An ILP approach to Multi Hypothesis Tracking

Erik Liland

Report submitted to

Norwegian University of Science and Technology

Department of Engineering Cybernetics

O.S. Bragstads Plass 2D

Elektroblokk D, Gløshaugen,
7034 Trondheim, Norway

December 2016

Abstract

Tentativ som bare det

Autonomous surface vessels are, at least in navigational sense, unmanned ships which can be used to transport cargo and people as well as being used for surveillance and other tasks. To ensure a safe journey, the route planer must have a real time image of its surroundings in addition to map data. In the maritime environment, this is primarily done by rotating radars mounted on the ship itself. The challenge then is to know which measurement (reflection) belongs together from scan to scan. This report documents a survey of multi target tracking methods with a walk-through of derivations and implementations of an N-scan target oriented multi hypothesis tracker algorithm. The algorithm is able to account for missed targets and false measurements. As each set of measurements (scan) is received, the algorithm predicts the position to its known targets and accepts all measurements inside a certain area around the estimated position. Each of the new measurements are scored based on their distance from the estimated position, the uncertainty of the estimate and the (estimated) amount of clutter in each scan. This method allows for correlation of measurements multiple steps back in time, hence average out white noise. The algorithm was tested and found successful on both simulated data with noise as well as recorded data from a S-band maritime radar.

Contents

Abstract ii

List of Figures **v**

Nomenclature **1**

1 Introduction **1**
1.1 Motivation 1
1.2 Autosea 1
1.3 Problem description 1
1.4 Goals 1
1.5 Outline of report 2

2 Survey of multi-target tracking methods **2**
2.1 Tracking 2
2.2 Tracking system 2
2.3 Nearest Neighbor Filter 2
2.4 Probabilistic Data Association Filter 3
2.5 Joint Probabilistic Data Association Filter 4
2.6 Multi Hypothesis Tracker 5
2.6.1 Hypothesis Oriented MHT 5
2.6.2 Track Oriented MHT 6

3 Algorithm walk-through **6**
3.1 Flowchart 6
3.2 State estimation 7
3.3 Gating 8
3.4 Scoring 9
3.5 Clustering 9
3.6 Association 10
3.7 N-scan pruning 11

4 Linear programming **11**
4.1 Problem formulation 12
4.2 Solvers 14

5 Results **14**
5.1 Testing scheme 14
5.2 Simulation data 15
5.3 Simulations 17

5.3.1 Tracking performance 18
5.4 Runtime 23

6 Discussion **26**

7 Conclusion **26**

List of Figures

1 Algorithm flowchart 7
2 Validation region 10
3 Pruned track hypothesis tree 11
4 Track hypothesis tree 15
6 Simulation results for all solvers in scenario 1 18
7 Simulation results for all solvers in scenario 2 19
8 Simulation results for all solvers in scenario 3 20
9 Simulation results for all solvers in scenario 4 21
10 Simulation results for all solvers in scenario 5 22
11 Scenario 1 23
12 Scenario 2 24
13 Scenario 3 24
14 Scenario 4 25
15 Runtime for scenario 5 25

1 Introduction

1.1 Motivation

This report is the result of TTK4550 Engineering cybernetics specialization report. Where the aim is to *let the student specialize in a selected area based on scientific methods, collect supplementary information based on literature search and other sources and combine this with own knowledge into a project report.* The learning outcome of this project is to give the student an extensive knowledge in a current problem, good knowledge in related topics and relevant scientific literature.

1.2 Autosea

This project is a part of the Sensor fusion and collision avoidance for autonomous surface vessels (Autosea) project, which is a collaboration between NTNU AMOS and Maritime Robotics, DNV GL and Kongsberg Maritime. The vision for the project is to *attain world-leading competence and knowledge in the design and verification of methods and system for sensor fusion and collision avoidance for autonomous surface vessels.*

1.3 Problem description

The proposed title for this project (and the following master-thesis) was "Multi-target tracking using radar and AIS". The key idea behind this formulation is that radar and AIS have different strengths and weaknesses, and if utilized properly, the strengths of both system can be exploited, while the weaknesses can be reduced. The following task was proposed for this project:

- Write a survey on multi-target and multi-sensor tracking methods.
- Implement a multi-target tracking method that fuses radar data with AIS data under benign assumptions.
- Describe the method and summarize the findings in a report.

1.4 Goals

The goals for this project is to implement a radar tracking algorithm in Python based on one of the surveyed methods, test this with simulated and real data and analyze the performance for different conditions and discuss methods for improvement.

1.5 Outline of report

In section 2, different methods for target tracking and data association are presented in varying depth, more on the most relevant methods and an overview on the others. In section 3, the algorithm based on the selected approach is thoroughly explained, and in section 4 the integer optimization problem that arises in section 3 is elaborated. The results are presented in section 5 and discussed in section 6.

2 Survey of multi-target tracking methods

The aim of this section is to give the reader a brief overview of tracking as a problem and a feeling for the most popular methods, their assumptions and strong and weak properties as seen from an 2D maritime anti collision perspective.

2.1 Tracking

Tracking of an object (target) is the process of estimating its state (i.e. position and velocity) based on discrete measurements from an observation system. An observation system can be a radar, sonar or any other sensor that passively or actively detects objects within an area or volume.

2.2 Tracking system

A tracking system can be interpreted as either the complete system from the signal processing level to the finished tracks, or as in this text: *A system that associates consecutive measurements from an observation system, and initiate or assigns them to tracks.* A track is a subset of all the measurements from the observation system that is believed to originate from the same target. The challenge knowing which measurement originating from which (real) target is the core at any tracking system. This association problem is non-trivial even under ideal situations, and the addition of spurious measurements and missed targets only increases the complexity.

There has been developed a large variety of methods to overcome this association problem, and most of them have several sub-methods. In the following subsections, some of the most common and popular methods will be presented.

2.3 Nearest Neighbor Filter

The Nearest Neighbor Filter (NNF)is the simplest approach in tracking, where one always select the closes neighbor as the consecutive measurement in the track. This approach suffers from being very vulnerable to clutter and dense target scenarios. It can be somewhat improved by estimating an a-priori state through a Kalman Filter and selecting the nearest neighbor to the estimate. This extension is sometimes refereed to as Nearest Neighbor Standard Filter (NNSF). Under the (normal) assumption that each target can at maximum generate one measurement, the NNF and NNSF are both single-target methods. They can however be expanded to multi-target variants by formulating the problem as a global least squares integer optimization problem. With this extension the NNSF is almost becoming a zero-scan multi hypothesis tracker. NNF and NNSF are the only method presented which is non-probabilistic, and does not assume specific models for noise, etc.

2.4 Probabilistic Data Association Filter

Probabilistic Data Association Filter (PDAF) is a *single-target* Bayesian association filter which is based on single scan probabilistic analysis of measurements. At each scan the filter is calculating a most probable measurement based on a weighted sum on the measurement innovations inside a validation region.

$$\hat{\mathbf{y}} \triangleq \sum_{j=1}^{m} \beta_j \tilde{\mathbf{y}}_{\mathbf{j}} \tag{1}$$

where

$$\begin{aligned} \beta_j &= \text{the probability of measurement j to be the correct one} \\ \tilde{y}_j &= y_j - \hat{y}_j \quad \text{the j-th measurement innovation} \end{aligned}$$

PDAF is computationally modest (approximate 50% more computationally demanding than a Kalman Filter [1] p.163) and have good results in an environment with up to about 5 false measurement in a 4σ validation region [1]. PDAF does not include track initialization and assumes that at most one measurement can originate from an actual target. It also assumes that clutter is uniformly distributed in the measurement space and that the targets history is approximated by a Gaussian with a calculated mean and covariance (single scan).

PDAF can be used for multiple targets, but only as multiple copies of the single-target filter [2]. Since PDAF is generating a "best guess"-measurement from all the measurements inside its validation region, it can suffer from track coalescence. This coalescence occurs when two targets have similar paths, and the resulting tracks will be an "average" of the two (actual) tracks. There has been done some work to overcome this coalescence [3].

2.5 Joint Probabilistic Data Association Filter

Joint Probabilistic Data Association Filter (JPDAF) is a *multi-target* extension of the Probabilistic Data Association Filter in which joint posteriori association probabilities are calculated for every target at each scan. Both PDAF and JPDAF use the same weighted sum (1), the key difference is the way the weight β_j is calculated. Whereas PDAF treats all but one measurement inside its validation region as clutter, in JPDAF the targets which interacts (one cluster) are treated as connected and the connected β_js are computed jointly across the cluster set with a given set of active targets inside the cluster. The probability of a measurement j belonging to a target t is [2]

$$\begin{aligned}\beta_j^t &= \sum_{\chi} P\{\chi|Y^k\}\hat{\omega}_{jt}(\chi) \\ \beta_0^t &= 1 - \sum_{j=1}^{m} \beta_j^t\end{aligned} \tag{2}$$

where

$$P\{\chi|Y^k\} = \frac{C^{\phi}}{c} \prod_{j:\tau_j=1} \frac{exp[-\frac{1}{2}(\mathbf{\tilde{y}_j^{t_j}})^T S_{t_j}^{-1}(\mathbf{\tilde{y}^{t_j}})]}{(2\pi)^{M/2}|S_{t_j}|^{1/2}} \prod_{t:\delta_t=1} P_D^t \prod_{t:\delta_t=0} (1 - P_D^t). \tag{3}$$

Where

$$\begin{aligned}\beta_j^k &= \text{the probability that measurement j belongs to target k} \\ \beta_0^k &= \text{the probability that no measurement belongs to target k} \\ \chi &= \text{All feasible events} \\ Y^k &= \text{all candidate measurements up to and included time k} \\ \hat{\omega}_{jt} &= \begin{cases} 1, \text{if } \chi_{jt} \text{ occurs} \\ 0, \text{otherwise} \end{cases} \\ m &= \text{number of measurements}\end{aligned}$$

Since the JPDAF is calculating joint probability for all the combinations of measurement associations in the cluster, the computation demand is growing exponentially with the numbers of tracks and measurements in the cluster. A real time implementation of the JPDAF has been developed and patented by QinetiQ [4], and described in [5], and an approach for avoiding track coalescence has been proposed by [3].

2.6 Multi Hypothesis Tracker

Multiple hypothesis tracking (MHT) is a category of methods that are based around the concept of evaluating multiple combinations of measurements to data association, and rank them with respect to their score. In contrast to PDA methods which in some cases will estimate an "average" of two tracks as the true on (coalesce), MHT methods split when in doubt. The original MHT algorithm was presented in [6], where a hypothesis oriented MHT was developed. Following this, a track oriented MHT was proposed in [7] and improved by [8].

2.6.1 Hypothesis Oriented MHT

Hypothesis Oriented Multiple Hypothesis Tracker (HOMHT) or Measurement Oriented Multiple Hypothesis Tracker (MOMHT) is a fully Bayesian approach where direct probabilities of global joint measurement-to-target association hypothesis are calculated. The algorithm initiates tracks and handles missing measurements, it has a recursive nature and it allows for clustering for quicker computation. One of the main benefits of MHT is the ability to utilize multiple scans (history) to aid in the data association, in other words to use all the available data when taking decisions. TOMHT was developed under the assumption that at most one measurement can originate from each target in each scan, and that a target does not necessary show on every scan (Probability of detection, $P_D < 1$). When evaluating the probability of a hypothesis, the MHT takes into account the false-alarm statistics of the measurement system, the expected density of targets and clutter and the accuracy of the target estimates.

The probability of each data association hypothesis was developed by Reid in [6]

$$P_i^k = \frac{1}{c} P_D^{N_{DT}} (1 - P_D)^{(N_{TGT} - N_{DT})} \beta_{FT}^{N_{FT}} \beta_{NT}^{N_{NT}} \left[\prod_{m=1}^{N_{DT}} N(\mathbf{Z_m} - \boldsymbol{H}\bar{\mathbf{x}}, \boldsymbol{B}) \right] P_g^{k-1} \tag{4}$$

where

$$
\begin{aligned}
P_i^k &= \text{the probability of hypothesis } \Omega_i^k \text{ given measurements up through time } k \\
P_D &= \text{the probability of detection} \\
\beta_{FT} &= \text{the density of targets} \\
\beta_{NT} &= \text{the density of previously unknown targets that have been detected} \\
N_{DT} &= \text{number of designated target} \\
N_{FT} &= \text{the number of false targets} \\
N_{NT} &= \text{the number of new targets} \\
N_{TGT} &= \text{is the number of targets} \\
\mathbf{Z_m} &= \text{the m-th measurement in the current scan} \\
\boldsymbol{H} &= \text{the observation matrix} \\
\boldsymbol{B} &= \text{the measurement covariance}
\end{aligned}
$$

2.6.2 Track Oriented MHT

Track Oriented Multiple Hypothesis Tracker (TOMHT) is a "bottom-up" approach where the tracks are assumed initialized, and for each scan the track splits whenever there are more than one feasible measurement in the validation region (in addition to the no-measurement hypothesis). The new track hypothesis state is generated from the posteriori filtered estimate from a Kalman Filter, and a score/cost is calculated using (20) from [8]. Following the addition of a new track hypothesis, the tracks are divided into clusters where all tracks which share measurements from the N-latest scan are in one cluster. The clusters can then be analyzed as standalone global problems to find the beast possible combination of (possibly mutual exclusive) measurement associations. Linear programming (LP)- and Integer Linear Programming (ILP)-based methods as proposed by [9] can be used to find the best combinations of newly created track hypotheses in accordance to the assumption that a measurement only can be assigned to one target and that one target can maximally create one measurement.

To limit the size of the track hypothesis tree, the unused edges of the track hypothesis tree will be removed after N scans.

3 Algorithm walk-through

3.1 Flowchart

Figure 1 shows a flowchart of the track oriented MHT algorithm presented in this section. An important difference between this approach compared to Reid's original measurement oriented MHT [6], is the need for external initialization of targets.

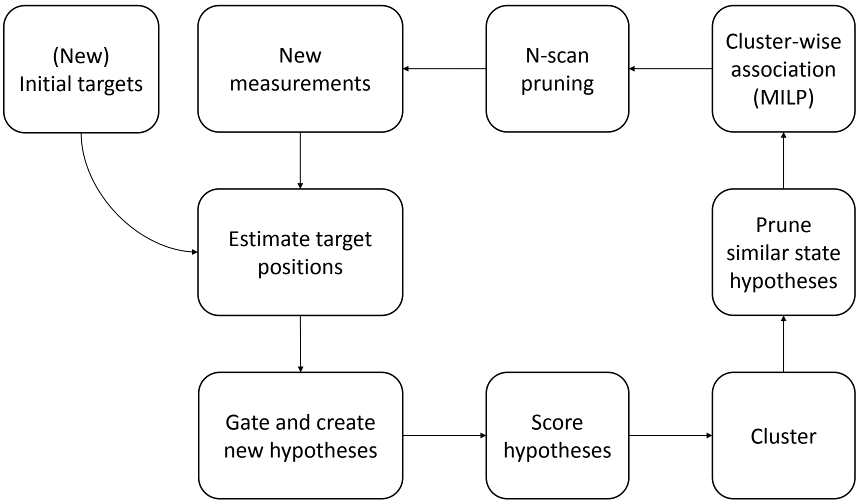


Figure 1: Algorithm flowchart

3.2 State estimation

When a new set of measurement is arriving, it is desirable to "guess" where the target might be before looking for matching measurements. This can be done though a model of the targets dynamics and a state estimator. For the purpose of tracking ships in a local frame (Cartesian plane), we compose a state vector with four states

$$\mathbf{x} = \begin{bmatrix} x & y & \dot{x} & \dot{y} \end{bmatrix}^T \tag{5}$$

where the change of velocity (acceleration) is assumed zero. The latest assumption is compensated with an increased system noise covariance in the model. For a linear system, a Kalman Filter is an optimal estimator which have the following time evolution assumptions.

$$\begin{aligned} \mathbf{x}(k+1) &= \mathbf{\Phi}\mathbf{x}(k) + \mathbf{\Gamma}\mathbf{w} \\ \mathbf{z}(k) &= \boldsymbol{H}\mathbf{x}(k) + \mathbf{v} \end{aligned} \tag{6}$$

where

$$
\begin{aligned}
\mathbf{\Phi} &= \text{the state transition matrix} \\
\mathbf{\Gamma} &= \text{the disturbance matrix} \\
\mathbf{w} &= \text{the process noise} \\
\boldsymbol{H} &= \text{the measurement matrix} \\
\mathbf{v} &= \text{the observation noise}
\end{aligned}
\tag{7}
$$

. The procedure of estimating the target state at the next time step is according to the "time update" equations of the Kalman Filter.

$$
\begin{aligned}
\bar{\mathbf{x}}(k+1) &= \mathbf{\Phi}\hat{\mathbf{x}}(k) \\
\bar{\boldsymbol{P}}(k+1) &= \mathbf{\Phi}\hat{\boldsymbol{P}}(k)\mathbf{\Phi}^T + \boldsymbol{Q}
\end{aligned}
\tag{8}
$$

Where Q is the system covariance matrix. The model have the following parameters,

$$
\mathbf{\Phi} = \begin{bmatrix} 1 & 0 & T & 0 \\ 0 & 1 & 0 & T \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \mathbf{\Gamma} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix} \quad \boldsymbol{H} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}
$$

$$
\boldsymbol{Q} = \sigma_v^2 \begin{bmatrix} \frac{T^3}{3} & 0 & \frac{T^2}{2} & 0 \\ 0 & \frac{T^3}{3} & 0 & \frac{T^2}{2} \\ \frac{T^2}{2} & 0 & T & 0 \\ 0 & \frac{T^2}{2} & 0 & T \end{bmatrix} \quad \boldsymbol{R} = \sigma_r^2 \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}
$$

where T is the time between the current and the previous measurement, σ_v^2 is the system velocity variance and σ_r^2 is the measurement variance. This model is very common due to its simplicity, and is used in among others [6], [10] and [11].

3.3 Gating

To avoid calculating the likelihood for all possible combinations of targets and measurements, some sort of selection criteria is needed when creating new hypotheses. One way of doing this gating is to select all measurements that are within a certain confidence region (generally an ellipsoid), as done in [6].

$$
\begin{gathered}
\boldsymbol{B} = \boldsymbol{H}\bar{\boldsymbol{P}}\boldsymbol{H}^T + \boldsymbol{R} \\
(\mathbf{Z_m} - \boldsymbol{H}\bar{\mathbf{x}})^T\boldsymbol{B}^{-1}(\mathbf{Z_m} - \boldsymbol{H}\bar{\mathbf{x}}) \leq \eta^2
\end{gathered}
$$

Where η is the gate size. Chi-Square value with a given confidence interval and two degrees of freedom. For each measurement inside the region, calculate the a posteriori state and covariance using the "measurement update" equations in the Kalman Filter (9) and create a new hypothesis with this state and covariance as initial.

$$\begin{aligned}
\tilde{\mathbf{y}} &= \mathbf{z} - \boldsymbol{H}\bar{\mathbf{x}} \\
\boldsymbol{S} &= \boldsymbol{H}\bar{\boldsymbol{P}}\boldsymbol{H}^T + \boldsymbol{R} \\
\boldsymbol{K} &= \bar{\boldsymbol{P}}\boldsymbol{H}^T\boldsymbol{S}^{-1} \\
\hat{\mathbf{x}}(k) &= \bar{\mathbf{x}} + \boldsymbol{K}\tilde{\mathbf{y}} \\
\hat{\boldsymbol{P}}(k) &= (\boldsymbol{I} - \boldsymbol{K}\boldsymbol{H})\,\bar{\boldsymbol{P}}
\end{aligned} \tag{9}$$

3.4 Scoring

Each hypothesis are scored according to [8]:

$$\begin{aligned}
NLLR_{t,j}(k) &= \frac{1}{2}\left[\tilde{y}_k^T S_{tj}(k)^{-1}\tilde{y}_k\right] + \ln\frac{\lambda_{ex}|2\pi S_{tj}(k)|^{1/2}}{P_{D_t}(k)} \\
\tilde{y}_k &= z_j(k) - \hat{z}_t(k|k-1)
\end{aligned} \tag{10}$$

where the cumulative NLLR is

$$l_t^k \triangleq \sum_{l=0}^{k} NLLR_{t,j(t,l)}(l) \tag{11}$$

3.5 Clustering

Since the global problem of finding the optimal selection of hypotheses is growing exponentially with the number of hypotheses, it is computationally beneficial to split the problem into smaller problem. This can only be done to targets that does not share any measurements within the N-latest time step. This is done efficiently through breath-first-search or depth-first-search on a graph made from the hypothesis tree.

Confidence	70%	80%	90%	95%	97.5%	99%	99.5%
η^2	2.41	3.22	4.61	5.99	7.38	9.21	10.60

Table 1: χ^2 values for two degrees of freedom for selected confidence values

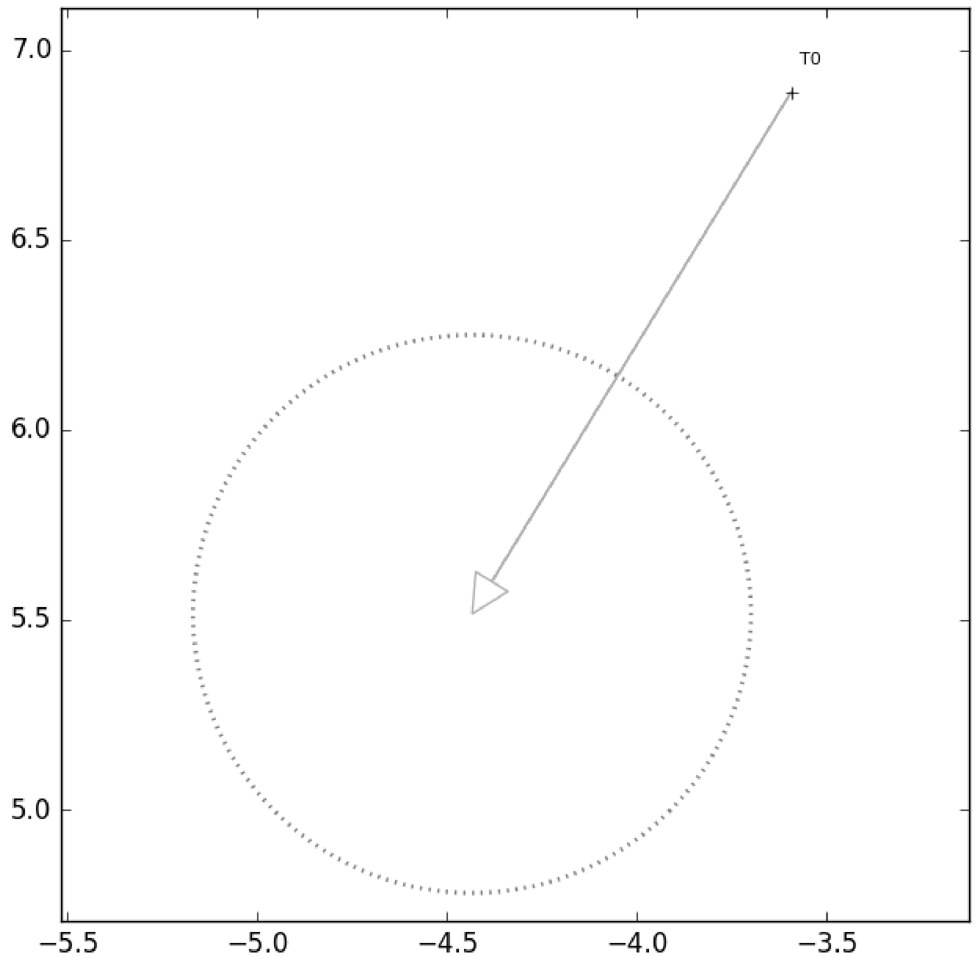


Figure 2: Validation region

3.6 Association

When the targets are divided into independent clusters, each of them can be treated as a global problem where we want to minimize the cost of the selected hypotheses (leaf nodes), while fulfilling the constraints that each measurement can only be a part of one track and that minimum and maximum one hypothesis can be selected from each target. Since only binary values, selected of not selected, is desired for selection of hypotheses, the problem becomes a integer linear optimization problem.

3.7 N-scan pruning

To keep the computational cost within reasonable limits, it is necessary to limit the amount of time steps backwards in time that the algorithm computes. This is done by removing all but the active hypothesis at the current root node, and assign the remaining hypothesis as new root node.

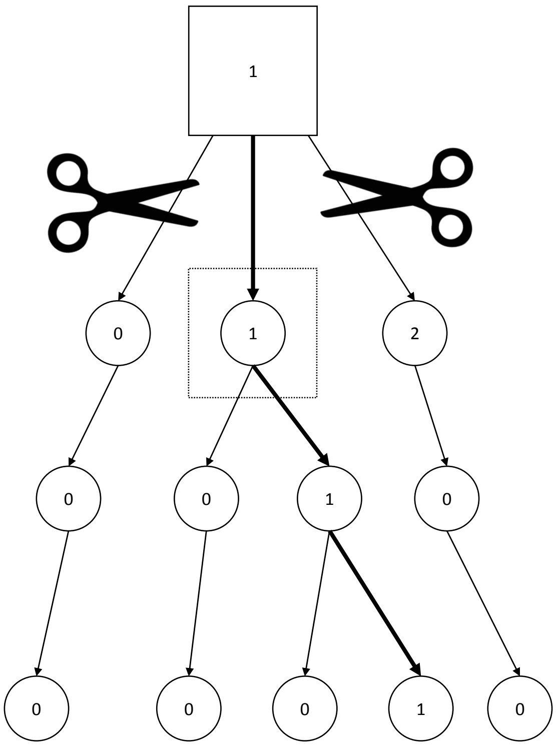


Figure 3: Pruned track hypothesis tree

4 Linear programming

The aim of this section is to elaborate the use of linear programming to solve the data association problem in MHT that arises when there are multiple (possible mutual exclusive) possibilities of measurement arrangements within the existing set of tracks. As with any optimization problem, we need an objective function which tells us how good or bad a given assignment is, and a set of constraints that limits the solution to the physical limits and our assumptions.

4.1 Problem formulation

Storms and Spieksma [9] are suggesting an Integer Linear Programming (ILP) scheme with Linear Programming (LP) relaxation and Greedy Rounding Procedure (GRP) as solvers.

$$\begin{aligned} \underset{f}{\text{minimize}} \quad & f = \sum_{z \in \mathbf{Z^*}} c_z x_z \\ \text{subject to} \quad & \sum_{z \in \mathbf{Z^*}, z(k) = z^k_{i_k}} x_z = 1, \forall k = 1, \ldots, N \text{ and } i_k = 1, \ldots, M_k \\ & x_z \in \{0, 1\} \end{aligned} \tag{12}$$

where, $c_z = -lnQ_z$ and the likelihood for a track is

$$Q(z) = \prod_{k=1}^{N} (P^k_\phi)^{\Delta_{i_k}} \left\{ \left[\frac{P_d f^k_\delta(z^k_{i_k}|z)}{\lambda_\varphi f^k_\varphi(z^k_{i_k})} \right]^{\delta^k_{i_k}} \left[\frac{\lambda_\nu f^k_\nu(z^k_{i_k}|z)}{\lambda_\varphi f^k_\varphi(z^k_{i_k})} \right]^{\nu^k_{i_k}} \right\}^{(1-\Delta_{i_k})} \tag{13}$$

where,

$$\Delta_{i_k} = \begin{cases} 1, i_k = 0(\text{dummy-report}) \\ 0, \text{otherwise} \end{cases}$$
$$P_\phi^k = \begin{cases} 1 - P_d, z_{i_k}^k \text{is a missing report} \\ 0, \text{otherwise} \end{cases}$$
$$\nu_{i_k}^k = \begin{cases} 1, z_{i_k}^k \text{initiates a track} \\ 0, \text{otherwise} \end{cases}$$
$$\delta_{i_k}^k = \begin{cases} 1, z_{i_k}^k \text{proceeds as a track} \\ 0, \text{otherwise} \end{cases}$$
$$P_d = \text{probability of detection}$$
$$\lambda_\varphi = \text{expected number of false alarms (Poisson distribution)}$$
$$\lambda_\nu = \text{expected number of new targets (Poisson distribution)}$$
$$f_\nu^k = f_\varphi^k = \frac{1}{\pi r^2}, \text{where r is the sensor range}$$
$$f_\delta^k = \frac{e^{-\frac{1}{2}[z_{i_k}^k - h(\bar{s}(t_k))]^T B^{-1} [z_{i_k} - h(\bar{s}(t_k))]}}{\sqrt{(2\pi)^n |B|}}$$
$$n = \text{dimension of measurement vector}$$
$$h(\cdot) = \text{transformation of Cartesian to polar}$$
$$\bar{s} = \text{predicted state vector}$$
$$B = \text{covariance of } z_{i_k}^k - h(\bar{s}(t_k))$$

This approach uses a rather inelegant summation notation which is not on a standard (I)LP format.

In this work we propose a more compact formulation of the data association problem on ILP standard form.

$$\begin{aligned} \text{maximize} \quad & \mathbf{c}^T \mathbf{x} \\ \text{s.t.} \quad & \boldsymbol{A_1} \mathbf{x} \leq \mathbf{b_1} \\ & \boldsymbol{A_2} \mathbf{x} = \mathbf{b_2} \\ & \mathbf{x} \in \{0, 1\} \end{aligned} \tag{14}$$

Where $\boldsymbol{A_1}$ is a $N_1 \times M$ binary matrix with N_1 real measurements and M track hypotheses (all leaf nodes), where $\boldsymbol{A_1}(i,j) = 1$ if hypothesis j are utilizing measurement i, 0 otherwise. The measurements and hypothesis are indexed by the order they are visited by a depth first search (DFS). $\boldsymbol{A_2}$ is a $N_2 \times M$ binary matrix where N_2 is the number of targets in the cluster and $\boldsymbol{A_2}(i,j) = 1$ if hypothesis j belongs to target i. $\mathbf{b_1}$ is a N_1 long vector with ones and $\mathbf{b_2}$ is a N_2 long vector with ones. $\mathbf{c}$ is a N long vector with a measure of the goodness of the track hypotheses. For example in Figure 4 at time step 2, the A matrices and C vector would be:

$$\begin{aligned}
\boldsymbol{A_1} &= \begin{bmatrix} 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \mathbf{b_1} = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} \\
\boldsymbol{A_2} &= \begin{bmatrix} 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \end{bmatrix} \mathbf{b_2} = \begin{bmatrix} 1 \\ 1 \end{bmatrix} \\
\mathbf{c} &= \begin{bmatrix} \lambda_1 & \lambda_2 & \lambda_3 & \lambda_4 & \lambda_5 & \lambda_6 & \lambda_7 & \lambda_8 & \lambda_9 \end{bmatrix}^T
\end{aligned} \tag{15}$$

4.2 Solvers

There are a lot of of-the-shelf integer linear program (ILP) and mixed integer linear program (MILP) solvers on the marked, both free open source and commercial. Since our problem is formulated on standard form, it can easily be executed on several solvers, and we can compare runtime and performance. In this report, the following solver are tested:

- CBC (Free, COIN-OR)
- CPLEX (Commercial (Free academic), IBM)
- GLPK (Free, GNU)
- Gurobi (Commercial (Free academic), Gurobi)

5 Results

5.1 Testing scheme

The evaluation of the MHT algorithm is two-sided, firstly the algorithm must be able to track under challenging conditions, secondly it must be able to do this without having an ever growing computationally cost. The first performance metric is how well the algorithm is estimating the true position to the object it is tracking. This is measured as the Euclidean distance:

$$\Delta P = \|\mathbf{p}_{track} - \mathbf{p}_{target}\|_2 \tag{16}$$

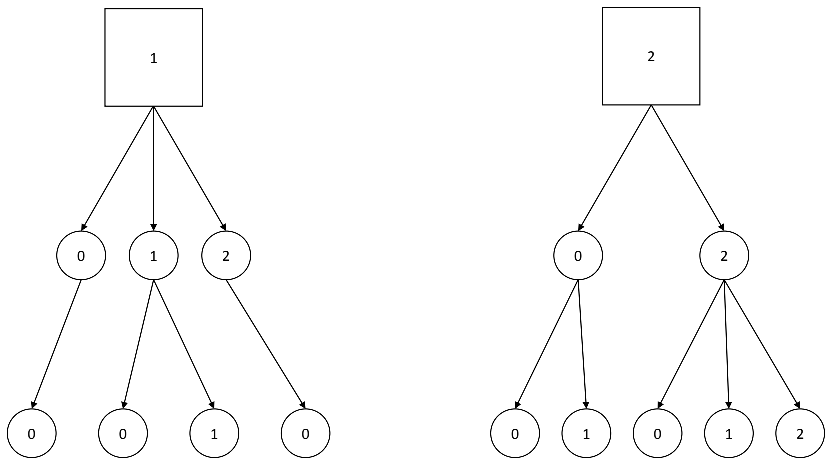


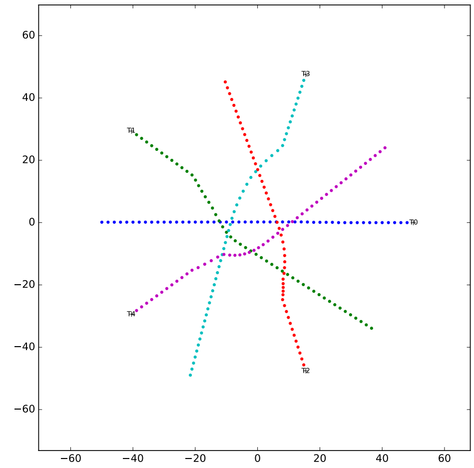
Figure 4: Track hypothesis tree

where the track is considered correct if $\Delta P \leq \varepsilon_p$ for all t after initial convergence. If a track is deviating more that the threshold and is never within the threshold again, it is considered lost at the time-step it exceeded the threshold. If the track should converge after exceeding the threshold, it is considered restored at the time-step it is returning within the limit. The algorithm is tested on five scenarios:

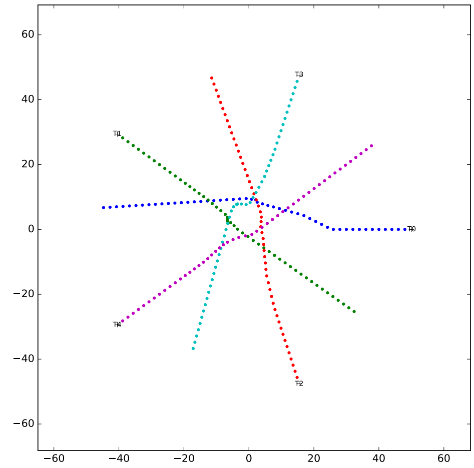
- Five fully cooperating ships
- Five partially cooperative ships
- Five ships avoiding obstacles with large space
- Five ships avoiding obstacles with little space
- Five parallel ships

5.2 Simulation data

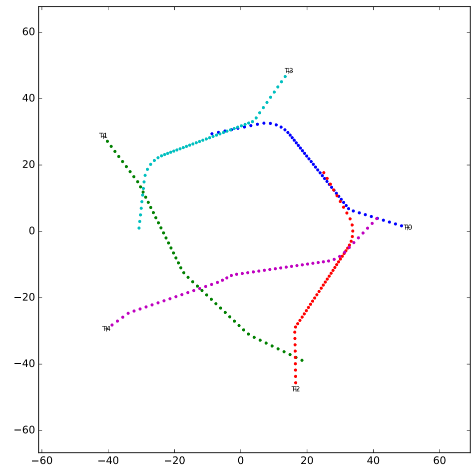
Scenario one through four is generated as a recording of time and position from an Autonomous Surface Vessel (ASV)-simulator with collision avoidance (COLAV) which is a project under work by D. Kwame Minde Kufolaor at NTNU. The targets are configured such that they need to maneuver to avoid collision with each



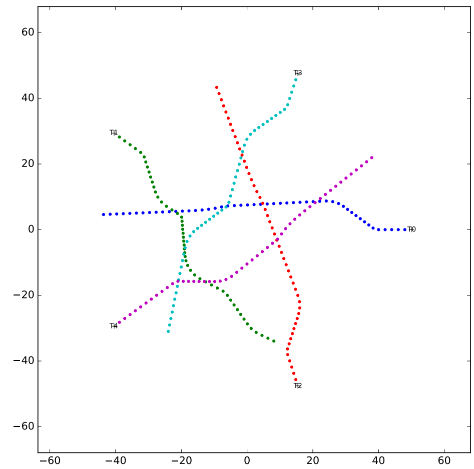
(a) First scenario



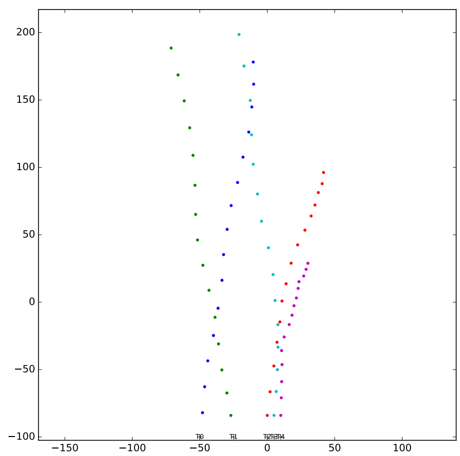
(b) Second scenario



(c) Third scenario



(d) Fourth scenario



(e) Fifth scenario

other, which allows for tracking of maneuvering targets in close proximity to each other. These scenarios where sampled at 1 Hz which is a little faster than a normal high speed vessel radar at 0.8 Hz (48 rpm). The fifth scenario is generated as a part of this project and is composed by linear parallel paths with white Gaussian system noise as maneuvering. This data set where sampled at 0.5 Hz which is a little higher that a normal maritime costal radar at 0.4 Hz (24 rpm).

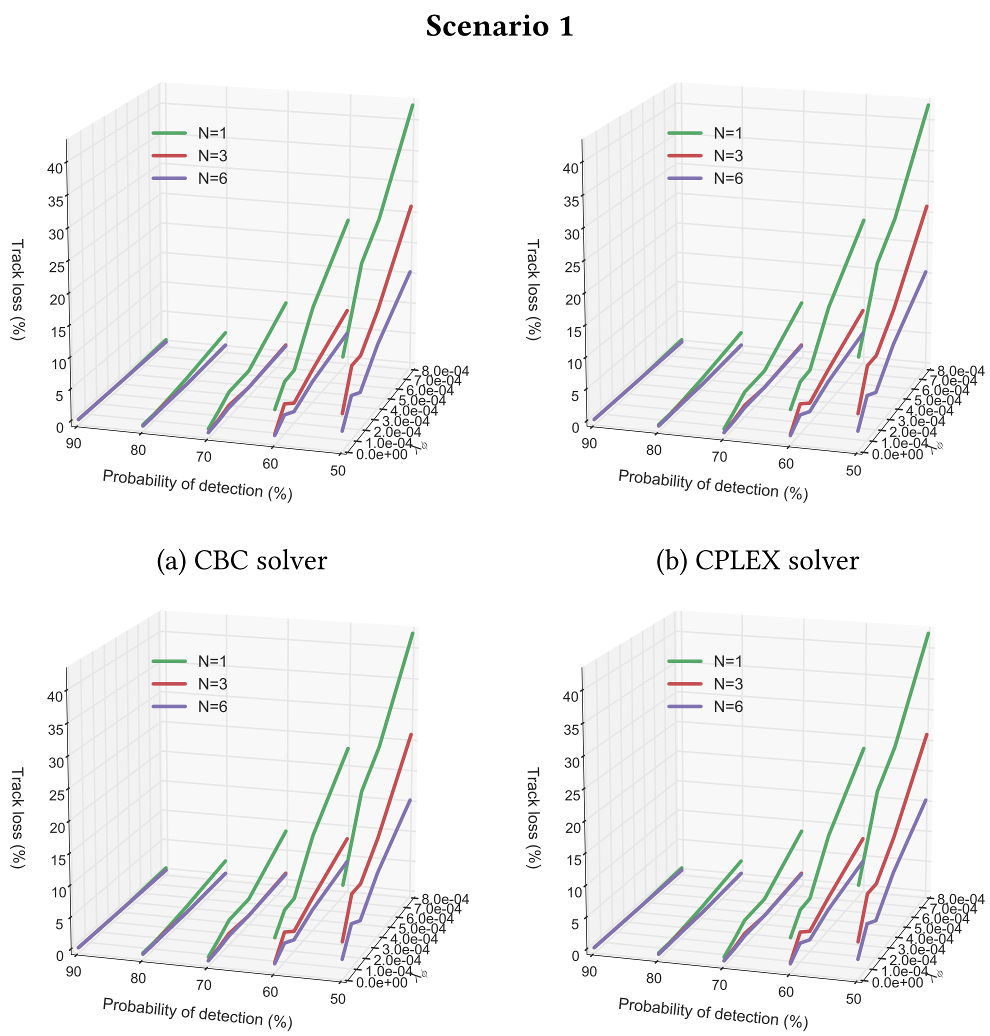
5.3 Simulations

Scenario one through four was simulated with the following variations with all four solvers.

$$\begin{aligned}
\mathbf{P_D} &= \begin{bmatrix} 0.5 & 0.6 & 0.7 & 0.8 & 0.9 \end{bmatrix} \\
\mathbf{N} &= \begin{bmatrix} 1 & 3 & 6 \end{bmatrix} \\
\boldsymbol{\lambda}_{\phi} &= \begin{bmatrix} 0 & 1 \cdot 10^{-4} & 2 \cdot 10^{-4} & 4 \cdot 10^{-4} & 8 \cdot 10^{-4} \end{bmatrix}
\end{aligned}$$

Each variant was simulated 100 times with different seeded random clutter measurements and miss detections. For each simulation, the estimated tracks where compared with the true tacks and categorised in successful and lost tracks, the track loss threshold ϵ_p was 4 meters. Figure 6 - 9 show the averaged track loss percentage.

5.3.1 Tracking performance



(a) CBC solver

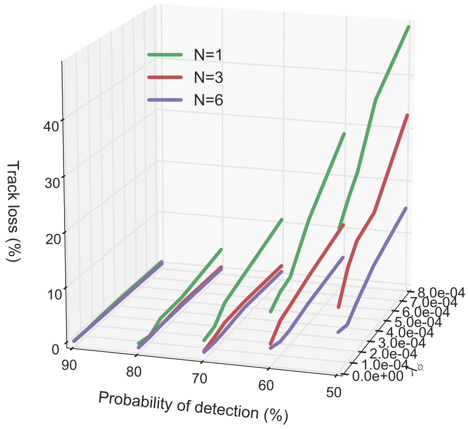
(b) CPLEX solver

(c) GLPK solver

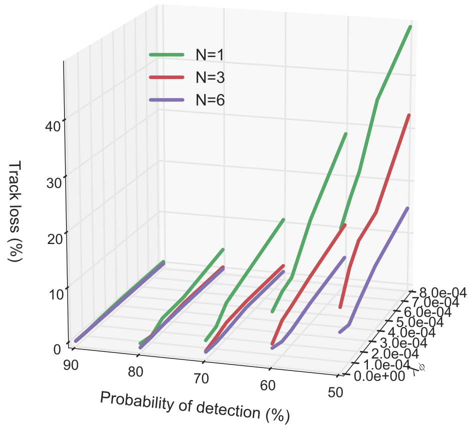
(d) GUROBI solver

Figure 6: Simulation results for all solvers in scenario 1

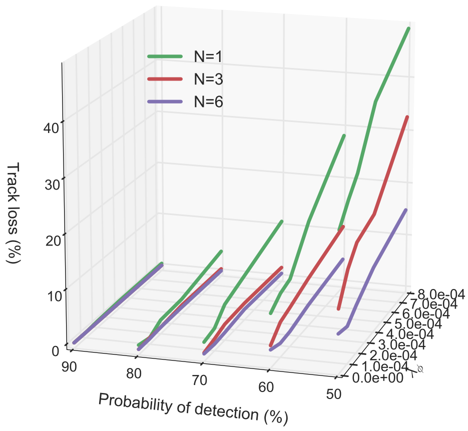
Scenario 2



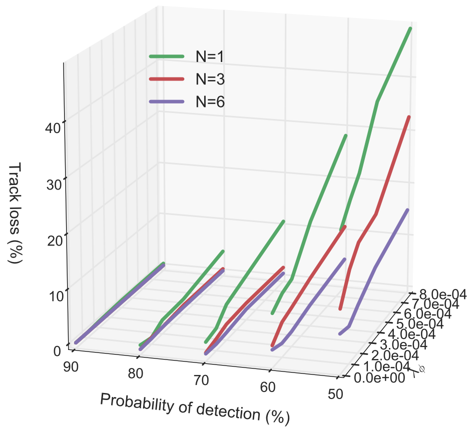
(a) CBC solver



(b) CPLEX solver



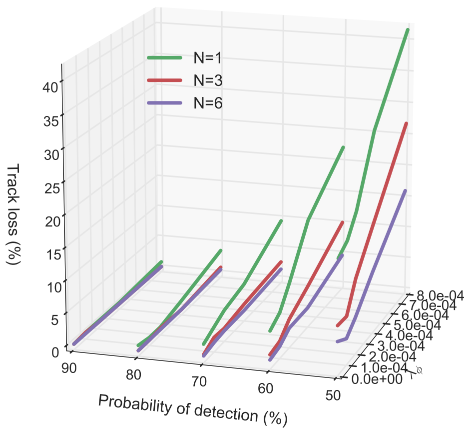
(c) GLPK solver



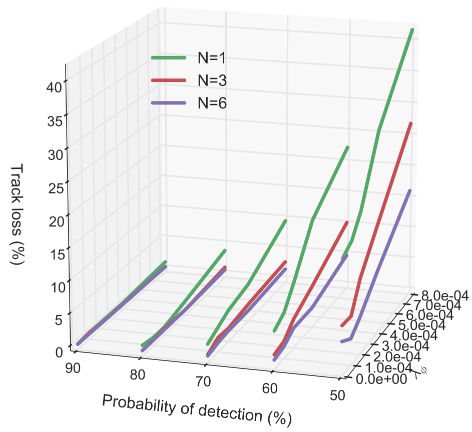
(d) GUROBI solver

Figure 7: Simulation results for all solvers in scenario 2

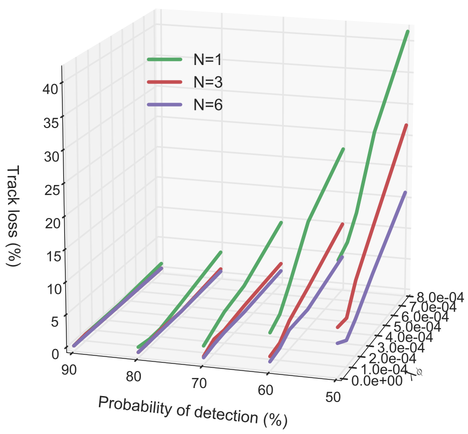
Scenario3



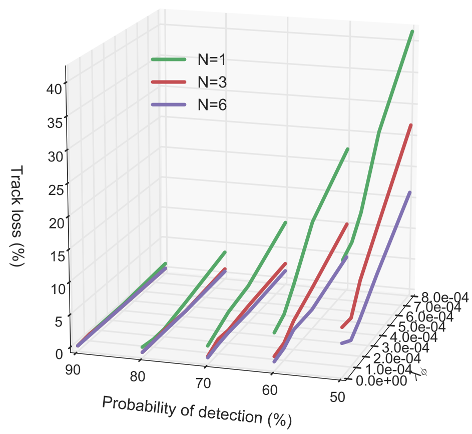
(a) CBC solver



(b) CPLEX solver



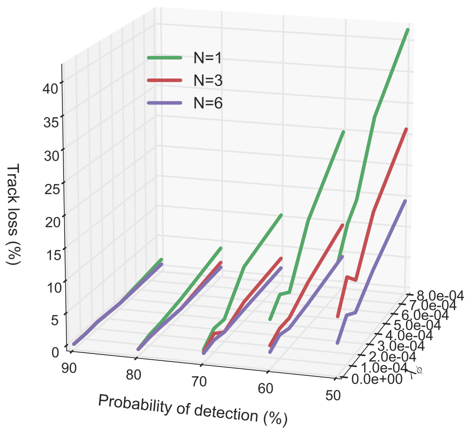
(c) GLPK solver



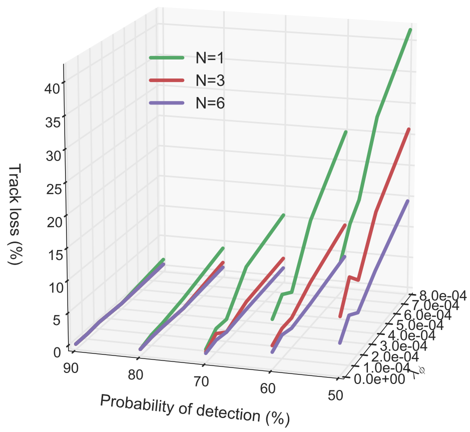
(d) GUROBI solver

Figure 8: Simulation results for all solvers in scenario 3

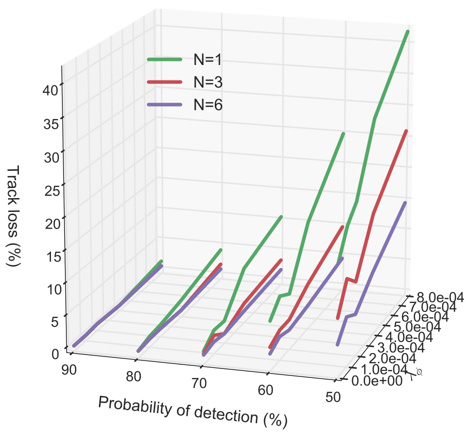
Scenario 4



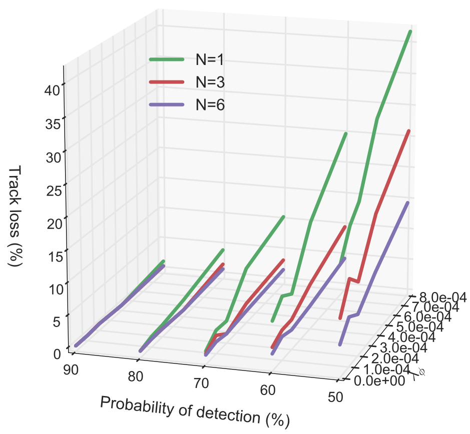
(a) CBC solver



(b) CPLEX solver



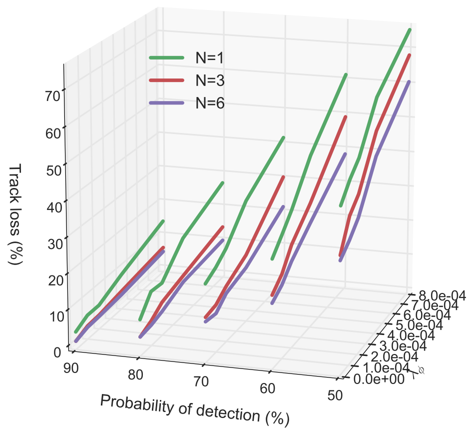
(c) GLPK solver



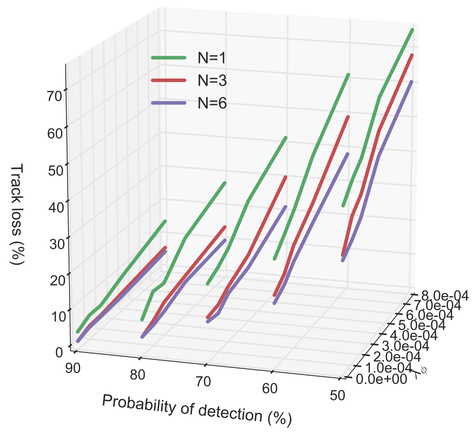
(d) GUROBI solver

Figure 9: Simulation results for all solvers in scenario 4

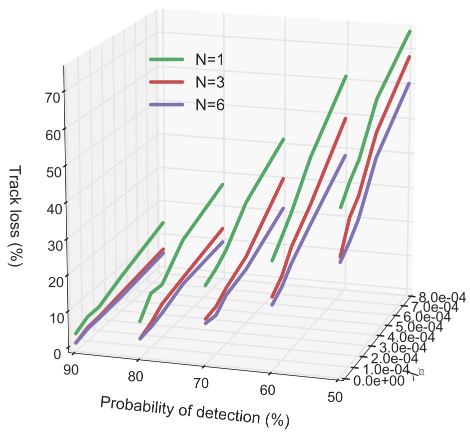
Scenario 5



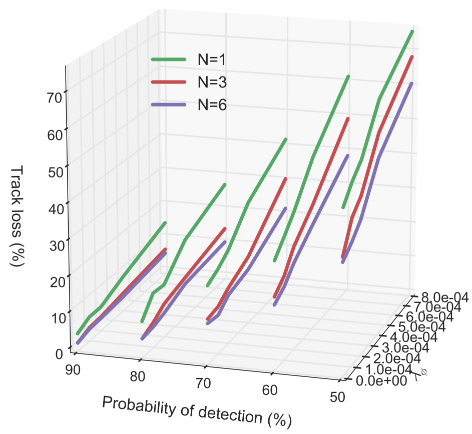
(a) CBC solver



(b) CPLEX solver



(c) GLPK solver



(d) GUROBI solver

Figure 10: Simulation results for all solvers in scenario 5

From the simulations it can be observed that the different solvers perform practically identically when it comes to track performance. The number of lost tracks is proportional with the clutter level at a rate which is inverse proportional with the probability of detection P_D. An interesting observation is the return on investment regarding the number of scans to evaluate (N-scan). For instance, with $P_D = 0.7$ the difference between $N = 3$ and $N = 6$ is marginal, at least when the clutter level is within reasonable levels. With $P_D = 0.5$ we see that the pay-off is much higher for the extra computational cost with 15% improvement between $N = 3$ and $N = 6$.

5.4 Runtime

Figure 11 to 15 displays average the runtime for the entire scenario for the different solvers. From these it can be seen that the GLKP solver was the fastest one in general,

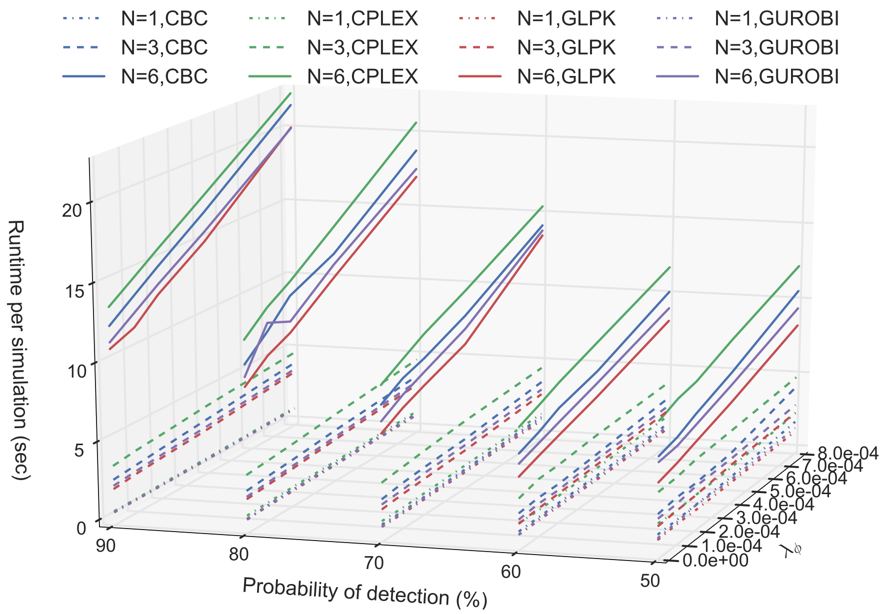


Figure 11: Scenario 1

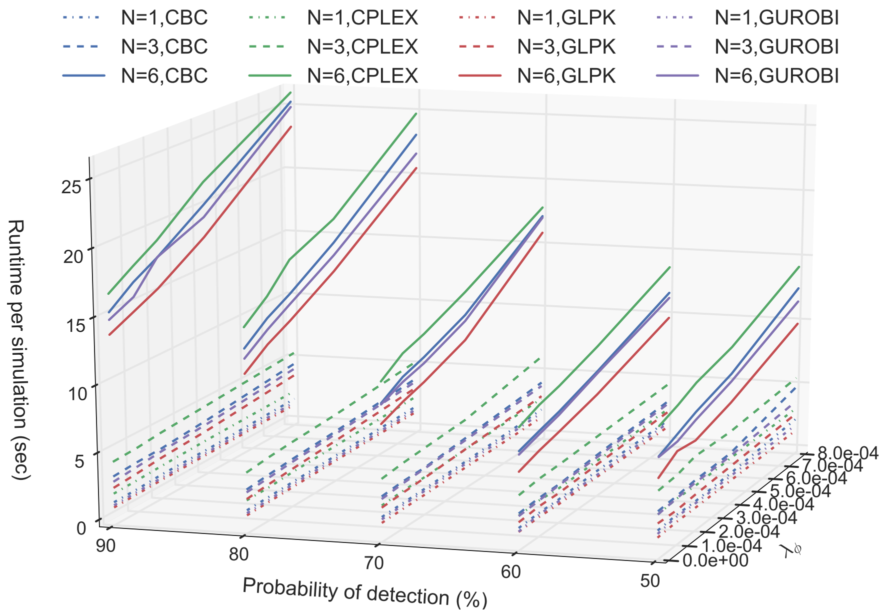


Figure 12: Scenario 2

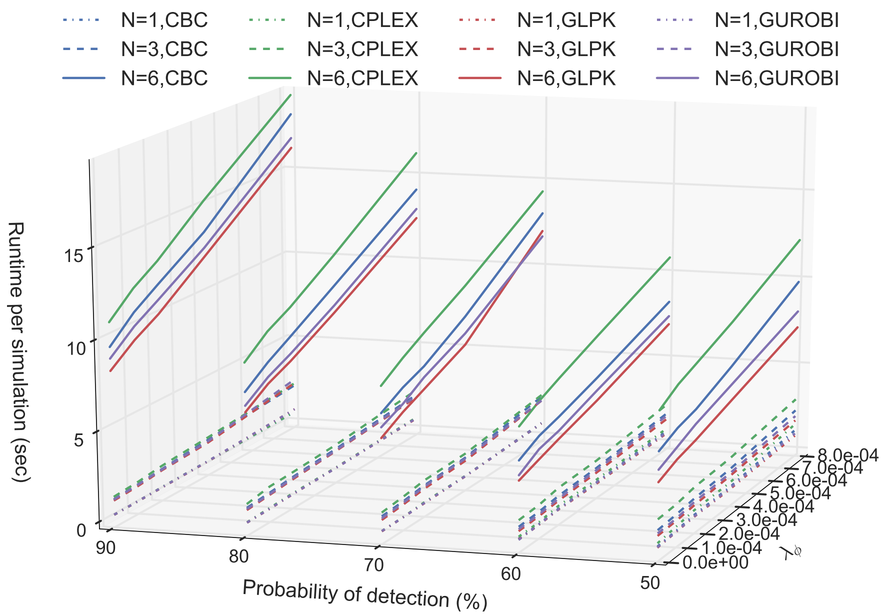


Figure 13: Scenario 3

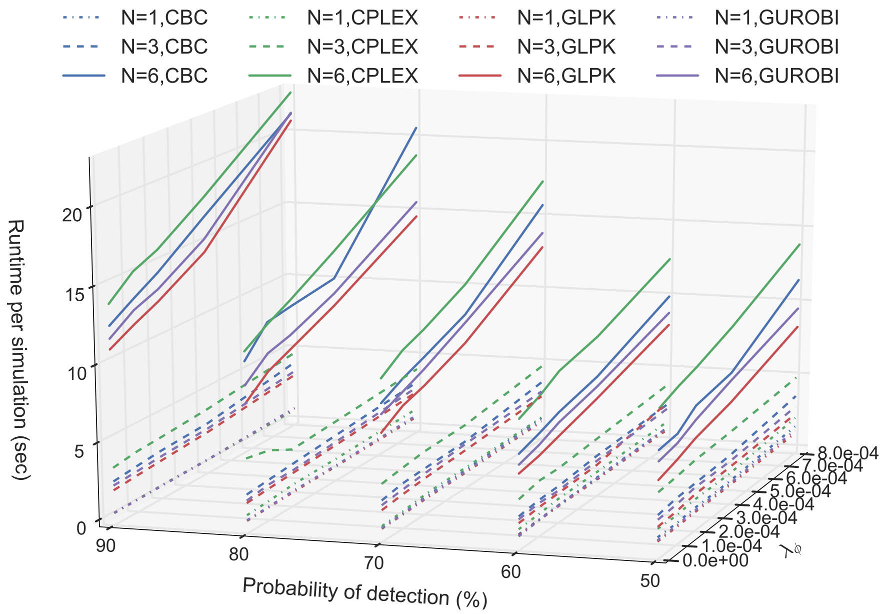


Figure 14: Scenario 4

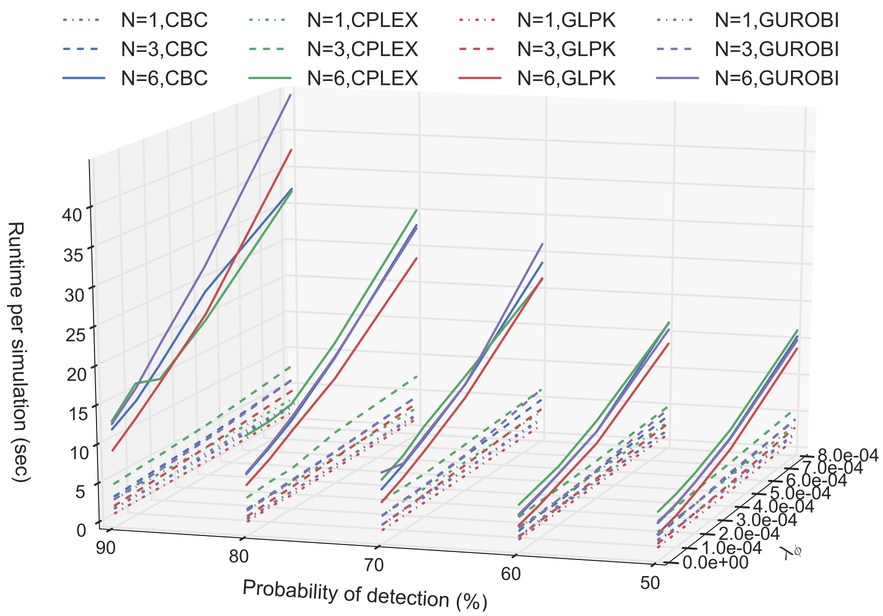


Figure 15: Runtime for scenario 5

6 Discussion

7 Conclusion

References

[1] Y. Bar-Shalom and T. E. Fortmann, "Algorithms for Tracking A Single Target In Clutter," in *Tracking and Data Association*, pp. 119–185, New York: Academic, 1998.

[2] T. Fortmann, Y. Bar-Shalom, and M. Scheffe, "Sonar tracking of multiple targets using joint probabilistic data association," *IEEE Journal of Oceanic Engineering*, vol. 8, no. 3, pp. 173–184, 1983.

[3] H. A. P. Blom and E. Bloem, "Probabilistic Data Association Avoiding Track Coalescence," in *IEEE Transactions on Automatic Control (2000)*, vol. 45, pp. 247–259, 200.

[4] QinetiQ, "Efficient hypothesis management," 2003.

[5] P. Horridge and S. Maskell, "Real-time tracking of hundreds of targets with efficient exact JPDAF implementation," in *2006 9th International Conference on Information Fusion, FUSION*, 2006.

[6] D. B. Reid, "An algorithm for tracking multiple targets," *1978 IEEE Conference on Decision and Control including the 17th Symposium on Adaptive Processes*, vol. 17, no. 6, pp. 843–854, 1978.

[7] T. Kurien, "Issues in the design of practical multitarget tracking algorithms," in *Multitarget-Multisensor Tracking: Application and Advances*, ch. 3, pp. 219–245, Artech House, first ed., 1990.

[8] Y. Bar-Shalom, S. S. Blackman, and R. J. Fitzgerald, "Dimensionless score function for multiple hypothesis tracking," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 43, no. 1, pp. 392–400, 2007.

[9] P. Storms and F. Spieksma, "An LP-based algorithm for the data association problem in multitarget tracking," *Computers and Operations Research*, vol. 30, no. 7, pp. 1067–1085, 2003.

[10] S. Coraluppi, C. Carthel, M. Luettgen, and S. Lynch, "All-Source Track and Identity Fusion," no. June, 2000.

[11] E. Brekke, O. Hallingstad, and J. Glattetre, "Improved target tracking in the presence of wakes," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 48, no. 2, pp. 1005–1017, 2012.