Optimizing Vessel Trajectory Compression

Giannis Fikioris[1], Kostas Patroumpas[2] and Alexander Artikis[3,1]
[1]*Institute of Informatics & Telecommunications, NCSR Demokritos, Athens, Greece*
[2] *Information Management Systems Institute, Athena Research Center, Athens, Greece*
[3]*Department of Maritime Studies, University of Piraeus, Greece*
gfikioris@iit.demokritos.gr, kpatro@athenarc.gr, a.artikis@unipi.gr

***Abstract*—In previous work we introduced a trajectory detection module that can provide summarized representations of vessel trajectories by consuming AIS positional messages online. This methodology can provide reliable trajectory synopses with little deviations from the original course by discarding at least 70% of the raw data as redundant. However, such trajectory compression is very sensitive to parametrization. In this paper, our goal is to fine-tune the selection of these parameter values. We take into account the type of each vessel in order to provide a suitable configuration that can yield improved trajectory synopses, both in terms of approximation error and compression ratio. Furthermore, we employ a genetic algorithm converging to a suitable configuration per vessel type. Our tests against a publicly available AIS dataset have shown that compression efficiency is comparable or even better than the one with default parametrization without resorting to a laborious data inspection.**

***Index Terms*—AIS, genetic algorithm, maritime data analytics**

I. Introduction

Thanks to the Automatic Identification System (AIS), tracking vessels across the seas provides a powerful means for maritime safety and environmental protection. However, large amounts of streaming AIS positional updates from vessels can hardly contribute additional knowledge about their actual motion patterns. Vessels are generally expected to maintain straight, predictable routes at open sea, except in cases of adverse weather conditions, accidents, traffic restrictions, etc. In [11] a maritime surveillance system was introduced, involving a *trajectory detection* module that can provide summarized representations of vessel trajectories by consuming AIS positional messages *online*. The key idea behind the proposed summarization is that keeping only some *critical points* may be enough to reconstruct with tolerable accuracy the original course of each vessel. Indeed, instead of retaining every incoming position for every vessel or even applying a costly multi-pass trajectory simplification algorithm, this method drops positions along trajectory segments of "normal" motion characteristics. In addition, the retained critical points can be marked with suitable *annotations*, i.e., indicating stops, turning points, changes in speed, etc. The resulting *trajectory synopsis* per vessel is derived from those judiciously annotated critical points and can approximately reconstruct its original course. This framework has been further enhanced to run against scalable streaming data in cluster infrastructure and enable Big Data Analytics [17] not only in the maritime domain, but also in aviation [12].

This methodology can provide reliable trajectory synopses with little deviations from the original course by discarding at least 70% of the raw data as redundant; and in case of frequent updates, even less than 1% of the data need be retained. However, such trajectory compression is very sensitive to *parametrization*. Multiple rules and spatiotemporal conditions are examined for each incoming AIS location per vessel, e.g., difference in heading, speed, acceleration, etc. with respect to its previous location or its known motion pattern. So, ill selection of parameter values may have a strong impact on the quality of the resulting trajectory synopses, the achieved compression ratio, or both. Default parameter values for extracting synopses can be assigned after painstaking exploration of a given dataset with the valuable advice of domain experts. Still, in any other AIS dataset, this process may need be repeated. Last, but not least, this kind of trajectory summarization applies the same parametrized conditions over all monitored vessels, irrespective of their type, tonnage, length, etc. This "one size fits all" approach lacks flexibility and cannot easily cope with the varying mobility patterns of vessels at sea.

In this paper, our goal is to fine-tune the selection of these parameter values. We take into account the type of each vessel (passenger, cargo, fishing, etc.) in order to provide a suitable configuration that can yield improved trajectory synopses, both in terms of approximation error and compression ratio. As many parameters are involved in this optimization problem, we employ a genetic algorithm that iterates over several combinations of the parameter values until converging to a suitable configuration per vessel type. This offline fine-tuning of parameters can then be used by the online trajectory summarization method with significant gains in terms of approximation quality and space savings. Overall, our tests against a publicly available AIS dataset [15] have shown that compression efficiency is comparable or even better than the one with default parametrization without resorting to a laborious data inspection.

The remainder of this paper proceeds as follows. Section II surveys related work. Section III provides an overview of the mobility events employed in online summarization of evolving trajectories of vessels. Section IV analyzes the suggested methodology for fine-tuning the parameters used in trajectory compression, while Section V discusses the customization of the genetic algorithm. Section VI reports results from our empirical study against a real AIS dataset. Finally, Section VII summarizes the paper and outlines future research directions.

II. Related Work

Our approach on trajectory synopses over streaming AIS positions involves a kind of *online* path simplification. Offline techniques like [8], [10] cannot apply, since complete trajectories must be available in advance. In contrast, we consider evolving trajectories where fresh locations are received online. Samples retained in the synopses should keep each compressed trajectory as much closer to the original one, as in fitting techniques [1], [9], which minimize approximation error. The one-pass approach in [9] discards points buffered in a sliding window until the error exceeds a given threshold. The notion of safe areas in [14] keeps samples that deviate from predefined error bounds regarding speed and direction. Dead-reckoning policies like [18] and mobility tracking protocols in [4] are usually employed on board of the moving objects, hence they do not seem applicable for AIS position reports. More recently, the focus on online trajectory simplification is focused on error bounds. For example, the ageing-aware approach in [7] uses a bounded quadrant mechanism to chose samples for the summary. Local distance checks and optimizations for higher compression are employed in [5]. A novel spatiotemporal cone intersection technique involving Synchronous Euclidean Distance (SED) is applied in [6] achieving more aggressive compression. A survey and empirical study of various trajectory simplification techniques is available in [19].

Even though such generic online methods can provide reliable summaries over vessel trajectories, they entirely lack support for mobility-annotated features in the retained samples. The maritime surveillance platform in [11] aims at tracking vessel trajectories and also recognizing complex events (e.g., suspicious vessel activity). Its trajectory compression module applies a sliding window over the streaming positions, and periodically reports annotated "critical" points (stop, turn, speed change, etc.) being retained for each vessel's synopsis. Tests indicated that less than 5% of the raw data suffice to offer reliable trajectory approximations. Further enhanced to a stream-based application (*Synopses Generator*) for modern cluster infrastructures, it can detect mobility events with richer semantics (multiple annotations per location, more refined conditions) with minimal latency [17]. With extra rules, the reported mobility events can also act as notifications to timely trigger detection of more *complex events* [13], analogous to those applied by CEP-traj [16]. However, this framework requires careful parametrization of its various conditions, which we aim to fine tune in this paper using a genetic algorithm.

III. Online Summarization of Vessel Trajectories

In this Section, we overview the mobility events detected along vessel trajectories and the parameters involved in their specification by the Synopses Generator. In depth analysis of the applied trajectory summarization is available in [11], [12].

The Synopses Generator accepts AIS positional messages (of types 1, 2, 3, 18, 19, 27) and extracts three attributes, namely *MMSI*, *Longitude*, and *Latitude*, as well as the *Timestamp* assigned to each message upon arrival to a base station. However, this data is not error-free [3]. To reduce this *noise*, a series of single-pass heuristics is applied to filter incoming locations online and can actually eliminate up to 20% of the raw AIS positions [11]. Even though some locations could be misreported as noisy, this hardly affects the resulting trajectory synopses as their critical points can still be identified from the fresh locations that keep continuously arriving.

Noiseless positional updates per vessel are chronologically buffered in memory. In particular, the *most recent portion* of each evolving trajectory is available as a distinct sequence of m 'clean', time-ordered locations (e.g., $m = 5$). From this batch of locations, the *mean velocity* $\overrightarrow{v}_m$ of that vessel can be estimated, as well as several derived spatiotemporal features (distance, travel time, overall change in heading, etc.). To avoid considering obsolete locations in velocity calculations, a maximum *historical timespan* ω is set, e.g., discarding any locations received one hour before current timestamp τ_{now}.

The Synopses Generator applies single-pass heuristics for detecting mobility events of the following types:

- *Stop* indicates a stationary vessel once its instantaneous speed v_{now} (estimated from its two most recent locations) remains below threshold v_{min} (e.g., 0.5 knots) over a period of time. The first location qualifying to this condition is annotated as $stopStart$, while the last one as $stopEnd$. To cope with the side-effects of sea drift, vessel agility or GPS discrepancies, any fresh locations are ignored while no significant displacement is observed (less than a *distance* threshold D, e.g., 50 meters) from the previous one. If the next location is away more than D, the stop event ends. Overall, two critical points are issued in order to capture the duration of this phenomenon; any intermediate locations are discarded from the synopsis.
- *Slow motion* means that during some time interval a vessel consistently sails at low speed below a threshold v_θ (e.g., < 5 knots). The first and last AIS positions in such subtrajectory should be annotated as critical points to indicate the duration of this event.
- *Change in Heading*: Once there is a difference in heading of more than a given angle threshold $\Delta\theta$ (e.g., $> 4^o$) with respect to mean velocity $\overrightarrow{v}_m$, the previously reported location marks a turning point and qualifies as critical. Since vessels usually make smooth turns due to their large size and safety regulations, a series of such turning points may be issued to better approximate this course.
- *Speed change* indicates that the vessel has either accelerated or decelerated. Such critical points are detected once the rate of change for speed deviates more than a given threshold α (e.g., 25%) from its mean speed v_m computed along the past m locations, i.e. $\left|\frac{v_\mathrm{now} - v_\mathrm{m}}{v_\mathrm{now}}\right| > \alpha$. To mark the duration of this event, one critical point (*speedChangeStart*) is issued when a vessel starts speeding up or slowing down; once the speed stabilizes, the respective location is marked as $speedChangeEnd$.
- *Communication gaps* are detected when a vessel has not relayed any fresh AIS position over a time period ΔT, e.g., the past 10 minutes. A pair of locations are marked

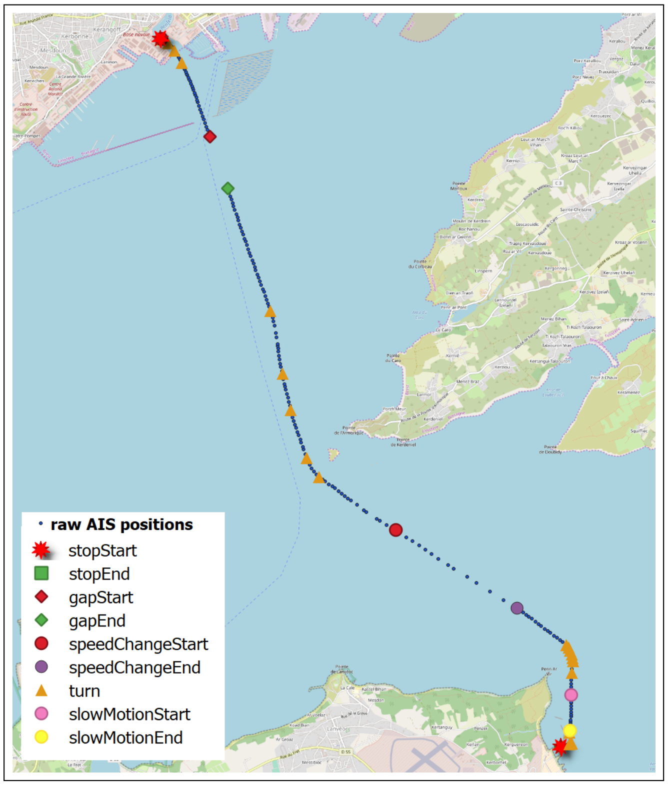


Fig. 1. Critical points detected along a vessel trajectory.

as critical: $gapStart$ to signify when contact was lost, and $gapEnd$ once it gets restored.

Figure 1 illustrates the process of critical point detection. Note that such critical points are issued at operational latency, i.e., in milliseconds. This derived stream of trajectory synopses keeps in pace with the incoming raw streaming data, which gets incrementally annotated with semantically important mobility events. This module can also achieve dramatic compression (sometimes keeping even less than 1%) of the raw streaming data with tolerable error in the resulting approximation. The actual course of a vessel can be approximated using time-based interpolation between successive critical points to estimate positions discarded from synopses [11].

IV. Fine-tuning of Compression Parameters

Trajectory compression is very sensitive to *parametrization*. Table I presents the parameters and their default values, which have been set with the valuable advice of domain experts specifically for an AIS dataset in Brest area [15]. However, in any other AIS dataset, different values may be needed, depending on the geographic area, the sampling frequency, the types of monitored vessels, etc. Moreover, this kind of trajectory summarization applies the same parametrized conditions over all vessels in the data, irrespective of their type, tonnage, length, etc. This approach lacks flexibility and cannot cope with the varying mobility patterns of the vessels. For example, a larger ship takes turns more smoothly, while a fishing or tug boat can make sharper turns. This would allow more relaxed (i.e., greater) angle threshold for larger ships without harming the quality of their synopses. On the other hand, having a stricter angle threshold for smaller ships would entail a more accurate approximate route.

TABLE I
Parameters involved in synopses of vessel trajectories

Symbol	*Parameter*	*Value range*	*Default value*
$\Delta\theta$	Angle threshold (^{o})	$2\dots25$	4
m	Buffer size (locations)	$3\dots50$	5
ΔT	Gap Period (seconds)	$200\dots5000$	1800
ω	Historical timespan (seconds)	$300\dots5000$	3600
v_{min}	No speed threshold (knots)	$0.05\dots2$	0.5
v_θ	Low speed threshold (knots)	$0.05\dots8$	5
α	Speed ratio	$0.01\dots0.8$	0.25
D	Distance threshold (meters)	$2\dots100$	50

In this paper, we address such issues by taking into account the type of each vessel (passenger, cargo, fishing, etc.) in order to provide a suitable configuration that can yield improved trajectory synopses. We employ a genetic algorithm that iterates over several combinations of the parameter values until converging to a suitable configuration per vessel type. This offline fine-tuning of parameters can then be used by the online trajectory summarization method with significant gains in terms of approximation quality and space savings.

A. Optimization Function

We want to keep as few AIS positions as possible, i.e., minimize the *compression ratio*, while at the same time we also need to minimize the *approximation error* in the resulting trajectory synopses. The compression ratio is defined as:

$$Ratio = \frac{\textit{number of critical points for all vessels}}{\textit{number of noiseless points for all vessels}}$$

The parameters that we optimize do not affect the noise reduction filters; thus, the number of noiseless positions is fixed for a given input dataset of AIS messages.

Typically, the approximation error is quantified with the Root Mean Square Error (in meters):

$$RMSE = \sqrt{\frac{\sum_{p \in \textit{noiseless points}} H^2(p, p')}{\textit{number of noiseless points for all vessels}}}$$

where $H(\cdot)$ is the Haversine distance between each original location p from its time-synchronized point p' in the synopsis.

We define our optimization function as

$$(RMSE + r)^n \times Ratio \qquad (1)$$

where n and r are a hyper-parameters. A term r has to be added to *RMSE* in order to avoid solutions to the optimization problem in which *RMSE* becomes 0 by opting for no compression at all. No such term is needed for *Ratio* because the ranges of compression parameters prevent *Ratio* to reach 0.

B. Optimization Algorithm

Figure 2 illustrates our approach using a genetic algorithm to optimize the trajectory compression parameters per vessel type (Table I). Genetic algorithms can solve optimization

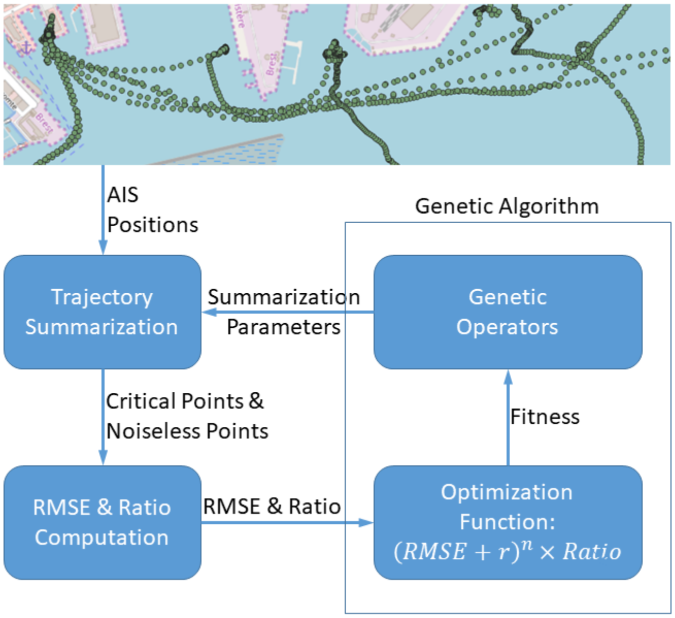


Fig. 2. Processing flow to fine-tune parameters for trajectory compression.

problems that are often intractable because of their large space of parameters. In each generation, the algorithm keeps a population of *individuals*. In our case, each individual is a tuple that contains specific values for the trajectory compression parameters. The individuals of the first generation are picked using a uniform distribution. Then, the following genetic operators are applied, until convergence: selection, crossover and mutation. In the next section, we describe the customization of this genetic algorithm over an AIS dataset.

V. Customization of the Genetic Algorithm

We trained the genetic algorithm on the publicly available AIS dataset of Brest, France [15]. We divided these locations by vessel type and evaluated our approach on the types with the most AIS messages, as listed in Table II.

For each vessel type, we performed 6-fold cross-validation, where each of the 6 parts of the dataset contains the same number of raw AIS locations. We compare the *RMSE* and the compression *Ratio* derived from the genetic algorithm with those achieved by the default values of the trajectory compression parameters. Recall that the default values (Table I) are applied to all vessel types, while the genetic algorithm is trained separately for each vessel type.

The genetic algorithm was customized as follows:

- We used *Tournament Selection* of size 3, which repeatedly and randomly picks 3 individuals and selects the one with the best fitness, until the desired number of total individuals has been picked.
- We adopted *single-point crossover*, with a crossover probability of 0.4.
- We employed *Gaussian Mutation*, which adds random Gaussian noise to each value of an individual with some probability. The mutation probability was set to 0.8 since the Gaussian Mutation restricts mutation. The probability that each trait of an individual would be mutated was set to 0.5.

In a preliminary stage to set hyper-parameters r and n of our optimization function in (1), we selected a small part of the dataset (spanning 6 days) and trained the genetic algorithm for different combinations of r and n. We chose values for these hyper-parameters so that *RMSE* and compression *Ratio* fall below some specified thresholds. Table II presents the thresholds used for each vessel type. As shown in this table, we used three different combinations of thresholds. Each combination was chosen based on the movement and size of each vessel type in order to ensure that the derived trajectory approximation is both reliable and lightweight. For example, fishing boats often move in a zig-zag manner, which requires a large number of critical points to be annotated, while passenger boats are quite large, making larger RMSE values more tolerable. Table II presents the values of r and n for each vessel type as discovered by this preprocessing step.

VI. Empirical Analysis

Next, we report results from an empirical study of the proposed methodology using a genetic algorithm (GA) towards fine-tuning of parameters for trajectory compression.

A. Experimental Setup

The Synopses Generator[1] is developed in Scala using the DataStream API of *Apache Flink* with *Apache Kafka* as broker for streaming messages. The genetic algorithm for fine-tuning the parameters of Synopses Generator is implemented in Python, using the evolutionary framework DEAP [2]. All tests were conducted on an Intel® Core™ i7-4790 CPU @ 3.60GHz machine with 16GB RAM.

B. Experimental Results

Regarding the training cost of GA, we measured the time required to train five sixths of the dataset, for each run of the 6-fold cross-validation. Table II reports the average training time and the standard deviation of the cost among runs. As expected, there is a direct correspondence between dataset size (in terms of AIS messages per vessel type) and this cost.

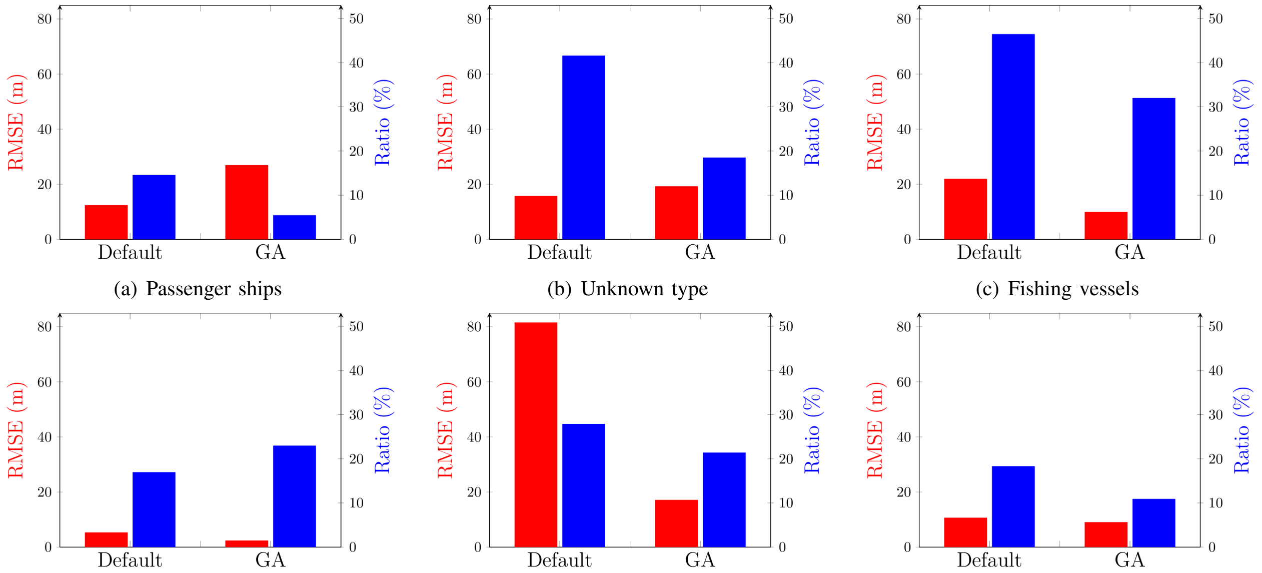
With respect to compression efficiency, Figure 3 compares results for *RMSE* (red bars) and compression *Ratio* (blue bars) between the default parametrization and the fine-tuned one suggested by the GA. Table III lists the fine-tuned parameter values for each vessel type, i.e., those that achieved the lowest score according to (1) over all folds.

1) Passenger ships: These generally follow straight courses at open seas, making their synopses easier to maintain. The default parameters achieve a rather low RMSE of about 15m. As passenger ships often generally have quite large sizes, both in width and in length, such low RMSE values are rather strict and lead to poorer compression. GA takes advantage of this, by only increasing RMSE to an acceptable value just under

[1] https://github.com/DataStories-UniPi/Trajectory-Synopses-Generator

TABLE II
DATASET AND GENETIC ALGORITHM CONFIGURATION PER VESSEL TYPE

	Brest dataset information		*Threshold*		*Hyper-parameters*		*Training Cost*
Vessel type	*AIS messages*	*Vessel count*	*RMSE*	*Ratio*	r	n	*Mean time* ± *Standard deviation*
Passenger	4,792,487	17	30m	10%	17	0.8	5.2 hours ± 16 minutes
Unknown	3,466,765	115	15m	15%	10	1.0	4.5 hours ± 8 minutes
Fishing	3,288,577	161	30m	30%	17	0.7	4.5 hours ± 39 minutes
Tug	1,411,761	15	15m	15%	2	1.6	1.8 hours ± 4 minutes
Cargo	1,198,228	184	30m	10%	13	0.8	1.5 hours ± 2 minutes
Military	802,045	12	15m	15%	10	1.4	1 hour ± 3 minutes



(a) Passenger ships

(b) Unknown type

(c) Fishing vessels

(d) Tug boats

(e) Cargo ships

(f) Military vessels

Fig. 3. Comparison of compression efficiency for the main vessel types available in the dataset. In each plot, the two bars on the left concern the use of the default parameter values, while the two bars on the right represent averages of measurements over all folds from the genetic algorithm (GA).

30m and manages to keep only a third of the critical points in the synopses. This improvement is possible according to the thresholds specified in Table II.

Regarding the fine-tuned parameters (Table III), note that less effort is made to keep critical points. The values for *angle threshold* ($\Delta\theta$) and *speed ratio* (α) are much higher than the default ones, leading to less detailed, yet fairly accurate synopses compared to the ones created with the default values.

2) Vessels of unknown type: Default parametrization yields a fair RMSE at the expense of a very poor compression, which keeps more than 60% of the original AIS locations. In contrast, GA provides a comparable RMSE and halves the number of critical points. This effect should be attributed to the unsuitability of default values for this subset of the data. From Table III, it is clear that GA relaxes the *angle threshold ($\Delta\theta$)* and suggests a lower value for *speed ratio (α)*. Instead, under the default parametrization (Table I), the resulting synopses unnecessarily keep too many critical (especially *turning*) points, whereas several *change-in-speed* points are missed.

3) Fishing vessels: In general, such trajectories are far from straight, thus a larger fraction of original points must be kept as critical to sustain a reasonable approximation error. With the default parameters, both RMSE and compression ratio are high compared to other vessel types. Instead, the optimal parameter values discovered by the GA offer much better synopses: Ratio is close to the desired threshold of 30%, while RMSE falls substantially under the threshold value of 30 meters.

This considerable improvement is mostly due to suitable selection of smaller values for *buffer size* (m) and *distance threshold* (D) in Table III, compared to the default ones (Table I). Since fishing boats often make zig-zag or circular manoeuvres, larger values in these parameters (as in the default configuration) may lead to inaccurate estimates of their mean velocity, thus yielding many extra critical points.

4) Tug boats: Tug boats move mainly inside the port at low speed, making it easier to create an accurate synopsis with few points. This is also noticeable in the very small RMSE and low compression ratio achieved with the default values. GA achieves an even better RMSE with only a small increase of the compression ratio. Note that the ratio value slightly exceeds the threshold of 15%, most likely because the dataset used for hyper-parameter tuning might not be enough representative of the motion patterns overall.

Fine-tuning suggests very small values for parameters *angle threshold* ($\Delta\theta$) and *speed ratio* (α) in Table III, probably due to the simple motion paths observed for this type of vessels.

TABLE III
PARAMETERS PICKED BY THE GENETIC ALGORITHM PER VESSEL TYPE

Parameter	*Default*	*Passenger*	*Unknown*	*Fishing*	*Tug*	*Cargo*	*Military*
$\Delta\theta$	4	10.71	17.58	18.99	4.96	17.5	11.68
ΔT	1800	200	400	200	450	2500	2600
m	5	50	21	3	29	3	3
ω	3600	2750	1500	3550	2300	1750	4800
v_{min}	0.5	2.0	1.52	0.41	0.84	0.81	0.88
D	50	76.51	44.45	23.97	2.0	15.12	22.96
α	0.25	0.63	0.01	0.01	0.01	0.01	0.01
v_θ	5	4.58	1.02	0.61	6.06	0.82	0.45

This might seem surprising given that the ratio threshold was set to 20% only, but it actually makes sense: tug boats move slowly with few changes in heading and speed, thus rarely generating critical points, even with low parameter values.

5) Cargo ships: The pattern of movement for such vessels is different from other types. This is confirmed by the unusually high RMSE and moderate compression ratio achieved with default parametrization. GA improves a little the compression ratio, but manages to reduce RMSE by a large margin.

These results actually validate our main argument: different mobility patterns require different degrees of compression. Admittedly, default values may offer fair compression for several vessel types, but not for the larger and bulkier cargo vessels. GA is able to "learn" this different behaviour, and subsequently chooses suitable values for the parameters in order to offer more reliable synopses. Note in Table III that the *distance threshold* (D) takes a very low value. Like in most other vessel types, the *speed ratio* (α) value is as low as possible. Instead, under default parametrization, α was relatively high and the synopses failed to keep enough *change-in-speed* critical points, hence the unusually increased RMSE.

6) Military vessels: Just like tug boats, movement of military ships is also rather simple, as manifested by the low RMSE and ratio that the default parameters can offer. With the GA, we manage to decrease both measures by a small amount, while also well satisfying the thresholds set in Table II. The fine-tuned parametrization (Table III) suggests a surprisingly low value for the *low speed threshold* (v_θ). This means that GA discovered that such vessels often move at speeds higher than 0.45 knots, so it considers slow motion less noteworthy for the synopses. In case the value of v_θ was set higher, more critical points would be generated, scarcely offering any extra details on the approximate course, but worsening the compression ratio without any significant benefit on RMSE.

VII. SUMMARY AND FUTURE WORK

We have been developing a trajectory detection module that can provide summarized representations of vessel trajectories by consuming AIS positional messages online. In this paper, we presented an approach for fine-tuning the selection of parameter values for this module. Taking into account the type of each vessel, we employed a genetic algorithm converging to a suitable configuration of parameters that can yield improved trajectory synopses, both in terms of approximation error and compression ratio. Our empirical analysis showed that compression efficiency may be better than the one under default parametrization, without resorting to time-consuming data inspection. In future work, we plan to investigate the effects of automated parameter value selection on the performance of composite maritime event recognition [13] based on such compressed trajectories. Further fine-tuning of parameters may be also possible by considering the fluctuating frequency of updates and the diverse motion patterns per vessel.

ACKNOWLEDGMENTS

This work has received funding from the EU Horizon 2020 RIA program INFORE under grant agreement No 825070.

REFERENCES

[1] H. Cao, O. Wolfson, and G. Trajcevski, "Spatio-temporal data reduction with deterministic error bounds," *VLDB Journal*, vol. 15, no. 3, pp. 211–228, 2006.

[2] F.-A. Fortin, F.-M. De Rainville, M.-A. Gardner, M. Parizeau, and C. Gagné, "DEAP: Evolutionary algorithms made easy," *Journal of Machine Learning Research*, vol. 13, pp. 2171–2175, jul 2012.

[3] C. Iphar, A. Napoli, and C. Ray, "Detection of false AIS messages for the improvement of maritime situational awareness," in *OCEANS*, 2015, pp. 1–7.

[4] R. Lange, F. Dürr, and K. Rothermel, "Efficient real-time trajectory tracking," *VLDB Journal*, vol. 20, no. 5, pp. 671–694, 2011.

[5] X. Lin, S. Ma, H. Zhang, T. Wo, and J. Huai, "One-pass error bounded trajectory simplification," *PVLDB*, vol. 10, no. 7, pp. 841–852, 2017.

[6] X. Lin, J. Jiang, S. Ma, Y. Zuo, and C. Hu, "One-pass trajectory simplification using the synchronous Euclidean distance," *VLDB Journal*, vol. 28, no. 6, pp. 897–921, 2019.

[7] J. Liu, K. Zhao, P. Sommer, S. Shang, B. Kusy, J. Lee, and R. Jurdak, "A novel framework for online amnesic trajectory compression in resource-constrained environments," *IEEE Trans. Knowl. Data Eng.*, vol. 28, no. 11, pp. 2827–2841, 2016.

[8] C. Long, R. C.-W. Wong, and H. Jagadish, "Trajectory simplification: On minimizing the direction-based error," *PVLDB*, vol. 8, no. 1, pp. 49–60, 2014.

[9] N. Meratnia and R. de By, "Spatiotemporal compression techniques for moving point objects," in *EDBT*, 2004, pp. 765–782.

[10] J. Muckell, P. Olsen, J.-H. Hwang, C. Lawson, and S. Ravi, "Compression of trajectory data: A comprehensive evaluation and new approach," *Geoinformatica*, vol. 18, no. 3, pp. 435–460, 2014.

[11] K. Patroumpas, E. Alevizos, A. Artikis, M. Vodas, N. Pelekis, and Y. Theodoridis, "Online event recognition from moving vessel trajectories," *GeoInformatica*, vol. 21, no. 2, pp. 389–427, 2017.

[12] K. Patroumpas, N. Pelekis, and Y. Theodoridis, "On-the-fly mobility event detection over aircraft trajectories," in *ACM SIGSPATIAL*, 2018, pp. 259–268.

[13] M. Pitsikalis, A. Artikis, R. Dréo, C. Ray, E. Camossi, and A.-L. Jousselme, "Composite event recognition for maritime monitoring," in *DEBS*, 2019, pp. 163–174.

[14] M. Potamias, K. Patroumpas, and T. Sellis, "Sampling trajectory streams with spatiotemporal criteria," in *SSDBM*, 2006, pp. 275–284.

[15] C. Ray, R. Dréo, E. Camossi, A.-L. Jousselme, and C. Iphar, "Heterogeneous Integrated Dataset for Maritime Intelligence, Surveillance, and Reconnaissance," *Data in Brief*, vol. 25, 2019.

[16] F. Terroso-Saenz, M. Valdés-Vela, E. den Breejen, P. Hanckmann, R. Dekker, and A. F. Skarmeta-Gómez, "CEP-traj: An event-based solution to process trajectory data." *Information Systems*, vol. 52, pp. 34–54, 2015.

[17] G. A. Vouros, C. Doulkeridis, G. Santipantakis, A. Vlachou, N. Pelekis, H. Georgiou, Y. Theodoridis, K. Patroumpas, E. Alevizos, A. Artikis, G. Fuchs, M. Mock, G. Andrienko, N. Andrienko, C. Ray, C. Claramunt, E. Camossi, A.-L. Jousselme, D. Scarlatti, and J. M. Cordero, "Big data analytics for time critical maritime and aerial mobility forecasting," in *EDBT*, 2018, pp. 612–623.

[18] O. Wolfson, A. Sistla, S. Chamberlain, and Y. Yesha, "Updating and querying databases that track mobile units," *Distrib. Parallel Dat.*, vol. 7, no. 3, pp. 257–287, 1999.

[19] D. Zhang, M. Ding, D. Yang, Y. Liu, J. Fan, and H. T. Shen, "Trajectory simplification: An experimental study and quality analysis," *PVLDB*, vol. 11, no. 9, pp. 934–946, 2018.