

Reactive Scheduling for LINC-NIRVANA

Juergen Berwein, Aleksei Pavlov, Florian Briegel, Wolfgang Gaessler and Clemens Storz
Max Planck Institute for Astronomy, Koenigstuhl 17, 69117 Heidelberg, Germany

ABSTRACT

LINC-NIRVANA is a Fizeau interferometer for the LBT. The instrument combines the two 8.4 m telescopes into one image plane. The fixed geometry of the telescope and the adaptive optics of the instrument put constraints on the observation schedule. Environmental changes influences the execution of observations. We present a robust and reactive scheduling strategy to achieve high observation efficiency and scientific results with our instrument.

Keywords: LBT, LINC, NIRVANA, MPIA, Scheduling, Observation Support Software

1. INTRODUCTION

The MPIA is leading an international consortium of institutes in building an instrument called LINC-NIRVANA,¹ the **LBT INterferometric Camera and Near-IR / Visible Adaptive INterferometer for Astronomy**.

LINC-NIRVANA is a Fizeau interferometer for the Large Binocular Telescope (LBT)² doing imaging in the near infrared (J, H, K - band). Multi-conjugated adaptive objects is used to increase sky coverage and to get diffraction limited images over a 2 arcminutes field of view.

The operation of LINC-NIRVANA (LN) requires large amount of setup and operational procedures. Interferometric observing with LINC-NIRVANA requires therefore a new paradigm, because there exists considerably less freedom in scheduling observations than in traditional astronomy. For example, since the LBT interferometer depends on the earth's rotation to present different object position angles, we may need first a particular position angle on a particular source, then move to grab another position angle on another source for a while, and then return to the previous target. Also certain environmental constraints have to be evaluated permanently during the observation. Thus, an advanced and reactive scheduling tool is mandatory to take maximum advantage of precious observing time. In this article we propose an approach of reactive scheduling, which maximizes the scientific return and operational efficiency of LBT.

2. DEFINITIONS

The software of the observation and preparation process is mainly divided into two major parts. First the software components which provide the functionality to prepare sets of observations and second software tools to support the observer and operator respectively. The following list describes the terminology of those components.

2.1. Packages

- OPS (Observation Preparation Software):

The observation preparation software consists of several software tools to define the overall data structure of an observation program. Executional relevant data like constraints and division into multiple observation blocks, depending on the amount of parallactic angles needed for an observation, will be defined there. Usually this task is done in advance before starting the actual nightly observation procedure.

- OSS (Observation Support Software):

The observation support software consists of a set of software tools to proceed the scheduling and execution of observation blocks. The workflow is described in section 4.2., and the dataflow inside the OSS is described in section 4.1. As seen in Figure 1, the OSS is placed between the more scientific oriented package, the OPS, and the more instrument oriented package, the ICS. The data structure from the OPS (Observation Programs) are processed, evaluated and sent as instrument level commands to the ICS.

Further author information: (Send correspondence to Juergen Berwein)

Juergen Berwein: E-mail: berwein@mpia.de, Telephone: +49 (0)6221 528 324

Aleksei Pavlov: E-mail: pavlov@mpia.de, Telephone: +49 (0)6221 528 329

Observatory Operations: Strategies, Processes, and Systems, edited by David R. Silva, Rodger E. Doxsey,
Proc. of SPIE Vol. 6270, 627010, (2006) · 0277-786X/06/\$15 · doi: 10.1117/12.671261

- **ICS (Instrument Control Software)** The instrument control software provides the low level functionalities for the instrument hardware and the overlaying processing logic. In normal operation mode, the ICS should not be accessible to the observer and can be seen as a blackbox system. Commands and parameters are passed over a standard interface to the ICS (see. Figure 1). Errors and status feedback will be passed from the ICS to the observation support package OSS.

2.2. OSS Modules

- **Repository:**
The repository is a data storage and data exchange module between the OPS and the OSS. It is most likely to be implemented as a standard relational database management system.
- **Scheduler:**
The scheduler provides functionality to sort imported OTUs (see 2.3) and their containing OBs in the chronologically most reasonable order. This execution order is only a recommendation to support the observer. Depending on how accurate and high-performance this scheduler is, observations over a certain time interval e.g. one night of observation, could be performed automatically.³ In our case the implementation is a support tool.
- **Executor:**
The executor is a software tool to execute observation procedures by using data from its OB. The observation procedures send commands and parameters to the lower instrument control software (see Fig. 1). The observation procedures are implemented in scripts, which are executed during runtime and fed with data from the appropriate OB.

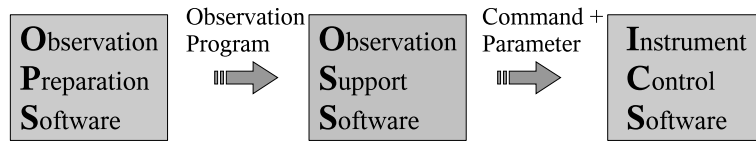


Figure 1. Main software packages of LINC-NIRVANA.

2.3. Observation Data-Structure

The logical data structure can mainly be divided into three hierarchical ordered components.⁴

1. **OP (Observation Program):**
The observation program contains the entire scientific relevant data of scientific goals and instrument relevant parameters. It can contain multiple sets of observation targets.
2. **OTU (Observation Target Unit):**
An observation target unit consists of parameters to evaluate and execute the underlying observation blocks. Also the observation blocks corresponding execution constraints are included. The OTU is a subset of the OP and contains all informations for one target.
3. **OB (Observation Block):**
The observation block is the smallest executable unit. Each one of these units have to be scheduled during the operational process. Usually in this data structure only instrument relevant data is included.

For detailed information on the data structure please see A. I. Pavlov et.al.⁵

3. OPERATIONAL SCENARIOS

At the LINC-NIRVANA interferometer there are mainly two special characteristics which define the operational process chain.

3.1. Observation Modes

In current astronomical observation, instruments provide two different operation modes. First an instrument can be operated in a so called **visitor mode**.⁶ In this mode the observer is physically present to execute the predefined observations. If the observation does not provide the expected scientific results to the observer, parametric changes of the observation setting can be applied.⁷ The observation can be redone or adopted until expectations are fulfilled. Usually the observer has direct access to the instruments highlevel settings. On less complex instruments the observer is able to use the knowledge about the instrument to adopt and optimize the instruments behavior for the scientific desires. As LINC-NIRVANA reaches a certain point of complexity and time constraints, to ensure an economical operation, the observation can be processed only by a trained operator, who is familiar with the instrument and its characteristics.⁸ The observing scientist delivers a set of observation requests. The operator decides, depending on the instrument and environmental constraints, which observation can be processed at what time. In this mode a detailed preparation and analysis is required to get an optimum amount of high quality scientific data. The goal for LINC-NIRVANA is to provide a semi like operation service. In this **service mode** the work procedure is quite different. A scientist will be present at the observatory and is performing observations for several scientists, who prepare their observations with the Observation Preparation Software to a level of detail, at which a third person can execute the observation safely. In this case the system has to automatically analyse certain instrument and environmental constraints and find the optimal execution start time. Therefor a highly optimized scheduling procedure is required.

3.2. Instrument characteristics

The instrument itself has characteristics which make an exact time planing necessary. To fully reconstruct images with the Data Reduction Software, which provide the highest possible spatial resolution in all image directions, a set of at least three single images, taken at different parallactic angles, is needed. Therefor an observation target unit has to be divided into at least three OBs, which have to be executed at a different time. OBs from different observation target units have to be chronological arranged to achieve a maximum amount of observations during the visit periode.

3.3. Operational Constraints

To accomplish the maximum information quality, certain constraints have to be considered. Basically these constraints can be divided into dynamical and statical constraints. The dynamical constraints are those, which can unpredictable change during an observation e.g. the seeing or the strehl ration. During the observation the instrument provides those informations, which will be used to validate predefined constraints of future OB execution.

- Average Seeing
- Strehl Ratio
- Signal-To-Noise Ratio

On the other hand the static constraints can also change in time but can be predicted like the airmass to the target object. The following list shows the currently considered static constraints.

- Multiple Instrument setups (i.e filters)
- Multiple Parallactic Angles
- Air Mass

- Moon angular distance

Each of the static constraints can be defined as a time interval during the observation in which a constraint is fulfilled. The sum of them gives a so called time slot for each OB. These time slots will be calculated in advance and will be used to perform the scheduling for one observation night (see fig. 2).

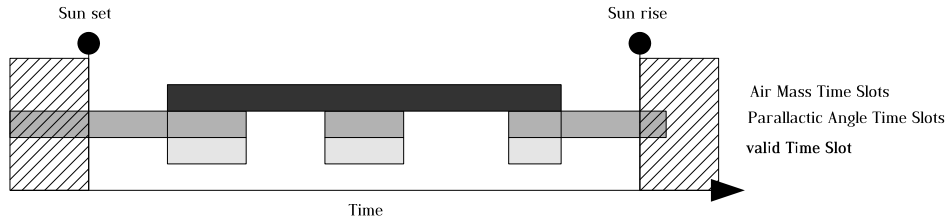


Figure 2. The figure above shows the definition of the time slots resulting from the static constraints air mass and parallactic angle. The overlapping time intervals define the valid time slots for an OB.

Static constraints like the air mass and the desired parallactic angle therefore can be defined by time intervals, in which all those environmental conditions are within a valid range.

4. SYSTEM OVERVIEW

The system can be mainly divided into two major processing units. First of all these units, which can be used offline. Offline process like preparing the observation program via the OPS and then doing a prescheduling can be done in advance without any feedback from the instrument. During this process, high level parameters in terms of scientific and instrumental requirements are defined. A detailed observation plan can be prepared. Environmental constraints defined from the observer via the OPS are not considered in the prescheduling process.

The second part is the online processing unit. This units contain the scheduling and the execution process. Both of them can only be processed during the observation itself. A predefined environmental constraint set from the OPS are evaluated by analysing the feedback of the instrument. The process of permanent evaluation of the instrument feedback and comparing it to the constraints requires a reactive scheduling.

4.1. Dataflow

The flow of the data from the OPS to the OSS including the scheduling unit and execution unit is described in Figure 3. After the observer has defined the entire OP, the data is stored in a data storage unit (repository). This unit is a central unit accessible from either the OPS or the OSS for data exchange and process logging. When the observation process has started, either in advance for a prescheduling or during the observation time, one or more OTUs are loaded from the data storage to create a schedule for the included OBs. The defined schedule will be stored again in the data storage to eventually edit the schedule for future use. If the schedule is ready for execution, each OB is send to the execution process when an OB is next to be executed. During this process, information about the execution state and the current environmental situation, is collected and fed back to the scheduling process to do a rescheduling if necessary.

4.2. Workflow

The workflow of a typical observation, usually done during an one night observation, is shown in Figure 4. Three major steps will be applied to the procedure. First a set of observation target units (OTU) is selected from a data source containing the predefined observation program (OP). Each OTU consists of one or more observation blocks (OB) depending on the amount of parallactic angles needed for the scientific goal. The OTUs are loaded into the scheduling software unit,⁹ in which the OBs will be chronologically arranged in the optimal time order. After the schedule is set, the observation will start by executing each OB sequentially. During the execution of an OB, the environmental situation is monitored, e.g. the seeing, which is part of the constraints.

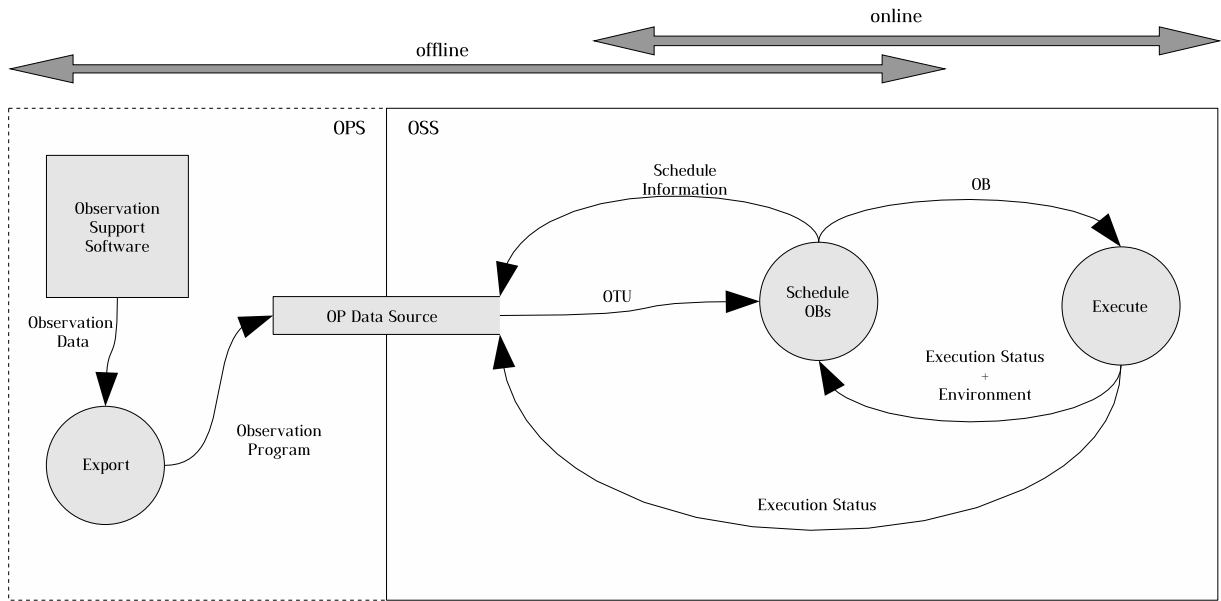


Figure 3. The figure above shows the data flow from the Observation Preparation Software through a Data Storage most likely a relational database management system to the OSS.

If an OB was executed, the detected conditions will be used to analyse the constraints for the next executable OB. If the conditions are out of the desired range, the scheduled execution sequence will be adopted to the new circumstances. Another aspect of rescheduling is the execution status of the OBs. If one OB is executed successfully, all remaining OB of the same OTU get higher prioritized in the execution sequence. This has to be done to achieve the goal of completeness for the entire OTU.

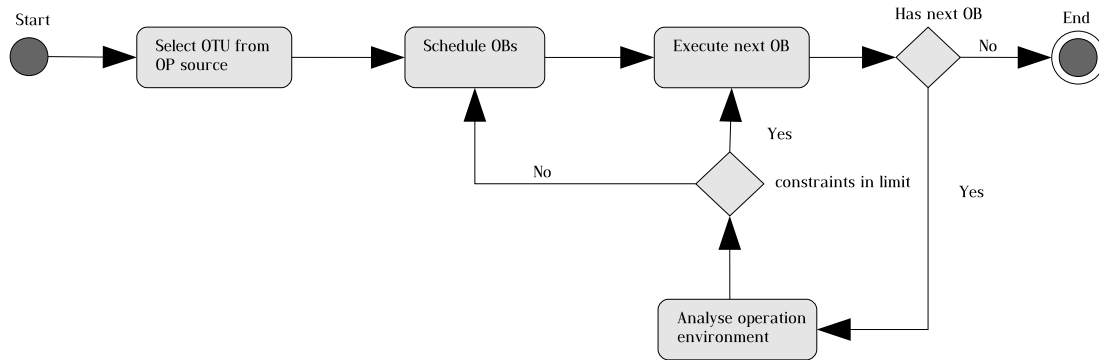


Figure 4. The figure above shows the workflow of a typical observation sequence. Exceptions are not handled in this diagram.

5. REACTIVE SCHEDULING

Providing an appropriate sequence of OBs for a defined time interval, usually one observation night, a mechanism for preordering the OB in a way, that the workload of the telescope is optimized, has to be found. As seen in figure 4 the scheduling process has to be run only once if all predefined constraints stay in limit over the entire

night. In case the constraints of upcoming OBs are not in limit, an on-the-fly rescheduling has to be done.⁹ This permanent rescheduling called 'Reactive Scheduling' has to provide methods, which fulfill the following criteria:

- High-performance, because of rapid reactive scheduling.
- Transparent to the observer (operator).
- Consistant in terms of the scientific goals.

5.1. Strategy

The goal of an optimized scheduling strategy is to find a sequence of OB executions, which have the following criteria:

- min. collisions between OBs.
- max. executable OBs.
- min. telescope wait time between two executions.

The simplest way to find the optimal arrangement of OBs, is to do a brut force search over all possible schedulable pathes. The disadvantage of this method is the very high need of computing power. The tolerance (Tol) of each observation block is defined by the possible placements of an execution within the OBs time slot.

$$Tol_n = SlotEnd_n - SlotStart_n - ExposureTime_n$$

The amount of possible schedulable pathes (NPath) is described as following:

$$NPath = Tol_1 * Tol_2 * Tol_3...Tol_{n-1} * Tol_n$$

The goal of all scheduling methodes, is to find the path within all possible pathes, which fullfills the three criterias described above as best as possible. In case of just ten observation blocks, with a tolerance of ten minutes for each time slot, the needed calculation cost exceeds the power of todays workstations and does not fullfill the demand of rapid reactive rescheduling, whenever a constraint is not in the valid range¹⁰ (see section 3.3). For this reason a compromise between the quality of the scheduling strategy and the complexity has to be found.¹¹ The strategy which was chosen to achieve this goal, is a combination of sorting methods known from computer science and additionally fitting methods, which is used in case of a scheduling conflict. The sorting methods are in general used to assign hardware resources like processor time in multiple process/thread environments. In case of assigning the telescope resources to observation blocks, most of the nonpreemptive scheduling methods can be used for the sorting process, while the combination of sorting and fitting defines a more effective work sequence of OBs, than just by defining the sequence manually. For LINC-NIRVANA the all over process of finding the optimal sequence of OB execution is defined by two subprocesses. First the list of unsorted OBs have to be sorted in a chronological way used by methods shown in section 5.2. After the scheduling order has been defined a fitting process is started, in which the algorithms tries to serialize as many as possible OBs by shifting the execution time within the defined time slots of each OB (see section 5.3). Figure 5 shows the processing pipeline.

5.2. OB Sorting

1. Priority:

In advance all OPs and their containing OBs have to be divided into several scientific ranking bands¹¹ by a Proposal Acceptance Committee. During the first sorting step all OBs are ordered in the descending order of their priorities. This way it is guaranteed that the highest prioritized observation block will be scheduled first. Also all OBs of an OTU have to be grouped together to ensure that the entire OB set will be executed prioritized.

2. FIFO (First In, First Out):

To minimize instrument wait time at the beginning of the observation period, the observation blocks are ordered by its time slot start time. This method describes the principle of a queue or first-come, first-served (FCFS) behavior. The OB which comes in first is handled first, what comes in next waits until the first is finished, etc. In case of the observation scheduling, each OB will be executed as soon as possible. If a previous OB execution is delayed, the maximum time buffer at the end of an OB execution is guaranteed. The risk of running out of time for an execution is therefore minimized. Each set of OBs with the same priority is sorted separately.

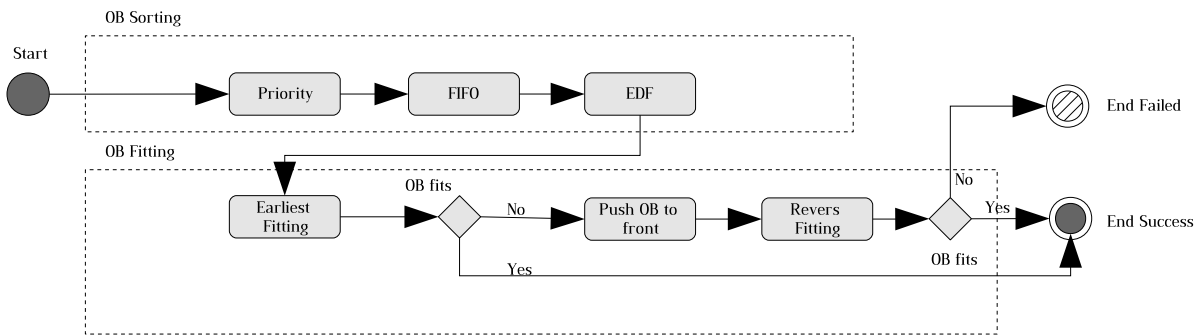


Figure 5. Processing pipeline on the entire scheduling process. The process is divided into two major parts, the sorting and the fitting.

3. EDF (Earliest Deadline First):

EDF is a very common used scheduling method in computer science. In case of LINC-NIRVANA this method is used, whenever the start time of a time slot of more than one OB of the same priority set are equal. The scheduling order by just using the FIFO method would cause ambiguities. The result of EDF causes the observation block which has the tightest deadline to have virtually a higher priority. The danger of running out of time for an OB execution minimizes therefore. Figure 6 shows a set of three OBs. All of them have the same priority and have been ordered by FIFO and EDF. As it is shown the second and third OB have an ambiguity, thus to minimize the risk of running out of time for execution of the third OB, the second OB is put first into to queue.

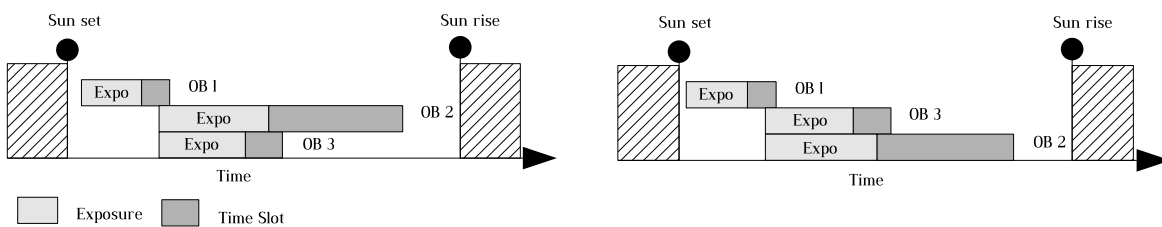


Figure 6. The figure above shows a set of three OBs with the same priority, which have been preordered by the FIFO (left) and then additionally by the EDF (right) method.

5.3. OB Fitting

1. Earliest Fitting

Within the previous scheduling methods it is most likely that conflicts between two or more OB will occur, because all execution times are still set to the beginning of the time slot and have not been validated against each other. Until this point only the order of OB has been defined. Now those OBs execution times

have to be arranged within the time slots, that the amount of conflicts will be minimized. If a conflict between the already scheduled OBs and the OB execution, which is about to be inserted into the schedule, appears, an earliest fitting process is stated. The upcoming execution will be shifted to the first possible fitting in the scheduling line. If at the end of this process no conflict between the upcoming OB execution and the previous scheduling happen, the execution will be inserted into the valid scheduling line. Figure 7 shows this process in detail.

The sequence in Figure 7 shows the fitting process of an upcoming schedulable OB into an already scheduled time line. The time line is basically defined by a sequence of atomic time slices e.g. into seconds. The fitting analyses chronologically each time slice. If a conflict is found (Step 1), the entire execution time will be added and shifted to the next time slice (Step 2). If a conflict does not appear (Step 3) any more, one shifted time slice stays and the rest is shifted further on (Step 4). Each time a non conflicted time slice is found, one time slice remains in that field and the remaining time slices will be shifted further. If at the end of the upcoming OB time slot the rest of the summed up time is zero or one, the first slot to fit into was found. If one or more time slices have a higher value than one, the execution time can not be fitted into the schedule time line.

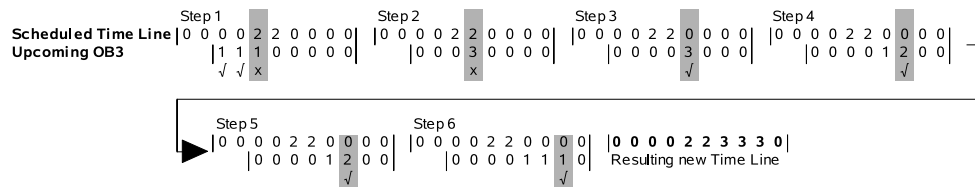


Figure 7. The sequence of the earliest fitting process. In this example '0' is the placeholder for an unused time slice, '1' is the id of an already scheduled OB in the scheduling time line and '1' a currently occupied time slice in the upcoming OB.

2. Reverse Fitting

If all the above strategies lead into a conflict, the reverse fitting method comes in place. The reverse fitting method puts the upcoming non schedulable OB into the first place of the ordered OB list (see Fig. 8 left), which was ordered by priority, FIFO and EDF. Then the list from the first until the last successfully scheduled OB is rescheduled by the above described fitting method(see Fig. 8 right). If the entire OB set is scheduled without conflict, the upcoming OB is set to the time line. If not, the upcoming OB is not schedulable within this sequence.

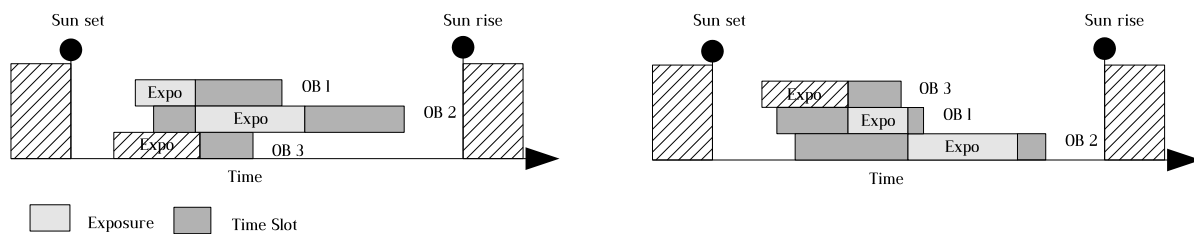


Figure 8. On the right hand side OB3, which has a lower priority, will end up into an unresolved conflict with OB1 and OB2. If OB2 and OB3 are shifted, OB3 can be fit inside the schedule, by changing the scheduling order of the OBs. Therefore the order of the OBs to schedule has to be changed.

6. CONCLUSIONS

Since the instrument LINC-NIRVANA is a highly complex instrument, which has to deal with special constraints, like multiple parallactic angles per observation, a mainly automatized scheduling algorithm has to be included into the working process. A sequence of sorting and fitting algorithms have been implemented and partially evaluated. The used algorithms were chosen most reasonable, to give the observer the most transparent overview of the process. The decisions made by the scheduling process can be easily evaluated by the observer and eventually can be customized by changing parameter inside the OB to force a rescheduling. More complex and optimized scheduling algorithms definitely result in a higher performance, what is defined by the lower rate of non schedulable OBs, but will be less transparent and more expensive in terms of computing time.

Until this point of the OSS software developing process, the above described strategy was implemented in a prototype. Further on, the strategy will be evaluated in more detail and eventually be adopted to more constraints, which the instrument will require in future.

The current status of the system describes a first step to a more automatized and performance optimized observation procedure for LINC-NIRVANA.

7. ACKNOWLEDGMENTS

We are grateful to the entire LINC-NIRVANA team at the Max-Planck Institute in Heidelberg (Germany), MPI for Radioastronomy in Bonn (Germany), University of Cologne (Germany), Astronomical Observatory of Rome in Monte Porzio Catone (Italy) and Arcetri Astrophysical Observatory in Firenze (Italy) for their professional astronomical and engineering support.

REFERENCES

1. T. M. Herbst, R. Ragazzoni, A. Eckart, and G. Weigelt, "The LINC-NIRVANA interferometric imager for the large binocular telescope," 2004. Proceeding of the SPIE Conference, vol. 5492-40.
2. J. Hill and P. Salinari, "The large binocular telescope project," 2004. Proceeding of the SPIE Conference, vol. 5489.
3. J. Bresina, M. Drummond, K. Swanson, and W. Edgington, "Automated management and scheduling of remote automatic telescopes," 1993.
4. W. Gaessler, "LINC-NIRVANA Final Design Review, operations, LN-MPIA-FDR-OPS-001."
5. A. Pavlov, "Observation preparation software for LINC-NIRVANA," 2006. Proceeding of the SPIE Conference, vol. 6270-39.
6. K. L. Casertano, "Service observing and data quality control: Some lessons learned from hubble space telescope."
7. A. Chavan, G. Giannone, D. Silva, T. Krueger, and G. Miller, "Nightly scheduling of ESO's very large telescope," 1998.
8. A. Maurizio, "Support tools for the VLT operations: the NTT prototyping experience."
9. M. D. Johnston, "Scheduling tools for astronomical observations."
10. K. Swanson, J. Bresina, and M. Drummond, "Robust telescope scheduling," 1994.
11. P. Puxley, "Queue mode observations," tech. rep., GEMINI Observatory, 2001.