THE AEROSPACE CORPORATION

Choosing Ground Stations with Gusto

multi-objective optimization for ground station selection

Dr. Leah Ruckle
Karel Marshall

The Aerospace Corporation
Modeling & Simulation Department

23 February 2022

OTR: 2022-00410

Outline

- Background & Motivation
- Basics of Genetic Algorithms
- Gusto Implementation
- Detailed Fictional Case Study Analysis Example
- Wrap Up

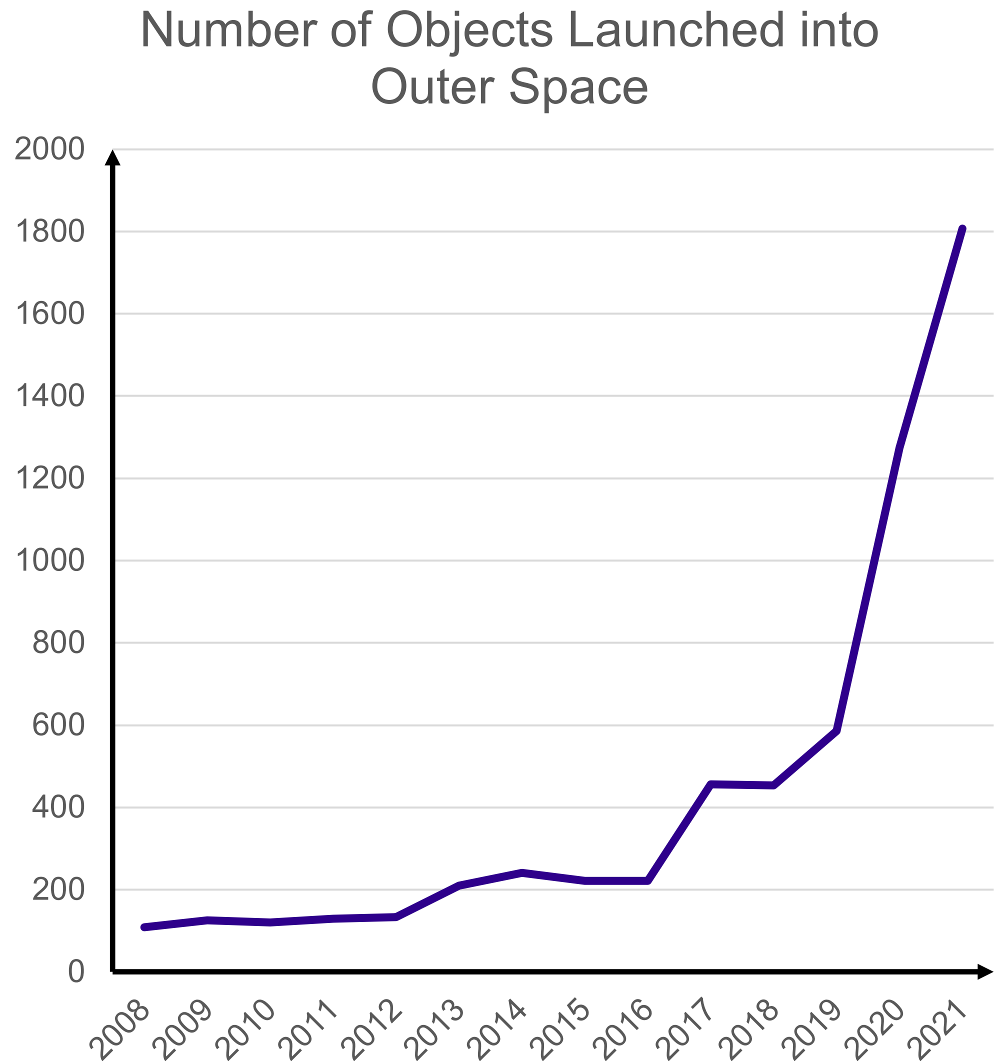
Background/Motivation

History of the Commercial Augmentation Services (CAS) Efforts

- Historically, government agencies have been the dominant force in US space activities. But as technical solutions arise in the commercial world, the US military is looking to buy solutions from private entities

- Due to the exponential increase of the number of active satellites over the past few years, the DoD is looking to supplement the Space Force's Satellite Control Network (SCN) with commercial ground vendors

- The CAS effort brings increased agility and resiliency to the existing ground station architecture by:
  - *Expanding network diversity*
  - *Introducing communications diversity*
  - *Establishing a backup system*



Data Source: UNITED NATIONS Office for Outer Space Affairs

APPROVED FOR PUBLIC RELEASE

Introducing Gusto for Ground Station Architecture Identification

THE MOTIVATION

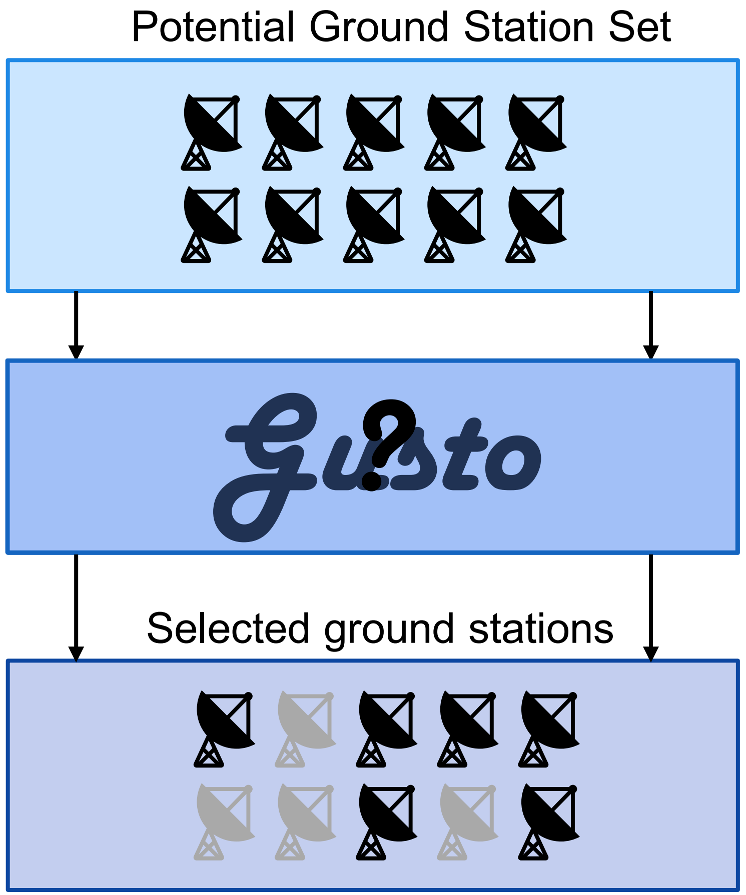
The recent increase in satellites in space has placed increased pressure on ground systems, but there is an opportunity to use the growing number of commercial ground service providers.

THE NEED

A methodology for choosing which of the available ground stations will best work together to satisfy the needs of a given set of satellites.

OUR APPROACH: *Gusto*

- Gusto is a ground station selection tool that leverages a multi-objective genetic algorithm to identify high-quality sets of ground stations to support satellite contact needs
- Can be used to identify "blue sky" ground station sets or augmentation sets to support an existing ground architecture



Basics of Genetic Algorithms

- Genetic algorithms (GA) are a set of iterative problem-solving operations, used to find a set of high value solutions to an optimization problem
- Made popular by scientist John Holland, GA tend to converge to optimal values by applying Darwin's theory of natural selection in which unfit options are eliminated from the solution pool[1]

The Process of a Genetic Algorithm

1. GA start with a "population" of randomly-generated candidate solutions

2. The properties of each "individual" of the population are evaluated for fitness against the objective function

3. At the end of each evaluation round, new individuals are generated from the highest performing individuals

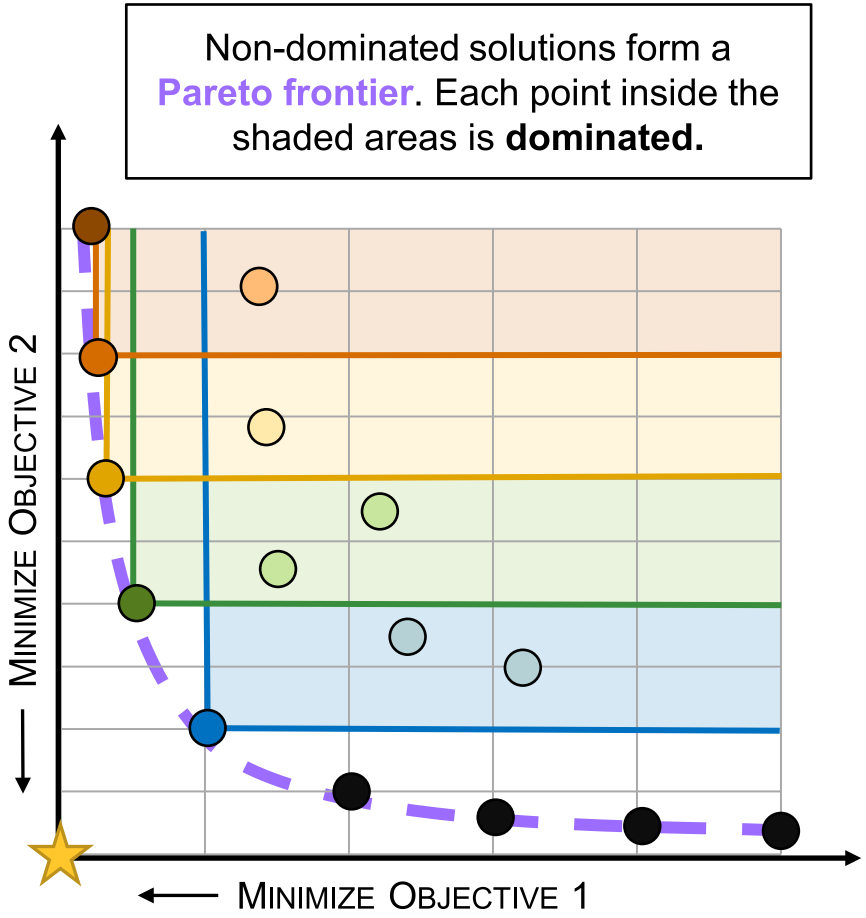
4. The lowest-performing individuals are removed to make room for the new individuals – keeping the population count the same

5. The algorithm runs until either a sufficient fitness level has been obtained, or until the max number of rounds has been achieved

[1] J. Holland. "Genetic Algorithms and Adaptation". Adaptive Control of Ill-Defined Systems, New York, 1984: 317-333.

Multi-Objective Optimization

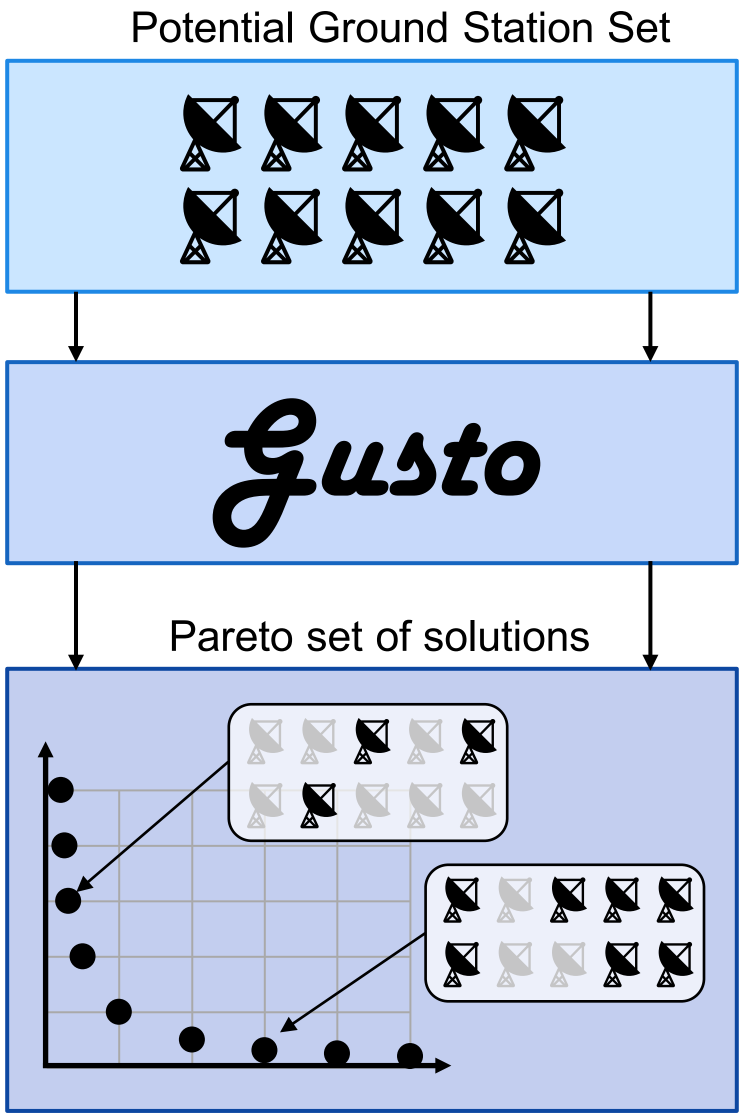
- Many real-world problems involve more than one objective
  - *Tradeoffs between different aspects of the problem*
  - *e.g., cost vs. performance*
- Ignoring the multi-objective nature or assigning *a priori* the weights to the objectives may lead to poor final solutions
- In multi-objective optimization, all objectives are optimized *simultaneously,* and the outcome is a *Pareto frontier*
- Pareto frontier: the set of non-dominated solutions
  - *Non-dominated solution: improving performance in one objective sacrifices performance in at least one other objective*
  - *Most multi-objective optimization algorithms cannot guarantee Pareto optimality, but produce a Pareto-efficient set of high-quality solutions*
- Many multi-objective optimization algorithms have been developed to find or approximate Pareto frontiers[1,2]
  - *NSGA-II – a popular genetic algorithm developed by Deb et al.[3] that includes crowd distancing*



[1] R.T. Marler and J.S. Arora. "Survey of multi-objective optimization methods for engineering". Structural and multidisciplinary optimization 26.6 (2004): 369-395.
[2] A. Konak, D.W. Coit, and A. Smith. "Multi-objective optimization using genetic algorithms: a tutorial". Reliability engineering & system safety 91.9 (2006): 992-1007.
[3] K. Deb, et al. "A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II," in IEEE Transactions on Evolutionary Computation, vol. 6, no. 2, pp. 182–197, April 2002.

Gusto: Ground Stations Selection Optimization

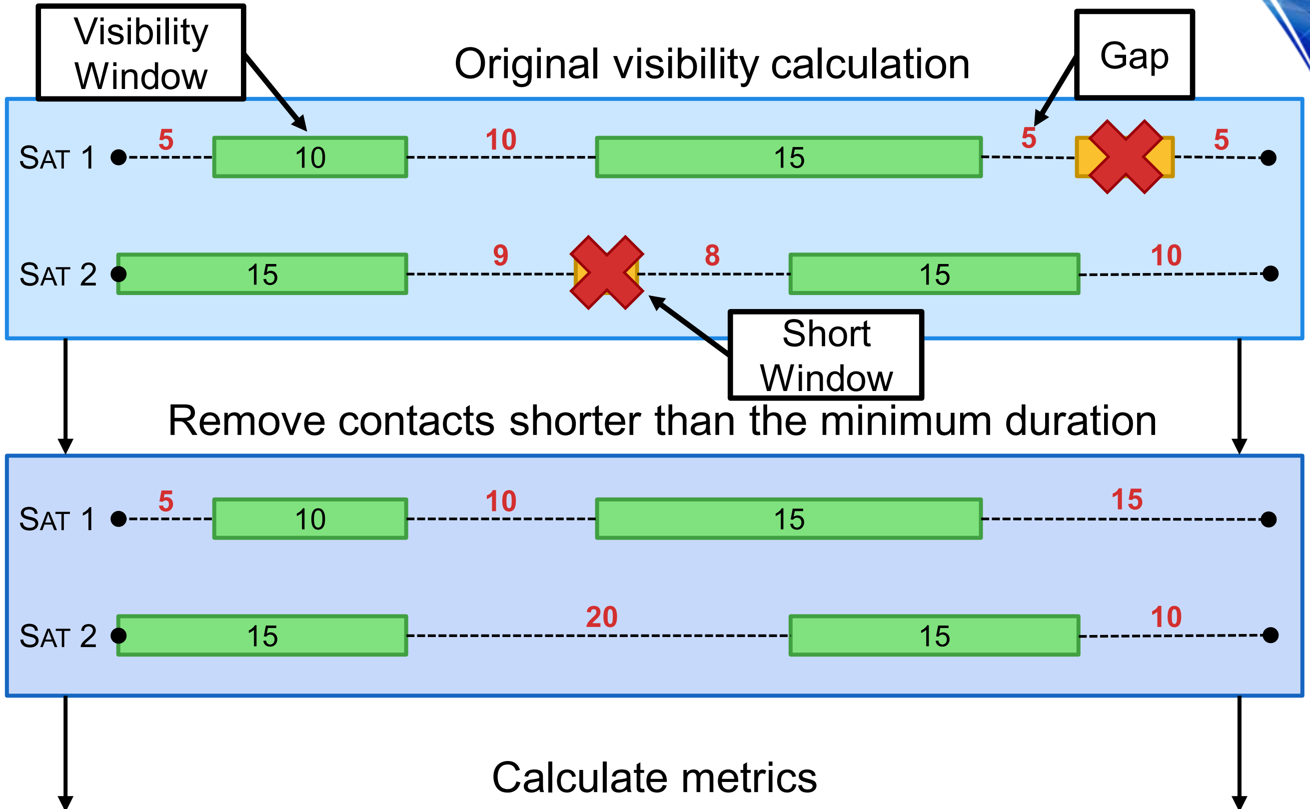
- Since multiple considerations and tradeoffs come into play when selecting a set of ground stations, Gusto uses multi-objective optimization to identify a *set* of potentially desirable solutions
- Gusto is written in Python 3.8, and uses the NSGA-II implementation from the pymoo[1] package
- Each ground station is modeled as a binary decision variable, which Gusto sets to add/exclude the site from the ground architecture
  - *Can be run with a set of real ground station locations or "imaginary" locations to identify good areas for new sites*
- Current objectives are primarily visibility-based:
  - *Minimize the number of ground stations*
  - *Minimize the average and maximum gaps in ground station access for non-geostationary satellites*
  - *Maximize the average and minimum number of simultaneously seen ground stations for geostationary satellites*



[1] J. Blank and K. Deb, pymoo: Multi-Objective Optimization in Python, in IEEE Access, vol. 8, pp. 89497-89509, 2020, doi: 10.1109/ACCESS.2020.2990567

Performance Metrics Terminology & Calculation: Gap

- Gap: time period in which the satellite cannot be supported by any ground station
- Gap-based metrics are useful for assessing support to non-geostationary satellites
- Calculation is performed assuming that any compatible ground station with sufficient visibility is acceptable
  - *Visibility is constrained by each satellite's minimum elevation angle requirement*
- Visibility periods shorter than the minimum contact duration are filtered out
- Average and maximum gap statistics are taken over the entire set of gaps across all satellites



$$\text{Avg Gap} = \frac{5+10+15+20+10}{5}$$

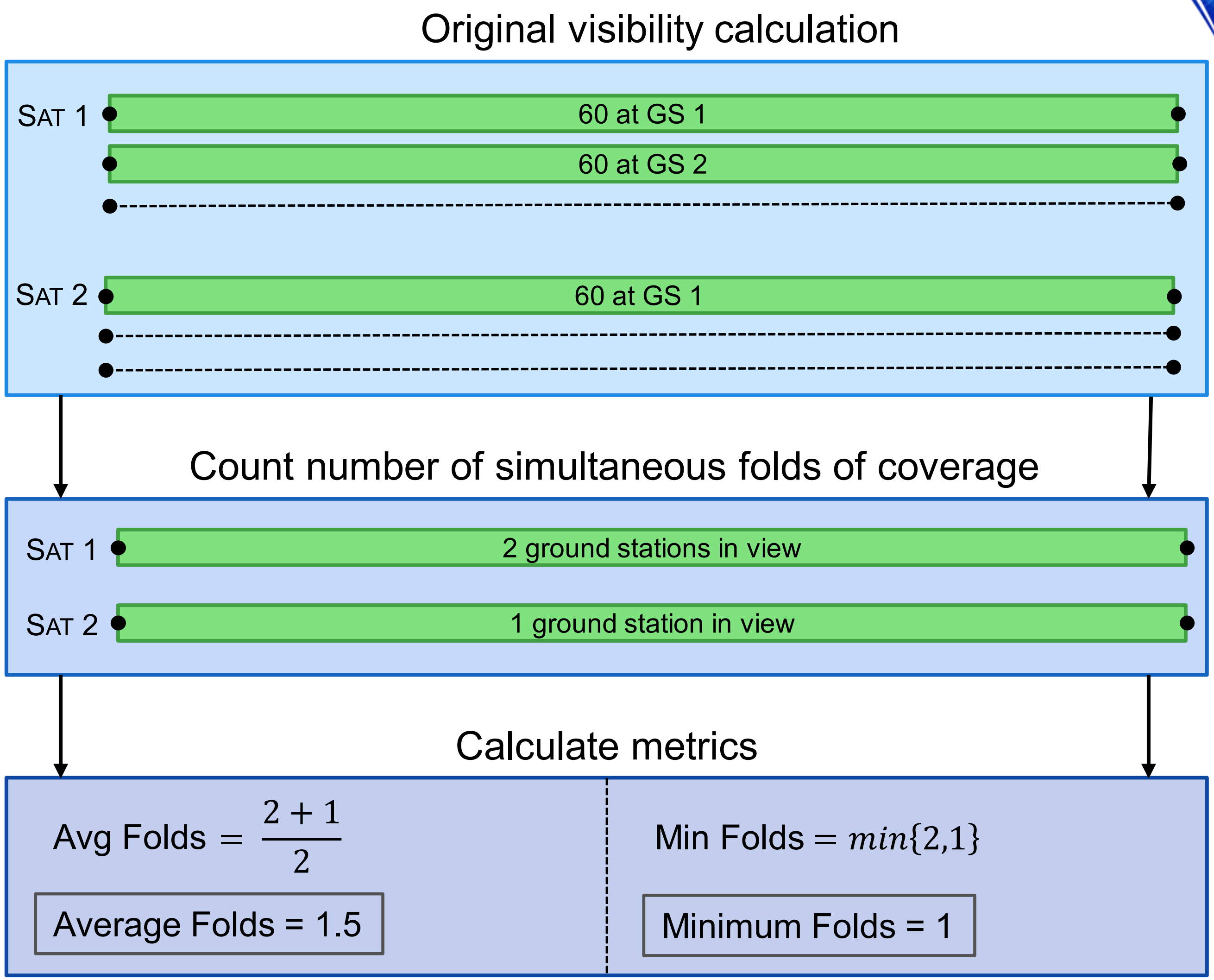
Average Gap = 12

$$\text{Max Gap} = max\{5, 10, 15, 20, 10\}$$

Maximum Gap = 20

Performance Metrics Terminology & Calculation: Folds of Coverage

- Folds of coverage: number of ground stations that could simultaneously support a satellite
- Folds of coverage metric is useful for assessing support to geostationary satellites since they will not experience periodic gaps
- In this visibility calculation, we track which ground stations can see which satellites…
- …Then count the number of visible ground stations for each satellite
- Average and minimum folds statistics are taken across all satellites
  - *If the satellites do not have constant folds over the simulation time, the average is time-weighted*



Example Analysis

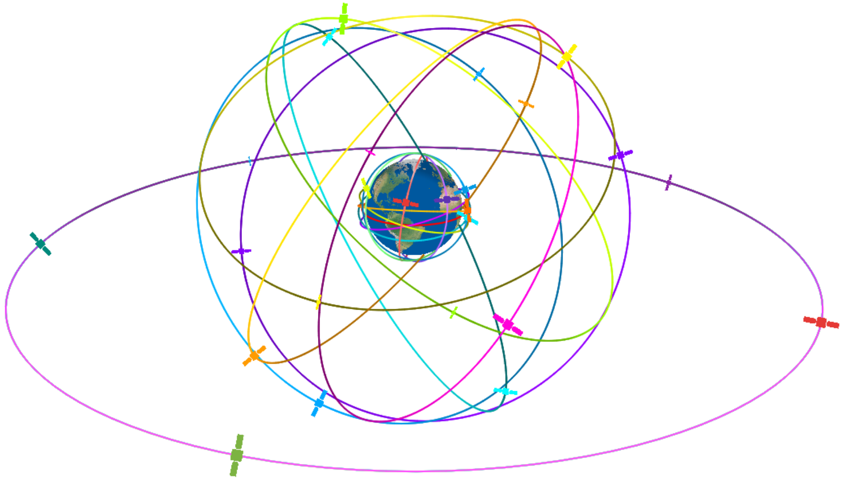
The fictional ACME Space Corporation needs a set of ground stations to support their space architecture

APPROVED FOR PUBLIC RELEASE

Problem Introduction

- ACME Space Corporation has big plans for their upcoming space architecture – almost thirty satellites spread across LEO, MEO and GEO
- ACME wants to buy time on commercial ground antennas to support these satellites rather than build and maintain their own
  - *Considering support from several providers rather than just a single-provider partnership*
  - *ACME does not currently have any ground stations so this will be a "blue sky" analysis as opposed to an augmentation analysis*
- Contact requirements:

Metric	Satellites	Required	Goal
Max Gap	Non-GEO	120 minutes	60 minutes
Avg Gap	Non-GEO	90 minutes	30 minutes
Min Folds	GEO	1 fold	2 folds
Avg Folds	GEO	1.5 fold	3 folds



ACME Space Corporation's Space Architecture

<u>SUN-SYNCHRONOUS ORBIT</u>

- 4 satellites
- *LTANs*: 0000, 0300, 0600, 0900
- 800 km altitude
- 10-minute min contact duration

<u>LOW EARTH ORBIT</u>

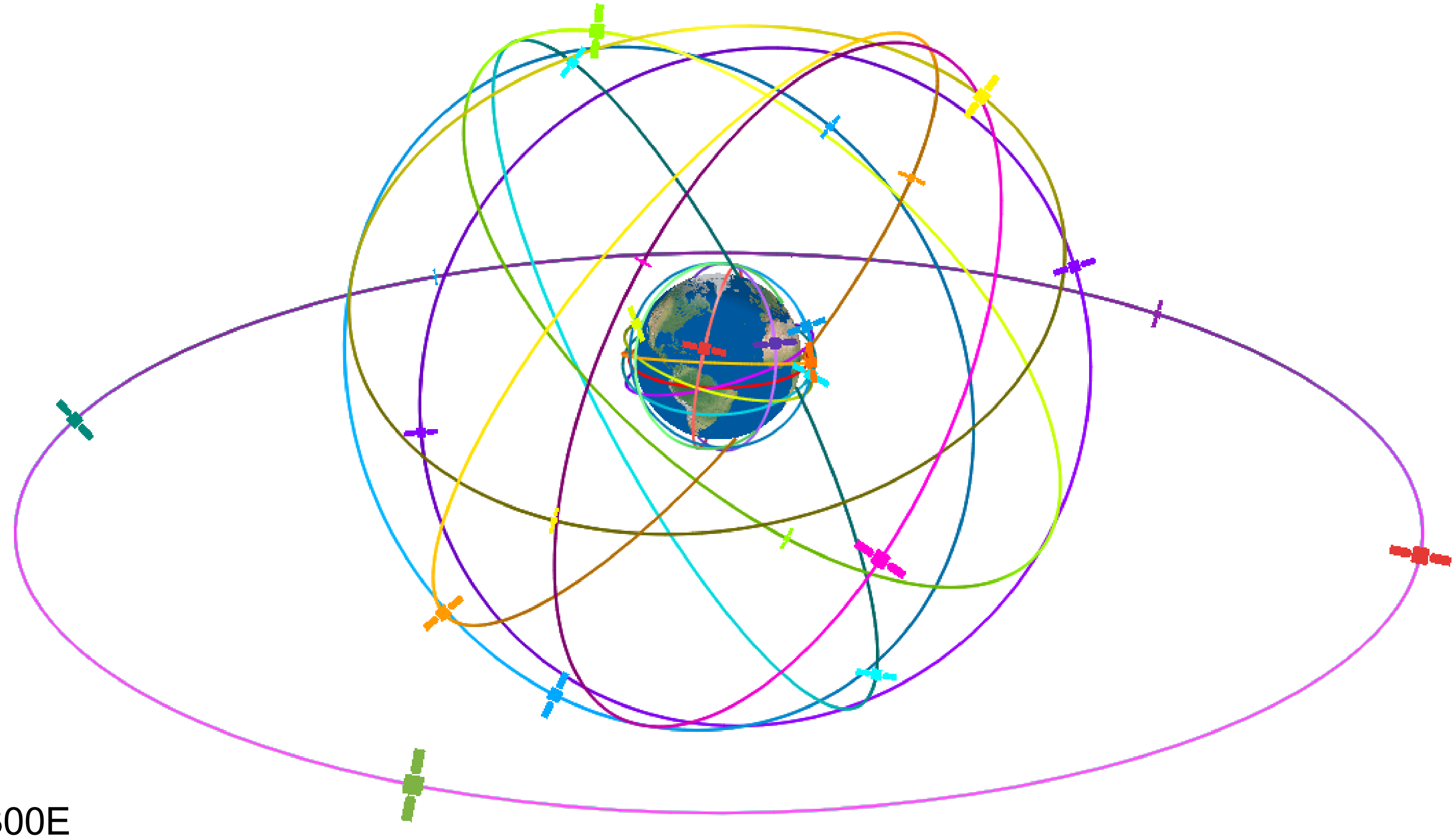
- 4 satellites in Walker 15°:4/4/1
- 1 equatorial satellite (no inclination)
- 1000 km altitude
- 10-minute min contact duration

<u>MEDIUM EARTH ORBIT</u>

- 14 satellites in Walker 60°:14/7/1
- 20,000 km altitude
- 20-minute min contact duration

<u>GEOSTATIONARY ORBIT</u>

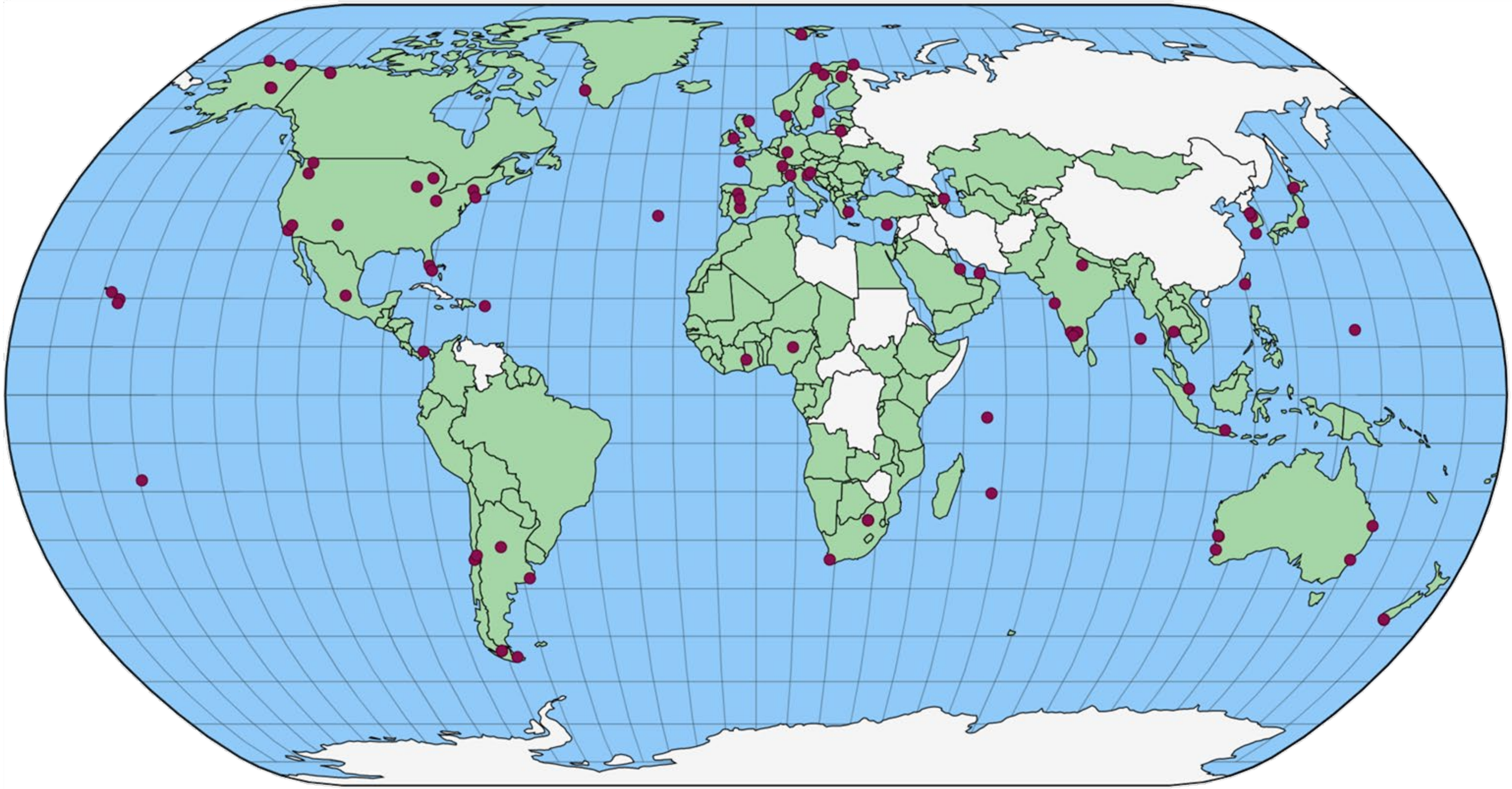
- 6 satellites
- Equally-spaced around GEO belt
- *Slots*: 0 E, 60 E, 120E, 180E, 240E, 300E



*Orbits in all regimes are circular

Set of Commercial Ground Stations Under Consideration

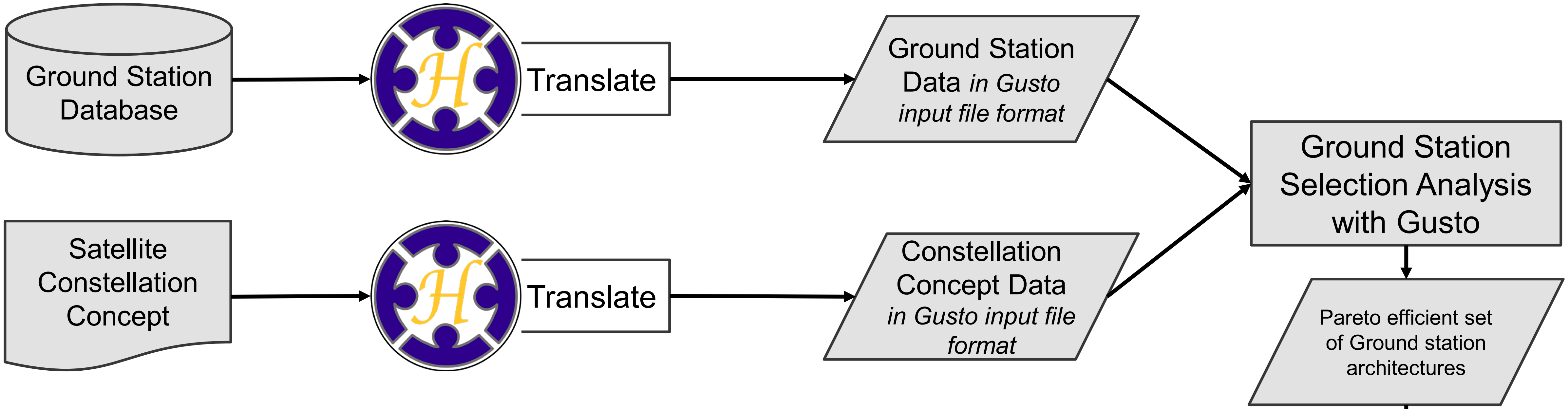
92 ground station options from 5 different commercial providers

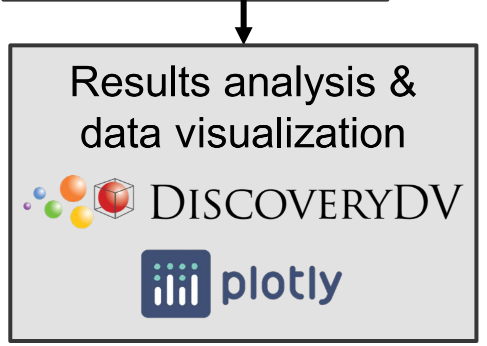


Gusto identifies high-performing subsets of these 92 ground stations

APPROVED FOR PUBLIC RELEASE

Digital Engineering-Enabled Analysis Workflow





Huddle[1] is a data translation service at The Aerospace Corporation that facilitates the translation of data from various sources into the necessary format for analysis tools

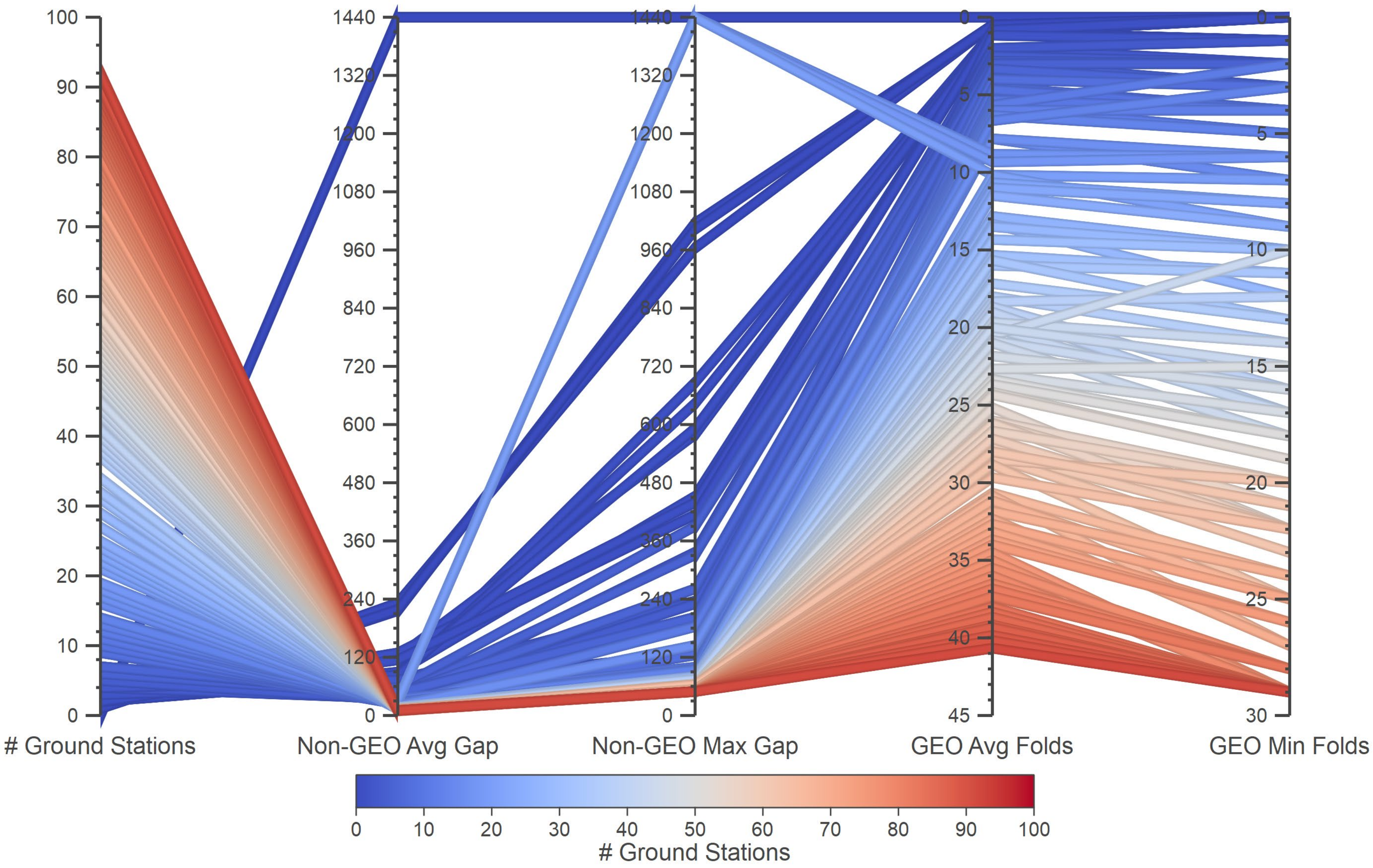
DiscoveryDV[2] is an interactive multi-dimensional data visualization and analysis tool at The Aerospace Corporation that supports the communication of complex analyses to decision makers

Plotly Dash[3] is an open-source Python package for building browser-based data visualization dashboards and plots

[1] A. Rice, C. Iwata, and L. Ruckle. "Authoritative Source of Truth (ASOT) to Analysis Using Huddle". Ground System Architectures Workshop. The Aerospace Corporation. 2021.
[2] Contact Joshua Kollat (joshua.b.kollat@aero.org) at The Aerospace Corporation for more details.
[3] Plotly Technologies Inc. Collaborative data science. Montréal, QC, 2015. https://plot.ly.

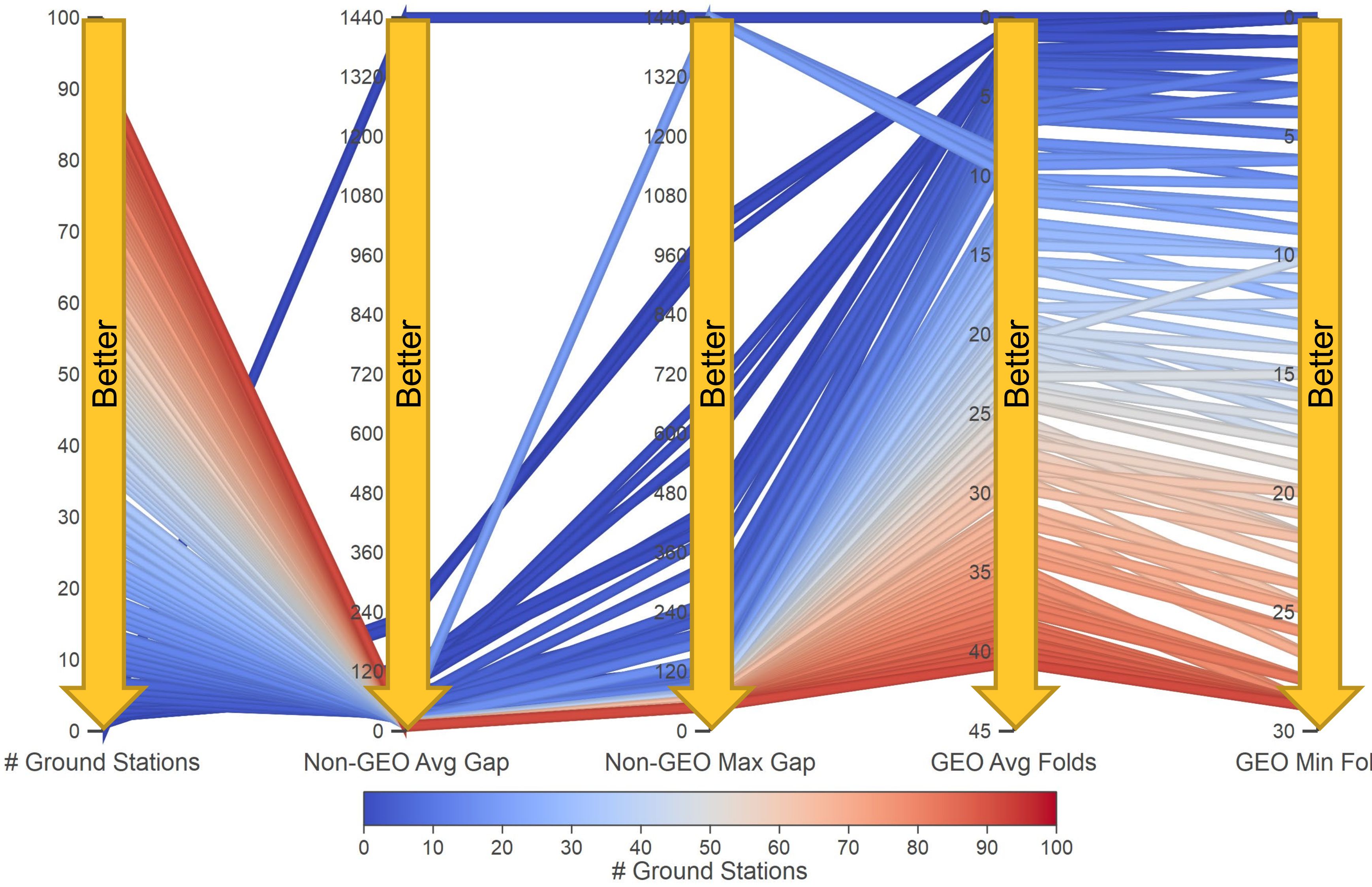
Using a Parallel Coordinate Plot to Visualize Analysis Results

- Parallel coordinate plots are an effective visualization for viewing the Pareto frontier across multiple dimensions
- All the axes are shown parallel to each other
- A single solution is represented by a line
- In this plot, each axis is one of the optimization objectives



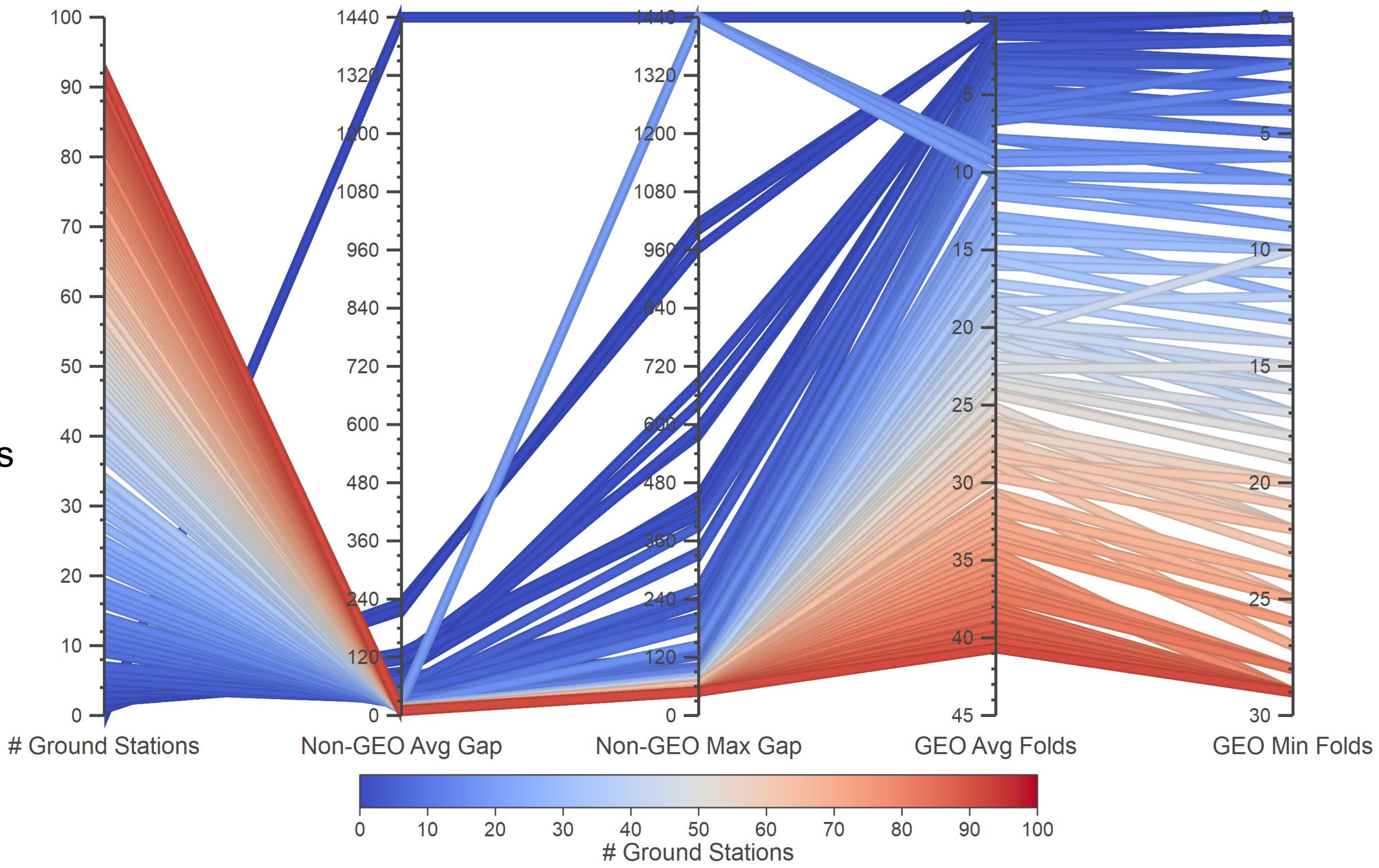
Using a Parallel Coordinate Plot to Visualize Analysis Results

- Parallel coordinate plots are an effective visualization for viewing the Pareto frontier across multiple dimensions
- All the axes are shown parallel to each other
- A single solution is represented by a line
- In this plot, each axis is one of the optimization objectives
- The axes are oriented such that better performance is towards the bottom



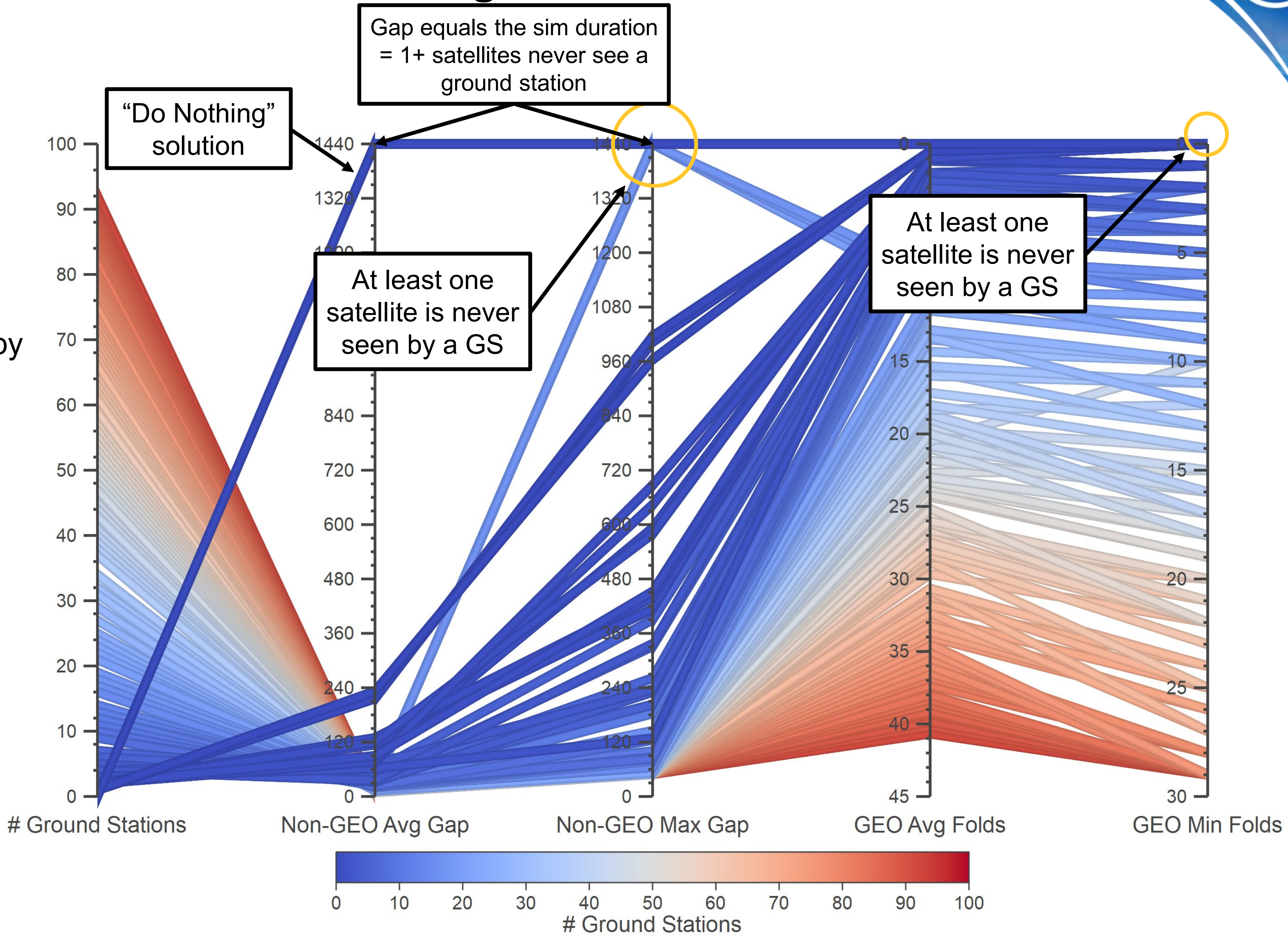
Ground Station Count vs. Performance Tradeoff

- Each solution (line) is colored by its number of ground stations
- Solutions with higher numbers of ground stations have higher performance
  - *Red lines are low on performance axes*
- Shows the tradeoff between adding more ground stations and getting more performance



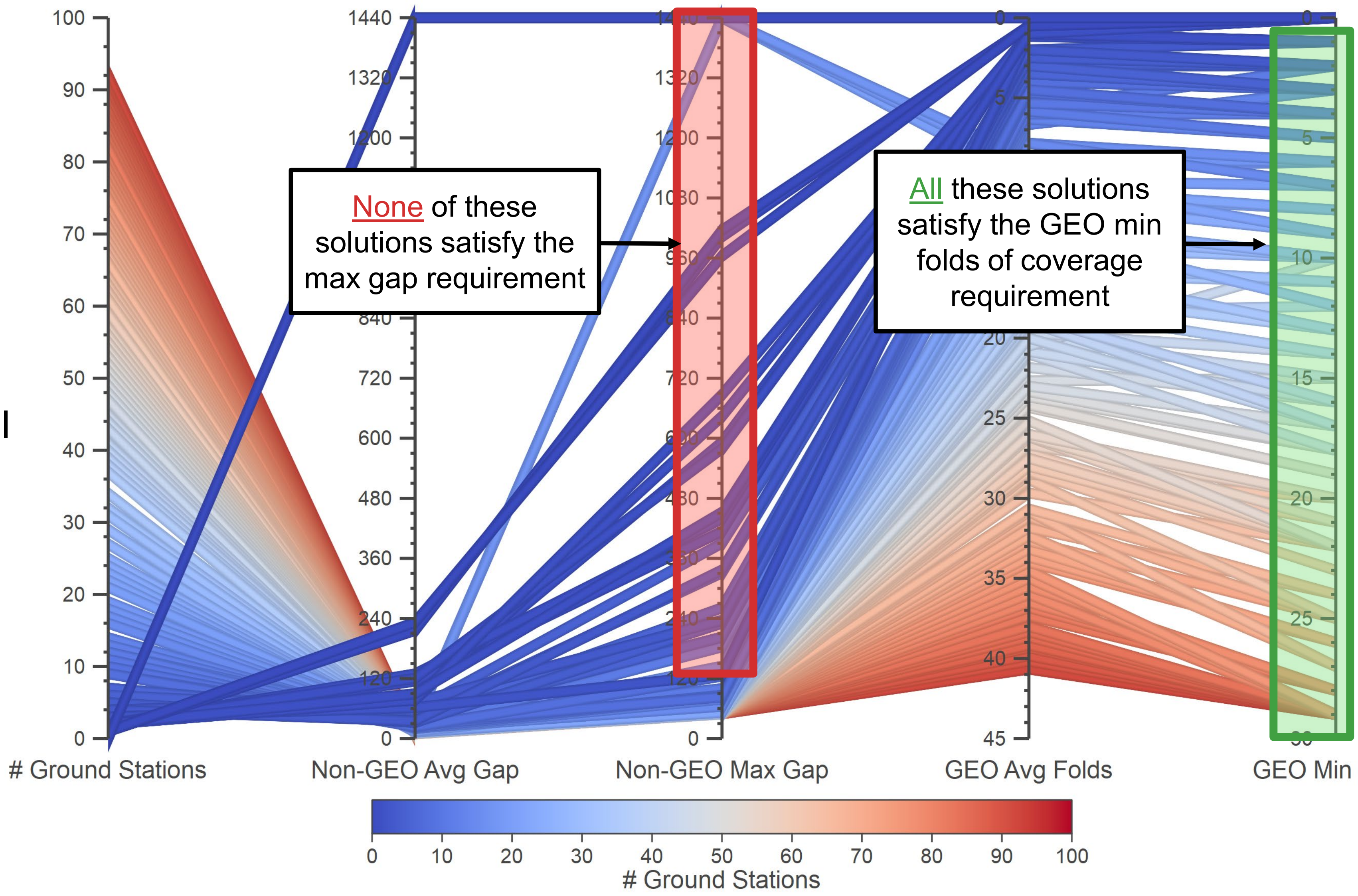
Identification of Outliers & Poor-Performing Solutions

- Better is towards the bottom of the plot therefore poor-performing solutions are near the top
- When a gap equals the simulation duration, it indicates that the satellite is *never seen* by a ground station
- Outlier: solution in which the average gap equals the simulation duration
  - *"Do nothing solution"*
- Poor performers: solutions in which the max gap equals the simulation duration…
- …or min folds equals zero amongst the GEO satellites
  - *Indicates that at least one satellite is never seen by a ground station*



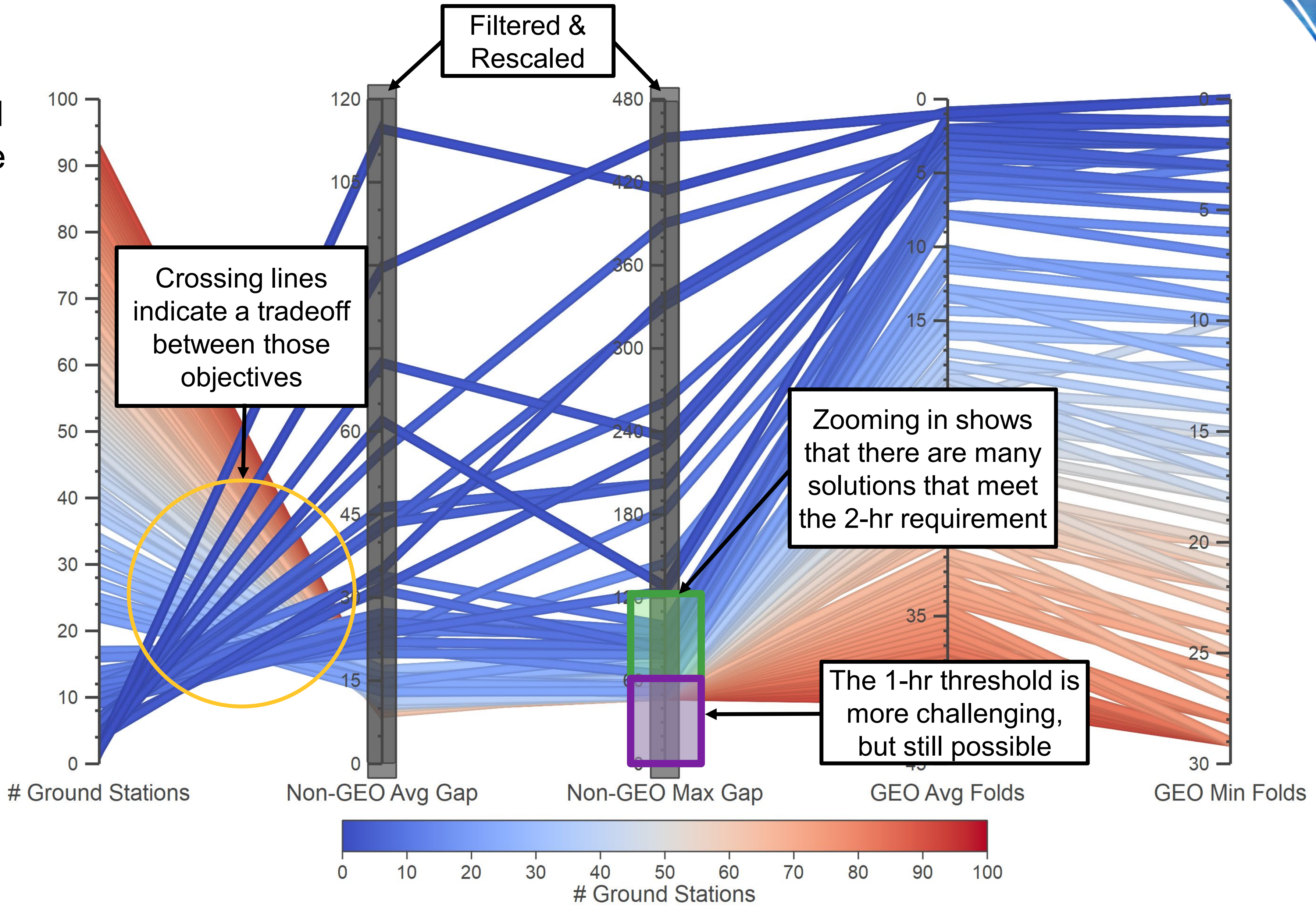
Maximum Gap Requirement is Most Constraining

- Most solutions satisfy the minimum folds of coverage threshold for the GEO satellites (1-fold)
- Fewer solutions satisfy the max gap threshold for the non-GEO satellites (2-hr)
- Maximum gap requirement is more constraining and will drive our down-selection of solutions



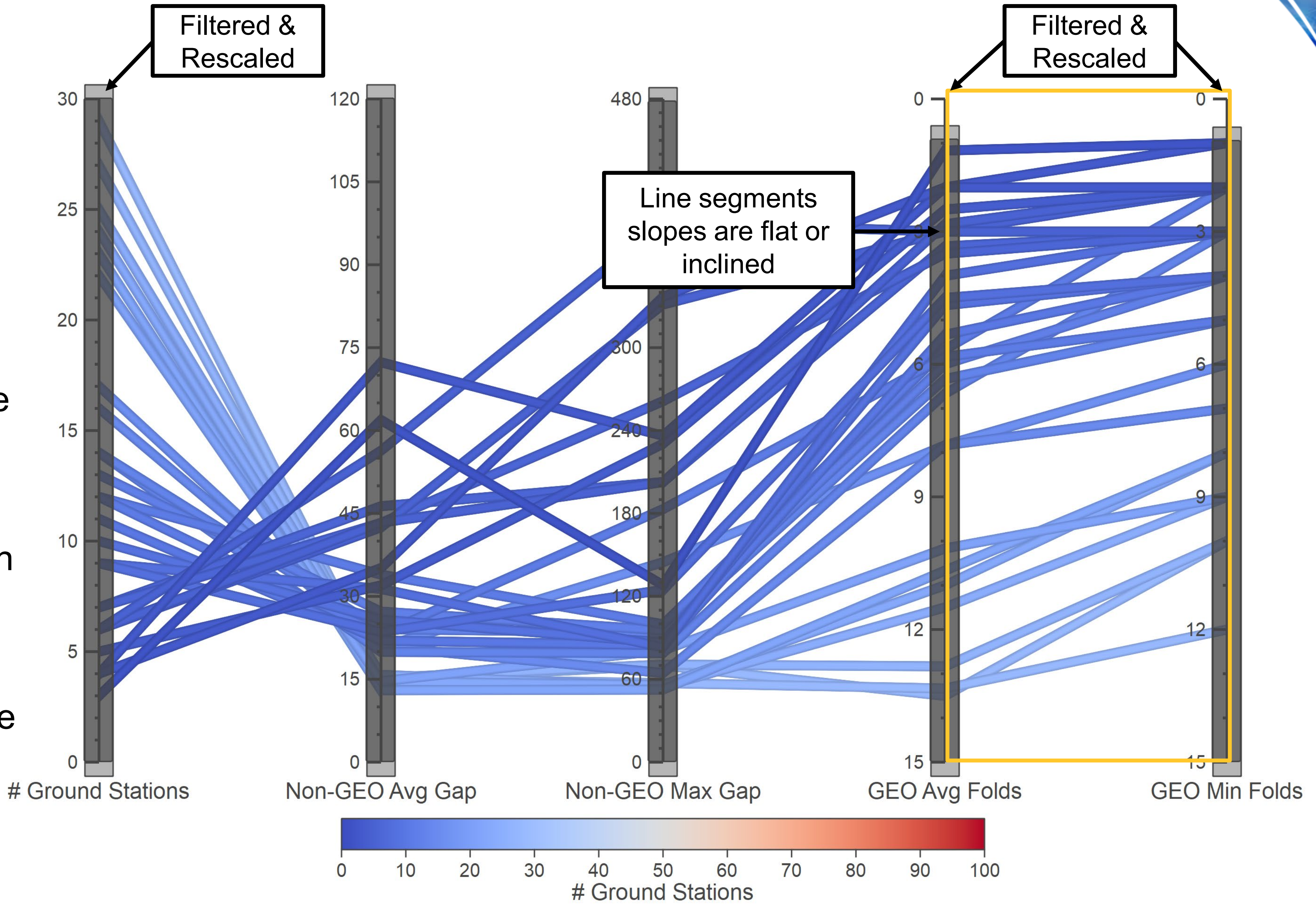
Zooming in on Gap Axes Shows Solutions That Meet Requirements

- Outliers and poor-performers can be excluded by filtering and rescaling the average and max gap axes
- There are solutions that meet at least the 2-hr threshold
- The 1-hr threshold is significantly more challenging, but possible
- Tradeoffs between objectives manifest as crossing lines between those respective axes



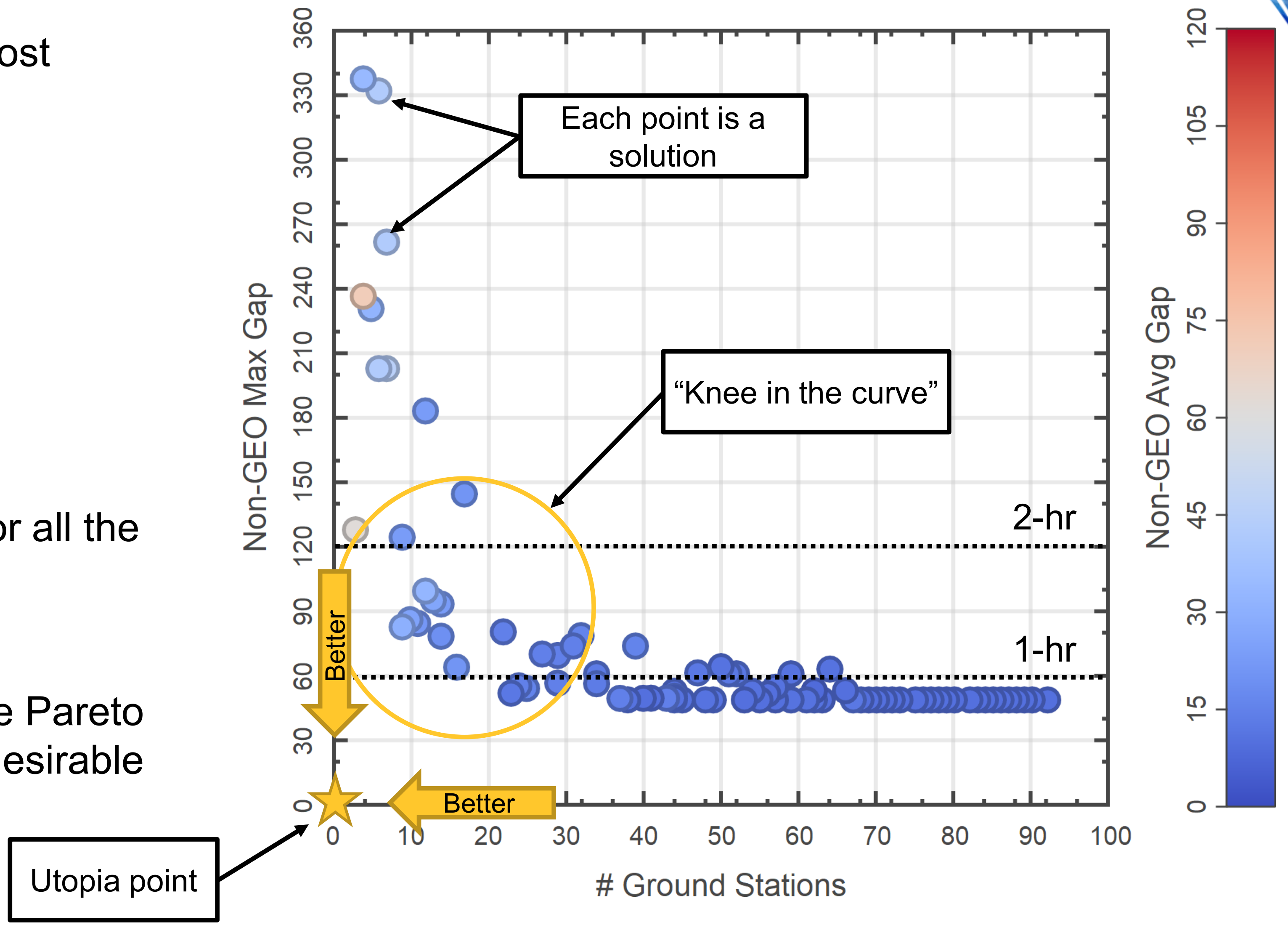
Focusing on GEO Folds of Coverage Metrics

- The relationship between the GEO folds of coverage axes can be better seen when the axes are on the same range
- Line segment slopes between the average GEO folds and min GEO folds are always flat or inclined…
- …Confirms that the minimum is always less than or equal to the average
- (Nearly) parallel lines indicate objectives that scale together



Focus on Non-GEO Maximum Gap vs. Ground Station Count Tradeoff

- The maximum gap requirement is the most challenging to satisfy
  - *Maximum Gap Requirement: 2 hours*
  - *Maximum Gap Goal: 1 hour*
- Applied filters:
  - *Non-GEO Max Gap < 360 minutes*
  - *Non-GEO Avg Gap < 120 minutes*
  - *GEO Min Folds of Coverage > 1 site*
- <u>Utopia point</u>: point with the ideal value for all the objectives (often unobtainable)
  - *Marked with the gold star*
- <u>“Knee in the curve”</u>: the region where the Pareto frontier bends, representing potentially desirable compromise opportunities

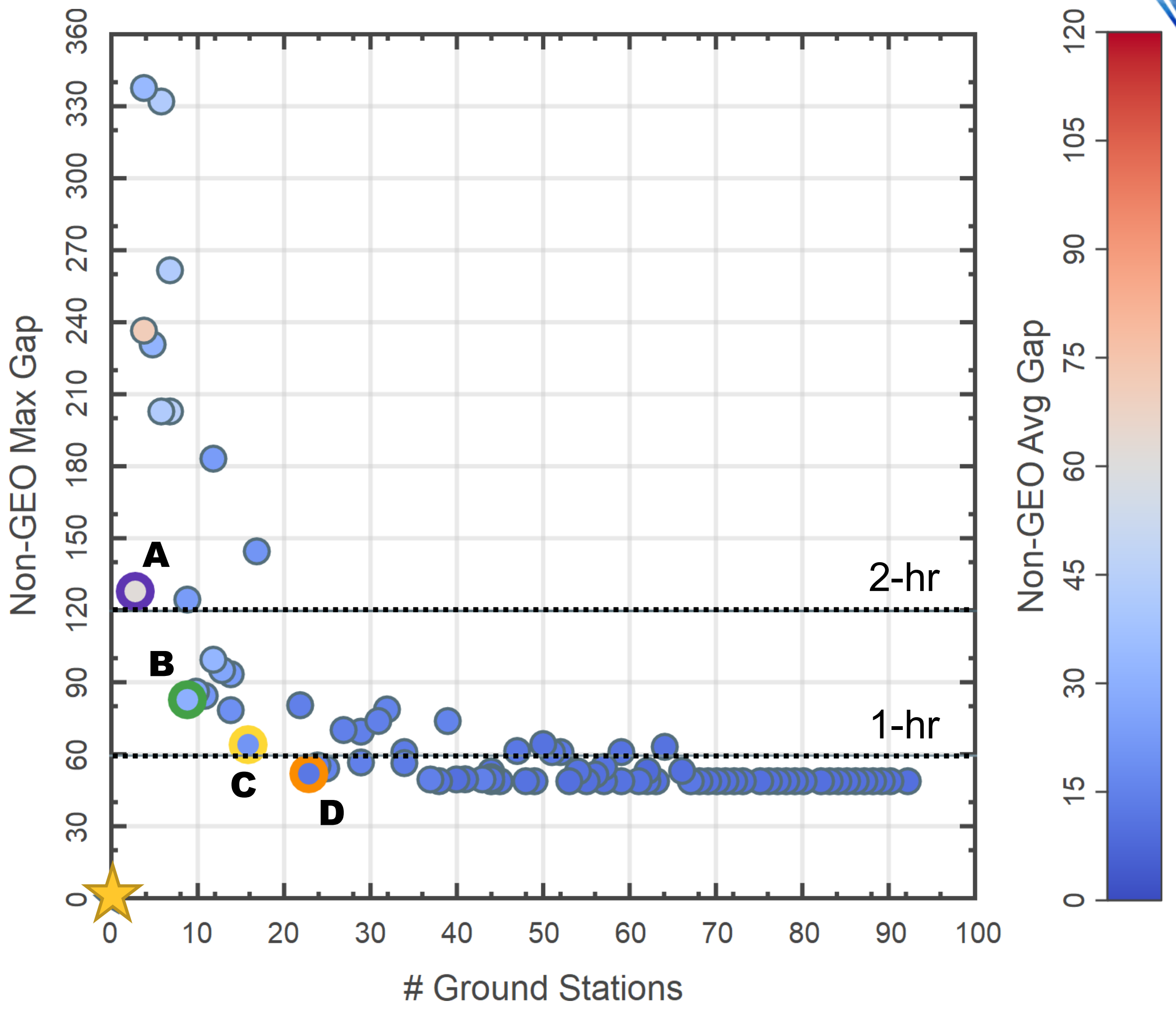


Selecting Four Points of Interest

- Let's select four points of interest along the Pareto frontier within the "knee in the curve" region

	# Ground Stations	Non-GEO Max Gap	Non-GEO Avg Gap	GEO Min Folds	GEO Avg Folds
Soln A	3	128.1	61.9	1	1.167
Soln B	9	83.17	31.33	3	4
Soln C	16	64.55	20.91	7	7.833
Soln D	23	52.5	13.02	8	10.67

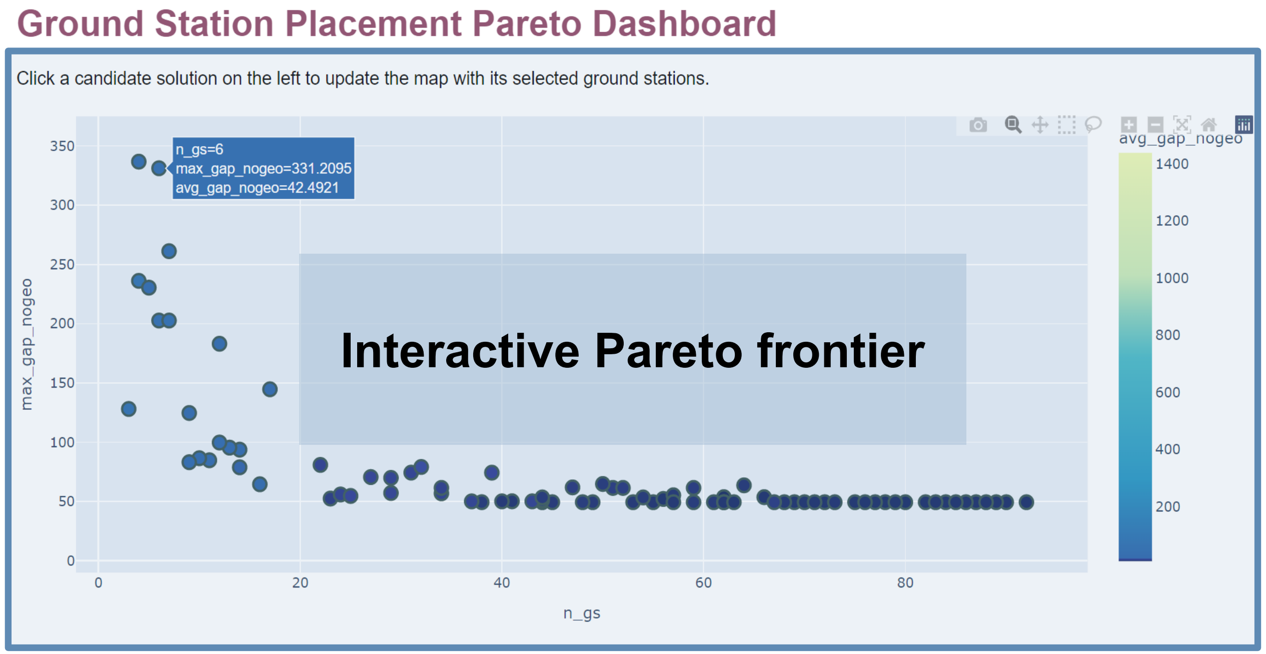
- *Solution A* doesn't strictly satisfy the 2-hr maximum gap requirement, but it only uses 3 ground stations
- The next smallest option, *Solution B*, uses 9 ground stations and easily satisfies the 2-hr limit
- *Solution C* very nearly satisfies the 1-hr threshold, at the cost of an additional 7 ground stations
- The smallest solution to satisfy the 1-hr threshold, *Solution D*, uses 23 ground stations

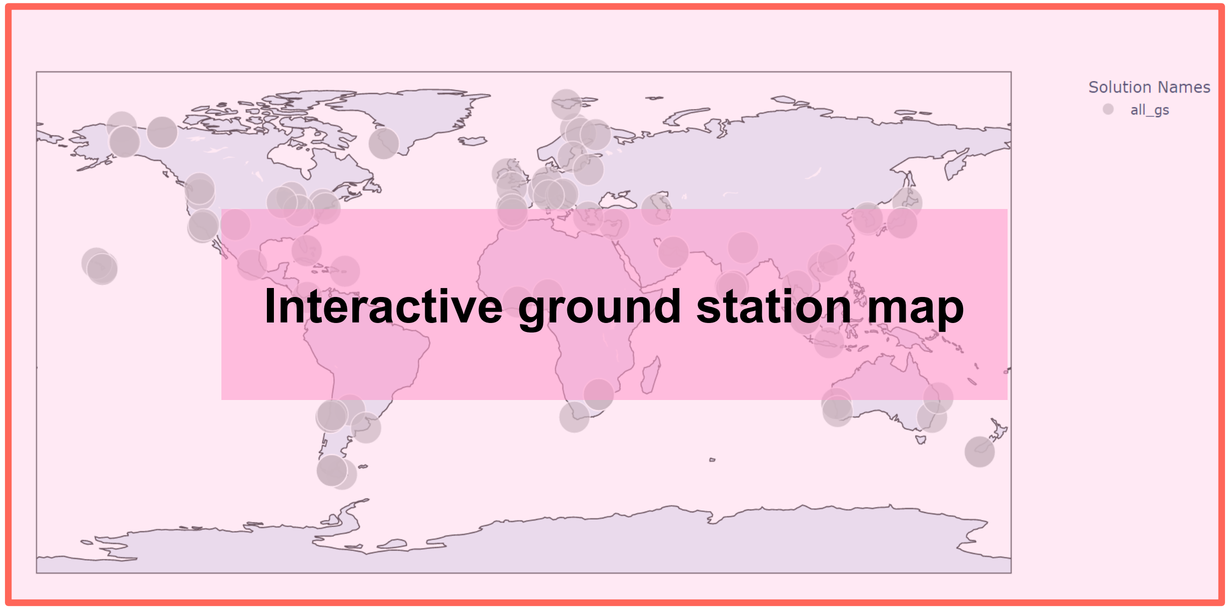


Visualizing Selected Results

Dashboard Showing Candidate Solutions

- The ***Ground Station Pareto Dashboard*** allows analysts to interactively study the Pareto frontier
- Developed using the Plotly Dash[1] Python framework, the dashboard is composed of:
  - *A clickable scatterplot showing the Pareto frontier against chosen metrics*
  - *A world plot that updates with the selected ground stations for a Pareto-efficient solution*
- Allows users to obtain more information by clicking and/or hovering over points on either plot





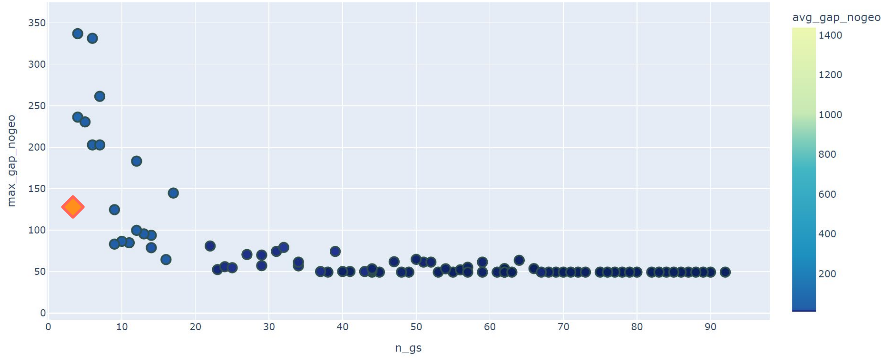
[1] S. Hossain, "Visualization of Bioinformatics Data with Dash Bio," in Proceedings of the 18th Python in Science Conference (SciPy 2019), pp. 126–133, 2019

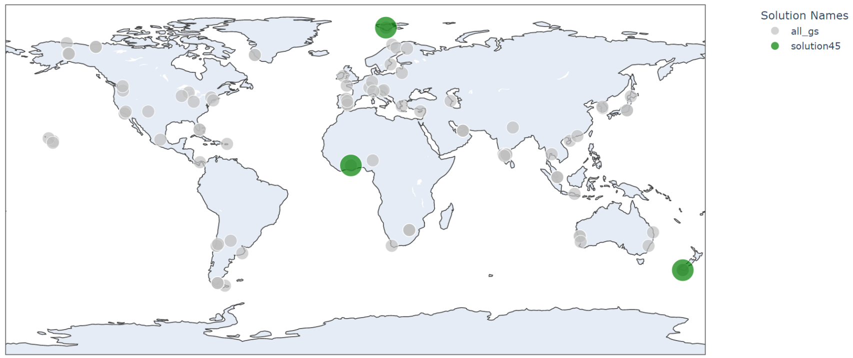
APPROVED FOR PUBLIC RELEASE

Maps of Some Select Results

Dashboard Showing Candidate Solutions

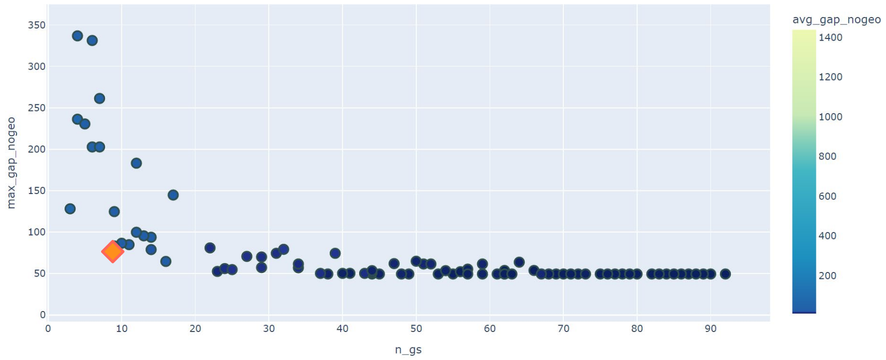
Solution A

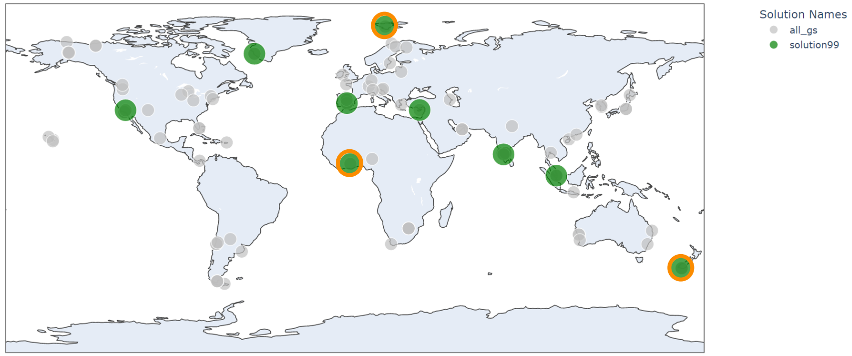




Although there are only three sites in Solution A, they are well-spaced in latitude and longitude.

Solution B



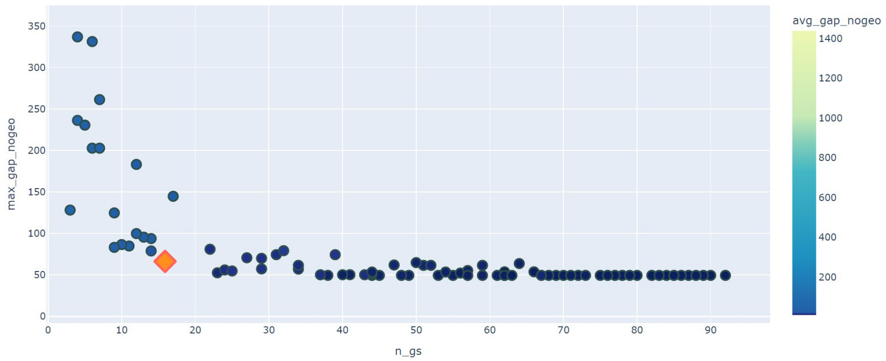


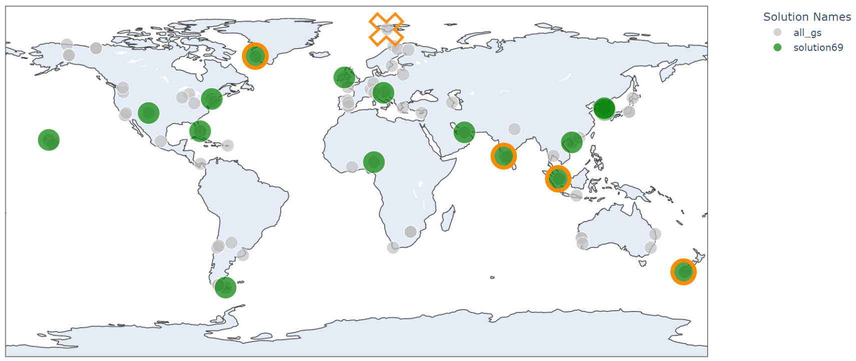
All three sites from Solution A are chosen in Solution B. Additional sites are well-spread in latitude and longitude.

Maps of Some Select Results

Dashboard Showing Candidate Solutions

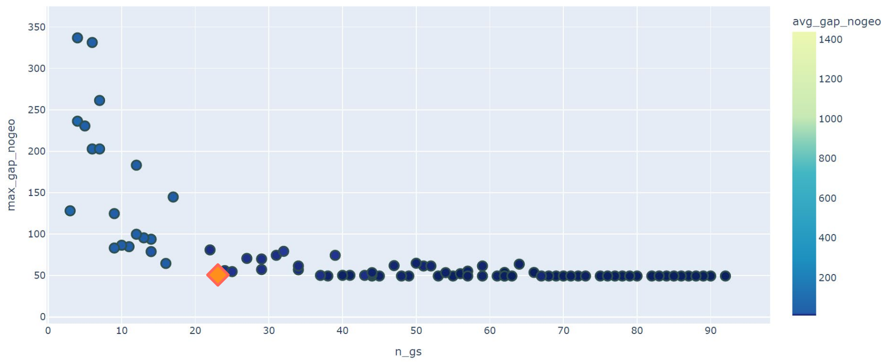
Solution C

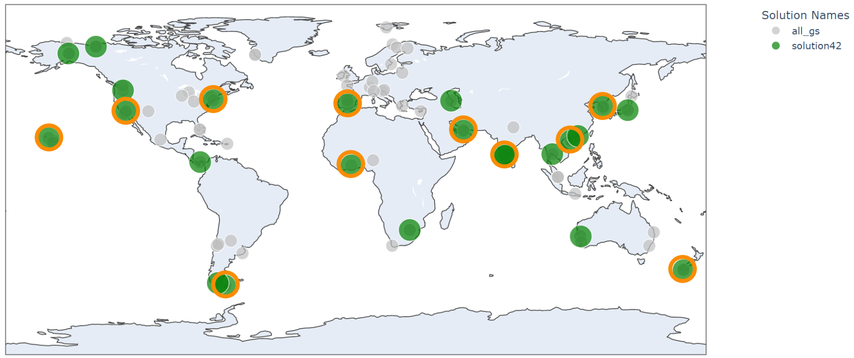




Although Solution C loses the high-latitude site from Solutions A & B, nearly half of the sites in Solution B appear in Solution C.

Solution D



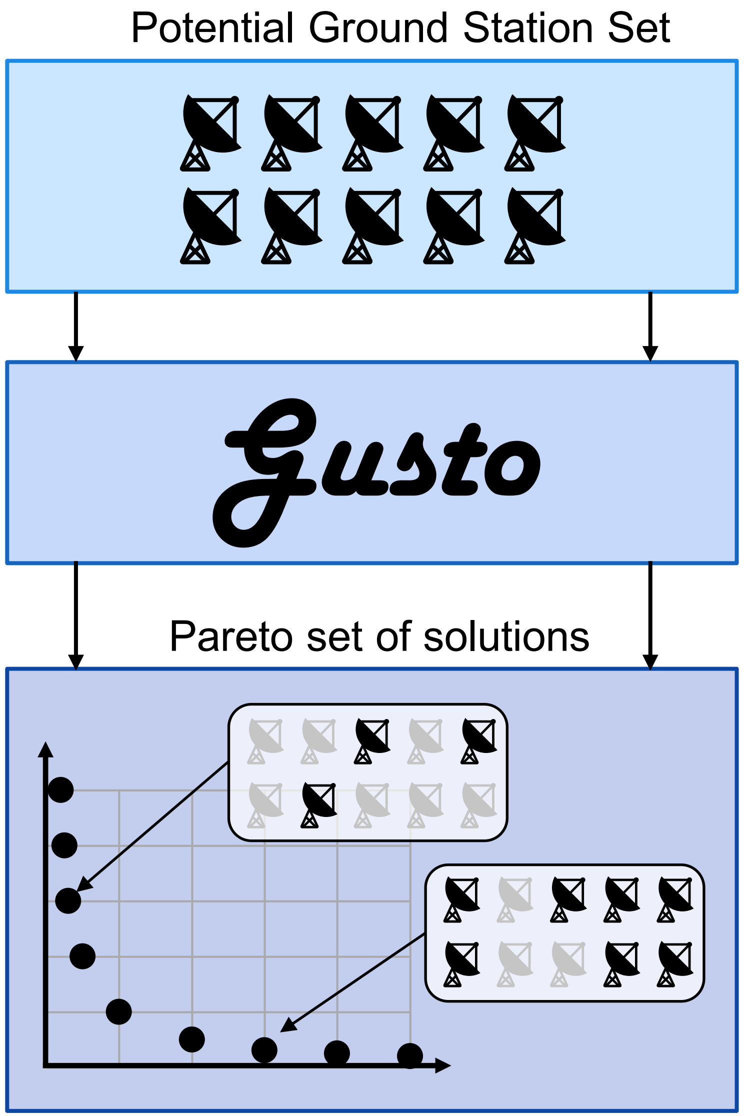


Approximately ~66% of sites in Solutions A & B, and ~50% of sites in Solution C appear in Solution D. Changes primarily occur in the higher latitudes of the northern hemisphere.

Summary

Summary of Gusto

- Gusto is a ground station selection tool that leverages a multi-objective genetic algorithm to identify high-quality sets of ground stations to support satellite contact needs
- Can be used to identify “blue sky” ground station sets or augmentation sets to support an existing ground architecture
- Current objectives are primarily visibility-based:
  - *Minimize the number of ground stations*
  - *Minimize the average and maximum gaps in ground station access for non-geostationary satellites*
  - *Maximize the average and minimum folds of coverage for the geostationary satellites*
- Future objectives and constraints could include other considerations:
  - *Cost to use, modify, or build a ground station*
  - *More-detailed satellite-ground station compatibility*



Thank You

APPROVED FOR PUBLIC RELEASE