Greedy and K-Greedy algorithms for the Multi Target Tracking and Data Association Problem

Federico Perea, Huub W. de Waard

***Abstract*—The multidimensional assignment (MDA) problem is a combinatorial optimization problem arising in many applications, for instance Multi Target Tracking (MTT). The objective of a MDA problem is to match groups of $d \in \mathbb{N}$ objects (also called d-tuples) in such a way that the solution with the optimum total cost is found. It is well-known that the MDA problem is NP-hard. In this paper three new heuristics to solve the MDA problem arising in MTT are presented. They are all based on the semi-greedy approach introduced in an earlier research. The three heuristics of polynomial complexity. Experimental results on the accuracy and speed of the proposed algorithms in MTT problems are provided in the last section of the paper.**

***Index Terms*—OR in telecommunications, OR in military, Decision support systems, Combinatorial optimization, Heuristics.**

I. Introduction

According to Bar-Shalom and Xiao 1995, *tracking* is the processing of measurements obtained from a target in order to maintain an estimate of its current state, which typically consists of:

- Kinematic components - position, velocity, acceleration, turn rate, etc.
- Feature components - radiated signal strength, spectral characteristics, radar cross-section, target classification, etc.
- Constant or slowly varying parameters - aerodynamic parameters, etc.

One of the major difficulties in the application of multi-target tracking involves the problem of associating measurements received by a sensor, also called plots, with the appropriate target, forming a so-called *track hypothesis*. It is assumed that each target in the coverage of the scanning radar can produce a maximum of one measurement during a radar scan. To determine which measurements are likely candidates to originate from a certain track hypothesis, a correlation gate is positioned at the predicted position of the track hypothesis in the measurement space(Blackman 1986). The measurement is said to correlate with the track hypothesis if it falls within the defined correlation gate. As an example consider a track t. In the new scan two new measurements are received, p_1 and p_2. In Figure 1, the square represents the predicted position of track t, and the dots are the measurements p_1, p_2 . The circle represents the gate that corresponds to track t. In this example plot p_1 correlates with track t while plot p_2 does not.

F. Perea Departamento de Matemática Aplicada II. Escuela Superior de Ingenieros. Camino de los descubrimientos sn. 41092, Sevilla (Spain) Tel.: +34 654111231 Fax: +34 954486175 perea@us.es.

H.W. de Waard huub.dewaard@nl.thalesgroup.com.

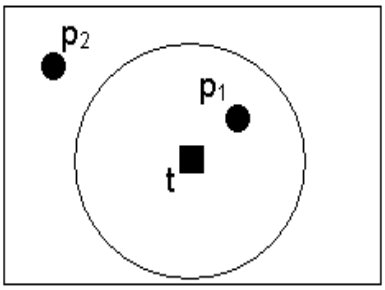

Fig. 1. Correlation gate.

The data association problem (DAP) for a number of data sets ≥ 3 is mathematically termed **NP**-hard (Deb et al. 1992 and Spieksma and Woeginger 1996). Here a data set corresponds with the data collected during a full scan. More precisely, the computational cost to determine an optimal solution, which assigns measurements to track hypotheses, can grow at a rate much faster than polynomial as the number of observations contained in the data sets increases. Since the pioneering work of Sittler 1969, who coined the term data association, a number of algorithms has been developed over the past 35 years to solve the DAP. (For an overview of different algorithmic approaches see Pattipati et al. 2000).

An elegant and efficient approach to solve the DAP is based on the application of assignment algorithms. In this approach the data association problem is formulated as a Multidimensional Assignment (MDA) problem. MDA problems are extensions of the classical Assignment problems to higher dimensional cases. In Multi-Target Tracking (MTT), the coefficients of the corresponding objective function are computed using the results of the state estimator. An early reference to the MDA problem can be found in Pierskalla 1968.

Morefield 1977 showed that the multi-target multi-scan data association problem could be expressed as a discrete optimization problem, for which mathematical programming methods are applicable, see Blackman and Popoli 1999. Morefield 1977 employed the branch-and-bound algorithm to solve the data association problem for a very sparse scenario, see Pattipati et al. 2000. Pattipati et al. 1989 and 1990 and later Poore 1992 and 1994, formulated the multi-target (multi-sensor) tracking problem as a MDA problem, and developed a multistage Lagrangian relaxation approach, following the work by Frieze and Yadegar 1981, to solve the MDA problem as a series of classical (two dimensional) assignment problems which are solvable in polynomial time. Rijavec et al. 1992 introduced a sliding window technique to consider only the last W data sets

(scans or frames). The problem is to find solutions for the optimization problem, using the measurements contained within the window and the track hypotheses which already exist outside the window. In Figure 2 those track hypotheses are related with the data sets with a number $\leq M$. Poore and Drummond 1996 presents a more general moving formulation for MHT. Given d data sets, the objective is to find the assignment of measurements to track hypotheses (targets) which optimizes the defined cost function. The measurements in the W data sets are associated with the list of track hypotheses, resulting in a $(W+1)$-dimensional data association problem. Capponi 2004 used a fast semi-greedy polynomial time algorithm to generate a set of required solutions from which the best solution was selected. He provided a theoretical upper bound for the difference in cost value between the unknown optimal solution and the first returned solution. In this paper we prove that this upper bound is tight. Another variation of greedy algorithms for the MTT problem is the GRASP algorithm proposed by Murphy et al. 1998. Greedy algorithms have been widely used to solve problems such as set-packing problems, set-covering problems, travelling salesman problem, etc., see Vazirani 2001. Despite their vast use, for some instances they should not be used, see Bang-Jensen et al. 2004 for a detailed description of cases in which the Greedy algorithm not only fails in finding the optimal solution but it obtains the worst possible one. Nevertheless, in this paper we present Greedy algorithms for their suitability to MTT problems and extend them to K-greedy versions, see Hausmann and Korte 1978 for an early reference on K-greedy algorithms.

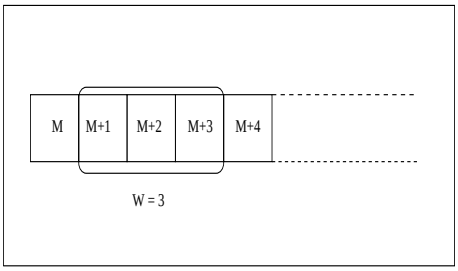

Fig. 2. The sliding window contains only three data sets: $d = 3$.

The rest of the paper is organized as follows. Section II is devoted to formulate the MTT problem as a MDA problem. In Section III the three greedy-based algorithms presented in this paper are introduced, which are extended to K-greedy in Section IV. Section V summarizes the experimental results obtained. The paper concludes with a brief summary of the contributions of this work.

II. Preliminaries: MTT as a MDA problem

In this section the multi-target tracking problem is formulated as a MDA problem. A more complete description of this process can be found in Reid 1979 and Poore 1994. Let us first describe the MTT problem.

Suppose that a sensor starts observing the airspace periodically at a time $y_0 = 0$. During the first scan, corresponding with the time interval $[y_0, y_1)$, the first set of measurements is received. Parameter y_1 denotes the point in time in which the first scan of the sensor finishes and the second one begins. Analogously, the second set of measurements, corresponding with the second scan is received within the interval $[y_1, y_2)$, and so on. The set of measurements collected at scan k is defined as $Z(k) = \{z_i^k\}_{i=1}^{M_k},\ \forall\ k = 1, \ldots, N$, where M_k denotes the total number of measurements received during scan k and z_i^k is the i^{th} measurement received within scan k. The cumulative data set for N scans is defined as $Z^N = \{Z(1), \ldots, Z(N)\}$. A *track hypothesis* t is defined as a set of measurements of Z^N such that it contains at most one measurement of each scan and consists of at least one measurement. This can be mathematically expressed as:

$$t \subset Z^N\ :\ |t| \geq 1,\ |t \cap Z(k)| \leq 1\ \forall\ k = 1, \ldots, N. \quad (1)$$

A feasible partition of Z^N is a set of track hypotheses $\delta = \{t_1, \ldots, t_{|\delta|}\}$ satisfying two conditions:

1) It must cover the whole data set Z^N, $Z^N = \bigcup_{j=1}^{|\delta|} t_j$.
2) Any two track hypotheses belonging to the same partition must not have common measurements, that is, they must be disjoint, $t_i \cap t_j = \emptyset\ \forall\ i \neq j,\ i, j = 1, \ldots, |\delta|$.

For the sake of readability, in the rest of the paper we will refer to feasible partitions as partitions. The set of all possible partitions of Z^N is denoted by $\Delta(Z^N)$.

After having defined what track hypotheses and partitions are, the goal is to find a best partition of Z^N, which needs not be unique. We will consider that "a best partition" is a partition that is most likely to correspond with the actual situation. To get such a partition a quality measure $Q(t)$ is assigned to each track hypothesis $t \subset Z^N$, which expresses how well each measurement of t fits the target's assumed dynamical model. Reid 1979 was one of the first to introduce a likelihood function $Q(t)$ for each track hypothesis t. A similar approach can be found in Poore 1994 and Storms and Spieksma 2003. Since the objective is to find a partition that is most likely to be true, it is necessary to maximize $\prod_{t\in\delta} Q(t)$ over the set of all possible partitions $\Delta(Z^N)$.

In order to write the DAP as a MDA problem, a linear objective function has to be defined. When for each track hypothesis t of Z^N the track formation cost is defined as $w(t) = \log(Q(t))$, the partition that maximizes $\prod_{t\in\delta} Q(t)$ also maximizes $\sum_{t\in\delta} w(t)$, since the logarithm is a monotonic function. An added benefit is also to reduce round-off errors that result from multiplying small numbers, such as likelihood functions. If with each partition $\delta = \{t_1, \ldots, t_{|\delta|}\}$ a weight $W(\delta) = \sum_{i=1}^{|\delta|} w(t_i)$ is associated, the goal is to find a partition δ^* such that:

$$W(\delta^*) = \max_{\delta\in\Delta(Z^N)} W(\delta). \quad (2)$$

Since the number of track hypotheses of the problem explosively grows with the number of scans, a sliding window approach is used. The main idea behind the sliding window technique consists of considering only the measurements of the last d scans, assuming that the assignments of the previous scans are fixed (see Figure 2). So, if a new scan is performed

the window slides one scan onwards, discarding the oldest scan of the previous window. The other $d-1$ scans are maintained. After including the new scan, the number of scans within the window is again restored to d scans. So, each time a set of measurements is received, the number of considered scans remains constant and the complexity of the corresponding MDA problem does not increase. A description of the method is provided by Poore 1992.

Now we are in a position to formulate the MTT problem as a MDA problem. A track $t = \{z_{i_0}^0, z_{i_1}^1, \ldots, z_{i_d}^d\}$ is either present in a partition δ or it is not, where $Z(0) = \{z_i^0\}_{i=1}^{M_0}$ is the set of previously established tracks. This corresponds to a 0-1 decision, which can be represented by the following decision variables:

$$x_{i_0,i_1,\ldots,i_d} = \begin{cases} 1 & \text{if } t = \{z_{i_0}^0, z_{i_1}^1, \ldots, z_{i_d}^d\} \in \delta \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

Let $c_{i_0,i_1,\ldots,i_d}$ be the weight of track $t = \{z_{i_0}^0, z_{i_1}^1, \ldots, z_{i_d}^d\}$. Then, the formulation of the MTT problem as a MDA problem is:

$$\begin{aligned} \max \quad & \sum_{i_0=1}^{M_0} \sum_{i_1=1}^{M_1} \cdots \sum_{i_d=1}^{M_d} c_{i_0,i_1,\ldots,i_d} x_{i_0,i_1,\ldots,i_d} \\ \text{s.t.:} \quad & \sum_{\substack{j=0 \\ j \neq k}}^{d} \sum_{i_j=1}^{M_j} x_{i_0,\ldots,i_d} = 1, \ i_k = 1, \ldots, M_k, \ k = 1, \ldots, d, \\ & x_{i_0,i_1,\ldots,i_d} \in \{0,1\} \ \forall \ i_0, i_1, \ldots, i_d. \end{aligned} \tag{4}$$

The constraints of the problem force each measurement to be in one, and only one, track hypothesis. That is, a measurement cannot originate from two different objects.

The size of the problem can also be reduced by excluding implausible track hypotheses. To do so, a target is only associated with a measurement if and only if the measurement falls within the gate or validation region for the target, as explained in Figure 1. The final set of track hypotheses considered is denoted by T_R.

Although the problem size is reduced by considering only the last d scans and by excluding implausible track hypotheses, the resulting MDA problem is still **NP**-hard for window sizes $d \geq 2$, since this leads to a $(d+1)-$dimensional assignment problem. When tracking multiple targets, a solution to the MDA problem must be given before the following scan of the sensor begins. This justifies the development of efficient approximation algorithms that provide good solutions for MDA problems.

The equivalence of the MHT (Multi-Hypotheses Tracking) formulation of MTT as a MDA problem was demonstrated in the work of Poore 1994. In this paper, a general formulation was presented for the multiple target tracking problem. It also discusses the LP formulation of Morefield 1977 in more depth. Specifically, the latter LP formulation is more general than the MDA problem since it can address different types of assignments such as merged measurements and some types of multiassignment. Algorithms for the multidimensional assignment problem on the other hand can be orders of magnitude faster than those for a general LP zero-one solver.

III. Two new Greedy-based algorithms

In this section the three algorithms under discussion are presented. The first one is algorithm SGTS (Semi Greedy Track Selection) introduced in Capponi 2004. The other two algorithms are variations of SGTS. In those algorithms, named Multi Greedy (MG) and Multi Greedy Rewarded (MGR), the way of choosing the best track at each step differs from algorithm SGTS.

A. SemiGreedy Track Selection algorithm

In this section we briefly recall algorithm SGTS presented in Capponi 2004. This algorithm first sorts the elements in T_R by decreasing weight. Afterwards it chooses the first track hypothesis in T_R as the first track of the solution. Then it removes from T_R all the track hypotheses intersecting the selected one. At this point, it selects the track hypothesis x with maximum weight from the remaining set and removes all track hypotheses intersecting x. SGTS keeps doing that until there is no track hypothesis to be selected. This part of SGTS is called SGTS1, and provides a set of tracks that constitute the first solution of the algorithm. A pseudocode of SGTS1 is:

SGTS1(T_R)
  $p_s = \{\}$
  $i \leftarrow 0$
  $T_i \leftarrow T_R$
  repeat
    $i \leftarrow i + 1$
    $x_i \leftarrow t \in T_{i-1}$ that maximizes $\omega(t)$
    $p_s \leftarrow p_s \cup x_i$
    $E_i \leftarrow \{t \in T_{i-1} : x_i \cap t \neq \emptyset\}$
    $T_i \leftarrow T_{i-1} \setminus E_i$
  until $T_i = \emptyset$
  return p_s

To obtain a larger variety of solutions, algorithm SGTS runs again from the complete T_R but starting with the first track t_i in T_R that has not yet been included in any of the previous solutions, which guarantees a new solution. The algorithm stops either when no other new solution can be generated or when we have run out of computational time. After that it chooses the solution with the highest weight among those that have been calculated, which does not have to be p_s necessarily.

1) Approximation factor: In Capponi 2004 it is proven that the value of the solution generated by SGTS1 approximates the value of the optimal one within a guaranteed factor depending only on the dimension of the window used.

Let OPT be an optimal partition. Let S_1 be the partition obtained as the solution returned by SGTS1. Denote by $W(\text{OPT})$ and $W(S_1)$ the values of the optimal solution and the first solution computed by SGTS respectively. It is proven that for a fixed d-dimensional assignment problem $W(\text{OPT}) \leq dW(S_1)$.

The following question arises in the context of improving the approximation guarantee: can the approximation guarantee of algorithm SGTS be improved by a better analysis? The following example states that the answer to this question is "no", i.e., the upper bound presented above for algorithm SGTS is tight.

Example 3.1: Consider the following data.

Track	Plot 1	Plot 2	...	Plot d	Weight
t_1	1	0	...	0	1
t_2	0	1	...	0	1
...	...	...	...	...	...
t_d	0	0	...	1	1
t_{d+1}	1	1	...	1	1+ε

TABLE I
TIGHT EXAMPLE.

Suppose that the data sets coming from d different scans consist of exactly one measurement in each scan. In Table I the track hypotheses that have been formed are shown. Track t_i is constituted by plot $i\ \forall\ i = 1, \ldots, d$ and track t_{d+1} is formed by all plots. The weights corresponding with all tracks are represented in the last column.

One may check that the set of pairwise disjoint tracks maximizing the sum of weights is OPT = $\{t_1, \ldots, t_d\}$. Nevertheless, the solution generated by SGTS1 would be the set consisting of only one track, $S_1 = \{t_{d+1}\}$. It follows that $W(\text{OPT}) = d$ and $W(S_1) = 1 + \varepsilon$. Thus, one has that:

$$W(OPT) = \frac{d}{1+\varepsilon} W(S_1) \quad \forall\ \varepsilon > 0. \tag{5}$$

Making $\varepsilon \to 0$ in the previous example concludes that the bound presented cannot be improved.

B. Multi-Greedy algorithm

In this section, the second approximation algorithm for solving the MDA problem is presented, named MG (Multi-Greedy). The only difference with algorithm SGTS is that, instead of selecting tracks that maximize weights, there will be chosen tracks that maximize *effectiveness*. Therefore, MG1 is the part of algorithm MG that finds the first solution. We define the effectiveness of a track hypothesis as the quotient between its weight and the number of measurements that constitute it. In other words, for every t track hypothesis, the effectiveness of t is defined as

$$\alpha(t) = \frac{w(t)}{|t|}. \tag{6}$$

In the rest of the algorithm, the steps taken are analogous to those taken in SGTS.

1) Approximation factor for algorithm Multi-Greedy: Denote by d and OPT the dimension of the window used and an optimal solution to the corresponding MDA problem. For every iteration i, define OPT_i to be the set of track hypotheses that are part of the optimal solution and have measurements in common with the tracks in E_i, $OPT_i = OPT \cap E_i$. Remember that x_i denotes the track hypothesis with highest efficiency and E_i denotes the set of available tracks that intersect x_i, at each step i of the algorithm. The following lemma is proven in Capponi 2004.

Lemma 3.1: The number of track hypotheses contained in the set OPT_i is less than or equal to the number of measurements contained in x_i, that is, $|OPT_i| \leq |x_i|$.

Afterwards, an analogous result to the Proposition 1 Capponi 2004 follows.

Theorem 3.1: Algorithm MG1 is a d factor approximation algorithm for a d-dimensional assignment problem, i.e.,

$$W(OPT) \leq dW(S^{MG1}), \tag{7}$$

where S^{MG1} is the first solution found by algorithm MG1.

Proof: For all i, we have that $\frac{w(t)}{|t|} \leq \frac{w(x_i)}{|x_i|}\ \forall\ t \in T_{i-1}$. Therefore

$$\begin{aligned} W(OPT_i) &= \textstyle\sum_{t \in OPT_i} w(t) \leq \sum_{t \in OPT_i} \frac{w(x_i)}{|x_i|}|t| \\ &= \textstyle\frac{w(x_i)}{|x_i|} \sum_{t \in OPT_i} |t| \leq \frac{w(x_i)}{|x_i|} d|OPT_i|. \end{aligned} \tag{8}$$

From Lemma 3.1 and the equation above we deduce that $W(OPT_i) \leq dw(x_i)$. Thus, one has that

$$W(OPT) = \sum_i W(OPT_i) \leq d \sum_i w(x_i) = dW(S_1). \tag{9}$$

And the result follows. ■

The following example shows that the approximation guarantee is tight.

Example 3.2: Consider the MTT problem described in Example 3.1, with weights $w(t_1) = \frac{1}{d},\ w(t_2) = w(t_3) = \cdots = w(t_d) = 0,\ w(t_{d+1}) = 1 - \varepsilon$.

It is not difficult to see that the optimal solution and the solution returned by MG1, and their respective values, are:

$$OPT = \{t_{d+1}\},\ W(OPT) = 1 - \varepsilon, \quad S_1 = \{t_1, t_2, \ldots, t_d\},\ W(S_1) = \frac{1}{d} \tag{10}$$

Suppose that the approximation factor given in Theorem 3.1 can be improved to $d' < d$. Then it should be satisfied that $W(OPT) \leq d'W(S_1)$. Consider $\varepsilon > 0$ such that $\varepsilon < 1 - \frac{d'}{d}$. Then one has:

$$\begin{aligned} W(OPT) &= 1 - \varepsilon = (1 - \varepsilon) dW(S_1) \\ &> (1 - (1 - \tfrac{d'}{d}))dW(S_1) = d'W(S_1). \end{aligned} \tag{11}$$

This contradicts the fact that d' is an approximation factor for algorithm MG1, and we conclude that the approximation factor d for algorithm MG1 cannot be improved.

C. Multi-Greedy Rewarded algorithm

Now the third approach for solving the MDA problem is presented, named MGR (Multi-Greedy Rewarded). It is again a greedy-based algorithm. The idea is based on the fact that for some MDA problems the more elements the associations have the better. For MTT this means that tracks with many plots are preferred to those that contain only a few measurements. This reasoning seems to be logical, and will be tested in Section V. Therefore, at each step we choose $t \in T_{i-1}$ that maximizes $|t|\omega(t)$

1) Guarantee bound: Following the same reasoning as we did for algorithms SGTS and MG, see sections III-A1 and III-C1, we conclude the following result.

Theorem 3.2: Algorithm MGR1 is a d^2 factor approximation algorithm for a d-dimensional assignment problem, i.e.,

$$W(OPT) \leq d^2 W(S_1). \tag{12}$$

Proof: For all i, we have that $w(t)|t| \leq w(x_i)|x_i| \ \forall \ t \in T_{i-1}$. Therefore

$$\begin{aligned} W(OPT_i) &= \textstyle\sum_{t \in OPT_i} w(t) \leq \sum_{t \in OPT_i} \frac{w(x_i)}{|t|}|x_i| \\ &= w(x_i)|x_i| \textstyle\sum_{t \in OPT_i} \frac{1}{|t|} \leq w(x_i)|x_i||OPT_i|. \end{aligned} \tag{13}$$

From Lemma 3.1 and the equation above we deduce that $W(OPT_i) \leq w(x_i)|x_i|^2$. Thus, one has that

$$W(OPT) = \sum_i W(OPT_i) \leq d^2 \sum_i w(x_i) = d^2 W(S_1). \tag{14}$$

And the result follows. ∎

The following example shows that the given guarantee bound is tight.

Example 3.3: Consider the instance of Example 3.1 with the following weight function w: $w(t_1) = \cdots = w(t_d) = d$ and $w(t_{d+1}) = 1 + \varepsilon$. Then MGR would take t_{d+1} with total weight $1 + \varepsilon$ since $|t_i|w(t_i) = d \ \forall \ i = 1, \ldots, d$ and $|t_{d+1}|w(t_{d+1}) = d(1 + \varepsilon)$ while $\{t_1, \ldots, t_d\}$ is an optimal solution with total weight d^2.

IV. K-Extensions of our greedy algorithms

The three proposed heuristics can be extended in the following way. At each step the track hypothesis with the highest value among all available track hypotheses is not selected. Instead, we choose at each step a group of K *disjoint* track hypotheses, just named K-group. Such a group is constituted by K track hypotheses with no measurements in common maximizing the sum of weights, K being a positive integer. After selecting this first K-group, we remove every K-group of track hypotheses that has at least one measurement in common with the previously selected group.

It could happen that there is no group of K disjoint hypotheses available and there still are measurements that have not been assigned to any track hypothesis yet. Then, the maximum weight $(K-1)$-group is selected. We continue reducing the size of the groups until all measurements have been assigned to track hypotheses. This procedure generates a first partition of the set of measurements Z^N, that is, a first feasible solution.

Afterwards, to obtain a larger variety of feasible solutions, we start the algorithm from that K-group with the highest value among all K-groups that have at least one track that has not been included in any previous solution. Note that this process always generates new solutions since we are including, at least, one new track in the solution. This procedure is repeated until $X \in \mathbb{N}$ different solutions have been generated or the allowed computational time has been consumed. This kind of algorithms is knwon in the literature as K-Greedy algorithms, see Hausmann and Korte 1978 for an early reference.

In the rest of the section we apply this extension to the algorithms previously presented: SGTS, MG and MGR.

A pseudocode of the algorithm that generates the first solution returned by K-SGTS, from now on called K-SGTS1, is:

K-SGTS1(T_R)
  $n = K$
  $p_s = \{\}$
  while $T_R \neq \emptyset$
    $i \leftarrow 0$
    $T_i \leftarrow T_R$
    $T_i^n \leftarrow \{(t^1, \ldots, t^n) : t^j \in T_R, t^j \cap t^h = \emptyset, \ \forall \ j, h\}$
    repeat
      $i \leftarrow i + 1$
      $x_i \leftarrow t \in T_{i-1}^n$ that maximizes $\sum_{j=1}^n \omega(t^j)$
      $p_s \leftarrow p_s \cup x_i$
      $E_i \leftarrow \{t \in T_{i-1} : \cup_{j=1}^n (x_i^j \cap t) \neq \emptyset\}$
      $T_i^n \leftarrow \{t \in T_{i-1}^n : t^j \notin E_i \ \forall \ j = 1, \ldots, n\}$
      $T_i \leftarrow T_{i-1} \setminus E_i$
    until $T_i^n = \emptyset$
    $T_R \leftarrow T_i$
    $n \leftarrow n - 1$
  end
  return p_s

In the pseudocode above, p_s denotes the solution that K-SGTS1 generates. Note that the size of the groups (n) decreases every time we cannot find a feasible group but there still are some measurements that have not been assigned to tracks. The algorithm stops, at the latest, when $n = 1$.

In order to study the complexity of K-SGTS, the following notation will be used:

- K denotes the size of the group of track hypotheses at the beginning of the algorithm, $K \in \mathbb{N} - \{0\}$.
- d denotes the number of scans considered, $d \in \mathbb{N} - \{0, 1\}$.
- $l = |T_R|$ is the number of plausible track hypotheses, $l \in \mathbb{N}$. Note that this number can be very large and depends on the number of measurements, the number of scans and the method used to discard implausible tracks.
- In each scan s there are M_s measurements, $M_s \in \mathbb{N}$, $s = 1, \ldots, d$.

To begin with, the complexity of the algorithm K-SGTS1 is determined. In this analysis we will calculate the computational complexity of K-SGTS1 in its first loop, that is, when $n = K$ (the cases corresponding with other values of n are analogous). Several steps can be differentiated:

1) Building T_0^K.
   For each possible group of K track hypotheses, we have to check if all the measurements that constitute them are different. In each track hypothesis there are at most d measurements, each of them coming from different scans. Therefore, the total number of operations to be done in order to check the feasibility of a K group is, in the worst case, equal to Kd. Since there are $\binom{l}{K}$ possible K-groups of track hypotheses, the number of comparisons for building T_0^K is

$$dK\binom{l}{K}. \tag{15}$$

2) Ordering T_0^K.
Mergesort or Heapsort algorithms sort a list of elements with complexity $O(n \log n)$. Therefore the number of operations for ordering T_0^K is

$$\binom{l}{K} \log \binom{l}{K}. \tag{16}$$

3) Taking the first group of K track hypotheses.
This step takes a constant amount of time.
4) Selecting the other K-groups of track hypotheses. The algorithm checks for every K-group $t \in T_0^K$ that every measurement j contained in t has not been selected yet. If t satisfies this condition, it is chosen to be part of the solution. If not, the following element of T_0^K is checked and so on. The maximum number of operations to be done is

$$dK(\binom{l}{K} - 1). \tag{17}$$

Joining the previous steps, the complexity of this part of the algorithm follows:

$$O(l^K \log l). \tag{18}$$

Analogously, the complexity of the loop corresponding with any other value of n is:

$$O(l^n \log l). \tag{19}$$

Thus, the total complexity of K-SGTS1 is given by,

$$\sum_{n=1}^{K} O(l^{n \log l}) = O(l \log l + \cdots + l^{K-1} \log l + l^K \log l) = O(l^K \log l). \tag{20}$$

Proposition 4.1: K-SGTS1 is a $O(l^K \log l)$ algorithm.

Proof: The discussion above. ■

The complexity of K-SGTS has not been calculated yet. To do so, we have to consider that the algorithm calculates not only one solution but X solutions. The computational complexity of finding solution i, $i > 1$, can easily be calculated from the fact that only steps 3 and 4 have to be performed (T_0^K needs not to be built and sorted again). So, the reader may note that the computational complexity of generating solution i is $O(l^K)\ \forall\ i = 2, \ldots, X$.

Theorem 4.1: For a fixed size $K \in \mathbb{N}$, the computational complexity of algorithm K-SGTS is polynomial, and given by $O(l^K \log l)$.

Proof: Let G_i be the maximum number of operations to find solution i, $i = 1, \ldots, X$. Then, the complexity of K-SGTS is

$$\sum_{i=1}^{X} G_i = O(l^K \log l) + \sum_{i=2}^{X} O(l^K) = O(l^K \log l). \tag{21}$$

■

Algorithms MG and MGR can be generalized following a similar reasoning as in Section **??**. Again, the main idea is to take at each step the group of K disjoint track hypotheses that maximize the sum of effectiveness, instead of taking only the one that individually maximizes the effectiveness. These heuristics shall be called K-MG and K-MGR, respectively.

Following the same steps as in the proof for the complexity of K-SGTS in **??**, the following results are proven.

Theorem 4.2: For a fixed $K \in \mathbb{N}$, the computational complexity of algorithms K-MG and K-MGR is polynomial, given by $O(l^K \log l)$, where l denotes the number of track hypotheses considered at the beginning of the algorithm.

Corollary 4.1: Algorithms MG and MGR are of polynomial complexity, given by $O(l \log l)$.

V. Experimental results

In this section, the most relevant results obtained from the multi-target tracking problems tested are presented. In those experiments, the performance of algorithms 2-MGR, MGR, 2-SGTS, SGTS, 2-MG and MG is analyzed.

For these experiments we used three kinds of data:

1) Simulated scenarios 1 (sim1). These scenarios consisted of groups of planes that start flying in straight lines from different points in the space to the same area. Afterwards they keep their trajectory until they disappear from the radar coverage. The number of planes in those scenarios vary from 48 till 80.
2) Simulated scenarios 2 (sim2). The same as sim1 but the number of planes vary from 2 till 10.
3) Real data (real). Those data were collected in October 2005, on a vessel 40 miles from the Dutch shore in the North Sea. The used radar sensor has a maximum detection range of 250 km.
Figure 3 shows a picture of the real situations under consideration, where each arrow represents a flying object. The objective is to associate the new measurements received in every scan to existing flying objects, to new targets or to false alarms.

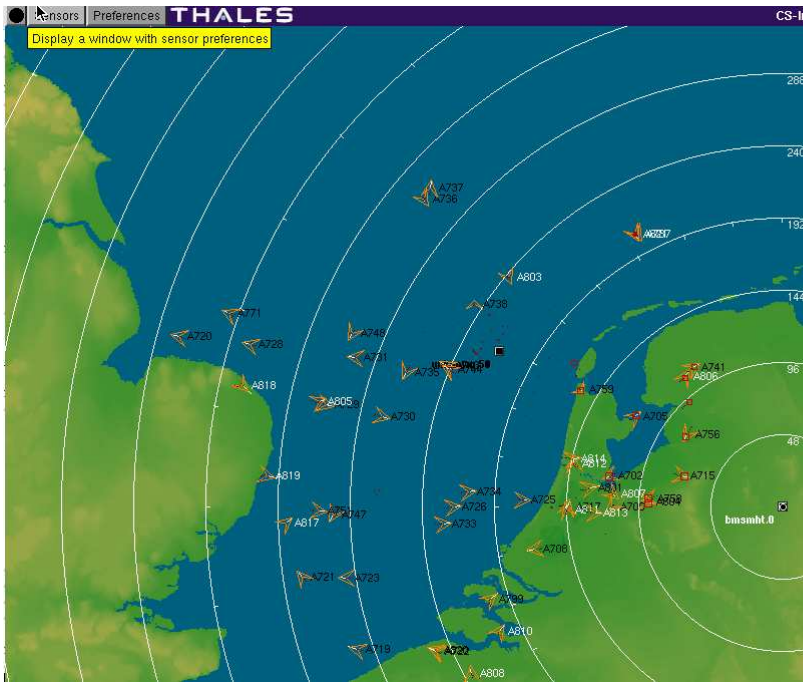

Fig. 3. Planes flying over the North Sea.

	d = 4	d = 5	d = 6
sim1	524	528	573
sim2	223	229	237
real	1648	1607	1589

TABLE II
NUMBER OF MDA PROBLEMS SOLVED

All scenarios were tested for window sizes $d = 4, 5, 6$, generating this way 5, 6 and 7 dimensional assignment problems respectively. The number of MDA problems solved in each case is summarized in table II (in total 7158).

For each MDA problem, the following algorithms were executed:

- an optimal solution was found by using the free optimization software lp_solve, which uses a branch and bound (B&B) algorithm.
- SGTS algorithm generating 100 solutions, $X = 100$.
- 2-SGTS algorithm generating 100 solutions, $X = 100$.
- MG algorithm generating 100 solutions, $X = 100$.
- 2-MG algorithm generating 100 solutions, $X = 100$.
- MGR algorithm generating 100 solutions, $X = 100$.
- 2-MGR algorithm generating 100 solutions, $X = 100$.

In the rest of the section, a summary of the performance of each algorithm is shown.

Table III shows the average computation time necessary to find the optimal solution and to find the first solution of the four approximation algorithms considered. It can be seen

Algorithm	B&B	MGR	2-MGR	
CPU time (seconds)	0.01604	0.00073	0.00079	
Algorithm	2-MG	SGTS	2-SGTS	MG
CPU time (seconds)	0.00116	0.00072	0.00075	0.00123

TABLE III
COMPUTATION TIME.

that the approximation algorithms found a first feasible near-optimal solution around 100 times faster than the branch and bound algorithm. Another conclusion is that no significant difference in computation time was found between the six proposed approximation algorithms when changing from $K = 1$ to $K = 2$.

But not only speed is important in an approximation algorithm. The accuracy of the generated solution should be also considered. Table IV shows the frequency in which each of the algorithms proposed found the optimal solution. Additionally, in order to give a more complete result, the third rows show a measurement of the accuracy of each algorithm, which is calculated by the formula $100\frac{\text{ALGORITHM VALUE}}{\text{OPTIMAL VALUE}}$. Remember that for every algorithm the first 100 solutions were calculated. Note that the performances of the six algorithms in their first solution are also shown in the third and fifth columns of each table.

From table IV it can be noted that the fact of generating not only one solution but more than one is justified as the performance of each algorithm significantly improves when generating more than one solution. The following question

Algorithm	MGR	MGR1	2-MGR	2-MGR1
Frequency of optimality	70.41%	64.80%	69.03%	61.66%
Accuracy	98.95%	98.22%	98.94%	98.06%

Algorithm	MG	MG1	2-MG	2-MG1
Frequency of optimality	30.32%	22.20%	33.32%	22.25%
Accuracy	94.63%	92.57%	94.99%	92.66%

Algorithm	SGTS	SGTS1	2-SGTS	2-SGTS1
Frequency of optimality	70.62%	65.92%	69.47%	62.67%
Accuracy	99.01%	98.25%	99.02%	98.08%

TABLE IV
QUALITY OF SOLUTIONS.

regarding this matter arises: how many solutions should be generated to assure an acceptable guarantee that the best solution of the algorithm is found? One approach to answer this question is to calculate the frequencies in which the best solution generated by each algorithm is one of the first i solutions, $i = 1, \ldots, 100$. In table V, the cumulated

Solution number	MGR	2-MGR	MG	2-MG	SGTS	2-SGTS
10	79.76%	78.29%	48.94%	44.95%	85.57%	85.41%
20	82.98%	81.63%	57.46%	52.68%	90.52%	90.21%
30	85.63%	84.37%	64.51%	51.28%	92.81%	92.50%
40	88.05%	84.37%	69.73%	59.79%	94.61%	94.39%
50	90.03%	86.88%	73.61%	69.65%	95.87%	95.61%
60	92.04%	89.02%	78.50%	74.61%	97.00%	96.80%
70	94.65%	91.34%	83.99%	80.90%	97.89%	97.70%
80	96.84%	94.53%	90.15%	87.70%	98.79%	98.57%
90	98.60%	98.27%	94.39%	92.80%	99.51%	99.24%
100	100%	100%	100%	100%	100%	100%

TABLE V
POSITION OF THE BEST SOLUTION FREQUENCIES.

frequencies in which the best solution of each algorithm was found among the i first ones, where i is the number in the first cell of each row, are shown. For instance, according to our experiments, to obtain the best solution between the first 100 solutions with a probability greater than or equal to 0.9, it will be necessary to calculate:

- 50 solutions for algorithm MGR.
- 70 solutions for algorithm 2-MGR.
- 80 solutions for algorithm MG.
- 90 solutions for algorithm 2-MG.
- 20 solutions for algorithm SGTS.
- 20 solutions for algorithm 2-SGTS.

A quick look at the results shown in this section suggests that algorithms K-SGTS and K-MGR perform much better than algorithms K-MG in Multi Target Tracking problems, for $K = 1, 2$. A reason for that could be that in K-MG algorithm tracks constituted by a small number of measurements are rewarded. This fact is illogical for MTT problems, although it could be valid in general MDA problems. It is surprising for the authors that the fact of rewarding tracks with great number of plots does not seem to give an improvement, as K-SGTS and K-MGR seem to perform likewise.

Since there seems to be no difference in time from one algorithm to another, and the highest accuracy of our experi-

ments was given by 2-SGTS, we suggest choosing algorithm 2-SGTS for MTT problems. From the tests performed, running algorithm 2-SGTS generating only 20 solutions seems to be a good trade-off between speed and accuracy.

CONCLUSIONS

In this paper three heuristics for solving the multidimensional assignment problem are presented, named K-SGTS, K-MG and K-MGR. Theoretical analysis prove that the three heuristics are of polynomial complexity. Besides, upper bounds showing that the solution given by the algorithm is "close" to the optimal solution are provided for the three algorithms. Although the algorithms are presented in a MTT context, as well as the experiments, the theoretical reasonings are valid for every MDA problem in general.

In order to show their applicability, the algorithms presented were tested for multi-target tracking problems. No significant difference in accuracy from the case $K = 1$ to the case $K = 2$ was observed for any of the three classes presented.

The difference in computational time was not significant from the case $K = 1$ to $K = 2$, nor between the different heuristics.

To finish the paper we recommend to run algorithm 2-SGTS generating only 20 solutions, since that seems to be a good trade-off between speed and accuracy.

ACKNOWLEDGMENTS

The authors want to thank Wojciech Mlynarczyk, André Bonhof and José Benito for their help during the implementation phase. This research has been supported by a Marie Curie fellowship of the European Community program "Improving Human Research Potential and the Socio-economic Knowledge Base" under contract Number HPMI-CT$-2002-00221$.

REFERENCES

[1] Bang-Jensen, J., Gutin, G. and Yeo, A. (2004) When the greedy algorithm fails. Discrete Optimization 1, 121-127 .

[2] Bar-Shalom, Y. and Xiao-Rong Li. (1995) Multitarget-Multisensor Tracking: Principles and Techniques. Storrs, CT: YBS Publishing.

[3] Blackman, S.S. (1986) Multiple-target tracking with radar applications. Artech House.

[4] Blackman, S.S. and Popoli, R. (1999) Design and Analysis of Modern Tracking Systems. Artech House.

[5] Capponi, A. (2004) Polinomial Time Algorithm for Data Association Problem in Multitarget Tracking. IEEE Transactions on Aerospace and Electronic Systems 40, pp. 1398-1410.

[6] Deb, S., Pattipati, K.R. and Bar-Shalom, Y. (1992) A new algorithm for the generalized multidimensional assignment problem. Proc. IEEE Intn'l Conf. on Systems, Man and Cybernetics 249-254.

[7] Frieze, A.M. and Yadegar, J. (1981) An algorithm for solving 3-dimensional assignment problems with application to scheduling a teaching practice. Journal of the Operational Research Society, 32, pp. 989-995, 1981.

[8] Hausmann, D. and Korte, B. (1978) K-Greedy Algorithms for Independence Systems. Zeitschrift für Operations Research, 22. pp. 219-228.

[9] Morefield, L. (1977) Application of 0-1 integer programming to multitarget tracking problem. IEEE Transactions on Automatic Control, AC-22. pp. 302- 312.

[10] Murphey, R., Pardalos, P. and L. Pitsoulis (1998) A GRASP for the Multitarget Multisensor Tracking Problem In DIMACS Series Vol. 40, American Mathematical Society, pp. 277-302.

[11] Pattipati, K.R. and Bar-Shalom, Y. and Washburn, R.B. (1989) Passive multisensor data association using a new relaxation algorithm. Proc. American Control Conf, Pittsburg, 3:2, 617-2,627.

[12] Pattipati, K.R. and Bar-Shalom, Y. and Washburn, R.B. (1990) Passive multisensor data association using a new relaxation algorithm. In: Multitarget-Multisensor Tracking: Advanced Applications (ed: Bar-Shalom, Y.), pp. 219-246.

[13] K.R. Pattipati, R.L. Popp and T. Kirubarajan (2000), Survey of assignment techniques for multitarget tracking. In: Y. Bar-Shalom and W.D. Blair, Editors, Multitarget-Multisensor Tracking: Advanced Applications vol. III, Artech House, Boston, pp. 77-159 (Chapter 2).

[14] W. Pierskalla, (1968) The multidimensional assignment problem. Operations Research 16, pp. 422-431.

[15] Poore, A.B. (1994) Multidimensional Assignment Formulation of Data Association Problems Arising from Multitarget and Multisensor Tracking. Computational Optimization and Applications 3, pp. 27-57.

[16] Poore, A.B. and Drummond, O.E., (1996) Track Initiation and Maintenance Using Multidimensional Assignment Problems, Network Optimization, (Editors) P.M. Pardalos, Et Al, Springer-Verlag.

[17] Poore, A.B., Rijavec, N. and Barker, T. N. (1992) Data association for track initiation and extension using multiscan windows. In Proceedings of the SPIE Signal and data processing of small targets. SPIE, 1698. Orlando, FL, pp. 432-441.

[18] Reid, D.B. (1979) An Algorithm for Tracking Multiple Targets. IEEE Transactions on Automatic Control, AC-24,6, pp. 843-854.

[19] Rijavec, Barker, T. and Poore, A. (1992). Data association for track initiation and extensio nusing multiscan windows, Signal and Data Processing of Small Targets 1992, Proc. SPIE 1698, pp. 432-441.

[20] Sittler, R.W. (1964) An optimal Data Association Problem in surveillance theory. IEEE Trans. Mil. Electron 8, pp. 125-139.

[21] Spieksma, F.C.R. and Woeginger, G.J. (1996) Geometric three-dimensional assignment problems. European Journal of Operations Research 91, pp. 611-618.

[22] Storms, P.P.A. and Spieksma, F.C.R. (2003) An LP-based algorithm for the data association problem in multitarget tracking. Computers & Operations Research 30, pp. 1067-1085.

[23] Vazirani, V.V. (2001) Approximation Algorithms. Springer.