Scenario Complexity for Unmanned Aircraft System Traffic

Min Xue*
NASA Ames Research Center, Moffett Field, CA 94035

Minh Do†
SGT Inc., NASA Ames Research Center, Moffett Field, CA 94035

A quick and accurate assessment of complexity for a given traffic scenario can help plan and schedule flights to alleviate traffic bottleneck and mitigate operation risks, especially for unmanned aircraft system traffic management where high traffic density or complexity is expected. This work introduces a traffic scenario complexity metric that was constructed based on the number of potential conflicts weighted by the conflict resolution cost associated. The cost associated with a conflict is calculated based on the corresponding conflict resolution maneuvers. To obtain the conflict resolution maneuvers, a Mixed Interger Linear Programming (MILP) based optimization is formulated with the vehicle model and conflict management parameters involved. To evaluate the complexity metrics, an approach of using measurements from high-fidelity simulations is proposed. The complexity measurements for 920 randomly-generated scenarios were generated through high-fidelity simulations and treated as the ground truth. Two statistical methods: Pearson and Alternative Conditional Expectations were applied for analysis. The results show that the metric using the number of flights has low correlation with the scenario complexity according to the correlation coefficients calculated by both methods. The Alternative Conditional Expectations method shows that the proposed metric has better correlation with the complexity than the metric using the number of potential conflicts.

I. Introduction

In conventional aviation, where traffic is managed by air traffic controllers, measuring and predicting airspace complexity can assist planning traffic flow changes, sector level staffing needs, and other operational decisions. For UAS traffic, a good and quick approximation of traffic complexity can help assess traffic scenarios, re-plan flight routes and schedules to alleviate traffic bottleneck, and mitigate operation risk. It can also help categorize traffic senarios for traffic management studies. Real-time or near real-time complexity assessment is needed for both research and operations in UAS traffic management.

Simple complexity metrics like aircraft count or traffic density can be misleading, especially for cases when traffic is highly organized, e.g. a set of aircraft fly parallelly where the aircraft count is high but the effort level of managing traffic is low. Over the past years, methods have been developed based on the cognitive complexity, such as Dynamic Density (DD) metric [1–3], where key features are weighted according to controller's workload ratings and a weighted sum of these key features is then used as the final measurement. Meanwhile, complexity metrics based on the intrinsic traffic disorder were developed as well to reveal the complexity in a sector regardless of controller's workload. Delahaye et. al. [4] proposed a geometrical approach based on the properties of relative positions and relative speeds of aircraft in a sector to obtain time histories of traffic divergence, convergence, and sensitivity. By modeling aircraft trajectories as a linear dynamical system, Delahaye et. al. [4, 5] also developed an aggregate metric using the entropy of the dynamic system. Based on that, the velocity vector field methods [5, 6] were developed to compute complexity maps for given traffic scenario snapshots.

However, measuring UAS traffic complexity is different from traditional traffic. First, it is likely there will be no human controller to maintain separations, therefore, controller workload rating can not be used to help construct cognitive complexity metrics. Second, while air traffic controllers resolve conflicts based on existing procedures and rules in conventional aviation, autonomous UASs follow their multi-point trajectory plans and perform avoidance maneuvers guided by a predefined conflict management model. Finally, since a large number of UASs are envisioned to operate in a given traffic scenario, the complexity metric needs to be computed near real-time for hundreds of vehicles.

*Aerospace Research Engineer, Aviation Systems Division. Mail Stop 210-15. AIAA senior member.
†Research Scientist, Planning & Scheduling Group, Intelligent System Division, Mail Stop 269-3.

In recent UTM related studies, Bulusu et. al. [7] studied the relationship between capacity and flow rate using three conflict management models to obtain an initial understanding of the operational capacity of UAS traffic, however, the scope of the study did not include intrinsic complexity of a given traffic scenario caused by the schedule and layout of flight plans. Meanwhile, the Fe^3 [8] simulator was developed to compute traffic statistics through high-fidelity Monte Carlo simulations. Although the simulator has the capability of providing accurate traffic complexity measurements, its performance may not be suitable for a real-time application. In this paper, a complexity metric is developed based on the number of potential conflicts weighted by the resolution cost associated with each conflict. Using the simulation mesurements from the Fe^3 simulator as the ground truth for compleixty, the proposed metric was compared against several popular metrics based on the hundreds of random-generated scenarios. The results showed that the correlation between the proposed metric is better than the other metrics.

In this work, Section II introduces the new complexity metric and the methods used to generate the metric are described as well including the mathematical programming formulation for solving pairwise conflicts. Section III presents the experiments, evaluations, and discussion. Metrics were compared against the measurements from the high-fidelity simulations. Section IV concludes the study.

II. Methodology

The number of potential conflicts and the number of aircraft within a distance threshold are popular metrics in the past studies [1–3, 9]. In order to differentiate difficulties involved in various encounters, indirect features have been proposed in dynamic density methods [1–3], such as the angle of convergence, the time to conflict, the horizontal proximity, and conflict resolution difficulty based crossing angle. In the linear dynamic system approach, Delahaye et. al. [5] constructed matrices for different types of encounters and used eigenvalues of these matrices as an indicator of the degree of organization of each encounter.

To safely and efficiently handle a large volume of UAS traffic, the future UAS traffic management system is envisioned to be managed by predefined autonomous conflict managemnet services (centralized or decentralized) instead of air traffic controllers. This opens up the possibility of taking advantage of the autonomous conflict management model when computing the difficulty associated with each conflict. While different encounters or conflicts (e.g. small angle crossing vs. head-on conflict) introduce different complexities, different conflict management models (e.g. different parameter setting, heading change vs. speed change) may also result in difference in complexity. Incorporating conflict management models can help accurately compute the cost involved in an encounter.

In this section, the general conflict management structures together with parameters are first introduced. A mathematic programming formulation incorporating vehicle model and conflict management parameters is then presented to compute pairwise conflict resolution maneuvers. Finally, the complexity metric built upon the conflict resolution maneuvers are proposed.

A. Conflict Management Model

A typical conflict management model applies three phases when resolving conflicts [10]: strategic conflict management, separation provision, and collision avoidance. The strategic conflict management changes flight plans to resolve conflicts and normally occurs prior to departure. The separation provision is a tactical process for conflicts predicted while airborne to prevent well-clear violations. The collision avoidance is triggered when the well-clear definition is violated and serves as the last layer of conflict resolution. Figure 1 shows the typical conflict management model structure (not to scale), which includes separation provision and collision avoidance. The Near Mid-Air Collision (NMAC) represents the last layer of separation required to avoid physical contacts. The well-clear threshold d_{WC} provides minimal space for collision avoidance for which both ownship and intruder can take actions as the last effort before colliding with each other and serves as the boundary between the collision avoidance and separation provision phases. The look-ahead time (LAT) defines the time horizon for trajectory prediction. The conflict warning threshold (CWT) defines the time t_{CWT} when a conflict warning becomes effective. The conflict resolution threshold (CRT) defines the time t_{CRT} when the ownship can start to maneuver to resolve the conflict. Although a heading-change resolution was shown in the figure, this structure can be utilized for other conflict resolution options such as speed change and altitude change. Apparently, different settings in conflict management models will yield different complexity. For instance, a larger value of conflict resolution threshold will require more space to resolve a conflict thus yield higher complexity for a given traffic scenario. Also a conflict management model with heading change only will demand a larger airspace than a model with additional speed changes. Since developing the well-clear definition is still an open research question [11, 12], a distance of 50 feet is arbitrarily chosen for d_{WC} in this study. The conflict resolution threshold t_{CRT} is set to 30 seconds.

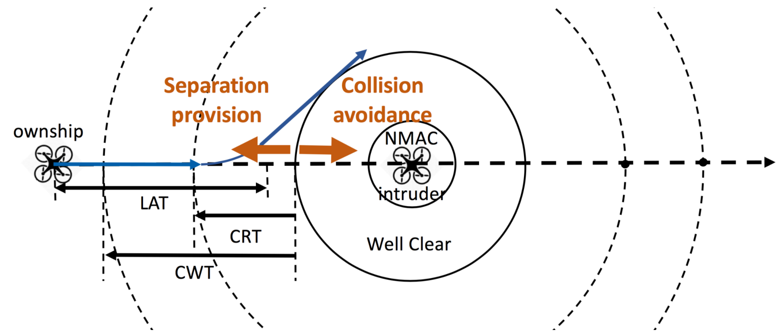


Fig. 1 Structure used in the conflict resolution algorithms

B. Method to Compute Pairwise Conflict Resolutions

A mixed integer linear programming was formulated to compute minimum deviations needed for mitigating pairwise conflicts in a traffic scenario when a large amount of vehicles are involved. Over the past years, many Mixed Integer Linear Programming (MILP) based methods have been proposed to solve dynamic obstacle avoidance path planning and conflict-free trajectory planning for UASs [13–21]. These studies showed that solving conflict-free trajectories in long horizon for multiple aircraft demands expensive computational time, which makes any application with a large number of aircraft prohibitive. However, because only a quick approximation is required for measuring a traffic scenario complexity, computing the minimum deviation for each pairwise conflict should be sufficient. Therefore, a MILP problem is formulated to solve a large amount of pair-wise conflicts to achieve the real-time computation performance.

1. Vehicle Trajectory Model

The aircraft trajectory can be characterized by a discrete-time linear model [17, 18, 20] (Eqn. 1) and associated initial conditions (Eqn. 2), where $[p_{x,t}, p_{y,t}]$ respesent the position vector at time t for an aircraft. $[v_{x,t}, v_{y,t}]$ and $[a_{x,t}, a_{y,t}]$ are velocity and acceleration vectors at time t, respectively, and Δt is the time step used in the formulation. $[x_0\ y_0\ \dot{x}_0\ \dot{y}_0]$ are initial position and velocity of the aircraft.

$$\forall\, t \geq 0,$$
$$\begin{bmatrix} p_{x,t+1} \\ p_{y,t+1} \\ v_{x,t+1} \\ v_{y,t+1} \end{bmatrix} = \begin{bmatrix} 1 & 0 & \Delta t & 0 \\ 0 & 1 & 0 & \Delta t \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} p_{x,t} \\ p_{y,t} \\ v_{x,t} \\ v_{y,t} \end{bmatrix} + \begin{bmatrix} \frac{1}{2}\Delta t^2 & 0 \\ 0 & \frac{1}{2}\Delta t^2 \\ \Delta t & 0 \\ 0 & \Delta t \end{bmatrix} \begin{bmatrix} a_{x,t} \\ a_{y,t} \end{bmatrix} \tag{1}$$

$$\begin{bmatrix} p_{x,0} & p_{y,0} & v_{x,0} & v_{y,0} \end{bmatrix}^{\mathrm{T}} = \begin{bmatrix} x_0 & y_0 & \dot{x}_0 & \dot{y}_0 \end{bmatrix}^{\mathrm{T}} \tag{2}$$

In order to capture the aircraft dynamics, dynamics constraints including speed, acceleration, and turn rate are applied. A maximum speed v_{max} is imposed and the quadratic constraints $\sqrt{v_x^2 + v_y^2} \leq v_{max}$ are linearized using a K-sided polygon technique proposed in Richards's work [13] to make the computation efficient (shown in Eqn. 3), where K can be adjusted to meet a given precision requirement. Ten constraints with $K = 10$ should provide sufficient approximation in application.

$$\forall\, i\in[1...N],\ \forall\, t\in[t_{si}, t_{ei}],\ \forall\, k\in[1...K]:$$
$$v_{x,i,t} \cdot \sin(\frac{2\pi k}{K}) + v_{y,i,t} \cdot \cos(\frac{2\pi k}{K}) \leq v_{max} \tag{3}$$

To introduce a minimum speed v_{min}, the speed vector has to lie outside of a K-sided polygon with at least one of the binary variables $c_{t,k}$ being nonzero as shown in Eqn. 4, where M is a sufficient large positive number to disable a constraint when the binary variable is one.

$$\begin{gathered}\forall\ i{\in}[1...N],\ \forall\ t{\in}[t_{si},t_{ei}],\ \forall\ k{\in}[1...K]:\\ v_{x,i,t}\cdot\sin(\frac{2\pi k}{K})+v_{y,i,t}\cdot\cos(\frac{2\pi k}{K})\geq v_{min}-M{\cdot}c_{t,k}\\ \sum_{k=1}^{K}c_{t,k}\leq K-1\\ c_{t,k}\in[0,1]\end{gathered}\tag{4}$$

The maximum turn rate ω_{max} is introduced and approximated through the maximum acceleration using the product of the maximum turn rate and the maximum speed [13, 15]:

$$a_{max}=v_{max}\cdot\omega_{max}\tag{5}$$

And to satisfy the maximum acceleration constraint a_{max}, the acceleration vector needs to stay inside of a K-sided polygon:

$$\begin{gathered}\forall\ i{\in}[1...N],\ \forall\ t{\in}[t_{si},t_{ei}],\ \forall\ k{\in}[1...K]:\\ a_{x,i,t}\cdot\sin(\frac{2\pi k}{K})+a_{y,i,t}\cdot\cos(\frac{2\pi k}{K})\leq a_{max}\end{gathered}\tag{6}$$

2. Separation Constraint

Besides vehicle dynamics constraints, aircraft have to satisfy a minimum separation distance from each other to avoid collisions. A well-clear distance d_{wc} is required for the minimum separation, therefore, quadratic separation distance constraints are introduced to maintain well-clear among aircraft. The same K-sided polygon technique is applied to linearize the quadratic separation constraints. The relative position vector has to stay outside of the polygon with at least one zero binary varilable $c_{t,k}$.

$$\begin{gathered}\forall\ i,j{\in}[1...N],\ \forall\ i{\neq}j,\forall\ t{\in}[t_{si},t_{ei}],\ \forall\ k{\in}[1...K]:\\ (p_{x,i,t}-p_{x,j,t})\cdot\sin(\frac{2\pi k}{K})+(p_{y,i,t}-p_{y,j,t})\cdot\cos(\frac{2\pi k}{K})\geq d_{wc}-M{\cdot}c_{t,k}\\ \sum_{k=1}^{K}c_{t,k}\leq K-1\\ c_{t,k}\in[0,1]\end{gathered}\tag{7}$$

3. Finite Horizon

To reduce the computational time, the finite horizon technique is applied to limit the time horizon of the MILP formulation to a time window during which a conflict resolution maneuver happens. According to the general conflict resolution structure described in Section II.A, the time horizon of each pair-wise conflict can be reduced to a finite range: $[t_{collision}-t_{crt},t_{collision}+t_{crt}]$ (shown in Fig. 2), where the $t_{collision}$ is calculated based on the original flight plan.

4. Objective

Finally, within the finite horizon, the objective function is to minimize the deviation from aircraft's original path as shown in Eqn. 8, where t_{si} and t_{ei} denotes the $t_{collision}-t_{crt}$ and $t_{collision}+t_{crt}$ for aircraft i, respectively. And a trajectory point at time t in the original path is represented by $q_{i,t}$.

$$\min_{\mathbf{p},\mathbf{a}}\ J=\sum_{i=1}^{N}\sum_{t=t_{si}}^{t_{ei}}\sqrt{(p_{x,i,t}-q_{x,i,t})^2+(p_{y,i,t}-q_{y,i,t})^2}\tag{8}$$

To linearize the final objective and avoid the quadratic equations for computing distances, the distances are expressed as an additional set of linear constraints for any aircraft i as in Eqn. 9:

$$\begin{gathered}\forall\ i{\in}[1,N],\forall\ t{\in}[t_{si},t_{ei}],\ \forall\ k{\in}[1...K]:\\ (p_{x,i,t}-q_{x,i,t})\cdot\sin(\frac{2\pi k}{K})+(p_{y,i,t}-q_{y,i,t})\cdot\cos(\frac{2\pi k}{K})\leq d_{i,t}\end{gathered}\tag{9}$$

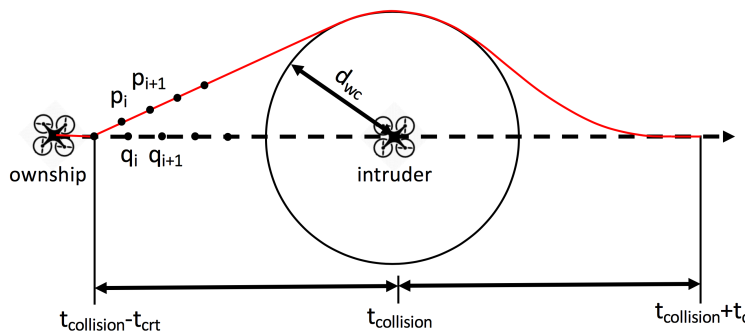


Fig. 2 Using finite horizon based on conflict management structure

and the objective can then be linearized as:

$$\min_{\mathbf{p},\mathbf{a}} J = \sum_{i=1}^{N} \sum_{t=t_{si}}^{t_{ei}} d_{i,t} \tag{10}$$

With the formulation (Eqn. 1-7, 9, 10), the conflict resolution maneuvers for a large number of pairwise conflicts can then obtained. Besides t_{crt} and d_{WC}, this formulation can be adjusted easily to a different conflict management model. For instance, if the conflict management model only allows heading changes, it can be achieved by setting v_{min} and v_{max} closely. This MILP model is implemented in C/C++ and solved by Solving Constraint Integer Programs (SCIP) [22, 23], an open-source optimization suites supporting MILP, quadratic, and several other types of constraint. The experiments show that for a large random set of scenarios with the number of flights increased from 5 to 100 flights, the median and average running time are 15 and 20 seconds on a Core I7 Mac laptop, respectively. Considering there are many approaches, such as using commerical MILP solvers, parallel computing, etc, reducing the computational time to near real-time should be feasible.

C. Scenario Complexity Metric

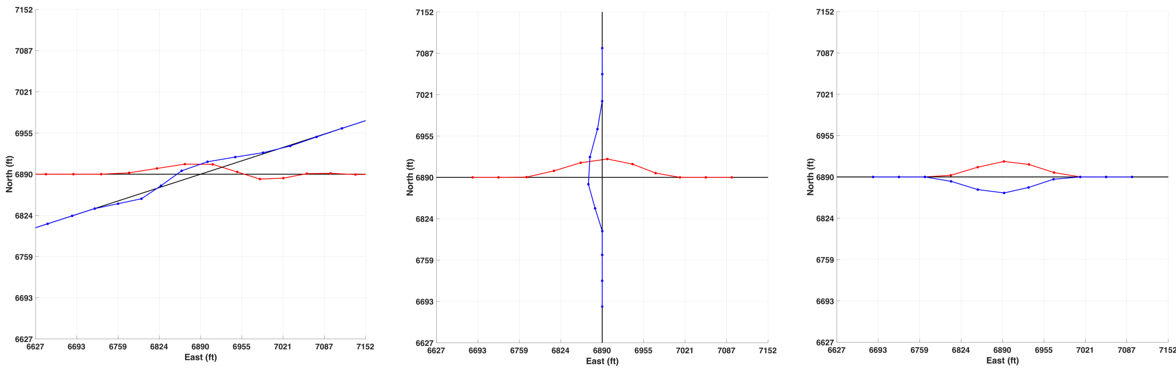
The cost of a pairwise conflict is defined as the sum of deviations from the original path in addition to the conflict count (shown in Eqn. 11), where C_{fi} and C_{fj} are the deviation costs associated with two involved flights, respectively. The deviation cost for a flight involved in a conflict is defined as the temporal deviation, which is calculated by the percentage of the distance deviation multiplied by the conflict resolution duration. p_{t_s} and p_{t_e} are the beginning and ending points of the resolution maneuver for a flight, respectively. $\overline{p_t p_{t+1}}$ denotes the distance between p_t and p_{t+1}, while $\overline{p_{t_s} p_{t_e}}$ is the distance between the beginning point p_{t_s} and the ending point p_{t_e}.

$$\begin{aligned} C &= 1.0 + C_{fi} + C_{fj} \\ C_{f_m} &= \left| \frac{\sum_{t=t_s}^{t_e} \overline{p_t p_{t+1}} - \overline{p_{t_s} p_{t_e}}}{\overline{p_{t_s} p_{t_e}}} \right| \cdot (t_e - t_s),\ where\ m = i\ or\ j \end{aligned} \tag{11}$$

Figure 3(a), 3(b), and 3(c) present the conflict resolution maneuvers computed by the MILP formulation in three different encounter situations: crossing with a small angle, crossing with a large angle, and a head-on conflict. According to the calculation using Eqn. 11, the costs for these three encounters are: 1.26, 1.32, and 1.29, respectively. It shows that with the setup of MILP formulation, the large-angle crossing has a higher cost than the head-on encounter, while the small-angle crossing has a lower cost than the other two.

The final measurement of scenario complexity metric SC can then be expressed as Eqn. 12, where nc is the number of potential conflicts associated with the original flight schedule in a given scenario without any conflict resolutions.

$$SC = \sum_{k=1}^{nc} C_k \tag{12}$$



(a) Crossing with a small angle (b) Crossing with a large angle (c) A head-on conflict

Fig. 3 Conflict resolution maneuvers in three different encounters

III. Experiments

In this section, random scenarios were first generated for experiments. Then the metric called the number of resolution maneuvers was produced by the Fe^3 [8] simulator. Using the simulation generated measurements as the ground truth for the scenario complexity, the proposed complexity metric SC was then analyzed and compared using statistical methods.

A. Test Scenarios

To evaluate complexity metrics, random scenarios with a large variety of complexities need to be generated. Several criteria were used to ensure high traffic intensity and comparability in scenarios: First, a 1.3x1.3 nautical mile region is defined (shown as the red box in Fig. 4) and all flights are required to go through the predefined region with origin and destination outside of the region; Second, at most one turning point is allowed other than the origin and destination in a flight plan; Third, all flights are set to depart within a five-minute window; Lastly, the target ground speeds of all flights are in the range of 5 meter per second and 20 meter per second. Fig. 4 shows a sample scenario with 30 vehicles, where circle, cross, diamond markers represent origins, destinations, and mid-points respectively.

The number of aircraft in these scenarios varies from 5 to 50 (or in density from 3 to 30 $vehicle/nmi^2$). Fig. 5 shows the percentage of scenarios with and without conflicts during the process of generating scenarios. When the traffic density increases, the likelihood of having conflicts increases and reaches 100 % around 15 $vehicle/nmi^2$. As scenarios without conflicts will have zero scenario complexity based on the proposed metric, therefore, only the scenarios have have potential conflicts are used in experiments. As a result, a total of 920 scenarios were created and used in this work with 20 scenarios at each level of density.

B. Evaluation

The Fe^3 [8] simulator has the capability of simulating traffic systems by incorporating critical models in an air traffic system, which includes vehicle models, conflict management models, wind model, communication, navigation, and surveillance models. Detailed description of those models and capabilities can be found in previous work [8]. The simulator can provide statistical measurements for metrics such as losses of seperation, number of conflict maneuvers, extra flight distance, number of conflict warning, and energy consumed. Since this study focuses on scenario intrinsic complexity, uncertainties on wind, communication, navigation, and surveilance were not included in the simulations. The deterministic measurements of the number of conflict resolution maneuvers was used as the ground truth for complexity. The number of conflict resolution maneuvers measures the resolution moves issued during the simulation. As the time step size in Fe^3 is 0.5 seconds, the number of conflict resolution maneuvers also reflects the resolution duration.

The Pearson method in R [24] was used to compute the correlations. Figure 6(a), 6(b), and 6(c) show the correlation plots for three matrices: the number of flights, the number of potential conflicts, and the proposed complexity metric SC. The correlation coefficients with the number of conflict resolution maneuvers (measurements from the Fe^3 simulator)

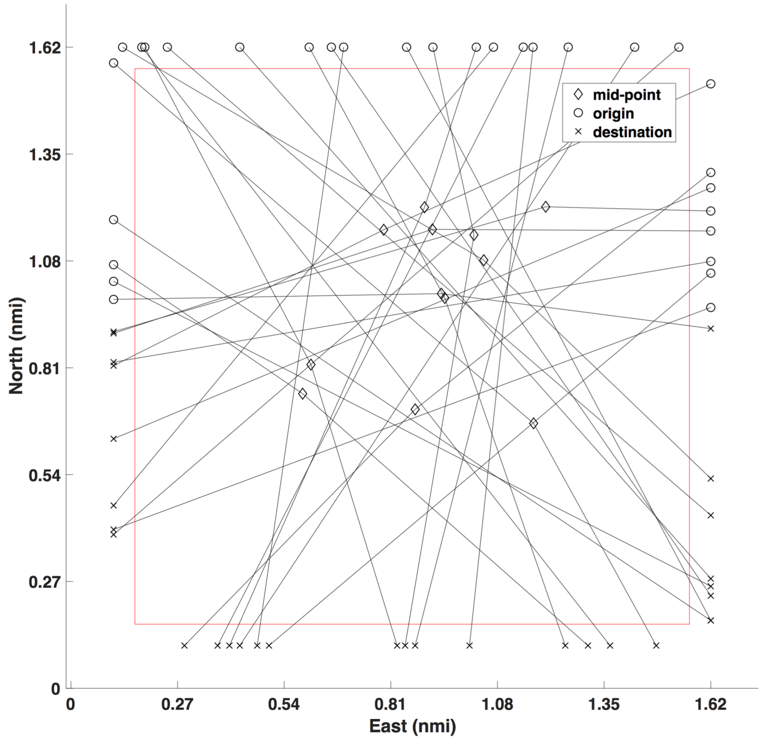


Fig. 4 A Sample Scenario with 30 vehicles

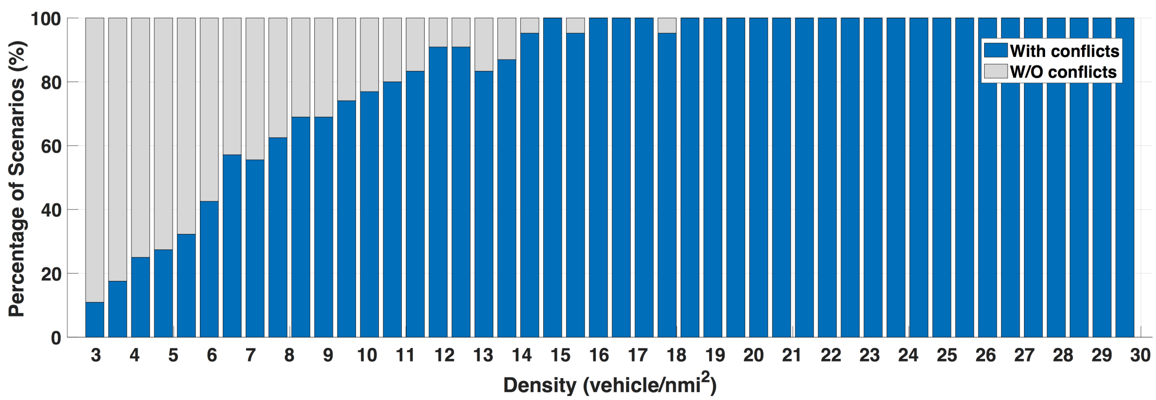
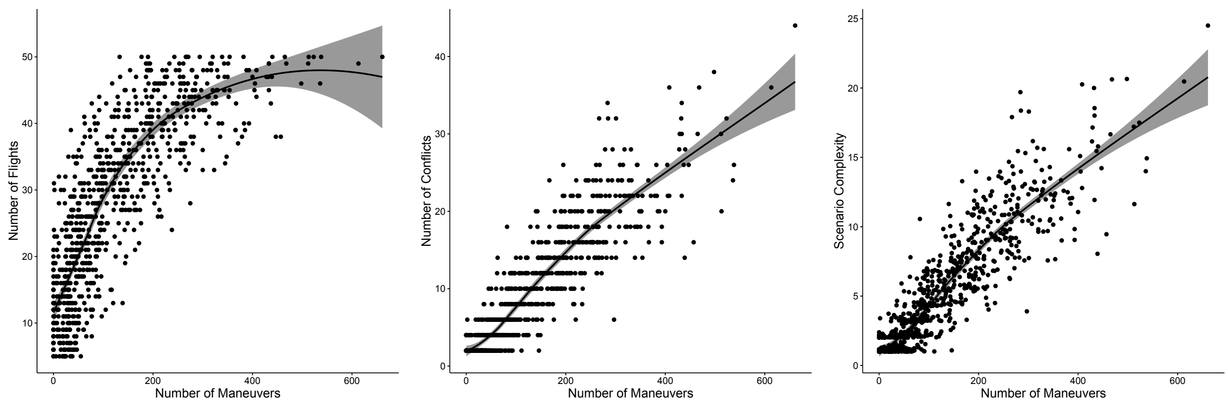


Fig. 5 Likelihood of Conflicts at Different Density Levels

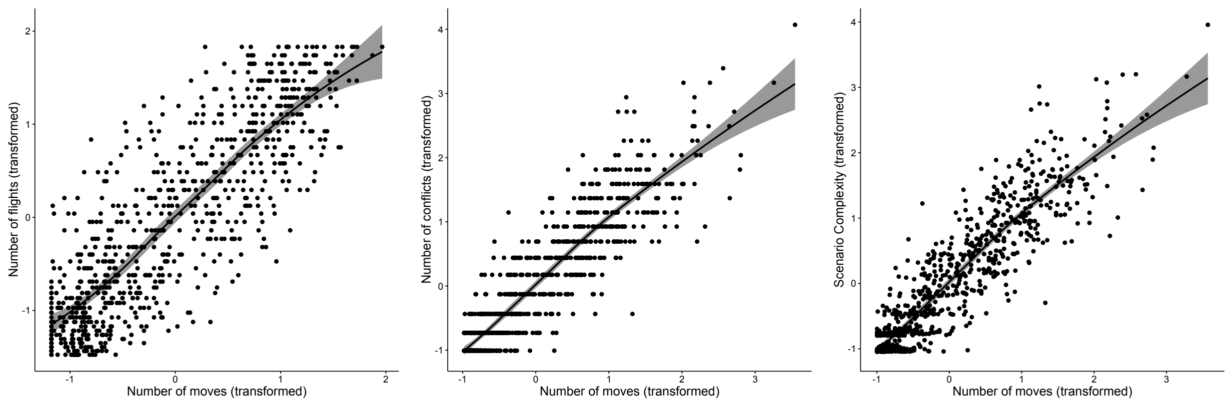
are 0.78, 0.9, and 0.9, respectively. p values for all three correlation tests are less than $2.2x10^{-16}$, which suggests with high significance that all three metrics have positive correlation with the complexity produced by simulations. The number of conflicts and the complexity metric SC have much higher correlation than the number of aircraft.

As the Pearson method is designed for checking linear correlations, therefore, a maximal correlation method [25] was used to capture non-linear association and the Alternative Conditional Expections (ACE) package [26] in R was used to compute such maximal correlations. Figure 7(a), 7(b), and 7(c) present the plots after applying transformation in the ACE package. The correlation coefficients for number of flights, number of conflicts, and scenario complexity are then 0.85, 0.908, and 0.913, respectively. The correlation coefficients suggest that the complexity metric SC is more correlated to the number of resolution maneuvers that were generated by the Fe^3 simulator than the number of conflicts.



(a) Number of flights (R = 0.78) (b) Number of conflicts (R = 0.9) (c) Senario Complexity (R=0.9)

Fig. 6 Pearson Correlation with the Number of Conflict Resolution Maneuvers



(a) Number of flights (R = 0.85) (b) Number of conflicts (R = 0.908) (c) Senario Complexity (R=0.913)

Fig. 7 Maximal Correlations with the Number of Conflict Resolution Maneuvers

C. Discussion

The analysis shows that the number of flights (or density) is much less correlated with the complexity compared to the other two metrics. For instance, as shown in Fig. 6(a), it is common that 40 flights (23 flights/nmi^2) may share the same complexity as 5 flights (3 flights/nmi^2), which makes the number of flights (or density) a bad indicator of complexity. On the other hand, by taking into account the complexity associated with each conflict, the proposed complexity metric *SC* showed better correlation than the number of potential conflicts. However, the proposed metric did not improve the correlation coefficient that much as it is essentially derived from the number of potential conflicts and shares a similar pitfall: the consequences of these conflicts are simply missed. For instance, there might exist secondary conflicts as the conflict resolution maneuvers changed the original trajectories. On the other hand, even though the conflict management structure and parameters were considered, it is still not exactly the same as the one applied in the simulations. Many other factors that affect the complexity are also missed or not as accurate as in the simulator, such as vehicle trajectory model, wind, communication, uncertainty and so on. The Fe^3 simulator was developed to address all these factors, whereas, the goal of this study is to identify quickly-computed metrics that can approximate the complexity as close as possible. From this perspective, the proposed complexity metric *SC* is the best compared to the other two. In addition, using the measurements from high-fidelity simulations as the ground truth, this work developed a mechanism of testing and evaluating any proposed complexity metric for future work.

IV. Conclusion

In UAS traffic management, a quick assessment of complexity for a given traffic scenario can help re-plan flights to alleviate traffic bottleneck and mitigate operation risks, and the traffic complexity measurement can also help categorize traffic scenarios for traffic management studies. In this work a complexity metric was constructed based on the number of potential conflicts weighted by the associated resolution costs. The cost associated with a conflict is calculated based on the corresponding conflict resolution maneuvers. A MILP-based optimization was formulated to obtain the conflict resolution maneuvers. To assess and compare complexity metrices, around 1,000 scenarios at different density level were generated. The Fe^3 simulator was used to run these scenarios and generate a metric measurement called the number of conflict resolution maneuvers for each scenario. The measurements from the simulations were then treated as the ground truth for the scenario complexity. Statistical tools were applied to examine the correlations for three metrics: the number of flights, the number of potential conflicts, and the proposed complexity metric. The analysis showed that the number of flights has much lower correlation with the scenario complexity than the other two according to the correlation coefficients calculated by both Pearson and ACE statistics methods. The ACE maximal correlation method shows that the proposed complexity metric has better correlation with the ground truth than the number of potential conflicts. With the simulation based measurements, future work will focus on investigating if combined simple features can better represent scenario complexity through regressions.

References

[1] Sridhar, B., Sheth, K., and Grabbe, S., "Airspace Complexity and its Application in Air Traffic Management," *2nd USA/Europe Air Traffic Management R&D Seminar*, Orlando, FL, 1998.

[2] Kopardekar, P., and Magyarits, S., "Measurement and Prediction of Dynamic Density," *The 21st Digital Avionics Systems Conference (DASC)*, Irvine, CA., 2002.

[3] Masalonis, A., Callaham, M., and Wanke, C., "Dynamic Density and Complexity Metrics for Realtime Traffic Flow Management," *5th USA/Europe Air Traffic Management R&D Seminar*, Budapest, Hungary, 2003.

[4] Delahaye, D., and Puechmorel, S., "Airspace Complexity: Towards Intrinsic Metrics," *3rd USA/Europe Air Traffic Management R&D Seminar*, Napoli, Italy, 2000.

[5] Delahaye, D., Puechmorel, S., Hansman, J., and Histon, J., "Air Traffic Complexity based on Non Linear Dynamical Systems," *Air Traffic Control Quarterly*, Vol. 12, No. 4, 2004, pp. 367–388.

[6] Ishutkina, M. A., and Feron, E., "Describing Air Traffic Complexity Using Mathematical Programming," *AIAA 5th Aviation, Technology, Integration, and Operations Conference (ATIO)*, 2005.

[7] Bulusu, V., Sengupta, R., Mueller, E. R., and Xue, M., "A Throughput-Based Capacity Metric for Low-altitude Airspace," *AIAA Aviation Forum*, Atlanta, GA., 2018.

[8] Xue, M., Rios, J., Silva, J., Ishihara, A., and Zhu, Z., "Fe3: An Evaluation Tool for Low-Altitude Air Traffic Operations," *AIAA Aviation Forum*, Atlanta, GA., 2018.

[9] Klein, A., "An Efficient Method for Airspace Analysis and Partitioning Based on Equalized Traffic Mass," *6th USA/Europe Seminar on Air Traffic Management Research and Development*, Baltimore, MD, 2005.

[10] Johnson, M., and Larrow, J., "UTM Research Transition Team Sense and Avoid Working Group Technical Work Package #2: UTM Conflict Management Model," Tech. Rep. Technical Report, NASA, 2018.

[11] Duffield, M. O., and McLain, T. W., "A Well Clear Recommendation for Small UAS in High-Density ADS-B-Enabled Airspace," *AIAA SciTech Forum*, Grapevine, Texas, 2017.

[12] Weinert, A., Campbell, S., Vela, A., Schuldt, D., and Kurucar, J., "Well-Clear Recommendation for Small Unmanned Aircraft Systems Based on Unmitigated Collision Risk," *AIAA Journal of Air Transportation*, Vol. 26, No. 3, 2018.

[13] Richards, A., and How, J. P., "Airspace Trajectory Planning with Collision Avoidacen Using Mixed Integer Linear Programming," *Proceedings of the 2002 American Control Conference*, Anchorage, AK, 2002.

[14] Bellingham, J. S., Tillerson, M., Alighanbari, M., and How, J. P., "Cooperative Path Planning for Multiple UAVs in Dynamic and Uncertain Environments," *Proceedings of the 41st IEEE Conference on Decision and Control*, Las Vegas, NV, 2002.

[15] Schouwenaars, T., How, J., and Feron, E., "Receding Horizon Path Planning with Implicit Safety Guarantees," *Proceedings of the 2004 American Control Conference*, Boston, MA, 2004.

[16] Ma, C. S., and Miller, R. H., "MILP Optimal Path Planning for Real-Time Applications," *Proceedings of the 2006 American Control Conference*, Minneapolis, MN, 2006.

[17] Forsmo, E. J., Grotli, E. I., Fossen, T. I., and Johansen, T. A., "Optimal Search Mission with Unmanned Aerial Vehicles using Mixed Integer Linear Programming," *2013 International Conference on Unmanned Aircraft Systems (ICUAS)*, Atlanta, GA, 2013.

[18] Radmanesh, M., Kumar, M., Nemati, A., and Sarim, M., "Dynamic Optimal UAV Trajectory Planning in the National Airspace System via Mixed Integer Linear Programming," *Journal Aerospace Engineering*, Vol. 230, No. 9, 2016, pp. 1668–1681.

[19] Alonso-Ayuso, A., Escudero, L. F., and Martin-Campo, F. J., "Exact and Approximate Solving of the Aircraft Collision Resolution Problem via Turn Changes," *Transportation Science*, Vol. 50, No. 1, 2014, pp. 1–12.

[20] Radmanesh, M., Nemati, A., Sarim, M., and Kumar, M., "Flight Formation of Quad-Copters In Presence of Dynamic Obstacles Using Mixed Integer Linear Programming," *Proceedings of the ASME 2015 Dynamic Systems and Control Conference*, Columbus, OH, 2015.

[21] Cafieri, S., and Omheni, R., "Mixed-integer Nonlinear Programming For Aircraft Conflict Avoidance by sequentially Applying Velocity and Heading Angle Changes," *European Journal of Operational Research*, Vol. 260, No. 1, 2017, pp. 283–290.

[22] Achterberg, T., Berthold, T., Koch, T., and Wolter, K., "Constraint Integer Programming: A New Approach to Integrate CP and MIP," *Integration of AI and OR Techniques in Constraint Programming for Combinatorial Optimization Problems*, edited by L. Perron and M. A. Trick, Springer Berlin Heidelberg, Berlin, Heidelberg, 2008, pp. 6–20.

[23] Achterberg, T., "SCIP: solving constraint integer programs," *Mathematical Programming Computation*, Vol. 1, No. 1, 2009, pp. 1–41. doi:10.1007/s12532-008-0001-1, URL `https://doi.org/10.1007/s12532-008-0001-1`.

[24] R Core Team, *R: A Language and Environment for Statistical Computing*, R Foundation for Statistical Computing, Vienna, Austria, 2016. URL `http://www.R-project.org/`, ISBN 3-900051-07-0.

[25] Beriman, L., and Friedman, J. H., "Estimating Optimal Transformations for Multiple Regression and Correlation," *Journal of the American Statistical Association*, Vol. 80, No. 391, 1985, pp. 580–598. doi:10.1080/01621459.1985.10478157.

[26] Spector, P., Friedman, J., Tibshirani, R., Lumley, T., and andJonathan Baron, S. G., *ACE and AVAS for Selecting Multiple Regression Transformations*, 2016. R package version 1.4.1 — For new features, see the 'Changelog' file (in the package source).