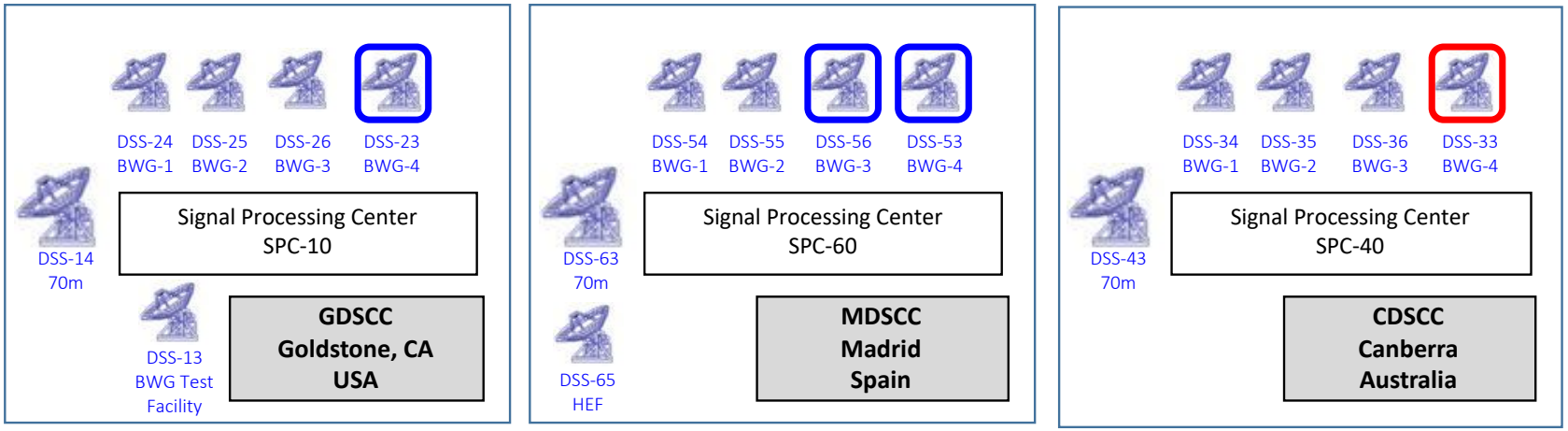
User Preference Optimization for Oversubscribed Scheduling of NASA's Deep Space Network

Jet Propulsion Laboratory
California Institute of Technology

Mark D. Johnston, JPL/CalTech

IWPSS 2019

DSN Overview



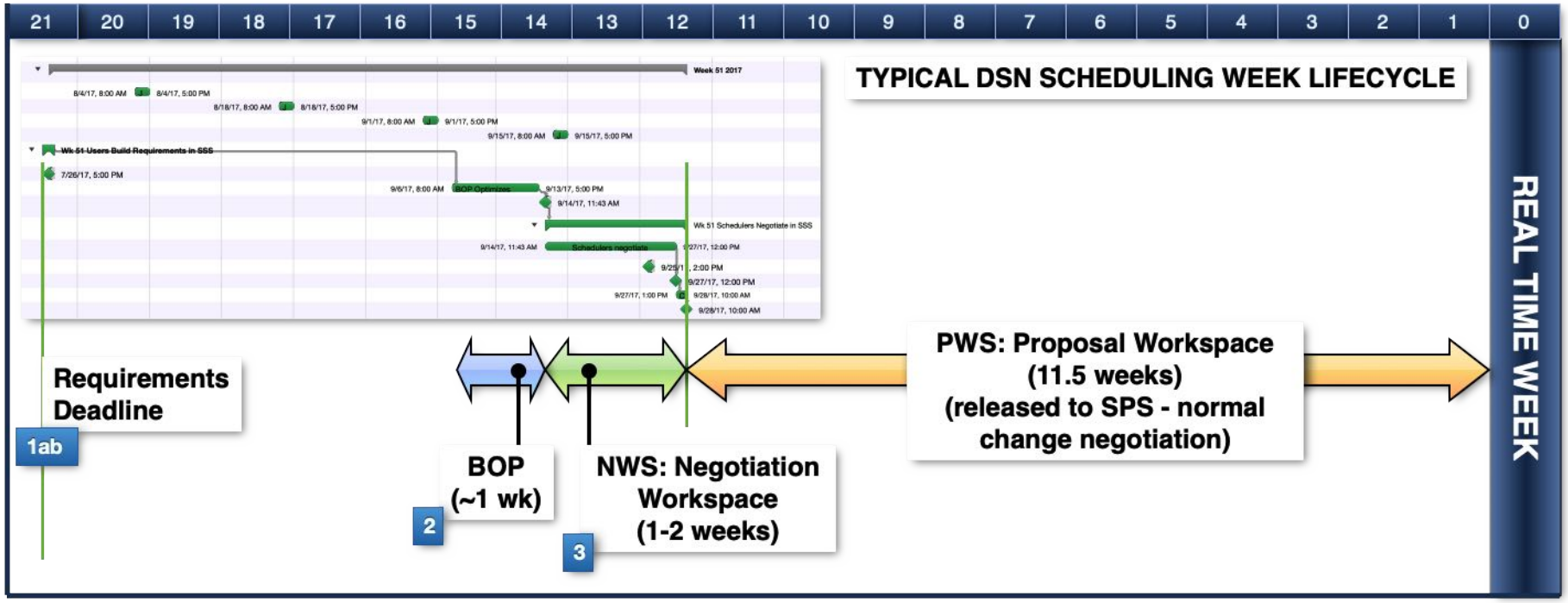
New Antennas Not Completed Yet

80-kW BWG

20-kW BWG

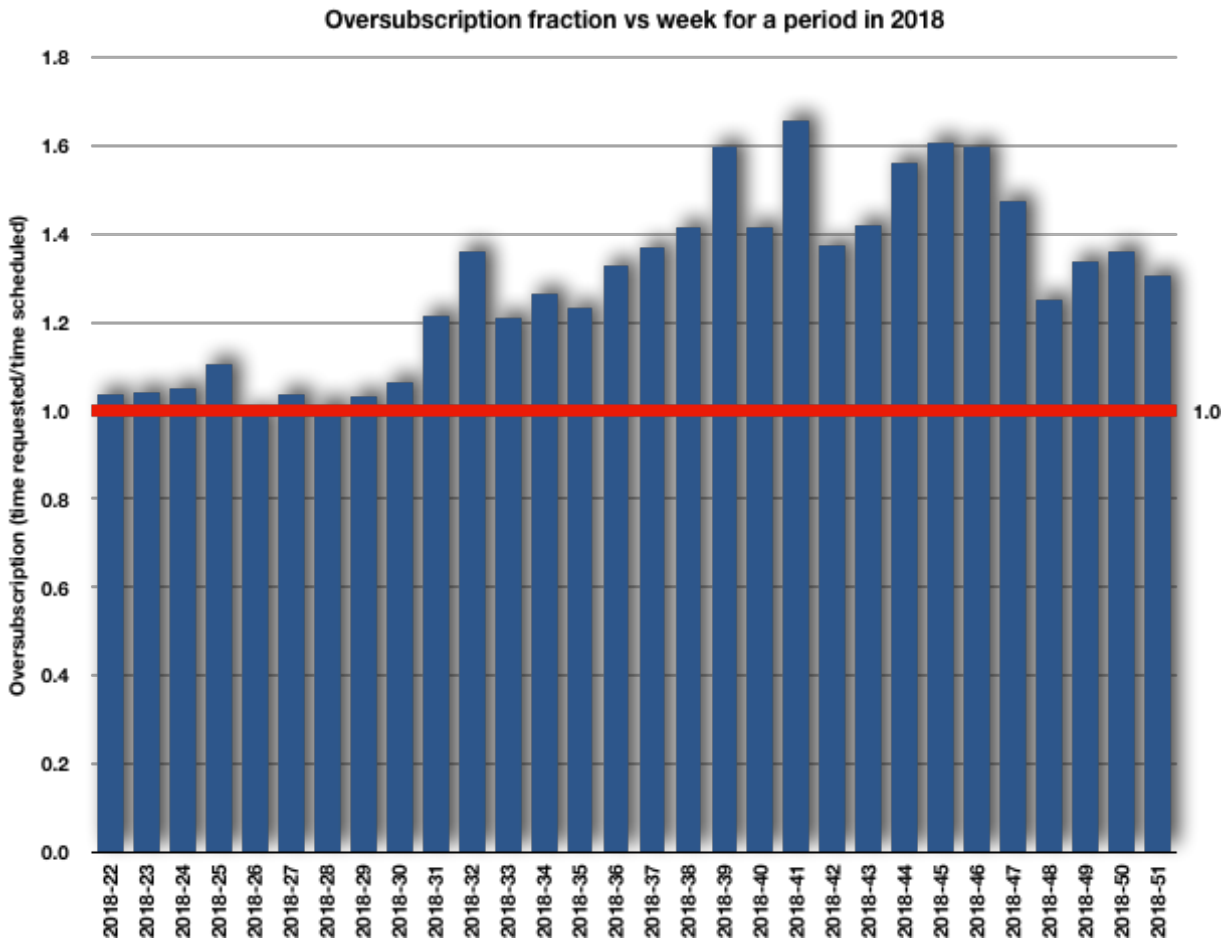
DSN Scheduling – The Process

- DSN scheduling is done on a rolling weekly basis so that baseline schedules are available ~3-4 months before execution
- Process is consensus-based with peer-to-peer collaboration and negotiation mediated by a web application, Service Scheduling Software, or S^3



DSN Oversubscription Has a Wide Variation

Ranges from near zero to over 60% at times



Priority/Preference Scheduling

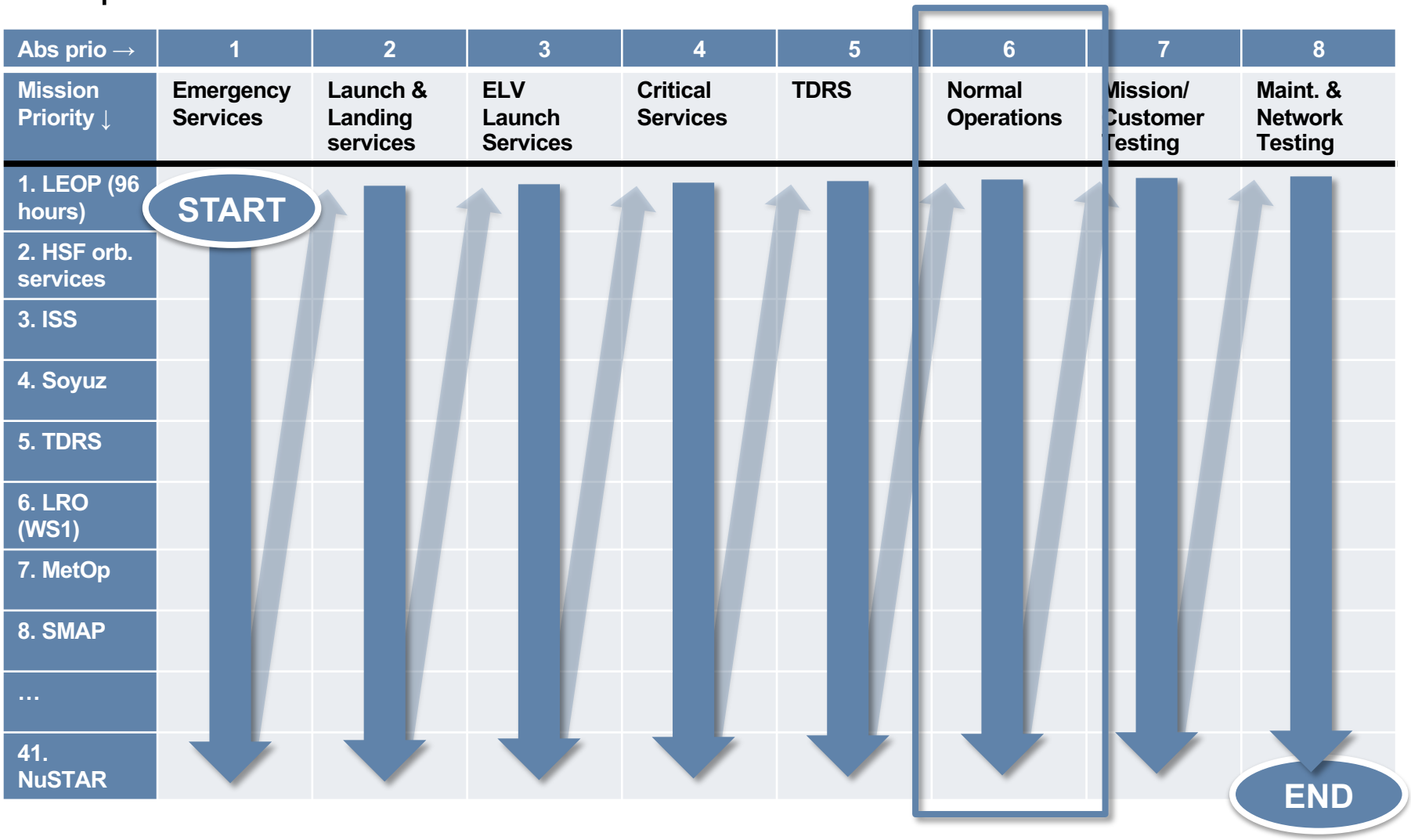
- DSN has been tasked to develop a priority scheduling system, motivated by desire to accomplish:
  - Reduction in effort and time to prepare and publish baseline schedules
  - Reduction in manual conflict reduction before negotiation starts
  - More consistency in number of negotiated weeks, to support mission planning and sequencing cycles
  - Consistency with other NASA SCaN networks — NEN & SN (Near Earth Network and Space Network)

What do the other SCaN Networks do?

- Both NEN and SN use a mission priority scheme, but there are big differences from the DSN:
  - NEN and SN both operate on a much shorter scheduling time frame than DSN: 3 weeks
  - NEN and SN are both entirely scheduled by a central org
    - full schedules are not published – only per-mission subsets directly to authorized users (driven by security considerations)
    - Changes are made via the central org only
- For priorities, both NEN and SN use a similar “2-dimensional” approach, with two priority lists
  - “Absolute Priority List” – relative importance based on mission activity category
  - “Mission Priority List” – rank ordered list of missions based on relative programmatic and mission priorities
  - Goddard Space Flight Center has a process for updating these lists - with NASA HQ as final authority

NEN/SN Priority Scheduling

2-D priorities



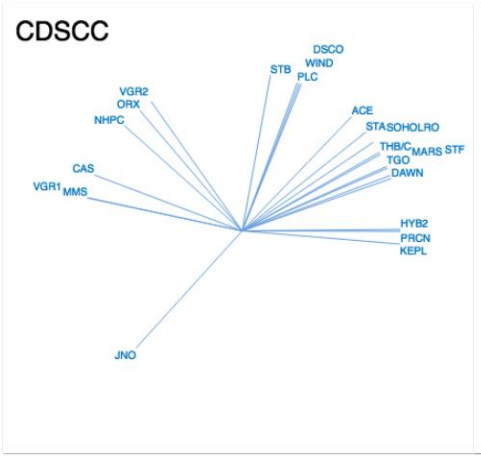
Considerations for DSN

- DSN does *not* have a mission priority list or equivalent
- Additionally, there are major drawbacks to a static mission priority list for DSN:
  - Given observed levels of DSN demand, a *strict* mission priority scheme for DSN (like NEN/SN) could leave missions at the bottom of the list completely out of the schedule
  - Many DSN users have *requirements that change* with mission phase or with planned science activities, week to week or day to day – not reflected well by a static list
  - Given the *variable mix of activities* and mission phase updates, a mission priority list would have to change so frequently as to be essentially useless

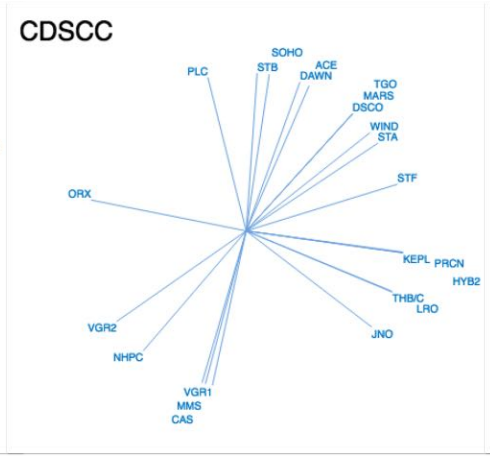
Viewperiod Alignment

- For some sample weeks:

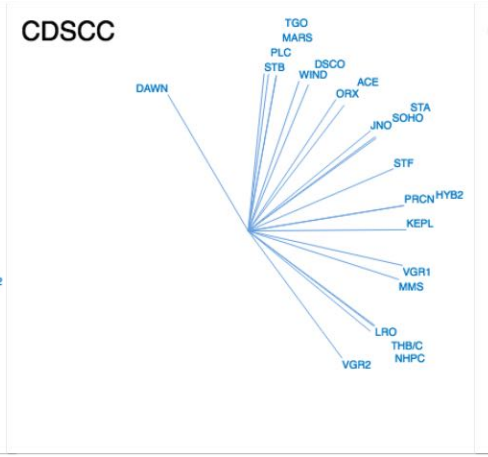
2017 week 13



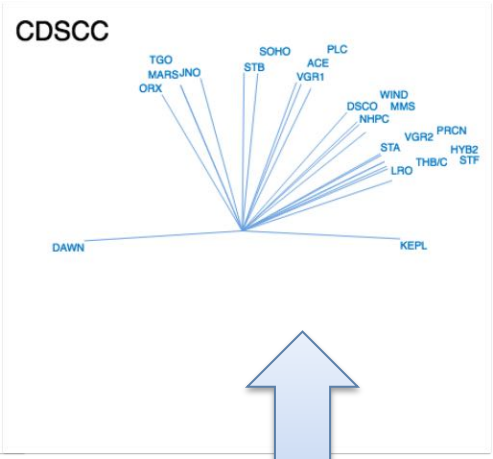
26



39



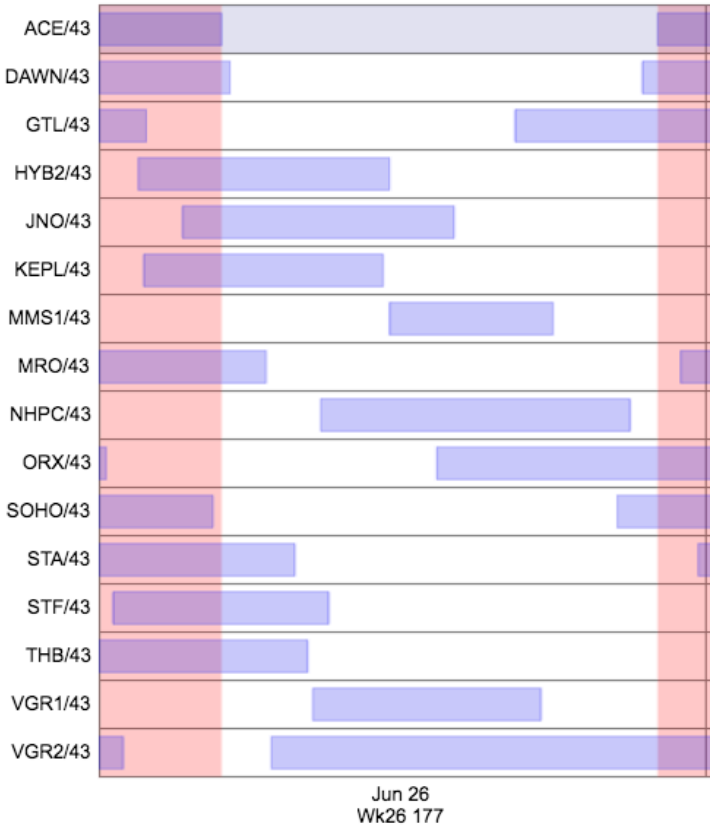
51

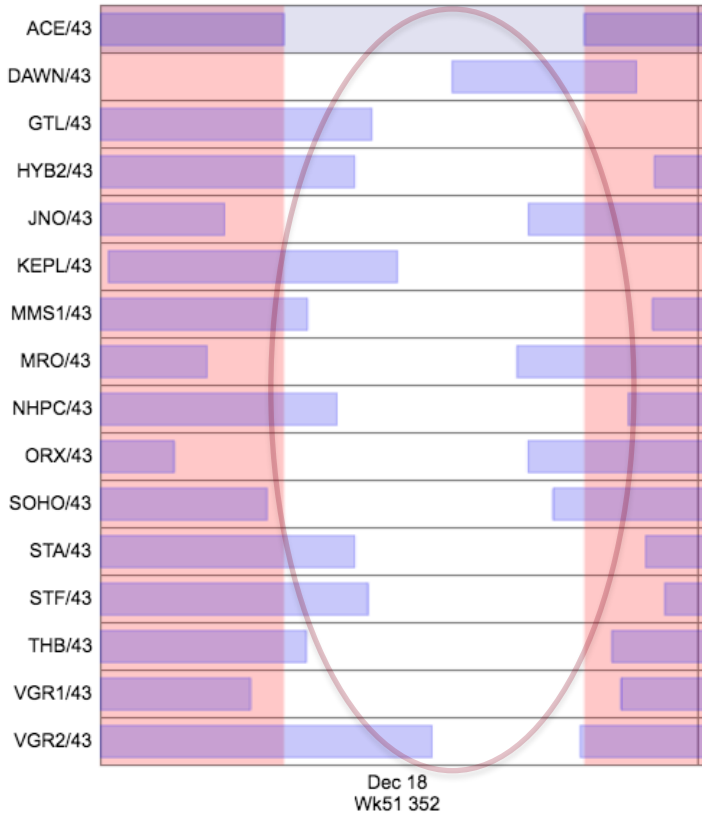


Heavy concentration of missions in the same part of the sky – exacerbates contention for antenna time

Viewperiod Alignment

- Illustration of viewperiod overlap
  - reference mission ACE (red band) on DSS-43
  - 2017 weeks 26 and 51

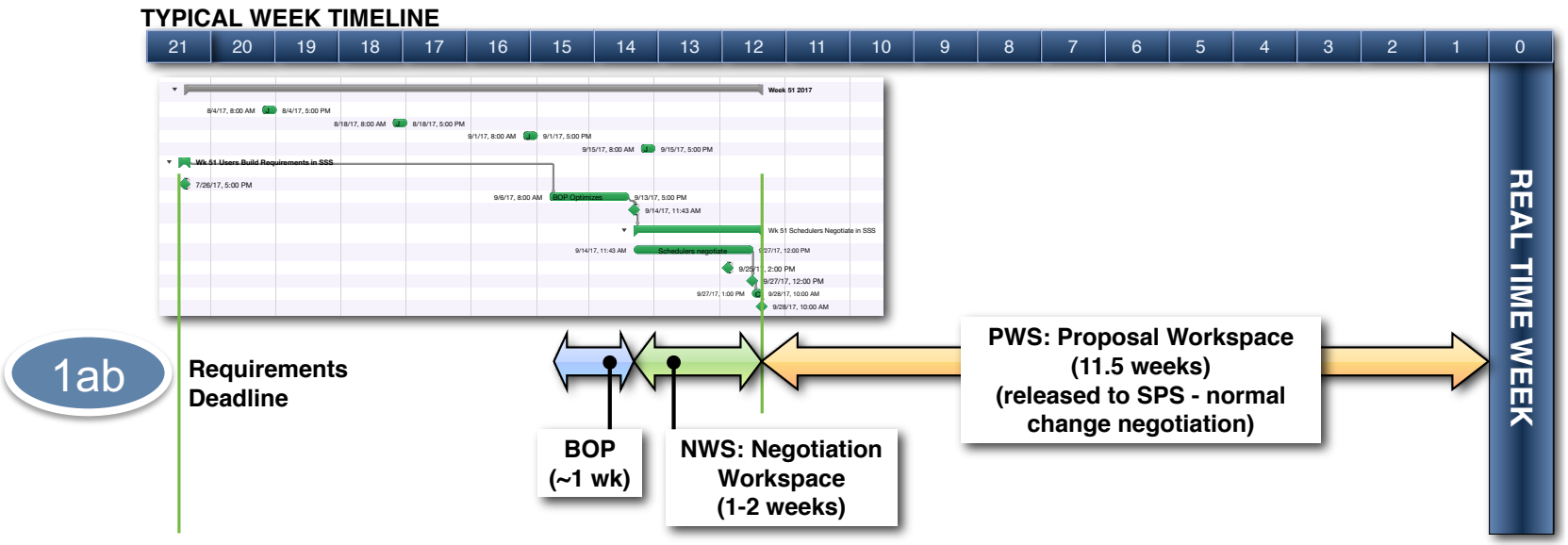




Alternatives with DSN Applicability

- Objective: infuse priority *and preference* considerations into SSS to reduce manual deconfliction effort, shorten the requirement-to-baseline negotiation timescale, while still allowing for peer-to-peer negotiation of the DSN schedule and flexibility for unforeseen occurrences
- How:
  1. Import supportability-limited “caps” per user into SSS
  2. Require users to specify (or default) requirement importance in terms of
     - Absolute “event” priority
     - User preference level
  3. Provide filtering tools to only consider highest priority/highest preference level requests within cap limits per user – i.e. at or near supportability limits
  4. Provide new auto scheduling algorithms (multiobjective/AI search) to optimize satisfaction of highest user preference level requests while deconflicting a workspace respecting event priority levels
  5. User negotiation is “up” from supportable levels rather than “down” from 20-50% oversubscription

Priorities/Preferences at Requirements Submission



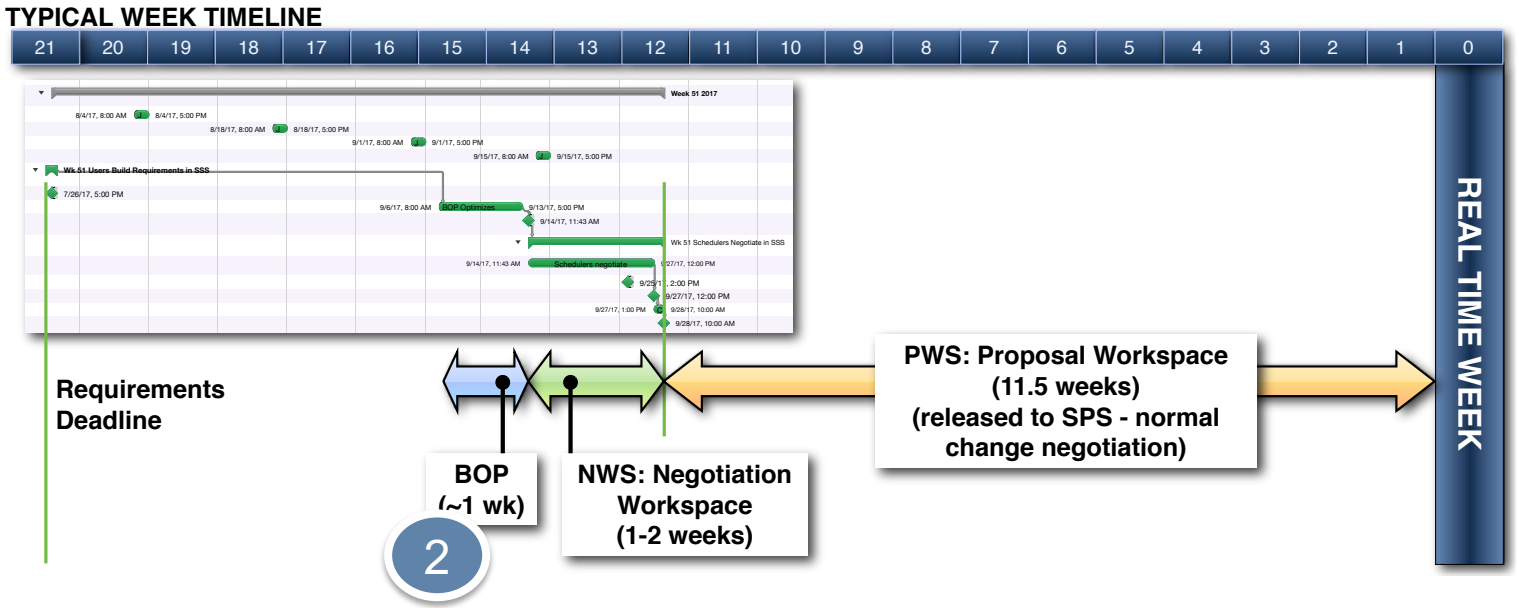
- At mid-range requirements submission:

1a – *Limit* each week's submitted requirements to a cap – calculated to be supportable

  - Based on Loading Analysis & Planning S/W (LAPS) integration of all user's planning requirements
  - Will require updated estimates of requirements + potential deconfliction if contention levels are too high

1b – Each mission/project *prioritizes their own inputs into tiers* reflecting their relative importance

Priorities/Preferences for Manual Deconfliction

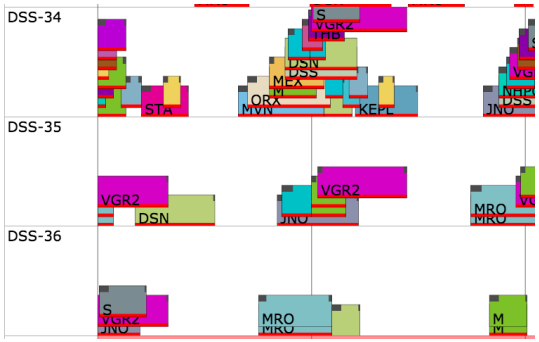


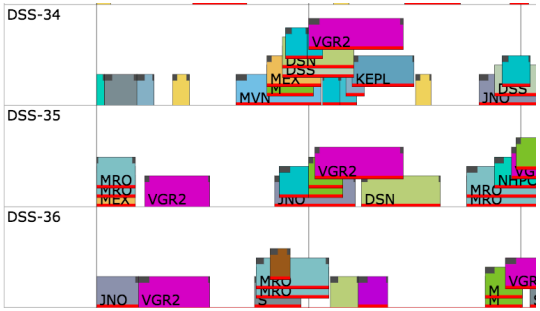
- At manual deconfliction time:
  - Run *preference optimization algorithms* to maximize priority/preference of scheduled activities per mission/project that can be fit into the schedule *without conflicts*
    - Use all provided flexibility (tracking duration, splitting across DSCCs, min/max gaps, etc)
    - Minor manual adjustments still needed, but much reduced compared to current process

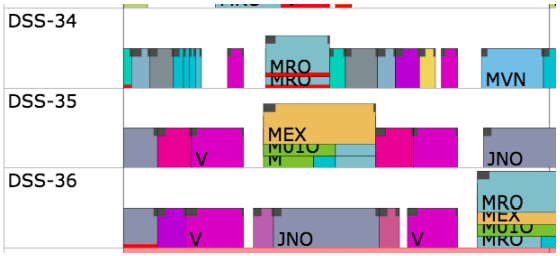
2

What does manual deconfliction involve?

1. **Create a 1-week workspace** containing all pushed requirements and (candidate) tracks from all users
   - these were generated mostly independently (except Mars), and are generally in high contention with each other
2. **Run the scheduling engine suite of strategies** to layout tracks, deconflict, and spread out conflicts to the greatest extent possible, *given user specified constraints and flexibilities* on duration, gaps, splittable specifications, antenna options, etc.
3. **Interactively edit the schedule** to shift, move, split, shrink, delete and otherwise manipulate tracks to reduce conflicts while meeting the "intent" of user requirements and taking into account special events

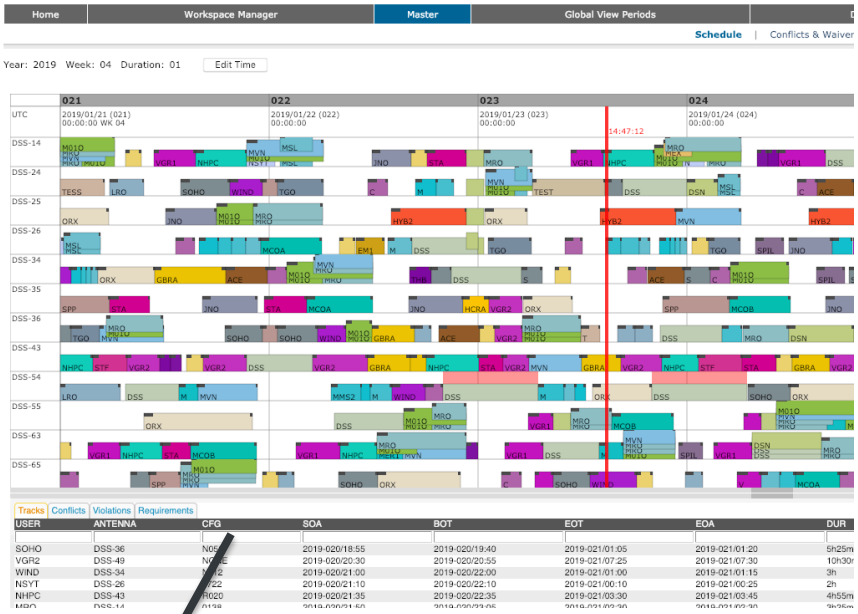






Using Limits in the Schedule view: W/S or Master

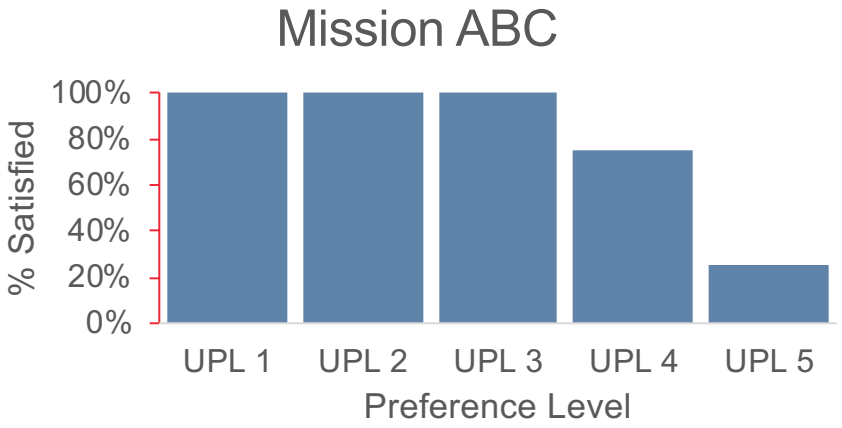
- Add new tab “Limits” after “Requirements” in schedule view
- Include grid with columns:
  - Year-wk, user, trk limit, trk total in w/s, over limit (e.g. red if over), trk limits by priority, trk limits by preference
- Year-Week and User entries are hyperlinks that jump to Request Editor Limit Tracking with selectors pre-set



					Workpace by Priority Level							Workpace by Preference Level				
Year-Week	User	w/s total	Limit	Overage:	1	2	3	4	5	6	7	1	2	3	4	5
2019-04	ACE	23.00	26.00	-3.00	20.0	3.0	0.0	0.0	0.0	0.0	0.0	20.0	3.0	0.0	0.0	0.0
2019-04	CHDR	21.00	21.00	0.00	21.0	0.0	0.0	0.0	0.0	0.0	0.0	21.0	0.0	0.0	0.0	0.0
2019-04	DSCO	1.00	1.00	0.00	1.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0
2019-04	DAWN	122.00	106.00	16.00	56.0	32.0	20.0	14.0	0.0	0.0	0.0	56.0	32.0	20.0	14.0	0.0

Metrics (cont.)

- Conflict count continues to be a key driving metric for SSS: the DSN process requires conflicts be cleared or waived at some gate, managed by SPO office
- Requirement violation count will also continue to be a key driving metric for scheduling engine operations:
  - All else being equal, changes that do not increase violations, while reducing conflicts, are preferred when possible
- New metrics can take advantage of additional information related to user preference levels:
  - Per-user fraction of total requested time by preference level



New Metrics (cont.)

- Overall user preference level satisfaction, cumulative to level i

$$S_i = \frac{1}{N}\sqrt{\sum_{users}\left(\frac{T_{Si}}{T_{Ri}}\right)^2}$$

where

S_i = (cumulative) satisfaction of preference level ≤ i

T_{Si} = time scheduled at preference level ≤ i

T_{Ri} = time requested at preference level ≤ i

N = # users

Optimal value is 1 at each level; range is [0,1]

New Metrics (cont.)

- Satisfied event priority (EP) request fraction at level p

$$S_{EP=p} = \frac{\sum T_{S,EP=p}}{\sum T_{R,EP=p}}$$

where

$S_{EP=p}$ = satisfied fraction of event priority $EP = p$

$T_{S,EP=p}$ = time scheduled at event priority $EP = p$

$T_{R,EP=p}$ = time requested at event priority $EP = p$

Best value is 1 at each level; range is [0,1]

New Metrics (cont.)

- Overall antenna utilization (as fraction of allocable* time)
  - SOA-EOA, i.e. non-gap time in the schedule
  - Caveat: maximizing this is correlated with maximizing setup and teardown, e.g. by superfluous splitting, and is therefore not a key optimization criterion
- Overall tracking duration (as fraction of allocable* time)
  - BOT-EOT
  - Due to allocation to different preference levels, this (scalar) is not a key optimization criterion
    - E.g. for two schedules with the same amount of scheduled tracking time, one corresponding to lower preference levels would be a worse schedule

* “Allocable” in this context means activity time that is free to be scheduled within a workspace; this could be work category determined to excluded “fixed” downtime or other activities

Priority/Preference - Algorithm Use

- Initial design for incorporating user-specified priorities/preferences completed and documented – to enable User Preference Optimization (UPO)
- Algorithm trade study and assessment in progress through summer
  - Constraint-based conflict-directed AI search
  - Squeaky-wheel optimization (SWO)
  - Multi-objective optimization

Conclusions

- The described new paradigm for DSN scheduling addresses several key objectives and has many advantages
  - Reduction in manual schedule editing effort to reduce conflicts prior to schedule negotiation
  - Reduction in negotiation effort and time to clear conflicts and baseline the schedule
  - More uniformity in the number of negotiated weeks, to support extended (multi-month) mission planning and sequencing cycles
  - Consistency with the other NASA SCaN networks (the Near-Earth Network and Space Network) in using priorities as a means to generate and deconflict schedules from a disparate set of user requirements The approach we have described above has several compelling aspects:
  - Users are asked for minimal additional information over what is already provided for scheduling
  - The thorny issue of assigning relative priorities to missions is avoided
  - The oversubscription of DSN resources is managed at the initial submission gate, thus speeding all downstream processes by not having to address what to reduce – this decision being left in the hands of the users who are best able to judge relative priorities of their own inputs

While there remains significant work to go to validate and fully implement this approach, it should provide a major quantitative reduction in effort and cost in the DSN scheduling process.