# Post Servicing Mission 4 Parallel Observing with the Hubble Space Telescope

**Thomas Civeit, Mark Giuliano, Reiko Rager**

Space Telescope Science Institute
3700 San Martin Drive, Baltimore, MD 21218
{civeit, giuliano, rager}@stsci.edu

**Abstract**

The scientific instruments of the Hubble Space Telescope are located at fixed positions in the telescope focal plane. This feature allows more than one instrument to operate in parallel, therefore increasing the potential scientific productivity of Hubble. However, the support of parallel observing in the ground system is not straightforward, as it requires the coordination of multiple observers with different science goals. We present a new process for obtaining parallel observations that has been implemented and will be used in operations after Servicing Mission 4 (SM4). It is designed to solve problems that occurred with an earlier parallel implementation and it also takes full advantage of the instruments being installed in SM4. This new process ensures the scientific fidelity of parallel programs and creates, in advance of observations, operational matches that guarantee the scientific completion of the programs. Moreover, it provides better insight to the operations teams as they build long-range plans and weekly calendars. We describe how parallel and primary observations are coordinated using a flow network formalism. Finally, we report on how the new process has been used in operations to date.

## 1. Introduction to Parallel Observing

The scientific instruments of the Hubble Space Telescope (HST) are mounted in bays behind the primary mirror and are located at fixed positions in the telescope focal plane (Figure 1). It is therefore possible to increase the scientific productivity of HST by observing simultaneously with more than one instrument. Observing programs for HST can be categorized into two types depending on the pointing requirements. *Primary* observation programs have explicit telescope pointing specifications and may use one or more instruments. *Parallel* observation programs do not have specific predefined telescope pointings and are restricted to use certain instruments. Parallel programs propose for generic regions of the sky over which they could obtain useful data if the telescope happened to be observing in that vicinity using a different instrument. A typical example would be fields at a galactic latitude above 20 degrees. In general, observers of parallel programs are interested in doing similar observations at dozens or hundreds of parts of the sky using multiple wavelengths. Since all the instruments' fields of view are fixed relative to one another in the focal plane, the pointing and orientation of the primary instrument on the sky dictate the potential parallel target positions. The orientation of the spacecraft along its roll axis (i.e. boresight) may therefore constrain the scheduling of parallel observations. Parallel programs are opportunistic and depend on the available prime programs defined by other observers to determine possible target pointings.

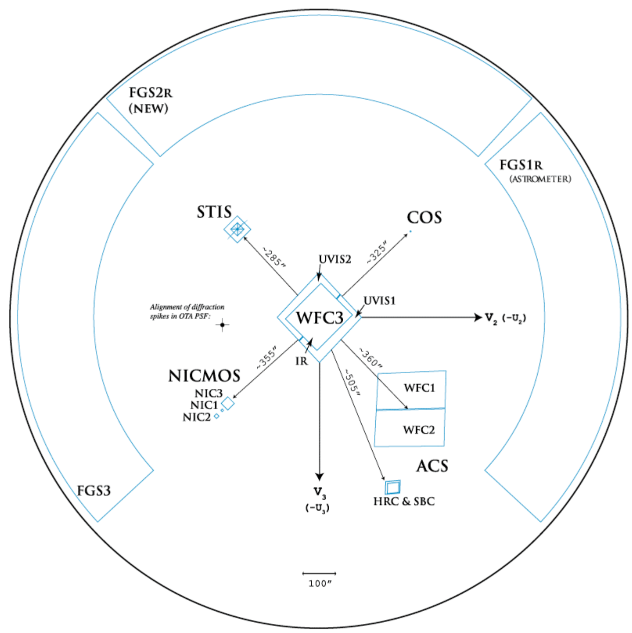



**Figure 1:** The post SM4 HST field-of-view with the location of the scientific instruments and Fine Guidance Sensors (FGS) in the focal plane. The scale in arcseconds is indicated.

The final HST servicing mission using a space shuttle is scheduled to take place in 2009. Servicing Mission 4 (SM4) will install two new scientific instruments, Cosmic Origins Spectrograph (COS) and Wide Field Camera 3 (WFC3). Over 500 COS orbits a year, more than 10% of usable HST orbits, are expected after SM4. COS has a small aperture and generates a relatively small amount of data, freeing up the time on the on-board Solid State

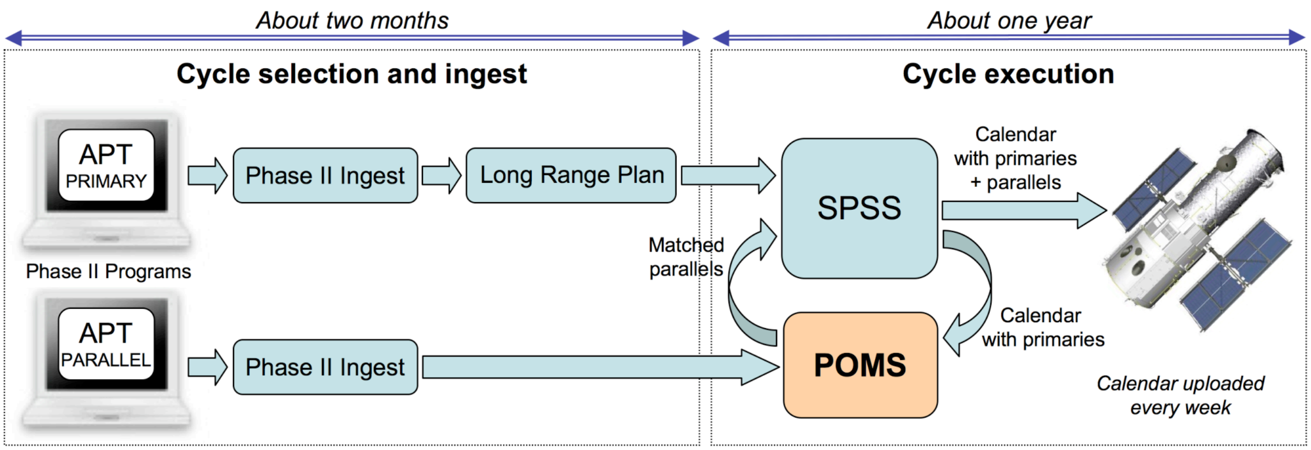



**Figure 2**: The previous parallel operations process flow with POMS. Observers specify observation programs using APT GUI. Submitted programs are prepared for scheduling (Phase II ingest process) at STScI. The short-term scheduler SPSS makes a one-week calendar with primary programs, POMS then matches parallels to primaries. Finally, SPSS schedules parallels based on the POMS output.

Recorder (SSR) to transfer data from other instruments. In addition, most of these observations will observe a single fixed target for multiple orbits. Hence, this presents an ideal orbit structure for parallel observing because it facilitates long sequences of parallel exposures. A goal of the HST mission is to use these COS orbits to enable parallel science. However, the existing process for parallels called Parallel Observation Matching System (POMS) (Lucks, 1992; Henry and Butschky, 1999) had several serious deficiencies. As a result, a working group was formed in early 2007 to explore options for a post SM4 strategy for executing parallel observations. The group, consisting of operations staff, software developers and scientists, examined different operations concepts and software procedures to perform parallel observing with COS. This led to the Parallel Observation Processing System (POPS), an operating paradigm that solves the POMS issues and that uses an innovative approach to determine operational matches from parallel to primary observations.

After a quick introduction to the HST mission in the next section, we list the lessons learned from the previous POMS system in section 3. An overview of the new POPS system is given in section 4, introducing the different software subsystems that define, select and execute parallel programs. The latter are described in the following three sections where particular attention is given to the match selection process in section 6. Finally, we report on the POPS usage for the 17th HST mission cycle before giving our conclusions.

## 2. Hubble Mission Background

Launched in 1990, the Hubble Space Telescope is a general purpose space observatory that provides support for near-infrared, visible, and ultraviolet frequencies. HST is in a low Earth orbit approximately 600 km above the Earth and orbits the Earth every 96 minutes. HST accepts new observations in *cycles,* time periods of typically 1 to 1.5 years long. In each cycle, thousands of new observations are chosen by a time allocation committee to be executed by HST.

Detailed requirements for observation programs are defined by observers during a two-step process, referred to as Phase I (initial call) and Phase II (planning and implementation). Section 3 describes the POPS integration with these phases. Below we describe the generic flow of HST observation programs through the Phase I and II processes.

Each observer submits a *program* that specifies an observing time for a set of specific or generic targets. Observing time is grouped in terms of *visits*, each of which is a series of one or more consecutive orbits. Each *orbit* contains a certain amount of useful time when the target can be observed called the *visibility period*. Exposures on the observer targets will execute in these periods. Once accepted programs are fully defined, the cycle-ingest process creates a long-range plan that integrates the previous cycle's leftover observations with the newly prepared observations. Finally, the short-term scheduler builds a weekly flight calendar, which is generated eleven days before the calendar begins to be executed on the spacecraft.

## 3. Problems with Previous Approach

Parallel observations were performed on HST from 1991 to 2005 using POMS to match parallel to primary observations. However, there were serious problems with

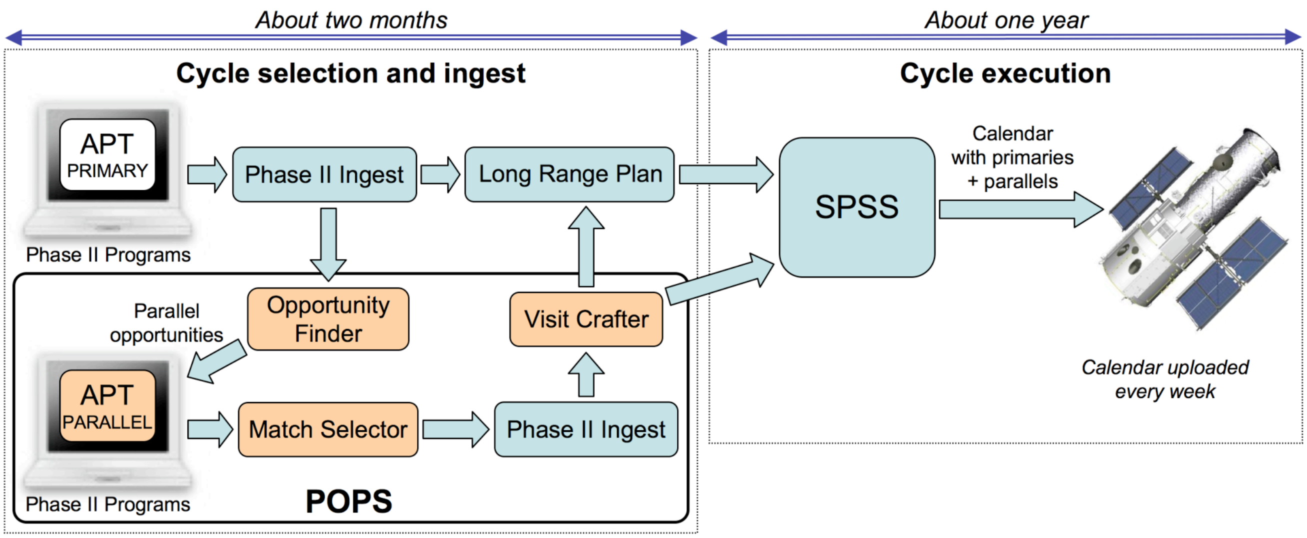



**Figure 3:** The new parallel operations process flow with POPS. Primary programs are submitted and processed first. Parallel programs are defined, matched to primes and prepared for scheduling. The short-term scheduler SPSS schedules primes and parallels together.

the POMS-based parallel process, which made parallel observation unpopular with both observers and operators. As a result, we had no parallel programs in two recent HST observation cycles. The POMS-based process had the following problems:

**Not ensuring scientific validity of parallels.** With POMS, parallel programs were constructed with no knowledge of which primary targets were available to match against. Parallel observers would not see what prime targets they matched until after the observation occurred onboard. As a result parallel observers could not plan their observations to best fit particular primes, and could not ensure that the prime targets that they matched against were scientifically useful. This made planning parallel programs too unpredictable for observers.

**Not ensuring scientific completion.** Using POMS, a parallel program was deemed 'complete' by Hubble operations when a predetermined number of parallel orbits were executed. However, the acquired data did not always achieve 'scientific completion' where the observers' research objectives were obtained. Various factors go into scientific completion. For example, a program may need to take data at three or more wavelengths, by repeating the exposures at a fixed pointing using different filters. Another program may need the same exposure repeated multiple times to reduce noise in the data. Yet another program may need data from different parts of the sky instead of data from one part of the sky. The POMS paradigm did not allow the construction and matching of programs to ensure that these conditions held. POMS enabled parallel observations but did not ensure that the observations were scientifically useful.

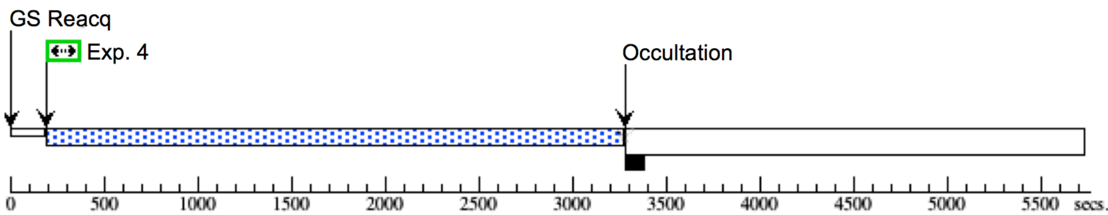
**Cumbersome operations procedures.** Because POMS was run at a time close to the actual upload to the telescope (see Figure 2), there was no time to resolve any problems that were discovered during a POMS run. If a parallel observation failed during short-term scheduling, there would be no time to fix it even though in some cases, a little change in a parallel observation would fix the problem. Likewise, the late process prevented exploiting possible improvements to a successful observation. For example, changing the telescope orientation angle of a primary observation may increase the parallel observing time.

## 4. POPS Overview

In order to resolve the aforementioned issues, we designed a new parallel process. The Parallel Observation Processing System (POPS) is integrated with the existing HST Phase I and Phase II tools. POPS refers to both the new process as well as a collection of software tools that have been built to support the new process. Figure 3 illustrates the operational flow with POPS and shows the POPS software components (Opportunity Finder, APT Opportunity Server, Match Selector and Visit Crafter).

After Phase I proposals are submitted, the selection committee approves or rejects proposals, both primary and parallel, according to the scientific merit and the time availability. In Phase II, the parallel process goes through three major steps; (1) defining parallel programs, (2) operationally matching parallels to primes, and (3) scheduling parallel programs on flight calendars. These steps are summarized below and details are provided in

**4(a):** An orbit structure suitable for a parallel opportunity




**4(b):** An orbit structure not suitable for a parallel opportunity

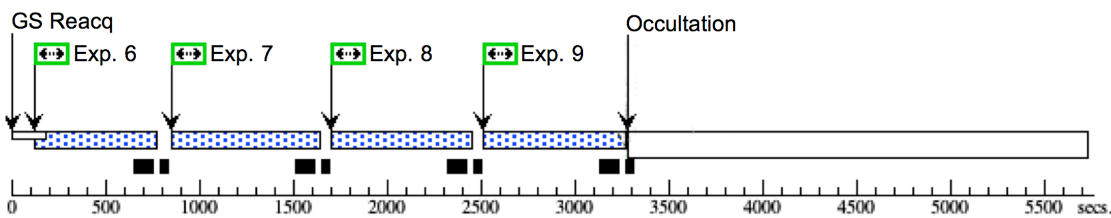



**Figure 4**: Orbit structure. The figure on the top shows a prime orbit structure that is desirable for a parallel opportunity. It has one long exposure (shown by a shaded rectangle) without any readouts (shown by lower black rectangles) overlapping it. The rest of the orbit, starting around 3300 seconds, cannot be used for exposures because the target is occulted by the Earth. In contrast, the figure on the bottom shows a structure that is not desirable for a parallel opportunity because of frequent readouts during the visibility period. Although exposures from prime and parallel instruments can occur simultaneously, their readouts cannot as all science instruments transfer their data to the same Solid State Recorder.

sections 5 through 7 showing how the new process solves the problems of the previous approach.

**Definition of Parallel Programs**

- In POPS process, prime observers write and submit their observation programs before parallel observers.
- After primary observers submit their Phase II observation programs, STScI operations identifies prime visits that are available for parallel observing, using the POPS Opportunity Finder. Based on the Opportunity Finder, selected prime visits with the same pointing within a single program are grouped into units called *prime opportunities*.
- Observers of accepted parallel proposals use the Astronomer's Proposal Tool (APT) Opportunity Server to look for prime opportunities that satisfy their pointing and exposure duration requirements, and write an observation specification that match them. A parallel program consists of a set of *parallel observations* where each parallel observation is a set of single orbit visits that can be executed on one prime opportunity.

**Selection of Operational Matches**

- STScI operations staff reviews the overall pool of requested matches between parallel observations and prime opportunities, resolves match conflicts using the POPS Match Selector tool, and makes assignments of parallel observations to prime opportunities. We call the assignments *operational matches*.

**Scheduling of Parallel Programs**

- The POPS Visit Crafter checks the feasibility of the operational matches and adjusts parallel observations as needed.
- Finally, during the flight calendar creation time, the parallel visits are readied for flight and scheduled simultaneously with their matched prime visits.

## 5. Definition of Parallel Programs

In the POPS process, prime observers write and submit their observation programs before parallel observers. Parallel programs are defined after opportunities are identified in the submitted prime programs.

### Identification of Prime Opportunities

After the detailed Phase II programs are submitted, operations staff finds and publishes potential prime opportunities. A prime opportunity is a set of one or more prime visits from a single primary program that could potentially match a parallel observation. Note that a parallel observation may or may not use the totality of the opportunity orbits, and consequently more than one parallel observation can be assigned to a single prime opportunity. All visits in a prime opportunity must satisfy a set of specific requirements:

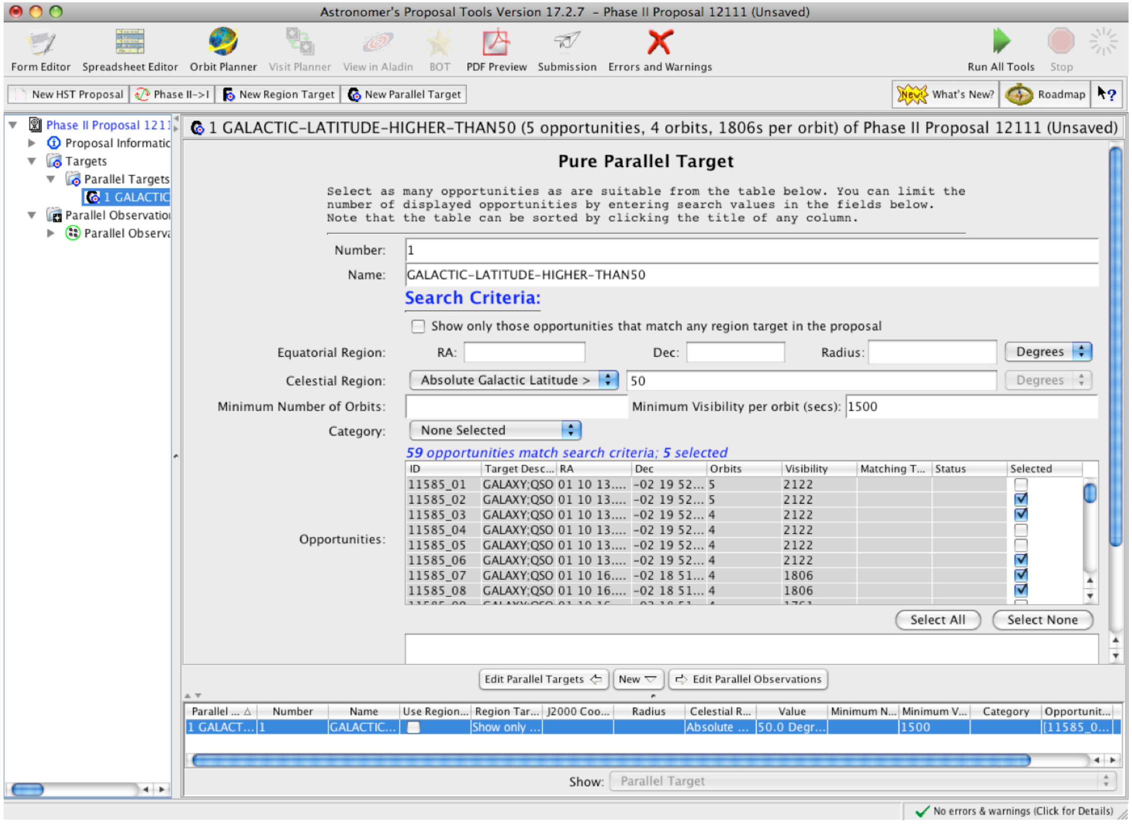



**Figure 5:** APT Opportunity Server GUI. Parallel observers can learn available opportunities and make selections for their parallel observations.

• **Orbit structure**. An opportunity orbit should have contiguous exposures longer than a predefined duration. The data readouts to onboard SSR occur only at the end of the orbit in order to avoid conflicts between the prime readouts and the parallel readouts. Only the prime visits using specified instruments (e.g. COS) are considered. Figure 4 illustrates two orbit structures; one that is ideal for a parallel opportunity and one that is not.

• **Pointing**. Visits have the exact same telescope pointing. In order to keep it steady, all prime visits need be executed at the same telescope orientation. Also, no solar system targets are allowed.

The POPS Opportunity Finder checks all the primary observation programs, and identifies and records prime opportunities. Some requirements, such as the required minimum duration of exposures, can be modified by input parameters. Users can adjust the parameters to reach a good balance of the quality and the quantity of opportunities. In order to process hundreds of prime programs across multiple processors, the Opportunity Finder uses the Sun Grid Engine.

## Parallel Program Writing

HST observers use APT (Roman, *et al*. 2004), a Java-based GUI, to specify and submit their Phase II observation programs. With APT, a primary observer can specify target locations, the exact configuration of an exposure in a visit, and check each visit's duration with overhead. We have made significant modifications to APT to support parallel observing. These modifications include adding the Opportunity Server to present observers with a list of available opportunities, and supporting the notion of parallel observations with single orbit visits. With APT, parallel observers gain the insight into what prime opportunities (i.e. pointing and duration) are available.

With the new process, parallel observers can see what kind of prime opportunities exist in the current pool of primary programs and choose the ones that fit their science objectives. APT's Opportunity Server retrieves a list of opportunities from STScI through a web server and shows the properties of each opportunity (Figure 5). A parallel observer can search and narrow down the choice of opportunities based on the target location, the types of targets, the number of orbits and the duration of visibility per orbit.

At least two opportunities should be requested for each observation in order to reduce the possibility of conflicts with other parallel programs and to increase the probability of achieving science completion. APT signals a warning when this preference is not followed.

Once a set of opportunities is selected for a new observation, APT automatically restricts the number of visits that the observation can contain to the minimum number of orbits among the selected opportunities. It also restricts the duration of the target visibility each parallel visit can use to the minimum visibility duration among the selected opportunities. Using the minimum orbits and visibility ensures that the parallel observation can be scheduled with any of the selected prime opportunities. APT also imposes other restrictions on parallel observations, such as which scientific instruments can be used.

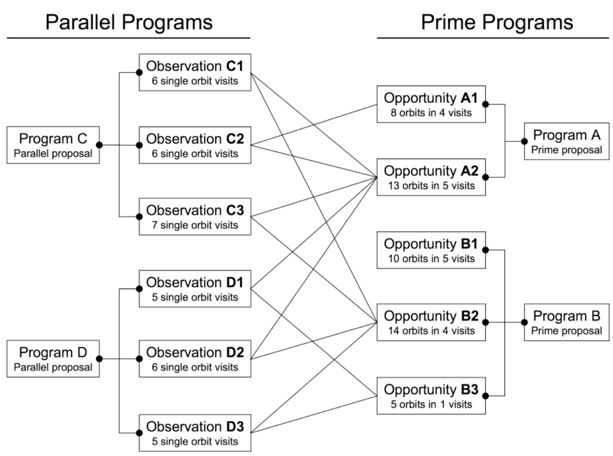



**Figure 6:** Opportunities are extracted from prime programs, then requested by parallel observers.

Figure 6 illustrates an example of requested matches between parallel observations and prime opportunities. There are two primary programs, A and B. A contains two prime opportunities and B contains three prime opportunities, as found by the Opportunity Finder. The resulting five prime opportunities have different orbit capacities. There are two parallel programs, C and D. Program C consists of three parallel observations, totaling 19 single orbit visits. Program D consists of three parallel observations, totaling 16 single orbit visits. The observer of program C has selected prime opportunities A2 and B2 for observation C1, A1 and A2 for observation C2, and A2 and B2 for observation C3. Prime opportunity A2 has 13 orbits and so can potentially accommodate a combination of the 5-orbit parallel observation D1 and the 6-orbit parallel observations C1, C2 or D2.

# 6. Selection of Operational Matches

Parallel observers submit their program specifications after having selected possible matches for their parallel observations with opportunities extracted from the prime programs. To ensure scientific completion of the parallel programs, each parallel observation is assigned to execute with a single prime opportunity out of the multiple requested matches selected by parallel observers. Since the APT system lets parallel observers independently request matches for their observations, there may be conflicts that STScI operations has to resolve. The assignment process is performed by the Match Selector tool that determines the list of operational matches. We will use the four programs (A, B, C and D) introduced in the previous section (Figure 6) to illustrate the match selection procedure. Operational matches obey the following rules:

• A prime opportunity can be used to satisfy multiple parallel observations as long as the number of available prime orbits allows it. For instance, the 14 orbits of opportunity B2 can potentially host both parallel observations D3 and C1, or any combination of parallel observations C1, C3, D2 or D3 while the total orbit usage is smaller than or equal to the orbit capacity of the opportunity.

• A single prime visit can host orbits from multiple parallel observations. In prime program A, one of the five visits of opportunity A2 could be matched to two single orbit visits from parallel observations C2 and D1.

• One prime orbit cannot host more than one parallel orbit.

• All visits of the same parallel observation are assigned to a single opportunity. This means that the 6 orbits of the parallel observation C1 cannot be distributed on opportunities A2 and B2.

• Observations from a single parallel program can be assigned to opportunities from different prime programs. In parallel program C, it is allowable to assign observation C1 to opportunity A2 and observation C3 to opportunity B2, even though opportunities A2 and B2 do not belong to the same prime program.

In addition to these constraints, preferred operational matches are those that minimize the impact on the spacecraft schedule, which means avoiding additional constraints on prime visits and facilitating possible repairs when the plan does not execute as expected.

Selecting operational matches is thus equivalent to finding a many-to-one mapping, which minimizes a cost function, from the set of parallel observations to the set of opportunities (both being obviously disjointed). Possible maps can be generated by iteratively finding a maximum bipartite matching in a bipartite graph, whose vertices are the observations and the opportunities, and whose edges are the requested matches.

To increase scientific productivity, a mapping having a maximum number of matches will always be preferred. The value of a cost function will be a tie breaker when several maximum matchings exist. Note that STScI operations staff may decide to manually specify a set of operational matches, based on strategic or scientific decisions. In such case, the latter are simply created before searching for a mapping, and the set of requested matches is updated to reflect the new restrictions. The overall Match Selector algorithm can be sketched as:

```
select_matches
  FOR n_iterations
    randomize graph traversal
    (mapping, cost) ⇐ find_mapping
    store (mapping, cost)
       when better than its previous value
  END
  RETURN mapping and cost

find_mapping
  mapping ⇐ match opportunities requested
                by a single observation
  WHILE a matching can be found
    mapping ⇐ mapping + find maximum matching
    update orbit capacity
        of the matched opportunities
    remove requested matches
        of matched observations
  END
  cost ⇐ compute cost of mapping
  RETURN mapping and cost
```

Iterations in `find_mapping` are needed because a maximum bipartite matching is a one-to-one matching and we allow several observations to match the same opportunity. Such design inherently gives priority to assignments from one parallel observation to one opportunity (which corresponds to a single iteration in `find_mapping`). This feature is actually intended since a one-to-one map is preferable from the operations perspective. Mappings are iteratively generated in `select_matches` in order to minimize the cost function. Different mappings can be obtained by randomizing the graph traversal used in the maximum bipartite matching algorithm, which is described in the next section.

## Maximum Bipartite Matching

It is well known that a maximum bipartite matching in a bipartite graph can be easily found by computing the maximum flow in the corresponding flow network. The augmenting path algorithm (Ford and Fulkerson 1962, Edmonds and Karp 1972) computes the maximum flow and, in the case of a bipartite graph, runs in $O(V*E)$, where $V$ is the number of vertices (i.e. number of parallel observations + number of opportunities) and $E$ is the number of edges (i.e. number of requested matches). The flow network is constructed as follows:

• each parallel observation or prime opportunity corresponds to one vertex.

• each requested match corresponds to one directed edge from a parallel observation to a prime opportunity.

• a super source vertex is added with edges to all parallel observation vertices.

• a super sink vertex is added with edges from all prime opportunity vertices.

• all edges are assigned a capacity of 1.

One may think that edge capacities could be used to represent the number of orbits available in opportunities or requested by parallel observations, hence removing the need for iterations. It is, unfortunately, not a correct approach since the same parallel observation cannot be assigned to multiple prime opportunities. An example of the iterative maximum bipartite matching algorithm is presented below in Figures 7a-e. The orbit capacity and usage of the prime opportunities as well as the orbit requirement of the parallel observations are shown in parentheses.

The initial flow network corresponds to the set of parallel programs (C, D), prime programs (A, B), and requested matches that were shown in Figure 6. Although it is not matched, opportunity B1 is included in the network for consistency.

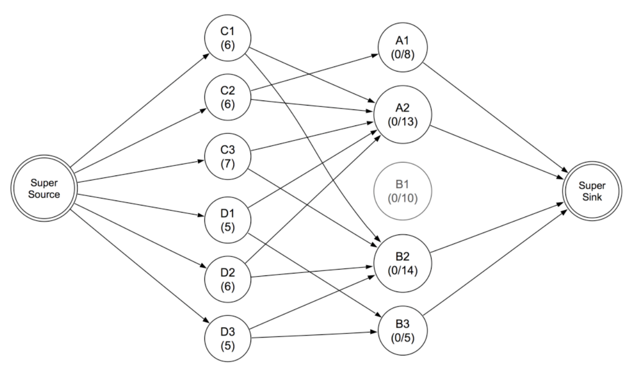



**Figure 7a:** Flow network in the first iteration. All edges have a capacity of 1.

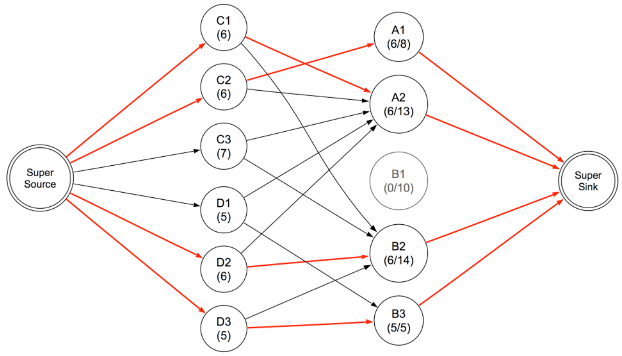



**Figure 7b:** At the end of the first iteration. The maximum flow, shown in red, is equal to 4.

Parallel observation C2 is automatically assigned to opportunity A1 since the latter has no other match requests. After a maximum flow is found, the resulting matches are created and the orbit duration of the parallel observations is removed from the orbit capacity of the opportunities.

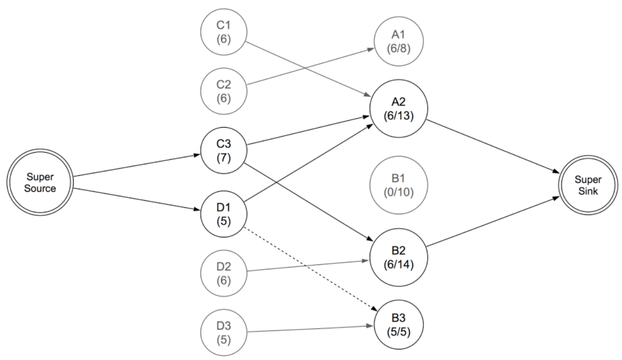



**Figure 7c:** Flow network in the second iteration. All edges have a capacity of 1.

In the second iteration, the network is adjusted to only include non-matched parallel observations. All five orbits of opportunity B3 are used by parallel observation D3, consequently the requested match D1-B3 is deactivated. This applies for any opportunity which no longer satisfies the orbit capacity requirement of a match.

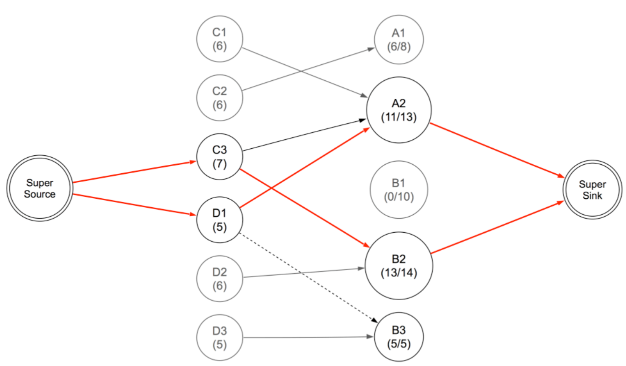



**Figure 7d:** At the end of the second iteration. The maximum flow, shown in red, is equal to 2.

After a new maximum flow has been found, the remaining non-matched parallel observations C3 and D1 are assigned to opportunities B2 and A2, respectively.

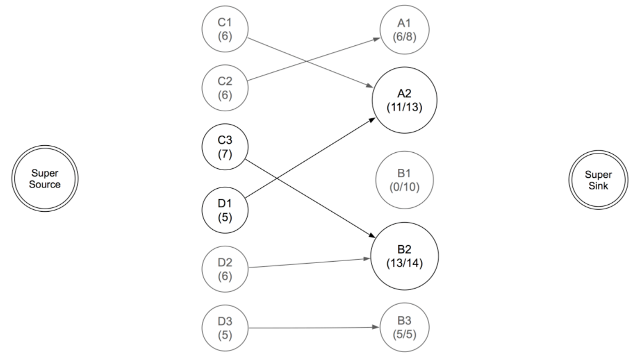



**Figure 7e:** Final state, a complete mapping has been found.

All of the six parallel observations are matched, so there is no need for a new iteration. While computing maximum bipartite matchings, different augmenting paths can be found during the network traversal by simply randomizing the order of the adjacent vertices returned for a given vertex. They will, of course, all converge to the same mapping when there is a unique solution.

## Selection Criteria

Although the matching cardinality is the supreme selection criterion, each mapping is evaluated by computing its cost. The latter is computed from a set of criteria, whose weight can be adjusted by the user. The total cost is the weighted sum of all the criteria. There are currently three criteria:

• Minimize the number of prime visits that will need to have the same spacecraft orientation. This constraint has to be added when multiple prime visits host several orbits of the same parallel observation.

• Minimize the number of matched opportunities that have no extra visit available for a redo. A redo is necessary when a visit execution failed.

• Minimize the number of parallel programs whose observations are not fully matched.

The weights of the first two criteria impact the way multiple parallel observations are assigned to a single prime opportunity. Two options are usually available, as shown in Figure 8.

The first option consists in creating matches from one parallel observation to one prime visit as much as possible. This option has the advantage of minimizing the number of orientation constraints added to the prime visits. However, its drawback is to reduce the orbit capacity of the prime visits. The second option is to assign orbits back-to-back. It has the exact opposite effect: it makes full use of the prime visits but it also tends to increase the need for orientation links between prime visits, thus potentially limiting the schedulability of a visit. Since no option is better than the other in the general case, the Match Selector tries both strategies as it generates different mappings. The decision is then made by evaluating the cost function.

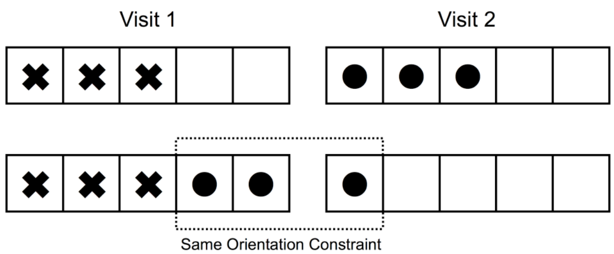



**Figure 8:** Top: one-to-one match from two 3-orbit long parallel observations (crosses and circles) to two visits of the same opportunity. Bottom: alternative back-to-back assignment.

## 7. Scheduling of Parallel Programs

Once parallel observations are operationally matched, the Visit Crafter adjusts parallel programs to fit better in the matched prime orbits. Later, during the cycle execution phase, the SPSS scheduler creates a week long calendar of HST observations every week. Prime and matched parallel visits are scheduled together to minimize the situation where prime visits are scheduled at times when the matched parallel visits cannot be placed.

### Scheduling

The Visit Crafter checks the feasibility of parallel programs by first creating the orbit timeline of the matched prime programs and test-scheduling the parallel visits. Parallel visits are constructed in APT to fit in the shortest visibility among the chosen opportunities. However, the real visibility varies from orbit to orbit. In order to fully utilize the available visibility, the Visit Crafter has a capability to lengthen certain exposures.

In the POMS process, a weekly schedule of HST observations was created first with primary programs only, then parallel programs were added later. About 5% of the parallels matched to primes by POMS failed to schedule on the calendar because of slight differences in target visibility between prime and parallel instruments. In order to avoid such scheduling failures, the SPSS scheduler was modified to schedule both primaries and matched parallels simultaneously, such that SPSS looks for a start time that satisfies both the primary visit and the attached parallel visits.

### Operations

Figures 2 and 3 show the previous and new operational flow of parallel observation processing. The light-colored boxes show the steps that operations staff has to take to support parallels. As we can see, the new process using POPS requires more steps, compared with the previous process using POMS. However, the extra steps occur at the start of a cycle. One problem with POMS is that the work happens at schedule creation time, which does not allow the time necessary to resolve any problems or concerns. With POPS, most work occurs at the start of the mission cycle. If desired, parallel programs can be test-scheduled with matched primary observation on SPSS. This minimizes any last minute scheduling problem at the short-term schedule creation time. We expect few problems at scheduling time, with more parallels successfully scheduled.

## 8. HST Mission Cycle 17 and evolution

At the start of the new parallels rework, SM4 was slated to occur in August 2008. Cycle 17 Phase I proposals were submitted in March 2008. Six parallel programs were submitted, and the selection committee approved three, requesting 592 orbits. Upon examination, we realized that all three programs were looking for similar prime opportunities at high galactic latitudes.

The Phase II primary programs were submitted in July 2008. Since we noticed all three parallel observers were interested in opportunities at high galactic latitudes, we modified the Opportunity Finder to filter out opportunities closer to the galactic equator. We ran the Opportunity Finder dozens of times with varying the visibility duration threshold and minimum number of orbits required. We decided on a set of opportunities with orbit-visibility greater than 1000 seconds and visibility-sum-over-the-opportunity greater than 6000 seconds. The resulting opportunity set consisted of 100 opportunities covering 433 orbits.

At this point we knew that all three parallel programs needed the same kind of prime opportunities and that we did not have enough prime opportunity orbits to accommodate all of the parallel programs. Since it was the first year with the new parallels process, we invited the observers of the three selected parallel programs to STScI to instruct them in how to use APT to specify their parallel science. We decided to use that chance to let the observers divide the prime opportunities among themselves. This way, the institute would not need to make difficult matches later, and also parallel programs could be written to fully utilize available visibility. As a result of this manual process, the parallel match procedure was not used operationally as the parallel investigators had already done the matching. We do however expect to use the match capability in future HST cycles.

The Phase II programs of parallel proposals were submitted in September 2008 and were processed through the Visit Crafter with no problems. Unfortunately, technical problems on shuttle Atlantis and a failure in equipment onboard HST postponed the shuttle launch. As of this writing, SM4 is planned for mid-May 2009. No Cycle 17 parallel observation has been performed on HST at this time.

## Conclusion

We implemented a new scheduling process to obtain parallel observations with HST, taking maximum advantage of the instrument complement after SM4. Assignments of prime opportunities to parallel observations now occur during the planning and implementation phase after observer programs have been submitted. This operating paradigm contrasts with the earlier implementation where observers were asked to submit parallel observing templates that were not matched to prime science opportunities until a weekly flight calendar was built. The new implementation provides tools to extract prime opportunities from prime programs, create possible matches, resolve conflicts, select operational matches and prepare parallel visits before being executed on HST. This improvement solves the issues of the previous process by

• *Ensuring scientific validity of parallels*. Parallel programs meet science goals as parallel observers construct programs based on available prime opportunities.

• *Ensuring scientific completion*. Observations from parallel programs are assigned to execute with prime opportunities at cycle ingest time to ensure the programs complete.

• *Avoiding cumbersome operations procedure*. Scheduling problems can be found and fixed well in advance of parallel program execution.

## References

Ford, L. R., and Fulkerson, D. R., *Flows in Networks*. Princeton U. Press, Princeton, N.J., 1962

Edmonds, J., and Karp, R. M., *Theoretical Improvements in Algorithmic Efficiency for Network Flow Problems*. Journal of the ACM (JACM), v.19 n.2, p.248-264, April 1972

Henry, R. L., and Butschky, M. 1999. Recent Improvements to HST Parallel Scheduling in Astronomical Data Analysis Software and Systems VIII, ASP Conference Series, Vol. 172, 1999, Editors: D.M. Mehringer, R.L. Plante, and D.A. Roberts, pp. 81-84.

Lucks, M. 1992. Detecting Opportunities for Parallel Observations on the Hubble Space Telescope in *Proceedings of the 1992 Goddard Conference on Space Applications of Artificial Intelligence*", ed. J.L. Rash, NASA Conference Publication 3141 (Greenbelt: NASA), pp. 29-44.reprinted in *Telematics and Informatics*, 9, pp. 331-347

Roman, A. J., Douglas. R., Downes, R, Krueger, A., and Peterson, K. 2004. Astronomer's proposal tool: the first two years of operation. In *Optimizing Scientific Return for Astronomy through Information*. SPIE, Vol. 5493, pp.351-358