22nd International Conference on Information Fusion
Ottawa, Canada - July 2-5, 2019

Real-time Ship Track Association: a Benchmark and a Network-Based Method

Lin Wu[a], Yongjun Xu[a], Fei Wang[a], Qiang Lu[a], Miao Hu[b]
[a]Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China
[b]Guangdong Key Laboratory of Big Data Analysis and Processing, Guangzhou, China

***Abstract*—Multiple systems are tracking ships independently, including cooperative as well as non-cooperative ones. Most track association works focused on association cost functions and optimal track matching matrix was determined by solving a multidimensional assignment problem, assuming synchronous detections of fixed number of targets received periodically from an ideal communication channel, which is not satisfied when tracking targets in a large scale by heterogeneous methods. We put forward an evolving target network-based association approach to create, update, merge, separate and destroy targets over time, which centers on targets rather than tracks. In the network, nodes are fused ships corresponding to tracks from one or more sources and weights of directed edges pointing to each node are association beliefs constituting its basic probability assignment (BPA) function in Dempster-Shafer theory (DST). Once a new message is received, this network evolves by combining the corresponding node's BPA with a new one calculated according to the message.**

This method is evaluated by an extensible benchmark consisting of an open source track simulator ICTShips and a free visualization software ICTShipVisual. The benchmark consists of four scenes with 32 to 102,231 targets observed by 3 sources at present, having GERs (Gap Error Ratio) from 0.787 to 1,412. Error ratios of our track association method under these scenes range from 0% to 0.539%.

Keywords—Track Association, Real-Time, Target Network, Dempster-Shafer Theory

I. INTRODUCTION

Considering that maritime transportation represents over 80% of total world merchandise trade by volume in 2015 [1], Maritime Situation Awareness (MSA) is essential for safety as well as economic benefits. The basis of MSA is real-time ship tracking. The term "real-time" indicates that systems must process messages at a rate faster than receiving them.

Nowadays, multiple systems are tracking ships independently, including cooperative ones like Automatic Identification System (AIS), ARGOS, BeiDou Navigation Satellite System (BDS) [2] and Inmarsat-C as well as non-cooperative ones like terrestrial and satellite-borne radars and remote sensing platforms. According to the International Convention for the Safety of Life at Sea [3], ships of 300 gross tons and upwards on international voyages, 500 tons and upwards for cargos not in international waters and passenger vessels are required to fit an AIS transceiver [4], which broadcasts kinematic, identity and voyage related information automatically. According to the regulation in China [5], deep sea fishing vessels must report their names, positions and timestamps at least 6 times per day using a device compatible with VMS, including: ARGOS, BDS and Inmarsat-C. As part of targets is not cooperative, terrestrial and satellite-borne radars and remote sensing platforms are also used to monitor targets. The encoding scheme of targets in each source is independent, such as MMSI in AIS and Beidou terminal ID in BDS. To form a unified picture, tracks using different encoding schemes need to be associated.

This work was supported in part by the National Natural Science Foundation of China (NSFC) [grant number 61602447]; Innovation Foundation of the Chinese Academy of Sciences [grant number CXJJ-17-M116]; and Opening Project of Guangdong Key Laboratory of Big Data Analysis and Processing [grant number 201804]. The authors gratefully acknowledge the support of K.C. Wong Education Foundation, Hong Kong.

II. RELATED WORKS

It is conventional to model track association as a multidimensional assignment problem in existed works [6][7][8], assuming synchronous data received periodically from multiple sources tracking fixed number of targets, which is not satisfied when monitoring targets in a large scale by heterogeneous sensors. As it is NP hard to solve the multidimensional assignment problem and need suboptimal solution via Lagrangian relaxation when there are more than 2 sources [9], researchers often simplify it to a situation with only 2 sensors in their simulation section [6][10][11][12][13][14][15]. Assuming that the same set of targets are being tracked by 2 sensors, an optimal matrix is solved every step by minimizing the sum of association cost between tracks:

$$\min_{H} \sum_{i=1}^{N} \sum_{j=1}^{N} C_{ij} H_{ij} \tag{1}$$

Where N is the number of tracks in each source, C_{ij} is the association cost between track i from source 1 and track j from source 2, H_{ij} is the indicator of whether this track pair is associated. As the focus is "track" and most efforts were put on cost functions between them, track association is often called "track-to-track association" [7][8][11][12][13][14][15][16].

Bar-Shalom and Chen considered a scenario with three sensors, but only 2 targets were simulated in evaluation [7]. Testa [17] and Mazzarella [18] considered the condition when missed detections exist in source S_1, and associated each track in S_1 to one track in S_2 who is tracking all targets. When systematic bias is present, Zhu et al. [19] and Tian [8] calculated the similarity between reference patterns as cost functions. In the case of asynchronous data, Sun et al. [20] searched potential matched data points under spatial and temporal constraints, which introduced extra complexity. Most works solved the track association problem in each step independently and is memoryless. To improve association accuracy, history information need to be used. Danu et al. [21] and Liu et al. [22] proposed association methods based on bithreshold, which confirmed track pairs when they had been matched for at least M times in the past N steps. Tokta et al. [14] and Tian et al. [23] proposed sliding window algorithms for calculating the test statistic using multiple frames of data.

978-0-9964527-8-6©2019 ISIF

TABLE I. PROBABILITY OF CORRECT ASSOCIATION IN LITERATURES ORDERED BY GER

Authors	Target count	Gap/km	StDev(pos)/meter	GER	Pr(correct)/%
Hambrick et al. [6]	6	3	0.1-0.5	6000	98-83
Zhu et al. [27]	20-50	45-28	10, 3.16	4500 - 2800	97-95
Zhu et al. [11]	5-50	13-4.2	60, 80	163-52.5	89-98
Yang et al. [25]	15, 30	7.7, 5.5	60	128, 91.7	96.69-98.92
Tian, W. et al. [13]	10-60	6.3-15	60	105-250	91
Li B. et al. [12]	5-50	25-7.8	60, 80	97.5-31.3	91-99
Zhu et al. [19]	30	3.7	30, 40	92.5	90
Habtemariam et al. [10]	2	0.71	4, 10	71	100
Danu et al. [21]	50	28	0, 1000	28	100
Qi et al. [16]	15	0.77	80, 60	9.63	85
Tokta and Hocaoglu [14]	40	0.316	134, 190	1.66	95
Chen [28]	10	0.3-0.9	200, 300	1-3	53.67-97.67
B. Kragel[15]	20	173	173	1	83
Zhang et al. [26]	NA	NA	150-670	NA	76.08
Zhang et al. [26]	730	6.0	NA	NA	NA

To evaluate association methods, a common way is to run off-line Monte Carlo simulations with several targets measured by two synchronous sensors [12][16]. As association accuracy is affected by positioning accuracy as well as gaps between targets [24][21], it is difficult to compare results in literature using different data sets, see TABLE I. A paper may adopt different parameters in simulations, so we use commas as separations in the table. The average gap between targets was given only in a few works [6, 7]. When it is not given, we estimate it by the following equation:

$$Gap = \sqrt{\frac{A}{N}} \tag{2}$$

Where A is the area of region, N is the total number of targets. We use the ratio of average gap to positioning error (i.e. GER) to represent the difficulty of track association, see the equation below.

$$GER = \frac{\text{Gap}}{\max(\sigma_i)} \tag{3}$$

Where σ_i is the positioning error of source i. Data from multiple sources may have different positioning accuracy, and we use the maximum standard deviation of positioning errors in the calculation. The exact value of association accuracy ratio is given in only a few works [25, 26], so we estimate it from figures when not given. In the simulation of Hambrick and Blair [6], gaps between targets were 3 units, standard deviation of position ranged from 0.1 to 0.5 units. We take one unit as 1 km in the table.

In spite of parameters listed in the table, some other parameters were also considered in studies, such as errors in angle measurements [21] and systematic biases [16], which can be calibrated in pre-process step by methods like alignment-correlation [25], coherent point drift [27] and human-computer interactions.

To achieve real-time association, time complexity must meet the demand of immediate process. However, very limited attention was paid to it. Time complexity of real-time association is affected by target count as well as message rate. As track association was usually performed off-line in batch in previous works [12, 29, 30], time consumed was affected by target count and steps simulated, so we list them along with running time in TABLE II. We have shown in the previous work [31] that about 250 AIS messages per second have been received from August 2012 to May 2015, and the rate is increasing every year. In addition, more than 50 000 Chinese fishing vessels were already equipped with BeiDou terminals in 2016 [2]. To associate tracks of global ships in real-time, the algorithm should be able to process hundreds of messages from tens of thousands targets per second, which is hard for current multidimensional assignment-based methods using typical PCs.

TABLE II. TIME COMPLEXITY OF TRACK ASSOCIATION IN LITERATURE ORDERED BY TARGET COUNT

Authors	Target count	steps	Time/seconds
Li et al. [12]	5-50	300	0.4731-4.9251
Duan [35]	8, 16	150	30, 200
Tian et al. [13]	10-60	150	0.1-1.3
Tokta and Hocaoglu [14]	40	200	50
Gao et al. [30]	100	10	0.28
Zhang et al. [32]	<1000	1	9.4-10.9
Wang [29]	3200	1	4.4

As a conclusion, there are three problems to be addressed for large scale real-time track association:

- Applicability: Widely used multidimensional assignment algorithms are not applicable when the intersection among target sets is small or even empty;
- Efficiency: Track association algorithms should be able to process asynchronous data from multiple sources in real-time at a rate faster than that data is producing, which is generally not satisfied with typical PCs especially when tracking more than 100 targets by heterogeneous sensors;
- Evaluation: No open-source benchmark is available in the community, making it hard to compare massive algorithms in literature.

III. NETWORK-BASED TRACK ASSOCIATION

Instead of using association matrix, we put forward a network-based method to associate tracks in real-time, taking ships as nodes and association beliefs as weights of directed edges, see Fig. 1. This network evolves using DST whenever new data is received.

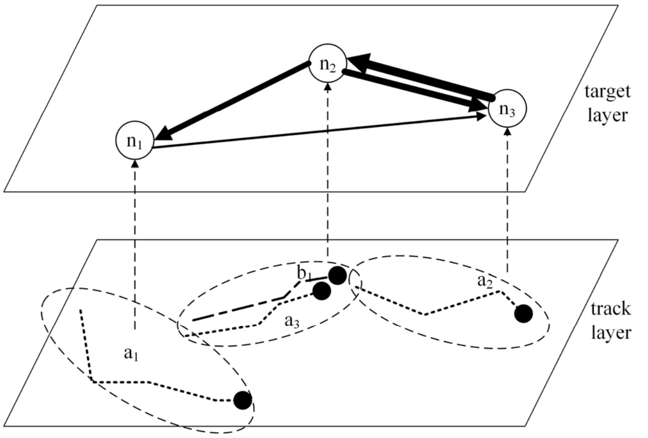

Fig. 1. The center shifts from "tracks" to "targets" in the network-based method. Each node in the target layer corresponds to one or more tracks in the track layer from source a and b. The width of each arrow in the target network represents that directed edge's weight. Edges pointing from non-singleton sets are not depicted in this figure.

A. Basics of DST

DST is a mathematical theory of evidence developed by Dempster [33] and Shafer [34]. As Shafer pointed out [37], DST is based on two ideas: "the idea of obtaining degrees of belief for one question from subjective probabilities for a related question, and Dempster's rule for combining such degrees of belief when they are based on independent items of evidence". In DST, a piece of evidence is represented by a belief function instead of a Bayesian probability distribution. One advantage of a belief function over a Bayesian probability distribution is that a belief function can express our ignorance explicitly. The other advantage is that sometimes direct probabilities are not available or it's too complex to get them. When evidence accumulates, uncertainty in belief functions will diminish. In this paper, we take each message as an evidence about a target's equality to other targets. In the target network, weights of directed edges pointing to each node are association beliefs constituting its BPA function. Once a new message is received, this network evolves by combining the corresponding node's BPA with a new one calculated according to the message.

In DST, the frame of discernment Ω is an exhaustive set of mutually exclusive answers to the question of interest. The true answer we are seeking is a singleton element in Ω. The total unitary mass is distributed among subsets of Ω. The mass of subset A represents the degree of belief allocated exactly to A, which is denoted as m(A) and called BPA. In DST, m(∅)=0.

When two pieces of evidence from independent information sources are represented by a BPA function respectively, we can combine them to get a new one. Given two BPAs m_1 and m_2, we can get $m_{1\oplus 2} = m_1 \oplus m_2$ by Dempster's combination rule:

$$m_{1\oplus 2}(A) = \frac{\sum_{B\cap C=A} m_1(B)m_2(C)}{1-k} \tag{4}$$

Where $k = \sum_{B\cap C=\emptyset} m_1(B)m_2(C)$, $A \neq \emptyset, k \neq 1, A \subseteq \Omega, B \subseteq \Omega,, C \subseteq \Omega$.

B. Modelling Track Association as an Evolving Target Network

The weights of edges pointing to each node in Fig. 1 constitutes its BPA function. A node n_k in target layer corresponds to one or more tracks in track layer:

$$n_k = \{t_k^1, t_k^2, \dots\} \tag{5}$$

The weight $w_{i,k}$ of a directed edge <n_i, n_k> represents the degree of belief allocated to the hypothesis that n_i and n_k are actually the same target, satisfying:

$$(\sum_{i=1}^{N} w_{i,k}) + w_{\Omega,k} = 1, \qquad i \neq k \tag{6}$$

$$\Omega = \{n_1, n_2, \dots, n_N\} \tag{7}$$

$$0 \leq w_{i,k} = m_k(\mathrm{i}) \leq 1, n_i \in \Omega \tag{8}$$

$$0 \leq w_{\Omega,k} = m_k(\Omega) \leq 1 \tag{9}$$

Where N is the total number of nodes in target layer, $w_{\Omega,k}$ is the mass assigned to universe, $m_k(\mathrm{i})$ is the BPA function of node n_i. When $w_{i,k}$ is above threshold, node n_i will be merged into n_k and thus track association is performed:

$$n_k = n_k \cup n_i \tag{10}$$

If n_k has not been updated by track t_k^j for a threshold time (reasons include: t_k^j was false measurements; t_k^j has be associated to a target mistakenly and their distance becomes larger and larger afterwards), the track will be removed from n_k, and thus separation or destroying (when this track does not belong to other nodes either) happens.

By modeling track association as a target network, the focus of research will be the evolution of the network.

C. Evolution of Target Network

We propose a DST-based evolution method in this paper, taking each data received as one evidence about relations among nodes in the target network, see Fig. 2 and Fig. 3.

In the initial state, there is no node in the target network. Whenever a new message is received, we calculate the mahalanobis distance between positions in it and nearby targets n_i (after dead reckoning to the same time using constant velocity model). Then we normalize these distance values into $Assoc(n_i)$ by an S-shaped membership function:

$$\mathrm{S(x; a, b, c)} = \begin{cases} 0, & x < a \\ 2(\frac{x-a}{c-a})^2, & a \leq x \leq b \\ 1 - 2(\frac{x-c}{c-a})^2, & b < x \leq c \\ 1 & x > c \end{cases} \tag{11}$$

If the maximum value of $Assoc(n_i)$ is low, it is probable that this message is from a new target or a false measurement. So we assign the mass to Ω using the following equation:

$$m'(\Omega) = 1 - S\big(max\big(Assoc(n_i)\big); 0, 0.5, 1\big), n_i \in \Omega \tag{12}$$

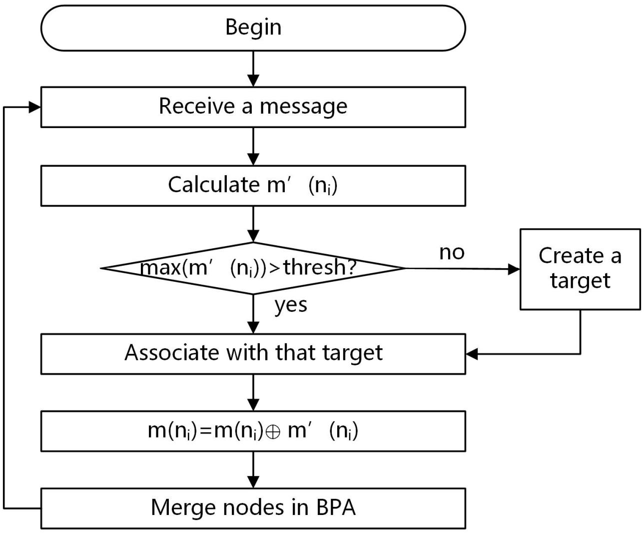

Fig. 2. Flow chart of real-time track association. When a new message is received, the BPA function $m^{'}(n_i)$ of this message versus existing targets will be calculated. If the max belief is larger than threshold, this message will be associated with that target. Otherwise, a new target will be created. After association with a target node n, $m^{'}(n_i)$ will be combined with this node's $m(n_i)$ using DST to update the edges connecting to this node. If there is an edge from node m with weight larger than threshold, m will be merged into node n.

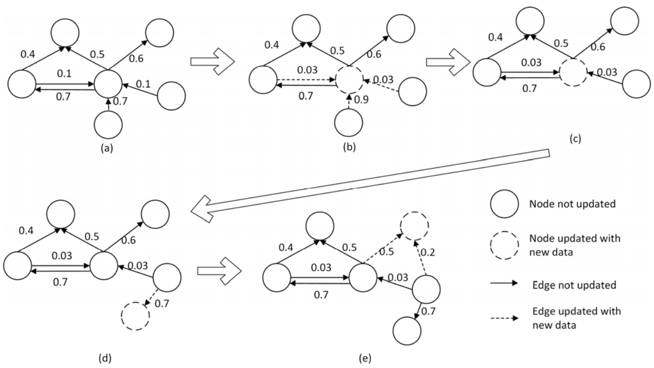

Fig. 3. An example of a target network evolving over time. (a) Current state of the network. (b) When a new data point comes, its BPA m'(ni) with candidate targets is calcualted. The data will be associated with a target if association belief is above threshold. In this case, m'(ni) will be combined with the BPA m(ni) of this target to get an udpated one. (c) Two targets are merged because the weight of the edge connecting them is above threshold. (d) Another message comes, and a new target is created because it fails to associate with any existed target at this moment. (e) One more message comes, which is associated with an existed target. The weights of two edges are updated.

Then we assign masses to other subsets of Ω by normalizing $Assoc(n_i)$

$$m'(n_i) = \frac{Assoc(n_i)}{\sum_{n_j \subseteq \Omega} Assoc(n_j)} \tag{13}$$

If $\max(m'(n_i))$ is above a threshold (e.g., 0.8), then this new message is associated to the corresponding node, otherwise a new node will be created. After that, the node's BPA function $m(n_i)$ will be updated by combining with this $m'(n_i)$ using DST. In this way, the network evolves over time, and nearby nodes are merged when the weights of edges connecting them is above threshold. If a node has not been updated by any message for a threshold time, it will be destroyed.

In this paper, we only considered continuous kinematic states and assigned masses to singleton elements as well as universe. When discrete states like colors are considered, masses can also be assigned to non-singleton elements.

The advantage of this method over multidimensional assignment is as follows:

- No synchronization nor high intersection ratio among multiple sources is required, as the network evolves with each data point received;
- Information in history tracks are fused into the network by DST, improving the association accuracy;
- With each new message received, BPA is calculated according to the distance from the position in it to those of nearby nodes and then combine it using DST rather than seeking a global optimal solution every step in the multidimensional assignment method, thus having higher efficiency when tracking large number of targets.

IV. Benchmark

The benchmark consists of an open source track simulator ICTShips [36] and a free visualization software ICTShipVisual. The ICTShips project can be forked from a git repository: https://gitee.com/iOceanPlus/open_multisource_targetdata_simulator (recommended) or https://github.com/iOceanPlus/Product_MultiSource_Target Data_Simulator_OpenSource, and the visualization software can be downloaded from https://pan.baidu.com/s/1-8fgxedtnEtcfjXMdwyBzw.

The simulator reads parameters from files and then simulates detections of targets from multiple sources. Simulated messages are sent to a RabbitMQ server [40], which is an open source message broker. Track association and other processes fetch data from RabbitMQ server and send results back, see Fig. 4. RabbitMQ is supported on a number of languages [41]: Java, .NET, Ruby, Python, PHP, Objective-C and Swift, Scala, JavaScript, C/C++, Go, etc. To communicate with it using languages not supported yet, we can develop an adapter as a bridge. From the Web UI of RabbitMQ management, we can see details of the message queue that track association process is consuming: consumer utilization (defined as proportion of time that a queue's consumers could take new messages), message count waiting to be consumed, count of consumers, etc. If a queue has a consumer utilization of 100% then it never needs to wait for its consumers, it is always able to push messages out to them as fast as it can [42]. In this case, the consumer can said to be a real-time software.

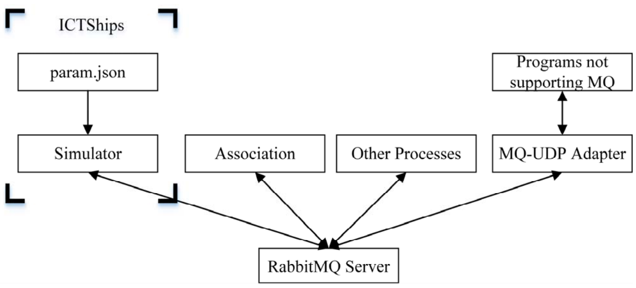

Fig. 4. Track association and other processes communicate with ICTShips through RabbitMQ.

Fig. 5 shows the architecture of the simulator, consisting of target initial states, motion models, measurement models and transmission models. In the simulator, each target is installed with (or observed by) several types of devices, which sense states of targets according to measurement models and send messages into channels according to transmission models. After a latency of transmission, packets not lost are received by some data sources. As an open source project, researchers can customize this simulator or just run it. A reference motion model has been implemented in the simulator, see Fig. 6. The initial states of targets are from a snapshot of global AIS data and can be customized if needed, see Fig. 7.

Target density [31] in different regions varies, resulting in different average gaps, which can be calculated according to equation (2). To evaluate the performance of track association under scenarios with various levels of target density and GERs, 3 scenes are configured with different bounding regions: SparseShip45, RegularShip32 and DenseShip42. To evaluate if an algorithm can associate global targets in real-time, GlobalShip110K is configured with 102 231 ships observed, about 8,930 messages received per second. We list main parameters of these scenarios in TABLE III, TABLE IV and TABLE V.

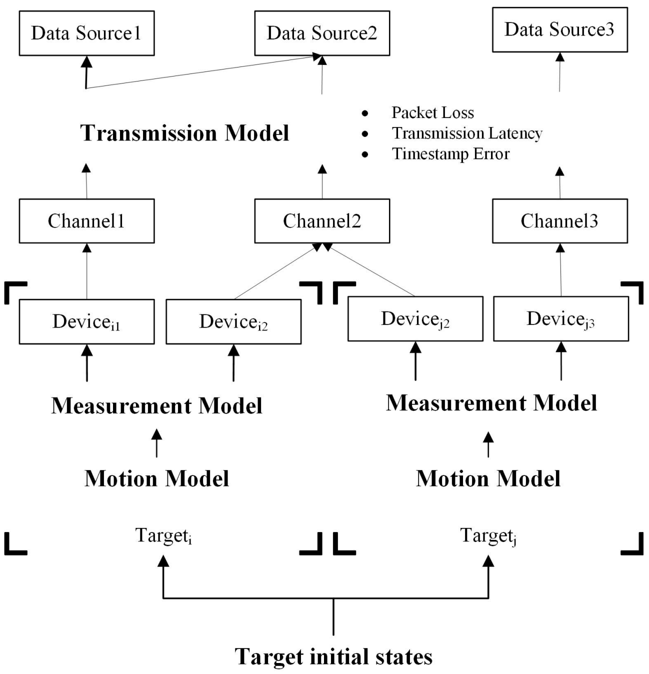

Fig. 5. Architecture of simulator

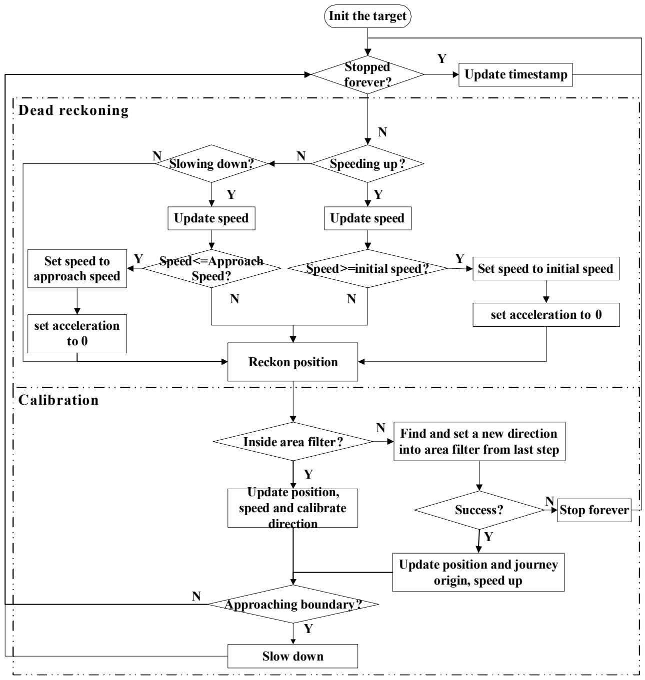

Fig. 6. An example of target motion model implemented in the open source simulator ICTShips. Targets move in great circle routes with constant speed until they are going to meet land or boundary. Then they slow down with constant acceleration and change directions randomly to move into water again. After the new direction is set, targets speed up to their initial speed again.

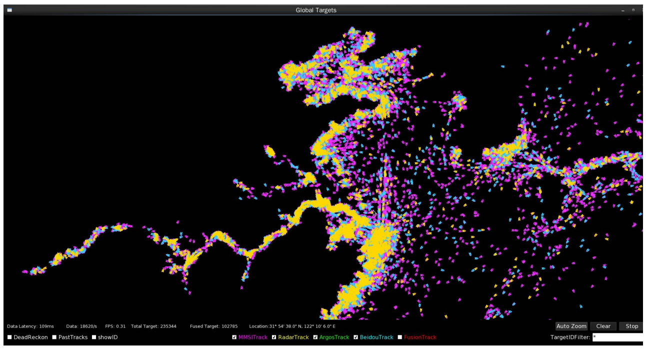

Fig. 7. Screenshot of ICTShipVisual with targets from multiple sources around China marked by different colors

TABLE III. CONFIGURATIONS OF MEASUREMENT MODELS IN ICTSHIPS

device	StDev(pos) /meter	StDev (SOG) /knot	StDev ([a]COG) /degree	[b]Interval /second
Beidou	15	0.5	5	20
AIS	15	0.5	5	12
Radar	DenseShip42: 100 GlobalShip110K: 50 Others: 500	0.5	5	2

[a] COG: Course Over Ground.
[b] Interval: Interval of sending a message from a device.

TABLE IV. PARAMTERS OF 4 SCENES IN ICTSHIPS

Scene	Center/(latitude, longitude)	Targets observed	Gap/km	GER
SparseShip45	(31.415,124.079)	45	15.66	31.3
RegularShip32	(38.942,118.378)	32	1.06	2.12
DenseShip42	(31.381,121.501)	42	0.0787	0.787
GlobalShip110K	Global	102,231	70.6	1412

TABLE V. TARGET DETECTION RATIO AND PACKET LOSS RATIO

Source	Targets detected / %	Packet loss ratio / %
Beidou	GlobalShip110K: 20 Others: 55	GlobalShip110K: 30 Others: 10
AIS	90	GlobalShip110K: 60 Others: 10
Radar	GlobalShip110K: 10 Others: 100	GlobalShip110K: 10 Others: 0

V. RESULTS

Association error ratio is widely used in literature to evaluate association algorithm, and we define it in the following equation:

$$error\_ratio = \frac{N_{er}}{N_{assoc}} \tag{14}$$

Where N_{er} is the number of targets mistakenly associated, N_{assoc} is total number of targets after association. In spite of error_ratio, we propose to use the ratio of associated targets to true number of targets to evaluate how often miss association occurs:

$$target\_ratio = \frac{N_{assoc}}{N_{truth}} \tag{15}$$

Where N_{truth} is the true number of targets.

To evaluate the convergence speed of algorithm, we propose **convergence time**, which is defined as the time before target_ratio and error_ratio converge. As history information is utilized, target_ratio and error_ratio will gradually converge.

The benchmark and real-time association both ran in the same computer, with 6-core CPU Intel(R) Xeon(R) CPU E5-2609 v3 @ 1.90GHz and 8GB DDR4 memory @ 2.4 GHz. The error_ratio and target_ratio is changing as targets move and the algorithm runs. After 735 seconds, average target_ratio and error_ratio is around 1.0055 and 0.539% respectively in DenseShip42. The target_ratio drops to 1 after about 200 seconds and error_ratio remains 0% in RegularShip32 and SparseShip45. Under GlobalShip110K, target_ratio varies between 0.9998 and 1.0001 after about 30000 seconds and error_ratio converges to 0.1%. An example of tracks before and after association is in Fig. 8. The results are depicted in Fig. 9, Fig. 10, Fig. 11 and Fig. 12, and summarized in TABLE VI.

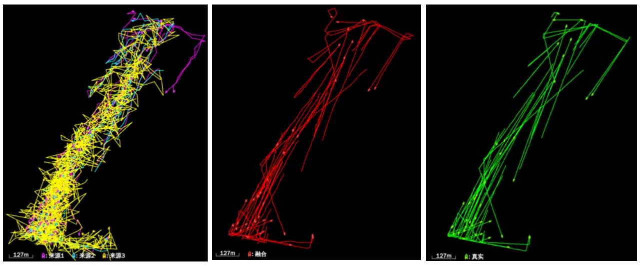

Fig. 8. Tracks before and after association in DenseShip42. Target density in this area is so high that noise is larger than the average gap among targets. Left: Tracks from 3 sources marked by different colors. Middle: Results of track association and filtering. Right: Ground truth of DenseShip42.

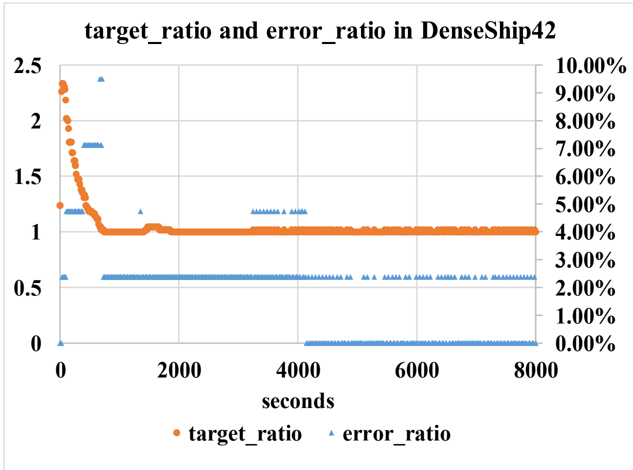

Fig. 9. The evolvement of target_ratio and error_ratio in DenseShip42. Left vertical axis: target_ratio. Right vertical axis: error_ratio.

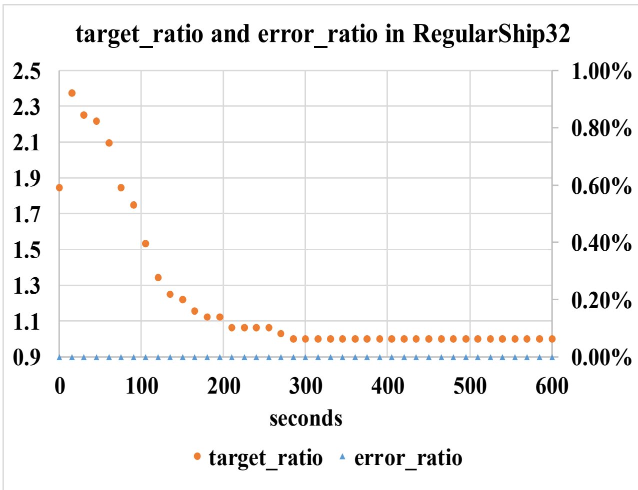

Fig. 10. The evolvement of target_ratio and error_ratio in RegularShip32. Left vertical axis: target_ratio. Right vertical axis: error_ratio.

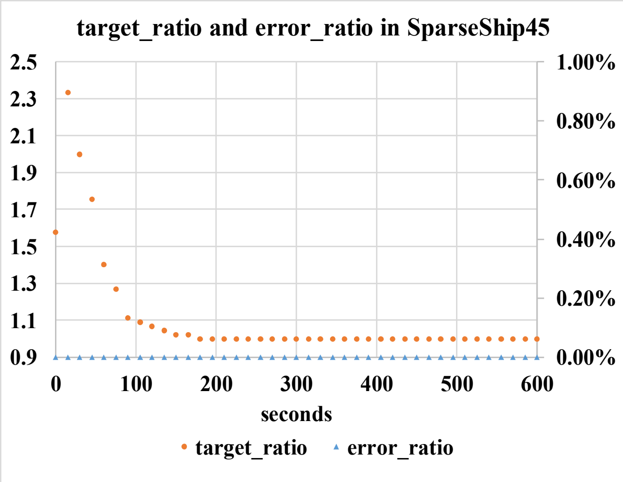

Fig. 11. The evolvement of target_ratio and error_ratio in SparseShip45. Left vertical axis: target_ratio. Right vertical axis: error_ratio.

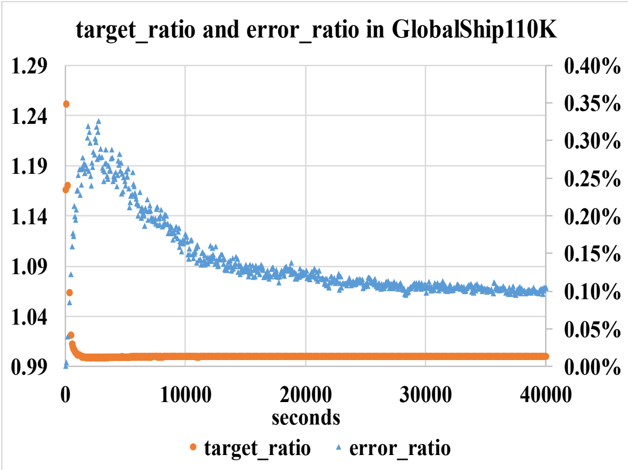

Fig. 12. The evolvement of target_ratio and error_ratio in GlobalShip110K. Left vertical axis: target_ratio. Right vertical axis: error_ratio.

TABLE VI. RESULTS OF ASSOCIATION UNDER 4 SCENES

Scene	target_ratio	error_ratio/%	GER	convergence time/seconds
SparseShip45	1.0000	0.000	31.3	210
RegularShip32	1.0000	0.000	2.12	210
DenseShip42	1.0055	0.539	0.787	735
GlobalShip110K	0.9998~ 1.0001	0.096	1412	30000

VI. CONCLUSIONS

Instead of using assignment algorithms to match tracks, we put forward an evolving target network-based method to create, update, merge, separate and destroy targets over time, which centers on targets rather than tracks. This method does not require synchronization or high intersection ratio among multiple sources. Results show that it can associate more than 8,930 tracks per second from 3 sources when 102,231 ships are detected, efficient enough for real-time systems in most cases and significantly faster than existed works listed in TABLE II.

To evaluate the track association method, we described an extensible benchmark consisting of an open source track simulator ICTShips and a free visualization software ICTShipVisual. To compare association difficulties under different scenes, the notion GER was put forward, and four scenes with GERs from 0.787 to 1,412 were configured: DenseShip42, RegularShip32, SparseShip45 and GlobalShip110K. Error ratios of track association range from 0% to 0.539%, lower than those listed in TABLE I. The track simulator ICTShips is customizable, compromising target initial states, motion models, measurement models and transmission models. Contributions from the community to this open-source project are expected.

DST assigns a belief mass to each element of the power set, but in this paper only singleton sets and universe are considered. In future works, masses can be assigned to non-singleton sets when discrete states like colors are used to calculate association likelihood. Furthermore, the evolution of target networks is not limited to the method based on DST as described in this paper and is worth further study. Other future works include adding systematic bias and covariance in measurement models and extending motion models for the benchmark.

REFERENCES

[1] UNCTAD, Review of Maritime Transport 2016. http://uncta.org/en/pages/publicationwebflyer.aspx?publicationid=1650, 2016 (Accessed 16 January, 2018).

[2] State Council Information Office of the People's Republic of China, China's BeiDou Navigation Satellite System. http://www.beidou.gov.cn/zt/xwfbh/bdbps/gdxw_1363/201710/P020171208599812943426.pdf, 2016(Accessed 16 January 2018).

[3] IMO, International Convention for the Safety of Life at Sea (SOLAS), 1974.

[4] ITU-R, Technical characteristics for an automatic identification system using time-division multiple access in the VHF maritime mobile band, *Recommendation ITU-R M.1371-4*, 2001.

[5] China's Ministry of Agricultureh, Regulation on deep sea fishing vessel monitoring. http://www.moa.gov.cn/govpublic/YYJ/201501/t20150120_4342021.htm, 2014 (Accessed 13 December 2018).

[6] D. Hambrick, W.D. Blair, Multisensor track association in the presence of bias, IEEE Aerospace Conference, 2014.

[7] Y. Bar-Shalom, H. Chen, Multisensor track-to-track association for tracks with dependent errors, 43rd IEEE Conference on Decision and Control, 2005, pp. 2674-2679.

[8] W. Tian, Reference pattern-based track-to-track association with biased data, IEEE Transactions on Aerospace and Electronic Systems, 2006, 52: 501-512.

[9] T. Denœux, N. E. Zoghby, V. Cherfaoui, and A. Jouglet, Optimal object association in the Dempster-Shafer framework. IEEE Transactions on Cybernetics, 2017, 44(12), 2521-2531.

[10] B. Habtemariam, R. Tharmarasa, M. Mcdonald, T. Kirubarajan, Measurement level AIS/radar fusion, Signal Processing, 106 (2015) 348-357.

[11] H. Zhu, W. Wang, C. Wang, Robust track-to-track association in the presence of sensor biases and missed detections, Information Fusion, 27 (2016) 33-40.

[12] B. Li, N. Liu, G. Wang, L. Qi, Y. Dong, Robust track-to-track association algorithm based on t-distribution mixture model, IEEE 20th International Conference on Information Fusion, 2017.

[13] W. Tian, Y. Wang, X. Shan, J. Yang, Track-to-Track Association for Biased Data Based on the Reference Topology Feature, IEEE Signal Processing Letters, 21 (2014) 449-453.

[14] A. Tokta, A.K. Hocaoglu, A Fast Track to Track Association Algorithm by Sequence Processing of Target States, IEEE 21st International Conference on Information Fusion (FUSION), 2018, pp. 1280-1284.

[15] B. Kragel, S. Herman, N. Roseveare, A comparison of methods for estimating track-to-track assignment probabilities, IEEE Transactions on Aerospace and Electronic Systems, 48 (2012) 1870-1888.

[16] L. Qi, K. Dong, Y. Liu, J. Liu, T. Jian, Y. He, Anti-bias track-to-track association algorithm based on distance detection, Iet Radar Sonar & Navigation, 11 (2017) 269-276.

[17] L. A. Testa, C. D. Pickle, and S. S. Blackman, A Dempster-Shafer approach to multi-sensor track association. IEEE Aerospace Conference, 2016.

[18] F. Mazzarella, A. Alessandrini, H. Greidanus, et al., Data Fusion for Wide-Area Maritime Surveillance. COST MOVE Workshop on Moving Objects at Sea, Brest, 2013.

[19] H. Zhu, S. Han, Track-to-Track Association Based on Structural Similarity in the Presence of Sensor Biases, Journal of Applied Mathematics, 2014.

[20] L. Sun, W. Zhou, A multi-source trajectory correlation algorithm based on spatial-temporal similarity. International Conference on Information Fusion, 2017.

[21] D. Danu, A. Sinha, T. Kirubarajan, M. Farooq, B. Dan, Fusion of over-the-horizon radar and automatic identification systems for overall maritime picture, in: International Conference on Information Fusion, 2007.

[22] Y. Liu, H. Wang, Y. He, et al. New track-to-track correlation algorithms based on bithreshold in a distributed multisensor information fusion system. Journal of Computer Science, 9(12):1695-1709, 2013.

[23] X. Tian, Y. Bar-Shalom. Sliding window test vs. single time test for Track-to-Track Association. 11th International Conference on Information Fusion, 2008.

[24] W. Tian, Y. Wang, X. Shan, and J. Yang, Analytic performance prediction of track-to-track association with biased data in multi-sensor multi-target tracking scenarios. Sensors, 13(9), 2013, pp. 12244-12265.

[25] J. Yang, Q. Song, C. Qu, Y. He, Track-to-Track Association Technique in Radar Network in the Presence of Systematic Errors, Journal of Signal & Information Processing, 04 (2015) 288-298.

[26] H. Zhang, Y. Liu, Y. Ji, L. Wang, J. Zhang, Multi-Feature Maximum Likelihood Association with Space-borne SAR, HFSWR and AIS, Journal of Navigation, 70 (2016) 359-378.

[27] H. Zhu, M. Wang, K.V. Yuen, H. Leung, Track-to-Track Association by Coherent Point Drift, IEEE Signal Processing Letters, 24 (2017) 643-647.

[28] C.Z. Chen, R. Huai-Lin, H.U. Feng-Gang, A New Track Correlation Method in Distributed Radar Network, Electronic Information Warfare Technology, 23 (2008) 27-30.

[29] Q. Wang, An Accelerated Track Association Algorithm for a Large Number of Targets, Telecommunication Engineering, 54 (2014) 1641-1645.

[30] Y. Gao, X. Chen, Y. Wang, M. Wu, Improved ant Colony Solution Algorithm Accelerated by GPU in Track Correlation, Journal of Northwestern polytechnical university, 34 (2016) 514-519.

[31] L. Wu, Y. Xu, Q. Wang, F. Wang, Z. Xu, Mapping Global Shipping Density from AIS Data, Journal of Navigation, 70 (2017) 67-81.

[32] H. Zhang, Y.X. Liu, J. Zhang, Y.G. Ji, Z.Q. Zheng, Target point tracks optimal association algorithm with surface wave radar and automatic identification system, Journal of Electronics & Information Technology, 37 (2015) 619-624.

[33] A.P. Dempster, Upper and lower probabilities induced by a multivalued mapping, The annals of mathematical statistics, 38 (1967) 325-339.

[34] G. Shafer, A mathematical theory of evidence, Princeton university press Princeton, 1976.

[35] Y. Duan, Study of Track Correlation Algorithm in Multi-radar and Multi-target tracking, Thesis to Ludong University for the Degree of Master, 2015.

[36] L. Wu, ICTShips: Multisource Target Data Simulator. https://gitee.com/iOceanPlus/open_multisource_targetdata_simulator, 2017 (Accessed 20 November 2017).

[37] G. Shafer, Perspectives on the theory and practice of belief functions, International Journal of Approximate Reasoning, 4 (1990) 323-362.

[38] R. A. Singer, A. J. Kanuyuck, Computer Control of Multiple Site Correlation, Automatica, 7 (1971) 455 – 463.

[39] L. M. Kaplan, Y. Bar-Shalom, W. D. Blair, Assignment costs for multiple sensor track-to-track association, IEEE Transactions on Aerospace & Electronic Systems, 2008, 44(2) 655-677.

[40] Pivotal Software Inc., Home page of RabbitMQ. http://www.rabbitmq.com/, 2017 (Accessed 21 November 2017).

[41] Pivotal Software Inc., Clients & Developer Tools. http://www.rabbitmq.com/devtools.html, 2017 (Accessed 21 November 2017).

[42] M. Simon, Finding bottlenecks with RabbitMQ 3.3. https://www.rabbitmq.com/blog/2014/04/14/finding-bottlenecks-with-rabbitmq-3-3/, 2014 (Accessed 21 November 2017).