PERGAMON

Computers & Operations Research 30 (2003) 1067–1085

computers & operations research

www.elsevier.com/locate/dsw

An LP-based algorithm for the data association problem in multitarget tracking ☆

P.P.A. Storms[a,*], F.C.R. Spieksma[b]

[a] *Acklin B.V., Parkstraat 1a, NL-4818 SJ Breda, Netherlands*

[b] *Faculty of Economic and Applied Economic Sciences, Katholieke Universiteit Leuven, Naamsestraat 69, B-3000 Leuven, Belgium*

Received 1 October 2000; received in revised form 1 June 2001

Abstract

In this work we present a linear programming (LP) based approach for solving the data association problem (DAP) in multiple target tracking. It is well-known that the DAP can be formulated as an integer program. We present a compact formulation of the DAP. To solve practical instances of the DAP we propose an algorithm that uses an iterated K-scan sliding window technique. In each iteration we solve the LP relaxation of an integer program and next apply a greedy rounding procedure. Computational experiments indicate that the quality of the solutions found is quite satisfactory.

Scope and purpose

The purpose of this article is to present an alternative approach to solving data association problems that arise from multitarget tracking. As a starting point we use the work reported by Poore et al. (Comput. Optim. Appl. 3 (1994) 27; SPIE 1954 (1993) 552) in which the data association problem is viewed as a multidimensional assignment problem. Instead of using Lagrangian relaxation techniques (SIAM J. Optim. 3 (1993) 554; SPIE 1955 (1993) 172; SPIE 1954 (1993) 564; SPIE 2561 (1995) 448), we apply linear programming (LP) relaxation.

A feasible solution is constructed from the relaxed solution by a new heuristic greedy rounding procedure called GRP.

The choice for a heuristic approach is justified by the hard real-time requirements that are inherent to multitarget tracking.

The application of the GRP lies mainly within the field of (air) surveillance systems for both the military and the civilian domain. Although the focus of this paper is on single sensor systems (i.e. a surveillance radar), the scope can be extended to sensor data fusion systems in which multiple (dissimilar) sensors

☆ An abstract corresponding to this work appeared in the proceedings of FUSION 2000 [1]; part of this work was supported by EU Grant 14084 (APPOL).

* Corresponding author.

E-mail addresses: patrick@acklin.nl (P.P.A. Storms), frits.spieksma@econ.kuleuven.ac.be (F.C.R. Spieksma).

PII: S0305-0548(02)00057-6

are involved. More generally, real-time algorithms that provide good-quality solutions for multidimensional assignment problems have a broad scope of applications in the field of robotics (e.g. planning of robot arm movement), communications and the military domain (e.g. weapon assignment, sensor resource allocation, etc.).

Keywords: Multitarget tracking; Data association; Linear programming

1. Introduction

A basic question in the field of (air) surveillance systems is: how to keep track of all observed targets using information from all available sensors? Sensors, like radar or infrared, detect and locate objects (*targets*) that are present within their coverage. Each time a detection (also called *measurement* or *plot*) takes place, several features are measured, such as the location of the object (expressed in range, bearing and elevation) and Doppler shift (a measure for the velocity in the direction of the sensor).

In this work, we restrict ourselves to a single scanning surveillance radar, providing two-dimensional positional information (i.e. range and bearing). Furthermore, we assume that the radar's measurements are processed after each scan, i.e. a full rotation of the radar antenna.

The so-called Data Association Problem (DAP) is now to find out which sensor detections originate from which target. More precisely, when assuming that for each scan at most one measurement can be assigned to a target, a problem arises if more than one measurement in a scan is a candidate to be assigned to the same target. Even more, a single measurement may be a candidate to be associated with two or more targets. The DAP aims at finding an assignment of measurements to targets such that a measurement is associated with at most one target and such that a target receives at most one measurement per scan. Notice that, in operational circumstances, the DAP has to be solved after each scan, for all collected scans of measurements.

Complicating factors when solving instances of the DAP are the presence of spurious plots (*false alarms*), noise contaminated measurements, (deliberate) target maneuvres, a sensor detection probability $<100\%$ which leads to missed detections, and hard real-time requirements. Each received plot either has to be associated to an existing target, or has to be labeled as a new target or a false alarm. A *track* is defined as a sequence of measurements which are assigned to the same target.

The DAP is illustrated by Fig. 1. The circle represents the sensor's coverage, the dots are sensor plots that are received in sensor scans A–F. Interconnected dots represent a track, i.e. the plots are assumed to originate from the same target. Track A2–B1–C2–D2–F3 has a missed detection for scan E. Plots A1 and E1 are false alarms.

Notice that each individual plot is considered to be a potential target and is initialized as a tentative track. A track is declared "confirmed" if there is sufficient confidence that the track represents a true target, e.g. when the track consists of a minimum number of plots.

The goal of this work is threefold:

- to present a compact, integer programming (IP) formulation of the DAP,
- to propose an algorithm that is capable of solving a series of instances of the DAP, and
- to test this algorithm on randomly generated instances.

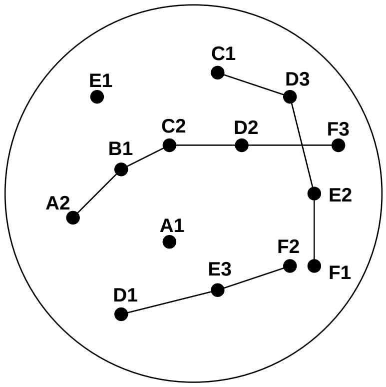

Fig. 1. Example of the data association problem.

Let us now review some of the literature.

A well-known DAP solution method is *Multiple Hypothesis Tracking* (MHT), based on a formulation by Reid [2]. The algorithm enumerates possible assignments (plot to existing track, plot to new track and plot to false alarm) and represents them in a hypothesis tree. For each hypothesis a likelihood of truth is calculated, after which the most probable assignment is chosen as the solution for the DAP. Notice that the solution found after scan k is not necessarily a part of the solution for scan $k+1$. Other less likely hypotheses are maintained, since they might eventually be used when future plots are received. MHT deals with the combinatorial explosion of possible assignments by using a number of pruning techniques, for instance based on clustering (see e.g. [3]).

Instead of MHT's enumerative approach, alternative methods view DAP as an integer programming problem. More specific, the problem is described as a multidimensional assignment problem (see e.g. [4] for an early reference), a problem in the field of combinatorial optimization.

An early contribution in this direction dates back to 1977, when C.L. Morefield suggested to apply 0–1 integer programming to the DAP [5]. More recent work on this subject is reported in [6–12]. In these papers the DAP is not only formulated as a multidimensional assignment problem, in addition a number of ways to solve the problem, notably by using Lagrangian relaxation, are suggested.

This work (see also [13]) is organized as follows. Section 2 is devoted to an IP formulation of the DAP; we propose a compact formulation of the integer program. In Section 3, we describe the heuristic algorithm, and in Section 4, we present some computational results. These are discussed in Section 5 and Section 6 contains the conclusions.

2. A compact IP formulation of the DAP

In this subsection, we propose a compact formulation of the integer programming formulation of the DAP. This compact formulation follows the formulation of the axial multidimensional assignment problem as suggested in e.g. [14] and avoids multiple summation signs and multiple indices. In Section 2.1, we introduce some notation; in Section 2.2 the decision variables are defined, and the (linear) constraints are introduced. Finally, Section 2.3 develops the objective function of the model.

2.1. Notation

A radar is assumed to receive measurements periodically. Suppose that at time $t_0 = 0$ the radar system starts observing the region, then the first set of measurements is taken within the interval $[t_0, t_1)$, where t_1 is the time when the first scan ends and the second scan begins. The duration T of each scan is assumed to be constant, so scan k ends at time $t_k = kT$. A set of measurements in scan k is defined as follows:

$$Z(k) = \{\mathbf{z}_{i_k}^k\}_{i_k=1}^{M_k}, \quad \text{for all } k = 1, \ldots, N. \tag{1}$$

Here, N denotes the number of scans, M_k is the total number of measurements received in scan k and $\mathbf{z}_{i_k}^k$ is the i_kth measurement vector within scan k ($1 \leqslant i_k \leqslant M_k$). Thus, notice that each $\mathbf{z}_{i_k}^k$ is a vector containing e.g. range, bearing and elevation data. For instance, in the example described in Fig. 1 we have 6 scans (so $N = 6$), each scan consisting of, respectively, 2, 1, 2, 3, 3 and 3 measurements (so $M_1 = 2$, $M_2 = 1$, $M_3 = 2$, $M_4 = 3$, $M_5 = 3$ and $M_6 = 3$).

In order to be able to deal with false alarms, missed detections and track initiation, the concept of a *dummy report* (see [9]) is introduced. To do so, let us redefine (1) by adding a dummy measurement, indexed $i_k = 0$, to each of the sets $Z(k)$. Formally:

$$Z(k) = \{\mathbf{z}_{i_k}^k\}_{i_k=0}^{M_k}, \quad \text{for all } k = 1, \ldots, N. \tag{2}$$

We refer to a non-dummy measurement as a *true* measurement. Now, we define the set Z as the Cartesian product of the sets $Z(k)$, $k = 1, \ldots, N$. Thus,

$$Z = Z(1) \times Z(2) \times \cdots \times Z(N) = \prod_{k=1}^{N} Z(k). \tag{3}$$

An element $\mathbf{z} \in Z$ is an N-dimensional vector whose kth entry ($\mathbf{z}(k)$) corresponds to a measurement in the kth scan. Because we do not allow a vector consisting solely of dummy reports, we define $Z^* = Z \backslash \{z_0^1, z_0^2, \ldots, z_0^N\}$. We will refer to any element $\mathbf{z} \in Z^*$ as a track. Unlike any true measurement, a dummy report can belong to an arbitrary number of tracks.

Let us now explain how the dummy report allows us to deal with missed detections, false alarms and track initiation.

The set $\{\mathbf{z}_{i_1}^1, \ldots, \mathbf{z}_{i_{k-1}}^{k-1}, \mathbf{z}_0^k, \mathbf{z}_{i_{k+1}}^{k+1}, \ldots, \mathbf{z}_{i_N}^N\}$ denotes a track with a missed detection in the kth scan, where $i_1, \ldots, i_{k-1}$ and $i_{k+1}, \ldots, i_N$ are non-zero indices. Notice that more than one missed detection can appear in a track.

A false alarm in scan k is represented as a track with exactly one true measurement, i.e. $\{\mathbf{z}_0^1, \ldots, \mathbf{z}_{i_k}^k, \ldots, \mathbf{z}_0^N\}$, where i_k is a non-zero index.

Finally, track initiation takes place at the first time a non-zero index appears in a measurement sequence, which contains at least two true measurements. Each dummy report *after* track initiation is a missed detection.

Let us illustrate all this by reconsidering the example described in Fig. 1. We have $Z = \{\{0, A_1, A_2\} \times \{0, B_1\} \times \{0, C_1, C_2\} \times \{0, D_1, D_2, D_3\} \times \{0, E_1, E_2, E_3\} \times \{0, F_1, F_2, F_3\}\}$. The three tracks indicated in Fig. 1 are: $\{A_2, B_1, C_2, D_2, 0, F_3\}$, $\{0, 0, C_1, D_3, E_2, F_1\}$ and $\{0, 0, 0, D_1, E_3, F_2\}$.

2.2. Variables and constraints

For each $\mathbf{z} \in Z^*$ we introduce a 0–1 decision variable, which is defined as follows.

$$x_z = \begin{cases} 1 & \text{if track } z \text{ is selected,} \\ 0 & \text{otherwise.} \end{cases} \tag{4}$$

Notice that this choice of decision variables implies that no two measurements from the same scan can be assigned to the same track. Consider now the following linear constraints.

$$\sum_{\mathbf{z} \in Z^*, \mathbf{z}(k) = \mathbf{z}^k_{i_k}} x_z = 1 \quad \text{for all } k = 1, \ldots, N \text{ and } i_k = 1, \ldots, M_k. \tag{5}$$

These constraints imply that each *true* measurement must belong to precisely one track. It is straightforward to verify that any integral x satisfying constraints (5) yields a solution to the DAP and, vice versa, any feasible solution to the DAP satisfies these constraints.

2.3. Objective function

In order to complete the model we need to specify an objective function, or more exactly we need to compute a cost-coefficient for each track $\mathbf{z} \in Z^*$. To do this we follow an approach described in [9].

We define a quality measure for a track $\mathbf{z} \in Z^*$ in the form of a likelihood function Q_z. In analogy with Reid's ranking of MHT hypotheses [2], the core part of this function is a composite score of how good each plot $z^k_{i_k} \in z$ fits the target's assumed dynamical model (e.g. targets move in a straight line with constant velocity). When a plot exactly fits this model, it is located at the predicted position. Since the measurements are noise contaminated, the deviation between the predicted position and the plot itself will follow the assumed normal distribution for the measurement noise. The corresponding probability density function is denoted by f_δ. For a track's state prediction we use an extended Kalman filter. Furthermore, we assume false alarms and new tracks to be uniformly distributed over the sensor's coverage. The corresponding probability density functions are denoted by f_φ and f_ν, respectively.

Under these assumptions, the likelihood Q_z for some track $\mathbf{z} = \{\mathbf{z}^1_{i_1}, \ldots, \mathbf{z}^N_{i_N}\}$ is given by the following equation [9]:

$$Q_z = \prod_{k=1}^{N} (P^k_\phi)^{\Delta_{i_k}} \left\{ \left[\frac{P_\mathrm{d} f^k_\delta(\mathbf{z}^k_{i_k} | \mathbf{z})}{\lambda_\varphi f^k_\varphi(\mathbf{z}^k_{i_k})} \right]^{\delta^k_{i_k}} \cdot \left[\frac{\lambda_\nu f^k_\nu(\mathbf{z}^k_{i_k} | \mathbf{z})}{\lambda_\varphi f^k_\varphi(\mathbf{z}^k_{i_k})} \right]^{\nu^k_{i_k}} \right\}^{(1-\Delta_{i_k})}, \tag{6}$$

where

$$\Delta_{i_k} = \begin{cases} 1, & i_k = 0 \text{ (dummy report),} \\ 0 & \text{otherwise,} \end{cases}$$

$$P^k_\phi = \begin{cases} 1 - P_\mathrm{d}, & \mathbf{z}^k_{i_k} \text{ is a detection,} \\ 1 & \text{otherwise,} \end{cases}$$

$$v_{i_k}^k = \begin{cases} 1, & \mathbf{z}_{i_k}^k \text{ initiates a track,} \\ 0 & \text{otherwise,} \end{cases}$$

$$\delta_{i_k}^k = \begin{cases} 1, & \mathbf{z}_{i_k}^k \text{ proceeds an existing track,} \\ 0, & \text{otherwise,} \end{cases}$$

P_{d} is the probability of detection, λ_φ the expected number of false alarms (Poisson distributed), λ_v the expected number of new targets (Poisson distributed), $f_v^k = f_\varphi^k = 1/\pi r^2$, where r is the sensor's range,

$$f_\delta^k = \frac{\mathrm{e}^{-1/2[\mathbf{z}_{i_k}^k - h(\mathbf{s}(t_k))]^{\mathrm{T}} B^{-1} [\mathbf{z}_{i_k}^k - h(\mathbf{s}(t_k))]}}{\sqrt{(2\pi)^n |B|}},$$

n is the dimension of measurement vector, $h(\cdot)$ the transformation of Cartesian to polar coordinates, $\mathbf{s}$ the predicted state vector and B the covariance matrix of $\mathbf{z}_{i_k}^k - h(\mathbf{s}(t_k))$.

Notice the division in (6) by the probability of all measurements being false alarms. In this way, the assignment to false alarms serves as a reference for the quality of the actual partition.

Now the likelihood of a single track $\mathbf{z}$ is known, the likelihood of a partition of Z^* into tracks is computed by the product of the individual track likelihoods. Notice that this product equals the score of a global hypothesis as defined by Reid [2]. The objective function we obtain looks as follows.

$$\text{Maximize} \prod_{\mathbf{z} \in Z^*} Q_z x_z. \tag{7}$$

In order to complete the desired IP model, a linear objective function is required. Instead of maximizing the product of track likelihood values, we want to minimize the sum of track cost values. Let the cost of track $\mathbf{z}$ be defined as

$$c_z = -\ln Q_z. \tag{8}$$

Notice that $0 < Q_z < 1$ yields positive cost values and $Q_z > 1$ results in negative cost values. The division by the probability of all measurements being false alarms, as stated by Eq. (6), yields that false alarms have zero cost, since then $Q_z = 1$. Thus, for tracks with positive cost values it is better to assign all measurements to false alarms. The goal is now to minimize the sum of the cost of all tracks in a feasible partition. The complete model for the DAP after N scans can now be formulated as follows:

$$\begin{aligned} &\text{Minimize} \sum_{\mathbf{z} \in Z^*} c_z x_z \\ &\text{Subject to} \\ &\sum_{\mathbf{z} \in Z^*, \mathbf{z}(k) = \mathbf{z}_{i_k}^k} x_z = 1 \qquad \text{for all } k = 1, \ldots, N, \\ &\qquad\qquad\qquad\qquad \text{and } i_k = 1, \ldots, M_k. \\ &\qquad x_z \in \{0, 1\} \end{aligned} \tag{9}$$

Notice that in contrast to previous formulations (see e.g. [9], [12]), variables have a single index and there are no longer multiple summations present in the formulation.

3. A solution method for the DAP

Of course, in practice model (9) has to be solved after each scan. This requirement has certain implications for the way an instance of the DAP is solved. First, there is a need to be efficient. More concrete, the time to solve an instance should not surpass the rotation time of the radar; thus, a solution is required fast. This motivates a heuristic approach. Second, there is a difference in importance when considering the assignment of plots in early scans versus the assignment of plots in more recent scans. These implications motivate a technique, called the K-scan sliding window technique (see [9]), which indeed reduces the computational burden, and does not reconsider the assignment of plots in early scans. Let us now explain this technique in more detail.

Assume that an integer $K \geqslant 2$ is given and suppose that a DAP instance is to be solved for a new scan $N+1$. Then the K-scan window ($N \geqslant K$) "slides" one position ahead and embraces the scans $N-K+2,\dots,N+1$. The *retained region* is now defined as the set of scans $N-K+2,\dots,N$ that were already present within the window *before* it was moved to its new position. Scan $N-K+1$ now is no longer part of the window and is added to the *discarded region*, i.e. the set of scans outside the sliding window. The assignment of measurements in scan $N-K+1$ which was found after scan N becomes now fixed and will not change in future scans. In other words, when using the K-scan sliding window we fix the assignment of plots in scans $1,2,\dots,N-K$, depending upon the solution of the DAP after scan N. Thus, we no longer solve model (9), instead our approach follows the following pattern.

For $N = 1,2,\dots,K+1$ we solve model (9) heuristically (we describe in Section 3.2 how). Then we can define

$$S^{K+1} = \{\mathbf{z} \in Z^* \mid \mathbf{z} \textit{ is in the solution found after scan } K+1\} \tag{10}$$

and

$$Z_{K+1}(0) = \{(\mathbf{z}^1_{i_1},\mathbf{z}^2_{i_2}) \mid \exists \mathbf{z}^3_{i_3},\mathbf{z}^4_{i_4},\dots,\mathbf{z}^{K+1}_{i_{K+1}}, \text{ with } (\mathbf{z}^1_{i_1},\dots,\mathbf{z}^{K+1}_{i_{K+1}}) \in S^{\mathrm{K}+1}\}. \tag{11}$$

Thus, we identify the pairs of plots from $Z(1)\times Z(2)$ that are assigned to the same track in a solution found after $K+1$ scans. In the K-scan sliding window technique these particular assignments are now permanent. Any solution found from now on will feature these pairs. Even more, the assignment found between the pairs $(\mathbf{z}^1_{i_1},\mathbf{z}^2_{i_2})$ from $Z_{K+1}(0)$ and the measurements from $Z(3)$ in the solution found after $K+2$ scans remains permanent thereafter, and so on. Mathematically, we express it as follows. Let, using (11)

$$Z^{\kappa} = Z_N(0) \times Z(N-K+1) \times \cdots \times Z(N). \tag{12}$$

Notice that an element $\mathbf{z} \in Z^{\kappa}$ is a vector consisting of $K+1$ entries such that $\mathbf{z}(2)$ refers to a measurement from $Z(N-K+1)$, $\mathbf{z}(3)$ refers to a measurement from $Z(N-K+2)$, up to $\mathbf{z}(K+1)$ referring to a measurement from $Z(N)$ and $\mathbf{z}(1)$ refers to an element from $Z_N(0)$, i.e. a tuple of $N-K$ measurements. When we reindex the elements of $Z_N(0)$ using $\mathbf{z}^0_{i_0}$, $i_0 = 1,\dots,\max_{k=1,\dots,N-K} M_k \equiv M_0$

(i.e. the number of fixed assignments) we have the following model:

$$
\begin{aligned}
&\text{Minimize} \quad \sum_{\mathbf{z}\in Z^{\kappa}} c_z x_z\\
&\text{Subject to}\\
&\sum_{\substack{\mathbf{z}\in Z^{\kappa},\\ \mathbf{z}(k-N+K+1)=\mathbf{z}^k_{i_k}}} x_z = 1 \qquad \text{for all } k = N-K+1,\dots,N,\\
&\qquad\qquad\qquad\qquad \text{and } i_k = 1,\dots,M_k.\\
&\sum_{\mathbf{z}\in Z^{\kappa},\mathbf{z}(1)=\mathbf{z}^0_{i_0}} x_z = 1, \qquad i_0 = 1,\dots,M_0,\\
&x_z \in \{0,1\}.
\end{aligned}
\tag{13}
$$

We solve this model for $N \geqslant K+2$. Notice that a K-scan sliding window results in a $K+1$ dimensional assignment problem.

3.1. Problem size reduction

The size of the solution space Z^{κ} is reduced by preventing implausible tracks to be found. In other words, some tracks are so unlikely that the corresponding variables are never considered explicitly. First, this is achieved by performing the *gating* test for each track-plot pair after each scan [2]. Only if the plot fits the track's predicted state well enough, a new track is formed. The criterion for this goodness of fit test is called a *gate* or *validation* region and is given by

$$[\mathbf{z}^k_{i_k} - h(\mathbf{s}(t_k))]^T B^{-1} [\mathbf{z}^k_{i_k} - h(\mathbf{s}(t_k))] \leqslant \eta^2, \tag{14}$$

where $[\mathbf{z}^k_{i_k} - h(\mathbf{s}(t_k))]$ is the difference between the position of the actual plot and a target's predicted position. B is the covariance matrix of $[\mathbf{z}^k_{i_k} - h(\mathbf{s}(t_k))]$. Since f^k_δ is normally distributed, the gate is based on the χ^2 distribution. For instance, a 99% validation region for 2D positional measurements (implying 2 degrees of freedom) is obtained by $\eta^2 = 9.21$.

Second, we allow no track to have more than p (a prespecified parameter) subsequent missed detections. Consequently, if no plot has been associated with a track hypothesis for more than p scans, the track is considered terminated.

Finally, for the IP problem instance we can further decrease the number of decision variables by excluding all track hypotheses with positive cost values. This is allowed since there always exists a better assignment of lower costs for the measurements within such tracks, namely the zero-cost false alarm assignment. Note that track hypotheses with positive cost values are not excluded from the solution space, since they can grow better when future plots are received.

3.2. A greedy rounding procedure (GRP)

In this subsection, we discuss how a DAP problem instance, in the form of model (13), is solved with an LP[10,12] based rounding procedure called GRP.

By changing the integrality constraint $x_z \in \{0, 1\}$ to $0 \leqslant x_z \leqslant 1$ we obtain the LP relaxation of the IP problem. An LP instance can now be solved by the Simplex algorithm [15]. If the optimal solution $\mathbf{x}_r$ of the LP-relaxation of model (13) is fully integer-valued (in this case all decision variables will either have the value 0 or 1) then the solution $\mathbf{x}_r$ is optimal for model (13) [16, pp. 167].

In case one or more decision variables are not integer valued, we apply a procedure called greedy rounding procedure (GRP) (compare e.g. [17]). It selects iteratively a fractionally valued decision variable and sets it to 1. The consequences of this change are propagated over the other decision variables, in such a way that the constraints are not violated. In this manner, a feasible (but not necessarily optimal) solution for the original IP is constructed.

Greedy Rounding Procedure (input $\mathbf{x}_r$)
$R := Z^{\kappa}$
while $R \neq \emptyset$ **do**
  $\mathbf{z}^* := \arg\max_{\mathbf{z} \in R} x_z$
  $I(\mathbf{z}^*) := \{\mathbf{z} \in R \mid \exists j, \text{ with } \mathbf{z}(j) \neq \mathbf{z}_0^l \text{ for some } l \text{ and } \mathbf{z}(j) = \mathbf{z}^*(j)\}$
  $x_{z^*} := 1$
  for all $\mathbf{z} \in I(\mathbf{z}^*)$ **do**
    $x_z := 0$
  $R := R \setminus I(\mathbf{z}^*)$

Thus, the input for the greedy rounding procedure is the solution $\mathbf{x}_r$ of the LP-relaxation of model (13). If $\mathbf{x}_r$ contains decision variables with fractional solutions then GRP chooses the variable with the highest value lower than 1 and sets it to 1. Let this variable be x_{z^*}. The selection criterion for x_{z^*} is based on the assumption that a variable with a high fractional value is more likely to be part of the optimal integral solution than a variable with a value closer to zero.

If there is no unique variable, i.e. a tie occurs, then the one with lowest cost is selected.

A consequence of setting some variable x_z to 1 is, that other variables have to be set to 0 in order to respect the constraints. In fact, the set $I(\mathbf{z})$ defined above equals the set of tracks that share a true measurement with a measurement in $\mathbf{z}$. Thus, when x_z is set to 1, all variables in $I(\mathbf{z})$ must be set to 0.

Once a variable is forced to a certain value, it is not allowed to change any more. To achieve this, all rounded variables and all implicated variables are discarded from the rounding procedure. In this way, GRP will never set the value of a variable twice. This deletion of variables also applies to the initial LP solution, i.e. all variables with value 1 and all zero-valued variables implicated by them, are removed. The greedy rounding algorithm repeats the actions of selection, rounding and deletion until there are no variables left. The outcome will then be a feasible solution to model (13).

Observe that the GRP can be implemented to run efficiently, however it is clear that there is no guarantee that this heuristic procedure yields solutions close to the IP's optimal objective function value. However, the experiments discussed in the following section show that GRP *does* result in acceptable plot-to-track assignments of which the objective function value is close to the optimum value.

4. Experiments

The layout of the experiments is as follows. The simulated sensor is a single scanning surveillance radar that provides 2D positional data (i.e. range and bearing) within a range of 50 km. We constructed 35 scenarios divided over 4 groups called A, B, C and D. Groups A, B and C consist of 10 scenarios each while group D consists of five scenarios. In each scenario targets are observed for 30 scans. The duration of a scan is 8 s. The measurement noise is set to $\sigma_{\mathrm{r}} = 0.015$ km and $\sigma_{\mathrm{b}} = 0.0052$ rad for range and bearing respectively. Furthermore, the value of P_{d} (the probability of detecting a target) is assumed to be 0.90.

Targets are assumed to move in a straight line with a constant uniformly distributed random velocity between 0.1 and 1 km/s for groups A, B, C, and between 0.1 and 0.5 km/s for group D. The process noise on this model is assumed to be 0.05 m/s^2.

The initial positions of the targets in scenarios of groups A, B and C are uniformly distributed over a square area of 100 km × 100 km, while targets in scenarios of group D are uniformly distributed over a square area of 3 km × 3 km. Notice that targets can enter and leave the sensor's coverage.

Initially, before the first scan, there are no targets present. The number of new targets that appear in each scan is Poisson distributed with an average of $\lambda_v = 1$ for scenarios of groups A–C and with an average of $\lambda_v = 3$ for scenarios of groups D. The value of p (the number of subsequent missed detections allowed in a track) was set to 2.

The average number of false alarms per scan λ_φ (Poisson distributed) is set to 1, 5, 25 and 2 for the scenarios of groups A, B, C and D, respectively. Clearly, the scenarios in group A will result in only a few plots for each scan, while group C contains much harder scenarios. The scenarios of group D are the hardest to solve since targets are closely spaced, meaning that almost each plot-track pair is eligible for association.

Fig. 2 depicts a typical scenario of group C. A plot is indicated by a cross. Actual target trajectories are represented by interconnected plots. The remaining plots are false alarms. In Fig. 2 the detections of 30 scans are superimposed.

For all 35 scenarios we apply the K-scan sliding window technique. For scenarios of groups A–C we use five different widths of the K-scan sliding window, namely $K = 3$, 4, 5, 6 and 7. For scenarios of group D we use $K = 3$. Table 1 gives an overview of the parameter settings in each of the scenarios.

In total, this results in (5 × 30 scenarios × 30 scans) + (1 × 5 scenarios × 30 scans) = 4650 IP problem instances. In principle, the number of variables in such an instance can be as large as 22×10^9 for a 7-scan sliding window and an average of 30 measurements per scan. However, by applying the gating test, this number is reduced to about 1400.

We solve these instances as described in Section 3, i.e. we compute the LP-relaxation and apply the GRP to obtain a feasible solution to instances of model (13). We also compute the optimum integral values for these instances with a Branch & Bound algorithm. All these computations were carried out on a *Sun Ultra SPARC 2* machine with 256 mb of RAM using a public domain software package *LP Solve* (version 1.5, November 1994).[1]

[1] The package was obtained from ftp://ftp.ics.ele.tue.nl/pub/lp_solve.

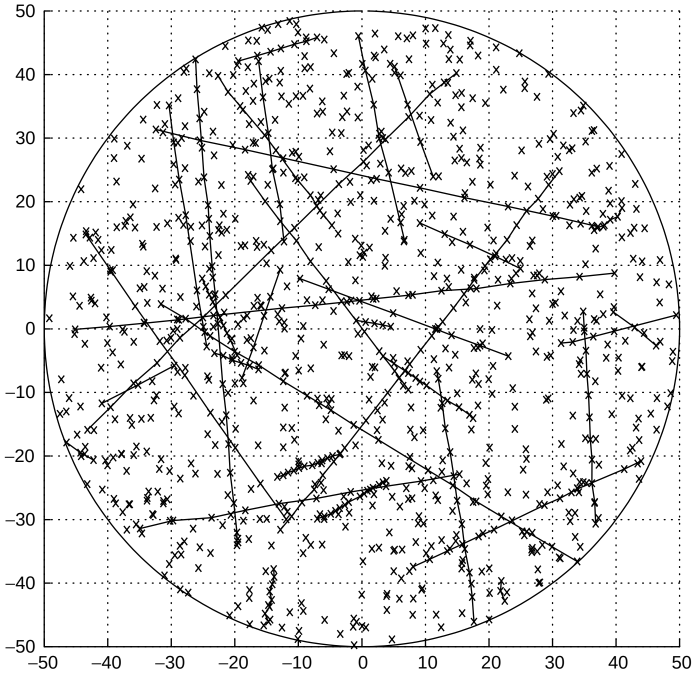

Fig. 2. A scenario of group C (30 scans superimposed).

Table 1
Parameter settings

Scenario	A	B	C	D
Number of scans	30	30	30	30
P_d	0.90	0.90	0.90	0.90
Speed of target	$U[0.1, 1.0]$	$U[0.1, 1.0]$	$U[0.1, 1.0]$	$U[0.1, 0.5]$
Area of target's appearance	100×100	100×100	100×100	3×3
λ_v	1	1	1	3
λ_φ	1	5	25	2
K	3, 4, 5, 6, 7	3, 4, 5, 6, 7	3, 4, 5, 6, 7	3

The results are shown in Tables 2–5. For each scenario three sub-rows exist. The first subrow (R) indicates the number of times (out of 30 scans) that an LP relaxed problem instance is *not* solved to integrality immediately, i.e. GRP's rounding mechanism had to recover a feasible integer valued solution. Second, the maximum difference between the optimum value and the value of the GRP solution is shown in the second subrow (Δ). If no roundings are performed, the maximum difference has no meaning. This is denoted by a "—". Finally, the value of the optimum solution (found by Branch & Bound) after thirty scans is shown in the third sub-row (O).

With respect to the topic of running times we restrict ourselves here to the following general remarks. Solving the LP relaxation and applying GRP took in general 0.07 s for a simple scenario

Table 2
Results GRP vs. Branch & Bound group A

Scenario		K 3	4	5	6	7
1	R	0	1	1	1	1
	Δ	—	0	0	0	0
	O	−1149.837	−1152.998	−1152.998	−1152.998	−1152.998
2	R	0	0	0	0	0
	Δ	—	—	—	—	—
	O	−2322.716	−2322.716	−2322.716	−2322.716	−2322.716
3	R	5	5	4	4	4
	Δ	0	0	0	0	0
	O	−1944.926	−1945.080	−1947.479	−1947.479	−1947.479
4	R	0	1	1	1	1
	Δ	—	0	0	0	0
	O	−1149.837	−1152.998	−1152.998	−1152.998	−1152.998
5	R	0	0	0	0	0
	Δ	—	—	—	—	—
	O	−2071.957	−2071.957	−2073.534	−2073.534	−2073.534
6	R	2	3	4	5	5
	Δ	0	0	0	0	0
	O	−1974.222	−1974.222	−1981.077	−1981.077	−1981.077
7	R	4	2	2	2	2
	Δ	3.370	1.946	1.946	1.946	1.946
	O	−2971.537	−2987.063	−2987.063	−2987.063	−2988.080
8	R	2	2	2	2	2
	Δ	0	0	0	0	0
	O	−2324.850	−2324.850	−2324.850	−2324.850	−2324.850
9	R	3	3	3	3	3
	Δ	0	0	0	0	0
	O	−1637.577	−1649.329	−1649.329	−1649.329	−1649.329
10	R	3	3	2	2	2
	Δ	0	0	0	0	0
	O	−1778.245	−1776.503	−1779.865	−1780.948	−1780.948

(3-scan sliding window, group A), 1.7 s for a harder scenario (7-scan sliding window, group C) up to 4.5 s the hardest scenario in group D. The running times for computing the optimum solution by means of Branch & Bound were about 0.09, 2.26 up to 5.7 s, respectively. Notice that the latter

Table 3
Results GRP vs. Branch & Bound group B

Scenario		K				
		3	4	5	6	7
11	R	2	4	4	5	6
	Δ	0.969	1.007	1.008	1.008	1.009
	O	−1757.197	−1763.268	−1765.636	−1765.636	−1765.636
12	R	1	1	1	1	1
	Δ	0	0	0	0	0
	O	−1485.215	−1488.597	−1488.597	−1488.597	−1488.597
13	R	3	4	4	5	6
	Δ	0.87	1.057	1.057	1.057	1.057
	O	−1882.858	−1886.774	−1886.774	−1886.774	−1886.774
14	R	2	2	2	2	2
	Δ	0	0	0	0	0
	O	−1826.735	−1826.735	−1826.735	−1826.735	−1826.735
15	R	1	1	1	1	1
	Δ	0	0	0	0	0
	O	−1549.809	−1552.625	−1552.625	−1552.625	−1552.625
16	R	0	0	0	0	0
	Δ	—	—	—	—	—
	O	−1345.276	−1345.276	−1345.276	−1345.276	−1345.276
17	R	3	4	5	6	7
	Δ	0	0	0	0	0
	O	−1462.369	−1463.135	−1463.135	−1463.135	−1463.135
18	R	1	1	4	4	4
	Δ	0	0	2.568	2.568	2.568
	O	−1585.837	−1585.892	−1590.316	−1590.316	−1590.316
19	R	2	2	2	2	2
	Δ	0	0	0	0	0
	O	−2088.965	−2091.325	−2091.325	−2091.325	−2091.325
20	R	1	1	1	1	1
	Δ	0	0	0	0	0
	O	−2105.915	−2120.993	−2125.458	−2125.458	−2125.458

algorithm does not give any guarantees on maximum running times; in the worst case the entire solutions space is searched.

Table 4
Results GRP vs. Branch & Bound group C

Scenario		K				
		3	4	5	6	7
21	R	1	1	1	0	0
	Δ	0	0	0	—	—
	O	−1600.534	−1607.869	−1607.869	−1611.367	−1611.367
22	R	0	0	0	0	0
	Δ	—	—	—	—	—
	O	−1579.554	−1588.361	−1588.383	−1588.383	−1588.383
23	R	0	0	0	0	0
	Δ	—	—	—	—	—
	O	−1049.066	−1048.773	−1051.028	−1050.967	−1051.601
24	R	0	0	0	0	0
	Δ	—	—	—	—	—
	O	−1593.235	−1593.235	−1596.823	−1597.449	−1597.449
25	R	3	4	5	6	6
	Δ	0	0	0	0	0
	O	−1032.891	−1045.471	−1046.388	−1046.419	−1046.419
26	R	0	1	2	3	3
	Δ	—	0	0	0	0
	O	−1519.282	−1522.409	−1527.799	−1527.277	−1527.797
27	R	0	3	3	3	4
	Δ	—	0.8	0.84	0.84	0.84
	O	−860.817	−863.922	−863.922	−876.911	−876.911
28	R	0	0	3	4	4
	Δ	—	—	0	0	0
	O	−1804.592	−1813.740	−1821.480	−1823.615	−1828.225
29	R	0	0	0	0	0
	Δ	—	—	—	—	—
	O	−1750.232	−1774.398	−1774.398	−1774.398	−1774.398
30	R	0	0	0	0	0
	Δ	—	—	—	—	—
	O	−1520.259	−1521.250	−1522.693	−1522.693	−1526.000

5. Discussion

The experiments show that there is a distinction between scenarios of groups A, B and C on the one hand and scenarios of group D on the other hand. Indeed, 4225 out of the 4500 (94%)

Table 5
Results GRP vs. Branch & Bound group D

Scenario		K
		3
31	R	13
	Δ	6.01
	O	−4278.040
32	R	11
	Δ	3.35
	O	−4295.400
33	R	11
	Δ	8.82
	O	−3497.922
34	R	16
	Δ	8.87
	O	−4339.604
35	R	15
	Δ	7.35
	O	−3908.557

IP problem instances from scenarios of groups A, B and C are solved by their LP relaxation, i.e. the Simplex algorithm immediately yields integer valued solutions. For the remaining 275 problem instances, the GRP, recovers a feasible (integer valued) solution from the fractional valued solution of the LP relaxation. For these instances, GRP succeeds in recovering an optimal solution in 195 cases, i.e. the maximum difference in objective function value equals zero. For the remaining 75 instances the recovered solutions are near-optimal with a maximum deviation of 3.37 (scenario 7, $K = 3$). Compared to the absolute objective function value, this difference is marginal. From these results it can be concluded that the GRP procedure performs quite satisfactory. Moreover, it follows from these experiments that raising the number of false alarms does not increase difficulties with respect to solving these instances.

When considering the 150 instances from scenarios from group D (see Table 5), it turns out that only 84 out of the 150 (56%) IP problem instances have an LP relaxation that is integral. Also, the difference between the value of the LP relaxation and its integer optimum has increased somewhat compared to scenarios from other groups, although GRP still results in near-optimal solutions (within 1%).

Let us now address the issue of computation times. Since the majority of instances of scenarios from groups A–C have integer valued LP-relaxations, we focus here on the instances of scenarios from group D. In Tables 6 and 7 we collected for the 5 scenarios of group D the instances for which the LP relaxation is fractional. For each scan in which the LP relaxation results in non-integer solutions, the number of fractional valued variables is shown (column indicated by F). The subsequent columns show the computation time (ms) of Branch & Bound (t_{BB}), LP relaxation (t_{LP}) and

Table 6
Results GRP vs. Branch & Bound group D (scenarios 31–33)

Scenario	Scan	F	t_{BB}	t_{LP}	t_R	$\frac{t_{BB}}{t_{LP}+t_R}$	O	Δ
31	5	24	1660	150	30	6.44	−194.66	4.53
	8	12	1560	1220	150	1.14	−590.49	4.44
	10	25	2430	1530	290	1.34	−950.90	6.01
	11	10	1880	1510	150	1.13	−1122.80	4.16
	12	17	1390	1350	40	1.00	−1324.57	1.66
	13	13	2580	2110	80	1.18	−1514.03	3.00
	14	15	2380	2130	80	1.08	−1702.83	1.33
	15	8	2080	1840	50	1.10	−1905.50	1.84
	17	8	1710	1320	30	1.27	−2275.98	1.63
	18	7	1390	1030	30	1.31	−2463.47	1.90
	23	5	510	470	10	1.06	−3147.65	0.00
	24	8	390	370	10	1.03	−3310.35	0.28
	27	23	2440	2050	310	1.04	−3791.40	1.55
32	10	7	160	120	20	1.14	−772.52	0.00
	13	9	380	210	10	1.72	−1126.42	0.00
	16	25	2210	810	60	2.48	−1557.57	2.69
	17	15	1160	1090	50	1.02	−1751.15	1.90
	20	19	2110	1630	120	1.21	−2273.23	0.88
	21	6	1220	1000	20	1.20	−2462.58	0.87
	22	10	2130	1060	50	1.92	−2638.91	0.82
	23	11	1690	1570	100	1.01	−2798.67	3.35
	24	29	3950	2060	320	1.65	−2977.56	2.30
	25	17	5690	2750	120	1.98	−3181.42	1.91
	27	4	4560	4470	120	0.99	−3651.98	0.99
33	12	12	3060	1310	40	2.26	−911.89	1.75
	13	14	940	570	50	1.51	−1051.29	1.16
	15	3	1440	1230	30	1.14	−1353.07	0.63
	16	4	650	600	10	1.07	−1526.28	0.00
	17	19	2050	1460	120	1.22	−1682.64	1.58
	19	20	1430	1390	150	0.93	−1982.99	8.82
	21	4	540	460	10	1.15	−2268.75	0.18
	22	8	590	450	30	1.23	−2381.74	2.30
	23	28	870	530	60	1.47	−2518.28	0.69
	27	15	1140	1060	140	0.95	−3055.02	3.81
	30	6	1170	1080	30	1.05	−3497.92	0.34

rounding (t_R) respectively. The ratio between computation times of Branch & Bound and GRP is given by $t_{BB}/(t_{LP}+t_R)$. The last two columns show the objective function value of the optimal solution (O) and the distance from GRP's solution to this optimum (Δ).

Concerning computation time GRP performs equally good or better than Branch & Bound. GRP is 50% faster than Branch & Bound in about 20% of the times rounding had to be performed and 20%

Table 7
Results GRP vs. Branch & Bound group D (scenario 34, 35)

Scenario	Scan	F	t_{BB}	t_{LP}	t_R	$\frac{t_{BB}}{t_{LP}+t_R}$	O	Δ
34	3	7	20	30	0	0.67	−50.50	0.00
	7	6	240	190	10	1.20	−306.88	0.85
	8	7	330	310	30	0.97	−421.35	1.83
	9	8	1110	750	60	1.37	−557.23	1.99
	10	12	2590	2280	130	1.07	−732.56	2.15
	11	12	1510	1320	70	1.09	−907.82	0.00
	13	3	1440	1360	20	1.04	−1303.65	0.49
	14	3	2620	1490	10	1.75	−1445.97	0.19
	15	10	3310	1040	50	3.04	−1592.21	0.34
	16	13	2070	1100	40	1.82	−1170.55	1.26
	17	4	2520	2430	40	1.02	−1941.34	0.55
	18	3	2210	2150	20	1.02	−2154.68	6.66
	19	3	2240	2240	20	0.99	−2344.84	6.66
	22	7	1690	1480	30	1.12	−2913.26	2.25
	25	21	1480	1360	240	0.92	−3481.10	8.87
	29	7	1510	450	20	3.21	−4166.92	4.60
35	3	5	40	50	0	0.80	−68.42	0.00
	4	12	160	150	20	0.94	−144.93	6.13
	5	10	200	140	10	1.33	−224.67	0.14
	12	11	1420	420	20	3.22	−1024.30	2.75
	14	3	300	290	0	1.03	−1322.10	0.00
	18	25	2330	600	60	3.53	−1928.71	7.35
	19	3	800	590	0	1.36	−2092.90	0.21
	20	15	730	670	60	1.00	−2257.53	2.43
	21	3	1120	1050	10	1.06	−2436.57	0.00
	22	13	890	740	30	1.16	−2603.45	2.93
	23	4	1100	1060	0	1.04	−2784.79	0.09
	24	14	700	630	30	1.06	−2953.88	1.82
	28	9	980	880	20	1.09	−3555.98	2.06
	29	15	1460	1260	80	1.09	−3714.72	7.20
	30	18	1520	940	110	1.45	−3908.56	0.83

faster in about 50% of the times that rounding had to be performed. In some cases, performance gain may be over 300%. (Occasionally, the ratio $t_{BB}/(t_{LP}+t_R)$ drops below 1. This can be explained by system noise during the performance measurements). However, it is perhaps not so much the gain in time that is interesting. Instead, the importance of GRP stems from the fact that it offers a guarantee of limited running times. When using an algorithm for the data association problem it is of utmost importance to be able to quickly have a solution available (see Section 3).

An interesting question is the following: How well does model (13), and in particular the cost-coefficients of this model, capture the DAP? In a situation where the true tracks are known, we can give a (partial) answer to this question as follows. Ideally, one would want that the objective function value of the solution given by the true tracks (say V_T) coincides with the optimum of model (13) (say OPT). Of course, it is possible that OPT $< V_{\mathrm{T}}$ (then OPT contains tracks that yield a better solution value than the true tracks), and also OPT $> V_{\mathrm{T}}$ is possible (in which case, due to gating or the K-scan sliding window technique, the solution given by the true tracks has been eliminated from consideration). We observe that only in three cases OPT $> V_{\mathrm{T}}$. The deviations OPT $- V_{\mathrm{T}}$ are 33.830, 54.709 and 38.155 for scenario 10 (group A), 27 (group C) and 29 (group C) respectively. It can be safely concluded that the recovery of actual tracks is satisfactory.

When looking at the Branch & Bound objective function values after thirty scans, the maximum difference between any two sliding window widths is 16.505 for group A (scenario 7), 19.54 for group B (scenario 20) and 24.166 for group C (scenario 29). Compared to the absolute values, increasing the size of the sliding window yields only marginal improvements on the quality of the recovered tracks.

Furthermore it must be noticed that for group A the objective function value does not improve for $K \geqslant 5$, with the exception of some minor changes in scenario 7 and 10. A similar observation can be made for group B. The scenarios for group C however, still show a decrease in objective function value for $K = 7$ (scenarios 23, 28, 30). Because of the trade-off between computational demand and solution quality, a 5-scan sliding window seems a sensible choice for all scenarios considered in these experiments.

6. Conclusions

In this section, we formulate our conclusions. We have modeled the data association problem as an integer programming problem. We have described an LP-based algorithm using a K-scan sliding window to solve instances of the data association problem. This algorithm consists of solving the LP-relaxation of the associated integer programming formulation, and next applying a greedy rounding procedure called GRP. From the computational experiments that we performed it follows that

- GRP yields near-optimal solutions, using a computation time that is guaranteed to be limited.
- Increasing the parameter λ_{φ} (the average number of false alarms per scan) does not seem to increase the difficulty of solving the resulting instance.
- When targets originate from a relatively small area the problem instances become more difficult.
- There seems to be no gain in having the size of the sliding window more than 5.

Acknowledgements

The authors are indebted to J.N. Driessen and H.W. de Waard at Thales Naval Nederland for their valuable comments on this paper.

References

[1] Storms PPA, Spieksma FCR. An LP-based algorithm for the data association problem in multitarget tracking. Proceedings of the Third International Conference on Information Fusion, vol. 1:TuD2, 2000. p. 10–6.

[2] Reid DB. An algorithm for tracking multiple targets. IEEE Transactions on Automatic Control 1979;AC-24(6).
[3] Volgenant A, Tesselaar B, Absil FGJ. Clustering in doelvolgalgoritmen van radarsystemen. Kwantitatieve Methoden 2001;67:27–49 ((in Dutch)).
[4] Pierskalla WP. The multidimensional assignment problem. Operations Research 1968;16:422–31.
[5] Morefield L. Application of 0–1 integer programming to multitarget tracking problems. IEEE Transactions on Automatic Control 1977;AC-22:302–12.
[6] Bojilov LV. Some algorithms for solving assignment problem in multisensor multitarget tracking. Comptes rendus de l'Academie bulgare des Sciences 1994;47:65–8.
[7] Deb S, Yeddanapudi M, Pattipati K, Bar-Shalom Y. Generalised S-D assignment algorithm for multisensor-multitarget state estimation. IEEE Transactions on Aerospace and Electronic Systems 1997;33:523–38.
[8] Pattipati KR, Deb S, Bar-Shalom Y, Washburn RB. Passive multisensor data association using a new relaxation algorithm. In: Bar-Shalom Y, editor. Multitarget-multisensor tracking: advanced applications. Norwood: Artech House, 1990. p. 219–46.
[9] Poore AB. Multidimensional assignment formulation of data association problems arising from multitarget and multisensor tracking. Computational Optimization and Applications 1994;3:27–57.
[10] Poore AB, Rijavec N. A Langrangian relaxation algorithm for multidimensional assignment problems arising from multitarget tracking. SIAM Journal on Optimization 1993;3:554–63.
[11] Poore AB, Rijavec N. A numerical study of some data association problems arising in multitarget tracking. In: Hager WW, Hearn DW, Pardalos PM, editors. Large Scale optimization. Boston: Kluwer Academic Publishers, 1994. p. 347–70.
[12] Poore AB, Robertson AJ. A new multidimensional data association algorithm for multisensor-multitarget tracking. SPIE 1995;2561:448–59.
[13] Storms PPA. A multidimensional assignment approach to multitarget tracking. Master's Thesis, Maastricht University, 1998.
[14] Queyranne M, Tardella F, Spieksma FCR. A general class of greedily solvable linear programs. Mathematics of Operations Research 1998;23:892–908.
[15] Dantzig GB. Linear programming and extensions. Princeton, NJ: Princeton University Press, 1963.
[16] Parker RG, Rardin RL. Discrete optimization. New York: Academic Press, 1988.
[17] Miller JL, Franz LS. A binary rounding heuristic for multi-period variable-task duration assignment problems. Computers and Operations Research 1996;23:819–28.

Patrick P.A. Storms is a consultant at Acklin B.V. (http://www.acklin.nl) and is specialized in intelligent software agent systems. Formerly he was a designer of naval Command & Control systems at Thales Naval Systems Nederland (http://www.thales-nederland.nl).

Frits C.R. Spieksma is a Professor at the Department of Applied Economics of the Katholieke Universiteit Leuven. His research interests concern combinatorial optimization problems and their applications (http://www.econ.kuleuven.ac.be).