

Space debris cataloging of GEO objects by using meta-heuristic methods.

Inauguraldissertation
der Philosophisch-naturwissenschaftlichen Fakultät
der Universität Bern

vorgelegt von

M. Zittersteijn

Aus den Niederlanden

Leiter der Arbeit:

Prof. Dr. T. Schildknecht
Astronomisches Institut

Ko-Leiter der Arbeit:

Dr. A. Vananti
Astronomisches Institut

Dr. V. Martinot
Thales Alenia Space France

J. C. Dolado Perez
Centre National d'Etudes Spatiales

Space debris cataloging of GEO objects by using Meta-Heuristic methods.

Inauguraldissertation
der Philosophisch-naturwissenschaftlichen Fakultät
der Universität Bern

vorgelegt von

M. Zittersteijn

Aus den Niederlanden

Leiter der Arbeit:

Prof. Dr. T. Schildknecht
Astronomisches Institut

Ko-Leiter der Arbeit:

Dr. A. Vananti
Astronomisches Institut

Dr. V. Martinot
Thales Alenia Space France

J. C. Dolado Perez
Centre National d'Etudes Spatiales

Von der Philosophisch-naturwissenschaftlichen Fakultät angenommen.

Bern, 1-Februari-2017

Der Dekan:
Prof. Dr. (Name)

Abstract

Currently several thousands of objects are being tracked in the Medium Earth Orbit (MEO) and Geosynchronous Earth Orbit (GEO) regions through optical means. The problem faced in this framework is that of Multiple Target Tracking (MTT). The MTT problem becomes an NP-hard combinatorial optimization problem as soon as its dimension S becomes $S \geq 3$. In regions with a high density of objects the MTT problem will have to have this dimension in order to avoid ambiguous solutions. With the advent of improved sensors and a heightened interest in the problem of space debris, it is expected that the number of tracked objects will grow by an order of magnitude in the near future. This research aims to identify an algorithm capable of addressing the problem of space debris cataloging in the MEO and GEO regions, in particular for highly dense regions, without possessing a restrictive computational complexity. In an attempt to find an approximate solution of sufficient quality several Population Based Meta Heuristic (PBMH) algorithms are implemented and tested on simulated optical measurements. In addition to this, a novel way of orbit determination is presented which is based on an existing $S = 2$ tracklet association method. These first results show promise as one of the tested algorithms (the Elitist Genetic Algorithm (EGA)) consistently displays the desired behavior of finding good approximate solutions before reaching the optimum. Furthermore, the results suggest that the algorithm has a polynomial time complexity when finding approximate solutions. The algorithm is also applied to real observations, where it also performs as desired.

Acknowledgements

This study has been co-funded by the Centre National d'Etudes Spatiales (CNES), Thales Alenia Space France, and the Astronomical Institute of the University of Bern (AIUB), through the CNES PhD financing scheme. I would like to thank these institutions, especially those people that have been directly involved. Juan Carlos Dolado Perez of the CNES, Vincent Martinot of Thales Alenia Space France, and Thomas Schildknecht of the AIUB have not only contributed in terms of funding, but also in terms of numerous intellectual contributions. Our regular meetings and their insights have been essential to the development of this PhD work. Alessandro Vananti of the AIUB deserves special mentioning, as he was my responsible and supervisor in my day to day activities.

List of Publications

- M Zittersteijn, A Vananti, T Schildknecht, JC Dolado Perez, and V Martinot. Associating optical observations of space debris in geo with an elitist genetic algorithm. In *Final Stardust Conference*, ESA-ESTEC, Noordwijk, the Netherlands, 2016c.
- M Zittersteijn, A Vananti, T Schildknecht, JC Dolado Perez, and V Martinot. Associating optical measurements of geocentric objects with a genetic algorithm: application to experimental data. In *Proceedings of 67th International Astronautical Congress*, Guadalajara, Mexico, 2016b.
- M Zittersteijn, A Vananti, T Schildknecht, JC Dolado Perez, and V Martinot. Associating optical measurements of meo and geo objects using population-based meta-heuristic methods. *Advances in Space Research*, 58(9):1778–1792, 2016a.
- M Zittersteijn, A Vananti, T Schildknecht, JC Dolado Perez, and V Martinot. A study of the time complexity of an elitist genetic algorithm used for associating optical observations of meo and geo space debris. In *Proceedings of workshop on Key Topics in Orbit Propagation Applied to Space Situational Awareness (KEPASSA)*, Toulouse, France, 2015c.
- M Zittersteijn, A Vananti, T Schildknecht, JC Dolado Perez, and V Martinot. Associating optical measurements and estimating orbits of geocentric objects through population-based meta-heuristic methods. In *Proceedings of 66th International Astronautical Congress*, Jerusalem, Israel, 2015b.
- M Zittersteijn, A Vananti, T Schildknecht, JC Dolado Perez, and V Martinot. A genetic algorithm to associate optical measurements and estimate orbits of geocentric objects. In *Proceedings of 6th European Conference for Aeronautics and Space Sciences*, Krakow, Poland, 2015a.

Glossary

- AIUB** Astronomical Institute of the University of Bern. iii, 102
- AMR** Area to Mass Ratio. xii, 55–59, 63–65, 127
- AR** Admissible Region. x, 14–21, 23, 42
- BFGS** Broyden-Fletcher-Goldfarb-Shanno. 47
- BVAR** Boundary Value Admissible Region. x, 14, 21, 22, 42, 69
- CCD** Charged Coupled Device. 6, 8
- cdf** cumulative distribution function. 89
- CMOS** Complementary Metal-Oxide Semiconductor. 6
- CNES** Centre National d’Etudes Spatiales. iii
- CPHD** Cardinality Probability Hypothesis Density. 13
- DE** Differential Evolution. 37, 38, 94, 126, 127
- DNA** Deoxyribonucleic Acid. 29
- EGA** Elitist Genetic Algorithm. i, xix, 35, 94, 101, 110, 119, 126–128
- GA** Genetic Algorithm. x, 28–30, 34–37, 93, 126
- GEO** Geosynchronous Earth Orbit. i, xii, 3–8, 48, 49, 54–57, 125, 127, 128
- GNSS** Global Navigation Satellite System. 4

- GRASP** Greedy Randomized Adaptive Local Search. 14, 28
- IOD** Initial Orbit Determination. 41–43, 52, 63, 88, 93
- ISS** International Space Station. 125
- IVAR** Initial Value Admissible Region. ix, x, 14, 16–18, 20, 42
- LEO** Low Earth Orbit. 3–5, 125
- LLR** Log Likelihood Ratio. 24
- MCMC** Markov Chain Monte Carlo. 71
- MEO** Medium Earth Orbit. i, xii, 3, 4, 54, 56–59, 125, 127
- MHT** Multiple Hypothesis Tracking. 14, 22–25, 27, 125, 129
- MTT** Multiple Target Tracking. i, x, 8, 10, 14, 22, 25, 27–29, 32, 39, 41, 43, 71, 72, 74, 125, 126
- NORAD** North American Aerospace Defense Command. 94
- OBVIOD** Optimized Boundary Value Initial Orbit Determination. xix, 42, 43, 53, 60, 73, 101–104, 113, 126, 127, 129
- PBIL** Population Based Incremental Learning. xi, 36, 37, 94, 126, 127
- PBMH** Population Based Meta Heuristic. i, 13, 25, 26, 28, 29, 71, 93, 126, 127
- pdf** probability density function. xi, 13, 31, 47, 48
- RADAR** Radio Detection and Ranging. 4, 10
- RMS** root mean square. 18, 54, 101
- TLE** Two Line Element. 3, 8, 94
- TSP** Traveling Salesman Problem. x, 12, 28, 29, 32
- USSTRATCOM** US Strategic Command. 3

List of Figures

1.1	The number of tracked geocentric space objects over time http://www.esa.int/spaceinimages/Images/2013/04/Object_catalogue . Accessed on 17/11/2016.	2
1.2	The number of tracked geocentric space objects over time, categorized based on their nature http://www.esa.int/Our_Activities/Operations/Space_Debris/Analysis_and_prediction . Accessed on 17/11/2016.	2
1.3	Damage caused by an object of a few millimeter in diameter that impacted a sheet of metal at hyper-velocity. http://www.esa.int/Our_Activities/Operations/Space_Debris/Hypervelocity_impacts_and_protecting_spacecraft . Accessed on 17/11/2016.	4
1.4	Different sources of perturbing accelerations as a function of the distance from the center of the Earth. The Area to Mass Ratio (AMR) used to compute the solar radiation pressure is $0.01 m^2/kg$. (Montenbruck and Gill, 2000)	6
1.5	Definition of measured angles right ascension α and declination δ	7
1.6	A CCD image of an ASTRA cluster (Schildknecht, 2007)	8
1.7	A schematic representation of a tracklet, which is a series of closely spaced optical observations.	8
1.8	Standard survey scenario applied to geostationary objects with $i < 3$ deg and $e < 0.1$. Coordinates are in topocentric right ascension α and declination δ . Each cross represents the center of the field of view of the telescope, the telescope scans in fixed right ascension and varying declination (also known as a fence). The observations are simulated for the night of 11-6-2014 for a part of the geosynchronous population.	10
1.9	A zoom in of four consecutive fences.	11

1.10	An overview of different methods used and under investigation in the domain of space debris cataloging.	15
1.11	An example of an Initial Value Admissible Region (IVAR) (Tommei et al., 2009).	18
1.12	An example of the discretization of an IVAR with Delaunay triangulation (Tommei et al., 2009).	19
1.13	A schematic representation of the propagation through time of two Admissible Region (AR)s in the Poincaré element space. The overlap means that the probability that the two tracklets belong to the same object is non-zero, in this case the two tracklets are associated. (Fujimoto and Scheeres, 2011)	20
1.14	The topography of the initial value loss function, over the IVAR. As can be seen, it is challenging to robustly find the global minimum point. (Siminski et al., 2014)	22
1.15	The topography of the boundary value loss function, over a Boundary Value Admissible Region (BVAR). This is for a set number of revolutions. (Siminski et al., 2014)	23
1.16	Three measurements and two propagated states with their uncertainties. Due to the overlap in uncertainty and inclusion of multiple measurements the association becomes ambiguous (Blackman, 2004).	25
1.17	A tree structure containing all possible tracks that originate from measurement 1. Here the N-scan pruning technique is depicted, where half of the tree is discarded in hindsight (in this case after two fences), here the left branch is deleted from the tree (Blackman, 2004).	26
2.1	A graphical representation of the MTT problem	30
2.2	A graphical representation of the Traveling Salesman Problem (TSP) problem	31
2.3	A flowchart of a Genetic Algorithm (GA)	32
2.4	An example of a discrete probability density function made by the relative fitness values in the population.	33
2.5	Transformation from an individual represented by a string of binary values to a roulette wheel representation.	35
2.6	Randomly selecting the cutting point in the roulette wheel and mapping it back to the string representation.	35

2.7 Exchanging the two halves of the individuals to create two new individuals. 36

2.8 Uniform crossover exchanges an element between two individuals with a crossover probability $p_{crossover}$ 36

2.9 A few examples of schemata for an individual represented by nine binary numbers. 36

2.10 An example of the mutation operator. 37

2.11 Flowchart of the Population Based Incremental Learning (PBIL) algorithm 38

2.12 Graphical representation of differential evolution (Storn and Price, 1997). 41

3.1 The non-dimensional orbit arc length T versus the parameter x (Gooding, 1990). 47

3.2 The probability density function (pdf) of the Mahalanobis distance with three tracklets. The covariance of the computed angular values is not neglected in these results. Therefore the values are distributed according to a χ^2 distribution. The covariance of the computed angular rates is neglected. 49

3.3 The pdf of the Mahalanobis distance with three tracklets. The covariance of the computed angular values is not neglected in these results. Therefore the values are distributed according a to χ^2 distribution. The covariance of the computed angular rates is neglected. It is seen that neglecting the covariance of the angular rates leads to Mahalanobis distance values that are not χ^2 distributed. 50

3.4 The topography of the loss function for a set of three tracklets. The topography of the loss function of the angular rates (left) and angular positions (right) are shown separately in order to show that they are similar. The total topography would be the summation of the two topographies shown. 51

3.5 Loss function topography, $N = 2$, separation is 1 hour. 52

3.6 Loss function topography, $N = 2$, separation is 6 hours. 52

3.7 Loss function topography, $N = 2$, separation is 12 hours. 52

3.8 The topography of the loss function for three tracklets where the first and last tracklets are spaced at six hours from each other, the third tracklet is positioned halfway between the first and the third tracklet. 52

3.9	The topography of the loss function for three tracklets where the first and last tracklets are spaced at 12 hours from each other, the third tracklet is positioned halfway between the first and the third tracklet.	53
3.10	The topography of the loss function for two tracklets that are spaced at 24 hours from each other.	54
3.11	The probability distribution of the minimized loss function values for two tracklets.	55
3.12	A comparison between a low noise and high noise case. Two tracklets spaced at varying fractions of an orbital interval were used in these orbit determinations, the results shown are the average results over 1000 simulations. The AMR value is $0.02 m^2/kg$	57
3.13	A comparison between a low AMR and high AMR case. Two tracklets were used in these orbit determinations, the results shown are the average results over 1000 simulations. A low noise value is used of $\sigma_0 = 1''$	58
3.14	A comparison between an object in MEO orbit and an object in a GEO orbit. Two tracklets were used in these orbit determinations, the results shown are the average results over 1000 simulations. The low noise and low AMR values are used in this plot.	59
3.15	A comparison between a high noise and a low noise case in MEO. Two tracklets were used in these orbit determinations, the results shown are the average results over 1000 simulations. The low AMR value is used in this figure.	60
3.16	A comparison between a high AMR and a low AMRf case in MEO. Two tracklets were used in these orbit determinations, the results shown are the average results over 1000 simulations. The low value of $\sigma_0 = 1''$ is used for the noise.	61
3.17	The distributions of the observed and computed angular rates for two tracklets and an arc length of 242 hours. Here the object's motion is Keplerian, therefore the computed values will (on average) match the observed ones.	63
3.18	The distributions of the observed and computed angular rates for two tracklets and an arc length of 242 hours. Here the object's motion is perturbed, therefore the computed values will not match the observed ones.	63

3.19 A χ^2 distribution compared to a noncentral χ^2 distribution. They both have two degrees of freedom, this corresponds to the two tracklets scenario. A value of $\lambda = 10$ is used. 64

3.20 A non-central χ^2 distribution depicting the distribution of the Mahalanobis distance in the case where two tracklets are positioned close to a problematic geometry. The non-centrality parameters is $\lambda = 35.9$ in this example. The x-axis label 'd2' refers to the Mahalanobis distance. 65

3.21 The evolution of the λ values as a function of the length of the orbit arc for a high and low AMR object. 66

3.22 The semi-major axis and eccentricity values of the computed orbits in case of a low and a high AMR. 67

3.23 The probability distribution function of the Mahalanobis distance for two tracklets with a length of the orbit arc of 115 [hr]. This length of the orbit arc is close to the ideal geometry of the Lambert problem. Apparently the Lambert solver that is currently used has problems when dealing with this geometry in a multiple revolution case. The x-axis label 'd2' refers to the Mahalanobis distance. 68

3.24 λ versus length of the orbit arc between two tracklets. An increase in AMR does not necessarily mean an increase in λ . Note that 1000 hours is about 41 days. 69

3.25 Two tracklets that are 121 hours apart. Due to the noise, different solutions exist. 70

3.26 The semi-major axis versus the eccentricity of computed orbits for different numbers of orbital revolutions. Both these orbital regimes are able to closely approximate the attributed angular rates of two tracklets that are spaced at 121 hours from each other. 71

4.1 Probability density function of the fitness values that a group of three tracklets can have. Three tracklets can be arranged as three single tracklets (N=1/N=1/N=1), a group of two tracklets and a single tracklet (N=2/N=1) and a group of three tracklets N=3. 77

5.1	results of the model test for increasing number of observations. These are average results over 1000 calculations. It is seen that for a small number of observations the m_0^2 differs from the σ_0^2 .	81
5.2	Monte Carlo simulation for object 31306. Here the a posteriori uncertainty on the angular rates is used. The values are slightly too small to fit the χ^2 distribution. When the a priori uncertainty is used the values would fit exactly to the χ^2 distribution.	82
5.3	The performance of the algorithm with different methods to compute the uncertainties on the angular rates. The performance of the algorithm is not much affected by the method where no a priori information is necessary.	83
5.4	Values the fitness can have, depending on the number of tracklets N. Here the normalization factors are taken into account	84
5.5	Values the fitness function can have, depending on the number of tracklets N. Here the normalization factors are not taken into account. Note that the distributions overlap significantly for this detection probability.	85
5.6	Comparing the N = 3 and the (N = 2 / N = 1) distributions. They overlap slightly in this case. The detection probability is set at $P_d = 0.9$, the false alarm probability at $P_f = 0.01$, and $\gamma = 0$.	86
5.7	Comparing the N = 3 and the (N = 2 / N = 1) distributions. Here the $P_d = 0.2$, the consequence is that the distributions overlap significantly and that the N = 1 value is small. The false alarm probability is $P_f = 0.01$ $\gamma = -50$.	86
5.8	Different weights for the normalization constants will result in a different performance of the algorithm. The γ parameter is set to zero.	87
5.9	Varying the weights for the normalization factors W_n and the value of the γ parameter have a large influence on the algorithm performance.	88
5.10	Comparing the N = 3 and the (N = 2 / N = 1) distributions. Here the normalization factors are taken into account and γ is used to bring the distributions close together.	89

5.11	Values the fitness function can have, depending on the number of tracklets N . Here the normalization factors are not taken into account ($W_n = 0$). Note that the distributions overlap significantly for this detection probability.	90
5.12	Values the fitness function can have, depending on the configuration of three tracklets.	91
5.13	Values the fitness function can have, depending on the number of tracklets N . Here the γ parameter is optimized.	92
5.14	A comparison between different γ parameter settings. The $\gamma = -40$ setting is the optimized setting, where the distributions overlap with $p = 1 \cdot 10^{-3}$ at maximum.	93
5.15	The difference in performance when tournament selection is used as opposed to the the previous scheme based on relative fitness. The lines denoted with 'opt' represent the optimal solutions.	94
5.16	The observations used in the first test case. These are Geosynchronous objects on nearly circular orbit, separated in inclination.	97
5.17	The average best fitness individual found per generation. Average value is based on 100 runs.	97
5.18	The average k-matrix at the end of the run vs. the true k-matrix for the first test case.	98
5.19	The observations used in the second test case. These are closely spaced objects belonging to one of the ASTRA clusters.	99
5.20	The average best fitness individual found per generation. Average value is based on 100 runs.	100
5.21	The average k-matrix at the end of the run vs. the true k-matrix for the second test case.	100
5.22	The observations used in the third test case. This test case contains both, objects that are separated in inclination, as well as an ASTRA cluster and a single tracklet.	101
5.23	The average best fitness individual found per generation. Average value is based on 100 runs.	102
5.24	The average k-matrix at the end of the run vs. the true k-matrix for the third test case.	102
5.25	Attributed values of the observed objects in the ASTRA cluster.	104
5.26	Average best fitness value per generation, averaged over 100 runs. The results of the individual runs is shown in gray.	105

5.27	Average k-matrix at the end of 500 generations. The EGA consistently converges to the same solution.	106
5.28	The difference in right ascension and declination between the least squares solution and the OBVIOD solution. The x-axis label ‘doy’ stands for ‘day of year’.	107
5.29	Residual in right ascension for each observation and for each object.	108
5.30	Residuals in declination for each observation and for each object.	109
5.31	The average k-matrix found at different stages of the search. Both the percentage of the optimum fitness as well as the time at which the solution is found are given. The intensity scale is the same as in Figure 5.27.	111
5.32	Computation time needed to reach an approximate solution of a certain quality (90% of optimum) versus the number of tracked objects. Two lines are fitted, a polynomial and an exponential function. Each point represents the average computation time over 100 runs.	112
5.33	The attributed and propagated angular positions at the epoch of the third tracklet for one object. Although the distributions are centered on the same position, the uncertainty of the propagated positions is considerable.	114
5.34	The attributed and propagated angular rates for one object. In this parameter space the distributions are centered on the same values and have a similar uncertainty.	115
5.35	The attributed and propagated angular positions at the epoch of the third tracklet for all four objects in the cluster. Although the distributions are centered on the same position, the uncertainty of the propagated positions is considerable. . .	116
5.36	The attributed and propagated angular rates. In this parameter space the distributions are centered on the same values and have a similar uncertainty.	117
5.37	The propagated covariance of the angular position of one object. The ellipse is drawn at one standard deviation.	118
5.38	The propagated covariance of the angular rates of one object. The ellipse is drawn at one standard deviation.	119
5.39	The Mahalanobis distance between the attributed and propagated state are distributed according to a four degrees of freedom χ^2 distribution.	120

5.40 In this figure the ‘optimum’ solution is the straight line that represents the true solution. The ‘original’ solution is the algorithm as it was applied to real data in Section 5.5. The ‘pairs’ solution is the solution where the full initial population is based on the pairwise checking results, the ‘half pairs’ solution initializes only half of the population in this manner. The acronym SSR refers to the search space reduction method being applied. 122

5.41 A zoom-in of the different performance of the different schemes. 123

5.42 The average k-matrix at the end of 500 generations, with the original implementation of the algorithm. 124

5.43 Difference between the search space reduction solution and the original implementation. 124

5.44 Difference between the ‘pairs’ solution and the original implementation. 125

5.45 Difference between the ‘half pairs’ solution and the original implementation. 125

5.46 Difference between the ‘shalf pairs search space reduction’ solution and the original implementation. 126

List of Tables

5.1	Settings of the parameters	96
5.2	EGA parameter settings	103
5.3	Orbital elements as determined by Optimized Boundary Value Initial Orbit Determination (OBVIOD)	106
5.4	Goodness of fit values for both models.	111

Contents

1	Introduction	1
1.1	Optical observations of space debris	7
1.2	Multiple Target Tracking	11
1.3	P vs. NP complexity	13
1.4	Existing methods	14
1.4.1	Admissible regions	16
1.4.2	Multiple Hypothesis Tracking	24
1.5	Outline of the thesis	27
2	PBMH methods	29
2.1	Motivation	29
2.2	Genetic Algorithms	31
2.3	Population Based Incremental Learning	38
2.4	Differential Evolution	39
3	Initial Orbit Determination: the OBVIOD method	43
3.1	Motivation	43
3.2	The optimized boundary value method	45
3.2.1	Lambert problem	45
3.3	Generalization of the optimized boundary value method	47
3.4	Orbit determination performance	55
3.5	Limitations	62
3.6	Multiple revolution solutions	70
4	Application to space debris cataloging	73
4.1	Solution representation	73
4.2	Fitness function	74

5	Results	79
5.1	Attributed angular rates uncertainties	79
5.2	Fitness function parameter settings	84
5.2.1	Tuning the γ parameter	90
5.3	Selection operator of the GA	93
5.4	Simulated test cases	94
5.4.1	Test case 1: four easy objects	96
5.4.2	Test case 2: four objects in a cluster	99
5.4.3	Test case 3: mixed case with eight objects	101
5.5	Real observations	102
5.6	Time complexity	110
5.7	Improving the convergence rate	112
5.7.1	Search space reduction method	113
5.7.2	Improving the convergence: results	121
6	Conclusions	127

Chapter 1

Introduction

In 1957 the Sputnik satellite was launched, it is the first man made object to orbit the Earth. That launch marked the beginning of the space age, and a rapid increase in the number of Earth orbiting satellites. Because the rate at which objects reenter the atmosphere is relatively low, the total number of orbiting objects keeps accumulating over time. Therefore about 95% of the current Earth orbiting objects are labeled as space debris. Space debris is defined as follows (Flury and Johnson, 1999):

Orbital debris is herein defined as any man-made object, which is non-functional with no reasonable expectation of assuming or resuming its intended function, or any other function for which it is or can be expected to be authorized, including fragments and parts thereof.

In Figure 1.1 the number of tracked space objects can be seen over time according to the country or state of origin.

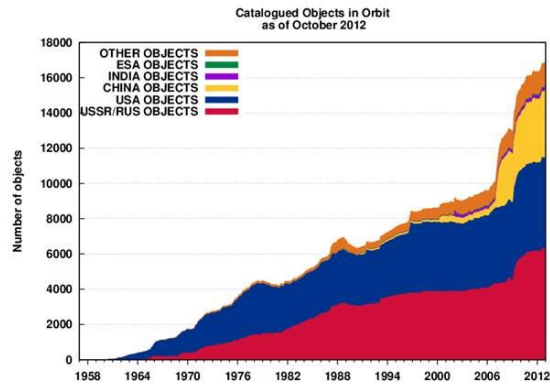


Figure 1.1: The number of tracked geocentric space objects over time http://www.esa.int/spaceinimages/Images/2013/04/Object_catalogue. Accessed on 17/11/2016.

Figure 1.2 shows the objects categorized by their nature.

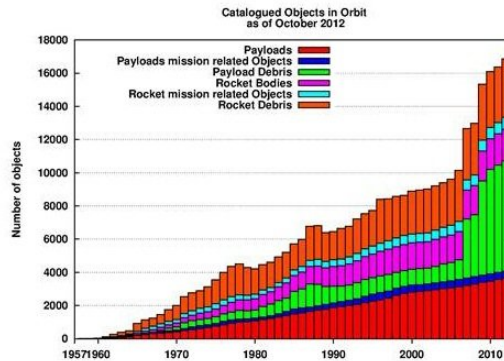


Figure 1.2: The number of tracked geocentric space objects over time, categorized based on their nature http://www.esa.int/Our_Activities/Operations/Space_Debris/Analysis_and_prediction. Accessed on 17/11/2016.

From Figure 1.1 the increasing trend in the number of tracked objects is

apparent. Note that this trend is not only due to the increase in the number of launches, but also due to the improved capabilities to track objects in Earth orbit. Two major events can be seen in Figure 1.1. The first is the Chinese anti satellite test that was conducted in 2007. Here the Chinese demonstrated their capability of eliminating a satellite in Low Earth Orbit (LEO) by destroying it with a missile. The satellite in question was an old, defunct, Chinese satellite called the Fengyun 1-C. The other sharp increase in the number of objects is caused by the Cosmos-Iridium crash in 2009. This crash demonstrated the real threat that space debris poses for the remaining functional satellites. The main consequence of these events is the large number of new objects that have been created.

In a worst case scenario a cascading effect can be caused by such collisions, where each collision generates more objects which in turn cause more collisions. This scenario is also known as the Kessler syndrome. New launches and collisions are not the only sources of space objects. Also so-called break up events generate new debris objects. To date some 200 of these events have been recorded. They are thought to occur mainly due to residual fuel left in upper rocket stages which causes the stage to explode. Currently about 16000 objects are tracked by the US Strategic Command (USSTRATCOM) and are publicly available through the Two Line Element (TLE) catalog¹. The catalog includes objects down to about 10 cm in diameter in the LEO region and about 1 m diameter in the higher Medium Earth Orbit (MEO) and Geosynchronous Earth Orbit (GEO) regions. As the interest in space debris increases, and the quality and quantity of the sensors improves, the number of tracked objects can increase by an order of magnitude in the near future. Figure 1.3 shows the damage that can be done by an object in the LEO region, where the relative velocities can be up to 15 km/s.

Objects of the size shown in Figure 1.3 cannot be tracked, even though they can cause catastrophic collisions. Tracking these objects represents a significant challenge, one that not only requires improved sensor systems but also improved algorithms capable of processing the large amount of data that results from these improved sensors.

The three main categories of Earth orbits are the LEO (up to 2000 km altitude), MEO (from 2000 km to about 36000 km altitude), and GEO (around

¹<https://www.space-track.org/>



Figure 1.3: Damage caused by an object of a few millimeter in diameter that impacted a sheet of metal at hyper-velocity. http://www.esa.int/Our_Activities/Operations/Space_Debris/Hypervelocity_impacts_and_protecting_spacecraft. Accessed on 17/11/2016.

the 36000 km altitude) regions. Roughly stated it can be said that the LEO region is used mostly for scientific and Earth observing satellites, the MEO region is where the Global Navigation Satellite System (GNSS) satellites are positioned, and the GEO region is where the majority of the the telecommunication and weather satellites satellites are placed. The tracking methods that are used mainly depend on the region in which the tracked objects reside. For LEO objects Radio Detection and Ranging (RADAR) and optical sensors can be used. For the MEO and GEO regions in general only optical sensors are used, because the objects are too far away to be observed with a RADAR system. The other aspect that has an influence on the nature of the problem are the perturbing forces that act on the orbiting bodies. An overview of these perturbations can be seen in Figure 1.4. In LEO the bodies are subjected to heavy perturbations by the non-homogeneous gravity field of the

Earth and its atmosphere. Whereas in MEO and GEO Earth's gravity field has less influence on the objects' orbit. Besides that, the atmospheric drag can be neglected at these altitudes. The third body perturbations caused by the Sun and Moon play a larger role in the high altitude orbits than in the LEO region.

Space debris cataloging consists of two problems. The first problem is the problem of identification, where each observation of a space object has to be associated to an object. The identification problem can be divided into two sub problems, one being the identification of an already cataloged object (tracklet-to-orbit association), and the other being the identification of a new object (tracklet-to-tracklet association). This work only focuses on the tracklet-to-tracklet association problem. The second problem is the estimation of the objects' state, or the orbit determination problem. As will be discussed further on, these two problems are interrelated. The number of tracked objects is expected to increase by an order of magnitude in the near future. These objects will still be in the same space as the currently tracked population, therefore the object density will increase. The goal of this thesis is therefore:

Purpose statement

To identify an algorithm capable of addressing the problem of space debris cataloging in the GEO region, in particular for highly dense regions, without possessing a restrictive computational complexity.

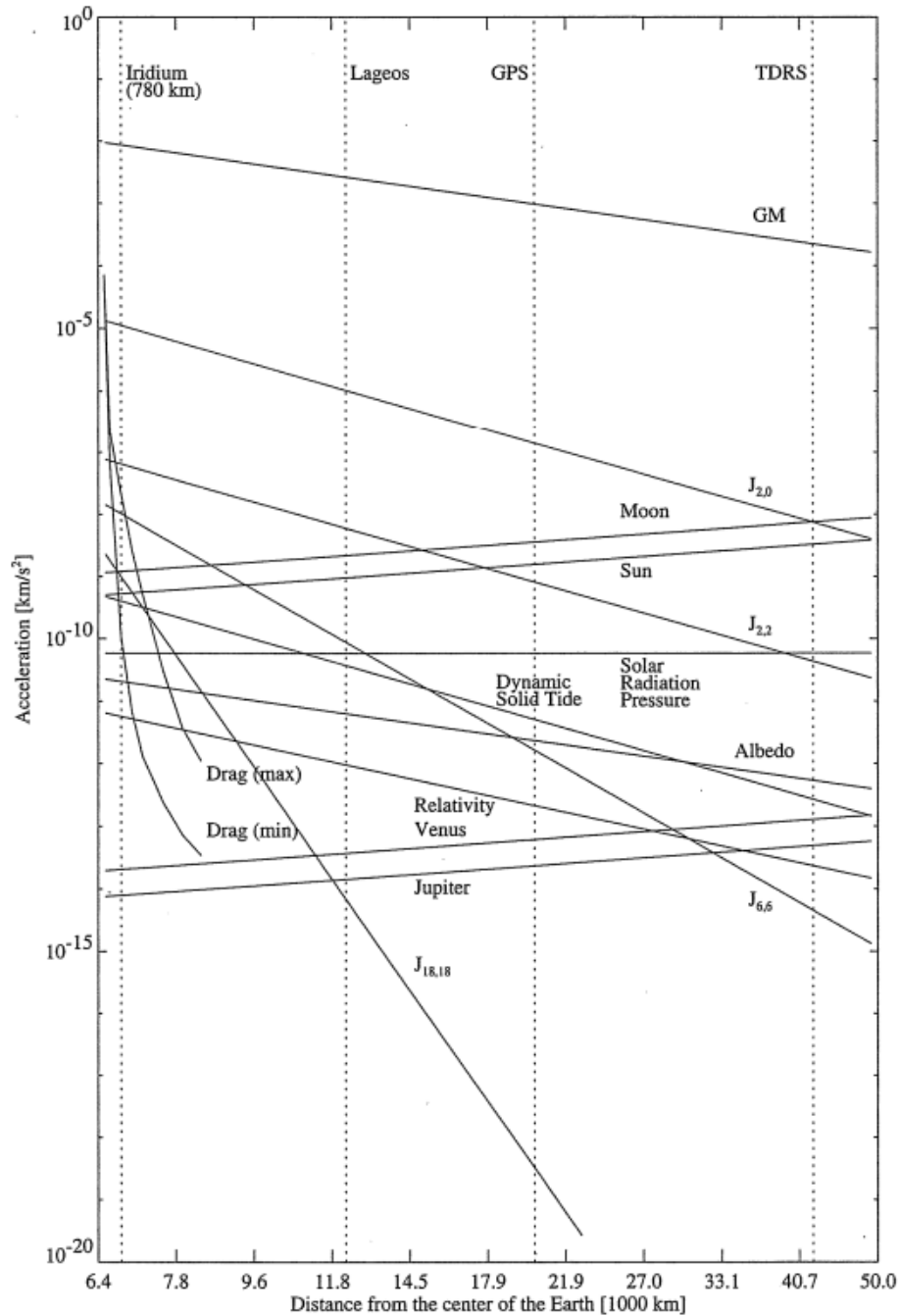


Figure 1.4: Different sources of perturbing accelerations as a function of the distance from the center of the Earth. The Area to Mass Ratio (AMR) used to compute the solar radiation pressure is $0.01 \text{ m}^2/\text{kg}$. (Montenbruck and Gill, 2000)

1.1 Optical observations of space debris

In this work only optical measurements are considered. The optical measurements are treated after the astrometry has been performed, therefore the presented work does not cover the e.g. telescope pointing or image processing algorithms that are needed. With an optical system a pair of angles $(\alpha, \delta)_t$ at an epoch t can be observed, where the angles are defined as in the schematic given in Figure 1.5. This topocentric reference frame is centered on the optical sensor, and oriented in the same way as the geocentric equatorial coordinate system (i axis parallel to the vernal equinox direction and ij plane parallel to the equatorial plane)(Vallado, 2007).

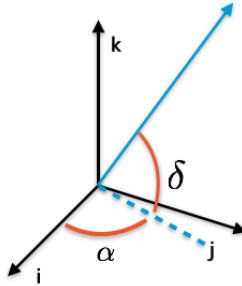


Figure 1.5: Definition of measured angles right ascension α and declination δ .

Typically space debris is observed with Charged Coupled Device (CCD) or Complementary Metal-Oxide Semiconductor (CMOS) sensors, which results in images such as that shown in Figure 1.6. In this figure an image of one of the ASTRA clusters is seen.

The astrometric position of each object is extracted through image processing software which uses a reference star catalog to identify the stars that are present in the frame. The exact position of the object is then calculated with respect to these reference stars. For the optical data obtained at the Zimmerwald observatory the uncertainty of the extracted position is around $0.7'' \geq \sigma \leq 1''$. Note that in this tracking mode the rotation of the Earth is not corrected for, therefore the stars appear as stripes and a GEO object is a point. Other tracking modes are possible. This thesis does not concern itself with the extraction of the astrometric position. The interested reader is referred to (Früh, 2011; Schildknecht, 2007; Schildknecht et al., 1997; Schild-

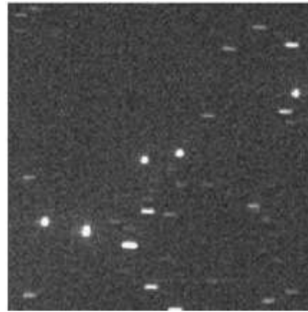


Figure 1.6: A CCD image of an ASTRA cluster (Schildknecht, 2007)

knecht, 1994).

Space debris is typically observed multiple times in rapid succession. Such a series of observations is also called a tracklet or very short arc and consists of four to seven observations made at 15 s to 30 s intervals (in the case of the Zimmerwald observatory). A schematic representation of such a series is given in Figure 1.7.

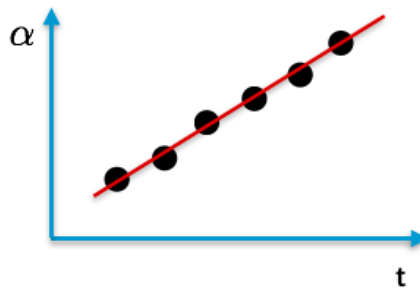


Figure 1.7: A schematic representation of a tracklet, which is a series of closely spaced optical observations.

A series of observations is made in order to be sure that an object has been observed, and to obtain a larger number of observations which is beneficial when e.g. a least squares estimator is applied to them. The tracklet formation process is where the individual observations are associated to one another in order to form the correct series of observations. Such a process is described

in more detail in (Früh and Schildknecht, 2012). The work presented in this thesis aims to exploit this strategy of observing space debris, and therefore exclusively considers that space debris is observed in this manner. A feature of a tracklet is that it describes such a short portion of the object’s orbit (e.g. for a GEO about 2-5 min of a 24 hr period) that a straight line can approximate the object’s motion during that time span. By fitting a straight line to the observations in a tracklet both the average position and the angular rate of the object are found. This is a so-called attributable as given in Equation 1.1 (Tommei et al., 2007). The epoch of the attributable is the average epoch of the observations in the series.

$$\theta_t = \left(\alpha, \delta, \dot{\alpha}, \dot{\delta} \right)_t \quad (1.1)$$

An important aspect of the observations and tracklets is the quantification of the their uncertainty. In general it is assumed that the noise on a single measurement is Gaussian. Therefore a single measurement is described with a mean value and a standard deviation σ_0 . The σ_0 can be characterized and depends on the telescope, CCD hardware, and image processing software used. However it is arduous to do so and is in itself not perfectly accurate. Another option is to use the uncertainties that are provided by the least squares straight line fit. In this work all measurement errors are assumed to be Gaussian with no bias.

The tracklets are collected through two types of observation strategies: the survey and the follow-up strategies. In the survey strategy the sky is ‘scanned’ by a telescope (typically with a relatively large field of view). The aim is to discover new objects by observing as many objects as possible in such a way that they can be successfully identified and that their orbit can be determined. The most interesting objects discovered by these surveys are then observed through a follow-up campaign. In the follow-up strategy the orbit of the object is already roughly known, and the new observations serve to improve the orbit of that object. In Figure 1.8 an example of a simulated survey can be seen. Here a number of GEO objects from the TLE catalog are simulated and observed through a survey type strategy. The telescope is used to scan portions of the sky with fixed right ascension and varying declination. It starts at a certain position and it takes seven images (in order to form the tracklets afterwards), it then moves one field of view in declination and repeats the process. It does this five times before moving back to the first position. The telescope scans in such a manner for e.g. one hour,

after which it moves one hour in right ascension and continues to scan at the new position. This is done in order to attempt to re-observe the objects that it has observed in the first position. Each of such scans with a fixed right ascension and varying declination is called a fence.

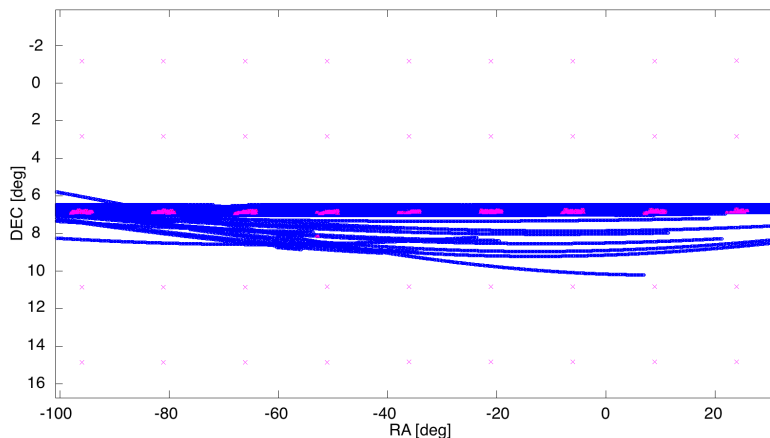


Figure 1.8: Standard survey scenario applied to geostationary objects with $i < 3$ deg and $e < 0.1$. Coordinates are in topocentric right ascension α and declination δ . Each cross represents the center of the field of view of the telescope, the telescope scans in fixed right ascension and varying declination (also known as a fence). The observations are simulated for the night of 11-6-2014 for a part of the geosynchronous population.

This work mainly focuses on treating observations collected with a survey type strategy because in this strategy the identity of each object is truly unknown. In this situation the so called Multiple Target Tracking (MTT) problem is encountered. Section 1.2 describes this problem in more detail.

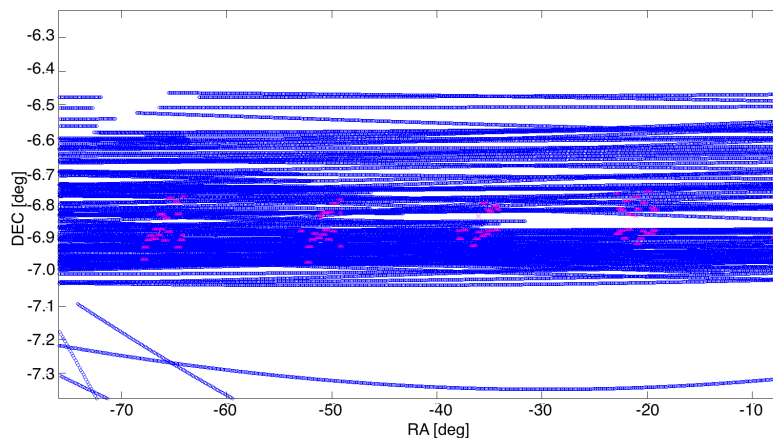


Figure 1.9: A zoom in of four consecutive fences.

1.2 Multiple Target Tracking

In this section the problem addressed in this thesis is defined. The problem of tracking space debris can be generalized to the MTT problem. In this problem a set of tracklets, grouped in a number of S fences (also known as scans in e.g. the RADAR domain), is given. The goal is to find the total number of objects from which the tracklets originate as well as the state of each of these objects. This problem is encountered in numerous domains, examples being defense (Rakdham, 2009), and the tracking of particles resulting from high energy collisions in particle physics (Pusztaszeri et al., 1996). The MTT problem also takes the occurrence of both false alarms (sporadic measurements) and missed detections into account. The problem consists of two interrelated parts, namely data association and state estimation. In the data association part the observations from the different scans have to be associated to the correct object. The state estimation part then takes these associated groups of observations and estimates the target state. This leads to a search for the permutation that results in the target state estimates that best approximate the measurements, according to a certain metric. The number of scans S that are used in the problem correspond to its dimension. For a dimension of $S \geq 3$ the number of possible permutations greatly increases and the problem becomes NP-hard (Poore and Gadaleta, 2006). For instance, in the case where $S = 2$ with two tracklets per fence,

there will be a total of seven possible permutations. However for the $S = 3$ case with two tracklets per fence, there will be 87 of these permutations (Aristoff et al., 2013). This increase in the number of permutations leads to the problem falling in a different category of complexity (the NP category), this is explained in more detail in Section 1.3.

The dimension of the problem is equal to the number of fences S (also called an S -D MTT problem). Each fence consists of n_s tracklets, where $s = 1, 2, \dots, S$, a tracklet is denoted by $i = 1, \dots, N$ within the total set of tracklets. Each of the tracklets belongs to at most one object denoted by $j = 1, 2, \dots, J$. The missed detection probability is denoted by P_d , a missed detection is taken into account by adding a dummy tracklet z_{s_0} to every fence. A permutation is denoted by a which lies in the set \mathcal{A} that contains all possible permutations. Each object j has a cost f_j that contributed to the total cost of a certain permutation a that contains multiple objects (up to N maximum, where N is the total number of tracklets). Summing those costs gives the total cost of a permutation, as written in Equation 1.2. Now the combinations of tracklets from different fences needs to be optimized in order to reduce the total cost, subject to the constraints given in 1.3, 1.4, and 1.5. The constraint in 1.4 means that the tracklets within each fence cannot be associated to each other. The constraint in Equation 1.5 means that each tracklet has to be assigned to something.

$$\min_{a \in \mathcal{A}} \sum_{j=1}^J f_j \quad (1.2)$$

$$k_{i,j} \in \{0, 1\} \quad (1.3)$$

$$\sum_{i_s=1}^{i_s=n_s} k_{i_s,j} \leq 1 \quad \forall j, \forall s \quad (1.4)$$

$$\sum_{j=1}^J k_{i,j} = 1 \quad \forall i \quad (1.5)$$

If $k_{i,j} = 1$ it means that the tracklet i is associated to object j . The function f_j depends on the estimated object state, the missed detection probability, the false alarm probability, the uncertainties in the observed and computed values, the number of tracklets, and the total number of fences considered. This so-called fitness function is further elaborated in Section 4.2.

1.3 P vs. NP complexity

The question of P vs. NP, or $P = NP$, has to do with the so-called computational complexity that can be assigned to a given problem. It has been called one of the most important open standing questions in theoretical computer science to date. And it is one of the seven problems that the Clay mathematics institute has defined as the 'millenium prize problems'. A reward of a million US dollars is given to the discoverers of the solution to any of these outstanding problems. Informally speaking the class P consists of problems that are easy to solve, and the NP class consists of problems that are very difficult to solve. Therefore stating that $P = NP$ is essentially saying that all these difficult to solve problems have simple solutions. Another characteristic of a NP problem is that if one provides the solution, this solution is easily verified (with P-complexity). One example of an NP-problem is the prime factorization problem, which is frequently used in cryptography. In this problem a number is given which is a product of n prime numbers, where n is a given integer. The problem is to find these n prime numbers. In order to do so one needs to perform a search through many combinations of prime numbers, which is a procedure with an NP-complexity. However if the solution is provided, one multiplication of the n prime numbers is enough to verify its correctness. If one proved that $P = NP$, the implications would be enormous, already just by considering the field of cryptography.

Saying that problems in P are easy and that those in NP are difficult is quite vague. Therefore a more quantifiable definition is given to the two classes. The P stands for 'polynomial time'. This means that the computation time needed to solve the problem varies as a polynomial function of the problem size. For example if the problem size is denoted by n , the computation time might vary by n^2 . This can also be written as $O(n^2)$, or more generally by $O(n^k)$. Such a relationship is also called the computational complexity of a problem. It is important to note that the P and NP classes are not concerned by the computation time of a problem with a fixed size, they are only considering the computational complexity. The problems in P might have a large computation time (e.g. if k is large), however it will dwindle in comparison to the computation time needed to solve NP problems. The NP acronym stands for 'non-deterministic polynomial time'. In other words, no optimal solution to the problem can be found in a deterministic fashion with a polynomial time complexity. These problems possess an exponential time complexity, e.g. $O(k^n)$. For instance with a $k = 2$ the computation

times with $n = 2$ would be $2^2 = 4$ and $2^2 = 4$ for the P and NP problems respectively. However for a problem size of $n = 10$ it becomes $10^2 = 100$ and $2^{100} = 1.3 \cdot 10^{30}$ for a problem in P and NP respectively. Clearly, any problem that is in NP quickly becomes unsolvable within a realistic computation time.

The majority of the NP problems are in a specific class called the NP-complete problems. The characteristic that sets these problems apart is that they can be adapted to resemble each other. Therefore if a polynomial time algorithm can be found for one of the NP-complete problems, it would be valid for all NP-complete problems since it can be adapted to solve all of the others. If one is capable to solve such a problem in polynomial time it would be a big step towards proving that $P = NP$. An example of a NP-complete problem is the Traveling Salesman Problem (TSP). For a more in depth overview of the advancements in this field of research the reader is referred to (Cook, 2000; Sipser, 1992).

1.4 Existing methods

In this section the most promising existing methods that are currently developed or under development are discussed. Three main categories can be defined being the $S = 2$ methods, the $S \geq 3$ methods, and the statistical methods. Statistical methods aim to model the population of objects through a probability distribution function. They represent a very different approach to the space debris cataloging problem. The current state of the art methods are depicted in Figure 1.10.

In Figure 1.10 the proposed Population Based Meta Heuristic (PBMH) method(s) are highlighted with a red box. The statistical methods represent very different approaches to the problem of cataloging space debris. Instead of attempting to identify each object and its state, they try to model the population of objects with a probability density function (pdf). A recent example is the work done in (Hussein et al., 2012). There are still limits to the algorithm described in that work though, for example the number of objects has to be assumed, and the computational complexity is prohibitive. Other promising algorithms are for instance the Cardinality Probability Hypothesis Density (CPHD) algorithm (Wei et al., 2015), and the multi-Bernoulli approach (Brandon and Vo, 2014). According to the P vs. NP problem theory there is no way to find the optimum solution to a NP problem within

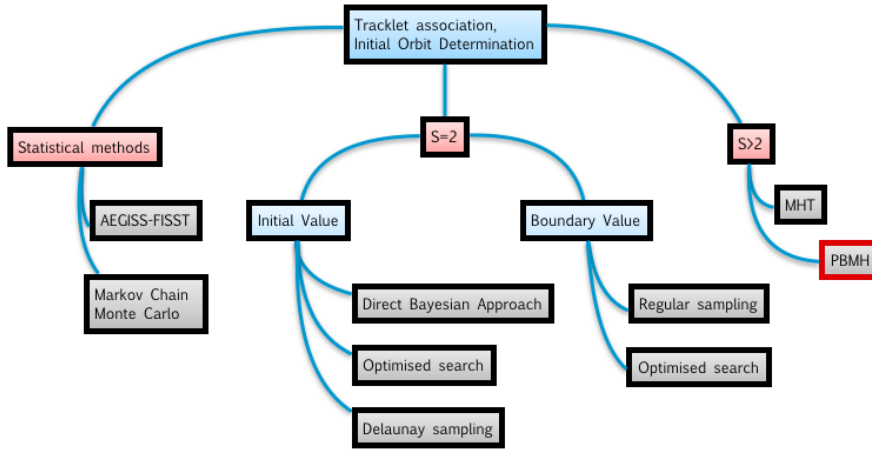


Figure 1.10: An overview of different methods used and under investigation in the domain of space debris cataloging.

a polynomial time in a deterministic way. However there exists a family of algorithms that specifically aims at addressing NP problems without incurring the exponential computational complexity. These are the heuristic algorithms. The heuristic algorithms seek to approximate the optimum solution to the S -dimensional MTT problem with a polynomial time complexity. In the context of the MTT such a heuristic algorithm might be of value. Where the $S = 2$ methods occupy one end of the spectrum and the Multiple Hypothesis Tracking (MHT) method occupies the other, the heuristic algorithms might be able to partly fill the middle of this spectrum. Besides that they have not been studied much in this domain. Two algorithms that have been applied to the MTT problem are the Lagrangian relaxation algorithm (Deb et al., 1997) and the Greedy Randomized Adaptive Local Search (GRASP) algorithm (Robertson, 2001).

The prevailing methods at the moment are those based on the Admissible Region (AR). These methods fall under the $S = 2$ category and are the first to be discussed. After that the MHT algorithm is described which is a part of the $S \geq 3$ category. The MHT algorithm is the currently accepted algorithm used to address the MTT problem in dense regions.

1.4.1 Admissible regions

Throughout the literature two main classes of ARs can be found, being the Initial Value Admissible Region (IVAR) and Boundary Value Admissible Region (BVAR) formulation. In (Siminski et al., 2014) a good overview of several of the most relevant papers and methods involving ARs is presented. An AR is a region within the state space which is defined as containing plausible solutions. For both the initial value problem and the boundary value problem there are four values that are fixed as a result of observations, leaving two values to be estimated in order to define the state of the object. Any set of two additional values which would complete the six element set needed to define an orbit is called an hypothesis. An AR method optimally solves the $S = 2$ MTT problem. This is a problem in P because the number of combinations between two fences depends in a polynomial fashion on the number of tracklets in each fence. These algorithms are therefore favorable in terms of computational complexity, the drawback is that a hard decision to associate two tracklets is made based on the information contained by just those two tracklets. In the majority of the cases (where the object density is relatively low) this works. However in situations with high object densities (e.g. clusters and break-up scenarios) ambiguous solutions start to exist, and wrong decisions can be made.

Initial value approaches

A short overview of the problematic and the proposed methods are presented here, for a more complete explanation the following papers can be considered: (Siminski et al., 2014), (Fujimoto et al., 2014b), (Fujimoto et al., 2013b), (Milani et al., 2010), (Siminski et al., 2013a), (Roscoe et al., 2013), (Fujimoto et al., 2013a), (Tommei et al., 2007), (Siminski et al., 2013b). The initial value problem uses the attributable denoted in Equation 1.1. A point can be made here that the quality of the angular rates and the knowledge of its uncertainty might be questionable. Propagating this uncertainty in time may thus lead to a bad definition of the angular rates in a future epoch, which can lead to false positive correlations (Siminski et al., 2014). This problem can be somewhat diminished by using more observations in one tracklet. Two questions related to this uncertainty are formulated by (Roscoe et al., 2013):

- How accurate do angle rates need to be in order to represent the orbital state directly?

- How accurate do they need to be to properly refine our bounds of the AR?

(Roscoe et al., 2013) shows that the angular rate errors transform linearly to orbit velocity errors, making the answer to the first question relatively straightforward. As for the second question, the admissible region can be padded to account for the possible errors, effectively enlarging the AR.

Six values are needed in order to define the state of the observed object, therefore a hypothesis is made which in the case of the initial value problem is:

$$p = (\rho, \dot{\rho}) \quad (1.6)$$

Where ρ is the range between the observer and the object. With these values the geocentric position and velocity can be determined through the relations 1.7, 1.8:

$$\vec{r}(t) = \vec{R}_{obs} + \rho \vec{u} \quad (1.7)$$

$$\dot{\vec{r}}(t) = \dot{\vec{R}}_{obs} + \dot{\rho} \vec{u} + \rho \dot{\vec{u}} \quad (1.8)$$

where \vec{u} contains the angular positions and \vec{R}_{obs} is the geocentric position of the observer.

The admissible region can be defined by imposing constraints on the possible hypotheses, in this case the range and range-rate values. An example of an AR is taken from (Fujimoto et al., 2013b) where it is defined as:

- $C_1 = \{(\rho, \dot{\rho}) : E \leq 0\}$
- $C_2 = \{(\rho, \dot{\rho}) : 1.03 \leq \rho \leq 8.53, -5 \leq \dot{\rho} \leq 5\}$
- $C_3 = \{(\rho, \dot{\rho}) : 1.03 \leq r_p\}$
- $C_4 = \{(\rho, \dot{\rho}) : r_a \leq 15\}$

where E is the specific geocentric energy of the object, and r_a and r_p are the apogee and perigee heights respectively. Units of length are in Earth radii

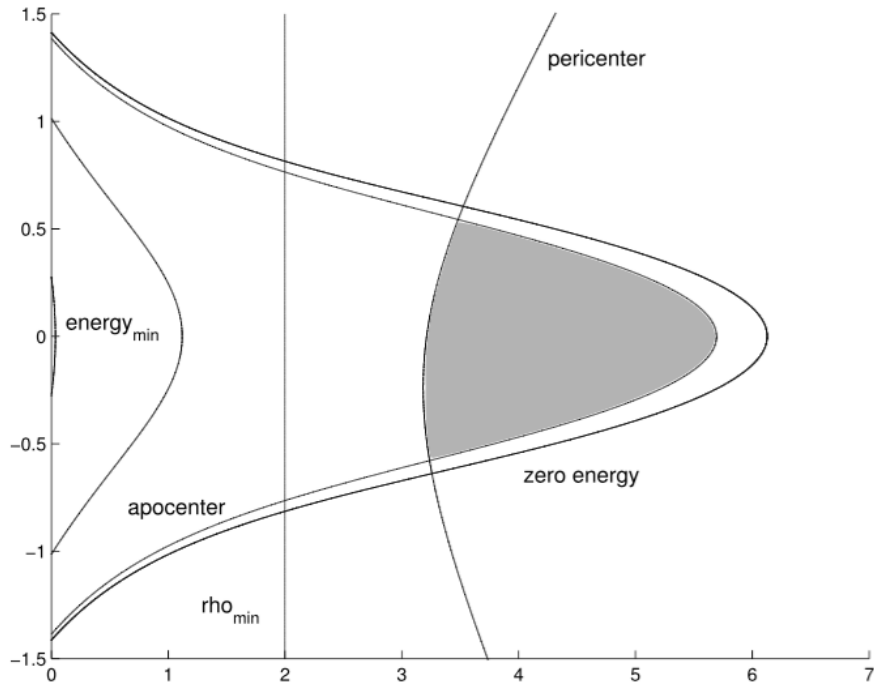


Figure 1.11: An example of an IVAR (Tommei et al., 2009).

and time is in hours. A typical IVAR can look like the one shown in Figure 1.11.

The AR describes a subset of the orbital element space in which the true orbit should lie. In order to find this true orbit different methods are suggested. The first method applied is called the **Virtual Debris** method (Milani et al., 2010; Tommei et al., 2009). Here the AR is discretized to allow for sampling, the chosen method is that of Delaunay triangulation. This results in a sampling as shown in Figure 1.12.

This is done in order to ensure an even distribution of sampling points. For each sample point a 'virtual debris' is generated which is propagated to the epoch of a next observation. For each virtual debris an attribution penalty K_4^i is defined, which depends on the difference between the computed and true attributed quantities at the next epoch. A maximum threshold is set for the attribution penalty under which a virtual debris is accepted as the possible true object, when this happens a decision is made to associate the two tracklets.

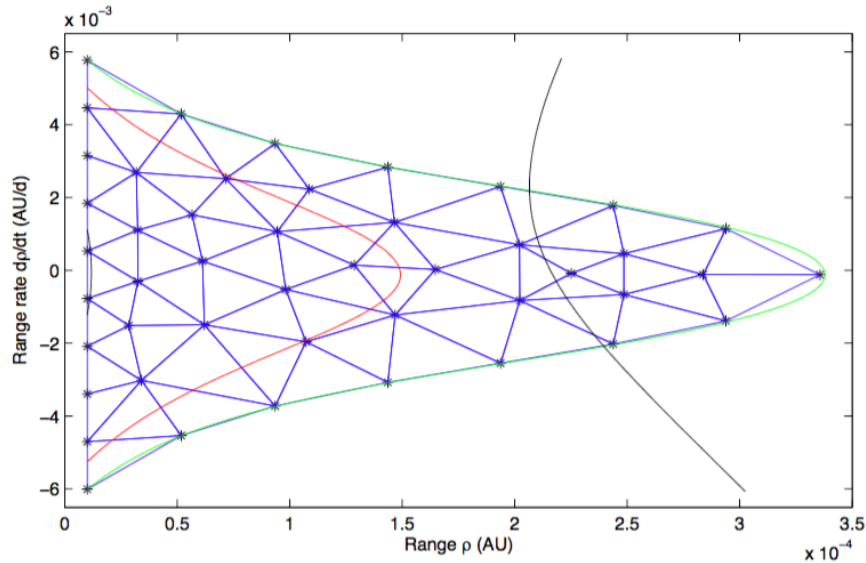


Figure 1.12: An example of the discretization of an IVAR with Delaunay triangulation (Tommei et al., 2009).

Once an orbit based on two attributables is found it has to be confirmed. In order to do this a least squares fit is made to the two tracklets and the orbit is propagated to a time t_3 of a possible third tracklet. If the computed attributable is close enough to the observed attributable the orbit is finally accepted as being true.

The fact that the entire AR has to be sampled, and that each sample is propagated is not very optimal. Another important point to make is that this is not so much a global approach, since the Virtual Debris algorithm will allow for association of the data, but still a least squares fit has to be made in the end (however, this is the case for all of the methods described in this section).

Another approach is described by (Fujimoto et al., 2014b, 2013b,a) called the **Direct Bayesian Admissible Region** method. First an AR for each tracklet is defined and transformed to Poincaré orbital elements. Each AR is discretized into 375.000 subsets. By using the Fokker-Planck equation and using the fact that each AR is a pdf the entire AR can be propagated through time. For each couple of tracklets that are tested the second tracklet is propagated to the epoch of the first tracklet.

If the a posteriori pdf is non zero the tracklets are considered to be possibly correlated. This process is depicted schematically in Figure 1.13.

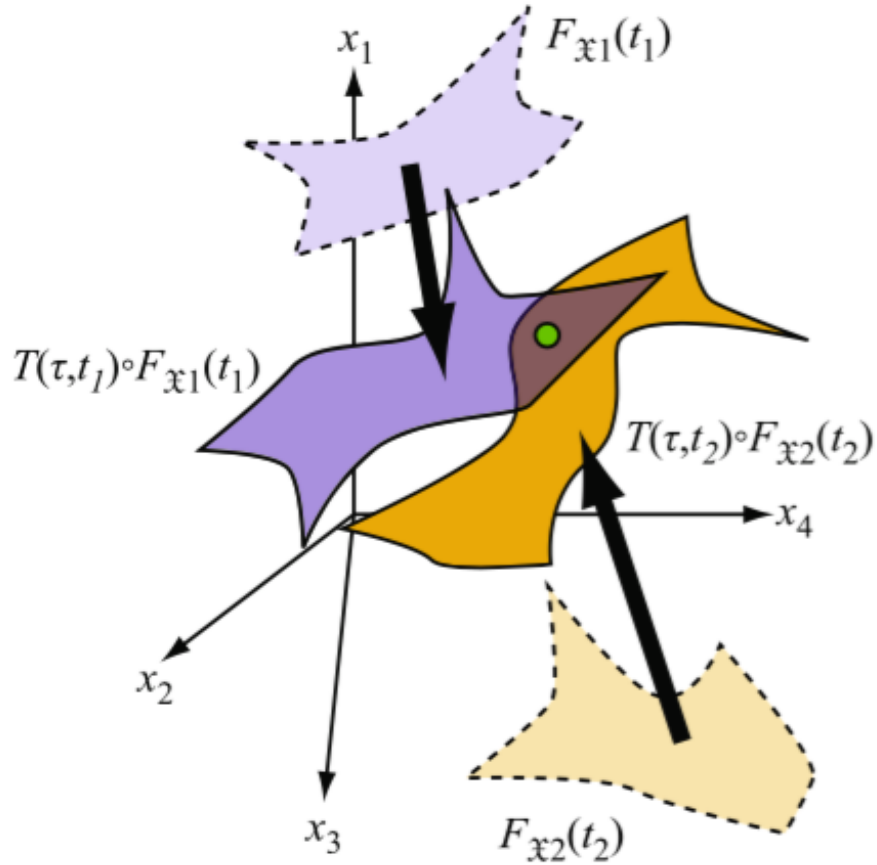


Figure 1.13: A schematic representation of the propagation through time of two ARs in the Poincaré element space. The overlap means that the probability that the two tracklets belong to the same object is non-zero, in this case the two tracklets are associated. (Fujimoto and Scheeres, 2011)

These tracklet pairs are passed on to a least squares estimator which refines the orbit determination and confirms the correlation (if the root mean square (RMS) is below a certain threshold). When such a least squares estimator is used the method is called the Hybrid method. Notice that in this case as well, the AR method does not provide a truly global approach since the orbit still needs refinement and the correlation still needs confirmation.

A test is introduced in (Fujimoto et al., 2014b), which attempts to filter out so called multi-rev (multiple revolutions) solutions. A multi-rev solution is a solution where a tracklet of a faster, lower altitude object, is accidentally associated to an object with a longer orbital period. For instance, a faster orbiting object in a GTO orbit can be mistaken for a GEO object. In order to check for these errors the single observations within the tracklet are used. Provided that the tracklet consists of at least four observations we can see if the RMS within the tracklet has a linear drift. If this is the case it means that the tracklet has a different orientation than the orbit that was fitted to it, giving an indication that the tracklet is wrongly associated.

Instead of sampling the entire AR, (Siminski et al., 2013a) conduct a **Best Hypotheses Search on an Iso-Energy grid**. Two problems are addressed in this method, one is the problem of discretizing the AR. When the time between two tracklets increases, the sampling grid should be more dense. In order to decrease the necessary computation time the new Iso-Energy grid is introduced, which takes samples from lines on which the energy is constant. The second improvement is made by introducing a search algorithm. This means that not the whole AR is sampled, leading to a reduction in the number of propagated hypotheses. The loss function to be minimized is defined by the Mahalanobis distance in 1.9.

$$L(x_0) = \left((a - \hat{a}(x_0))^T \left(\hat{C}(x_0) + C \right)^{-1} (a - \hat{a}(x_0)) \right) \quad (1.9)$$

Where a contains the angular positions at the second epoch, $\hat{a}(x_0)$ is the computed (propagated) angular position. The covariance matrix at the second epoch is C_1 and the propagated value is $\hat{C}(x_0)$, x_0 is the hypotheses that is being tested. This results in a topography such as the one shown in Figure 1.14.

The search algorithm is generally not able to find the global minimum. When the difference between observation epochs is less than one revolution the global minimum and the local minimum coincide. However when the time between observations is large enough to allow for several revolutions there will be several local minima. In this case the AR is split up into several smaller regions based on the number of revolutions (the iso-energy grid is useful here), and each sub space is treated separately. The global minimum is computed by comparing all the local minima and selecting the smallest one. When the AR is divided into too many subsections the topology can

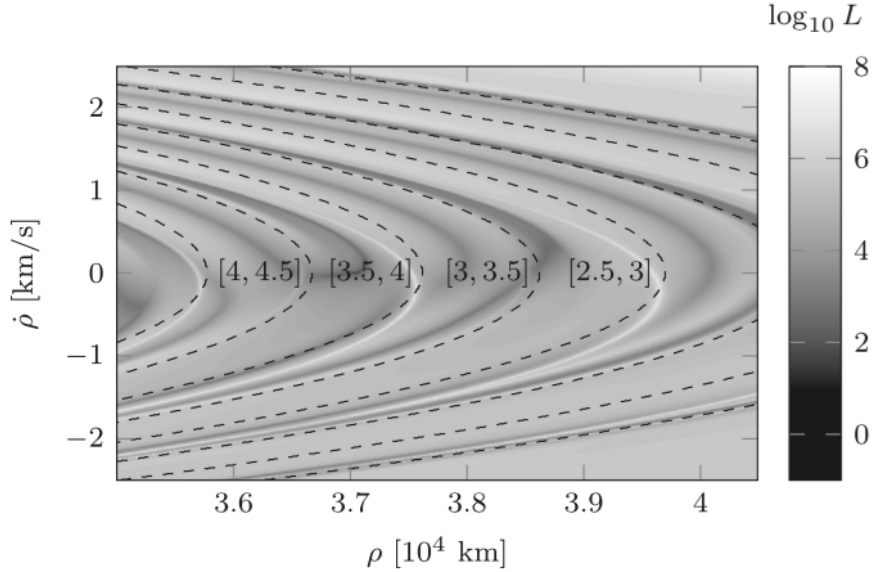


Figure 1.14: The topography of the initial value loss function, over the IVAR. As can be seen, it is challenging to robustly find the global minimum point. (Siminski et al., 2014)

become too difficult to treat robustly with a search algorithm. In such a case a solution could be to perform a parameter transformation, in (Siminski et al., 2014) the use of the semi-major axis a_0 and the relative range $\tilde{\rho}_0$ at the first epoch is suggested. This adaptation helps to remedy the problem up to a certain point, but it does not solve the problem completely.

Boundary value approaches

The admissible region concept is not as much studied in the boundary value problem as it is in the initial value problem. Papers dealing with this problematic are (Schumacher et al., 2013), (Siminski et al., 2014), (Roscoe et al., 2013). The hypothesis consists of a range value per tracklet epoch, and the four fixed values are the angular positions of the first and second tracklet:

$$\bar{p} = (\rho_1, \rho_2) \quad (1.10)$$

For each hypothesis an orbit can be determined by solving the Lambert problem. The strength of the boundary value method is that its AR does not rely

on angular velocity measurements, which can be of doubtful quality (Schumacher et al., 2013) depending on the observation strategy and telescopes used. The angular velocities could be used to come up with potential associations, and only for those candidate tracklet pairs the Lambert problem will be solved. In (Schumacher et al., 2013; Roscoe et al., 2013; Fujimoto et al., 2014a) the entire BVAR is sampled in order to find a good candidate object. Also in this context an optimization is possible, as proposed in (Siminski et al., 2014).

In the optimized approach again a search method is applied to a topography defined by a loss function as given in Equation 1.11.

$$L_{S=2}(m, \bar{p}) = \left(\dot{\bar{z}} - \hat{\dot{z}} \right)^T C_{\dot{\bar{z}}}^{-1} \left(\dot{\bar{z}} - \hat{\dot{z}} \right) \quad (1.11)$$

Where $\dot{\bar{z}} = (\dot{\alpha}_1, \dot{\delta}_1, \dot{\alpha}_2, \dot{\delta}_2)$, and the $\hat{\cdot}$ denotes the computed values. The covariance is given by $C_{\dot{\bar{z}}} = \Sigma_{\dot{\bar{z}}} + \Sigma_{\hat{\dot{z}}}$, where $\Sigma_{\dot{\bar{z}}}$ is the covariance matrix of the attributed values and $\Sigma_{\hat{\dot{z}}}$ is the covariance matrix of the computed values. The loss function is defined with the angular velocities, so even though these velocities are not directly used in the orbit determination they are used here to describe the likelihood of an association.

A given hypothesis defines a Lambert problem, the solution to this problem will give us six orbital elements. With these elements the angular velocities at the observation epochs can be modeled. The resulting topography is much more favorable, as shown in Figure 1.15.

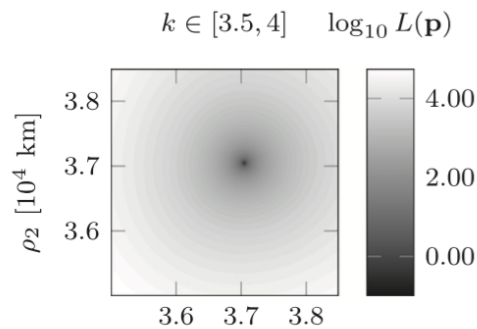


Figure 1.15: The topography of the boundary value loss function, over a BVAR. This is for a set number of revolutions. (Siminski et al., 2014)

The loss function has to be minimized for different numbers of revolution m between the epoch of the first tracklet and the epoch of the last tracklet. The minimum value among those results represents the most likely orbit of the hypothetical object. If the loss function value falls below a certain threshold then those two tracklets are associated to each other, and the resulting orbit is used as the initial orbit of the object.

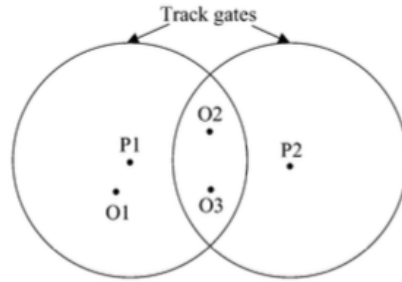
1.4.2 Multiple Hypothesis Tracking

In the MTT community the MHT principle is widely regarded as the accepted method to address the problem (Blackman, 2004). The strength of this method is that it can postpone the decision to associate a measurement until more information is available. Throughout this section the work in (Blackman, 2004) is followed. A lot of literature is written on this topic, a selection of good sources are (Blackman, 2004; Aristoff et al., 2013; Poore and Gadaleta, 2006). It bases its decisions on more information than just two data points (or two tracklets, such as all the $S = 2$ AR methods discussed in Section 1.4.1). This becomes especially important when dealing with closely spaced objects where ambiguous solutions exist. In Figure 1.16 this problem can be seen in a schematic representation. In this figure the two propagated states P1 and P2 are so close together that their uncertainties start to overlap. Within these two uncertainties three measurements, O1, O2, and O3, are made. In this situation it is not clear which measurement should be associated to which object. Therefore more information is needed before making this decision.

The MHT algorithm will initiate a so-called hypothesis for each of the possible combinations between the measurements and propagated objects, including the possibility that the measurements do not stem from either of the propagated objects or that they are false alarms. Each of these hypotheses are then propagated to the next fence where the same process repeats itself. A track is a series of measurements stemming from different fences. Each track can be evaluated according to a certain loss function. It is common to use a likelihood ratio such as the one given in Equation 1.12 which is based on Bayes' rule.

$$\text{LR} = \frac{p(D|H_1)P_0(H_1)}{p(D|H_0)P_0(H_0)} = \frac{P_T}{P_F} \quad (1.12)$$

In this equation $p(D|H_1)$ denotes the probability of measurement D being



O1, O2, O3 = Observation positions
P1, P2 = Predicted target position

Figure 1.16: Three measurements and two propagated states with their uncertainties. Due to the overlap in uncertainty and inclusion of multiple measurements the association becomes ambiguous (Blackman, 2004).

generated by hypothesis H_1 and $P_0(H_1)$ is the a priori probability of the hypothesis H_1 . The H_0 is the hypothesis where all measurements are declared false alarms. The probability that the hypothesis is true is given by P_T , the probability that it consists fully of false alarms is given by P_F . It is general practice to use the Log Likelihood Ratio (LLR). This gives the expression in Equation 1.13. The LLR is also called the track score $L(k)$, where k denotes the fence.

$$\text{LLR} = \frac{P_T}{P_F} \quad (1.13)$$

Now the track score at fence k can be updated as such: $L(k) = L(k - 1) + \Delta L(k)$ where $\Delta L(k)$ is the update value. This update value differs depending on the hypothesis. It can be either a false alarm, a missed detection, or the continuation of the track. In this way the score is kept for each track, adding all these track scores within the same hypothesis gives the hypothesis score. At any time these scores can be converted back to probabilities. It is straightforward now to compare different hypotheses and to choose the most probable one at any time.

Because the MHT method considers all possible hypotheses it will find the optimal solution given a set of measurements. As was discussed earlier in Sections 1.2 and 1.3, this quickly becomes computationally unfeasible. For example, two fences with two measurements each already gives seven possible

hypotheses. Adding a third fence with again two measurements leads to a total number of 87 possible hypotheses (Aristoff et al., 2013). This problem needs to be addressed if the MHT principle is to be used. The manner in which this is done is where MHT methods can differ. One possibility is called N-scan pruning where the search space is pruned by selecting the part that contains the most probable hypothesis to date. At the current fence the decision is taken to cut half of the tree N fences before. The half that contains the best hypothesis so far is kept. This process is depicted in Figure 1.17.

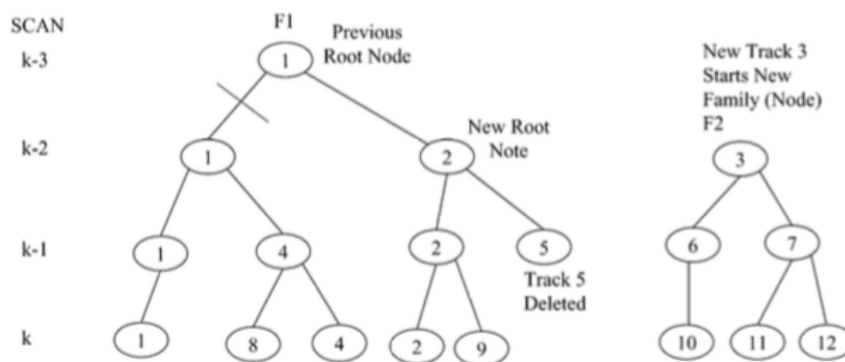


Figure 1.17: A tree structure containing all possible tracks that originate from measurement 1. Here the N-scan pruning technique is depicted, where half of the tree is discarded in hindsight (in this case after two fences), here the left branch is deleted from the tree (Blackman, 2004).

In the example in Figure 1.17 a $N = 2$ N-scan pruning is used. In reality it may be necessary to use a $N = 5$ N-scan pruning (Blackman, 2004). In any case, this still means that the problem is NP-hard, and is thus only applicable to small problems that involve only a few tracked objects. Another way to handle the increase in the number of combinations is the m-best implementation, this approach introduces the parameter m that can tune the compromise that is made between the computation time and the amount of hypotheses that are considered. Here only the m tracks with the highest probabilities of being correct are considered. A different method to restrict the number of combinations is by using track oriented MHT, where the hypotheses on fence $k - 1$ are pruned.

Even though MHT is considered to be the accepted method to address the

MTT problem it is not further investigated during this thesis work. The reasons for this are twofold. Firstly, MHT is a well developed method that has been used extensively (and successfully) throughout many different domains. Simply applying this well known algorithm to the problem at hand did not seem to be of much interest. The second reason is that although MHT is a powerful algorithm it still attempts to optimally solve the $S \geq 3$ MTT problem. Finding the optimum solution is good, but it brings a prohibitively large computation time with it if no simplifications to the problem are made.

1.5 Outline of the thesis

The thesis is structured as follows. In the upcoming chapter arguments are given as to why PBMH algorithms could be of use in the space debris cataloging framework. The algorithms that have been applied to the problem are described afterwards. Each of the algorithms has problem specific design elements. In Chapter 4 these elements, which are the fitness function and the solution representation, are described. In Chapter 3 the initial orbit determination method that is used in the PBMH algorithms is presented. In Chapter 5 the results of the different algorithms are presented. Through three simulated test cases the best algorithm is identified. This algorithm is applied to a set of real data to show that it performs as expected with real observations. The time complexity of the algorithm is discussed, also some preliminary results are presented on possible ways to reduce this time complexity. Finally, in Chapter 6, the conclusion are drawn and the possible future work is outlined.

Chapter 2

Population Based Meta Heuristic methods

In this chapter the algorithms that are applied to the space debris cataloging problem are discussed. All algorithms are discussed in their general form. An interesting characteristic of these algorithms is that they are easily adaptable to solve a wide range of problems (hence the name *meta* heuristic). The problem specific adaptations that are made will be discussed in Chapter 4. First the motivation for using these algorithms is discussed, after which each of the algorithms is explained separately.

2.1 Motivation

Current methods are at two extremes of a spectrum. Either the method is compromising its accuracy (the $S = 2$ methods) or it has an unfavorable NP computational complexity which leads to simplifications that have to be made (MHT (Blackman, 2004)). An overview of these existing methods can be found in Section 1.4. Ideally a new algorithm would be able to cover a new part in the 'spectrum' between these two types of solutions. To that end it is proposed to study meta heuristic algorithms. These algorithms are capable of approximating the optimum solution to problems that are of NP complexity. This approximate solution is then obtained with a P complexity. This means that an approximate solution to the $S \geq 3$ MTT problem can be found with a P computational complexity.

Meta heuristic methods have been explored to a certain extent in the MTT

community. A few examples are the works of (Deb et al., 1997) and (Robertson, 2001) which use the Lagrangian relaxation and GRASP algorithms respectively. And in (Turkmen et al., 2006) and (Chen and Hong, 1997) a Genetic Algorithm (GA) is used to track a few simulated objects with simple dynamics. As mentioned in (Poore and Gadaleta, 2006) the Lagrangian relaxation and GRASP algorithms are quite well developed. The GA and other PBMH methods remain a relatively unexplored topic in the domain of MTT. There is reason to believe that these methods are a promising approach to this problem. The MTT problem is a combinatorial optimization problem. It can be represented in a graph as shown in Figure 2.1.

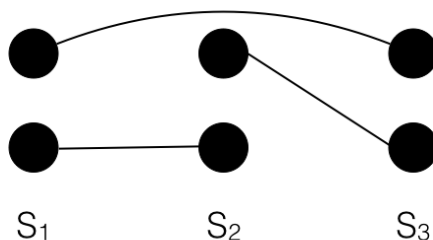


Figure 2.1: A graphical representation of the MTT problem

In Figure 2.1 the $S_{1,2,3}$ represent the first, second and third fence respectively. Each node represents a tracklet, within each fence the tracklets belong to different objects. The nodes can be connected which signifies an association between those tracklets. If a node is not connected to any other node it means that it is a different object that has only one tracklet, or that it is a false alarm. Each connection has a certain weight. This weight can be calculated with the use of a so-called fitness function which is discussed in depth in Section 4.2. The goal is to find which connections minimize the total weight while respecting the constraints set in Section 1.2.

The MTT problem has a lot in common with other well known combinatorial optimization problems. An example is the TSP. In Figure 2.2 a graphical representation of this problem can be seen.

In the TSP each node represents a city, and each edge represents the distance to be traveled between the cities. The goal here is to find the order in which one should visit each of the cities such that the total traveled distance is minimized. This is an NP combinatorial optimization problem that has

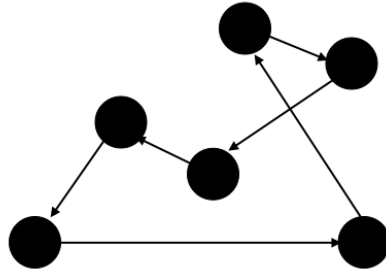


Figure 2.2: A graphical representation of the TSP problem

been addressed by using PBMH algorithms (Moon et al., 2002). Other such problems are the flow shop sequencing problem (Reeves, 1995), and the set covering problem (Beasley and Chu, 1995). As such it is expected that these methods will yield good results when applied to the MTT problem.

2.2 Genetic Algorithms

Genetic Algorithms are being developed since the 1960s (Goldberg, 1999). Nature has a powerful optimization scheme, which is natural selection. All beings pass on their characteristics to the next generations through the form of Deoxyribonucleic Acid (DNA). Two parents can do this by creating a child which inherits a part of each parents' DNA. Whether or not the individual will pass on their characteristics depends on whether or not they are fit enough to do so. By implementing these rules an algorithm can be developed. The flow chart of the GA is given in Figure 2.3.

The algorithms starts by generating an initial population of individuals. An individual is a term to describe a possible solution to the problem at hand. The way in which an individual is defined is important, since it needs to be able to describe any solution within the search space. A group of such individuals make up a population. After initializing the population the algorithm proceeds to evaluate each individual separately. The evaluation is done according to a so-called fitness function. This fitness function is the second important problem specific design element in the GA. Both the definition of the individual and the fitness function for the space debris tracking problem are developed in Chapter 4. The iterations in a GA are called gener-

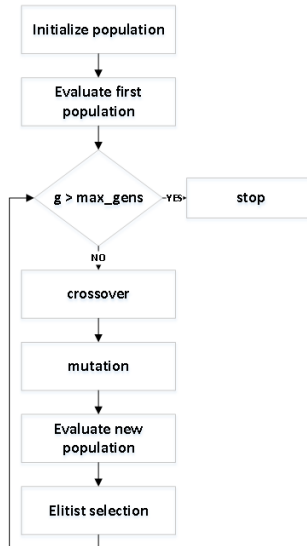


Figure 2.3: A flowchart of a GA

ations, and are denoted by g . In the shown flowchart the stopping criterion is a maximum number of generations max_{gens} . After evaluating the current population a series of operators are used to generate a new population, which represents the next generation. The first operator to be applied is the selection operator.

Selection operator The selection operator selects an individual from the population to be used in the crossover operator. It is important to select the individuals in a proper manner as it will have a large impact on the performance of the algorithm. This operator needs to mimic the process of natural selection. Therefore it is logical to use the fitness values of the individuals and to base the selection on that. Different selection schemes exist (Goldberg, 1999), in this section two of them are discussed. The first version uses the relative fitness of each individual. This relative fitness is given by Equation 2.1. As the algorithm will attempt to minimize the fitness value, the inverse of the fitness value is used to define the relative fitness.

$$f_{y_{rel}} = \frac{1}{f_y} \frac{1}{\sum_{y=1}^Y (1/f_y)} \quad (2.1)$$

In Equation 2.1 $y = 1, \dots, Y$ denotes a specific individual in the population

and f_y is the fitness value of that individual. The relative fitness values describe a discrete pdf over the population. An example of such a distribution is given in Figure 2.4.

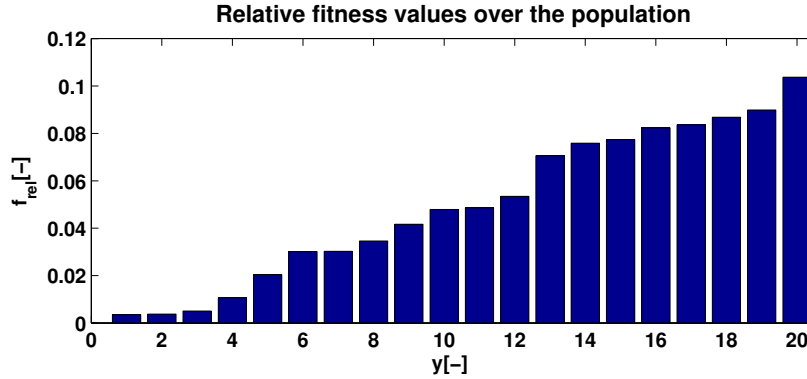


Figure 2.4: An example of a discrete probability density function made by the relative fitness values in the population.

From this pdf two individuals can be selected at random. This manner of selecting individuals is called the *relative fitness selection* operator. A variation of this operator exists where the relative fitness is scaled. This is done to ensure a continuous selection pressure during the run of the algorithm. If the relative fitness values are not scaled it is possible to arrive at a population of individuals that all have similar fitness values. In this case the relative fitness selection operator selects from a nearly uniform distribution and the algorithm becomes similar to the inefficient random walk algorithm. The fitness can be scaled in different ways (Goldberg, 1999). In the current implementation a linear scaling is used as shown in Equation 2.2. Another option would be a polynomial scaling.

$$f_y^{scaled} = a f_y + b \quad (2.2)$$

In Equation 2.2 the parameter a is computed as shown in Equation 2.3.

$$a = \frac{(C - 1) \mu_{fitness}}{\Delta} \quad (2.3)$$

Here C is the scaling parameter, $\mu_{fitness}$ is the mean fitness value and Δ is

given by $\Delta = f_{max} - \mu_{fitness}$. The parameter b is given by Equation 2.4.

$$b = \frac{\mu_{fitness} - (C - 1)}{\Delta\mu_{fitness}} \quad (2.4)$$

So the only parameter that is user defined in the relative fitness selection operator is the scaling parameter C , if linear scaling is used.

An alternative selection operator is the so called *tournament selection* operator. This operator is simpler in design but can be more effective in maintaining the selection pressure during the algorithm run. In this operator two individuals are selected at random from a uniform distribution. These two individuals are then compared to each other and the one with the better fitness value is selected. In Chapter 5 a comparison is made between these two operators.

As an example the TSP is again considered. Taking five cities: Amsterdam (A), New York (NY), Boston (B), Hong Kong (HK), and Paris (P), the task is to find the route that visits all cities once with the minimum distance traveled. An individual in this problem represents a sequence in which the cities are visited, for example:

- $x_1 = (A, NY, B, HK, P)$
- $x_2 = (NY, HK, A, B, P)$
- $x_n = (x_{1,n}, x_{2,n}, x_{3,n}, x_{4,n}, x_{5,n})$

where the last case is the generalized case. The fitness function in this case can be written as in Equation 2.5.

$$f_{TSP} = \sum_{i=1}^{i=4} d_{i,i+1} \quad (2.5)$$

Where $d_{i,i+1}$ is the distance from city $x_{i,n}$ to city $x_{i+1,n}$. In the MTT problem an individual represents a hypothesis on the associations of the tracklets. The fitness function will evaluate how well these association can describe the observations. Chapter 4 goes into more detail on this topic.

Crossover operator The goal of a crossover operator is to exchange information between two or more individuals. In general two individuals are used in this operator, although it is possible to use more (Goldberg, 1999).

Several variations of the crossover operator exist. The most popular ones are discussed in this section. A straightforward version is the *1-point crossover* operator. In this operator the individual is cut at one randomly chosen point. A 'cut' in an individual means that it now consists of two parts that could be exchanged with another individual. Before this can be done the individual has to be represented in a slightly different way. Often a comparison to a roulette wheel is made. This transformation can be seen in Figure 2.5. In this example an individual is used that is represented by a string of binary values. Such a representation can be found regularly. In most cases an individual can be represented by a string of binary, integer, or real numbers.

$$(1\ 0\ 0\ 1\ 0\ 1\ 0\ 1) \longrightarrow \begin{array}{ccc} & 1 & 1 & 0 \\ 0 & & & 0 \\ & 1 & 0 & 1 \end{array}$$

Figure 2.5: Transformation from an individual represented by a string of binary values to a roulette wheel representation.

In this roulette wheel representation a point is chosen at random between two values. This point is then mapped back to the original string representation as shown in Figure 2.6.

$$\begin{array}{ccc} & 1 & 1 & 0 \\ 0 & & & 0 \\ & 1 & 0 & 1 \end{array} \xrightarrow{\quad} (1\ 0|0\ 1\ 0\ 1\ 0\ 1)$$

Figure 2.6: Randomly selecting the cutting point in the roulette wheel and mapping it back to the string representation.

The two individuals are then cut at that point and their halves are exchanged as shown in Figure 2.7.

The 1-point crossover operator is one version of the more general *n-point crossover*, where n number of cutting points are selected at random. The number n cannot exceed the length of the string representation.

A different approach is that of the *uniform crossover* operator. In this operator each entry in the string representation is exchanged with that of another

$$\begin{array}{c}
 (1\ 0|0\ 1\ 0\ 1\ 0\ 1) \\
 (0\ 1|0\ 1\ 1\ 0\ 1\ 1) \\
 \downarrow \\
 (0\ 1|0\ 1\ 0\ 1\ 0\ 1) \\
 (1\ 0|0\ 1\ 1\ 0\ 1\ 1)
 \end{array}$$

Figure 2.7: Exchanging the two halves of the individuals to create two new individuals.

individual with a certain probability p_{xover} . This process is depicted in Figure 2.8.

$$\begin{array}{ccc}
 (1\ 0\ 0\ 1\ 0\ 1\ 0\ 1) & & (0\ 0\ 0\ 1\ 1\ 1\ 0\ 1) \\
 \updownarrow \updownarrow \updownarrow \updownarrow & \longrightarrow & \\
 (0\ 1\ 0\ 1\ 1\ 0\ 1\ 1) & & (1\ 1\ 0\ 1\ 0\ 0\ 1\ 1)
 \end{array}$$

Figure 2.8: Uniform crossover exchanges an element between two individuals with a crossover probability p_{xover}

The GA works because it 'builds' the solution over the number of generations by combining so called 'building blocks' or 'schemata' (Goldberg, 1999). These pieces of information consist of a group of elements within the individual. Examples of such schemata are given in Figure 2.9.

$$\begin{array}{c}
 (1\ *\ * \ * \ * \ * \ * \ *) \\
 (1\ * \ * \ * \ 0\ 1\ * \ * \ *) \\
 (* \ * \ * \ * \ * \ * \ 0\ 1\ 0)
 \end{array}$$

Figure 2.9: A few examples of schemata for an individual represented by nine binary numbers.

In Figure 2.9 the \star denotes a wildcard value. Some schemata will have a better fitness value than others. For instance any individual that starts

with a '1' can be better than those that start with a zero. The crossover operator's ability to combine such schemata is what makes it as efficient as it is. What is special about the GA is that it takes the dependencies between the parameters in the individual into account by using these building blocks. If the operator exchanges building blocks between individuals and the resulting individual has a bad fitness value, then that individual will likely not be selected for the next generation. Likewise, by combining building blocks from highly fit individuals the solution will be improved. For instance, if tracklets 1, 6, and 20 belong to the same object then associating those tracklets will lead to a good fitness value. These three tracklets are thus dependent on each other and form a building block that is a part of the optimum solution.

Mutation operator The crossover operator is powerful but still constrained to only use the information already available. It is said that the crossover operator is the exploitation part of the GA. To include a certain level of exploration the mutation operator is introduced. This operator can change an individual at random and as such can introduce new information into the population. Each element in an individual is mutated with a probability p_{mute} . An example of this operation is given in Figure 2.10.

$$\begin{array}{c} (0\ 0\ 0\ 1\ 1\ 1\ 0\ 1) \\ \downarrow \\ (1\ 0\ 0\ 1\ 1\ 1\ 0\ 1) \end{array}$$

Figure 2.10: An example of the mutation operator.

Elitist selection If this step is added to the regular GA the algorithm becomes an Elitist Genetic Algorithm (EGA). In this case the best few individuals are always copied to the population in the next generation. This ensures that the information contained in the best individuals is conserved. It also means that the best fitness value in the population will either improve or stay the same. In a regular GA this is not the case since the entire population is always replaced. As shown in Chapter 5, this operation can have a significant impact.

2.3 Population Based Incremental Learning

The Population Based Incremental Learning (PBIL) algorithm is a close relative of the GA. However, it uses less operators and in some cases converges faster than the GA. It aims at learning a probability distribution from which highly fit individuals can be sampled. This goal is achieved by using the information that is present in a given population of individuals. The probability distribution P for an individual is defined by the probability that a given element in that individual has a certain integer value. From this distribution a new individual can be sampled by sampling a value at random (according to the probability distribution) for each element. The flowchart for this algorithm is given in Figure 2.11. The probability distribution P is initialized to a uniform distribution. This means that initially $P_{i,j} = 1/i$ for $i = 1, 2, \dots, j$ and $j = 1, 2, \dots, N$, the i and j denote the row and column respectively. From this distribution the first population is sampled. These individuals are all evaluated as usual. In the implementation presented in this work of PBIL, only the very best individual is used to update the probability distribution. This is done according to the update rule in Equation 2.6 (Baluja, 1994).

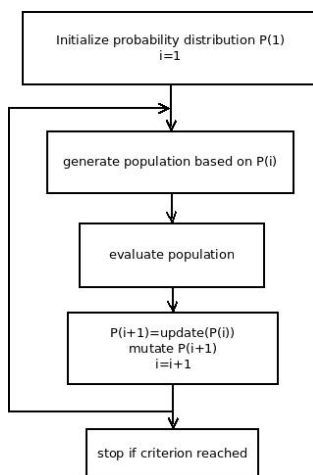


Figure 2.11: Flowchart of the PBIL algorithm

$$P_{i,j} = (P_{i,j} (1 - LR)) + (LR \cdot y_{i,j}^*) \quad (2.6)$$

The probability distribution is denoted by P . The best individual in the

population is denoted by y^* . The parameter LR is the Learning Rate parameter. It dictates how much impact the current best solution has on the probability distribution. Different update schemes exist, where also the worst individual of the population is used. Experimental results showed that the scheme that only uses the best individual in the population performs best. In PBIL there is also a mutation operator in order to ensure versatility in the population and to provide a mechanism to escape from local minima. The mutations can either be performed directly on the individuals (as in the GA) or on the probability distribution itself. In the current implementation it is opted to mutate the distribution. This is done by randomly adding or subtracting a fixed value from a parameter in the distribution P . This algorithm has shown promise, since in many cases it outperforms the standard GA (Baluja, 1994). One pitfall of this algorithm is that it assumes that all the parameters (in this case tracklet associations) are independent from one another. In the GA these dependencies are taken into account, because it uses the crossover operator. The strength of the GA lies with this crossover operator and its capability of successfully combining 'building blocks' (thus with strong inter-dependencies) to come up with promising solutions. It can therefore be expected that if these inter-dependencies are very strong, the PBIL algorithm could have poorer performance than the classical GA. For further reading about the PBIL algorithm the reader is referred to the work of (Baluja, 1994).

2.4 Differential Evolution

Differential Evolution was first thought of in the 1990s in order to address problems with continuous variables. Since then efforts have been made to adapt the original algorithm so that it can be applied to discrete (combinatorial) problems as well (Prado et al., 2010) (Onwubolu, 2009). These efforts have been successful in some cases, however the applicability of Differential Evolution (DE) to combinatorial optimization problems remains debatable. The main difference between DE and GA is the way in which new individuals are produced. In the GA this is done by the crossover and mutation operators. In DE the difference between (real valued) candidate solution vectors is used to guide the search process.

The notation $x_{g,i,j}$ is adopted. Here g denotes the generation number, the

i the number of the individual and j the parameter number within that individual. The x means that it is a current member of the population. From these population members three are selected at random to construct the so called mutation vector $v_{g,i}$ as shown in Equation 2.7.

$$v_{g,i} = x_{g,r_1} + F(x_{g,r_2} - x_{g,r_3}) \quad (2.7)$$

Here the $r_1 \neq r_2 \neq r_3$. The parameter F is the scaling parameter. It dictates in what measure the difference vector will perturb the other solution. Now the mutation vector is used to construct a trial vector $u_{g,i}$. This is done by combining the mutation vector $v_{g,i}$ with $x_{g,i}$ by exchanging parameter values between the two vectors. A certain crossover probability P_x is set. For each parameter 'j' a random number is generated, if this number is lower than P_x then $u_{g,i,j} = v_{g,i,j}$, otherwise $u_{g,i,j} = x_{g,i,j}$. The final step is to evaluate the new vector $u_{g,i}$, if it is better than the current solution $x_{g,i}$ then it will replace it. Otherwise the original solution is kept and copied to the next generation without making any changes. In Figure 2.12 a graphical representation is given of the DE search method (Storn and Price, 1997).

Different schemes exist to create the mutation vector. They differ by the way in which they select the solution vector to change and the number of vectors that are used. In the current implementation the scheme in Equation 2.8 is applied.

$$v_{g,i} = x_{g,best} + F(x_{g,r_2} - x_{g,r_3}) \quad (2.8)$$

As can be seen in the equation, the difference vector is always applied to the best solution in the population.

As mentioned earlier, this method was first conceived for applications on real valued problems. There are two ways to adapt DE to handle discrete variables. One way is to adapt the main operator of the algorithm as given in Equation 2.7. The other way is to transform the solution vectors from the discrete domain to the real domain and to leave the differential operator as it is. Different options to this end are outlined in (Prado et al., 2010) and (Onwubolu, 2009). Here it was opted to transform the solution vectors to the real domain. The transformation used is called the *forward backward transformation*. The idea is to first transform the population from the discrete to real domain using the forward transformation, then DE can be applied as usual, after this the new candidate solution vectors are transformed back to

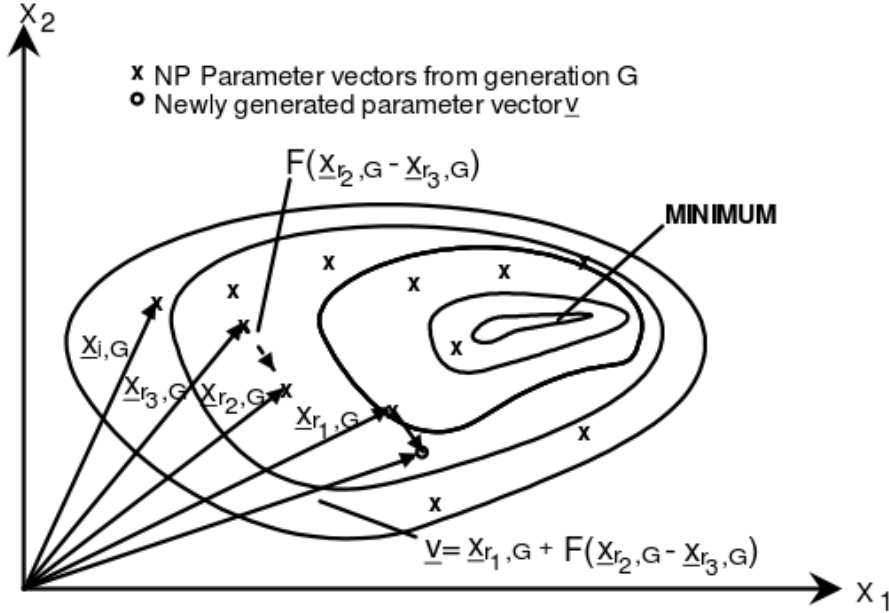


Figure 2.12: Graphical representation of differential evolution (Storn and Price, 1997).

the discrete domain with the backward transformation. In Equation 2.9 the forward transformation is given.

$$\hat{x}_{g,i} = -1 + \alpha x_{g,i} \quad (2.9)$$

In the above equation the α is fixed to a value of $1 \cdot 10^{-3}$. The backward transformation is the inverse of the above equation, giving Equation 2.10.

$$x_{g,i} = \text{ROUND} \left[\frac{1}{\alpha} (1 + \hat{x}_{g,i}) \right] \quad (2.10)$$

This scheme has performed well on several problems such as the traveling salesman problem (Onwubolu, 2009). This reason, together with its easy implementation, led to the decision to use it in the current implementation.

All of the algorithms described in this chapter are applied to the MTT problem. In Chapter 5 the parameter settings and results are presented and discussed.

Chapter 3

Initial Orbit Determination: the OBVIOD method

3.1 Motivation

A MTT algorithm needs to be able to estimate the state of the tracked objects. Which state estimator is preferable will depend on the underlying problem and the object dynamics. Space debris cataloging requires many orbit determinations. Therefore if the orbit determination algorithm that is used is not robust it means that many man hours are spent in attempts to determine the orbit manually. A least squares estimator is a popular choice due to its ability to provide both a maximum likelihood state estimate and the uncertainties of each dimension of that state. In space debris tracking the least squares method is used extensively. However a minimum number of observations are required for the least squares estimator to work in a robust manner. In general the minimum number of observations needed for a least squares estimator to work is equal to the number of parameters that area estimated. However, in the case of space debris often more observations are needed. This is due to the typical observation strategies that are used in space debris tracking. An object is often observed in very short arcs of several observations. Since each short arc of observations only constrains a small portion of the object's orbit (typically about two to five minutes of a 24 hour orbit) the orbit is not well constrained, even if the total number of observations might suggest otherwise. An Initial Orbit Determination (IOD) method is used to provide an initial orbit in a robust manner with

a small amount of observations. This initial orbit can then be used as an initial value in a least squares estimator to further improve the orbit estimate if necessary. Apart from being robust the IOD method should also provide a consist indicator of the quality of the orbit. In this way the quality of orbits with different numbers of tracklets can be compared. Initial orbit determination is a topic which has been addressed extensively. Examples of well known methods are the Gauss and Laplace methods (Beutler, 2004) which use three optical observations $(\alpha, \delta)_{1,2,3}$ to determine the six orbital elements. Although these methods are interesting, they do not exploit the manner in which space debris is observed. In this work a new IOD method is developed called the Optimized Boundary Value Initial Orbit Determination (OBVIOD) method. It relies heavily on previously developed work on the optimized boundary value problem (Siminski et al., 2014).

A method is needed that fulfills the aforementioned constraints of robustness, consistency, and initial orbit quality. Besides that it needs to be able to exploit the use of tracklets as apposed to single observations. The current state of the art $S = 2$ methods that are discussed in Section 1.4.1 work exclusively with tracklets and attributables (as opposed to the other methods). Therefore these methods became the focus of this study. These methods use two tracklets at a time and seek to determine whether or not these tracklets stem from the same object. The algorithm developed during this thesis aims to do the same, but for two or more tracklets. This means that whichever $S = 2$ method is deemed to be best suited for our needs, it will have to be generalized such that it can work with two or more tracklets at a time. Two categories of $S = 2$ methods can be defined. One is the IVAR approach and the other is the BVAR approach. The pros and cons of the most important methods in each category are briefly discussed in the following.

The main advantage of the boundary value methods is that they do not rely on the attributed angular rates $\dot{\alpha}$ and $\dot{\delta}$. These are two attributed values which are randomly distributed with a large uncertainty (e.g. a normal distribution with a Gaussian uncertainty), so if an AR is based on them (as is done in the initial value method) the entire AR will be off (Schumacher et al., 2013). The question of how to include these uncertainties into the IVAR is still being addressed (Roscoe et al., 2013). Therefore the boundary value formulation will in general be more robust to uncertainties in the observations. Of the different boundary value methods the optimized version

developed by (Siminski et al., 2014) seems the most promising. Therefore it is chosen as a basis for the OBVIOD method. Note that this choice is based only on a literature study, and not on a direct comparison of performance.

3.2 The optimized boundary value method

The optimized boundary value method (Siminski et al., 2014) is the $S = 2$ algorithm that best fulfills the criteria that were set. In order to be used in an $S \geq 3$ MTT algorithm it has to be generalized to be able to determine an orbit with two or more tracklets. As there are now more than two tracklets, there is a choice to be made. Which tracklets should be involved in the definition of the Lambert problem? From the point of view of the IOD it is favorable to cover a maximum portion of the arc in order to maximize the accuracy of the solution. Therefore it is chosen to always use the $(1, N)$ tracklet pair to define the Lambert problem. Because the Lambert problem is such a vital component of the OBVIOD method it is described separately in the next subsection.

3.2.1 Lambert problem

For completeness' sake the Lambert problem will be briefly outlined. There is a wealth of information to be found on this topic, it is a relatively old problem that has been addressed extensively. In case the reader is interested in learning about it the following sources are recommended: (Battin, 1999; Gooding, 1990; Lancaster and Blanchard, 1969). In (Gooding, 1990) the Lambert problem is stated as follows:

An unperturbed orbit, about a given inverse-square-law centre of force, C say, is to be found connecting to given points, P_1 and P_2 , with a flight time $\Delta t = t_2 - t_1$ that has been specified. There will always be at least one solution, and the actual number (N say) depends on the 'kinematic geometry' of the triangle CP_1P_2 and the value of Δt ; it will be assumed, for convenience and with no loss of generality, that $\Delta t > 0$.

From this description the input quantities are defined as $r_1 = P_1 - C$ and $r_2 = P_2 - C$, the force of the central point C is described by the central gravity coefficient μ , and the length of the orbit arc Δt . The angle ϕ is the

angle between the vectors r_1 and r_2 and can be between $0 < \phi < 2\pi$. If the Δt is large enough, more than one solution are possible. These additional solutions are called multi-revolution solutions and they occur in pairs. The number of revolutions is denoted by m . Therefore the output quantities are $2m + 1$ solutions, each consisting of a radial velocity V_r and a transverse velocity V_t at epoch t_0 and t_1 .

In order to find a solution to this problem an iterative procedure is needed. One parameter needs to be defined on which to iterate. The choice of this parameter has a significant impact on the quality of the solution. Since the semi-major axis a is directly related to the orbital energy, it is a likely candidate. However, this would result in having solutions in pairs or no solutions at all (depending on the geometry of the problem). Since at least one solution is required (and possible), this parameter choice does not fulfill the requirements. Instead of using a a new parameter is defined that is dependent on a . It is given in Equation 3.1 (Gooding, 1990).

$$x^2 = 1 - \frac{s}{2a} \quad (3.1)$$

Where s is the semi-parameter of the triangle CP_1P_2 . The variable that is iterated can vary from one Lambert problem solver to another. Although the parameter x performs well (Lancaster and Blanchard, 1969; Gooding, 1990; Izzo, 2015), there is still a search for a parameter that might improve the solution procedure, such as in (Sun et al., 1987). If the length of the orbit arc is expressed in a non-dimensional form as in Equation 3.2 Figure 3.1 is obtained.

$$T = \sqrt{\frac{8\mu}{s^3}} \Delta t \quad (3.2)$$

The value of T is known, and the x values that correspond can be found. This requires an iterative search, where the initial guess of x plays an important role in terms of convergence speed. Note that for more than one revolution there can be one or two solutions to x . In the case where two solutions are possible they are called the left and right hand solutions. The iterative search can for instance be a Newton-Raphson method, the Halley iteration method (Gooding, 1990), or Householder iterations (Izzo, 2015). After x is known, the V_t and V_r at times t_1 and t_2 can be computed. This computation can be different between different solvers as well. All in all, a Lambert solver can differ in four things: the choice of the iteration parameter, the iteration

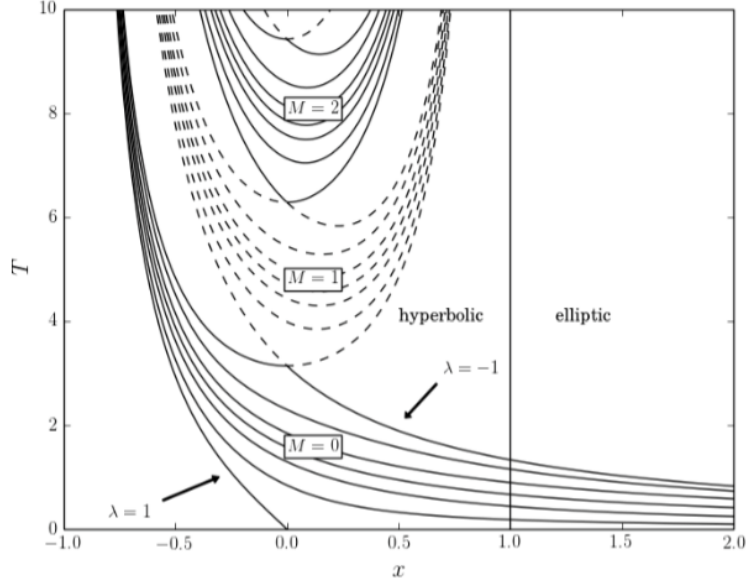


Figure 3.1: The non-dimensional orbit arc length T versus the parameter x (Gooding, 1990).

scheme, the initial value of the iteration parameter, and the computation of the final velocities.

3.3 Generalization of the optimized boundary value method

The optimized boundary value method was developed in the work of (Siminski et al., 2014), it is described in Section 1.4.1. That method is able to handle two tracklets at a time, as such it needs to be generalized so that it can handle two *or more* tracklets. This generalization is straightforward. Instead of having two tracklets, now there are N tracklets, each with an attributable. This gives us the vector in Equation 3.3.

$$\bar{\Theta} = (\bar{\theta}_1, \bar{\theta}_2, \dots, \bar{\theta}_N)^T \quad (3.3)$$

Furthermore, the assumption that the uncertainties on the single observations is Gaussian still stands. Consequently the uncertainties on all the

attributables are assumed to be Gaussian as such:

$$\bar{\theta}_t = (\alpha, \delta, \dot{\alpha}, \dot{\delta}) \sim \mathcal{N}(\mu_{\bar{\theta}_t}, \Sigma_{\bar{\theta}_t}) \quad (3.4)$$

In the original $S = 2$ version of this algorithm the loss function was given by Equation 1.11. Since the errors are all still Gaussian, the same loss function can be used for the more general case. The result is given in Equation 3.5.

$$L_{S \geq 2}(m, \bar{p}) = (\bar{\theta} - \hat{\theta}(m, \bar{p}))^T C_{\bar{\theta}}(m, \bar{p})^{-1} (\bar{\theta} - \hat{\theta}(m, \bar{p})) \quad (3.5)$$

Where the covariance is given by $C_{\bar{\theta}} = \Sigma_{\bar{\theta}} + \Sigma_{\hat{\theta}}(m, \bar{p})$ and $\bar{p} = (\rho_1, \rho_N)$. The $\Sigma_{\hat{\theta}}(m, \bar{p})$ is the covariance of the computed attributables, it can be found by propagating the uncertainties of the individual observations. Unfortunately this poses a problem. In order to propagate the uncertainties, the partial derivatives between the computed attributables and the observations have to be known. A finite difference scheme is used to obtain these (Vallado, 2007), no analytic method can be used because of the iterations used in the Lambert problem solver. In a finite difference scheme one variable is slightly altered and the parameters are estimated again. The difference between the original and the now slightly different parameter values are used to estimate the partial derivative with respect to the variable that was changed. In this case the variables to change are the attributed values. For a slight change in one of these values a new orbit has to be determined to find the new parameter values. However, to determine that orbit the uncertainties on the computed attributables are needed as well. This means that in order to get the full covariance matrix one already needs to have it. For the angular rates this is not a large problem, because even without the uncertainties on the computed rates the minimized loss function is still distributed very close to a χ^2 distribution. For the angular positions, however, this does pose a problem, since their uncertainty certainly does have an impact on the distribution. In the case of the angular positions it is opted to only take the partials with respect to the attributed positions into account. These attributed positions are slightly changed (one standard deviation), and a new orbit is computed through a Lambert solver (keeping the range values fixed). This method approximates the covariance close enough to have a near χ^2 distribution. The resulting distribution with the approximated covariance of the computed angular positions is shown in Figure 3.2.

When the covariance is neglected the results in Figure 3.3 are obtained. It

3.3. GENERALIZATION OF THE OPTIMIZED BOUNDARY VALUE METHOD 49

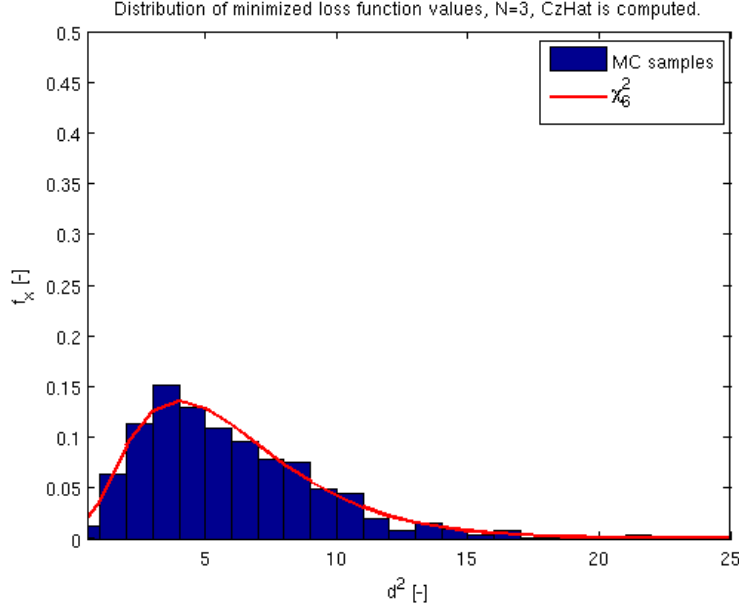


Figure 3.2: The pdf of the Mahalanobis distance with three tracklets. The covariance of the computed angular values is not neglected in these results. Therefore the values are distributed according to a χ^2 distribution. The covariance of the computed angular rates is neglected.

is clear that the values are no longer χ^2 distributed, their average value is $3.4 \cdot 10^{13}$.

By adding new values to the loss function its topography might change. In Figure 3.4 the topography of this loss function for three tracklets can be seen. The loss function is minimized with the use of the Broyden-Fletcher-Goldfarb-Shanno (BFGS) method (Press et al., 2007). In the current implementation the initial point is computed with Equation 3.6, as done in (Siminski et al., 2014).

$$\rho_i^{init} = -c_i + \sqrt{c_i^2 + a(m)^2 - \|\bar{r}_{s,i}\|^2} \quad (3.6)$$

In Equation 3.6 the i subscript denotes the tracklet number 1 or N . The c_i is given by $c_i = \bar{r}_{s,i} \cdot \bar{u}_i$. The vector from the Earth's center to the observer for tracklet i is given by $\bar{r}_{s,i}$. The $a(m)$ is the semi-major axis that corresponds to m revolutions between epochs t_1 and t_N .

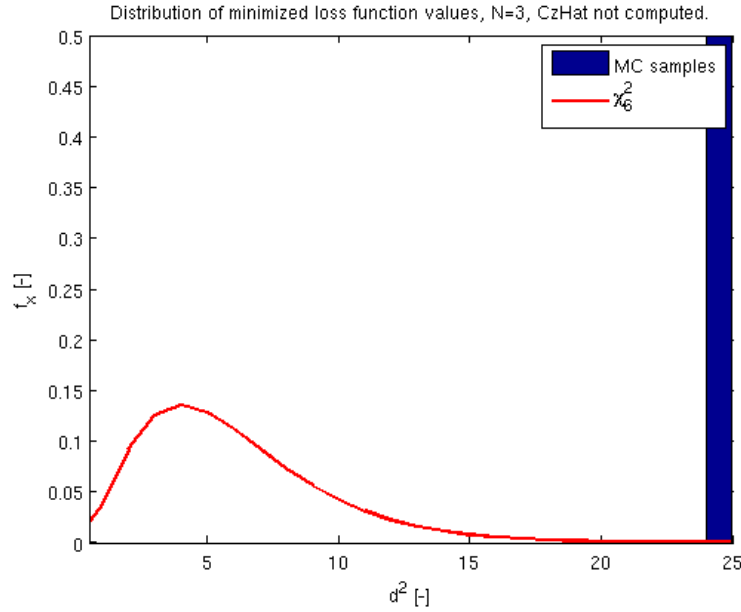


Figure 3.3: The pdf of the Mahalanobis distance with three tracklets. The covariance of the computed angular values is not neglected in these results. Therefore the values are distributed according to a χ^2 distribution. The covariance of the computed angular rates is neglected. It is seen that neglecting the covariance of the angular rates leads to Mahalanobis distance values that are not χ^2 distributed.

As seen in this figure, the topography is again smooth and has one global minimum point within the given range bounds. It is important to note that the fact that there is one unique minimum point has never been proved mathematically. Therefore caution is advised when applying this method. In Figures 3.5 to 3.10 some example topographies can be seen for an object in GEO, two tracklets are used in all of these cases. The object follows a Keplerian motion. Note that in each of the figures the range values cover a large range of values. This is done in order to get a better idea of what the topography looks like when very loose constraints are imposed. In Figure 3.5 the topography can be seen in the case where there is one hour between the epoch of the first and second tracklet. It shows that there is a large region where no valid solutions are possible (the uniform white region), this is because the orbits here are hyperbolic. The ρ_1 is strongly correlated to the ρ_2

3.3. GENERALIZATION OF THE OPTIMIZED BOUNDARY VALUE METHOD51

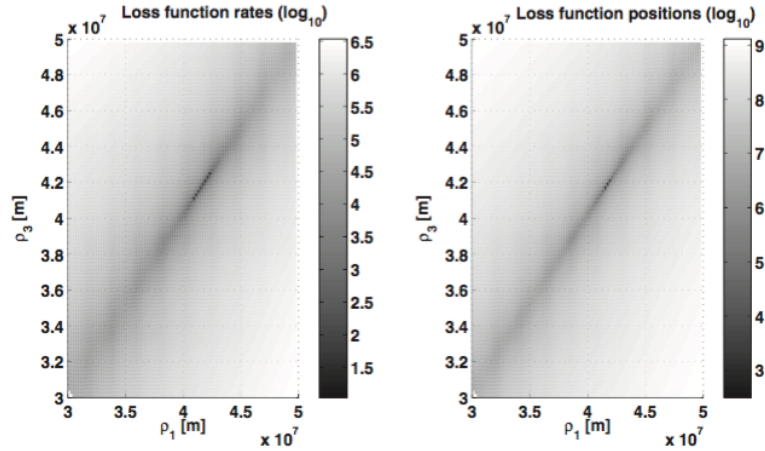


Figure 3.4: The topography of the loss function for a set of three tracklets. The topography of the loss function of the angular rates (left) and angular positions (right) are shown separately in order to show that they are similar. The total topography would be the summation of the two topographies shown.

in the 1 hour case. This is a typical behavior that becomes stronger when the length of the orbit arc between two tracklets becomes smaller. Any deviation from the diagonal trend signifies (roughly) an increase in eccentricity of the resulting orbit. This is why the loss function value is relatively small along the diagonal, and sharply increases (until the orbits are not valid anymore i.e. they become hyperbolic) towards the off diagonal. The topography in this case contains one global minimum which can be found with a typical gradient descent algorithm.

Figure 3.6 depicts the situation where there is six hours of separation between the two tracklets. For a GEO object this is near the optimum problem geometry (a quarter revolution between the tracklets), as will be further explained in Section 3.5. This favorable geometry is reflected in the topography. It is a smooth topography with one global minimum point, and almost the entire region contains valid orbit solutions. In Figure 3.8 the effect of adding a third tracklet can be seen. Due to the addition of the attributable of the third tracklet, the topography becomes much steeper. The angular position of the third tracklet is not included in the definition of the Lambert problem, therefore there will be a non zero difference between the attributed and

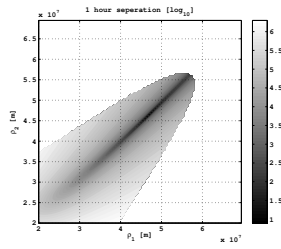


Figure 3.5: Loss function topography, $N = 2$, separation is 1 hour.

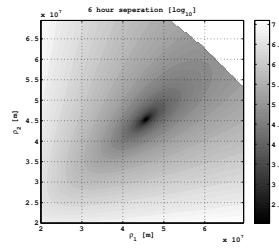


Figure 3.6: Loss function topography, $N = 2$, separation is 6 hours.

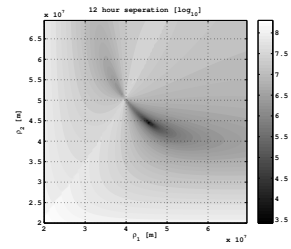


Figure 3.7: Loss function topography, $N = 2$, separation is 12 hours.

computed angular position for that tracklet. However, the topography is still favorable as it remains smooth and still contains one minimum point.

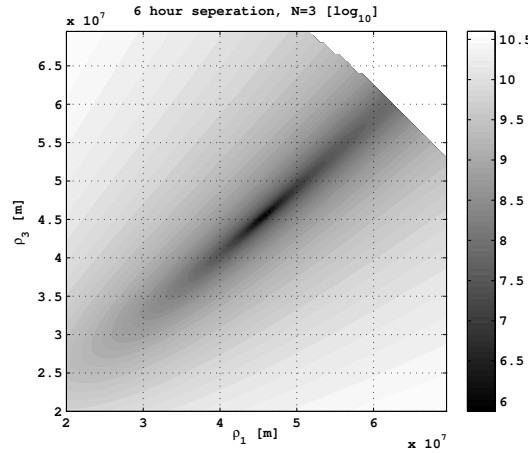


Figure 3.8: The topography of the loss function for three tracklets where the first and last tracklets are spaced at six hours from each other, the third tracklet is positioned halfway between the first and the third tracklet.

The Lambert problem has two geometries where problems occur. These are at an angle of $\phi = \pi + 2\pi k$ and $\phi = 2\pi k$ where $k = 0, 1, 2, \dots$. In both cases the orbital plane is not defined by the two geocentric vectors (together they form a straight line), and therefore an infinite amount of solutions exists. Figure 3.7 shows the situation where two tracklets have 12 hours between them. This is close to the $\phi = \pi$ case. As can be expected, the topography in this case is not favorable, as it shows multiple minimum points. In this

3.3. GENERALIZATION OF THE OPTIMIZED BOUNDARY VALUE METHOD53

case a gradient descent algorithm might not be sufficient to find the global minimum point. This situation should therefore be avoided. Besides the difficult topography, this geometry brings other problems with it as discussed in more detail in Section 3.5. In Figure 3.9 a third tracklet is added in between the first and third tracklet. The first and last tracklets are spaced at 12 hours from each other. It is clear that the third tracklet has a large influence on the topography. However, from this example it does not seem to solve the problem of having multiple minimum points. Therefore, also with more than two tracklets, the difficult geometries of the Lambert problem should be avoided. A solution to this problem could be to identify a tracklet pair that has a good geometry, and use that pair in the orbit determination. This could be an interesting improvement, but it is not considered in this work.

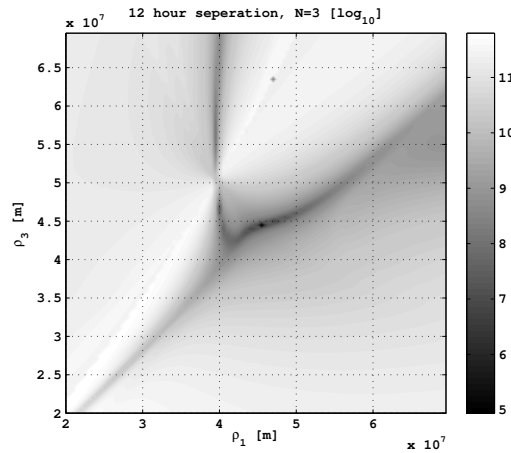


Figure 3.9: The topography of the loss function for three tracklets where the first and last tracklets are spaced at 12 hours from each other, the third tracklet is positioned halfway between the first and the third tracklet.

Figure 3.10 shows the topography where the two tracklets are spaced at 24 hours from each other, close to the $\phi = 2\pi + 2\pi k$ point. Here again the topography is much more complex than in the more favorable geometries. Knowing that it is possible to have a topography with several minimum points means that the choice of the initial point is important to ensure that the gradient descent algorithm converges towards the correct solution.

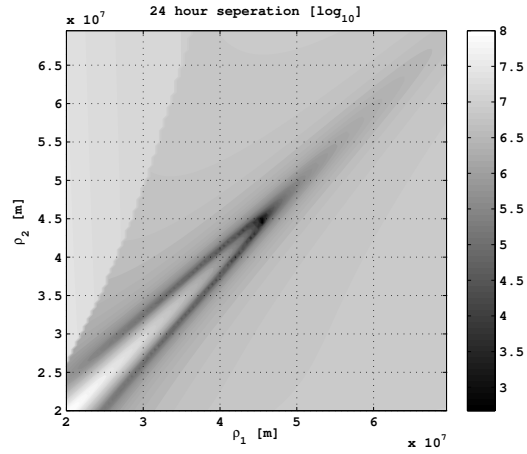


Figure 3.10: The topography of the loss function for two tracklets that are spaced at 24 hours from each other.

The main difference with the original $S = 2$ method is the number of elements in the vector of Gaussian random variables. In the original $S = 2$ algorithm there were four values, being the angular rates at the first and second epoch. Now the vectors contain $4N$ values, since angular positions are now also taken into account. What this means is that the values of the minimized loss function will be distributed differently. They will still be χ^2 distributed, but the degree of freedom f will be different, depending on the number of tracklets used. The attributed angular positions of the first and last tracklet are exactly intersected by the computed values. This is because these two tracklets are used to define the Lambert problem. Therefore these two angular positions (that is, four values) are not to be counted as a degree of freedom. Furthermore, two parameters are being estimated, which are the (ρ_1, ρ_N) range values. These two parameters should also be subtracted from the total degree of freedom. In the end the degree of freedom of the χ^2 distribution can be computed by $f = 4N - 6$. Examples of such distributions can be found in Figures 3.11 and 3.2.

Note that this method is reminiscent of the Gooding IOD method (Gooding, 1996). In the Gooding method three lines of sight are used to determine an orbit. The range values at the first and third epochs are varied. For each range hypothesis a Lambert problem is formulated and solved, the resulting orbit is used to compute the angular position of the object at the second

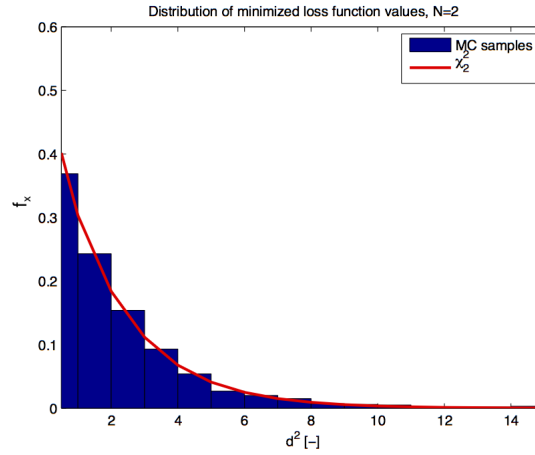


Figure 3.11: The probability distribution of the minimized loss function values for two tracklets.

epoch. The distance between the observed and computed angular positions at the second epoch has to be minimized. That procedure is similar to that of the method described in this section, except that the OBVIOD method also minimizes the distance between the attributed and computed velocities. On the other hand the OBVIOD method is also similar to the Siminski method (Siminski et al., 2014). Where the Siminski method only used angular velocities in its loss function, OBVIOD can use both angular velocities and angular positions. Other similar methods are those of (Virtanen et al., 2012; Muinonen et al., 2012), where the range values are varied in order to best fit the angular positions (as with the Gooding method), but they are varied through a (Markov-Chain) Monte-Carlo scheme.

3.4 Orbit determination performance

Although the quality of the determined orbits is not the first priority of the OBVIOD method, it is examined to some extent in this section. The performance of the algorithm is quantified by looking at how close the computed orbit can approximate the true geocentric positions of the object at the observation epochs. This error in the geocentric position is directly related to the errors in the right ascension α , declination δ , and range ρ . Several tests are performed. In each test the first and last tracklets are spaced on different

intervals. From one twelfth of an orbital period to a full orbital period. These tests are performed for both a GEO and a MEO object. Note that the error in the geocentric position is directly translatable into an error in the (α, δ, ρ) variables. However, the geocentric position lends itself to easier interpretation and is therefore used in these tests. The expected error in the position (tangential to the line of sight), resulting from an error in the astrometric position (α, δ) , is given by $\sigma_p = \rho\sigma_0$, where σ_p is the error in the position and σ_0 is the astrometric error. For a GEO object, observed with an error of $\sigma_0 = 1''$, this gives $\sigma_p = 204$ m of position error for a range of $\rho = 42000$ km. For a MEO object that error is $\sigma_p = 126$ m. Note that this position error only gives an idea of the order of magnitude in two dimensions, as no information is available on the range values. A RMS error is computed as shown in Equations 3.7 and 3.8.

$$\|p - \hat{p}\|_i = \sqrt{(x - \hat{x})_i^2 + (y - \hat{y})_i^2 + (z - \hat{z})_i^2} \quad (3.7)$$

In Equation 3.7 a $\hat{\cdot}$ hat accent denotes the computed values.

$$\text{RMS}_p = \sqrt{\frac{1}{n} (\|p - \hat{p}\|_1^2 + \|p - \hat{p}\|_2^2 + \dots + \|p - \hat{p}\|_i^2 + \dots + \|p - \hat{p}\|_n^2)} \quad (3.8)$$

Here n denotes the total number of observations. Finally the relative RMS is defined as in Equation 3.9.

$$\text{RMS}_{p_{rel}} = \frac{\text{RMS}_p}{\|p\|_{average}} \quad (3.9)$$

Where $\|p\|_{average}$ is the norm of the average geocentric position. By using the relative RMS orbits from different regimes can be compared. Each object is modeled by a spherical space craft. The absorption and reflection coefficients are $c_a = 0.5$ and $c_r = 0.8$ respectively. A gravity field model up to degree and order four is included in the force models. Any degree and order beyond that point will introduce perturbations on the millimeter level for MEO objects (and less for objects in GEO), which is well below the measurement noise. No third body perturbations are taken into account.

In Figure 3.12 a comparison between the relative error versus the time between two tracklets can be seen. The comparison is between a low noise case

where $\sigma_0 = 1''$ and a high noise case with $\sigma_0 = 10.0''$. The low noise case is representative for the ZimSMART telescope (Fujimoto et al., 2014b; Herzog et al., 2010), and the high noise case represents a worst case scenario that is typically not encountered in the space debris tracking.

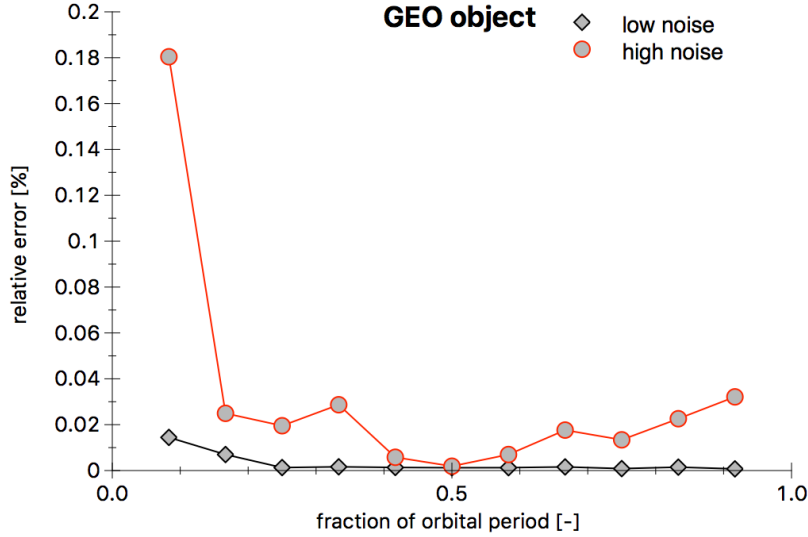


Figure 3.12: A comparison between a low noise and high noise case. Two tracklets spaced at varying fractions of an orbital interval were used in these orbit determinations, the results shown are the average results over 1000 simulations. The AMR value is $0.02 m^2/kg$.

The noise clearly has an impact on the results. The error increases towards an orbit arc length of zero and of a full revolution. This is due to the unconstrained orbital plane in the Lambert problem that exists at these points. The method does not seem concerned by the infinite possible solutions of the Lambert problem that exists for half an orbital period. A relative error of 0.02% corresponds roughly to 8400m. Figure 3.13 shows a comparison between a GEO object with an AMR value of $0.02 m^2/kg$ and an object with $AMR = 2.0 m^2/kg$. The initial orbital parameters of the objects are identical. The high AMR value is one hundred times larger than that of a typical operational satellite in GEO. Higher values than this are present in the space debris population (Schildknecht et al., 2005), but they are rare.

Interestingly, the AMR does not necessarily have a bad impact on the error of the orbit determination. This is explained by the estimated orbital ele-

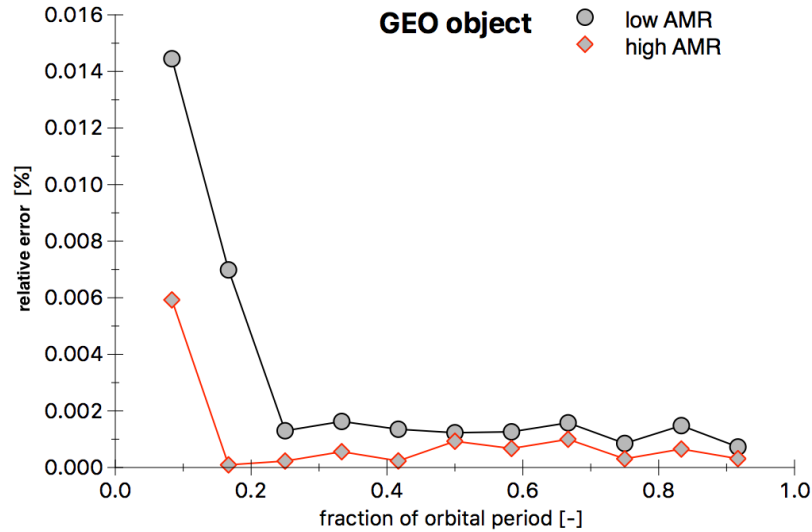


Figure 3.13: A comparison between a low AMR and high AMR case. Two tracklets were used in these orbit determinations, the results shown are the average results over 1000 simulations. A low noise value is used of $\sigma_0 = 1''$.

ments which are able to absorb the effect that the AMR introduces. This is further shown in the results presented in Section 3.5. Figure 3.14 shows the comparison between an object in MEO and one in GEO.

The results in Figure 3.14 show that the orbital regime does not have a large impact on the quality of the orbit determination. Note that the time is measured in fractions of the orbital period, and that the orbital period of a MEO object is approximately half that of a GEO object. Certainly the MEO object is subject to stronger perturbations due to the non-homogeneous gravity field, but these effects are absorbed by the orbital elements. Also realize that the errors displayed are the relative errors. So the absolute error will be less for the MEO object than for the GEO object. Figure 3.15 displays the results for a MEO object with low noise and a high noise on the measurements.

Finally, in Figure 3.16 a comparison can be seen between a low and a high AMR case in MEO. As can be seen, it behaves quite similarly to the GEO case, where the AMR does not necessarily lead to a worse precision with respect to the geocentric positions.

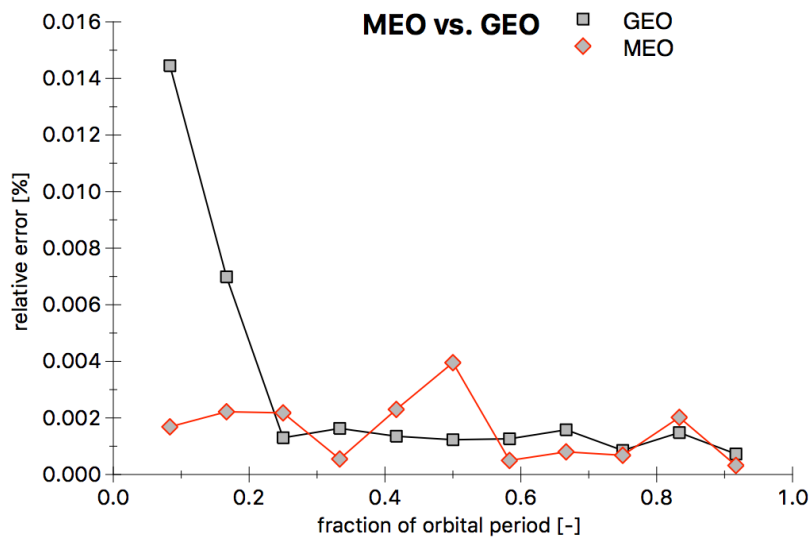


Figure 3.14: A comparison between an object in MEO orbit and an object in a GEO orbit. Two tracklets were used in these orbit determinations, the results shown are the average results over 1000 simulations. The low noise and low AMR values are used in this plot.

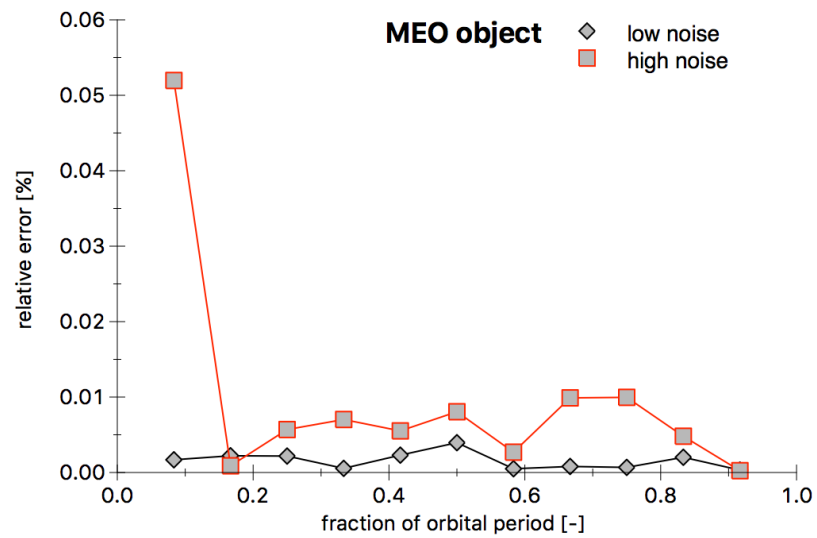


Figure 3.15: A comparison between a high noise and a low noise case in MEO. Two tracklets were used in these orbit determinations, the results shown are the average results over 1000 simulations. The low AMR value is used in this figure.

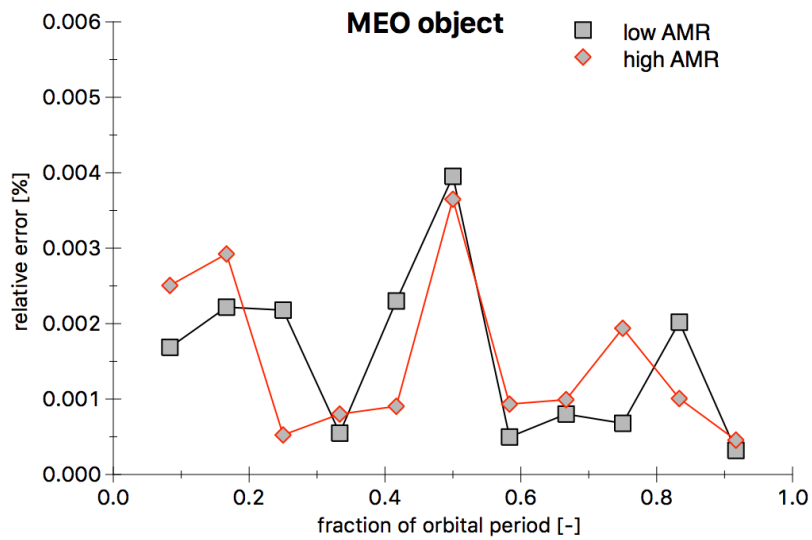


Figure 3.16: A comparison between a high AMR and a low AMR case in MEO. Two tracklets were used in these orbit determinations, the results shown are the average results over 1000 simulations. The low value of $\sigma_0 = 1''$ is used for the noise.

3.5 Limitations

The OBVIOD method is a robust initial orbit determination method capable of finding good approximate orbits. However, it relies on solving the Lambert problem, where a Keplerian motion of the object is assumed. In reality the object follows a perturbed motion, and as such the dynamics models used in the orbit determination will not exactly match the true motion of the object. This mismatch of models will have an effect on the results of the orbit determination and the values of the minimized loss function 3.5. It can be written as follows:

$$X \sim \mathcal{N}(0, \sigma^2) \rightarrow X^2/\sigma^2 \sim \chi^2 \quad (3.10)$$

Note that the X^2/σ^2 is the Mahalanobis distance where in this case the $X = (\bar{\Theta} - \hat{\Theta})$. The resulting variable will be distributed as follows:

$$\bar{\Theta} \sim \mathcal{N}(\mu_{\bar{\Theta}}, \sigma_{\bar{\Theta}}^2)$$

$$\hat{\Theta} \sim \mathcal{N}(\mu_{\hat{\Theta}}, \sigma_{\hat{\Theta}}^2)$$

$$\bar{\Theta} - \hat{\Theta} \sim \mathcal{N}(\mu_{\bar{\Theta}}, \sigma_{\bar{\Theta}}^2) - \mathcal{N}(\mu_{\hat{\Theta}}, \sigma_{\hat{\Theta}}^2)$$

$$\bar{\Theta} - \hat{\Theta} \sim \mathcal{N}(\mu_{\bar{\Theta}} - \mu_{\hat{\Theta}}, \Sigma_{\bar{\Theta}} + \Sigma_{\hat{\Theta}}) \quad (3.11)$$

When the motion of the object is Keplerian the $\mu_{\bar{\Theta}} - \mu_{\hat{\Theta}} = 0$ since the model is able to perfectly match the object's trajectory. That is why, in that case, the Mahalanobis distance is χ^2 distributed.

The effect of a perturbed motion of the objects is that the $\mu_{\bar{\Theta}} - \mu_{\hat{\Theta}} \neq 0$. This is clearly seen in Figures 3.17 and 3.18 where results are shown for a two tracklet case with an arc length of 242 hours (a little over ten orbital periods).

In the situation shown in Figure 3.18 the Mahalanobis distance will not be χ^2 distributed. It will now be distributed according to the non-central χ^2 distribution. The results in Figure 3.18 were obtained with a constraint on

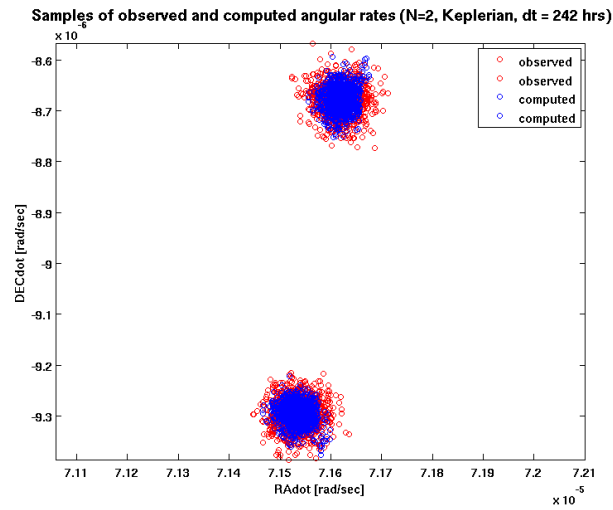


Figure 3.17: The distributions of the observed and computed angular rates for two tracklets and an arc length of 242 hours. Here the object's motion is Keplerian, therefore the computed values will (on average) match the observed ones.

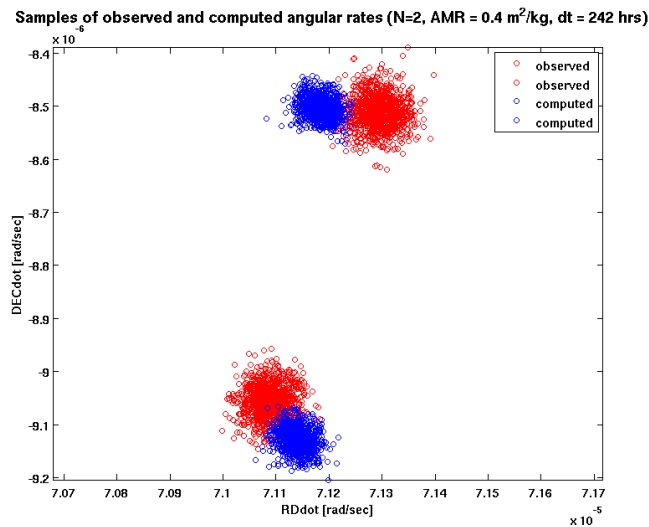


Figure 3.18: The distributions of the observed and computed angular rates for two tracklets and an arc length of 242 hours. Here the object's motion is perturbed, therefore the computed values will not match the observed ones.

the number of revolutions between the two tracklet epochs. The probability density function of the non-central χ^2 distribution is given in Equation 3.12.

$$f_x(x; k, \lambda) = \sum_{i=0}^{\infty} \frac{e^{-\lambda/2} (\lambda/2)^i}{i!} f_{Y_{k+2i}}(x) \quad (3.12)$$

Here x is the random variable, k is the degree of freedom, and λ is the non-centrality parameter which is defined as follows:

$$\lambda = \sum_{i=1}^n \mu_i^2 \quad (3.13)$$

Here n is the total number of random variables considered and $f_{Y_q}(x)$ is the χ^2 distribution with q degrees of freedom, μ_i is the mean value of the i^{th} variable, normalized by its standard deviation. The non-centrality parameter λ can therefore be determined by knowing the mean and standard deviation of the random variables that are involved in the computation. An increase in the λ parameter will have an effect as shown in Figure 3.19.

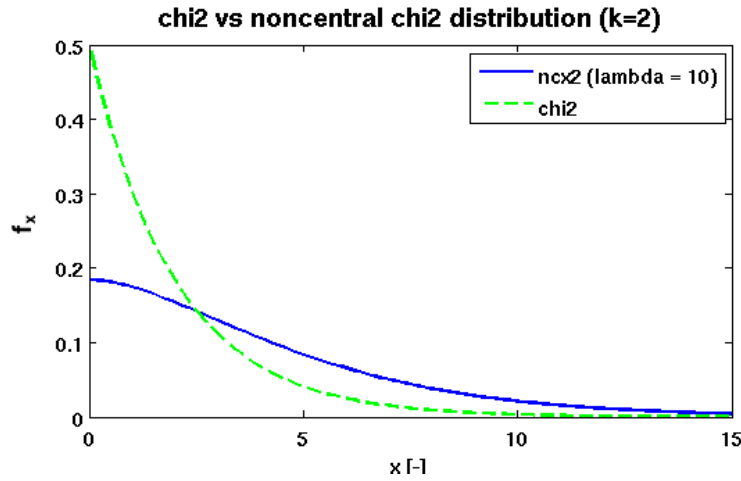


Figure 3.19: A χ^2 distribution compared to a noncentral χ^2 distribution. They both have two degrees of freedom, this corresponds to the two tracklets scenario. A value of $\lambda = 10$ is used.

An example of a non-central χ^2 distributed Mahalanobis distance is given in Figure 3.20. This distribution is a result of two tracklets being spaced at about five days with an AMR of $0.04[m^2/kg]$. Clearly, the λ parameter is

correctly computed and the distribution approximates the samples. The λ parameter is computed by performing one IOD without any noise, this will give the mean values of the observed minus computed vector $\Delta\bar{\Theta}$. These mean values are normalized with the covariance of the attributed values, this is the same covariance that is used in the computation of the Mahalanobis distance.

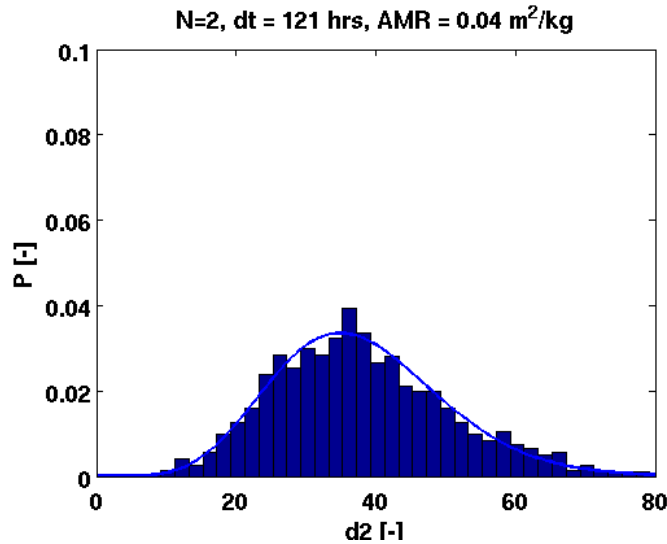


Figure 3.20: A non-central χ^2 distribution depicting the distribution of the Mahalanobis distance in the case where two tracklets are positioned close to a problematic geometry. The non-centrality parameter is $\lambda = 35.9$ in this example. The x-axis label 'd2' refers to the Mahalanobis distance.

The evolution of the λ parameter over time is therefore of interest. The value of this parameter will change with an increase in the length of the orbit arc between tracklets, or with an increase in the perturbations (e.g. higher AMR). In Figure 3.21 the λ parameter can be seen for different times of flight and for two AMR values. The vertical lines in the figure mark the positions of half and full periods of the object that is tracked. From the figure it becomes clear that the method suffers from strong instabilities near the half and full revolution periods. This is explained by the fact that the method uses the Lambert problem to compute orbits. The orbital plane is not important to approximate the geocentric positions at the two epochs, however in order to approximate the angular rates it is important. This is the reason why in

the Figures 3.12 until 3.16 almost no peak at a half period is seen, but in Figure 3.21 there are. This certainly constraints the algorithm's limits, and poses some strong design constraints on the observation scenarios. Note that the distribution shown in Figure 3.20 has a relatively high λ value. This is because the length of the orbit arc is close to the $0.5P + 9P/2 = 121 [hr]$ line, here P denotes the orbital period.

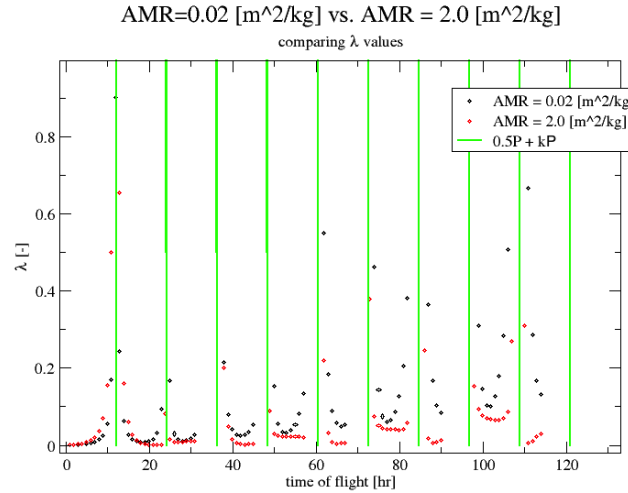


Figure 3.21: The evolution of the λ values as a function of the length of the orbit arc for a high and low AMR object.

Another interesting feature in Figure 3.21 is that the object with a high AMR does not always have a larger λ value. Especially when the length of the orbit arc increases, the λ value is consistently lower than that of the low AMR object. Apparently it is possible to better approximate the attributed values in this high AMR situation. Figure 3.22 shows the semi-major axis and eccentricity of the computed orbit for each of the points in Figure 3.21. This figure clearly shows how the semi-major axis changes as a function of the length of the orbit arc in order to compensate for the higher AMR value. The eccentricity is consistently larger for the high AMR case, and maintains a more or less constant value.

Two tracklets form an ideal geometry when the angle ψ between the first and second geocentric positions r_1 and r_2 , is $\psi = \frac{1}{2}\pi + k\pi$ where $k = 0, 1, 2, \dots$. This angle ensures that the tracklets are as far away as possible from the

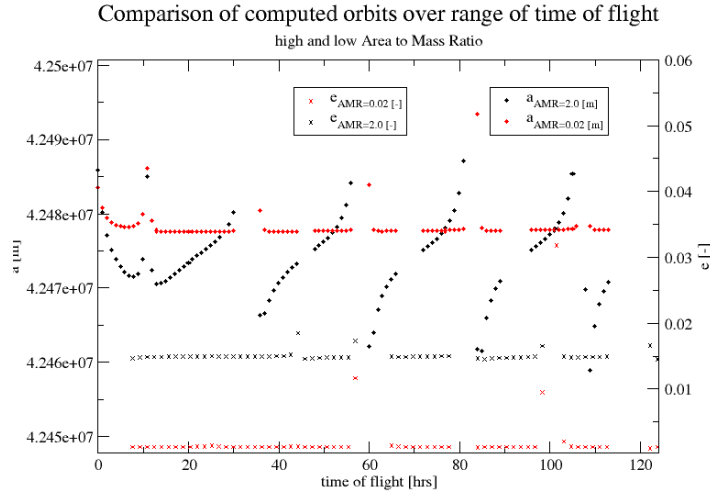


Figure 3.22: The semi-major axis and eccentricity values of the computed orbits in case of a low and a high AMR.

degenerate geometries. Still though, in some cases the computed λ value is not at all in line with the trend. In Figure 3.23 the distribution of the Mahalanobis distance is seen with a length of the orbit arc of $dt = 115 [hr]$ (or 4.75 orbital revolutions) between the two tracklet epochs. This situation should have led to a λ value of about $\lambda = 0.15$. It seems that the Lambert solver that is currently used ((Izzo, 2015)) has issues when trying to resolve any multiple revolution problem where $\psi = \frac{3}{2}\pi + 2\pi k$. In Figure 3.24 the λ parameter is shown as a function of the length of the orbit arc between two tracklets. The two tracklets are always placed in the ideal geometry. Each point where the tracklets are placed at $\psi = \frac{3}{2}\pi + 2\pi k$ is missing from the plot due to the aforementioned issue.

Again it is seen that a higher AMR value does not necessarily mean that the λ value is higher as well. The orbit is still able to absorb the effect that the increased AMR has on the observations. Concretely this means that this method cannot discern a low AMR from a high AMR object. On the positive side this means that it is able to associate tracklets that belong to high AMR objects. On the downside it means that it might wrongly correlate two closely spaced objects by absorbing any differences in their observations with the orbital elements.

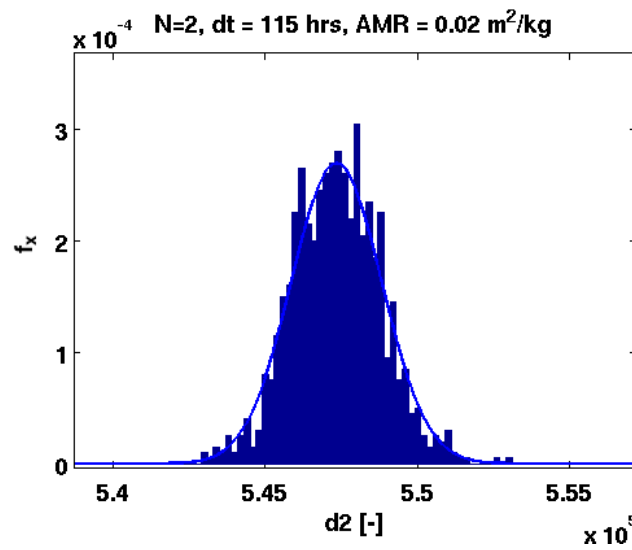


Figure 3.23: The probability distribution function of the Mahalanobis distance for two tracklets with a length of the orbit arc of 115 [hr]. This length of the orbit arc is close to the ideal geometry of the Lambert problem. Apparently the Lambert solver that is currently used has problems when dealing with this geometry in a multiple revolution case. The x-axis label 'd2' refers to the Mahalanobis distance.

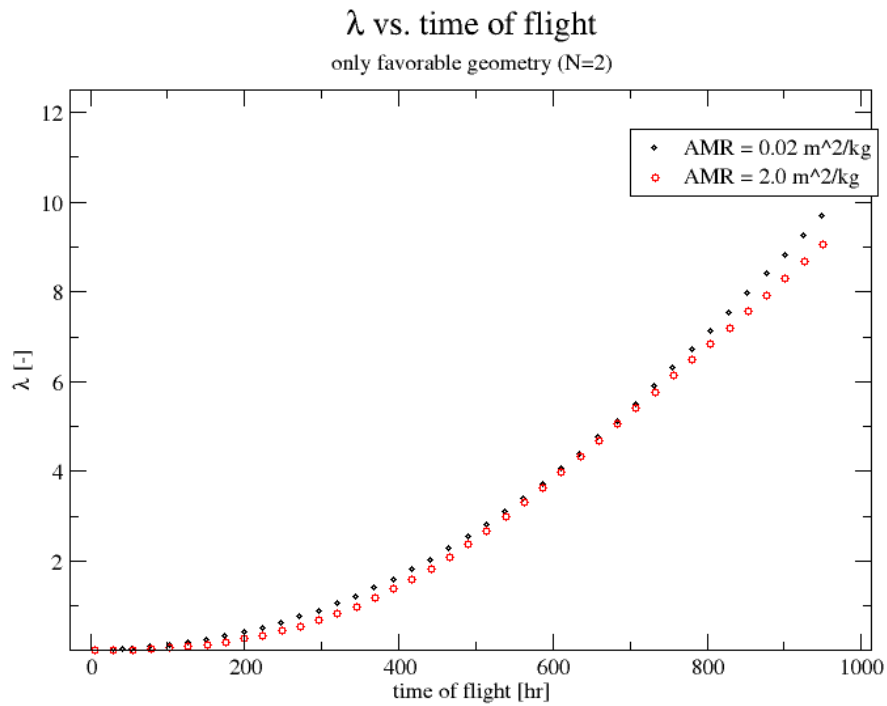


Figure 3.24: λ versus length of the orbit arc between two tracklets. An increase in AMR does not necessarily mean an increase in λ . Note that 1000 hours is about 41 days.

3.6 Multiple revolution solutions

In Figures 3.17 and 3.18 the number of revolutions between the two epochs was constrained. When we remove that constraint, it becomes clear that the solutions can be ambiguous. In Figure 3.25 the results are shown for different regimes of orbital revolutions.

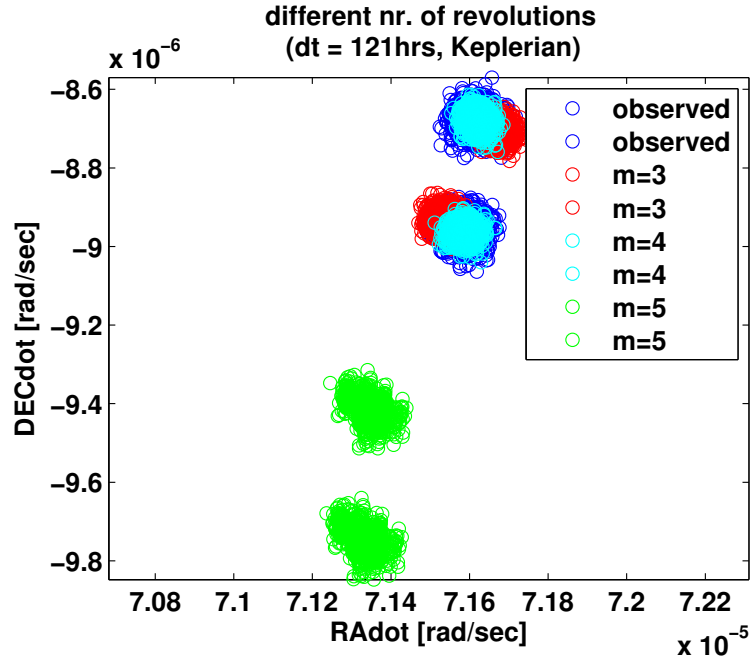


Figure 3.25: Two tracklets that are 121 hours apart. Due to the noise, different solutions exist.

In Figure 3.25 the m denotes the minimum number of revolutions (e.g. $m = 3$ means that the object will have performed between three and four revolutions between the epochs). Clearly, different computed orbits are capable of approximating the attributed values closely. Note that the attributed values in Figure 3.25 belong to an object that follows a Keplerian motion. Without a constraint on the number of revolutions the orbit determination method might favor solutions with the wrong number of orbital revolutions. Therefore, if the number of revolutions between the epochs is not constrained to one setting, the computed values $\hat{\Theta}$ will not be normally distributed. It will

be a combination from different Normal distributions, each of these distributions corresponding to a solution with a different number of orbital revolutions. Without these values being normally distributed, it is not possible to estimate the non-central χ^2 distribution and it could lead to difficulties associating the tracklets correctly. In Figure 3.26 the semi-major axis versus the eccentricity of the different computed orbits can be seen for the $m = 3$ and $m = 4$ cases.

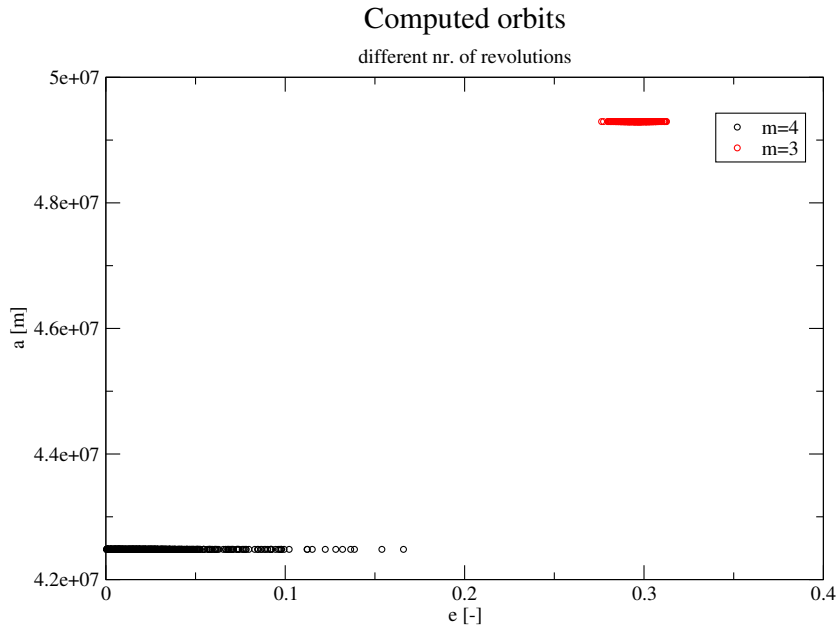


Figure 3.26: The semi-major axis versus the eccentricity of computed orbits for different numbers of orbital revolutions. Both these orbital regimes are able to closely approximate the attributed angular rates of two tracklets that are spaced at 121 hours from each other.

This problem is addressed by setting narrow constraints on the BVAR, particularly on the minimum and maximum semi-major axis values that are allowed. The more time there is between two tracklets, the more narrow the constraints have to be set, until ultimately the constraint is exactly the true semi-major axis. In other words, if tracklets have to be associated over longer time spans then more a priori information on the semi-major axis is needed.

Chapter 4

Application to space debris cataloging

Although PBMH algorithms are meta-heuristic they still have two elements that need to be tailored to the problem on which they are applied. These elements are the fitness function and the solution representation (also known as an 'individual'). This chapter will focus on these elements as well as on a method to reduce the search space of possible permutations in the MTT problem.

4.1 Solution representation

A PBMH algorithm works with a population of individuals. Each individual represents a potential solution and is evaluated to determine its so-called fitness. The fitness of an individual is a measure of the quality of that solution. The individual has to be able to accurately represent any valid solution in the search space. In this work an individual represents the associations between the tracklets. Equation 4.1 shows the general notation of a k-matrix. The k-matrix notation was introduced in the work of (Schneider, 2012), where the most likely k-matrix was sought through the application of Markov Chain

Monte Carlo (MCMC) computations.

$$K = \begin{pmatrix} 1 & 0 & \dots & 0 \\ k_{2,1} & k_{2,2} & \dots & 0 \\ \dots & \dots & \ddots & 0 \\ k_{i,1} & k_{i,2} & \dots & k_{i,j} \end{pmatrix} \quad (4.1)$$

In the k-matrix any entry $k_{i,j}$ can only have a value of 1 or 0. If $k_{i,j} = 1$ it signifies that the tracklet in row i is associated to the object in column j . The k-matrix is defined in such a way that the first tracklet is always associated to the first object. Following this logic the k-matrix becomes a lower triangular matrix.

4.2 Fitness function

The fitness function dictates to which solution the algorithm converges and how it converges to that solution. Therefore the optimum fitness value must coincide with the optimum solution, and any improvement in the fitness value must represent an improvement in the solution. These are the only two requirements for a fitness function, so one can choose to be creative with the definition of the function. The fitness function evaluation is generally the bottleneck of the algorithm in terms of computation speed. There are ways to deal with this directly. For instance by using a function that approximates the true fitness function but is fast to evaluate (Jin, 2005). In this work the fitness function is based on the underlying physics of the problem. It incorporates available information on the MTT problem at hand and uses it to quantify a probability that a certain solution could be true. What follows is a derivation of the fitness function used.

In the fitness function the detection probability P_d and the false alarm probability P_f have to be taken into account. The P_d describes the probability to detect the object. This probability depends on, e.g., the image processing software and the brightness of the object. The P_f describes the probability of having a sporadic measurement due to, e.g., measurement noise. We define the event A as the event where none of the N tracklets are a false alarm. The event B is the event where all of the N tracklets are detected. Event C is the event where the object is not detected in the remaining $S - N$ fences.

Event D occurs when the measurements stem from the object that is represented by the modeled object. The probability of this event is modeled with a multivariate Normal distribution as given in Equation 4.2, since the errors are assumed to be Gaussian. Note that this probability cannot be evaluated in the $N = 1$ case, since the OBVIOD method cannot define an orbit for a single tracklet.

$$\mathcal{N}(\bar{x}, \bar{\mu}_x, \Sigma_{\bar{x}}) = \frac{1}{\sqrt{(2\pi)^n |\Sigma_{\bar{x}}|}} e^{-\frac{1}{2}(\bar{x} - \bar{\mu}_x)^T \Sigma_{\bar{x}}^{-1} (\bar{x} - \bar{\mu}_x)} \quad (4.2)$$

Where \bar{x} is the vector containing the random variables, $\bar{\mu}_x$ is the vector containing the mean values, $\Sigma_{\bar{x}}$ is the covariance matrix and n is the dimension of the vector \bar{x} . The probabilities of having each of these events occur are given in Equations 4.3 to 4.6.

$$P(A) = (1 - P_f)^N \quad (4.3)$$

$$P(B) = P_d^N \quad (4.4)$$

$$P(C) = (1 - P_d)^{S-N} \quad (4.5)$$

$$P(D) = \begin{cases} \mathcal{N}(\bar{\Theta}, \hat{\Theta}(\mathbf{m}^*, \bar{p}^*), \Sigma_{\bar{\Theta}}) & \text{if } N \geq 2 \\ 1 & \text{if } N = 1 \end{cases} \quad (4.6)$$

In case of Equation 4.6 a value of '1' is used for the $N = 1$ case. An empirical parameter will be introduced that allows the $N = 1$ fitness to be tuned.

All the events are assumed to be independent from one another. The conjunction of all the events is given in Equation 4.7. It describes the total probability that a single object exists with the given set of tracklets.

$$P_x = \begin{cases} \mathcal{N}(\bar{\Theta}, \hat{\Theta}(\mathbf{m}^*, \bar{p}^*), \Sigma_{\bar{\Theta}}(\mathbf{m}^*, \bar{p}^*)) (1 - P_d)^{S-N} P_d^N (1 - P_f)^N & \text{if } N \geq 2 \\ (1 - P_d)^{S-1} P_d (1 - P_f) & \text{if } N = 1 \end{cases} \quad (4.7)$$

Here $x = 1, 2, \dots, X$, where X is the total number of hypothetical objects that the k-matrix proposes. The asterisk superscript denotes an optimized value.

To obtain the total probability that a k-matrix is correct, all the probabilities of the individual objects have to be combined as shown in Equation 4.8.

$$P_y = \prod_{x=1}^X P_x \quad (4.8)$$

Where $y = 1, 2, \dots, Y$ denotes the k-matrix, and Y denotes the population size. When we take the negative log-likelihood of Equation 4.7, and introduce the tuning parameter γ , we obtain Equation 4.9.

$$f_x = \begin{cases} L_{N \geq 2}(\mathbf{m}^*, \bar{\mathbf{p}}^*) - \ln \left(\frac{1}{\sqrt{(2\pi)^n |\Sigma_{\Theta}(\mathbf{m}^*, \bar{\mathbf{p}}^*)|}} \right) + & N \geq 2 \\ - (S - N) \ln(1 - P_d) - N \ln(P_d) - N \ln(1 - P_f) & \\ - \ln \left((1 - P_d)^{S-1} P_d (1 - P_f) \right) + \gamma & N = 1 \end{cases} \quad (4.9)$$

And Equation 4.8 becomes Equation 4.10.

$$f_y = \sum_{x=1}^X f_x \quad (4.10)$$

Expressions similar to Equation 4.10 can be found throughout the MTT literature, examples being (Blackman, 2004) and (Deb et al., 1997). In Figure 4.1 the possible fitness values that a group of three tracklets can have are shown. Three tracklets can be arranged as three single tracklets ($N=1/N=1/N=1$), a group of two tracklets and a single tracklet ($N=2/N=1$), and a group of three tracklets $N=3$. Note that the ($N=1/N=1/N=1$) fitness value and the ($N=2/N=1$) fitness values can be shifted by changing the γ parameter. For now a value of $\gamma = 0$ is maintained. This value ensures that the fitness value distributions do not overlap.

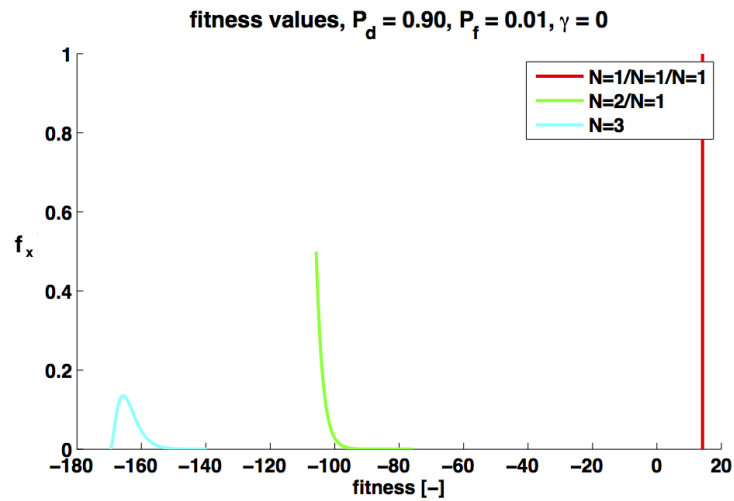


Figure 4.1: Probability density function of the fitness values that a group of three tracklets can have. Three tracklets can be arranged as three single tracklets ($N=1/N=1/N=1$), a group of two tracklets and a single tracklet ($N=2/N=1$) and a group of three tracklets $N=3$.

Chapter 5

Results

This section presents the test cases and their results. The majority of the tests are applied to simulated measurements. This is because in this case the true solution to the problem is known, which makes it easier to judge the performance of the algorithm. Through several test cases the best algorithm is identified. The fitness function and the parameters that can be tuned are discussed in more detail afterwards. The algorithm is then applied to real observations in order to show that also in this case it performs as desired. The second to last section is dedicated to a study of the time complexity of the algorithm. As these algorithms are stochastic the time complexity has to be determined through experimental means. The results suggest that the complexity is polynomial. The final section investigates the effects of different strategies to improve the convergence speed of the algorithm.

5.1 Attributed angular rates uncertainties

The values of the uncertainty of the angular rates ($\dot{\alpha}, \dot{\delta}$) differ depending on what approach is taken. Either the uncertainty is computed a priori, through the expression given by (Fujimoto et al., 2014a). Or it is computed a posteriori through the least squares method. The a priori approach relies on the expression given in Equation 5.1.

$$\sigma_{rate}^2 = \sigma_0^2 \frac{12}{l(l-1)(l+1)dt^2} \quad (5.1)$$

Here l is the length of the tracklet (nr of observations) and dt is the time

between each observation. With the expression in Equation 5.1 the Mahalanobis distance between the attributed and computed attributables are χ^2 distributed. When the a posteriori uncertainty is used it is not perfectly fitting to a χ^2 distribution. Instead it will be distributed according to a non-central χ^2 distribution. The least squares method estimates the variance of a single observation through the following equation:

$$m_0^2 = \frac{v^T P v}{n - u} \quad (5.2)$$

here the v is the vector with the observed minus computed values, the $P = I$ is the identity matrix, n is the number of observations, and u is the number of parameters to be estimated. The so-called model test will result in a value of '1' if the used model accurately fits the observations. For the model test to be close to a value of one also enough observations are needed, and the uncertainty on the observations has to be Gaussian. The model test is given in Equation 5.3.

$$r_{modeltest} = \frac{m_0^2}{\sigma_0^2} \quad (5.3)$$

For different numbers of observations n the plot as shown in Figure 5.1 can be made. From the plot it becomes clear that for a low number of observations the least squares estimator is not able to correctly estimate the uncertainty. However, the m_0^2 does come close to the true value and does this independently from any a priori information on the observation uncertainty. The differences in performance using σ_0^2 or m_0^2 are shown in Figures 5.2 and 5.3. The correctness of a k-matrix indicates the portion of the matrix that coincides with the optimal solution. As can be seen in Figure 5.3 using the posteriori uncertainty does not have a noticeable impact on the performance. Therefore one can conclude that if the observation uncertainty is not known a priori, it will not have a profound impact on the performance of the algorithm.

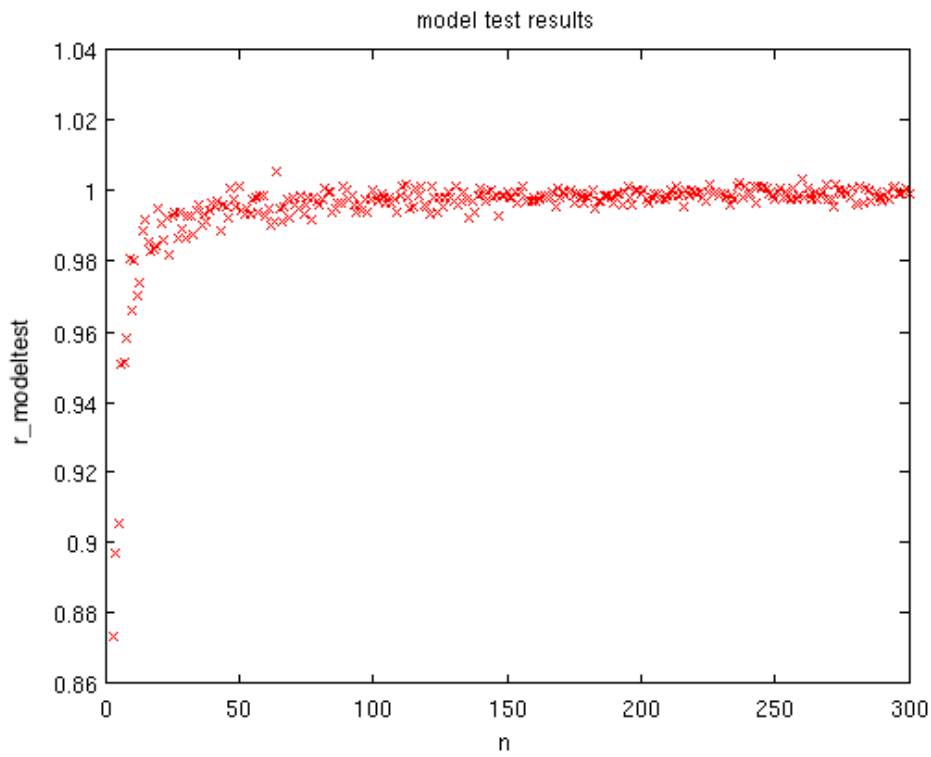


Figure 5.1: results of the model test for increasing number of observations. These are average results over 1000 calculations. It is seen that for a small number of observations the m_0^2 differs from the σ_0^2 .

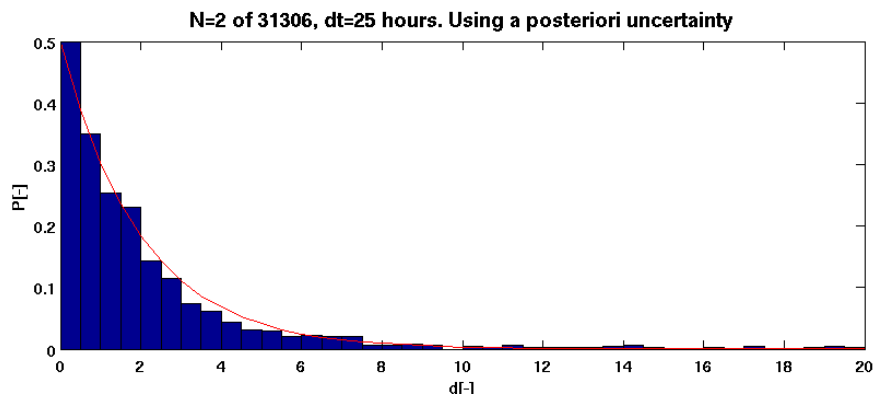


Figure 5.2: Monte Carlo simulation for object 31306. Here the a posteriori uncertainty on the angular rates is used. The values are slightly too small to fit the χ^2 distribution. When the a priori uncertainty is used the values would fit exactly to the χ^2 distribution.

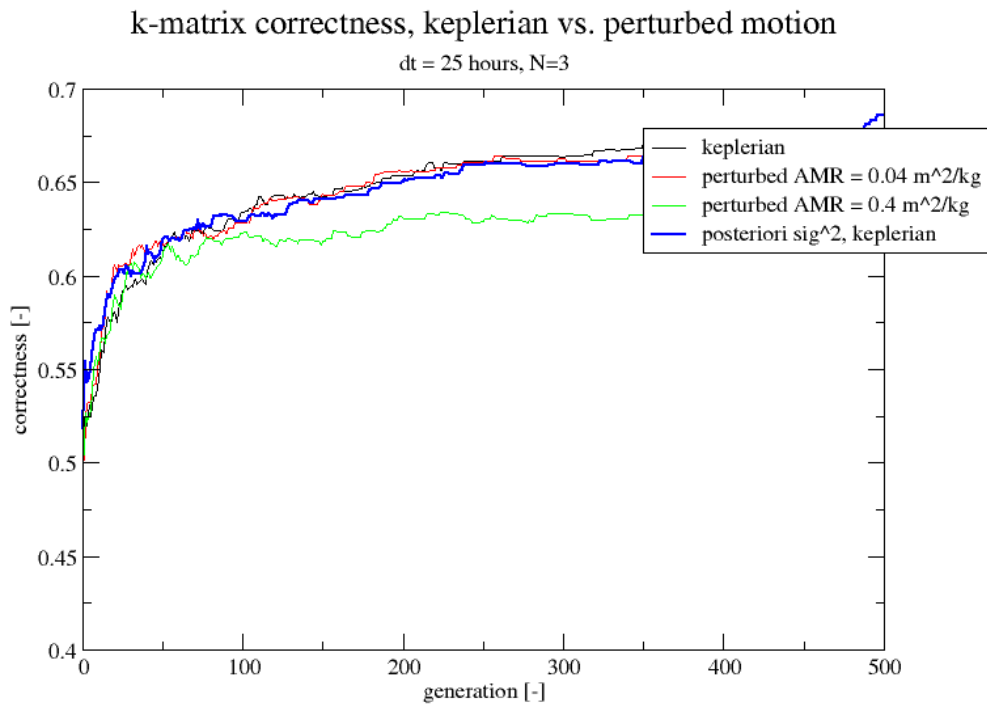


Figure 5.3: The performance of the algorithm with different methods to compute the uncertainties on the angular rates. The performance of the algorithm is not much affected by the method where no a priori information is necessary.

5.2 Fitness function parameter settings

The fitness function 4.9 described in Section 4.2 has a tuning parameter γ included for the $N = 1$ case. Besides that the role of the normalization factor that appears in the $N \geq 2$ case was not well understood, since an improvement in the performance was observed when this term was eliminated from the function. This section will therefore explore the impact that these two parameters have on the performance of the algorithm.

The normalization term in Equation 4.9 stems from the normalization factor that appears in the normal distribution as shown in Equation 4.2. In Equation 4.2 this term serves to ensure that the total volume of the multi-variate Gaussian distribution is equal to 1. The role that these factors play after the negative log-likelihood is taken is not yet clear. In Figure 5.4 the possible values that the fitness function can have are shown for different numbers of tracklets. The Mahalanobis distance is distributed according to a χ^2 distribution, these distributions are shifted by the remaining factors in the fitness function. The main contributors to this shift are the normalization factors.

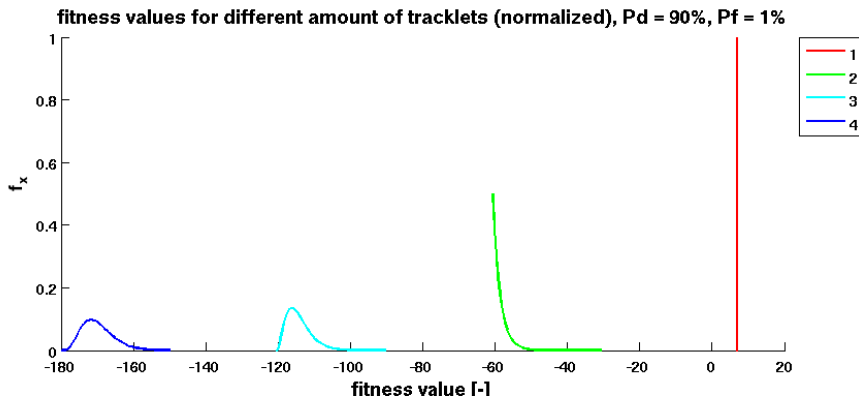


Figure 5.4: Values the fitness can have, depending on the number of tracklets N . Here the normalization factors are taken into account

In Figure 5.4 it is seen that the possible values that the fitness function can have given a certain number of tracklets are clearly separated from each other. This means that the algorithm will favor larger groups of tracklets. The criteria for favoring e.g. a group of $N = 3$ tracklets over a group of $N = 2$ tracklets is that the fitness value of $N = 3$ tracklets should always be better than the fitness values of $N = 2$ and $N = 1$ combined. This is

because the third tracklet of the $N = 3$ group has to be put somewhere else, and this change will have an effect on the total fitness value of the k-matrix. Therefore the distributions of the $N = 3$ fitness and that of the ($N = 2 / N = 1$) fitness have to be non-overlapping. In Figure 5.4 this is the case, thanks to the normalization factors and a $\gamma = 0$. In Figure 5.5 it can be seen what happens when the Normalization terms are taken out of the fitness function.

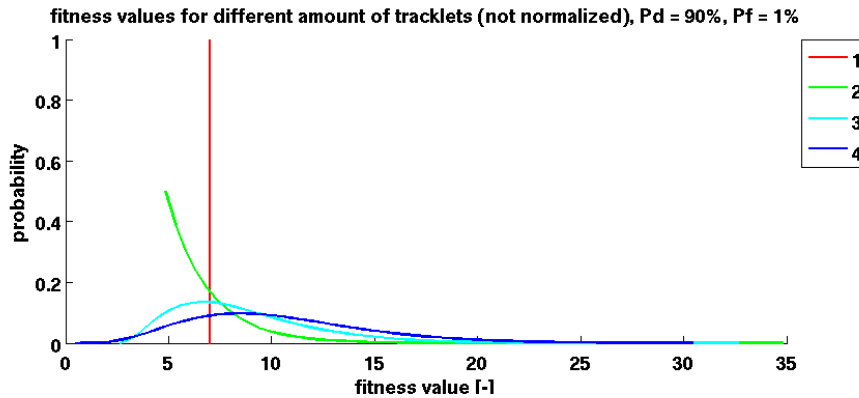


Figure 5.5: Values the fitness function can have, depending on the number of tracklets N . Here the normalization factors are not taken into account. Note that the distributions overlap significantly for this detection probability.

Here the probability distributions overlap. As stated before, the criteria is that e.g. the $N = 3$ distribution should not overlap with the ($N = 2 / N = 1$) distribution. As is shown in Figure 5.6, the probability that for $N = 3$ the fitness value is larger than for ($N = 2 / N = 1$) is small. Therefore it is expected that the algorithm will still perform well in this case, even without the normalization factors.

The fact that the ($N = 2 / N = 1$) distribution almost does not overlap with the $N = 3$ distribution in Figure 5.6 is also due to the probabilities P_d and P_f . If for instance the detection probability is lowered, the distributions will start to overlap more. This is shown in Figure 5.7. This means that there is a distinct probability that the ($N = 2 / N = 1$) situation has a better fitness value than the $N = 3$ case. Also the $N = 1$ value has shifted towards a smaller number. In this case it is certain that the correct correlations will not be found, and smaller groups of tracklets will be favored. This is also intuitive: if the detection probability is lowered, then the probability of having a large group of tracklets becomes smaller.

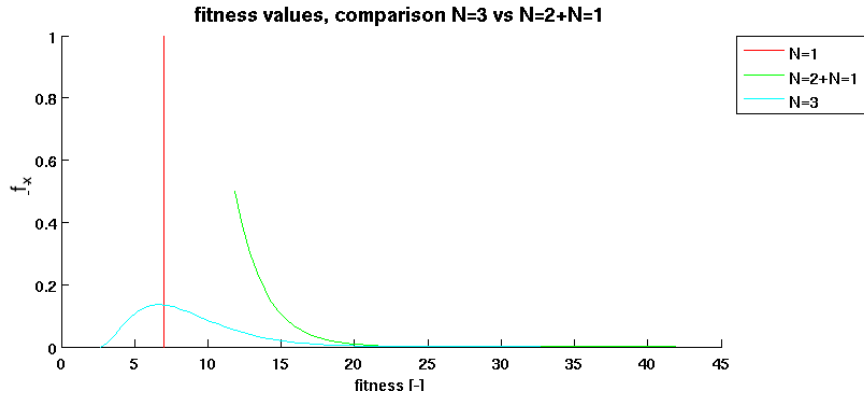


Figure 5.6: Comparing the $N = 3$ and the ($N = 2 / N = 1$) distributions. They overlap slightly in this case. The detection probability is set at $P_d = 0.9$, the false alarm probability at $P_f = 0.01$, and $\gamma = 0$.

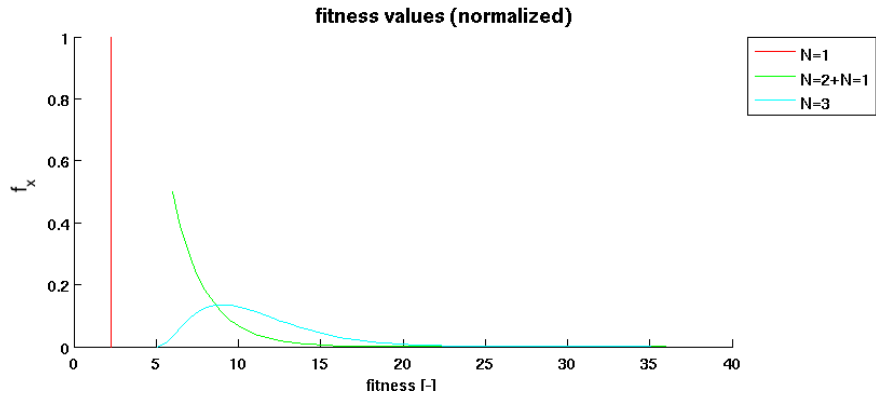


Figure 5.7: Comparing the $N = 3$ and the ($N = 2 / N = 1$) distributions. Here the $P_d = 0.2$, the consequence is that the distributions overlap significantly and that the $N = 1$ value is small. The false alarm probability is $P_f = 0.01$ $\gamma = -50$.

In Figure 5.8 the effect of weighting the normalization factors can be seen. The 'correctness' of the k-matrix is shown versus the generation number. The two most extreme cases, $W_n = 0$ and $W_n = 1$, are represented by Figures 5.5 and 5.4 respectively. The results in Figure 5.8 are as expected. When $W_n = 1$ the fitness value is made up in part by the normalization factors. So the emphasis is more on this part of the function than on the results of the orbit determination. When $W_n < 1$ the emphasis shifts towards the quality

of the orbit determinations. Finally when $W_n = 0$ the normalization factors are not taken into account at all, and the orbit determinations are the main focus of the algorithm.

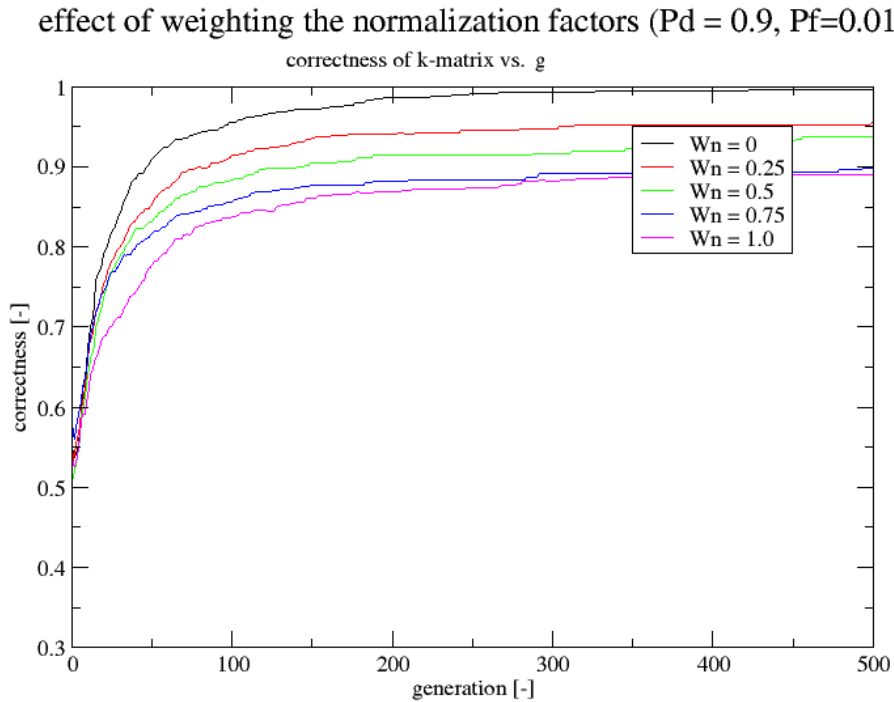


Figure 5.8: Different weights for the normalization constants will result in a different performance of the algorithm. The γ parameter is set to zero.

So far the γ parameter has always been set to zero. Its use is explained by looking at e.g. Figure 5.6. Here the distributions of $N = 3$ and ($N = 2 / N = 1$) are compared. By changing the γ parameter in Equation 4.9 the $N = 1$ (and consequently the ($N = 2 / N = 1$)) distribution can be shifted. This can be used to further separate the distributions, or to cause them to overlap more. In Figure 5.9 the effect of changing the γ parameter can be seen.

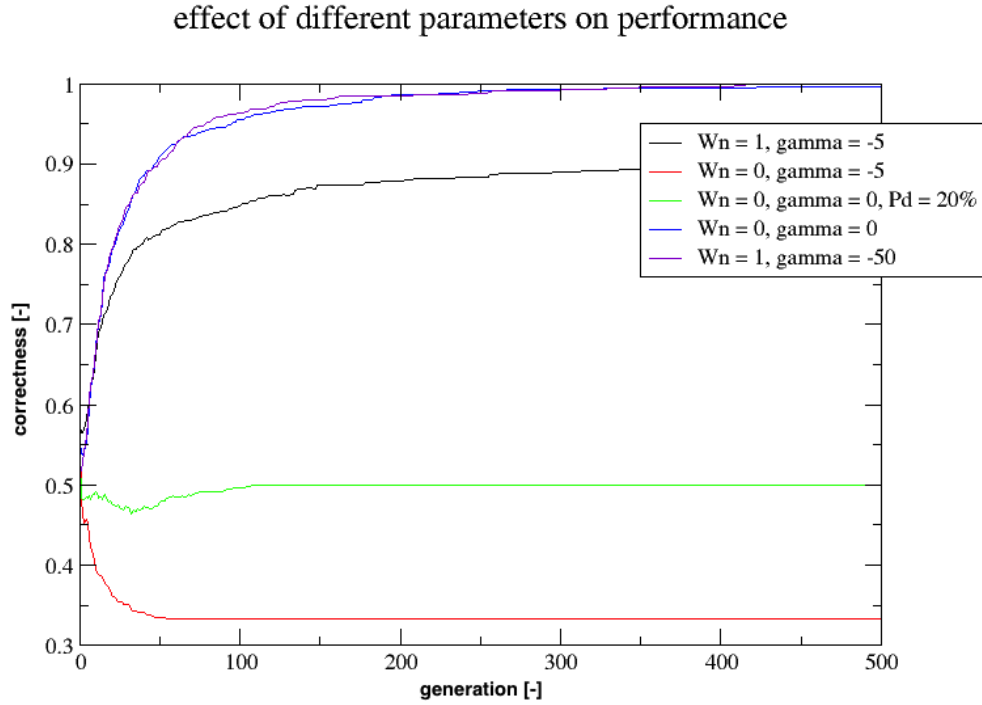


Figure 5.9: Varying the weights for the normalization factors W_n and the value of the γ parameter have a large influence on the algorithm performance.

For reference the $W_n = 0, \gamma = 0$ curve is plotted in Figure 5.9. It is seen that when $\gamma = -5$, with all other parameters being kept the same, the algorithm tends towards the wrong solution. This is because the $N = 1$ fitness value is close to zero in this case. With a fitness value of almost zero, the $N = 1$ scenario is the preferred one in terms of optimizing the total fitness value. Therefore the algorithm will tend towards a solution where all tracklets belong to different objects. Alternatively when $\gamma = 0$ and the

$P_d = 0.2$ (as in Figure 5.7) it is also clear that the algorithm will not tend to a solution that has tracklet groups of $N = 3$. Instead it will find solutions that consist of $(N = 2 / N = 1)$ configurations, since that is what minimizes the fitness function. When the normalization factors are taken into account ($W_n = 1$) the γ parameter has a different impact on the performance. For instance when $W_n = 1$ and $\gamma = -5$ the performance is about the same as when $W_n = 1$ and $\gamma = 0$, since the $N = 1$ value has changed relatively little in this case. The performance of the $W_n = 0$ curve can be obtained by tuning the γ parameter, even when $W_n = 1$. The distributions as shown in Figure 5.10 can be obtained by setting $\gamma = -50$, here the distributions barely overlap. Indeed it is found that the performance is very similar to the case where $W_n = 0$ and $\gamma = 0$.

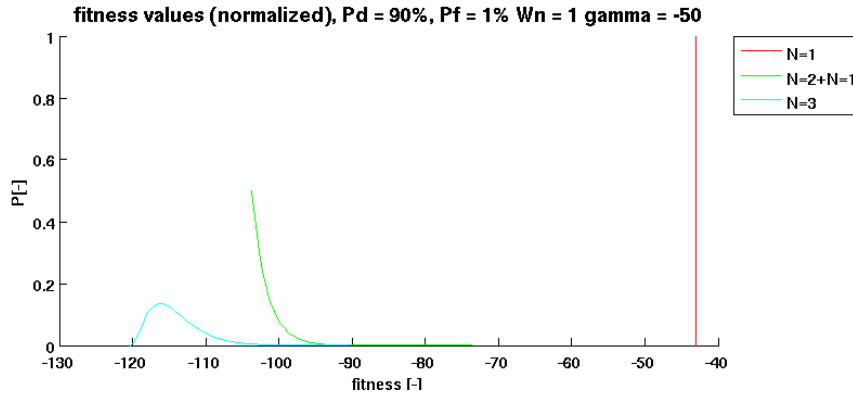


Figure 5.10: Comparing the $N = 3$ and the $(N = 2 / N = 1)$ distributions. Here the normalization factors are taken into account and γ is used to bring the distributions close together.

As a conclusion it can be said that the normalization terms should be taken into account. These terms follow directly from the theory of the problem and there is no strong reason to take them out of the fitness function. Besides that, although the $W_n = 0$, $\gamma = 0$ case performs best, the same performance can be achieved by tuning the γ parameter while having $W_n = 1$. It is therefore decided that the fitness function is left unchanged.

5.2.1 Tuning the γ parameter

The impact of the γ parameter is explained by looking at the probability distributions of the fitness value for different tracklet configurations, as shown in Figure 5.11. By changing the γ parameter, these distributions can be moved with respect to each other.

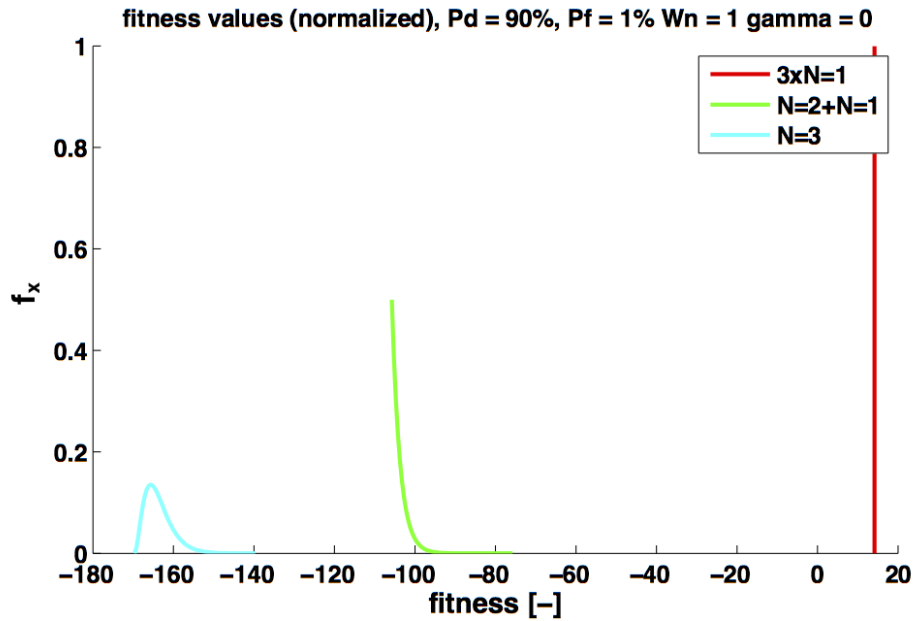


Figure 5.11: Values the fitness function can have, depending on the number of tracklets N . Here the normalization factors are not taken into account ($W_n = 0$). Note that the distributions overlap significantly for this detection probability.

The distributions of $N = 3$, ($N = 2/N = 1$), and ($N = 1/N = 1/N = 1$) are compared to each other. These are the three configurations that any group of three tracklets can take. The performance will increase as these distributions come closer together. When the distributions are moved towards each other the Mahalanobis distance becomes more important. This Mahalanobis distance is a value that is directly related to the quality of the IOD. The performance starts to decrease when the overlap between the distributions becomes significant. If the distributions overlap, a wrong solution might become favored over the correct solution. An extreme example is shown in Figure 5.12. Here the $N = 3$ fitness is almost always worse than the ($N =$

$2/N = 1$) fitness. And the $(N = 1/N = 1/N = 1)$ fitness is always better than either of the other two. With this γ value, the EGA will always favor single tracklets, it will converge to the solution where no tracklets are associated to each other.

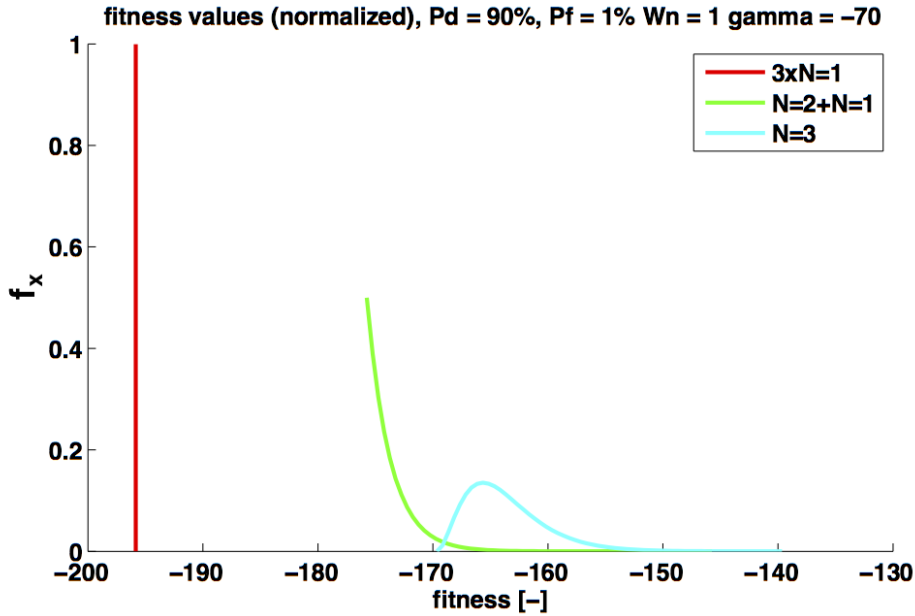


Figure 5.12: Values the fitness function can have, depending on the configuration of three tracklets.

The γ parameter is tuned in such a way that the distributions are close to each other, but without creating any significant overlap. Since it is known that these are χ^2 distributions, it is possible to determine the point at which a certain fitness value is unlikely to occur. In this case it is said that there can only be a $1 \cdot 10^{-3}$ probability that the fitness value of one tracklet configuration is in the range of possible fitness values of another configuration. In other words, this means that in one of a 1000 cases, there is a possibility that the wrong solution is favored by the algorithm.

For a χ^2 distribution with degrees of freedom $d = 2$ (as is the case for the $N = 2$ situation), the threshold value that includes $1 - p$ of the distribution is $t_{1-p}^2 = 13.82$ where t_x^f is the cumulative distribution function (cdf) of the χ^2 distribution with degree of freedom f . For the $N = 3$ situation this values is

$t_{1-p}^6 = 22.46$ (degree of freedom $f = 6$). The minimum fitness values $f_{N=2}^{min}$ and $f_{N=3}^{min}$ are as shown in Equation 4.9, with the Mahalanobis distance set to zero $L_{N \geq 2}(m^*, \bar{p}^*) = 0$. With these values defined, we can set the criteria for the γ parameter as shown in Equation 5.4.

$$3f_{N=1}(\gamma) > f_{N=2}^{min} + t_{1-p}^2 + f_{N=1}(\gamma) > f_{N=3}^{min} + t_{1-p}^6 + t_{1-p}^2 \quad (5.4)$$

In Equation 5.4 the γ parameter influences the $f_{N=1}$ values. The γ parameter is tuned to a value of $p = 1 \cdot 10^{-3}$, the distributions as shown in Figure 5.13 are found. The resulting value is $\gamma = -40$.

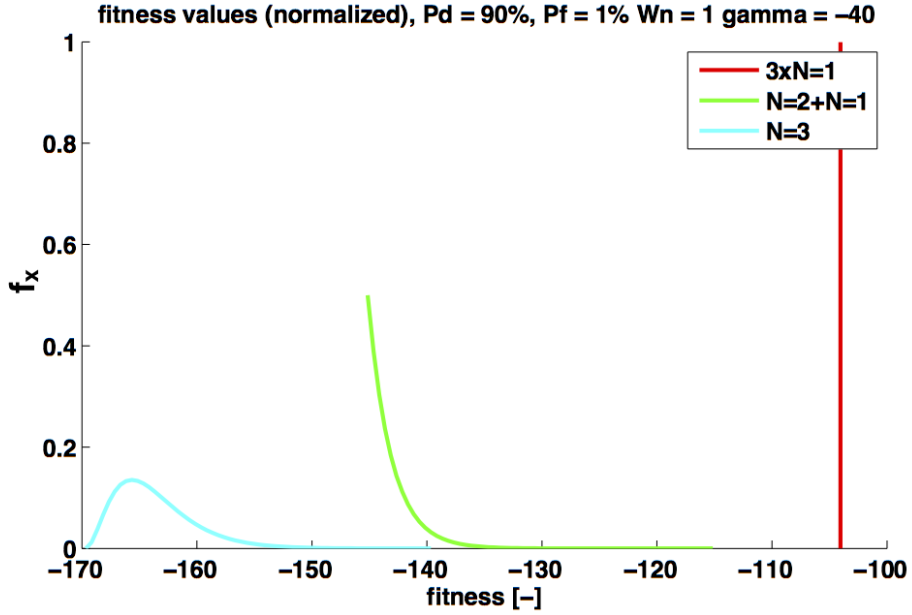


Figure 5.13: Values the fitness function can have, depending on the number of tracklets N . Here the γ parameter is optimized.

In Figure 5.14 a comparison can be seen in the k-matrix correctness over the course of an EGA run, with different γ settings. The correctness of a k-matrix reflects how much of the k-matrix coincides with the optimal k-matrix. A value of '1' means that the k-matrix is the optimal k-matrix. From this figure we can see that the γ parameter has a significant impact on the performance. The optimized γ value (with $p = 1 \cdot 10^{-3}$) indeed performs much better than the default $\gamma = 0$ setting. In this case though, it seems to be a conservative setting, as more performance can be won by increasing the value even further.

The $\gamma = -50$ setting clearly outperforms the $\gamma = -40$ setting. The $\gamma = -50$ value comes at the cost of an increased risk to make the wrong associations, as the overlap between the distributions is larger than for the $\gamma = -40$ case.

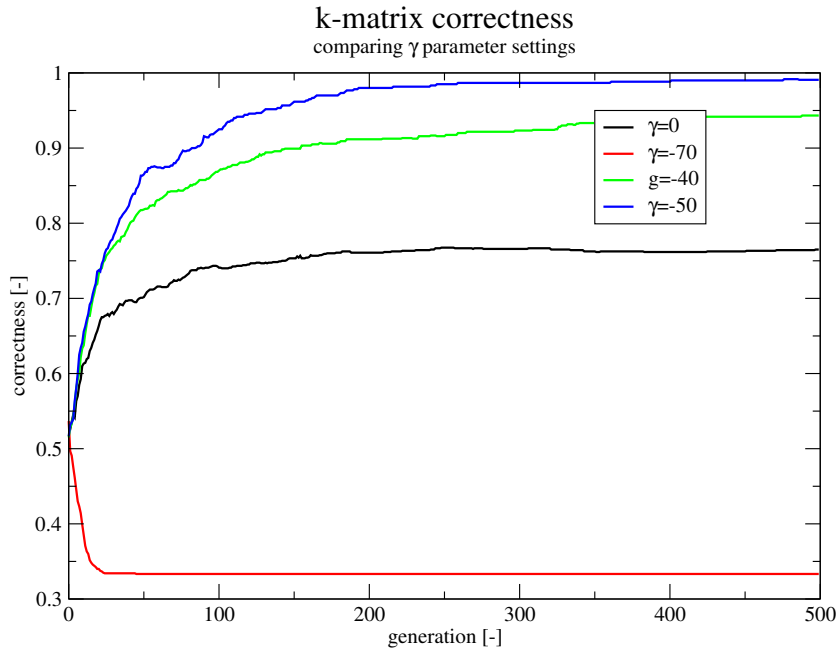


Figure 5.14: A comparison between different γ parameter settings. The $\gamma = -40$ setting is the optimized setting, where the distributions overlap with $p = 1 \cdot 10^{-3}$ at maximum.

5.3 Selection operator of the GA

The choice of selection scheme can have a large impact on the performance of the algorithm. As shown in Figure 5.15 there is a noticeable change in performance when the tournament selection scheme is used.

Apparently this selection scheme is able to maintain a good amount of selection pressure. This is important in order to avoid stagnation of the algorithm when the fitness values in the population become similar to each other. When that happens a selection scheme based on relative fitness will start to select individuals more or less from a uniform distribution. This means that the algorithm becomes reminiscent of a random walk algorithm, something that

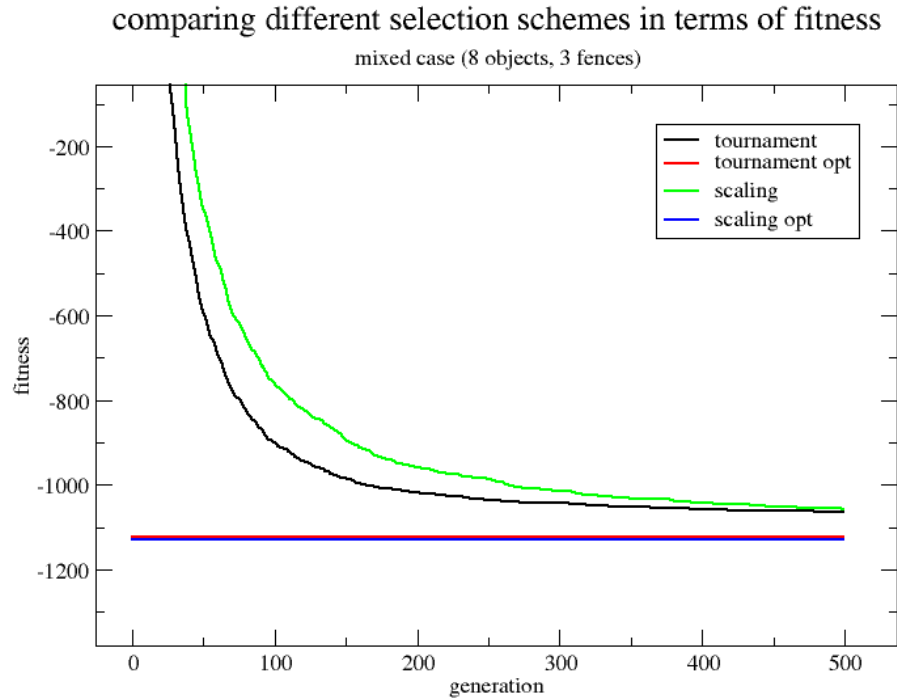


Figure 5.15: The difference in performance when tournament selection is used as opposed to the the previous scheme based on relative fitness. The lines denoted with 'opt' represent the optimal solutions.

is quite ineffective. Since the tournament selection operator does not rely on the relative fitness of the individuals it does not have this problem, and it will therefore not tend towards this random walk behavior. The tournament selection scheme is the preferred selection scheme and is therefore used throughout the tests unless stated otherwise.

5.4 Simulated test cases

In this section three cases are presented. All the objects treated in these test cases are in geosynchronous, near circular orbits. Each case is based

on simulated observations, where the objects stem from the TLE-Catalog¹ and are propagated with a Keplerian motion. At this stage only objects that follow a Keplerian motion are considered. In each case the observations are collected at three epochs within the same night. The fences are spaced at two hours intervals in Right Ascension. Furthermore, the observations are made in the topocentric frame associated with the Zimmerwald observatory (Herzog et al., 2010). No visibility conditions other than the elevation of the object are considered here. A Gaussian noise is added to each individual observation with a standard deviation of $\sigma = 1''$. The settings of the algorithms are kept constant, they can be found in Table 5.1. Note that the stopping criterion of 150 generations is the same for each algorithm. This is deemed to be fair, since all algorithms have a similar computation cost when evaluating and creating a generation. To reduce the computation time a look up table is maintained. In this table the resulting Mahalanobis distance of each IOD is stored, therefore the same orbit is not determined twice. The first test case involves four objects that are clearly separated, this should be a relatively easy task. For the second study case one of the ASTRA clusters is used. These are again four objects but spaced close together (about 0.06 degrees in declination for the given epochs). It is expected that the convergence of the algorithm will be slower when compared to the first case, since the difference between the correct and erroneous solutions are smaller. For the last test case the two previous situations are mixed. This gives a total of eight objects. It is expected that the algorithm will find the correct associations among the four 'easy' targets relatively soon, and will need more time to correctly distinguish the cluster. For each test case the average convergence is presented. As the algorithms are stochastic it is unrealistic to try to derive any bounds on the convergence behavior through analytic means. Therefore the average performance over multiple runs is presented. In this work the algorithms are applied 100 times for each test case. Each time an algorithm is applied it starts from a different random seed. However, the observations remain unchanged after the noise is added one time. This is done to ensure that the optimum solution stays at exactly the same value. This way the purely the performance of the algorithm is observed, and not an additional influence by a varying optimum value. Also the average k-matrix found at the last generation is shown and compared to the true k-matrix.

Four PBMH algorithms are applied to each test case, namely the GA, the

¹<https://www.space-track.org/>

Table 5.1: Settings of the parameters

-	GA	EGA	PBIL	DE
pop. size	2N	2N	2N	2N
p_{mute}	$1/N$	$1/N$	$1/N$	$1/N$
$p_{crossover}$	0.5	0.5	0.5	0.5
% copied to next generation	-	10	-	-
α	-	-	-	$1 \cdot 10^{-3}$
F	-	-	-	0.8
LR	-	-	0.5	-
Mutation shift	-	-	$1/i$	-
max_{gens}	150	150	150	150
γ	0	0	0	0

EGA, PBIL, and the DE algorithms. These tests serve to help determine how well each algorithm performs when applied to this problem, and to identify which algorithm is the most promising.

The measurements are simulated based on the TLEs as published by space-track ², in each example the objects are denoted with their North American Aerospace Defense Command (NORAD) ID number. A TLE of interest is read and converted to Keplerian elements. This orbit is then defined at the TLE epoch and propagated to the desired epochs (spaced at an equal distance of two hours from each other). For each tracklet seven observations are used. These are spaced at 30 s intervals from one another.

5.4.1 Test case 1: four easy objects

In Figure 5.16 the observations for the first test case can be seen. Here it is seen that all the tracklets are spaced at considerable distances (about three degrees in position and $6 \cdot 10^{-4}$ deg/s in angular rates) from each other in all dimensions. Therefore this test case should be an easy one to solve for the algorithm. The numbers noted in the legend of the figure correspond to the NORAD ID numbers of the objects.

In Figure 5.17 the average best solution per generation can be found. The straight line represents the optimum solution.

²www.space-track.com

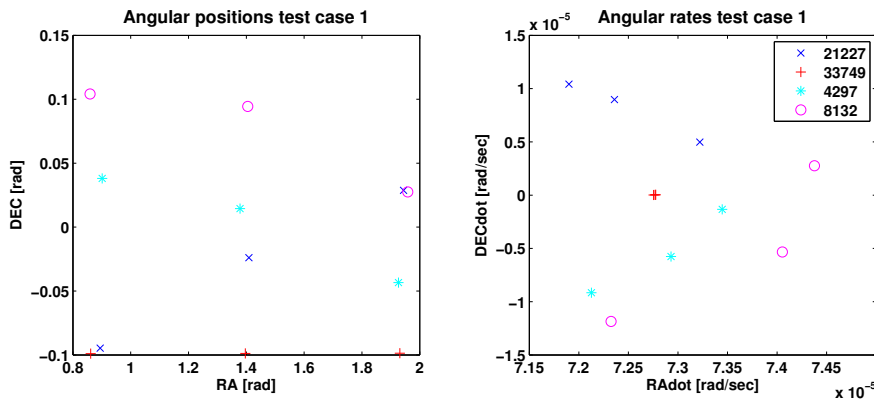


Figure 5.16: The observations used in the first test case. These are Geosynchronous objects on nearly circular orbit, separated in inclination.

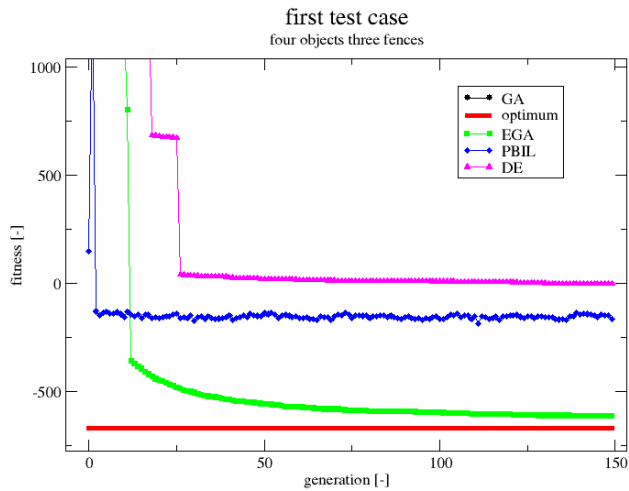


Figure 5.17: The average best fitness individual found per generation. Average value is based on 100 runs.

From these results some clear conclusions can be drawn. First of all, the results of the standard GA are not to be seen in this plot. This is because this algorithm is not capable of finding a solution with a fitness value in the range denoted on the y-axis. Surprisingly the DE algorithm already performs

much better than the GA. This is unexpected since the DE algorithm is an algorithm that is known to have problems when facing a combinatorial problem (Onwubolu, 2009). The PBIL algorithm seems to converge the fastest, but it reaches a plateau early in the run. This signifies that the algorithm has found a local minimum and is trapped. The clear winner among these algorithms is the EGA. Recall that the only difference between the GA and the EGA is that in the EGA the best individuals are always copied to the next population. Therefore the computational cost is the same for both algorithms. Apparently it is of paramount importance that the information contained in the top individuals is not lost. This also explains the relative success of DE with respect to the GA, since in DE the best individuals are always kept as well. These results are also reflected in the average k-matrix that each algorithm found at the end of the run (at generation number 150). The entries of the average k-matrix are given by $\bar{k}_{i,j} = \sum_{r=1}^R k_{i,j}^r / R$, where $r = 1, 2, \dots, R$ indicates the number of the run, $R = 100$ is the total number of runs, and \bar{k} is the average matrix. In Figure 5.18 these k-matrices are compared to each other and to the optimum solution.

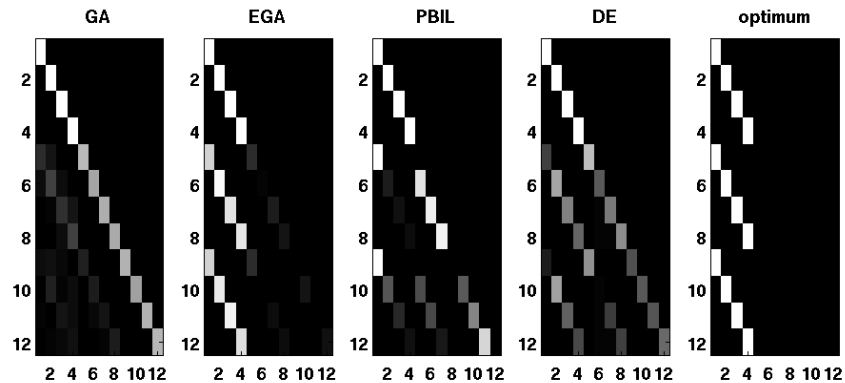


Figure 5.18: The average k-matrix at the end of the run vs. the true k-matrix for the first test case.

The results shown in Figure 5.17 are in accordance with those shown in Figure 5.18. The GA is barely able to distinguish any correct associations, instead it tends to put all the tracklets on the diagonal of the k-matrix (effectively assigning each tracklet to a different object). The PBIL algorithm has indeed found a local minimum which is in part correct. It consistently finds one of the objects, however it is not able to form the complete 3-tuples of tracklets

of the other objects. Finally the DE algorithm consistently finds correct associations but is not able to form the complete groups of tracklets either. The success of the EGA can be accredited to the fact that it takes the interdependencies among the parameters (tracklets) into account.

5.4.2 Test case 2: four objects in a cluster

The second test case concerns a satellite cluster. Satellite clusters form one of the most challenging situations in space debris tracking because of the close proximity that the objects have to each other. Therefore it is an interesting test case for the proposed algorithms. In Figure 5.19 the observations for this test case are found.

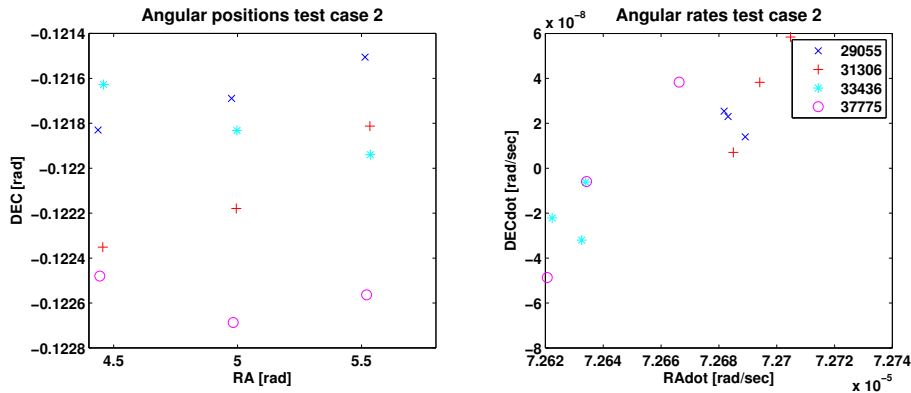


Figure 5.19: The observations used in the second test case. These are closely spaced objects belonging to one of the ASTRA clusters.

In Figure 5.20 the average best solution per generation can be found. The results show a similar behavior as in the first test case. An interesting observation here is that the GA manages to parallel the performance of PBIL. This is because a wrong association in this scenario will not necessarily lead to a bad fitness value, due to the tracklets being closely spaced, and their angular rates being similar to one another. Again the importance of keeping the best solutions is shown. The EGA clearly outperforms the other algorithms. In Figure 5.21 the average k-matrices at the end of the run can be found.

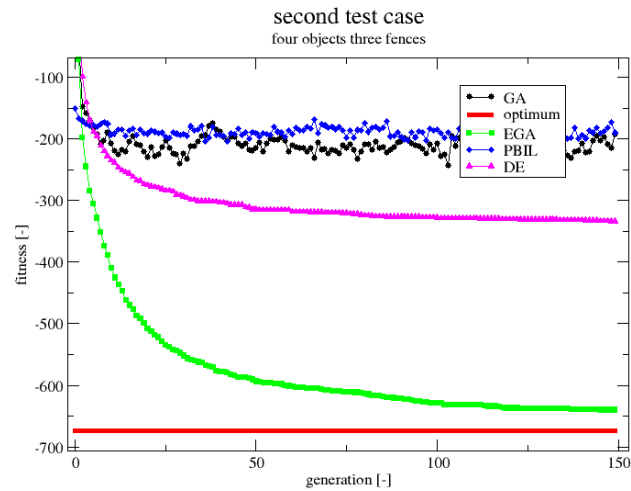


Figure 5.20: The average best fitness individual found per generation. Average value is based on 100 runs.

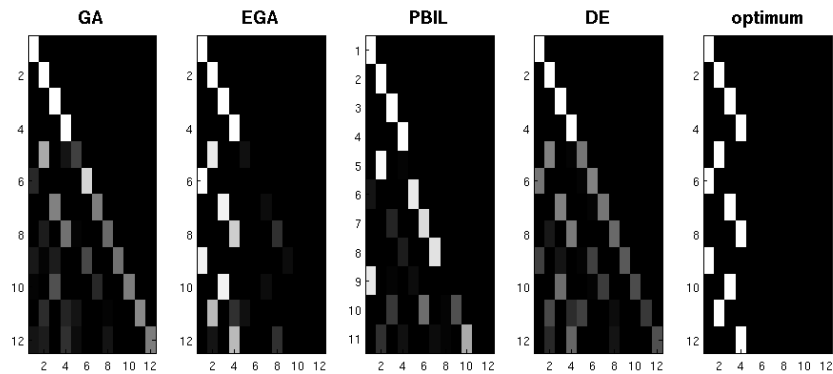


Figure 5.21: The average k-matrix at the end of the run vs. the true k-matrix for the second test case.

In these results it is again evident that the EGA is the only algorithm capable of approximating the correct solution in a reliable way. PBIL again gets stuck in a local minimum, and DE finds good associations but is unable to form the complete groups of three tracklets.

5.4.3 Test case 3: mixed case with eight objects

In the third test case the two previous cases are mixed. The expectation is that the algorithms are able to quickly find the correct associations for the easy targets and will focus on the cluster afterwards. In Figure 5.22 the observations used in this test case are seen.

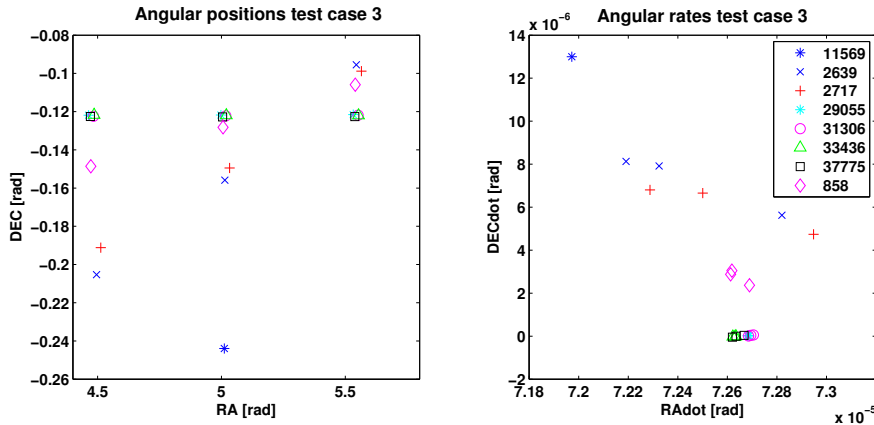


Figure 5.22: The observations used in the third test case. This test case contains both, objects that are separated in inclination, as well as an ASTRA cluster and a single tracklet.

In Figure 5.23 the average best fitness value per generation can be seen. These results are again in accordance with the results of the previous test cases. The EGA is clearly the only algorithm capable of finding a reasonable solution

Figure 5.24 shows the average k-matrix found at generation 150. The first four columns of this matrix correspond to the four objects that are in the cluster. It is clear that the EGA would need more time to distinguish these objects with certainty. Columns five until seven correspond to the objects 2639, 2717 and 858, which are the relatively easy targets. Although the EGA is still unsure about the objects in the cluster, it has associated these three easy targets correctly. This shows that the EGA is able to identify the objects in the more sparsely populated areas with relative ease, after which it focuses on the high density regions such as satellite clusters. Also, it has identified the stand-alone tracklet without any issues.

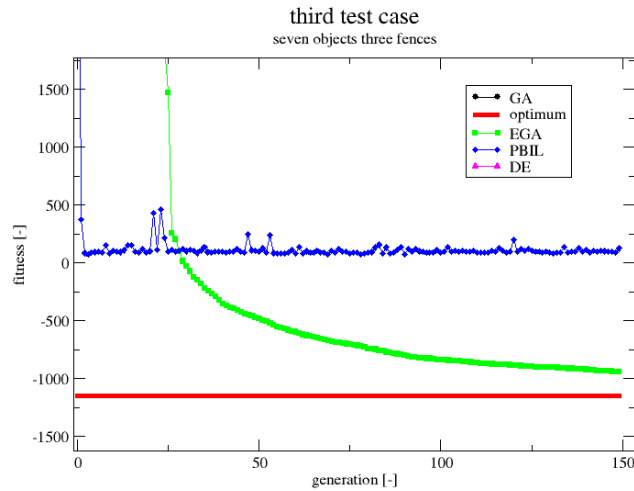


Figure 5.23: The average best fitness individual found per generation. Average value is based on 100 runs.

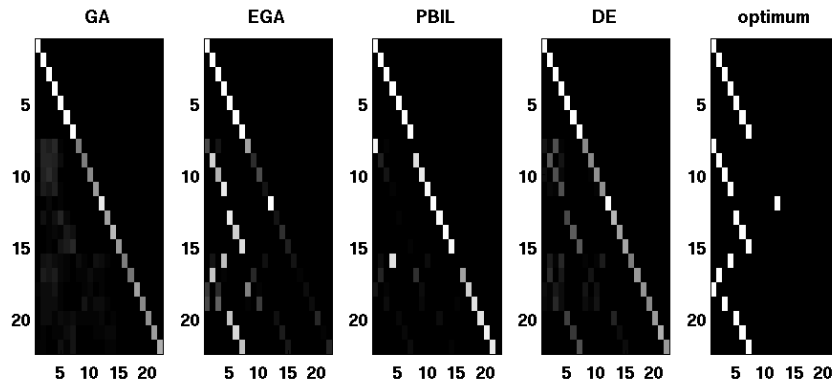


Figure 5.24: The average k-matrix at the end of the run vs. the true k-matrix for the third test case.

5.5 Real observations

The algorithm is applied to a set of observations that were taken of the ASTRA 19.2E cluster. This dataset is chosen because it is very similar to the simulated datasets used in Section 5.4. If the results differ from those

Table 5.2: EGA parameter settings

Pop. size	p_{mute}	$p_{crossover}$	% copied	γ	Max. gens
2N	1/N	0.5	10	-39	500

found in the previous simulated test scenarios, it will be purely because of the difference between simulated and real situations. Table 5.2 displays the parameter settings that were used by the algorithm.

All observations stem from the night of 17-18 March 2016 and were made by the ZimSMART telescope. The attributed values are shown in Figure 5.25. Figure 5.26 shows the convergence behavior of the algorithm. The best fitness value per generation is shown for each individual run as well as the average best fitness value per generation. The plot can be interpreted by considering the average best fitness per generation. The lines that are formed by the individual runs (in gray) give an impression of the uncertainty during the run. This uncertainty is significant. In some cases the optimum solution is found after just a few generations, and in some cases the optimum solution is not found even after 500 generations.

The average k-matrix over 100 runs is shown in Figure 5.27. The result in Figure 5.27 shows that the EGA works as desired. Even though the uncertainty in the best fitness value per generation might suggest a high uncertainty in the final solution, the average k-matrix in Figure 5.27 shows that this is not the case. It consistently finds four objects, which corresponds to the number of objects in the ASTRA 19.2E cluster³. The orbit and the RMS of each of the objects is given in Table 5.3. The RMS is calculated with respect to the individual observations, it is defined as follows:

$$\text{RMS} = \sqrt{\frac{\bar{v}^T \bar{v}}{n}} \quad (5.5)$$

where \bar{v} is a vector containing the observed minus computed values, and n is the number of observations.

Table 5.3 shows that each object is in a near geostationary orbit, which is as expected. One purpose of this initial orbit is to provide a good starting point for an orbit improvement. Therefore the difference between the OBVIOD solution and a least squares solution should be small (i.e. on the order of the observation noise) to ensure convergence of the least squares method. Fig-

³ <http://www.ses.com/4232744/19-2-east> (Accessed 10.08.2016).

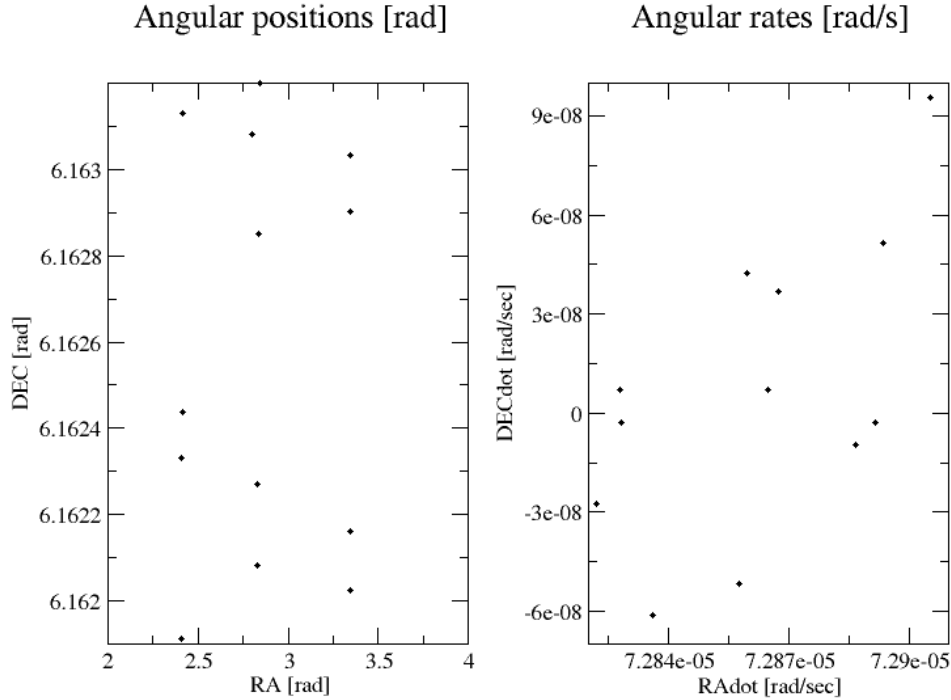


Figure 5.25: Attributed values of the observed objects in the ASTRA cluster.

ure 5.28 shows the difference in right ascension and declination between the OBVIOD and SATORB (Beutler, 2004) solutions. SATORB is an implementation of a least squares estimator used for routine orbit improvements at the Astronomical Institute of the University of Bern (AIUB). The least squares estimator employs a purely Keplerian model of the satellite dynamics. This is done such that the models used by OBVIOD and SATORB are the same. The uncertainty of a single observation is $\sigma_{ob} = 1''$, which translates to about $5 \cdot 10^{-6}$ rad. The difference between the two solutions is smaller than this uncertainty, therefore the initial orbit provided by the OBVIOD method is of sufficient quality.

In Figures 5.29 and 5.30 the residuals with respect to the individual observations can be found. They are shown for each of the four objects. An important observation to make is that all the objects have residuals that are consistent to one another. This reinforces the belief that the tracklets are

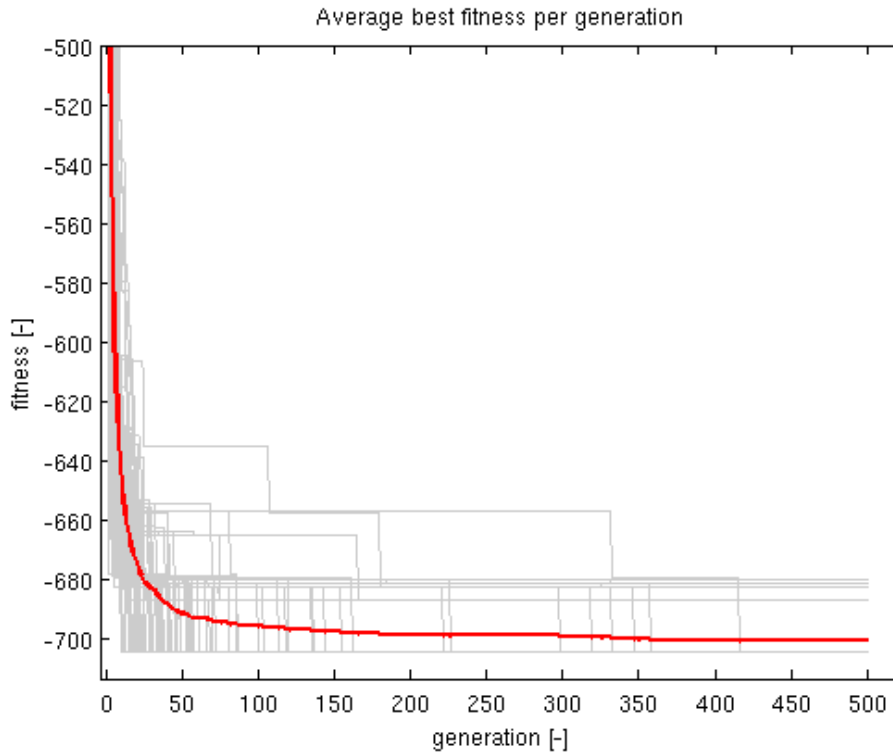


Figure 5.26: Average best fitness value per generation, averaged over 100 runs. The results of the individual runs is shown in gray.

correctly associated to each other.

In Figure 5.30 an increase is seen in the residuals of the declination for the observations 8-14. This can be explained by considering the workings of the OBVIOD method. The OBVIOD method uses the first and last tracklets in the orbit determination to define a Lambert problem. Since it uses this Lambert problem the orbit will always precisely intersect the attributed angular positions of the first and last tracklet. Therefore the residuals with respect to the individual observations in the first and last tracklet are also expected to be small. The tracklet that is in between the first and last tracklets is not involved in the definition of the Lambert problem, and therefore the residuals might be larger for those observations.

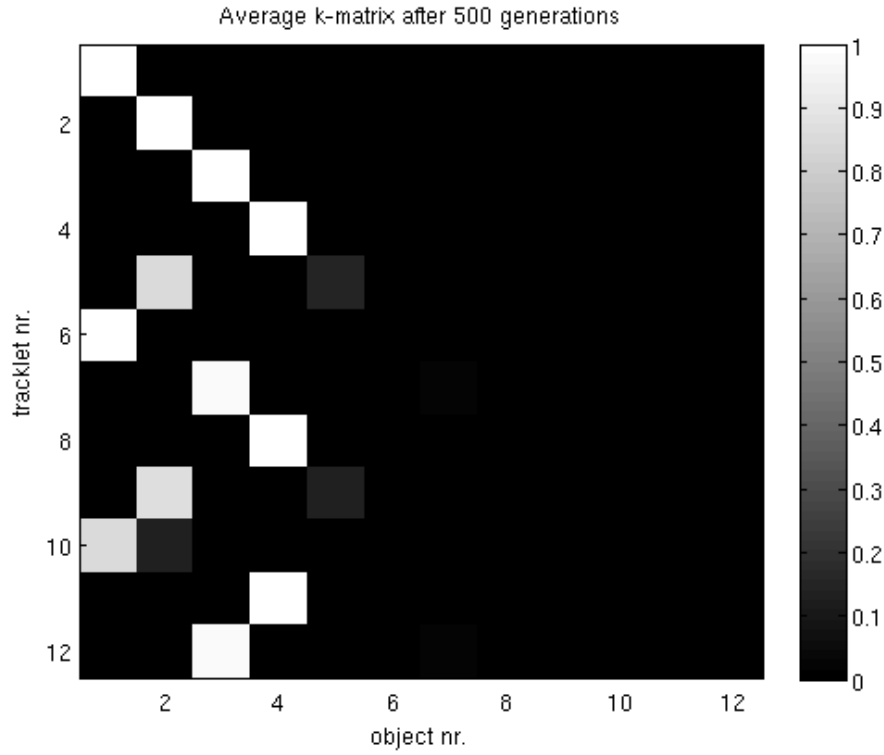


Figure 5.27: Average k-matrix at the end of 500 generations. The EGA consistently converges to the same solution.

Table 5.3: Orbital elements as determined by OBVIOD

Sat. nr.	a [m]	e [-]	i [deg]	Ω [deg]	ω [deg]	M [deg]	RMS [rad]
1	$4.216 \cdot 10^7$	$3.2 \cdot 10^{-4}$	0.04	103.0	204.5	-170.6	$8.86 \cdot 10^{-7}$
2	$4.216 \cdot 10^7$	$5.7 \cdot 10^{-4}$	0.044	56.6	-86.4	166.7	$1.16 \cdot 10^{-6}$
3	$4.217 \cdot 10^7$	$2.0 \cdot 10^{-4}$	0.085	99.6	-48.5	85.8	$1.08 \cdot 10^{-6}$
4	$4.217 \cdot 10^7$	$4.9 \cdot 10^{-4}$	0.091	66.7	-45.9	116.2	$1.14 \cdot 10^{-6}$

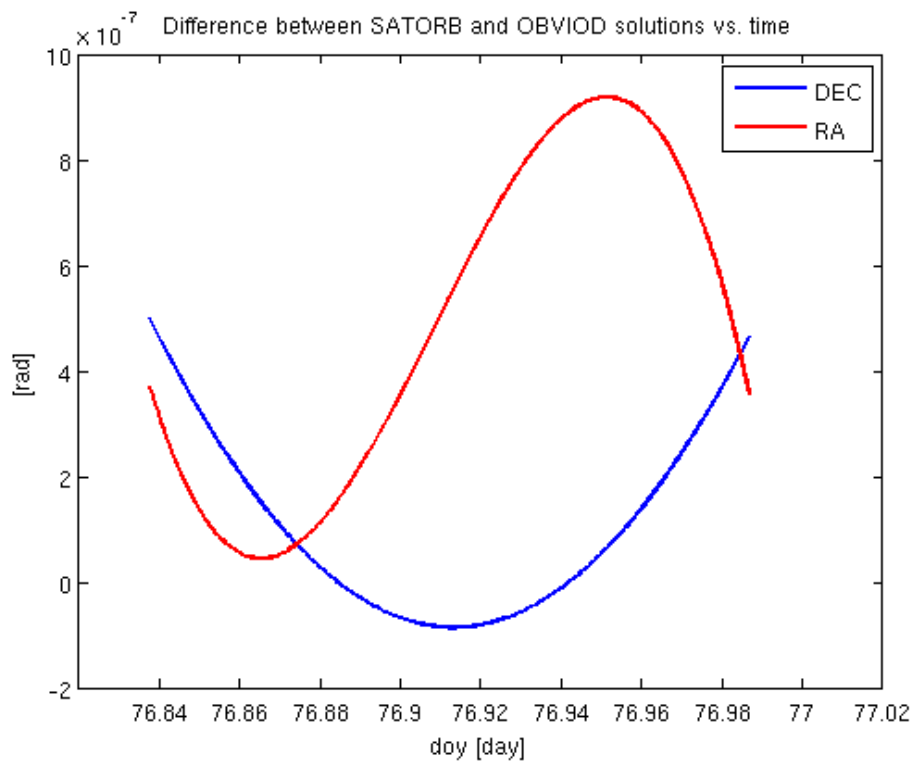


Figure 5.28: The difference in right ascension and declination between the least squares solution and the OBVIOD solution. The x-axis label ‘doy’ stands for ‘day of year’.

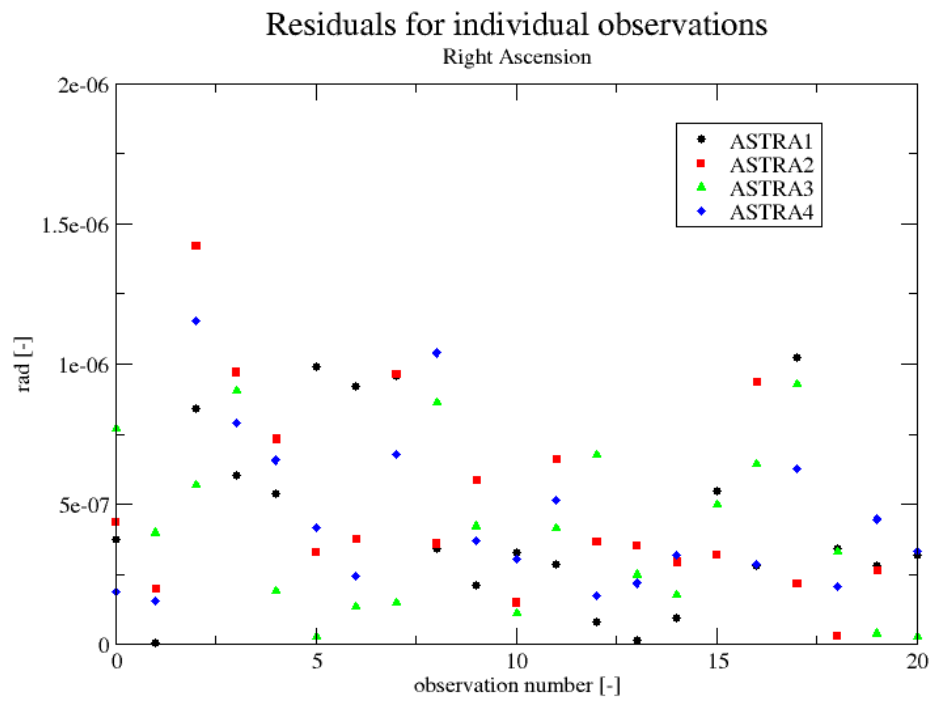


Figure 5.29: Residual in right ascension for each observation and for each object.

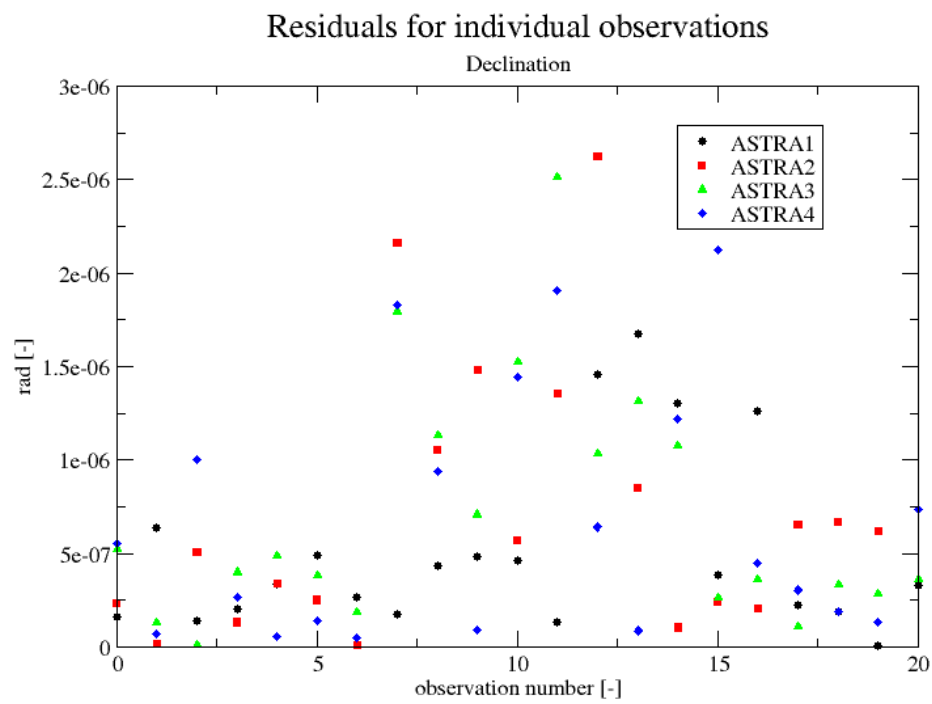


Figure 5.30: Residuals in declination for each observation and for each object.

5.6 Time complexity

As mentioned in the introduction, the challenge is to find a good approximate solution in a reasonable computation time. It is clear that the EGA is capable of finding a good approximate solution, the question that remains is whether it can do so in a realistic (polynomial) computation time.

The EGA is a stochastic method. Therefore it is difficult if not impossible to say anything about its behavior in a deterministic way. In cases where it is possible to deduce a characteristic, its use is questionable. An example is that the upper bound on the convergence behavior for a regular GA is the same as that of an algorithm that generates solutions purely at random (Oliveto et al., 2007). In the work of (Oliveto et al., 2007) an overview is given of the recent investigations in the time complexity of Evolutionary Algorithms (the GA is a type of Evolutionary Algorithm). One of the conclusions is that the function to be optimized has to be considered when analyzing the convergence behavior.

With this in mind the following experiment was devised. The EGA is applied to problems of different sizes. As it is a stochastic algorithm, it is applied 100 times and the average performance is studied. The problem size is said to change when the number of tracked objects changes. In order to keep the difficulty of the problem the same when increasing the number of objects they are spaced at one degree increments in inclination, but are otherwise in identical orbits. They are geosynchronous objects, a tracklet is observed every hour for a total of three tracklets per object.

Figure 5.31 shows the average k-matrix at different stages of the computation for a problem with five objects. Four matrices are shown, at 80%, 90%, 95% of the optimum value, and the true k-matrix. The computational burden is concentrated at the beginning of the run. A look up table is used, therefore the computation time needed per generation quickly becomes relatively low. The improvement in the solution from 80% to 95% is therefore relatively cheap from a computational standpoint.

In Figure 5.32 the absolute computation time in seconds is plotted against the number of objects that are being tracked. It is the time needed to have a solution at 90% of the optimum fitness value. These tests were performed on a machine with a 3.16 GHz CPU. The tests involved 11 tracked objects at maximum. Because for each number of tracked objects the algorithm has to be run 100 times, it was chosen to not go further than 11 objects (about two hours of computation) in order to limit the computation time. Two

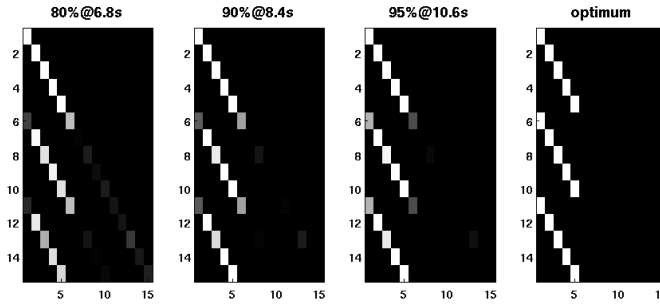


Figure 5.31: The average k-matrix found at different stages of the search. Both the percentage of the optimum fitness as well as the time at which the solution is found are given. The intensity scale is the same as in Figure 5.27.

Table 5.4: Goodness of fit values for both models.

-	polynomial	exponential
Sum of Squares due to Error (SSE)	4.44	12.52
R-square	0.999	0.998
adjusted R-square	0.999	0.997
RMSE	0.70	1.18

lines are fitted to these points, a polynomial and an exponential function. These models were fitted in a least squares sense with the MATLAB curve fitting toolbox. The goodness of fit values are given in Table 5.4, from this table it becomes apparent that the data is well described by a polynomial model. The SSE is the sum of squares of the differences between the model and the actual data points. The R-squared statistic is the square of the correlation between the predicted and actual response values. When the R-squared value is corrected for the number of degrees of freedom the adjusted R-squared value is found. Finally the RMSE is an estimate of the standard deviation of the data, it can be found by taking the square root of the SSE divided by the number of degrees of freedom. The determined parameters for the exponential function are $a = 3.00$ and $b = 0.29$ and for the polynomial they are $a = 0.11$ and $b = 2.69$.

The population size and the number of objects per k-matrix scale linearly with the total number of tracklets ($\text{pop.size} = 2N$, $X \sim N$). This means that the computation time needed to evaluate a generation will scale with N^2 . The parameter b of the polynomial fit is greater than two for several reasons. The

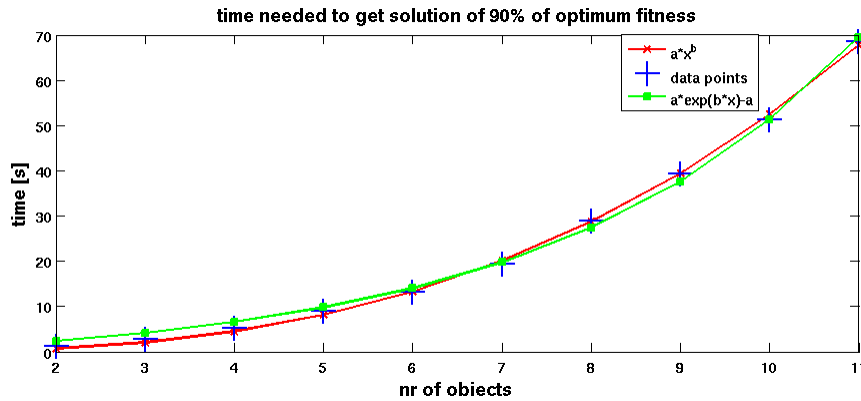


Figure 5.32: Computation time needed to reach an approximate solution of a certain quality (90% of optimum) versus the number of tracked objects. Two lines are fitted, a polynomial and an exponential function. Each point represents the average computation time over 100 runs.

first is that the number of generations needed to arrive at a good approximate solution will increase when more objects are tracked. However, due to the look-up table it is relatively cheap to compute additional generations after the initial few generations. Therefore, this has a minor impact on the total computation time. Also the computation time needed by the crossover and mutation operators is relatively small, but will still make a contribution. Therefore it is to be expected that a polynomial trend fits well to the data points. This polynomial behavior holds true for the range that is shown (two to 11 tracked objects). Although it is expected that the polynomial trend continues outside of this range, there is no way to be certain without adding additional data points.

5.7 Improving the convergence rate

The search space reduction algorithm described in Section 5.7.1 is applied to the third simulated test case. The change in performance is measured in terms of fitness function values versus the generations, and the average k-matrix throughout the algorithm run. It is expected that the search space reduction will reduce the number of wrongly associated tracklets during the run and that it might result in a faster convergence of the EGA. This section first explains the implementation of the search space reduction. Afterwards

the results are presented for the search space reduction scheme and for a ‘smart’ population initialization.

5.7.1 Search space reduction method

Problem specific information has to be used in order to reduce the search space. In this case the determined orbits can be used to prune the search space. The idea is that during the EGA run the algorithm is able to identify the ‘easy’ targets that are in low density environments. These easy targets are found by propagating an orbit to the epochs of stand-alone tracklets. All the tracklets that are far away (a to-be-defined metric) from the propagated state are excluded from possibly being associated to that object. These excluded associations are then removed from the search space. This method will allow the algorithm to resolve the easy targets quicker, and to concentrate on the more challenging situations such as satellite clusters.

The decision on whether or not to propagate an object is based on the minimized loss function value of that object. This value is the Mahalanobis distance, and it is χ^2 distributed with a degree of freedom $f = 4N - 6$ where N is the number of tracklets associated to the object. The Mahalanobis distance d^2 is converted to a probability by taking the probability that a random sample x of the χ^2 distribution will result in a value that is d^2 or larger $P(x \geq d^2)$. This is expressed by the Equation given in 5.6.

$$P(x \geq d^2) = 1 - F(d^2; f) \quad (5.6)$$

Where $F(d^2; f)$ is the cumulative distribution function of the χ^2 distribution, it is given in Equation 5.7.

$$F(d^2; f) = \frac{\gamma(\frac{f}{2}, \frac{d^2}{2})}{\Gamma(\frac{f}{2})} \quad (5.7)$$

In Equation 5.7 the $\gamma(\cdot, \cdot)$ is the lower incomplete gamma function, and the $\Gamma(\cdot, \cdot)$ is the gamma function, which are given in Equations 5.8 and 5.9 respectively.

$$\gamma(\frac{f}{2}, \frac{d^2}{2}) = \int_0^{\frac{d^2}{2}} t^{\frac{f}{2}-1} e^{-t} dt \quad (5.8)$$

$$\Gamma\left(\frac{f}{2}\right) = \left(\frac{f}{2} - 1\right)! \quad (5.9)$$

This is the chosen way in which to convert a Mahalanobis distance value to a probability for this application. It could be that there is a better alternative, but this is not further explored.

In Figures 5.33 and 5.34 the attributed (the values contained in the attributable vector) and propagated angular positions and rates can be seen. Right away it is seen that the uncertainty of the propagated angular positions is large with respect to the uncertainty in the attributed values. In the case of the angular rates the uncertainties are quite similar in magnitude.

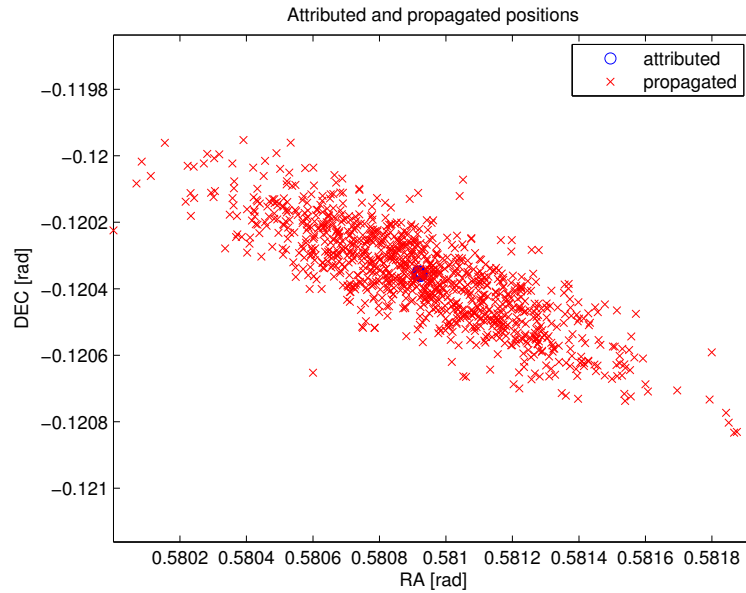


Figure 5.33: The attributed and propagated angular positions at the epoch of the third tracklet for one object. Although the distributions are centered on the same position, the uncertainty of the propagated positions is considerable.

In Figures 5.35 and 5.36 the same results as in the previous two figures are shown, but for all four objects in the cluster.

From these figures it become apparent that the objects are much better separated in position than in angular rate. To be able to keep a probabilistic

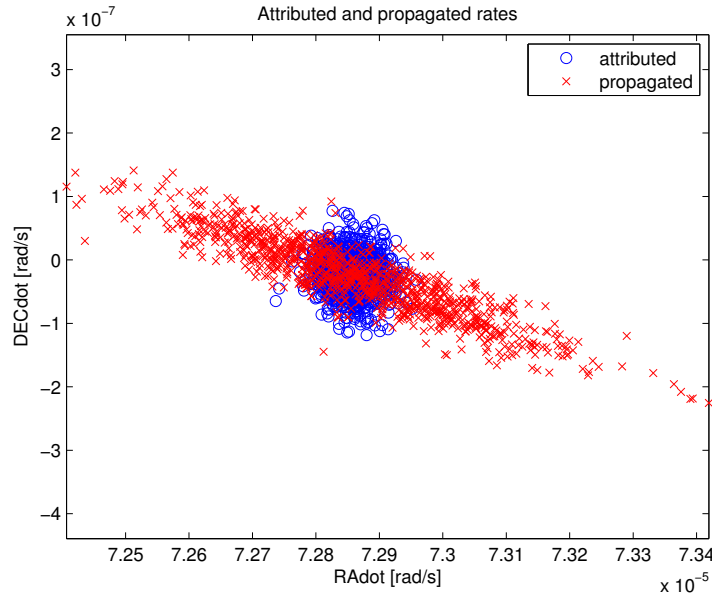


Figure 5.34: The attributed and propagated angular rates for one object. In this parameter space the distributions are centered on the same values and have a similar uncertainty.

approach the covariance of the propagated state is needed. This covariance has to be computed through a finite difference scheme. No analytic method is possible since the OBVIOD algorithm uses the Lambert problem to determine the orbit. The Lambert problem uses iterations to find the correct solution, and therefore it is not possible to analytically derive the partial derivatives needed. The uncertainties on the individual observations are propagated to uncertainties on the propagated state. This gives the result as shown in Figure 5.37 and 5.38.

Note that in this approximation a Gaussian uncertainty is assumed. In this case the finite difference scheme is able to describe the uncertainty well with a Gaussian distribution. This might not be the case for more extreme (longer time of flight, high AMR) examples. But for our current needs (reducing the search space) it is sufficient. The Mahalanobis distance between the attributed and propagated state can now be computed. If the Mahalanobis distance is larger than a set value the association is removed from the search space. Figure 5.39 shows this χ^2 distribution.

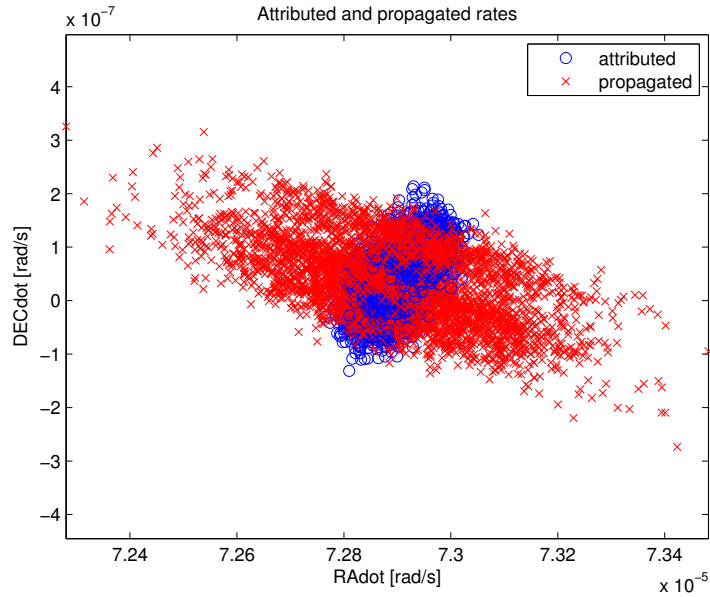


Figure 5.35: The attributed and propagated angular positions at the epoch of the third tracklet for all four objects in the cluster. Although the distributions are centered on the same position, the uncertainty of the propagated positions is considerable.

The threshold value of the Mahalanobis distance at which an association is removed is defined by the probability that a random value sampled from the χ_4^2 distribution is smaller than the set threshold. For instance, for a probability of $P = 0.9999$ that value is 23.5.

These ‘illegal’ associations are stored in a table. With this table each new k-matrix is checked for illegal associations. If an illegal association is found it is repaired by a separate repair operator. This operator removes the wrongly associated tracklet from the association and replaces it with an association that is ‘legal’. The wrongly associated tracklet is put on the diagonal of the k-matrix, where it represents a new object consisting of one tracklet only.

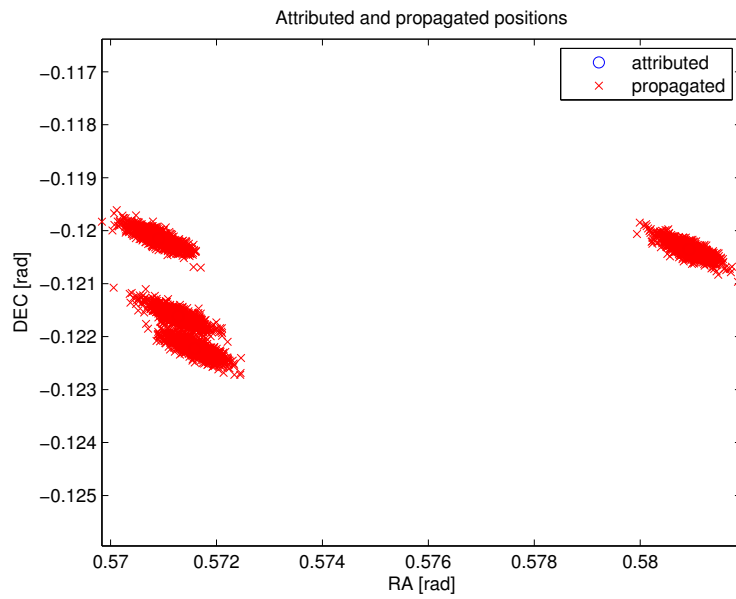


Figure 5.36: The attributed and propagated angular rates. In this parameter space the distributions are centered on the same values and have a similar uncertainty.

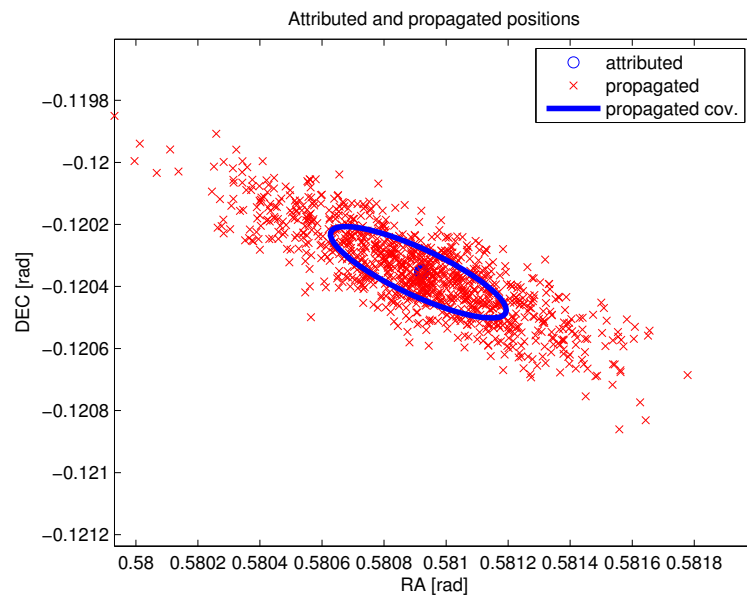


Figure 5.37: The propagated covariance of the angular position of one object. The ellipse is drawn at one standard deviation.

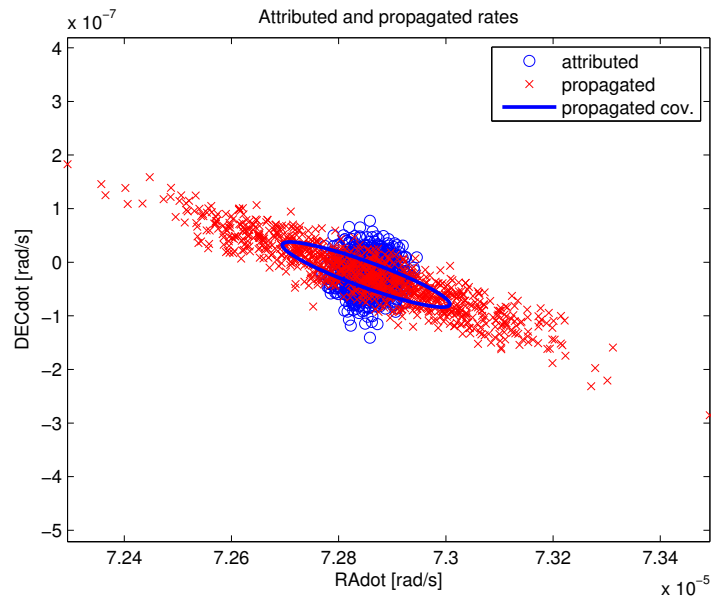


Figure 5.38: The propagated covariance of the angular rates of one object. The ellipse is drawn at one standard deviation.

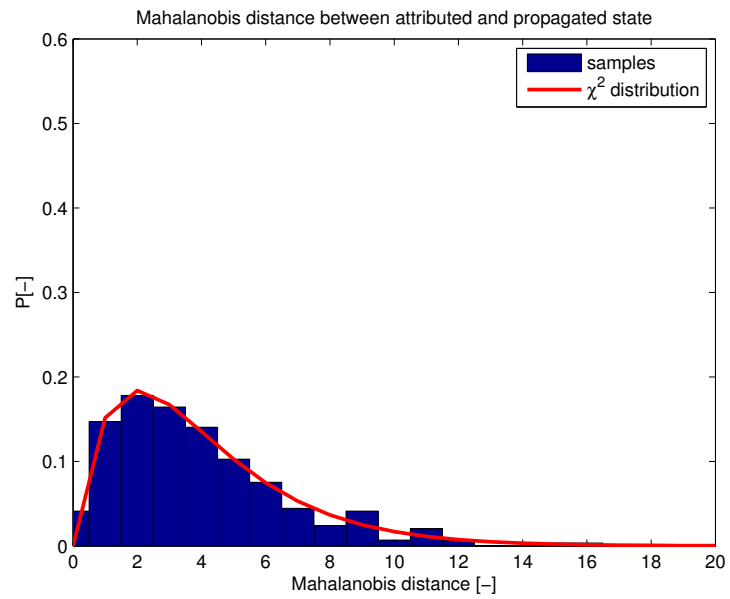


Figure 5.39: The Mahalanobis distance between the attributed and propagated state are distributed according to a four degrees of freedom χ^2 distribution.

5.7.2 Improving the convergence: results

An additional method was implemented in which first the $S = 2$ problem was treated. The result of the $S = 2$ problem is a set of all tracklet pairs and the minimized Mahalanobis distance that followed from the orbit determination. This Mahalanobis distance is used to quantify the probability that the two tracklets stem from the same object (by using Equation 5.6). A k-matrix can now be generated based on these probabilities. Each tracklet is associated to another tracklet according to the corresponding probability. If it is not associated to any other tracklet it is left as a single tracklet (on the diagonal of the k-matrix). For the result in Figure 5.40 denoted with ‘halfpairs’ the whole initial population is generated in this fashion. Although the EGA now starts at a much better point, it converges to the same solution as the original implementation. To have a part of the initial population spread out uniformly over the search space one half was replaced with k-matrices that are purely random. This is the ‘half pairs’ solution in the figures. This method seems to have the same performance as the one where the total initial population is based on the pairwise results. Finally the search space reduction and the ‘half pair’ methods are combined, resulting in a similar performance to the ‘half pairs’ solution.

In Figure 5.41 a zoom-in of the final generations can be seen. There are small differences between the solutions, but it is not clear whether these differences are due to the random nature of the algorithms or not. In any case, both the ‘pairs’ method and the search space reduction method come with a significant penalty in terms of computational costs. From these results it seems clear that it is not worth using these alternative approaches.

Figure 5.42 shows the average k-matrix at the end of 500 generations with the original implementation. In Figures 5.43 to 5.46 the differences between the average k-matrix of each alternative implementation and the original implementation are shown. Also in terms of the average k-matrix no large differences are found. In conclusion it can be said that the tested methods do not yield any improvements over the original implementation. Also in terms of computation time no improvements are found. The numerical propagation of the covariances that are needed in the Search Space Reduction (SSR) scheme require a large amount of computation time. Also the ‘smart’ initialization of the population requires an initial computation cost of computing the pair wise solutions. No precise study has been done on the computation time due to time constraints.

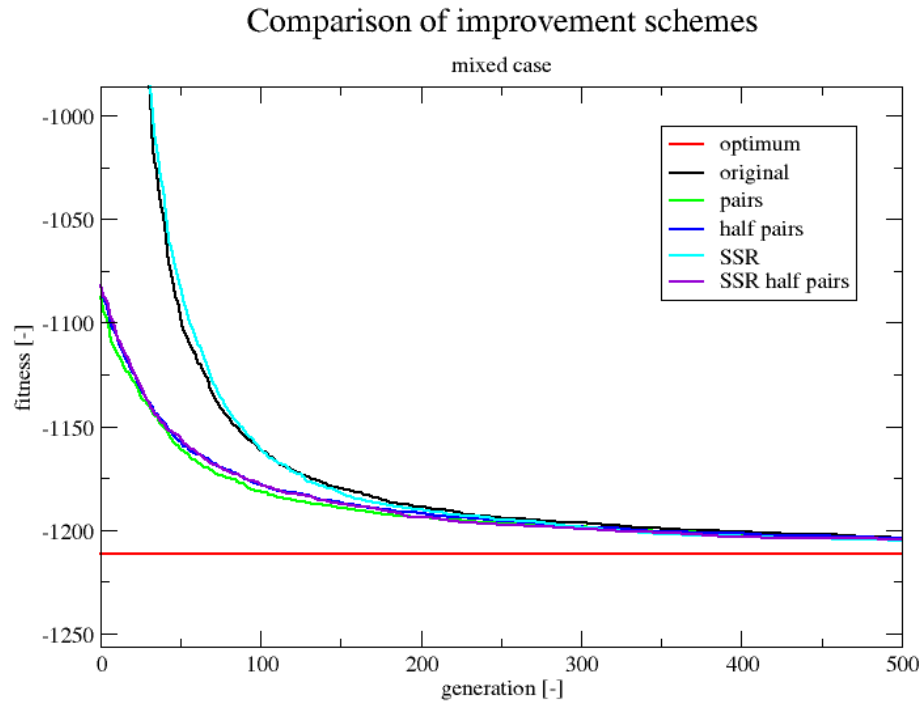


Figure 5.40: In this figure the ‘optimum’ solution is the straight line that represents the true solution. The ‘original’ solution is the algorithm as it was applied to real data in Section 5.5. The ‘pairs’ solution is the solution where the full initial population is based on the pairwise checking results, the ‘half pairs’ solution initializes only half of the population in this manner. The acronym SSR refers to the search space reduction method being applied.

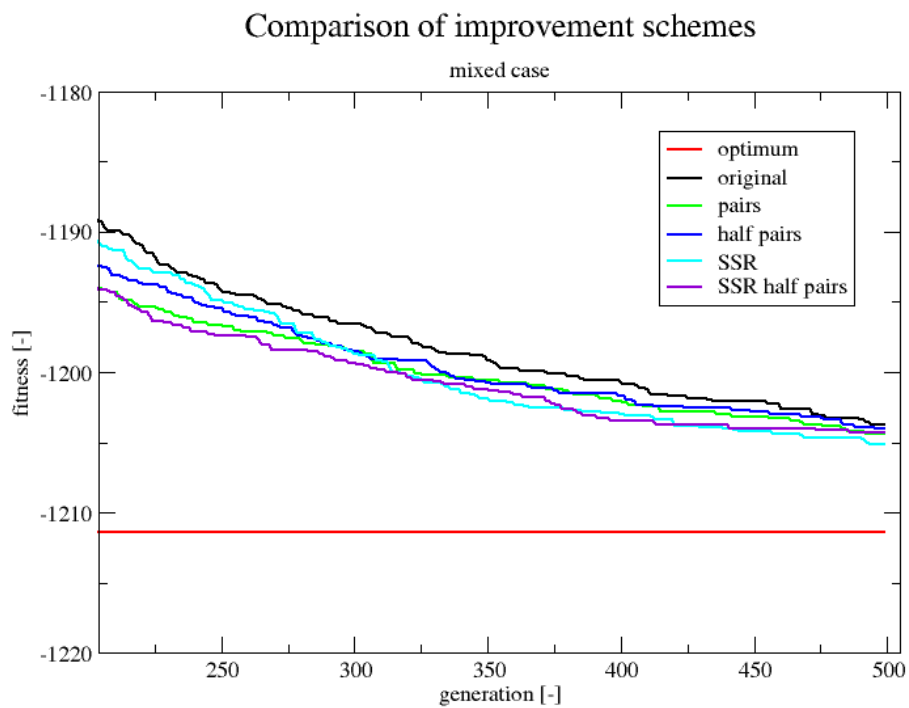


Figure 5.41: A zoom-in of the different performance of the different schemes.

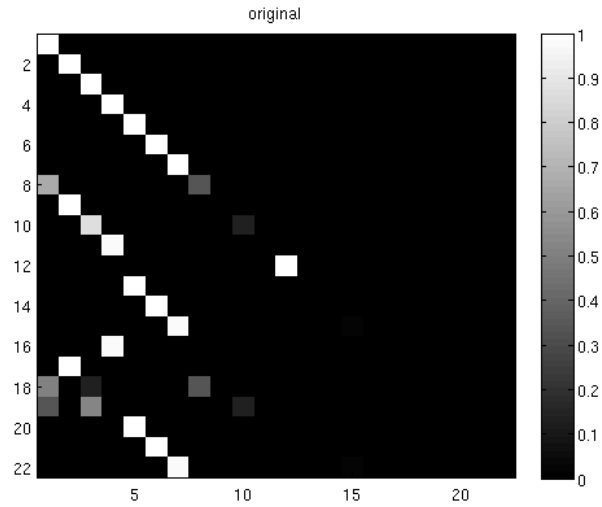


Figure 5.42: The average k-matrix at the end of 500 generations, with the original implementation of the algorithm.

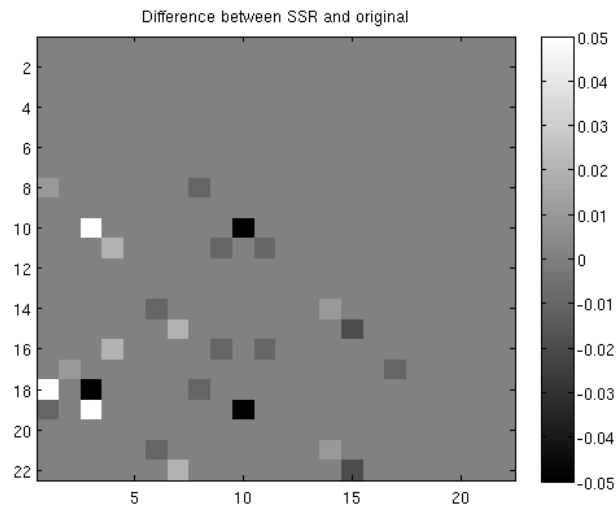


Figure 5.43: Difference between the search space reduction solution and the original implementation.

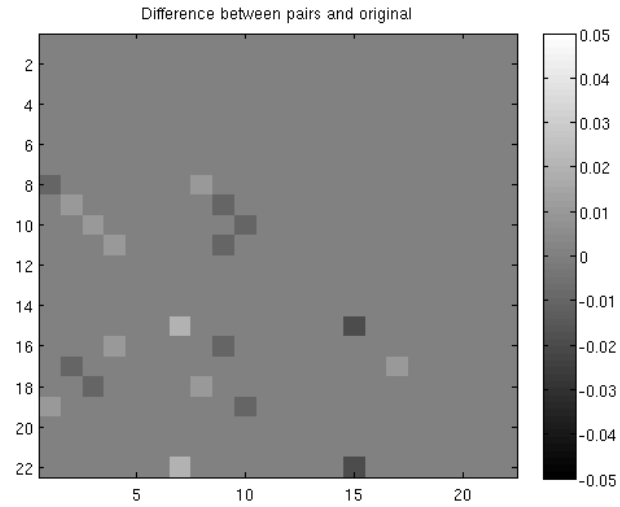


Figure 5.44: Difference between the ‘pairs’ solution and the original implementation.

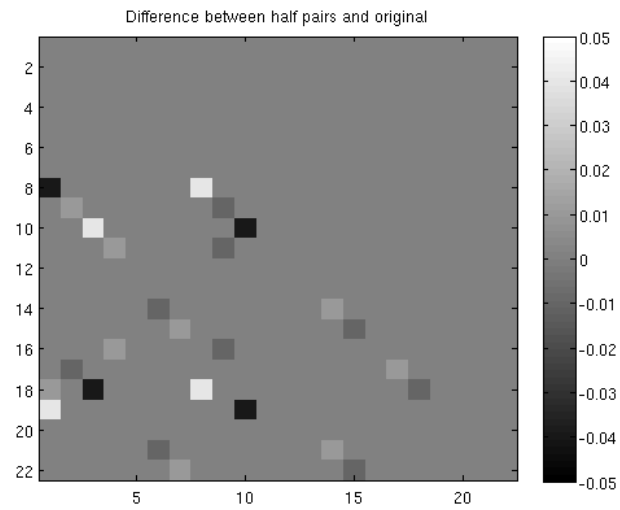


Figure 5.45: Difference between the ‘half pairs’ solution and the original implementation.

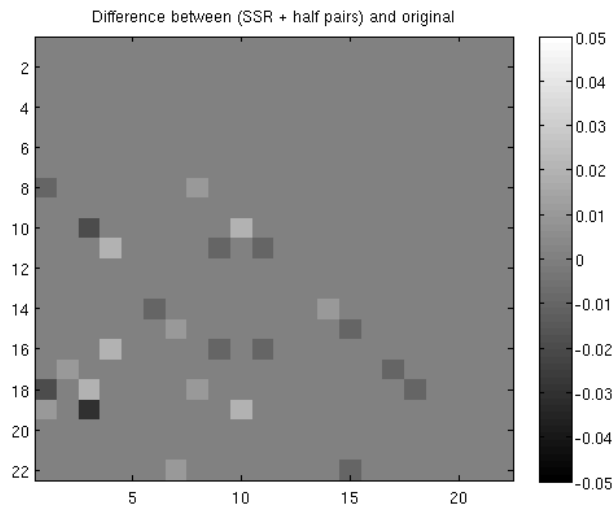


Figure 5.46: Difference between the ‘half pairs search space reduction’ solution and the original implementation.

Chapter 6

Conclusions

The Cosmos-Iridium collision in 2009, has demonstrated that the threat that space debris poses is not negligible. Currently about 16000 objects are tracked, sizes range from a 10 cm cross section in the LEO region to a one meter cross section in the GEO region. The space debris cataloging problem is different for the LEO region than for the MEO and GEO regions. This is mainly due to the perturbations that act on the objects and the different observables that are available for the different regions. In this work the focus is on the MEO and GEO regions. Here optical observations are used, and the perturbations due to Earth's non-homogeneous gravity field and its atmosphere are small. The space debris cataloging problem can be divided into several sub-problems, for optical observations these are: tracklet-to-catalog association, tracklet-to-tracklet association, and initial orbit determination. In this work mainly the tracklet-to-tracklet association and initial orbit determination problems are treated. The current methods that address these problems perform well for non-dense regions, where the objects are spaced far from one another. In dense environments such as break-up events and satellite clusters using these methods can lead to ambiguous solutions. The MHT method can take into account more data than the other methods, and is therefore capable of solving the cataloging problem in denser regions. However, this algorithm possesses an exponential computational complexity and needs to be simplified in order to work. In the near future the number of tracked objects is expected to grow by an order of magnitude. With that in mind the purpose of this thesis is to identify an algorithm capable of addressing the problem of space debris cataloging in the GEO region, in particular for highly dense regions, without possessing a restrictive computational com-

plexity.

Space debris cataloging can be described by the MTT problem. In this problem the unknowns are the number of objects in the surveillance region and their states, and the given information is a set of tracklets divided over a number of S fences. Each fence consists of tracklets that all stem from different objects or are false alarms. For $S \geq 3$ this is a NP-hard combinatorial optimization problem. The PBMH algorithms are known to perform well on NP-hard (combinatorial) problems. They excel at finding a good approximate result with a reasonable (polynomial) computational complexity. Therefore this family of algorithms could represent an interesting new approach to address the space debris cataloging problem. Besides that, these algorithms are left relatively unexplored in the MTT community. This work could therefore offer new insights to the problem and the way to address it. The algorithms that were applied are the PBIL, DE, GA, and the EGA algorithms. They were chosen based on a literature study, where each of the algorithms was shown to have success when applied to one or more combinatorial NP-hard problems. The algorithms were applied to three simulated test cases. In these test cases all objects followed a Keplerian motion, a Gaussian noise with $\sigma = 1''$ was added to the individual observations. The first test case consisted of four easy to track objects (low spatial density case), the second test case used observations of the ASTRA cluster (high density test case), and the last test case was a mix of both amounting to eight tracked objects. The orbits of the objects are determined with the newly developed OBVIOD method. This method is specifically developed to be used in these PBMH MTT algorithms, it is based on previous work detailed in (Siminski et al., 2014). The OBVIOD method is able to determine an orbit for two or more tracklets, and to give a statistically significant quantity that is related to the probability that the group of tracklets stems from the same object. The method from (Siminski et al., 2014) was chosen as a basis for this method since it seems to be the most promising $S = 2$ tracklet association method.

In each of the test cases the EGA was the only algorithm able to consistently converge towards the optimum solution. This can be explained by the fact that the EGA takes the dependencies between the associations into account by using the crossover operator, and that it always keeps the most valuable information in the population (elitism). The regular GA discards the whole population at the end of every generation, it therefore does not keep the

best information in the population. The DE algorithm works with difference vectors between different individuals. These difference vectors have a physical meaning when applied to a continuous problem (e.g. position, velocity, or acceleration). However for a combinatorial problem this is not the case, and therefore the difference vectors do not make much sense. The search is therefore very random in the case of DE, meaning that it is not able to find good solutions in a structured way. Finally the PBIL algorithm attempts to construct a probability distribution from which high fitness (high quality) individuals can be sampled. The weakness of this method is that it does not take the dependencies between parameters into account. If the parameters for the specific problem have strong dependencies then this algorithm will not perform very well. From the results it becomes clear that in the space debris cataloging problem in the MEO and GEO regions the parameters are strongly dependent on each other.

The OBVIOD method is shown to be robust, and it is able to provide good quality initial orbits that lie within the measurement noise. However, one should be careful when using this method as its performance depends on the observation geometry and the orbital regime of the object. If the first and last tracklets are placed close to half an orbital period or a full orbital period apart from each other the method is not able to compute a proper orbit (for near circular orbits). This is because the Lambert problem is used for the orbit computations, and in those geometries where the angle between the geocentric position vectors of the object at the first and last epochs is $\phi = \pi + 2\pi k$ or $\phi = 2\pi k$ there are infinite solutions to the Lambert problem (as the orbital plane is not constrained). The loss function of the OBVIOD method is the Mahalanobis distance. The topography of this function is smooth and contains one minimum point whenever the observation geometry is favorable. When the observations are close to the problematic geometries the topography might include several local minimum points. Another interesting observation is that an increased level of perturbation of the object does not necessarily translate to worse loss function values. For example, if the AMR of an object is increased the orbital parameters are still able to absorb the effect that has.

After having identified the EGA as being the best algorithm among the tested PBMH algorithms it was applied to a set of real data. The ASTRA cluster was observed at three different epochs, providing three tracklets for each of

the four objects in the cluster. The algorithm performed well on this data set as well, without showing any large differences to the simulated test cases.

The computational complexity of the EGA applied to the space debris cataloging problem has to be determined through experimentation. An experiment was devised where the number of objects ranged from two to 11, each observed over three fences. The average computation time needed to reach a close to optimum solution was measured. A polynomial and an exponential trend were fitted to these data points. The polynomial curve fits best, however this is not a proof that the algorithm possesses a polynomial complexity.

A few different strategies were tested to see if the algorithm could converge faster if problem specific information is introduced. One approach was to implement a so-called search space reduction algorithm, which eliminated parts of the search space during the EGA run. This was done by propagating orbits that have a high probability of being correct. All the tracklets that are too far away from the propagated state are marked as non valid associations. In the next generation no k-matrix is allowed to propose such a non valid association. A different approach was to first solve the $S = 2$ problem. The resulting tracklet pairs and their orbits were then used to initialize the population of the EGA. Both these approaches, as well as a combination of them, did not yield satisfactory results. Little difference in performance between the original implementation and the alternative implementations were found.

One important topic is the application of the algorithm to large data sets of 500-1000 tracklets. It could be straightforward to test this by evaluating all the individuals in a population in a parallel manner. It is even possible to get a better-than-linear speed up from implementing the EGA in a parallel fashion (Calegari et al., 1997).

If the algorithm is shown to work well on large data sets in its given form, it would be interesting to try to reduce the computation complexity of the algorithm. In the presented work already some results of the search space reduction and ‘smart’ initialization of the population have been shown. These methods are still not understood well enough and should be studied in more detail together with other possible ways to improve the computation times.

Something that has not been studied at all and which would be of high inter-

est is to see how the computation time scales as a function of the dimension of the problem.

Also the OBVIOD method might be improved. Currently it is restricted to the use of Keplerian motion, however it should be straightforward to implement a boundary value shooting method that can take perturbations into account. This would approximate the motion of the object more accurately. However, what is perhaps more important is that this modification will also (to some extent) address the problem of the difficult-to-solve-for geometries. This could mean that the OBVIOD method could be used even when the first and last tracklets are placed in a troublesome geometry. The downside of this is that to solve the resulting boundary value problem the object's state has to be propagated numerically, which means that it will take more computation time to compute orbits.

With these results a survey strategy could be designed where the difficult observation geometries are avoided. Another area of interest would be to test the algorithm for robustness. Tests can be made where e.g. the object density, the false alarm rate, or the missed detection probability are increased. The results from these tests would give new insight to the operational limits of the algorithm. Furthermore it would be good to compare the performance of the algorithm directly to a MHT implementation in order to find out in which regimes it is better to use one over the other.

Bibliography

- 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. Technical report, Numerica Corporation, 2013.
- S. Baluja. Population based incremental learning: A method for integrating genetic search based function optimization and competitive learning. Technical report, Carnegie Mellon University, 1994.
- R. H. Battin. *An Introduction to the Mathematics and Methods of Astrodynamics, Revised Edition*. Number ISBN-13 978-1563473425 in AIAA Education. AIAA Education Series, 1999.
- J. E. Beasley and P. C. Chu. A genetic algorithm for the set covering problem. *European Journal of Operational Research*, 1995.
- G. Beutler. *Methods of Celestial Mechanics*. Springer-Verlag, 2004.
- S. S. Blackman. Multiple hypothesis tracking for multiple target tracking. *IEEE AE Systems Magazine*, 2004.
- J. A. Brandon and B. Vo. A labeled multi-bernoulli filter for space object tracking. In *2014 AAS/AIAA Spaceflight Mechanics Meeting*, pages 11–15, 2014.
- P. Calegari, F. Guidicci, P. Kuonen, and D. Kobler. Parallel island-based genetic algorithm for radio network design. *Journal of Parallel and Distributed Computing*, 47:86–90, 1997.
- 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, 1997.

- S. Cook. The p versus np problem. In *Clay Mathematical Institute; The Millennium Prize Problem*, 2000.
- S. Deb, M. Yeddanapudi, K. Pattipati, and Y. Bar-Shalom. A generalized s-d assignment algorithm for multisensor-multitarget state estimation. *IEEE Transactions on Aerospace and Electronic Systems*, 33(2), 1997.
- W. Flury and N. Johnson. Position paper on orbital debris. *IAA*, 31:169–191, 1999.
- C. Früh. *Identification of Space Debris*. PhD thesis, Astronomical Institute of the University of Bern, 2011.
- C. Früh and T. Schildknecht. Object image linking of earth orbiting object in the presence of comets. *Advances in Space Research*, 49:594–602, 2012.
- K. Fujimoto and D. Scheeres. Short-arc correlation and initial orbit determination for space-based observations. In *Proceedings of the 2011 Advanced Maui Optical and Space Surveillance Technologies Conference*, 2011.
- K. Fujimoto, J. Herzog, T. Schildknecht, and D. J. Scheeres. Improvements to optical track association with the direct bayesian admissible region method. *AMOS*, 2013a.
- K. Fujimoto, D. J. Scheeres, J. Herzog, and T. Schildknecht. Association of short-arc optical tracks via the direct bayesian admissible region: theory and application. *Proc. '6th European Conference on Space Debris*, 2013b.
- K. Fujimoto, K.T. Alfriend, and T. Schildknecht. A boundary value problem approach to too-short arc optical track association. In *American Astronomical Society 14-201*, 2014a.
- 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), 2014b.
- D. E. Goldberg. *Genetic Algorithms in search, optimization and machine learning*. Addison-Wesley, 1999.
- R. H. Gooding. A procedure for the solution of lambert's orbital boundary-value problem. *Celestial Mechanics and Dynamical Astronomy*, 48(2):145–165, 1990.

- R. H. Gooding. A new procedure for the solution of the classical problem of minimal orbit determination from three lines of sight. *Celestial Mechanics and Dynamical Astronomy*, 66(4):387–423, 1996.
- J. Herzog, C. Früh, and T. Schildknecht. Build-up and maintenance of a catalogue of geo objects with zim-smart and zimsmart 2. In *61st International Astronautical Congress, Prague, Czech Republic. IAC-10 A*, volume 6, 2010.
- 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), 2012 15th International Conference on*, pages 2065–2072. IEEE, 2012.
- D. Izzo. Revisiting Lambert’s problem. *Celestial Mechanics and Dynamical Astronomy*, 121:1–15, 2015.
- Y. Jin. A comprehensive survey of fitness approximation in evolutionary computation. *Soft Computing*, 9(1):3–12, 2005. doi: 10.1007/s00500-003-0328-5.
- E. R. Lancaster and R. C. Blanchard. *A unified form of Lambert’s theorem*. National Aeronautics and Space Administration, 1969.
- A. Milani, D. Farnocchia, A. Rossi, T. Schildknecht, and R. Jehn. Orbit determination of space objects based on sparse optical data. *MNRAS*, 2010.
- O. Montenbruck and E. Gill. *Satellite orbits, Models, Methods and Applications*. Springer-Verlag Berlin Heidelberg, 2000.
- C. Moon, J. Kim, G. Choi, and Y. Seo. An efficient genetic algorithm for the traveling salesman problem with precedence constraints. *European journal of operational research*, 140:606–617, 2002.
- K. Muinonen, M. Granvik, D. Oszkiewicz, and T. Pieniluoma. Asteroid orbital inversion using a virtual-observation markov-chain monte carlo method. *Planetary and Space Science*, 73, 2012.
- P. Oliveto, J. He, and X. Yao. Time complexity of evolutionary algorithms for combinatorial optimization: a decade of results. *International Journal of Automation and Computing*, 4(3):281–293, 2007.

- G. C. Onwubolu. *Differential Evolution: A handbook for global permutation based combinatorial optimization*. Springer, 2009.
- A. B. Poore and S. Gadaleta. Some assignment problems arising from multiple target tracking. *Mathematical and computer modeling*, 43, 2006.
- R. S. Prado, R. C. P. Silva, F. G. Guimaraes, and O. M. Neto. Using differential evolution for combinatorial optimization: A general approach. Technical report, Instituto Federal de Minas Gerais, 2010.
- W.H. Press, S.A. Teukolsky, W.T. Vetterling, and B.P. Flannery. *Numerical Recipes: The Art of Scientific Computing*. Cambridge University Press, 2007.
- J. F. Pusztaszeri, P. E. Rensing, and T. M. Liebling. Tracking elementary particles near their primary vertex: A combinatorial approach. *JGO*, 9: 41–46, 1996.
- B. Rakdham. *Efficient multiple hypothesis track processing of boost-phase ballistic missiles using IMPULSEA-generated threat models*. PhD thesis, Naval Postgraduate School, 2009.
- C. R. Reeves. A genetic algorithm for flowshop sequencing. *Computers and Operations Research*, 22(1):5–13, 1995.
- 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.
- C. W. T. Roscoe, Jr. P. W. Schumacher, and M. P. Wilkens. Parallel track initiation for optical space surveillance using range and range-rate bounds. *AAS 13-767*, 2013.
- T. Schildknecht. *Optical astrometry of fast moving objects using CCD detectors*. PhD thesis, University of Bern, 1994.
- T. Schildknecht. Optical surveys for space debris. *Astronomy and Astrophysics Review*, 14:41–111, 2007.
- T. Schildknecht, U. Hugentobler, and A. Verdun. Optical observations of space debris with the zimmerwald 1-m telescope. *Advances in Space Research*, 19(2):221–228, 1997.

- T. Schildknecht, R. Musci, W. Flury, J. Kuusela, J. de Leon, and L. De Fatima Dominguez Palmero. Properties of the high area-to-mass ratio space debris population in geo. In *2005 AMOS Technical Conference Proceedings, Kihei, Maui, HI*, 2005.
- 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.
- P. Schumacher, M. Wilkins, and C. Roscoe. Parallel algorithm for track initiation for optical space surveillance. *Proc. '6th European conference on Space Debris'*, 2013.
- J. A. Siminski, H. Fiedler, and T. Schildknecht. Track association performance of the best hypotheses search method. *Proc '6th European Conference on Space Debris*, 2013a.
- J. A. Siminski, O. Montenbruck, H. Fiedler, and M. Weigel. Best hypotheses search on iso-energy grid for initial orbit determination and track association. *AAS 13-239*, 2013b.
- J.A. Siminski, O. Montenbruck, H. Fiedler, and T. Schildknecht. Short-arc tracklet association for geostationary objects. *Advances in Space Research*, 53(8), 2014.
- M. Sipser. The history and status of the p versus np question. In *Proceedings of the twenty-fourth annual ACM symposium on Theory of computing*, pages 603–618. ACM, 1992.
- R. Storn and K. Price. Differential evolution - a simple and efficient heuristic for global optimization over continuous spaces. *Journal of Global Optimization*, 1997.
- F. T. Sun, N. X. Vinh, and T.J. Chern. Analytic study of the solution families of the extended godal's time equation for lambert problem. *Journal of the Astronautical Sciences*, 35:213–234, 1987.
- G. Tommei, A. Milani, and A. Rossi. Orbit determination of space debris: admissible regions. *Celestial Mechanics and Dynamical Astronomy*, 2007.

- G. Tommei, A. Milani, D. Farnocchia, and A. Rossi. Correlation of space debris observations by the virtual debris algorithm. In *Proc. of the Fifth European Conference on Space Debris*, volume 30, 2009.
- I. Turkmen, K. Guney, and D. Karaboga. Genetic tracker with neural network for single and multiple target tracking. *Neurocomputing*, 69, 2006.
- D. Vallado. *Fundamentals of astrodynamics and applications*. Number ISBN10: 1881883140 in Space technology library. Microcosm press, third edition, 2007.
- J. Virtanen, M. Granvik, K. Muinonen, and D. Oszkiewicz. Bayesian orbit computation tools for objects on geocentric orbits. -, 2012.
- B. Wei, B. Nener, and W. Liu. Tracking of space debris via cphd and consensus. In *Control, Automation and Information Sciences (ICCAIS), 2015 International Conference on*, pages 436–441. IEEE, 2015.

Erklärung

gemäss Art. 28 Abs. 2 RSL 05

Name/Vorname:

Matrikelnummer:

Studiengang:

Bachelor Master Dissertation

Titel der Arbeit:

LeiterIn der Arbeit:

Ich erkläre hiermit, dass ich diese Arbeit selbständig verfasst und keine anderen als die angegebenen Quellen benutzt habe. Alle Stellen, die wörtlich oder sinngemäss aus Quellen entnommen wurden, habe ich als solche gekennzeichnet. Mir ist bekannt, dass andernfalls der Senat gemäss Artikel 36 Absatz 1 Buchstabe r des Gesetzes vom 5. September 1996 über die Universität zum Entzug des auf Grund dieser Arbeit verliehenen Titels berechtigt ist. Ich gewähre hiermit Einsicht in diese Arbeit.

Ort/Datum

Unterschrift

Michiel Zittersteijn

Date of birth: 02-05-1988

Nationality: Dutch

Phone: +41789568695

E-mail: zittersteijnmichiel@gmail.com

Niesenweg 2

3012, Bern

Switzerland

Education

02/2014-present

Astronomical Institute of the University of Bern (AIUB)

Bern, Switzerland

PhD Candidate

- Funded by the AIUB, CNES, and Thales Alenia Space France.
- The goal is to develop an algorithm capable of maintaining a catalog of space debris in the MEO and GEO regions in a robust and autonomous manner.
- Performing innovative research on orbit determination and data association methods.
- Working extensively with optimisation techniques, applied to both continuous and combinatorial optimisation problems.

09/2011-09/2013

Institut Supérieur de l'Aéronautique et de l'Espace (SUPAERO/ENSICA)

Toulouse, France

Master of Science program in Aeronautical and Space Systems

Major in Space Systems Engineering.

Research project: Application of a reinforcement learning (RL) algorithm to a flapping wing experiment. Built a simulator and applied an RL algorithm to design a controller for the flapping wing.

09/2006-07/2011

Technical University of Delft

Delft, the Netherlands

Bachelor of Science program in Aerospace Engineering

Final design project: designing a propulsion system to enable manned missions to Mars. Responsible for trajectory optimisation. Worked in a team of 10 students.