Model-based heterogeneous optimal space constellation design

Katherine Mott
Department of Aerospace and Ocean Engineering
Virginia Tech
Blacksburg, VA 24060
Email: kemott@vt.edu

Dr. Jonathan Black
Department of Aerospace and Ocean Engineering
Virginia Tech
Blacksburg, VA 24060
Email: jonathan.black@vt.edu

***Abstract*—Few tools exist for designing constellations of heterogeneous satellites. A new modular tool for total mission design of heterogeneous constellations, including spacecraft design, orbit selection, and launch manifestation, is proposed. The component modules and algorithms are discussed, including a novel crossover method for genetic algorithms and a novel constraint formulation for launch manifestation of maneuverable vehicles. Finally, the expandability of the tool to multiple domains and various applications is highlighted. The main benefit of the described tool is that it allows analysts to quickly and easily design complex system architectures for space systems with a variety of objectives, providing cost savings and enabling timely responses to changing mission needs.**

I. Introduction

With the increasing popularity of small, micro, and nanosatellites, the economic viability of launching several satellites to do a task hitherto performed by a single large satellite is increasing. Traditional constellation design methodology and tools are not equipped to compare the performance of a typical constellation of identical satellites to the performance of a heterogeneous constellation, which is comprised of different satellites of varying capabilities. In such a constellation, tasking and payloads may be divided among several satellites instead of being aggregated on a single satellite. This task creates problems not only in orbit design but also in packaging and distribution of sensors and payloads. Additionally, the solution space grows with the increased number of variables. Proposed efforts by private companies to launch constellations of thousands of satellites highlight the need for a method of comparing various architectures and regimes in which large numbers of decision variables must be determined.

Existing efforts use genetic algorithms to design the orbits of large constellations. However, little focus has been given to simultaneous determining spacecraft parameters and orbital parameters, although the two are intimately linked. To address the unique challenges of heterogeneous constellation design, a model-based systems engineering tool was created. This tool uses modules representing physical systems on the satellite, dynamic behavior, and mission performance. Current modules include individual sensors, communication systems, solar panels, satellite propagation, coverage metrics, dilution of precision calculation, monetary cost, and launch vehicles. The models are provided to the optimizer, which evaluates subsystem combinations and orbit design options. The integration between spacecraft, payload, and orbit design permits a holistic approach to mission design that avoids the elimination of viable mission architectures early in the design process.

This research uses a genetic algorithm to generate a non-dominated set of solutions that approximate the Pareto frontier of the multiobjective optimization problem, describing configurations of heterogeneous satellites that accomplish the given mission. A genetic algorithm was selected because the solution space may be complex and non-convex as determined by the user-defined constraints and objectives. The stochasticity of the algorithm assists in traversing the solution space, may enable the algorithm to reach areas unavailable to convex optimizers, and reduces the likelihood that the solver will become stuck in a local minimum. However, the extent to which the problem is nonlinear and the tuning parameters chosen for the genetic algorithm can affect convergence. A variable length genetic algorithm is employed, which allows the number of satellites and number of planes in the proposed constellation to vary. The genetic algorithm also assigns various sensor packages to the satellites, influencing the satellite size and monetary cost. This research introduces design methodology and tools that can be applied to a variety of missions. Particularly, missions with several objectives or missions whose target areas are asymmetric about the equator will benefit most from this analysis. Missions with payloads of differing orbital and access requirements will also benefit from disaggregation using this tool. Because individual sensors, mission goals, launches, and constraints are modelled separately and interpreted and combined by the optimizer software, the analysis is modular and therefore easily varied for different applications.

II. Background

Analytically and geometrically derived solutions exist for specific types of satellite missions, such as those requiring global coverage [1]–[4]. Likewise, there are well-documented methods for maneuvering between known configurations of satellites or to known target locations [4], [5]. However, more complicated mission design efforts involving noncircular orbits, perturbations, regional coverage, heterogeneous constellations, constellation reconfiguration, non-coverage objectives, or multiple objectives are impossible to solve analytically.

978-0-9964527-7-9 ©2018 ISIF

Modern constellation design therefore relies heavily on numerical simulation and optimization techniques. For example, the DISCO constellation design methodology uses statistical coverage models and a genetic algorithm to determine optimal constellation and payload parameters when comparing traditional homogeneous constellations to heterogeneous constellations comprised of multiple spacecraft of differing capability [6]. DISCO has been applied to various scenarios including fire detection and weather [7], [8].

Similar efforts in industry and in other domains highlight interest in the concept of heterogeneous swarms of vehicles. In particular, several Defense Advanced Research Projects Agency (DARPA) projects are of particular interest. The DARPA Gremlins program, in development by Dynetics, aims to use lightweight, unmanned air systems as a basis for a variety of payloads [9]. The DARPA Hydra program seeks to extend U.S naval capabilities by augmenting manned Navy vessels with modular, unmanned, undersea payloads, describing a system that is not only heterogeneous but also multi-domain [10]. Finally, the DARPA Complex Adaptive System Composition And Design Environment (CASCADE) program calls for the development of a "unified view of system behavior" for complex, system-of-systems architectures, highlighting the need for tools that effectively manage such complex systems. The proposed research is capable of addressing these needs and providing a platform with broad applicability to multi-domain mission design problems.

The research described below is a follow-on to the DISCO methodology that remedies shortcomings in the original formulation. Despite its utility, DISCO is limited by the need to pre-define the architectures to be considered. It is also not suitable for assignment problems in which commercial off the shelf components (COTS) are used. DISCO does not consider cases of constellation resilience and reconfiguration. Finally, DISCO was reliant on commercial software for calculating accurate access metrics, so parallelization was cost prohibitive. The new simulation software addresses these shortcomings. Previous applications of this tool include the development of a Mars-based communications, navigation, and data relay constellation [11].

III. Methodology

The tool uses a modular approach to allow the expansion of the code as the needs of end users change. An outline of the interaction between high-level modules is shown in Fig. 1. Assets, constraints, mission objectives, and optimization properties are defined prior to the beginning of the optimization process, during which candidate solutions are evaluated for fitness based on the stated objectives. Additional asset types, such as ground or air-based sensors, could be added during the asset definition stage if suitable dynamics models are provided.

A. Optimizer

The design space for spacecraft and constellation optimization problems is large, as each spacecraft requires several parameters to define its configuration and orbit. Furthermore, the constraint and objective functions used to evaluate solutions are not guaranteed to be convex, so convex optimization techniques cannot be used. Instead, a stochastic optimization technique, a modified version of the Nondominated Sorting Genetic Algorithm II (NSGA-II), is employed. Because NSGA-II generates nondominated fronts for problems with multiple objectives, a variety of feasible solutions can be found without assigning arbitrary weights to the different objectives. The genetic algorithm has four parts: sorting, selection, crossover, and mutation. An optional fifth part, constraint handling, is also discussed. Solutions are represented by a "genome," a series of numbers that correspond to solution properties. The algorithm described improves upon standard genetic algorithms by allowing genome lengths to vary, allowing constellations of various sizes to be modeled without resorting to turning segments of the genome "off" using flagging. Additional improvements in genetic algorithm research such as epsilon boxing, dynamic population sizing, and re-invigoration can be combined with this novel crossover technique [12].

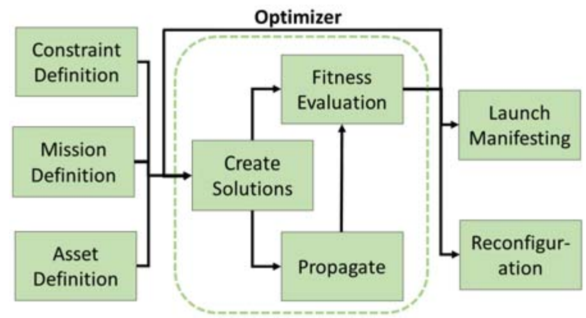

Fig. 1. Module interaction during optimization.

1) Sorting: NSGA-II groups candidate solutions from the current and previous generations into a series of nondominated fronts based on one or more fitness values. Crowding distance, a measure of the difference in fitness between nearby solutions, is calculated within the nondominated fronts. To encourage elitism and diversity, NSGA-II generates the next set of parents by prioritizing the most dominant solutions and then the solutions with the largest crowding distance within a front [13].

2) Selection: Once the set of parents has been established, tournament selection is used to create pairs for crossover. By default, binary tournament selection is used, though users can define a different number of competitors if desired. In tournament selection, the current candidate is compared to a new candidate each round. If the candidates are of different nondominated fronts, the less dominated candidate is chosen. If candidates are of the same front, the candidate with the larger crowding distance is chosen. Once one parent has been chosen, the process is repeated after removing that parent from the selection pool to ensure that the second parent will be different from the first. The number of pairs selected is equal to half the population size, rounded up.

3) Crossover: The crossover algorithm used in this tool is a departure from the standard crossover technique used in most genetic algorithms, including NSGA-II. The algorithm is designed to permit length change in the genome. A previous example of a variable length genetic algorithm is the exG algorithm developed at the California Institute of Technology [14]. Unlike exG, the crossover algorithm used divides the genome into segments of different types and ensures that crossover will only occur between like segments. First, a random value is compared to the length change rate (LCR) to determine if length changing is permitted. If a length change can occur, the starting and ending segment type are chosen randomly. The segment numbers are then chosen randomly for both parents. The starting and ending position within a segment is also chosen randomly but must be the same for both parents. Furthermore, limits are imposed on the maximum and minimum number of segments of each type and selections changed if the crossover will violate these limits. Once the portions to be swapped have been determined, a weighted swap is performed between the parents, resulting in two children. The crossover operation is performed for each pair chosen in the selection section, forming the next generation. Pseudocode for the crossover operation is shown below.

l =rand(1)
Choose starting and stopping segment types
i_1, i_2 =empty - indices to be swapped
if Starting and stopping types are the same **then**
  Choose the starting and stopping segment numbers for each
  if $l > LCR$ **then**
    No length change - shift the stopping segment number of the longer segment back so it matches the length of the shorter segment
  else if the swap will violate the segment number limits **then**
    Reduce the size of the larger segment being moved
  end if
  Choose the starting and stopping position in the segment.
  Add indices from starting and stopping positions to the first position in i_1 and i_2
else
  Choose the starting segment numbers for the starting segment type. The stopping segment number will be the last segment of this type.
  if $l > LCR$ **then**
    No length change - shift the starting segment number of the longer segment forward so it matches the length of the shorter segment
  else if the swap will violate the segment number limits **then**
    Reduce the size of the larger segment being moved
  end if
  Choose the starting position in the segment.
  Add indices from starting positions to the end of the segment type to the first positions of i_1 and i_2
  for j = 1: number of segment types between starting and stopping **do**
    Add indices from the start of the type to the end of the shorter type to the j+1 positions of i_1 and i_2
  end for
  Choose the stopping segment numbers for the stopping segment type. The starting segment number will be the first segment of this type.
  if $l > LCR$ **then**
    No length change - shift the stopping segment number of the longer segment backward so it matches the length of the shorter segment
  else if the swap will violate the segment number limits **then**
    Reduce the size of the larger segment being moved
  end if
  Choose the stopping position in the segment.
  Add indices from the start of the segment type to the stopping positions to the last positions of i_1 and i_2
end if
r =rand(1)
child1 = parent1; child2 = parent2
for k =1: number of segment types from starting to stopping **do**
  $s_1 =$ portion of first parent corresponding to $i_1(k)$
  $s_2 =$ portion of first parent corresponding to $i_2(k)$
  Extend s_1 or s_2 so lengths are equal
  $s_2 = s_2 - r(s_2 - s_1)$
  $s_1 = s_1 - r(s_1 - s_2)$
  Shorten s_1 or s_2 back to original lengths
  Remove s_1 from child1 and insert s_2 in its place
  Remove s_2 from child2 and insert s_1 in its place
end for

As an example, consider a constellation and spacecraft design problem in which the genome represents properties of the satellite hardware (such as the communications subsystem, assigned payloads, etc.), orbits (such as the orbital elements at some epoch), and individual satellites (such as the type of satellite, the orbit, the position along the orbit, and thrust vectors for maneuvers). The variable length crossover operation described above would allow the number of satellite types, the number of orbits, and the individual satellites to vary as a result of the optimization. This process is useful when the required number of satellites, orbits, or types is not known exactly. By using this technique, a heterogeneous constellation can be easily modeled, as the user needs only to model base satellite types (discussed further in the Assets section) rather than specific constellation architectures.

4) Mutation: A predefined mutation rate determines the likelihood that an individual element of the genome will mutate. For each element of the genome, choose a random number and compare it to the mutation rate. If the random number is less than or equal to the mutation rate, change that element of the genome to a random value in the set of admissible values for that element. Perform this process for every element of every genome in the new generation.

5) Constraint Handling: Constraints can be handled in one of two ways. The first option evaluates constraints for each genome immediately after the mutation phase. If a genome violates a constraint, an element of the genome that affected that constraint is mutated. Then, all of the constraints are rechecked. The process continues until the genome satisfies all constraints. This method can be useful if the evaluation of the constraints is straightforward, as it directly affects the relevant parameters. However, the code will become stuck in the loop if there are no valid solutions, and could take a long time to find a solution if the solution space is small. Furthermore, it does not consider the extent to which a constraint is violated.

The second option incorporates constraints into the evaluation function. Constraints are checked as close to the beginning of the evaluation function as possible. For each violated constraint, a penalty value is increased based on the severity of the violation unless the constraint is fixable. If the constraint is fixable, the genome can be modified using a known process in order to satisfy the constraint without affecting any other constraints. The constraint is then satisfied. If no constraints were violated, the code continues. If one or more constraints were violated, the fitness is set to the value of the penalty and the remaining code skipped. One advantage of this method is that it considers the extent to which the constraints are violated, unlike the first method. The genetic algorithm will minimize the constraint violation, eventually pushing the solution into the admissible space. Furthermore, if no valid solutions exist, solutions will be found that minimize the number of constraints violated. A disadvantage of this method is that it could take a long time if no initial solutions are near the admissible space. Neither constraint method currently considers the ability of an inadmissible solution to satisfy the mission objectives, though the second method could easily be adapted to consider the inadmissible solution performance.

The current constraint types used by the design tool are link budgets and fuel constraints. The link budget between two or more assets can be established, and the bit error rate or signal to noise ratio compared to required values. The fuel constraint calculates the mass of fuel needed to perform any maneuvers assigned to a particular satellite with known mass and specific impulse, then determines if the fuel mass needed exceeds the allowed mass for the satellite. The fuel constraint is fixable. If the fuel constraint fails, the maneuver values are reduced such that the total fuel needed for all maneuvers is equal to the maximum allowable fuel mass.

B. Assets

Assets are defined prior to beginning an optimization routine. For a constellation and spacecraft design problem, the default asset types are subsystems, satellite structures, orbit definitions, and individual satellites. Each asset is modeled as a structure containing information about relevant properties for that asset. The properties can either be constants not affected by the optimization or variables for which a range or set of admissible values is defined. If the property is a variable, the structure also contains information about the location of the variable in a genome where there is exactly one of each segment type. Subsystems are defined first. Fields for subsystems include the type of subsystem (such as transmitter, battery, camera, etc.) and necessary properties. The fields for satellite structures can include parameters (such as transmit frequency, power, etc.) assigned directly rather than through subsystems, but generally just includes sensor assignment information. A satellite structure designates subsystems or subsystem types as either optional or required. If a satellite structure has multiple options for a given subsystem, the genome will contain a value between zero and one that is processed to select either one of the options or a lack of assignment if the subsystem is optional. An orbit definition contains fields corresponding to classical orbital elements. It can include all or only some of the six parameters necessary to specify the position of a satellite. The final asset type, the individual satellite, must contain fields specifying the spacecraft structure and orbit that are used by that satellite. It also contains the classical orbital elements that are not included in the orbit definition. For example, if the orbit definition defines the orbit by specifying semimajor axis, eccentricity, inclination, longitude of the ascending node, and argument of periapsis, the individual satellite only needs to contain the true anomaly. However, if the orbit definition instead describes only the plane by using inclination and longitude of the ascending node, the individual satellite must contain the semimajor axis, eccentricity, argument of periapsis, and true anomaly. Additionally, the individual satellite may contain information describing maneuvers and the times at which they occur.

The way in which assets are formulated easily permits the addition of new asset types. Ground stations could be assigned subsystems and locations. Other vehicles such as balloons, airplanes, drones, automobiles, and submarines can be defined using subsystems and can be assigned to trajectories or positions in the same manner as satellites, provided suitable dynamic models are available.

C. Propagation

There are two methods available for propagating satellite position over time. The first, more flexible method is a numerical propagator. Conservative forces are determined by taking the gradient of the potential. The gravitational potential energy per unit mass due to a nonspherical body can be characterized using a spherical harmonics representation,

$$V = -\frac{\mu}{r}\Bigg(1+\sum_{l=2}^{\infty}\Big(\frac{a_e}{r}\Big)^l\sum_{m=0}^{l}\big[\overline{C_{lm}}\cos(m\lambda) + \overline{S_{lm}}\sin(m\lambda)\big]\overline{P_{lm}}(\sin(\phi))\Bigg) \quad (1)$$

where V is the gravitational potential energy per unit mass; μ is the standard gravitational parameter for the planet; r, λ, and ϕ are the radius from the planet's center to the satellite, the longitude of the satellite, and the geocentric latitude of the satellite; a_e is the reference radius of the planet used to develop the model; $\overline{C_{lm}}$ and $\overline{S_{lm}}$ are coefficients; and $\overline{P_{lm}}(\sin(\phi))$ is the normalized associated Legendre polynomial of degree l and order m, evaluated for $\sin(\phi)$. The propagator currently truncates the series after the fourth degree terms ($l = 4$) to avoid excessive computation time, though additional terms can be included as needed. The magnetic field of a body could be modeled in a similar manner. The spherical gradient of the potential is taken to obtain the conservative force in the spherical coordinate frame. The force can then be rotated to the planet-centered inertial frame prior to integration. Nonconservative forces, such as drag and control forces, can also be modeled and rotated to the inertial frame. The equations of motion are then integrated numerically using an ordinary differential equation (ODE) solver. By controlling the tolerances of the ODE solver and the force and potential models, satellite positions can be solved with a high degree of accuracy. Furthermore, numerical propagators can be developed for other vehicle types and the equations of motion solved simultaneously. However, this method is computationally intense and becomes infeasible for large numbers of satellites.

The second method uses the Lagrange variation of parameters method to determine the secular effects of the low-degree zonal harmonic terms on the osculating orbital elements over time [15] [16]. The secular effect on the semimajor axis a, eccentricity e, and inclination i are negligible, so those parameters will be treated as constant over time. The rate of changes of the longitude of the ascending node, Ω, the argument of periapsis, ω, and the mean anomaly, M, due to the spherical planet model and second and fourth order terms are

$$\begin{aligned}\dot{\Omega}_{sec} = &-\frac{3J_2a_e^2n\cos(i)}{2p^2} \\ &+\frac{3J_2^2a_e^4n\cos(i)}{32p^4}\Big(12-4e^2-(80+5e^2)\sin^2(i)\Big) \\ &+\frac{15J_4a_e^4n\cos(i)}{32p^4}\Big(8+12e^2-(14+21e^2)\sin^2(i)\Big)\end{aligned} \tag{2}$$

$$\begin{aligned}\dot{\omega}_{sec} = &\frac{3nJ_2a_e^2}{4p^2}\Big(4-5\sin^2(i)\Big)+\frac{9nJ_2^2a_e^4}{384p^4}\Big(56e^2 \\ &+(760-36e^2)\sin^2(i)-(890+45e^2)\sin^4(i)\Big) \\ &-\frac{15J_4a_e^4n}{128p^4}\Big(64+72e^2-(248+252e^2)\sin^2(i) \\ &+(196+189e^2)\sin^4(i)\Big)\end{aligned} \tag{3}$$

$$\begin{aligned}\dot{M} = &\,n+\frac{3na_e^2J_2\sqrt{1-e^2}}{4p^2}\Big(2-3\sin^2(i)\Big) \\ &+\frac{3na_e^4J_2^2}{512p^4\sqrt{1-e^2}}\Big(320e^2-280e^4 \\ &+(1600-1568e^2+328e^4)\sin^2(i) \\ &+(-2096+1072e^2+79e^4)\sin^4(i)\Big) \\ &-\frac{45J_4a_e^4e^2n\sqrt{1-e^2}}{128p^4}\Big(-8+40\sin(i)-35\sin^2(i)\Big)\end{aligned} \tag{4}$$

where $p=\frac{h^2}{\mu}=a(1-e^2)$ is the semi-parameter, n is the mean motion, and the J_2 and J_4 constants are the negatives of the second and fourth degree non-normalized zonal harmonic coefficients. The rates are constant, so the values of Ω, ω, and M can be found by multiplying the rates by the time t and adding that value to the initial value. It is then necessary to find the eccentric and true anomalies, E and ν, to calculate the Cartesian spacecraft position and velocity. The relationship between the mean and eccentric anomalies is described by Kepler's equation, $M=E-e\sin(E)$. Since Kepler's equation is transcendental in E, the equation must be solved numerically. Because solving Kepler's equation at every time step for every satellite would be cumbersome, the solution was precomputed for a 50x50 grid of mean anomalies and eccentricities. Bicubic interpolation is then used to find the eccentric anomaly for any combination of mean anomaly and eccentricity. The interpolation surface and its error relative to the actual solution is shown in Fig. 2. The true anomaly can then be computed from the eccentric anomaly.

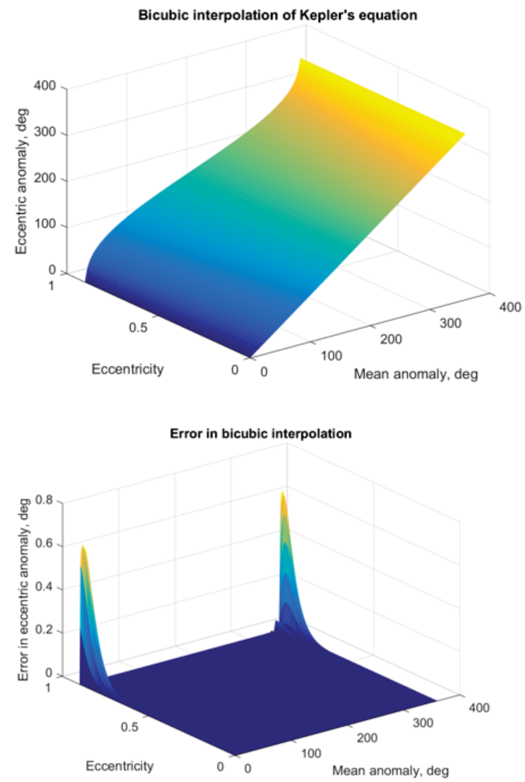

Fig. 2. Interpolation surface and error for Kepler's equation.

Maneuvering: The modeling tool supports impulsive maneuvering. Propagation occurs until the time of a maneuver is reached. The changes in velocity Δv imparted by the maneuvers are then added to the velocities of the corresponding satellites. The position remains unchanged, so the old position and new velocity are used as initial conditions for further propagation. Propagation then continues until the next maneuver time is reached. It is also possible to model more complex control schemes if they are of a well-known form or are analytically derived. Although a large body of research exists with regards to optimal maneuvering of spacecraft, the inclusion of an optimization routine for each satellite's maneuvers would be computationally prohibitive.

D. Evaluation

Because constellation design usually has several conflicting goals, it is a multiobjective optimization problem. Even simple constellations strive to balance the desire for increased performance and the ability to complete a mission with the desire for low monetary cost. A constellation's fitness is calculated for each objective. The fitness metrics can either be combined using user-defined weights to create a single objective or treated separately by a multiobjective optimization algorithm such as NSGA-II. There are currently several objective function types implemented, though the modular approach makes it easy to add new types.

1) Coverage: A coverage definition contains a region of interest, assets, and figures of merit. A region of interest on the planet's surface is defined. The region can either be the entire planet, a region with specified latitude and longitude bounds, or a region defined by a series of boundary points. A grid of points is then generated within the boundary. The assets assigned to the coverage definition can be subsystems, satellite types, orbits, or individual satellites. Any individual satellite assigned to or that has been assigned at least one assigned asset is included in the coverage analysis. At each time step of the coverage scenario, the position vector from each grid point to each included satellite is calculated. The elevation angle of the satellite with respect to the plane tangent to the planet at the grid point can also be calculated. If the elevation angle is greater than or equal to some specified minimum, the satellite is considered to be in view of the grid point. If the satellite is not permitted to change its attitude or if continuous coverage is required, the boresight angle must be calculated to determine if the grid point is in view of the satellite. If the angle from the satellite's boresight vector, usually the nadir vector, to the grid point is less than or equal to the satellite or sensor field of view, then the grid point is in view of the satellite. If the satellite and grid point are mutually in view, then the grid point is considered capable of accessing the satellite at that point in time. A three dimensional array for access between any satellite and any grid point at any point in time can be formed. Using this array and the relative position information, various figures of merit such as revisit time, percent coverage, and positional dilution of precision can be calculated.

2) Resilience: The access array generated in the coverage scenario can also be used as a measure of resilience for the constellation. Because the access array is stored after the initial coverage calculations are performed, the performance of a degraded constellation can be calculated without necessitating the recalculation of accesses. By removing one or more satellites from the access array, figures of merit for the reduced constellation can be calculated. By performing these calculations for different combinations of removed spacecraft, it is possible to acquire bounds on the constellation performance in the event of one or more failures. It can also be determined which satellites are most critical to the mission, allowing redundant satellites to be deployed if needed. A full enumeration of all possible combinations of removals can be performed when the number of satellites and removals is relatively small, but it may be necessary to resort to heuristic techniques for large problems.

3) Cost: Cost modeling for space systems in early phases of development is difficult. Few satellite components are commercial off the shelf (COTS) parts that have set prices. These COTS parts are normally only available for cubesats or other small satellites. If exact pricing is available for specific components, that pricing is used in estimating cost. Otherwise, cost models based on historic satellite costs must be used. Currently, a low fidelity mass model is used to approximate the spacecraft mass, which is then used to approximate the nonrecurring and recurring costs for the satellite. For a simple communications satellite, transmission power is used as the main determinant for mass. The total power used by the communcations system can be approximated as twice the transmitted power due to inefficiencies in the system [17]. Researchers at the Massachusetts Institute of Technology determined a relationship between the communication system power and the satellite dry mass through analysis of previous nongeosynchronous communications satellites. The relationship is

$$M_{dry} = 7.5 P_{PL}^{0.65} \tag{5}$$

where P_{PL} is the payload power in watts and M_{dry} is the spacecraft dry mass in kg. The same work also shows that the payload mass M_{PL} is approximately $M_{PL} = 0.27 * M_{dry}$. [18] Once payload and dry masses are known, the Unmanned Space Vehicle Cost Model (USCM8) is used to calculate the recurring and nonrecurring costs. The total cost for N satellites is $T = T_1 * N^{1+ln(0.95)/ln(2)}$, where T is the total satellite cost and T_1 is the nonrecurring cost of the first flight unit. [17] More accurate cost and mass models can be employed if more subsystem parameters are defined.

The determination of an optimal launch manifest for a given constellation can be modeled as a binary linear programming (BLP) problem, as discussed in the following section. Because solving the BLP problem for each candidate solution would be prohibitively computationally expensive, an approximate launch manifest is instead developed. Some set of launch vehicles and corresponding launch locations is provided by the user. Properties such as the approximate cost of a launch vehicle, the mass the vehicle can carry to a 185km circular orbit m_{185}, and the specific impulse I_{sp} of the vehicle are available from the manufacturer. For each unique orbit, the Δv required to move each launch vehicle from a 185km circular orbit at launch inclination or higher to the desired orbit is calculated. The orbital cost function is then used to approximate the mass capacity m_{lim} for the launch vehicles going to the desired orbit, $m_{lim} = m_{185}/(1.1 * e^{\Delta v/(g * I_{sp})} - 0.1)$ [17]. The number of launch vehicles of each type that would be needed for a single orbit is approximated by adding satellites from that orbit, in no particular order, to a launch vehicle until the mass limit is reached, then adding an additional vehicle of that type. For each orbit, the launch vehicle with the minimum total cost to launch all satellites is chosen. The sum of the launches for all orbits is used as the launch cost for the constellation. This method does not provide an optimal manifest, but does provide a feasible one.

E. Launch

Once the optimization is complete and a constellation determined, a final optimal launch manifest can then be determined. Launch manifestation for heterogeneous satellite constellations has been considered previously [19]. This work extends heterogeneous constellation manifestation to include the ability for satellites to maneuver to their final locations using a two burn minimum energy Lambert targeter. The launch manifest problem can be modeled as a BLP problem and solved using Matlab's mixed integer linear programming solver, which uses constraint relaxation and the branch-and-bound method. The following parameters are necessary for the calculation.

Decision variables:

- x_{ij}: 1 if satellite i is assigned to launch vehicle j, 0 otherwise
- ω_{jk}: 1 if launch vehicle j is assigned to orbit k, 0 otherwise
- y_{jl}: 1 if launch vehicle j is assigned to launch location l, 0 otherwise

Constants:

- m_{jkl}: maximum allowable mass for launch vehicle j to orbit k from launch location l
- m_i: dry mass of satellite i, kg
- f_{ik}: fuel required to move satellite i from orbit k to its final orbit, capped at the maximum fuel mass for the satellite, kg
- r_{ik}: 1 if satellite i can hold enough fuel to get from orbit k to its final orbit, 0 otherwise
- α_{jkl}: 1 if launch vehicle j can go from launch location l to orbit k, 0 otherwise
- c_{jl}: cost in dollars to launch vehicle j from location l. Currently, cost is independent of location.
- I: total number of satellites
- J: total number of launch vehicles, equal to the number of launch vehicle types times I
- K: total number of unique orbits
- L: total number of launch locations

1) Constraints: The following constraints must be satisfied by the launch manifest.

1) Each satellite is assigned to exactly one launch vehicle.

$$\sum_{j=1}^{J} x_{ij} = 1 \quad \forall i = 1, 2, \ldots, I \tag{6}$$

2) A launch vehicle is assigned to the same number of orbits as it is locations.

$$\sum_{l=1}^{L} y_{jl} = \sum_{k=1}^{K} \omega_{jk} \quad \forall j = 1, 2, \ldots, J \tag{7}$$

3) For a launch vehicle assigned to a specific launch location and orbit, the combined mass of all satellites assigned to the launch vehicle and the fuel required to get those satellites from the launch vehicle's orbit to their final orbit must be less than or equal to the mass capacity of launch vehicle going to the assigned orbit from the assigned launch location.

Define a "big M" parameter M such that if the launch vehicle is not assigned to either the orbit or the launch location, the constraint is satisfied automatically. Specifically, choose M equal to the sum of the maximum wet mass of all of the satellites.

$$\begin{aligned}&\sum_{i=1}^{I} (m_i + f_{ik}) x_{ij} \leq m_{jkl} + M(1 - \omega_{jk}) + M(1 - y_{jl}) \\ &\quad \forall j = 1, 2, \ldots, J \quad \forall k = 1, 2, \ldots, K \\ &\quad \forall l = 1, 2, \ldots, L\end{aligned} \tag{8}$$

4) If a launch vehicle is assigned to a specific launch location and orbit, launches from the specified launch location to the specified orbit using the specified launch vehicle must be allowed. Specifically, the launch vehicle must be available at the launch location and the inclination of the orbit must be in the allowable range for the launch location.

$$\begin{aligned}&\omega_{jk} - \alpha_{jkl} + y_{jl} - 1 \leq 0 \\ &\forall j = 1, 2, \ldots, J \quad \forall k = 1, 2, \ldots, K \quad \forall l = 1, 2, \ldots, L\end{aligned} \tag{9}$$

5) If a satellite is assigned to a launch vehicle, and the launch vehicle is assigned to an orbit, then the satellite must be capable of reaching its final orbit from the launch orbit, subject to the constraints on the maximum propellant for the satellite.

$$\begin{aligned}&\omega_{jk} - r_{ik} + x_{ij} - 1 \leq 0 \\ &\forall i = 1, 2, \ldots, I \quad \forall j = 1, 2, \ldots, J \quad \forall k = 1, 2, \ldots, K\end{aligned} \tag{10}$$

6) If at least one satellite is assigned to a launch vehicle, then that launch vehicle must be assigned to at least one orbit.

$$\sum_{i=1}^{I} x_{ij} - I \sum_{k=1}^{K} \omega_{jk} \leq 0 \quad \forall j = 1, 2, \ldots, J \tag{11}$$

7) A launch vehicle can be assigned to at most one orbit.

$$\sum_{k=1}^{K} \omega_{jk} \leq 1 \quad \forall j = 1, 2, \ldots, J \tag{12}$$

8) x_{ij} is binary.

$$x_{ij} \in \{0, 1\} \quad \forall i = 1, 2, \ldots, I \quad \forall j = 1, 2, \ldots, J \tag{13}$$

9) ω_{jk} is binary.

$$\omega_{jk} \in \{0, 1\} \quad \forall j = 1, 2, \ldots, J \quad \forall k = 1, 2, \ldots, K \tag{14}$$

10) y_{jl} is binary.

$$y_{jl} \in \{0, 1\} \quad \forall j = 1, 2, \ldots, J \quad \forall l = 1, 2, \ldots, L \tag{15}$$

2) Objective function: The optimal launch manifest will minimize the cost of the launches assigned to launch locations, which is equivalent to minimizing the cost of all assigned launches.

$$\min \sum_{j=1}^{J} \sum_{l=1}^{L} c_{jl} y_{jl} \tag{16}$$

F. Reconfiguration

A total constellation reconfiguration can be performed in the event of a change in objectives or a failure of one or more satellites. In these cases, the previously determined optimal solution minus any failed satellites acts as a starting point for the new constellation. The optimization then determines the maneuvers to be performed by each satellite, subject to fuel constraints. Reconfiguration extends the life of a constellation by allowing it to accomplish secondary missions or by salvaging a mission in the event of an unexpected occurrence.

IV. Information fusion applications

The research being done is applicable to information fusion because it enables the design of complex, multi-domain systems that can be used to gather data. The ability to intelligently determine sensor placement on various structures aids designers in reducing costs while still satisfy mission objectives that may be complicated and competing. The tool's ability to evaluate configurations of COTS components and to manifest satellites on rideshare opportunities will assist universities and other budget-limited groups in performing in space-based missions and acquiring meaningful data. Furthermore, the ability to design and launch large constellations of small satellites enables constellations that collaborate on-orbit to perform high performance computing tasks and reduce the data sent to the ground. Overall, the tool aids in the creation of large-scale systems gathering raw data that can be fused to provide meaningful results.

V. Conclusion

A new tool for the design and evaluation of heterogeneous spacecraft constellations was discussed. The software provides guidance on both orbit design and satellite design for heterogeneous constellations, an area in which few simulation resources are currently available. The extant modules were outlined and potential expansions indicated. This tool represents a powerful expansion in capability, particularly for the growing domain of large constellations of low cost satellites. Potential applications include the development of communications constellations for responsive space, fire detection systems, constellations performing on-board big data analysis and image processing, and constellations using on-board machine learning. It can also be used to fuse vehicles and sensors from multiple domains, determining the best combination of assets from a variety of options.

Acknowledgments

This material is based upon work supported by the National Science Foundation Graduate Research Fellowship Program under Grant No. DGE-1651272. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation. Support for this work was also provided by the Virginia Tech Institute for Critical Technology and Applied Science and the Ted and Karyn Hume Center for National Security and Technology.

References

[1] J. G. Walker, "Circular orbit patterns providing continuous whole earth coverage," Royal Aircraft Establishment, Tech. Rep. 70211, 1970.

[2] M. P. W. Daniele Mortari and C. Bruccoleri, "The flower constellations," *Journal of Astronautical Sciences*, vol. 52, no. 1, 2004.

[3] M. E. A. Jeremy J. Davis and D. Mortari, "The 3-d lattice theory of flower constellations," *Celestial Mechanics and Dynamical Astronomy*, vol. 116, no. 4, pp. 339–356, 2013.

[4] J. J. Davis, "Constellation reconfiguration: Tools and analysis," Ph.D. dissertation, Texas A&M University, 2010.

[5] D. J. Thomas, K. Tetreault, K. Mott, K. M. Nastasi, and J. T. B. et al, "Real-time on-board estimation and optimal control of autonomous micro-satellite proximity operations," in *Proceedings of the AIAA SciTech Forum*. AIAA, 2016.

[6] R. E. Thompson, "A methodology for the optimization of disaggregated space system conceptual designs," Ph.D. dissertation, Air Force Institute of Technology, 2015.

[7] R. E. Thompson, J. M. Colombi, J. Black, and B. J. Ayres, "Model-based conceptual design of disaggregated space system architectures," *Systems Engineering*, vol. 18, no. 6, 2015.

[8] ——, "Model-based conceptual design optimization methods: Disaggregated weather system follow-on," *Journal of Spacecraft and Rockets*, vol. 52, no. 4, pp. 1021–1037, 2015.

[9] Dynetics, "The gremlins program fact sheet," http://www.dynetics.com/_files/strike-systems/Dynetics%20Gremlins.pdf, 2017.

[10] DARPA, "Hydra," https://www.darpa.mil/program/hydra.

[11] K. E. Mott and J. T. Black, "Heterogeneous constellation design methodology applied to a mars-orbiting communications and positioning constellation," in *Proceedings of the AAS Astrodynamics Specialist Conference*. AAS, 2017.

[12] J. B. Kollat and P. M. Reed, "The value of online adaptive search: a performance comparison of nsgaii, ϵ-nsgaii and ϵmoea," in *International Conference on Evolutionary Multi-Criterion Optimization*. Springer, 2005.

[13] S. A. Kalyanmoy Deb, Amrit Pratap and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: Nsga-ii," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 2, 2002.

[14] C. Y. Lee and E. K. Antonsson, "Variable length genomes for evolutionary algorithms," in *Genetic and Evolutionary Computation Conference*, 2000.

[15] D. A. Vallado, *Fundamentals of Astrodynamics and Applications, 4th ed.* Microcosm Press, 2013, ch. General Perturbation Techniques, pp. 609–654.

[16] R. H. Merson, "The motion of a satellite in an axi-symmetric gravitation field," *Geophysical Journal of the Royal Astronomical Society*, vol. 4, no. 17, 1961.

[17] D. F. E. James R. Wertz and J. J. Puschell, Eds., *Space Mission Engineering: The New SMAD*. Microcosm Press, 2015, pp. 298–299, 316, 486.

[18] P. N. Springmann and O. L. de Weck, "Parametric scaling model for nongeosynchronous communications satellites," *Journal of Spacecraft and Rockets*, vol. 41, no. 3, 2004.

[19] J. M. Colombi, L. D. Buckle, J. T. Black, and S. G. Nurre, "Optimal launch manifesting for heterogeneous disaggregated satellite constellations," *Journal of Spacecraft and Rockets*, vol. 54, 2017.