A Polynomial Time Algorithm for the Multidimensional Assignment Problem in Multiple Sensor Environments

Agostino Capponi *
Thales Naval Nederland B.V.
The Netherlands

H.W. de Waard †
Thales Naval Nederland B.V.
The Netherlands

Abstract – *This is the first half of a two-part paper devoted to the multidimensional assignment problem (MDA) in multiple target tracking. Here a new polynomial-time algorithm for solving the MDA problem is proposed. The dimension of the formulated assignment problem is given by two parameters: the number of available sensors and the last time instant when association decisions are revokable. This algorithm is then used in the companion paper [4] where a decomposition approach aiming at reducing the number of candidate associations and then creating independent MDA subproblems is proposed. The algorithm proposed here is applied inside each independent MDA subproblem.*

Keywords: Data association problem, clustering, tracking, filtering.

1 Formulation of the MDA problem

The objective of multiple target tracking is to partition the data received from sensors into sets of measurements, or track hypotheses, produced by the same target [1]. The central problem in the application of multiple target tracking is to associate measurements with the appropriate track hypotheses. Each measurement can be associated to an existing track hypothesis, a new track hypothesis or a false alarm. Different formulations of the data association problem have been proposed in the literature. In this paper only the multidimensional assignment formulation is addressed. Assume that S sensors scan the surveillance region. Every sensor stores the produced measurements in a shared data space. The stored measurements can be retrieved by means of a query. At fixed timestamps $t_1, t_2, \ldots, t_k$ the sets of measurements are retrieved. More particularly at timestamp t_i the measurements produced by all sensors in the interval $[t_{i-1}, t_i)$ are collected. Figure 1 illustrates the mechanism of storing and retrieving measurements. There are S sensors that produce measurements and send them to the shared data space where they are stored. The module Retrieve Sensor Measurements retrieves the measurements from the shared data space.

We denote by $Z_j(i)$ the set of measurements retrieved by sensor j at timestamp t_i, formally $Z_j(i) = \{z_k^{(i,j)}\}$ where t_k is the actual measurement time, i and j denote

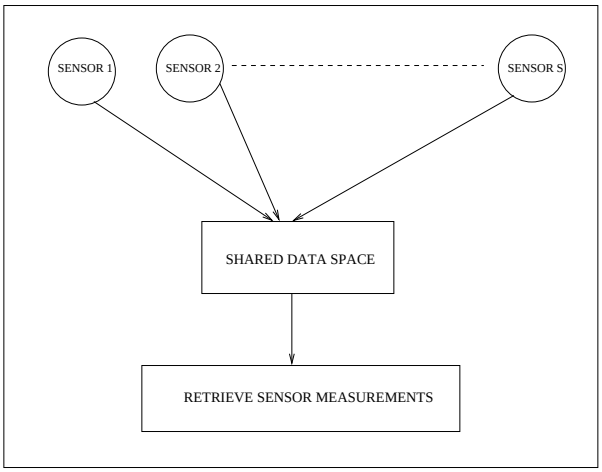

Fig. 1: Mechanism of store/retrieval of sensor measurements.

respectively the timestamp when the measurement is retrieved and the number of the sensor that has produced the measurement, $k = 1, 2, \ldots |M^{(i,j)}|$ and $M^{(i,j)}$ is the set of all measurements retrieved from sensor j at timestamp t_i. The set of measurements collected at timestamp t_i from all available S sensors is denoted by $Z(i) = \{Z_1(i), Z_2(i), \ldots, Z_S(i)\}$. We call such a set a *complete* frame.

Note that a measurement in the frame $Z_j(i)$ may emanate from an existing target or be a false alarm. The cumulative set of all measurements collected from all S sensors until the timestamp t_N is denoted by Z^N and mathematically expressed as $Z^N = \{Z(1), Z(2), \ldots, Z(N)\}$. A *track hypothesis* is defined as a set of measurements such that for every complete frame at most one measurement comes from a certain sensor. Formally if z is a track hypothesis, then $|z \cap Z_j(k)| \leq 1$ for each j and k. The process of partitioning sensor measurements into track hypotheses can be seen as a partitioning problem over the cumulative data set Z^N. A partition γ of Z^N is defined by:

- $\gamma = \{z_1, \ldots, z_k : z_i \neq \emptyset \text{ for each } i\}$,
- $z_i \cap z_j = \emptyset$ for each $i \neq j$,

*A.Capponi. Email: agostino.capponi@nl.thalesgroup.com
†H.W. de Waard. Email: huub.dewaard@nl.thalesgroup.com

- $|z_i \cap Z_j(k)| \leq 1$ for each j and k
- $Z^N = \bigcup_{j=1}^{k} z_i$

Here, each element z_i is a track hypothesis and k denotes the number of track hypotheses in the partition γ. The second constraint expresses that two track hypotheses in the partition do not share measurements. The third constraint states that each track hypothesis can contain at most one measurement in the data set retrieved from a sensor at a certain timestamp. Finally the fourth constraint states that all measurements must be covered by the track hypotheses in γ. The goal in data association is to find the *best* partition of Z^N. Informally the best partition is the one most likely to correspond with the real world situation. In order to define it formally a cost-coefficient is needed for each track hypothesis $z \in Z^N$. An approach described in [8] is used where a quality measure for a track hypothesis $z \in Z^N$ is defined in the form of a *likelihood function* $Q(z)$. The objective of this function is to assign a score of how good each measurement $z_k^{(i,j)} \in z$ fits the target's assumed measurement model. The measurements are noise contaminated. The deviation between the predicted position in the measurement space and the measurement itself is assumed to follow the normal distribution. The likelihood of a partition of Z^N is computed by the product of the individual track likelihoods. If we now associate each track hypothesis z with the weight $w(z) = \log(Q(z))$ and define $W(\gamma) = \sum_{i=1}^{t} w(z_i)$, the data association problem can be formulated as a *maximum weighted set partitioning problem*, where the goal is to find the partition γ^* such that:

$$W(\gamma^*) = \max_{\gamma \in \Gamma} W(\gamma) \tag{1}$$

where Γ denotes the set of all partitions of Z^N. We call γ^* an *optimal partition*.

Ideally we would like to solve the DAP every time a complete frame of measurements is collected. The problem as formulated above includes measurements collected from all available S sensors during all timestamps. Such problem has been shown to be NP-hard [6] when the number S of sensors used is ≥ 3. In order to reduce the complexity of the problem, a deferred logic approach is used in which only the last complete frames of measurements take part in the problem solving process. Recall that a complete frame has been defined above as the set of measurements collected from all available sensors at a fixed timestamp.

A technique for considering only the last W complete frames is the W-sliding window approach introduced by Poore [8, 9]. The main idea is to consider only the measurements of the W complete frames that lie within the window as well as earlier established track hypotheses. If a new complete frame $Z(N)$ of measurements is received, the window slides one complete frame ahead so it again embraces the last W complete frames.

This implies that all decisions concerning measurements collected before the timestamp t_{N-W} (outside the window) become fixed by the last solution fulfilling eq.1. For

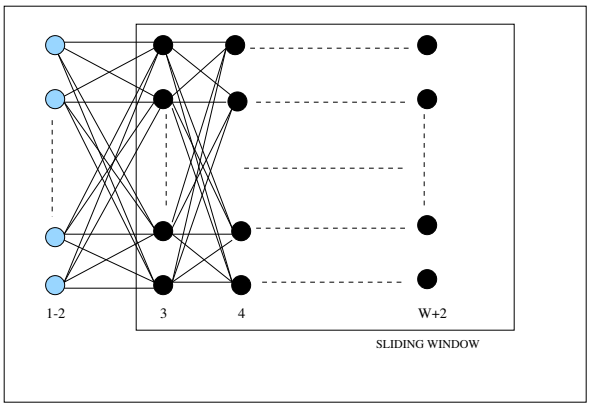

Fig. 2: Data association posed as a multidimensional assignment problem

$N = 1, 2, \ldots, W+1$ the problem is solved using a data association algorithm. When the $(W+2)^{th}$ complete frame of measurements is received the association decisions between measurements of the first two complete frames become fixed (they correspond to the circles in fig. 2 colored in blue). All the track hypotheses containing measurements outside the sliding window are removed from consideration unless the measurements falling outside the window have been associated by the last solution fulfilling eq. 1.

Mathematically speaking denote by $Z_{W+1}(0)$ the set of all the track hypotheses that are part of the last solution computed and cut at the beginning of the window. This is the set of all established track hypotheses. After that $W+2$ complete frames have been received the data association of the measurements in the first three complete frames remains fixed and so on. Formally, at the N-th timestamp, we define

$$Z^*(N) = Z_N(0) \cup Z(N-W+1) \cup \cdots \cup Z(N)$$

and denote by Γ^* the set of all partitions of $Z^*(N)$. The goal is now to find the partition $\gamma^* \in \Gamma^*$ with the highest sum of track weights.

It is important to note that a W-sliding window implies the solution of a $(W \times S)+1$-dimensional assignment problem. This is due to the fact that there are W complete frames, each consisting itself of S frames of measurements (one frame for each sensor) plus one additional frame consisting of established track hypotheses.

2 The Semi-Greedy Track Selection algorithm

The goal of this section is to illustrate the polynomial time algorithm Semi-Greedy Track Selection (SGTS). This algorithm solves the multidimensional assignment problem as formulated in the previous section. Subsection 2.1 describes the algorithm and Subsection 2.2 determines the complexity order estimation of the proposed algorithm.

2.1 Description and Analysis of SGTS

The SGTS is a semigreedy algorithm that first generates a number of different partitions of sensor measurements into

track hypotheses. Subsequently it selects the partition with the highest sum of track weights. It receives in input a set of weighted track hypotheses, denoted by Z_*. Recall that the weight of a track hypothesis is the logarithm of the likelihood of the track hypothesis as described in Section 1. The output is the partition among the ones generated with the highest sum of track weights. SGTS runs as follows. First it sorts all track hypotheses in Z_* by decreasing weight. Then it generates the partitions. Starting from an *initial* track hypothesis, every partition is constructed by repeatedly including the track hypothesis with the highest weight that does not share measurements with any of the previously selected track hypotheses. In order to guarantee that all generated partitions are different, for every partition an *initial* track hypothesis that has not yet been included in any of the previously generated partitions is chosen. This is accomplished according to the following criteria:

- First Partition. The *initial* track hypothesis is the track hypothesis with the highest weight.
- i-th Partition. The *initial* track hypothesis is the track hypothesis with the highest weight among the ones that have not yet been included in any of the previously $i-1$ generated partitions.

The algorithm stops when no other partition can be generated because each input track hypothesis has already been included in some previously generated partition. Of course it is possible to fix a maximum number of partitions that SGTS can generate. This results in a high flexibility because the number of partitions that are generated before selecting the one with the highest sum of track weights can be adapted to the available computation time. The partition with the highest sum of track weights is the one returned by SGTS and is used to freeze the assignments of measurements outside the window (see previous section). An example of a run of the SGTS algorithm is shown in figure 3.

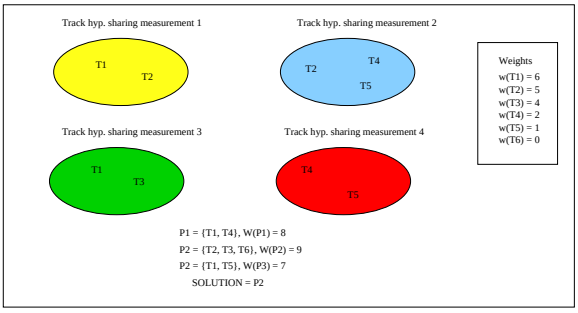

Fig. 3: Example of a run of SGTS.

In figure 3 there are six track hypotheses and four measurements forming the tracks. The first partition P1 consists of track hypotheses T1 and T4. The track hypothesis T1 is the first one selected because it has the highest weight. The track hypothesis T4 is selected next because it is the one with the highest score after track hypotheses T2 and T3 have been eliminated due to the reason that they share measurements with T1. The selection of track hypothesis T4 implies the deletion of the track hypotheses T5 and T6 since they both share measurement 4. So the first partition is complete since there is no other track selectable. The second partition P2 contains T2 as an initial track hypothesis and this is correct since T2 has not been selected in the previous partition P1. Subsequently track hypotheses T1, T4 and T5 are removed because they share measurements with T2. The next selected track hypothesis is T3 which does not share any measurement with the last remaining track hypothesis T6. The selection of the track hypothesis T6 completes the second partition P2. In order to generate P3 SGTS starts selecting T5 as an initial track hypothesis because T1, T2, T3 and T4 have already been selected in the previous two partitions. After selecting T5, the track hypotheses T4 and T2 and T6 are removed because they share measurements with T5. Hence the next track hypothesis selected is T1 because it is the one with the highest weight within the remaining track hypotheses. Since T1 and T3 share measurement 3 the track hypothesis T3 is deleted and no other track hypothesis can be further selected for partition P3. At this point each track hypothesis has been selected in some of the first three partitions, hence SGTS stops because it does not find an initial track hypothesis to start a new partition and outputs partition P2 which is the one with the highest weight.

It is important to note that SGTS generates a number of different partitions. Based upon some apriori additional information an operator might decide to select a partition that is not necessarily the one with the highest sum of track weights. Another advantage of SGTS is the mathematical guarantee on the quality of the solution returned. More particularly it will be shown that the value of the first solution generated by SGTS approximates the value of the optimal solution within a guaranteed factor depending on the dimension $d = W \times S + 1$ of the assignment problem. The proof given here is based upon a result given in [7] concerning the approximation of the weighted set packing problem. As follows, the steps followed by the algorithm to generate the first solution are described in pseudocode. The module of SGTS that generates the first solution is called SGTS1. In order to simplify the understanding of the following pseudocode, the main variables used are described:

- x_i is the track hypothesis selected as part of the first solution at the i-th iteration.
- Z_i contains the track hypotheses that are eliminated at the i-th iteration because they have measurements in common with the selected track hypothesis x_i.
- T_i contains the track hypotheses that can still be part of the first solution because they do not intersect any of the track hypotheses $x_1, x_2, \ldots, x_{i-1}, x_i$ selected in the first i iterations of SGTS1.

ALGORITHM SGTS1(Z_*)

$i \leftarrow 0$
$T_i \leftarrow Z_*$
repeat
  $i \leftarrow i + 1$
  $x_i \leftarrow z \in T_{i-1}$ that maximizes $w(z)$
  $Z_i \leftarrow \{z \in T_{i-1} : x_i \cap z \neq \emptyset\}$
  $T_i \leftarrow T_{i-1} \setminus Z_i$
until $T_i = \emptyset$
return$\{x_1, x_2, \ldots, x_i\}$

Assume OPT to be an optimal partition. At each iteration i of the loop *repeat* we define OPT_i as the set of track hypotheses that are part of the optimal solution and have measurements in common with the sets in Z_i. Mathematically this is expressed as $OPT_i = OPT \cap Z_i$. Furthermore denote by S_1 the solution returned by SGTS1.

Lemma 1: *The number of track hypotheses contained in the set OPT_i is not greater than the number of measurements contained in the track hypothesis x_i, i.e $|OPT_i| \leq |x_i|$.*

Proof. Firstly note that the track hypotheses in OPT_i are pairwise disjoint because they are also contained in the optimal partition, being $OPT_i = OPT \cap Z_i$. Secondly note that each track hypothesis contained in OPT_i is also contained in Z_i by definition of OPT_i. Since all track hypotheses contained in Z_i have measurements in common with x_i by definition of Z_i, each track hypothesis in OPT_i has at least one measurement in common with the track hypothesis x_i. Figure 4 illustrates the extreme situation that can occur, i.e each track hypothesis in OPT_i shares only one measurement with x_i.

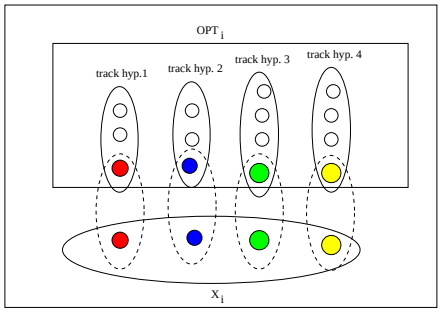

Fig. 4: Number of track hypotheses in OPT_i = number of measurements in x_i

The conclusion is that $|OPT_i| \leq |x_i|$. □

Proposition 2: *For a fixed d-dimensional assignment problem the value of the solution computed by SGTS1 approximates the value of OPT within a factor of d,* i.e.

$$W(OPT) \leq d \times W(S_1)$$

Proof. Due to the second instruction of the loop *repeat* $w(z) \leq w(x_i)$ for any track hypothesis $z \in T_{i-1}$. Therefore:

$$W(OPT_i) = \sum_{z \in OPT_i} w(z) \leq w(x_i)|OPT_i|$$

The lemma 1 allows to claim that $W(OPT_i) \leq w(x_i)|x_i|$. Since we are solving a d-dimensional assignment problem, the cardinality of each track hypothesis will never be greater than d, thus $W(OPT_i) \leq d \times w(x_i)$. It follows that

$$W(OPT) = \sum_i W(OPT_i) \leq d \sum_i w(x_i) = d \times W(S_1).$$

□

A guaranteed approximation factor has been provided for the first solution generated by the algorithm. This means that the solution returned by SGTS which not necessarily coincides with the first solution generated can give an approximation factor equal to or smaller than d. This is better illustrated in fig. 5

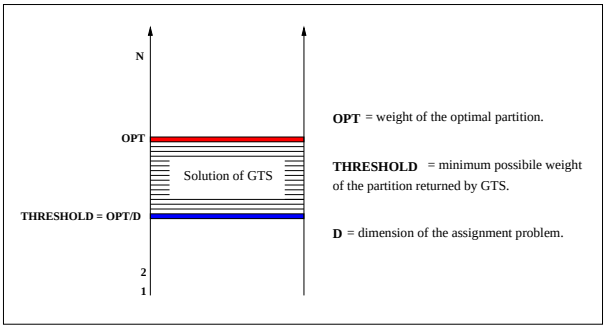

Fig. 5: Guaranteed approximation factor of SGTS

A more detailed analysis of the SGTS algorithm as well as experimental results are presented in [2].

2.2 Complexity Order Estimation of SGTS

This subsection gives a detailed analysis of the complexity of the SGTS algorithm. This analysis is decomposed in two major parts: the number of operations needed to sort the track hypotheses and the number of operations needed to generate the different solutions. The following parameters are used for the analysis:

- $t = |Z_*|$ is the cardinality of the input track hypothesis set of the algorithm.
- n is the cardinality of the set of measurements that are contained in the track hypotheses of Z_*.
- $d = (W \times S) + 1$ is the dimension of the assignment problem, where W is the number of complete frames that are taken into account and S the number of sensors that have produced the measurements.

Let $\{m_1, m_2, \ldots, m_n\}$ be an arbitrary ordering of the measurements contained in the track hypotheses of Z_*. In order to simplify the notation we use j to indicate m_j, i.e the j-th measurement in the ordering. An array M of size n containing positive integer values is used to check whether or not a certain measurement belongs to some other track hypothesis already selected in the solution that is being computed. Initially for any $1 \leq j \leq n$ the value of $M[j]$ is set to 0. Every time a track hypothesis z is selected as a part of a solution, $M[j]$ increases by one for every measurement j contained in the track hypothesis z. Since all measurements must be covered when a solution has been computed, we have that $M[j] = i$ for any $1 \leq j \leq n$ after computing the i-th solution and before starting the computation of the $(i+1)$-th solution.

The $QuickSort$ algorithm is used to sort the track hypotheses by decreasing weight. This means that the number of operations needed by the sorting process is $O(t \log t)$.

It remains to calculate the number of operations needed to generate a solution. We start with examining the number of operations needed to select the *initial* track hypothesis. Here we distinguish between the first partition and another partition as follows.

- First partition. It requires a constant number of operations. It corresponds with the first track hypothesis in the ordering.
- i-th partition ($i > 1$). It requires a constant number of operations if every time that SGTS generates a partition a pointer to the first track hypothesis in the ordering that has not yet been included in any of the previously generated partitions is saved.

All the other track hypotheses (second, third, etc...) are selected as a part of the i-th partition ($i \geq 1$) as follows. For any scanned track hypothesis z in the ordering SGTS checks that every measurement j contained in z is such that $M[j] = i - 1$, i.e it has not yet been covered by any track hypothesis of the i-th solution. If this is the case, then it selects the track hypothesis as a part of the solution and increments by one the value $M[j]$ for any j in z. If not, then it scans the next track hypothesis in the ordering. This is repeated until all track hypotheses have been scanned. Note that every track hypothesis in the ordering can contain at most d measurements since we are solving a d-dimensional assignment problem. Thus it can be safely concluded that the number of operations needed to generate the different partitions is:

- First partition. It takes $O(t \log t + td)$. The first term represents the number of operations required by the sorting algorithm and the second term indicates the number of operations needed to check if every track hypothesis can be included in the partition (using the information contained in the array M).
- i-th partition ($i > 1$). It requires $O(td)$. Here the complexity order estimation is given only by the number of operations needed to check for every track hypothesis if it can be included in the partition that is being generated since the sorting algorithm is executed only once at the beginning.

The complexity order estimation $T(K)$ for the generation of the first K partitions is given by the sum of the single terms $T(P_i)$ representing the complexity to generate each partition P_i, $1 \leq i \leq K$. Formally

$$T(K) = \sum_{i=1}^{K} T(P_i)$$

Expressing $T(K)$ as a function of K and t we get $T(K) = O(Kt \log t + Kt)$. The number of generated partitions can be assumed constant, thus the complexity order estimation of $SGTS$ is $O(T \log T)$. Experimental results for the single sensor case [2] show that generally after generating twenty partitions there is not any significant improvement in the quality of the solution.

3 GreedySet vs Clusterized GreedySet

The multidimensional assignment problem (MDA) basically consists of solving two subproblems. The first subproblem is the track hypothesis formation using the complete frames within the window and the earlier established track hypotheses. The second one is the solution of the assignment problem using the track hypotheses obtained after the completion of the first stage. The process of track hypothesis formation itself can be further decomposed in other two phases: the correlation tests and the track hypothesis likelihood calculations. The aim of the correlation tests is to prevent implausible track hypotheses to be formed. The correlation test is achieved by application of the *gating* test for each track hypothesis-measurement pair [10]. The criterion for this goodness of fit test is defined by a *gate* or *validation* region. If the correlation test is verified, then a likelihood must be assigned to the track hypothesis and this involves the estimation of the track hypothesis state vector.

Although finding a solution for the MDA problem is the most important of the two subproblems, it must be noted that the amount of CPU time taken by the track hypothesis formation process is much higher [3]. This motivates the study of methods that try to reduce the amount of time needed for correlation tests and likelihood assignments to track hypotheses. A new approach has been proposed in the companion paper [4] to address this issue. The proposed approach creates clusters containing disjoint sets of track hypotheses using the available correlation ellipsoids of the track hypotheses. Every measurement in the complete frame received at a particular timestamp is then assigned to a cluster and correlation tests are carried out only for track hypotheses in the cluster. Finally a number of smaller independent MDA subproblems is determined using the information about the past hystory of the track hypothesis. For a detailed explanation of the approach the reader is referred to [4].

4 Experiments

4.1 Tested features

The goal of the experiments conducted is to test the advantages of the mean child approach described in the companion paper [4] with respect to the standard approach that checks for all possible track measurement correlations. Before proceeding with the description of the experiments we briefly remind the approach proposed in [4]. It considers trees consisting of all tracks originated from a fixed measurement falling within the window or from an established track obtained from the previous solution. For each of these trees a mean track, which is intuitively a track representing the mean dynamical behaviour of all tracks in the tree, is calculated using the predicted state and covariance matrices of all tracks in the tree. The generated mean tracks are then used to produce clusters, which consist of mean tracks whose associated uncertainty ellipsoids intersect. Whenever a measurement is received, it is first checked if it falls within the cluster size of the some existing clusters. If so, then a mean track measurement correlation test is performed for every mean track in the cluster and in the case when the test is successful, the measurement is correlated with all tracks in the tree. If not, a new cluster whose only element is the the track consisting of this single measurement is created. Finally independent MDA subproblems are derived and each subproblem is solved applying the SGTS algorithm [2].

Some parameters have been introduced in order to measure the relevant features of the mean child approach and the resulting tracking accuracy.

Before describing in detail these parameters, some notation and terminology is introduced. Let k be the number of measurements in the current frame and t the number of trees formed so far. Given a measurement m and a track z we write $m \rightarrow z$ if the measurement m correlates with the track z and $m \xrightarrow{not} z$ if it does not. There are now all ingredients to introduce the test parameters used. They are:

- **Density of correlations**. Let S be the set of mean track measurement pairs such that the measurement correlates with the mean track of the tree. Formally we define the set S as follows:

  $$S = \{(m_i, MT_j) : m_i \rightarrow MT_j, 1 \leq i \leq k, 1 \leq j \leq t\}$$

  For any pair $(m_i, MT_j) \in S$ denote by $c_{i,j}$ the number of tracks in the tree T_j which correlate with the measurement m_i (note that MT_j is the mean track of T_j) and by t_j the number of tracks in the tree T_j.

  The parameter μ_{WD} is defined as:

  $$\mu_{WD} = \min\{\tfrac{c_{j,i}}{t_j} : (m_i, MT_j,) \in S\}$$

  This parameter says which is the minimum possible number of tracks in a tree which correlate with the considered measurement, where the minimum is calculated over all mean track measurement pairs such that the correlation test is fulfilled.

  The parameter μ_{BD} is defined as

  $$\mu_{BD} = \max\{\tfrac{c_{j,i}}{t_j} : (m_i, MT_j) \in S\}$$

  This parameter says which is the maximum possible number of tracks in a tree which correlate with the considered measurement, where the maximum is calculated over all mean track measurement pairs such that the correlation test is fulfilled.

  Ideally we would like both these two parameters to be one. This would mean that whenever a measurement correlates with the mean track of a tree, then it also correlates with all the tracks of the tree. This would imply that in the case when a mean track measurement correlation test is fulfilled we can immediately correlate the measurement with all the tracks in the tree without performing the additional measurement track correlation test for all tracks in the tree. Therefore our proposal to use only mean child to make correlations would be fully justified.

- **Effectiveness of correlations**. Let S be the set of mean track measurement pairs such that the measurement does not correlate with the mean track of the tree, but it correlates with at least one track in the tree. Formally

  $$S = \{(m_i, MT_j) : m_i \xrightarrow{not} MT_j \text{ and } \exists z \in T_j : m_i \rightarrow z\}$$

  However we want to distinguish between the case when likely track measurement correlations have not been created because of the failure of the preliminary mean track measurement correlation test from the case when only very poor track measurement correlations have been discarded (this can be considered as a further advantage of the mean child approach). In order to do that we associate every pair (MT_j, m_i) in S with a weighted factor $w(MT_j, m_i)$ defined as follows:

  $$w(MT_j, m_i) = \sum_z p_c(z) : z \in T_j \text{ and } m_i \rightarrow z$$

  where $p_c(z)$ denotes the conditional likelihood of the track z in the tree T_j (see [4] for further details). The parameter μ_E is defined as follows:

  $$\mu_E = \begin{cases} \sum_{(MT,m)\in S} \frac{w(MT,m)}{|S|} & \text{if } |S| \neq 0 \\ 0 & \text{if } |S| = 0 \end{cases}$$

  Ideally we would like that every time that the mean track measurement correlation test is not fulfilled, no track in the tree associated with the mean track correlates with the measurement. In other words, we would like the parameter μ_E to be zero.

- **Redundancy of correlations**. Define S as the set of all mean track measurement pairs such that the measurement correlates with the mean track, but it does

not correlate with any track of the associated tree. Formally

$$S = \{(MT_j, m_i) : m_i \rightarrow MT_j \text{ and } \forall z \in T_j, m_i \xrightarrow{not} z\}$$

Mathematically speaking, the parameter μ_R is defined as

$$\mu_R = \frac{|S|}{t*k}.$$

This parameter gives a measure of the roughness of the mean child approach for deciding correlations. If the value of the parameter is 0, it means that every time the measurement correlates with the mean child of a tree, it also correlates with at least one track in the tree.

- **Correlation reduction factor** (*CRF*). The purpose of this parameter is to compare the number of track measurement correlation tests performed using the mean child approach described in [4] with the number of track measurement correlation tests which would be performed without using it. Consider the set S of mean track measurement pairs such that the mean track correlates with the measurement. Formally we define the set

$$S = \{(MT_j, m_i) : m_i \rightarrow MT_j\}$$

Denoting by t_j the number of tracks in the tree T_j and by t_{tot} the total number of tracks, i.e $t_{tot} = \sum_{j=1}^{t} t_j$, we can formally define the parameter CRF as:

$$CRF = \frac{t*k}{t_{tot}*k} = \frac{t}{t_{tot}}$$

All the parameters described in this subsection will be evaluated on a special scenario described in the next subsection.

4.2 Simulation results

The simulated scenario consists of four groups of aircrafts located in a three dimensional space. Each group consists of three closely spaced aircrafts. All aircrafts fly at the same height. In each group the initial distance along the x axis between an aircraft and the closest to it is 500 meters. All aircrafts follow a straight trajectory with a constant velocity of 500 meters per second within the coverage area of the radar sensor. The simulated radar sensor used has a maximum detection range of 100 km. The measurement noise is set to 0.03 km in range and to 0.3 degrees in bearing and elevation. The process noise is set to $50\frac{m}{s^2}$. A complete frame of measurements is collected every 1.5 seconds and correlations between the measurements in the frame and existing tracks are attempted. A false alarm is generated at every sampling time within the neighbourhood of one group of aircrafts. All the aircrafts will final converge to the same final position. The picture 6 illustrates the described scenario.

Twenty significative samples, each corresponding to an instant time when the mean child approach is applied, have been extracted from the collected results. For each of the considered samples the parameters described in the previous subsection are illustrated by means of Matlab graphics.

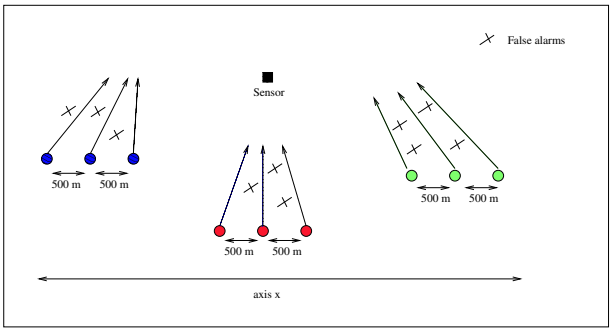

Fig. 6: Simulated Scenario

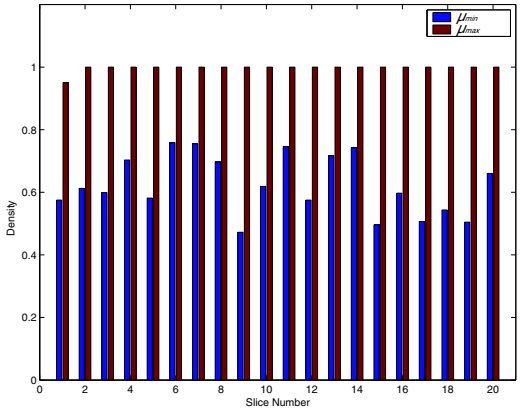

Fig. 7: Density of correlations

The graphic in fig.7 leads us to conclude that for the tested scenarios at least half of the tracks in every tree correlate with the measurement when the preliminary mean track measurement correlation test is fulfilled. Furthermore it turns out that at every sampling time there is at least one tree and one measurement such that the measurement fulfills the preliminary correlation test with the mean track and then correlates with every track of the tree.

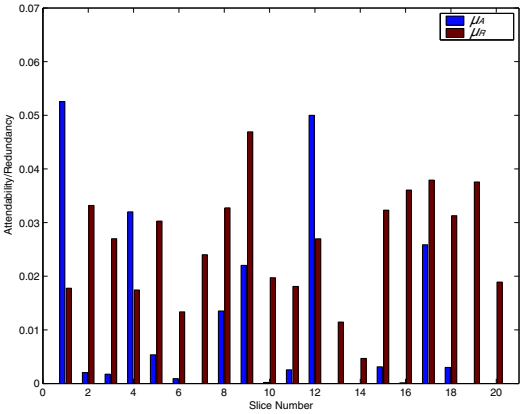

Fig. 8: Attendability/Redundancy of Correlations

The graphic in fig.8 shows that only very few and poor track measurement correlations would have been created in

the case when the preliminary mean track measurement correlation test is not fulfilled. Also it happens very few times that the preliminary mean track measurement correlation test is positive and no track in the tree correlates with the measurement. More detailed tests have shown that this happens only when the tree is very sparse, i.e it contains only very few tracks (normally on the order of 2-5 tracks). Due to space limitations the results of these tests will be reported only in [5].

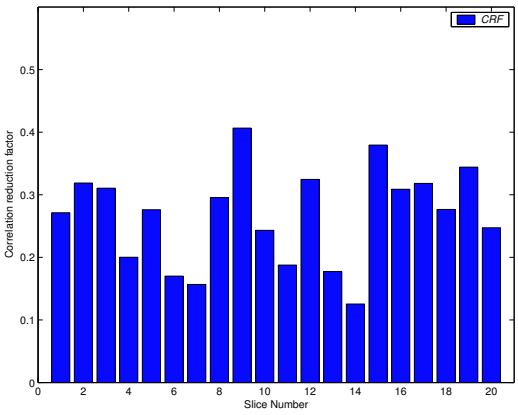

Fig. 9: Correlation Reduction Factor

Finally it turns out from the graphic in fig.9 that a reduction in the number of track measurement correlation tests greater than 60% is obtained using the mean child approach.

5 Conclusions

An algorithm for solving the multidimensional assignment problem in multiple sensor environments has been proposed in this paper. The algorithm has been shown to be polynomial with the number of track hypotheses. Also the amount of memory needed to recover a solution is very limited. It is only necessary to store the sensor measurements, the track hypothesis set and an integer array of a size equal to the number of the measurements to check if two track hypotheses intersect. Since the number of sensor measurements is never greater than the number of track hypotheses it can be safely concluded that the amount of memory used grows polynomially with the number of track hypotheses. The quality of the solution is mathematically guaranteed to be always within a certain distance from the optimal solution. Such a distance only depends on the dimension of the problem solved. Experiments presented in [2] have shown that the quality of the solutions returned is quite satisfactory already when the number of generated partitions is about twenty, the number of complete frames which are taken into account is 3 and only one sensor is used.

The experiments described in section 4.2 have shown that the cluster decomposition approach based on mean tracks described in the companion paper [4] provides a significant reduction in the number of measurement track correlations (fig. 9). In addition it has been shown that the mean track is a good indicator for possible correlations due to the following reason. In the case when a measurement does not correlate with the mean track of the tree then it is also very likely that such measurement does not correlate with any likely track of the same tree (fig. 8). And conversely, if the measurement correlates with the mean track of the tree, then it will also correlate with many other tracks of the tree (fig. 7).

6 Acknowledgments

This research has been supported by a Marie Curie fellowship of the European Community programme "Improving Human Research Potential and the Socio-economic Knowledge Base" under contract number HPMI-CT-2002-00221"

References

[1] S.S. Blackman and R.Popoli. *Design and Analysis of Modern Tracking Systems*. Artech House, 1999.

[2] A. Capponi. A polynomial algorithm for the data association problem in multitarget tracking. *Submitted to IEEE Transaction on Aerospace and Electronic systems*, 2004.

[3] M. Chummun, T. Kirubarajan, K. Pattipati, and Y. Bar-Shalom. Fast data association using multidimensional assignment with clustering. *IEEE Transactions on Aerospace and Electronic Systems*, Vol. 37(No. 3), 2001.

[4] H.W. de Waard and A. Capponi. An efficient approach to decompose the multidimensional assignment problem. *To appear in Information Fusion 2004*.

[5] H.W. de Waard and Agostino Capponi. A cluster based approach for the measurement track correlation problem. *In preparation for IEEE Transactions on Aerospace and Electronic systems*.

[6] S. Deb, K.R. Pattipati, and Y. Bar-Shalom. A new algorithm for the generalized multidimensional assignment problem. *Proceedings IEEE International Conference on Systems, Man and Cybernetics, Chicago, IL*, pages 249–254, 1992.

[7] M.M. Halldorsson. Approximations of weighted independent set and hereditary subset problems. *Journal of Graph Algorithms and Applications*, Vol. 4(No. 1):1–16, 2000.

[8] A.B. Poore. Multidimensional assignment formulation of data association problems arising from multitarget and multisensor tracking. *Computational Optimization and Applications*, Vol. 3:27–57, 1994.

[9] A.B. Poore, N. Rijavec, M. Liggins, and V.C. Vannicola. Data association problems posed as multidimensional assignment problems: Problem formulation. *SPIE*, Vol. 1954:552–564, 1993.

[10] D.B. Reid. An algorithm for tracking multiple targets. *IEEE Transactions on Automatic Control*, Vol. AC-24(No. 6), December 1979.