Scheduling for Rapid Response Imaging using Agile, Small Satellite Constellations

Sreeja Nag and Alan S. Li

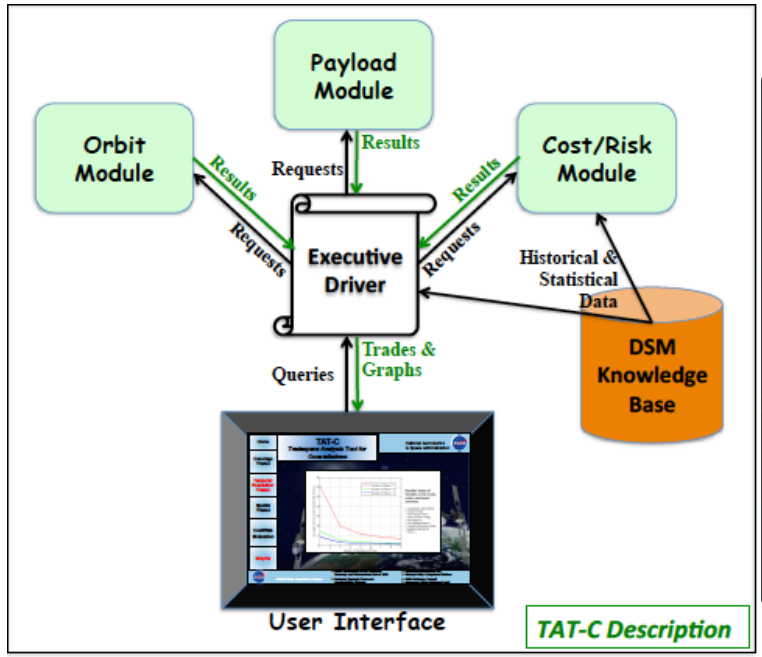
NASA Ames Research Center / Bay Area Environmental Research Institute

April 25, 2017

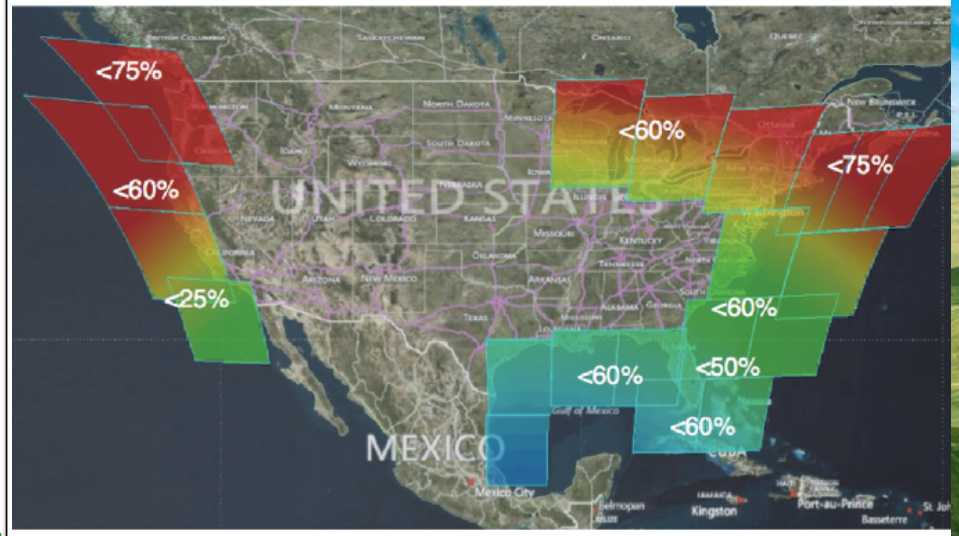
Bay Area Environmental Research Institute

Premise

Lots of research on constellation design, scheduling ops for single spacecraft, downlinks for constellations, UAV path planning and industry advances in ADCS… however little on combining all of them for responsive remote sensing



Tradespace Analysis Tool for Constellations (TAT-C)
NASA GSFC



GEO-CAPE scheduling based on spatial resolution, instrument specs and cloud cover (MILP, CP)
NASA ARC

NASA Unmanned Aerial Traffic Management Project
NASA ARC

Bay Area Environmental Research Institute

Problem Summary

Given a global set of images, a fixed constellation of satellites with agile ADCS, MOI/ADCS specs and coverage constraints, what is the fastest way to cover those images?

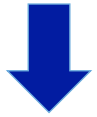
- Need a linear-time algorithm, generalizable for any constellation and targets
- Using Landsat as case study w/ a 14 day revisit. Daily revisit needs ~15 satellites or 4 satellites with triple the FOV.
- Instead assuming a 20 kg satellite platform to try the option of agile pointing
- Landsat takes ~24 s to transverse over its FOV of ~185 km. It snaps 236 times a second and integrates pushbroom images over 30 s (to allow for redundancy). We assume snapshot imagers and <1s integration
- The images, constellation/satellite number, specs and constraints (e.g. clouds, ground station outage) are assumed modular for generality

Bay Area Environmental Research Institute

Breaking Down the Problem

Orbital Mechanics => Access Times for Satellite, given discrete pointing options

Attitude Analysis => Extended Kalman filter + PID control => Switching Times for Satellite, given discrete pointing options



Optimization over satellites, pointing options, time and required images => Dynamic Programming

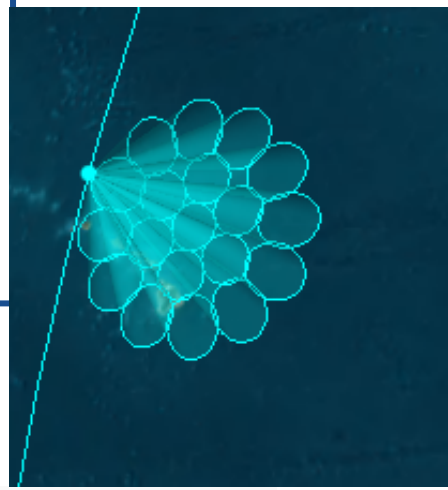
- Use MATLAB, STK, MS Connect to simulate ***orbital mechanics*** and compute access reports: For every satellite, every pointing option and every image, when to when is it visible
- Use MATLAB Simulink to compute pointing switch time
- Use an optimization method (DP, MILP) to find the best schedule of pointing per satellite, that stably views (for upto one second) the maximum number of given images in given time

Bay Area Environmental Research Institute

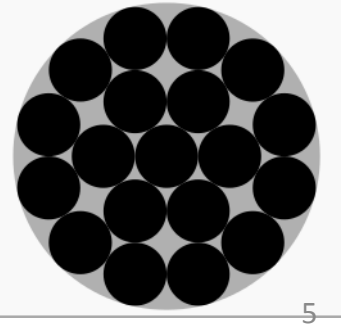
Breaking Down the Problem

- 16986 Landsat images imported as lat, lon, alt
- Limit simulation to one day at time resolution of 1 sec
- FOV = 15 deg, 710 km 98.2 deg orbit, max slew < 30 deg

```
PointNumber:          0
Lat:                  -1.42862442392132e+000
Lon:                  2.53334889992827e-002
Alt:                  0.00000000000000e+000
NumberOfAccesses:     24
17     2.16472247288690e+004    2.16682342027050e+004
16     2.16803324637031e+004    2.17019188513764e+004
19     2.75134863884224e+004    2.75352535285753e+004
18     2.75312718745623e+004    2.75551035673686e+004
7     2.75540277660315e+004    2.75832837351813e+004
6     2.75839494706595e+004    2.76132749995163e+004
15     2.76124735607921e+004    2.76351519882569e+004
14     2.76310365887511e+004    2.76549902722753e+004
19     3.34074109272497e+004    3.34350444375388e+004
2     3.34335923377278e+004    3.34633601183685e+004
1     3.34644770731449e+004    3.34917946612874e+004
5     3.34928682064443e+004    3.35227203785727e+004
14     3.35214143114703e+004    3.35484367052491e+004
19     3.92990388572563e+004    3.93214955190352e+004
```



- All possible pointing options discretized to 19 options based on the 2D circle-packing-in-circle solution

19	$1+\sqrt{2}+\sqrt{6}\approx 4.863...$	0.8034...	Proved optimal by Fodor in 1999.[8]	

https://en.wikipedia.org/wiki/Circle_packing_in_a_circle

Bay Area Environmental Research Institute

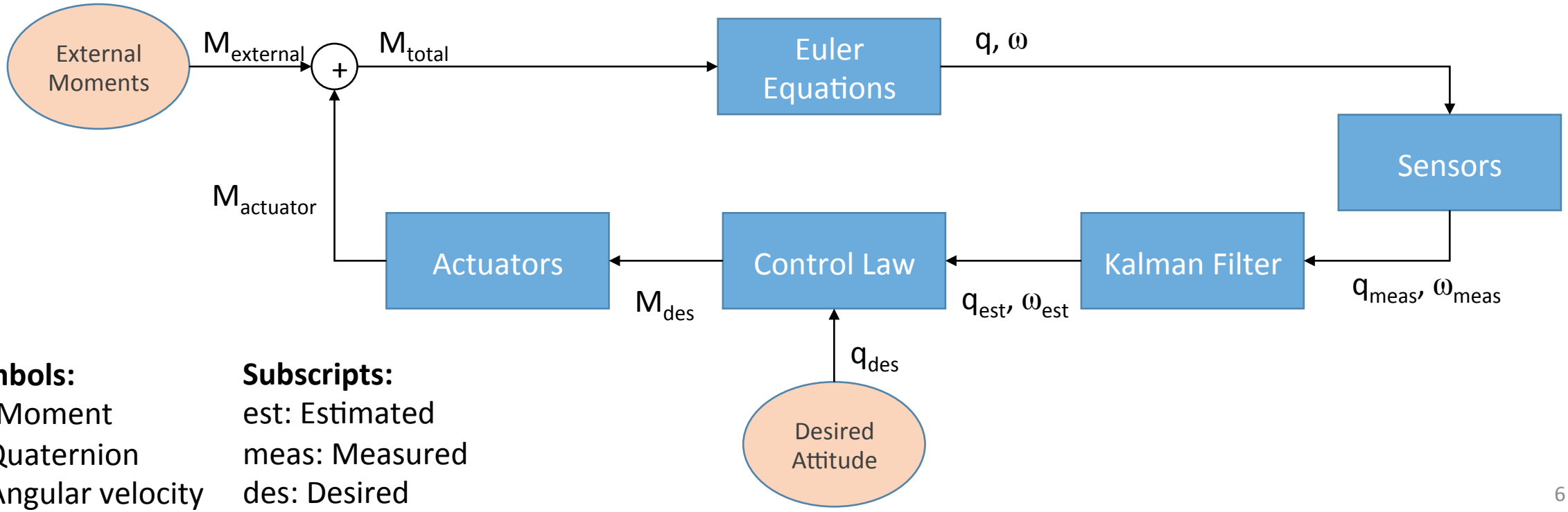
Attitude Control

Assumptions:
- No external moments
- Satellite inertial properties evenly distributed as a cube
- Sensor error: 0.1° in pointing, 0.01° error in rate gyro
- Reaches goal when pointing error < 0.2°

Specifications:
Mass = 20 kg
Max. moment = 0.025 Nm
Max. momentum storage = 0.5 Nms

Implementation:
- Actuators: 4 Reaction wheels
- Sensors: Sun sensor, magnetometer, rate gyro (with bias)
- Bang-bang control, with PD control when approaching desired attitude

Simplified Block Diagram



Symbols:
M: Moment
q: Quaternion
ω: Angular velocity

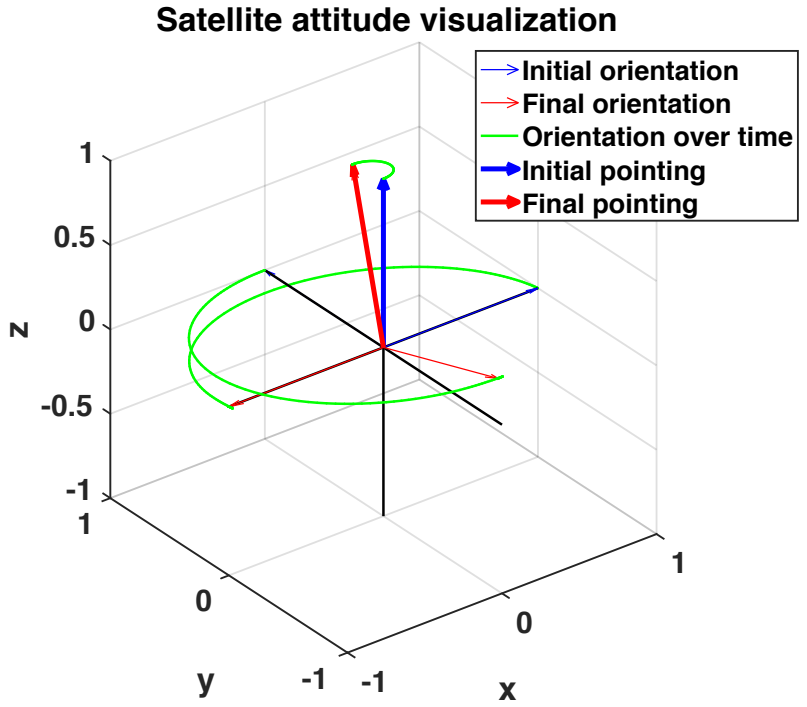
Subscripts:
est: Estimated
meas: Measured
des: Desired

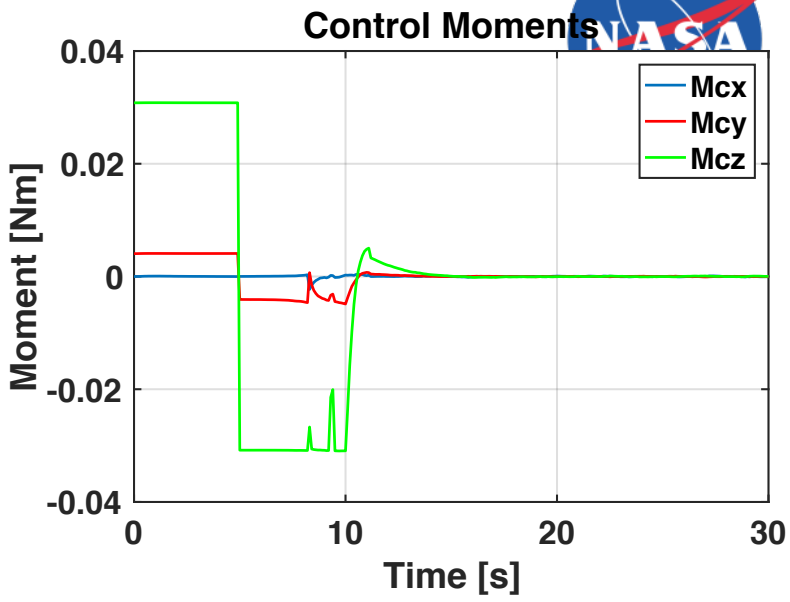
Bay Area Environmental Research Institute

Attitude Control

Full pointing mode:

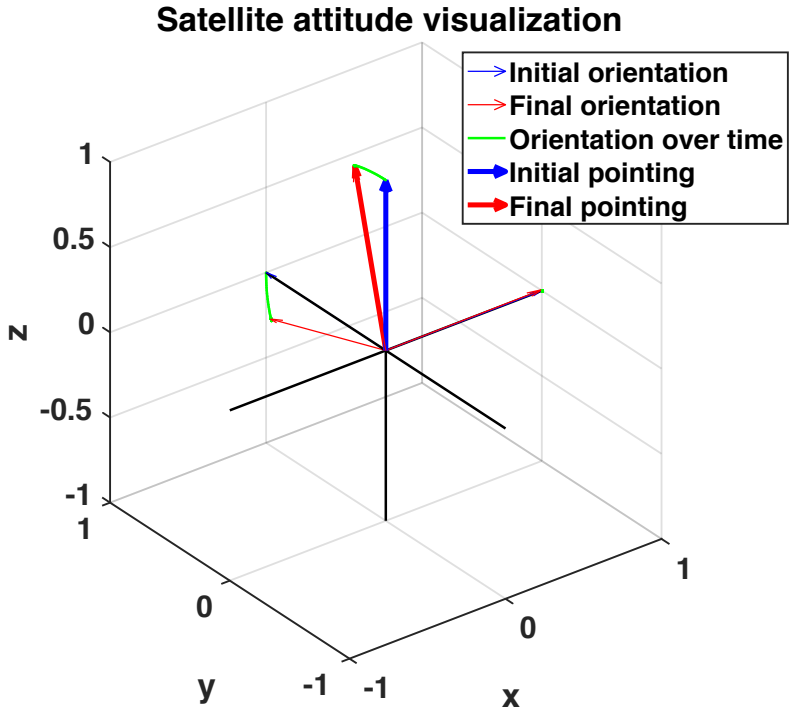
- Align all axes to desired
- Useful for aligning solar panels
- Requires more energy (E = 38.2 J)
- Slower in general

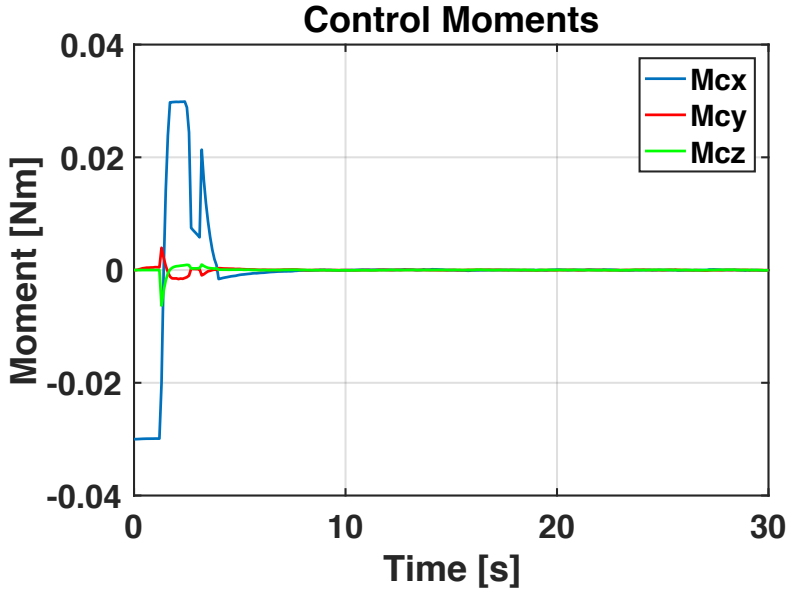




Fast pointing mode:

- Align ONLY pointing axes
- Useful for only aligning pointing direction
- Requires less energy (E = 2.8 J)
- Faster in general





Bay Area Environmental Research Institute

Breaking Down the Problem

- Access Matrix (includes constraints, based on the ***orbital mechanics*** solution)

Pointing Options (1-19) per satellite ->

<= Time Steps over simulation time (1-86400)

	1	2	3	4	5	6	7	8	9	10	11	12
378	10180	10188	10189	NaN	NaN	10173	10179	11310	10203	10191	NaN	NaN
379	10180	10188	10189	NaN	NaN	10173	10179	11310	10203	10191	NaN	NaN
380	10180	10188	10189	NaN	NaN	10173	10179	11310	10203	10191	NaN	NaN
381	10180	10188	10189	NaN	NaN	10173	10179	11310	10203	10191	NaN	NaN
382	10180	10188	10189	NaN	NaN	10173	10179	11310	10203	10191	NaN	NaN
383	10180	10188	10189	NaN	NaN	10173	10187	11310	10203	10191	NaN	NaN
384	10180	10188	10189	NaN	NaN	10173	10187	11310	10203	10191	NaN	NaN
385	10180	10188	10189	NaN	NaN	10173	10187	11310	10203	10191	NaN	NaN
386	10180	10188	10189	NaN	NaN	10173	10187	11310	10203	10191	NaN	NaN
387	10180	10188	10189	NaN	NaN	10172	10187	11310	10202	10191	NaN	NaN
388	10180	10188	10189	NaN	NaN	NaN	10187	11310	10202	10191	NaN	NaN
389	10180	10188	10189	NaN	NaN	10179	10187	11310	10202	10191	NaN	NaN
390	10180	10188	10189	NaN	NaN	10179	10187	11310	11311	10191	NaN	NaN
391	10180	10188	10189	NaN	NaN	10179	10187	11310	11311	10190	NaN	NaN
392	10180	10201	10189	NaN	NaN	10179	10187	11310	11311	10203	NaN	NaN
393	10188	10201	10189	NaN	NaN	10179	10187	11310	11311	10203	NaN	NaN
394	10188	10201	10189	NaN	NaN	10179	10187	11310	11311	10203	NaN	NaN
395	10188	10201	10189	NaN	NaN	10179	10187	11310	11311	10203	NaN	NaN
396	10188	10201	10189	NaN	NaN	10179	10187	11310	11311	10203	NaN	NaN
397	10188	10201	10189	NaN	NaN	10179	10187	11310	11311	10203	NaN	NaN
398	10188	10201	10189	NaN	NaN	10179	10187	11310	11311	10203	NaN	NaN
399	10188	10201	10189	NaN	10180	10179	10187	11310	11311	10203	NaN	NaN
400	10188	10201	10202	NaN	10180	10179	10187	11310	11311	10203	10190	NaN
401	10188	10201	10202	NaN	10180	10179	10187	11310	11311	10203	10190	NaN
402	10188	10201	10202	NaN	10180	10179	10187	11310	11311	10203	10190	NaN
403	10188	10201	10202	NaN	10180	10179	10187	11310	11311	10203	10190	NaN
404	10188	10201	10202	10189	10180	10179	10187	11310	11311	10203	10190	NaN
405	10188	10201	10202	10189	10180	10179	10187	11324	11311	10203	10191	NaN
406	10188	10201	10202	10189	10180	10179	10187	11324	11311	10203	10191	NaN
407	10188	10200	10202	10189	10180	10179	10187	11324	11311	10203	10191	NaN
408	10188	10200	10202	10189	10180	10179	10187	11324	11311	10203	10191	NaN
409	10188	10200	10202	10189	10180	10179	10199	11324	11311	10203	10191	NaN
410	10188	10200	10202	10189	10180	10179	10199	11325	11311	10203	10191	NaN
411	10188	11309	10202	10189	10180	10179	10199	11325	11311	10203	10191	NaN
412	10188	11309	10202	10189	10180	10187	10199	11325	11311	10203	10191	NaN
413	10188	11309	10202	10189	10180	10187	10199	11325	11311	10203	10191	NaN

- Cost Matrix (Pointing change based on the ***ADCS solution***)

From Pointing Option #1-19

To Pointing Option #1-19

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
1	1	5	5	5	5	5	5	11	11	11	11	11	11	11	11	11	11	11	11
2	6	1	5	5	10	9	5	5	8	10	11	11	11	11	12	11	11	9	5
3	5	5	1	5	8	10	11	9	5	5	8	11	11	10	11	11	12	11	10
4	5	9	5	1	5	5	10	11	10	9	5	5	8	11	11	11	11	10	12
5	5	10	9	5	1	5	5	11	12	11	10	9	5	5	8	10	11	10	11
6	6	8	10	11	5	1	5	10	11	11	12	12	11	10	5	5	8	10	11
7	9	5	5	10	9	5	1	10	11	10	11	11	12	11	11	10	5	5	8
8	10	6	10	11	12	11	11	1	6	9	11	11	12	12	11	12	12	10	5
9	10	9	7	11	10	11	11	5	1	6	9	11	11	12	12	12	12	12	10
10	11	11	6	5	11	12	11	10	5	1	6	9	11	11	12	12	11	12	11
11	11	11	9	8	10	10	12	11	10	5	1	6	10	10	11	12	12	11	12
12	10	11	11	6	5	11	12	12	11	10	5	1	6	9	11	11	12	12	11
13	11	11	11	10	7	11	9	11	12	11	10	5	1	6	9	10	11	12	12
14	11	12	11	11	6	5	10	12	11	12	12	10	5	1	6	9	10	11	12
15	10	10	11	11	10	7	10	12	12	11	12	12	10	5	1	6	9	10	11
16	10	11	12	11	11	6	5	11	12	12	11	12	12	10	5	1	6	9	10
17	10	11	10	12	11	10	9	10	11	12	12	11	12	12	10	5	1	6	9
18	9	5	11	12	11	11	6	9	10	11	12	12	11	12	12	10	5	1	6
19	10	7	11	10	11	11	10	6	9	11	11	12	12	11	12	12	11	5	1

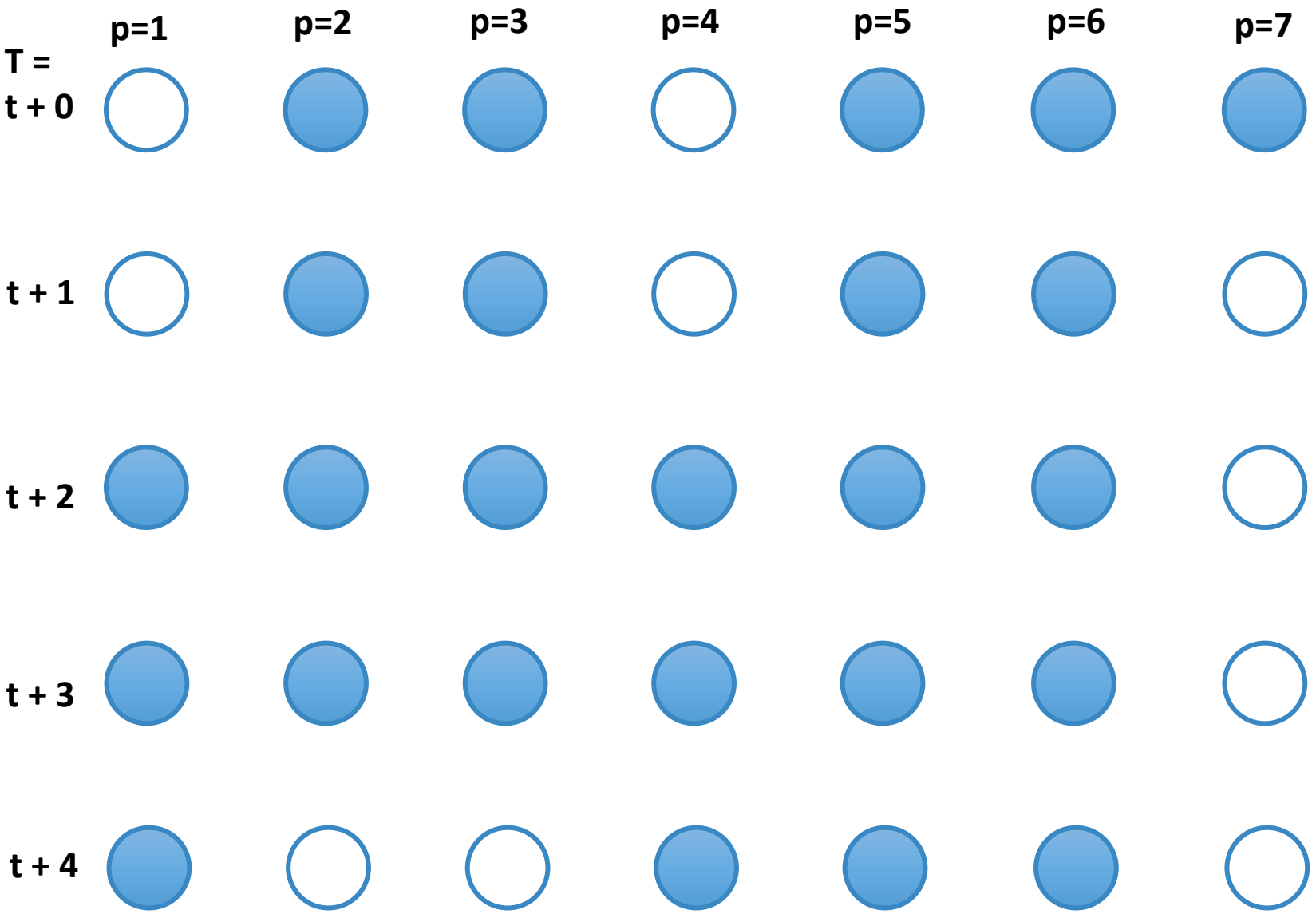
MILP?

Heuristic Optimization?

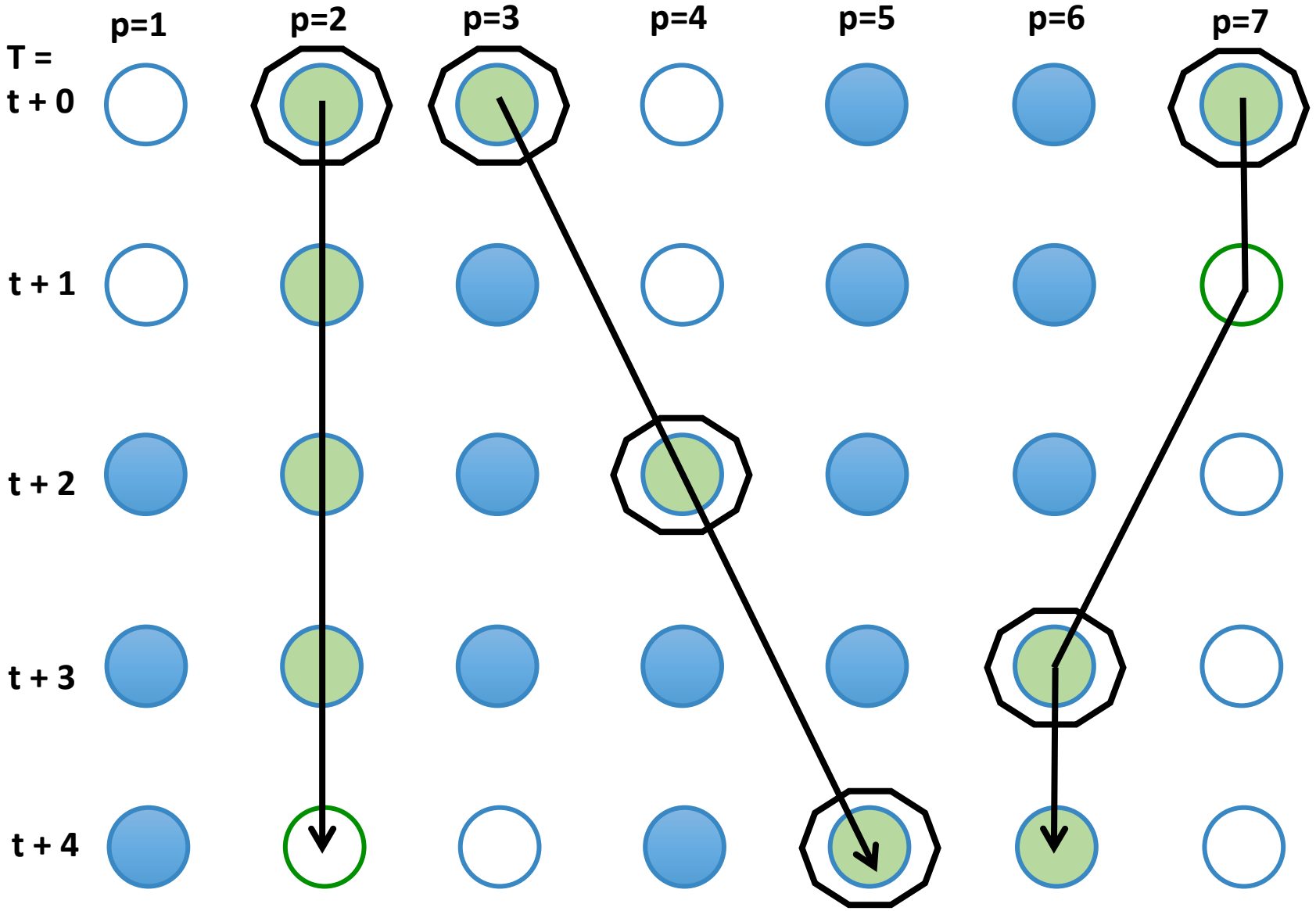
Dynamic Programming?

Bay Area Environmental Research Institute

Dynamic Programming Approach



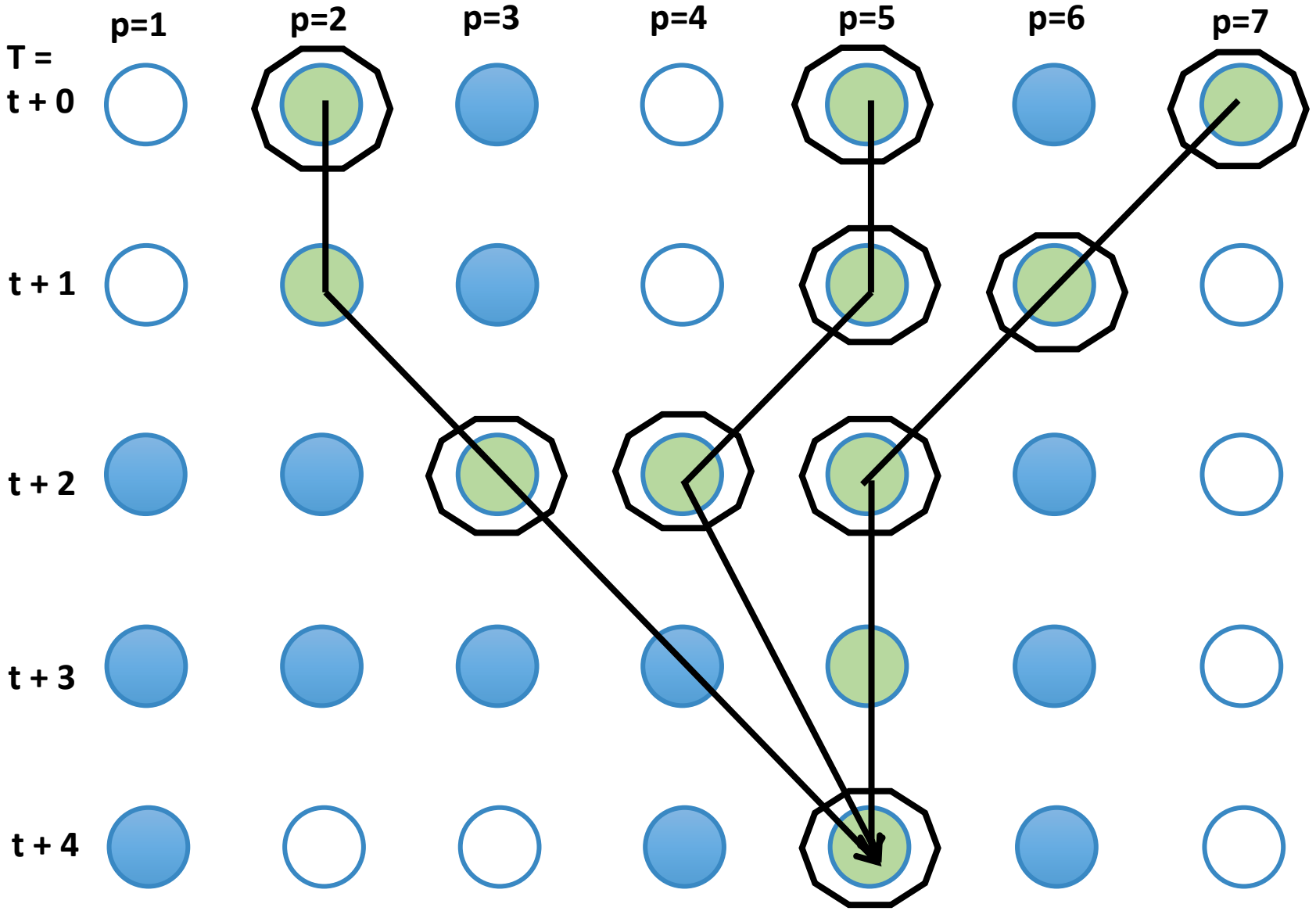
Dynamic Programming Approach



Example sub paths #1, #2, #3 shown

- Unique Images in the paths are circled in black and traced images colored green
- Paths have to take into account pointing switching time and skip that many rows/time steps

Dynamic Programming Approach



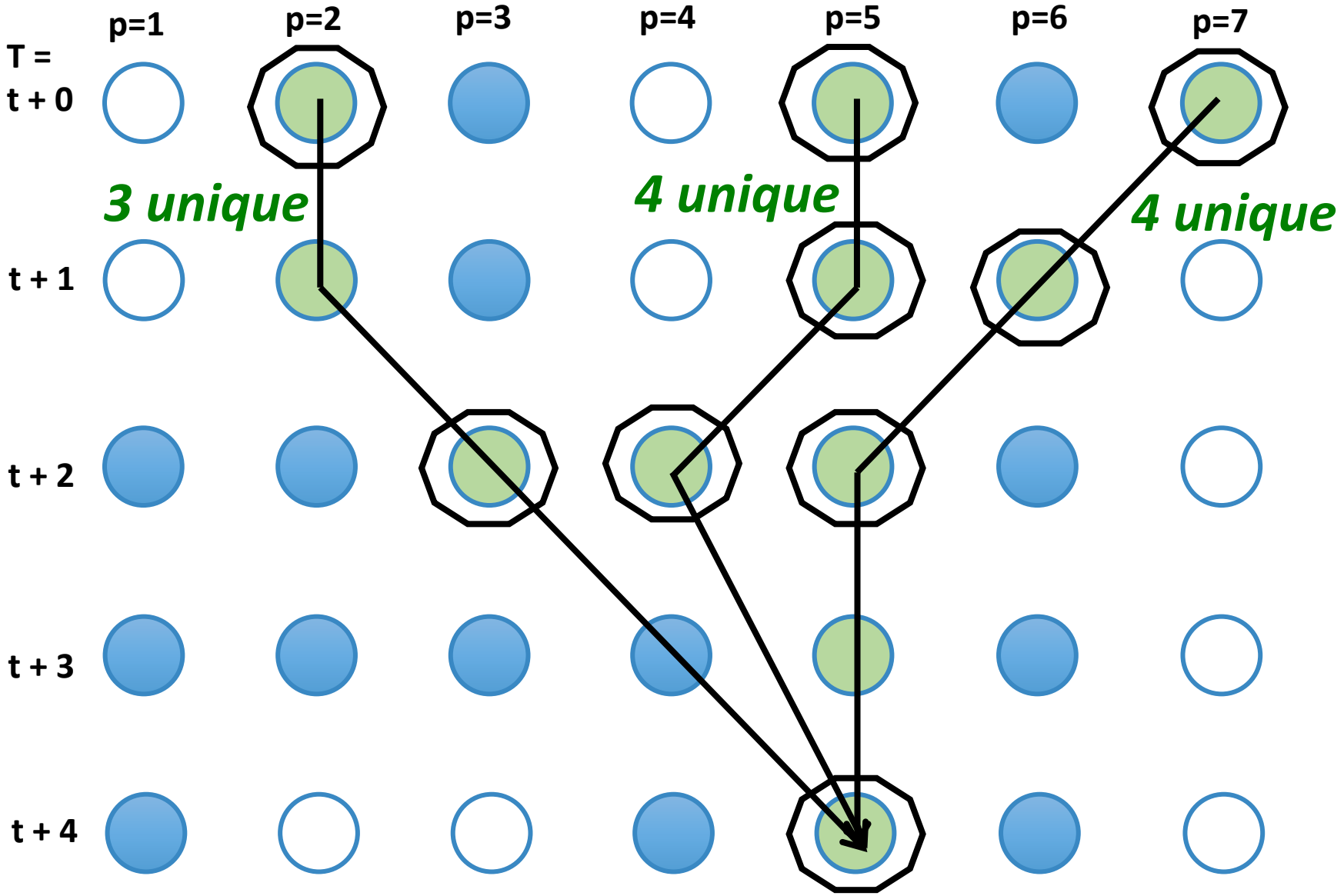
Proposed Algorithm:

- For every time step and every node:
- Find all paths leading to it (from 19 pointing options) and select+store the 'best' path/s only
- 'Best' is evaluated in 2 levels:

1) Path/s with the maximum unique number of seen images
2) Paths/s for which there are the least number of remaining opportunities for the seen images in the path

- No recalculation is needed because the unique images, paths and opportunities are stored up to the previous node and can be appended to.

Dynamic Programming Approach



Proposed Algorithm:

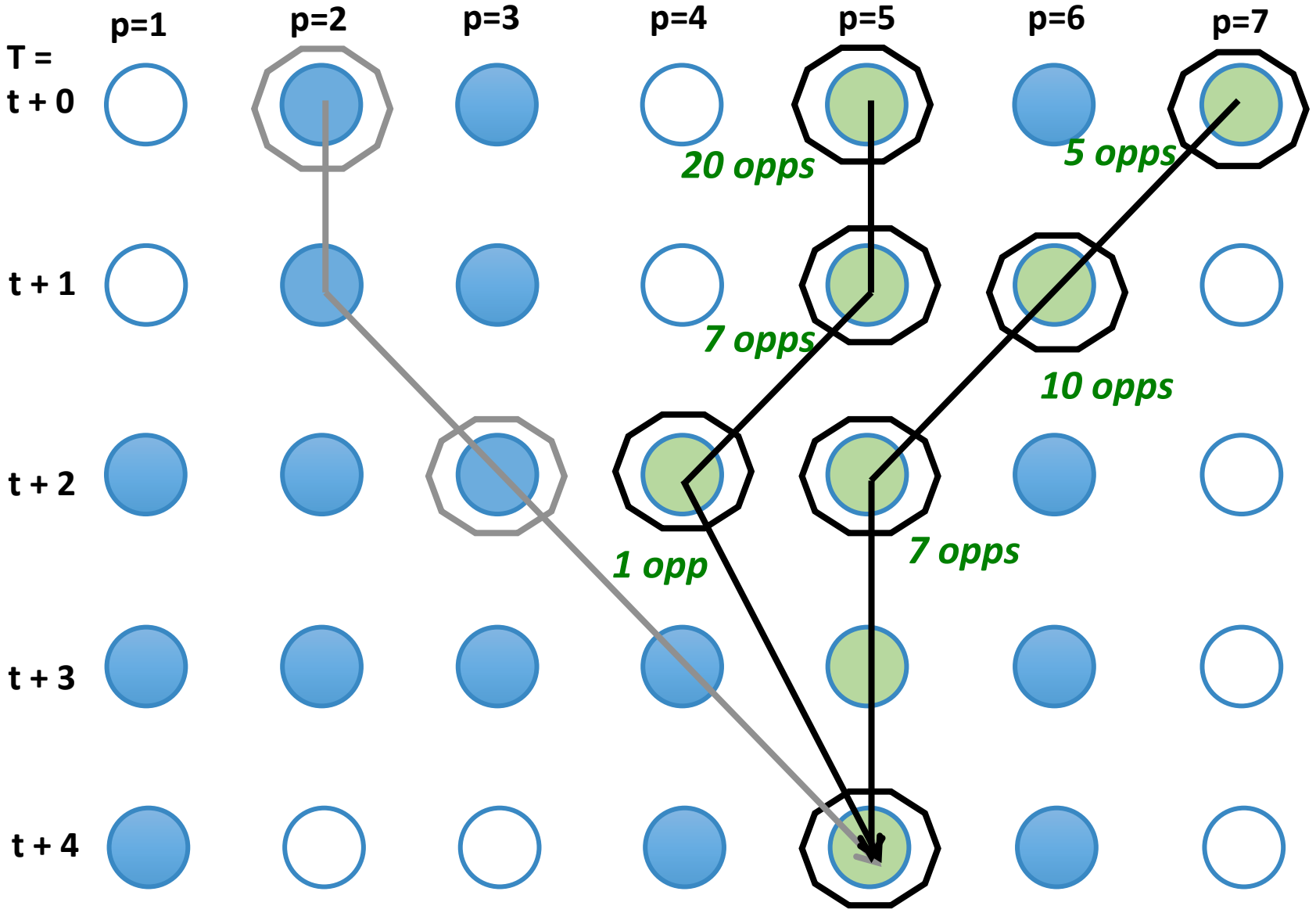
- For every time step and every node:
- Find all paths leading to it (from 19 pointing options) and select+store the 'best' path/s only
- 'Best' is evaluated in 2 levels:

1) Path/s with the maximum unique number of seen images
2) Paths/s for which there are the least number of remaining opportunities for the seen images in the path

- No recalculation is needed because the unique images, paths and opportunities are stored up to the previous node and can be appended to.

Bay Area Environmental Research Institute

Dynamic Programming Approach



Proposed Algorithm:

- For every time step and every node:
- Find all paths leading to it (from 19 pointing options) and select+store the 'best' path/s only
- 'Best' is evaluated in 2 levels:

1) Path/s with the maximum unique number of seen images
2) Paths/s for which there are the least number of remaining opportunities for the seen images in the path

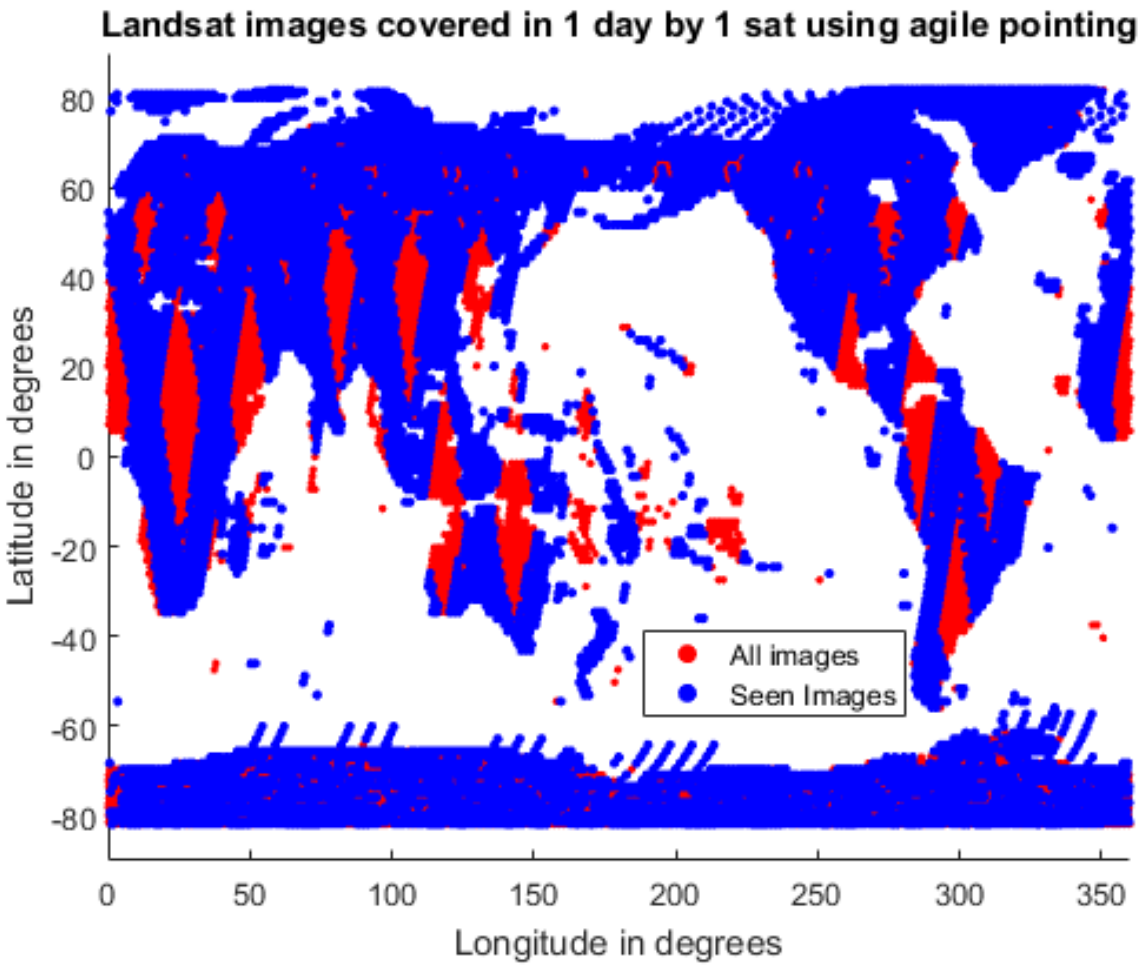
- No recalculation is needed because the unique images, paths and opportunities are stored up to the previous node and can be appended to.

Bay Area Environmental Research Institute

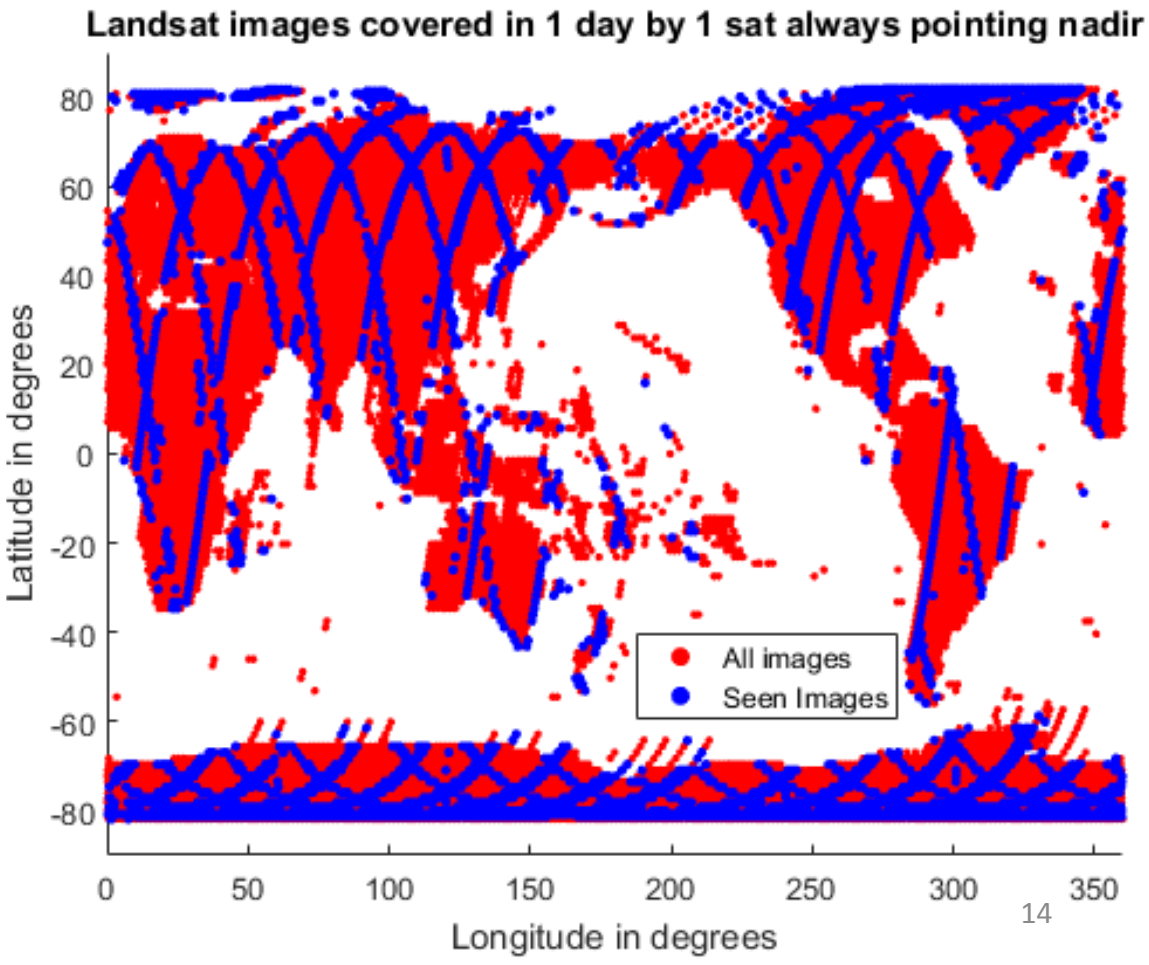
Results using a Single Satellite

Over a full day's worth of simulation/86400 seconds:

- Using our proposed DP algorithm



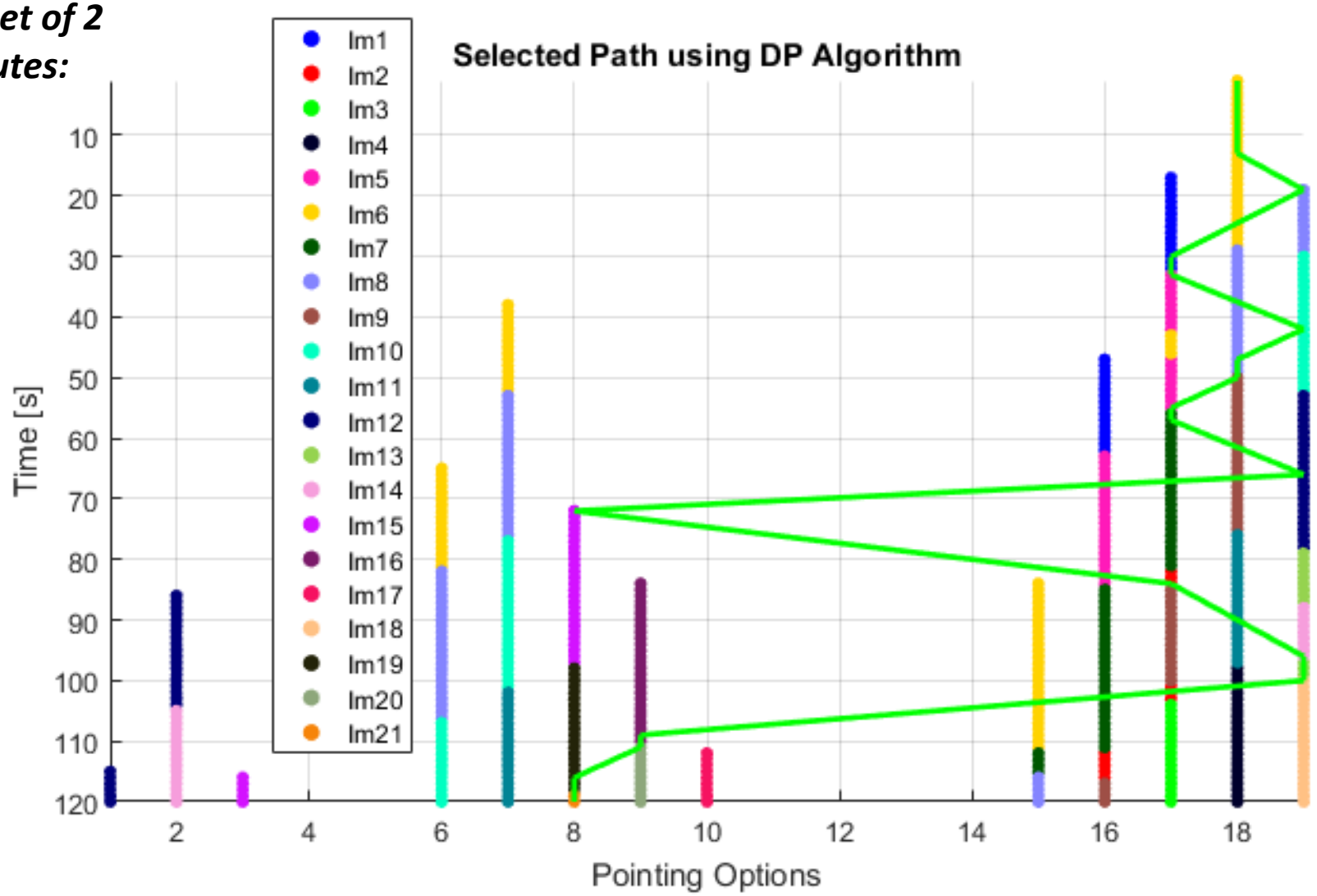
- Using a fixed Landsat sensor, as is



Bay Area Environmental Research Institute

Results using a Single Satellite

An example subset of 2 minutes:



- Image #1 is not an image
- Of 21 possible images in these 2 minutes, 18 were seen.
- In comparison, 2-7 images were seen for fixed pointing and only 1 when scanning
- Note the preference for off-nadir pointing

Bay Area Environmental Research Institute

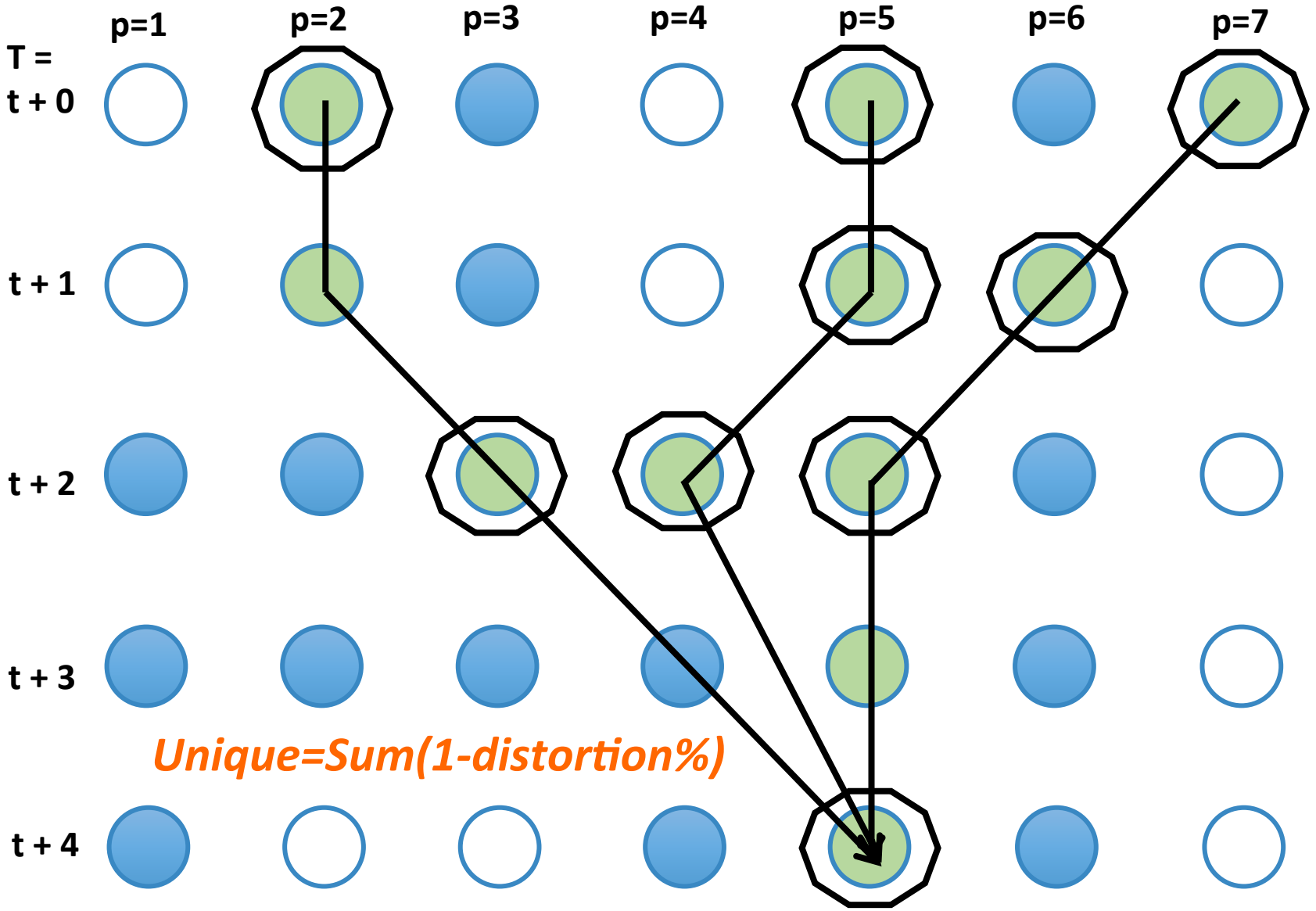
Results using a Single Satellite

All algorithms are linear in time. Verified in simulation.

Over a full day's worth of simulation/86400 seconds:

- Of 14724 possible images, 11900 were seen.
- In comparison, max 4894 images were seen using the static single-look conops and 3079 images using the whiskbroom/scanning approach
- Algorithm covered 77.5% from possible images and 70% from total … 2.5 times the static case and 3.86 times the scanning approach
- However, <6% of the seen images are nadir-viewing and >65% have maximum distortion
- Image distortion can be added to the path-selection algorithm by weighting the seen images with *(1-distortion%) where %=f(pointing) and leftover images with *(1+distortion%)

Dynamic Programming Approach



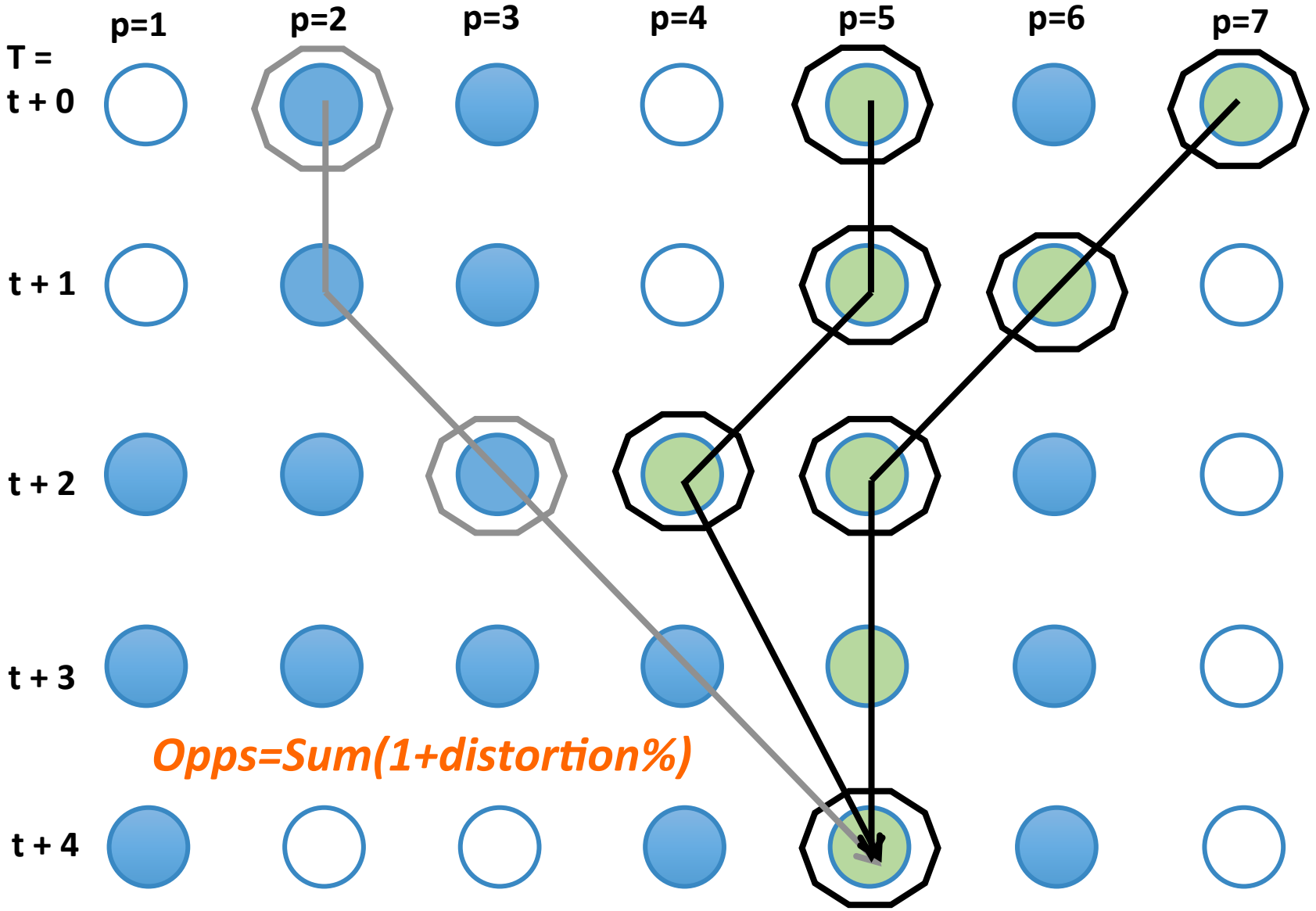
Modified Algorithm:

- For every time step and every node:
- Find all paths leading to it (from 19 pointing options) and select+store the 'best' path/s only
- 'Best' is evaluated in 2 levels:

1) Path/s with the maximum weighted unique number of seen images
2) Paths/s for which there are the least number of weighted remaining opportunities for the seen images in the path

- No recalculation is needed because the unique images, paths and opportunities are stored up to the previous node and can be appended to.

Dynamic Programming Approach



Modified Algorithm:

- For every time step and every node:
- Find all paths leading to it (from 19 pointing options) and select+store the 'best' path/s only
- 'Best' is evaluated in 2 levels:

1) Path/s with the maximum weighted unique number of seen images
2) Paths/s for which there are the least number of weighted remaining opportunities for the seen images in the path

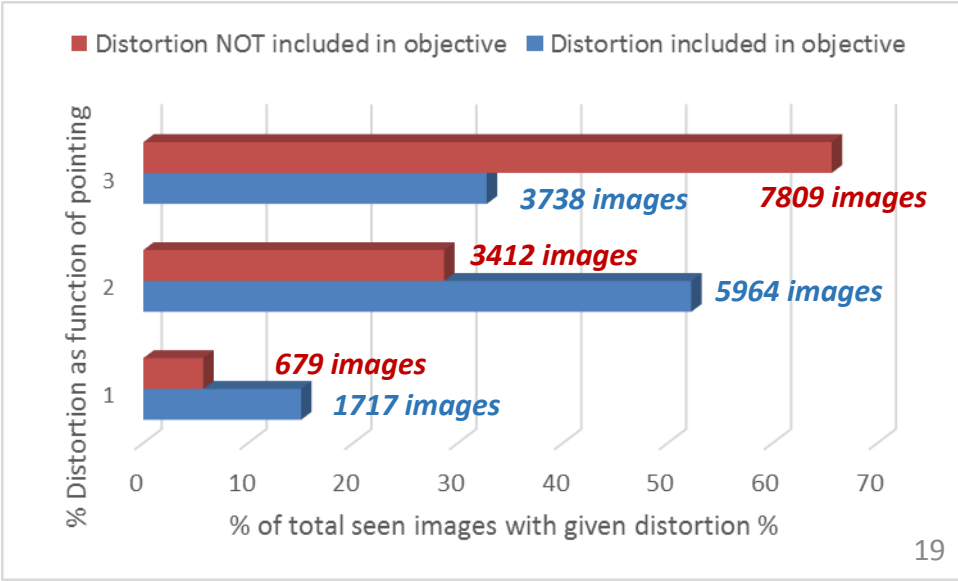
- No recalculation is needed because the unique images, paths and opportunities are stored up to the previous node and can be appended to.

Bay Area Environmental Research Institute

Modified Results using a Single Satellite

Over a full day's worth of simulation/86400 seconds:

- Of 14724 possible images, 11418 were seen instead of 11900.
- The algorithm covered 77.5% of the possible and 67% from the total Landsat selection.
- As before, fixed pointing covers only 32% of the possible images so the algorithm does ~2.4 times better than the static conops
- Adding distortion minimization to the objective function moves more images to the 0% and 8% GSD distortion.
- 9548.5 effective images (weighted by 1-percDist) were seen instead of 8785.4 effective images (w/o weights)

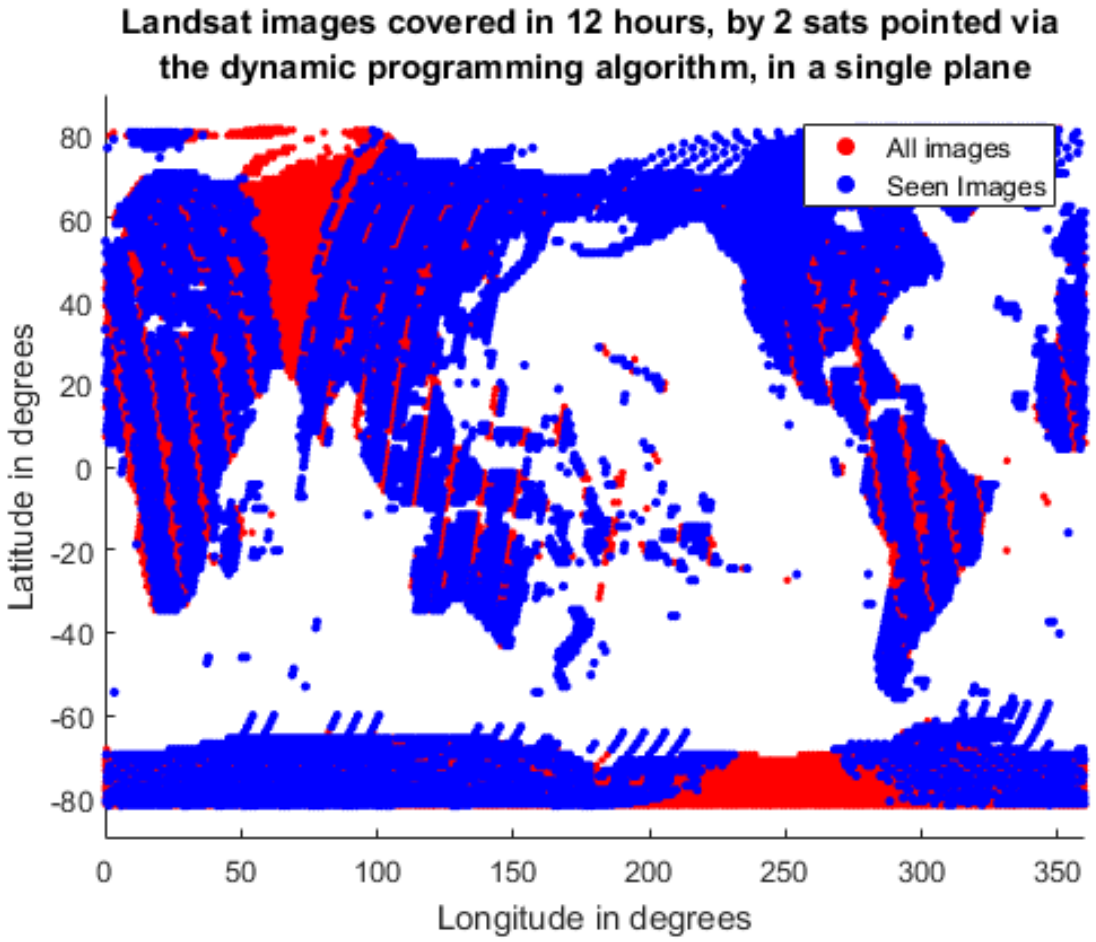


Bay Area Environmental Research Institute

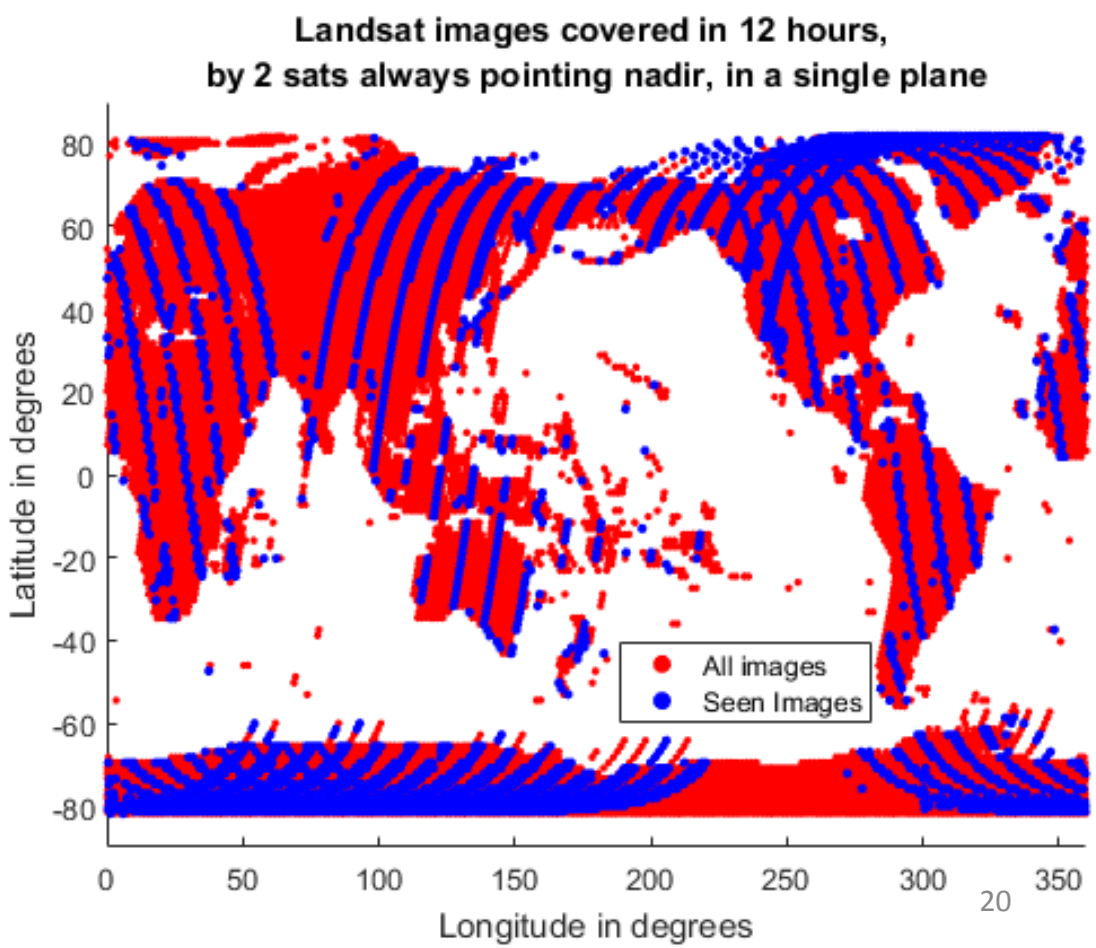
Results using a Constellation

Over 6 hours of simulation/43200 seconds:

- Using our proposed DP algorithm



- Using a fixed Landsat sensor, as is



Bay Area Environmental Research Institute

Results using a Constellation

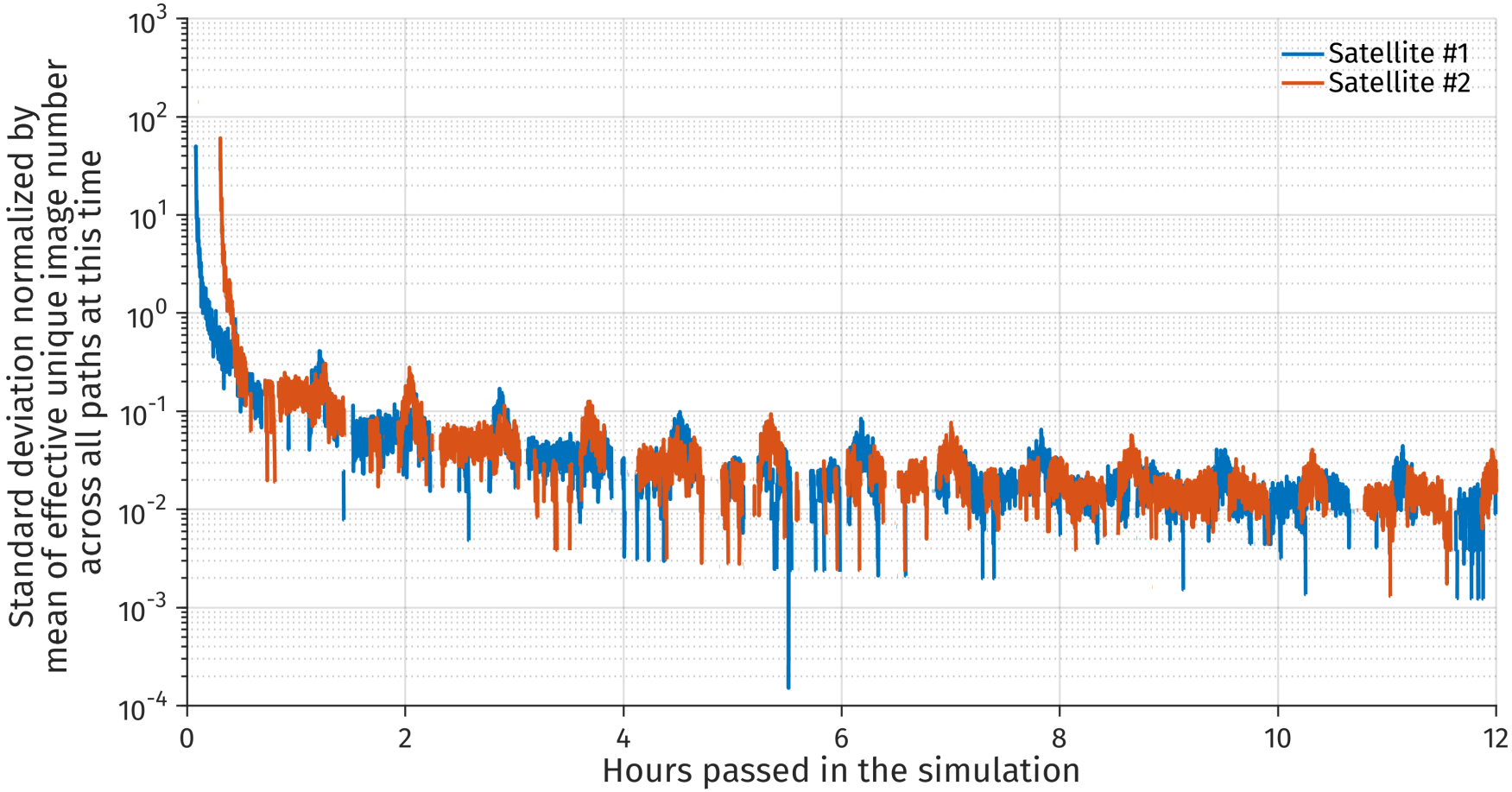
Over 6 hours of simulation/43200 seconds using 2 satellites, 180 deg apart in the same plane:

- Of 14163 possible images, 10847 were seen (note half the time as before).
- In comparison, 4366 images were seen using the static fixed pointing conops, where in the satellite always points nadir, i.e. 60% lesser images
- Our algorithm covered 76.6% from possible images and 65% from total … 2.5 times the number obtained using the fixed pointing approach
- BUT there were 2230 unique images, common between those imaged by the two satellites because the DP algorithm evaluates uniqueness per satellite path.
- I.E., the satellites in the sim should know the other's optimum images, as frequently as possible to avoid repeats WHILE each satellite should have as long a sim time window to optimize its path efficiently AND in a real scenario, more time allows for checking which images were actually collected.

Bay Area Environmental Research Institute

Results using an Informed Constellation

Same algorithm implemented on the constellation simulation however, individual path optimization is for every X hours AND each satellite is made aware of the images seen by the optimum schedule/path of all others every X hours



- At any given time, how different are the potential paths from each other?
- Unique images seen deviate lesser across paths with increased sim time
- Knee @ 45 mins
- Number of unique paths over time oscillates like white noise every 15-30 mins – okay to cut off.

Bay Area Environmental Research Institute

Results using an Informed Constellation

Same algorithm implemented on the constellation simulation however, individual path optimization is for every X hours AND each satellite is made aware of the images seen by the optimum schedule/path of all others every X hours

Of 14163 possible images, 10847 were seen w/o information sharing (i.e. X = 12 hours) and 4366 were seen without any active pointing

X	Unique Images seen	Repeat images (not counted)	Improvement from X=12 hrs	Improvement from no agility
6 hours	10948	1751	0.9%	150.7%
3 hours	11027	1410	1.6%	152.5%
1.5 hours	11137	1018	2.7%	155%
45 minutes	11430	0	5.4%	161.8%

Unique images seen increases with decreasing X and is the maximum for min repeat. Optimum X = 45 mins for this 2 sat, 1 plane, 180 deg phase constellation but depends on the knee position for others, which depends on the constellation structure.

Bay Area Environmental Research Institute

Future Improvements

- Apply to different constellation structures with varying satellite numbers to check sensitivity of X (time horizon for optimization and sharing). Apply to other point groups.
- Modify the access matrix by removing eligible images based on time-dependent Cloud Cover, Ground Station downlink/access, illumination conditions
- Add uncertainty characterization in the algorithm as a function of physical limitations of momentum wheels used in ADCS and position/attitude knowledge uncertainties
- Explore better orienteering algorithms in terms of runtime/memory and effective images covered (w/ distortion and potentially signal to noise ratio)
  - ***Travelling Salesman (TSP) approach*** can be viable if images can be maximized (instead of all visited at least once) and if distortion can be added to nodes depending on the edge (same image, different pointing).

Image #	From Image #	'From' Time (s)	To Image #	'To' Time (s)
100	[2 4 8 10 16…]	[4 1 20 2 2…]	[101 102 200…]	[10 1 5…]
101	[100 90 95…]	[10 1 16…]	[110 111 112…]	[1 5 7…]
… and so on for each of the 14724 unique images				

Acknowledgments:

NASA ESTO QRS Grant for funding the project

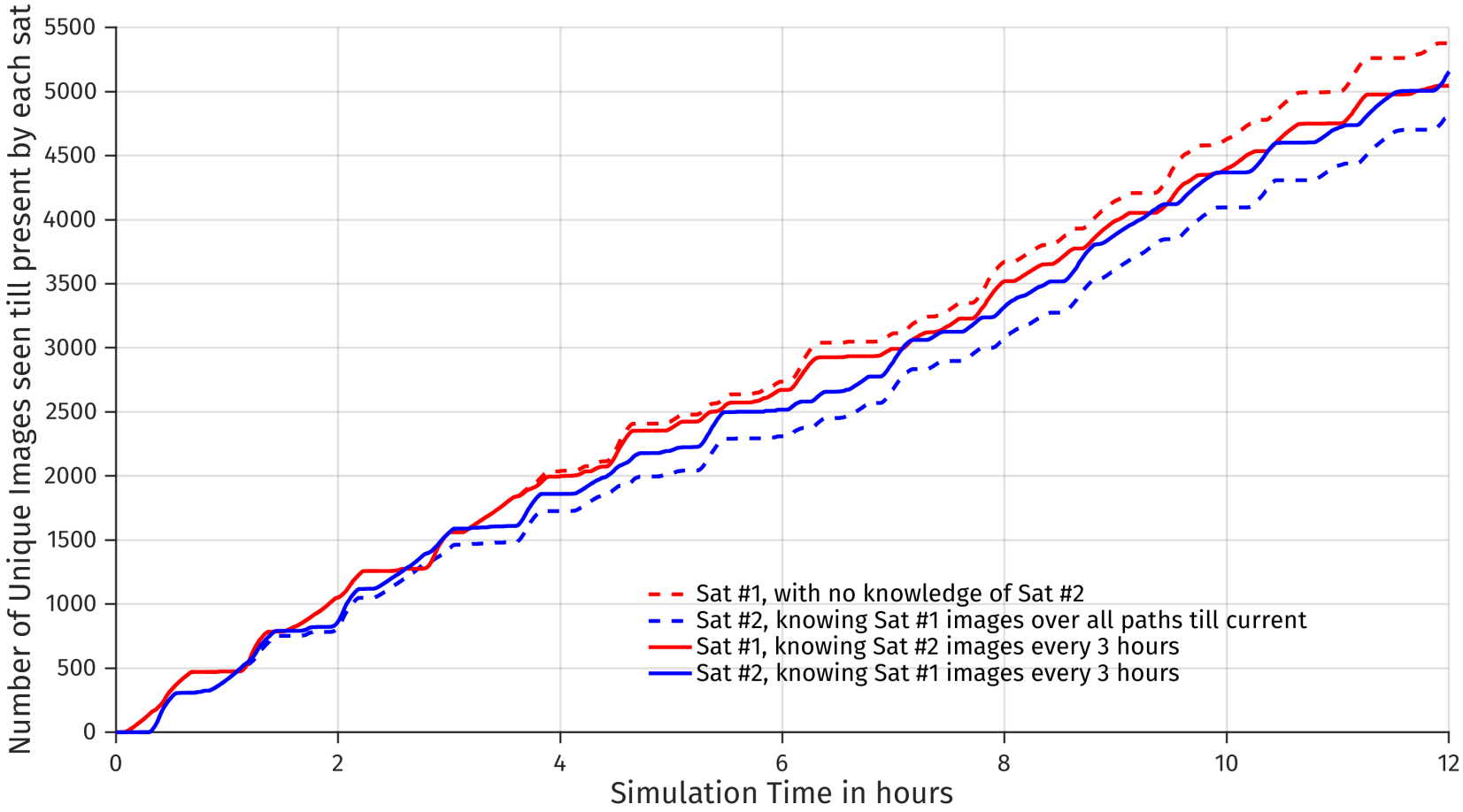
Questions?

Sreeja.Nag@nasa.gov

Bay Area Environmental Research Institute

Results using an Informed Constellation

Same algorithm implemented on the constellation simulation however, individual path optimization is for every X hours AND each satellite is made aware of the images seen by the optimum schedule/path of all others every X hours



- Unique paths over time is oscillates like white noise every 15-30 mins.
- Alternative – Satellites could avoid all images in any possible path/ schedule of the other satellites. This reduces unique image repeats to a similar order BUT the image distribution across satellites is not fair

Pointing Options per satellite