SALT scheduling software

Brent Miszalski

SALT Research Fellow

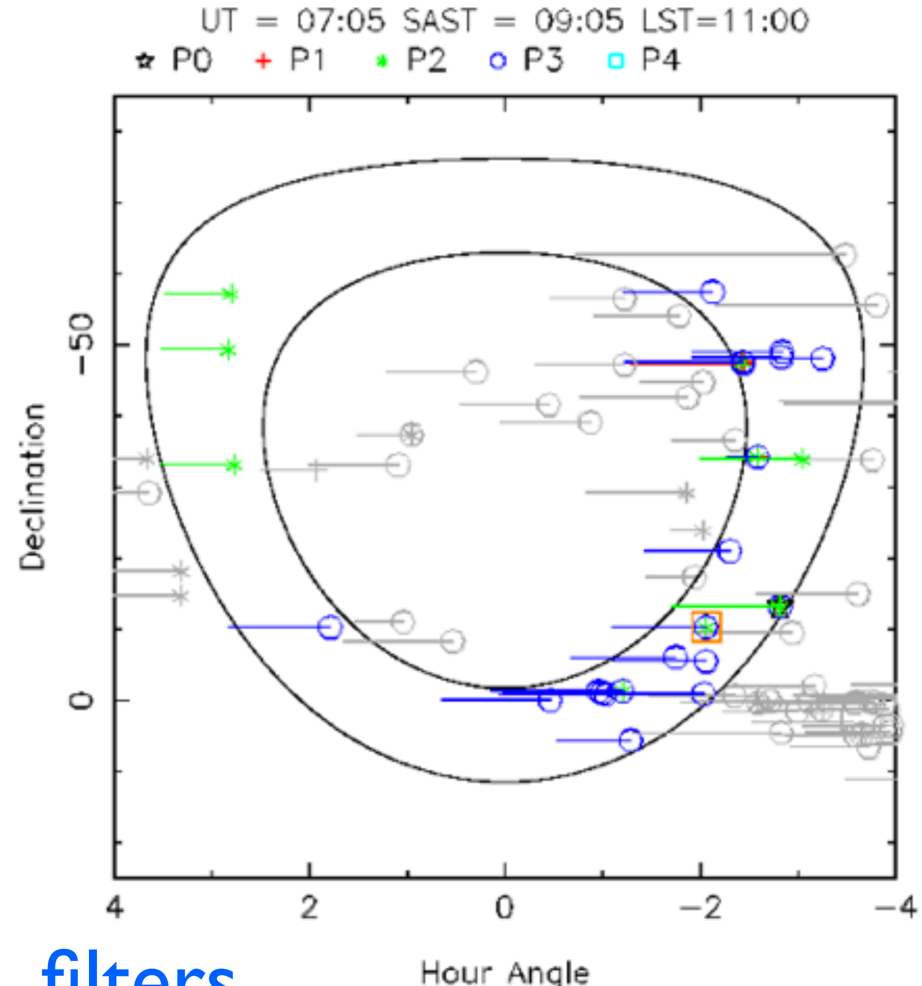
brent@saao.ac.za

National Research Foundation

SAAO South African Astronomical Observatory

HA tool

N(Targets)	503
Proposal Code	2012-1- [submit]
PI (surname)	[submit]
Moon	Any [submit]
Transparency	Any [submit]
Seeing (min)	0 [submit]
Seeing (max)	10 [submit]
Priority	0,1,2,3 [submit]
Manual	SAST 9.09 HA min 4 max -4 Dec min 25 max -85 [REPLOT]
SAST controls	[-1h] [-0.2h] [+0.2h] [+1h]
Windowing	x1 4 x2 -4 y1 25 y2 -85 [REPLOT] [WEST] [EAST] [EXTRAGALACTIC] [MAGELLANIC] [DEFAULT]
Reset all	[RESET ALL]



interactive web page

live science database queries

filters similar to OPT

<-----targets move to left with increasing time

block length lines include acquisition

time critical targets are in pink and time windows are *not* yet considered

Loaded MOS masks have RSS mode in GREEN, missing ones are in RED.

Objects < 25 deg from Moon are not showed (if not time critical) and MD = Moon Distance (deg

Multiple-visit objects now have date last obs (YYYY-MM-DD) and wait time.

CCAS blocked objects now have orange boxes and **CCAS** warning on this page (azimuth=54.5-65

Wiki page

Moon: 01:47 +14:12 (99%)

P0

Select	Proposal	PI	RSS	SCAM	WM	ObsTime	East (SAST)	West (SAST)	Target	Coords	Visits	Seeing	Moon	MD	Transparency	Progress

P1

Select	Proposal	PI	RSS	SCAM	WM	ObsTime	East (SAST)	West (SAST)	Target	Coords	Visits	Seeing	Moon	MD	Transparency	Progress
10454	2012-1-GU-001	Husser		SLOT	10454	2415	10:20 to ----	---- to 14:18	GK Vir	14:15:36.41 +01:17:18.2	0 / 1	2.5	Grey	163.0	Thin Clouds	P1: 0.0% [0/14200] P2: 19.1% [5030/26400] P3: 6.1% [5230/86320]
10464	2012-1-GU-001	Husser		SLOT	10464	2715	09:36 to ----	---- to 13:44	WD 1333+005	13:36:16.03 +00:17:32.6	0 / 1	2.0	Dark	165.2	Thin Clouds	P1: 0.0% [0/14200] P2: 19.1% [5030/26400] P3: 6.1% [5230/86320]
14230	2012-1-UNC_RSA_RU-002	Kannappan	FP		14230	3328	10:42 to ----	---- to 14:20	rs1403	14:27:24.46 +03:04:20.3	0 / 1	1.5	Bright	160.0	Thin Clouds	P1: 76.3% [9741/12772] P2: 43.4% [6656/15340] P3: 0.0% [0/19709]

P2

Select	Proposal	PI	RSS	SCAM	WM	ObsTime	East (SAST)	West (SAST)	Target	Coords	Visits	Seeing	Moon	MD	Transparency	Progress
10455	2012-1-GU-001	Husser		SLOT	10455	2415	10:20 to ----	---- to 14:18	GK Vir	14:15:36.41 +01:17:18.2	0 / 2	2.5	Grey	163.0	Thin Clouds	P1: 0.0% [0/14200] P2: 19.1% [5030/26400] P3: 6.1% [5230/86320]
10457	2012-1-GU-001	Husser		SLOT	10457	2415	08:05 to ----	---- to 12:29	SDSS J121258.25-012310.1	12:12:58.25 -01:23:10.2	0 / 1	2.5	Grey	153.4	Thin Clouds	P1: 0.0% [0/14200] P2: 19.1% [5030/26400] P3: 6.1% [5230/86320]
10465	2012-1-GU-001	Husser		SLOT	10465	2715	09:36 to ----	---- to 13:44	WD 1333+005	13:36:16.03 +00:17:32.6	0 / 1	2.0	Dark	165.2	Thin Clouds	P1: 0.0% [0/14200] P2: 19.1% [5030/26400] P3: 6.1% [5230/86320]
11580	2012-1-IUCAA-004	Bhattacharya	LS PG900		11580	3259	10:14 to 11:31	16:18 to 17:35	GRB 110715A	15:50:44.09 -46:14:06.5	0 / 3	1.2	Dark	138.8	Photometric	P1: 0.0% [0/13140] P2: 0.0% [0/19692] P3: 0.0% [0/13968]
11818	2012-1-GU-004	Kollatschny	LS PG900		11818	1970	04:32 to 05:39	10:35 to 11:42	ESO374-G25	10:03:23.62 -37:23:45.2	0 / 1	4.0	Dark	125.5	Any	P1: 48.6% [1970/4050] P2: 0.0% [0/6700] P3: 0.0% [0/22700]

P3

posal	PI	RSS	SCAM	WM	ObsTime	East (SAST)	West (SAST)	Target	Coords	Visits	Seeing	Moon	MD	Transparency	Progress
2-1-GU-001	Husser		SLOT	10456	2415	10:20 to ----	---- to 14:18	GK Vir	14:15:36.41 +01:17:18.2	0 / 5	2.5	Grey	163.0	Thin Clouds	P1: 0.0% [0/14200] P2: 19.1% [5030/26 P3: 6.1% [5230/863
2-1-GU-001	Husser		SLOT	10458	2415	08:05 to ----	---- to 12:29	SDSS J121258.25-012310.1	12:12:58.25 -01:23:10.2	0 / 5	2.5	Grey	153.4	Thin Clouds	P1: 0.0% [0/14200] P2: 19.1% [5030/26 P3: 6.1% [5230/863
2-1-GU-001	Husser		SLOT	10466	2715	09:36 to ----	---- to 13:44	WD 1333+005	13:36:16.03 +00:17:32.6	0 / 5	2.0	Dark	165.2	Thin Clouds	P1: 0.0% [0/14200] P2: 19.1% [5030/26 P3: 6.1% [5230/863
2-1-RSA_OTH-019	Woudt	LS PG900		10670	4007	05:54 to 07:11	09:54 to 11:11	CSS090331:102843-081927	10:28:43.00 -08:19:27.0	0 / 1	2.0	Grey	131.1	Thin Clouds	P3: 36.5% [8835/24
2-1-IUCAA-002	Srianand	LS PG900		10873	3103	09:17 to ----	---- to 13:34	J1321-0041	13:21:39.86 -00:41:51.9	1 / 3 Last: 2012-05-11 Wait: 1 days	1.5	Grey	165.1	Photometric	P3: 41.3% [48387/1

SALT scheduling software requirements

- Create a basic queue based on observing conditions and input date
  - Prefill with P0 (P1?) and time critical blocks
  - Populate with other blocks that minimise idle time
  - Modular - plug in optimal scheduling algorithm
  - Planning - reuse code to simulate a semester
- Good level of block display and interaction

requirements (cont)

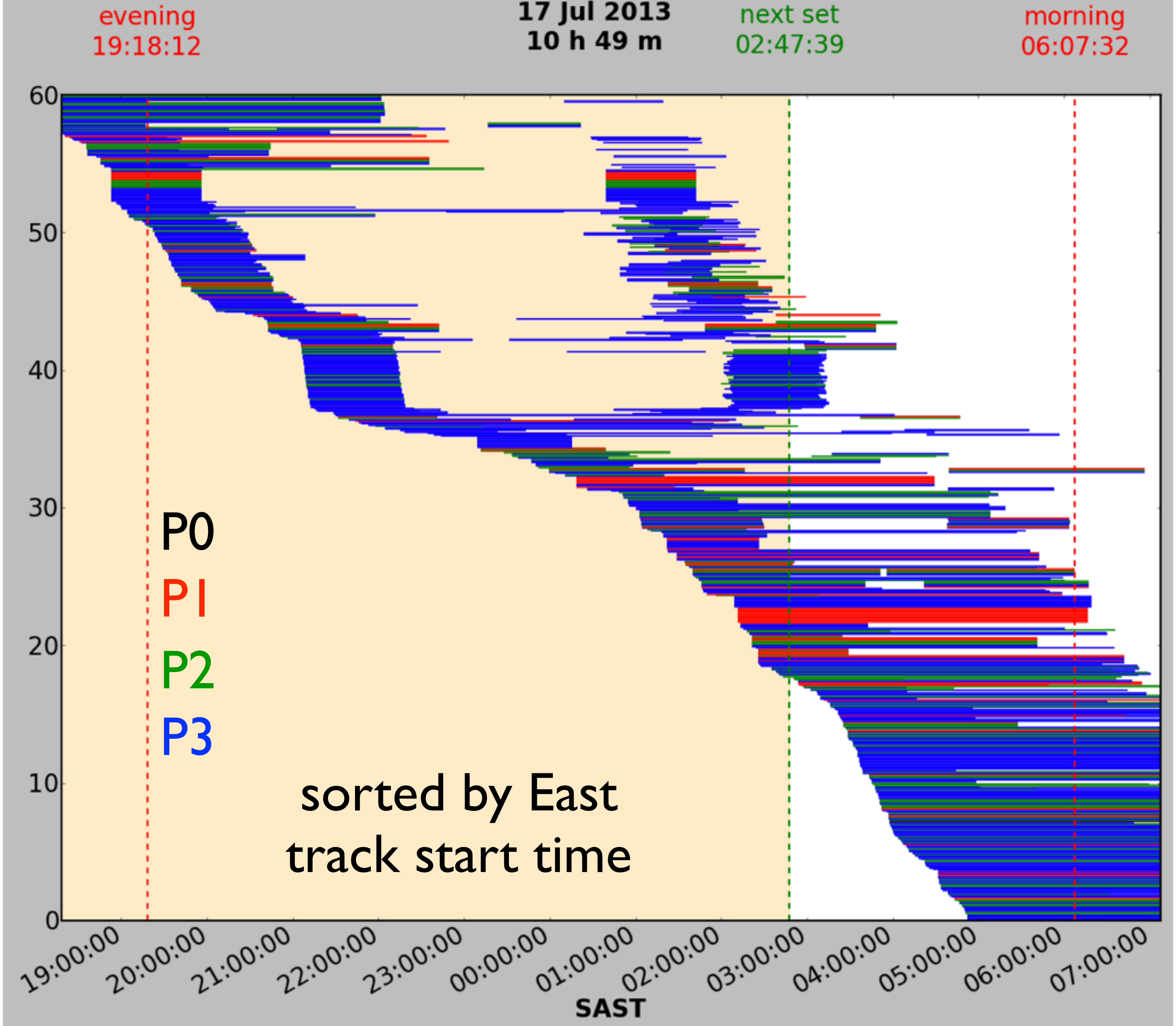
- Address a number of issues not easily done now
  - Be aware of moon illum fraction and moon rise/set during night
  - If time critical or P0 or ToO, ignore moon
  - If proposal is time critical and one block already observed, bump up in priority or highlight it
  - Go to blocks at right time (need to know best windows, crucial for MOS, equatorial tracks)
  - Aware of calibration time AFTER science in blocks

Work done so far

matplotlib

PyEphem

- Test application written to read in block info (SQL query dumped to FITS table)
- Aim: display blocks using native Python date format and matplotlib
- PyEphem module used for twilight times, moon rise/set/illumfrac at Sutherland
- Times stored in Python DateTime format


evening
19:18:12
17 Jul 2013
10 h 49 m
next set
02:47:39
morning
06:07:32
60
50
40
30
20
10
0
P0
P1
P2
P3
sorted by East
track start time
19:00:00
20:00:00
21:00:00
22:00:00
23:00:00
00:00:00
01:00:00
02:00:00
03:00:00
04:00:00
05:00:00
06:00:00
07:00:00
SAST

Progress on Scheduler

- Now at point where we can start to implement a basic queue! hooray!
- Progress:
  - Matplotlib display of queue for a night!
  - SubBlock python class functional
  - Moon handled properly now (except lunar illum calc is a bit off)
- N.B. Code is still very much in a simple state and not all bells and whistles have been added => Focus is to get main stuff working

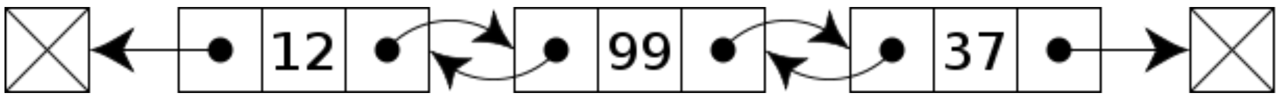
Sub Block

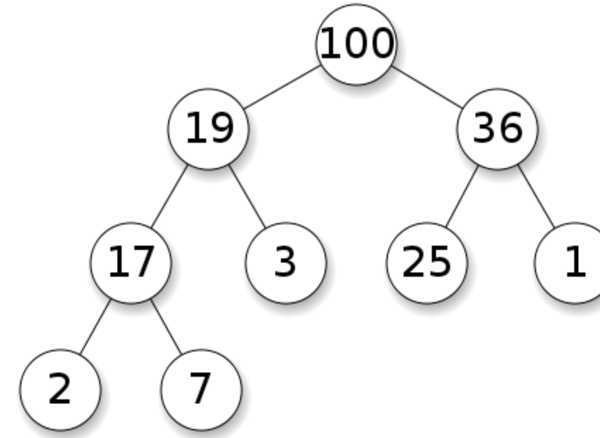
- Initially tried to have a Block class that included all time windows (east AND west track), but this became way too complicated (different moons encountered by different tracks)
- Simplified to SubBlock - only defined by a single time window (e.g. a window you can point to object and have enough track to complete the ObsTime)
- Things now work very well! Though in queue will have to keep track of other subblocks with same BlockID (can use references to other SubBlocks)

Queue me up

- Current test program just has a simple list of P1,P2, P3 blocks
  - Activate/Choose a random subblock and then go through all other subblocks
  - If they don't overlap with already active blocks, activate new subblock
  - Display queue in matplotlib
- Immediate next steps:
  - Get a basic optimisation going (e.g. simulated annealing) and explore randomisation steps

Queue class

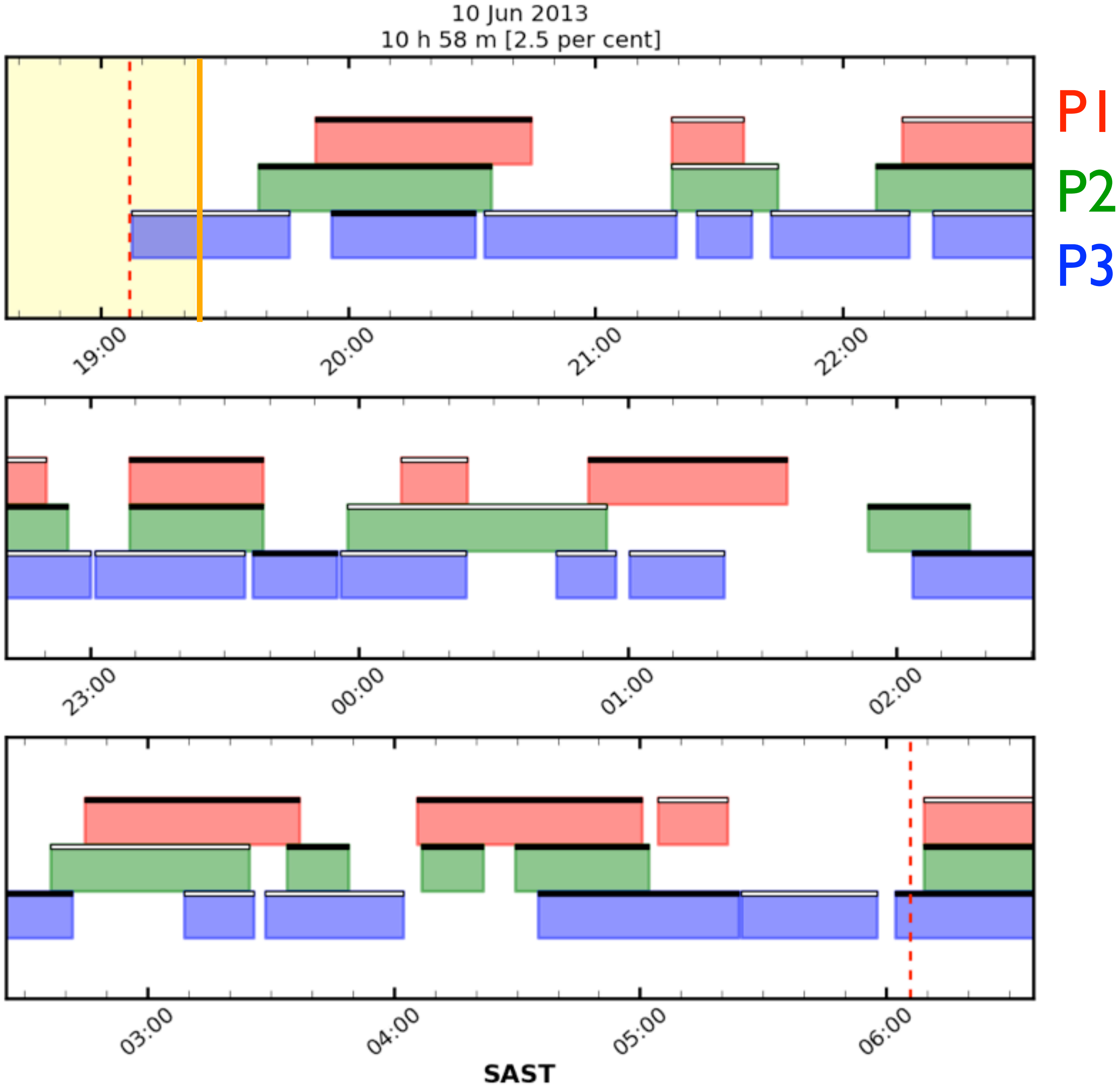


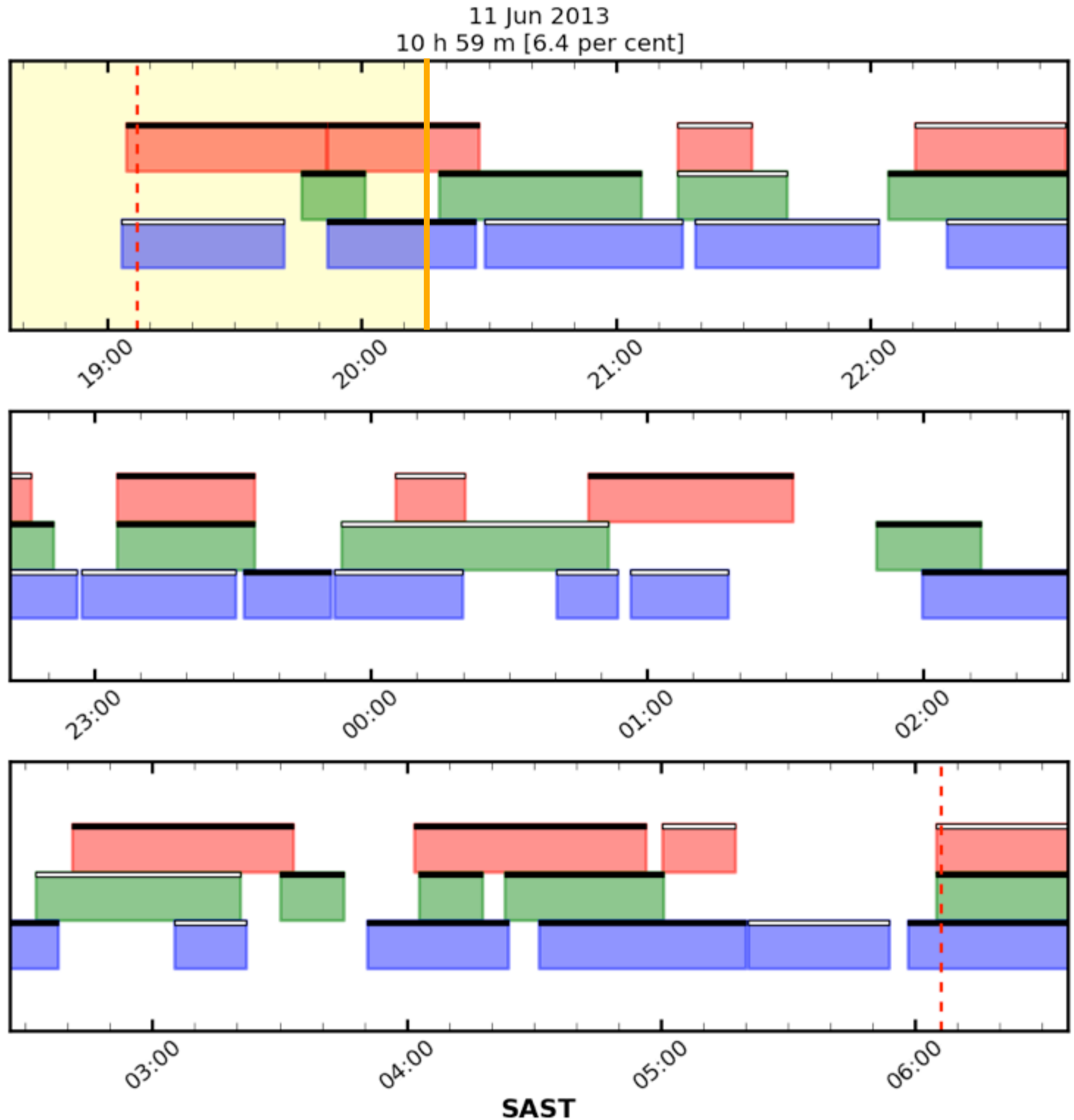


- Basic implementation in place
- Need a container class for the queue that stores individual Block class instances
- Queue itself: for now, a Python list
- May be sufficient for queue randomisation
- Could make use of helper priority queues (e.g. in the form of a heap)
- May need a way to select ‘best’ block in a given time interval for many intervals in the night

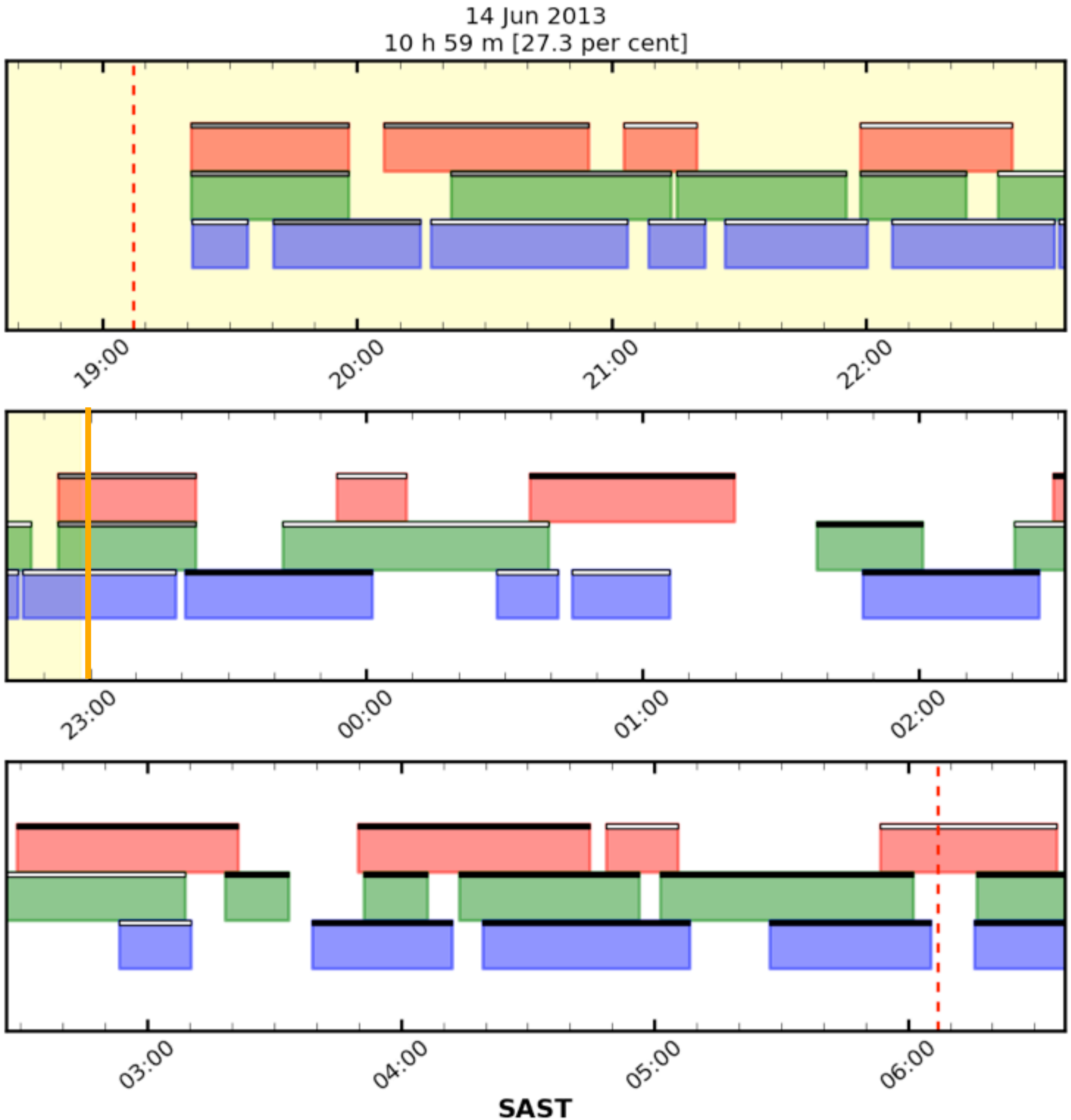
Block display examples (not an optimisation at all)

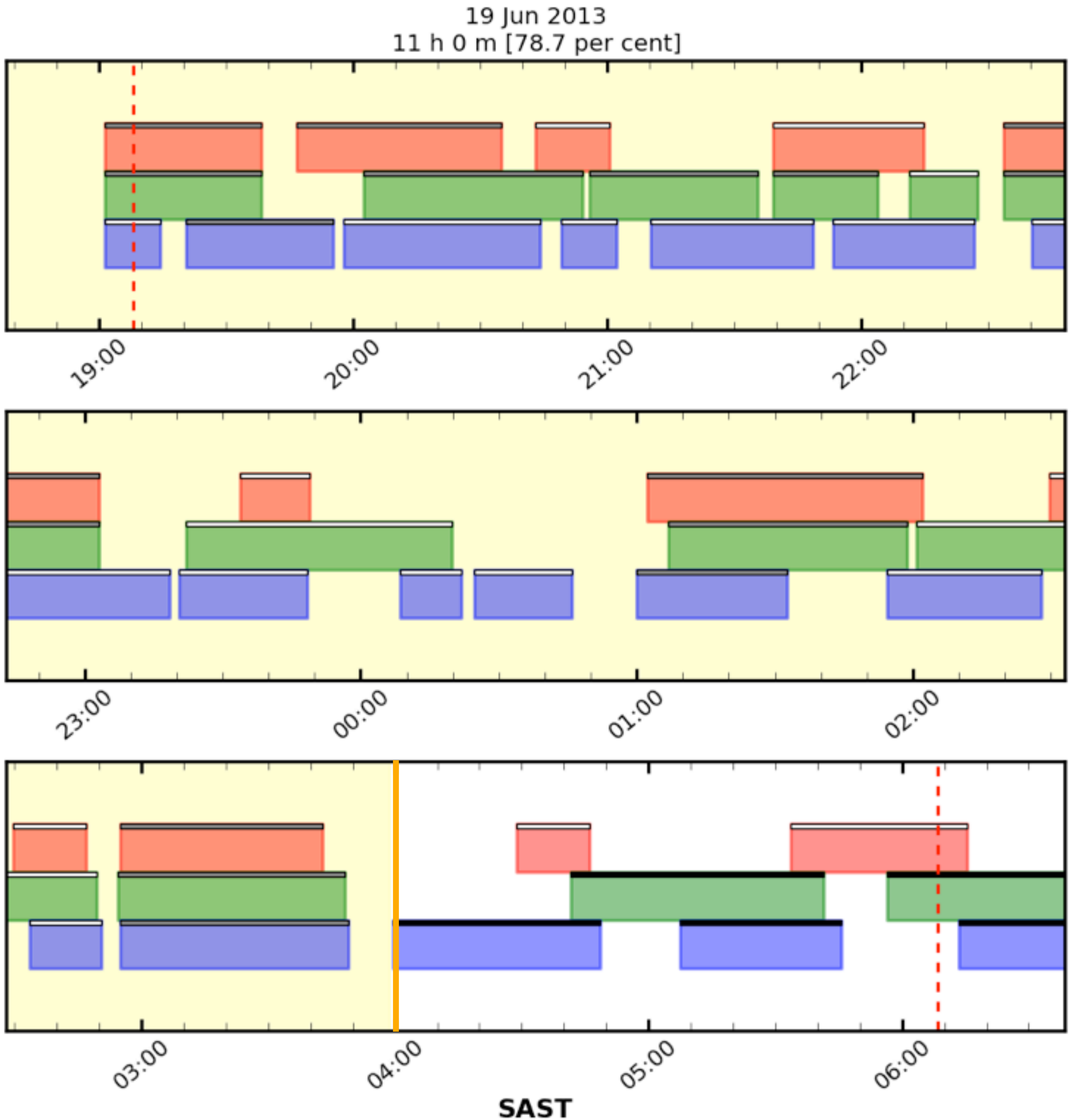
blocks are clickable => go to WM proposal page

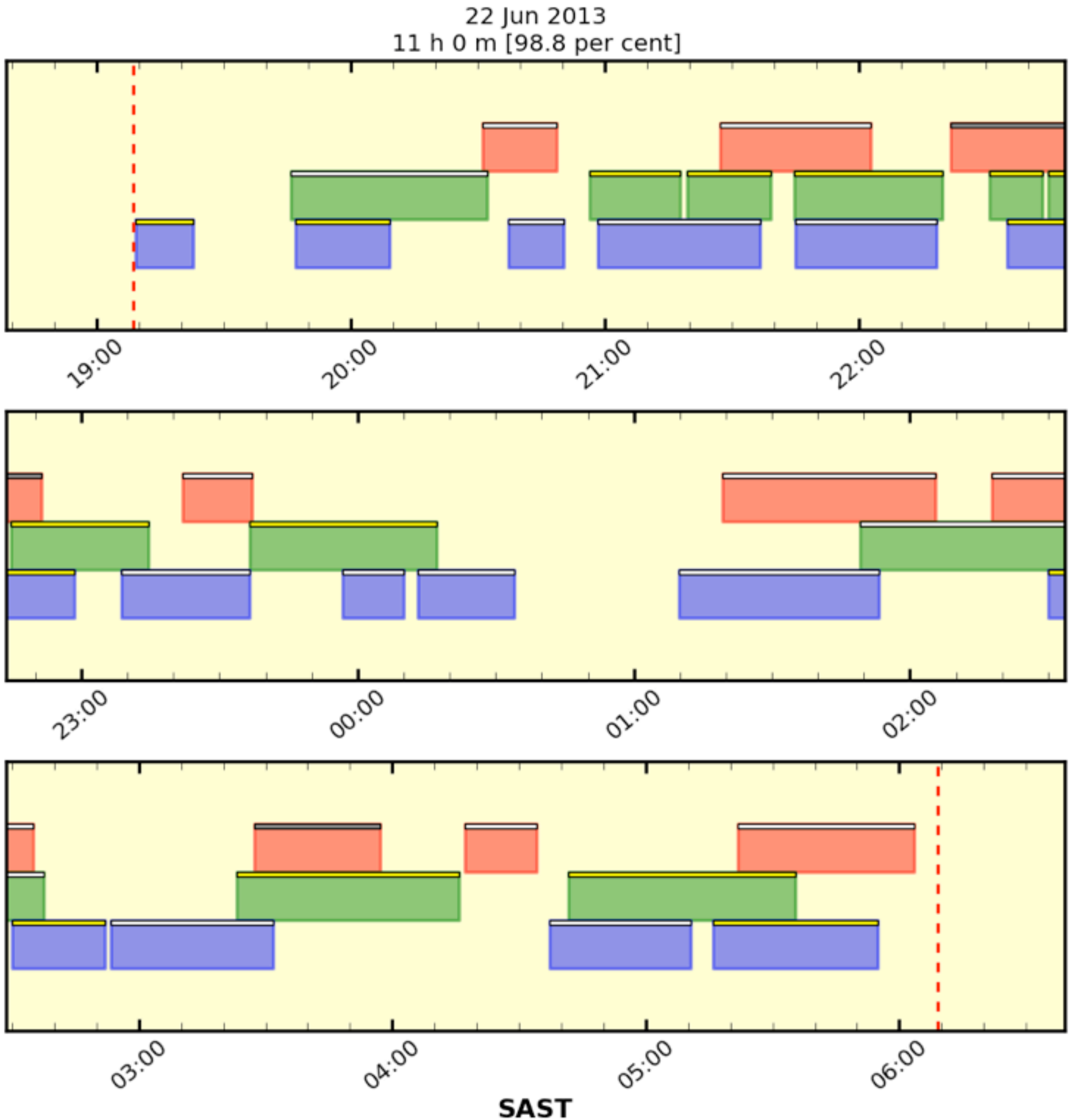


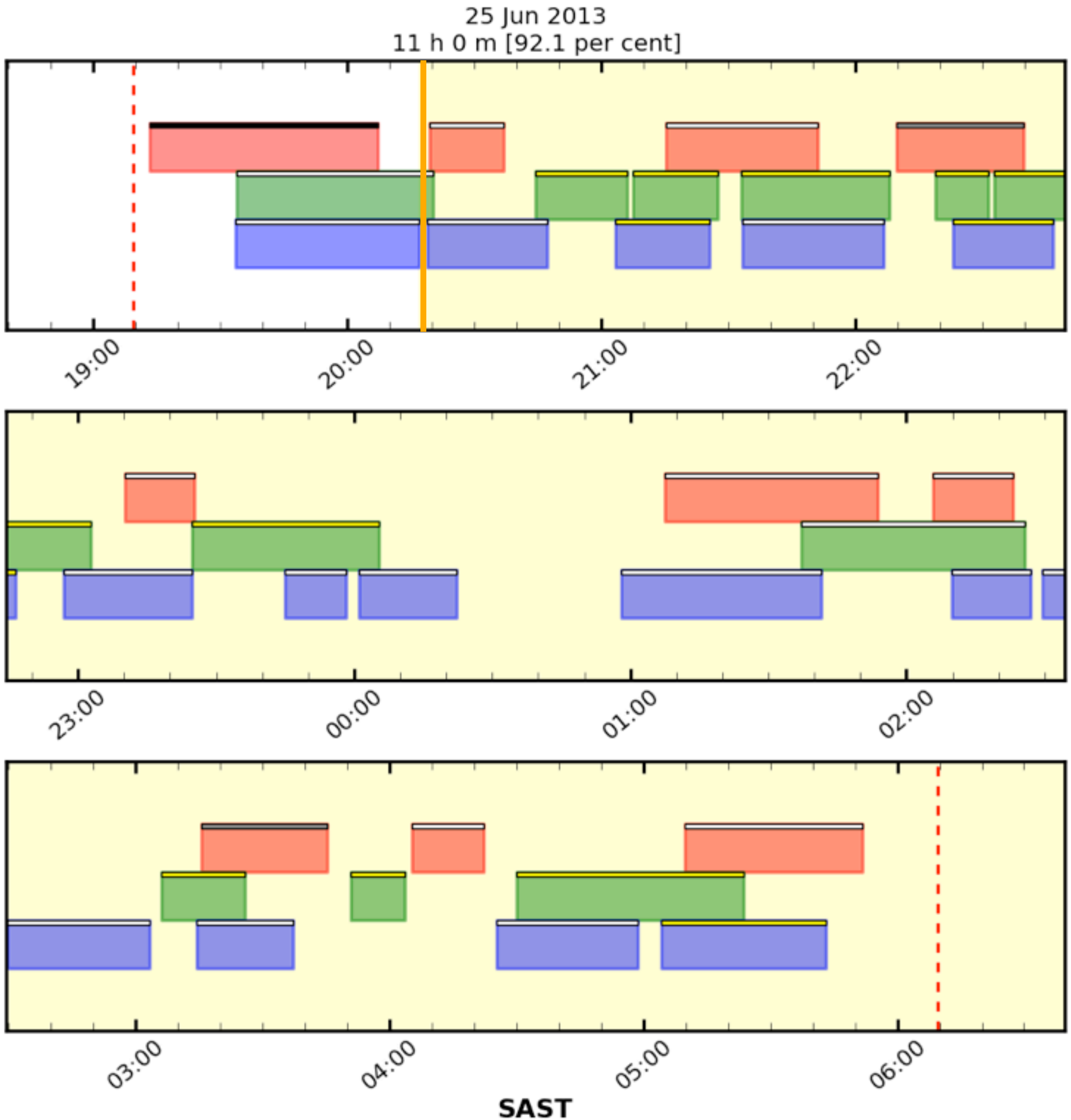


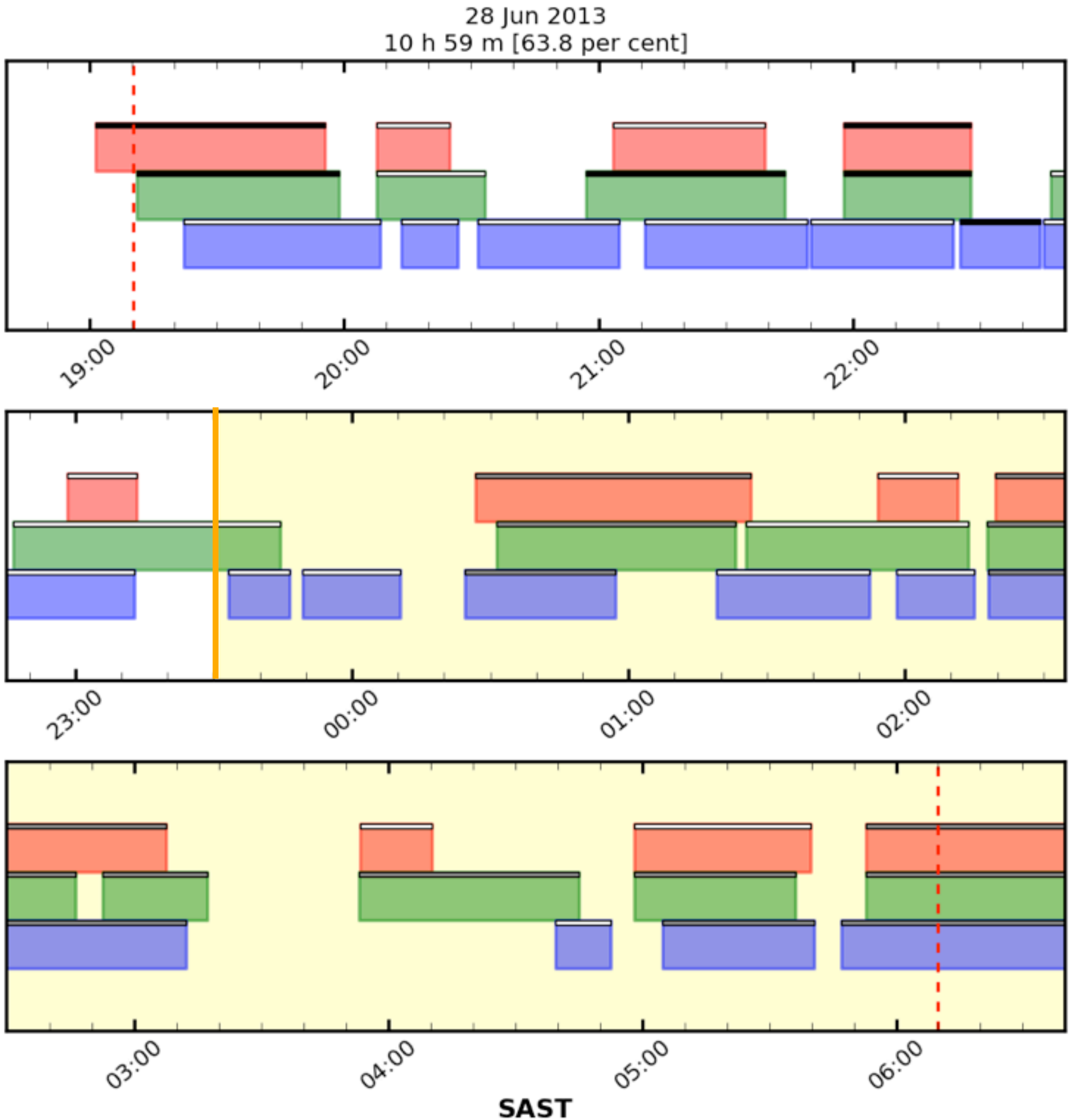
11 Jun 2013
10 h 59 m [6.4 per cent]
19:00
20:00
21:00
22:00
23:00
00:00
01:00
02:00
03:00
04:00
05:00
06:00
SAST

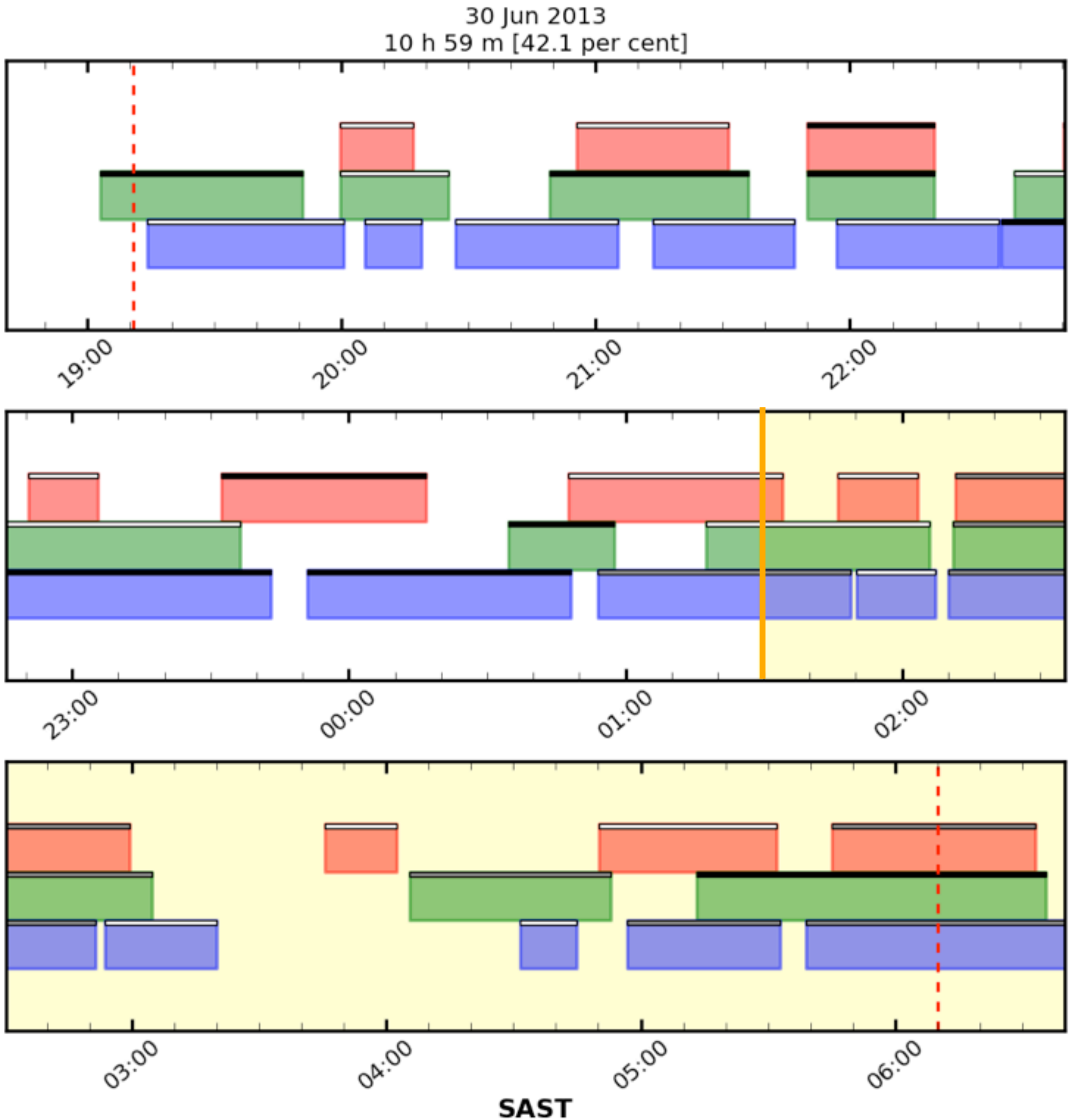

14 Jun 2013
10 h 59 m [27.3 per cent]
19:00
20:00
21:00
22:00
23:00
00:00
01:00
02:00
03:00
04:00
05:00
06:00
SAST


19 Jun 2013
11 h 0 m [78.7 per cent]
19:00
20:00
21:00
22:00
23:00
00:00
01:00
02:00
03:00
04:00
05:00
06:00
SAST


22 Jun 2013
11 h 0 m [98.8 per cent]
19:00
20:00
21:00
22:00
23:00
00:00
01:00
02:00
03:00
04:00
05:00
06:00
SAST


25 Jun 2013
11 h 0 m [92.1 per cent]
19:00
20:00
21:00
22:00
23:00
00:00
01:00
02:00
03:00
04:00
05:00
06:00
SAST

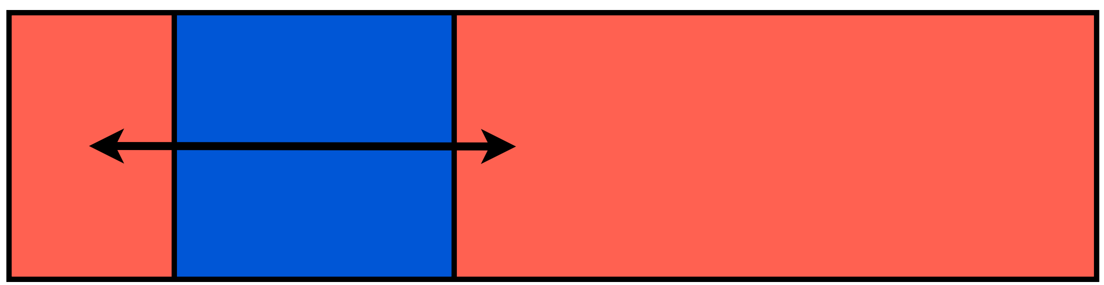

28 Jun 2013
10 h 59 m [63.8 per cent]
19:00
20:00
21:00
22:00
23:00
00:00
01:00
02:00
03:00
04:00
05:00
06:00
SAST


30 Jun 2013
10 h 59 m [42.1 per cent]
19:00
20:00
21:00
22:00
23:00
00:00
01:00
02:00
03:00
04:00
05:00
06:00
SAST

Queue randomisation

Visibility Window

can point to block any time in this window where TrackTime > ObsTime

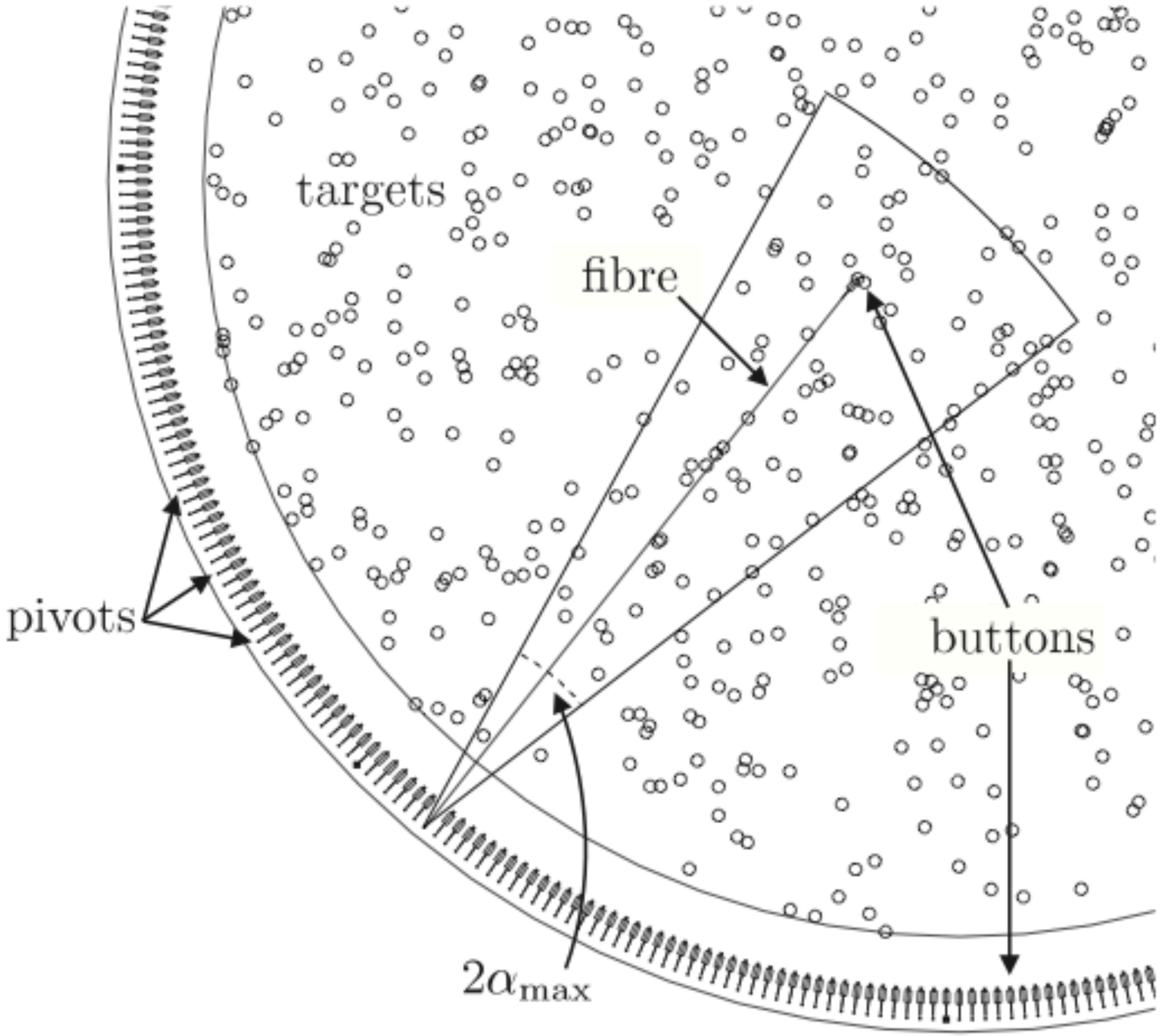


ObsTime

time needed to observe block, minus calibrations at the end

Next step: Implement simulated annealing algorithm with randomisation steps works well with priorities (e.g. Miszalski et al. 2006)

MOS field config by simulated annealing Miszalski et al. 2006, MNRAS, 371,1537



2dF on AAT
400 fibres

complex optimisation problem

9 target priorities
P1 (low) to P9 (hi)

Objective function

The randomization of each fibre during the annealing schedule is accepted only if the Metropolis algorithm is satisfied. A randomization is accepted with probability determined by

$$P = \begin{cases} 1 & E_2 \geqslant E_1 \\ e^{\Delta E/T} & E_2 < E_1, \end{cases} \tag{1}$$

where $\Delta E = E_2 - E_1$ and $e^{\Delta E/T}$ is only accepted if $\xi < e^{\Delta E/T}$, where $\xi \in [0, 1]$ is a random number. Traditionally, if $E_2 < E_1$ the operation would not occur, but the Metropolis algorithm ensures a non-zero possibility of acceptance that provides the algorithm with the ability to avoid local maxima. Here, E is the energy or quality of the field as determined by the objective function

$$E = \sum_{i}^{\mathrm{NPiv}} \left[\beta^{p_i} + \delta \sum_{j}^{\mathrm{NAssoc}} \beta^{p_j} \right] \left(\frac{\alpha_i - \alpha_{\max}}{\alpha_{\max}} \right)^{\gamma}, \tag{2}$$

where β, γ and δ are real parameters, $\alpha_{\max}$ is the maximum angular fibre deviation, and α_i, p_i represent the angular fibre deviation and priority of the allocation to pivot i. The first β term is a priority weighting identical to that used by Campbell et al. (2004). The second β term is a close-pairs constraint that favours NAssoc close-pairs around the target allocated to pivot i. This term is later used to

Randomisation steps

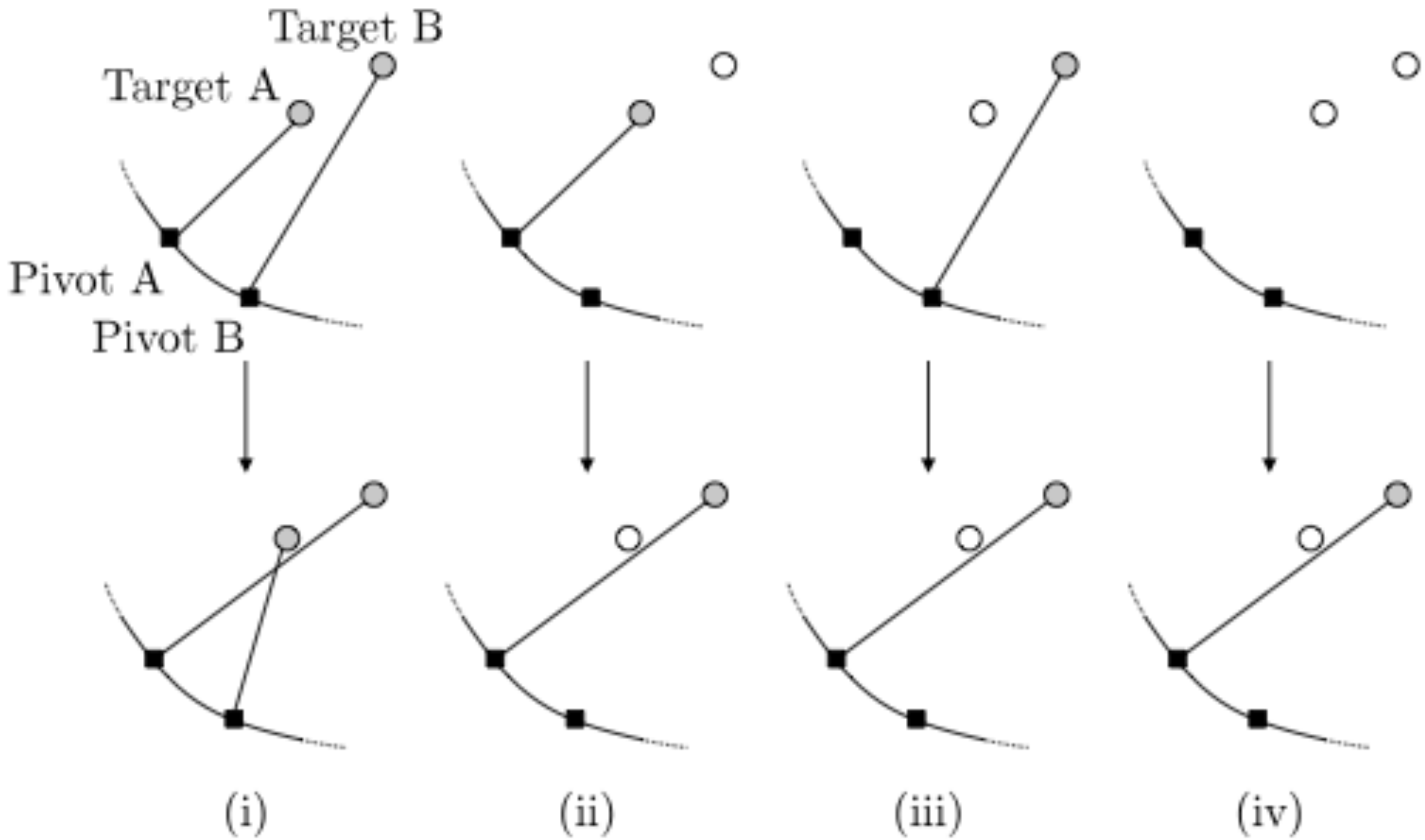
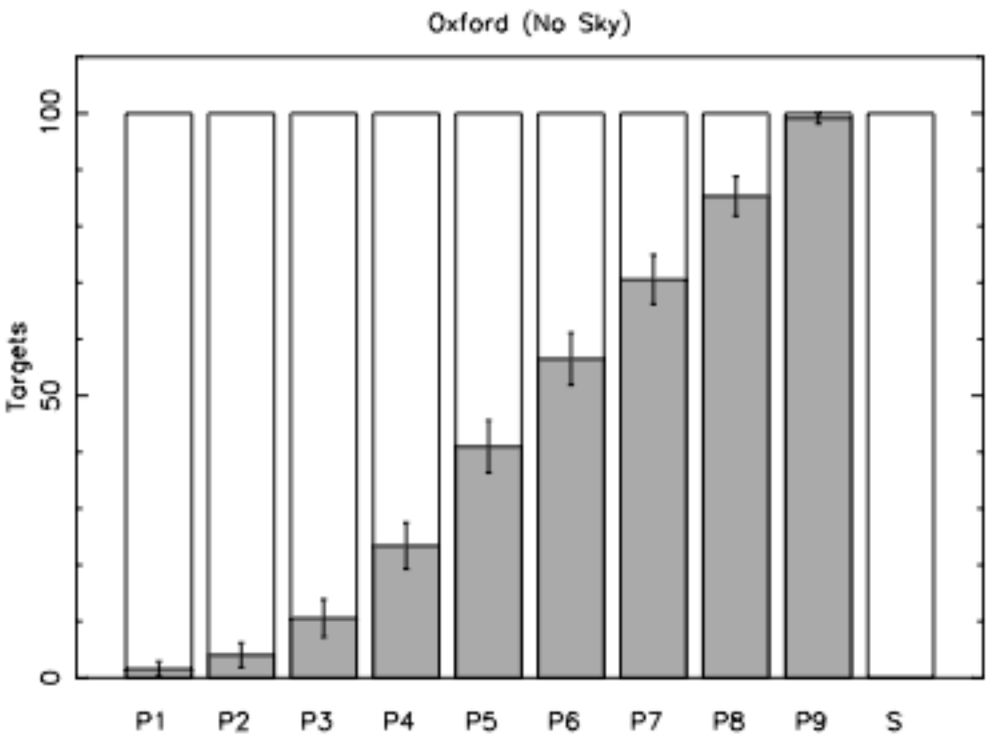


Figure 3. The four randomization cases possible when given PivotA and a randomly selected target TargetB. The type of randomization is determined by the possibility that PivotA may be allocated to TargetA and TargetB may be allocated by PivotB. The initial configuration (top) is changed which results in the final randomized state (bottom).

Priority distribution

Oxford algorithm



Simulated annealing