RQ-Law TOFs using the Q proximity quotient. Histogram bin width (right side): 6 h.

Table 4
Goodness-of-fit analysis of the TOF models. ε: estimation error.

	Pearson r	Slope	Intercept	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

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

2.4. Problem formulation and cost function

The STSP seeks an optimal sequence π of target visits over a set of targets E, minimizing the cumulative ΔV cost function, C_π:

$$\begin{aligned} \min_{\pi} \quad & C_\pi = \sum_{k=0}^{n-1} c\left(E_k^{\pi(k)}; E_k^{\pi(k+1)}; \Theta\right) \\ \text{s.t.} \quad & \pi(0) = \begin{cases} h & \text{if provided} \\ \text{free} & \text{otherwise} \end{cases} \\ & \pi(n) = \begin{cases} h & \text{if Hamiltonian cycle} \\ d & \text{if decommissioning} \\ \text{free} & \text{otherwise} \end{cases} \\ & \{\pi(k)\}_{k=1}^{n-1} = T \setminus \{h\} \\ & \pi(k) \in T \cup \{d\}, \quad k = 0, 1, \ldots, n \\ & E_{k+1} = \Phi(E_k, \Delta t_k), \quad k = 0, 1, \ldots, n-1 \\ & \Delta t_k = \text{TOF}\left(E_k^{\pi(k)}; E_k^{\pi(k+1)}; \Theta\right), \quad k = 0, 1, \ldots, n-1 \end{aligned} \tag{16}$$

where $c\left(E_k^{\pi(k)}; E_k^{\pi(k+1)}; \Theta\right)$ represents the ΔV required for a transfer from the departure state $E_k^{\pi(k)}$ to the target state $E_k^{\pi(k+1)}$, given spacecraft parameters Θ, which include performance specifications and guidance policy parameters.

The environment state E_k at time t_k comprises the state of all bodies. After each transfer k the environment is propagated subject to the dynamics f over the transfer time Δt_k, resulting in $E_{k+1} = \Phi(E_k; \Delta t_k)$; here, Φ denotes the state transition function derived from integrating f. The transfer time Δt_k is determined by the TOF function $\text{TOF}\left(E_k^{\pi(k)}; E_k^{\pi(k+1)}; \Theta\right)$, dependent on the departure and target states at time t_k and the spacecraft parameters Θ.

In this study $c\left(E_k^{\pi(k)}; E_k^{\pi(k+1)}; \Theta\right)$ represents the ΔV required for an RQ-Law transfer from $E_k^{\pi(k)}$ to $E_k^{\pi(k+1)}$, defined in Eq. (15), and $\text{TOF}\left(E_k^{\pi(k)}; E_k^{\pi(k+1)}; \Theta\right)$ represents the TOF estimator defined in Eq. (14). f represents the secular J_2 perturbation defined in Eq. (12). The initial state of the spacecraft $E_0^{\pi(0)}$ is fixed to the centroid (weighed by RCS) of the Iridium 33 cloud. The final state $E_n^{\pi(n)}$ is constrained to a circular decommissioning orbit at 250 km altitude (all other orbital parameters free).

2.5. Policy performance baseline: the Dynamic RAAN walk spacecraft routing heuristic

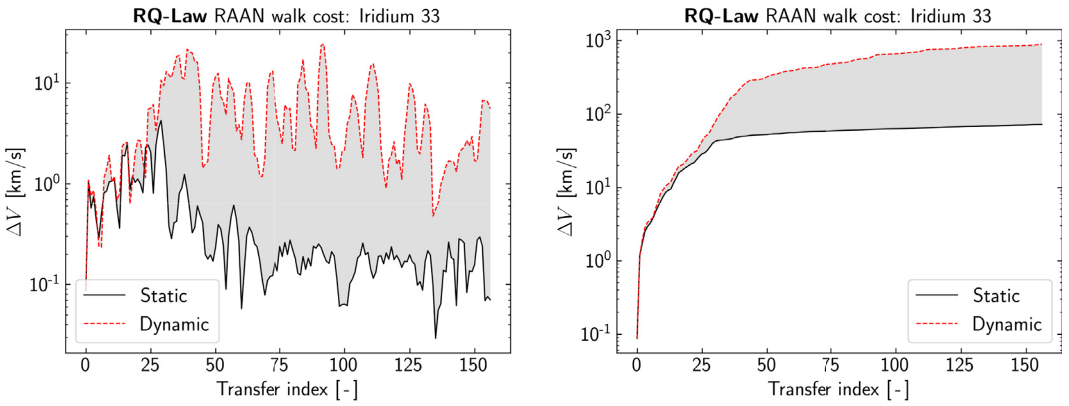
To contextualize NCO policy performance for the dynamic STSP in LEO ADR missions, we introduce the Dynamic RAAN Walk as a high-performing baseline that clarifies the relative gains of learned policies. Izzo et al. (2015) showed that the optimal solution of the static STSP closely resembles a monotonically increasing RAAN walk. This result is intuitive as transfer cost is primarily driven by plane change cost, and plane change cost (Eq. 17) is primarily driven by the RAAN gap that must be closed (Izzo et al., 2015; Medioni et al., 2023) for orbits with relatively high inclination: this includes the orbits of most Earth-orbiting spacecraft (Boley and Byers, 2021) and all the debris clouds under consideration (Fig. 7).

$$\Delta V_\gamma = 2V_0 \sin\left(\frac{\gamma}{2}\right) \tag{17a}$$

$$\gamma = \arccos\left(\cos i_1 \cos i_2 + \sin i_1 \sin i_2 [\cos \Omega_1 \cos \Omega_2 + \sin \Omega_1 \sin \Omega_2]\right) \tag{17b}$$

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 (Izzo et al., 2015). Fig. 5 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 (Izzo et al., 2015), as RAAN drift rates are independent of the original ranking (Eq. (12)).

A novel spacecraft routing heuristic is proposed to obtain high-quality approximate solutions for the dynamic STSP, which will be referred to as Dynamic RAAN Walk (DRW). The DRW is defined as a greedy search over a nearest-RAAN target ranking policy, which simplifies to the classic RAAN walk in the static case. Fig. 6 shows the performance of the DRW heuristic for the full Iridium 33 ADR STSP against two exact tree search approaches: Beam Search (BS, refer to Freitag and Al-Onaizan (2017)) and Nearest-Neighbor (NN) search (Lowerre,



(a) Cost per transfer.

(b) Cumulative cost.

Fig. 5. Impact of RAAN drift on the cost of the RAAN walk through the Iridium 33 debris cloud.

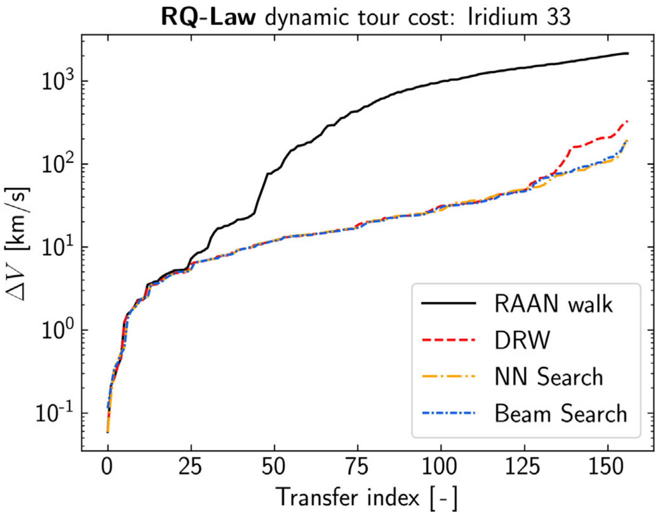


Fig. 6. Cumulative cost of tours traversing the Iridium 33 debris cloud as a function of transfer index, under RAAN drift. BS beam width: 20.

Table 5
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

1976). The computational cost of each method is reported in Table 5. As can be seen in Fig. 6, the DRW yields comparable performance to that of the searches at a fraction of the computational cost.

2.6. Policy performance benchmark: near-optimal routing with heuristic combinatorial optimization

To benchmark NCO policies for the STSP, we employ a modular solver based on population-based HCO, which yields near-optimal solutions for comparative assessment. The choice for HCO is motivated by the widespread use of HCO methods to solve STSPs in literature (Izzo et al., 2015; Narayanaswamy et al., 2023; Medioni et al., 2023; Federici et al., 2021; Ricciardi and Vasile, 2019) and the availability of highly performant, open-source heuristic global optimization libraries such as `pygmo`[3] (Biscani and Izzo, 2020) and `pymoo`[4] (Blank and Deb, 2020), which greatly eases the benchmarking and selection of diverse HCO 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 (Biscani and Izzo, 2020; Coello et al., 2007).

An archipelago comprising 16 islands, each populated by 80 individuals (decision vectors, representing permutations of length n, where n is the number of targets to visit) is used to perform the optimization. Initial populations are sampled using a combination of uniform (Mitchell et al., 2022; Eberl, 2016; Knuth, 1997) and distance-based permutation sampling using the Mallows model (Mallows, 1957; Diaconis, 1988) under the Hamming distance (Waggener and Waggener, 1995; Irurozki, 2014): the latter is done to leverage approximate solutions obtained using the DRW heuristic. Decision vectors are encoded into $\mathbb{R}^n$ using random-keys encoding (Bean, 1994), and the `pygmo` Simple Genetic Algorithm is used to evolve the populations over 5 evolutionary periods of 500 generations. Empirical results show this to be a reliable approach to obtain near-optimal solutions for the 10-, 30- and 50-transfer ADR mission scenarios considered in this work.

3. Statistical modelling of the active debris removal environment

Training NCO policies with RL requires a large amount of realistic and diverse STSP scenarios. This section intro-

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

[4] https://pymoo.org/

duces a novel procedure to create statistical models of space debris clouds. A brief historical overview of the space debris crisis is presented in Section 3.1 along with the case study considered in this work: an ADR mission targeting the Iridium 33 debris cloud. Section 3.2 then discusses the implementation of the STSP environment for NCO.

3.1. 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 (Klinkrad, 2006), 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) (ESOC, 2024). Awareness of the problem has grown rapidly in recent years (Hall, 2014; Locke et al., 2024). The removal of large uncompliant objects from orbit has been the primary focus of ADR mission design up to the present day (Bonnal et al., 2013, 2004, 2021, 2014, 2020, 2017, 2021, 2016).

The 2024 NASA OTPS Phase 2 report (Locke et al., 2024) 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 (Kelso, 2009; NRC, 2011). Iridium 33 is one of the most widely studied clouds in LEO, other notable clouds being the Cosmos 2242 (Kelso, 2009), Fengyun 1C (Johnson et al., 2008) and Cosmos 1408 (Pardini and Anselmo, 2023) clouds. The Gabbard diagram (Johnson et al., 1984) of the four clouds can be seen in Fig. 7, including Radar Cross-Section (RCS) and expected decay time data. 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[5], and decay time data is obtained from the ESA Database and Information System Characterising Objects in Space (DISCOS) database[6].

Fig. 8 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—. As the cost of redezvous is primarily driven by plane change cost, and this by RAAN change for high inclination orbits (Izzo et al., 2015), ADR missions are bound to require extreme amounts of ΔV.

3.2. Environment

The state of the targets is described using Keplerian elements (Izzo et al., 2015). The NCO policy is trained in realistic ADR scenarios based on the Iridium 33 debris cloud. These scenarios are generated using a statistical model of the real debris cloud. The model is obtained by fitting the observed state of the cloud (each Keplerian element, possibly other parameters) using the parametric statistical models that best match each observation. All 19 parametric statistical models available in PyTorch[7] (Paszke et al., 2019) are considered. Goodness of fit is assessed using the Kolmogorov–Smirnov (KS) test (Guthrie and Heckert, 2016). 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. 9 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 (von Neumann, 1951) when the Inverse Cumulative Distribution Function (ICDF) of the parametric model is defined and implemented in PyTorch, and with vectorized rejection sampling (Hastings, 1970) if the ICDF is not available.

The environment state E is propagated through the sequential decision-making process as per Eq. (16).

4. Neural combinatorial optimization for spacecraft routing

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 (François et al., 2019). NCO offers the attractive prospect of alleviating the scaling issues of exact approaches, while removing the need for hand-crafted heuristics (Berto et al., 2024b), and has shown promising performance on various CO problems (Berto et al., 2024b), and has been shown to achieve high-quality results, especially when coupled with advanced policy search procedures (François et al., 2019).

The `RL4CO`[8] NCO library is used to implement and train the STSP routing policy. `RL4CO` is a benchmark library for NCO based on PyTorch (Paszke et al., 2019) with standardized, modular, and highly performant imple-

[5] https://celestrak.org
[6] https://discosweb.esoc.esa.int
[7] https://pytorch.org/docs/stable/distributions.html
[8] https://rl4.co

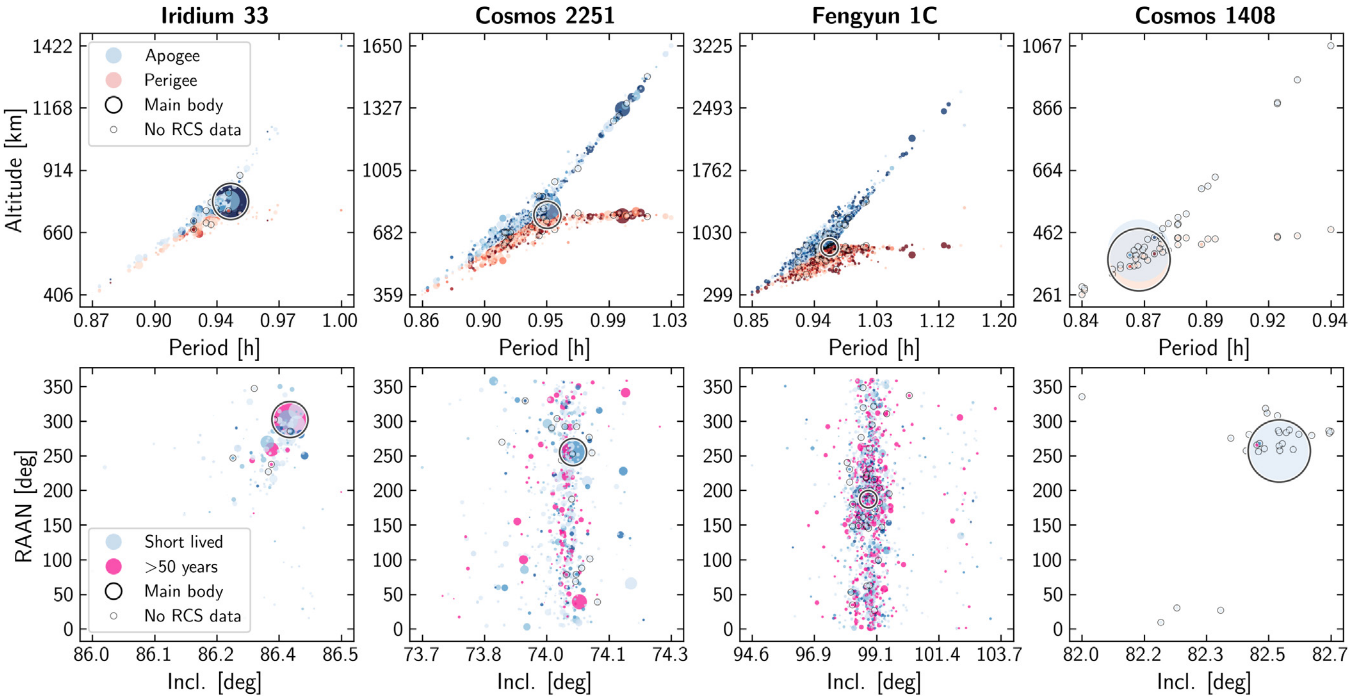


Fig. 7. Gabbard diagram of the Iridium 33, Cosmos 2251, Fengyun 1C and Cosmos 1408 debris clouds as of October 2024. Point size proportional to RCS. Color intensity in top row proportional to lifetime before natural decay. Data obtained from Celestrak. Own work.

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 October 2024, and estimates for the year 2050 and 2100. Estimates are obtained from the ESA DISCOS.

	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

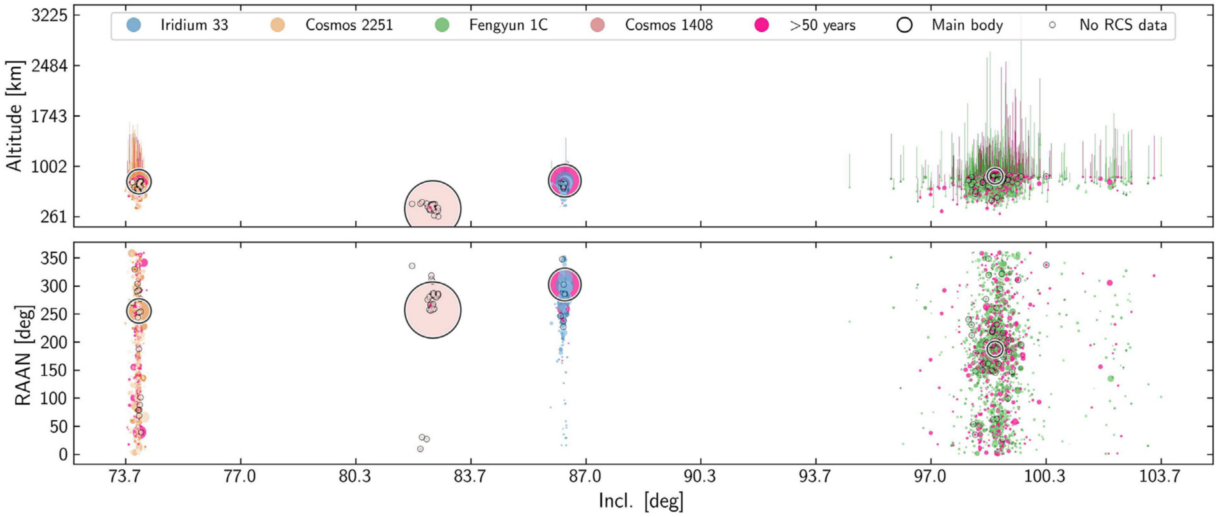


Fig. 8. Joint altitude-RAAN-inclination diagram of the Iridium 33, Cosmos 2251, Fengyun 1C and Cosmos 1408 debris clouds as of October 2024. Point size proportional to RCS. Color intensity proportional to lifetime before natural decay. Color trails, top: perigee to apogee altitude. Data obtained from Celestrak. Own work.

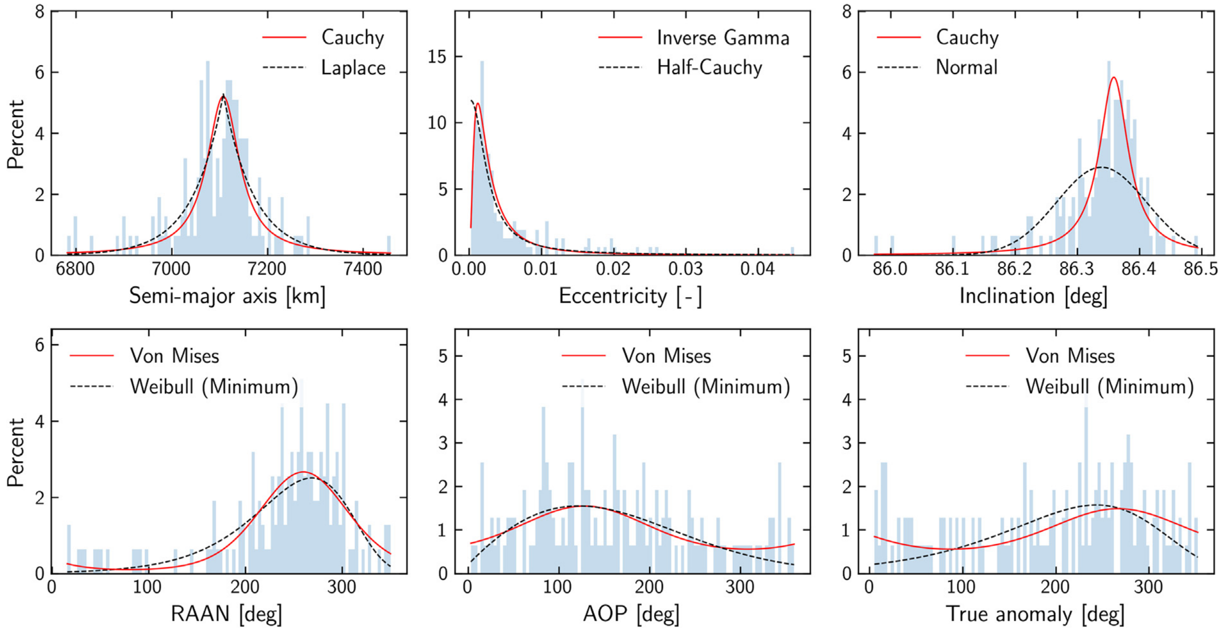


Fig. 9. Statistical model of the translational state of the Iridium 33 debris cloud. Histograms: observations. Curve, red: parametric model that best matches the observations, where goodness of fit is measured using the KS statistic. Curve, black: second-best parametric model.

mentations of various environments, policies and RL algorithms, covering the entire NCO pipeline (Berto et al., 2024b). This section is structured as follows. Section 4.1 introduces the NCO routing policy. Policy search procedures are introduced in Section 4.2. Section 4.3 discusses policy training with RL and presents a trade-off between three RL algorithms. RL algorithm selection was followed by an ANOVA to determine the sensitivity of policy performance to 13 key hyperparameters; the analysis and its results are discussed in Section 4.4. An ablation study to determine the most critical architectural component of the NCO policy follows in Section 4.5. Section 4.6 discusses hyperparameter optimization, followed by an analysis of the impact of training dataset size on policy performance in Section 4.7. Lastly, Section 4.8 presents an analysis of the performance and generalization capabilities of the final NCO policy.

4.1. Policy

An autoregressive attention-based policy[9] is used to solve the routing problem. Introduced by Kool et al. (2019), the policy encodes the input graph using a feedforward layer combined with a Graph Attention Network (GAT) and decodes the solution using a Pointer Network (PM) based on Vinyals et al. (2017). A visual representation of this architecture can be seen in Fig. 10. Kool et al. (2019) trained this policy using the REINFORCE RL algorithm, achieving considerably better performance than other architectures. François et al. (2019) found this policy to be a highly efficient learning component in their comprehensive analysis of learning performance in NCO methods.

4.2. Policy search

Provided with the state of the system, the policy generates a probability distribution over all remaining targets, which after training should represent the likelihood that picking any given target next is the optimal decision to make. The policy search or decoding strategy determines how actions are taken based on the probability distribution generated by the learned policy. Advanced policy search strategies have been found to greatly improve the performance of NCO algorithms when increasing model size yields diminishing returns (François et al., 2019). Three standard policy search strategies (François et al., 2019) are considered in this work: greedy search, stochastic search and Beam Search (BS).

4.3. 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 a trade-off to determine the best RL algorithm to train STSP routing policies.

4.3.1. Stochastic policy gradient

The REINFORCE algorithm, introduced by Williams (1992), 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

[9] https://rl4.co/docs/content/api/zoo/constructive_ar/#models.zoo.am.policy.AttentionModelPolicy

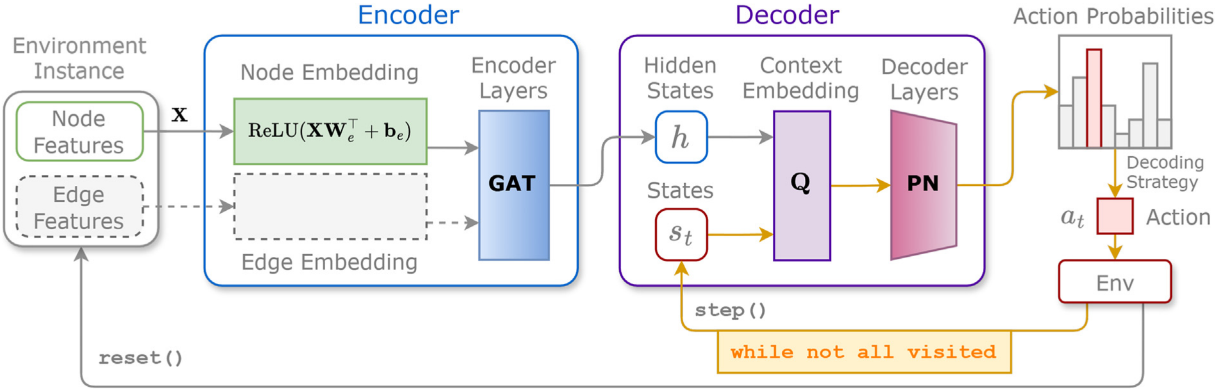


Fig. 10. Architecture of the autoregressive, attention-based policy used in this work. The encoder comprises a fully connected network and a GAT with a feedforward layer. Edge embeddings are omitted as the STSP graph is fully connected. The decoder constructs at each step a context embedding **Q** used as the query for the PN attention mechanism. This diagram is based on the general `RL4CO` policy architecture diagram by Berto et al. (2024b). Refer to Kool et al. (2019) and Vinyals et al. (2017) for more information about the internal structure of. the GAT and PN.

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.

4.3.2. Advantage actor-critic

Advantage Actor-Critic (A2C), introduced by Konda and Tsitsiklis (1999) and extended by Mnih et al. (2016), 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.

4.3.3. Proximal policy optimization for autoregressive policies

Proximal Policy Optimization (PPO), proposed by Schulman et al. (2017), 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. (2019), 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 (Kool et al., 2019).

4.3.4. Algorithm selection

The three RL algorithms under consideration were traded-off on a 10-transfer ADR STSP scenario based on the Iridium 33 debris cloud. Training was repeated 5 times using different random seeds to ensure the trade-off was robust. To benchmark policy performance all ADR STSP instances in the validation dataset were 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 7 lists the most relevant training settings and policy architecture hyperparameters. With the exception of the embedding size of 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, 250 GB 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 8 and illustrated in Fig. 11a and Fig. 11b, indicate that A2C consistently outperforms REINFORCE and the modified PPO across multiple metrics. As expected BS is the best performing policy search strategy (Kool et al., 2019; François et al., 2019). A2C achieves the lowest mean ΔV and optimality gap, with values of 111 km/s and 38% respectively when using BS to search the trained policy. In contrast, REINFORCE and PPO exhibit higher mean ΔV values of 173 km/s and 155 km/s, and optimality gaps of 115% and 92% respectively.

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

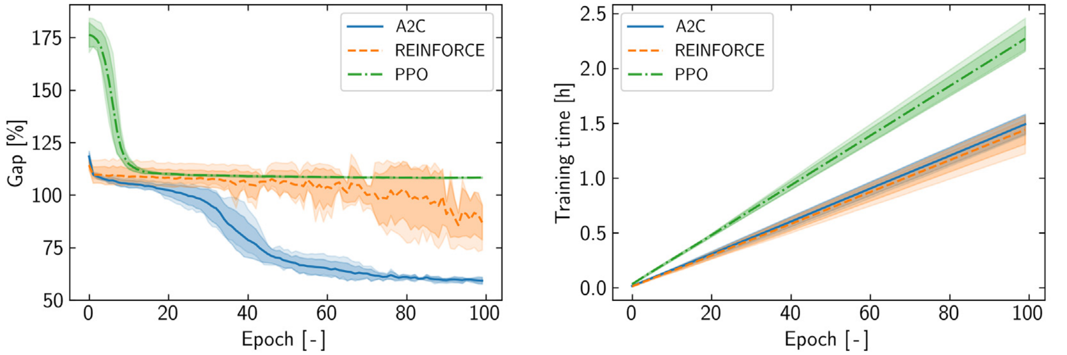
These observations suggest that A2C not only produces better-performing policies but also does so with training efficiency similar to REINFORCE and superior to PPO. The lower standard deviations in ΔV and optimality gap for A2C indicate more stable and reliable policy learning.

Table 7
Training and policy architecture settings used to compare RL algorithm performance.

Training dataset	**Batch size**	**Optimizer**	**Learning rate**	**Epochs**	**Embedding size**	**Policy search**
1 M scenarios	32768	Adam	1,00E-04	100	256	Stoch.

Table 8
RL training performance using REINFORCE, A2C and PPO. Policy performance was measured over 1000 unseen STSP scenarios based on the Iridium 33 cloud. Training time only reported for stochastic policy search. Values given as: $\mu_{\pm\sigma}$. Bold: best result. Multiple results highlighted if best cannot be decided based on the observed mean and variance.

	REINFORCE			**A2C**			**PPO**		
Search strategy	Greedy	Stoch.	BS	Greedy	Stoch.	BS	Greedy	Stoch.	BS
ΔV [km/s]	206,4$_{\pm 39,7}$	212,7$_{\pm 39,7}$	172,6$_{\pm 29,6}$	129,6$_{\pm 28,1}$	130,1$_{\pm 28,4}$	**111,0$_{\pm 19,7}$**	169,4$_{\pm 36,6}$	178,4$_{\pm 36,4}$	154,6$_{\pm 30,0}$
Optimality gap [%]	156,7$_{\pm 56,0}$	164,6$_{\pm 57,2}$	114,7$_{\pm 43,1}$	61,3$_{\pm 39,1}$	61,9$_{\pm 39,2}$	**38,3$_{\pm 29,0}$**	110,6$_{\pm 50,3}$	121,7$_{\pm 50,1}$	92,1$_{\pm 41,6}$
Training time [h]	—	**1,44$_{\pm 0,13}$**	—	—	**1,49$_{\pm 0,09}$**	—	—	2,27$_{\pm 0,11}$	—
Inference time [ms]	**36,1$_{\pm 4.5}$**	43,7$_{\pm 11.1}$	86,5$_{\pm 6.5}$	**37,9$_{\pm 7.4}$**	43,8$_{\pm 11.7}$	87,1$_{\pm 6.6}$	**35,3$_{\pm 4.8}$**	43,5$_{\pm 12.9}$	86,6$_{\pm 11.0}$



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

(b) Training time for each algorithm. Shaded areas: 1 standard deviation range, min-max range.

Fig. 11. RL training performance summary.

Despite PPO's effectiveness in various reinforcement learning tasks (Schulman et al., 2017), 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. Additionally, the loss of detailed gradient information due to the simplified MDP formulation could hinder effective policy updates.

Based on these results, A2C was 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.

4.4. Hyperparameter impact determination with ANOVA

Optimizing the performance of NCO models involves tuning a multitude of hyperparameters. Identifying the hyperparameters with the greatest impact on model performance is critical to prioritize them for further optimization (Hastie et al., 2009). Analysis of Variance (ANOVA) serves as a robust statistical procedure to assess the significance of multiple factors on model performance simultaneously (Kutner, 2005).

Orthogonal arrays, particularly Taguchi factorial designs, facilitate efficient experimentation by systematically varying hyperparameters across predefined levels while minimizing the number of required experimental runs (Taguchi and Konishi, 1987; Taguchi, 1993; Maghsoodloo et al., 2004). The Taguchi L27 orthogonal array, also known as L_{27}-$A3^{13-10}$ fractional factorial design (Guthrie and Heckert, 2016), is specifically designed to measure the linear effects of up to 13 factors at three levels, making it suitable for comprehensive hyperparameter analysis with limited resources (Guthrie and Heckert, 2016). Table 9 presents the Taguchi L27 orthogonal array utilized in this study.

An ANOVA was conducted using the `statsmodels` library (Seabold and Perktold, 2010) to evaluate the linear (or main) effects of 13 hyperparameters on the model's per-

Table 9
Taguchi L27 orthogonal array, also known as L_{27}-$A3^{13-10}$ fractional factorial design (Guthrie and Heckert, 2016).

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

formance. The 13 hyperparameters considered, their function, and the rationale for considering each one can be seen in Table 10.

Table 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 (Kutner, 2005; Guthrie and Heckert, 2016). Residual normality was verified by the Shapiro–Wilk test (Shapiro and Wilk, 1965; Guthrie and Heckert, 2016) ($p = 0.57$) and the Anderson–Darling test (Guthrie and Heckert, 2016) ($p = 0.27$). Homoscedasticity was verified by visual inspection of the residuals. Having verified both assumptions, the ANOVA was considered valid to diagnose the main effects of the 13 hyperparameters under consideration.

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 above the confidence threshod of 0.05 (0,0058 and 0,0487 respectively). Embedding dimension has both the lowest p-value and highest R^2 value (25.20%), indicating it is the most critical hyperparameter for policy performance. The number of encoder layers follows with an R^2 of 10.96%, indicating a meaningful but slightly lesser influence. All other hyperparameters do not show statistically 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 learning higher dimensional representations of target states 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. In terms of policy architecture, the 5 most critical hyperparameters for policy performance are encoder parameters, with embedding dimension impacting both the encoder and decoder. These effects suggest that the GAT encoder is the most critical architectural component of the network for policy performance. The goal of the following two sections is to validate this claim, and to optimize the two critical hyperparameters identified in this analysis: embedding dimension, and the number of layers of the GAT encoder.

4.5. Ablation study

In order to quantitatively assess the impact of the GAT encoder on overall model performance, an ablation study was conducted comparing four encoder variants. Specifically, the GAT encoder was first removed and then replaced with alternative architectures that maintain equal depth and a comparable number of trainable parameters, thereby isolating the effect of the encoder design on the NCO policy. The study did not consider ablations to the PN decoder, as the pointer mechanism is fundamental to generating combinatorial solutions (Vinyals et al., 2015) and empirical analysis indicates that policy performance is minimally sensitive to variations in decoder parameters (Table 11).

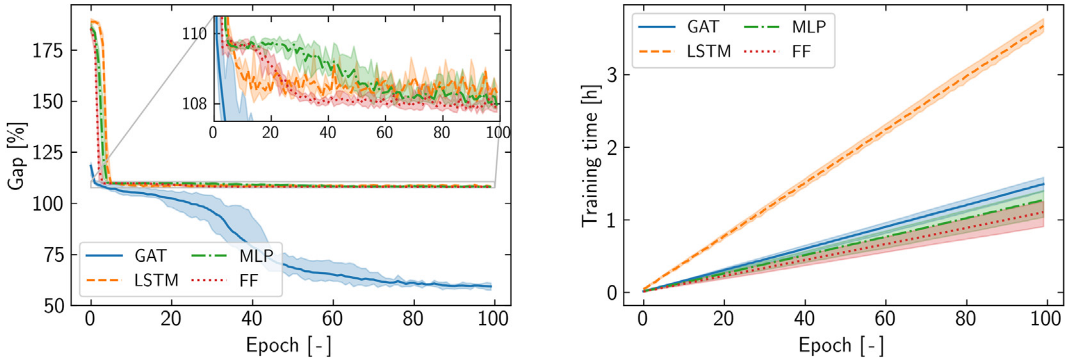
Three alternative encoders were considered in addition to the baseline three-layer GAT encoder: (i) a single-layer Feed-Forward (FF) Network, reducing the number of encoder parameters from 1.58 M to 66 k (effectively eliminating the encoder component); (ii) a three-layer Multi-Layer Perceptron (MLP) with the hidden dimension chosen such as to equate the total number of parameters of the MLP to that of the baseline GAT encoder; and (iii) a three-layer LSTM encoder akin to that used in the original PN implementation (Vinyals et al., 2017), with the hidden dimension chosen such as to equate the combined parameter count of the multi-layer LSTM (plus the projection

Table 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 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	Architectural impact
Embedding Dimension	3.46E + 09	1	10.9	**0.005776**	25.20%	**Encoder**, decoder
Number of Encoder Layers	1.50E + 09	1	4.73	**0.048680**	10.96%	**Encoder**
Feedforward Hidden Size	7.82E + 08	1	2.46	0.140893	5.70%	**Encoder**
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%	**Encoder**
Actor Learning Rate	6.66E + 08	1	2.10	0.171374	4.86%	—
Critic Hidden Dimension	6.18E + 08	1	1.94	0.186560	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%	**Encoder**
Tanh Clipping	2.11E + 08	1	0.66	0.429799	1.54%	Decoder
Critic Learning Rate	3.58E + 07	1	0.11	0.742551	0.26%	—
Temperature	1.58E + 06	1	0.00	0.944904	0.01%	Decoder
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%**	



(a) Learning curves of each policy, represented by the vaildation reward. Shaded areas: min-max range.

(b) Training time of each policy. Shaded areas: min-max range.

Fig. 12. Training performance summary for the four policies considered in the ablation study.

Table 12
Ablation study summary. Policy performance was measured over 1000 unseen STSP scenarios based on the Iridium 33 cloud. Values given as: $\mu_{\pm\sigma}$. Bold: best result. Multiple results highlighted if best cannot be decided based on the observed mean and variance.

Ablation	Depth [-]	Trainable parameters [-]	Optimality gap [%]	Training time [h]	Inference time [ms]
GAT (Baseline)	3	1.582 M	$\mathbf{52.72_{\pm 35.69}}$	$\mathbf{1.35_{\pm 0.14}}$	$\mathbf{45.5_{\pm 4.6}}$
FF	1	**66.8 k**	$87.46_{\pm 14.09}$	$\mathbf{1.10_{\pm 0.13}}$	$\mathbf{44.4_{\pm 16}}$
MLP	3	1.583 M	$87.34_{\pm 40.66}$	$\mathbf{1.27_{\pm 0.15}}$	$\mathbf{47.3_{\pm 9.0}}$
LSTM	3	1.589 M	$87.21_{\pm 40.64}$	$3.74_{\pm 0.13}$	$63.2_{\pm 8.6}$

layer) to that of the baseline GAT encoder. Training was conducted for 100 epochs on NVIDIA L40 GPU systems (16 vCPUs, 250 GB RAM) using the settings reported in Table 7. All policy and training parameters remained unchanged in the four experiments with the exception of the encoder.

Fig. 12 displays the learning curves of the four policies, and Table 12 reports the key characteristics and performance for the four policies. The optimality gaps reported were obtained using the BS policy search strategy. The baseline GAT encoder achieved a mean optimality gap of

Table 13
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	4,8316240	0,115377	3,28%
ED^2	4,72E + 08	1.0	29,455497	**0,012275**	19,97%
NL^2	3,77E + 07	1.0	2,3521440	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%

Table 14
Optimality gaps obtained for each run in the grid search. BS was used 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%

Table 15
Number of trainable parameters for each configuration considered in the grid search, as multiples of parameters of the 2-layer, 128 ED model.

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

52.72% with a mean training time of 1.35 h and a mean inference time of 45.5 ms. In contrast, the FF, MLP, and LSTM replacements yielded significantly higher optimality gaps (ranging from approximately 87.21% to 87.46%), with the LSTM variant incurring a markedly longer mean training time of 3.74 h. Given that the $\mu_{\pm\sigma}$ intervals do not overlap between the baseline and its alternatives, the data robustly indicate that the GAT encoder is the critical architectural element driving performance.

4.6. Hyperparameter optimization

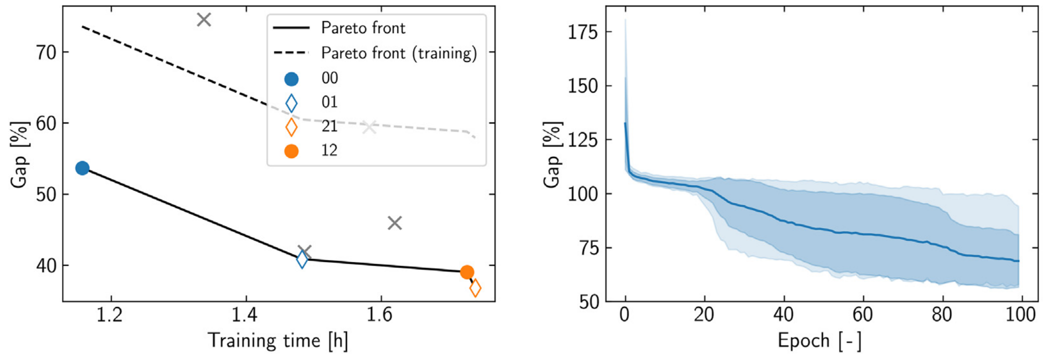
A full factorial (grid) search was conducted to optimize the embedding dimension and number of encoder layers, considering three levels. The levels considered for each hyperparameter are those previously presented in Table 10. Table 14 shows the optimality gaps obtained by each of the models in the grid search, using BS to decode the policies. No statistically significant improvement is observed in comparison with the RL algorithm trade-off results in Table 8. Fig. 13b shows the combined learning curve all grid search runs. Fig. 13a shows the Pareto front of the grid search. An interesting feature in Table 14 is the underperformance of models in the diagonal. The amount of trainable parameters of the models considered can be seen in Table 15.

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 13. The Shapiro–Wilk test ($p = 0.9004$) and Anderson–Darling test ($p = 0.153$) again confirm that the residuals are normally distributed; homoscedasticity was verified by visual inspection of the residuals, confirming the validity of the ANOVA. 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 further support the conclusion that network architecture, and particularly the architecture of the encoder, is the principal driver of model performance for this policy, and suggest that the performance of the NCO policy considered in this work improves asymptotically as a function of network size.

4.7. Impact of training dataset size on policy performance

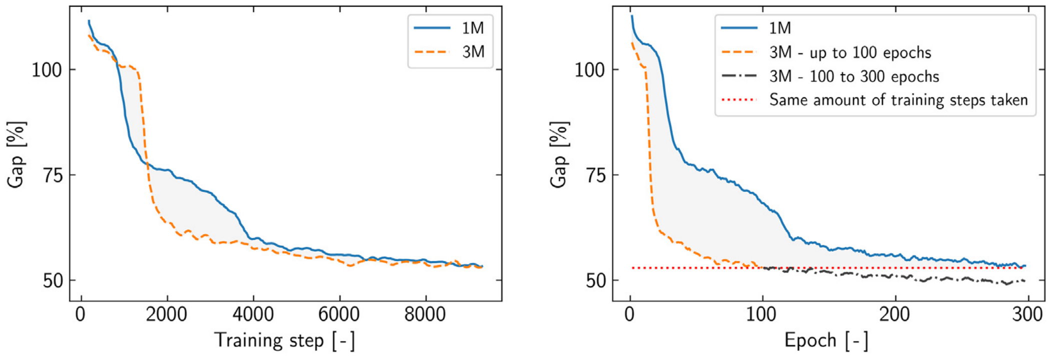
The amount of data that the model is exposed to during training is key for its performance (Berto et al., 2024b). This is especially the case for attention-based ML models (Vaswani et al., 2017; Kaplan et al., 2020; Hoffmann et al., 2022). Fig. 14a shows the effect of increasing the size



(a) Pareto front of the full factorial design. BS is used to search the learned policy in the testing stage, yielding the performance represented above.

(b) Grid search learning curve. During training the policy is decoded using greedy search to obtained the training validation reward depicted in the figure.

Fig. 13. Summary of the performance and training of the policy architectures considered in the grid search.



(a) Learning curves of the final model as a function of training step. Note the impact of increasing training dataset size from 1M to 3M.

(b) Final learning curves. The performance of the model asymptotically approaches a similar limit at 50% optimality gap.

Fig. 14. Final learning curves.

Table 16
Policy performance compared to the DRW STSP heuristic for STSPs with 10, 30 and 50 targets. Values given as: $\mu_{\pm\sigma}$. Performance measured on test datasets of 1000 missions. Bold: best result.

Number of nodes	**10**		**30**		**50**	
Heuristic	AM BS	DRW	AM BS	DRW	AM BS	DRW
ΔV [km/s]	**106,4**$_{\pm 16,2}$	122,1$_{\pm 31,4}$	573,5$_{\pm 83,9}$	**208,1**$_{\pm 46,2}$	1041,8$_{\pm 68,2}$	**261,2**$_{\pm 52,6}$
Optimality gap [%]	**32,6**$_{\pm 25,5}$	50,5$_{\pm 36,8}$	616,0$_{\pm 133}$	**50,4**$_{\pm 37,2}$	555,4$_{\pm 97,4}$	**63,9**$_{\pm 38,3}$
Inference time [ms]	**99,2**$_{\pm 0,0}$	2494,9$_{\pm 0,0}$	**3592,0**$_{\pm 0,0}$	8534,5$_{\pm 0,0}$	**8630,0**$_{\pm 0,0}$	16147,1$_{\pm 0,0}$

of the training dataset from 1 M to 3 M tours of 10 transfers. Convergence speed and consistency was observed to improve. No gains in model performance were observed from using larger training datasets: in both cases model performance plateaus to a validation optimality gap in training of approximately 50%. Fig. 14b shows the final learning curves, extending training from 100 to 300 epochs using the 3 M tour training dataset. Performance gains from the extended training were marginal. This result confirms that the architecture of the policy is the factor limiting further learning.

4.8. Final performance and generalization to larger routing problems

To evaluate the capacity of the final trained 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 16 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 inference 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.

5. Conclusion

This study evaluated the applicability and effectiveness of Neural Combinatorial Optimization (NCO) methods for space Vehicle Routing Problems (VRPs), with a focus 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 debris cloud was constructed to generate millions of realistic ADR scenarios in which to train NCO policies. An electric propulsion ADR spacecraft concept was employed, and Lyapunov Feedback Control (LFC) was used to generate low-thrust trajectories. Specifically, the Rendezvous Q-Law LFC guidance policy was applied to produce six-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, incorporating the secular J_2 perturbations on the Right Ascension of the Ascending Node (RAAN) and the 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 three RL algorithms: REINFORCE, Advantage Actor-Critic (A2C), and Proximal Policy Optimization (PPO). A2C was found to yield the best performance. A fractional factorial ANOVA was conducted to study the impact of 13 key hyperparameters on model performance, finding embedding dimension and the number of layers in the GAT encoder to be the two most critical factors. An ablation study confirmed that the GAT encoder is the most critical architectural element for model performance. A 3-level grid search followed to determine the optimal combination of embedding dimension and encoder layers. Model performance was found to improve asymptotically with network size. Lastly, the impact of training dataset size was investigated, finding that using larger training datasets did not yield significant gains in final performance, confirming that policy architecture is the factor limiting further learning.

The trained NCO policy in combination with a Beam Search policy search strategy achieved a mean optimality gap of 32% with respect to the near-optimal Heuristic Combinatorial Optimization (HCO) benchmark in 1000 missions with 10 transfers. The NCO policy outperformed the Dynamic RAAN Walk (DRW) heuristic in both mission cost and runtime. This result shows that NCO methods can be effective for ADR missions with a limited number of targets and indicate potential for efficient, autonomous multi-rendezvous routing solutions by means of NCO policies. Policy performance declined however in scenarios with more targets (30- and 50-visit sequences) than seen in training. Achieving robust performance in scenarios with variable numbers of visits remains a challenge to be solved.

Future research should follow along three main directions of improvement: NCO policy performance and efficiency, robustness in scenarios with uncertain numbers of nodes and greater target variety, and multiple spacecraft routing problems. As to the first, alternative NCO policy architectures, deep RL approaches that incorporate tree search strategies (such as Monte Carlo Tree Search), and transfer learning techniques using foundational NCO models are an attractive option to improve policy performance and efficiency. Secondly, procedures to expose policies to variable-length routing problems during training must be pursued. Lastly, adapting existing NCO approaches for multi-agent VRP to tackle multiple spacecraft routing problems is a promising research direction.

This research demonstrates the viability and potential of NCO methods to learn effective and efficient routing policies for space VRPs, in particular short-horizon routing problems and high-throughput autonomous decision-making in uncertain scenarios where solving complex optimizations on-board may not be a possibility. This also applies to other problems in spacecraft autonomy, such as sensor allocation under uncertainty. However, to realize their full potential further development is required to improve policy efficiency, and robustness in scenarios with uncertain numbers of visits. The further development and 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.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

This research would not have been possible without the support of my supervisor at the Delft University of Technology and Jesús Ramírez at SENER Aerospace & Defence. I would like to extend my thanks to Lucrezia Marcovaldi, Álex Cuenca and David Bermejo for their comradery and commitment, 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. Lastly, I would like to express my deepest gratitude to my family, and to all those that were there all along. Victoria Catalán, Román Chiva, Francesco Branca, Alejandro Montoya, Martín Camús, Silvio Peressini, Neil Richez and you —you know who you are—.

The authors would like to thank the anonymous reviewers for their helpful comments and feedback, which strengthened the overall manuscript.

Appendix A. Nomenclature

A.1. Abbreviations

A2C	Advantage Actor-Critic
ADR	Active Debris Removal
ANOVA	Analysis of Variance
AOP	Argument of Perigee
CO	Combinatorial Optimization
DISCOS	ESA Database and Information System Characterising Objects in Space
DNN	Deep Neural Network
DRW	Dynamic RAAN Walk
ECI	Earth-Centered Inertial frame
FF	Feed-Forward network
GAT	Graph Attention Network
GIT	Gridded Ion Thruster
GNN	Graph Neural Network
GPU	Graphics Processing Unit
KS	Kolmogorov–Smirnov
LC	Lyapunov Control
LFC	Lyapunov Feedback Control
LEO	Low Earth Orbit

Appendix A (*continued*)

A2C	Advantage Actor-Critic
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
MLP	Multi-Layer Perceptron
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
RL	Reinforcement Learning
RQ-Law	Rendezvous Q-Law
RCS	Radar Cross-Section
SMA	Semi-major Axis
STSP	Spacecraft Traveling Salesman Problem
TOF	Time of Flight
TSP	Traveling Salesman Problem
VRP	Vehicle Routing Problem

A.2. Greek Symbols

α	In-plane thrust angle	–
β	Out-of-plane thrust angle	–
γ	Relative inclination	rad
ΔV	Delta-V (change in velocity)	ms^{-1}
$\Delta_r, \Delta_t, \Delta_n$	Perturbing accelerations in radial, tangential, normal directions	ms^{-2}
$\Delta L_{[-\pi,\pi]}$	Difference in true longitude wrapped to $[-\pi, \pi]$	rad
μ	Earth's gravitational parameter	$3.986e14m^3s^{-2}$
θ	True anomaly	rad
Θ	Vector of spacecraft and guidance law parameters	rad
σ	Standard deviation	–
ω	Argument of Perigee	rad
Ω	Right Ascension of the Ascending Node	rad
Φ	State transition function derived from integrating the dynamics f	–
Ψ	Gauss variational equations state-space input matrix in MEE	–

References

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

Bean, J.C., 1994. Genetic algorithms and random keys for sequencing and optimization. ORSA J. Comput., 6(2), 154–160. URL:https://pubsonline.informs.org/doi/abs/10.1287/ijoc.6.2.154.

Berto, F., Hua, C., Luttmann, L. et al., 2024a. PARCO: learning parallel autoregressive policies for efficient multi-agent combinatorial optimization. URL: http://arxiv.org/abs/2409.03811. arXiv:2409.03811 [cs].

Berto, F., Hua, C., Park, J. et al., 2024b) RL4CO: an extensive reinforcement learning for combinatorial optimization benchmark. URL: http://arxiv.org/abs/2306.17100. arXiv:2306.17100 [cs].

Berto, F., Hua, C., Zepeda, N.G. et al., 2024c. RouteFinder: towards foundation models for vehicle routing problems. URL: http://arxiv.org/abs/2406.15007. doi:10.48550/arXiv.2406.15007 arXiv:2406.15007.

Betts, J.T., 1998. Survey of numerical methods for trajectory optimization. J. Guid., Control, Dynam., 21(2), 193–207. URL:https://arc.aiaa.org/doi/10.2514/2.4231.

Biesbroek, R., Aziz, S., Wolahan, A., et al., 2021. The clearspace-1 mission: ESA and clearspace team up to remove debris. In: In Proc. 8th European Conference on Space Debris. ESA Space Debris Office, Darmstadt, Germany, p. 10, URL: https://conference.sdo.esoc.esa.int/proceedings/sdc8/paper/320/SDC8-paper320.pdf.

Biesbroek, R., Innocenti, L., Wolahan, A., et al., 2017. E.DEORBIT – ESA'S active debris removal mission. In: In Proc. 7th European Conference on Space Debris. ESA Space Debris Office, Darmstadt, Germany, p. 10, URL: https://conference.sdo.esoc.esa.int/proceedings/sdc7/paper/1053/SDC7-paper1053.pdf.

Biscani, F., Izzo, D., 2020. A parallel global multiobjective framework for optimization: pagmo. J. Open Source Software, 5(53), 2338. URL: https://joss.theoj.org/papers/10.21105/joss.02338.

Blank, J., Deb, K., 2020. Pymoo: multi-objective optimization in python. IEEE Access, 8, 89497–89509. URL: https://ieeexplore.ieee.org/document/9078759. doi:10.1109/ACCESS.2020.2990567. Conference Name: IEEE Access.

Boley, A., Byers, M., 2021. Satellite mega-constellations create risks in Low Earth Orbit, the atmosphere and on Earth. Scient. Rep. 11, 10642. https://doi.org/10.1038/s41598-021-89909-7.

Bonnal, C., Ruault, J.-M., Desjean, M.-C., 2013. Active debris removal: recent progress and current trends. Acta Astronaut., 85, 51–60. URL: https://www.sciencedirect.com/science/article/pii/S0094576512004602. doi:10.1016/j.actaastro.2012.11.009.

Borelli, G., Trisolini, M., Massari, M., et al., 2021. A comprehensive ranking framework for active debris removal missions candidates. In: Proc. 8th European Conference on Space Debris, p. 10.

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

Cohen, J., 1988. Statistical Power Analysis for the Behavioral Sciences, 2nd ed. Routledge, New York. https://doi.org/10.4324/9780203771587.

Conversano, R.W., Wirz, R.E., 2013. Mission capability assessment of CubeSats using a miniature ion thruster. J. Spacecr. Rock., 50(5), 1035–1046. URL: doi: 10.2514/1.A32435.

Cormen, T.H., Leiserson, C.E., Rivest, R.L., et al., 2009. !!1048550/arXiv.1602.01783, URL: http://arxiv.org/abs/1602.01783. arXiv:1602.01783 [cs].

Morante, D., Sanjurjo Rivo, M., & Soler, M. (2021). A Survey on Low-Thrust Trajectory Optimization Approaches. Aerospace, 8(3). URL: https://www.mdpi.com/2226-4310/8/3/88. doi:10.3390/aerospace8030088.

Narayanaswamy, S., & Damaren, C.J. (2023). Equinoctial Lyapunov Control Law for Low-Thrust Rendezvous. Journal of Guidance, Control, and Dynamics, 46(4), 781–795. URL: doi: 10.2514/1.G006662. doi:10.2514/1.G006662. Publisher: American Institute of Aeronautics and Astronautics _eprint: https://doi.org/10.2514/1.G006662.

Narayanaswamy, S., Wu, B., Ludivig, P. et al. (2023). Low-thrust rendezvous trajectory generation for multi-target active space debris removal using the RQ-Law. Advances in Space Research, 71(10), 4276–4287. URL: https://www.sciencedirect.com/science/article/pii/S0273117722011656. doi:10.1016/j.asr.2022.12.049.

von Neumann, J., 1951. Various techniques used in connection with random digits. J. Res. National Bureau of Stand., URL: https://mcnplanl.gov/pdf_files/InBook_Computing_1961_Neumann_JohnVonNeumannCollectedWorks_VariousTechniquesUsedinConnectionwithRandomDigits.pdf.

NRC (2011). Limiting Future Collision Risk to Spacecraft: An Assessment of NASA's Meteoroid and Orbital Debris Programs. Washington, D.C.: National Academies Press. URL: http://www.nap.edu/catalog/13244. doi:10.17226/13244.

O'Reilly, D., Herdrich, G., & Kavanagh, D.F. (2021). Electric Propulsion Methods for Small Satellites: A Review. Aerospace, 8(1), 22. URL: https://www.mdpi.com/2226-4310/8/1/22. doi:10.3390/aerospace8010022. Number: 1 Publisher: Multidisciplinary Digital Publishing Institute.

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

Paszke, A., Gross, S., Massa, F., et al., 2019. PyTorch: an imperative style, high-performance deep learning library. In: In Proceedings of the 33rd International Conference on Neural Information Processing Systems 721. Curran Associates Inc., Red Hook, NY, USA, pp. 8026–8037.

Petropoulos, A., 2004. Low-Thrust Orbit Transfers Using Candidate Lyapunov Functions with a Mechanism for Coasting. In AIAA/AAS Astrodynamics Specialist Conference and Exhibit Guidance, Navigation, and Control and Co-located Conferences. American Institute of Aeronautics and Astronautics, p. (p. 16).. https://doi.org/10.2514/6.2004-5089, URL: https://arc.aiaa.org/doi/10.2514/6.2004-5089.

Petropoulos, A. (2005). Refinements to the Q-law for low-thrust orbit transfers.

Petropoulos, A., Grebow, D., Jones, D., et al., 2017. GTOC9: Methods and Results from the Jet Propulsion Laboratory Team.

Ricciardi, L., & Vasile, M. (2019). Solving multi-objective dynamic travelling salesman problems by relaxation. In GECCO '19: Proceedings of the Genetic and Evolutionary Computation Conference Companion (p. 9). Prague Czech Republic: Association for Computing Machinery. 2007, doi:10.1145/3319619.3326837 pages.

Russel, S., Norvig, P., 2020. Artificial Intelligence: A Modern Approach, volume 175. Pearson Education (4th ed.)..

Saunders, C., Forshaw, J., Lappas, V., et al., 2014. Business and economic considerations for service oriented active debris removal missions. Proceedings of the International Astronautical Congress, IAC 3, 1729–1743.

Schulman, J., Wolski, F., Dhariwal, P. et al. (2017). Proximal Policy Optimization Algorithms. URL: http://arxiv.org/abs/1707.06347. doi:10.48550/arXiv.1707.06347 arXiv:1707.06347 [cs].

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

Sellmaier, F., Boge, T., Spurmann, J., et al., 2010. On-Orbit Servicing Missions: Challenges and Solutions for Spacecraft Operations. In: In SpaceOps 2010 Conference. American Institute of Aeronautics and Astronautics, Huntsville, Alabama, p. 11. https://doi.org/10.2514/6.2010-2159, URL: https://arc.aiaa.org/doi/10.2514/6.2010-2159.

Shan, M., Guo, J., & Gill, E. (2016). Review and comparison of active space debris capturing and removal methods. Progress in Aerospace Sciences, 80, 18–32. URL: https://www.sciencedirect.com/science/article/pii/S0376042115300221. doi:10.1016/j.paerosci.2015.11.001.

Shapiro, S.S., Wilk, M.B., 1965. An analysis of variance test for normality (complete samples)†. Biometrika 52 (3–4), 591–611. https://doi.org/10.1093/biomet/52.3-4.591.

Sorenson, S.E., Pinkley, S.G.N., 2023. Multi-orbit routing and scheduling of refuellable on-orbit servicing space robots. Comput. Industr. Eng., 176, 108852. URL:https://www.sciencedirect.com/science/article/pii/S0360835222008403. doi:10.1016/j.cie.2022.108852.

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

Taguchi, G., Konishi, S., 1987. Orthogonal arrays and linear graphs: tools for quality engineering. American Supplier Institute, Google-Books-ID: _5BRnQAACAAJ.

Varga, G.I., Perez, J.M.S., 2016. Many-revolution low-thrust orbit transfer computation using equinoctial Q-Law Including J2 and Eclipse Effects. ICATT, URL:https://indico.esa.int/event/111/contributions/346/attachments/336/377/Final_Paper_ICATT.pdf.

Vaswani, A., Shazeer, N., Parmar, N. et al. (2017). Attention is All you Need. In Advances in Neural Information Processing Systems. Curran Associates, Inc. volume 30. URL: https://proceedings.neurips.cc/paper_files/paper/2017/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html.

Vinyals, O., Fortunato, M., & Jaitly, N. (2015). Pointer Networks. In Advances in Neural Information Processing Systems. Curran Associates, Inc. volume 28. URL: https://proceedings.neurips.cc/paper_files/paper/2015/hash/29921001f2f04bd3baee84a12e98098f-Abstract.html.

Vinyals, O., Fortunato, M., Jaitly, N., 2017. Pointer Networks. URL: http://arxiv.org/abs/1506.03134. doi:10.48550/arXiv.1506.03134 arXiv:1506.03134.

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

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

Wiedemann, C., Krag, H., Bendisch, J. et al. (2004). Analyzing costs of space debris mitigation methods. Advances in Space Research, 34(5), 1241–1245. URL: https://www.sciencedirect.com/science/article/pii/S0273117704001048. doi:10.1016/j.asr.2003.10.041.

Williams, R.J., 1992. Simple statistical gradient-following algorithms for connectionist reinforcement learning. Mach. Learn. 8 (3), 229–256. https://doi.org/10.1007/BF00992696.

Yang, H., Hu, J., Li, S. et al., 2024. Reinforcement-learning-based robust guidance for asteroid approaching. J. Guid., Control, Dynam., 47(10), 2058–2072. URL:https://arc.aiaa.org/doi/10.2514/1.G008085.