An Introduction to *Sked*
or
How to *Sked*ule a VLBI Session

John Gipson

NVI, Inc./GSFC

2010 IVS General Meeting

Hobart, Australia

2010 February 7-10

Sked is used to/in:

1. Schedule sessions
2. List schedules
3. Check schedules
4. Analyze schedules
5. Correlator setup
6. Simulation studies
7. Determine media requirements
8. …

Sked is used to/in:

I will focus mostly on 1.

1. **Schedule sessions**

2. List schedules

3. Check schedules

4. Analyze schedules

5. Correlator setup

6. Simulation studies

7. Determine media requirements.

8. …

I hope to cover…

- My goals for sked

- History: Inception to Date

- Where does *sked* fit in?

- Master command

- Downtime

- Manual and Automatic Scheduling

- Overview of Automatic scheduling algorithm

- Generating an R1 schedule

- Fill In mode

- New sky coverage calculation

- Conclusion/Recap

1. Bug free.

2. User friendly

3. Less user input required

    A. Less input required→ fewer chances of error.

4. Make *sked* smart, so that you don't have to be.

5. Better schedules:

    A. More observations.

    B. Better sky coverage.

    C. Less idle time.

6. Better tools for analyzing schedules.

History		
1978	**Basic program created (Nancy Vandenberg)**	
	command line input	
	manual selection of scans	
	catalogs for sources, stations, equipment	
1981	Automatic calculation of antenna motion and tape handling	
1988	**Automatic selection of observations (Autosked)**	
	Optimization by strict covariance.	
1992	Evaluation of schedules using SOLVE simulations	
	Creation of pseudo-databases to evaluate formal errors.	
1993	**Autosked merged into standard version.**	
	"Strange" schedules	
1995	**Beginning of rule based schedules.**	
1996	Mark IV/VLBA recording mode support added	
	Last time sked documentation updated.	
1997	Numerous changes	
	Support for VEX files	
	Y2K fixes,	
	new Java-based catalog interface	
	S2 and K4 support	

History (Cont)	
2002	**John Gipson takes over development/maintenance** Fill-in mode Best-N Source Selection
2004	**Linux port by Alexey Melnikov** Beginning of death of HP-sked Astrometric option Specify min, max observing targets for set of sources.
2005	Full support of Disk-based recording: Mark5A, Mark5B, KK5
2006	Downtime: Ability to specify when an antenna is unavailable.
2007	Resurrection of covariance optimization. Found and fixed various bugs in algorithms. Still not used routinely. By-product: sked can predict formal errors internally.
2008	Master command Read session setup from master file. Check session against master file.
2009	Station limit raised from 32$\rightarrow$64, and made parameter.

History (Cont)	
2010	Now— Update documentation *(only 14 years out of date!)* Prepare for VLBI2010 **What else do you want?**

Contents of Schedule File

- General information about experiment:
- Sources
    - Positions
    - Flux models, positions
- Station positions
- Antenna information
    - Horizon mask, slew rates, etc.
- Equipment
    - Receivers
    - Recorders
- Recording mode setup
- Schedule information: Source, stations, durations
- Date generated
- Sked version used.
- Scheduling parameters
    - SNR targets, maximum scan lengths
    - Procedure timing
    - Recording mode: stop& stop, continuous, adaptive
    - Major options
    - Minor options

This list is not complete!

Everything required to run the session

Debugging info

Everything required to generate the schedule

Master File (Experiment, date, stations)

Sked file from similar session

New sked file

Catalogs:
Sources
Stations
Equipment

sked

Scheduler:
- •Selects:
    - •Sources
    - •Stations
    - •Recording Mode
- •Sets
    - •Scheduling parameters
    - •Optimization parameters

Master File (Experiment, date, stations)

Sked file from similar session

Catalogs:
Sources
Stations
Equipment

sked

New sked file

CDDIS

Correlator

Stations

When the schedule is complete, the scheduler puts it on CDDIS, where it is picked up stations and the correlators.

It is also available on the session web page for future reference.

1. Initial setup:

   A. Station selection—usually obtained from master file.

   B. Mode selection—frequencies, BW, 1-2 bit, etc. (set by session type.)

   C. Start, stop times—usually from masterfile.

   D. Source selection—manual or use "Best N"

   E. Flux update—needed because flux models change.

   F. SNR targets.

   G. Downtime—this is when one or more stations are unavailable.

2. Setup Major and Minor options. (Usually don't change).

3. Schedule scans, either "by hand" or automatically.

4. Review schedule.

5. Iterate if necessary.

6. Save the schedule.

Before 2008, users would set up a schedule by doing the following:

1. Start with a similar schedule: `cp r1412.skd r1413.skd`
2. For the new schedule, determine the following from the masterfile:
   A. Start time;
   B. Stop time;
   C. Stations;
3. Modify the new schedule accordingly.
4. Change the $EXPER parameter in the schedule.

Before 2008, users would set up a schedule by doing the following:

1. Start with a similar schedule: `cp r1412.skd r1413.skd`
2. For the new schedule, determine the following from the masterfile:
    A. Start time;
    B. Stop time;
    C. Stations;
3. Modify the new schedule accordingly.
4. Change the $EXPER parameter in the schedule.

There were several schedules with incorrect stations or start and stop times.

The Master command checks the schedule against the masterfile.

It uses the $EXPER parameter in the *sked* file as a key.

There are two modes:

1. **Master check**: Compares *sked* file against master, and reports discrepancies.
2. **Master get:** Reads the master file to determine a) start and stop time; b) stations; c) Correlator; d) Scheduler. Puts this in the schedule file.

This makes generating a schedule much **easier, and reduces the chance of error.**

Frequently stations are unavailable for part of a session. The most common reason is that the station drops out to **participate in an Intensive.**

In the past, schedulers had to "stop" and "start" the station manually.
1. Generate schedule to start of downtime.
2. Remove station from subnet.
3. Generate schedule until just after end of downtime.
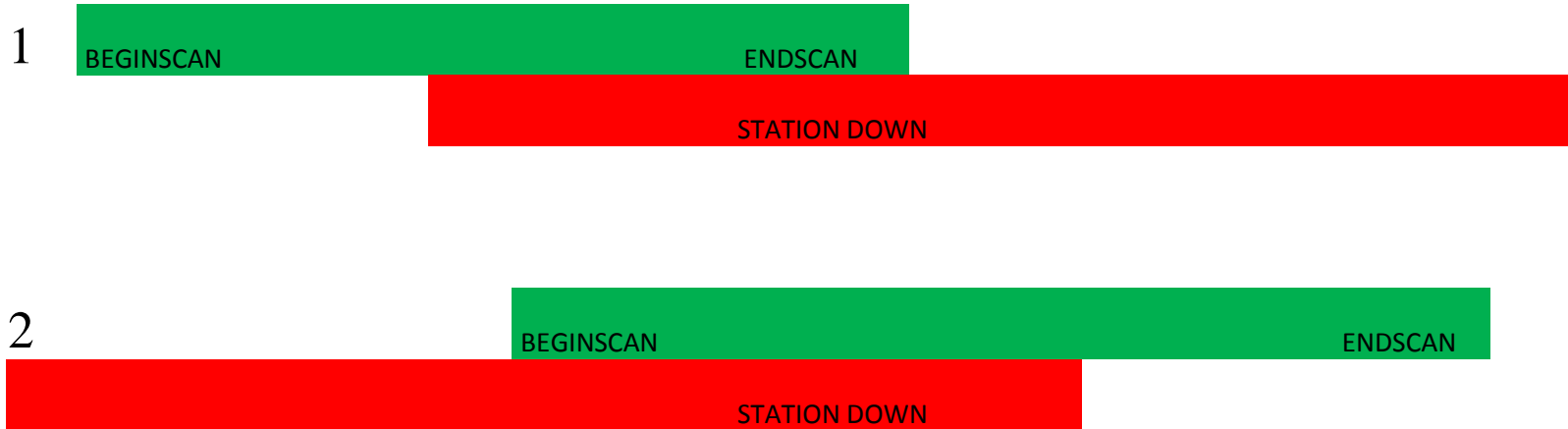4. Put station back in subnet.

The downtime setting makes this unnecessary.

The user **specifies which stations will be down, and for how long.**

Sked does the rest.

1

| BEGINSCAN | ENDSCAN |
| STATION DOWN |

2

| BEGINSCAN | ENDSCAN |
| STATION DOWN |

A scan is rejected because the station is down if:
1. **The end of the scan is in downtime**.
2. **The beginning of the scan is in downtime**

This worked fine for scheduling downtime due to the Intensives.

1

| BEGINSCAN | ENDSCAN |

STATION DOWN

2

BEGINSCAN ENDSCAN

STATION DOWN

3

BEGINSCAN ENDSCAN

STATION DOWN

A scan is rejected because the station is down if:
1. **The end of the scan is in downtime**.
2. **The beginning of the scan is in downtime**
3. **There is downtime in the middle of a scan:**

Two modes of scheduling:

1. **Manual mode.** User specifies everything about a scan.

2. **Automatic mode.** Sked determines best scan based on heuristic rules. These rules are a distillation of what works.

These modes can be intermixed. Sked can run in auto-mode for a while, then scheduler can insert scans by hand, and return to auto-mode.

User specifies everything about the scan. General form:

```
/ src_name [start time] [sub Subnet] [dur duration]
```

A user can specify scans that cannot be observed.

src_name is required. The terms in brackets are optional. If they are omitted, sked will try to schedule a scan with the maximum number of stations meets the SNR targets, as soon as possible after the current time. If it detects problems (doesn't meet SNR targets, a station is unavailable, etc.,) *sked* will notify the user and ask if they want to proceed.

Example:
```
/ 3c84 start 17:30:00 sub NyWzWt
```

A typical R1 has about 900 scans→900 commands if scheduling manually.

In automatic mode, the user instructs *sked* to schedule a network until some endtime.

```
? Auto subnet endtime
```

Example:

```
Auto KkWfNyMa 19:30:00
```

Generate scans with Kokee, Westford, NyAlesund, Matera until 19:30:00.

Example:

```
Auto _ End
```

Generate scans with full subnet from the current time to end of the session.

A single command in automatic mode **instead of 900 in manual mode.**

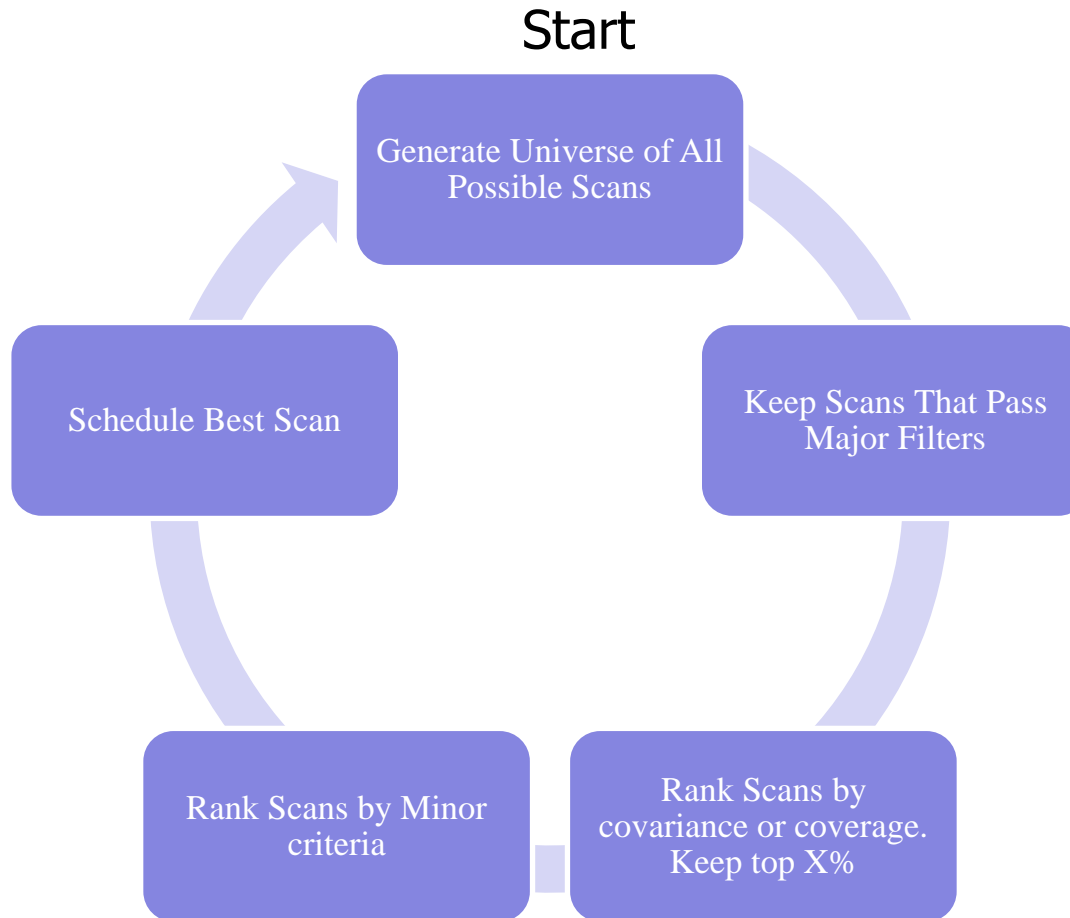If `endtime` is not specified, sked will schedule a single scan:

```
Auto _
```

In automatic mode, user specifies general parameters for scheduling the session. *Sked* generates and schedules the scans.

Two kinds of parameters (or options):
1. **Major options** determine what scans are considered for inclusion.
2. **Minor options** determine the *ranking* of the scans generated. The highest ranked scan is selected.
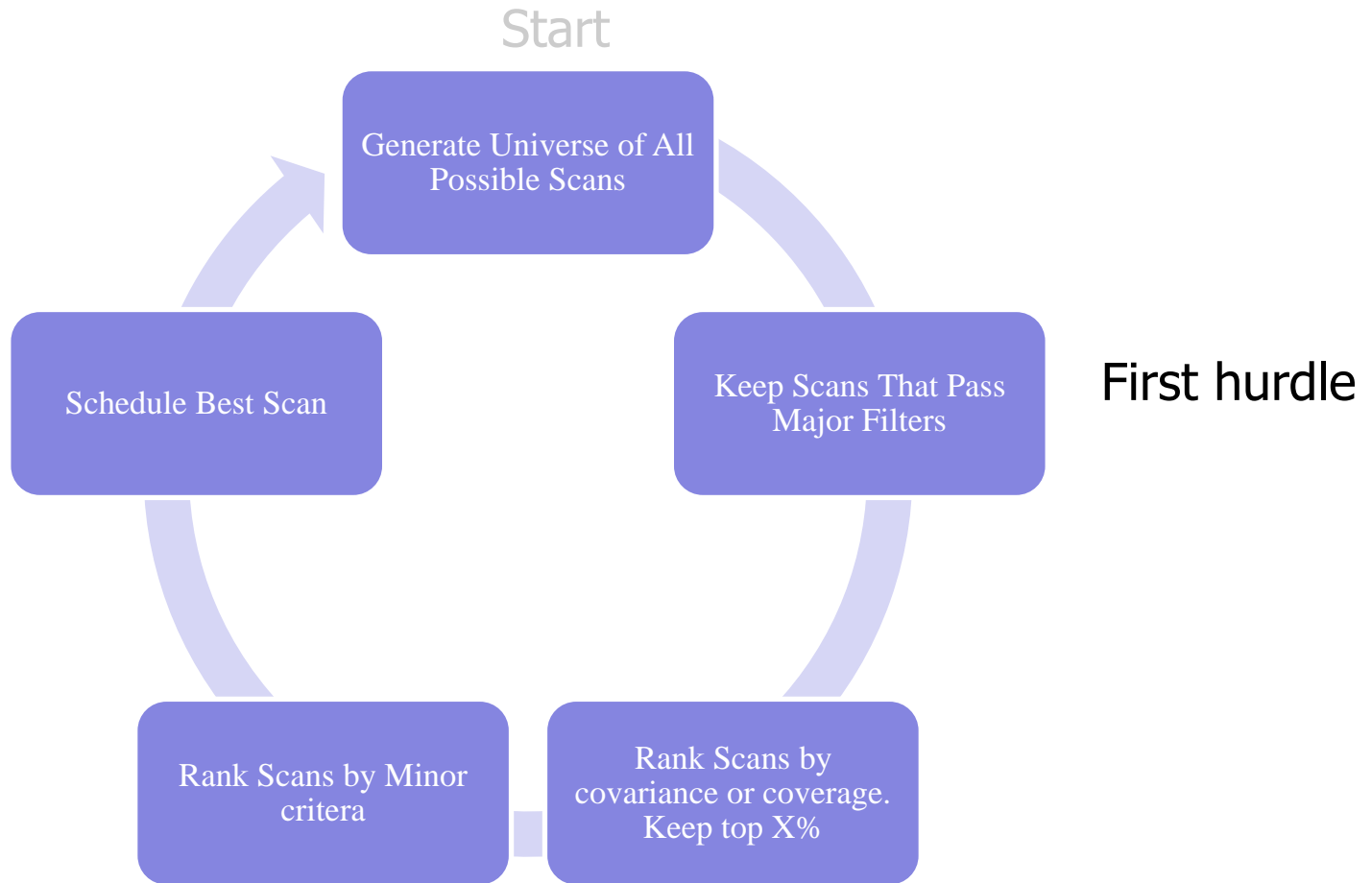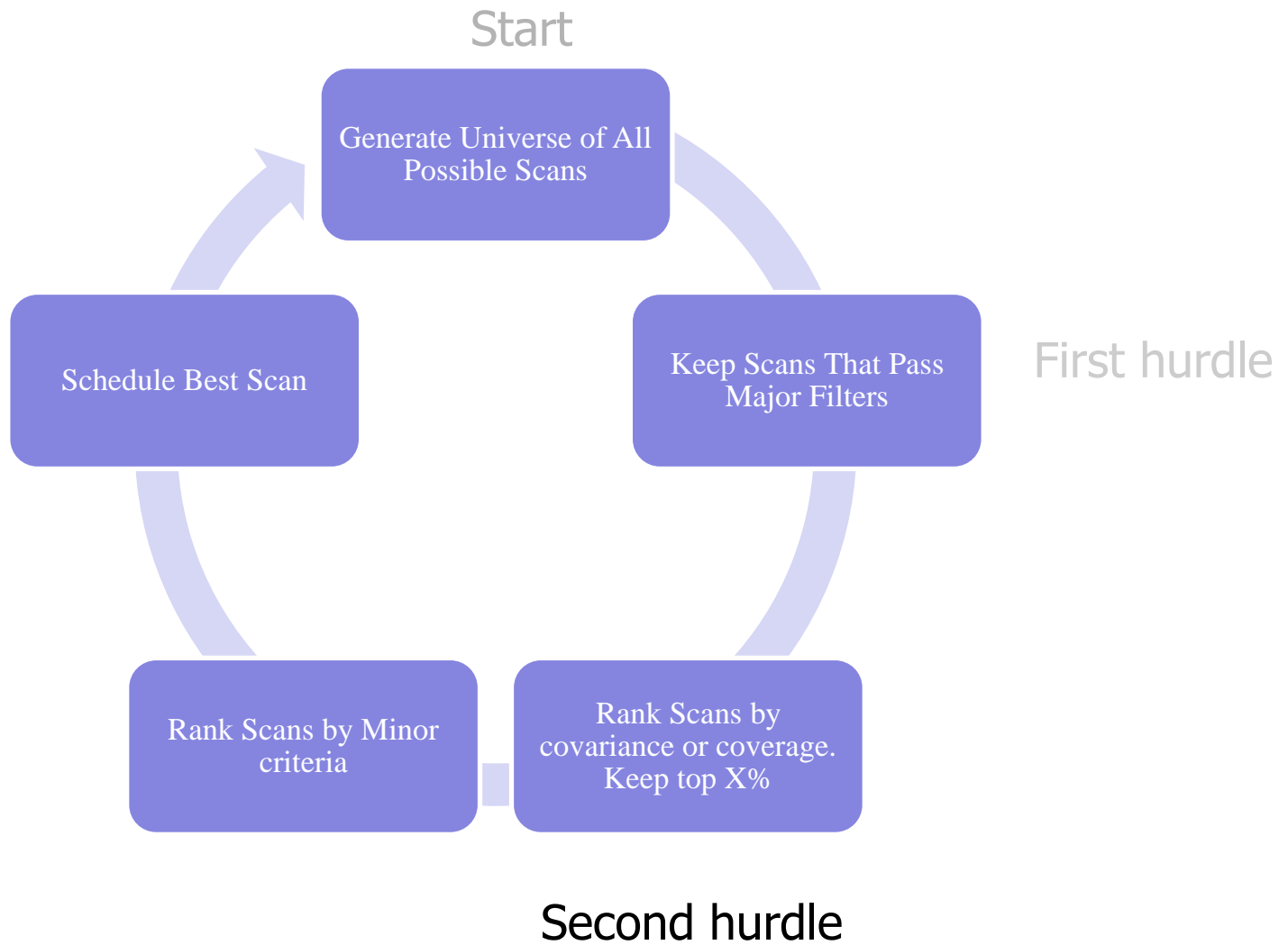
Start

- Generate Universe of All Possible Scans
- Keep Scans That Pass Major Filters
- Rank Scans by covariance or coverage. Keep top X%
- Rank Scans by Minor criteria
- Schedule Best Scan

Start

Generate Universe of All Possible Scans

Keep Scans That Pass Major Filters

First hurdle

Schedule Best Scan

Rank Scans by Minor critera

Rank Scans by covariance or coverage. Keep top X%

Start

Generate Universe of All Possible Scans

Keep Scans That Pass Major Filters

First hurdle

Rank Scans by covariance or coverage. Keep top X%

Rank Scans by Minor criteria

Schedule Best Scan

Find the best scan

Second hurdle

Start

Generate Universe of All Possible Scans

More time? Then continue!

First hurdle

Keep Scans That Pass Major Filters

Schedule Best Scan

Find the best scan

Rank Scans by Minor criteria

Rank Scans by covariance or coverage. Keep top X%

Second hurdle

All Possible Scans

Generate universe of all possible scans with:
1. Current network
2. Current sources
3. At Current time.

Source too close to Sun

Use Major filters and SNR targets to eliminate scans.

Scans that Pass Major Filters

Source too close to Sun

Source observed too recently

Scans that Pass Major Filters

Scan doesn't meet SNR targets on enough baselines

Use Major filters and SNR targets to eliminate scans.

Source too close to Sun
Source observed too recently
Source too close to previous source
Scans that Pass Major Filters
Subnet too small
Station takes too long to slew
Scan doesn't meet SNR targets on enough baselines

Use Major filters and SNR targets to eliminate scans.

In generating scan, start with largest network possible. If necessary, sequentially discard stations until meet targets.

Scans that Pass Major Filters

Scans that Pass Major Filters

Rank and throw away bottom

1. Rank sources by **sky-coverage** or by **covariance**.
2. Keep top X% (usually 25-50%) and throw away rest

A means of specifying the **properties** of a **good scan**. Some examples:

1. Number of observations
2. Time from last scan until this scan starts
3. Time from last scan until this scan starts
4. Presence of one or more stations
5. Presence of an astrometric source
6. LoEl scans
7. LoDec scans
8. Covariance
9. Sky Coverage
10. Minimizes idle time of stations
11. …
12. …

About 20 overall.

1. A minor option can be either ON or OFF.
2. The "ON" options are assigned weights by the scheduler.
3. For each minor option that is on, *sked* calculates a score.
4. The final score for each scan is the weighted sum.

$$Total\ Scan\ Score = \sum_{options} Wt_j Score_j$$

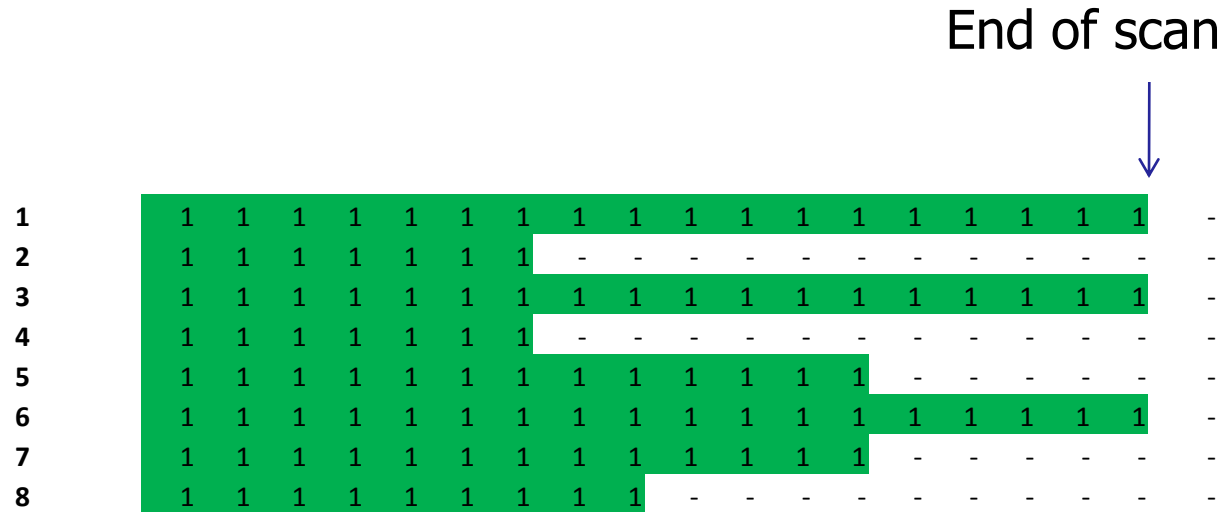**The "best scan" is the scan with the highest total score.**

This is the scan that is scheduled.

```
jmg/scedules>cp r1412.skd r1413.skd   Start with a similar schedule
jmg/schedules>sked r1413.skd          Open it in sked.
```
… sked displays info about schedule
```
? param exper r1413        Change the experiment key
? master get               Initialize the session using the masterfile
```
… sked displays information about the new session.

… sked prompts you if it finds potential problems.
```
? best 60                  Select the best sources for this session
```
… sked finds the best sources, and reports problems.
```
? Down kk-wz 18:00-20:00       Set downtime because of intensives.
? Auto _ end               Generate schedule until end of the session
```
…sked generates scans and schedules them, displaying the progress on the screen.
```
? Wr                       Write out the schedule
? Quit
```

End of scan

1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	-
2	1	1	1	1	1	1	1	-	-	-	-	-	-	-	-	-	-	-	-
3	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	-
4	1	1	1	1	1	1	1	-	-	-	-	-	-	-	-	-	-	-	-
5	1	1	1	1	1	1	1	1	1	1	1	1	1	-	-	-	-	-	-
6	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	-
7	1	1	1	1	1	1	1	1	1	1	1	1	1	-	-	-	-	-	-
8	1	1	1	1	1	1	1	1	1	-	-	-	-	-	-	-	-	-	-

By default, *sked* tries to schedule full net-work scans.
The next scan would start as soon after the end of the scan as possible.
This can leave a lot of time when stations are not observing.
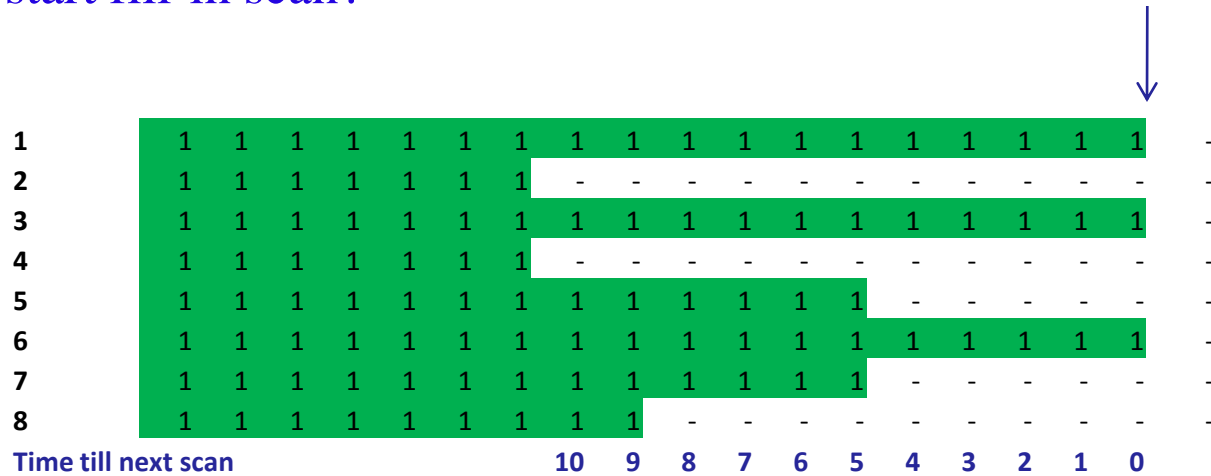"Fill In Mode" attempts to decrease idle time by filling in the gaps.

End of scan

When to start fill-in scan?

1	1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 -
2	1 1 1 1 1 1 1 - - - - - - - - - - - -
3	1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 -
4	1 1 1 1 1 1 1 - - - - - - - - - - - -
5	1 1 1 1 1 1 1 1 1 1 1 1 1 - - - - - - -
6	1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 -
7	1 1 1 1 1 1 1 1 1 1 1 1 1 - - - - - - -
8	1 1 1 1 1 1 1 1 1 - - - - - - - - - - -

Time till next scan   10 9 8 7 6 5 4 3 2 1 0

Strategy: Find possible starting times, and calculate how much idle time we could eliminate.

Idle Stations	# Stations	Time Left	Maximum Fill Time
2,4	2	10	20
2,4,8	**3**	**8**	**24**
2,4,5,7,8	5	4	20

2nd option maximizes possible fill time

1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	-
2	1	1	1	1	1	1	1	-	-	2	2	2	2	2	2	2	2	-	-
3	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	-
4	1	1	1	1	1	1	1	-	-	2	2	2	2	2	2	2	-	-	-
5	1	1	1	1	1	1	1	1	1	1	1	1	1	-	-	-	-	-	-
6	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	-
7	1	1	1	1	1	1	1	1	1	1	1	1	1	-	-	-	-	-	-
8	1	1	1	1	1	1	1	1	1	2	2	2	2	2	2	2	2	-	-

Look for possible scans, and schedule best one.

Allowable spillover

Repeat until all the idle time is filled.

Fill In mode results in:
1. More observations
2. Fewer large network scans
3. More smaller network scans

What does it mean to have good sky coverage?

Hard to quantify.

Here are two possibilities:

1. No large holes.
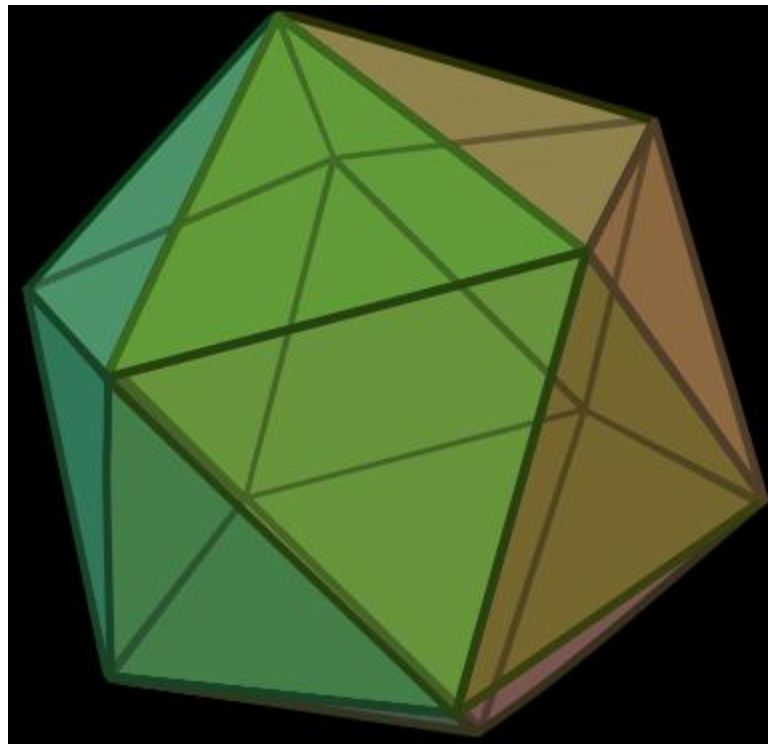2. Average distance of any point on the sky to an observation is small.

To calculate, need to "pixelate" the sphere.

Start with icosahedron.
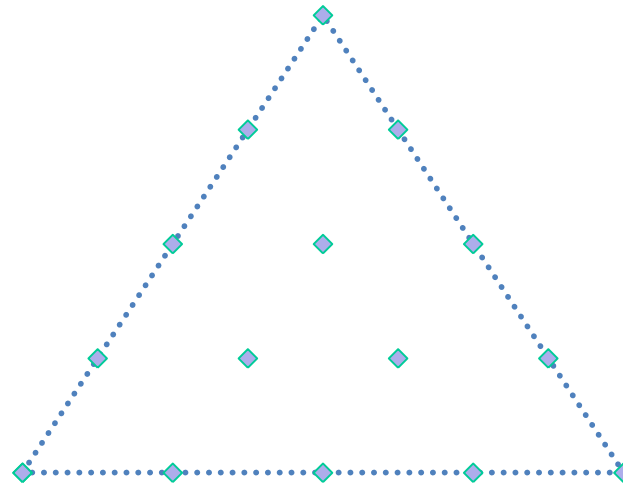
Pixelate each face.

1. Each point corresponds to pixel.
2. Pixels on sides are shared by two faces.
3. Pixels at the corners are shared by 5 faces.

Pixelating a Sphere

And keep dividing

J. Baumgardner (1985)
M. Tegmark (1996)
G. Hart (200)
and others…
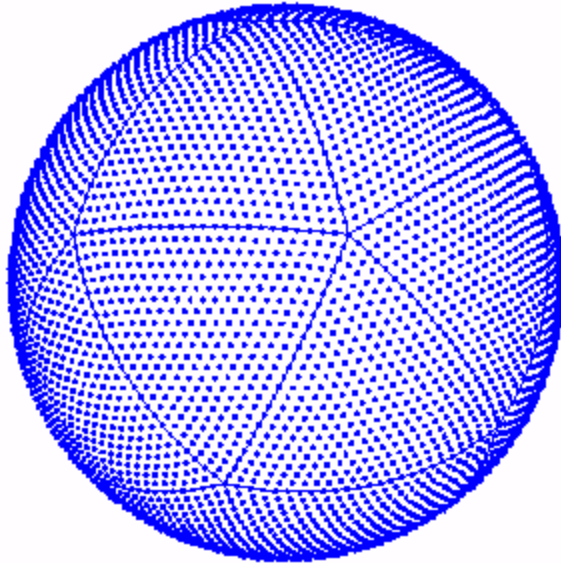
Number of pixels=40N(N-1)+12

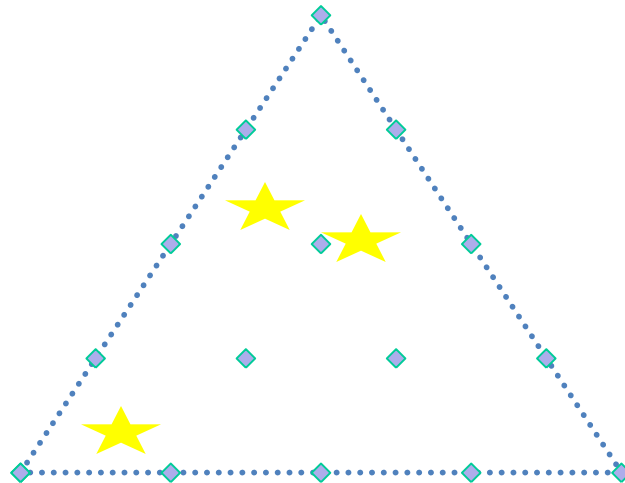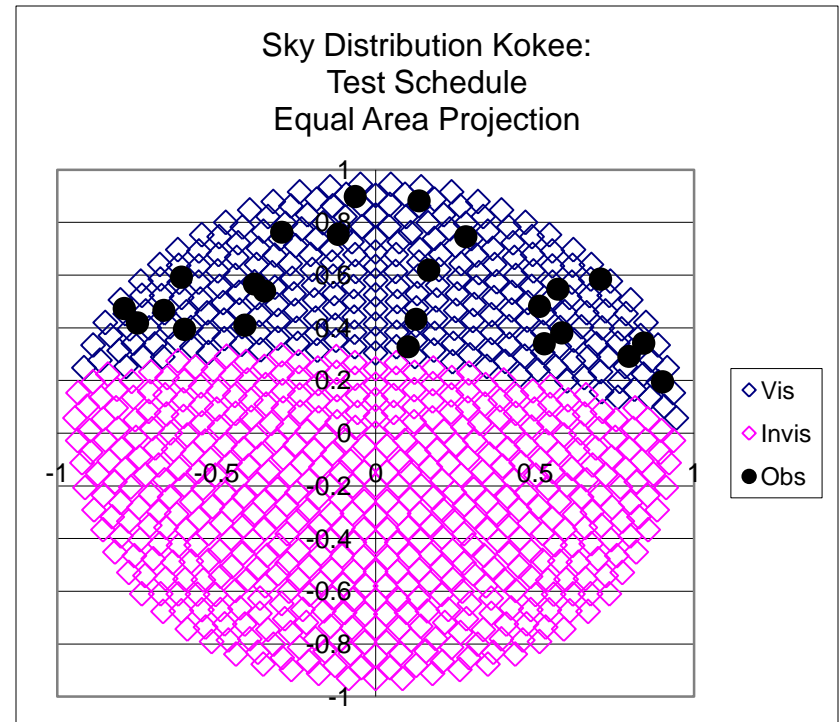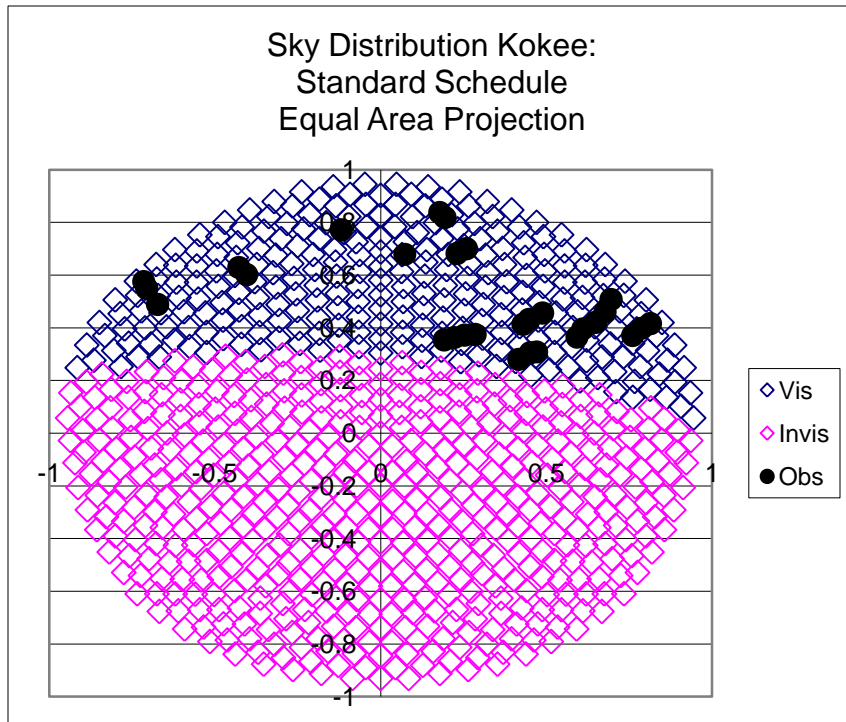| N	Number Pixels	Pixel Radius (Deg)
4	492	5.2
5	812	4.0
6	1212	3.3
7	1692	2.8
8	2252	2.4
9	2892	2.1
10	3612	1.9

For a given schedule:
1. For each pixel, find the distance to the nearest observation. Call this Dist_j
2. Find the average value of Dist_j over all pixels. This is the average distance of any point on the sky to an observation.
3. Find the largest value of Dist_j. This is the largest hole in the sky.

Sky Distribution Kokee:
Standard Schedule
Equal Area Projection

Sky Distribution Kokee:
Test Schedule
Equal Area Projection

	Average	Max Dist
Standard Intensive	10.9	32.6
Test Intensive	8.2	22.7

Conclusions

1. *Sked* is an essential tool for geodetic VLBI.

2. My goals are:

    A.   No known bugs!

    B.   Make it more user friendly.

    C.   Protect users from making errors.

    D.   Better schedules.

3. *Sked* continues to evolve.

4. User input is critical to its continued improvement.