International Workshop on Planning & Scheduling for Space, 2025
(Presentation Only)

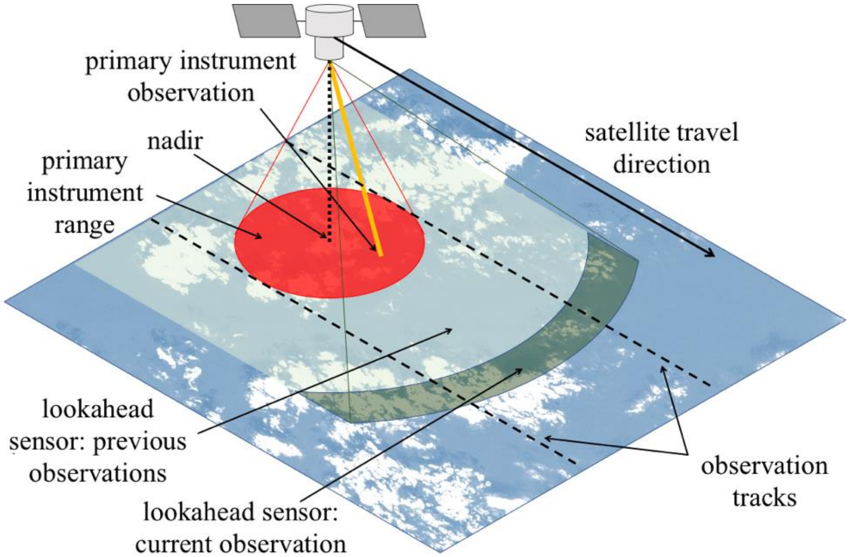
Search Applications for Integrated Planning and Execution of Satellite Observations using Dynamic Targeting

Akseli Kangaslahti, **Itai Zilberstein**, Alberto Candela, Steve Chien

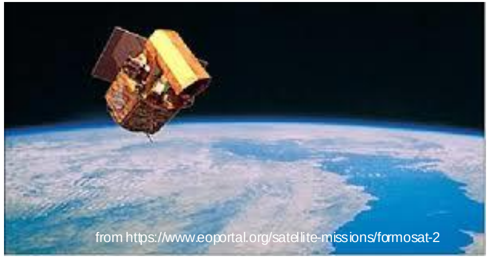
Jet Propulsion Laboratory
California Institute of Technology

Dynamic Targeting Concept

- Dynamic Targeting (DT): use look-ahead sensor to optimize plan for agile primary sensor
  - Identify high utility targets to observe
    - e.g. storm features
  - Identify low utility targets to avoid
    - e.g. clouds

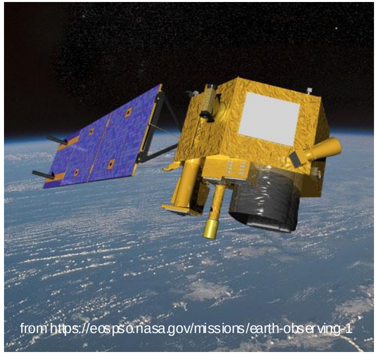


Related Work



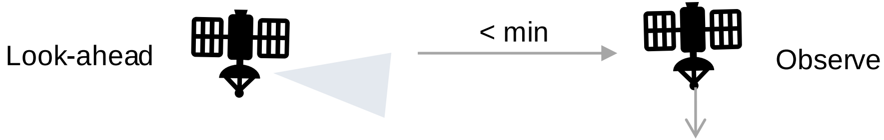
- Operational:
  - FORMOSA-2: examine weather (Liao and Yang 2005)
  - TANSO-FTS-2: avoids clouds (Suto et al. 2021)
  - EO-1: modified schedule (Chien et al. 2005)

- Non-operational:
  - Greedy algorithms (Chien and Troesch 2015)
  - Storm hunting, cloud avoidance (Swope et al. 2021; Candela et al. 2022; Candela, Swope, and Chien 2022, 2023)
  - **Tree-search, dynamic utility, slewing (Kangaslahti et al. 2024)**

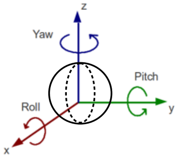


Dynamic Targeting Challenges

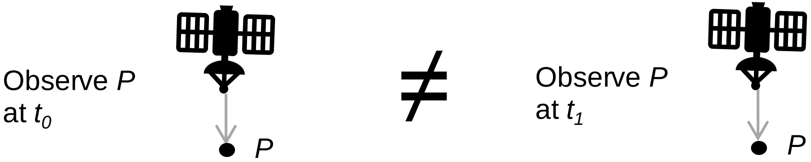
- *Real-time planning*



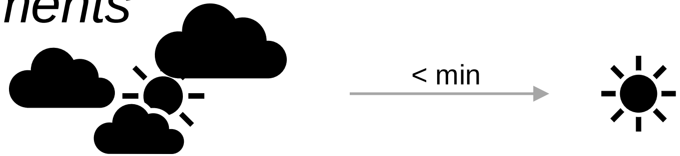
- *Real-world constraints*



- *Complex utility*



- *Dynamic environments*

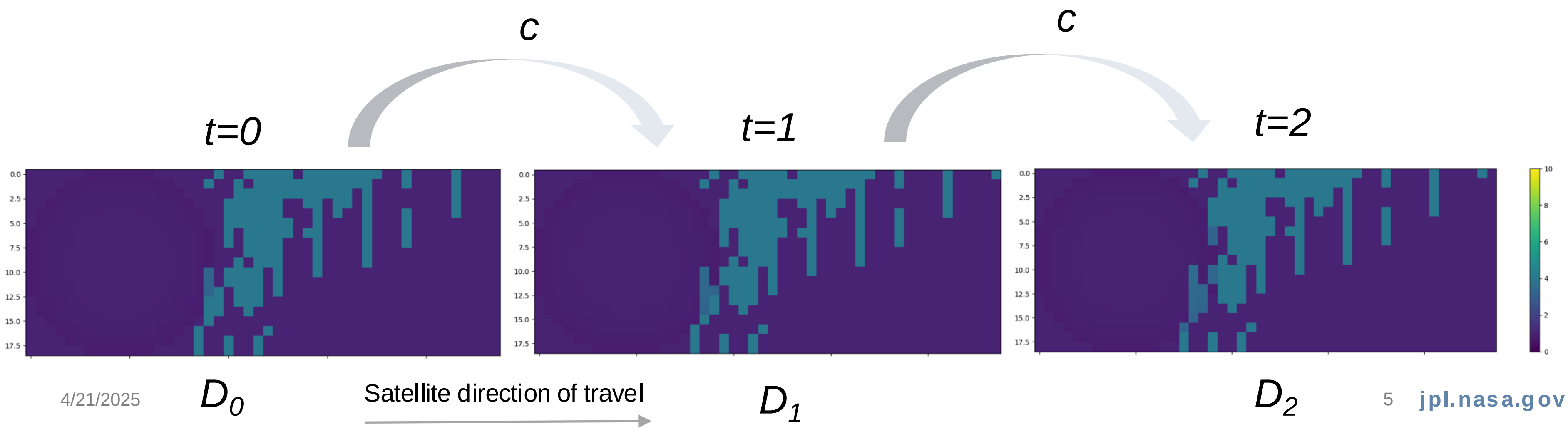


- Planning vs. execution time
- Power, primary sensor slew
- Time-dependent, action-dependent
- Assume static environment

Model Definition

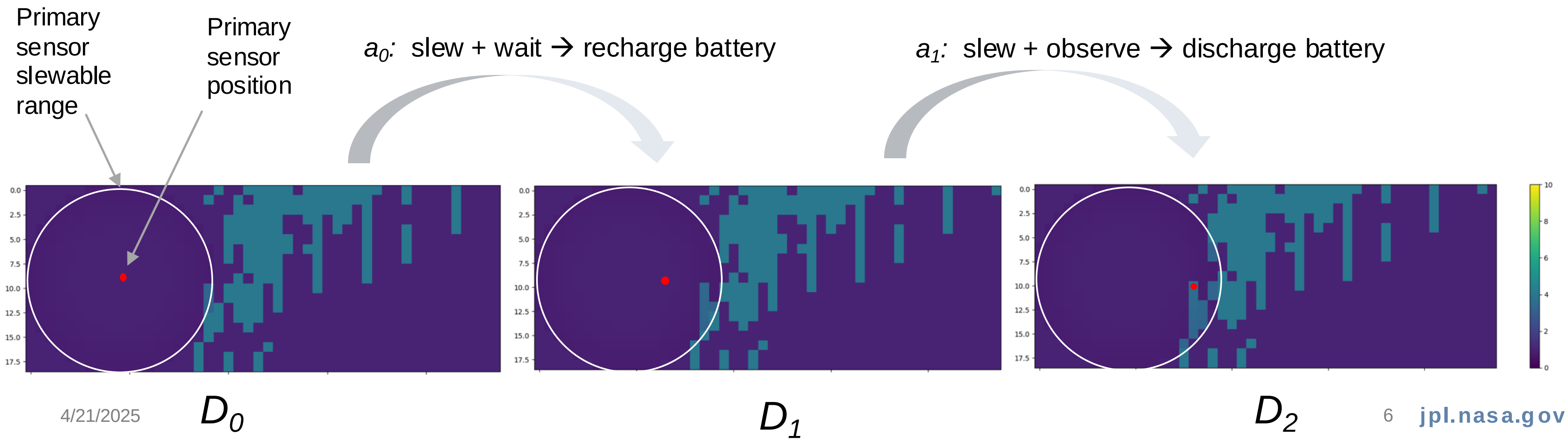
Discretized Dynamic Targeting

- *H:* Planning horizon composed of *n* cycles
- *c:* Time between cycles
- *D:* Set of look-ahead views



Model Definition

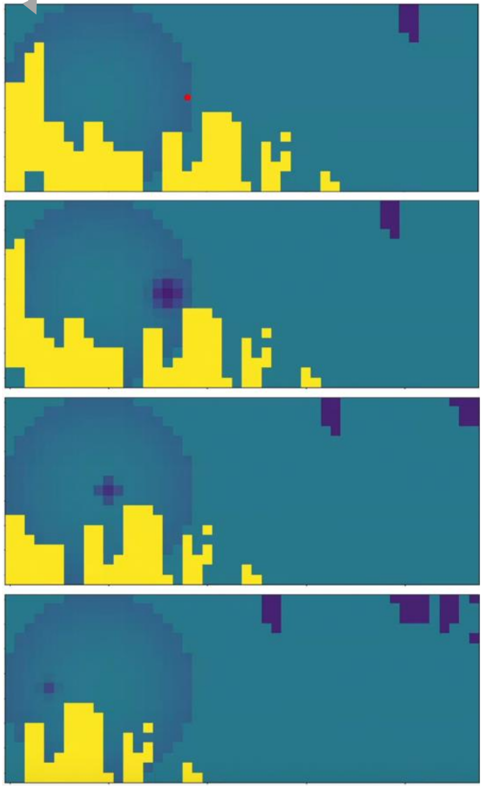
- *C:* constraints
  - Power: can only observe with sufficient charge
  - Slew: rest-to-rest time w/ acceleration & max velocity (slew can span multiple cycles)
- *A:* actions
  - Observe: discharge battery (optional slew)
  - Wait: recharge battery (optional slew)



Model Definition

- *U:* utility function
  - Base utility 4^L where $L \in \{0, 1, 2\}$ is pixel class
  - Linear off-nadir penalty relative to distance
  - After observation, apply Gaussian kernel and linearly recover utility over time
- Goal:
  - Construct a plan $\pi = (a_0, \ldots, a_{n-1})$ where $a_i \in A$ such that C is satisfied and U is maximized
  - One action per cycle

Ring due to off-nadir penalty
Brighter colors = higher utility



t=0
Observation

t=1
Utility of nearby points decrease

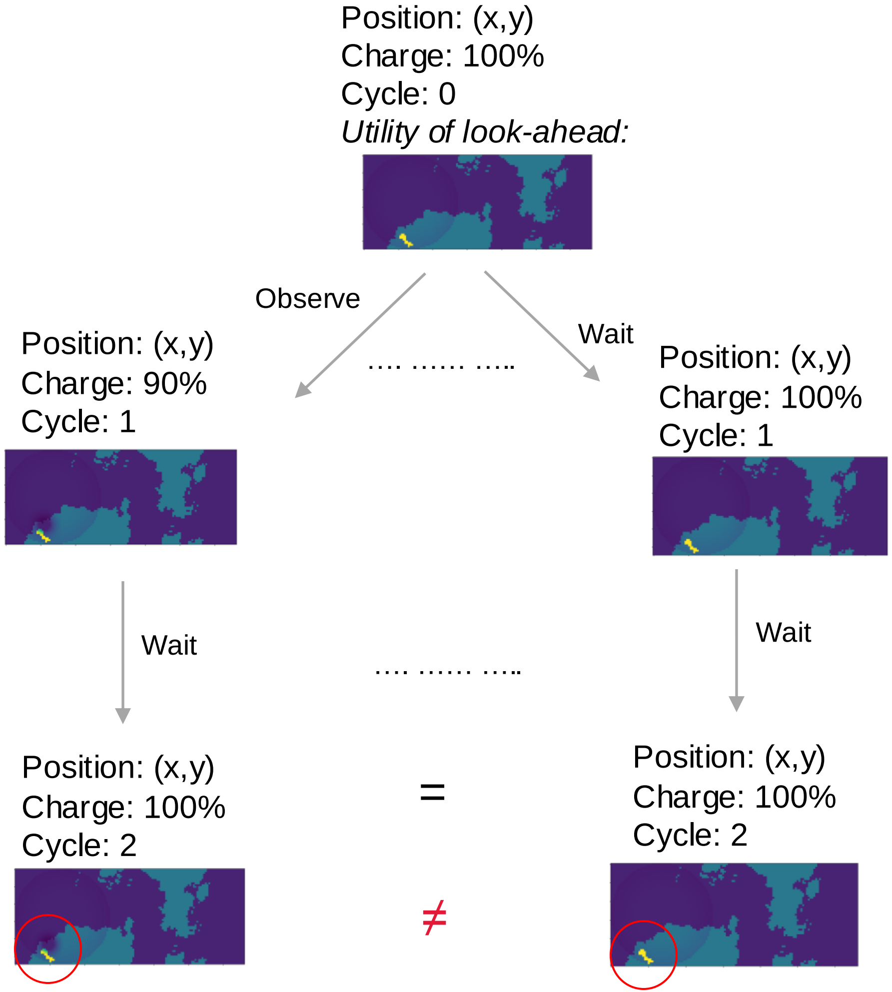
t=2

Utility recovers over time

t=3

State Space

- State:
  - Sensor position
  - State of charge
  - Look-ahead view / time / cycle
  - *Utility function*
- Search tree:
  - States are nodes
  - Actions are edges
  - Each depth corresponds to a cycle
- Exhaustive search: $O((m^2)^d)$
  - *d:* search depth (number of cycles)
  - *m:* diameter of reachable range of sensor



Algorithms

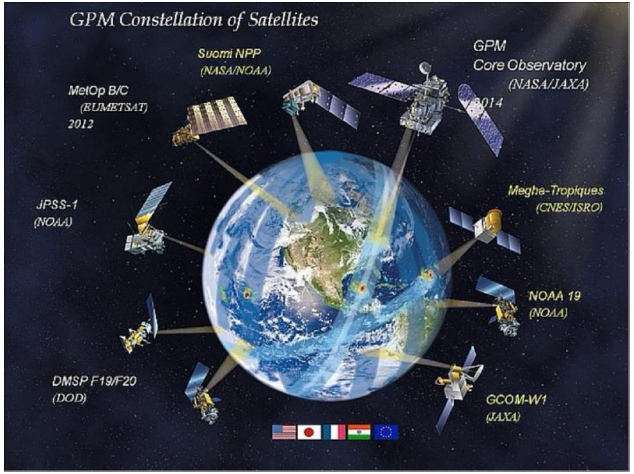
- Nadir Only (NO): observe nadir when sufficient power
- Greedy (G): locally optimal action
- Beam Search (BS): *k locally best* actions
- Partitioned Depth-First Search (PDFS): best action in *k* spatial partitions
- Monte-Carlo Tree Search (MCTS): UCB
- A*: *h* = Upper bound (admissible)
- Upper Bound (UB): infinite slewing capability and non-decreasing utility

- Re-planning: each cycle, prune search tree and continue planning with new look-ahead

- Theoretic Results:
  - PDFS, BS, A*: $O(m^2k^d)$
  - Locally optimal action in searched space: BS, PDFS, A* ≥ G

Experiments

- Datasets:
  - GPM: Storm hunting
  - MODIS: Cloud avoidance



from https://www.eoportal.org/satellite-missions/gpm

Parameter	Value	Description
Primary Sensor Range	15^o	Max off-nadir angle for primary sensor, from SMICES (Swope et al. 2021)
Look-ahead Sensor Range	45^o	Off-nadir angle for look-ahead sensor, from SMICES (Swope et al. 2021)
Power Recharge Rate	1%/ cycle	Rate at which battery power increases
Power Discharge Rate	1%/ obs	Rate at which battery power decreases during cycles when an observation is made
Cycle Time	3s	Total time allocated for each cycle
Gaussian Kernel Size	75 km × 75 km	Size of Gaussian kernel used in utility models
Gaussian Kernel Standard Deviation	15 km	Standard deviation of Gaussian kernel used in utility models
Angular Acceleration	$1.08°/s^2$	Angular acceleration of slewing along the pitch and roll axes. The value we use is realistic for small satellites
Maximum Angular Velocity	5.40°/s	The maximum angular velocity of slewing along the pitch and roll axes. The value we use is realistic for small satellites
Processing Time	0.1 s	Amount of time allocated for observation acquisition and processing
Minimum Utility Multiplier	0.25	The minimum utility multiplier in our utility model
Maximum Off-Nadir Penalty	20%	Penalty for an observation as far off-nadir as possible

Planning vs. Execution

- Compute resources onboard are scarce
- Planning action for next cycle cannot take longer than *c*
- Planning time subtracted from available slew time
- Tune search parameters (*d, k*) to trade-off search vs. execution

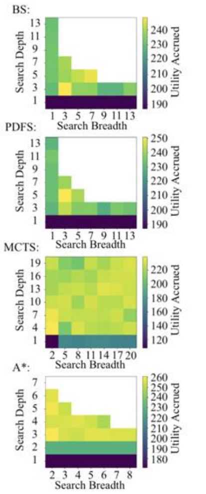


Figure 4: GPM data set parameter selection results for beam search (top), partitioned depth-first search (top middle), Monte Carlo tree search (bottom middle), and A* search (bottom).

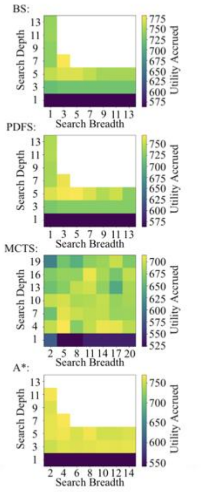


Figure 5: MODIS data set parameter selection results for beam search (top), partitioned depth-first search (top middle), Monte Carlo tree search (bottom middle), and A* search (bottom).

Table 2: Optimal Parameters Found during Parameter Selection

	GPM Data set	MODIS Data set
BS	Depth = 3, Width = 3	Depth = 7, Width = 3
PDFS	Depth = 5, Width = 3	Depth = 5, Width = 5
MCTS	Depth = 4, Width = 2	Depth = 4, Width = 5
A*	Depth = 3, Width = 2	Depth = 7, Width = 4

Table 3: Maximum Single Cycle Run-Time during First Parameter Sweep for Optimal Parameters (ms)

	GPM Data set	MODIS Data set
BS	40.7457	475.0683
PDFS	172.4098	484.0328
MCTS	100.7869	30.8206
A*	36.8192	441.7007

Results

Table: Total Accrued Utility

	GPM Data set	MODIS Data set
NO	16098	34713
G	19901	37521
BS	23960	43555
PDFS	24614	43238
MCTS	22488	41273
A*	23992	43070
UB	37953	71038

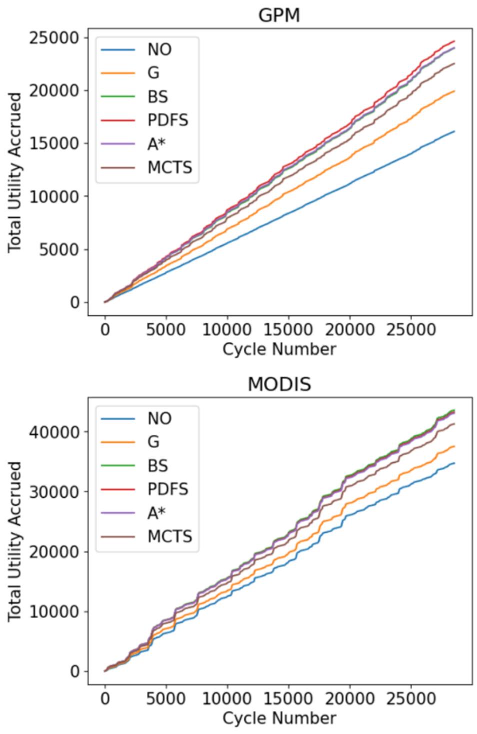


Figure 6: Total utility accrued vs. cycle number for all planning algorithms in GPM final testing (top) and MODIS final testing (bottom).

Conclusion

- Tree-based search algorithms outperform previous work
- Trading off planning and execution time important to DT
    - Slewing + observing dominates execution time
- Future work:
    - Anytime algorithms
    - Continuous search space
    - Dynamic environments & uncertainty
    - Execute DT onboard a satellite (coming soon!)
- Apply DT to other domains

jpl.nasa.gov

Portions of this research were carried out by the Jet Propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration (80NM0018D0004). This work was supported by the Earth Science and Technology Office (ESTO), NASA.