J Optim Theory Appl (2015) 167:195–218
DOI 10.1007/s10957-015-0705-0

Multiple Space Debris Collecting Mission: Optimal Mission Planning

Max Cerf

Received: 24 May 2014 / Accepted: 17 January 2015 / Published online: 31 January 2015

Abstract Successive missions must be planned to clean the near-Earth space from the heaviest debris. The problem mixes combinatorial optimization to select and order the debris, and continuous optimization to define the orbital maneuvers. In order to reduce the costs all missions have to be achieved by identical expendable vehicles with a minimum fuel requirement. The solution method proposed consists in three stages. Firstly the orbital transfer problem is solved for all pairs of debris and for discretized dates, considering a generic transfer strategy suited either to a high-thrust or to a low-thrust vehicle. The results are stored in cost matrices defining a response surface. Secondly a simulated annealing algorithm is applied to find the optimal mission planning. The cost is assessed by interpolation on the response surface. The convergence is quite fast, yielding an optimal mission planning. Thirdly the successive missions are re-optimized in terms of transfer maneuvers without changing the debris order. These continuous problems yield a refined performance requirement for designing the removal vehicle. The solution method is illustrated on a representative application case.

Keywords Space debris · Orbital transfer · Simulated annealing · Nonlinear programming

Mathematics Subject Classification 49N25 · 65K99 · 70M20 · 90C27 · 90C35 · 90C90 · 93C95

M. Cerf (✉)
AIRBUS Defence and Space, Les Mureaux, France
e-mail: max.cerf@astrium.eads.net

Abbreviations

LEO	Low earth orbit
SSO	Sun-synchronous orbit
RAAN	Right ascension of ascending node
SDC	Space debris collecting
TSP	Travelling salesman problem
RSM	Response surface modeling
SA	Simulated annealing
NLP	Nonlinear programming

1 Introduction

1.1 Space Debris

The near-Earth space is crowded by debris of all sizes. These debris originate from the fragmentation or corrosion of the old spacecrafts (spent satellites and launcher upper stages) released on orbit at the end of their operational life since the 1960. An efficient way to limit their proliferation is to remove the spent observation satellites evolving on near-circular polar orbits in the altitude range 700–900 km altitude. Several studies recommend a removal rate of five heavy debris per year in order to stabilize the debris population [1–4].

Dedicated vehicles must be designed for such removal missions. This paper addresses the problem of planning the successive missions so that they can be achieved at minimal cost by identical expendable vehicles.

1.2 Problem Statement

The cleaning program consists in launching a series of dedicated vehicles, each one being in charge of removing several heavy debris (typically five debris per vehicle). With the current state of the art, a reusable concept cannot be envisioned. A series of expendable vehicles is necessary in order to progressively clean the near-Earth space from the most dangerous debris.

In order to limit the development cost the vehicles used for the successive missions should be identical. The profile of a single space debris collecting (SDC) mission is defined by the selected debris and the successive transfer maneuvers to travel from a debris to another.

The mission duration (typically 1 year) includes the transfers between the successive debris and the operations applied to each of them. The mission cost is driven at the first order by the SDC vehicle initial mass. This gross mass comprises the fuel required by the powered maneuvers (orbital transfers, deorbitation if performed by the vehicle itself) and the masses of the sub-systems used for the debris processing (rendezvous, capture, and deorbitation if performed by an autonomous kit supplied to the debris).

The design of the SDC vehicle is based on the most expensive mission: This ensures that the same design is compliant of all the successive missions planned. The SDC problem formulates as a complex variant of the travelling salesman problem (TSP) mixing discrete and continuous variables and with a time-dependent cost.

1.3 Method Overview

Finding the exact solution of a combinatorial problem requires an enumeration algorithm, either explicit or implicit. Such enumeration algorithms are applicable only to limited size problems. For large instances only approximate solutions can be hoped in a fixed computation time, using either heuristics approaches or stochastic algorithms [5–7]. Various approaches have been proposed for the debris removal problem depending on the number of debris and missions, on the orbital transfer scenario and on the mission duration.

- In [8] an exact solution is sought by implicit enumeration (branch and bound). The transfer scenario considers a drift orbit in order to comply with a bounded mission duration. The method allows finding the optimal compromise between fuel and duration, but it is restricted to a single mission and it can be applied only to a small number of debris.
- In [9] an exact solution is sought by explicit enumeration (brute force approach) without duration constraint. This allows simplifying the orbital transfer scenario which consists in waiting for the RAAN alignment before starting the rendezvous maneuvers. This strategy simplifies the transfer cost assessment, so that the method can be applied to several missions and a larger number of debris, but the mission duration is not bounded.
- In [10] an approximate solution is sought by a heuristic approach (Series Method) previously applied to near-Earth asteroids tour problems in the solar system. A co-elliptic transfer scenario is considered to minimize the transfer cost, but the mission duration is not bounded. The authors indicate that a future scenario improvement will consist in using the natural drift due to the first zonal term in order to control the overall duration. The method can be applied to several missions and large number of debris.
- In [11] an exact solution is sought by explicit enumeration for a set of five preselected debris. The mission scenario considers successive deorbit and transfer phases using low-thrust propulsion and locally optimal steering laws along each thrust arc. The first zonal term is not taken into account, and the plane changes are achieved through a constant out-of-plane thrust component. For each possible sequence of debris a bi-objective optimization yields the Pareto front in terms of duration and consumption. Large computation loads are required due to the exhaustive exploration of sequences.
- In [12] an approximate solution is sought by a branch and prune algorithm. The transfer scenario is based on an impulsive solution of the Lambert problem taking into account the first zonal term. A brute force approach is applied to assess all the transfers between debris pairs, sweeping on the dates and numbers of revolutions. The tree of the possible scenario is then pruned by applying thresholds on the mission cost and the mission duration with a penalization on the debris features (mass, area).

The method yields an impulsive mission scenario that is finally refined considering a low-thrust vehicle.
- In [13] an approximate solution is sought by an ant colony optimization algorithm. Each ant represents a single mission visiting several debris. An auction and bidding method allows debris exchanges and coordination between the ants. The transfer scenario assumes impulsive maneuvers at the debris orbit intersections without first zonal term effect. The method is well suited to several missions and large number of debris, but it is restricted to short mission durations and small RAAN changes.

1.4 Solution Method Proposed

Only approximate solution can be envisioned due to the complexity of the global mission planning problem. Among the existing algorithms, simulated annealing (SA) has proved quite successful on large TSP instances. We have therefore selected a simulated annealing approach to tackle the SDC problem. Compared to the TSP, the SDC problem presents additional issues due to the edge valuations and their time dependency. Indeed a SA algorithm tries millions of solutions before achieving a satisfactory convergence. Each trial solution is defined by a debris order and the visiting dates. Assessing the exact cost of a trial solution requires solving a series of hard optimal control problems to find the minimum fuel trajectories between the successive debris. In order to apply a SA to the SDC problem with reasonable computation times, it is not possible to solve "on-line" these optimal control problems for each trial solution. An instantaneous cost function must be devised. In order to get a sufficient confidence in the SA results, this cost function must be both robust (i.e., yield a cost value whatever the input data) and reliable (i.e., yield a cost value representative of a real optimized transfer).

The approach proposed consists in using a response surface modeling (RSM) based on cost matrices. More precisely, the optimization process is split into three successive stages.

- The first stage consists in building cost matrices. These cost matrices store the costs of all the possible elementary transfers between debris for a mesh of discretized dates. They result from a series of nonlinear optimizations (NLP) based on a simplified generic transfer strategy adapted to the mission specificities and to the vehicle propulsion system (high thrust or low thrust).
- The second stage consists in finding the optimal mission planning with a SA algorithm. The algorithm is derived from the one applied to a classical TSP with additional variables (rendezvous dates) and a RSM-based cost function. The cost of a trial solution is assessed by interpolation in the cost matrices spanning the possible transfers and dates. The solution defines the optimal mission planning.
- The third stage consists in a refined trajectory optimization. Indeed the RSM yields an approximate cost value by interpolation in the cost matrices. The refined optimization consists in fixing the debris order for the successive missions as given by the simulated annealing, and optimizing the rendezvous dates and the maneuvers using a real trajectory simulation. This is a standard nonlinear programming

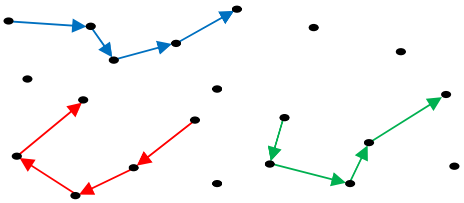

Fig. 1 Illustration of the SDC problem

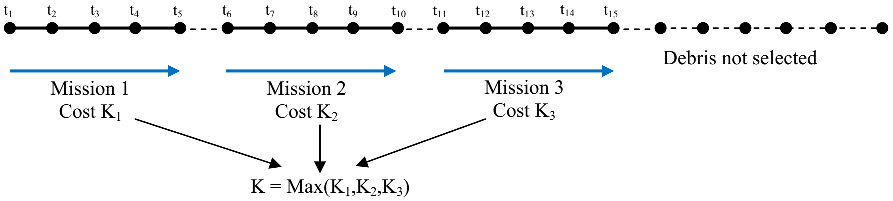

Fig. 2 SDC cost function

problem with continuous variables. The solution yields the performance requirement for the vehicle design.

2 Problem Formulation

The goal is to design a minimal mass vehicle compliant of a series of successive removal missions. The missions must be planned so as to minimize the fuel required by the most expensive mission, while achieving a mean removal rate of five debris per year. The problem formulates as a path problem (debris selection and ordering) with embedded transfer problems (maneuvers to go from a debris to another).

2.1 Path Problem

The problem features are first illustrated on an instance corresponding to the application case of Sect. 4. The path problem is then formulated in the general case.

2.1.1 Problem Features

The Fig. 1 illustrates an instance of the SDC problem with 21 candidate debris (represented by black points). The goal is to define three missions (represented respectively by blue, green and red arrows) removing each one 5 debris. Only 15 debris out of 21 will be visited, leaving 6 debris on orbit. In terms of graph vocabulary, the debris are nodes and each mission is an opened sub-path of 4 edges representing the orbital transfers.

The cost evaluation is illustrated on the Fig. 2. The 15 selected debris are visited at the respective increasing dates $t_1 < t_2 < \cdots < t_{15}$. The respective costs of the three missions are K_1, K_2 and K_3. The cost K of the cleaning program is the cost of the

Table 1 TSP versus SDC

	TSP	SDC
Number of nodes visited	N	$m \times n \leq N$
Path definition	Single closed path	Several opened sub-paths
Node positions	Fixed	Moving
Edge valuations	Fixed length	Time-dependent
Cost function	Path length	Maximum sub-path cost
Time constraint	None	Upper bound

most expensive mission. The total duration of the cleaning program ($t_{15} - t_1$) must be lower than 3 years in order to achieve the targeted removal rate (5 debris per year).

The SDC problem is a variant of the TSP with the following differences:

- The debris orbits have different precession rates, so that their relative configuration evolves with the time. The cost of going from the debris j to the debris k depends on the starting date t_j and the arrival date t_k, making thus the SDC problem time dependent.
- Instead of a single closed path visiting all the nodes, the debris are gathered in several sub-paths (missions). Not all debris are visited, and the cost is measured from the most expensive sub-path.
- There is a global time constraint due to the targeted removal rate of five debris per year.

The differences between the SDC problem and the TSP are summarized in the Table 1.

2.1.2 Path Problem Formulation

We denote N the total number of debris, n the number of debris to visit per mission and m the number of missions planned. The number of debris visited is $m \times n$ out of N candidates. The $m \times n$ selected debris are visited at the successive dates $t_1, t_2, \ldots, t_{m \times n}$. The debris order and the rendezvous dates have to be optimized.

A path is composed of the N candidate debris in a given order and visited at given dates. The path is subdivided into m successive sub-paths of n debris each. For the debris number $j (1 \leq j \leq N)$, we denote:

- d_j the position of the debris j on the path. The set $(d_1, d_2, \ldots, d_N)$ is a permutation of $(1, 2, \ldots, N)$.
- t_j the rendezvous date with the debris j. The dates are strictly increasing along the path: $t_1 < t_2 < \cdots < t_N$.
- C_j the cost of the transfer from the debris d_j at the date t_j to the next debris d_{j+1} at the date t_{j+1}.
- The cost C_j is a function of $(t_j, t_{j+1}, d_j, d_{j+1})$. It represents the fuel mass required by the orbital transfer.

The mission number $i (1 \leq i \leq N)$ is the sub-path dealing with the n successive debris of the path starting from the debris at the position $(i - 1) \times n + 1$ to the debris at the position $(i-1)\times n+n$. We denote K_i the cost of the mission number i and K the cost of the most expensive mission which is to be minimized.

$$\min_{d_j,t_j} K \text{ with } \begin{cases} K = \max\limits_{1\leq i\leq m} (K_i) \\ K_i = \sum_{j=1}^{n} C_{(i-1)n+j} \end{cases}$$

The cost function K accounts for the debris from the position 1 to the position $m\times n$. The remaining debris from $m\times n+1$ to N are left on orbit. The goal is to choose the debris order on the path (d_j) and the rendezvous dates (t_j) in order to minimize the cost of the most expensive mission.

A major issue lies in the valuation of the edges representing the orbital transfers between successive debris on the path. The valuation C_j of the jth edge is the propellant required to perform the orbital transfer. The formulation of the transfer problem is presented in the next section.

2.2 Transfer Problem

Finding the minimal fuel trajectory from a debris to another is a difficult optimal control problem in the general case. This transfer problem is simplified by considering a generic transfer strategy adapted to the mission specificities and to the vehicle propulsion system. The optimal control problem reduces thus to a nonlinear programming problem with two variables and one constraint that can be solved in an efficient manner.

2.2.1 Debris Orbits

Most low Earth orbit (LEO) debris move on near-circular orbits. At a given date t_0 a circular orbit is defined by its radius or semi-major axis $a(t_0)$ and two angles $I(t_0)$ and $\Omega(t_0)$ orientating the orbital plane in the Earth inertial reference frame (Fig. 3). The inclination I is the angle of the orbital plane with the Earth Equator. The line of nodes is the intersection of the orbital plane with the Equator. The RAAN Ω is the angle between the X axis of the Earth reference frame and the direction of the ascending node (node crossed with a northwards motion).

The orbital parameters are constant in the Keplerian model. The main perturbation to this model comes from the Earth flattening. The Earth equatorial bulge adds a perturbing force (J_2 zonal term) on the motion, causing a precession of the orbital plane. The RAAN precession rate [14] depends on the orbit radius a and inclination I.

$$\dot{\Omega} = -\frac{3}{2} J_2 \sqrt{\mu} R_T^2 a^{-\frac{7}{2}} \cos I$$

with $R_T = 6378137\,\text{m}$ (Earth equatorial radius), $\mu = 3.986005.10^{14}\,\text{m}^3/\text{s}^2$ (Earth gravitational constant), $J_2 = 1.08266$ (first zonal term)

The J_2 perturbation causes no secular change on the semi-major axis, the eccentricity and the inclination. The precession rate $\dot{\Omega}$ is therefore constant, and the RAAN evolves linearly with the time: $\Omega(t) = \Omega(t_0) + \dot{\Omega}(t-t_0)$

Fig. 3 Orbital parameters

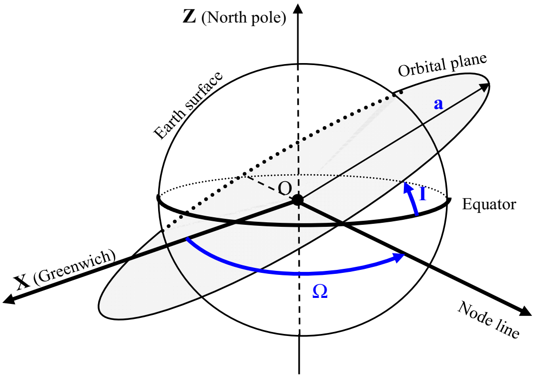

Table 2 Successive orbits during the transfer

	Propelled transfer 1	Drift phase	Propelled transfer 2	
Orbit	Debris 1	Drift start	Drift finish	Debris 2
Date	t_1	t_{d1}	t_{d2}	t_2
Radius	a_1	a_d	a_d	a_2
Inclination	I_1	I_d	I_d	I_2
RAAN debris 1	$\Omega_1(t_1)$	$\Omega_1(t_{d1})$	$\Omega_1(t_{d2})$	$\Omega_1(t_2)$
RAAN debris 2	$\Omega_2(t_1)$	$\Omega_2(t_{d1})$	$\Omega_2(t_{d2})$	$\Omega_2(t_2)$
RAAN vehicle	$\Omega_v(t_1) = \Omega_1(t_1)$	$\Omega_v(t_{d1})$	$\Omega_v(t_{d2})$	$\Omega_v(t_2) = \Omega_2(t_2)$

For a sun-synchronous orbit (SSO), the precession rate matches the Sun direction motion (0.986 deg/day). This property favors the Earth observation since a given latitude is always flown over at the same local solar time. Most debris stemming from spent observation satellites are consequently on nearly sun-synchronous orbits.

2.2.2 Transfer Strategy

The simplifications of the transfer problem are based on the mission specificities:

- The orbits of the targeted debris (old observation satellites) are assumed to be circular. The real orbits of such debris have indeed negligible eccentricities ($e < 0.01$).
- The mean removal rate (five debris per year) leaves enough time to use the J_2 nodal precession in order to perform the RAAN change at null fuel consumption.

The generic transfer strategy consists in bringing the vehicle on a circular drift orbit and wait until the RAAN change is completed. The transfer is split into a first propelled transfer from the debris 1 orbit to the drift orbit, a waiting duration on the drift orbit, and a second propelled transfer from the drift orbit to the debris 2 orbit. The transfer starts at a given date t_1 and ends at a given date t_2. The orbital parameters of the successive orbits are denoted in the Table 2.

The rendezvous in anomaly with the debris 2 is neglected in terms of both duration and consumption compared to the overall transfer. This generic transfer strategy using the J_2 precession to control the RAAN at null fuel consumption is near-optimal as long as a sufficient duration ($t_2 - t_1$) is allocated. For short durations, this strategy would no longer be possible and the RAAN change should be realized by propelled maneuvers at the expense of a larger fuel consumption. Two modelings of the propelled transfers are considered depending whether the SDC vehicle uses a high-thrust (Sect. 2.2.3) or a low-thrust propulsion system (Sect. 2.2.4).

Remark 2.1 It can be more economical to complete the transfer at a date prior to t_2 when the initial orbit precession rate is sufficient to naturally compensate the RAAN difference between the debris 1 and 2 within the allocated duration. In such a case the drift orbit is useless. The vehicle waits on the initial orbit (debris 1), and the transfer toward the debris 2 takes place when the RAAN difference is nullified. The final date is then lower than t_2. This case is accounted in the transfer strategy by allowing a drift orbit identical to the debris 1 orbit and by adding an optional waiting phase on the debris 2 orbit until reaching the fixed final date t_2.

2.2.3 High Thrust Propulsion

In the case of a high-thrust engine the propelled orbital transfers are modeled as Hohmann transfers [14,15] with impulsive maneuvers. Each transfer (from debris 1 to drift, then from drift to debris 2) is achieved by two impulses with split inclination change. The inclination of the intermediate elliptical orbit is computed using a near-optimal approximation derived by Lisowski [15]. The approximation consists in minimizing the sum of the squared velocity impulses (instead of the impulses norm). An analytical solution can thus be found with a limited deviation from the true minimum. The transfer strategy is depicted on the Fig. 4, with the velocity impulses associated to the Hohmann transfers.

The Hohmann transfer durations (about 1h) are negligible wrt the drift duration (several days or weeks). The RAAN precession due to the J_2 may be neglected during these transfers:

$$\begin{aligned} t_{d1} \approx t_1 &\Rightarrow \Omega_v(t_{d1}) \approx \Omega_v(t_1) \\ t_2 \approx t_{d2} &\Rightarrow \Omega_v(t_2) \approx \Omega_v(t_{d2}) \end{aligned}$$

The transfer total cost is the sum of the four velocity impulses. It does not depend on the vehicle thrust level.

2.2.4 Low-Thrust Propulsion

In the case of a low-thrust engine the propelled orbital transfers are modeled as Edelbaum transfers with continuous thrusting. Each transfer (from debris 1 to drift, then from drift to debris 2) is achieved in minimum time with continuous inclination change [14,16,17]. The Edelbaum model assumes a constant acceleration level. In order to get a refined assessment of the transfer duration and cost, the solution is computed in two stages: first with the initial acceleration level and then with the average acceleration level estimated from the first solution. The transfer strategy is depicted on the Fig. 5 with the spiraling trajectories associated to the Edelbaum transfers.

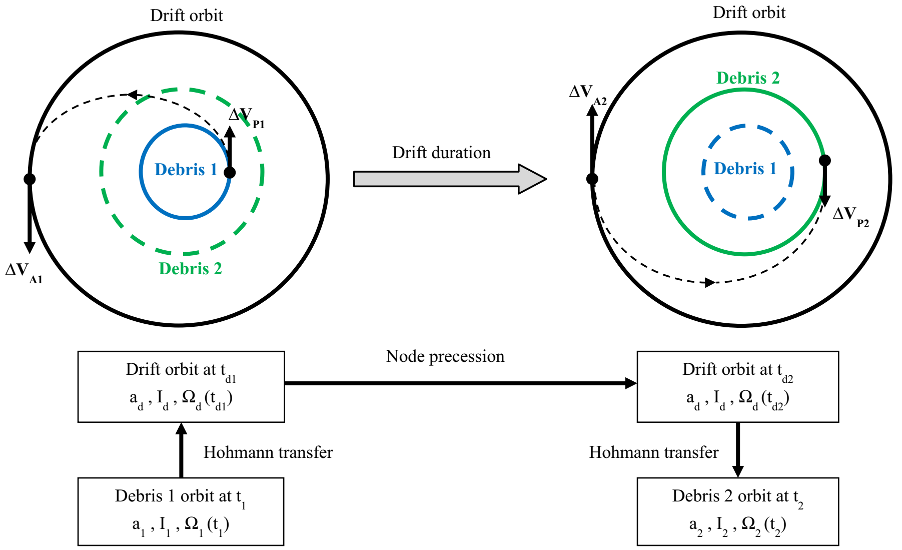

Fig. 4 High-thrust transfer strategy

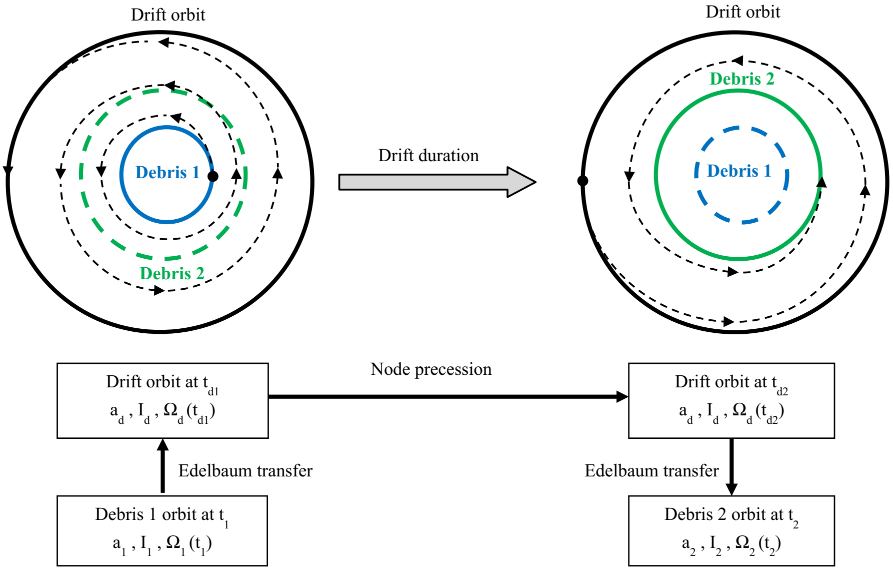

Fig. 5 Low-thrust transfer strategy

Opposite to the high-thrust case, the durations of the Edelbaum transfers are no longer negligible wrt the drift duration and they induce significant RAAN changes that are accounted as follows.

The Edelbaum solution yields the minimal time transfer between mutually inclined circular orbits, assuming a constant acceleration level. The Edelbaum model is based on an averaging of the dynamic equations assuming that the orbit remains circular throughout the transfer. During each revolution, the thrust direction keeps a constant angle with the orbital plane, with a sign switch at the antinodes. This averaged control law does not modify directly the RAAN. The RAAN evolution is only due to the J_2 perturbation, which acts constantly throughout the transfer phases. Knowing the evolution of the mean orbit radius $a(t)$ and inclination $I(t)$ throughout the Edelbaum transfer, the mean RAAN precession rate can be computed:

$$\dot{\Omega}(t) = -\frac{3}{2} J_2 \sqrt{\mu} R_T^2 a(t)^{-\frac{7}{2}} \cos I(t)$$

The RAAN variation during the propelled transfer is then assessed by a numerical integration along the Edelbaum trajectory from the initial date t_1 to the final date t_2.

$$\begin{array}{ll} \Omega_v(t_{d1}) = \Omega_v(t_1) + \int_{t_1}^{t_{d1}} \dot{\Omega}_v(t)\mathrm{d}t & \text{(during the first propelled transfer)} \\ \Omega_v(t_2) = \Omega_v(t_{d2}) + \int_{t_{d2}}^{t_2} \dot{\Omega}_v(t)\mathrm{d}t & \text{(during the second propelled transfer)} \end{array}$$

The velocity impulse associated to the Edelbaum solution is obtained as the product of the mean acceleration level (denoted f) by the transfer duration $(t_2 - t_1) : \Delta V = f(t_2 - t_1)$. The transfer total cost is measured by summing the velocity impulses of the two propelled transfers (from debris 1 to drift, then from drift to debris 2). Opposite to the high-thrust case, the cost depends on the vehicle thrust level.

Remark 2.2 The transfer strategy based on Edelbaum transfers is not globally optimal. Indeed, the Edelbaum solution yields the minimal time transfer without taking into account the RAAN change. The RAAN change is assessed a posteriori along the Edelbaum trajectory. The drift orbit parameters a_d and I_d must then yield the adequate precession $\dot{\Omega}_d$ rate to achieve the required RAAN final value.

This may lead to a more costly drift orbit regarding the velocity impulses. Cheaper solutions could be found by performing a part of the RAAN change during the propelled transfers. The possible cost gain may be significant depending on the relative durations of the propelled phases wrt the drift phase. Variants of the Edelbaum solution have been derived considering alternative constraints [17] (RAAN change instead of inclination change, altitude bound). For the SDC problem, an analytical solution taking into account the three transfer phases (propelled—drift—propelled) has been investigated and it will be presented in a separate paper.

It is assumed for the SDC problem that a sufficient acceleration level is available on the vehicle, so that the propelled durations remain small wrt the drift duration. With this assumption the Edelbaum transfer strategy can be considered as nearly optimal.

2.2.5 Transfer Problem Formulation

The fuel consumed by the transfer is linked to the velocity impulse by the rocket equation [14,15]:

$$\Delta V_{12} = v_e \ln \frac{M_1}{M_2} \Leftrightarrow m_c = M_1 - M_2 = M_1 \left(1 - e^{-\frac{\Delta V_{12}}{v_e}}\right)$$

with v_e: engine exhaust velocity. M_1: initial vehicle mass (date t_1), M_2: final vehicle mass (date t_2).

For a given initial mass M_1, minimizing the fuel consumption is equivalent to minimizing the velocity impulse. The velocity impulse is preferred as cost function rather than the mass consumed. It is indeed an intrinsic cost measure independent on the transfers previously realized by the vehicle, opposite to the mass consumption which depends on the vehicle gross mass at the transfer beginning. It is therefore more suited to the optimization of the successive transfers required by the SDC mission (see Sect. 3.3 for the mission cost assessment).

The transfer starts at the fixed date t_1 and finishes at the fixed date t_2. The variables are the drift orbit radius a_d and inclination I_d. The initial vehicle RAAN is $\Omega_v(t_1) = \Omega_1(t_1)$, and the final RAAN value must coincide with the RAAN of the debris 2: $\Omega_v(t_2) = \Omega_2(t_2)$.

The problem formulates as a reduced nonlinear programming (NLP) problem with 2 variables and 1 constraint:

$$\min_{a_d, i_d} \Delta V_{12} \text{ s.t. } \Omega_v(t_2) = \Omega_2(t_2)$$

3 Solution Method

The global problem consists in a series of continuous problems (transfer trajectories between debris) embedded within a combinatorial problem (path between the selected debris). It mixes integer variables (debris selection and order) and real variables (rendezvous dates, drift orbit parameters). Even taken separately these sub-problems are intrinsically hard. The global problem cannot be solved in a direct manner. The approach proposed consists applying a SA algorithm to the path problem, using a RSM to assess the transfer costs.

3.1 Simulated Annealing

Simulated annealing is a stochastic optimization algorithm inspired by the metallurgic process of annealing [6,7]. It has been applied successfully to combinatorial problems with a large number of local minima, and particularly to the TSP. Starting from an initial solution it allows escaping local minima by accepting random uphill moves with a varying probability level and exploring widely the cost function landscape. The probability level is progressively decreased through a "temperature" parameter T. When the temperature is progressively lowered, the solution freezes on the best minimum found. The main settings of the algorithm are the initial temperature, the decrease rate and the definition of the random perturbations (or moves) applied to the current solution.

Four elementary moves are implemented for the SDC problem: insertion, swap, permutation and date shift (Fig. 6a–d). The insertion, swap and permutation modify the debris order on the path similar to the classical TSP [7]. The date shift changes the date of a node while keeping the path order. The new date remains comprised between the previous and next node date.

A single evaluation (or try) of the SA algorithm consists in:

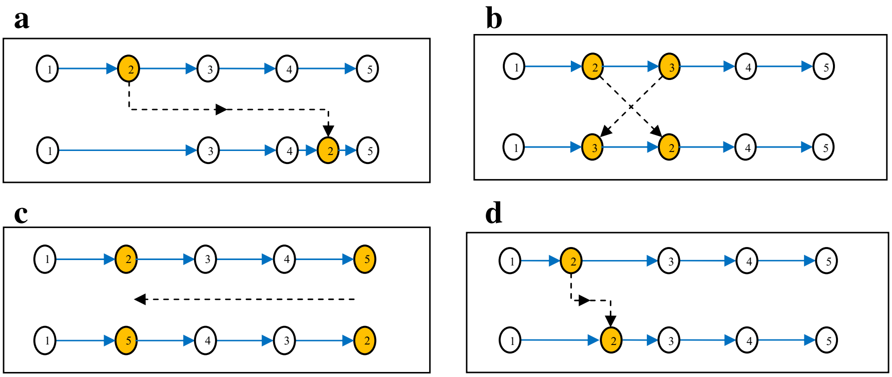

Fig. 6 **a** Insertion (2 is inserted after 4). **b** Swap (2 and 3 are exchanged). **c** Permutation (the leg from 2 to 5 is reversed). **d** Date shift (the date 2 is shifted between date 1 and date 3)

- Selecting randomly one of the three elementary path moves (insertion, swap, permutation)
- Selecting randomly the nodes where the move is applied
- Performing the move to get the trial path
- Selecting randomly a node on the path
- Shifting randomly the node date between the previous and the next node date
- Assessing the cost of the trial solution
- Accepting the try with the probability level defined by the current temperature

An iteration of the SA algorithm consists in decreasing the temperature with a fixed rate α after a given number of tries. A typical decrease rate value is $\alpha = 0.999$ every 1,000 tries (this depends on the problem size).

The algorithm is initiated either with a random solution or with a greedy solution. For example, the initial solution can be built by the best insertion method: The nodes are inserted successively in the path at the position minimizing the cost. The initial temperature is set in order to accept a random perturbation of the initial solution with a 90 % probability. This allows large solution changes during the first iterations. When no progress is made, a local search is performed by trying systematically all the elementary perturbations on the last solution. If this search is successful, the iterations are restarted from the improved solution, else the algorithm is stopped.

Millions of trials are necessary to achieve a satisfactory convergence of the SA algorithm. Assessing the exact cost of a trial solution requires solving a series of transfer problems. Although the NLP transfer problem can be solved in a robust and efficient manner by a nonlinear optimizer, the computation still requires a few seconds discarding an on-line optimization. In order to get reasonable computation times, the costs are assessed using a RSM based on cost matrices.

3.2 Response Surface Modeling

The cost of an elementary transfer from any debris to any other depends on the starting date and on the transfer duration. For a given starting date τ and a given transfer

Date τ / Duration $\Delta\tau$		Debris k $(1 \leq k \leq N)$	
Debris j $(1 \leq j \leq N)$		Cost $C(\tau,\Delta\tau,j,k)$	

Fig. 7 Cost matrix for the date τ and the duration $\Delta\tau$

duration $\Delta\tau$, the transfer costs between all the pairs of the N candidate debris are assessed. This requires $N \times (N-1)$ elementary optimizations whose results are stored into a $N \times N$ cost matrix represented on the Fig. 7.

The value $C(\tau, \Delta\tau, j, k)$ stored at the row j and column $k \neq j$ of the matrix is thus the minimum velocity impulse to go from the debris j at the date τ to the debris k at the date $\tau + \Delta\tau$. The matrix diagonal is unfilled at this stage. It will be used later to account for the cost of the debris operations. In order to account for the time dependency of the SDC problem, a series of cost matrices are assessed for a mesh of discretized starting dates and transfer durations covering the time span of the cleaning program.

We denote:

T_0	the date of the beginning of the cleaning program	
ΔT	the total duration of the cleaning program	
n_t	the number of discretized starting dates	
n_d	the number of discretized transfer durations	
τ_i	the starting date number i in the grid	$(1 \leq i \leq n_t)$
$\Delta\tau_d$	the transfer duration number d in the grid	$(1 \leq d \leq n_d)$

For any starting date τ_i, any duration $\Delta\tau_d$ and any pair of debris j and $k \neq j, C(i, d, j, k)$ is the cost of the transfer going from the debris j at the date τ_i to the debris k at the date $\tau_i + \Delta\tau_d$. The sub-matrix $C(i, d, 1{:}N, 1{:}N)$ of size $N \times N$ contains the costs of all the elementary transfers starting at the date τ_i with a duration $\Delta\tau_d$. $N \times (N-1)$ optimizations are required for solving the associated transfer problems. Some transfers may be unfeasible in the prescribed duration due to bounds on the drift orbit parameters (minimal altitude) that limit the available precession rate. In such cases, the corresponding matrix element is set to an arbitrarily large value, so that it will not be selected during the path optimization. The total number of $N \times N$ sub-matrices is $n_t \times n_d$, corresponding to the mesh of n_t dates and n_d durations. Some of these sub-matrices are useless when they correspond to a final date $(\tau_i + \Delta\tau_d)$ beyond the ending date of the cleaning program $(T_0 + \Delta T)$. These useless matrices are not computed and filled with large values indicating the transfer unfeasibility.

The Fig. 8 illustrates the mesh of $n_t \times n_d$ cost sub-matrices spanning the dates of the cleaning program.

A total of $n_t \times n_d \times N \times (N-1)$ optimizations is run to fill the mesh of cost matrices. This mesh is then used within the simulated annealing algorithm to assess the cost of a trial solution through a RSM as follows.

A trial solution is a single path visiting the N candidates debris as for the TSP. It is defined by the successive debris numbers $(d_1, d_2, \ldots, d_N)$ and the successive rendezvous dates $(t_1 < t_2 < \cdots < t_N)$. The pth edge on the trial path goes from the debris d_p at the date t_p to the debris d_{p+1} at the date t_{p+1}. The corresponding transfer

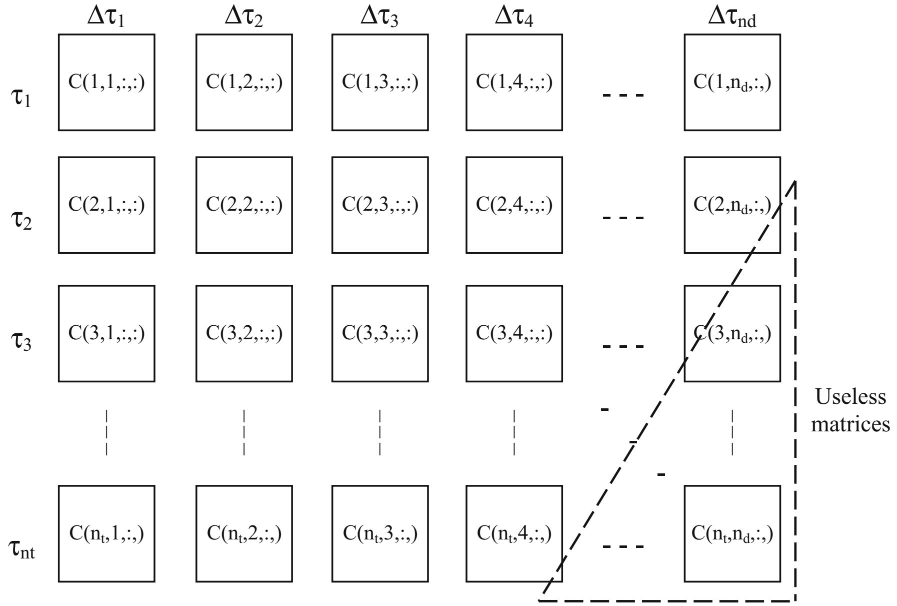

Fig. 8 Mesh of discretized cost matrices

duration is denoted $\Delta t_p = t_{p+1} - t_p$. The RSM consists in a bilinear interpolation on the mesh of cost matrices $C(i, d, j, k)$ with the following steps:

- Locating the starting date interval (index i): $\tau_i \leq t_p < \tau_{i+1}$
- Locating the duration interval (index d): $\Delta\tau_d \leq \Delta t_p < \Delta\tau_{d+1}$
- Selecting the matrices elements at the row d_p (starting debris) and the column d_{p+1} (arrival debris)
- Performing a bilinear interpolation on the intervals $[\tau_i; \tau_{i+1}]$ and $[\Delta\tau_d; \Delta\tau_{d+1}]$

This yields the interpolated cost $C_{\text{int}}(t_p, \Delta t_p, d_p, d_{p+1})$ of the pth edge $(1 \leq p \leq n \times m)$. The path is then sub-divided into m sub-paths of n nodes each, representing the successive missions. The cost K_i of the ith sub-path is the sum of its edge costs, and the cost of the trial solution is the cost of the most expensive sub-path: $K = \text{Max}(K_1, \ldots K_m)$

3.3 Mission Cost

The actual cost function for the SDC problem is the fuel consumption per mission, which is the driver for the vehicle design. This consumption depends on the vehicle mass through the rocket equation. It is not an adequate measure for the transfer valuations since it depends on the transfer location along the path. Rather than the fuel consumption, the cost matrices store the velocity impulses, which are intrinsic measures of the transfer costs.

The interpolated cost $C_{\text{int}}(t_p, \Delta t_p, d_p, d_{p+1})$ gives the required velocity impulse ΔV_p for the pth transfer going from the debris d_p at the date t_p to the debris d_{p+1} at the date $t_{p+1} = t_p + \Delta t_p$.

The propellant consumed for this pth transfer is then assessed from the rocket equation.

$$m_c(t_p) = M(t_p)\left(1 - e^{-\frac{\Delta V_p}{v_e}}\right)$$

where $M(t_p)$ is the vehicle gross mass at the p^{th} transfer beginning. v_e is the engine exhaust velocity.

In addition to the transfer maneuvers, the mission assessment must also account for the debris operations, in terms of both durations and released masses. A fixed duration denoted Δt_{oper} is allocated to each debris operations (observation, capture and deorbitation). This duration is directly taken into account when building the cost matrices by including a last waiting sequence of duration Δt_{oper} in the transfer modeling, once the targeted debris is reached. An elementary transfer going from a debris 1 at the date t_1 to a debris 2 at the date t_2 is by this way completed at the date $t_2 - \Delta t_{\text{oper}}$. The operation durations between the successive transfers are thus implicitly accounted in the path valuation through the RSM.

The operation costs denoted C_{oper} are stored on the cost matrix diagonals, so that they can be accounted in the mission global assessment. Two deorbitation options are envisioned: either a deorbitation of the debris by the SDC vehicle or an autonomous deorbitation of the debris with a "kit" supplied by the SDC vehicle.

The storage depends on the deorbitation option as follows.

- The first option consists in a deorbitation of the debris by the SDC vehicle. For each debris the velocity impulse ΔV_{oper} required by the deorbitation depends on the debris altitude and it can be assessed a priori. The deorbitation velocity impulses of the N debris are stored on the cost matrix diagonals. For an edge going from the debris j to the debris k, the deorbitation cost $\Delta V_{\text{oper},k}$ of the debris k is added to the transfer interpolated cost, resulting in an additional fuel consumption.
- The second option consists in an autonomous deorbitation of the debris using a "kit" supplied by the SDC vehicle. The kit of mass m_{oper} is attached to the debris, and then, the debris is released to perform the deorbitation maneuver. In that option, the masses of the N kits designed respectively for the N debris are stored on the cost matrix diagonals. At the end of the propelled transfer arriving on the debris k the vehicle gross mass is decreased from the kit mass $m_{\text{oper},k}$ released to the debris k.

The cost of the debris operations is thus taken into account within the path optimization, whatever the deorbitation option selected.

It is also possible to consider weights w_k on the debris list to account for their priority, depending for example on their dangerousness. These weights come as multipliers on the cost matrix columns (arrival debris). The cost of an edge going from the debris j to the debris k is then assessed as: $w_k\left(C_{\text{int}}(t_p, \Delta t_p, j, k) + C_{\text{oper}}(k)\right)$

3.4 Practical Process

The debris orbits are retrieved at a given date from a database like the TLE of the NORAD [18]. A mesh of n_t discretized starting dates and n_d transfer durations is chosen in order to span the cleaning program forecast dates $[T_0;\ T_0 + \Delta T]$. The grid step results from a compromise between the response surface accuracy and the total computation time. The following choices based on the mean duration per mission and on the mean duration per transfer yield in practice an adequate compromise:

- $n_t \approx n \times m$ to associate one starting date per selected debris with

$$\begin{aligned} \tau_1 &= T_0 && \text{(cleaning program starting date)} \\ \tau_{nt} &= T_0 + \Delta T && \text{(cleaning program ending date)} \end{aligned}$$

- $n_d \approx$ n to associate one transfer duration per sub-path debris with

$$\begin{aligned} \Delta\tau_1 &= \Delta T/m/2/n && \text{(minimum = half of the mean transfer duration)} \\ \Delta\tau_{nd} &= \Delta T/m/2 && \text{(maximum= half of the mean mission duration)} \end{aligned}$$

In order to avoid extrapolations that could lead to erroneous cost assessments, it is also necessary to keep some “bounding matrices” in the mesh, particularly:

- A last row with a starting date greater than the ending date of the cleaning program. This lead to choose as last starting date: $\tau_{nt} = T_0 + \Delta T$
- On each row (with a starting date τ_i) either the maximal transfer duration ($\Delta\tau_{nd}$), or the smallest transfer duration $\Delta\tau_d$ exceeding the ending date of the cleaning program ($\tau_i + \Delta\tau_d > T_0 + \Delta T$).

The $n_t \times n_d \times N \times (N - 1)$ elementary transfer optimizations are run to fill the cost matrices. These optimizations are independent from each other, and they are parallelized on several processors. Each optimization is a NLP problem with 2 variables (drift orbit) and 1 constraint (final RAAN value). To spare some computation time, a filter discards the useless cases (ending date exceeding the end of the cleaning program) and the unfeasible transfers (requiring a drift altitude out of the allowed bounds). For such cases an arbitrary large cost value is stored in the corresponding matrix element, so that it will not be selected on the path.

The optimization variables (drift orbit radius and inclination) are initialized automatically, depending on the debris relative RAAN values. The convergence is typically achieved in a few seconds.

The simulated annealing algorithm is applied, with the cost function assessed by a RSM based on the cost matrices. The variables are the debris order and the rendezvous dates. The debris operations are accounted in terms of duration (taken into account the cost matrices) and cost (stored on the matrix diagonals depending on the deorbitation option).

At the convergence a solution path is issued defining the m missions of n debris each, with the optimized rendezvous dates. The convergence is achieved after some million trial solutions, with a typical computation time of a few minutes similar to a classical TSP problem.

The cost function for the simulated annealing has been computed through a RSM based on interpolations. The real cost is actually nonlinear, and it can be significantly different of the RSM cost, depending on the mesh discretization. In order to refine the mission planning, and to get a reliable cost assessment, the missions are re-optimized using a simulation-based software. The debris order is fixed, as well as the mission initial and final dates. For each mission, the rendezvous dates and the intermediate drift orbits are re-optimized to minimize the total ΔV. The refined assessment yields the requirement for the SDC vehicle design.

4 Application Case

The optimization method is illustrated on an application case with 21 debris. The cleaning program consists of 3 missions visiting 5 debris each one over a total duration of 45 months. Each mission requires the launch of an expendable SDC vehicle. This application case assumes that the SDC vehicle is equipped with a high-thrust propulsion system, but the solution method applies identically to the low-thrust case. For confidentiality reasons regarding current design studies, this application case considers fictitious debris and takes the velocity impulse as cost function instead of the vehicle mass. Nevertheless this application case is representative of the targeted debris population and of the overall optimization process.

4.1 Debris List

A list of 21 debris on circular orbits is considered, with the altitude ranging from 700 to 900 km, the inclination ranging from 97 to 99 deg and the initial RAAN between 0 to 360 deg. The values of altitude, inclination and initial RAAN are uniformly distributed in their respective intervals. The Table 3 provides the orbital parameters of the 21 debris, with their nodal precession rate in the last column. The orbits are nearly sun-synchronous with precession rates close to the Sun precession rate (360 deg in 365.25 days = 0.986 deg/day).

For real cases, the orbital parameters are retrieved from official databases like the NORAD TLEs [18].

4.2 Cleaning Program Specification

The goal is to design the lightest vehicle able to perform the 3 successive missions. Each mission has to visit 5 debris, so that 15 debris out of the 21 candidates will be visited. An average duration of 3 months per debris is considered, leading to a total duration of 45 months (1,370 days) for the overall cleaning program. A 5 days duration is also allocated to each debris operations. The altitude of the drift orbits is bounded between 400 and 2,000 km. The mission consists in visiting successively the debris, without deorbitation maneuvers. The cost function is the total velocity

Table 3 List of 21 candidate debris

Debris number	Altitude (km)	Inclination (deg)	Initial RAAN (deg)	Precession rate (deg/day)
Debris 1	700	97.0	0	0.8429
Debris 2	710	97.3	90	0.8745
Debris 3	720	97.6	180	0.9058
Debris 4	730	97.9	270	0.9367
Debris 5	740	98.2	018	0.9672
Debris 6	750	98.5	108	0.9975
Debris 7	760	98.8	198	1.0273
Debris 8	770	97.1	288	0.8260
Debris 9	780	97.4	36	0.8565
Debris 10	790	97.7	126	0.8866
Debris 11	800	98.0	216	0.9165
Debris 12	810	98.3	306	0.9460
Debris 13	820	98.6	54	0.9752
Debris 14	830	98.9	144	1.0040
Debris 15	840	97.2	234	0.8094
Debris 16	850	97.5	324	0.8389
Debris 17	860	97.8	72	0.8681
Debris 18	870	98.1	162	0.8969
Debris 19	880	98.4	252	0.9254
Debris 20	890	98.7	342	0.9536
Debris 21	900	99.0	360	0.9815

impulse ΔV required for the orbital transfer maneuvers (for real application cases, the cost function is the vehicle gross mass). The vehicle is assumed to be powered by a high-thrust engine.

4.3 Cost Matrices

The first stage of the solving method consists in building the cost matrices. Each cost matrix contains the transfer costs from any debris to any other for a given starting date and a given transfer duration. Filling one cost matrix requires solving 20×21 elementary transfer problems.

The matrices are assessed for a grid of discretized starting dates and durations, in order to span the total duration of the 3 missions. The application case specifies that 15 debris have to be visited within a 45 months period. We choose the following discretization:

- 16 starting dates ranging from 0 to 45 months
- 6 transfer durations ranging from 20 to 200 days

Table 4 Cleaning program (SA solution)

	Dates (days)	Debris visited	Total ΔV (m/s)
Mission 1	0–545.3	16–20–21–5–17	820.0
Mission 2	552.7–935.7	15–3–14–11–8	838.0
Mission 3	942.1–1365.9	1–4–9–7–12	837.5

Table 5 Mission planning (SA solution)

Mission 1			Mission 2			Mission 3		
Debris number	Date (days)	ΔV (m/s)	Debris number	Date (days)	ΔV (m/s)	Debris number	Date (days)	ΔV (m/s)
16	3.1	285.8	**15**	552.7	145.0	**1**	942.1	154.7
20	184.8	224.4	**3**	616.0	375.5	**4**	1014.6	362.8
21	375.0	216.0	**14**	771.5	140.4	**9**	1179.8	246.9
5	488.7	93.9	**11**	831.9	177.1	**7**	1268.0	73.2
17	545.3	**820.0**	**8**	935.7	**838.0**	**12**	1365.9	**837.5**

The total number of optimizations is $16 \times 6 \times 20 \times 21 = 40{,}320$. Each optimization is achieved in about 10 s, leading to a total computation time of 112 h. With a parallelization on 10 processors, the task is completed in a half day. The date and duration grids and the cost matrices are written in an output file. The file is directly usable by the simulated annealing algorithm in order to build the RSM.

4.4 Path Optimization

The second stage of the solving method consists in finding the optimal mission planning leading to the minimal ΔV requirement per mission. The cost is assessed by interpolation in the cost matrices. The convergence of the simulated annealing is obtained in about 10 min with 200 million trials.

The Table 4 presents the cleaning program found with the simulated annealing algorithm. The corresponding mission planning is detailed in the Table 5. It can be observed than the respective missions have close ΔV ranging from 820 to 838 m/s. This balanced cost between the missions is an indication of a good behavior of the simulated annealing algorithm.

A sample of the last iterations (Table 6) shows that several paths yield close cost values. The minimal cost found (838 m/s) is the cost of the most expensive mission which is in this case the second mission (Tables 4 and 5).

4.5 Refined Solution

In order to get a reliable cost assessment, the three missions are re-optimized using a simulation-based software. The debris order is fixed, as well as the mission initial and final dates. For each mission, the rendezvous dates and the intermediate drift orbits are

Table 6 Simulated annealing iterations (sample)

ΔV (m/s)	Debris order																				
900.6	6	18	14	3	15	19	16	7	4	1	17	20	21	5	10	13	2	8	12	11	9
891.2	6	18	14	3	15	19	16	7	4	1	20	17	21	5	10	13	9	8	11	2	12
889.2	2	13	17	5	21	15	3	14	11	8	1	4	9	7	12	19	16	18	6	10	20
884.0	2	13	17	5	21	15	3	14	11	8	1	4	9	7	12	19	16	18	6	10	20
881.7	2	13	17	5	21	15	3	14	11	8	1	4	9	7	12	19	16	18	6	10	20
879.3	1	21	5	9	20	15	3	14	11	8	16	19	4	7	12	18	6	13	2	17	10
871.8	1	21	5	9	20	15	3	14	11	8	16	19	4	7	12	18	6	13	2	17	10
838.0	16	20	21	5	17	15	3	14	11	8	1	4	9	7	12	19	2	13	18	6	10

Table 7 Cleaning program (simulation)

	Dates (days)	Debris visited	Total ΔV (m/s)
Mission 1	0–545.3	16–20–21–5–17	811.1 (820.0)
Mission 2	552.7–935.7	15–3–14–11–8	711.9 (838.0)
Mission 3	942.1–1365.9	1–4–9–7–12	785.1 (837.5)

Table 8 Mission planning (simulation)

Mission 1			Mission 2			Mission 3		
Debris number	Date (days)	ΔV (m/s)	Debris number	Date (days)	ΔV (m/s)	Debris number	Date (days)	ΔV (m/s)
16	3.1	287.1	**15**	552.7	141.5	**1**	942.1	119.2
20	183.1	210.8	**3**	563.3	291.8	**4**	976.8	411.8
21	389.3	202.2	**14**	781.7	132.2	**9**	1143.4	183.5
5	513.6	111.0	**11**	823.0	146.4	**7**	1177.3	70.6
17	545.4	**811.1**	**8**	935.8	**711.9**	**12**	1365.8	**785.1**

re-optimized to minimize the total mission ΔV. The rendezvous with the successive debris is constrained with a maximum RAAN deviation of 1 deg.

The Table 7 presents the cleaning program found after the dates and maneuvers re-optimization, for the high-thrust case. The total ΔV per mission is given in the last column (in parenthesis the previous RSM assessment from the simulated annealing). The corresponding mission planning is detailed in the Table 8.

Comparing to the SA results using the RSM cost assessment, all the missions are improved owing to the drift orbit parameters refined optimization. The main changes are marked by the orange-colored cells.

- The improvement on the mission 2 comes mainly from the second leg whose ΔV is reduced of 85 m/s. This is explained by the starting date advance (−50 days), which increases the transfer duration up to 218 days. Such a solution could not be detected with the RSM, because the mesh was defined with a maximum transfer duration of 200 days. In such a case, it could be useful to iterate the whole process with an updated discretization.
- The improvement on the mission 3 comes mainly from the advance of the second rendezvous date (−40 days). The ΔV of the second leg is increased of about 50 m/s, but this is globally counterbalanced by the gains on the first and the third legs.

Table 9 RAAN values at the mission beginning

Mission 1 (t_1 = 3.1 days)		Mission 2 (t_6 = 552.7 days)		Mission 3 (t_{11} = 942.1 days)	
Debris number	RAAN (deg)	Debris number	RAAN (deg)	Debris number	RAAN (deg)
16	-33.4	**15**	-38.6	**1**	74.1
20	-15.1	**3**	-39.4	**4**	72.4
21	3.0	**14**	-21.1	**9**	122.8
5	21.0	**11**	2.5	**7**	74.1
17	74.7	**8**	24.5	**12**	117.2

Table 10 Re-optimized missions

Mission 1			**SDC vehicle**		**Transfer**		**Drift orbit**		**Debris orbit**		**RAAN (deg)**	
Edge number	Initial debris	Final debris	Date (days)	Total DV (m/s)	Duration (days)	DV (m/s)	Altitude (km)	Inclination (deg)	Altitude (km)	Inclination (deg)	SDC Vehicle	Debris
1		16	3.1	0	3.1	0			850.0	97.50	326.6	326.6
2	16	20	183.1	287.1	180.0	287.1	708.0	98.84	890.0	98.70	156.7	156.6
3	20	21	389.3	497.9	206.2	210.8	715.8	99.20	900.0	99.00	22.2	22.1
4	21	5	513.6	700.1	124.3	202.2	695.4	98.90	740.0	98.20	155.0	154.8
5	5	17	545.4	811.1	31.8	111.0	712.9	98.24	860.0	97.80	185.7	185.4

Mission 2			**SDC vehicle**		**Transfer**		**Drift orbit**		**Debris orbit**		**RAAN (deg)**	
Edge number	Initial debris	Final debris	Date (days)	Total DV (m/s)	Duration (days)	DV (m/s)	Altitude (km)	Inclination (deg)	Altitude (km)	Inclination (deg)	SDC Vehicle	Debris
1		15	552.7	0	3.1	0			840.0	97.20	321.7	321.4
2	15	3	563.3	141.5	10.6	141.5	830.3	96.93	720.0	97.60	330.5	330.2
3	3	14	781.7	433.3	218.4	291.8	572.5	98.55	830.0	98.90	209.2	208.8
4	14	11	823.0	565.5	41.3	132.2	812.4	98.93	800.0	98.00	250.5	250.2
5	11	8	935.8	711.9	112.9	146.4	825.9	97.10	770.0	97.10	341.4	340.9

Mission 3			**SDC vehicle**		**Transfer**		**Drift orbit**		**Debris orbit**		**RAAN (deg)**	
Edge number	Initial debris	Final debris	Date (days)	Total DV (m/s)	Duration (days)	DV (m/s)	Altitude (km)	Inclination (deg)	Altitude (km)	Inclination (deg)	SDC Vehicle	Debris
1		1	942.1	0	3.1	0			700.0	97.00	74.7	74.1
2	1	4	976.8	119.2	34.8	119.2	702.9	97.32	730.0	97.90	105.4	105.0
3	4	9	1143.4	531.0	166.6	411.8	412.4	98.29	780.0	97.40	295.9	295.3
4	9	7	1177.3	714.5	33.9	183.5	759.2	98.07	760.0	98.80	328.0	327.5
5	7	12	1365.8	785.1	188.6	70.6	763.7	98.69	810.0	98.30	158.8	158.1

A refined mesh discretization may help capturing these nonlinearities of the cost function within the RSM.

Another interesting observation can be done by considering for each mission the RAAN of the selected debris at the mission starting date (Table 9). It can be noted that the debris assigned to each mission are somewhat gathered by their initial RAAN values, in increasing order. This is not an absolute rule, since the optimal order also depends on the other orbital parameters (radius and inclination). When large RAAN differences exist at the mission starting date (for example for the third and fifth debris of the mission 3), they are nullified by long transfer durations (up to 6 months).

The re-optimized mission planning is detailed in the Table 10. The last column checks the RAAN constraint at the successive rendezvous with the debris. Some observations can be done on the green- and orange-colored cells.

- The short drift durations (green cells) correspond to a drift orbit close to the starting orbit. The natural precession on the initial orbit is nearly sufficient to reach the targeted RAAN value, so that there is no need for changing significantly the precession rate (cf Remark 2.1). This reduces the ΔV required for the transfer.

- The large drift durations (orange cells) correspond to low-altitude drift orbits. These transfers require a large RAAN change achieved by both an accelerated precession rate (low altitude) and a long duration (about 6 months). The costs of these legs represent about half the mission cost.

5 Conclusions

In order to clean the near-Earth space from the most dangerous debris, a cleaning program is envisioned. It consists of several successive missions, performed by identical vehicles, in order to achieve a mean removal rate of five debris per year. A solution method is proposed for the planning of these successive Space Debris Collecting missions. The goal is to minimize the fuel required by the most expensive mission, with the perspective of designing a generic vehicle compliant of the cleaning program. The problem mixes combinatorial optimization to select and order the debris among a list of candidates, and continuous optimization to fix the rendezvous dates and to define the minimum fuel orbital maneuvers. The solution method proposed consists in three stages.

Firstly the orbital transfer problem is simplified by considering a generic transfer strategy suited either to a high-thrust or a low-thrust vehicle. A response surface modeling is built by solving the reduced problem for all pairs of debris and for discretized dates, and storing the results in cost matrices. This first stage is parallelized on several processors. The results of this series of optimizations are stored in cost matrices.

Secondly a simulated annealing algorithm is applied to find the optimal mission planning. The cost function is assessed by interpolation on the response surface based on the cost matrices. This allows the convergence of the simulated algorithm in a limited computation time, yielding an optimal mission planning.

Thirdly the successive missions are re-optimized in terms of transfer maneuvers and dates without changing the debris order. This continuous control problem is simulation based, taking into account the problem nonlinearities that were not captured by the response surface modeling. It yields a refined solution with the performance requirement for designing the future Space Debris Collecting vehicle.

The method is applicable for a large list of debris and for various assumptions regarding the cleaning program (number of missions, number of debris per mission, total duration, deorbitation scenario, high- or low-thrust vehicle). It is exemplified on an application case considering a high-thrust propulsion system, with 3 missions to plan, each mission visiting 5 near sun-synchronous debris to be selected in a list of 21 candidates.

Weights can be attributed to the debris in order to account for their dangerousness and to assign if desired a priority in the selection process. The generic transfer strategy can be considered as near optimal as long as a significant duration is allocated to the drift phases. For a low-thrust vehicle, the available acceleration level must be sufficient, in order that the propelled transfers do not exceed a few days. The overall optimization process is automatized, and a mission planning can be established in a few hours considering various specifications.

Acknowledgments This work was carried out at Airbus Defence and Space in 2013–2014 in the frame of the internal R&D. I would like to thank the R&D team project for having supported this work.

References

1. Position paper on space debris mitigation. International Academy of Astronautics (2005)
2. Klinkrad, K. (s.d.): Space debris models and risks analysis. ISBN:3-540-25448-X
3. Liou, J.: An assessment of the current LEO debris environment and what needs to be done to preserve it for future generations (2008)
4. Walker R., Martin C.E., Stokes P.H., Wilkinson J.E. : Studies of space debris mitigation options using the debris environment long term analysis (DELTA) model. In: 51st International Astronautical Congress, Brazil, pp. IAA-00_IAA.6.6.07. Rio de Janeiro (2000)
5. Nemhauser, G.L., Rinnooy Kan, A.H.G., Todd, M.J.: Optimization. In: Handbooks in Operations Research and Management Science, vol. 1. Elsevier, Amsterdam (1989)
6. Dréo J., Pétrowski A., Siarry P., Taillard E.: Métaheuristiques Pour l'optimisation Difficile. Eyrolles, Paris (2003)
7. Schneider, J.J., Kirkpatrick, S.: Stochastic Optimization. Springer, Berlin (2006)
8. Cerf, M.: Multiple space debris collecting mission, debris selection and trajectory optimization. J. Optim. Theory Appl. **156**, 761–796 (2013). doi:10.1007/s10957-012-0130-6
9. Braun, Vitali, Lüpken, A., Flegel, S., Gelhaus, J., Möckel, M., Kebschull, C., Wiedemann, C., Vörsmann, P.: Active debris removal of multiple priority targets. Adv. Space Res. **51**, 1638–1648 (2013)
10. Barbee, B.W., Alfano, S., Piñon, E., Gold, K., Gaylor, D.: Design of spacecraft missions to remove multiple orbital debris objects. AAS 12-017. In: 35th Annual AAS Guidance and Control Conference (2012)
11. Zuiani, F., Vasile, M.: Preliminary design of debris removal missions by means of simplified models for low-thrust many revolution transfers. Int. J. Aerosp. Eng. 836250 (2012)
12. Olympio, J.T., Frouvelle, N.: Space debris selection and optimal guidance for removal in the SSO with low thrust propulsion. Acta Astronaut. **99**, 263–275 (2014)
13. Stuart, J., Howell, K., Wilson, R.: Application of multi-agent coordination methods to the design of space debris mitigation tours. In: 24th International Symposium on Space Flight Dynamics (2014)
14. Chobotov, V.: Orbital Mechanics, 3rd edn. AIAA Education Series (2002)
15. Vallado, D.A.: Fundamentals of Astrodynamics and Applications, 3rd edn. Space Technology Library (2007)
16. Leitmann, G.: Theory of maxima and minima. In: Optimization Technique with Applications to Aerospace System. Academic Press, New York (1962)
17. Conway, B.A.: Spacecraft Trajectory Optimization. Cambridge University Press, Cambridge (2010)
18. NORAD Two Line Elements. http://www.celestrak.com/NORAD/elements/