engineering proceedings

*Proceeding Paper*

# Extended Object Tracking (EOT) Performance Comparison for Autonomous Driving Applications †

**Tolga Bodrumlu, Mehmet Murat Gozum and Abdurrahim Semiz ***

AVL Research and Engineering, Istanbul, Turkey; murat.hozum@avl.com (M.M.G.); abdurrahim.semiz@avl.com (A.S.)

* Correspondence: tolga.bodrumlu@avl.com

† Presented at the 10th International Electronic Conference on Sensors and Applications (ECSA-10), 15–30 November 2023; Available online: https://ecsa-10.sciforum.net/.

**Abstract:** Extended object tracking is crucial for autonomous driving, as it enables vehicles to perceive and respond to their environment accurately by considering an object's shape, size, and motion over time. Two commonly used methods for extended object tracking, Joint Probabilistic Data Association (JPDA) and Gaussian Mixture Probability Hypothesis Density (GM-PHD), were compared in autonomous vehicles using radar data. Both JPDA and GM-PHD perform well in tracking multiple extended objects, but GM-PHD demonstrates a performance advantage, especially in terms of the Generalized Optimal Sub-Pattern Assignment (GOSPA) metric, which measures the accuracy of tracked object positions in comparison to their actual positions.

**Keywords:** extended object tracking; radar; GM-PHD; JPDA

## 1. Introduction

Radar technology has revolutionized the way that gather information about the surrounding environment, providing valuable insights into the movement and position of the objects. One of the most important applications of radar technology is the extended object tracker (EOT), which is a sophisticated system that enables the detection and tracking of multiple objects simultaneously.

The EOT is a critical component of many modern systems, including air traffic control, weather monitoring, and autonomous driving. By using advanced algorithms and sophisticated hardware, these systems can accurately identify and track the position, speed, and direction of objects in real-time, even in challenging weather conditions. The main idea of the EOT is to track the kinematics and estimate the size of the objects nearby the ego vehicle [1]. As per [2] the shape is often as-sumed to be rigid, i.e., non-changing. Tracking an extended object is in general a highly complex problem due to the non-linearity of the resulting estimation problem.

Various tracking algorithms have been developed for EOT. One popular approach is the multi-hypothesis tracking (MHT) algorithm [3] which generates multiple hypotheses about the object's position and shape based on the measurements. Another approach is the joint probabilistic data association (JPDA) algorithm [4], which assigns probabilities to each measurement-object association and updates the probabilities over time. More recently, deep learning-based object detectors have been proposed, such as the Cen-terNet [5] and YOLO [6] algorithms, which use convolutional neural networks to predict the object's position and size.

One notable work in this area is the method proposed by Granström et al. [7] , who propose a probabilistic approach to extended object tracking based on the concept of random matrices. Their method also uses a Bayesian filter to estimate the object's state and shape, and it is shown to be effective in tracking vehicles in cluttered environments.

check for updates

**Citation:** Bodrumlu, T.; Gozum, M.M.; Semiz, A. Extended Object Tracking (EOT) Performance Comparison for Autonomous Driving Applications. *Eng. Proc.* **2023**, *56*, 0. https://doi.org/

Academic Editor: Firstname Lastname

Published: 15 November 2023

Another approach is presented by Ba-Ngu Vo and Wing-Kin Ma [8], who propose a GM-PHD (Gaussian Mixture Probability Hypothesis Density) filter for extended object tracking. Their method uses a probability hypothesis density to represent the object's state and shape and is shown to be effective in tracking multiple objects with varying shapes and sizes.

In this paper, two methods are compared for the purpose of extended object tracking which are JPDA and GM-PHD. The main reason for comparing these two algorithms here is to see which algorithm's results are at the desired level and to determine the algorithm we will use in real-time applications in the future. Both algorithms are coded in the MATLAB environment. The Driving Scenario Designer Toolbox [9] of MATLAB is used for testing the implemented algorithms. The created scenarios include an ego vehicle and three surrounding vehicles. Measurements are taken from the radar on the ego vehicle, and the designed algorithms are executed and compared with ground truth data.

The remainder of this paper is structured as follows. Firstly, details of both algorithms are explained, then the Generalized Optimal Sub-Pattern Assignment metric which is defined for comparison is given. Finally, the simulation system and results are pre-sented and future work is discussed.

## 2. Extended Object Tracking

### *2.1. Joint Probabilistic Data Association Tracker*

The JPDA algorithm is a statistical approach used to solve the problem of data association in multiple target tracking [4]. The problem of data association involves assigning observations to a set of targets, where the observations are potentially noisy.

The JPDA algorithm considers all possible combinations of the observations and targets, and computes the probability of each association based on the likelihood of the observations given the targets, as well as the prior probability of each target being present. These probabilities are then combined using Bayes' rule to obtain the posterior probability of each target being associated with each observation.

The JPDA Tracker is an implementation of the JPDA algorithm for multiple target tracking. The main steps of the algorithm are summarized as follows:

- Initialization: Initialize the target states and their associated covariances, and set the prior probabilities of each target being present.
- Prediction: Predict the state and covariance of each target based on a motion model, and update the prior probabilities of each target being present.
- Data Association: Compute the likelihood of each observation given each target, and use the JPDA algorithm to compute the posterior probability of each target being associated with each observation
- Track Update: Update the state and covariance of each target based on the associated observations, and compute the likelihood of each track.
- Track Management: Decide whether to create a new track for an unassociated observation, or terminate a track if its likelihood falls below a certain threshold.

The general structure of the JPDA filter flowchart shown Figure 1.

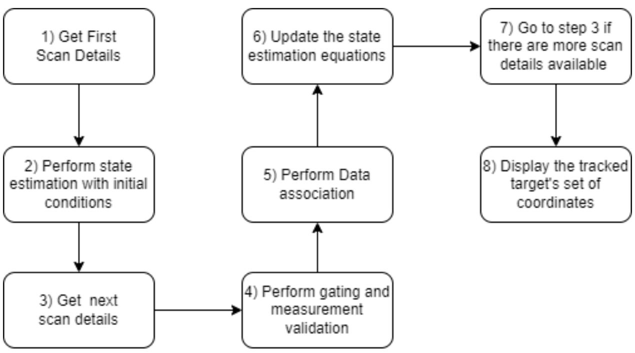


**Figure 1.** JPDA Tracker Flowchart.

*2.2. Gaussian Mixture Probability Hypothesis Density Filter*

The GM-PHD filter is a Bayesian filtering algorithm that is commonly used for multiple target tracking. It is an extension of the Probability Hypothesis Density (PHD) filter, which is a non-parametric approach to multi-target tracking that represents the state of the targets using a single probability density function.

The GM-PHD filter extends the PHD filter by modeling the probability density function of the targets as a mixture of Gaussian components. Each Gaussian component corresponds to a potential target, and its mean and covariance represent the state and uncertainty of the target.

The GM-PHD filter consists of two main steps: prediction and update.

- Prediction: The means and covariances of the Gaussian components are propagated using a motion model. The weights of the Gaussian components are also updated based on the predicted probability of each target being present.
- Update: The filter incorporates the measurement information by calculating the likelihood of each measurement given each Gaussian component. The weights of the Gaussian components are then updated based on the likelihood and the predicted probability of each target being present.

The GM-PHD filter is efficient and scalable, as it can handle a large number of targets and clutter measurements. It also provides a probabilistic representation of the estimated target states, which can be used for decision-making and sensor fusion. General structure of the GM-PHD filter flowchart is shown Figure 2.

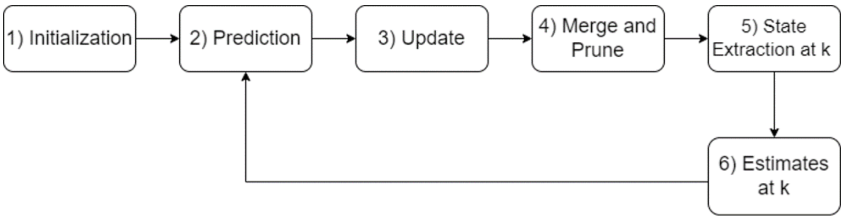


**Figure 2.** GM-PHD Flowchart.

## 3. Comparison Metric

The GOSPA (Generalized Optimal Sub-Pattern Assignment) metric was proposed by Rahmathullah et al. [10] as a way to evaluate the performance of extended object tracking algorithms. The metric compares the estimated and true object trajectories and provides a quantitative measure of the tracking accuracy. It is a generalization of the Optimal Sub-Pattern Assignment (OSPA) metric that considers not only the distances between individual detections/tracks, but also the cardinality differences between the sets being compared.

The GOSPA metric measures the similarity between two sets of detections/tracks based on the distances between their individual elements, as well as the number of false alarms and missed detections. The metric is defined by four parameters: $p$, $c$, $\alpha$, and $\beta$. The parameter $p$ determines the order of the distance metric used to compare individual detections/tracks (e.g., $p = 2$ corresponds to Euclidean distance). The parameters $c$, $\alpha$ , and $\beta$ control the trade-off between the cardinality differences and the distances between the sets.

The GOSPA metric is a useful tool for evaluating the performance of multiple object tracking algorithms, as it provides a comprehensive measure of their accuracy and robustness.The code structure of the GOSPA algorithm explained Algorithm 1 briefly.

**Algorithm 1:** GOSPA Metric.

**Input:** Set of predicted tracks $P$ and set of true tracks $T$
**Output:** GOSPA distance
$g = (c, p, fp, m)$; **for** *each true track* $t \in T$ **do**
  $D_t =$ ; **for** *each predicted track* $p \in P$ **do**
    $D_t(p) =$ calculate distance between $t$ and $p$;
  $D_t = \text{sort}(D_t)$; **if** $|D_t| > m$ **then**
    $D_t = D_t(1 : m)$;
  $c_t = \sum_{d \in D_t} d$; $m_t = |D_t|$; $c = c + c_t$; $m = m + m_t$;
$p = |P|$; $fp = p - m$; $g = (c, p, fp, m)$; $GOSPA = \sqrt{(g_c + \alpha^2 g_p + \beta^2 g_{fp} + \gamma^2 g_m)}$;
**return** $GOSPA$;

## 4. Simulation System and Results

### *4.1. Simulation System*

MATLAB environment is used as the simulation system. Both JPDA Tracker and GM-PHD algorithm are implemented in MATLAB environment. Implemented algorithms have been tested with radar data obtained from Driving Scenario Toolbox. The studies are carried out on a single radar data on the ego vehicle. In the tested scenarios, the number of vehicles in the environment is fixed.

In order to provide more reliable measurement values from surrounding vehicles to the algorithm and to make more accurate decisions about which measurement belongs to which vehicle, a clustering algorithm has been applied to the obtained radar measurements. The clustering algorithm is selected as Density-Based Spatial Clustering of Applications with Noise (DBSCAN)[11]. This clustering algorithm takes two parameters as input along with the incoming measurement values. These two parameters are a radius value $\epsilon$ and a minimum number of points required to form a dense region (minPts). In the simulations $\epsilon$ value is taken as 20 and minPts value is taken as 3.

The simulation scenario created in Driving Scenario Designer Toolbox[9]. In this scenario, the vehicles around the ego vehicle do not only move on a straight road. To make the scenario a bit more complex, lane changes were introduced and the vehicles were brought closer together. In the next section, the implementation of both JPDA and GM-PHD algorithms to the simulation system and the results obtained are shown.

#### 4.1.1. JPDA Tracker

In this section JPDA Tracker algorithm which is explained in section 2.1 is implemented in the MATLAB environment.The studies are carried out on a single radar data on the ego vehicle.

A covariance ellipse is created by using the velocity and position of the ego vehicle with the radar data. These created ellipses are also seen in Figure 3. Parameters used in the JPDA Tracker algorithm and ellipse generation and their definitions are shown in the Table 1. Initial values of these parameters are given as follows.

$$C_v = \text{diag}(10{,}10), \quad C_{rw} = \text{diag}(10{,}10{,}1{,}1),$$
$$A_p = \text{diag}(1{,}1{,}1), \quad C_r = \text{diag}(0.3{,}0.3{,}0.1{,}0.1)$$
$$A_r = \begin{bmatrix} 1 & 0 & 0.05 & 0 \\ 0 & 1 & 0 & 0.05 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix},$$
$$C_p = \text{diag}(0.2{,}0.1{,}0.1)$$

**Table 1.** JPDA Paramaters.

| Symbol | Definition |
|---|---|
| $C_v$ | Measurement Noise Covariance |
| $C_{rw}$ | Error Covariance for Kinematic State |
| $A_r$ | Transition for Kinematic State |
| $A_p$ | Transition for Shape Parameters |
| $C_r$ | Pos x,y and Vel x,y Covariance Matrix |
| $C_p$ | Major, minor axes and azimuth angle of ellipse |

When looking at the Figure 3, it can be seen that the position estimations of the vehicles surrounding the ego vehicle are quite close to the ground truth data. However, it is observed that errors in position estimation begin to increase as the vehicles get close to each other. When the vehicles are close to each other data association part can be a problem due to the distinguishing part.

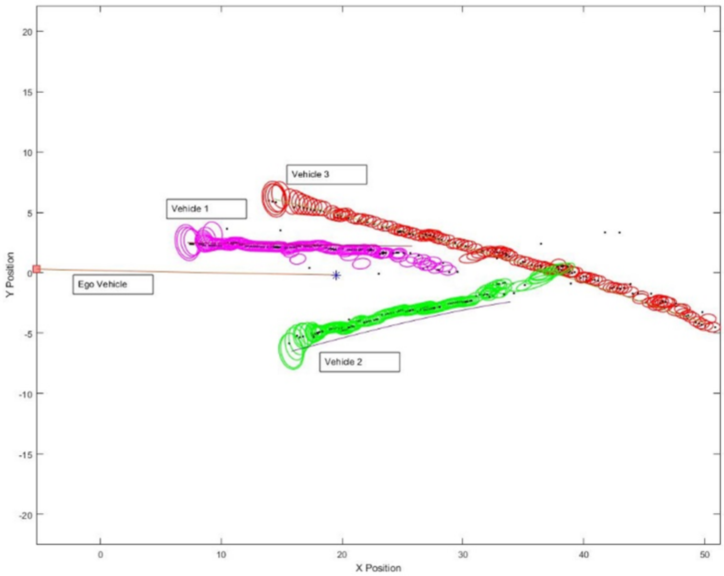


**Figure 3.** Tracking Results Obtained for JPDA Algorithm.

#### 4.1.2. GM-PHD

In this section GM-PHD algorithm which is explained in Section 2.2 is implemented in the MATLAB environment. Studies are carried out on a single radar data on the ego vehicle. The data used here is the same as the data used in the JPDA algorithm. Some of the using parameters and their definitions are shown in the Table 2. The initial values of these parameters are also given as follows.

**Table 2.** GM-PHD Paramaters.

| Symbol | Definition |
|---|---|
| $v_{sigma}$ | Q noise matrix amplitude |
| $P_s$ | Survival probability |
| $P_d$ | Probability of detection |
| $w_{birth}$ | Weighting of birth terms |
| $m_{birth}$ | Mean of birth terms |
| $P_{birth}$ | Covariance of birth terms |

$$v_{sigma} = 1, \quad P_s{=}0.99, \quad P_d = 0.99$$
$$w_{birth} = 0.03, \quad P_{birth} = \mathrm{diag}(1,1,1,1)$$

$$m_{birth} = \begin{bmatrix} 14.400 & 15.800 & 7.200 \\ 5.832 & 2.901 & 2.999 \\ 6.100 & -6.500 & 2.500 \\ -1.407 & 0.7636 & -0.057 \end{bmatrix}$$

When looking at the Figure 4, it can be seen that the position estimations of the vehicles surrounding the ego vehicle are quite close to the ground truth data. It can be observed that even when vehicles approach each other, the predictions made do not shift to other vehicles. It can easily be said that the data association part works more effectively in the GM-PHD algorithm when compared with Figures 3 and 4.

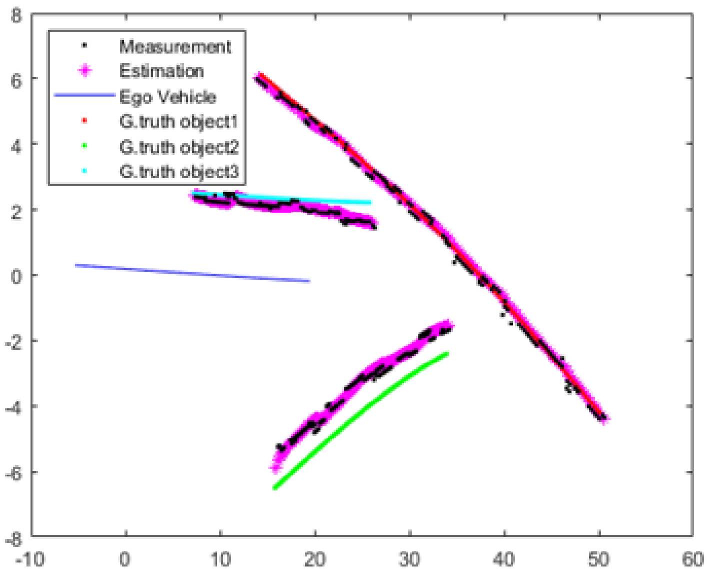


**Figure 4.** Tracking Results Obtained for GM-PHD Algorithm.

### *4.2. Performance Comparison of JPDA and GM-PHD Algorithms*

The results obtained in Sections 4.1.1 and 4.1.2 qualitatively illustrates the good performance for both algorithms. However, we need a quantitative metric which shows how accurate the performance of each algorithm. The GOSPA metric defined in section 3 is used to illustrate the tracking performance of JPDA and GM-PHD algorithms. Based on their values (the performance gets better as GOSPA value decreases), the performances of the algorithms are compared. The parameters p and c are chosen to be 2 and 100, respectively. Figure 5 illustrates the comparison of GOSPA values for the tracking performance of JPDA and GM-PHD algorithms.

The GOSPA values comparison illustrates that both algorithms are good at keeping track of the surrounding objects. For both of them, the GOSPA values are quite low. It can be seen that the GOSPA values obtained with GM-PHD are much lower than the ones obtained with JPDA. The GOSPA values oscillates between 1 and 1.5 for GM-PHD whereas the GOSPA values obtained with JPDA is varying between 1.5 and 5. In brief, the GOSPA results quantitatively illustrates that the tracking performance of GM-PHD is superior than JPDA.

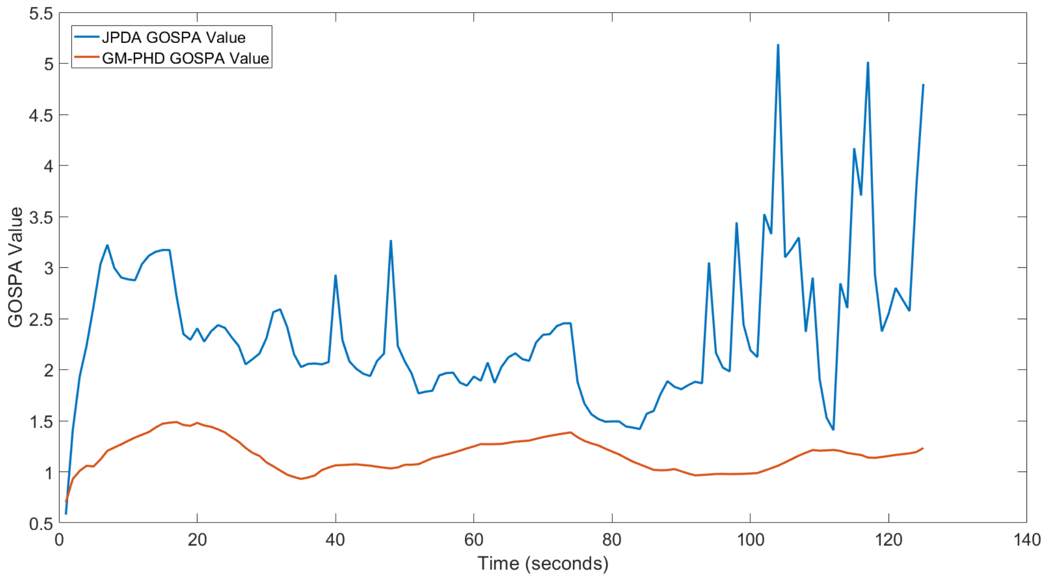


**Figure 5.** The GOSPA values comparison of JPDA and GM-PHD algorithms for tracking performance.

## 5. Conclusions

In this paper, JPDA and GM-PHD methods are compared with each other for the purpose of extended object tracking. Both methods have advantages and disadvantages with respect to each other. In autonomous driving applications, there was no study which compares the performance of the both methods based on only radar data. GOSPA metric was chosen to compare the performance of GM-PHD and JPDA. The simulation platform was chosen to be MATLAB Driving Toolbox.

Simulation results show that both algorithms are good at keeping track of the surrounding vehicles which is understood from the low GOSPA values at all time stamps of the simulation. It is also clearly observed that GOSPA values obtained with GM-PHD are much lower than the GOSPA values obtained with JPDA which shows the superiority of GM-PHD over JPDA based on the performance. Although it is known that GM-PHD is able to keep track of the objects when the number of objects is varied unlike JPDA, GM-PHD is not optimal to estimate the cardinality of the objects when the newborn object is modeled as static and not measurement-driven. In the future, we aim to keep track of the surrounding objects by using adaptive-birth GM-PHD algorithm in the real-time implementations for the autonomous driving applications.

**Author Contributions:** Conceptualization, T.B., M.M.G. and A.S.; Data gathering, T.B., M.M.G. and A.S.; Formal analysis, T.B. and M.M.G.; Investigation, T.B., M.M.G. and A.S.; Methodology, T.B., M.M.G. and A.S.; Project administration, T.B.; Resources, T.B.; Supervision, T.B., M.M.G.; Validation, T.B., A.S. and M.M.G.; Visualization, T.B. and A.S.; Writing—original draft, T.B., M.M.G. and A.S.; Writing—review and editing, T.B., and M.M.G. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research received no external funding.

**Conflicts of Interest:** The authors declare no conflict of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript, or in the decision to publish the results.

## References

1. Granstrom, K.; Baum, M. A tutorial on multiple extended object tracking. *techrxiv* **2022**.
2. Granstrom, K.; Baum, M.; Reuter, S. Extended object tracking: Introduction, overview and applications. *J. Adv. Inf. Fusion* **2017**, *12* , 139–174. https://doi.org/10.48550/arXiv.1604.00970
3. Reid, D.B. An algorithm for tracking multiple targets. *IEEE Trans. Autom. Control* **1979**, *24* , 843–854. https://doi.org/10.1109/TAC.1979.1102177.

4. Fortmann, T.; Bar-Shalom, Y.; Scheffe, M. Sonar tracking of multiple targets using joint probabilistic data association. *IEEE J. Ocean. Eng.* **1983**, *8* , 173–184. https://doi.org/10.1109/JOE.1983.1145560.
5. Zhou, X.; Wang, D.; Krahenbuhl, P. Objects as points. *arXiv* **2019**. https://doi.org/10.48550/arXiv.1904.07850.
6. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You only look once: Unified, real-time object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016. https://doi.org/10.48550/arXiv.1506.02640.
7. Granstrom, K.; Lundquist, C.; Orguner, U. Extended Target Tracking using a Gaussian-Mixture PHD filter. *IEEE Trans. Aerosp. Electron. Syst.* **2012**, *48*, 3268–3286. https://doi.org/10.1109/TAES.2012.6324703.
8. Vo, B.-N.; Ma, W.-K. Gaussian mixture probability hypothesis density filter for multi-target tracking. *IEEE Transactions Signal Processing* **2006**, *54*, 4091–4104. https://doi.org/10.1109/TSP.2006.881190.
9. MathWorks. *Automated Driving Toolbox*™. MathWorks: Natick, MA, USA, 2021.
10. Rahmathullah, A.S.; Garcia-Fernandez; Svensson, L. Generalized optimal sub-pattern assignment metric. In Proceedings of the 20th International IEEE Conference on Information Fusion, Xi'an, China, 10–13 July 2017. https://doi.org/10.23919/ICIF.2017.8009645.
11. Ester, M.; Kriegel, H.-P.; Sander, J.; Xu, X. A density-based algorithm for discovering clusters in large spatial databases with noise. In Proceedings of the 2nd International Conference on Knowledge Discovery and Data Mining, Portland, OR, USA, 2–4 August 1996. https://doi.org/10.5555/3001460.3001507.