Constellation Operations and Instrument Analysis for Earth Science Missions using TAT-C

ESTO Earth Science Technology Forum
Moffet Field, CA
June 11, 2019

Sreeja Nag¹ and Paul T. Grogan²
Jonathan Verville³, Daniel Selva⁴, Afreen Siddiqi⁵, Phil Dabney³

¹ Bay Area Environmental Research Institute
² Stevens Institute of Technology
³ Goddard Space Flight Center
⁴ Texas A&M University
⁵ Massachusetts Institute of Technology
Constellation Architecture Questions

- With smaller satellites, newer launches, commercial providers.... what does a constellation look like?
- When and where to add new sensors?
- Which instruments can be placed on which platforms?
- Where are synergies with existing platforms?
- Lack of existing information systems tools to study NOS architecture trades during conceptual design phase:

<table>
<thead>
<tr>
<th>Tool</th>
<th>License</th>
<th>Simulation</th>
<th>Automation</th>
<th>Trades?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Systems Tool Kit (STK)</td>
<td>Commercial</td>
<td>Full mission operations</td>
<td>STK connect commands (w/integration module)</td>
<td>Slow</td>
</tr>
<tr>
<td>FreeFlyer</td>
<td>Commercial</td>
<td>Full mission operations</td>
<td>Runtime API (w/license tier)</td>
<td>No</td>
</tr>
<tr>
<td>General Mission Analysis Tool (GMAT)</td>
<td>Open (Apache 2.0)</td>
<td>Orbital dynamics</td>
<td>GMAT script</td>
<td>No</td>
</tr>
<tr>
<td>Orekit</td>
<td>Open (Apache 2.0)</td>
<td>Low-level library</td>
<td>Java/Python API</td>
<td>No</td>
</tr>
</tbody>
</table>
TAT-C Vision and Unique Capabilities

The **Trade-space Analysis Tool for Constellations** using Machine Learning (TAT-C ML) will provide a framework to facilitate **Pre-Phase A** investigations of Distributed Spacecraft Missions (DSM) and optimize DSM designs to **a priori science goals**.

1. Science-first: leave architecture as a variable
2. Modular software components extensible by user community
3. Efficient pre-Phase A analysis made scalable with intelligent/learning technologies
4. Can be utilized with operations optimizers, planners and different concept of operations
Combinatorial Architecture Tradespace

• Architecture variables:
  – Number of satellites [1, 2, 4, 6, 8, 12]
  – Constellation geometry Walker Delta with [1, 2, 3, 4] planes
  – Orbital plane specifications
    • Altitude [400, 500, 600, 700] km
    • Inclination [30, 50]°
  – Satellite bus specifications [A, B, C]
  – Instrument specifications [A, B, C, D]
  – Ground network [A, B, C, D]

• Resulting architectural tradespace has around
  \(6 \times 4^4 \times 2 \times 3 \times 4 \times 4 = 18\) million alternatives

• Knowledge-driven algorithms to search design space

TAT-C Functions and Modules

Graphical User Interface (GUI)

Tradespace Search Executive (TSE)

Architecture

Orbit

Orbital Metrics

Instrument

Instrument Metrics

Launch Vehicle

Launch Metrics

Maintenance

Maintenance Metrics

Cost & Risk

Cost & Risk Metrics

Value

Value Metrics

Tradespace Validation

Knowledge Base
Both interfaces generate a JSON file that defines the constraints of the trade space

- Simple Panel – simplifies process of creating JSON file
- Advanced Panel – allows the user to edit every part of the JSON file

Comparing Global Metrics: Revisit Time
- Shows avg/min/max revisit time for each architecture
- Annotations show number of satellites per architecture and costing data

Exploring Global Metric Variation: Revisit Time
- Heatmap over Area of Interest
  - Red: high average revisit time
  - Blue: low average revisit time
- Annotations show lat/lon bounds of cursor
Application Case: Sustainable Land Imaging (SLI)

• Existing government programs
  - U.S. Landsat (7 and 8)
  - E.U. Sentinel-2 (A and B)

• New commercial systems
  - Planet Flock (100+, 4 kg): 1 (x28), 1b (x28), 1c (x11), 2e (x20), 2p (x12), 2k (x48), 3m (x4), 3p’ (x4), 3s (x3), 3k (x12), 4a (x20)
  - Planet RapidEye (5 x 150 kg)

• Plausible SLI Scenarios:
  1. Monolithic: Landsat 10 is a single satellite
  2. Distributed: Landsat 10 is a distributed spacecraft mission with multiple satellites
  3. Hybrid: Landsat 10 provides government cross-calibration for numerous commercial platforms
• TAT-C is key enabler of Bart Forman AIST16
• Provides orbital dynamics into Forman LIS OSSE machine learning model
• Also provides cost and risk analysis to prioritize spacecraft and instrument trade-offs
Constellation Types

Delta Homogeneous Walker
- Decisions = [h,inc,#sats,#pl,f]
- Fixed altitude and inclination
- Evenly spread RAAN and mean anomalies

Train
- Decisions = [h,LTAN,#sats,ΔLTAN]
- Fixed altitude and LTAN
- RAAN and mean anomalies calculated to obtain desired ΔLTAN

Delta Heterogeneous Walker
- Decisions = [h,inc,#sats,#pl,f]
- Allows for a mix of h, inc
- FF: Enumerate all possible planes in homogeneous Walker and take all possible combinations of planes
- GA: Enforce evenly spread RAAN and mean anomalies

Ad-hoc
- Decisions = [#sats]
- Current implementation: Take #sats random satellites from Planet Labs constellation

Precessing Type
- Decisions = [#sats, rocket]
- Multiple relight rockets drop off h-i combis
- Current implementation: Take #relights, ΔV from available rockets and use to compute h-i spread for satellite batches within #sats

S. Nag, S.P. Hughes, J.J. Le Moigne, "Navigating the Deployment and Downlink Tradespace for Earth Imaging Constellations", International Astronautical Congress, Adelaide, Australia, September 2017
Tradespace Search Strategies

Enumerating

Search strategy

- Full-factorial enumeration
  - Simple but rigid formulation
  - Exhaustive but expensive computationally

- Evolutionary algorithm
  - Flexible formulation and still relatively simple
  - Anytime algorithm but may need many function evaluations

- Knowledge-driven optimization
  - More complex
  - Uses knowledge (existing and new) to find better architectures faster


N. Hitomi and D. Selva, "Constellation optimization using an evolutionary algorithm with a variable-length chromosome", IEEE Aerospace Conference, Big Sky, MT 2018
Application Case: Applying TAT-C to SLI

- **Mission Concept:**
  - Government-directed
  - 90-day reference simulation period
  - Global imaging mission

- **Design Space:**
  - Existing Landsat 8/9 satellites (with OLI)
  - Existing Sentinel-2 A/B satellites (with MSI)
  - New constellation with 1-10 satellites:
    - Sun-synchronous with 1-2 orbital planes
    - 400-800 km altitude
  - Standard launch vehicle database
  - Satellite bus evolved from Landsat 9 (with OLI)
  - Standard Landsat international ground network

- **Analysis Settings:** standard outputs
Application Case: Applying TAT-C to SLI

- New constellation with 1 satellite:
  - Satellite bus evolved from Landsat 9 (with OLI)
  - Sun-synchronous with 1 orbital plane
  - 600 km altitude
- Standard Landsat international ground network
Orbital Analysis

Evaluation of an architecture by propagating the orbit and calculating its coverage over entire mission duration, e.g. suitable monolithic-Landsat-10 spacecraft working alongside NASA, ESA spacecrafts for EO SLI.

Visualization of an architecture, simulated in the NASA General Mission Analysis Toolkit (GMAT):
Orbital Analysis

- Numerical simulation in C++

- **Orbit propagation:**
  - Computationally light, utilizing the point-mass model of Earth with consideration of J2-perturbations.
  - Option to include effect of atmospheric drag.

- **Coverage (dependent on Instrument):**
  - Generation of a set of grid-points within regions of interest
  - Calculation of access times of the grid-points
  - Supported FOV geometries:
    - Cone
    - Rectangular
    - Custom shaped

Illustration of satellite orbit and coverage
Orbital Analysis

**INPUTS:**

```
"orbit": {
  "@type": "Orbit",
  "keplerian": {
    "semimajorAxis": 6978.14,
    "inclination": 97.787,
    "eccentricity": 0.0,
    "periapsisArgument": 0.0,
    "rightAscensionAscendingNode": 0.0,
    "trueAnomaly": 0.0
  }
  "orientation": {
    "eulerAngle1": 0.0,
    "eulerAngle2": 0.0,
    "eulerAngle3": 0.0,
    "eulerSeq1": 1,
    "eulerSeq2": 2,
    "eulerSeq3": 3
  }
  "fieldOfView": {
    "geometry": "conical",
    "coneAnglesVector": [7.5, 7.5, 7.5],
    "clockAnglesVector": [0.0092, 179.9908, 180.0092, -0.0092],
    "alongTrackFov": 0.0024,
    "crossTrackFov": 15.0
  }
}
```

Inputs are orbit specifications of satellite and sensor FOV, orientation specifications.

**OUTPUTS:**

Outputs are access-events captured during the entire mission at each grid-point.

Initial orbit specs from TLEs

Instrument FOV specifications

**ACCESS EVENT #1**
- Satellite: Landsat-8 OLI
- Access From [JDUT1]: 2457954.41
- Access Duration [s]: 14e-3
- Satellite position [km]: (38.9 N, 77.0 W)
- Satellite velocity [km/s]: -0.11, 7.49, -0.27

**ACCESS EVENT #2**
- Landsat-9 OLI ...

**ACCESS EVENT #3**
- Sentinel-1B MSI ...

**ACCESS EVENT #1324**
- Landsat-8 OLI ...
Instrument Analysis

- Quantifies the **relative** instrument performance of the architecture using quality of the data-products gathered
- Implemented as a lightweight **Python package** called ‘InstruPy’.
- Three types of instruments supported
- Real world space instruments are complex and unique. Modeling of sophisticated systems with “equivalent” parameters of preselected basic-simple model. Tradeoff of **computational efficiency vs fidelity**

**Figure:** The high-level function of the InstruPy package is shown in the figure.

**SAR image of Capitol at DC** (`σ_{NEZ0}<-30 dB`)  
credit: Sandia Labs

**Simulated SAR image of Capitol at DC** (`σ_{NEZ0}=-15 dB`)  
credit: Sandia Labs

Example of how image metrics relate to quality of image
Instrument Analysis

**Inputs**

```
{
  "@type": "Passive Optical Scanner",
  "name": "Sentinel-2A MSI Band2",
  "mass": 290,
  "volume": 1,
  "power": 266,
  "dataRate": 450,
  "snrThreshold": 154,
  "orientation": {
    "convention": "SIDE_LOOK",
    "sideLookAngle": 0
  },
  "fieldOfView": {"sensorGeometry": "RECTANGULAR",
    "alongTrackFieldOfView": 0.00072,
    "crossTrackFieldOfView": 20.6},
  "opticsSysEff": 0.75,
  "apertureDia": 150e-3,
  "Fnum": 4,
  "focalLength": 600e-3,
  "scanTechnique": "PUSHBROOM",
  "numberOfDetectorsRowsAlongTrack": 1,
  "numberOfDetectorsColsCrossTrack": 28763,
  "detectorWidth": 7.5e-6,
  "operatingWavelength": 492.4e-9,
  "bandwidth": 66e-9,
  "quantumEff": 0.85,
  "targetBlackBodyTemp": 290,
  "bitsPerPixel": 12,
  "numOfReadOutE": 40
}
```

**Outputs**

```
(38.9 N, 77.0 W)
ACCESS EVENT #1
Landsat-8 OLI
SNR: 889.73
Dynamic Range: 39601
Along-track Pixel resolution [m]: 29.85
Cross-track Pixel resolution [m]: NA
Coverage: YES

ACCESS EVENT #2
Landsat-9 OLI

ACCESS EVENT #2
Sentinel-1B MSI
```

Outputs are metrics of observations made at the respective grid-point.

Inputs are the instrument specifications and access-events computed during Orbit Analysis.

Three types of instruments supported:

1. **Passive-Optical Sensor**: Modeling of passive imagers working at visible/ thermal/ UV electromagnetic spectrum
   - **Pushboom**; *Examples*: TIRS and OLI on Landsat 8, MSI onboard Sentinel 2A, 2B
   - **Whiskbroom**; *Examples*: ETM+ onboard Landsat 7, MODIS onboard Terra, Aqua
   - **Matrix**; *Examples*: PlanetScope PS2 onboard DOVE

2. **Synthetic Aperture Radar**: Modeling of active radars working at microwave electromagnetic spectrum. Only Stripmap type.
   *Examples*: SeaSat, PALSAR onboard JAXA’s ALOS, DLR’s TERRASAR-X, CSA’s RADARSAT

3. **Basic Sensor**
   To represent sensors whose sophisticated engineering models are not supported in InstruPy
   *Examples*: GLAS (LIDAR) onboard ICESAT, CATS onboard ISS.
   Very basic metrics (range of observation, elevation angle) used for more sophisticated data metrics e.g. SNR, NETD
Application Case: Applying TAT-C to SLI

- **One architecture** with Landsat-8, 9, Sentinel 2A, 2B and *new-satellite* Landsat-10 was analyzed by the Orbit and Instrument modules.

- Examples of performance metrics evaluated by these modules are shown … among the many available metrics which quantify the performance of the architecture used by the Optimizer to compare against other architectures.

*Bad SNR, Fine Resolution, probably imaged by the Sentinel MSI, not the Landsat OLI*

*Since all satellites in Polar orbits, low (good) revisit time at Poles.*
Application Case: Augmenting TAT-C for SLI

- Example **test satellites** - Cubesat with CCD matrix imager (2 deg x 55 deg Field Of Regard) and capability to maneuver.
- Current **reference satellites** – Landsat 8, 9 with pushbroom OLI, Sentinel 2A, 2B with pushbroom MSI, Sentinel 3A, 3B with pushbroom+12.6 deg off-nadir OLCI
- An optimal calibration opportunity ~ **minimizes the differences in target site viewing geometries.**

Application Case: Augmenting TAT-C for SLI

*User Customized criteria for allowed opportunity:*

- Max time difference ⇔ radiometric stability
- Max view zenith difference, solar zenith difference
- Min image size, reference satellites allowed
- Can be different per calibration location

How to design the optimum constellation architecture for Transfer Radiometers? Likely non-SSO and global coverage of invariant sites.

=> Augmented TAT-C
Key Takeaways and Next Steps

• Executing the NOS demands new methods and tools for conceptual design:
  – Synthesize multiple observation platforms
  – Use science-relevant metrics to inform architecture decisions
  – Search a combinatorially-large design tradespace
  – Flexible/modular interfaces for application-specific models

• TAT-C supports NOS by aiding rapid architecture analysis for future missions
  – Pareto optimal architecture for complex value modeling
  – Inputs for higher fidelity operations modeling; e.g. planning for adaptable sensors, calibration opportunities

• Ongoing work on hosted platform for TAT-C analysis on AIST Managed Cloud Environment (AMCE)
Acknowledgements

• Supported by ESTO Advanced Information Systems Technology (AIST) Program 2016 Project:
  “Generalizing Distributed Missions Design Using the Trade-Space Analysis Tool for Constellations (TAT-C) and Machine Learning (ML)”

• Thanks to other team members:
  – Jacqueline Le Moigne (TAT-C PI 2015-2018)
  – Steve Hughes, Mike Stark, Wendy Shoan (GSFC, GMAT)
  – Vinay Ravindra (BAERI)
  – Joseph Gurganus, Christian Billie, Gabriel Apaza (GSFC)
  – Pau Garcia Buzzi and Prachi Dutta (TAMU)
  – Matt Sabatini (Stevens Institute of Tech)
  – Eric Magliarditi (MIT)
Backup Slides
Source Code Availability

Goal is to have TAT-C tool and source code publicly available as open source, building a community around its use and evolution

- Source code managed through AIST Managed Cloud Environment (AMCE)
- Release of software requires following NASA’s Software Release Authority (SRA)
- Paperwork is the draft stage of submission with NASA Tech Transfer Office/ Strategic Partnership Office
Agenda

• Constellation Design to aid New Observing Strategies
• TAT-C Vision and Unique Capabilities
• Application Case: Sustainable Land Imaging (SLI)
• Overview of Core Functions and Modules:
  – Knowledge Base Schemas
  – Visualizations/User Interface
  – Orbital and Instrument Analysis
  – Launch Analysis
• Tradespace Search Capabilities
Instrument Analysis: Framework

Assumptions:

• Real world space instruments are **complicated** and unique.

• Modeling of sophisticated systems with “equivalent” parameters of preselected basic-simple model.

• Tradeoff of **simplicity** vs fidelity

Limitations currently undergoing work:

• Simplified physical models. **Example:** atmospheric losses, surface reflectivity not considered while evaluating radiance.

• Only one “feature” of instrument considered in analysis. **Example:** The Landsat-8 OLI can image at 9 bands, but only 1 band can be considered at a time.

**Example:** Optics assembly in Landsat-8 OLI is **reflective** with 3 mirrors, while the TIRS is **refractive**. Impractical to start building custom models of all “possible” optics. Instead we simplify analysis by allowing the user to input an **equivalent telescope focal-length, aperture diameter, F#**.
ESTO’s New Observing Strategy (NOS)

• Flexible constellation of multiple observations:
  – Space, airborne, in-situ
  – Spatially and temporally correlated observations

• Autonomous control and data collection:
  – Scalable and selective
  – Automatic targeting

• Challenges:
  – Technical: security, autonomy, calibration, processing, complexity
  – Social: new concepts, culture change, complexity

CYGNSS (NASA/Univ. Michigan)
Case Study: Sustainable Land Imaging (SLI)

- **Landsat Program (NASA/USGS):**
  - 47-year continuous record of moderate-resolution (~30 m) global land imagery
  - Solar reflective and thermal infrared
  - Spacecraft in mid-morning sun-synchronous orbit, 16-day repeat cycle

- **Sustainable Land Imaging Program:**
  - 2019 SLI Architecture Study to consider beyond Landsat 9 (2026 and beyond)
  - Maintain consistency for calibration, coverage, spectral/spatial characteristics, data quality, and availability
Knowledge Base Schemas

- Standard serialized object templates for module interoperability
  - Human- and machine-readable documents
  - Linked to (more formal) semantic definitions

- Key-value dictionaries in JavaScript Object Notation (JSON):

```json
{
  "@type": "GroundStation",
  "name": "Gilmore Creek",
  "latitude": 64.9764,
  "longitude": -147.5208,
  "elevation": 306.418
}
```
**ε-MOEA: Evolutionary algorithm**

- **Idea:** Evolutionary algorithms mimic natural evolution
  - Specifically, use ε-MOEA
    - Multi-objective
    - Steady-state algorithm
    - Maintains an archive of best solutions found so far
- **Challenge:** Develop crossover and mutation operators for complex constellation types (e.g., heterogeneous)
- **Pros:** Simple, flexible
- **Cons:** May need many function evaluations to converge

---

**Diagram:**

1. **START**
2. Create and evaluate initial population
3. Check Termination Criteria
   - continue
   - terminate
4. Generate offspring
5. Evaluate offspring and insert into population
6. Update population and archive
7. Increment Iteration $t \leftarrow t + 1$
8. **END**

Baseline evolutionary algorithm
**Key params:** 7-day simulation, 57x20 deg FOV, coverage grid over tropical regions, 3U CubeSat

**Design space**

<table>
<thead>
<tr>
<th>Decision</th>
<th>Options</th>
</tr>
</thead>
<tbody>
<tr>
<td># satellites</td>
<td>[1,2,3,6,12]</td>
</tr>
<tr>
<td># planes</td>
<td>[1,2,3]</td>
</tr>
<tr>
<td>Altitude</td>
<td>[400,500,600,700,800]</td>
</tr>
<tr>
<td>Inclination</td>
<td>[30,51.6,90]</td>
</tr>
</tbody>
</table>

**Some good architectures**

<table>
<thead>
<tr>
<th>#id</th>
<th>#sats</th>
<th>#planes</th>
<th>h</th>
<th>inc</th>
<th>Cost [M$]</th>
<th>Mean revisit [h]</th>
<th>Max revisit [h]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Arch-1</td>
<td>12</td>
<td>1</td>
<td>80</td>
<td>90</td>
<td>51</td>
<td>2.8</td>
<td>11.7</td>
</tr>
<tr>
<td>Arch-2</td>
<td>12</td>
<td>1</td>
<td>80</td>
<td>30</td>
<td>51</td>
<td>1.2</td>
<td>18.4</td>
</tr>
</tbody>
</table>
Key params: 7-day simulation, 57x20 deg FOV, coverage grid over tropical regions, 3U CubeSat, 3 objectives (cost, avg revisit, max revisit)

Design space

<table>
<thead>
<tr>
<th>Decision</th>
<th>Options</th>
</tr>
</thead>
<tbody>
<tr>
<td># satellites</td>
<td>[1,2,3,4,6,8,10,12,16,18,24]</td>
</tr>
<tr>
<td># planes</td>
<td>[1,2,3,4]</td>
</tr>
<tr>
<td>Altitude</td>
<td>[400,500,600,700,800]</td>
</tr>
<tr>
<td>Inclination</td>
<td>[30,51.6,90]</td>
</tr>
</tbody>
</table>

Some good architectures

<table>
<thead>
<tr>
<th>#id</th>
<th>#sats</th>
<th>#planes</th>
<th>h</th>
<th>inc</th>
<th>Cost [M$]</th>
<th>Mean revisit [h]</th>
<th>Max revisit [h]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Arch-1</td>
<td>24</td>
<td>1</td>
<td>80</td>
<td>90</td>
<td>58</td>
<td>1.4</td>
<td>11.6</td>
</tr>
<tr>
<td>Arch-2</td>
<td>24</td>
<td>1</td>
<td>80</td>
<td>30</td>
<td>58</td>
<td>0.6</td>
<td>18.3</td>
</tr>
</tbody>
</table>
Knowledge-Driven Optimization with Adaptive Operator Selection

- **Idea**: Maintain a pool of operators (e.g., different types of crossover or ops. based on expert knowledge) and use ML to learn which one(s) work best for the problem at hand.

- **Challenge 1 - Credit assignment**: Measure performance of each operator over time.
  - $c_{i,t} = \text{credit received by } o_i \text{ at iteration } t$
  - Example: $c_{i,t} \propto f(x^p) - f(x_{o_i t}^p)$

- **Challenge 2 - Operator selection**: Assign solutions to operators proportionally to their quality ($q_{i,t} = \text{quality of operator } o_i \text{ at iteration } t$). For example:

  $$q_{i,t+1} = (1 - \alpha) \cdot q_{i,t} + \alpha \cdot c_{i,t}$$

  $$p_{i,t+1} = p_{\text{min}} + (1 - |O| \cdot p_{\text{min}}) \cdot \frac{q_{i,t+1}}{\sum_{j=1}^{\left|O\right|} q_{j,t+1}}$$

  $\alpha \in [0,1] = \text{adaptation rate}$

  $p_{\text{min}} = \text{minimum selection probability}$

**Pros**: Minimizes NFE to achieve search performance

**Cons**: More complex
**KDO\AOS: On-line discovery of new operators**

- **Idea:** New operators can be discovered online using feature extraction.

- **Approach:**
  - Define set of base features.
  - Use **association rule mining** to search space of conjunctions of features for target region C (top 25% architectures).
  - Use **mRMR** to select top few (e.g., 4) features.
  - Automatically transform top features into operators.
  - Add operators to pool.
  - Repeat every few iterations.

\[
\text{mRMR: } \Phi_i = \Phi_{i-1} \cup \left( \max_{F_i \in \Phi_{i-1}} \left[ I(F_i, C) - \frac{1}{i-1} \sum_{F_j \in \Phi_{i-1}} I(F_i, F_j) \right] \right)
\]

**U:** All possible designs

**C:** Designs within target region

**F:** Designs with the feature

\[
supp(F) \equiv \frac{|F|}{|U|}
\]

\[
\text{conf } (F \Rightarrow C) = \frac{supp(F \cap C)}{supp(F)} \quad \text{(consistency, specificity)}
\]

\[
\text{conf } (C \Rightarrow F) = \frac{supp(F \cap C)}{supp(C)} \quad \text{(coverage, generality)}
\]
Launch Module

- Primary Function: Determines the “optimal” launch manifest for a given constellation
- Optimal: Lowest launch cost to the customer
- Assumptions:
  - A different launch vehicle must be used for each orbital plane
  - Customer pays the entirety of the launch cost
- Database Construction:
  - Launch vehicle information was gathered through vehicle user guides and the FAA Annual Compendium of Commercial Space Transportation: 2018
Launch Module Formulation

• An Integer Program is used to assign satellites within a plane to a launch vehicle

• Constraints:
  – Each satellite can only be placed on a single launch vehicle
  – Payload mass cannot exceed payload capacity of the launch vehicle
  – Payload volume cannot exceed volume capacity of a launch vehicle (assuming a cylindrical fairing)
Launch Module SLI Example

- SLI-2 Architecture
  74
- Constellation Type:
  - Delta Homogenous
- # of Satellites: 10
- # of Planes: 2

<table>
<thead>
<tr>
<th></th>
<th>Plane 1</th>
<th>Plane 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>LV Used</td>
<td>Atlas V 551</td>
<td>Atlas V 551</td>
</tr>
<tr>
<td>Satellites on LV</td>
<td>0,1,2,3,4</td>
<td>5,6,7,8,9</td>
</tr>
<tr>
<td>LV Cost</td>
<td>$165 M</td>
<td>$165 M</td>
</tr>
<tr>
<td>Total Cost</td>
<td>$330 M</td>
<td>$330 M</td>
</tr>
</tbody>
</table>
Tradespace Search Executive (TSE)

TSR.json

TSR processing

Decisions

Ex: altitude, inclination, numsats...

Search strategy
- FF
- GA
- KDO

Modules
- Orbits
- Cost&Risk
- Instruments
- ...

Tradespace Search Request
- Constellation types
- Ground networks
- Satellites/Instruments
- ...

Enumerating

Arch.json

gbl.json

cost_risk.json
Tradespace Search Request (TSR)

- Json file containing all user-provided information to define
  - The mission concept
  - The design space
  - Some settings for the different analyses

- The TSE processes the TSR.json and **formalizes** the design space
  - through a set of decisions and options, constraints, objectives to optimize (when applicable).

- The TSR may not contain all the information required to uniquely define a design space
  - So the TSE must fill out missing information with reasonable values and **resolve any ambiguities**
Full Factorial Enumeration

- **Idea**: Given a set of allowed options for each decision, enumerate the Cartesian product of the sets of allowed options
  - Nested for loops
  - Can be visualized as a tree

- **Challenge**: Different types of constellations require slightly different enumeration approaches

- **Pros**: Simple and complete, allows full sensitivity analysis

- **Limitations**: It may be computationally too expensive to enumerate all architectures.

Foreach $x$ in decision 1
    Foreach $y$ in decision 2
        Foreach $z$ in decision 3
            ...
            Arch(n++) = create_arch($x, y, z, ...$)

Example: for Homogeneous Walker, decisions are $[h, inc, #sats, #pl, f]$; ground network, satellite/payload type