OVERALL SYSTEMS & PERFORMANCE
FILE NOTES
AT/20.1/014

The Scheduling Problem.

mjk, 22 March 1986

Ray Norris has recently outlined a possible scheduling algorithm. This note examines some limitations to his scheme.

The Compact Array

The Compact Array of the AT has been designed expressly to provide high quality, wide field, high dynamic maps. We can therefore expect that in general the observers will be requesting a number (n) of essentially complete 12 hour scans. (My guess is that n will likely be 4 on average).

It is currently believed that the array will remain in a given configuration for around two weeks - that is, we would expect to complete a 4-day observation in a two month period.

How many sources can we observe in two weeks? If the sources are all carefully chosen we could conceivably observe 27 sources, allowing about 30 minutes between scans. But in the real world the sources are unlikely to be so conveniently organized. I have run a few checks with random numbers, in order to set some lower limits. The results are shown below.

Number of sources in schedule	mean dead time between scans (hours)		Extra days added to schedule
5	8.8	+/- 3.6	1.5
10	6.4	2.4	2.4
15	5.4	1.8	3.1
20	4.7	1.4	3.8
30	3.9	1.2	4.7
50	3.0	1.0	6.2
100	2.2	0.7	9.0
180	1.8	0.6	13.2
500	1.0	0.3	21.3

In summary, a 20 source schedule which should be completed in 10 days will likely require 14 days, with an average dead time of 4.7 hours between sources. Schedules of more than 20 sources do not seem to offer much relief - a 100-source schedule has 2.2 hours dead-time/source, but the penalty is high: a 4-day source will require around 230 days: and the "24-day source" will require almost 4 years.

The difficulties arise because the principal observing mode will be a 12 hour scan. Thus, unlike the VLA, VLBA or LBA, the CA has little flexibility.

Consequences for the scheduler.

What do we do with the dead-time? Can we reduce it?

- we could observe some sources in sub-12 hour segments. This is probably undesirable from the calibration point of view. Obtaining high dynamic maps will be an uphill battle at the best of times; multiplying the number of non-contiguous segments does not help. One strategy might be to differentiate between night and daytime observations. The nighttime sources would be guaranteed a full 12 hour run; the daytime sources, likely to be of lesser quality in any case, are the ones which may be sub-divided. This route leads to a book-keeping nightmare, and to probable madness.

- a simpler alternative would be to recognize two classes of observations - the serious mapping, and the pilot quick survey type. The schedule would be built around the mapping (12 hour scan) sources: everything else would take its chances, and fit in as best it can. These would soak up the dead-time. Any time remaining could be used for general observatory calibration purposes.

In this scheme the scheduling task is still severe, but probably tractable. Observers requesting short scans would in effect generate a list of the source positions; the scheduler would slot them into the overall schedule as convenient, and delete the entry in the request file once the observation is completed. Since the observer may wish to keep a calibration source adjacent to his observations, the scheduler does not have complete freedom in dissecting the observer's requests.

- a yet simpler scheme is to adopt a more rigid, VLA-style, approach: a tentative schedule, based on the 12 hour requests, would be drawn up after the TAC deliberations. The remaining time would then be parcelled out to those observers with survey-type requests. At the end of this operation all the time segments of the schedule will have been defined; the dynamical rescheduling, should it occur, would apply to these segments, and not to the elements within them. (We would need to find some way of describing the extent to which a survey request can be slipped in time - earliest start time, latest stop time, for example).

I favour this last method as it is the simplest. A more sophisticated scheme might be worth pursuing at a later date, when we have a better idea of the mix of survey/12 hour observations.

I suggest the following sequence, applied on a quarter-by-quarter basis :

1. Divide the quarter into 4 parts, each to be allocated to a different array configuration.

2. To each configuration slot those observations which must occur there: the "24-day" and "4-day" sources.

3. Generate a schedule based on these, with the "1-day" sources being distributed over the four configurations so as to give roughly equal time to each configuration.

4. Any time left over at this stage is given to the sub-12 hour requests.

5. Since we will need to complete the "24-day" sources in a finite lifetime, we will probably need to assign some priority to the rescheduling process: "24-day" sources must be observed; "4-day" really ought to be, but could be postponed; the others may suffer.

The arguments given above are somewhat attenuated if the observing requests consist of a high proportion of 1-day sources. In that case the basic schedule will be dealing with a quarter's worth of sources - 180, approximately. The scheduler thus has greater scope, and the dead-time will be reduced (to 1.8 hours/source).

LBA

The 12-hour unit is no longer relevant, so we can expect a much higher scheduling efficiency. I suggest that the observers' requests should take the form of modules, specified by an earliest start time, and a latest stop time. The actual scheduling operations can then be quite dynamic.