Simplified Conjunction Analysis using a Graph Database for Identifying High Risk Objects

Janet Cathell
Sceptre Analytics, Inc.
Anthony Holincheck
Sceptre Analytics, Inc.

ABSTRACT

Debris generating collisions are severe threats to space flight safety. Recently, McKnight et al. have called attention to specific objects in Low Earth Orbit (LEO) that represent an enhanced threat to the space environment. We present a schema using graph database technology to assist in the identification of other objects that have a high probability of collision or are otherwise problematic. With the new USSF space fence contributing observations, full N-on-N conjunction analysis of the space catalog has the potential to generate millions of possible collision events per day. By collecting multiple days worth of potential events it becomes possible to perform analysis of which objects are frequent offenders. In an operational environment where orbital neighbors under active control maneuver away from one object and potentially into the path of others, it becomes important to know not only the threats to primary objects but also the threats to objects in nearby orbits. Graph databases allow for the collection of information in a set of nodes (space objects) and relationships (conjunctions). Modern graph databases easily scale to hold billions of nodes and relationships. Graph database queries allow for easy generation of the list of objects that most frequently have a potential collision with an object. Through relationship queries that are optimized in graph DBs, it becomes trivial to also identify the list of objects that are threats to objects that are threats, etc. An open source implementation of the schema with sample conjunction analysis data is presented.

1. INTRODUCTION

Satellite operators perform conjunction assessments (CAs) to identify potential close approaches or collisions involving their own satellites. Several organizations also perform CAs for the entire set of payloads in orbit against the list of all objects including payloads, rocket bodies, and debris [1]. The USSF's 18 SPCS also shares conjunction predictions for the full N-on-N or all objects-against-all objects in the space catalog. This includes conjunctions where both objects are debris. For CAs to be most useful, they are typically forward looking predictions with days of lead time. This allows operators of active payloads time to plan and execute an avoidance maneuver if necessary. A typical CA will involve the use of best available ephemeris data, an estimate of orbit uncertainty, physical sizes of objects involved, and sophisticated probability of collision modeling. The current work is not about improving CA calculations but about analyzing a collection of CA results.

We propose collecting and storing a history of CA results and conjunction predictions in a flexible graph database. The set of past results can then be analyzed to identify high-threat objects. As shown by [2], certain objects pose a threat to the space environment shared by all operators, even if they are not an immediate collision threat to any specific object. A single-object break up event can also generate dangerous debris. We look at the entire set of objects and close approaches to help identify not only direct threats to specific payloads, but also indirect threats. As we enter an era where debris removal missions become possible [3] multiple debris mitigation strategies become cost-effective, especially compared to the operating and potential replacement cost of mega constellations.

The probability of collision, P_c, is the most common metric of collision risk. The need to move beyond the analysis of discrete conjunction events was described by [4]. We believe that the flexibility of graph databases allows for the creation of multiple risk metrics that can be tailored to the needs of individual operators. We present our approach for generating 12 months of single-day CA results along with our graph DB schema for storing them. We then present different queries and graph-analytic calculations which can then be performed on the data to identify objects that pose a high risk to the space environment.

Table 1: Number of conjunctions for various minimum distance values.

Minimum Distance (km)	Number of Conjunctions
10	460148
5	232936
2	94819
1	48981
0.5	21984
0.1	2387

2. CONJUNCTION SIMULATIONS

We implemented a basic technique for calculating the conjunctions for the full N-on-N catalog of objects that had TLEs available from the Space-Track.org website. For each day in 2020, we identified the most up-to-date TLE for each object in orbit. The TLE for each object was used to compute the Cartesian state vector for the object at 30 second intervals. Each state vector includes the three-dimension position and velocity of the object. The objects' minimum and maximum distance from the earth were computed for the 24 hour period along with maximum velocity magnitude. This was used to establish a basic apogee and perigee filter where objects that did not have overlapping heights during the day could be excluded for further consideration. For every 30 second time step, each object was compared with every other object to see if their relative distance was close enough for a potential collision. The distance threshold was computed by

$$D_{thresh} = 50\text{ km} + (T_s * (V_{max1} + V_{max2}))$$

T_s is the step size in seconds. V_{max} refers to each object's maximum velocity during the day in km per second. For objects with distances below that threshold a fine-grain mode was entered where the relative distance was checked every 0.1 s. The conjunctions were then identified with a start and stop time equal to when the distance crossed a 10 km threshold. The minimum distance and time of closest approach were also determined at the 0.1 s resolution. The conjunction predictions were then stored in text files for later ingest into a graph database.

During 2020 there were around 23000 space objects in orbit for at least part of the year. That means there were over 250 million comparisons needed each day. Most of those were screened out with the apogee and perigee filter. Of the millions of remaining comparisons each day our simulations identified over 460000 conjunctions with a minimum distance within 10 km.

3. GRAPH DATABASE

A graph is a mathematical structure that connects vertices to each other with edges. Each vertex can have zero or more edges. Graph databases are software applications that use this mathematical structure for conceptual storage and analysis of data. In graph DBs, a vertex is often called a Node and an edge is referred to as a Relationship. A relationship connects two nodes together. In our implementation the relationships have no direction, other schemas may have directed relationships. Consider a simple schema where the nodes are of type Person. The relationships can have several different types such as Person 1 is the "Child Of" Person 2. The relationship "Child Of" is a directed relationship. A relationship between person nodes without a direction could be something like "Coworker."

Our schema defines a Node type called `space_object`. The relationship between two space objects is called a `conjunction`. Graph DBs allow each node and relationship to store data in the form of properties. Properties are similar to columns in a traditional RDBMS table. Many Graph DBs are implemented with a NoSQL backend for physical storage. This means that properties can be added to individual instances of nodes and relationships without having alter existing entries in the database. In a traditional RDMBS, adding a column affects all rows in a table. The set of basic properties for our schema is presented in Fig. 1.

We imported our conjunction predictions into an instance of the Neo4j graph database server. Neo4j uses the Cypher query language. With the above schema, to retrieve the list of objects that had a conjunction with object 25544 you would execute the following query:

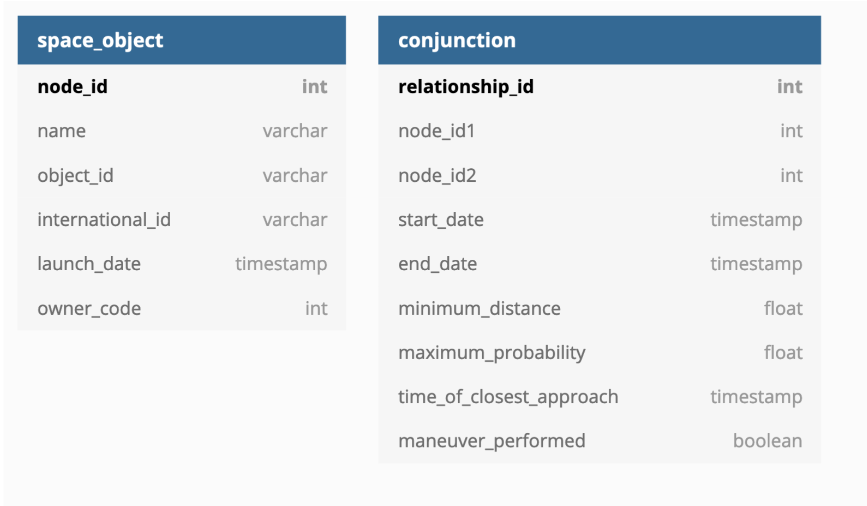

Fig. 1: A conceptual schema for conjunction information.

```
MATCH (a:space_object {object_id: '25544'})--(b:space_object)
RETURN b.name, b.object_id
```

To filter that query to only include objects that came within 1 km you can query on the relationship properties:

```
MATCH (a:space_object {object_id: '25544'})-[r:conjunction]-(b:space_object)
WHERE r.minimum_distance < 1
RETURN b.name, b.object_id, r.minimum_distance
```

Object 25544 is the ISS, which often maneuvers away from objects with predicted collision probabilities greater than 1 in 10000. The above query returns a list of 10 objects in 2020 that came within 1 km of ISS. Seven of those objects are cubesats that were launched from ISS, but three of them were unrelated rocket bodies.

Those objects may have passed close to the ISS infrequently enough to not be considered a threat. Consider though that each object might have participated in close approaches with other objects. The following query will look for neighbors of neighbors to the ISS:

```
MATCH (a:space_object {object_id: '25544'})-[r:conjunction*1..2]-(b:space_object)
WHERE r[1].minimum_distance < 1
RETURN b.name, b.object_id, r.minimum_distance
```

For 2020 this query produced around 150 objects. If the query is expanded to neighbors of neighbors of neighbors (n^3) then the number grows to over 8000. That is 8000 objects that are linked to the ISS by at least one conjunction within 1 km of another object. Even if most of these objects never come close to the ISS, they do have mutual close approaches. If one of these close approaches were to become a debris generating event (collision or breakup), then the threat of a small-scale cascade affecting the ISS grows. Consider a situation where two debris objects that are both one step removed from the ISS collide, how much advanced warning will there be before the subsequent debris collides with something that does come close to the ISS? It may be useful to screen for these types of collisions and take shelter when high probabilities are predicted in neighbors of neighbors. Graph DBs are optimized to perform queries such as the neighbors of neighbors and n^3. A typical RDBMS schema that attempts to implement a graph-like schema with foreign keys and join tables would have poor performance on those queries. The graph schema and the types of queries supported by graph DBs greatly simplifies obtaining these results.

Satellite operators can make use of other properties of the conjunction relationship such as `maneuver_performed` to compute cost or risk in the manner of their choice. Theoretical conjunction assessments can be performed for planned constellations. Operators can simulate up-coming mega constellations to estimate how often they will need to maneuver away from potentially threatening objects including threats posed by neighbors of neighbors. As more objects are added to the catalog either by new launches or the lowering of the RCS floor due to new contributing sensors such as the space fence, the graph DB approach can continue to scale to billions of nodes and relationships[1].

4. GLOBAL STATISTICS

The field of mathematics that makes a study of graphs and their properties is called Graph Theory. Algorithms developed by Graph Theorists are used to analyze graphs in diverse areas such as Social Network Analysis and fraud detection of credit card transactions. Several of these algorithms are used to compute statistics for each node that quantify the number and relative density of the relationships connecting each node. By computing these statistics for the entire graph we can potentially identify problematic space objects without having to start from any particular location in the graph of conjunctions.

The Katz centrality measures the number of relationships between a node and all the other nodes in the graph. It starts by first computing the number of neighbors and then counting their neighbors, etc. At each degree of relationship the count of neighboring nodes is weighted by an attenuation factor. Nodes with high Katz centrality have large numbers of relationships and neighbors with large numbers of relationships. The Meteor 2-2 Deb object has the highest Katz centrality in the graph. Objects with high centrality have a higher number of potential conjunctions and therefore may be good candidates for removal. An object with high Katz centrality is shown in Fig. 2.

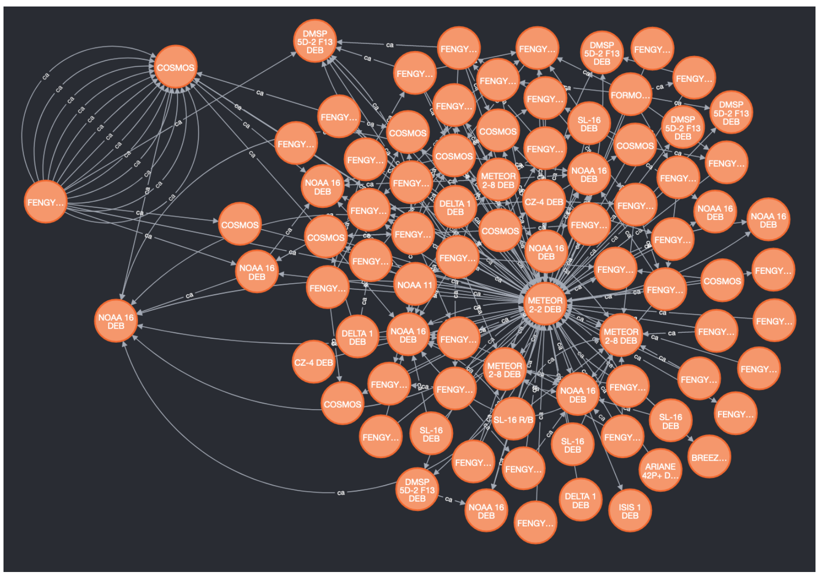

Fig. 2: An object with high Katz Centrality and its potential conjunctions.

Coreness is a measure of the degree of connections in sub-graphs. A high coreness implies being near the center of relatively isolated sub-graph. An object with high coreness is shown in Fig. 3. Objects like this could be targets for removal to improve the safety of their sub-graphs, especially if active payloads are members of their sub-graphs.

[1] `https://neo4j.com/product/neo4j-graph-database/scalability/`

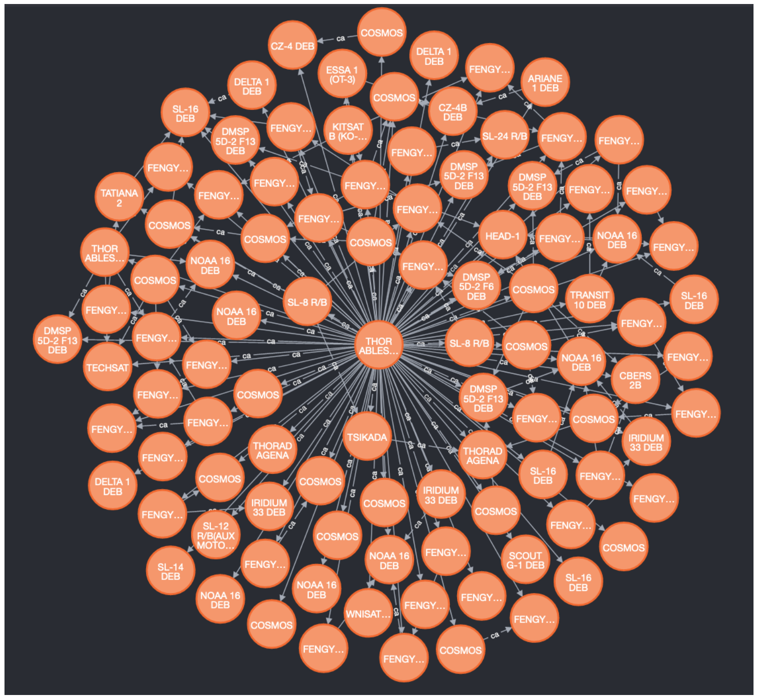

Fig. 3: An object with high Coreness and its potential conjunctions.

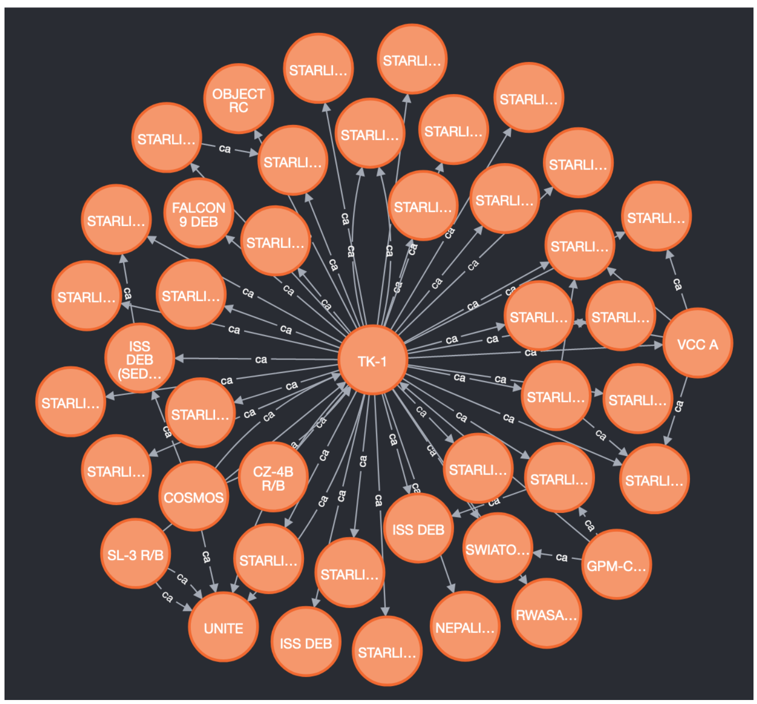

Fig. 4: An object with high Page Rank and its potential conjunctions.

The PageRank algorithm was very important in the early days of Google. It is commonly used to measure the importance of nodes in a graph by measuring the number and quality of links in the graph. The highest PageRank object is shown in Fig. 4.

Compared to coreness and Katz centrality, the high PageRank seems relatively unimportant in terms of identifying objects that pose risks to many other objects directly. However, consider the result when the neighbors of neighbors are plotted as in Fig. 5.

Fig. 5: An object with high Page Rank and potential conjunctions to degree 2.

When viewed with second degree connections, the importance of an object with high Page Rank seems to be as a connection point between relatively separated sub-graphs. Targeting these objects for removal may help prevent cascading debris events.

5. CONCLUSION

Conjunction Assessments are an important tool for flight safety. Extending the analysis of conjunctions to include neighbors of neighbors and beyond is an important way to identify potential threatening objects to the whole space environment. Graph databases are a flexible and scalable tool for storing historical conjunction information. Analyzing the graph of conjunctions using global statistics aids in identifying objects that may be targets for debris removal missions. Future work may include computing the change in value of graph statistics after specific objects are removed. Identifying the objects with the biggest improvement in graph statistics after their removal may be a better way of selecting debris removal targets.

6. REFERENCES

[1] T. S. Kelso and S. Alfano. Satellite orbital conjunction reports assessing threatening encounters in space (SOCRATES). In *Modeling, Simulation, and Verification of Space-based Systems III*, volume 6221, pages 1 – 9. International Society for Optics and Photonics, SPIE, 2006.

[2] Darren McKnight, Rachel Witner, Francesca Letizia, et al. Identifying the 50 statistically-most-concerning derelict objects in LEO. *Acta Astronautica*, 181:282–291, 2021.

[3] State-of-the-art small spacecraft technology. Technical report, NASA, October 2020.

[4] R. Frigm and L. Johnson. Total Probability of Collision as a Metric for Finite Conjunction Assessment and Collision Risk Management. In *Advanced Maui Optical and Space Surveillance Technologies Conference*, page 14, January 2015.