

# AN ACCELERATED IMM-JPDA ALGORITHM FOR TRACKING MULTIPLE MANEUVERING TARGETS IN CLUTTER

Ljudmil BOJILOV, Kiril ALEXIEV  
and Pavlina KONSTANTINOVA

## 1. Introduction

By far, the most complicated case in target tracking is to track multiple maneuvering targets in heavy clutter. Numerous methods and algorithms have been devoted to this problem and for any one of them *pros* and *cons* can be pointed out. Theoretically, for example, the MHT method is known to be the most powerful approach to tracking multiple maneuvering targets in clutter. This method, however, very often leads to combinatorial explosion and computational overload that restricts its implementation. Recently, numerous papers have been devoted to algorithms capable to compute a ranked set of assignments of measurements to targets. Such algorithms allowed for the first practical implementations of MHT the approach.

Another and much less complicated approach, especially for tracking maneuvering targets, is the Multiple Models (MM) approach. The most promising algorithm based on this approach is the Interacting Multiple Models (IMM) algorithm. At the price of some sub-optimality of its framework, this algorithm reaches best implementation in terms of speed and stability. However, in the presence of clutter the IMM algorithm most often fails. In the case of cluttered environment, the PDA (and JPDA) approaches can be implemented. When tracking multiple closely spaced targets, the JPDA algorithm can be implemented successfully even in the presence of heavy clutter. In a recent paper <sup>1</sup> we proposed an algorithm unifying features of the IMM and the JPDA algorithms. That algorithm proved to be good alternative to the MHT approach for clusters containing up to 4 targets and moderate level of clutter. However, when the number of targets in the cluster exceeds this limit the total number of all feasible hypotheses increases exponentially. In this paper we propose an extension of the algorithm in our previous work. <sup>1</sup> Instead of enumeration of all

feasible hypotheses we propose to use ranked assignment approach to find the first K-best hypotheses only. The value of K has to ensure that the weight of scores-sum of these K-best hypotheses prevails over the total sum.

This paper is organized as follows. In the next section we elaborate on our motivation and formulate the problem. The IMM\_JPDA algorithm is briefly described and the need of its extension is discussed. In the 3<sup>rd</sup> section the extended algorithm is described. The emphasis is on the extension of the algorithm. The 4<sup>th</sup> section presents simulation results. These results show that the extended algorithm performs better than the IMM\_JPDA algorithm in terms of speed, while at the same time preserves stability of tracking.

## **2. Motivation and Problem Formulation**

When several closely spaced targets form a cluster, the JPDA algorithm starts to generate all feasible hypotheses and to compute their scores. The set of all feasible hypotheses includes such hypotheses as ‘null’ hypothesis and all its derivatives. The consideration of all possible assignments including the ‘null’ assignments is important for optimal calculation of assignment probabilities.<sup>6</sup> If, for example, the score of every one of these hypotheses differs from any of the others by no more than one order of magnitude, it should not be possible to truncate some significant parts of all hypotheses. If, however, the prevailing share of the total score is concentrated in a small percent of the total number of all hypotheses, then the interest in considering only this small percent of all hypotheses becomes very high.

In order to investigate this idea, a typical example with five closely spaced targets with overlapping validation regions and shared measurements is used. In the first run (or first scenario) 17 measurements are disposed in the target gates, and in the second run (second scenario) 9 measurements are disposed. At every run all feasible hypotheses are generated and their scores are computed and summarized. The results are presented on figures 1 and 2. These two examples were chosen out of numerous experiments as typical for the algorithm performance.

The two plots of Figure 1 show how the individual scores of the sorted feasible hypotheses are distributed. Only the top six percents of all hypotheses for the first and second scenario are depicted on the figure. It can be seen that the scores of the hypotheses dramatically reduce their values. Even more informative is Figure 2, where the cumulative score’s distributions of the two scenarios are given. This figure confirms our expectations that only a small number of hypotheses concentrate the prevailing part of their total sum. One additional conclusion can be derived. The first scenario is much more complicated with more than 4930 hypotheses generated. In the second scenario, the generated hypotheses are approximately 550. It can be seen from

the figures that for the more complicated cases the expected effect stands out even more definitely.

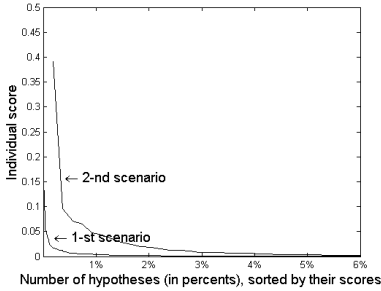


Figure 1: Hypotheses' score distribution.

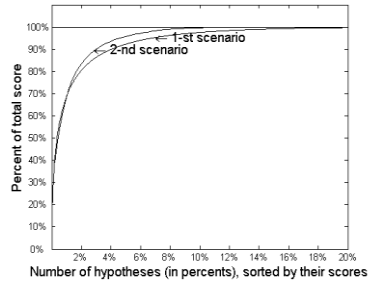


Figure 2: Cumulative score distribution.

The description of the algorithm proposed in our previous work<sup>1</sup> follows. For simplicity and without losing generality two models are assumed.

### 2.1. IMM-JPDA Algorithm Description

The IMM JPDA algorithm starts with the same step as IMM PDA algorithm,<sup>5</sup> but in cycle for every particular target in the cluster.

**Step 1.** Computation of the mixed initial conditions  $\hat{x}_i^{0t}$  for every target  $i$  and for the filter, matched to model  $t$ :

- a) mixed state estimate

$$\hat{x}_i^{0t}(k-1|k-1) = \sum_{s=1}^2 \hat{x}_i^s(k-1|k-1) \mu_{s|t}^i(k-1|k-1), \quad t=1,2 \quad (1)$$

Here, it is supposed that mixing probabilities  $\mu_{s|t}^i$  are already computed.

- b) mixed covariance estimate

$$P_i^{0t}(k-1|k-1) = \sum_{s=1}^2 \mu_{s|t}^i(k-1|k-1) \left\{ P_i^s(k-1|k-1) + [\hat{x}_i^s - \hat{x}_i^{0t}] [\hat{x}_i^s - \hat{x}_i^{0t}]^T \right\} \quad (2)$$

Here  $P_i^s$  is covariance update of model  $s$  for target  $i$ .

Next, some JPDA steps follows.

Step 2. State predictions  $\hat{x}_i^{0r}(k|k-1)$  and covariance predictions  $P_i^{0r}(k|k-1)$  for the next scan  $k$  for every target and for every model are calculated.

Step 3. In this step, after receiving the set of measurements at scan  $k$ , a clustering is performed. Further on, it is assumed that the algorithm will proceed with every particular cluster.

At this point, in the traditional JPDA algorithm, hypotheses generation has to be performed. However, to avoid combinatorial explosion we include here our innovation.

Step 4. Calculating ‘predicted model probabilities’:

$$\mu_i^t(k|k-1) = \sum_{s=1}^2 p_{st} \mu_i^s(k-1), \quad (3)$$

where  $\mu_i^t(k-1)$  is the probability that the model  $t$  is correct at scan  $(k-1)$  and  $p_{st}$  are Markovian switching probabilities.

Now, the individual model state predictions are merged for every particular target:

$$\hat{x}_i^0(k|k-1) = \sum_{t=1}^2 \hat{x}_i^{0r}(k|k-1) \mu_i^t(k|k-1). \quad (4)$$

Step 5. We are now ready to continue with the hypotheses generation and hypotheses score computation. Hypotheses generation is another combinatorial problem that will be discussed in the next section.

After generating all feasible hypotheses, hypothesis probability is computed by the expression

$$P^r(H_l) = \beta^{[N_M - (N_T - N_{nD})]} (I - P_D)^{N_{nD}} P_D^{(N_r - N_{nD})} g_{ij} \dots g_{mm}, \quad (5)$$

where

$\beta$  - is probability density for false returns,

$$g_{ij} = \frac{e^{-\frac{d_{ij}^2}{2}}}{(2\pi)^{M/2} \sqrt{|S|}} - \text{is probability density that measurement } j \text{ originates from}$$

target  $i$ , and the following additional notations are used:  $N_M$  - total number of measurements in the cluster,  $N_T$  - total number of targets,  $d_{ij}$  - statistical distance,  $N_{nD}$  - number of not detected targets. The step ends with the standard normalization

$$P(H_l) = \frac{P'(H_l)}{\sum_{l=1}^{N_H} P'(H_l)}, \quad (6)$$

where  $N_H$  is the total number of hypotheses.

Step 6. In this step, association probabilities are calculated. To compute for a fixed  $i$  the probability  $p_{ij}$  that observation  $j$  originates from track  $i$ , we have to take a sum over the probabilities of those hypotheses in which this event occurs:

$$p_{ij} = \sum_{l \in L_j} P(H_l), \quad \text{for } j = 1, \dots, m_i(k) \text{ and } i = 1, \dots, N_T, \quad (7)$$

where  $L_j$  is a set of indices of all hypotheses, which include the event mentioned above,  $m_i(k)$  is the number of measurements falling in the gate of target  $i$ , and  $N_T$  is the total number of targets in the cluster.

Step 7. After association probabilities computation, the JPDA algorithm continues as a PDA algorithm for every individual target. For every target the ‘merged’ combined innovation is computed

$$v_i(k) = \sum_{j=1}^{m_i(k)} p_{ij} v_{ij}(k). \quad (8)$$

Step 8. This is the last step of our description. At this step, our algorithm returns to the multiple model case by splitting ‘merged’ combined innovation from the previous equation. For every individual target and for every particular model the combined innovations are computed:

$$\mathbf{v}'_i(k) = \mathbf{v}_i(k) + H_i(k)\hat{\mathbf{x}}_i^0(k|k-1) - H_i'(k)\hat{\mathbf{x}}_i^{0r}(k|k-1). \quad (9)$$

The last few steps of this algorithm fully coincide with the well-known IMM PDA algorithm<sup>5</sup> and will be omitted from the current description.

### 3. Accelerating Extension to the IMM JPDA Algorithm

Our extension to IMM JPDA algorithm is directed to the most time consuming part of the algorithm, which concerns hypotheses generation and their scores computation. If we take as a simple example a cluster with 4 targets and 10 measurements distributed in their validation regions (Table 1), the total number of all feasible hypotheses for this example approaches 400. When, however, the number of targets in the cluster exceeds 5 or 6 and there are more than 15 measurements in their gates, the number of all hypotheses to be generated reaches thousands. To avoid these overwhelming computations we propose the next trade-off: to take into consideration only small part of all feasible hypotheses with highest scores and to concentrate on the prevailing share of the total score sum.

Table 1: Indices of the measurements falling in the gates of corresponding targets

T1	T2	T3	T4
0	0	0	0
4	6	3	1
8	7	4	2
9	8	5	3
		6	4
		9	

In order to find out the first K-best hypotheses we use an algorithm due to Murty<sup>2</sup> and optimized by Miller *et al.*<sup>3</sup> This algorithm gives a set of assignments to the assignment problem,<sup>4</sup> ranked in increasing order of cost. As a first step in solving this problem we have to define the cost matrix of the assignment problem. It can be seen that the score of any particular hypothesis (equation 5) is an expression of multipliers. The score of every feasible hypothesis, i.e., the probability of being true, can be calculated using a table similar to Table 1, but instead indices in the boxes of the Table 1 we need to put multipliers equal to probability of assigning the given measurement to the corresponding target (Table 2).

Table 2: Multipliers of the corresponding measurements

T1	T2	T3	T4
$\beta(1-P_D)$	$\beta(1-P_D)$	$\beta(1-P_D)$	$\beta(1-P_D)$
$g_{14}P_D$	$g_{26}P_D$	$g_{33}P_D$	$g_{41}P_D$
$g_{18}P_D$	$g_{27}P_D$	$g_{34}P_D$	$g_{42}P_D$
$g_{19}P_D$	$g_{28}P_D$	$g_{35}P_D$	$g_{43}P_D$
		$g_{36}P_D$	$g_{44}P_D$
		$g_{39}P_D$	

Now, properly combining indices from Table 1, thus generating every one of the feasible hypotheses we can at the same time multiply corresponding elements from Table 2, obtaining the score of the so generated hypothesis (equation 5). As it is well known, feasibility of hypothesis meets two important constraints: a) no target can create more than one measurement, and b) no measurement can be assigned to more than one target.

On the other side, every solution of the assignment problem represents a sum of elements of the cost matrix. We have to define this cost matrix in such way, that the value of every possible solution of the assignment can be potentially a score of some feasible hypothesis. Let us take logarithm from both sides of (5). From the left-hand side we obtain logarithm of hypothesis probability and, from the right-hand side, a sum of logarithms of elements from Table 2. This correspondence between multipliers in equation (5) and the sum of their logarithms gives a hint of how to construct the cost matrix and to solve the problem mentioned above.

We construct a cost matrix containing instead the elements of Table 2, their negative logarithms. If we find the optimal solution (in this particular case – the minimum) of the assignment problem with this cost matrix it will coincide with the hypothesis with highest probability, i.e., both the optimal solution and the highest probability hypothesis will associate the targets with the same measurements. The cost matrix of a cluster from Table 1 appears in Table 3.

Table 3: The cost matrix of the example

	<b>f1</b>	<b>F2</b>	<b>f3</b>	<b>f4</b>	<b>z1</b>	<b>z2</b>	<b>z3</b>	<b>z4</b>	<b>z5</b>	<b>z6</b>	<b>z7</b>	<b>z8</b>	<b>z9</b>
<b>T1</b>	$\ln^0$	×	×	×	×	×	×	$\ln_{14}$	×	×	×	$\ln_{18}$	$\ln_{19}$
<b>T2</b>	×	$\ln^0$	×	×	×	×	×	×	×	$\ln_{26}$	$\ln_{27}$	$\ln_{28}$	×
<b>T3</b>	×	×	$\ln^0$	×	×	×	$\ln_{33}$	$\ln_{34}$	$\ln_{35}$	$\ln_{36}$	×	×	$\ln_{39}$
<b>T4</b>	×	×	×	$\ln^0$	$\ln_{41}$	$\ln_{42}$	$\ln_{43}$	$\ln_{44}$	×	×	×	×	×

where

$$\ln^0 = -\ln[(1 - P_D)\beta], \quad \ln_{ij} = -\ln[g_{ij}P_D].$$

The symbol  $\times$  in the matrix represents one and the same value with only requirement to be greater than the greatest element out of the set of elements denoted with  $\ln$ . In order to use any of the widespread assignment algorithms, as well as the algorithm<sup>1</sup> for finding the K-best hypotheses, the matrix from Table 3 has to be added up to square matrix filling in the remaining rows with the same value  $\times$ . The first four columns of the matrix in Table 3 correspond to false measurements, i.e., assigning first row to first column, the second row to the second column, etc., means that no measurement originated from this target. Columns from five to thirteen represent the corresponding measurements falling in the validation regions of the targets.

When the algorithm for finding K-best assignments is initiated it will find K solutions of the problem with lowest sums of negative log-likelihood (or with highest probabilities). After receiving these K values their anti-logarithms have to be computed in order to obtain the K-best hypotheses probabilities. Next, these probabilities have to be normalized by equation (6), but now the sum is up to K:

$$P(H_l) = \frac{P'(H_l)}{\sum_{l=1}^K P'(H_l)}.$$

Henceforth, this algorithm fully coincides with the algorithm described in the previous section, continuing with the [step 6](#).

One important practical question, closely related to the proposed approach, arises in this regard: how many hypotheses K to be found out. When deciding on the value of K we have to realize that this value has to be optimal in some sense. On one hand, the smaller the value of K, the proposed algorithm performs faster. On the other hand, however, too small values of K can lead to distortion in assignment probabilities computation (equation 7). This question will be discussed in the next section.

#### 4. Simulation Results

We compare the algorithm presented in this paper with the same algorithm without acceleration discussed in previous section (our previous algorithm<sup>1</sup>). These two algorithms were tested extensively on a variety of scenarios involving different numbers of maneuvering and closely spaced targets and in presence of heavy clutter. We construct a set of scenarios with 3, 4 and 5 targets in a cluster and in the presence



of moderate and heavy clutter. The scenarios are similar to those from our previous paper<sup>1</sup> where we searched for the limits of the IMM JPDA algorithm in terms of the number of targets in a cluster.

The first step in preparing the common frame for testing is to decide how many K-best hypotheses need to be generated. We mentioned in the end of the previous section that the value of K has to be, in some sense, optimal so that: a) it is sufficiently small to ensure acceleration of the algorithm, and, in the same time, b) it is not too small to lead to distortion in computing assignment probabilities.

As it can be seen from Figure 1, the scores of feasible hypotheses decrease very rapidly and some 5-10 percents of them (Figure 2) cover more then 95 percents of the total score sum. However, as we know neither the total number, nor the total sum, we try to derive indirect criterion for determining the value of K. One possible expression can be

$$H(n) - H(n+1) < \alpha \cdot H(n),$$

where  $\alpha \ll 1$ . Here with  $H(n)$  the probability density of  $n^{\text{th}}$  hypothesis to be true is denoted. The implementation of this criterion, however, did not give stable results. The reason is that very often there are subsets of hypotheses with very close values of their scores, even in the beginning of the sorted hypotheses array. Another expression, providing for higher stability, is

$$H(n) < \alpha \cdot H(1). \quad (10)$$

In order to tune experimentally the value of  $\alpha$ , a range of experimental runs have been carried out. Every one run is performed with scenario with the same number of 6 targets and 12 measurements but with different reciprocal (relative) location. Averaging over 1000 runs, the following simulation results have been received (Table 4).

The first column of the Table 4 contains the different values of  $\alpha$ , the second and third columns contain respectively the mean and the largest number (worst case) of the first N-best hypotheses in accordance with (10). The fourth and fifth columns contain mean and lowest values of the ratio of the total score sum of these N-best hypotheses. In opposite to the hypotheses' number, the worst case for this ratio is its lowest value.

Table 4: Number of hypotheses and ratio of the total score sum for different values of  $\alpha$  as per equation (10)

$\alpha$	$N_{mean}$	$N_{wst}$	$R_{mean}$	$R_{wst}$
0.05	32	506	0.7842	0.6185
0.01	179	1510	0.9414	0.8587
0.005	286	2074	0.9690	0.9184
0.001	632	3350	0.9942	0.9784
0.0005	779	3797	0.9973	0.9899
0.0001	1082	4459	0.9995	0.9982

Now, we can choose the most suitable value for  $\alpha$ . For example, if we choose the value of  $\alpha = 0.005$ , after summation of the first 286 hypotheses we ensure, in average, the attainment of nearly 97 percent of the total score sum. If we take into account that the mean of the total hypotheses number for this experiment is 9780 we can conclude, that choosing the value of  $\alpha = 0.005$  we can generate and process the first 3 percent of all feasible hypotheses ensuring 97 percent of the total score sum. Similar conclusions can be drawn for  $\alpha = 0.01$ . Consequent experiments confirm that the values 0.01 and 0.005 for  $\alpha$  are equally appropriate.

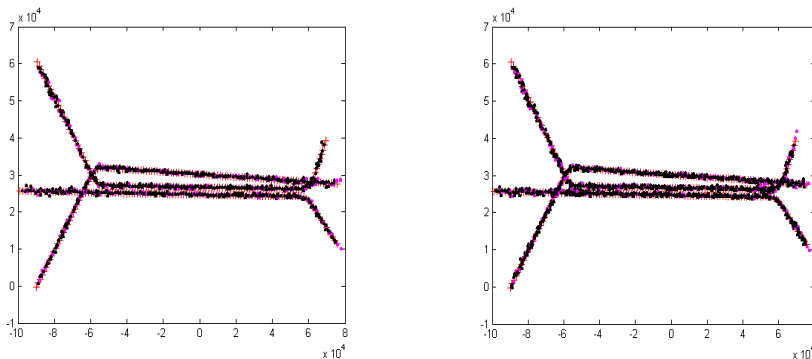


Figure 3: Three targets with crossing trajectories and Poisson parameter  $\beta V = 1$  for the left, and  $\beta V = 2$  for the right picture

To test further the presented algorithm we constructed a range of scenarios with increasing complexity in terms of number of targets and presence of clutter. The chosen scenarios include 3, 4 and 5 targets with closely spaced and crossing

trajectories (Figures 4 and 5). The included clutter has been modeled as a Poisson process with parameter  $\beta V$ , where  $B$  is spatial false alarm density and  $V$  is validation volume:

$$P(N = m_k | \beta V) = \frac{(\beta V)^{m_k} e^{-\beta V}}{m_k!}.$$

For every scenario two levels of clutter have been tested: with  $\beta V = 1$  to simulate moderate clutter, and  $\beta V = 2$  for heavy clutter. The results achieved can be summarized as follows:

**A. Scenario with 3 closely spaced targets.**

Table 5: Time per cluster in seconds for 3-targets scenario

	All hypotheses computation /targets in a cluster/		First K-best hypotheses only /targets in a cluster/	
	2 targets	3 targets	2 targets	3 targets
$\beta V = 1$	0.016	0.062	0.02	0.26
$\beta V = 2$	0.011	0.136	0.09	0.68

The comparison of presented algorithm with the algorithm where all feasible hypotheses are computed gives unexpected results – the latter algorithm spends less processing time (Table 5). Obviously the program frame for finding out the first K-best hypotheses is heavy and unsuitable for simple cases. Even so, both approaches give results far below the real time implementation threshold.

**B. Scenario with 4 closely spaced targets.**

Table 6: Time per cluster in seconds for 4-targets scenario

Targets in a cluster	All hypotheses computation Targets in a cluster		First K-best hypotheses only Targets in a cluster	
	3 targets	4 targets	3 targets	4 targets
$\beta V = 1$	0.03	3.94	0.79	3.39
$\beta V = 2$	0.22	124.7	3.42	9.86

It can be seen in this case (Figure 4) that when the scenario becomes denser the results become comparable (especially for clusters with 4 targets) and for the heaviest case ( $\beta V = 2$ ) the processing time for the first algorithm increases almost exponentially (Table 6). In the same time, the processing time for the new algorithm increases polinomially.

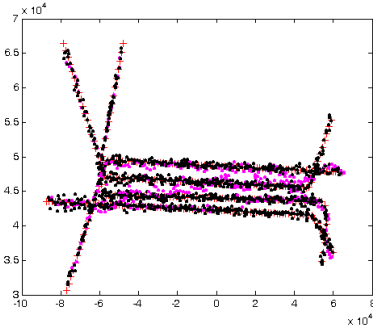


Figure 4: Four-target scenario with  $\beta V = 2$

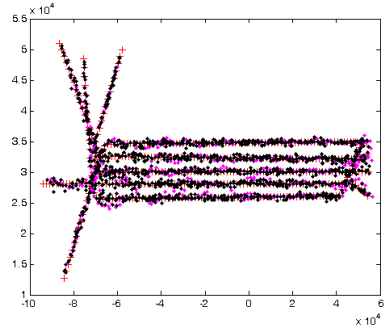


Figure 5: Five-target scenario with  $\beta V = 2$

C. Scenario with 5 closely spaced targets.

For this scenario (Figure 5) only the proposed algorithm has been tested. For the most dense case, when five closely spaced targets have to be tracked in heavy clutter we compute average time per scan  $t=8.7 sec$ . But as it can be seen from Table 7, when in a given scan all five targets fall into the cluster the processing time becomes twice the average time. It can be stated that this case is the limit for algorithm implementation.

Table 7: Time per cluster in seconds for 5-targets scenario

	First K-best hypotheses computation		
	Targets in a cluster		
Targets in a cluster	3 targets	4 targets	5 targets
$\beta V = 1$	0.35	1.16	8.2
$\beta V = 2$	1.58	6.36	15.4

## 5. Conclusions

In this paper a new algorithm is presented for tracking closely spaced targets in moderate and heavy clutter. This algorithm is an improved version of an algorithm previously presented earlier by the authors. Instead of all feasible hypotheses in the new algorithm only part of the hypotheses are generated. By means of an algorithm for finding the first K-best solutions of the assignment problem we generate the first K-best feasible hypotheses in terms of their probability of being true. This trade-off does not lead to observable assignment probability degradation and in the same time definitely speeds up the algorithm processing.

## Acknowledgment

The work on this paper was supported by the Center of Excellence BIS21 under Grant ICA1-2000-70016.

## Notes:

- <sup>1</sup> Ljudmil Bojilov, Kiril Alexiev and Pavlina Konstantinova, "An Algorithm Unifying IMM and JPDA Approaches," *Comptes Rendue de l'Academie Bulgare des Sciences* (to appear).
- <sup>2</sup> Katta G. Murty, "An Algorithm for Ranking All the Assignment in Order of Increasing Cost," *Operations Research* 16 (1968): 682-687.
- <sup>3</sup> Matt L. Miller, Harold S. Stone and Ingemar J. Cox, "Optimizing Murty's Ranked Assignment Method," *IEEE Transactions on AES* 33, 3 (July 1997): 851-862.
- <sup>4</sup> Roy Jonker and Ton Volgenant, "A Shortest Augmenting Path Algorithm for Dense and Sparse Assignment Problems," *Computing* 38 (1987): 325-340.
- <sup>5</sup> Yaakov Bar-Shalom, ed., *Multitarget-Multisensor Tracking: Advanced Applications*, (Norwood, MA: Artech House, 1990).
- <sup>6</sup> Samuel S. Blackman, *Multiple-Target Tracking with Radar Applications* (Norwood, MA: Artech House, 1986).

**LJUDMIL VLADIMIROV BOJILOV** see page 140.

**KIRIL METODIEV ALEXIEV** is assistant research professor at the Central Laboratory for Parallel Processing, Bulgarian Academy of Sciences. He received a M.Sc. degree from the Polytechnic Institute in Kiev in 1984 and a Ph.D. degree from the Technical University of Sofia in 1997. *E-mail*: alexiev@bas.bg.

**PAVLINA DIMITROVA KONSTANTINOVA** see page 106.