Coupled Low-Thrust Trajectory and Systems Optimization Via Multi-Objective Hybrid Optimal Control

Matthew A. Vavrina, a.i. solutions
Jacob A. Englander, NASA GSFC
Alexander R. Ghosh, University of Illinois

January 13, 2015
• Low-thrust trajectory & s/c hardware system are tightly coupled
  – Definition of traj. dependent on propulsion system, LV
  – SEP has variable power & dependent on array size

• Systems design problem
  – Different possible I_sp, power levels, number of thrusters, launch vehicle
  – Realistic engine, array models are discrete
  – Hybrid optimal control problem
  – Design space is multimodal, mixed parameter, often expansive

• Traditional approaches to sample trade space
  – Directly vary power & I_sp in optimization formulation
  – Simplified models, characteristic solutions
  – Parametric studies, grid searches

• Limitations
  – Trajectory opt. requires initial guess; locally optimal only
  – Only single-objective opt. strategies employed
  – Grid searches intractable
  – Limited fidelity w/out trading realistic hardware models
  – No full mapping of optimal trade space

BOL p_0: 30 kW, 3 thrusters

BOL p_0: 68 kW, 7 thrusters
Objective

Solve multi-objective, low-thrust systems optimization problem to fully map optimal systems trade spade

Method should be:

- Capable of **global** trajectory & systems parameter search
- Automated
- Free of user-defined initial guess
- Able to search broad design space
- Medium fidelity for preliminary design purposes
- Efficient
• Want to optimize any number of mission design metrics
  – e.g., payload mass, TOF, array size, ref. power, number of thrusters
  – Often coupled & competing
  – Fully map mission trade-offs between optimal solutions

• Optimize multiple objectives simultaneously
  – Entire set of optimal solutions
  – Goal: generate representation of Pareto front
  – Traditionally use repetitions of single objective technique
**Multi-objective Systems Optimization**

**Approach:** Solve coupled problem simultaneously w/ hybrid optimal control algorithm

- Multi-objective genetic algorithm (GA) as outer loop systems optimizer around direct-method inner loop trajectory optimizer
  - Non-dominated Sorting Genetic Algorithm II (NSGA-II) searches over systems parameters, defining trajectory problem
  - Monotonic basin hopping (MBH) + sequential quadratic programming (SQP) solves trajectory problem

---

**Initial generation**

- Population evolves via genetic operators

**Final generation**

- Population evolves via genetic operators
Genetic Algorithm

- Models Darwinian evolution
  - Mimic natural selection & reproduction
- Searches with population of designs
- Globally search design space
- No initial guess required
• Develops globally-optimal Pareto solutions using non-dominated sorting
  – Evolves population towards Pareto front

• Fitness assignment based on “nearness” to Pareto front
  – $x_1$ dominates $x_2$ if:
    \[ \forall p : f_p(x_1) \leq f_p(x_2) \quad p = 1, 2, \ldots, n_{obj} \]
    and
    \[ \exists p : f_p(x_1) < f_p(x_2) \quad p = 1, 2, \ldots, n_{obj} \]

• If neither design dominates other, they are non-dominant

• Non-dominated sorting:
  – Assign fitness based on design’s non-dominated front
  – Designs closer to Pareto front → better fitness & more mating opportunities
Need automated, robust method that does not require initial guess

- **Solution**: apply a global-local hybrid algorithm
  - Formulate problem based on Sims & Flanagan transcription
  - Monotonic basin hopping (MBH) drives global search
  - Gradient-based optimizer solves NLP (SNOPT used)

- Robust & efficient formulation

- Continuous thrust approximated
  - Trajectory discretized into segments
  - Impulsive $\Delta V$ at segment midpoint

- Efficient constraint handling
  - Gradients guide search
  - Robust & efficient formulation

- Proven approach in EMTG software (Evolutionary Mission Trajectory Generator)

From Sims and Flanagan
Monotonic Basin Hopping + SQP

- Stochastic, global search scheme
- No initial guess required
- Adept at multi-modal problems w/ clustered local minima
- Stochastic “hops” evaluated from base solution
  - Pareto distribution balances exploration & exploitation

![Diagram showing the process of Monotonic Basin Hopping with SQP. The diagram includes a graph with two axes: $f_1$ and $x_1$. The base design point is marked, followed by two hops: Hop 1 and Hop 2, leading to a point after gradient-based optimization.](image)
Multi-objective Systems Optimization Algorithm

• Synergistic relationship between outer & inner loops
• Generates globally optimal Pareto solutions for mission trade evaluation
• Any number of objectives viable
• Flexible to any unique mission constraints, trajectory constraints enforced in EMTG

Generate $P_1$, the initial parent population of trajectory problems defined by system design variables

Globally optimize each individual of $P_1$ via MBH+NLP algorithm to determine objective function values

Non-dominated sort $P_1$ to determine rank

Assign crowding distance to $P_1$

Selection: select individuals from $P_1$ via crowded tournament selection to form parent pool $S$

Crossover: generate initial offspring population, $Q_1$, from $S$ using uniform crossover

Mutate: Randomly alter individuals in $Q_1$

Globally optimize current $Q$ via MBH+NLP

Maximum generation?

No

Combine $P_1$ and $Q_1$ to form $R_t$

Non-dominated sort $R_t$ to generate $P_{t+1}$

Crowded tournament selection on $P_{t+1}$ to generate parent pool $S$

Uniform crossover of $S$ to create offspring population, $Q_{t+1}$

Mutate $Q_{t+1}$

Yes

Stop

Gen = 0

Gen = Gen + 1
Conclusions

• Hybrid optimal control algorithm developed for low-thrust spacecraft systems design
  – Outer loop: NSGA-II solves systems optimization problem
  – Inner loop: MBH+SQP solves trajectory optimization
• Generates globally optimal Pareto solutions for mission trade evaluation
• Automated
• Any number of objectives viable
• Ability to trade discrete, realistic hardware models
• General applicability to any interplanetary, low-thrust mission
  – Flexible to any unique mission constraints, trajectory constraints enforced in EMTG
• Can make large systems problems computationally tractable
Example Problem: ARRM

- Asteroid Robotic Retrieval Mission: return asteroid boulder or entire asteroid
  - Extensibility option is to return boulder from Deimos
  - Want to understand how return mass & TOF are affected by array size, # of thrusters
    → Multiple objectives: maximize return mass, minimize TOF, minimize BOL power, minimize # of thrusters (all coupled)

### Mission Parameters

<table>
<thead>
<tr>
<th>Description</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Launch window</td>
<td>1 year</td>
</tr>
<tr>
<td>Wait time at Bennu</td>
<td>[430, 700] days</td>
</tr>
<tr>
<td>Min. spacecraft mass with 2 thrusters</td>
<td>5991 kg</td>
</tr>
<tr>
<td>Additional dry mass per extra thruster</td>
<td>75 kg</td>
</tr>
<tr>
<td>Max. depart. mass if lunar gravity assist (C₃ ≤ 2.0 km²/s²)</td>
<td>11191 kg</td>
</tr>
<tr>
<td>Max. departure mass if direct launch (C₃ = 0.0 km²/s²)</td>
<td>10796 kg</td>
</tr>
<tr>
<td>Maximum C₃ if direct launch</td>
<td>6 km²/s²</td>
</tr>
<tr>
<td>Post-mission ΔV, Iₚ</td>
<td>75 m/s, 3000 s</td>
</tr>
<tr>
<td>Thruster duty cycle</td>
<td>90%</td>
</tr>
<tr>
<td>Solar array modeling</td>
<td>1/r²</td>
</tr>
<tr>
<td>Spacecraft bus power</td>
<td>2 kW</td>
</tr>
<tr>
<td>Propellant margin</td>
<td>6%</td>
</tr>
</tbody>
</table>

### System Design Variables

<table>
<thead>
<tr>
<th>Design Variable</th>
<th>Integer</th>
<th>Value</th>
<th>Resolution</th>
</tr>
</thead>
<tbody>
<tr>
<td>Launch option</td>
<td>[0, 1]</td>
<td>{Delta IV-H from LV curve, Delta IV-H with LGA}</td>
<td>-</td>
</tr>
<tr>
<td>Solar array size</td>
<td>[0, 20]</td>
<td>[30, 70] kW</td>
<td>2 kW</td>
</tr>
<tr>
<td>Launch window open epoch</td>
<td>[0, 4]</td>
<td>{2020, ..., 2029}</td>
<td>1 year</td>
</tr>
<tr>
<td>Flight time</td>
<td>[0, 26]</td>
<td>[700, 3300] days</td>
<td>100 days</td>
</tr>
<tr>
<td>Engine type</td>
<td>[0, 2]</td>
<td>{high-Isp, medium-thrust, high-thrust}</td>
<td>-</td>
</tr>
<tr>
<td>Number of engines</td>
<td>[0, 5]</td>
<td>[2, 7]</td>
<td>1</td>
</tr>
</tbody>
</table>
Pareto-Optimal Solutions
• Sharp increase in maximum return mass w/ increasing power
  – Increase in dry mass for increased power not accounted
Optimal Design Parameters

- Distinct grouping of engine modes based on TOF
  - Return mass plateaus for different engines
Backup
Example: Bennu Large-Mass Sample Return

- Asteroid Robotic Retrieval Mission (ARRM) Option B target

<table>
<thead>
<tr>
<th>Mission Objective</th>
<th>Return a large boulder from Bennu</th>
</tr>
</thead>
<tbody>
<tr>
<td>Launch Vehicle</td>
<td>Delta IV Heavy direct (C3 &lt; 6.0)</td>
</tr>
<tr>
<td></td>
<td>Delta IV Heavy with lunar flyby (C3 2.0)</td>
</tr>
<tr>
<td>Power System</td>
<td>Array power at 1 AU chosen by optimizer</td>
</tr>
<tr>
<td>Cell performance model</td>
<td>$1/r^2$</td>
</tr>
<tr>
<td>Spacecraft bus power</td>
<td>2.0 kW</td>
</tr>
<tr>
<td>Power margin</td>
<td>0%</td>
</tr>
<tr>
<td>Propulsion System</td>
<td>Thruster chosen by optimizer (high-Isp, medium thrust, or high-thrust versions of a large Hall thruster)</td>
</tr>
<tr>
<td>Number of thrusters</td>
<td>chosen by optimizer (2, 3, 4, 5, 6, 7); dry mass increases by 75 kg for each addtl thruster</td>
</tr>
<tr>
<td>Duty cycle</td>
<td>90%</td>
</tr>
<tr>
<td>Propellant tank</td>
<td>unconstrained</td>
</tr>
<tr>
<td>Mission Sequence</td>
<td>Direct travel to Bennu followed by direct return to C3 2.0 for lunar flyby capture</td>
</tr>
<tr>
<td>Inner-Loop Objective Function</td>
<td>Maximize sample return mass</td>
</tr>
<tr>
<td>Outer-Loop Objective Functions</td>
<td>Sample return mass</td>
</tr>
<tr>
<td></td>
<td>Solar array size</td>
</tr>
<tr>
<td></td>
<td>Number of thrusters</td>
</tr>
<tr>
<td></td>
<td>Flight time</td>
</tr>
</tbody>
</table>
## Bennu Sample Return: Outer-Loop Menu

<table>
<thead>
<tr>
<th>Power Supply at 1 AU</th>
<th>Launch Year</th>
<th>Flight Time Upper Bound</th>
<th>Thruster Type</th>
<th>Number of Thrusters</th>
</tr>
</thead>
<tbody>
<tr>
<td>Code</td>
<td>Array Output</td>
<td>Code</td>
<td>Year</td>
<td>Code</td>
</tr>
<tr>
<td>0</td>
<td>30</td>
<td>0</td>
<td>2019</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>32</td>
<td>1</td>
<td>2020</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>34</td>
<td>2</td>
<td>2021</td>
<td>2</td>
</tr>
<tr>
<td>3</td>
<td>36</td>
<td>3</td>
<td>2022</td>
<td>3</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>4</td>
<td>2023</td>
<td>4</td>
</tr>
<tr>
<td>20</td>
<td>70</td>
<td>5</td>
<td>2023</td>
<td>5</td>
</tr>
</tbody>
</table>

### Earth Departure Type

<table>
<thead>
<tr>
<th>Code</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Delta IV-H direct</td>
</tr>
<tr>
<td>1</td>
<td>Delta IV-H w/ LGA</td>
</tr>
</tbody>
</table>

102,060 possible combinations
Bennu Sample Return: Final Generation Trade Space
Bennu Sample Return: Evolution of Population
Bennu Sample Return:
Objective Space

Return mass vs. array size

Final journey mass increment (for maximizing sample ret.)
BOL power at 1 AU (kW)

Flight time (years)

Number of thrusters

Return mass vs. array size

Final journey mass increment (for maximizing sample ret.)
BOL power at 1 AU (kW)
Bennu Sample Return: Optimal Design Variables

**Departure Type**
- DeltaIV-H direct
- DeltaIV-H LGA

**Engine Type**
- High-Isp Engine
- High-Thrust Engine
- Medium-Thrust Engine

![Graph showing Boulder Return Mass vs Flight time for Departure Type](image1)

![Graph showing Boulder Return Mass vs Flight time for Engine Type](image2)
Bennu Sample Return: Two Trajectories

A 8-year mission with a 58 kW solar array returns a 20 ton boulder

A 3.3-year mission with a 70 kW solar array returns a 2.2 ton boulder