INNOVATION IN PLANNING SPACE DEBRIS REMOVAL MISSIONS USING ARTIFICIAL INTELLIGENCE AND QUANTUM-INSPIRED COMPUTING

David Snelling[(1)], **Ellen Devereux**[(1)], **Niven Payne**[(1)], **Matthew Nuckley**[(1)], **Giulia Viavattene**[(2)], **Matteo Ceriotti**[(2)], **Stephen Wokes**[(3)], **Giuseppe Di Mauro**[(3)], **and Harriet Brettle**[(3)]

[(1)] *Fujitsu Services, 22 Baker Street, London, UK*
[(2)] *James Watt School of Engineering, University of Glasgow, UK*
[(3)] *Astroscale, Harwell Science Park, UK*

ABSTRACT

This paper proposes an optimisation solution and tool-set for planning an active debris removal mission, enabling a single spacecraft to deorbit multiple space debris objects in one mission efficiently. A two-step strategy is proposed; first, an Artificial Neural Network is trained to predict the cost of orbital transfer to and disposal of a range of debris objects quickly. Then, this information is used to plan a mission of four captures from 100 possible debris targets using Fujitsu's quantum-inspired optimisation technology, called Digital Annealer, by formulating the problem as a quadratic unconstrained binary optimisation. In validation, this platform produced a 25% faster mission, using 18% less propellant when compared to an expert's attempt to plan the mission using the same assumptions, this solution was found 170,000 times faster than current methods.

Keywords: space; debris; removal; artificial neural network; quantum annealing; QUBO; Digital Annealer.

1. INTRODUCTION

In order to keep a safe access to and usage of space in the coming years, it will be necessary to limit the number of space debris objects, such as non-functional spacecraft and abandoned launch vehicle stages, that populate the low Earth orbit (LEO) region. In addition to design for demise of future space missions, debris objects currently in-orbit can be actively removed with dedicated missions. This process is known as active debris removal (ADR). To make an ADR mission commercially viable, planning is required on how to remove multiple space debris objects using a single removal spacecraft. This approach can reduce the overall launch cost and make ADR missions more commercially viable. This study aimed at optimising the removal and disposal of multiple debris objects using a single spacecraft (referred to as the "chaser" throughout this paper). The considered mission scenario requires a chaser that is able to rendezvous and dock with the debris object, then descend to a disposal orbit and release the object for deorbiting and re-entry in the Earth's atmosphere, before the chaser transfers to the next target object and the procedure repeats until propellant is depleted. The debris population used for this study was assumed to have a circular orbit with the same inclination and a variety of altitudes and right ascensions of the ascending nodes (RAANs).

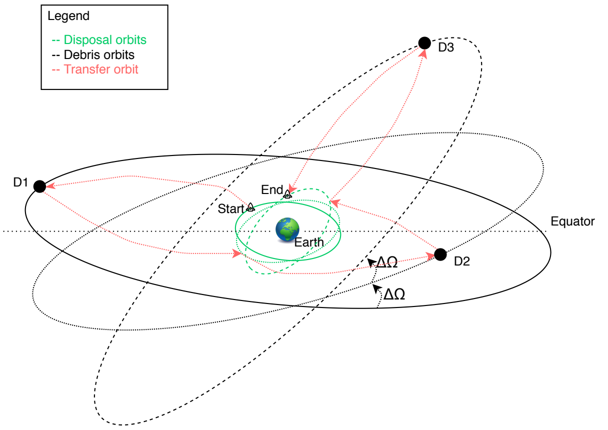

Figure 1. Schematic representation of the mission scenario.

A schematic representation of the mission scenario is provided in Fig. 1. From the point of view of mission design, the problem is challenging for two reasons: first, the debris objects to be disposed of, and the sequence of their disposal, shall be selected in such way that the overall mission cost is minimised. Second, each orbital transfer between debris orbit and disposal orbit must be designed to minimise the manoeuvring change in velocity (Δv) and/or time of flight (TOF).

To identify the best sequence for debris removal given an input set, all the permutations of debris should be explored and evaluated. According to the European Space Agency report [5], 95% of the trackable (larger than 10 cm) objects in near-Earth space are pieces of debris. The number sharply rises from 29,000 (larger than 10 cm) to 670,000 objects of sizes larger than 1cm, to more than 170 million for sized larger than 1mm. It follows that there are multiple trillions of permutations for ADR missions between these objects that would need to be investigated for their efficient removal. Since low-thrust transfers have no analytical closed-form solutions, an optimisation strategy must be used to find a solution to trajectory design problems, which are generally computationally demanding.

Artificial Neural Networks (ANN) can be trained to calculate the cost of a trajectory in terms of Δv and TOF, given the departure and arrival orbit, in a fraction of time needed using conventional optimisation methods. This provides an estimate that can help to select candidates for further analysis through optimal control problems. A similar methodology was designed and successfully used for multiple-asteroid missions [12, 14, 13]. During this study this methodology is applied for the first time in the design of multiple space debris removals. This approach allows the evaluation of the Δv and TOF for an ADR chaser for many pieces of debris at a modest computational cost. However, even with an ANN predicting the costs of optimal transfers that still leaves millions of options for an ADR mission sequence. This type of problem, where there are many more combinations of solutions than is sensible to calculate exhaustively, is referred to as combinatorial optimisation [7]. These problems scale exponentially which is incredibly computationally intensive for classical computing. However, they will be some of the first problems that quantum computers can solve via a process known as quantum annealing . Fujitsu has developed a piece of technology that is particularly good at solving combinatorial optimisation and is addressed in the same way as quantum annealers of the future will be, it is called the Digital Annealer (DA) [6, 8]. The DA and quantum annealers can only be used to solve combinatorial optimisation problems that are written in the form of a quadratic unconstrained binary optimisation (QUBO) [7].

2. THE SOLUTION

The proposed solution is a combination of the 2 technologies discussed, an ANN and the DA. The technologies are combined into the solution architecture described in the diagram in Fig. 2: first, a potential customer of the platform such as Astroscale submits their target debris data including the mass, altitude, initial position and desirability of the object to the platform. This data is passed to the ANN which provides a fast prediction of orbital transfer costs in terms of Δv and TOF and this information is passed to the DA which calculates an optimal sequence based on a range of priorities, such as mission length, fuel and desirability of the debris removal. The final stage of the platform includes calculating an accurate mission plan that defines the Δv, TOF and propellant usage based on the consumption throughout the mission. The following sections describe in detail the different components of the solution.

2.1. Equation of Motion and Dynamics

The dynamics of the spacecraft are described and propagated using modified equinoctial elements, in order to avoid numerical singularities for zero eccentricity and inclination. The modified equinoctial elements are defined from the orbital elements as follows:

$$
\begin{aligned}
p &= a(a - e^2) && (1)\\
f &= e\cos(\omega + \Omega) && (2)\\
g &= e\sin(\omega + \Omega) && (3)\\
h &= \tan(0.5i)\sin(\Omega) && (4)\\
k &= \tan(0.5i)\cos(\Omega) && (5)\\
g &= \omega + \Omega + \theta && (6)
\end{aligned}
$$

with p being the semi-latus rectum, f and g the elements that describe the eccentricity, h and k elements that describe the inclination, and L is the true longitude. Here, the semi-major axis, a, the eccentricity, e, the inclination i, the RAAN Ω, the argument of perigee ω, and the true anomaly θ constitute the classical orbital elements [1].

The differential equations describing the motion of the spacecraft can be formulated as:

$$\dot{\mathbf{x}}(t) = \mathbf{A}(\mathbf{x})\mathbf{a} + \mathbf{b}(\mathbf{x}) \tag{7}$$

with $\mathbf{a}$ being the spacecraft perturbing acceleration in radial, transversal, out-of-plane components, and $\mathbf{A}(\mathbf{x})$ and $\mathbf{b}(\mathbf{x})$ being, respectively, the matrix and the vector of the dynamics. A full definition of the matrix $\mathbf{A}(\mathbf{x})$ is given as [2]:

$$\mathbf{A} = \begin{bmatrix} 0 & a_{1,2} & 0 \\ a_{2,1} & a_{2,2} & a_{2,3} \\ a_{3,1} & a_{3,2} & a_{3,3} \\ 0 & 0 & a_{4,3} \\ 0 & 0 & a_{5,3} \\ 0 & 0 & a_{6,3} \end{bmatrix} \tag{8}$$

where:

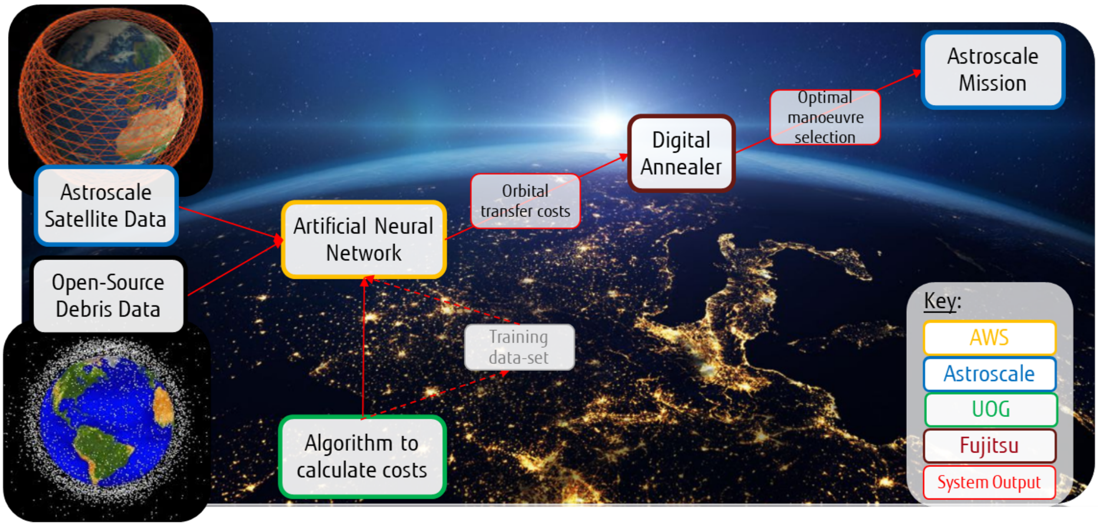

Figure 2. A solution architecture diagram for the optimiser platform. The colour outline shows who is responsible for each part of the solution. The red arrows depict the flow of information between each stage.

$$a_{1,2} = \frac{2p}{q}\sqrt{\frac{p}{\mu}} \tag{9}$$

$$a_{2,1} = \sqrt{\frac{p}{\mu}}\sin(L) \tag{10}$$

$$a_{2,2} = \sqrt{\frac{p}{\mu}}\frac{1}{q}\left((q+1)\cos(L)+f\right) \tag{11}$$

$$a_{2,3} = -\sqrt{\frac{p}{\mu}}\frac{g}{q}\left(h\sin(L)-k\cos(L)\right) \tag{12}$$

$$a_{3,1} = \sqrt{\frac{p}{\mu}}\cos(L) \tag{13}$$

$$a_{3,2} = \sqrt{\frac{p}{\mu}}\frac{1}{q}\left((q+1)\sin(L)+g\right) \tag{14}$$

$$a_{3,3} = \sqrt{\frac{p}{\mu}}\frac{f}{q}\left(h\sin(L)-k\cos(L)\right) \tag{15}$$

$$a_{4,3} = \sqrt{\frac{p}{\mu}}\frac{s^2\sin(L)}{2q} \tag{16}$$

$$a_{5,3} = \sqrt{\frac{p}{\mu}}\frac{s^2\cos(L)}{2q} \tag{17}$$

$$a_{6,3} = \sqrt{\frac{p}{\mu}}\left(h\sin(L)-k\cos(L)\right) \tag{18}$$

and the vector $\mathbf{b}(\mathbf{x})$ is [2]:

$$\mathbf{b} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ \sqrt{\mu p}\left(\frac{q}{p}\right)^2 \end{bmatrix} \tag{19}$$

where

$$q = 1 + f\cos(L) + g\sin(L) \tag{20}$$

$$s^2 = 1 + \chi^2 \tag{21}$$

$$\chi = \sqrt{h^2+k^2} \tag{22}$$

The spacecraft perturbing acceleration, $\mathbf{a}$, is provided by the low-thrust propulsion system $\mathbf{a_T}$, the acceleration due to the oblateness of the Earth $\mathbf{a_g}$ and the acceleration due to atmospheric drag $\mathbf{a_D}$, i.e.:

$$\mathbf{a} = \mathbf{a_T} + \mathbf{a_g} + \mathbf{a_D} \tag{23}$$

where the acceleration due to the thrust is given as:

$$\mathbf{a_T} = \frac{T_{max}}{m}\mathbf{N} \tag{24}$$

where T_{max} is the maximum thrust that can be generated from the considered propulsion system, and $\mathbf{N} = [N_r, N_\theta, N_h]^T$ indicates the acceleration direction and magnitude vector in radial, transverse, out-of-plane coordinates. The mass of the spacecraft m changes with time while it is thrusting as described by the following mass differential equation:

$$\dot{m} = -\frac{T_{max}|\mathbf{N}|}{I_{sp}g_e} \tag{25}$$

with $|\mathbf{N}|$ being the magnitude of $\mathbf{N}$, which accounts for the thrust throttling, I_{sp} being the specific impulse of the relative propulsion system and $\mathbf{g_e}$ the gravitational acceleration at Earth's surface as seen in Table 1.

The acceleration due to the Earth oblateness can be defined as follows:

$$\mathbf{a_g} = \mathbf{Q_r^T} \delta \mathbf{g} \tag{26}$$

where $\mathbf{Q_R} = [\mathbf{i_r} \mathbf{i}_\theta \mathbf{i_h}]$ is the transformation matrix from rotating local-vertical-local-horizontal frame to Earth-centred inertial (ECI) frame, whose components are:

$$\mathbf{i_r} = \frac{\mathbf{r}}{||\mathbf{r}||}, \quad \mathbf{i}_\theta = \mathbf{i_h} \times \mathbf{i_r}, \quad \mathbf{i_h} = \frac{\mathbf{r} \times \mathbf{v}}{||\mathbf{r} \times \mathbf{v}||} \tag{27}$$

with $\mathbf{r}$ and $\mathbf{v}$ being, respectively, the position and velocity vectors of the spacecraft in the ECI frame. The perturbation acceleration $\delta\mathbf{g}$ is formulated as:

$$\delta\mathbf{g} = \delta g_n \mathbf{i_n} - \delta g_r \mathbf{i_r} \tag{28}$$

where $\mathbf{i_n}$ is the local north direction:

$$\mathbf{i_n} = \frac{\mathbf{e_n} - (\mathbf{e_n^T} \mathbf{i_r})\mathbf{i_r}}{||\mathbf{e_n} - (\mathbf{e_n^T} \mathbf{i_r})\mathbf{i_r}||} \tag{29}$$

and

$$\delta g_n = -\frac{\mu \cos(\phi)}{r^2} \sum_{k=2}^{n} \left(\frac{R_e}{r}\right)^k P_k'(\sin(\phi)) J_k \tag{30}$$

$$\delta g_r = -\frac{\mu}{r^2} \sum_{k=2}^{n} (k+1) \left(\frac{R_e}{r}\right)^k P_k(\sin(\phi)) J_k \tag{31}$$

with μ being the Earth gravitational constant, R_e the equatorial radius of the Earth, $r = p/q$, $P_k \sin(\phi)$ represents the k-th degree Legendre polynomial whose derivative with respect $\sin(\phi)$ is $P_k' \sin(\phi)$, J_k is the zonal harmonic coefficient for $k = 2, ..., n$. For the purpose of this work, it is sufficiently accurate to consider $k = 2$ only since the effect are considered negligible beyond the second term.

The acceleration due to the atmospheric drag is characterised by radial, transversal and normal components as:

$$\mathbf{a_D} = [a_{D_r} \quad a_{D_\theta} \quad 0] \tag{32}$$

which are defined as follows:

Table 1. Constants and system parameters.

Constant	Value
g_e	9.8066 m/s^2
μ	$3.9860 \cdot 10^{14}$ m^3/s^2
R_e	$6378.14 \cdot 10^3$ m
J_2	$1082.639 \cdot 10^{-6}$
Parameter	**Value**
T_{max}	21 mN
I_{sp}	2000 s
m_0	400 kg
C_D	2.2
S	8 m^2

$$a_{D_r} = -0.5 \rho S C_D v v_r \tag{33}$$

$$a_{D_\theta} = -0.5 \rho S C_D v v_\theta \tag{34}$$

where ρ is the atmospheric density, which can be predicted using existing models such as the *Exponential Atmospheric Model* that considers the atmosphere as composed by an ideal gas at constant temperature in a hydrostatic equilibrium [11]. Also, S is the aerodynamic surface area, C_D is the drag coefficient and v is the spacecraft velocity magnitude, with v_r and v_θ being its radial and tangential components:

$$v_r = \sqrt{\frac{\mu}{p}} \left(f \sin(L) - g \cos(L)\right) \tag{35}$$

$$v_\theta = \sqrt{\frac{\mu}{p}} \left(1 + f \cos(L) + g \sin(L)\right) \tag{36}$$

The numerical values of the physical parameters which are used to calculate the dynamics of the spacecraft are detailed in Table 1.

2.2. Transfer Model

To realise the rendezvous transfers to and/or from space debris objects, the differences of altitude and Ω between the departure and arrival orbits need to be reduced to zero. It is decided that the thrust is utilised only to perform the change in altitude, while the gravitational perturbations (J_2) are exploited to achieve the change in Ω through *waiting manoeuvres*. From the Gauss equations [3], the orbital-averaged RAAN variation rate is:

$$\dot{\Omega} = -\left[\frac{3}{2} \frac{J_2 \sqrt{\mu} R^2}{a^{7/2}(1-e^2)^2} \cos(i)\right] \tag{37}$$

which is experienced by both the spacecraft and the debris.

On the basis of the dynamics model described and the range of assumptions, an example of a transfer is presented in Fig. 3, where the changes in altitude, RAAN Ω, burn manoeuvres and mass are shown as function of time. In the considered example, the first capture of a debris with an altitude of 533.18 km and Ω of 146.78 deg is performed.

The figure shows that the spacecraft starts at the *waiting* altitude (420 km) with an optimal Ω so that the final Ω of the debris can be matched at the end of the first transfer to reach the debris altitude. A constant capture time of 30 days (to allow for rendezvous and docking) is considered where the altitude is fixed and equal to the debris altitude to allow for the capture operations to take place. This can be further appreciated from the first subplot and second subplot (in order from the top) where, respectively, the altitude stays constant and Ω_{SC} of the spacecraft and Ω_{Debris} of the debris also match for the whole duration for the capture. Since during the capture there is no change in altitude, the thrust does not operate, as illustrate in the third subplot.

Thereafter, the spacecraft will carry the debris down to disposal orbit (390 km) to deorbit it. Once completed, the spacecraft transfers again up to the waiting altitude, where a waiting manoeuvre is performed in order to match the Ω of the next debris to capture. The mass of the system (forth subplot) decreases due to the propellant mass which is burned to thrust during the transfers. During the transfer to dispose the debris, the mass of the debris being carried is considered in the mass of the system.

The chaser solar arrays, which power the solar electric propulsion (SEP) system, are subject to blackout periods during solar eclipse conditions. For the computation of the low-thrust trajectories between debris orbits and disposal orbit, and viceversa, it is necessary to account for the discontinuity in power available to the chaser. In addition, an on/off duty-cycle is needed to prevent the thruster from exceeding the power capability of the chaser. It is therefore assumed to use a thruster duty-cycle that is synchronous with the eclipses on each orbit. In the eclipse model, for which a schematic representation is presented in Fig. 4, the Earth is assumed to be spherical and to project a cylindrical shadow region in the direction opposing the sun (assumed in the equatorial plane). In this approach, the thruster of the chaser is assumed to be off when travelling through the shadow region. In order to account for a suitable duty-cycle, the thruster of the chaser is considered to be off also in the cylindrical portion of the transfer which is symmetrically opposite to the shadow region. This allows the thruster to be on along opposing arcs on the orbit (north and south), thus allowing tangential accelerations that can change the orbit semimajor axis with negligible change in eccentricity. For this work, the angle θ between the centre of the shadow cylinder and the shadow terminator point is assumed a maximum value $\theta = 36°$, resulting that the SEP system can thrust along up to 60% of the orbit. This is taken into account on the dynamics of the model of the orbital transfer.

2.3. Artificial Neural Network for Space Debris Removal

A wide variety of ANNs exists to be suitable for multiple applications. A feedforward ANN is structured in layers, each presenting a number of neurons. The neurons of a layer are connected directly to neurons of the successive layer, so that the information moves from input layer y^1 through the hidden layers to the output layer y^L. Figure 5 presents a general illustration of a neural network with L layers.

The network needs to be trained with a database containing the corresponding inputs and outputs (or *targets*) with the purpose of *learning* the network function. The network function is intended to minimise the difference between the outputs generated by the network and the targets, i.e., to minimise the network error. The training of the network consists in determining the weights w_{jk}^l associated to each connection between the k-th and j-th neuron and the biases b_j^l of the j-th neuron of the l-th layer. So, the j-th neuron of the l-th layer equals to:

$$y_j^l = \mathcal{F}^l \left(\sum_k w_{jk}^l y_k^{(l-1)} + b_j^l \right) \tag{38}$$

where subscripts and superscripts identify the neuron and hidden layer, respectively. $\mathcal{F}^l$ is the transfer or activation function of the l-th layer, which maps from the neuron's weighted input values onto a single output value. The most commonly used activation function for feedforward networks is the *sigmoid* function, which is defined as follows:

$$s_\gamma(x) = \frac{1}{1 - e^{-x/\gamma}} \tag{39}$$

where γ defines the slope of the function. Using the sigmoid function as activation function, each j-th neuron of the hidden and outer layers is computed by means of the Eq. 38, as follows:

$$y_j = \frac{1}{1 - e^{-\left(\sum_k w_{jk}^l y_k^{(l-1)} + b_j^l\right)/\gamma_j}} \tag{40}$$

with y_j being the output of the j-th neuron and $y_j \subseteq (0,1)$. Differently, each i-th neuron of the input layer provides one component of the input vector, $X_i \subseteq \mathbb{R}$. It follows that the *network function* is a parameterised function that maps from an input set $\mathcal{X}$ to an output set $\mathcal{Y}$:

$$N = \mathcal{X} \subseteq \mathbb{R}^{n_i} \rightarrow \mathcal{Y} \subseteq (0,1)^{n_0} \tag{41}$$

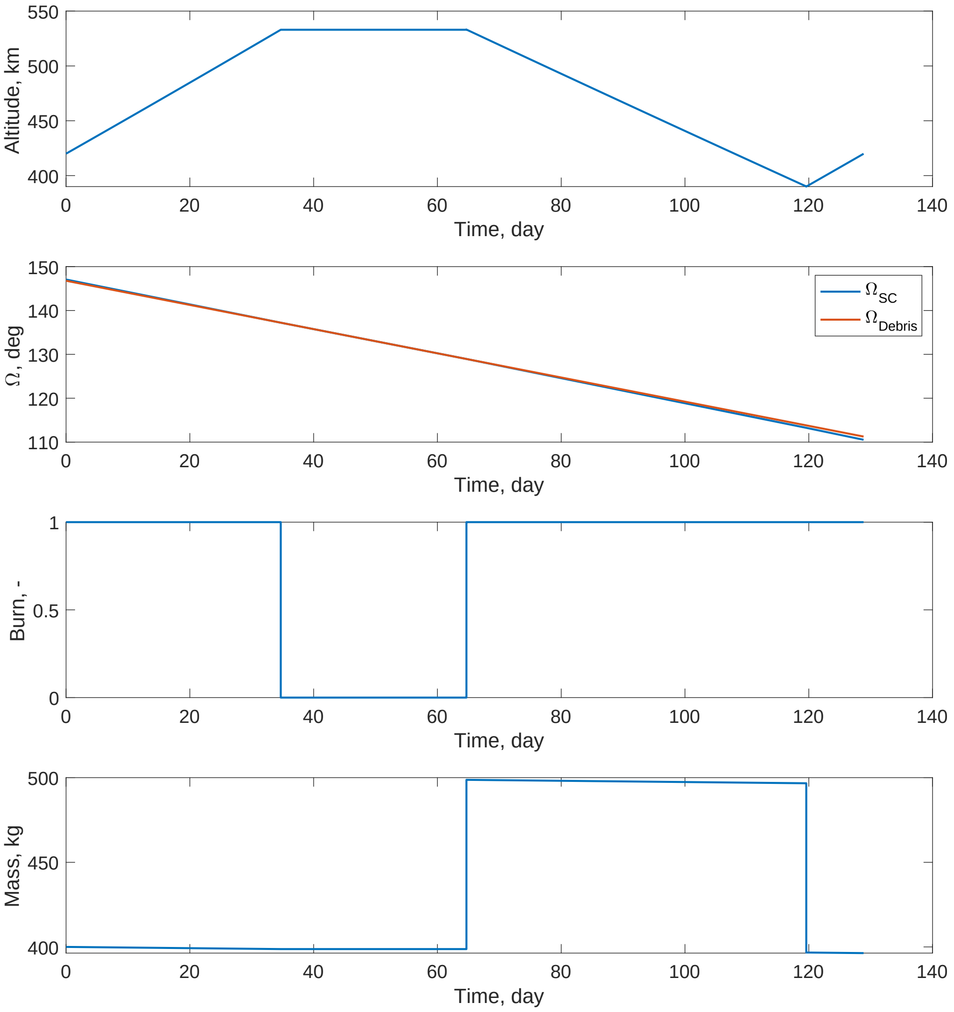

Figure 3. Altitude, Ω, burn manoeuvres and mass changes (from top to bottom) as function of time to perform one debris object capture and disposal.

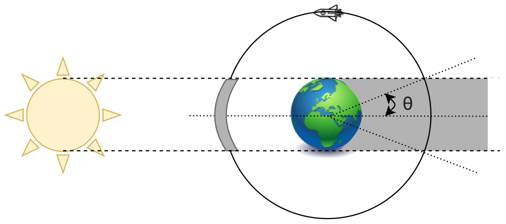

Figure 4. Eclipse model, where θ is the angle from the equator to the intersection of the edge of the eclipse and the orbit of the chaser.

with n_i and n_0 being the number of input and output neurons, respectively.

The performance of the network is affected by its architecture (i.e. number of layers and neurons) and by its hyper-parameters, such as learning algorithm, activation function, learning rate [10, 9]. In the following, the training database is generated, as presented in Section 2.3.1 and the design of the optimal ANN for this application is presented in Sec. 2.3.2.

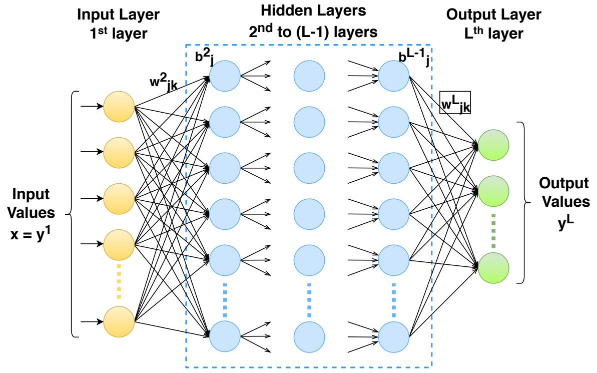

Figure 5. Illustration of an artificial Neural Network (ANN) with L layers.

2.3.1. ANN Training Database

The training database contains the input vector and the target output vector which are fed into the ANN for the training. The input vector includes the orbital parametrization of the debris object, its mass which needs to be carried to disposal and the initial mass of the spacecraft, as this will vary during the mission due to the propellant consumption. The target output vector includes the cost Δv and TOF of the low-thrust transfers between the departure and arrival satellites. Because of the limited eccentricity of the satellite orbits, circular orbits are assumed. Moreover, as a first step, it is chosen to consider coplanar transfers with the purpose of validating the model and methodology for a 'simpler' case scenario. It follows that the input vector $\mathbf{x}$ and output vector $\mathbf{y}$ can be defined as:

$$\mathbf{x} = [h_0, \Omega_0, m_0, m_{SC}] \tag{42}$$

$$\mathbf{y} = [\Delta v, t_{0,f}] \tag{43}$$

where h_0 and Ω_0 are the altitude and RAAN [3] describing the orbit of the space debris. The mass of the debris to dispose of is indicated with m_0, while the mass of the spacecraft at the beginning of the corresponding transfer is m_{SC}. The output vector contains the Δv indicates the velocity increment and $t_{0,f}$ the time of flight to perform the transfer between the two debris objects.

For the generation of both the training and test datasets, altitudes between 500 and 1500 km are considered and $\Omega = [0, 360]$ deg. The inclination is fixed to 87.9 deg and the disposal is at an altitude of 390km, which is also fixed. The mass of the satellite m_0 to dispose can vary from about 100 to 300 kg. The spacecraft mass can vary from 300 kg to 400 kg, with the latter being the starting mass, since the on-board propellant mass is 100 kg.

To compute the Δv and TOF of low-thrust transfers (*output*) between the pairs of debris, the dynamics of the spacecraft detailed in the previous section is used. The training database is build by permuting a subset of 300 satellites with different altitudes and Ω. The training database comprises a total of 90,000 low-thrust transfers.

To verify the generalisation property of the network, the database is divided into training set, validation set and test set. The training set is used for the training, while the validation and test sets contain new samples that are not included in the training. The validation set is used to verify that the overfitting does not occur during the training and the test set is used to test the performance of the network, after the training, with totally new cases. To this end, the validation-set mean square error (MSE) is often taken into consideration when studying the network performance.

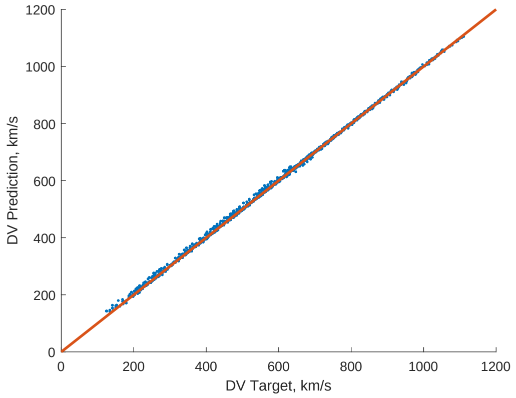

Figure 6. Performance of the network to estimate the Δv of the transfers between pairs of debris objects.

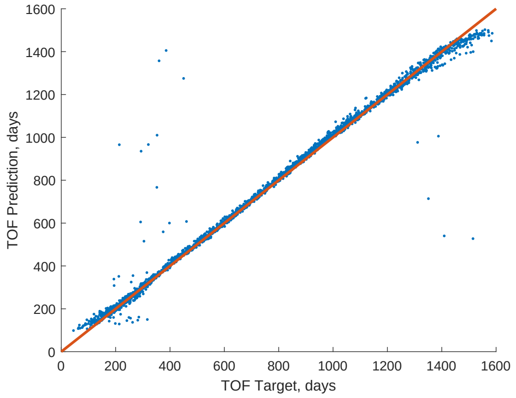

Figure 7. Performance of the network to estimate the TOF of the transfers between pairs of debris objects.

2.3.2. ANN Training

The ANN was trained in the Amazon Web Services (AWS) environment using the *Autopilot* feature in Amazon SageMaker Studio [4], 6,000 test samples were fed into the networks and their outputs were compared with respect to the targets values of Δv and TOF. Figures 6 and 7 represent the performance of the networks to estimate the Δv and TOF. The plots present how well the network outputs (Y-axis) fit the targets (X-axis) with respect to the test set. A perfect fit, thus a perfect performance of the network, is obtained when the data fall along the line with a unit slope and zero y-intercept. This means that the relationship between the outputs and the targets is $\mathbf{y} = \mathbf{x}$.

Furthermore, considering a bad prediction when a difference between the output and the target larger than 5% is obtained, no bad predictions are registered for the Δv and 63 bad predictions for the TOF. From this it follows that the mean accuracy achieved from the networks is 99.93% and 99.51% for the Δv and TOF estimation, respectively. This is considered to be a very high performance for the purpose of this work.

The DA will call the trained ANN to evaluate the Δv and TOF of thousands of captures and, ultimately, identify the sequence of captures which is the most convenient in terms of cost of mission.

2.4. Digital Annealer

Fujitsu's quantum inspired DA is a piece of silicon hardware that was specifically designed by to solve combinatorial optimisation problems [8]. These are some of the first problems that quantum computers will solve via a process known as quantum annealing. The DA is unique due to the fact it is more capable than conventional, parallel processing at solving these problems and it's scale at 8000-100,000 bits outranks all quantum computers to date [6]. This means it is capable of solving very large combinatorial optimisation problems very quickly without the complex quantum architecture. Due to the vast number of combinations available for any single ADR mission plan, this was an ideal use case for the DA.

Combinatorial optimisation problems are passed to the DA as QUBOs and this section will describe the QUBO used to optimise ADR mission planning. Prior to the implementation of the DA the ANN was called to evaluate the Δv and TOF of thousands of captures. The platform then passes the predictions from the ANN to the DA. The DA finds an optimal mission plan sequence based on a range of priorities by finding the minimum energy of the QUBO and, ultimately, identifies the sequence of captures which is the most convenient in terms of the cost of the mission.

To begin with each piece of debris is defined by the following properties: altitude, desirability, mass, RAAN (at start of mission), Δv, TOF, optimal Ω^{start}, and Ω^{end}. The latter 4 of which are provided by the ANN. Then the goals of the QUBO are to; minimise wait time, maximise desirability and constrain propellant. This is described mathematically below, first, there are the QUBO structural constraints which define the mission plan. The capture constraint says that only one piece of debris can be captured in any one capture slot:

$$Capt = C_p \sum_{c=0}^{C-1} \left[1 - \sum_{d=0}^{D-1} x_{d,c}\right]^2 \tag{44}$$

where C is the number of captures, D is the number of possible target pieces of debris, C_p is the capture constraint penalty term and $x_{d,c}$ is one when debris d is captured on manoeuvre c and zero otherwise. Then there's the debris constraint that says a piece of debris can only be captured and disposed of once:

$$Debr = D_p \sum_{d=0}^{D-1} \left[1 - sl_d - \sum_{c=0}^{C-1} x_{d,c}\right]^2 \tag{45}$$

where D_p is the satellite constraint penalty term and sl_d is a slack bit for debris that is not captured on this mission. Finally there is the constraint of the propellant for the EP system to make sure that the amount of propellant used does not exceed the amount available:

$$Propellant = P_w(100 - \sum_{c=0}^{C-1} \sum_{d=0}^{D-1} P_d x_{d,c})^2 \tag{46}$$

where P_d is the EP propellant mass used to complete the manoeuvre that captures debris d, from wait altitude to wait altitude and P_w is the weight associated with the propellant term. Then there are the objective terms which control the priorities of the results, first to maximise the desirability:

$$Desirability = -D_w \sum_{c=0}^{C-1} \sum_{d=0}^{D-1} D_d x_{d,c} \tag{47}$$

where D_w is the weight associated with the mission desirability term and D_d is the desirability value of the debris d. Then to minimise the wait between captures:

$$Wait = W_w \sum_{c=0}^{C-2} \sum_{d0=0}^{D-1} \sum_{d1=0}^{D-1} W_{d0,d1,c} x_{d0,c} x_{d1,c+1} \tag{48}$$

where the wait between each debris object is defined as:

$$W_{d0,d1,c} = mod\big((\Omega_{d0}^{end} + c \cdot TOF_{avg} \cdot drift_{d0}), 360) - mod((\Omega_{d1}^{start} + (c \cdot TOF_{avg} + TOF_{d0}) \cdot drift_{d1}), 360\big) \tag{49}$$

and

$$drift_d = \frac{d\Omega_d}{dt} \tag{50}$$

where TOF_{ave} is the average TOF in days of all debris removal manoeuvres, $\frac{d\Omega_d}{dt}$ in the RAAN drift per day at altitude debris d and W_w is the weight associated with the wait time term.

So the final Hamiltonian can be written as:

$$qubo = Capt + Debr + Propellant + Desirability + Wait \tag{51}$$

The mission optimisation platform finds the minimum energy of the QUBO for all of the input data set of debris very quickly. The sequence selected by the DA is then fed into the algorithm described in Sec. 2.3.1 to create a full mission plan for the customer.

3. VALIDATION METHODS

The primary assessment of the impact of the optimisation solution is a like for like comparison of a mission plan for 4 targets captured from a choice of 100 valid targets. The 100 satellite targets were a randomly created set of objects with random mass, position and altitude. The methods used to validate the efficacy of the platform were as follows: comparison with an industry expert processing the same input as the platform and coming to the best solution possible within 4 hours via an iterative approach and comparison with an exhaustive search of the possible results from the ANN. These methods were used to measure whether the DA adds value to the platform. For this task desirability of the captured debris is not taken into account as it would have been difficult for the ANN and the by hand method to consider this in a measurable way. Finally, the platform was also used to optimise a mission picking from 200 satellites for 10 captures to further test the capabilities of the platform beyond what was possible using the previously discussed methods.

All methods used the same set of input data and the same assumptions. The assumptions were as follows: all pieces of debris had orbits with eccentricity and inclination of zero and the duty cycle of the chaser was 23%. These assumptions were chosen to keep the minimum viable product of the platform simple and to match Astroscale's business case, however, they are representative of a real mission and therefore can be easily varied for future use cases. The results found using each of the methods were compared by running the final mission sequence through the algorithm applied in the final stage of the platform (also used to build the training data for the ANN). This provides a fair comparison between the methods.

4. RESULTS

The results from the validation activities described in the previous section can be viewed in Table 2 and Table 3. The inputs to the platform are the number of satellites considered by each method and the number of captures the solution should attempt. The sequence refers to the index of the satellites to be captured and the order in which they should be captured. The compute time is the time taken to select the sequence, this does not include the time required to run the ANN calculations or the final algorithm for the sequence as the rest of the platform can be sped up with a productionised solution so this is the most representative measurement of method capability. The approximate Δv and approximate TOF are the Δv and TOF calculated by each method for the optimisation task and the algorithm Δv and algorithm TOF are calculated using the algorithm imposed in the final stage of the platform this was used as a consistent measure of good between the approaches. The comparison to an expert working by hand is viewed in Table 2, it can be seen that for the same input data the optimisation platform compute time was more than 170,000 times shorter than the compute time for the expert using current methods. The algorithm Δv and TOF found by the expert also greatly exceeded those found by the full platform.

The exhaustive ANN search approach used different assumptions but has been included to highlight the need for the DA and the results from this comparison can be viewed in Table 3. For the initial cases of 10 satellites, the platform and the exhaustive search found the same sequence. For the case of 20 initial satellites, the ANN exhaustive search was asked to minimise TOF (since Δv and TOF are for the most part proportional) therefore, based on the ANN approximated results time was minimised. Although, the algorithm results for these runs show that there are some small inaccuracies presented by the prediction using the ANN, the platform found a more optimal result for the case of 3 captures with both reduced TOF and Δv. In the case with 4 captures, the ANN found a faster sequence however, the complete platform found a sequence that used less fuel, therefore, either of these results could be considered optimal depending on the customers priorities. The important difference between these results is the compute time. The exhaustive search of the ANN took nearly 200,000 times longer than the optimisation performed by the DA. It can be seen that computing more complex sequences scales linearly for the mission optimisation platform and exponentially for conventional computing methods. To demonstrate this scaling further, the last row of Table 3 shows that expanding the problem size from 4 captures chosen from 100 pieces of debris to 10 captures from 200 pieces of debris still takes less than 2 seconds to compute using the full platform. This expansion requires more than 4 times the number of bits (from 499 bits to 2199 bits) to describe the problem to the DA but scales the problem size by a factor of 8×10^{14}.

5. CONCLUSION

It can be seen from these results that the full platform incorporating both the ANN and the DA technologies provides a significant advantage to mission planning when compared to an exhaustive calculation and an experts best attempt using current methods. It is understood that current calculation methods are not of the standard that would be acceptable for mission planning but since this is the first optimiser tool of its kind there was little else to use for comparison. Therefore, comparison with an expert using iterative, by-hand methods for 4 hours supplied an appropriate base level comparison.

The comparison to an exhaustive calculation using just the ANN shows that calculating all of the possible permutations for each mission plan is computationally expensive as it takes nearly 170,000 times as long to calculate the optimal mission plan exhaustively than using the DA to find the best mission plan from the predictions. Therefore, at larger problem sizes the problem becomes intractable using exhaustive methods.

Finally, the comparison to an expert using current methods shows that there was an advantage in speed to select a solution, and an advantage in the optimisation of the selected sequences. It can be seen that the platform finds a more optimal sequence by comparing the algorithm values for each of the methods as the platform finds a 25% faster mission, using 18% less propellant when compared to the expert methods. In practice these results mean reduced chaser weight at launch leading to reduced launch

Table 2. Comparison of Mission Planning platform against a subject matter expert by hand.

Method	Inputs		Sequence	Compute time [s]	Proxy Δv [km/s] (propellant [kg])	Proxy TOF [days]	Algorithm Δv [km/s] (propellant [kg])	Algorithm TOF [days]
	Satellites	Captures						
Expert by hand	100	4	24, 1, 29, 54	14400	1.85 (41.8)	2283	1.479 (30.7)	2307
Full Platform	100	4	64, 1, 100, 44	0.083	1.100	1826	1.100 (24.9)	1717

Table 3. A comparison of the mission planning platform with an exhaustive search using the ANN.

Method	Inputs		Sequence	Compute time [s]	Proxy Δv [km/s]	Proxy TOF [days]	Algorithm Δv [km/s] (propellant [kg])	Algorithm TOF [days]
	Satellites	Captures						
Exhaustive ANN	10	3	8, 4, 1	56.40	0.67	384	0.828 (18.1)	395
		4	8, 4, 1, 7	445.80	1.05	630	1.204 (26.9)	627
	20	3	18, 12, 3	577.20	0.5	220	0.681 (15.8)	403
		4	19, 18, 12, 3	7765.20	0.71	333	0.889 (20.9)	545
Full Platform	10	3	8, 4, 1	0.040	0.83	365	0.828 (18.1)	395
		4	8, 4, 1, 7	0.040	1.28	588	1.204 (26.9)	627
	20	3	13, 4, 1	0.041	0.64	316	0.622 (13.7)	370
		4	11, 13, 4, 1	0.041	0.93	533	0.838 (19.0)	582
	200	10	84, 118, 103, 87, 113, 4, 144, 1, 127, 81	1.284	2.63	1521	1.819 (38.1)	1169

costs and a more efficient mission with minimal wait time resulting in reduced risk to mission completion. Furthermore, the platform found this solution in a fraction of a second, thousands of times faster than using current methods.

Overall, the performance of the optimisation solution is beneficial in three areas; speed of calculation, optimisation of the mission and additional options to the service operator. It was found that the mission optimisation tool reduced the overall launch cost and helped to remove space debris efficiently. Furthermore, the significant speed up in computation time allows for new options for mission planning such as rapid simulation of a variety of scenarios and priorities which would create improvements for spacecraft design, mission planning and price setting. Based on these outcomes, this solution offers opportunities to other use cases within the space sector such as earth observation, in-orbit servicing and rapid response to changing live mission conditions.

ACKNOWLEDGEMENTS

This research was carried out as part of the UK Space Agency grant “Advancing Research into Space Surveillance and Tracking: to assess how to make space debris removal missions more commercially viable”. The project leverages artificial neural network (ANN) based trajectory design algorithms developed by the University of Glasgow and Fujitsu’s quantum-inspired technology the Digital Annealer to solve the optimisation problems associated with ADR mission planning design. Amazon Web Services (AWS) provided cloud and artificial intelligence/ machine learning (AI/ML) tools and services to support the project. Astroscale provided the end-use case as a representative user of a multi-target mission optimisation tool and expertise throughout. Special thanks to Tom Gray from AWS, whose support throughout the project has been vital.

REFERENCES

1. Betts, J. T. 1998, Journal of Guidance, Control, and Dynamics, 21, 193
2. Betts, J. T. 2010, Practical Methods for Optimal Control and Estimation Using Nonlinear Programming, 2nd edn. (Philadelphia: SIAM Press)
3. Curtis, H. 2005, Orbital Mechanics for Engineering Students (Elsevier)
4. Das, P., Ivkin, N., Bansal, T., et al. 2020, in Proceedings of the Fourth International Workshop on Data Management for End-to-End Machine Learning, 1–7
5. ESA. 2021, How many space debris objects are currently in orbit?
6. Fujitsu. 2019, Quantum Future - Quantum Present, Tech. rep., Fujitsu
7. Glover, F., Kochenberger, G., & Du, Y. 2018, arXiv preprint arXiv:1811.11538
8. Matsubara, S., Takatsu, M., Miyazawa, T., et al. 2020, in 2020 25th Asia and South Pacific Design Automation Conference (ASP-DAC), IEEE, 667–672
9. Sánchez-Sánchez, C. & Izzo, D. 2018, Journal of Guidance, Control, and Dynamics, 41, 1122
10. Sanchez-Sanchez, C., Izzo, D., & Hennes, D. 2016, IEEE Symposium Series on Computational Intelligence, SSCI 2016, 1
11. Schroeder, D. V. 2000, An Introduction to Thermal Physics, 3rd edn. (Addison-Wesley)
12. Viavattene, G. & Ceriotti, M. 2019, 70th International Astronautical Congress (IAC), Washington DC, USA
13. Viavattene, G. & Ceriotti, M. 2020, Springer, Lectures Notes in Computer Science
14. Viavattene, G. & Ceriotti, M. 2020, AAS/AIAA Astrodynamics Specialist Conference