An Enhanced TSP-Based Approach for Active Debris Removal Mission Planning

João Batista Rodrigues Neto(✉) and Gabriel de Oliveira Ramos(✉)

Graduate Program in Applied Computing, Universidade do Vale do Rio dos Sinos,
São Leopoldo, RS, Brazil
jrneto@edu.unisinos.br, gdoramos@unisinos.br

Abstract. The extensive exploration of the Low Earth Orbit (LEO) has created a dangerous spacial environment, where space debris has threatened the feasibility of future operations. In this sense, Active Debris Removal (ADR) missions are required to clean up the space, deorbiting the debris with a spacecraft. ADR mission planning has been investigated in the literature by means of metaheuristic approaches, focused on maximizing the amount of removed debris given the constraints of the spacecraft. The state-of-the-art approach uses an inver-over and maximal open walk algorithms to solve this problem. However, that approach fails to deal with large instances and duration constraints. This work extends the state of the art, increasing its performance and modeling all the constraints. Experimental results evidence the improvements over the original approach, including the ability to run for scenarios with thousands of debris.

Keywords: Time-dependent traveling salesman problem · Space debris · Active debris removal · Genetic algorithm

1 Introduction

In the1960 s, space exploration began to boost the development of new technologies feasible through the use of satellites distributed in orbits. Among these orbits, Low Earth Orbit (LEO) was widely used for satellite networks with a large number of objects per service in orbit [20]. Over time, these satellites became depreciated, lose communication or got out of control, thus becoming space debris. A high population of debris represents a hazard to the operating structures in orbit, since they are objects out of control and at high speed [12].

According to some predictive models, a sufficiently large population of debris will increase the probability of collisions and, therefore, increase the debris population again, thus making this population increase recursively for many years. This phenomenon is known as Kessler syndrome and may cause the collapse of the LEO, rendering it useless for years [12].

In fact, the literature already pointed out that the debris population in LEO has already reached a critical point [16], and now measures to mitigate this situation

A. Britto and K. Valdivia Delgado (Eds.): BRACIS 2021, LNAI 13073, pp. 140–154, 2021.
https://doi.org/10.1007/978-3-030-91702-9_10

are needed. While there are documented techniques that would stabilize the debris population, for now, the only approach capable of reducing it consists in Active Debris Removal (ADR) missions [16]. These missions aim to clean up orbit space by forcing the re-entry of certain debris as performed by specific spacecraft. Due to the limited resources and time to perform the rendezvous maneuvers and the debris population size, the selection of the debris to be removed has became a hard combinatorial problem [3,4,7].

A basic ADR should consider the bounds of cost and duration, dependent position values of the moving debris along time [3,4], and also prioritizing the removal of the most threatener debris first, thus increasing the benefits of such missions [17]. Various approaches to this problem have been proposed in the literature [7,9,13]. However, most of these works share the same limitations: small instance sizes, unbounded approaches, and non-time-dependent modeling.

In this work, we propose an enhanced genetic algorithm to optimize ADR mission planning. Our approach builds upon the work by [9], improving its performance through a novel combination of genetic operators. The final algorithm resembles the original *inver-over* genetic operator, with modifications to its reversing strategy. Moreover, a new variant of the *k-opt* algorithm is implemented using a stochastic approach. Finally, the open-walk algorithm is improved with one additional constraint. Through extensive experiments, our approach yielded better solutions with instances larger than the previous largest ones [9].

2 Literature Review

In order to solve the ADR optimization problem, a few methods have been approached and documented in the last 10 years. Exact solutions were used by Braun et al. [2] with brute force, and branch and bound variations by [3,14,19]. However, in both classes of methods can only be applied to small instances. Approximate solutions were implemented using simulated annealing [4,7], reinforcement learning [25], and genetic algorithms [18,24]. Nonetheless, all of these works were also tested only on small instances.

On the other hand, a few approximate methods considered bigger instances. Barea et al. [1] used a linear programming method, which has a high complexity as drawback. Yang et al. [26] used a greedy heuristic, but requires instance-dependent parameters. Ant Colony Optimization was used in [13,21,27], but ignoring mission constraints, strongly simplifying the cost dynamics to reduce the complexity, or even leaving mission duration unbounded. Finally, Izzo et al. [9] and Kanazaki et al. [11] used genetic algorithms, though both did not model all the necessary mission constraints.

Generally speaking, the majority of the works do not prioritize debris by hazard, consider bigger instances, model the time-dependence, or bound the cost or the mission duration, meaning that most works fail to fully meet the ADR mission requirements. Building against this background, in this work we introduce the enhanced inver-over operator to deal with large instances, and an enhanced maximal open walk algorithm to model the cost and duration constraints while prioritizing the most threatener debris.

3 Problem Formulation

An ADR problem is the combinatorial problem of finding the correct sequence to rendezvous maneuvers towards debris in order to maximize the mission profit given some constraints. In this sense, ADR can be seen as a complex variant of the Travelling Salesman Problem (TSP), where one wants to find the minimum weight path in a dynamic complete graph, where the debris are the cities and the dynamic transference trajectories are the edges. The dynamicity is due to the time-dependent cost of the transference, so the correct generalized version of the TSP will be a time-dependent TSP (TDTSP). This work will make use of the integer linear TDTSP problem formulation by [7].

Hereafter, we will follow the notation typically used in the literature [7]. Let $V = \{1, ..., n\}$ be the set of n debris. The distance tensor is represented by $C = (c_{ijtm})$, where c_{ijtm} is the cost of transfer from debris i at epoch t to debris j at epoch $t+m$. Also, let $X = (x_{ijtm})$ be a binary tensor, where $X_{ijtm} = 1$ indicates that this transference is part of the solution and $X_{ijtm} = 0$ otherwise. The n_t possible epochs of departure and arrival are discretized following $n_t \geq n + 1$ and $M \leq n_t - (n - 1)$. Usually, in order to grant some freedom at the mission planning, n_t is far larger than n, while M limits the maximum duration of the transfers.

Along these lines, the problem of finding the optimized route can be modelled as finding the X matrix that minimizes the total cost with due respect to the constraints, which can be formulated as follows.

$$\min_{x} \sum_{i=1}^{n} \sum_{j=1}^{n} \sum_{t=1}^{n_t} \sum_{m=1}^{M} c_{ijtm} x_{ijtm} \tag{1}$$

$$\text{s.t.} \sum_{i=1}^{n} \sum_{j=1}^{n} \sum_{t=1}^{n_t} \sum_{m=1}^{M} x_{ijtm} = n \tag{2}$$

$$\sum_{i=1}^{n} \sum_{t=1}^{n_t} \sum_{m=1}^{M} x_{ijtm} = 1 \qquad j = 1, ..., n \tag{3}$$

$$\sum_{j=1}^{n} \sum_{t=1}^{n_t} \sum_{m=1}^{M} x_{ijtm} = 1 \qquad i = 1, ..., n \tag{4}$$

$$\sum_{j=1}^{n} \sum_{t=2}^{n_t} \sum_{m=1}^{M} t x_{ijtm} - \sum_{j=1}^{n} \sum_{t=1}^{n_t} \sum_{m=1}^{M} t x_{ijtm} = \sum_{j=1}^{n} \sum_{t=1}^{n_t} \sum_{m=1}^{M} m x_{ijtm} \qquad i = 2, ..., n \tag{5}$$

$$\sum_{i=1}^{n} \sum_{j=1}^{n} \sum_{t=1}^{n_t} \sum_{m=1}^{M} m x_{ijtm} \leq n_t \tag{6}$$

In the above formulation Eq. (1) represents the objective function and Eqs. (2)–(6) represent the problem constraints, to which we will refer simply as constraints (2)–(6) hereafter. Constraint (2) guarantees that the solution tensor has all the n debris. Constraints (3) and (4) ensure that there is no loops in the solution (one departure and one arrival transfer for each debris). Constraint (5) enforces the transfer duration. Finally, constraint (6) limits the total mission duration. As the result, this formulation has a $\mathcal{O}(2^n)$ search space, using n^4 binary decision variables and $3n - 1$ constraints.

3.1 Orbit Transfer

The trip between one debris and another require impulses ($\varDelta v$) of the thrusters to change the orbit of the spaceship. Low thrust propulsion systems can perform this maneuvers efficiently. However, they require the optimization of the trajectories to make the mission time available [5]. Determining a minimal fuel transfer trajectory between two debris is a complex optimization problem in general case. Thus, major works simplify this task by using a generic transfer strategy [4].

The major used transfer strategies are the Hohmann and Lambert transfers. Since this work's scope does not focus on the orbital transfer optimization problem, the mechanics of the transfers will be briefly described. In [6], Hohmann is described as a minimum two-impulse elliptic maneuver to transfer from coplanar orbits. Hohmann transfer is a high thrust transfer. Since debris are not always in co-planar orbits it is also presented a variation of the Hohmann transfer, namely the Edelbaum transfer, which is a three-impulse bi-elliptic transfer that allows transferences between non-coplanar orbits.

In [5], Lambert is described as two-impulse trajectory to transfer from coplanar orbits given a certain transference duration. Also, it is possible to make use of the J_2 gravitational earth perturbation to wait for the natural alignment of orbital planes, saving some fuel but increasing the mission duration [5,7].

Finally, the cost of a transference between two debris relies on the mass of the spacecraft, since the thrusters consume propellant mass at each impulse, as the mission goes on the cost of the transfers became lower due to the mass lost in the previous maneuvers. So, it is also possible to optimize the removal sequence taking in account the resultant masses of the objects [3].

4 Proposed Approach

In this work, the ADR problem is approached with an improved heuristic solution, similar to the method used by [9] with the inver-over operator in a Genetic Algorithm and the maximal open walk algorithm. The inver-over operator optimizes the total cost of the mission with a local search strategy through genetic operations on the individuals. The maximal open walk algorithm constraints the path. This work enhances this solution with a greedier implementation of the algorithms. Moreover, to avoid the local optimum, a stochastic 2-opt is proposed to improve the solution, creating new connections. Finally, the most rewardable open walk is extracted from the best individual as the final solution.

As this is a bi-optimization problem, the complexity was divided in two stages. In the first part, the effectiveness of a solution will be given by the total cost of the removal sequence, calculated with the Edelbaum transfer [6] (Eqs. (7) and (8)), a consistent and reliable cost approximation for cheap orbital maneuvers. In the second part, the effectiveness of a solution became the removed threat of the LEO, calculated with the sum of radar cross section (RCS) area of the removed debris. The RCS area is an abstraction of the size of the debris and is widely used for the threat calculation in the literature, being a measure about how much detectable an object is for a ground radar.

$$\varDelta v = \sqrt{v_0^2 - 2V_0V_f \cos\frac{\pi}{2}\varDelta i + v_f^2} \tag{7}$$

$$\cos\varDelta i = \cos i_1 \cos i_2 + \sin i_1 \sin i_2(\cos\varOmega_1 \cos\varOmega_2 + \sin\varOmega_1 \sin\varOmega_2) \tag{8}$$

$$T = \frac{1}{2}\sqrt{\frac{4\pi^2 a^3}{\mu}} \tag{9}$$

Also, in order to minimize the complexity of the problem, major works in the literature have assumed a few dynamics simplifications. In this work, the following assumptions were made with the same purpose:

- The time dependence of the problem is relaxed by the correlation explored in [9], where an optimal solution can remain optimal up to 50 d, depending on the size of the instance.
- Since the Edelbaum transfer is an optimized variation of the Hohmann transfer [6], the duration of the transference arcs can be computed using Kepler's third law of planetary motion (Eq. (9)), which measures the orbital transference duration of an object (spacecraft) between two orbits (debris).
- The transfer cost also depends on mass of the spacecraft, that will decrease during the mission, where the fuel mass will be consumed. Moreover, the gravitational effects of the earth on the spacecraft also influence the transfer cost. In this work, the masses of the objects and the orbital perturbation effects are neglected in the cost transfer dynamics.
- There are more steps of the rendezvous process to remove a debris, and each step take some time to be performed [15]. In this work the duration of the mission will be given by the sum of the duration of the transferences, the other stages will be neglected.

4.1 Inver-Over

The inver-over operator [22] is a unary genetic operator that resembles characteristics of mutation and crossover at the same time. The evolution of an individual is based on simple population-driven inversions and recombinations of genes. This is a well established operator, know by its good performance with larger instances [22].

Algorithm 1. Inver-over

```
generate random population P;
while not satisfied termination-condition do
  for each individual $S_i \in P$ do
    $S^{'} \leftarrow S_i$
    $c \leftarrow$ select random gene $\in S^{'}$
    repeat
      $S^{''} \leftarrow$ select random individual $\in P$ that $S^{''} \neq S^{'}$
      $c^{'} \leftarrow$ gene after $c$ in $S^{''}$
      if $c^{'}$ is the next or previous gene to $c$ in $S^{'}$ then
        exit from repeat loop;
      end if
      if $c^{'} > c$ then
        $c_{temp} \leftarrow$ gene after $c^{'}$ in $S^{'}$
      else
        $c_{temp} \leftarrow$ gene before $c^{'}$ in $S^{'}$
      end if
      inverse the section from the gene after $c$ until $c^{'}$ in $S^{'}$
      $c \leftarrow c_{temp}$
    until
    if $eval(S^{'}) \leq eval(S_i)$ then
      $S_i \leftarrow S^{'}$
    end if
  end for
end while
```

In this work a different version on the algorithm is used, mixing the original implementation with the [9] implementation. This work allows array cyclic inversions to happen, inversions that include the section of the last to the first gene. However, the next gene pick depends of the previous order of the selected genes for inversion. This inversion process is analogous to a crossover operator.

Furthermore, it is stated that to avoid the local optima, a process analogous to the mutation operator has to be used to create new connections that do not exist in population [22]. However, these mutations are not greedy and usually delay the convergence process, so for this implementation it was removed. The implemented algorithm pseudo-code is sketched in Algorithm 1.

4.2 Stochastic 2-Opt

The 2-opt optimization [8] is a TSP local search algorithm that adjusts the routing sequence greedily with simple inversions. The main idea is to break the route in two paths and reconnect it invertedly, if it improves the fitness so the inversion is kept in the solution. Unfortunately, this is an exact algorithm with a complexity of $\mathcal{O}(n^2)$, so a lot of solutions evaluations need to be performed in order to improve the solution. Nonetheless, there are other methods to improve the 2-opt performance, such as search parallelism and the Lin and Kernighan technique [10]. The present work made use of a stochastic approach for the algorithm.

The stochastic implementation relies on the observations that even with a N^2 search space, the actual number of improvements performed by the algorithm is roughly N [10]. So, with a random exploration of the space, there could be more

Algorithm 2. Stochastic 2-opt

```
given an individual $S_i \in P$;
$S^{'} \leftarrow S_i$
$moves \leftarrow \{\}$
while not satisfied termination-condition do
    repeat
        $c \leftarrow$ select random gene $\in S^{'}$
        $c^{'} \leftarrow$ select random gene $\in S^{'}$
    until $(c, c^{'}) \notin moves$
    $moves \leftarrow moves + (c, c^{'})$
    if $c^{'} < c$ then
        $tmp \leftarrow c$
        $c \leftarrow c^{'}$
        $c^{'} \leftarrow tmp$
    end if
    $S^{''} \leftarrow$ inverse the section from the gene $c$ until $c^{'}$ of $S^{'}$
    if $eval(S^{''}) \leq eval(S^{'})$ then
        $S^{'} \leftarrow S^{''}$
    end if
end while
if $eval(S^{'}) \leq eval(S_i)$ then
    $S_i \leftarrow S^{'}$
end if
```

chances of finding a profitable move. In order to keep control of the algorithm run time, a termination condition is used to limit the exploration. Also, a set of explored moves prevents duplicated evaluations. The implemented algorithm pseudo-code is sketched in Algorithm 2.

4.3 Maximal Open Walk

The maximal open walk proposed by [9] as "City Selection", is a separated algorithm that searches for the contiguous part of a Hamiltonian path with the maximal cumulative value limited to some constraint. The path is the optimal solution found, while the value and constraint are respectively, the threat, given by the RCS area, and the total cost. In this work, another constraint is added to this problem, the duration of the open walk, calculated with Kepler's third law. The implemented algorithm pseudo-code is sketched in Algorithm 3.

5 Experimental Evaluation

The objective of these experiments is to evaluate the performance of the approach, understand the improvement gain by each technique and find some optimal parameters. To preserve the comparability, all the runs used 20000 fixed iterations as the termination-condition of the inver-over algorithm, a 100 individuals population, the original method runs used 0.05 as ri (mutation probability), the constrained runs were performed with a cost constraint of 1000 m/s and a time constraint of 1 year.

Algorithm 3. Maximal Open Walk

```
given an individual $S_i \in P$;
$arcs \leftarrow \{\}$
for each pair of contiguous genes $(g^{'}, g^{''}) \in S_i$ do
    $arcs \leftarrow arcs + (cost(g^{'}, g^{''}), duration(g^{'}, g^{''}))$
end for
$walks \leftarrow \{\}$
for each of arc $a^{'} \in arcs$ do
    $walk \leftarrow \{a^{'}\}$
    $cost \leftarrow a^{'}_{cost}$
    $duration \leftarrow a^{'}_{duration}$
    for each of arc $a^{''}$ after $a^{'} \in arcs$ do
        if $cost + a^{''}_{cost} > max_{cost}$ or $duration + a^{''}_{duration} > max_{duration}$ then
            exit from second for loop;
        else
            $walk \leftarrow walk + a^{''}$
            $cost \leftarrow cost + a^{''}_{cost}$
            $duration \leftarrow duration + a^{''}_{duration}$
        end if
    end for
    $genes \leftarrow$ the genes of the arcs in $walk$
    $reward \leftarrow threat(genes)$
    $walks \leftarrow walks + (genes, cost, duration, reward)$
end for
return $walk \in walks$ with the biggest $walk_{reward}$
```

The data about the debris were extracted from the Satellite Catalog (SATCAT), a catalogue of all the objects on the Earth orbit, maintained by the United States Space Command (USSPACECOM). The following instances of the problem were extracted: Iridium-33, Cosmos-2251 and Fengyun-1C, with respectively 331, 1048 and 2653 debris. Data was collected at respectively 11-Jun-2021 00:06 UTC, 13-Jun-2021 22:06 UTC and 13-Jun-2021 22:06 UTC.

To preserve the comparability of some results, back-propagated instances were generated inputting the actual instances in a SGPD4 orbital propagator [23] that backtracked the debris positions back to 01-Jan-2015 at 00:00 UTC, the same epoch of the instances used by [9]. Unfortunately this process is not very precise, though still feasible. All the debris in the clouds were considered, including the ones that will decay during the mission time.

For the sake of clearness, the *Time (min)* values on the experiments are concerned to the computation time taken for the run, and *Std. dev.* is the abbreviation for Standard Deviation. All experiments were conducted on a public online machine with a 2.30GHz CPU and 12.69 GB of RAM. Also, for all the experiments, the statistical data results out of 10 independent runs.

5.1 Back-Propagated Instances

The experiments performed with the back-propagate debris are intended to provide comparative results to the work of [9] and guide the definition of the parameters. The inver-over algorithm implemented in this work differs from the original

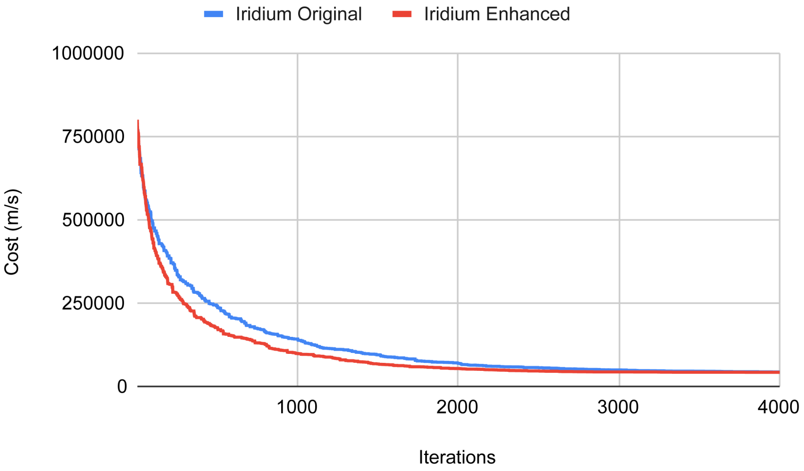

Fig. 1. Iridium-33 convergence results

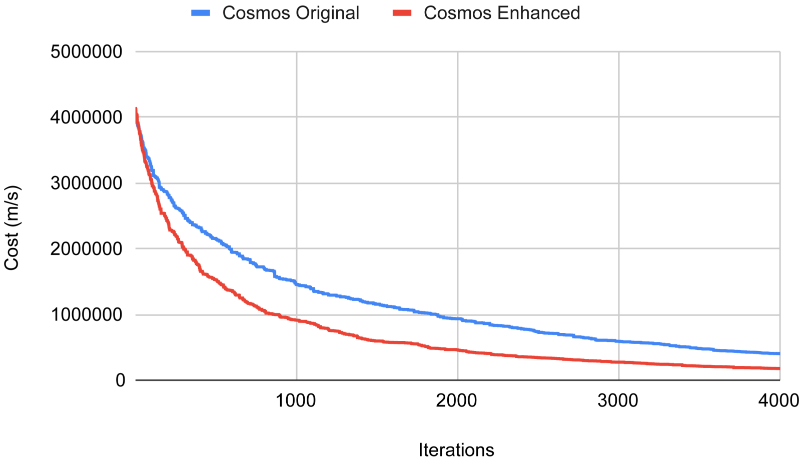

Fig. 2. Cosmos-2251 convergence results

implementation on two points, each point will be tested separately to show the advantages and justify its usage in this approach.

The changes made on the inversion implementation in this work aims to improve the convergence of the algorithm, to do so, this work approach is more population driven and less random mutated. To analyse the performance of the changes, the instances Iridium-33 and Cosmos-2251 were each run twice for 10 times, the first runs with the [9] implementation, and the other ones with this work implementation. The Figs. 1 and 2 demonstrate the improvement of the convergence, for a better visualization, just the first 4000 iterations were drawn.

Table 1. Inversion results

Instance	Method	Mean (min)	Std. dev. (min)
Iridium-33	Original	2.14	0.02
Iridium-33	Enhanced	**1.25**	0.02
Cosmos-2251	Original	12.09	0.07
Cosmos-2251	Enhanced	**7.30**	0.08

It is possible to notice a considerably change on the shape of the curve and the taken computation time in Table 1, making this works implementation convergence better. For the record, the majority of runs in each instance got the same final result, indicating that, for small and medium sized instances, the fast convergence does not deteriorate the result.

To deviate from the local optima, the stochastic 2-opt (S2opt) is used in this work. Parametric tests were conducted to analyse its performance when it matters to time and achieved result. Iridium-33 and Cosmos-2251 were submitted to 9 different runs, running 10 times each, with a different combination of two parameters: how often does the S2opt runs and with how many iterations at each time. All individuals of the population were processed at each S2opt iteration.

Since the search area of the S2opt is large, a range of possible attempts should be chosen, being neither too small, so no improvement move is found, or too big, so almost the whole search space is tested, turning it into a exact solution. In this work, the chosen range is from 10,000 to 1,000,000 attempts, while the parameters are equally spaced discrete values where its configuration do not underflow or overflow the range. The parameters per run are the following:

- Run 1: At each 100 main iterations, run S2opt with 100 iterations.
- Run 2: At each 100 main iterations, run S2opt with 500 iterations.
- Run 3: At each 100 main iterations, run S2opt with 1000 iterations.
- Run 4: At each 500 main iterations, run S2opt with 100 iterations.
- Run 5: At each 500 main iterations, run S2opt with 500 iterations.
- Run 6: At each 500 main iterations, run S2opt with 1000 iterations.
- Run 7: At each 1000 main iterations, run S2opt with 100 iterations.
- Run 8: At each 1000 main iterations, run S2opt with 500 iterations.
- Run 9: At each 1000 main iterations, run S2opt with 1000 iterations.

Analysing the results in Figs. 3 and 4 it is possible to state that due to the small size of the Iridium-33 debris, all the runs achieved the optimal solution. Also, the number of S2opt iterations is directly proportional to the computation time. And finally, the Runs 4, 7 an 8 have the better performances ratios, with low values of cost and time, among these, Run 7 is the best one.

Also, to preserve the idea of a competitive evolution, the following experiments use S2opt with an elitist strategy. This time, instead of running the S2opt for the whole population at each S2opt iteration, just the better individuals will be improved. Parametric tests were performed with 5 different sizes of elites, running 10 times each. To preserve the elitist characteristic, the elite group should not be greater than 30% of the population.

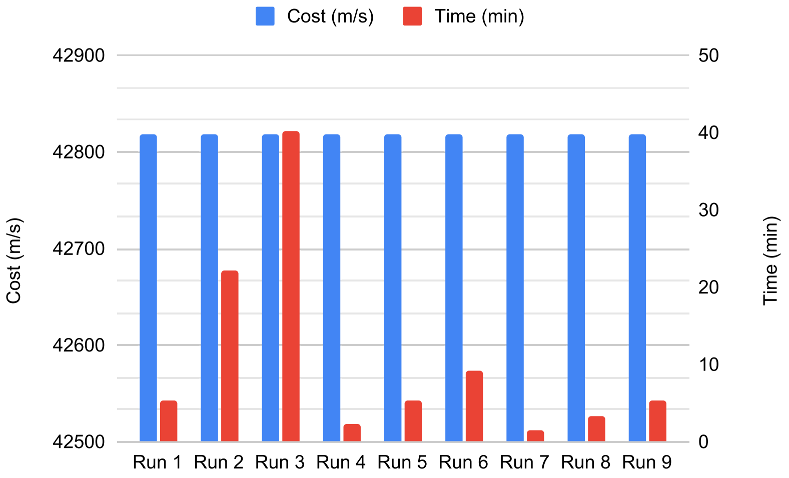

Fig. 3. Iridium-33 Stochastic 2-opt results

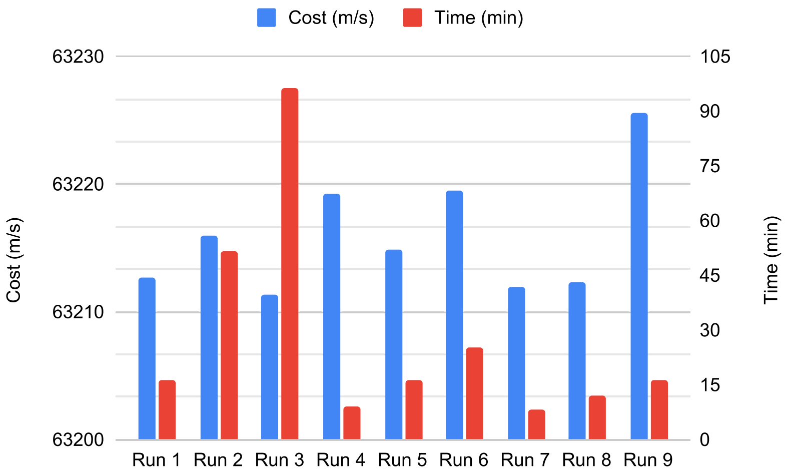

Fig. 4. Cosmos-2251 Stochastic 2-opt results

In these experiments, Iridium-33 will be discarded, since its small size does not help to fully analyse the performance of the algorithm. Here, the tests are set with the same parameters of the previous Run 7 (best run), at each 1000 main iterations, run S2opt with 100 iterations.

Analysing the results in Table 2 it is possible to state that the elitist improvement of the best 5 individuals at each iteration is the wise strategy to follow, having a lower cost with a little bit more computational time taken.

To emphasises the importance of each technique, ablation experiments were conducted for this solution. It is important to state that the Elitism is applied on the S2Opt, so there is no possible scenario using Elitism without S2Opt. Table 3,

Table 2. Elitist results

Run	Elite size	Cost (m/s)		Time (min)	
		Mean	Std. dev.	Mean	Std. dev.
Run 1	1	63,226	29.75	8.01	0.12
Run 2	5	**63,211**	0.10	7.54	0.05
Run 3	10	63,216	16.92	**7.52**	0.11
Run 4	20	63,217	22.01	8.02	0.13
Run 5	30	63,220	26.56	8.07	0.10

Table 3. Ablation results

Technique			Cost (m/s)		Time (min)	
Inver-over	S2Opt	Elitism	Mean	Std. dev.	Mean	Std. dev.
Y	Y	Y	**63,211**	0.10	**7.54**	0.05
Y	Y	N	63,211	2.14	8.27	0.31
Y	N	N	63,214	8.23	7.49	0.07
N	Y	Y	63,225	21.01	12.57	0.12
N	Y	N	63,252	36.87	13.34	0.14
N	N	N	63,256	42.70	12.50	0.08

summarizes the results of each combination. The experiments were performed on instance Cosmos-2251, with 10 runs each. The S2opt parameters are 100 runs at each 1000 main iterations, and the elitist parameter is 5.

With the ablation experiments, it is possible to understand how each technique of our approach affects the final results. It is clear that the conjunction of the techniques improves the found solutions.

5.2 Actual Instances

The experiments performed with the actual instances are intended to provide results for the present status of the debris clouds. The runs were performed using the best parameters found in the previous experiments. Also, for a comparative result, the original implementation was used with the actual instances, and its solutions, inputted to the enhanced maximal open walk of this work. For each instance were performed 10 runs, for the sake of clarity, the settings are: Enhanced inversion implementation, with S2opt running 100 iterations for the best 5 individuals at each 1000 main iterations, with a cost constraint of 1000 m/s and a time constraint of 1 year.

The results on both sections of Table 4 are from the same runs, the results at the bottom are given by the maximal open walk applied to the optimized path at the top. Being the missions objective: clear the maximum possible area are under the cost and time constraints [17], our approach focused on making an optimized use of the mission resources to outperforms other approaches. Cleaning more area, even if the mission duration and cost are bigger, bounded to time and cost constraints, of course.

Table 4. Final results

Instance	Method	Cost (m/s)		Duration (min)		Time (min)	
		Mean	Std. dev.	Mean	Std. dev.	Mean	Std. dev.
Iridium-33	Original	**55,788**	0.0	**16,345**	0.57	4.27	0.04
Iridium-33	Enhanced	55,810	12.69	16,346	0.80	**2.56**	0.03
Cosmos-2251	Original	186,896	4,512	52,069	1.23	46.52	1.29
Cosmos-2251	Enhanced	**156,067**	8,022	52,069	1.41	**31.00**	0.14
Fengyun-1C	Original	2,438,068	24,731	**134,904**	0.71	188.29	6.50
Fengyun-1C	Enhanced	**899,179**	11,468	134,905	0.82	**115.37**	5.24

Constrained results

Instance	Method	Cost (m/s)		Duration (min)		Area (m^2)	
		Mean	Std. dev.	Mean	Std. dev.	Mean	Std. dev.
Iridium-33	Original	971.16	26.17	34,722	3,785	3,984	196.0
Iridium-33	Enhanced	**952.42**	61.79	**34,720**	3,789	**3,993**	167.0
Cosmos-2251	Original	974.77	19.55	**26,675**	6,705	1,666	86.0
Cosmos-2251	Enhanced	**967.90**	19.62	33,163	5,009	**1,703**	41.0
Fengyun-1C	Original	**859.63**	270.0	**148.0**	84.52	671.4	45.4
Fengyun-1C	Enhanced	951.67	67.63	403.5	60.21	**905.5**	221.6

Summarizing, in the average of the runs, the enhanced approach decreased the cost by **26.51%** and the computation time by **35.56%**. Also, when constrained, the solutions of the enhanced approach produced paths that performs significantly better than the original approach solutions, cleaning **12.82%** more area under the same constraints. This is possible by a better usage of the constrained resources, increasing the cost by **5.28%** and the mission duration by **1.06%**. Finally, it is possible to notice that the most profitable mission is the enhanced Iridium-33, cleaning way more area with the same constraints.

6 Conclusions

The exploration of TSP approaches when dealing with space debris have considerably evolved the TDTSP problem modeling and its available solutions. However, it still lacks from approaches that fulfill the ADR mission requirements with a feasible performance and big instance sizes. This work proposed an enhanced method as a strong candidate to future approaches on TSP based ADR mission plannings. Using the inver-over as a fast convergence algorithm to deal with the greedy search throught fast inversions, the S2opt as a solution diversity creator to deviate the search from local optima, our method found optimized solutions in a feasible time out of large datasets. Real world instances were used to evaluate this approach performance and execute parametric tests, the retrieved results were considerably better than the original approach by [9].

However, this approach does not model the time dependence of the problem, meaning that the produced solutions may not be good solutions in a real scenario, since it does not consider the moving dynamics of the debris. Also, this approach first optimizes the cost of the solution, and later chooses the most rewardable sub-path, so it is not a fully bi-optimization algorithm. Meaning that solutions with a good cost versus reward ratio could be missed.

As future work, we would like to implement a time-dependent removal sequence to produce a complete solution for the ADR mission planning problem. We also plan to implement Lin and Kernighan's algorithm to improve the convergence of local search heuristic. The optimization of the transference cost with the consideration of body masses and J_2 effect on the transfers represents another interesting direction. Finally, we would like to improve the approximation of the mission duration with a more suitable equation for the Edelbaum (rather than Hohmann) transfer duration.

Acknowledgements. We thank the anonymous reviewers for their constructive feedback. This work was partially supported by FAPERGS (grant 19/2551-0001277-2).

References

1. Barea, A., Urrutxua, H., Cadarso, L.: Large-scale object selection and trajectory planning for multi-target space debris removal missions. Acta Astronautica **170**, 289–301 (2020). https://doi.org/10.1016/j.actaastro.2020.01.032
2. Braun, V., et al.: Active debris removal of multiple priority targets. Adv. Space Res. **51**(9), 1638–1648 (2013)
3. Cerf, M.: Multiple space debris collecting mission-debris selection and trajectory optimization. J. Optim. Theor. Appl. **156**, 761–796 (2013). https://doi.org/10.1007/s10957-012-0130-6
4. Cerf, M.: Multiple space debris collecting mission: optimal mission planning. J. Optim. Theor. Appl. **167**(1), 195–218 (2015). https://doi.org/10.1007/s10957-015-0705-0
5. Koblick, D., et al.: Low thrust minimum time orbit transfer nonlinear optimization using impulse discretization via the modified picard-chebyshev method. Comput. Model. Eng. Sci. **111**(1), 1–27 (2016). https://doi.org/10.3970/cmes.2016.111.001
6. Edelbaum, T.N.: Propulsion requirements for controllable satellites. ARS J. **31**(8), 1079–1089 (1961). https://doi.org/10.2514/8.5723
7. Federici, L., Zavoli, A., Colasurdo, G.: A time-dependent tsp formulation for the design of an active debris removal mission using simulated annealing (2019)
8. Flood, M.M.: The traveling-salesman problem. Oper. Res. **4**(1), 61–75 (1956). https://doi.org/10.1287/opre.4.1.61
9. Izzo, D., Getzner, I., Hennes, D., Simões, L.F.: Evolving solutions to tsp variants for active space debris removal. In: Proceedings of the 2015 Annual Conference on Genetic and Evolutionary Computation, pp. 1207–1214. GECCO '15, ACM (2015)
10. Johnson, D., McGeoch, L.A.: The traveling salesman problem: A case study in local optimization (2008)
11. Kanazaki, M., Yamada, Y., Nakamiya, M.: Trajectory optimization of a satellite for multiple active space debris removal based on a method for the traveling serviceman problem. In: 2017 21st Asia Pacific Symposium on Intelligent and Evolutionary Systems (IES), pp. 61–66 (2017). https://doi.org/10.1109/IESYS.2017.8233562

12. Kessler, D.J., Cour-Palais, B.G.: Collision frequency of artificial satellites: the creation of a debris belt. J. Geophys. Res. Space Phys. **83**(A6), 2637–2646 (1978). https://doi.org/10.1029/JA083iA06p02637
13. Li, H., Baoyin, H.: Optimization of multiple debris removal missions using an evolving elitist club algorithm. IEEE Trans. Aerospace Electron. Syst. **56**(1), 773–784 (2020). https://doi.org/10.1109/TAES.2019.2934373
14. Li, H.Y., Baoyin, H.X.: Sequence optimization for multiple asteroids rendezvous via cluster analysis and probability-based beam search. Sci. Chin. Technol. Sci. **64**(1), 122–130 (2020). https://doi.org/10.1007/s11431-020-1560-9
15. Liou, J.C.: An active debris removal parametric study for leo environment remediation. Adv. Space Res. **47**(11), 1865–1876 (2011). https://doi.org/10.1016/j.asr.2011.02.003
16. Liou, J.C., Johnson, N.: Instability of the present leo satellite populations. Adv. Space Res. **41**(7), 1046–1053 (2008). https://doi.org/10.1016/j.asr.2007.04.081
17. Liou, J.C., Johnson, N., Hill, N.: Controlling the growth of future leo debris populations with active debris removal. Acta Astronautica **66**(5), 648–653 (2010). https://doi.org/10.1016/j.actaastro.2009.08.005
18. Liu, Y., Yang, J., Wang, Y., Pan, Q., Yuan, J.: Multi-objective optimal preliminary planning of multi-debris active removal mission in LEO. Sci. Chin. Inform. Sci. **60**(7), 1–10 (2017). https://doi.org/10.1007/s11432-016-0566-7
19. Madakat, D., Morio, J., Vanderpooten, D.: Biobjective planning of an active debris removal mission. Acta Astronautica **84**, 182–188 (2013). https://doi.org/10.1016/j.actaastro.2012.10.038
20. Seumahu, E.S.: Exploration of the Equatorial LEO Orbit for Communication and Other Applications, pp. 217–228. Springer, Netherlands, Dordrecht (1996). https://doi.org/10.1007/978-94-011-5692-9_25
21. Stuart, J., Howell, K., Wilson, R.: Application of multi-agent coordination methods to the design of space debris mitigation tours. Adv. Space Res. **57**(8), 1680–1697 (2016). https://doi.org/10.1016/j.asr.2015.05.002
22. Tao, G., Michalewicz, Z.: Inver-over operator for the TSP. In: Eiben, A.E., Bäck, T., Schoenauer, M., Schwefel, H.-P. (eds.) PPSN 1998. LNCS, vol. 1498, pp. 803–812. Springer, Heidelberg (1998). https://doi.org/10.1007/BFb0056922
23. Vallado, D., Crawford, P.: SGP4 Orbit Determination, pp. 18–21 (2008). https://doi.org/10.2514/6.2008-6770
24. Wang, D., Li, L., Chen, L.: An efficient genetic algorithm for active space debris removal planning. In: 2019 IEEE Congress on Evolutionary Computation (CEC), pp. 514–521 (2019). https://doi.org/10.1109/CEC.2019.8790081
25. Yang, J., Hou, X., Hu, Y.H., Liu, Y., Pan, Q.: A reinforcement learning scheme for active multi-debris removal mission planning with modified upper confidence bound tree search. IEEE Access **8**, 108461–108473 (2020)
26. Yang, J., Hu, Y.H., Liu, Y., Pan, Q.: A maximal-reward preliminary planning for multi-debris active removal mission in leo with a greedy heuristic method. Acta Astronautica **149**, 123–142 (2018). https://doi.org/10.1016/j.actaastro.2018.05.040
27. Zhang, T., Shen, H., Li, H., Li, J.: Ant Colony Optimization based design of multiple-target active debris removal mission (2018). https://doi.org/10.2514/6.2018-2412