Generation-based Evolutionary Tool for the Optimization of Constellations (GenETOC)

Joshua Carden, Analytical Mechanics Associates Inc., 21 Enterprise Pkwy Suite 300, Hampton, VA 23666, (757)-864-7778, joshua.f.carden@nasa.gov
Shaun Deacon, 1 N Dryden St, Hampton, VA 23681, (757)-864-2467, shaun.a.deacon@nasa.gov
Paul Kessler, 1 N Dryden St, Hampton, VA 23681, (757)-864-3019, paul.d.kessler@nasa.gov
Paul Speth, 1 N Dryden St, Hampton, VA 23681, (757)-968-9952, paul.w.speth@nasa.gov

Abstract

With the rapid growth in the capabilities of smaller satellites, satellite architectures that replace a single, extremely capable spacecraft with multiple, cheaper ones are gaining in popularity. Unfortunately, the orbit design process for constellations can be significantly more involved, especially when the relative placement of the individual spacecraft within the constellation is not constrained by mission and/or science objectives. Optimizing a satellite constellation in the presence of multiple, competing objectives is a highly complex problem to which many traditional mathematical optimization methods cannot be applied and few tools exist to help mission designers search for promising candidate mission designs. The Generation-based Evolutionary Tool for the Optimization of Constellations (GenETOC) has been created to search for near-optimal constellation design options. GenETOC combines a modified version of the Non-dominated Sorting Genetic Algorithm II (NSGA II) with STK Components libraries (a 3rd party .NET package created by Analytical Graphics Inc.) to create a framework that enables a mission designer to generate a simulation that models the design problem and obtain a family of potential, near-optimal solutions that can be investigated more in detail.

GenETOC was developed in C# using the .NET framework with Windows Presentation Foundation (WPF) serving as the framework from which to create the graphical user interface (GUI). GenETOC user inputs can be categorized into three major data components: definition of the problem (areas of interest, satellite decision parameters, and sensor configurations), definition of performance objectives, and specification of the genetic algorithm (GA) parameters. In the problem definition component, the user is prompted to define the areas of interest against which the performance metrics will be computed, define the sensor parameters and attach them to specific spacecraft, select which satellite orbital parameters will be added to the decision space of the GA, and specify the range of desired values for each optimization parameter. For performance objectives, the user is presented with a list of available coverage and revisit performance based calculation options from which two metrics are chosen to serve as the objective functions that the GA will use to evaluate solutions during the optimization process. Finally, the definition of the GA parameters provides user control over the number of generations (number of optimization iterations), the population size (number of candidate constellations created in each generation), and the adaptive mutation and crossover threshold values (control parameters for how frequently each process occurs during the optimization).

GenETOC has been extensively tested to verify the individual components of the optimization process. The GA has been tested against a suite of GA test problems to confirm convergence to the known two and three-dimensional Pareto fronts. The coverage and revisit performance metrics obtained in GenETOC are compared with STK desktop scenarios, confirming the constellations are being appropriately modeled within GenETOC simulations. A walkthrough of a simple, example problem is provided to illustrate the workings of GenETOC and to demonstrate the output available to the mission designer.

Keywords – orbit constellations, coverage optimization, revisit optimization, genetic algorithm

Table of Contents

Introduction

Satellite Market Conditions

In the past several years the satellite market has seen a significant uptick as the cost of access to space has come down considerably due to the implementation of rideshare options and the ability of commercial launch service providers to offer satellite launches at a fraction of the historical cost. While this market evolution has generated a greater degree of corporate interests in space, it has also created a shift towards larger constellations of smaller, less capable systems. SpaceX's Starlink constellation is an example of this trend, where the mission design is no longer focused on optimizing a single orbit but rather on determining the optimal configuration that will minimize the number of satellites to achieve the same objective. This change in the philosophy has led to a similar shift for the mission designer, and typical methods such as parametric assessments are no longer able to adequately address the new optimization problem.

Satellite Constellation Design

Modern design problems with multiple satellites functioning as a constellation are frequently judged using metrics of success that are often conflicting in nature and thus cannot be directly optimized without introducing a weighting scheme to combine the various metrics of interest into a single optimizable function. Unfortunately, the tools available to mission designers have not kept pace with this growing complexity in the design space and few options exist to assist mission designers with the task of finding initially feasible design points from which higher fidelity analysis can be constructed.

To address this lack of available design tools and provide a flexible capability to aid mission designers in early concept formulation, the Space Mission Analysis Branch at NASA Langley has developed the Generation-based Evolutionary

Tool for the Optimization of Constellations (GenETOC). GenETOC combines a mission simulation engine with a Genetic Algorithm (GA) optimizer to create a framework that enables a mission designer to obtain a family of potential, near-optimal design solutions that can be investigated individually in more detail. Within GenETOC, designers are able to create the structure of mission simulation (simulation duration, areas of interest, and coverage definitions), specify which orbital parameters are allowed to vary within the optimization (and the permissible ranges of the variation), and select the optimization metrics against which the population of solutions will be evaluated.

Genetic Algorithms

Genetic Algorithms (GAs) are an attempt to mathematically represent aspects of Darwinian Evolution and have been utilized for a wide variety of applications since their original inception by Holland in 1975 [3]. GAs represent a population based search of an objective space, and as such do not suffer from many of the limitations of deterministic optimization schemes. When designed correctly, Genetic Algorithms do not require apriori knowledge of the problem, and this trait allows GAs to be used favorably in optimization problems with conflicting objectives.

In generational GAs, the algorithm establishes a base population and evolves that population generation by generation. During each generation, models that mimic natural selection, survival of the fittest, and mutation are employed to create offspring and establish the base population for the next successive generation. Natural selection and survival of the fittest push the population towards better and better solutions, while mutation ensures population diversity and allows for exploration of other areas of the trade space.

System Components

Overview

The GenETOC application can be broken down into three main subcomponents: the simulation engine, the optimizer, and the user interface. The simulation engine builds the constellation scenario within the program. GenETOC utilizes a third party library (STK Components) to model the constellation scenarios and obtain coverage data. Objective metrics are then calculated using the data produced by the constellation scenario.

GenETOC utilizes a slightly modified version of the Nondominated Sorting Genetic Algorithm II (NSGA II) algorithm [2] as the optimizer of the program. The optimizer loops through the process of creating satellite constellation scenarios, evaluating them based on user-defined criteria, and producing new constellations that drive the solution set towards improvement in the objective metrics. The GenETOC user interface is designed to give users an environment to build the mission scenario, set controls on the optimization and runtime processing, specify the output, and monitor the progress of the optimization.

A more detailed description of the major components of GenETOC is provided in the following sections.

Simulation Engine

The core function of the simulation engine is to calculate the coverage metrics of a constellation. To accomplish this, the simulation engine creates a full simulation of the mission scenario, which includes the physical components of the mission (satellite, sensor, and target definitions), the mission timeline, and the coverage objectives for the mission. With the full simulation of the mission scenario completed, the simulation engine can then perform the calculations for the objective metrics that GenETOC is optimizing towards. The simulation engine is divided into three key processes: simulation set up and scenario building, scenario propagation, and objective metric computation.

To build the constellation scenario, GenETOC leverages the libraries contained in STK Components software package. Using STK Components as a foundation, programmatic constructs to represent all potential components of a mission scenario were built. These constructs include satellites that are positioned using Keplerian orbital elements, sensors with defined fields of view, and ground target areas. The scenario is then given a time period for which the satellites should be propagated.

The propagation options within GenETOC are limited to the semi-analytic J2 and J4 propagators. While the simulation engine is capable of leveraging numerical propagation and full gravity and force models, GenETOC is intended to be a trade space exploration tool and the refinement of solutions using higher fidelity modeling is left to follow up analysis by the mission designer. This simplification reduces the required propagation time and assists with the goal of keeping the overall GenETOC execution time manageable.

After satellite propagation is complete, the specified areas of interest are discretized into grid points so that they can be approximated using a reference grid. The simulation engine steps through each grid point, storing a list of sensor access intervals during which the grid point is within the defined field of view. These access intervals each contain the start and stop times for each instance of the grid point coming into view of a satellite’s sensor. These time windows are then combined with any defined constraints (such as lighting conditions) to obtain a list of satisfaction intervals for which all viewing constraints are met.

The catalogue of satisfaction intervals for a target area is the set of data necessary to evaluate the coverage objectives used in GenETOC. The selectable objective metrics for GenETOC are the median time between revisits of a target area, the median number of revisits to a target area, the percent coverage of a target area, and the time until full coverage of a target area. The satisfaction interval data is parsed to compute the selected evaluation metrics for the analysis.

Optimizer

The optimizer is the driving force behind converging the solution space towards better performing constellation solutions. Since many of the coverage metrics (described in the upcoming section discussing the Simulation Engine) are competing objectives (higher performance in one objective sacrifices performance in other objectives) there is no singular optimal solution. A Multi-Objective GA, like the NSGA II, attempts to optimize for competing objectives and illustrate the tradeoffs between the conflicting optimization metrics. The desired result of the optimization is to achieve a Pareto Optimal Front [2]. The concept of Pareto optimality defines the goal of a Multi-Objective GA by establishing a front or series of discrete fronts, which represent the set of optimal solutions that exist. In any multi-objective problem, there is often a family of equally optimal solutions rather than a single optimal point. Figure 1 is an example of a Pareto optimal front, where every point on the curve is equally optimal.

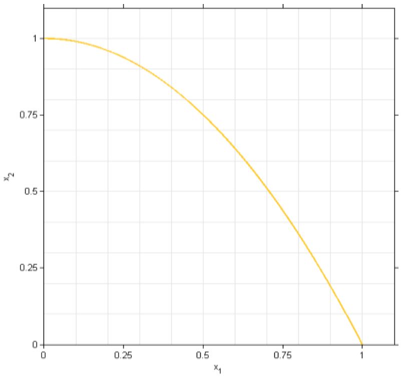


Figure 1. Pareto Optimal Front

Pareto optimality depends upon an idea called domination. A solution is said to dominate another if it is better in at least one objective, and no worse in any of the remaining objectives. To establish a Pareto front, this comparison is repeated over the entire population, and the subset of solutions which are non-dominated (no solution was found to dominate them) define the current Pareto front. Multi-Objective GAs create a series of Pareto fronts for each generation, and subsequent generations push the current fronts towards the true Pareto optimal front. The success of any Multi-Objective GA is based on how close the final solutions are to the true Pareto front.

The GA used for the GenETOC optimizer is a slightly modified version of the NSGA II algorithm [2]. The modifications made to NSGA II change the population creation routines to promote faster convergence (at the risk of losing some solution diversity) and increase the chances of finding high value solutions even during shortened executions of GenETOC. The optimizer constitutes the majority of main process flow for GenETOC (Figure 2). The optimizer receives inputs from the user interface and uses them as controls during the program's execution.

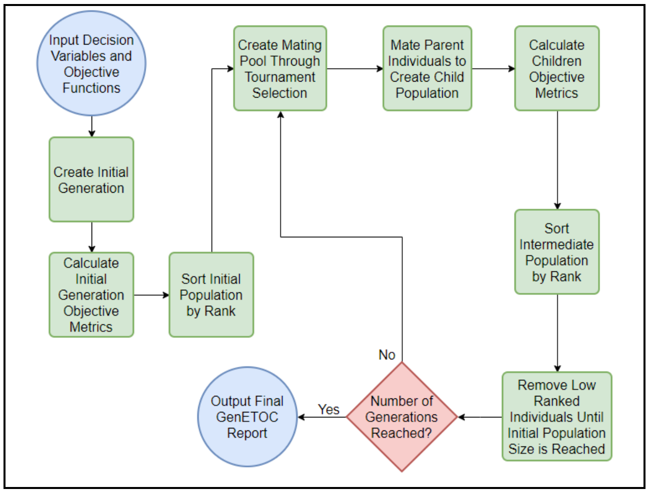

Input Decision Variables and Objective Functions
Create Initial Generation
Calculate Initial Generation Objective Metrics
Sort Initial Population by Rank
Create Mating Pool Through Tournament Selection
Mate Parent Individuals to Create Child Population
Calculate Children Objective Metrics
Sort Intermediate Population by Rank
Remove Low Ranked Individuals Until Initial Population Size is Reached
Number of Generations Reached?
No
Yes
Output Final GenETOC Report

Figure 2. GenETOC Iteration Process

The algorithm starts with an initial population of constellations (generation 0). The initial generation of constellations have randomized orbital parameters that are constrained within bounds set by the user. The objective metrics for each constellation are calculated using the simulation engine. After the objective metrics are calculated, the optimizer evaluates and sorts members of the population based on their performance. The sorting method used for the population of constellations is the non-dominated sorting algorithm (Figure 3). A constellation is considered 'dominated' by another when it is outperformed in every possible objective metric.

The constellation is also given a secondary ranking called crowding distance. Crowding distance is a measure of difference from a constellation's objective metrics to other constellations of the same rank. Higher crowding distance is more valuable as it promotes diversity of the solution space. Diversity is important to the end product of GenETOC because it ensures the mission designer has more options to explore within the near optimal solution set.

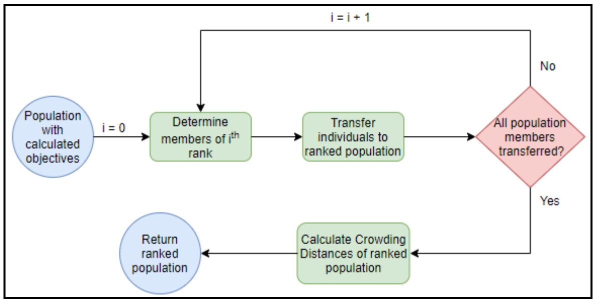

i = i + 1
No
Population with calculated objectives
i = 0
Determine members of ith rank
Transfer individuals to ranked population
All population members transferred?
Yes
Return ranked population
Calculate Crowding Distances of ranked population

Figure 3. Non-Domination Sorting Process

The mating process (Figure 4) is the process of combining the traits of more successful constellations from the population to see if the new solution produces a more successful constellation. Two randomly selected constellations from the mating pool become parents and pass on their traits to offspring constellations. Orbital parameters are passed to offspring through two methods: 1) by directly copying the parameters from a single parent to offspring, or 2) through a combination process called simulated binary crossover [2]. Offspring constellations are created until the number of offspring constellations is equal to the number of constellations in generation 0. After the offspring constellations are created, each orbital parameter of a constellation has a small chance of undergoing mutation [2].

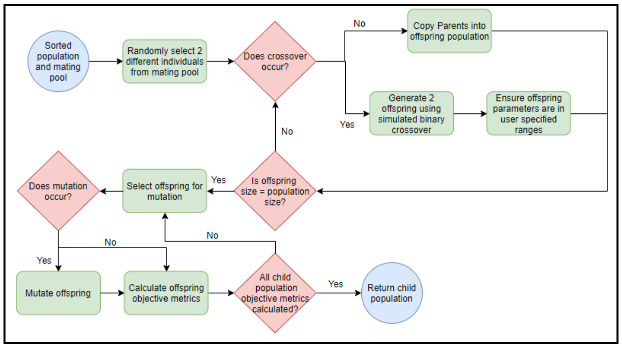

Sorted population and mating pool
Randomly select 2 different individuals from mating pool
Does crossover occur?
No
Copy Parents into offspring population
Yes
Generate 2 offspring using simulated binary crossover
Ensure offspring parameters are in user specified ranges
No
Does mutation occur?
Select offspring for mutation
Yes
Is offspring size = population size?
No
No
Yes
Mutate offspring
Calculate offspring objective metrics
All child population objective metrics calculated?
Yes
Return child population

Figure 4. Mating Process

Mutation is an important step of the optimization process; it provides mechanism to create more diversity in the solution space when the algorithm begins to converge towards a local optima. Adding in this chance of mutation enables the solution set to begin to drift out of the effective gravity well created by the localized optima.

Once mutation is complete, the objective metrics are evaluated for the offspring population using the simulation engine. The combined set of constellations from generation 0 and the offspring are sorted using the non-dominated sorting algorithm. The lowest performing member of the population of this combined set is removed until the number of constellations in the set matches the initial population size (Figure 5). If constellations are of the same tier in terms of domination, crowding distance becomes the determining factor for removal.

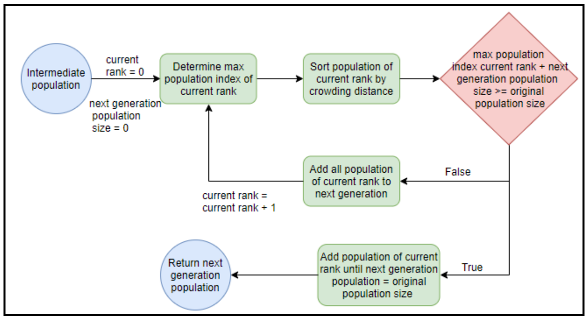

Intermediate population
current rank = 0
next generation population size = 0
Determine max population index of current rank
Sort population of current rank by crowding distance
max population index current rank + next generation population size >= original population size
False
Add all population of current rank to next generation
current rank = current rank + 1
True
Add population of current rank until next generation population = original population size
Return next generation population

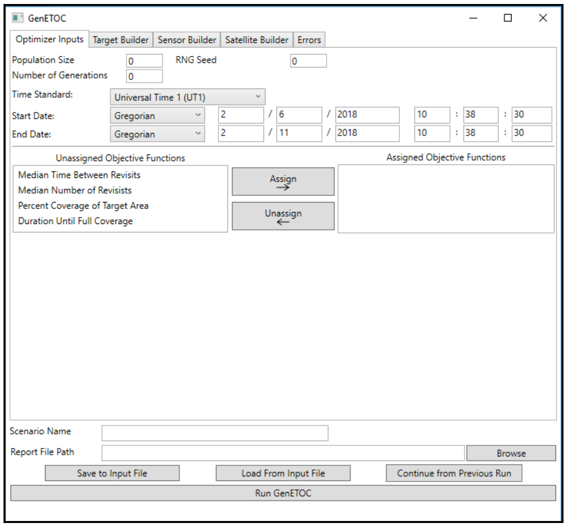
Figure 5. Chromosome Replacement Process

The new sorted set becomes generation 1. As there are no true tests for whether or not the optimal Pareto front has been found, the process flow that was outlined in this section is repeated until the user specified number of generations has been created and evaluated.

User Interface

The User Interface provides a mechanism for the user to control program setup, program execution, input settings, and data visualization. In the interface, there are four major input constructs provided to the user: optimizer settings, area of interest definition, sensor definition, and satellite definition.

Users can modify parameters of the genetic algorithm through the optimization control tab (Figure 6). The optimization tab contains settings for both the simulation engine and the optimizer. These settings control objective metric selection, population and generation counts for the optimizer, and time data for the scenario (epoch, start and stop times). GenETOC currently is capable of optimizing towards any combination of the four objectives.



Population Size 0 RNG Seed 0
Number of Generations 0
Time Standard: Universal Time 1 (UT1)
Start Date: Gregorian 2 / 6 / 2018 10 : 38 : 30
End Date: Gregorian 2 / 11 / 2018 10 : 38 : 30

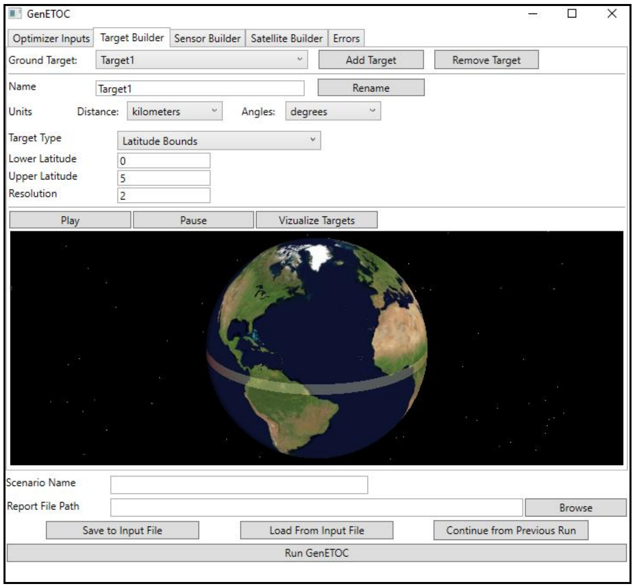
Unassigned Objective Functions
Median Time Between Revisits
Median Number of Revisits
Percent Coverage of Target Area
Duration Until Full Coverage

Assigned Objective Functions

Scenario Name
Report File Path

Figure 6. Optimization Screen

The area of interest input control (Figure 7) enables the user to create and modify the ground targets for use by the simulation engine. GenETOC supports five types of targets: point, circular, latitude bounds, longitude bounds, and global. The user can control the grid resolution of a target to ensure that the simulation fidelity is sufficient for the analysis problem. Users are able to display the created targets on an interactive globe to enable a visual confirmation that the target has been properly defined.



Ground Target: Target1

Name: Target1

Units Distance: kilometers Angles: degrees

Target Type: Latitude Bounds

Lower Latitude: 0

Upper Latitude: 5

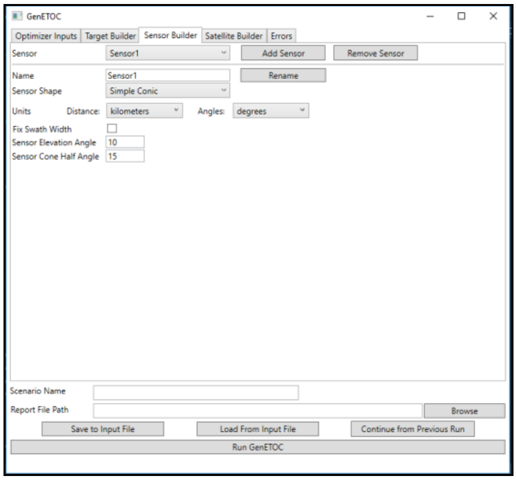
Resolution: 2

Scenario Name

Report File Path

Figure 7. Target Control Screen

The sensor input control tab (Figure 8) enables the user to create sensor configurations that can be attached to any satellite within the simulation. GenETOC supports both simple conic and rectangular prism sensor definitions. In the case of a conic sensor, the half angle of the cone is the user’s point of control and for rectangular prism sensors, the cross-track and along-track half angles are user definable. Users can also select a fixed swath width, which will remove the option to adjust angles, and the user control becomes the cross-track width of the swath on the ground. Algorithms inside of the simulation setup process will determine the appropriate angle values to maintain the fixed swath whenever the starting altitude of a constellation is adjusted.



Sensor: Sensor1

Name: Sensor1

Sensor Shape: Simple Conic

Units Distance: kilometers Angles: degrees

Fix Swath Width ☐

Sensor Elevation Angle: 10

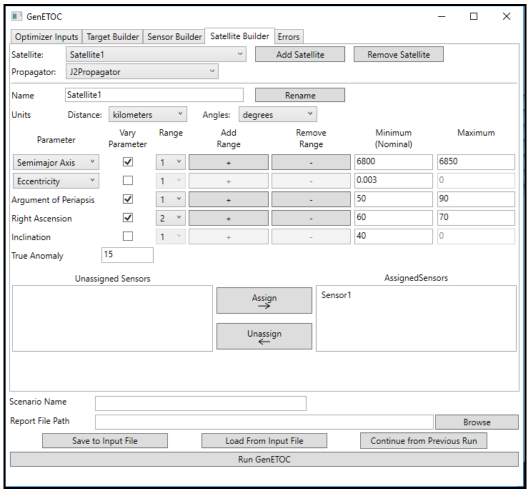
Sensor Cone Half Angle: 15

Scenario Name

Report File Path

Figure 8. Sensor Builder Screen

The satellite input control tab (Figure 9) is where the user can set the orbital parameters for each satellite. The user has the ability to fix or vary five orbital elements. The first two parameters contain the option for the user to specify any appropriate combination of semi major axis, eccentricity, apogee and perigee. The remaining three parameters are argument of periapsis, the right ascension of ascending node, and inclination. True anomaly is omitted from the user input controls due to some of the initial limitations in the optimizer setup. Within the simulation framework, merely swapping the true anomaly values of two satellites within the constellation would be viewed a unique solution. To avoid this situation, all satellites within a single orbit plane are assumed to be equally spaced in true anomaly. For each of the controllable parameter, the allowable variation can be constrained within one or more user-defined, discrete ranges.



Satellite: Satellite1

Propagator: J2Propagator

Name: Satellite1

Units Distance: kilometers Angles: degrees

Parameter	Vary Parameter	Range	Add Range	Remove Range	Minimum (Nominal)	Maximum
Semimajor Axis	☑	1	+	-	6800	6850
Eccentricity	☐	1	+	-	0.003	0
Argument of Periapsis	☑	1	+	-	50	90
Right Ascension	☑	2	+	-	60	70
Inclination	☐	1	+	-	40	0
True Anomaly	15					

Unassigned Sensors

AssignedSensors: Sensor1

Scenario Name

Report File Path

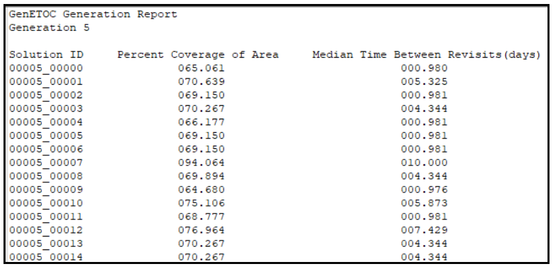
Figure 9. Satellite Builder Screen

The execution control options can be seen in the bottom of Figures 6, 7, 8, and 9 as these options are accessible from any tab in the user interface. These options give users control over file management of data that is generated during an execution of GenETOC. Users can specify the name of the scenario and where data output will be stored. Users are also given the option to save the current set of user inputs, which gives users a mechanism to save progress if interrupted and enables a mechanism for archiving the input set with the optimization results so that the traceability of the scenario input sets can be maintained. This section of the UI allows users to kick-off an optimization run or to continue processing from the end point of a previous run. At any point during execution, the user can decide to cancel the run and GenETOC will finish the current generation and safely termination execution.

Data Visualization and Results

GenETOC regularly reports constellation data each time a population is finished processing. The regular reports give users the option to save progress and monitor the current progression of the data. After each generation is complete, the data for that generation is stored within GenETOC's file management system. GenETOC also supports 2D data plotting for user analysis.

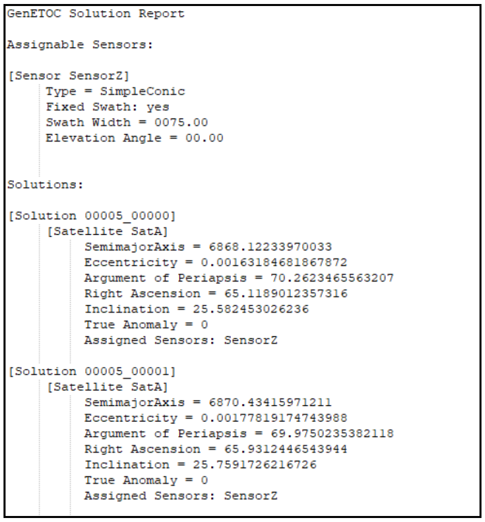
Population data from each generation is stored in GenETOC result and solution files. Result files (Figure 10) contain the objective performance of each constellation of the population. Solution files (Figure 11) contain an elaboration of each member of the population such as the orbital parameters of each satellite for further analysis if desired. The solution and result files can be cross-referenced using the unique ID system GenETOC has in place. The goal of this system is to enable a mission designer to first examine a particular constellation of interest in the report file; if the designer requires further elaboration on the particular constellation, the unique ID is traced to the solution file for further analysis.



```
GenETOC Generation Report
Generation 5

Solution ID     Percent Coverage of Area     Median Time Between Revisits(days)
00005_00000              065.061                           000.980
00005_00001              070.639                           005.325
00005_00002              069.150                           000.981
00005_00003              070.267                           004.344
00005_00004              066.177                           000.981
00005_00005              069.150                           000.981
00005_00006              069.150                           000.981
00005_00007              094.064                           010.000
00005_00008              069.894                           004.344
00005_00009              064.680                           000.976
00005_00010              075.106                           005.873
00005_00011              068.777                           000.981
00005_00012              076.964                           007.429
00005_00013              070.267                           004.344
00005_00014              070.267                           004.344
```

Figure 10. Result File



```
GenETOC Solution Report

Assignable Sensors:

[Sensor SensorZ]
     Type = SimpleConic
     Fixed Swath: yes
     Swath Width = 0075.00
     Elevation Angle = 00.00


Solutions:

[Solution 00005_00000]
     [Satellite SatA]
          SemimajorAxis = 6868.12233970033
          Eccentricity = 0.00163184681867872
          Argument of Periapsis = 70.2623465563207
          Right Ascension = 65.1189012357316
          Inclination = 25.582453026236
          True Anomaly = 0
          Assigned Sensors: SensorZ

[Solution 00005_00001]
     [Satellite SatA]
          SemimajorAxis = 6870.43415971211
          Eccentricity = 0.00177819174743988
          Argument of Periapsis = 69.9750235382118
          Right Ascension = 65.9312446543944
          Inclination = 25.7591726216726
          True Anomaly = 0
          Assigned Sensors: SensorZ
```

Figure 12 Solution File

GenETOC also creates data visualizations after each generation (Figure 12). Interim data reports are displayed at intervals for the user to monitor. In the case of erroneous inputs or a non-converging scenario, the interim data output helps the user identify any complications with the current scenario and visualize the state of convergence. For each generation, GenETOC plots each solution on a 2D scatter plot on the completion of each generation (note: this is only for a scenario with two objective functions. GenETOC currently does not support 3D+ graphing capabilities). The user can use this plot to visualize the trend of the data after each generation, to acquire more in-depth data on particular solutions, or to export an image of the plot to use. The user can also click on a data point in the scatter plot to open a summary window of that data point. The summary window contains the same data as the solution files for the selected point.

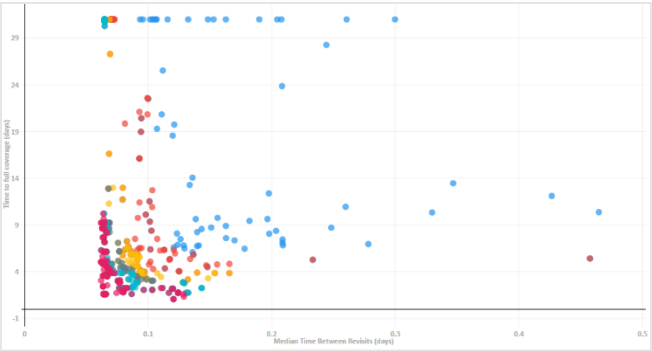


Figure 12. GenETOC Data Visualization

Example Design Application

To illustrate the application of GenETOC, we will examine a notional, simple mission design problem. In this example, a mission designer is tasked with designing a constellation to observe the air quality over eleven major cities across the world (Table 1). The constellation will have three CubeSats, constrained to a 500 km circular orbit. Each satellite will be equipped with a sensor with a 5 degree half angle swath. Given these constraints to the mission, the designer creates an optimization scenario for GenETOC to start exploring potential designs. The scenario is set over the period of one month. The constellation has two conflicting objectives: the first is to minimize the amount of time it takes for the constellation to visit each target at least once, the second is to minimize the median time between revisits of a target. Since the altitude is fixed with a circular orbit, the only variable parameters are the right ascension of ascending node and the inclination of the orbits; each parameter is set to vary between 0 and 90 degrees (Table 2).

To keep the program runtime manageable, the population size will be set to 500 and the generation size will be set to 15, which is enough genetic diversity to obtain a variety of solutions while keeping the runtime within 24 hours. The mission designer runs GenETOC to begin the optimization process.

Table 1. Sample Problem Targets

City	**Latitude** (degrees)	**Longitude** (degrees)
New York	40.7128	-74.006
Los Angeles	34.0522	-115.2437
Chicago	41.8781	-87.6298
London	51.5074	-0.1278
Florence	43.7696	11.2558
Cairo	30.0444	31.2357
Johannesburg	-26.2041	28.0473
Mumbai	19.076	72.8777
Seoul	37.5665	126.978
Sydney	-33.8688	151.2093
Rio De Janeiro	-22.9068	-43.1729

Table 2. Sample Problem Decision Variables

Satellite / Parameter	**Right Ascension of Ascending Node Ω (degrees)**		**Inclination *i* (degrees)**	
	Min	**Max**	**Min**	**Max**
Satellite 1	0	90	0	90
Satellite 2	0	90	0	90
Satellite 3	0	90	0	90

The first set of data generated by GenETOC (Figure 13) contains the variety of potential solutions from the first generation.

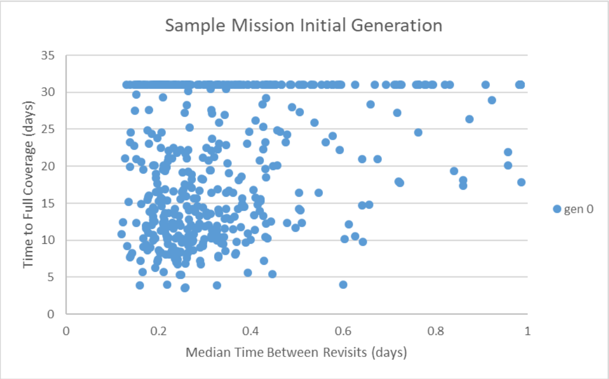


Figure 13. Initial population for the sample problem

The first generation contains the largest spread of solutions. Note the solutions with a value of 32 days for the 'Time to Full Coverage' objective have failed that objective. In other words, these solutions did not reach each target within the one-month scenario time.

GenETOC iterates each subsequent generation towards the optimal (Figure 14). By the fifth generation, most failed solutions have been phased out and replaced by solutions along a higher performance curve. The later generations no longer have the wide spread of the first generation. They are converged closer towards the optima and tightly packed together.

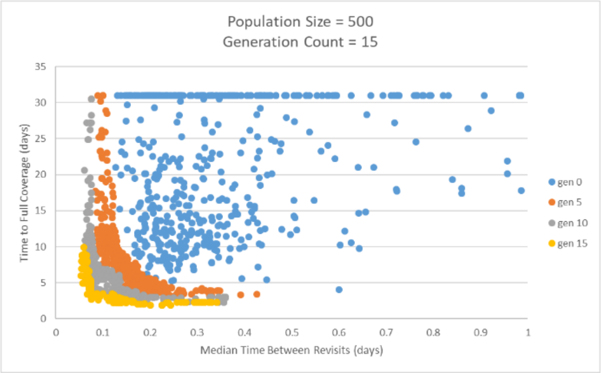


Figure 14. All populations for the sample problem

The final iteration of the program (generation 15) yields a more optimal front with no failed solutions.

GenETOC produces reports for each generation, giving the mission designer access to the performance metrics and orbital parameters of each solution. The mission designer will have access to a variety of optimized solutions allowing the mission designer to compare solutions based on other metrics, such as cost, feasibility, and scheduling. As an example, the mission designer could determine that the solutions with the fewest number of launches are of higher priority and select viable solutions with that constraint. Tables 3 and 4 contain data from four prominent solutions from the analysis.

Table 3. Sample Solution Performance

Solution	Median Time Between Revisits (days)	Time to Full Coverage (days)
15_04	0.053	5.962
15_06	0.243	1.865
15_08	0.122	2.232
15_19	0.122	2.385
15_22	0.070	3.439

Table 4. Sample Solution Parameters

Solution	Satellite 1		Satellite 2		Satellite 3	
	Ω	***i***	**Ω**	***i***	**Ω**	***i***
15_04	24.2	42.0	8.2	42.0	74.5	51.5
15_06	85.4	33.1	53.6	68.0	67.3	81.4
15_08	35.9	89.0	66.9	47.0	70.0	41.7
15_19	35.8	89.4	67.5	47.2	70.9	42.1
15_22	27.0	43.8	16.1	43.8	77.0	51.5

A successful GenETOC analysis will yield a variety of potential constellation configurations. Solution 4 is the best performing constellation for the median time between revisits objective, and solution 6 is the best performing constellation for the time to full coverage objective. Solutions 8 and 22 strike a better balance between both objectives. Each solution belongs to the same front, where no solution is strictly better in both solutions than any other. This provides the mission designer multiple avenues to approach the constellation design.

From this example case, GenETOC produced 500 potential solutions for further examination. The only constraints applied to the satellites are the original orbital parameters; it is up to the mission designer to apply additional constraints (launch vehicle requirements, costs requirements, etc.). For solution 4, satellites 1 and 2 are separated by less than a degree in inclination and less than 30 degrees right ascension. For solution 19, both the right ascension and inclination differ less than 5 degrees each. These are potential cases in which, given slight modifications, two satellites can be launched together. This is potentially beneficial for the mission as launching two satellites in one vehicle can cut costs and provide the mission more flexibility.

In this scenario, GenETOC presents a group of solution of similar quality for mission designer. The analysis done using GenETOC gives a starting point for the mission, where further exploration into solutions give the designer a first order view of potential mission configurations.

Conclusion

With the advent of small satellite constellations, the problem of coverage optimization increases in complexity as the number of satellites per constellation increases. This increase in complexity renders traditional analytical optimizations inefficient in comparison to automated solutions. Few orbit design tools are available to the mission designer that can provide a broad spectrum of near optimal solutions. GenETOC was designed to meet the growing complexity of the problem. It is able to efficiently give the mission designer a first look into a diverse set of near optimal solutions in the trade space.

To obtain the set of near optimal solutions, GenETOC relies on a GA for solution discovery. Generational GAs have the ability to start with a broad description of the design problem and narrowing down the potential avenues of solution exploration to a family of solutions along the optimal curve. The initial set of bounded randomly generated constellations guarantees a wide view of potential configurations, and the iteration process converges the family of solutions towards the optima. It is essential for a mission designer to have this breadth of constellation configurations to explore in the orbit design process.

GenETOC has the unique capability of applying useful features of a GA to a satellite constellation scenario. The Space Mission Analysis Branch at NASA Langley intends on using GenETOC as a tool to produce mission design options for exploration. In the absence of analytical or intuitive approaches to a constellation optimization problem, GenETOC will prove to be a useful tool in creating a first order look into the orbit design trade space.

References

[1] E. Zitzler, K. Deb, and L. Thiele, “*Comparison of Multiobjective Evolutionary Algorithms: Empirical Results*,” Evolutionary Computation, vol. 8, no. 2, Summer 2000: 173-195.

[2] Kalyanmoy Deb, Associate Member, IEEE, Amrit Pratap, Sameer Agarwal, and T. Meyarivan, “*A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II*,” IEEE Transactions on Evolutionary Computation, vol. 6, no. 2, April 2002

[3] J.H. Holland, *Adaptation in Natural and Artificial Systems*, University of Michigan press, Ann Arbor, 1975.

Biography

Joshua Carden

Shaun Deacon

Paul Kessler

Paul Speth