MOPSO Optimized Radar CBMeMBer Forward-Backward Smoothing Filter

Jiazheng Pei, Yong Huang, Yunlong Dong, and Xiaolong Chen(✉)

Naval Aviation University, Ermalu Road 188, Yantai, Shandong, China
roycerover@163.com, huangyong_2003@163.com,
china_dyl@sina.com, cxlcxl1209@163.com

Abstract. For the tracking of multiple maneuvering targets under radar observations, the Cardinality-Balanced Multi-Bernoulli based Sequential Monte-Carlo Filter (SMC-CBMeMBer) tracking algorithm gets its shortcomings that the estimation of number is inaccurate and the state estimation accuracy is degraded. This paper presents an improved tracking algorithm based on SMC-CBMeMBer smoothing filter. In the prediction process, the algorithm uses Multi-objectIve Particle Swarm Optimization (MOPSO), combined with the measured values at the current moment, to move the particles to the location where the posterior probability density distribution takes a larger value; Besides the smooth recursive method is used to smooth the filter value with multi-target measurement data, and the estimation accuracy of the algorithm is improved on the basis of sacrificing certain operation efficiency. The simulation results show that compared with the traditional filter and smoothing methods, the proposed algorithm performs better in terms of the accuracy of the estimation of the number of maneuvering targets and the accuracy of the target state estimation.

Keywords: Particle swarm optimization · Particle filter · CBMeMBer filter
Forward-backward smoothing

1 Introduction

Multi-target Multi-Bernoulli (MeMBer) filter [1] is another multi-target tracking method based on RFS proposed by Mahler after PHD [2] and CPHD [3] filter. For multi-objective nonlinear filtering, B-N Vo gives advantages of MeMBer [4] over the other two algorithms in terms of filtering accuracy and computational complexity, and on this basis, an improvement, i.e. Cardinality Balanced Multi-target Multi-Bernoulli (CBMeMBer) filter [5] was proposed to dispose the overestimation of the number of targets caused by MeMBer. The literature [6] proposed using Bernoulli random finite set to model the single-target motion state and complete forward-backward smoothing.

This work was supported in part by the National Science Foundation of China (U1633122, 61531020, 61501487), National Defense Science Foundation (2102024), the China Postdoctoral Science Foundation (2017M620862) and by the Special Funds of Taishan Scholars Construction Engineering, Young Elite Scientist Sponsorship Program of CAST.

L. Meng and Y. Zhang (Eds.): MLICOM 2018, LNICST 251, pp. 604–616, 2018.
https://doi.org/10.1007/978-3-030-00557-3_60

Bernoulli forward-backward smoothing improves the recognition accuracy of the disappearance of the target and the accuracy of the state estimation. In addition, it validates the feasibility of the Bernoulli filtering forward-backward smoothing filter.

Based on the known background, this paper proposes an improved multi-target tracking method based on CBMeMBer smoothing, which consists of forward filtering and backward smoothing. Forward filtering uses CBMeMBer filtering. Multi-objective particle swarm optimization is added between prediction and updating step. Particles move in the direction of higher posterior probability density; backward smoothing still uses CBMeMBer probability density to approximate the multi-objective smooth states, and obtain the backward recursive formula of CBMeMBer probability density parameter, thereby achieving multi-targets Backward recursive calculation of smooth state probability density. Finally, the simulation experiments show that the tracking performance of the newly proposed algorithm is better than the previous algorithm in the multi-objective maneuvering scenario.

2 CBMeMBer Smoothing Filter

In contrast to MeMBer filtering, the probability generation functional of CBMeMBer is more accurate than that of MeMBer in the updating process, which averts a potential cardinality deviation. The implementation of CBMeMBer filtering needs to satisfy the following assumptions: (1) the RFS of the newborn target state is formed by a multi-Bernoulli random finite set; (2) the clutter obeys the multi-target Poisson process, and the clutter density is not too large; (3) The target has a higher detection probability.

The CBMeMBer forward-backward smoothing filter utilizes more measurement information, and can improve the estimation performance of the target number and the targets' state.

2.1 Forward Filtering Process

Prediction. Suppose that Multi-objective posterior probability density at time $k-1$ is a form of multiple Bernoulli RFS.

$$\pi_{k-1} = \left\{ \left(r_{k-1}^{(i)}, p_{k-1}^{(i)} \right) \right\}_{i=1}^{M_{k-1}} \tag{1}$$

So as to the Multi-objective predict probability density is of the same form,

$$\pi_{k|k-1} = \left\{ \left(r_{P,k|k-1}^{(i)}, p_{P,k|k-1}^{(i)} \right) \right\}_{i=1}^{M_{k-1}} \cup \left\{ \left(r_{\Gamma,k|k-1}^{(i)}, p_{\Gamma,k|k-1}^{(i)} \right) \right\}_{i=1}^{M_{\Gamma,k}} \tag{2}$$

where

$$r_{P,k|k-1}^{(i)} = r_{k-1}^{(i)} \left\langle p_{k-1}^{(i)}, p_{S,k} \right\rangle \tag{3}$$

$$p^{(i)}_{P,k|k-1} = \frac{\left\langle f_{k|k-1}(x|\cdot), p^{(i)}_{k-1} p_{S,k} \right\rangle}{\left\langle p^{(i)}_{k-1}, p_{S,k} \right\rangle} \tag{4}$$

$f_{k|k-1}(\cdot|\xi)$ denotes single-target transfer probability at time k, $p_{S,k}$ denotes the survival probability. $\left\{ \left(r^{(i)}_{\Gamma,k|k-1}, p^{(i)}_{\Gamma,k|k-1} \right) \right\}_{i=1}^{M_{\Gamma,k}}$ refers to Bernoulli RFS newborn parameters at time k.

Update. The predicted multi-target density at time k is still in the form of multi Bernoulli $\pi_{k|k-1} = \left\{ \left(r^{(i)}_{k|k-1}, p^{(i)}_{k|k-1} \right) \right\}_{i=1}^{M_{k|k-1}}$, Then this posterior probability density can be represented by a multiple Bernoulli union.

$$\pi_k \approx \left\{ \left(r^{(i)}_{L,k}, p^{(i)}_{L,k} \right) \right\}_{i=1}^{M_{k|k-1}} \cup \left\{ \left(r_{U,k}(z), p_{U,k}(\cdot; z) \right) \right\}_{z \in Z_k} \tag{5}$$

Where

$$r^{(i)}_{L,k} = r^{(i)}_{k|k-1} \frac{1 - \left\langle p^{(i)}_{k|k-1}, p_{D,k} \right\rangle}{1 - r^{(i)}_{k|k-1} \left\langle p^{(i)}_{k|k-1}, p_{D,k} \right\rangle} \tag{6}$$

$$p^{(i)}_{L,k}(x) = p^{(i)}_{k|k-1}(x) \frac{1 - p_{D,k}(x)}{1 - \left\langle p^{(i)}_{k|k-1}, p_{D,k} \right\rangle} \tag{7}$$

$$r_{U,k}(z) = \frac{\sum_{i=1}^{M_{k|k-1}} \frac{r^{(i)}_{k|k-1} \left(1 - r^{(i)}_{k|k-1} \right) \left\langle p^{(i)}_{k|k-1}, \psi_{k,z} \right\rangle}{\left(1 - r^{(i)}_{k|k-1} \left\langle p^{(i)}_{k|k-1}, p_{D,k} \right\rangle \right)^2}}{\kappa_k(z) + \sum_{i=1}^{M_{k|k-1}} \frac{r^{(i)}_{k|k-1} \left\langle p^{(i)}_{k|k-1}, \psi_{k,z} \right\rangle}{1 - r^{(i)}_{k|k-1} \left\langle p^{(i)}_{k|k-1}, p_{D,k} \right\rangle}} \tag{8}$$

$$p_{U,k}(x; z) = \frac{\sum_{i=1}^{M_{k|k-1}} \frac{r^{(i)}_{k|k-1}}{1 - r^{(i)}_{k|k-1}} p^{(i)}_{k|k-1}(x) \psi_{k,z}(x)}{\sum_{i=1}^{M_{k|k-1}} \frac{r^{(i)}_{k|k-1}}{1 - r^{(i)}_{k|k-1}} \left\langle p^{(i)}_{k|k-1}, \psi_{k,z} \right\rangle} \tag{9}$$

$$\psi_{k,z}(x) = g_k(z|x) p_{D,k}(x) \tag{10}$$

Z_k denotes observation set, $g_k(\cdot|x)$ and $p_{D,k}(x)$ respectively refer to single-target measurement likelihood function and target detection probability given the state at time k. $\kappa_k(\cdot)$ is Poisson clutter intensity parameter.

2.2 Backward Smoothing Process

The purpose of smoothing is to use the data at time l to estimate the state value at time $k(l > k)$. Given the multi-Bernoulli smoothing density parameters $r_{k|l}$ and $p_{k|l}(\cdot)$, backward recursion is displayed as follow [7].

$$r_{k-1|l} = 1 - \left(1 - r_{k-1|l-1}\right)\left(\alpha_{B,k|l} + \beta_{B,k|l} \int \frac{p_{k|l}(\zeta)}{p_{k|k-1}(\zeta)} b_{k|k-1}(\zeta) d\zeta\right) \tag{11}$$

$$p_{k-1|l}(x) \propto p_{k-1|k-1}(x)\left(\alpha_{S,k|l}(x) + \beta_{S,k|l}(x) \int \frac{p_{k|l}(\zeta)}{p_{k|k-1}(\zeta)} f_{k|k-1}(\zeta|x) d\zeta\right) \tag{12}$$

where

$$\alpha_{B,k|l} = (1 - p_b)\frac{1 - r_{k|l}}{1 - r_{k|k-1}} \tag{13}$$

$$\beta_{B,k|l} = p_b \frac{r_{k|l}}{r_{k|k-1}} \tag{14}$$

$$\alpha_{S,k|l}(x) = \left(1 - p_{S,k|k-1}(x)\right)\frac{1 - r_{k|l}}{1 - r_{k|k-1}} \tag{15}$$

$$\beta_{S,k|l}(x) = p_{S,k|k-1}(x)\frac{r_{k|l}}{r_{k|k-1}} \tag{16}$$

2.3 Shortcomings of SMC-CBMeMBer Smoothing

Particle filter using Sequential Importance Sampling [8] (SIS) method, recursively sampling according to the weights of the particles to obtain an approximate distribution of posterior probabilities. An important flaw in resampling particles is the lack of particle diversity. When the observation information is so accurate that the peak of the likelihood function is narrow, the overlap space between the likelihood probability and the prior probability distribution is extremely limited. As an end, only a small fraction of overlapping particle weights will increase after the update. In addition, similar problems exist when the observation probability distributed at the tail of the prior distribution, since only a small part of the particles generated by the prior probability is located in the high likelihood region. It is very likely that the prediction result will lose important particles and miss good assumptions.

Another significant problem is that when faced with the actual initial state of the system is unknown, the state estimation of the system requires a large number of particles, which makes the calculation efficiency of particles greatly reduced.

3 MOPSO Optimized CBMeMBer Smoothing

In order to solve the problems described in Sect. 2.3 above, this chapter proposes an improved method that combines the particle swarm optimization algorithm with the CBMeMBer smoothing filter to improve the problem of the particle diversity.

3.1 Multi-objective Particle Swarm Optimization

Particle Swarm Optimization (PSO) is a cluster intelligent global optimization algorithm based on population search strategy jointly proposed by Kennedy and Eberhart [9].

In each iteration, the individual optimal solution is updated after one iteration of the particle p_j, and the global optimal solution in the entire particle swarm g_j are used, where $j = 1, 2, \cdots, n$, n is the maximum number of iterations.

Update each particle with the following speed update formula and position update formula, so that we can improve the position information carried by the particles,

$$v_i = wv_i + c_1 rand\left(p_j - x_i\right) + c_2 Rand\left(g_j - x_i\right) \tag{17}$$

$$x_{i+1} = x_i + rv_i \tag{18}$$

where *rand* and *Rand* are random numbers between 0 and 1, w is the inertia coefficient, c_1 and c_2 are positive numbers, x_i is the initial particle swarm state, v_i is the updated speed, and r is the constraint factor.

In the multi-objective optimization, this paper uses multi-objective particle swarm optimization (MOPSO). The MOPSO screens particles based on multiple objective functions by means of the NSGA-II algorithm [10]. The algorithm is an improvement over conventional genetic algorithms. The key steps are the following three processes.

(1) Non-dominated sort.

The algorithm stratifies the population based on individual non-inferiority levels. The individuals are included in the 1st front F_1, and they are given the non-dominated order $i_{rank} = 1$; Then continue to identify non-dominated solutions and repeat this until the whole population is assigned.

(2) Crowding distance calculation.

The individual crowding distances are calculated in order to be able to selectively rank individuals within the same non-dominant sequence i_{rank}. For different objective functions, we need to calculate repeatedly to obtain the individual's crowding distance.

(3) Recombination and selection.

Firstly, the entire population is placed in new descendants in descending order of non-dominated sorting i_{rank} until the population size exceeds the particle group N when placed in a certain layer F_j; secondly, filling is continued in descending order of individual crowding distances of F_j until the number of particle swarms reaches.

Besides, in order to avoid the problem of particle swarm optimization speeding up the convergence too quickly and causing a local optimal solution, this paper added a mutation mechanism to the algorithm to increase the diversity of particles.

3.2 Improved Smoothing Filter Optimized by PSO

The improved process is expressed as follow.

The whole algorithm can be illustrate by Fig. 1. Firstly, we need to initialize the essential parameters, such as existence probability r, particle state x and particle weight w. Then the parameters will be used to describe the state of targets which is estimated by a series of work: prediction, PSO optimization, updating and smoothing. It should be noted that the raw observation data is utilized in the PSO optimization and updating process. Secondly, we will prune the tracks whose existence probability r can not reach the threshold, merge the tracks which are adjacent to each other to one track. Finally, we resample the particles based in their weights. The existence probability r can be used to get the targets' number. The particles' weights and state can be calculated to estimate the targets' state and can be used to join in the next filter iteration. The detailed description is stated as follows.

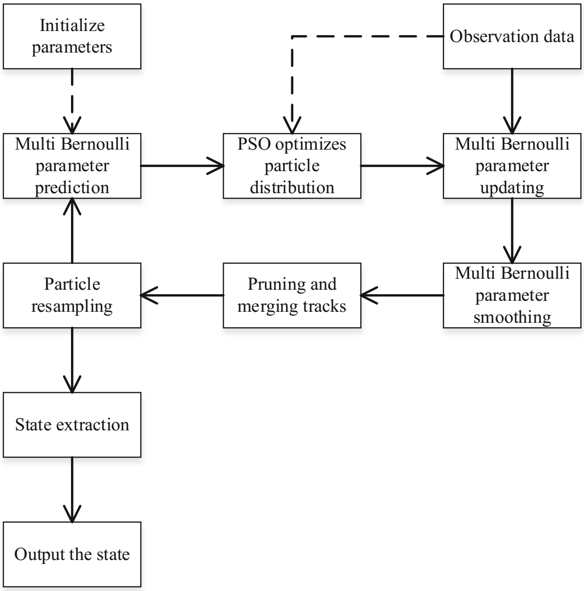
Initialize parameters
Observation data
Multi Bernoulli parameter prediction
PSO optimizes particle distribution
Multi Bernoulli parameter updating
Particle resampling
Pruning and merging tracks
Multi Bernoulli parameter smoothing
State extraction
Output the state

Fig. 1. The flow chart of PSO optimized CBMeMBer forward-backward smoothing

Prediction. Suppose that we have the posterior parameters at time $k-1$ are given with $r_{k-1|k-1}$ and $p_{k-1|k-1}$ in the form of particle filter:

$$p_{k-1|k-1} \approx \sum_{i=1}^{N_{k-1}} w_{k-1}^{(i)} \delta_{x_{k-1}^{(i)}}(x) \tag{19}$$

Then the prediction parameters can be displayed as:

$$r_{k|k-1} \approx p_b\left(1 - r_{k-1|k-1}\right) + r_{k-1|k-1} \sum_{i=1}^{N_{k-1}} w_{k-1}^{(i)} p_{S,k|k-1}\left(x_{k-1}^{(i)}\right) \tag{20}$$

$$p_{k|k-1}(x) = \sum_{i=1}^{N_{k-1}+J_k} w_{k|k-1}^{(i)} \delta_{x_k^{(i)}}(x) \tag{21}$$

where the particles and weights are generated as [7]:

$$x_k^{(i)} \sim \begin{cases} q_k\left(\cdot|x_{k-1}^{(i)}, z_k\right), \ i = 1 : N_{k-1} \\ s_k(\cdot|z_k), \ i = N_{k-1} + 1 : N_{k-1} + J_k \end{cases} \tag{22}$$

$$w_{k|k-1}^{(i)} = \begin{cases} \frac{r_{k-1|k-1}}{r_{k|k-1}} \frac{p_{S,k|k-1}\left(x_{k-1}^{(i)}\right) f_{k|k-1}\left(x_k^{(i)}|x_{k-1}^{(i)}\right) w_{k-1}^{(i)}}{q_k\left(x_k^{(i)}|x_{k-1}^{(i)}, z_k\right)}, i = 1 : N_{k-1} \\ \\ \frac{1 - r_{k-1|k-1}}{r_{k|k-1}} \frac{p_b b_{k|k-1}\left(x_k^{(i)}\right)}{J_k s_k\left(x_k^{(i)}, z_k\right)}, i = N_{k-1} + 1 : N_{k-1} + J_k \end{cases} \tag{23}$$

and p_b denotes probability of target newborn or reentry, $b_{k|k-1}\left(x_k^{(i)}\right)$ is spatial density of newborn process, q_k, s_k respectively denotes the survival density and newborn density.

Optimization. The particle swarm optimization algorithm is applied after the prediction step of the multi-Bernoulli filtering, combining the current time measurement, taking into account the distance between the particles and the measurement points. For the nth measurement, the objective function value of the ith particle is as follows.

$$fit_i^n = \exp\left[-\frac{1}{\sigma^2}\left(\boldsymbol{par}_n - \boldsymbol{meas}_i\right)^2\right], i = 1, \cdots, length(meas), n = 1, \cdots, N \tag{24}$$

where σ^2 denotes observation noise variance, ***meas*** denotes observation set, $\boldsymbol{meas}_i$ refers to the ith observation value, N is the total particle number. The particle swarm optimization algorithm motivates all particles to the Pareto optimal solution set by calculating the objective function value.

The optimized particle weights need to be redistributed by considering the particle's objective function values, and then normalized.

$$w_n = \sum\nolimits_i fit_i^n, w_n = \frac{w_n}{\sum_{n=1}^{N} w_n} \tag{25}$$

The optimization process makes the particles far away from the real state move to the areas which have higher posterior probability and improves the effect of each particle. Even when the initial state is unknown, the problem of particle filtering that requires a large number of particles for accurate state estimation is also attenuated.

Updating. After the optimization, the particle state, weight, and probability of existence $r_{k|k-1}$ participate in the update [11].

$$r_{k|k} = \frac{r_{k|k-1}\sum_{i=1}^{N_k}\tilde{w}_k^{(i)}}{1 - r_{k|k-1} + r_{k|k-1}\sum_{i=1}^{N_k}\tilde{w}_k^{(i)}} \tag{26}$$

$$p_{k|k}(x) \approx \sum_{i=1}^{N_k} w_k^{(i)} \delta_{x_k^{(i)}}(x) \tag{27}$$

$$\tilde{w}_k^{(i)} \propto l_k\left(z_k | x_{k|k-1}^{(i)}\right) w_{k|k-1}^{(i)} \tag{28}$$

In order to maintain the diversity of particles in the next smoothing process, the resampling operation is not performed after the update is completed, and is performed after the smoothing is completed.

Smoothing. Given $r_{k|l}$ and $p_{k|l} = \sum\limits_{i=1}^{N_{k|l}} w_{k|l}^{(i)} \delta_{x_{k|l}^{(i)}}(x)$, we

$$r_{k-1|l} \approx 1 - \left(1 - r_{k-1|\mathrm{k}-1}\right) \times \left(\frac{1 - r_{k|l}}{1 - r_{k|k-1}}(1 - p_b) + \frac{r_{k|l}}{r_{k|k-1}} p_b \sum_{j=1}^{N_{k|l}} w_{k|j}^{(j)} \frac{b_{k|k-1}\left(x_{k|l}^{(j)}\right)}{p_{k|k-1}\left(x_{k|l}^{(j)}\right)}\right) \tag{29}$$

$$p_{k-1|l}(x) \approx \sum_{i=1}^{N_{k-1|k-1}} \tilde{w}_{k-1|l}^{(i)} \delta_{x_{k-1|k-1}^{(i)}}(x) \tag{30}$$

$$\tilde{w}_{k-1|l}^{(i)} \propto \frac{1 - r_{k|l}}{1 - r_{k|k-1}}\left(1 - p_{S,k|k-1} w_{k-1|k-1}^{(i)}\right) + \frac{r_{k|l}}{r_{k|\mathrm{k}-1}} \sum_{j=1}^{N_{k|l}} p_{S,k|k-1} w_{k|l}^{(j)} \frac{f_{k|k-1}\left(x_{k|l}^{(j)} | x_{k-1|k-1}^{(i)}\right)}{p_{k|k-1}\left(x_{k|l}^{(j)}\right)} w_{k-1|k-1}^{(i)} \tag{31}$$

$$p_{k|k-1}\left(x_{k|l}^{(j)}\right) = \sum_{i=1}^{N_{k-1|k-1}} w_{k-1|k-1}^{(i)} f_{k|k-1}\left(x_{k|j}^{(j)} | x_{k-1|k-1}^{(i)}\right) \tag{32}$$

Track Pruning and Resampling. Only the track whose probability of existence gets greater than the threshold can be preserved. In the meanwhile, we need to select and reproduce particles with large weight values, i.e. resample the particle set.

$$\left\{x_i^k, \frac{1}{N}\right\}_{i=1}^{N} = \left\{x_i^k, w_i^k\right\}_{i=1}^{N} \tag{33}$$

4 Simulation

4.1 Maneuvering Target Cooperative Turning Model Establishment

In this CT model, the maneuvering target is a collaborative turn CT model. The state equation and measurement equation are as follows [12–14]:

$$X(k) = F(k)X(k-1) + \Gamma(k-1)v(k-1) \tag{34}$$

$$Z(k) = H(k)X(k) + W(k) \tag{35}$$

$$F(k) = \begin{bmatrix} 1 & \frac{\sin \omega T}{\omega} & 0 & \frac{\cos \omega T - 1}{\omega} & 0 \\ 0 & \cos \omega T & 0 & -\sin \omega T & 0 \\ 0 & \frac{1 - \cos \omega T}{\omega} & 1 & \frac{\sin \omega T}{\omega} & 0 \\ 0 & \sin \omega T & 0 & \cos \omega T & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{36}$$

$$\Gamma(k-1) = \begin{bmatrix} T^2/2 & T & 0 & 0 & 0 \\ 0 & 0 & T^2/2 & T & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}' \tag{37}$$

$$H(k) = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{bmatrix} \tag{38}$$

4.2 Experimental Simulation

Suppose that the radar was located in the origin point; the detection area was defined as [−2000 m, 2000 m] × [−500 m, 2000 m]; sampling interval $T = 1\,\text{s}$ and the total tracking time was set as 100 s. As for the CT model, the detection probability $p_{D,k} = 0.98$. Assume that there are multiple targets in the detection area for continuous motion within the detection area, and the ith target' state at time k was $\mathbf{x}_{k,i} = \left[x_{k,i}, y_{k,i}, \dot{x}_{k,i}, \dot{y}_{k,i}, \omega\right]^T$; the survival probability $p_{S,k} = 0.99$. Table 1 has set the initial state of each target and its starting and ending time, where the w was positive for a clockwise turn and negative for a counterclockwise rotation [15]. Figure 2 has displayed the true tracks of the simulation.

Table 1. The initial state, start and end time of targets

No	Start time (s)	End time (s)	Initial state
1	1	100	[1000, −10, 1500, −10, π/36]
2	10	100	[−250, 20, 1000, 3, −π/225]
3	10	100	[−1300, 11, 300, 10, −π/90]
4	10	70	[−1500, 43, 250, 0, 0]
5	20	80	[−250, 11, 750, 5, π/180]
6	40	100	[−500, −12, 1000, −12, −π/90]

In the particle swarm optimization process, the number of iterations $gen = 10$ and the fitness function threshold $Th_{fit} = 0.01$ are counted at a time t. In the state extraction process, the track pruning threshold $Th_{prune} = 10^{-3}$ and the maximum track number $T_{\max} = 100$ are set; the combined threshold $Th_{cap} = 10^{-5}$ of the track is set, the merging distance $D = 4\,\mathrm{m}$; the maximum number of particles $J_{\max} = 1000$, and the minimum $J_{\min} = 300$.

Assume that the location of clutter points is uniformly distributed in the detection area at each time, and its number obeys a poisson distribution with an average value of 20. Under the condition of a clutter density of $\lambda_c = 2 \times 10^{-6}\mathrm{m}^{-2}$.

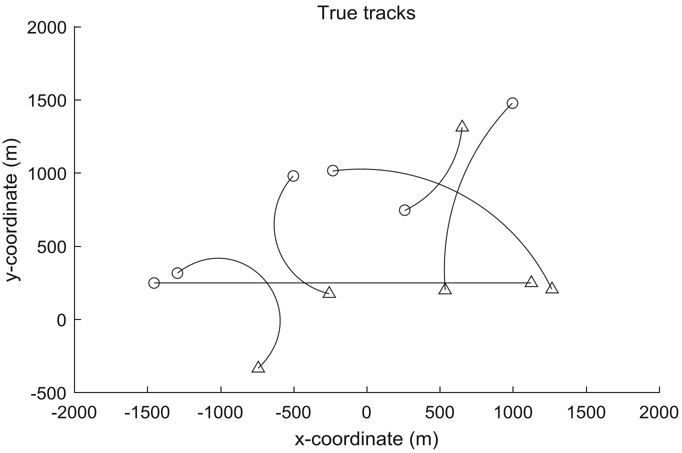

Fig. 2. Target actual movement status

After 100 times monte carlo experiments, the simulations found that all three can accurately estimate the number of targets, but the details can be seen at Fig. 3. when the number of targets changes, the new proposed algorithm is better than the general smoothing algorithm, both are better than the traditional filtering algorithm.

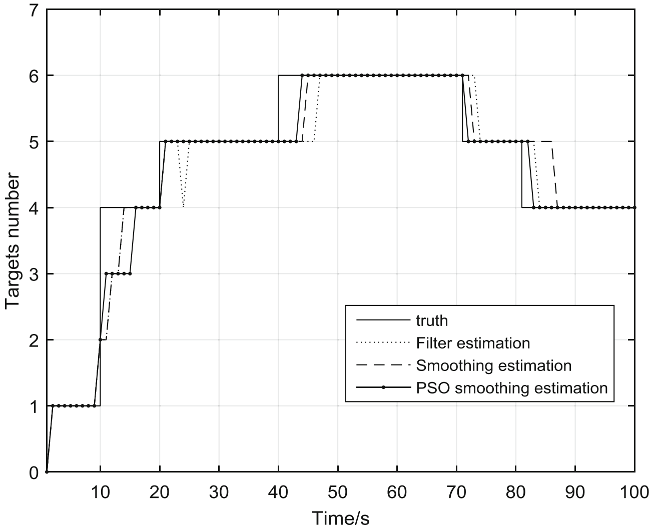

Fig. 3. Comparison of the number estimation of three tracking methods

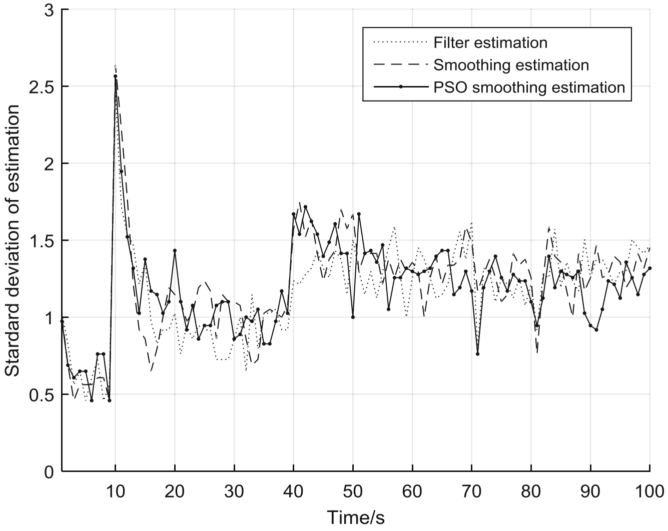

Fig. 4. Standard deviation of the three methods' estimation

From the Figs. 4 and 5, we can thoroughly analyze the standard deviation and OSPA error [16] of the three methods' estimation. In terms of specific estimation indicators, we can see that the two smoothing algorithms have better improvement effects than the traditional filtering methods in terms of estimation accuracy; while in the accuracy of the estimation number, the smoothing method has a certain degree of improvement over the filtering method. While the improved smoothing method again

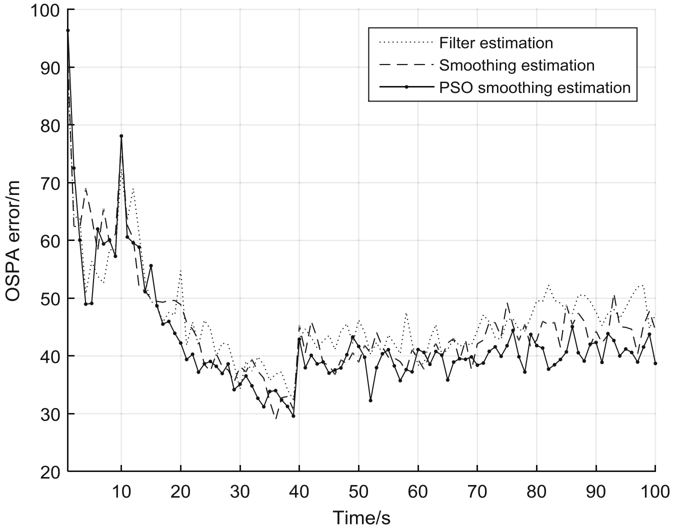

Fig. 5. OSPA error of the three methods' estimation

suppresses the standard deviation at the same level comparable to filtering. However, inevitably, the improved smoothing algorithm is longer in terms of running time than the former two kinds of algorithm. For more details, the numerical data was given in Table 2.

Table 2. The mean value of the standard deviation, OSPA error and running time

Mean value	CBMeMBer filter	CBMeMBer smoothing	PSO-CBMeMBer smoothing
Standard deviation	1.1606	1.2139	1.1882
OSPA error(m)	16.6089	15.0336	12.6689
Running time(s)	10.7	16.7	24.3

5 Summary

In order to deal with the tracking problem of multiple maneuvering targets, an SMC-CBMeMBer forward-backward smoothing filter with multi-objective particle swarm optimization was proposed. The simulation results show that the introduction of the particle swarm optimization algorithm improves the convergence performance of the CBMeMBer forward-backward smoothing filter algorithm for the maneuvering target, which improves the accuracy of the estimated number and position. In the next study, the predicted particles are selectively optimized, and the adaptive optimization is completed by setting the threshold of the objective function, focusing on improving the efficiency of the particle swarm optimization and saving the calculation cost.

References

1. Mahler, R.: Statistical Multisource-Multitarget Information Fusion, pp. 5–11. Artech House, Boston (2007)
2. Mahler, R.: Multitarget Bayes filtering via first-order multitarget moments. IEEE Trans. Aerosp. Electron. Syst. **39**(4), 1152–1178 (2003)
3. Mahler, R.: PHD filters of higher order in target number. IEEE Trans. Aerosp. Electron. Syst. **43**(4), 1523–1543 (2007)
4. Vo, B.T., See, C.M., Ma, N., et al.: Multi-sensor joint detection and tracking with the Bernoulli filter. IEEE Trans. Aerosp. Electron. Syst. **48**(2), 1385–1402 (2012)
5. Vo, B.T., Vo, B.N., Cantoni, A.: The cardinality balanced multi-target multi-Bernoulli filter and its implementations. IEEE Trans. Sig. Process. **57**(2), 409–423 (2009)
6. Vo, B.T., Clark, D., Vo, B.N., et al.: Bernoulli forward-backward smoothing for joint target detection and tracking. IEEE Trans. Sig. Process. **59**(9), 4473–4477 (2011)
7. Wong, S., Vo, B.T., Papi, F.: Bernoilli forward-backward smoothing for track-before-detect. IEEE Sig. Process. Lett. **21**(6), 727–731 (2014)
8. Liu, J.S., Chen, R., Logvinenko, T.: A theoretical framework for sequential importance sampling with resampling. In: Doucet, A., de Freitas, N., Gordon, N. (eds.) Sequential Monte Carlo Methods in Practice, pp. 225–246. Springer, New York (2001). https://doi.org/10.1007/978-1-4757-3437-9_11
9. Kennedy, J., Eberhart, R.: Particle swarm optimization. In: Proceedings of the IEEE International Conference on Neural Networks, pp. 1941–1948. IEEE Service Center, Piscataway (1995)
10. Deb, K., Agrawal, S., Pratap, A., Meyarivan, T.: A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: NSGA-II. In: Schoenauer, M., et al. (eds.) PPSN 2000. LNCS, vol. 1917, pp. 849–858. Springer, Heidelberg (2000). https://doi.org/10.1007/3-540-45356-3_83
11. Ouyang, C., Ji, H., Li, C.: Improved multi-target multi-Bernoulli filter. IET Radar Sonar Navig. **6**(6), 458–464 (2012)
12. Zhou, G., Pelletier, M., Kirubarajan, T., et al.: Statically fused converted position and doppler measurement Kalman filters. IEEE Trans. Aerosp. Electron. Syst. **50**(1), 300–318 (2014)
13. Yoon, J.H., Kim, D.Y., Bae, S.H., et al.: Joint initialization and tracking of multiple moving objects using doppler information. IEEE Trans. Sig. Process. **59**(7), 3447–3452 (2011)
14. Vo, B.N., Ma, W.K.: The Gaussian mixture probability hypothesis density filter. IEEE Trans. Sig. Process. **54**(11), 4091–4104 (2006)
15. Vo, B.T., Vo, B.N., Cantoni, A.: Analytic implementations of the cardinalized probability hypothesis density filter. IEEE Trans. Sig. Process. **55**(7), 3553–3567 (2007)
16. Ristic, B., Vo, B.N., Clark, D., et al.: A metric for performance evaluation of multi-target tracking algorithms. IEEE Trans. Sig. Process. **59**(7), 3452–3457 (2011)