Acta Futura 11 (2018) 49-55
DOI: 10.5281/zenodo.1139240

Acta Futura

# GTOC9: Results from the Xi'an Satellite Control Center (team XSCC)

HONG-XIN SHEN*, TIAN-JIAO ZHANG, AN-YI HUANG, ZHAO LI

STATE KEY LABORATORY OF ASTRONAUTIC DYNAMICS, XI'AN SATELLITE CONTROL CENTER, XI'AN, CHINA

**Abstract.** This paper describes methods used by the team from the Xi'an Satellite Control Center (XSCC) for solving the 9th Global Trajectory Optimization Competition (GTOC9) problem. The removal of all 123 pieces of debris is accomplished using 12 launches in about 8 years time span, and the performance index finally ranked third in the competition. We refined our results after the competition, and the improved solutions are also presented.

## 1 Introduction

The 9th global trajectory optimization competition (GTOC9) problem concerned the multiple debris removal in low Earth orbit in order to relieve the Kessler effect [1]. This was accomplished by multiple missions, where each mission is a multiple-rendezvous spacecraft trajectory where a subset of size $N$ of the 123 orbiting debris is removed by the delivery and activation of $N$ de-orbit packages. The goal was to rendezvous as many debris objects as possible. A performance index, which depended on the number of launches used penalized by an added quadratic cost that including the sum of propellant mass and de-orbiting kits, must be minimized subject to a variety of constraints. Earlier submissions of single missions were rewarded through a smaller base cost. The only manoeuvres allowed to control the spacecraft trajectory are instantaneous changes of the spacecraft velocity (specific impulse was 340s); the tour should last less than 2947 days. Details can be found in Reference [1]. In this paper, we describe the main design model and optimization methods for this problem.

The complexities of this problem are to manage multiple-missions and select suitable sequence to rendezvous all debris objects from a given set of 123. Since the overall sequence of missions / debris removed is too large to be explored exhaustively, a global optimization procedure based on Ant Colony Optimization (ACO) is employed to automatically generate a near-optimal solution, as a replacement of the global search [2].

## 2 Leg cost estimation and optimization

Optimal phasing is assumed to obtain a preliminary estimate of the transfer velocity increment $\Delta V$. The Hohmann transfer cost for a small change of the semimajor axis $\Delta a$ reduces to $\Delta V/V = 0.5\Delta a/a$; similarly, for a small eccentricity change, the relation $\Delta e$, $\Delta V/V = 0.5\Delta e$ holds while $\Delta V/V = 2\sin(0.5\Delta i)$ applies for inclination changes. An empirical relation is thus introduced to consider simultaneously the change of semimajor axis $\Delta a$, to account for the additional eccentricity change $\Delta e$ as well as for the change of incli-

*Corresponding author. E-mail: hongxin.shen@gmail.com

nation $\Delta i$ [3]:

$$\Delta V'/V = 0.5\sqrt{(\Delta a/a)^2 + \Delta e^2} + 2\sin(0.5\Delta i) \quad (1)$$

where $\Delta V'$ is one part of the total velocity increment $\Delta V$ for a body-to-body transfer. The smallest value of semimajor axis $a$ among the two objects involved in the leg, and the corresponding circular velocity $V$ are used.

Suitable object sequences are found by estimating the rendezvous times and cost of each leg in terms of velocity change $\Delta V$. The estimation procedure assumes that favorable opportunities occur only when the required plane change is the smallest, and therefore when the RAAN of the chaser is close to that of the target. To this purpose, the mission can take advantage of the J2 perturbation, which changes the RAAN of bodies orbiting the Earth with a rate that depends on semimajor axis and eccentricity. Objects with different orbits will therefore have different rates of change of $\Omega$. The cost required for a small RAAN change $\Delta\Omega$ is

$$\Delta V''/V = 2\sin(0.5\Delta\Omega) \quad (2)$$

where $\Delta V''$ is the other part composing the total velocity increment $\Delta V$ for a body-to-body transfer. Therefore, one obtains the total cost of a leg as $\Delta V = \Delta V' + \Delta V''$.

The search starts from finding suitable times for the transfer between any debris pair (objects $j$ and $k$). Each leg starts in rendezvous conditions with object $j$. It is necessary to wait for the RAAN difference $\Delta\Omega$ between objects $j$ and $k$ to become minimal in a prescribed time range in order to perform the transfer with a small amount of propellant consumption. Assuming $t^j$ as the leg starting epoch, the rendezvous time for the next object $k$ is denoted with $t^{jk}$. According to the GTOC9 rules the transfer time for each leg ranges between 5days and 30days, taking servicing time into account. Thus, $t^j + 5.5 = t^{jk}_{min} \le t^{jk} \le t^{jk}_{max} = t^j + 30$, where we have reserved a buffer of 0.5days for the orbit transfer time. Considering all possible debris pairs, the times when this favorable condition occurs are computed as:

$$t^{jk} = \begin{cases} \text{solve } \Delta\Omega(t) = 0, \text{if } \Delta\Omega(t^{jk}_{min})\Delta\Omega(t^{jk}_{max}) < 0, \text{else} \\ t^{jk}_{min}, \text{if } |\Delta\Omega(t^{jk}_{min})| < |\Delta\Omega(t^{jk}_{max})| \\ t^{jk}_{max}, \text{otherwise} \end{cases} \quad (3)$$

These times represent the theoretical rendezvous times for legs connecting the two objects; they are valid both for transfer from object $j$ to object $k$ and vice versa. Multiple-leg missions can be built based on the theoretical rendezvous times that have been determined with the procedure described above. All the permutations of $m$ targets selected among the $n$ possible objects should be considered. Mission starts at $t_1$ from target 1; rendezvous with target $i+1$ at the end of the $i$-th leg ($i = 1, 2, \ldots, m-1$) is assumed to happen at the best orbit alignment, that occurs after $t_i$, between the orbit planes of targets $i$ and $i+1$. Because the number of targets and the dimension of the target set are relatively large, all the sequences cannot be evaluated in reasonable times. Pruning technique or an global optimization strategy is required to find the best sequences when the number of possible solutions becomes too large. Instead of pruning technique, an ant colony optimization approach is used in this paper to obtain an near-optimal solution, by solving a variant of the well-known travelling salesman problem.

The actual missions are designed acounting for the full dynamics complexity and constraints by means of an ant colony optimization approach in continuous domain [4], which is used to obtain the optimal solution, i.e., minimum $\Delta V$, for each single transfer leg.

Assume the $i$-th leg starts at time $\tau_1 = t_i$, which is known from the solution of the previous leg, taking servicing time into account. Impulses are applied at times $\tau_2 < \tau_3 < \tau_4 = t_{i+1}$. The rendezvous/last impulse time $\tau_4 = t_{i+1}$ is unknown and only an estimation is available. Each arc is described by a limited set of 6 variables. Three variables define impulse times: $p_1 = \tau_4 - \tau_1$, $p_2 = (\tau_2 - \tau_1)/(\tau_4 - \tau_1)$, $p_3 = (\tau_3 - \tau_2)/(\tau_4 - \tau_2)$; the time of flight $p_1$, $p_2$ and $p_3$ vary between 0 and 1. $p_4$ is the impulse velocity change $\Delta V_1$, which varies between 0 and 800 m/s. $p_5$ and $p_6$ define the direction of the first impulse.


DOI: 10.5281/zenodo.1139240

Given the set of variables it is possible to evaluate the arc $\Delta V$. Initial time, position and velocity are known. Once the velocity after the impulse has been evaluated from the optimization variables, Kepler's problem is solved taking J2 perturbation into account, to evaluate position and velocity just before the following impulse (2-); again, the optimization variables provide the velocity components after the impulse (2+), and position and velocity at 3- are determined solving a perturbed Kepler's problem. The arc from point 3 to point 4 (the target position at $t_4$ is also evaluated by solving a perturbed Kepler's problem) is first solved as an unperturbed Lambert's problem to obtain the velocity components after the second impulse that would allow to intercept the target in the absence of perturbations. J2 effect would not allow the intercept with these values, so they are corrected with an iterative scheme to nullify the error between the (perturbed) positions at $t_4$ of chaser and target. The scheme is based on Newton's method and employs the numerical derivatives of final position with respect to initial velocity components [5]. For the legs with relatively long transfer time, four impulses may be required, and the procedure can fit this situation easily with minor changes.

Each arc is solved in sequence, starting with the values obtained at the end of the previous one. It is important to note that the optimal strategy for favorable opportunities is often to wait on the initial orbit for a relatively long time, until the orbit planes have become sufficiently close. Estimations and results of the leg optimization have been compared for a large number of debris pairs. Differences in terms of rendezvous times are typically limited at 1 or 2 days. However, optimization of $\Delta V$ usually are remarkably smaller than the corresponding estimations, particularly for high-$\Delta V$ legs. Thus, we discount the estimation typically by $\Delta V = 0.7(\Delta V' + \Delta V'')$, where $\Delta V'$ and $\Delta V''$ are given by Eq. (1) and Eq. (2).

## 3 Formulation of a variant of TSP

We consider an Active Debris Removal(ADR) task with $n$ missions which are responsible for the removal of a given set $S = \{s_0, s_1, ..., s_N\}, N = 122$ of 123 orbiting debris over a time horizon $[0, 2947]$ with reference time 23467[MJD2000]. The ADR task can be represented by a graph $G =< S, E >$ where the nodes are defined by the debris objects. The directed edge $(s_i, s_j) \in S \times S, s_i \neq s_j$ corresponds to the orbital

**Algorithm 1** *ACO for the ADR problem*

Set parameters, initialize pheromone trails
**while** The stopping criterion is not met **do**
  **for** all ants $k = 1, \dots, K$ **do**
    set mission index: $m = 1$
    **while** there are unselected debris **do**
      $Can(\Pi^{p,k})$ is the set of all unselected debris objects
      choose a debris $s_i$ randomly from $Can(\Pi^{p,k})$ as the start one in the $m$-$th$ mission
      remove the forbidden debris according to both the time constraints and tour length limitation
      **while** $Can(\Pi^{p,k}) \neq \emptyset$ **do**
        choose a debris $s_j \in Can(\Pi^{p,k})$ with probability $p_{ij}^k$
        add the chosen debris to the $m$-$th$ sub-path $\Pi_m^{p,k}$
      **end while**
      $m = m + 1$
    **end while**
    employ the 2-$Opt$, $insertion$ and $swap$ operators to improve the combined sequence $\Pi^k$
    divide the combined sequence into sub-paths $\{\Pi_1^k, \Pi_2^k, \dots, \Pi_n^k\}$ according to either time constraints or length limitation
  **end for**
  **for** all ants $k = 1, \dots, K$ **do**
    evaluate the solution $f(\Pi_1^k, \Pi_2^k, \dots, \Pi_n^k)$
    update the best-so-far solution $\Pi_1^*, \Pi_2^*, \dots, \Pi_n^*$
  **end for**
  update pheromones on the best-so-far sub-paths
**end while**
**return** $\Pi_1^*, \Pi_2^*, \dots, \Pi_n^*$

DOI: 10.5281/zenodo.1139240 

transfer made by a spacecraft from object $s_i$ to object $s_j$. Moreover, each edge has an associated transfer cost $\Delta v_{ij}$ in terms of velocity change as well as $\Delta m_{ij}$ in terms of fuel usage, it is characterized by a number of factors: $\Delta v_{ij} \equiv \Delta v_{ij}(s_i, s_j, t_i, t_j)$, where $t_i$ and $t_j$, respectively, are the departure and the arrival date associated with the transfer. Thus, a single mission can be viewed as an open sub-path in the graph. To be feasible, a sub-path must pass through each selected debris object only once, the servicing time between two successive debris rendezvous within the same mission should be greater than 5.5 days and not exceed 30 days. Moreover, the time gap of at least 30 days must be accounted for any two missions, and all the mission events must be finished within the allowed time horizon. The goal is to design a series of sub-paths able to cover all the 123 nodes, while minimizing the sum of overall mission costs and the penalty term related to the possible incomplete removals. The ADR problem is analogous to the Traveling Salesman Problem(TSP) [6], but there are two main differences between them:

- Instead of a single closed path visiting all the nodes in TSP, the debris are covered by several sub-paths(mission). The length of sub-path cannot exceed the upper limit $\Delta v_{max}$. Therefore, the overall path length is measured as the sum of all sub-path lengths.
- Unlike the constant distances between node pairs in the TSP, the debris are moving with different precession rates. Accordingly, the time varying cost of going from one debris object to another makes the ADR problem time-dependent.

For the sake of clarity, we present an integer programming formulation for the ADR problem. It is based on two sets of binary decision variables, $y_{ik}$, designating the removal of debris $s_i$ to the $k$-$th(1 \le k \le n)$ mission by the value 1 (and 0, otherwise), and $x_{ij}, i, j \in \{0, 1, \dots, N\}$, determining the debris disposal sequence in a single mission. $x_{ij}$ takes the value 1 when the vehicle proceeds from debris $s_i$ to debris $s_j$. Clearly, the search space defines over two finite sets of discrete decision variables $\{x_{ij}, i, j \in \{0, 1, \dots, N\}\}, \{y_{ik}, i \in \{0, 1, \dots, N\}, k \in \{1, 2, \dots, n\}\}$. A solution in the ADR problem can be represented through a set of $N$ variables, each of which is associated with a debris object. Here, solution components are debris, and they are to be visited in the order appearing in the solution. The decision variables $x_{ij}, y_{ik}$ indicate the debris to be removed in what order and in which mission. Hence, the ADR problem can be stated as follows:

$$\begin{aligned}
\min \quad & cn + \alpha \sum_{k=1}^{n} \Big( \sum_{\forall s_i \in S} \sum_{\forall s_j \in S} x_{ij} y_{ik} \Delta m_{ij} \Big)^2 && (4) \\
\text{s.t.} \quad & \sum_{\forall s_i \in S} x_{ij} \le 1, \forall s_j \in S && (5) \\
& \sum_{\forall s_j \in S} x_{ij} \le 1, \forall s_i \in S && (6) \\
& \sum_{\forall s_i \in S} \sum_{\forall s_j \in S} x_{ij} y_{ik} = \sum_{\forall s_i \in S} y_{ik} - 1, \forall k \le n && (7) \\
& \sum_{\forall s_i \in S} \sum_{\forall s_j \in S} x_{ij} y_{ik} \Delta v_{ij} \le \Delta v_{max}, \forall k \le n && (8) \\
& 5.5 \le t_j - t_i \le 30, \text{if } \exists k \le n, \forall s_i, s_j \in S, x_{ij} y_{ik} y_{jk} = 1 && (9) \\
& t_j - t_i \ge 35, \text{if } \exists k \le n, \forall s_i, s_j \in S, y_{ik} + y_{jk} = 1 && (10) \\
& t_j \le 26414, \text{if } \exists k \le n, \forall s_j \in S, y_{jk} = 1 && (11) \\
& t_j \ge 23467, \text{if } \exists k \le n, \forall s_j \in S, y_{jk} = 1 && (12) \\
& x_{ij}, y_{ik} \in \{0, 1\}, \forall s_i, s_j \in S, \forall k \le n && (13)
\end{aligned}$$

 DOI: 10.5281/zenodo.1139240

where $c$ is the base cost of each mission ranging from 45 to 55 (increasing linearly during the competition time frame),$n$ is the number of missions, $\alpha$ is set to be $2.0\times10^{-6}$ according to the GTOC9 rules. Eq. (4) states the objective function, where the fuel usage $\Delta m_{ij}$ is computed from impulsive $\Delta v_{ij}$ using the Rocket equation. Eqs. (5) and (6) ensure that a selected debris cannot be removed more than once. Furthermore, Eq. (7) signifies that the adjacent chosen debris in one mission must be arrived by the spacecraft consecutively. From the graph theory point of view, Eqs. (5)-(7) make the selected edges compose a series of paths without loops. Eq. (8) is the length limit of velocity change for a single sub-path. Eqs. (9)-(12) define all the time constraints. Eventually, Eq. (13) imposes the restriction on the decision variables.

## 4 Ant colony optimization

Ant colony optimization, denoted as ACO, is a metaheuristic framework for solving static combinatorial optimization problem. It takes inspiration from the following behavior of ant species: ants are able to find the shortest path from their home to a food source over a period of time [7].

ACO solves an optimization problem by a construction graph and uses $K$ artificial ants to walk on the graph where $K$ is the size of ant colony. Each ant constructs a solution iteratively and its behavior is guided by pheromone and heuristic information. The construction graph for the sequence optimization problem is the static ADR mission graph, in which the nodes are debris and each edge represents a transfer between two debris objects. The ACO metaheuristic is shown in Algorithm 1. Its main iterative procedure consists of three steps. At first, each ant acts in the same manner: starting from an empty solution $\Pi^p = \emptyset$, the partial solution $\Pi^p$ is extended incrementally by adding a debris object as a solution component from available candidates $Can(\Pi^p) \subset S$ according to a biased probabilistic mechanism. In particular, if a debris $s_j$ has not been previously removed, it can be selected with the probability:

$$p_{ij}^k = \begin{cases} \frac{\tau_{ij}^\alpha \cdot \eta_{ij}^\beta}{\sum_{l \in Can(\Pi^{p,k})} \tau_{il}^\alpha \cdot \eta_{il}^\beta}, & \text{if } l \in Can(\Pi^{p,k}) \\ 0, & \text{otherwise} \end{cases} \tag{14}$$

where $Can(\Pi^{p,k})$ is defined as the set of debris objects that can be added to a sub-path by the $k$-$th$ ant without violating either the time constraints or the limitation of the sub-path length. $\tau_{ij}$ is the pheromone of edge $(i,j)$ which corresponds to the transfer from the debris $s_i$ to $s_j$. The heuristic information is chosen as $\eta_{ij} = \frac{\lambda_1}{\Delta v_{ij}} + \frac{\lambda_2}{\Delta t_{ij}}$, where $\lambda_1$ and $\lambda_2$ are nonnegative real parameters that control the relative importance between transfer cost and time. Furthermore, $\alpha$ and $\beta$ are positive real parameters that control the relative importance between the pheromone and the heuristic information. Subsequently, a local search procedure is employed to improve every constructed solution. At the last step, pheromone is updated as follows: all pheromone trails are decreased uniformly through pheromone evaporation as to allow ants to forget bad solutions. Then, the pheromone deposited, the mount of which is proportional to the quality of the solution, guides subsequent ants to search in the promising regions of the search space. In this way, edges, associated with components in promising solutions, are reinforced with additional pheromone gradually. The iterative process terminates when a stopping condition is satisfied. In order to improve exploitation capacity of ACO, we use a modified version of the MAX-MIN ant system with respect to the classic one. In our algorithm, we propose an enhanced local search strategy by employing the 2-$Opt$, $insertion$ and $swap$ operators in turn. Once all these operators cannot find a better solution, local search stops. Otherwise, it continues until the stopping condition is met. The illustration of these operators is shown in Figure 1.

## 5 Results

A summary of the final solution submitted during the competition timeframe and achieving an objective function of 821MEUR, is provided in Table 1. Each mission ends with a dry mass $m_{dry} = 2000kg$. Shortly after the competition ended, an improved solution reaching an objective function value of 766MEUR was obtained refining all the body-to-body transfers, as given in Table 2. Under the competition pressure and stimulated by the ideas provided by the winner solution that made use of a reduced number of launches, we experimented with an increased weight of launches in Eq.(4). By doing so, our algorithms returned got several 10-mission solutions, among which the best oneachieving an objective function of 698MEUR is summarized in Table

DOI: 10.5281/zenodo.1139240 

**TABLE 1.** *Submitted solution*

| Mission | Number of objects | Debris ID | Start MJD2000 | End MJD2000 | $\Delta V$, m/s | Cost* |
|---|---|---|---|---|---|---|
| 1 | 16 | 7,67,12,48,122,63,61,19,107,41,11,82,115,45,85,47 | 62.09 | 405.03 | 2433.44 | 71.49 |
| 2 | 10 | 58, 90, 51, 72, 69, 10, 66, 28, 52, 64 | 476.4 | 618.81 | 1844.81 | 62.12 |
| 3 | 12 | 86, 84, 103, 16, 121, 92, 49, 20, 27, 54, 23, 36 | 668.95 | 960.88 | 1758.58 | 61.95 |
| 4 | 11 | 8, 43, 9, 55, 95, 73, 14, 102, 39, 113, 110 | 991.04 | 1195.71 | 1957.82 | 63.31 |
| 5 | 11 | 83, 75, 35, 119, 24, 108, 37, 112, 104, 32, 114 | 1322.00 | 1506.76 | 3176.94 | 83.41 |
| 6 | 9 | 74, 50, 94, 21, 97, 79, 120, 109, 77 | 1603.00 | 1763.17 | 2390.13 | 67.34 |
| 7 | 12 | 62, 1, 40, 76, 89, 99, 0, 15, 87, 59, 98, 116 | 1793.35 | 2012.06 | 3148.97 | 82.79 |
| 8 | 9 | 93, 70, 31, 105, 46, 88, 118, 18, 117 | 2079.42 | 2227.17 | 2667.12 | 71.07 |
| 9 | 9 | 5, 106, 53, 33, 17, 60, 68, 80, 71 | 2257.17 | 2409.36 | 2371.04 | 67.33 |
| 10 | 10 | 6, 2, 65, 81, 96, 100, 30, 4, 34, 26 | 2439.36 | 2588.5 | 2608.18 | 70.97 |
| 11 | 6 | 3, 42, 44, 56, 78, 111 | 2619.67 | 2701.14 | 1264.68 | 57.63 |
| 12 | 8 | 101, 22, 29, 38, 25, 91, 13, 57 | 2731.98 | 2833.56 | 2186.65 | 64.66 |

*Base cost: 55 MEUR.

**TABLE 2.** *Improved submitted solution*

| Mission | Number of objects | Debris ID | Start MJD2000 | End MJD2000 | $\Delta V$, m/s | Cost |
|---|---|---|---|---|---|---|
| 1 | 16 | 67,12,48,122,7,63,61,19,107,41, 11,82,115,45,85,47 | 73.06 | 393.40 | 1803.65 | 65.78 |
| 2 | 10 | 90,58,51,72,69,10,66,28,52,64 | 431.67 | 642.00 | 1147.32 | 59.16 |
| 3 | 12 | 86,84,103,16,121,92,49,23,20,54,27,36 | 672.00 | 969.09 | 1278.77 | 59.31 |
| 4 | 11 | 8,43,9,55,95,73,14,102,39,113,110 | 1000.00 | 1195.89 | 1885.13 | 64.75 |
| 5 | 11 | 83,75,35,119,24,108,37,112,104,114,32 | 1302.45 | 1530.06 | 2368.30 | 71.02 |
| 6 | 9 | 94,74,21,79,50,120,97,109,77 | 1580.41 | 1764.24 | 1883.52 | 64.37 |
| 7 | 12 | 1,62,40,76,89,99,0,15,87,59,98,116 | 1794.24 | 2017.40 | 2128.08 | 66.77 |
| 8 | 9 | 93,70,31,105,46,88,118,18,117 | 2047.40 | 2227.40 | 2015.48 | 64.61 |
| 9 | 9 | 5,53,106,17,60,80,33,68,71 | 2257.40 | 2405.00 | 1877.61 | 65.50 |
| 10 | 10 | 4,2,65,81,96,6,100,30,34,26 | 2436.58 | 2590.00 | 2186.32 | 68.22 |
| 11 | 6 | 3,42,44,56,78,111 | 2620.00 | 2697.00 | 927.96 | 57.23 |
| 12 | 8 | 101,22,91,13,57,25,38,29 | 2733.19 | 2937.13 | 1331.01 | 59.46 |

**TABLE 3.** *10-mission solution*

| Mission | Number of objects | Debris ID | Start MJD2000 | End MJD2000 | $\Delta V$, m/s | Cost |
|---|---|---|---|---|---|---|
| 1 | 14 | 38,103,95,57,16,118,50,23,117,55,113,20,79,27 | 0 | 281.73 | 2713.2 | 74.99 |
| 2 | 17 | 88,77,31,104,48,39,91,21,11,70,63,47,8,82,45,7,41 | 311.73 | 705.50 | 2727.3 | 77.29 |
| 3 | 13 | 94,75,0,18,2,6,108,24,44,120,26,67,119 | 735.50 | 955.00 | 2195.6 | 66.61 |
| 4 | 9 | 13,114,96,74,46,32,105,83,89 | 985.00 | 1090.28 | 2374.0 | 67.16 |
| 5 | 10 | 64,30,66,28,69,14,93,90,19,9 | 1127.19 | 1274.32 | 2241.2 | 66.31 |
| 6 | 12 | 107,61,5,4,78,92,53,25,111,109,56,42 | 1304.32 | 1500.00 | 2528.3 | 70.81 |
| 7 | 17 | 84,51,36,1,40,62,54,99,122,35,76,85,98,59,15,121,112 | 1530.00 | 1899.74 | 2473.9 | 72.79 |
| 8 | 11 | 97,115,22,102,86,110,65,10,100,34,73 | 2067.92 | 2285.00 | 1967.0 | 63.53 |
| 9 | 16 | 3,17,43,60,80,106,12,52,71,116,68,33,58,72,37,49 | 2315.00 | 2662.36 | 2909.5 | 80.76 |
| 10 | 4 | 29,101,87,81 | 2692.36 | 2787.02 | 1501.5 | 58.33 |

 DOI: 10.5281/zenodo.1139240

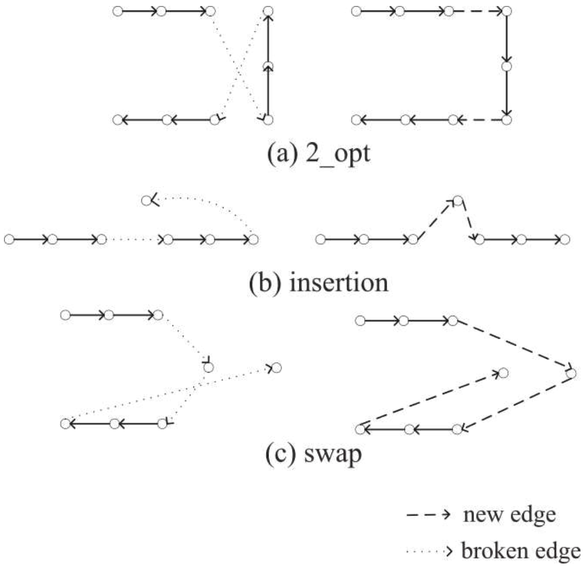


**FIGURE 1.** *The illustration of the local search operators, each of which transforms the left into the right. (a) 2-opt operator, (b) insertion operator, (c)swap operator.*

3. All the mission times in these tables refer to the permitted mission starting time i.e., 23467[MJD2000]. For fairness in the comparison, the maximum base cost, i.e., 55MEUR, is used for the above solutions.

## 6 Conclusions

The multiple debris rendezvous problem posed for the 9th edition of the GTOC is, essentially, a time-dependent combinatorial problem, and thus it can be casted into a dynamic variant to the standard Travelling Salesman Problem. Fortunately, multiple-revolutions transfers together with the time constraints considered in this particular competition allows for an accurate estimation of the single transfer cost and time, therefore the resulting dynamic TSP degenerates into a simpler version. As ant colony optimization ACO is one of the most acclaimed algorithms able tackle TSP problems [8], it was also implemented by the XSCC-ADL team to find near-optimal rendezvous sequences. An efficient local optimizer is also important to improve the overall search strategy adopted, as can be seen from the fact that we refined our submitted solution and got about 60MEUR performance index improvement only through local optimization. Finally, when we focused on the number of launches by adjusting the weight in the performance index, 10-mission solutions with remarkable performance improvement also can be found.

## References

[1] D. Izzo and M. Märtens. The Kessler Run: On the Design of the GTOC9 Challenge. *Acta Futura*, pages 11–24, 2018.

[2] Hong-Xin Shen, Tian-Jiao Zhang, Lorenzo Casalino, and Dario Pastrone. Optimization of Active Debris Removal Missions with Multiple Targets. *Journal of Spacecraft and Rockets*, pages 1–9, 2017.

[3] T. N. Edelbaum. Propulsion Requirements for Controllable Satellites. *ARS Journal*, 31(8):1079–1089, 1961.

[4] T. Liao, T. Stutzle, M. A. Montes de Oca, and M. Dorigo. A unified ant colony optimization algorithm for continuous optimization. *European Journal of Operational Research*, 234(3):597–609, 2014.

[5] L. Casalino and D. Pastrone. Active Debris Removal Missions with Multiple Targets. In *AIAA Paper 2014-4226*, 2014.

[6] D. Izzo, I. Getzner, D. Hennes, and L.F. Simoes. Evolving Solutions to TSP Variants for Active Space Debris Removal. In *Proceedings of the 17th annual conference on Genetic and evolutionary computation (GECCO 2015)*, 2015.

[7] M. Dorigo and L. M. Gambardella. Ant Colony System: A Cooperative Learning Approach to the Traveling Salesman Problem. *IEEE Transactions on Evolutionary Computation*, 1(1):53–66, 1997.

[8] L.F. Simoes, D. Izzo, E. Haasdijk, and A.E. Eiben. Multi-rendezvous Spacecraft Trajectory Optimization with Beam P-ACO. In *European Conference on Evolutionary Computation in Combinatorial Optimization*, 2017.