Planning Satellite Swarm Measurements for Earth Science Models: Comparing Constraint Processing and MILP Methods

Rich Levinson[1,2] **Samantha Neimoeller**[1] **Sreeja Nag**[1,3] **Vinay Ravindra**[1,3]

{rich.levinson, samantha.c.niemoeller, sreeja.nag, vinay.ravindra} @nasa.gov

[1]NASA Ames Research Center
[2] KBR Wyle Services, LLC
[3] Bay Area Environmental Research Institute
Moffett Field, CA 94035

Collaborators:

Mahta Moghaddam (University of Southern California)
Daniel Selva (Texas A&M University)

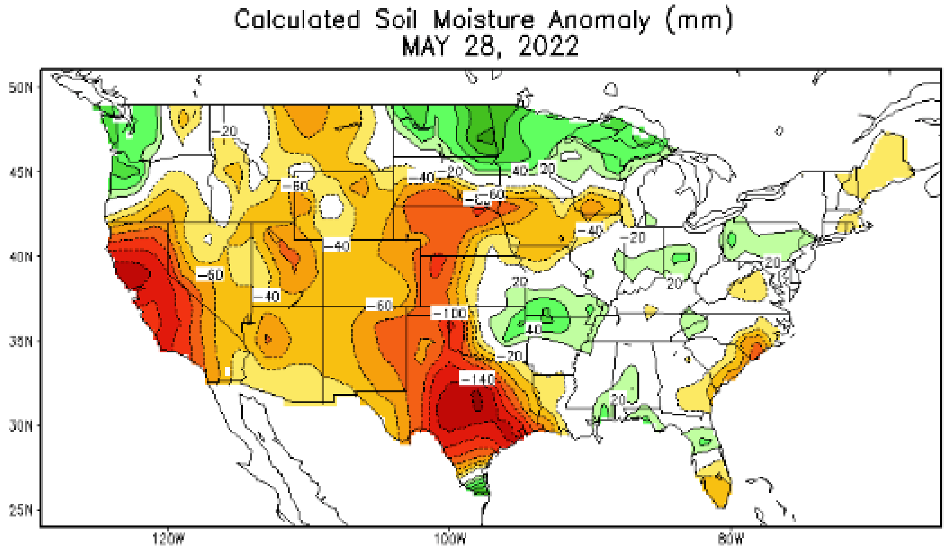
Funded by:

NASA Earth Sciences Technology Office (ESTO)

Background and Motivation

Soil Moisture Prediction

- Soil moisture is a key indicator for predicting floods, mudslides, wildfires, other phenomena



(NOAA/National Weather Service website)

Problem:

- We have initial solution for improving soil moisture prediction
- Our Constraint Processing planner follows a narrow beam of heuristically guided trajectories through a **huge** search space
- We don't know how optimal our heuristic solution is

Research Questions:

- How close to optimal is our heuristic solution?
- Can we model our constraints in Mixed Integer Linear Programming (MILP)?
- How long would it take MILP to prove optimality on our 6-hour plan horizon?

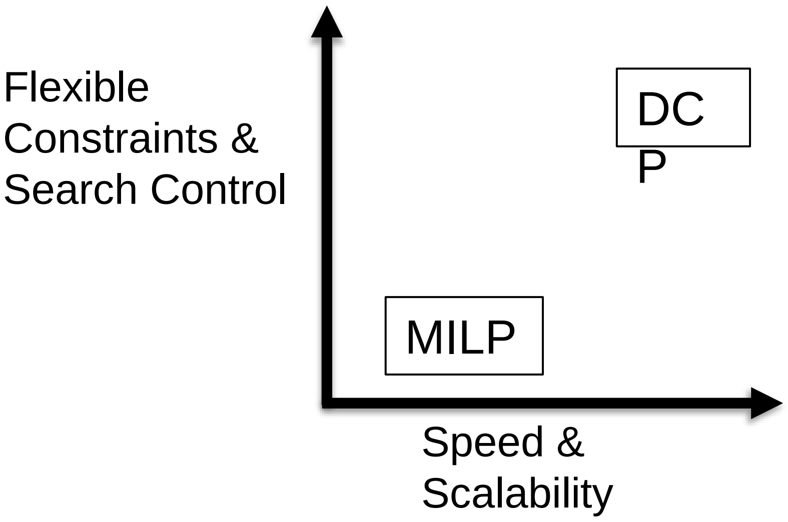
Dynamic Constraint Processing vs. Mixed Integer Linear Programming

Dynamic Constraint Programming (DCP)

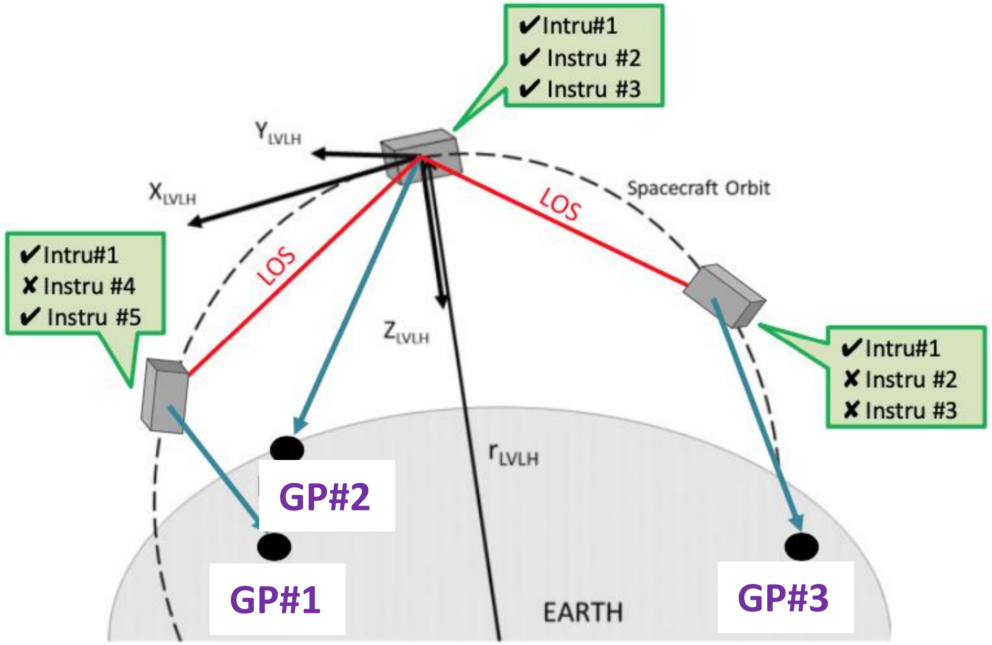
- *Suboptimal but Fast*
- *Constraints enforced "on-demand"*
- *Variables dynamically eliminated by constraint handlers*

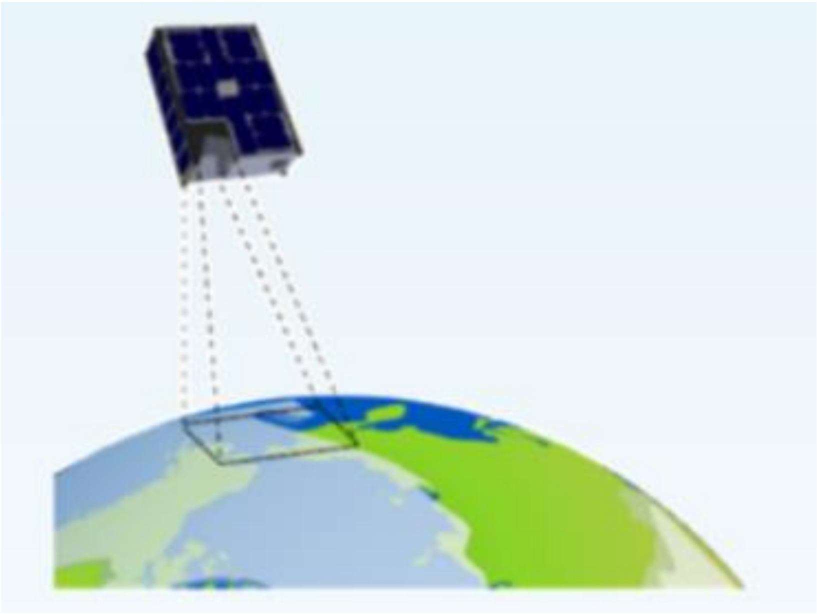
Mixed Integer Linear Programming (MILP)

- *Optimal but Slow*
- Less flexible constraint modeling
- Quantitative declarative constraints
- Scalability challenges



D-SHIELD:
Distributed Spacecraft w/ Heuristic Intelligence to Enable Logistical Decisions





- Multiple satellites w/ multiple instruments to observe *Ground Positions* (**GP**)
- Each satellite has 2 different sensors & can point at 61 different angles
- Each observation covers multiple GP (9 km x 9 km tile)

- *For each satellite:*
  Assign a sensor command for every *Time Point* (**TP**) when it can see any **GP**

D-SHIELD Constraints

Constellation-wide constraint

- **Duplicate Observations:**
  No duplicate GP observations (across all satellites)

Satellite-level mutex constraints (only do one thing at a time)

- **Image Lock** – hold viewing angle for 3 seconds per observation
- **Maneuver constraints** – slew time for changing view angle

Planner Input: GP Access Times & Command Choices

Search space *for each satellite*:

Timepoint (TP)	Command Choices	Ground Positions (GP) covered by command
10	L.32 (L-band, angle 32)	25, 26, 27
	P.32 (P-band, angle 32)	24, 25, 26, 27, 28
	P.34 (P-band, angle 34)	36, 38, 40, 47, 49

- Access times (TP)
- Command choices (sensor & view angle) for each access time (TP)
- List of GP covered by each command

Planner Input: Prediction and Measurement Errors

Soil Moisture *Prediction* Error

Ground Position (GP)	Timepoint (TP)	Prediction Error
15	100	0.02
	500	0.23

Measurement Error

Sensor Command	Ground Cover	Measurement Error
L.48		
	Forest	0.035
	Shrubs	0.025

↑ Error *increases* with time and rain

↓ Error *decreases* with "good" observation (measurement error < pred. error)

Planner Input: Prediction and Measurement Errors

Soil Moisture *Prediction* Error

Ground Position (GP)	Timepoint (TP)	Prediction Error
15	100	0.02
	500	0.23

Measurement Error

Sensor Command	Ground Cover	Measurement Error
L.48		
	Forest	0.035
	Shrubs	0.025

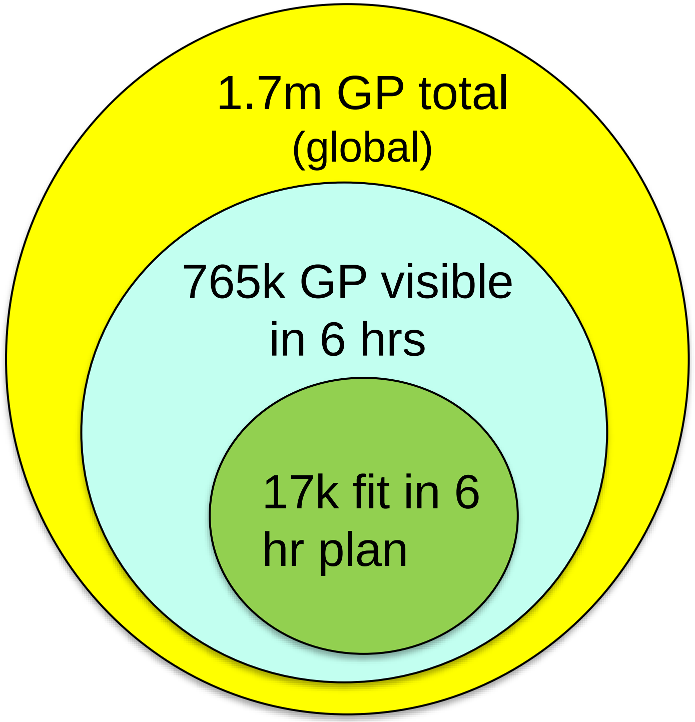
Goal:

Observe GP having most prediction error,
using measurements with least error

Search Space Combinatorics

Scenario: 3 satellites, 6-hour plan horizon:

- ~8700 Time Points (TP) when GP are visible
- ~55 command choices/TP (max: ~150 choices/TP)
- # nodes in search space $\left(\frac{\#\ cmd\ choices}{TP}\right)^{\#TP}$ = 55^{8700} = *"Infinity"*



Different Planning Models

Dynamic Constraint Processing (Levinson et al., IWPSS '21)

Qualitative Decision Variables:

- $x_{s,t}$ = the command choice for sat s at time t. $\forall\, t \in$ {All TP when sat s can see any GP}
  $x_{1,25} \in$ {L.32, L.34, P.33, P.34}
- Constraints are procedural (Python code) and called on-demand after each planner choice

MILP: *Binary* Decision Variables

- $x_{s,c,t}$ = 1 ↔ sat s executes command c at time t $x_{s,c,t} \in \{0,1\}$
- $y_{g,s,c,t}$ = 1 ↔ GP g is observed by sat s with command c at time t $y_{g,s,c,t} \in \{0,1\}$
- Constraints are *quantitative, declarative, and pre-enumerated*

Apples-to-Apples comparison:

- Requires identical inputs, model constraints, and metrics

Objective: Maximize reduction of GP model error

maximize Σ commandReward(cmd)

$\forall \;\; cmd \;\; \in plan\ P$

Plan:

TP	Cmd	Observed GP	Command Reward
10	L.32	25, 26, 27	0.25
20	P.21	33,35,39,40	0.32
30	p.25	50,21	0.20
objective: $\sum$ cmd rewards			**0.77**

commandReward (s, c, t) = reward for sat *s* executing cmd *c* at time *t*

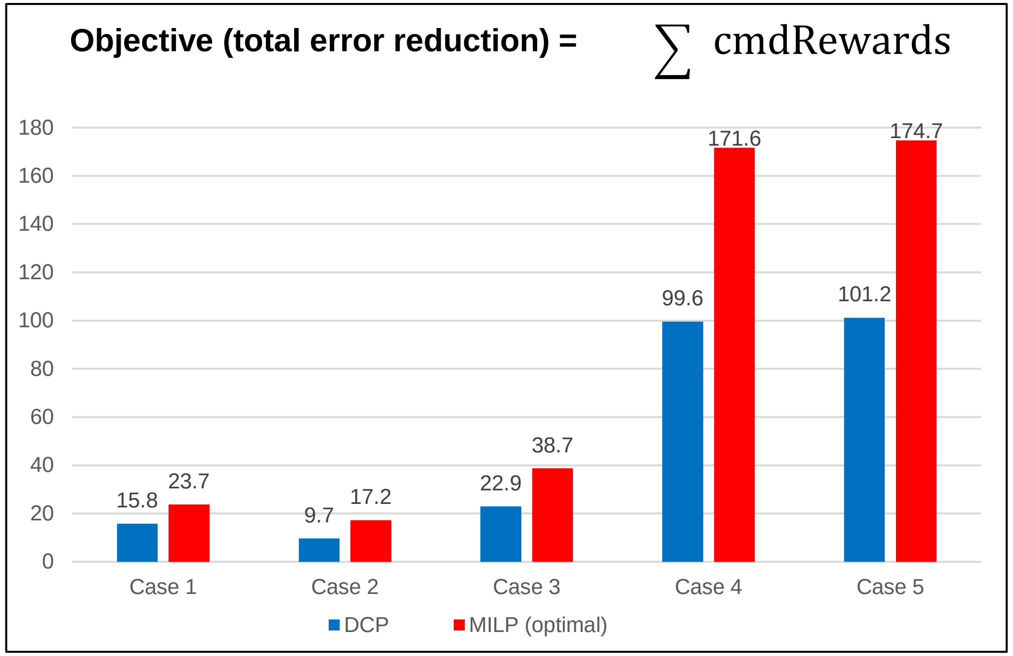
= Σ gpReward(g,c,t) $\forall\ GP\ g\ \in \{GP$ visible by sat *s* using cmd *c* at time *t*}

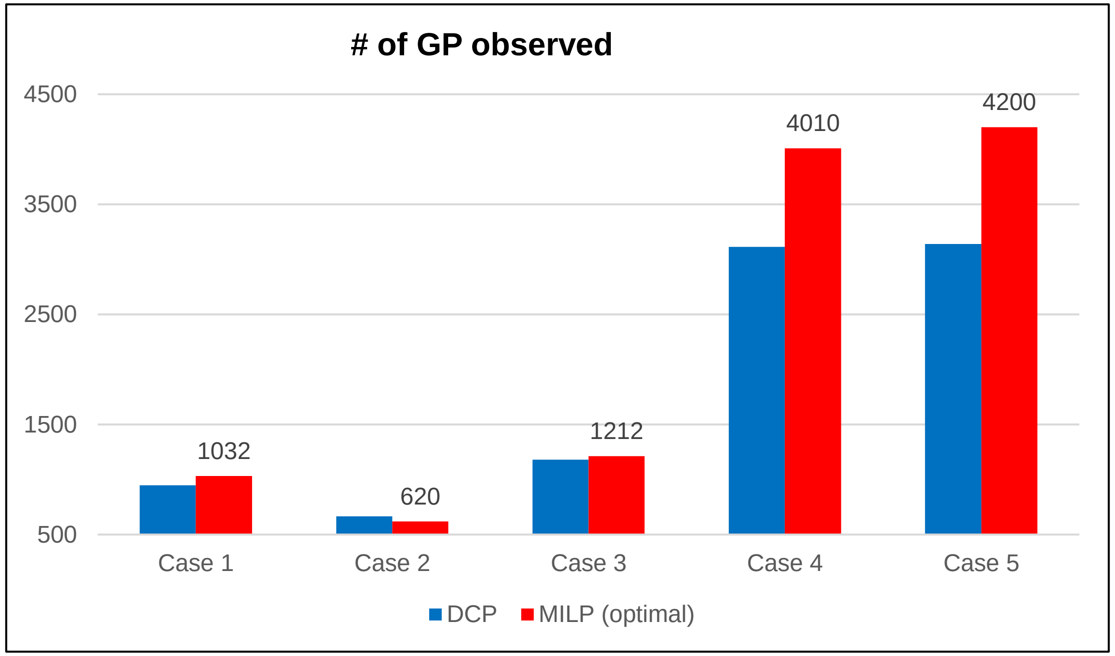
gpReward(g,c,t) = reward for viewing GP *g* with command *c* at time *t*

= $e_{g,t} - m_{c,b}$ where $e_{g,t}$ = *prediction* error for g at time *t*

$m_{c,b}$ = *measurement* error for command *c* in biome b (forest, shrub)

Comparison: Science Value



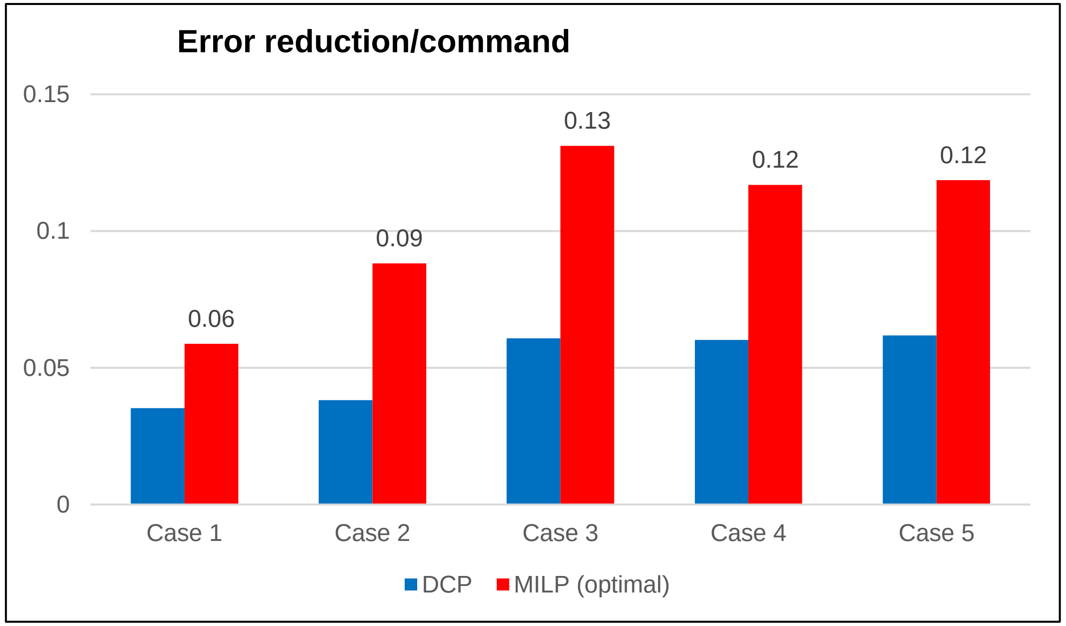


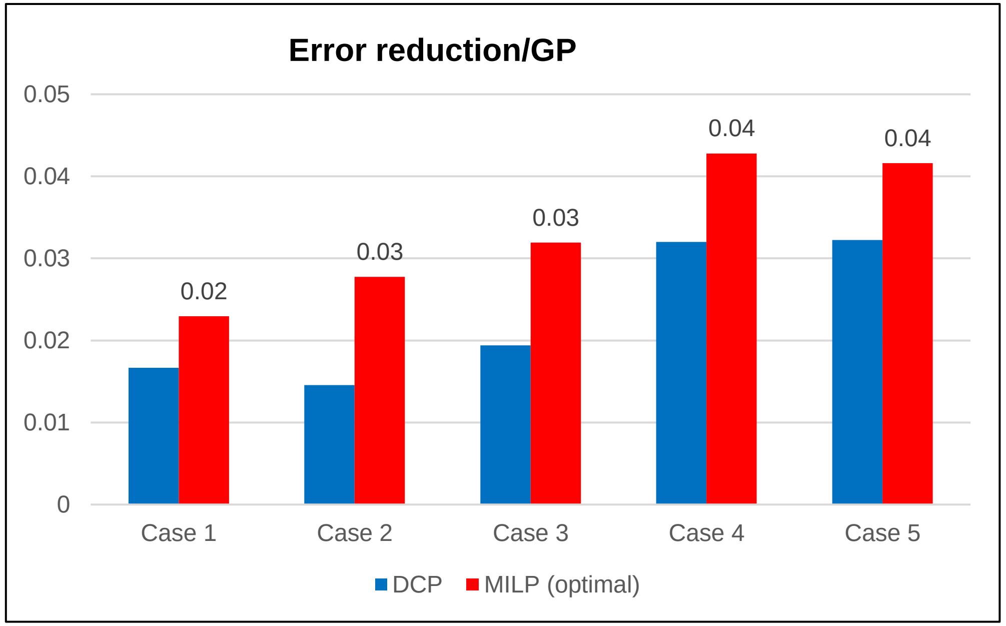
6 test cases with increasing plan horizons and complexity

- Cases 1 – 5: DCP achieves ~ 60 % optimal

- Case 6: **Unsolvable by MILP in 50 hour time limit, but DCP solves in 28 mins**

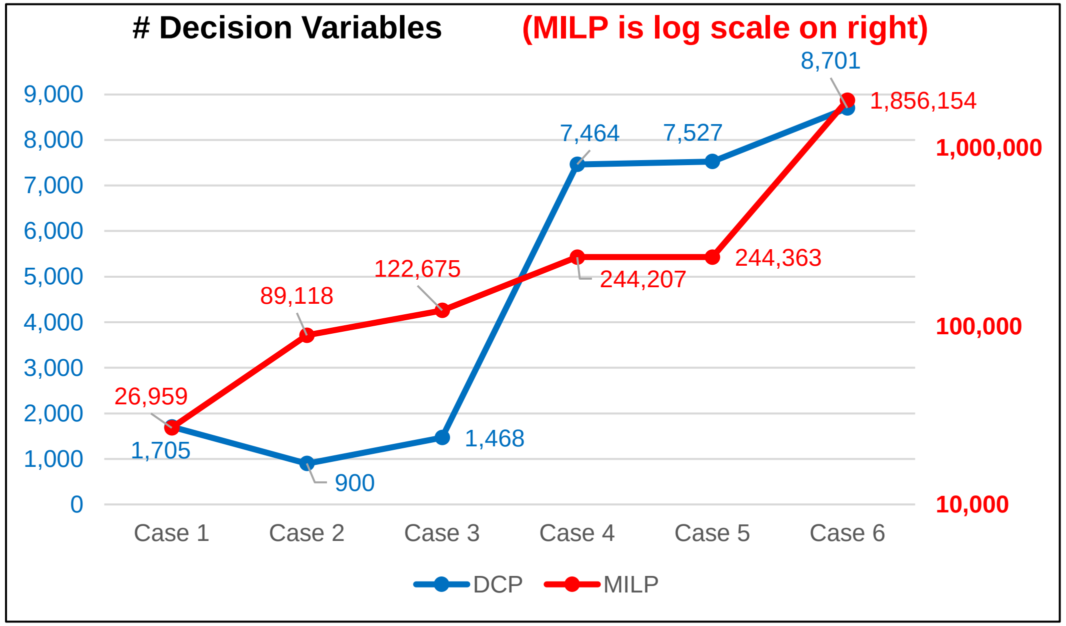
Comparison: Efficiency

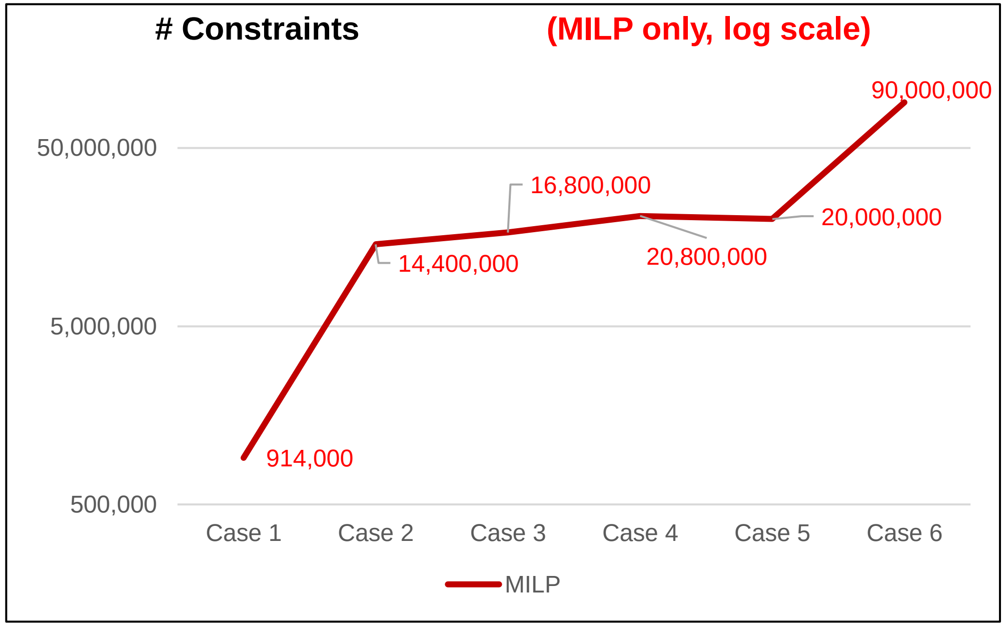




- MILP is always more efficient with higher objective rewards per command and per GP
- MILP plans always has fewer commands (makespan): ↓ Less slewing & energy cost

Comparison: Model Size





DCP: Scales linearly with # of timepoints

- # variables = # Timepoints in plan horizon
- # constraints is N/A because they are instantiated on demand

MILP: Always requires many more variables and constraints

- # vars = $\Sigma_{TP}\ \#\ commands(TP) + (\Sigma_{TP}\ \#\ commands(TP) * \#\ GP)$
- # constraints $\propto$ commands within 25 seconds of each other (mutex deconfliction)

Conclusion

	DCP	MILP
Optimal	✗	✓
Plan Efficiency	✗	✓
Speed	✓	✗
Flexible Constraints	✓	✗
Heuristic Search Control	✓	✗
Model Size/ Scalability	✓	✗

Both methods are useful, especially when used together

Future Work

- **Search Control:**
  - Monte Carlo Tree Search (MCTS)
  - Divide & Conquer (independent subproblems with TP gap delimiters)
- **Closing the execution loop**
  - Simulated satellites execute their plans with noise
- **New constraints:**
  - Relaxed 3-second image lock (not yet modelled in MILP)
- **New domain:**
  - wildfire prediction

Thank You

Questions?

rich.levinson@nasa.gov

video

Experiments

- Experiments ran on: 2020 MacBook Pro 13-inch 2.3 GHz Quad-Core Intel Core i7, 32 GB RAM.
- DCP solutions implemented in Python
- All MILP solutions used Gurobi 9.5.
- Data sets and software will be released open source

Six Test Cases

Experiment includes 6 test cases varying by:

- Plan horizon
  - 1,000 secs (~17 mins)
  - 1,800 secs (30 mins)
  - 7,200 secs (2 hours)
  - 21,600 secs (6 hours)
- Rainy or non-rainy GP cohort (rainy GP recently received rain)
- Triage heuristic
  - solve for the 15% most needy GP first
  - mitigation for MILP model size combinatoric explosion

Planner Objective: Maximize model improvement (error reduction)

Each GP: prior model error & cmd choices w/ measurement errors

Error(GP) = F(time, rain, biome type, instrument(s), view angle)

- ⬆ Error increases with time and rain
- ⬇ Error decreases with "good" observation (measurement error < model error)

GP	TP	Model Error	cmd	Measurement Error
2	27	0.08	L.41	0.14
	1500	0.39	P.48	0.07

➡

GP	Model Error	cmd	Measurement Error
2	~~0.39~~ 0.07	P.48	0.07

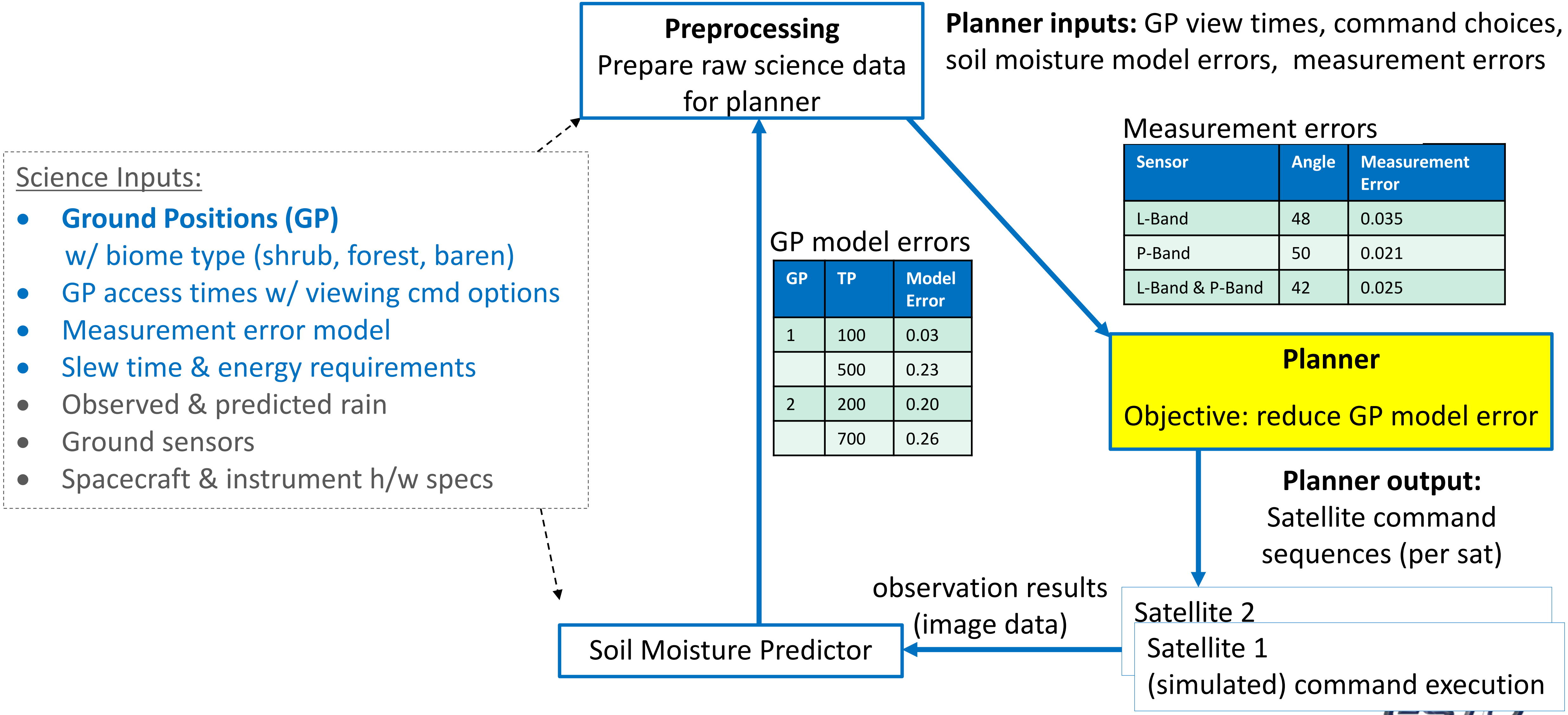
Planner Input: GP Access Times and Command Choices

For each satellite:
Search space of *access times*, *viewing options (commands)*, and *covered GP*

Command choice examples:
L.34 = <L-band, angle 34> ,
P.32 = <P-band, angle 32>

TP (time)	Command choices	GP covered by choice
1311:	L.32:	[3165]
	L.34:	[3445, 3446]
	P.33:	[3165]
	L.32 & P.32:	[3165]

D-SHIELD Architecture



Science Inputs:

- **Ground Positions (GP)**
  w/ biome type (shrub, forest, baren)
- GP access times w/ viewing cmd options
- Measurement error model
- Slew time & energy requirements
- Observed & predicted rain
- Ground sensors
- Spacecraft & instrument h/w specs

Preprocessing
Prepare raw science data for planner

Planner inputs: GP view times, command choices, soil moisture model errors, measurement errors

GP model errors

GP	TP	Model Error
1	100	0.03
	500	0.23
2	200	0.20
	700	0.26

Measurement errors

Sensor	Angle	Measurement Error
L-Band	48	0.035
P-Band	50	0.021
L-Band & P-Band	42	0.025

Planner

Objective: reduce GP model error

Planner output:
Satellite command sequences (per sat)

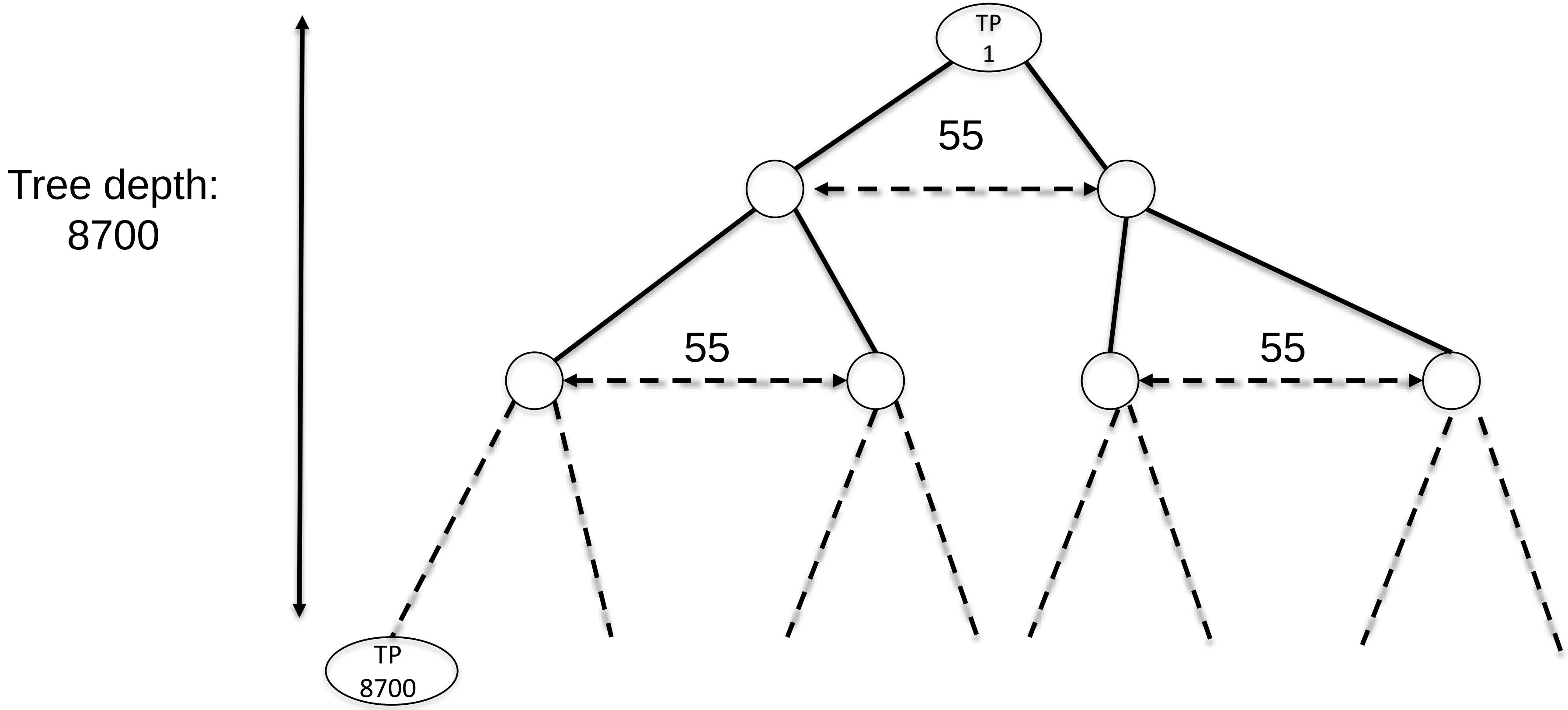
Satellite 2

Satellite 1
(simulated) command execution

observation results
(image data)

Soil Moisture Predictor

Search Space Combinatorics



- Each node is a decision var associated with a TP
- ~55 children/node (~55 cmd choices/TP)
- Tree Depth: 8700 = # of TP

Planner Model (MILP)

2 Binary Decision Variables

- $x_{s,c,t}$ = 1 ↔ sat s executes command c at time t, $x_{s,c,t} \in \{0,1\}$
- $y_{g,s,c,t}$ = 1 ↔ GP g is observed by sat s executes command c at time t, $y_{g,s,c,t} \in \{0,1\}$

Duplicate GP constraint: $\Sigma_{s,c,t}\ y_{g,s,c,t} = 1, \quad \forall_g \in G_{s,c,t}$

Mutex constraint (image lock and slew times block out other commands)

$x_{s,c,t_1} + x_{s,c,t_2} \leq 1 \quad \forall_{t1,t2} : t2 - t1 \leq 25 \quad (25 \text{ seconds} = \text{max slew time})$

GP Coverage Constraint: $y_{g,s,c,t} \leq x_{s,c,t}$ Tracks which GP are covered by planned commands

Objective: maximize the sum of gpRewards for all GP covered by plan: $\Sigma_g\ r_{g,c,t}\ y_{g,s,c,t}$

DCP vs. MILP: Model Size and Performance

Problem #	# Vars	#Constraints	Time to best sol (* = optimal)	Time to prove optimal	Makespan (# commands in plan)
1					
MILP	26,959	914 K	*** 156 s**	156 s	403
DCP	1,705		7 s		449
2					
MILP	89,118	14.4 M	*** 5 h**	16 h	195
DCP	900		5 s		254
3					
MILP	122,675	16.8 M	*** 13 h**	38 h	295
DCP	1,468		8 s		377
4					
MILP	244,207	20.8 M	*** 10.7 h**	10.8 h	1,468
DCP	7,464		1.5 m		1,656
5					
MILP	244,363	20 M	*** 24 h**	45.2 h	1,473
DCP	7,527		2.5 m		1,636
6					
MILP	1,856,154	**90 M**	***DNF***	***DNF***	***DNF***
DCP	8,701		28 m		6,104

- MILP could not solve 6-hour plan horizon within 50 hours, but DCP solves it in 28 minutes.
- MILP requires many more vars and constraints
- MILP makespan is always smaller

Objective: Maximize reduction of GP model error

- **gpReward(g,c,t)** = reward for viewing GP g with command *c* at time *t*

  = $r_{g,c,t}$ = $e_{g,t} - m_{c,b}$ (1)

  where $e_{g,t}$ = *prediction* err for g at time t

  $m_{c,b}$ = *measurement* error for command c in biome type b

- **cmdReward**(s,c,t) = sum of *gpRewards* for all GP observed by sat *s* using cmd *c* at time *t*

  = $\Sigma_{\forall g \in v_{s,c,t}} r_{g,c,t}$ (2)

  where $v_{s,c,t}$ = set of GP visible by sat s using cmd c at time t

- **Objective:** **Maximize** $\Sigma_{\forall c_n \in P}\, cmdReward(c_n)$ (3)

  max sum of all gpRewards for all GP covered by all commands in plan P

Identical metrics for DCP/MILP comparison: Equations 1, 2 and 3

Conclusion

Dynamic Constraint Processing (DCP)	Mixed Integer Linear Programming (MILP)
Pro • Speed • Better search control • Flexible modeling • Domain heuristics • Explainable AI	**Pro** • Provably optimal solutions • Relies on 3rd party solver (benefit of robust, heavily tested tool)
Con • Suboptimal solutions • Subject to local minima and path dependencies	**Con** • Slow • Limited model flexibility • Difficult to include domain specific heuristics • Solver variability • Declarative model requires pre-enumerating 10's of millions of constraints in model • Variable domains, constraints must be quantitative

Planner Input and Output

Inputs: TP and GP choices

Outputs: Plans for each satellite

Satellite 2 Timepoint (TP) choices:

Satellite 1 Timepoint (TP) choices:

Command choices for each TP

- Command search space = choices for every TP when a sat can observe GP
- 1 file for each satellite

Command choice examples:

L.34 = <L-band, angle 34> ,

P.32 = <P-band, angle 32>

TP (time)	Command choices	GP covered by choice
1311:	L.32:	[3165]
	L.34:	[3445, 3446]
	P.33:	[3165]
	L.32 & P.32:	[3165]

Ground Position (GP) choices:

- Choices for when & how to view each GP
- Science-value search space
- Measurement error depends on GP biome-type (shrub, forest, baren)
- One file for whole constellation

Choices for GP: 3165

Sat	TP (time)	Cmd Choices	Pred. Error	Meas. Error
1	1311	L.32	.008	.038
1	1311	P.33	.008	.017
1	1311	L.32 & P.32	.008	.010
2	1259	L.33	.042	.028
2	1259	P.33	.042	.028

Satellite 2 Plan:

Satellite 1 Plan:

Time	Command
[2-4]	**P.48**
[5-14]	Idle
[15-17]	**L.48**
[18-36]	Idle
[37-40]	Slew
[41-43]	**L.44**

Constraint Handlers

All constraints are enforced via choice propagation (forward consistency checking)

- Implemented by `propagateChoices(variable, value)`

Duplicate observation constraint handler examples:

Example 1: After observing GP 123, remove it from all future var domains (for all sats)

[x_{25}^{1}: {~~L.32: [123]~~,
L.33: [436349, 436350, 436351]]

Example 2: Removing GP results in empty variable domain, so remove the variable

~~[x_{36}^{1}: {P.42: [253]}]~~

- Empty variable domain means no observation can be taken at that time (unlike pure CSP)

Image Lock and **Slew time constraints** remove variables for infeasible observation times