A Genetic Algorithm to associate optical measurements and estimate orbits of geocentric objects

M. Zittersteijn ⋆†, A. Vananti⋆, T. Schildknecht⋆, J.C Dolado-Perez⋆⋆ and V. Martinot⋆⋆⋆
⋆Astronomical Institute of the University of Bern (AIUB)
Sidlerstrasse 5, 3012, Bern, Switzerland
⋆⋆Centre National d'Etudes Spatiales (CNES)
18 Avenue Edouard Belin, 31400 Toulouse, France
⋆⋆⋆ Thales Alenia Space (TAS)
26 Avenue Jean François Champollion, 31100 Toulouse, France
†Corresponding author: michiel.zittersteijn@aiub.unibe.ch

Abstract

Cataloging geocentric objects can be put in the framework of Multiple Target Tracking (MTT). Current work tends to focus on the $S = 2$ MTT problem because of its favorable computational complexity of $O(n^2)$. The MTT problem becomes NP-hard for a dimension of $S \geq 3$. The challenge is to find an approximation to the solution within a reasonable computation time. To efficiently approximate this solution a Genetic Algorithm is used. The algorithm is applied to a simulated test case. These results represent the first steps towards a method that can treat the $S \geq 3$ problem efficiently and with minimal manual intervention.

1. Introduction

Since the dawn of the Space age in the 1960s the population of geocentric man made objects has been steadily increasing. It is only in the past decade or so that the community has started to realize that the population is reaching a point where it poses a risk to the active satellites in Earth orbit. This risk has been proven to be non negligible with events such as the Cosmos-Iridium collision in 2009. Currently several thousands of geocentric objects are being tracked with the use of both optical and radar sensors.

The problem of cataloging geocentric objects can be put in the more general framework of Multiple Target Tracking (MTT). A thorough discussion of the different approaches to solve this problem and the difficulties encountered can be found in [11]. One intrinsic property of the MTT problem is that no single observation provides enough information to enable a full state estimate of the object. Therefore the problem of MTT is twofold. Both the observations have to be correctly associated to one another, and the states (in this case six orbital elements) of the resulting objects have to be estimated. In the context of cataloging geocentric objects part of the observations are typically made with a survey type observation scheme. After the observations are collected the problem is to match two or more sets of observations to each other. Such as m objects from one set (can be single observations or already cataloged objects) to n objects in another set. The only way to optimally solve the MTT problem is to try all the different combinations of observations and perform an orbit determination for each of these hypothetical objects. This quickly becomes computationally unfeasible. Therefore the MTT problem is said to have a dimension S. When the dimension is set to $S = 2$ only two observation epochs are considered at a time. This makes for the smallest amount of possible combinations (only pairs) which have to be considered. When $S \geq 3$ the number of possible combinations rapidly increases and the problem is said to become NP-hard [11 18]. Generally there are two ways to approach the MTT problem. Either the two parts of the problem are treated separately (then $S = 2$ for the association part) or a global method is used. A global method aims to find both, the correct associations, and the orbits in a joint manner. Currently there are efforts made on both approaches. Good examples of state of the art methods that treat the problem in two separate steps are the Direct Bayesian Admissible Region approach [6] and the Optimized Boundary Value Admissible Region approach [16]. These methods aim to correctly associate closely spaced sets of observations (a.k.a. tracklets) to each other, after which a refined orbit determination should be made (e.g. by using the Least Squares technique). The advantage of these techniques lies with the favorable computational complexity of $O(n^2)$. Disadvantages include the fact that a minimum of information is used on which to base a hard decision to associate tracklets. Another disadvantage is that these methods can only

treat pairs of tracklets, and cannot consider the association between an already cataloged object and a new tracklet. Although the global approaches are challenging from a computational point of view, several methods have been developed in recent years. A popular and well known algorithm is the Multiple Hypothesis Tracking (MHT) algorithm[4]. This algorithm enumerates all the possible combinations of tracklets and evaluates them. In order to keep the computational complexity reasonable the problem has to be greatly simplified though. Other approaches are often based on a probabilistic approach, such as the Adaptive Entropy-based Gaussian-mixture Information Synthesis (AEGIS) - Finite Set Statistics (FISST) method[8], and Markov Chain Monte Carlo (MCMC) computations[15]. The primary advantage of these methods is that they do not work with hard association decisions. This means that the associations can be based on more information than just two tracklets. The drawback of these methods is clear: the computational burden is significantly higher than for the case where $S = 2$. Ideally a method is required that can solve the MTT problem for a dimension of $S \geq 3$ but with a realistic (polynomial) complexity. By definition an NP-hard problem cannot be solved optimally in a polynomial time. The common approach to these type of problems is to seek an approximation to the optimum solution, which can be found in a reasonable computation time. To address this need the authors developed an algorithm that is based on a Genetic Algorithm (GA)[7] and employs an orbit determination technique that is based on the previously developed Optimized Boundary Value method[16]. This paper presents the methods in more detail and the results found based on the application of the algorithm to simulated data.
The first section of this paper describes the method of orbit determination used in this work. The second part is dedicated to the Genetic Algorithm and its relevance in the MTT framework. In the third section the algorithm is applied to simulated optical measurements and the results are discussed. Finally the conclusions are discussed in the fourth section.

2. The Optimized Boundary Value Orbit Determination (OBVOD) Approach

Originally the Optimized Boundary Value Method[16] was developed for the data association problem. In that problem only the associations are sought, after this a separate step is needed to perform the orbit determination for a set of more than two tracklets. In this section the Optimized Boundary Value Method is explained as well as its extension that allows for an orbit determination for more than two tracklets.
In the optical domain it is common practice to work with *tracklets* instead of single observations. A tracklet is made up of a series of single observations, typically spaced several tens of seconds from each other. With this series of position measurements an estimate can be made of the angular rate of the object. If we assume that the tracklet is short enough these angular rates can be estimated by fitting a straight line through the observation series. This line fit is made with the least squares parameter estimation technique. This gives four observables per tracklet, namely the right ascension, declination, rate of right ascension and the rate of declination.
With these four quantities per tracklet, any given pair of tracklets has a set of four derived angular rates.

$$\dot{z} = \left(\dot{\alpha}_i, \dot{\delta}_i, \dot{\alpha}_j, \dot{\delta}_j\right) \tag{1}$$

Here the $\dot{\alpha}$ denotes the rate of the right ascension, and $\dot{\delta}$ is the rate of the declination. The subscripts i and j denote the tracklet number of the first and second tracklet considered in the correlation process.
A hypothesis $\mathbf{p} = \left(\rho_i, \rho_j\right)$ is now made on the ranges towards the two epochs at which the tracklets were observed. The angular positions together with the hypothesized ranges make up two geocentric positions at the two tracklet epochs. This is the classic Lambert problem, and several methods have been developed to solve it[3]. By solving the Lambert problem an orbit can be found that perfectly intersects these two geocentric positions. This computed orbit leads to the computed angular rates.

$$\hat{\dot{z}} = \left(\hat{\dot{\alpha}}_i, \hat{\dot{\delta}}_i, \hat{\dot{\alpha}}_j, \hat{\dot{\delta}}_j\right) \tag{2}$$

In the Optimized Boundary Value Method it is assumed that there is a Gaussian noise on the measured angular positions. Consequently there is also a Gaussian noise on the angular rates. Therefore the likelihood that the observed angular positions together with the hypothesis $\mathbf{p}$ can correctly represent the derived angular rates $\dot{z}$ is given by the normal distribution in Equation 3.

$$P\left(\dot{z} \mid z, \mathbf{p}\right) = \mathcal{N}\left(\dot{z}, \hat{\dot{z}}, \Sigma_{\dot{z}}\right) \tag{3}$$

Here the $\Sigma_{\dot{z}}$ contains the sum of the uncertainties of both, the random variables $\dot{z}$, and $\hat{\dot{z}}$ as in Equation 4.

$$\Sigma_{\dot{z}} = C_{\dot{z}} + C_{\hat{\dot{z}}} \tag{4}$$

The uncertainties of the derived angular rates $C_{\dot{z}}$ follow from the least squares fit. For the computed angular rates the uncertainties on the angular positions are propagated, as shown in Equation 5.

$$C_{\hat{\dot{z}}} = \left(\frac{\partial \hat{\dot{z}}}{\partial z}\right)^T C_z \left(\frac{\partial \hat{\dot{z}}}{\partial z}\right) \tag{5}$$

In Equation 5 the C_z contains the uncertainties of the measured angular positions. Note that the partial derivatives cannot be computed in an analytical manner. In order to find the impact that the change of an angular position has on the computed angular rate, a Lambert problem has to be solved. This makes that the $C_{\hat{\dot{z}}}$ is quite expensive to compute. In practice it has been found that this part of the total uncertainty is often negligible (a factor 1000 smaller than $C_{\dot{z}}$). Only in the case where the tracklets are spaced closely to each other does this part of $\Sigma_{\dot{z}}$ play a significant role. The exact limit where the $C_{\hat{\dot{z}}}$ starts to be significant is not yet studied. In the work presented in this paper the tracklets are spaced about two hours from each other, in this case it is safe to assume that the $C_{\hat{\dot{z}}}$ is negligible.

Given these observables and assumptions, the optimum hypothesis has to be found. This is the hypothesis that maximizes the likelihood given in Equation 3. As is common practice in the MTT domain, the negative log-likelihood can be taken of this quantity [1]. This gives the cost function as given in Equation 6.

$$L(\dot{z} \mid z, \mathbf{p}) = \frac{1}{2}\left(\dot{z} - \hat{\dot{z}}\right)^T \Sigma_{\hat{\dot{z}}}^{-1} \left(\dot{z} - \hat{\dot{z}}\right) + \frac{1}{2}\ln\det\left(2\pi\Sigma_{\dot{z}}\right) + \ln(k) \tag{6}$$

In Equation 6 the factor k accounts for all the exterior probabilities to consider. Examples of these are the false alarm and missed detection probabilities. If furthermore we assume that the $\Sigma_{\dot{z}}$ stays constant during the search for $\mathbf{p}^*$ the equation reduces to the loss function used in [16]. Also the constant is dropped from the loss function since it does not affect the final hypothesis found.

$$L_{\dot{z}}(\dot{z} \mid z, \mathbf{p}) = \left(\dot{z} - \hat{\dot{z}}\right)^T \Sigma_{\hat{\dot{z}}}^{-1} \left(\dot{z} - \hat{\dot{z}}\right) \tag{7}$$

The quantity given in Equation 7 is also known as the Mahalanobis Distance. A favorable characteristic of this distance is that it is distributed according to a χ^2 distribution. In the two dimensional tracking problem this is of great benefit because a predetermined threshold value can be used to gate the tracklet pairs. Whenever the Mahalanobis Distance falls beneath this threshold the tracklet pair is correlated. The topography of the loss function is smooth and contains one minimum point for a given regime of orbital periods [16]. In the work presented here only tracklets that were observed in the same night will be considered. Therefore the only regime of orbital periods that we are interested in is the [0, 1] domain (assuming near geosynchronous orbits). In Figure 1 an example of the loss function topography can be seen. Here the two tracklets belong to the same geosynchronous object (NORAD ID: 33749).

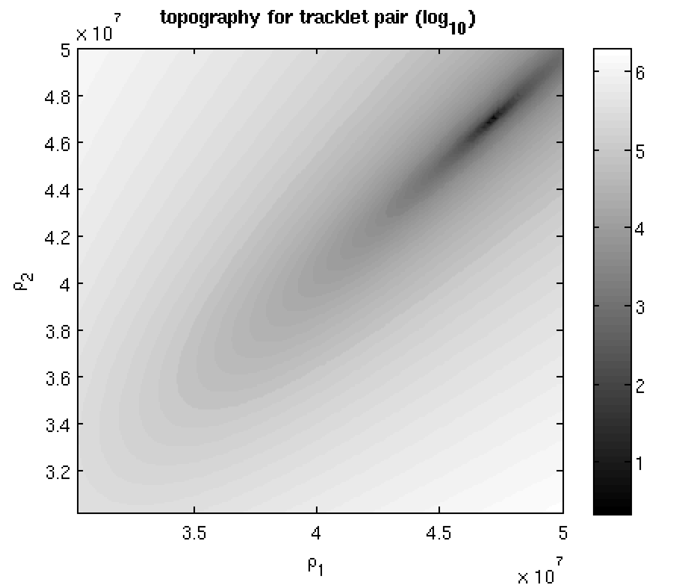

Figure 1: The $\log_{10}$ of the loss function for two tracklets that belong to the same object.

The Optimized Boundary Value Method can readily be extended to consider more than two tracklets. One part of the extension is straightforward. The vector of angular rates $\dot{z}$ will simply grow to include the rates of all the tracklets used. This is also the case for the covariance matrix Σ. The major difference between the two tracklet and the multiple tracklet case is that there is now information on the angular positions that can be used as well. When only looking at two tracklets the solution given by solving the Lambert problem will always go perfectly through the two angular positions. Therefore these measured angular positions are of no use in the two tracklet case. However when more than two tracklets are considered there will be a difference between the observed and computed angular positions for every tracklet except for the two tracklets used in the Lambert problem. Searching for the best hypothesis by using the angular positions of all tracklets other than the two that are used in the Lambert problem is reminiscent of the statistical ranging method[10]. A similar loss function as in Equation 7 can be set up for the angular positions.

$$L_z(z^\star \mid z, \mathbf{p}) = (z - \hat{z})^T \, \Sigma_z^{-1} \, (z - \hat{z}) \tag{8}$$

In Equation 8 the $z^\star$ denotes the set of the tracklets excluding the tracklets that are used in the Lambert problem. Throughout this work the first and last tracklet in a given set of tracklets are used in the Lambert problem. Figure 2 shows the topography of this loss function for a set of four tracklets, all belong to the same object (33749).

It is seen that again the topography is smooth and has one global minimum.
With these two parts in place a method can be derived that uses both the information on the angular rates as well as that of the angular positions. Equation 9 shows how the two original likelihoods can be combined to form a total likelihood.

$$P\left(z^\star, \dot{z} \mid z, \mathbf{p}\right) = \mathcal{N}\left(\dot{z}, \hat{\dot{z}}, \Sigma_{\dot{z}}\right) \mathcal{N}\left(z, \hat{z}, \Sigma_z\right) \tag{9}$$

Through a similar derivation as above we arrive at the following:

$$L\left(z^\star, \dot{z} \mid z, \mathbf{p}\right) = \left(\dot{z} - \hat{\dot{z}}\right)^T \Sigma_{\dot{z}}^{-1} \left(\dot{z} - \hat{\dot{z}}\right) + (z - \hat{z})^T \, \Sigma_z^{-1} \, (z - \hat{z}) \tag{10}$$

In this equation it is assumed that the Σ will stay constant for the positions and the rates during the search process. Figure 3 shows the topography of this combined loss function. The combined loss function is dominated by the loss function of the angular positions. Table 1 is an example of the steps made during the search for the optimum point in the combined loss function topography. Both the loss function for the rates, as well as for the positions, are listed. This search has been performed with the Broyden-Fletcher-Goldfarb-Shanno gradient descent method.

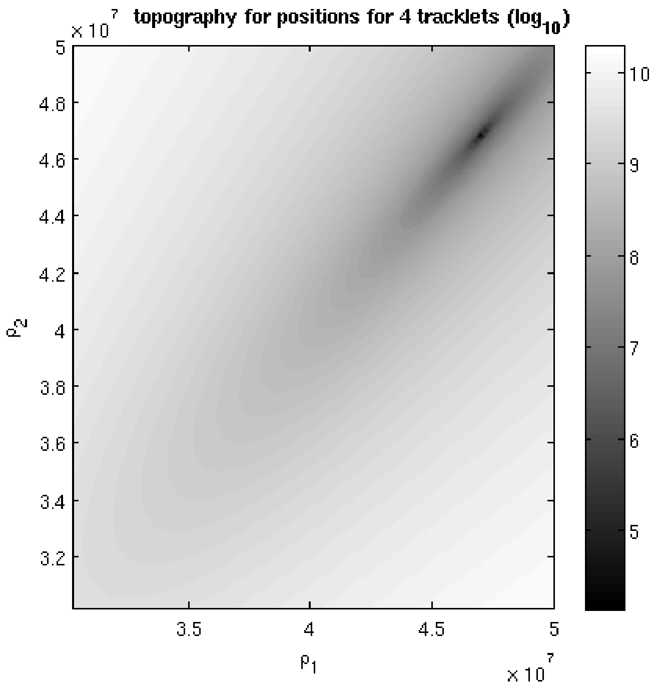

Figure 2: The $\log_{10}$ of the loss function on the angular positions for four tracklets that belong to the same object.

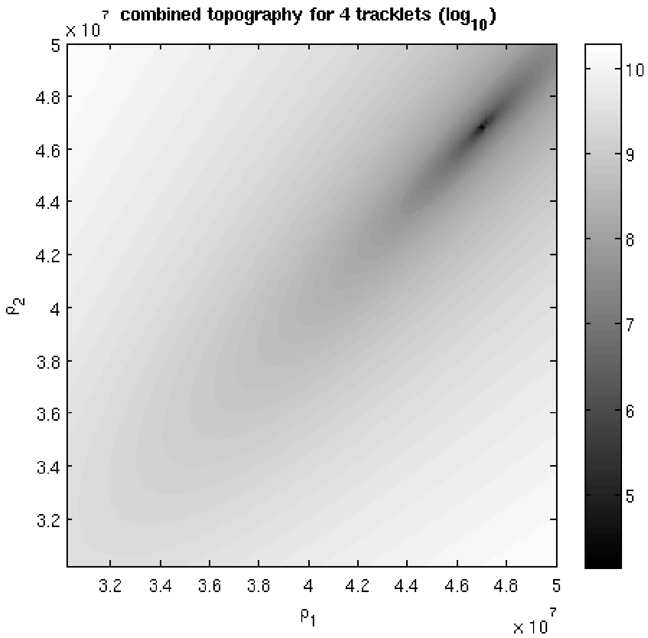

Figure 3: The $\log_{10}$ of the combined loss function for four tracklets that belong to the same object.

Table 1: Steps taken in the combined topography by the optimization scheme

rho1 [m]	rho2 [m]	L_z [-]	$L_{\dot{z}}$ [-]
4.4490E7	4.6278E7	3.4668E7	2.1315E4
5.2590E7	5.3342E7	1.2596E7	1.6248E4
4.7823E7	4.7785E7	3.0905E5	1.0954E3
4.6268E7	4.6209E7	1.7216E5	1.8733E2
4.6901E7	4.6906E7	1.8676E3	24.870
4.6832E7	4.6840E7	9.586	9.3328
4.6827E7	4.6835E7	2.2457E-2	8.5078
4.6827E7	4.6835E7	1.4925E-2	8.5145
4.6827E7	4.6835E7	1.4924E-2	8.5145
4.6827E7	4.6835E7	1.4925E-2	8.5145

From Table 1 it can be found that the optimum point for the two loss functions (Equations 7 and 8) do not coincide. This raises the question whether the two parts of the loss function should be treated equally.

One of the main points to be investigated further is the weighting of the two parts of the loss function. Other aspects that will need further attention are the computational costs involved and the validity of the assumptions that have been made.

3. Genetic Algorithm in Multiple Target Tracking

We can identify two main classes of MTT methods, one being the single frame methods, and the other being the multiple frame methods [11]. Single frame methods match m objects from one set (e.g. a catalog or group of tracklets) to n objects of a second set. An example of a single frame method is the nearest neighbor method. Multiple frame methods consider the data association between more than two data sets. A well known multiple frame method is the Multiple Hypothesis Tracking (MHT) method [4]. The advantage in using multiple frame methods, is that we can postpone the decision to correlate two tracklets. This postponing of a decision helps to identify false alarms and will decrease the number of false associations between tracklets. Therefore these methods will lead to a significant increase in performance when applied to an environment with a high density of targets. However, the downside to using multiple frame methods is the computational cost.

In the fields of computer science and mathematics one of the biggest unanswered questions is whether the statement P=NP holds. P and NP are classes in which a problem can be placed based on its complexity. P stands for Polynomial time, so it is solvable in $t = n^k$ where n is the dimension of the problem and k a fixed factor. In general these problems are thought to scale well with their size, and can be solved in a reasonable time (although k can still be very large). On the contrary, an NP problem can only be *checked* in polynomial time. So given a certain answer, we can see if this is correct or not rather quickly. However computing the answer can involve an enormous computation time and a bad scaling (as an example $t = k^n$).

The common approach to solving an NP-hard problem is to approximate the solution. This approximation of sufficient quality can perhaps be found within a reasonable computation time. To this end several algorithms have been developed, some of which have been applied to the MTT problem [11]. Examples of this are the Tabu search [17], Lagrangian relaxation [12], the Greedy Randomized Adaptive Search Procedure (GRASP) [14] and the Genetic Algorithm (GA) [18,5]. Of these algorithms the GRASP and Lagrangian relaxation algorithms are the most developed ones. As the performance of the GA has not been investigated extensively in the context of MTT it represents an interesting subject of study.

A flowchart of the GA is presented in Figure 4.

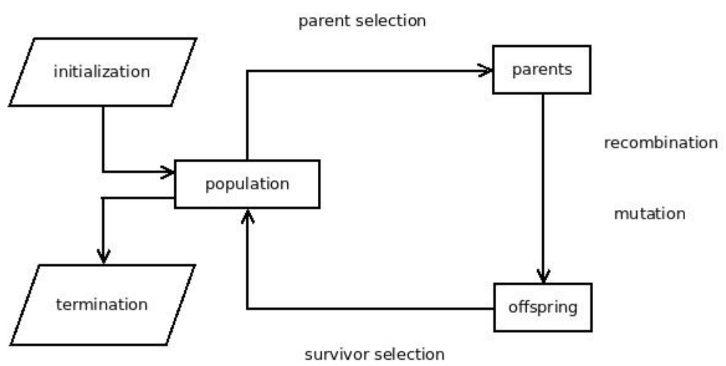

Figure 4: Flowchart of a Genetic Algorithm

Before the concepts behind each of the elements in Figure 4 can be discussed, the notions of an individual and a population have to be clarified. The GA is a population based algorithm. This means that it does not work with a single point in the search space. Instead it works with a group of points, also called individuals. These individuals together make up the population. The strength of a GA lies in the fact that it uses the information contained in the whole population. An individual must represent a valid solution to the problem that is being solved, therefore the representation of this solution is an important choice when designing a GA. In this work it is decided to use the so-called *k-matrix*, as it was introduced in [15]. A k-matrix contains all the information needed about the correlations among the tracklets. It is a lower triangular matrix as is shown in Equation 11.

$$K = \begin{pmatrix} 1 & 0 & \dots & 0 \\ k_{2,1} & k_{2,2} & \dots & 0 \\ \dots & \dots & \ddots & 0 \\ k_{i,1} & k_{i,2} & \dots & k_{i,j} \end{pmatrix} \tag{11}$$

The subscripts i and j denote the row and column respectively. An entry $k_{i,j}$ can only have a value of zero or one. This matrix can be understood better by viewing its columns as hypothetical objects, which consist of the tracklets where a "1" is in the row. The fact that it is lower triangular is user defined and comes from the definition that the first tracklet will always be associated to the first object, the second tracklet will belong to the second object etc. Furthermore, since each tracklet can only be associated to one object at a time the following constraint holds.

$$\Sigma_{j=1}^{N} k_{i,j} = 1 \quad \forall i \tag{12}$$

In Equation 12 the N denotes the total number of tracklets (which is the same as the total number of columns). Each individual in the population is thus a k-matrix, and represents a valid way in which to associate the tracklets.

The challenge in designing a GA lies mostly in the design of the fitness function. There are only two general requirements of a fitness function. One is that the fitness function should improve when the algorithm moves closer to the optimum solution. The second requirement is that the best fitness individual should coincide with the true solution. The fitness function used here is based on the loss function as given by Equation 10. The main difference between the fitness function and the previous loss function is that we cannot ignore the $\ln(k)$ term any longer. This is because we do not only compute the orbit, instead we now look at the total likelihood of a certain k-matrix being correct. Therefore the missed detection and false alarm probabilities should be taken into account. There are two cases to consider. Either an object consists of one tracklet only, or it consists of two or more tracklets. Let us first discuss the $N \geq 2$ scenario. Here enough information is present to evaluate the loss function as given by Equation 10. The manner in which the missed detection probability and the false alarm probability should be taken into account is not fully clear yet. Therefore instead of working with these probabilities we introduce two constant factors k_1 and k_2. Note that in any case the missed detection probability and the false alarm probability are difficult if not impossible to quantify correctly. They are influenced by many factors that are very different in nature, such as the weather, the sensor type and the image processing software used.

It is assumed that a survey type observation strategy is used for the collection of the observations. In such a strategy several *fences* are used to construct an observation scheme where geosynchronous objects should be re-observed in each of the fences. A fence is a region of the sky with fixed right ascension and varying declination values. The observation strategy used here will be explained in more detail in the following section. Equation 13 describes the total likelihood that a set of tracklets belong to the same object, and that the object was not observed in one or more of the fences. In this equation the k_1 factor serves as a penalty that is induced each time an object is not observed in a given fence.

$$P_{N\geq 2}\left(z^{\star}, \dot{z} \mid z, \mathbf{p}\right) = \mathcal{N}\left(\dot{z}, \hat{\dot{z}}, \Sigma_{\dot{z}}\right) \mathcal{N}\left(z, \hat{z}, \Sigma_{z}\right) k_1^{L-N} \tag{13}$$

In Equation 13 the L is the total number of fences. Again, taking the negative logarithm of this function and making the same assumptions as previously made we arrive at Equation 14 .

$$F_{N\geq 2}\left(z^{\star}, \dot{z} \mid z, \mathbf{p}\right) = \left(\dot{z} - \hat{\dot{z}}\right)^T \Sigma_{\dot{z}}^{-1} \left(\dot{z} - \hat{\dot{z}}\right) + \left(z - \hat{z}\right)^T \Sigma_{z}^{-1} \left(z - \hat{z}\right) - (L - N) \ln\left(k_1\right) \tag{14}$$

When there is only one tracklet to consider it can mean two things. Either the tracklet belongs to an object that has only been observed once, and has been missed in all other fences. Or it is a spurious measurement. Therefore it induces the penalty of k_1 for all fences except one, as is the case in Equation 14. An additional penalty is given for the case where it is a spurious measurement, this factor is denoted by k_2.

$$F_{N=1} = -\ln\left(k_2 + k_1^{(L-1)}\right) \tag{15}$$

This fitness function is applied to each object in a given k-matrix. At the end all the individual fitness values are summed to arrive at the total fitness of the k-matrix. Recall that summing the negative log likelihood expressions is the same as multiplying the probabilistic expressions as given in Equation 13. Note that this fitness function has to be minimized in order to derive the optimum solution.

The initialization of the population is done at random. Random initialization ensures a good first distribution of the individuals over the search space. Parent selection is performed randomly based on the relative fitness of the individual. These two individuals (a.k.a. the parents) are then used in the recombination process. Here the two individuals are combined to make two new individuals (a.k.a. the offspring). This recombining can be done in numerous ways. Here the common two-point crossover operator is used. In the two-point crossover operator two rows in the k-matrix are chosen at random, the resulting three parts of the k-matrices are exchanged between the parents to form the offspring. These new individuals are then subjected to the mutation operator. The mutation operator changes an individual at random, this is to ensure a certain amount of diversity within the population. A mutation probability p_{mute} is defined. For every row a random number is generated, if this number is lower than p_{mute} that row will be mutated. A row is mutated by randomly changing the column which has a value of "1", this is equivalent to randomly reassigning that tracklet to another object. This process of selecting parents and creating the offspring is repeated until a complete new population is produced. In a standard GA the whole population is now replaced by the new one. Another option is to use an *elitist* GA scheme (EGA). Here a certain percentage of the best individuals in the population is copied to the new population, replacing the worst individuals in the new population.

4. Application and Results

Now that both ingredients to the algorithm are discussed it can be applied to a test case. The algorithm is applied to simulated optical measurements. The objects that are used come from the TLE-Catalog[i] of the geosynchronous population. Each object is propagated during one night and a survey type observation scenario is used to collect the observations of this population. The sensor used in this survey has a field of view of 4.1°, which is the same as the ZimSMART telescope used for survey purposes at the Zimmerwald observatory. For each fence there are five fields of view placed in a strip, centered on the geostationary ring. In each field of view seven images are taken spaced 30 seconds from each other. Apart from collecting the necessary observations this ensures that each object has enough time to pass through the field of view before that field is revisited by the telescope. Therefore there will be no two tracklets of the same object observed in the same fencepost. This is an important aspect of the survey design, as it allows us to only consider the correlation of tracklets between different fences and ignore those within the same fence. After taking the seven observations in a given field of view the telescope moves to the next one. After the last field of view is reached at the bottom of the fence it moves back up to the first one again. Each fence is scanned in such a manner for two hours, after which the fence is moved two hours in right ascension. By doing this we attempt to re-observe objects in each fence. In Figure 5 the collected observations of four of the objects can be seen. The observer position is set at Zimmerwald observatory, but since no visibility conditions were taken into account the location of the observer is of little relevance.

Note that such an observation strategy can be carried out by one or several sensors. The only aspect that is relevant to the algorithm is the number of fences. Which sensor(s) and at what sites they are placed on is not of importance to finding the correct associations. The test case consists of four objects that have each been re-observed in all four fences. This means that this is a four dimensional ($S = 4$) problem with a total of 16 tracklets. The four objects have the following NORAD IDs: 33749, 21227, 4297, 8132. All four are geosynchronous objects, their orbital elements are listed in Table 2.

Table 2: Orbital elements of the four objects tracked

NORAD ID	a [m]	e [-]	i [deg]	Ω [deg]	ω [deg]	M [deg]
33749	4.2242E+07	2.148Ee-4	0.0261	23.852	34.596	201.70
21227	4.2241E+07	5.630E-5	9.1759	49.050	139.44	78.015
4297	4.2241E+07	6.162E-4	8.7786	313.31	329.56	32.376
8132	4.2241E+07	3.764E-4	13.056	328.86	182.34	194.97

[i] www.space-track.org/

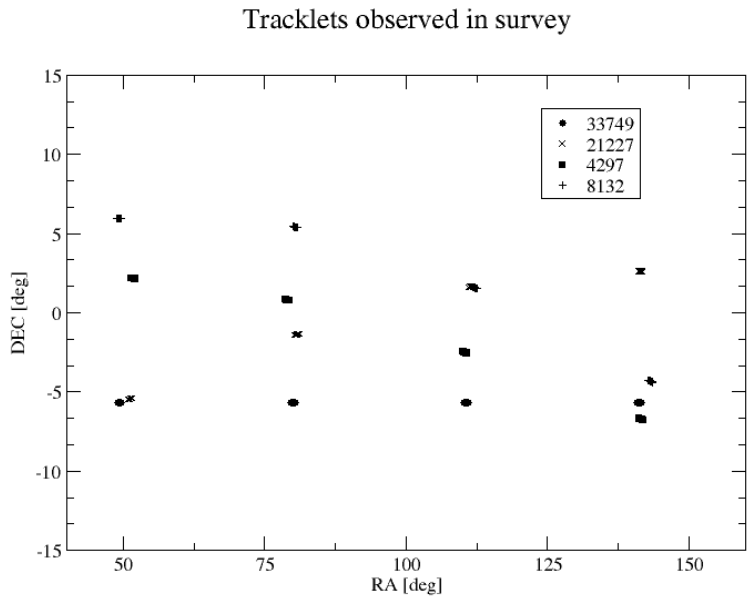

Figure 5: Four objects that are re-observed four times during the survey. The objects are taken from the TLE catalog, the numbers in the legend refer to their NORAD IDs.

Table 3: Parameter settings for GA and EGA algorithms

parameter	GA	EGA
p_{mute}	1/N	1/N
crossover	2 point	2 point
population size	N	N
percentage copied to next generation	N.A.	10%
k_1	1E-5	1E-5
k_2	1E-5	1E-5

The GA as described in the previous section is now applied to this data set. Genetic Algorithms are stochastic processes, therefore it is difficult to say something about their convergence from an analytical standpoint. Instead it is common practice to look at the average performance of the GA for a certain number of runs. In this test case the average fitness per generation over 20 runs is shown. Also the best individual found at that generation is shown. The worst individual found at each generation has such a large fitness value (e.g. 1E7) that it is not displayed in Figure 6. In Table 3 the parameter setting of both the GA and EGA can be found.

As can be seen in Table 3, the parameter settings for both algorithms are identical. They only differ in the fact that the EGA copies the top 10% of the current population to the next generation. The mutation probability is set in such a way that on average there will be one tracklet reassigned to another object per k-matrix. The k_1 and k_2 are set to small values. This is because we assume to know that each object has been re-observed four times and none of the tracklets is spurious.

In Figure 6 the fitness is normalized by the fitness of the true solution. Therefore a value of 1 corresponds to the optimum solution. From Figure 6 we see that both the GA as well as the EGA perform as desired. They rapidly decrease the fitness value before leveling out. The GA reaches a plateau after about 25 generations, it is unable to further improve its solutions by simply recombining and mutating the individuals that it has. The EGA keeps improving its individuals until it is close to a value of 1 after 150 generations. This is because the EGA always keeps the best solution found so far in the search. Also when looking at the best individual found per generation during the 20 runs it is clear that the EGA outperforms the regular GA.

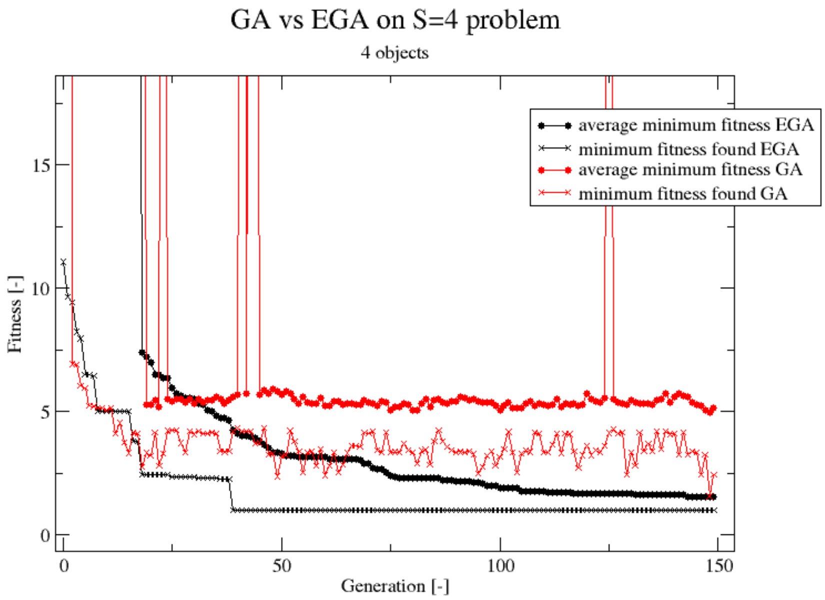

Figure 6: Fitness of best individual in the population vs. the generation number. The fitness is averaged over 20 runs.

An example of a k-matrix with about 5x the optimum fitness is given in Figure 7. Here the circles indicate the optimum solution, the crosses indicate the current solution displayed.

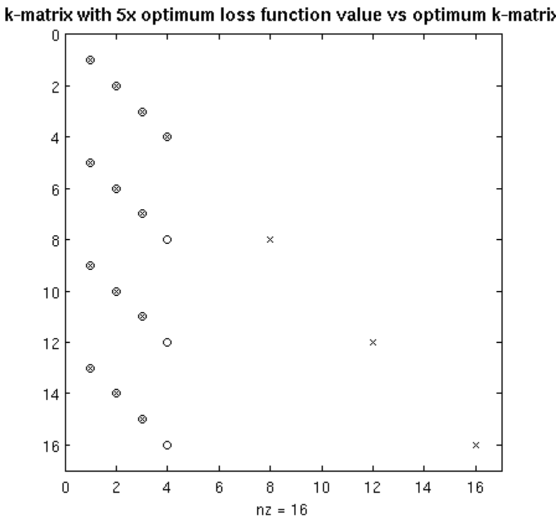

Figure 7: k-matrix with 5x optimum fitness value (crosses) compared to the optimum k-matrix (circles).

From Figure 7 it can be seen that although the k-matrix has a fitness of 5x the optimum, it is still a good approximation of the truth. From here the solution can be further refined. A simple enumerative scheme that explores the remaining possible combinations can easily find the optimum solution when using this approximation as a starting point. According to Figure 6 such a matrix is typically found around generation number 30 - 40.

These results show that the EGA indeed finds an approximation of sufficient quality well before the optimum solution is found. Even so there are still many open questions that need to be addressed. Perhaps the most important future point of research is on the computational complexity of the EGA and perhaps other meta-heuristic algorithms applied to the MTT problem. As stated in the goal of this work, we want a method that can process large data sets. To find out if this is a realistic goal a rule has to be derived about the scaling of the computation time as a function of the problem size. As this method is stochastic, the only way to do this is to derive an emperical rule by applying the method to data sets of different size. If it turns out that the computational complexity is unacceptable an effort can be made in reducing the search space that the algorithm has to explore. At the moment only the information on the k-matrix as a whole is used. The information contained by the hypothetical objects within each k-matrix is currently ignored. This information can be of great value though. Another point that should be made is that population based methods are particularly suited for a parallel architecture. All the individual k-matrices within a given generation are completely independent from each other, and can therefore be evaluated on separate processors.

5. Conclusions

The approach presented in this paper aims to provide a method that can treat the correlation and orbit determination problems simultaneously, and is able to efficiently process large data sets with minimal manual intervention. In this paper the first steps towards such a method have been described.

The method consists of two parts. One is the orbit determination method, which is heavily based on the tracklet to tracklet correlation method as described in [16]. The orbit determination method is dubbed the Optimized Boundary Value Orbit Determination (OBVOD) method. It uses both, the information on the angular rates, as well as on the angular positions. Besides determining the orbit it provides a statistical value on the likelihood that the computed orbit is representative for the measured angular positions and the derived angular rates. This quantity is used in the second part of the algorithm. The second part of the algorithm is a Genetic Algorithm. The Multiple Target Tracking (MTT) problem quickly becomes an NP-hard combinatorial optimization problem, as soon as the dimension $S \geq 3$. Genetic Algorithms have shown to be powerful tools in (approximately) solving NP-hard problems, also their application to the MTT problem has not been extensively researched [11]. The OBVOD method is used during the GA in the fitness evaluation of the individuals.

The OBVOD method works by minimizing the combination of the Mahalanobis distances on the angular positions and the angular rates. Since the topography of both of those quantities is smooth and has one minimum point (for a given range of orbital revolutions) it can be efficiently solved with a gradient descent algorithm. There are however still areas that need attention. One point to investigate is how far the minimum point of the angular positions lies from that of the angular rates. It has already been found that the combination of the two quantities will lead to a minimum point that is neither the minimization of the rates nor the positions. The relationship between the two quantities will be studied more carefully, and might lead to the decision to introduce weights.

The GA has shown to perform as expected. It shows an asymptotic behavior, rapidly decreasing the fitness at first before leveling out. It has also shown that it can find reasonable approximations well before it finds the optimum solution. These reasonable approximations can be further refined with a simple enumerative scheme to further decrease the fitness as far as possible. Two GAs were compared in this study, the regular GA and the elitist GA (EGA). The EGA has shown to outperform the GA consistently and with ease in this case. Concerning this part of the algorithm there are still many topics to be studied. First of all there are many other meta-heuristic methods that are more advanced than the simple GA. Examples are the Population Based Incremental Learning (PBIL) method [2], the Differential Evolution scheme [13], the Guided Mutation algorithm [19] and the Bayesian Network learning and simulating method [9]. Another point of interest is the computational complexity of the algorithm(s). As mentioned in the goal of this work, we want to have a method that is able to process large data sets. The scaling of the computation time vs. the size and dimension of the problem is still not known. Since these methods are all stochastic it is also not possible to say anything about this scaling through an analytical reasoning. Therefore the complexity has to be studied through tests with data sets of different size in order to derive an empirical rule. It is expected that the application of any meta-heuristic method on a data set of realistic size (about 500-1000 tracklets) will lead to unacceptable computation times. Because of this, perhaps the most important aspect to research further is the reduction of the search space. As of now the information of each hypothetical object is not used in the GA, only the information on the k-matrix as a whole is taken into account. The information contained by the hypothetical objects can be put to great use. For instance if a hypothetical object has a good fitness score, one could exclude many tracklets in the other fences and only consider those who are close to a propagated state of the hypothetical object. It is expected that such a scheme would make a significant impact on the necessary computation time.

Acknowledgment

This work has been funded equally by the three project partners under the PhD funding scheme of CNES. The funding parties are CNES, TAS (industrial partner) and the AIUB (host laboratory).

References

[1] J. M. Aristoff, J. T. Horwood, N. Singh, A. B. Poore, C. Sheaff, and M. K. Jah. Multiple hypothesis tracking (mht) for space surveillance: theoretical framework. In *AAS/AIAA Astrodynamics Specialist Conference*, Hilton Head, SC, August 2013.

[2] S. Baluja. Population based incremental learning: A method for integrating genetic search based function optimization and competitive learning. Technical Report CMU-CS-94-163, Carnegie Mellon University, 1994.

[3] R. H. Battin. *An Introduction to the Mathematics and Methods of Astrodynamics, Revised Edition*. American Institute of Aeronautics and Astronautics, 1999.

[4] S. S. Blackman. Multiple hypothesis tracking for multiple target tracking. *IEEE AE Systems Magazine*, 19:5 – 18, 2004.

[5] G. Chen and L. Hong. A genetic algorithm based multi-dimensional data association algorithm for multi-sensor-multi-target tracking. *Mathematical and computer modeling*, 26(4):57 – 69, 1997.

[6] K. Fujimoto, D. J. Scheeres, J. Herzog, and T. Schildknecht. Association of optical tracklets from a geosynchronous belt survey via the direct bayesian admissible region approach. *Advances in Space Research*, 53(2), 2014.

[7] D. E. Goldberg. *Genetic Algorithms in search, optimization and machine learning*. Addison-Wesley, 1999.

[8] I. I. Hussein, K. J. DeMars, C. Früh, R. S. Erwin, and M. K. Jah. An aegis-fisst integrated detection and tracking approach to space situational awareness. In *Information Fusion (FUSION)*, number 12965927, pages 2065–2072. IEEE, 2012.

[9] P. Larranaga, R. Etxeberria, J. A . Lozano, and J. M. Pena. Combinatorial optimization by learning and simulation of bayesian networks. In *Sixteenth conference on Uncertainty in artificial intelligence*, pages 343–352, 2000.

[10] D. Oszkiewicz, K. Muinonen, J. Virtanen, and M. Granvik. Asteroid orbital ranging using markov-chain monte carlo. *Meteoritics and Planetary Science*, 44(12):1897 – 1904, 2009.

[11] A. B. Poore and S. Gadaleta. Some assignment problems arising from multiple target tracking. *Mathematical and computer modeling*, 43(9-10):1074 – 1091, 2006.

[12] A. B. Poore and A. J. Robertson. A new lagrangian relaxation based algorithm for a class of multidimensional assignment problems. *Computational Optimization and Applications*, 8(2):129 – 150, 1997.

[13] R. S. Prado, R. C. P. Silva, F. G. Guimaraes, and O. M. Neto. Using differential evolution for combinatorial optimization: A general approach. In *International Conference on Systems Man and Cybernetics (SMC)*, pages 11 – 18, Istanbul, 2010. IEEE.

[14] A. J. Robertson. A set of greedy randomized adaptive local search procedure (grasp) implementations for the multidimensional assignment problem. *Computational Optimization and Applications*, 19(2):145 – 164, 2001.

[15] M. D. Schneider. Bayesian linking of geosynchronous orbital debris tracks as seen by the large synoptic survey telescopes. *Advances in Space Research*, 49(4), 2012.

[16] J.A. Siminski, O. Montenbruck, H. Fiedler, and T. Schildknecht. Short-arc tracklet association for geostationary objects. *Advances in Space Research*, 53(8):1184 – 1194, 2014.

[17] I. Turkmen and K. Gurney. Tabu search tracker with adaptive neuro-fuzzy inference system for multiple target tracking. In *Progress In Electromagnetics Research*, volume 65, pages 169 – 185, 2006.

[18] I. Turkmen, K. Guney, and D. Karaboga. Genetic tracker with neural network for single and multiple target tracking. *Neurocomputing*, 69(16 - 18):2309 – 2319, 2006.

[19] Q. Zhang. An evolutionary algorithm with guided mutation for the maximum clique problem. In *Transactions on evolutionary computation*, volume 9, pages 192 – 200. IEEE, 2005.