Topology-based Launch Collision Avoidance augmented with Differential Evolution

Luca Leoni[1,2], Giuseppe Di Campli Bayard de Volo[1], Ralph Kahle[1]

[1] *German Aerospace Center (DLR), German Space Operations Centre (GSOC)*
Münchener Str. 20, 82234 Weßling, Germany
{giuseppe.dicamplibayarddevolo, ralph.kahle}@dlr.de

[2] *Faculty of Aerospace Engineering, Delft University of Technology*
Kluyverweg 1, 2629 HS Delft, The Netherlands
l.leoni@student.tudelft.nl

***Abstract* – This work enhances the Launch COLlision Avoidance (LCOLA) tool ASSET (Ascent SafETy) developed at DLR/GSOC by introducing a continuous, topology-based methodology for conjunction detection across full launch windows. A two-dimensional time of closest approach search, formulated in the domain of liftoff time and mission elapsed time, and solved via differential evolution optimization, enables continuous liftoff coverage, and accurate launch window closures identification based on probability of collision and miss distance. Validation on a representative mission demonstrates full identification of risky conjunctions while reducing computational cost by two orders of magnitude compared to the discrete approach, significantly improving operational responsiveness and launch safety.**

I. INTRODUCTION

The number of space debris objects in Earth orbit continues to grow steadily (see Fig. 1), and the risks posed by this overpopulation demand unprecedented solutions to ensure safe and sustainable access to space [1]. Launch COLlision Avoidance (LCOLA) refers to the set of analyses and operational procedures designed to prevent collisions between a launch vehicle, including its components and payloads, and existing space objects during the ascent, injection, and early orbit phases. Its primary objective is to minimize the probability of collision (or, equivalently, maximize the miss distance) with operational satellites, dead spacecraft, and debris, thereby reducing the generation of new space debris and safeguarding both the mission and the overall space environment.

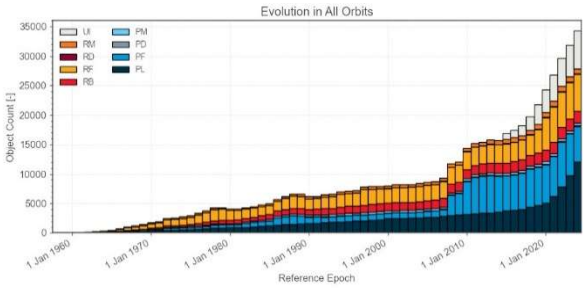

Fig. 1: Evolution of Resident Space Objects in all orbital regimes [1].

In recent years, several space agencies and industrial actors have developed their own LCOLA tools. Examples include the systems developed by Deimos Space UK [2], CNES [3], and the ASSET (AScent SafETy) software created by the Flight Dynamics Team at DLR/GSOC [4]. Despite independent development efforts, these organizations have reported similar challenges and limitations, primarily driven by the discretized LCOLA strategy commonly adopted across the industry. A detailed discussion of these limitations is provided in [5].

The conventional discretized LCOLA process is computationally expensive also due to the growing population of resident space objects (RSOs). It relies on sampling the launch window at fixed time intervals and screening each sample against every secondary object. Although effective when specific lift-off times are of interest, this approach cannot provide risk information between consecutive samples and may therefore overlook launch window closures. Moreover, the computational cost can become prohibitive, especially when multiple payloads or long screening periods are required, even when relatively coarse sampling steps are used.

A recent study introduced a new methodology, referred to as topology-based LCOLA, that aims to address these limitations [5]. However, the publication does not provide the algorithmic details necessary for direct implementation.

This work bridges that gap by implementing the topology-based LCOLA methodology within the ASSET software framework. In addition, it introduces a novel evolutionary optimization approach for the identification of the Time of Closest Approach (TCA), inspired by the method developed by Analytical Graphics Inc. (AGI) and implemented in its launch window analysis tool [6].

While the AGI approach relies on two separate mathematical algorithms for TCA identification, the proposed method consolidates the process into a single optimization routine. This is achieved through a differential evolution strategy, resulting in a more streamlined and computationally efficient formulation.

The remainder of this paper is organized as follows: section II describes the revised software architecture of

ASSET; section III details the proposed evolutionary topology-based LCOLA algorithm; section IV presents the results of a representative test case used for verification and validation; and section V provides concluding remarks and directions for future work.

II. Revised ASSET Architecture

The development of a topology-based LCOLA approach within the ASSET software required a redefinition of its software architecture, currently described in [4]. Fig. 2 presents the redefined topology-based ASSET software architecture. In particular, the user inputs adopted for the classical ASSET remain unchanged. The pre-processing step generates the inertial trajectories of the primaries at a fixed liftoff time step of two minutes within the launch window. This two-minute interval is selected to avoid computationally expensive interpolations, as the secondary catalogue provides ephemerides at the same temporal resolution.

Given the primary trajectories, a geometric apogee–perigee filter is applied to the secondary catalogue. The sieve filters are removed from the original ASSET framework, as they are computationally expensive and not required in the topology-based approach. Consequently, each primary–secondary pair proceeds to the screening step, where the proposed Differential Evolution (DE) TCA search is employed. The risky conjunctions identified in this step define the epochs at which the secondary covariances must be estimated using synthetic covariance generation [7]. Subsequently, the probability of collision (PoC) for each risky conjunction is computed, and the two launch-window closure algorithms (based on PoC and miss distance, described in [5], [6], respectively) are executed.

Finally, the post-processing step aggregates the launch-window closures into a JSON file, stores the risky conjunction parameters in a separate JSON file, and provides the user with a summary plot of PoC, miss distances, and launch-window closures (presented in the following section).

III. Evolutionary Topology-Based LCOLA

A. Reference frames

The proposed formulation relies on two reference frames. The inertial frame $\mathcal{F}_I$ is chosen as the Earth centered inertial (EME2000) frame used to propagate the primaries and store the catalogue objects. The Earth fixed frame $\mathcal{F}_E$ is the international terrestrial reference frame (ITRF2000), in which the primaries trajectories are provided as a function of the Mission Elapsed Time (MET) τ.

Let t_0 denote the liftoff epoch and $t = t_0 + \tau$ the corresponding absolute time. The inertial position of the secondary is given by $\mathbf{r}_s^I(t)$, directly available in $\mathcal{F}_I$. The primary position is first defined in $\mathcal{F}_E$ as $\boldsymbol{r}_p^E(\tau)$, and then rotated into the inertial frame through the Earth rotation matrix $A_{E\to I}(t)$:

$$\boldsymbol{r}_p^I(t_0,\tau) = A_{E\to I}(t_0+\tau)\boldsymbol{r}_p^E(\tau). \tag{1}$$

For any pair (t_0,τ) in the search domain, the primary-secondary distance $d(t)$ is evaluated in the inertial frame as follows:

$$d(t) = |\boldsymbol{r}_s^I(t_0+\tau) - \boldsymbol{r}_p^I(t_0,\tau)|. \tag{2}$$

B. LCOLA Topological Properties

Before detailing the algorithm, the distance function in (2) is analysed to determine whether, and under which conditions, it becomes singular in the vicinity of a stationary point. In other words, if the distance function is singular, a well-defined local minimum (i.e., the TCA) does not exist. Assume that the Earth rotation matrix can be approximated (without compromising the analysis) as

$$A_{E\to I}(t_0+\tau) \approx R_z(\Omega_E t_0)R_z(\Omega_E \tau), \tag{3}$$

where $R_z(\alpha)$ denotes a rotation about the z-axis by an angle α, and Ω_E is the Earth's angular velocity. Under this approximation, the inertial primary position is

$$\boldsymbol{r}_p^I(t_0,\tau) \approx R_z(\Omega_E t_0)R_z(\Omega_E \tau)\boldsymbol{r}_p^E(\tau), \tag{4}$$

and the relative vector $\boldsymbol{\Delta}(t_0,\tau)$ becomes

$$\boldsymbol{\Delta}(t_0,\tau) = \boldsymbol{r}_s^I(t_0+\tau) - R_z(\Omega_E t_0)R_z(\Omega_E \tau)\boldsymbol{r}_p^E(\tau). \tag{5}$$

Therefore, the gradient of the squared distance $f = |\boldsymbol{\Delta}|^2$ is given by

$$\frac{\partial f}{\partial t_0} = 2\boldsymbol{\Delta}\cdot\frac{\partial \boldsymbol{\Delta}}{\partial t_0}, \quad \frac{\partial f}{\partial \tau} = 2\boldsymbol{\Delta}\cdot\frac{\partial \boldsymbol{\Delta}}{\partial \tau}. \tag{6}$$

A stationary point satisfies $\frac{\partial f}{\partial t_0} \approx \frac{\partial f}{\partial \tau} \approx 0$. This can occur in three cases:

1. $\boldsymbol{\Delta} \perp \frac{\partial \boldsymbol{\Delta}}{\partial t_0} \wedge \boldsymbol{\Delta} \perp \frac{\partial \boldsymbol{\Delta}}{\partial \tau}$
2. $\boldsymbol{\Delta} = (0,0,0)$
3. $\frac{\partial \boldsymbol{\Delta}}{\partial t_0} = \frac{\partial \boldsymbol{\Delta}}{\partial \tau} = (0,0,0)$

The last two cases are not considered. In fact, case 2 is highly unlikely, and case 3 would require case 2 to hold for two satellites in GEO (to ensure $\frac{\partial \boldsymbol{\Delta}}{\partial t_0} = \mathbf{0}$); therefore, case 3 is also not of interest. Accordingly, only case 1 is discussed. Specifically, $\frac{\partial f}{\partial \tau}(t_0,\tau)$ in (6) corresponds to the classical TCA condition in satellite collision avoidance, namely a zero-range rate at TCA [8] ($\boldsymbol{\Delta}$ is orthogonal to the relative velocity). Furthermore, two orbital configurations are presented that render the

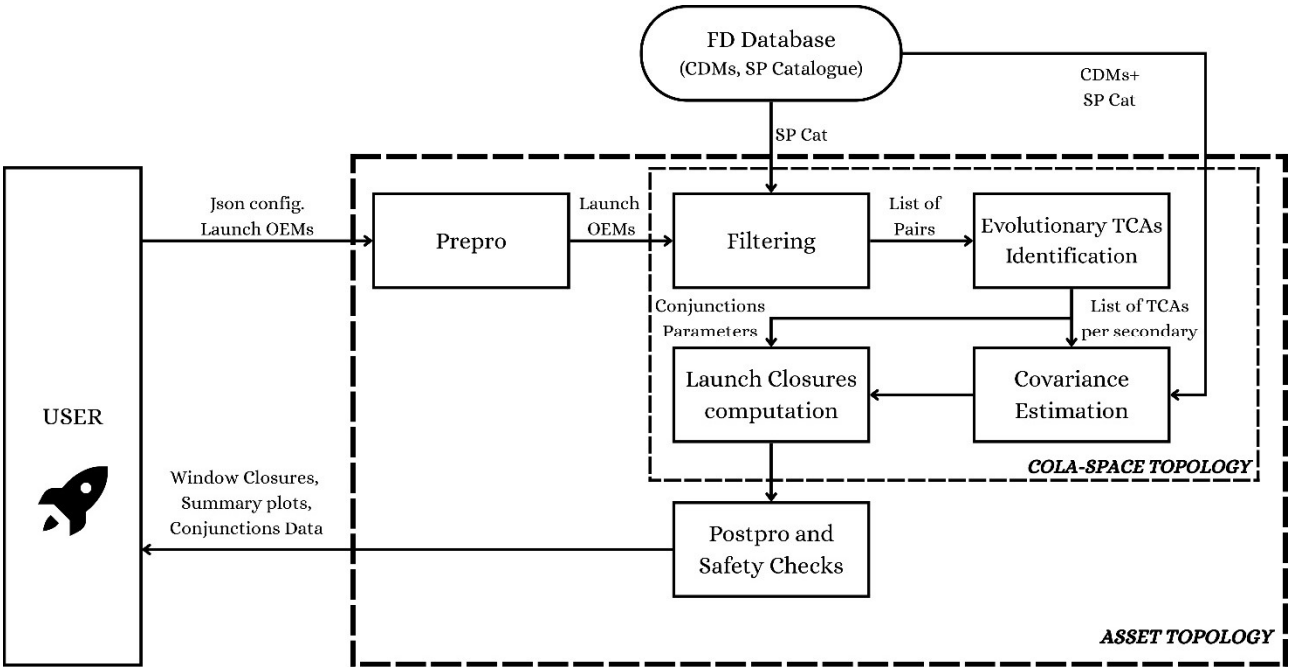

Fig. 2: ASSET Topology Software Architecture

distance function f singular; that is, there exists at least one direction in (t_0, τ) along which f remains constant:

i. *GEO-GEO configuration.* If both satellites are geostationary, circular, and equatorial, with semi-major axis (SMA) a_{GEO} and mean motion equal to the Earth's rotation rate Ω_E, then

$$f(t_0, \tau) = \text{const.},$$

$$\frac{\partial f}{\partial t_0}(t_0, \tau) = \frac{\partial f}{\partial \tau}(t_0, \tau) = 0, \ \forall (t_0, \tau),$$

and all second-order partial derivatives vanish as well.

ii. *Circular, coplanar, equal-SMA configuration.* Suppose both satellites move on circular orbits with the same semi-major axis a and inclination i, and thus the same mean motion n. Then, there exists an infinite set of liftoff epochs $t_0^\star$ for which the RAANs of the two orbits coincide in the inertial frame, so that the orbits become coplanar. For any such $t_0^\star$,

$$f(t_0^\star, \tau) = \text{const.}, \qquad \frac{\partial f}{\partial \tau}(t_0^\star, \tau) = 0 \quad \forall \tau,$$

and the second order derivatives vanish as well, apart from the second derivative with respect to the liftoff.

The proof of these statements is provided in Appendix A.

C. Methodology

The TCA search algorithm is based on a differential evolution (DE) minimization of (2). The DE algorithm employed in this study is implemented in the scipy.optimize library [9]. The overall procedure can be divided into two main steps: (i) identification of multiple intervals each containing a single TCA for each primary–secondary pair, and (ii) minimization of the distance within each of these intervals.

The identification of the intervals relies on a modified version of the algorithm described in [6]. Specifically, the launch window is sampled at a fixed time step, the corresponding inertial trajectories of the primary are generated, and the distances between each primary–secondary pair are computed at each discrete (t_0, τ). At this stage, the algorithm in [6] proposes the use of a mathematical procedure to identify, for each discrete τ, the absolute minimum distance across all liftoff epochs. However, this approach is not suitable for long launch windows, where multiple local minima may exist at a given τ across t_0. This situation arises when the launch window exceeds the orbital period of the secondary.

In the methodology proposed here, this step is performed by storing each discrete local minimum (in terms of t_0 and d) for every τ. This results in significant computational savings, as no additional mathematical procedure is required and, due to the subsequent DE refinement, the accuracy of the method is not compromised. To account for the presence of multiple local minima in long launch windows, the window is automatically split into N sub-intervals such that each sub-interval contains only one minimum. An example of a launch window divided into four sub-intervals, together with the storage of four local minima for a

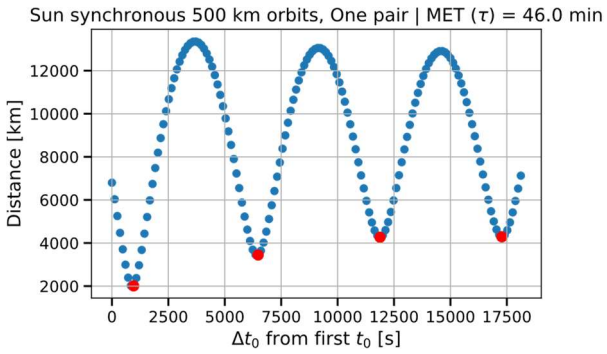

Fig. 3: Discrete $(d; t_0)$ points for an arbitrary primary-secondary pair and τ (46 minutes). Local minima highlighted in red.

primary–secondary pair at a representative τ, is shown in Fig. 3.

The dataset containing all stored (d, t_0, τ) points is then used to identify the intervals that contain a local minimum of (2) (thus, a TCA). Specifically, a discrete local minimum in the 2D space (d, τ) defines the center of the corresponding τ interval, with its boundaries determined by the neighbouring points. Around each discrete local minimum, the t_0 boundaries are defined as $\widetilde{t_0} \pm \Delta t_0$, where $\widetilde{t_0}$ denotes the liftoff epoch at the discrete local minimum and Δt_0 is the t_0 step size adopted to generate the inertial primary trajectories. Within each interval, the absolute minimum is guaranteed to be contained, and the distance function is ensured to be locally convex according to the discussion in subsection III-B, except for the identified singular cases that will be addressed later. Fig. 4 illustrates the dataset (d, t_0, τ), in which multiple discrete local minima are present. In particular, the t_0 values corresponding to two local minima at approximately the same τ differ by about 1.5 hours, which, in this case (500 km orbits), corresponds to the orbital period of the secondary object.

Given the intervals identified in the first stage of the algorithm, the second stage employs the DE optimizer to determine the absolute minimum within each interval, denoted as $(\bar{d}, \bar{\tau}, \bar{t_0})$. The DE configuration details are provided in subsection III-D. Fig. 5 presents the topology of a representative primary–secondary pair, including the optimization solution, from which the periodicity over t_0 and τ can be observed.

Subsequently, synthetic covariances are generated for each secondary object at each risky TCA epoch $(\bar{t_0} + \bar{\tau})$. The final step of the topology-based LCOLA approach consists of computing the probability of collision and the corresponding launch closure window intervals. In ASSET, the 2D Alfano PoC method P_{xy} proposed in [10] is adopted. After computing the PoC, the topology-based algorithms for launch closure window determination are applied. The miss distance-based algorithm is described in [6], whereas the PoC-based approach is presented in [5]. It was found that both

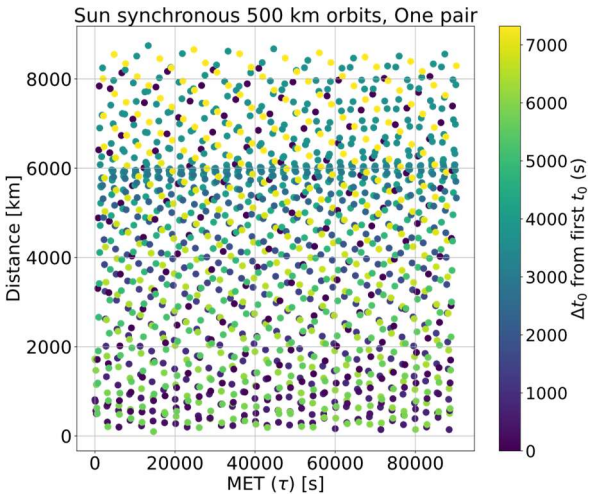

Fig. 4: (d, t_0, τ) dataset for a primary-secondary pair over a 2-hour launch window.

algorithms should be executed together, even when P_{xy} remains below the prescribed threshold (and a PoC screening was selected). Indeed, in the (t_0, τ) representation, the identified point of minimum miss distance does not necessarily coincide with the point of maximum PoC. This discrepancy arises because a larger miss distance in the tangential direction may yield a higher PoC than a smaller miss distance confined to the RN plane. The methodology has been adjusted to deal with the two singular cases. Specifically, in the GEO–GEO configuration, the discrete distances obtained in the first step are expected to be very similar across all discrete (t_0, τ) points. If this occurs, the pair can be discarded immediately when the distance is sufficiently large. Conversely, if the distances are small, the entire discrete distance dataset is retained and the optimization step is skipped, proceeding directly to the PoC computation and, if necessary, to the launch-closure algorithm. This latter step is applied when the risk level changes between two consecutive liftoffs, allowing the determination of the exact t_0 at which the risk level shifts. In contrast, case 2 is not fully degenerate, and the first step of storing the local minima for each τ can still be performed. Moreover, since in practice the singularity of case 2 is mitigated by perturbations and perfectly circular orbits do not occur, the previously described methodology remains valid even for these highly improbable scenarios.

D. DE Hyperparameters tuning

The tuning of the DE optimizer followed a step-by-step procedure, adjusting one hyperparameter at a time before proceeding to the next.

First, a benchmark optimization was defined. The hyperparameters were kept at their default values in the scipy.optimize library, except for the population size (set to 15) and the convergence criterion, where the relative tolerance was disabled and only an absolute tolerance ($atol = 10^{-5} s$) was retained. An initial run

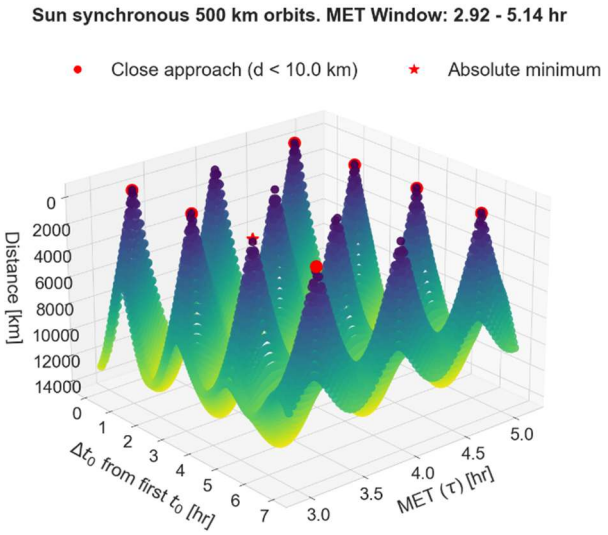

Fig. 5: 3D Topology for a primary-secondary pair after DE Optimization

with the benchmark configuration was performed on the entire secondary catalogue to identify all risky primary–secondary pairs. The tuning process was then conducted only on this restricted set, reducing the computational time. All reported CPU times refer to this second run.

To evaluate each configuration, three figures of merit were defined, addressing accuracy, computational time, and robustness. The accuracy metric, f_{acc}, defined in (7), assigns 100% accuracy ($f_{acc,i} = 1$) if the minimum distance and TCA differences with respect to the benchmark are within 1 m and 0.1 s, respectively. Otherwise, a linear penalty proportional to the deviation is applied, with the contribution set to zero if the miss-distance deviation exceeds 10 m or the TCA deviation exceeds 1 s.

$$f_{\text{acc}} = \frac{1}{2N}\sum_{\text{conj}}\left[\max\left(\min\left(\frac{1}{9}\left(\Delta x_{\text{conj}} - 10\right), 1\right), 0\right) + \right. \\ \left. + \max\left(\min\left(-\frac{10}{9}\left(\Delta t_{conj} - 1\right), 1\right), 0\right)\right] \quad (7)$$

where $conj$ denotes a conjunction identified in the tested configuration, N is the total number of conjunctions in that configuration, and Δx_{conj} and Δt_{conj} represent the absolute miss-distance (meters) and TCA (seconds) deviations with respect to the benchmark.

Computational time is measured directly via CPU time, while robustness is quantified through the Percentage of Detected Conjunctions (PDC), defined as the fraction of conjunctions detected relative to the benchmark.

The tuned hyperparameters, in chronological order, are: polish, absolute tolerance ($atol$), population size (NP), mutation factor (F), and recombination rate (CR). The polish option was set to False, as enabling it increased CPU time without improving accuracy.

Fig. 6 and Fig. 7 report the explored hyperparameter ranges and the corresponding results in terms of the three figures of merit. For absolute tolerance, PDC remains 100% across the tested range. This indicates that the LCOLA problem can be efficiently addressed with a two-stage approach: a coarse run with $atol$ = 0.1s to restrict the catalogue to risky secondaries and decrease CPU time, followed by a refinement run with $atol$ = 0.001s on the reduced set. For tuning the remaining hyperparameters, $atol$ = 0.1s was adopted.

The population size NP exhibits a direct trade-off between CPU time and accuracy. A value of NP = 10 was selected as it is the smallest population to achieve PDC = 100%. Mutation and recombination tuning is straightforward, as the optimal region is clearly identified in Fig. 7, leading to the selection of F = 0.3 and CR = 0.7. Table 1 summarizes the figures of merit for the benchmark and the two selected hyperparameter sets (differing in $atol$), together with the number of missed warnings and alarms relative to the benchmark. As expected, the configuration with $atol = 0.1\,s$ achieves PDC = 100% with lower CPU time than the $atol = 0.001\,s$ case, while the latter provides higher accuracy and detects all warnings and alarms.

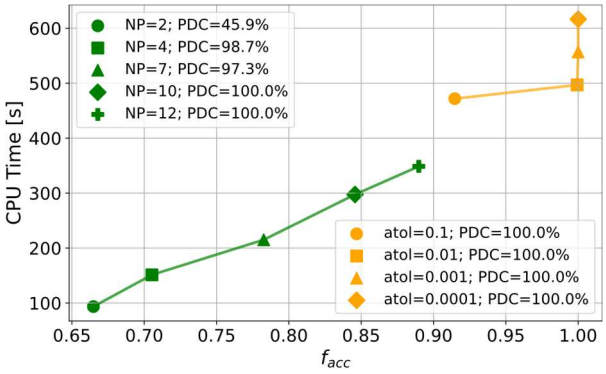

Fig. 6: Tuning results for population size (NP) and absolute tolerance ($atol$).

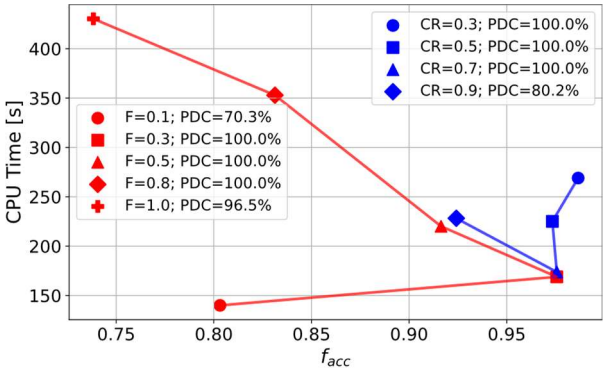

Fig. 7: Tuning results for mutation factor (F) and recombination rate (CR).

Table 1: Figure of merits for benchmark and tuned settings

Function of merit	Bench. run	Tuned run ($atol = 0.1s$)	Tuned run ($atol = 10^{-3}s$)
CPU Time [s]	720.234	173.843	205.019
$f_{\mathbf{acc}}$	1.0	0.976	0.995
PDC	100%	100%	100%
Missed warnings	0	4	0
Missed alarms	0	0	0

IV. Results

A. Test Case

The test case considered consists of a 60-minute launch window for four primaries (rocket upper stage and three payloads) over a one-day screening period for a Sun-Synchronous Orbit (SSO) at 500 km altitude. The screening is based exclusively on the probability of collision, with an alarm threshold of 10^{-5} and a warning threshold of 10^{-6}. Table 2 reports the total launch window closure time due to warnings and alarms, together with the number of identified closures (for the refinement LCOLA run). For this test case, the launch window is closed (including warnings) for 7.4% of its total duration. Whereas a hybrid screening scenario, with a miss distance threshold of 100 m, would result in a 28.3% window closure. Fig. 8 shows the ASSET Topology summary plot (restricted to a 2-minute liftoff coverage for clarity). The top panel displays the launch window status, with green, yellow, and red indicating safe, warning, and alarm conditions, respectively, based solely on PoC. The middle and bottom panels report the PoC and miss distance evolution for each primary–secondary pair. Multiple curves of the same colour correspond to recurrent TCAs of the same pair at slightly different liftoff epochs. Selecting any point on the curves reveals the associated conjunction parameters.

Table 2: Total warning and alarm durations and counts for LCOLA.

Risk Level	Total time [min]	Number of Occurrences
WARNING	3.332	266
ALARM	1.122	81
TOTAL	4.454	348

B. Verification and Validation

The evolutionary topology-based LCOLA implementation in ASSET was verified through targeted tests of the TCA search and closure window computation by comparison with the discretized ASSET and the in-house collision avoidance tool COLA. Conjunction identification was assessed in three scenarios: (i) single primary–secondary, (ii) single primary–multiple secondaries, and (iii) multiple primaries–multiple secondaries, with the objective of ensuring that ASSET Topology accurately identified all conjunctions. For consistency, the discretized ASSET was executed with a liftoff step size of 0.1 s.

In scenario (i), 30 TCAs were identified by both approaches, with ASSET Topology yielding smaller miss distances (e.g., 0.2 m versus 32.4 m). In scenario (ii), 31 TCAs were detected by both methods, while ASSET Topology achieved a 98.7% CPU time reduction and significantly reduced the most critical miss distance (e.g., from 460.8 m to 14.0 m). Scenario (iii) showed comparable improvements, with a 97.92% CPU reduction and further refinement of miss distances. Overall, the evolutionary topology-based LCOLA detected all conjunctions found by the discretized ASSET and refined miss-distance estimates at substantially lower computational cost. Despite the very fine liftoff grid adopted in the discretized ASSET, ASSET Topology achieved higher accuracy while drastically reducing CPU time. The PoC computation module and the PoC-based launch window closure algorithm were validated against COLA; Table 3 reports the results for a representative PoC-based closure. The miss-distance-based closure windows were verified against discretized ASSET using a fine liftoff grid, with results for a representative closure shown in Table 4.

These tests confirm that ASSET Topology correctly identifies conjunctions, computes PoCs, and determines closure windows, while delivering substantial computational speed-ups.

Table 3: Comparison of conjunction parameters between COLA and ASSET Topology within a PoC-based launch window closure.

Parameter	ALARM Start	Highest PoC	ALARM Stop
TCA (COLA)	17:37:37.630	17:37:37.798	17:37:37.967
TCA (Topology)	17:37:37.630	17:37:37.798	17:37:37.967
Miss. Dist. [m] (COLA)	874.5	1.2	875.7
Miss. Dist. [m] (Topology)	874.5	1.2	875.8
PoC (COLA)	9.84×10^{-6}	3.37×10^{-4}	9.92×10^{-6}
PoC (Topology)	9.84×10^{-6}	3.37×10^{-4}	9.92×10^{-6}

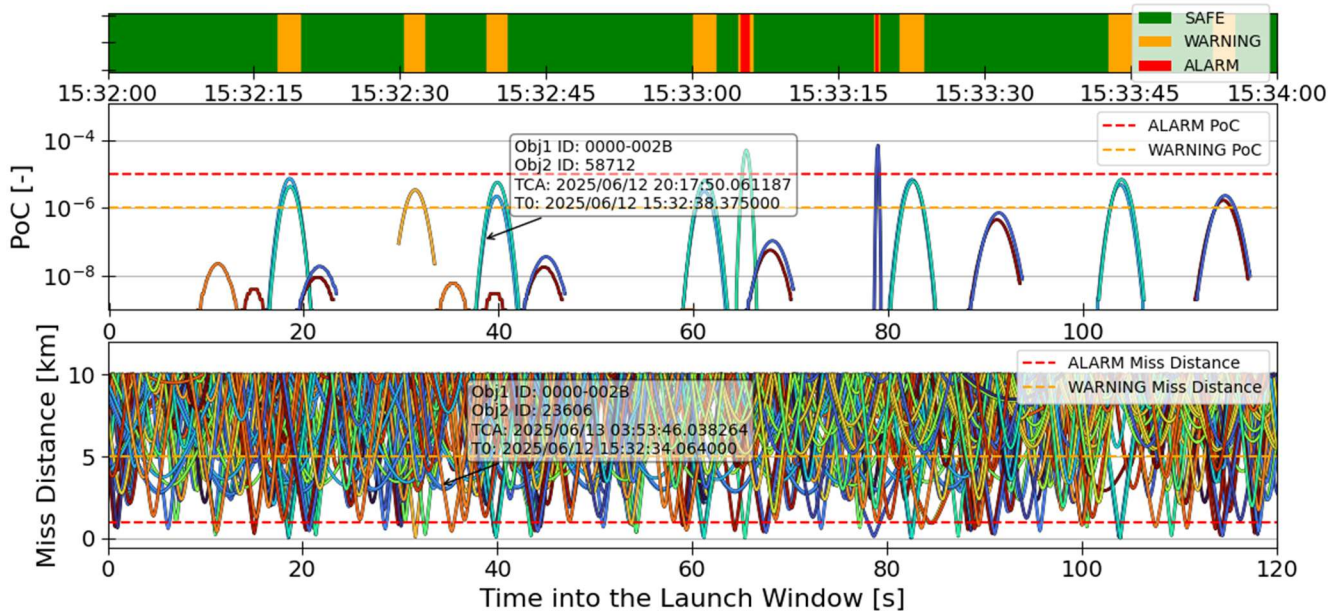

Figure 8: ASSET Topology Summary plot (close view on a 2-minutes launch window).

Table 4: Comparison of a miss-distance-based launch closure interval boundaries between discretized ASSET and ASSET Topology.

Interval Boundary	**Discrete ASSET**	**ASSET Topology**
WARNING start	15:04:17.273	15:04:17.273
WARNING stop/ALARM start	15:04:18.873	15:04:18.873
ALARM stop/WARNING restart	15:04:19.273	15:04:19.273
WARNING end	15:04:20.773	15:04:20.867

V. Conclusions and Outlook

This work presented the integration of a topology-based launch collision avoidance methodology within the ASSET software by DLR/GSOC, supporting the conventional discretized liftoff sampling approach with a continuous formulation in the (liftoff, mission elapsed time) domain. The time of closest approach search was reformulated as a two-dimensional minimization of the inertial primary–secondary miss distance and solved using a differential evolution optimizer.

Verification and validation against the discretized ASSET implementation and the in-house collision avoidance tool demonstrated that the evolutionary topology-based approach recovers all conjunctions identified by the discrete method while refining miss-distance estimates and substantially reducing computational cost. For the representative Sun-synchronous 500 km orbit test case considered, CPU time reductions of two orders of magnitudes were achieved relative to finely sampled discretized runs, while maintaining consistency in TCA epochs, miss distances, PoC values, and closure window boundaries. These results confirm that the continuous formulation overcomes the intrinsic limitation of discretized screening, namely the inability to guarantee coverage between sampled liftoff epochs within operationally acceptable CPU times, thereby improving both operational responsiveness and safety assurance.

For operational usage, a two-stage optimization strategy has been implemented, consisting of an initial coarse run with relaxed absolute tolerance to restrict the catalogue to potentially risky secondaries, followed by a refinement run with tighter tolerance on the reduced set. Furthermore, both window closure algorithms (PoC and miss distance based) are performed in parallel as the minimum miss distance in the topology does not necessarily coincide with the maximum PoC, due to the directional dependence of covariances. Current work focuses on the validation on extended launch windows, heterogeneous ascent profiles, and longer screening horizons to characterize worst-case runtime behaviour and scalability with growing catalogue sizes.

With this improved topology-based ASSET software, DLR is now providing a reliable, accurate and fast launch collision avoidance service to the growing launcher market.

VI. Acknowledgement

This research work has been supported by the German Federal Ministry of Defence through the technological research and development assignment “Responsive Space Capabilities”.

VII. REFERENCES

[1] ESA Space Debris Office, "Esa's annual space environment report", April 2025, edition 9.

[2] G. Falco, F. Toso, A. Ratcliffe, and S. Ortiz, "Launch collision avoidance sensitivity analysis", in Proceedings of the 8th European Conference on Space Debris. Darmstadt, Germany: European Space Agency, April 2021.

[3] J. Prézeau, P. Seimandi, C. Taillan, and M. Jouisse, "Collision risk assessment during launch and injection phases", in Proceedings of the 9th European Conference on Space Debris. Darmstadt, Germany: European Space Agency, April 2025.

[4] G. Di Campli Bayard de Volo, R. Kahle, A. Zollo, L. Sprengelmeyer, and S. Gruber, "Asset (ascent safety) – a new flight dynamics service for the safety of ascent trajectories and injection orbits," in International Spaceflight Dynamics Symposium 2024, Darmstadt, Germany, 2024.

[5] D. Oltrogge and S. Alfano, "Innovative Launch Collision Avoidance (LCOLA) tool prioritizing accuracy, launch access and efficiency", Journal of Space Safety Engineering, vol. 11, pp. 243–251, 2024. URL: https://doi.org/10.1016/j.jsse.2024.04.011.

[6] G. D. Bredvik and J. E. Strub, "Determination of acceptable launch windows for satellite collision avoidance", Astrodynamics 1991, pp. 345–356, 1992.

[7] A. Zollo, G. Di Campli Bayard de Volo, M. Weigel, S. Aida, R. Kahle, and J. F. San Juan Díaz, "Synthetic orbit uncertainty generation through regression analysis of historical conjunction data messages", Journal of Space Safety Engineering, vol.11, pp. 454–461, 2024. https://doi.org/10.1016/j.jsse.2024.06.001

[8] H. Klinkrad, Space Debris: Models and Risk Analysis. Springer Praxis Books, 2006.

[9] R. Storn and K. Price, "Differential evolution – a simple and efficient heuristic for global optimization over continuous spaces", Journal of Global Optimization, vol. 11, no. 4, pp. 341–359, 1997.

[10] S. Alfano, "Review of conjunction probability methods for short-term encounters", (AAS 07-148), 2007 Jan 28 – Feb 1.

VIII. APPENDIX

A. *Singular Orbital Configurations Demonstration*

(i) GEO-GEO Configuration:

For a perfectly geostationary satellite, the orbit is circular, equatorial, and has mean motion equal to the Earth's rotation rate Ω_E. In an inertial frame, the positions can be written as:

$$\boldsymbol{r}_s^{\mathrm{I}}(t) = R_z(\Omega_E t)\boldsymbol{r}_{s,0}, \quad \boldsymbol{r}_p^{\mathrm{I}}(t) = R_z(\Omega_E t)\boldsymbol{r}_{p,0},$$

where $R_z(\cdot)$ is the rotation matrix about the inertial z-axis and $\boldsymbol{r}_{s,0}$, $\boldsymbol{r}_{p,0}$ are constant vectors with $|\boldsymbol{r}_{s,0}| = |\boldsymbol{r}_{p,0}| = a_{\mathrm{GEO}}$. The relative position at time $t = t_0 + \tau$ is then

$$\boldsymbol{\Delta}(t_0,\tau) = \boldsymbol{r}_s^{\mathrm{I}}(t_0+\tau) - \boldsymbol{r}_p^{\mathrm{I}}(t_0+\tau) = \\ = R_z\big(\Omega_E(t_0+\tau)\big)\big(\boldsymbol{r}_{s,0} - \boldsymbol{r}_{p,0}\big).$$

Let $\boldsymbol{\Delta}_0 \coloneqq \boldsymbol{r}_{s,0} - \boldsymbol{r}_{p,0}$, then $f(t_0,\tau) = |\boldsymbol{\Delta}(t_0,\tau)|^2 = |\boldsymbol{\Delta}_0|^2$, which is constant for all (t_0,τ). Hence

$$\frac{\partial f}{\partial t_0}(t_0,\tau) = \frac{\partial f}{\partial \tau}(t_0,\tau) = 0, \quad \forall (t_0,\tau),$$

and all higher-order partial derivatives vanish as well.

(ii) Circular, coplanar, and equal-SMA Configuration:

Assume that both satellites move on circular orbits with the same semi-major axis a (same mean motion n) and inclination i. Because the topology is parametrised by the liftoff epoch t_0 and the mission elapsed time τ, changing t_0 corresponds to a rigid rotation of the full configuration about the Earth's rotation axis. As t_0 varies, the effective RAAN difference in the inertial frame can be written as

$$\Delta\Omega_{\mathrm{eff}}(t_0) = \Omega_s - \Omega_{p,0} - \Omega_E\big(t_0 - t_{0,init}\big),$$

where $t_{0,init}$ is the launch window start epoch, $\Omega_{p,0}$ is the primary's RAAN at $t_{0,init}$ and Ω_s is the secondary's RAAN. Therefore, there exists an infinite set of liftoff epochs $t_0^\star$ such that $\Delta\Omega_{\mathrm{eff}}(t_0^\star) = 0$, meaning that the two inertial orbital planes are coplanar.

For any such $t_0^\star$ and in a frame whose z-axis is orthogonal to the common orbital plane, their motions can be expressed as

$$\boldsymbol{r}_s^{\mathrm{I}}(t) = aR_z(nt)\boldsymbol{u}_s, \qquad \boldsymbol{r}_p^{\mathrm{I}}(t) = aR_z(nt)\boldsymbol{u}_p,$$

where $\boldsymbol{u}_s, \boldsymbol{u}_p$ are constant unit vectors in the plane. The relative position is then

$$\boldsymbol{\Delta}(t_0^\star,\tau) = \boldsymbol{r}_s^{\mathrm{I}}(t_0^\star+\tau) - \boldsymbol{r}_p^{\mathrm{I}}(t_0^\star+\tau) = \\ = a\,R_z\big(n(t_0^\star+\tau)\big)\,\big(\boldsymbol{u}_s - \boldsymbol{u}_p\big).$$

Let $\boldsymbol{\Delta}_0 \coloneqq a\big(\boldsymbol{u}_s - \boldsymbol{u}_p\big)$, then $f(t_0^\star,\tau) = |\boldsymbol{\Delta}(t_0^\star,\tau)|^2 = |\boldsymbol{\Delta}_0|^2$, which is constant in τ. Therefore,

$$\frac{\partial f}{\partial \tau}(t_0^\star,\tau) = 0 \quad \forall \tau,$$

and the second order derivatives vanish as well, apart from the second derivative with respect to the liftoff. In fact, dependence on t_0 is preserved, so the singularity is one-dimensional in (t_0,τ) −space, in contrast to the fully degenerate GEO-GEO case.