



VieSched++: A New VLBI Scheduling Software for Geodesy and Astrometry

Matthias Schartner¹  and Johannes Böhm 

Technische Universität Wien, Vienna, Austria

Received 2019 February 17; accepted 2019 April 3; published 2019 June 18

We present a new Very Long Baseline Interferometry (VLBI) scheduling software called VieSched++, which is a stand-alone tool of the Vienna VLBI and Satellite Software (VieVS). The scheduler is written in C++ and aims to be flexible and easy to use, with a modern graphical user interface while creating high-quality schedules. In this work, the general design concepts of the scheduling software are discussed and the major scheduling algorithms are explained. Additionally, deep insight into the optimization criteria is given. First tests demonstrate that VieSched++ is able to generate schedules of highest quality. The software can be downloaded from <https://github.com/TUW-VieVS>.

Key words: VLBI – scheduling

Online material: color figures

1. Introduction

Very Long Baseline Interferometry (VLBI) is one of the main space geodetic techniques with essential contribution for the determination of Earth Orientation Parameters (EOP) or the scale of the Terrestrial Reference Frame (TRF); see Schuh & Böhm (2013) or Sovers et al. (1998). Moreover, VLBI is unique in determining the Celestial Reference Frame (CRF) at radio frequencies (Fey et al. 2015), and is often used in astronomy as well. Typically, a geodetic or astrometric VLBI session lasts for either 1 or 24 hours with a fixed network of antennas. The first step of creating a VLBI experiment is to create an observing plan, the so-called schedule, which is necessary to coordinate the observations to make sure that at least two stations observe the same source simultaneously. Essentially, the schedule is a table that lists for each station which source should be observed at what time and how the observations should be recorded. This observing plan is then sent to the participating stations for observation. Thereafter, the session is correlated and a process called fringe-fitting is applied to prepare the data for the analysis software. Therefore, the schedule directly determines the observations, which can be used for parameter estimation and great care should be taken when generating schedules to get the best possible result.

While several software packages exist for analyzing geodetic or astrometric VLBI sessions, like Calc/Solve (Bolotin et al. 2014),

the Vienna VLBI and Satellite Software (VieVS; Böhm et al. 2018), OCCAM; Titov et al. 2004), c5++ (Hobiger et al. 2010), Where (Hjelle et al. 2017), or GINS (Bourda et al. 2007), the number of software packages that can create schedules of a geodetic or astrometric VLBI session is considerably smaller. A big variety of different analysis software is appreciated because it helps to verify the results, find and trace down bugs that might exist and also average out systematic inconsistencies between different implementations. Unfortunately, there are very few scheduling programs. Almost all of the geodetic VLBI schedules are generated using sked (Vandenberg 1999), while astrometric schedules are scheduled using Sched (Walker 2018). Astrometric schedules can be created using sked or Sched. For geodesy and astrometry, the only notable exception are AUSTRAL sessions (Plank et al. 2017b), which were scheduled using a Matlab-based scheduler (Sun 2013), which is part of VieVS. The existence of a new geodetic and astrometric VLBI scheduling software might increase the quality of the schedules, which will lead to improved observations and better results.

In the following work, a schedule always refers to a geodetic or astrometric schedule. Section 1.1 gives a brief overview of the challenges for scheduling VLBI sessions, while Section 1.2 discusses the design concepts of the new scheduling software VieSched++. The newly developed algorithms are presented in Section 2 covering optimization criteria 2.1, recursive scan selection 2.2, multi-scheduling and automatic source selection features 2.3, station-, source-, and baseline-based parameterization 2.4, and finally astrometric optimization 2.5. Finally, some initial results achieved by using VieSched++ are listed in Section 3 and an outlook and future plans are shown in Section 4.

¹ Corresponding author.



1.1. Scheduling

A typical geodetic schedule contains 100 sources and consists of roughly 1000 scans, which leads to 100^{1000} theoretical schedules. It is the scheduler's task to select one of those schedules that will then be used for the session. The challenge is to select one of those theoretically possible schedules, which will lead to good results. In reality, the total number of practical schedules is considerably less, due to the fact that typically not all sources are visible for each station at every time and many other conditions need to be applied. However, it is still large enough that it can be exploited for optimization and creating a good schedule is a serious challenge.

When a schedule is prepared, it is up to the scheduling software to decide which sources should be observed in which sequence and for how long. The software is creating the schedule scan after scan. It tries every possible combination of upcoming scans at the current scheduling time, which can be hundreds or even thousands of possibilities. Then it compares those possible scans using some metrics like the total duration of the scan, the improvement in sky coverage, and the number of observations; see Section 2.1. The scan that is considered the most beneficial based on those metrics is then selected and scheduled. After a scan is scheduled, this process starts over again and the steps are repeated until the schedule is finished. More information about the general scheduling concept of the scheduling software sked can be found in Gipson (2010) and for the Matlab VieVS scheduler we refer to Sun (2013).

Several different optimization criteria need to be considered with some contradicting each other. One example in geodesy is the benefit of having an evenly distributed sky coverage at each station versus the benefit of having a large number of observations. An evenly distributed sky coverage, especially at low elevations, helps to estimate tropospheric delays (Schuh & Böhm 2013) which are considered one of the biggest error sources in geodetic VLBI (Boehm et al. 2006), while a larger number of observations improves the accuracy of estimated geodetic parameters due to the higher redundancy. Both factors are very important for getting accurate geodetic results, but the optimization of the sky coverage would make the stations slow long distances between scans, while for achieving the maximum number of observations the slew time between scans should be kept low. This topic is also discussed in (Gipson 2010). One challenge is to find a sweet spot between those competing factors. Besides that, a typical network consists of antennas with different attributes making it necessary to consider different slew speeds, sensitivities, visibility areas, and antenna mounts. Additionally, the cable wrap has to be monitored for each station.

In the case of global geodetic VLBI, the network is often divided into subnetworks and multiple sources are observed simultaneously (Petrov et al. 2009). This concept is called

subnetting and further increases the complexity of the optimization problem exponentially.

1.2. Concept

The Department of Geodesy and Geoinformation (GEO) at Technische Universität Wien (TU Wien) is developing the Vienna VLBI and Satellite Software (Böhm et al. 2018). Part of this software is a VLBI scheduler written in Matlab (Sun 2013) and previously used for scheduling VLBI sessions with the Auscope network in Australia (Lovell et al. 2016), plus Hartebeesthoek in South Africa and Warkworth in New Zealand (Mayer et al. 2015). Based on the knowledge gained while developing and maintaining this tool, we have started to develop a new scheduling software from scratch. The new scheduling software is called VieSched++ and is written in modern C++ using an object-oriented software design. It is a stand-alone tool under the umbrella of VieVS. Although it is a stand-alone tool and Matlab is not required, the interaction between VieSched++ and the VieVS VLBI module is straightforward and results from the scheduler can be directly used in VieVS. The source code of VieVS including VieSched++ and the VLBI module is freely available at our GitHub page: <https://github.com/TUW-VieVS>.

VieSched++ consists of two packages, the scheduler (<https://github.com/TUW-VieVS/VieSchedpp>) and a graphical user interface (GUI; <https://github.com/TUW-VieVS/VieSchedppGUI>). The scheduler is controlled by an XML document that can be created either by hand or with the GUI. It supports all necessary features to create high-quality geodetic schedules, like automatic source selection with or without subnetting, tagalong-mode, fillin-mode, and scan duration estimation, as well as various optimization criteria like optimization of sky coverage and many more; see Section 2. Tagalong-mode is a technique that allows the addition of a station without influencing the performance of the remaining network in case the station tagged along is not able to participate during the session. This is especially useful for new stations that are not yet fully reliable. The schedule is first created without considering the tagalong station. After the schedule is finished, the tagalong station is added to the existing schedule. VieSched++ provides the option to also use tagalong-mode during station down times; see Section 2.4. Fillin-mode is a technique that is reducing the idle time of the stations. Different stations can have highly variable observing and slew times based on the antenna characteristics, therefore idle time can occur. If the fillin-mode is enabled, it is checked to see if it is possible to schedule additional scans using a reduced network during these idle times. For more detailed information about fillin-mode, see Section 2.2.

To be consistent with today's schedules created with sked, the sked catalogs (Vandenberg 1997) can be used as input parameters. The output is available in .skd and .vex format.

However, some of the newly created features, like the use of multiple observing modes and frequency setups, as well as different scan time alignments, can only be represented in .vex format using VieSched++. In general, the use of multiple observing modes would also be possible using .skd files, but this feature is not yet supported. Besides the schedule output, additional statistics and log files can be created as well as NGS observation files that can be directly used in VieVS for simulations.

The GUI is based on the Qt5 libraries and aims to be very intuitive and highly interactive. It provides a built-in help and multiple interactive schedule analysis and comparison features.

Both packages are easy to install and great care was taken to reduce the number of third-party libraries to a minimum. Besides the SOFA routines (Board 2019), no other prebuild library is necessary for the scheduler, although some features like improved log files are only available if you have a build version of the C++ BOOST (Ling 2011) libraries. The scheduler can be built easily using a provided cmake script file. The GUI additionally requires the Qt5 Libraries, and can be build using a qmake script file. VieSched++ is a multi-platform application and has been successfully tested on Linux and Windows using different compilers. While the installation on Linux is very easy and straightforward, installation on Windows can be a bit trickier. Therefore, we provide an executable program at our GitHub page (<https://github.com/TUW-VieVS/VieSchedpp/releases>), which solves this issue.

2. Algorithms

All algorithms are developed from scratch while they are carefully tested against previous implementations. In the following subsections, some major differences compared to other software packages like sked or the Matlab VieVS scheduler are shown and discussed.

2.1. Optimization Criteria

As discussed in Section 1.1, a schedule is typically created scan after scan. To select a scan, the software needs to know the time and antenna pointing at the end of the previous scan. Based on this information the next scan can be chosen. Similar to sked and the Matlab VieVS scheduler, VieSched++ uses a brute force approach to select the following scan (i.e., all possible next scans are calculated and evaluated). To select the best scan out of this pool, several optimization criteria are used. Every possible scan gains a score per optimization criterion, $score_{opt}$. The total score, $score_{scan}$, is the weighted sum of the scores per optimization criterion. Weight factors, $weight_{opt}$, can be used to combine multiple criteria similar to “minor options” in sked; see Gipson & Baver (2016) or Gipson (2010):

$$score_{scan} = \sum_{i_{opt}=0}^{n_{opt}} weight_{i_{opt}} \cdot score_{i_{opt}}. \quad (1)$$

The higher the weight $weight_{opt}$, the more the individual score $score_{opt}$ contributes to the total score $score_{scan}$. In the case of subnetting, additional combinations of two scans to two sources are created and evaluated. This approach is useful if a schedule for a global network is created. The amount of possible next scans and scan combinations can be between several hundreds and many thousands, based on the settings, source list and if subnetting is allowed or not. In the case of subnetting scans, the score is calculated for each scan individually and summed up:

$$score_{subnetting} = score_{scan_1} + score_{scan_2}. \quad (2)$$

The scan or scan combination with the highest score is selected and scheduled as the next scan.

Currently, VieSched++ supports nine optimization criteria; however, it is planned to add additional criteria (see Section 4):

- duration,
- sky coverage,
- number of observations,
- idle time,
- average stations,
- average sources,
- average baselines,
- low elevation, and
- low declination.

Choosing a good set of optimization criteria and weight factors are the key to creating an optimized schedule (Schartner et al. 2017).

Defining optimization criteria is tricky because the criteria should be generally valid no matter which network and stations are used. Additionally, it should be possible to compare single source scans with subnetting scans objectively. Because optimization criteria are the most critical part of the scheduling software, there are regular changes and improvements to the algorithms. The following subsections list a snapshot of the current implementation in VieSched++ Version *a585502*.

2.1.1. Duration

The optimization criterion for the duration of a scan is one of the trickiest to define, while one of the most important for a good schedule. For fast slewing VLBI Global Observing System (VGOS) antennas with high recording rates, observing a scan takes far less time than for slow slewing antennas with lower recording rates. The same is true for fillin-mode scans that include a reduced number of stations and are typically faster to perform. The idea behind this optimization criterion is that a high number of scans and a reduced slew time lead to more observations and better results. Therefore, scans that take shorter times to observe should gain a higher score in the evaluation. Preferring scans with a short duration leads to an increased number of scans, reduced slew times, and—depending on the

recording rate—to more scans to strong sources. The duration, dur , includes everything between the end of the previous scan until the end of the tested scan, which means it includes time for field system commands, slew time, possible idle time, calibration time, and observing time. In this context, the “end of the previous scan” is defined as the earliest time a in this scan participating station stops recording its previous scan. The score, $score_{dur}$, 3 is based on the minimum, dur_{\min} , and maximum required times, dur_{\max} , for all possible next scans as well as on the duration, dur , of the tested scans. Finally, a scale factor is applied based on the number of participating stations, n_{sta} . This scaling is necessary for combining single source scans and subnetting scans. Assuming a scan with all stations, the scan with the shortest duration gains a score of 1 while the scan with the longest duration gets a score of 0. Linear interpolation is used in between:

$$score_{dur} = 1 - \frac{dur - dur_{\min}}{dur_{\max} - dur_{\min}} \cdot \frac{n_{sta}}{n_{sta_{\max}}}. \quad (3)$$

2.1.2. Sky Coverage

An evenly distributed sky coverage helps to better estimate tropospheric delays, which are one of the major error sources in VLBI (Boehm et al. 2006). Therefore, a schedule can be optimized by planning observations at different azimuth and elevation angles over a short period of time.

There are several parameters that can be used to define the sky coverage: a distance function, $dist_{fun}$; an influence distance, $dist_{inf}$; a time function, $time_{fun}$; and an influence time, $time_{inf}$. Currently, there are three possible distance and time functions called *cosine*, *linear*, and *constant*.

The basic idea is that the antenna pointing at each station for this scan is compared to all scheduled antenna pointings of previous scans, i_{prev_scan} . Depending on the time and angular distance a score per antenna pointing, $sat_{dist_{i_{prev_scan}}}$ and $sat_{time_{i_{prev_scan}}}$ is calculated. Equation (4) lists the possible functions used to describe the dependency of the angular distance. This equation is used to calculate the saturation sat between the new observation and a previous observations i_{prev_scan} . The same formulas can be used for the dependency of the time:

$$sat_{dist_{i_{prev_scan}}} = \begin{cases} 0 & dist > dist_{inf} \\ dist_{fun}(dist) & dist < dist_{inf} \end{cases} \quad (4)$$

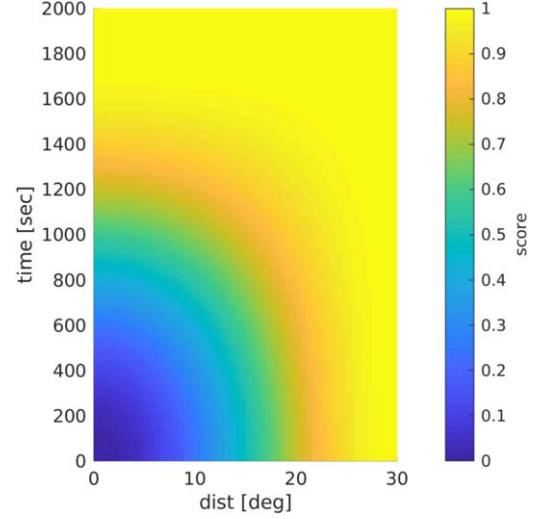


Figure 1. Example of sky coverage score function, $score_{i_{prev_scan}}$, from Equation (6) when a *cosine* function is used for both, the angular distance and the time, with $dist_{inf} = 30^\circ$ and $time_{inf} = 1800$ seconds. (A color version of this figure is available in the online journal.)

with

$$dist_{fun}(dist) = \begin{cases} 1 & dist_{fun} = constant \\ 1 - \frac{dist}{dist_{inf}} & dist_{fun} = linear \\ \frac{1}{2} + \frac{1}{2} \cdot \cos\left(\frac{dist \cdot \pi}{dist_{inf}}\right) & dist_{fun} = cosine \end{cases}. \quad (5)$$

Based on the saturation due to the time difference and angular distance to one previous observation, the score compared with this previous observation $score_{i_{prev_scan}}$ can be calculated:

$$score_{i_{prev_scan}} = 1 - (sat_{dist_{i_{prev_scan}}} \cdot sat_{time_{i_{prev_scan}}}). \quad (6)$$

Figure 1 illustrates an example how the score function could look like.

The total score per sky coverage, $score_{sky_{ista}}$, is the minimum score of each previous scan, $score_{i_{prev_scan}}$:

$$score_{sky_{ista}} = \min(sat_{i_{prev_scan}}). \quad (7)$$

Finally, the total score for the sky coverage optimization criterion is the sum of the improvements for each sky coverage divided by the number of participating stations in this session $n_{sta_{\max}}$. The normalization by the number of participating stations ensures that the maximum achievable score is 1:

$$score_{sky} = \sum_{i_{sky}=0}^{n_{sky}} score_{sky_{i_{sky}}} \cdot \frac{1}{n_{sta_{\max}}}. \quad (8)$$

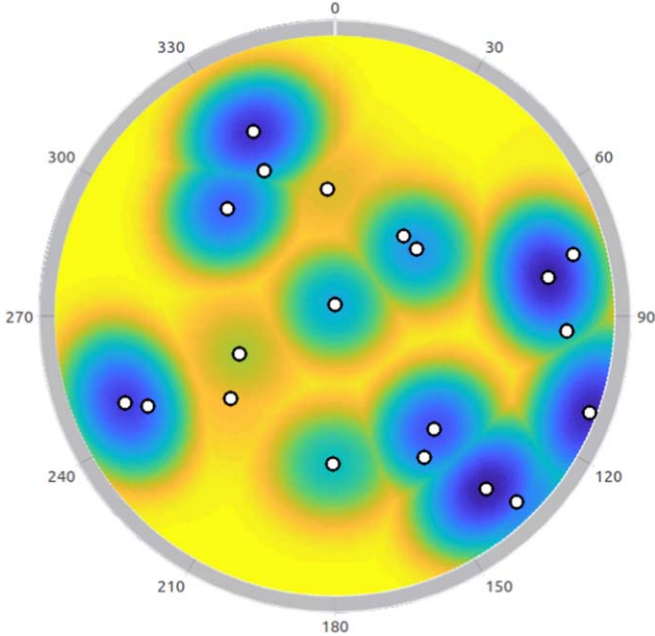


Figure 2. Example of sky coverage score, $score_{sky}$, from Equation (8). Observations in areas that are marked blue (dark) would gain a low score, while observations in areas that are marked yellow (bright) would gain a high score. Previous observations are marked with circles. The same functions and parameters were used as in Figure 1.

(A color version of this figure is available in the online journal.)

This means if all stations are participating in a scan and all observe a source at a point of the sky where no observations happened for some time, this scan would get the highest possible score of 1. Figure 2 illustrates possible sky coverage scores.

VieSched++ also supports the use of twin telescopes and more general multiple telescopes per observing site. In case of multiple telescopes per observing site, all of these stations share the sky coverage information with each other.

2.1.3. Number of Observations

A rule of thumb for evaluating a good schedule suggests that the higher the number of observations, the better the schedule (Gipson 2010). The score for the number of observations, $score_{n_{obs}}$, optimization criterion is based on the total number of observations, n_{obs} , per scan and the maximum theoretically possible number of observations, $n_{obs_{max}}$, assuming a scan with all stations, $n_{sta_{max}}$. While the maximum number of observations theoretically possible is defined by Equation (9), the actual number of observations per scan can be lower in VieSched++:

$$n_{obs_{max}} = \frac{n_{sta_{max}} \cdot (n_{sta_{max}} - 1)}{2}. \quad (9)$$

In VieSched++, it is possible that a scan has a lower number of observations due to the way how stations are removed from

a scan. There are several reasons why to remove a station from a scan, one of which is due to too long observing times. The maximum allowed observing time duration can be set through the parameters. If an observation needs more time to reach the target signal-to-noise ratio (S/N) than the maximum allowed observing time, VieSched++ removes this observation from the calculation and from all upcoming statistics and evaluations instead of removing all observations at the station. A station is only removed from a scan as soon as all observations with this station get removed. Additionally, it is possible to define for each source if observations at certain baselines should be ignored. This is necessary to imitate the Star scheduling mode (McCallum et al. 2017). However, it should be noted here that most of the time all or nearly all possible observations are scheduled. Only in rare scenarios when using stations with low sensitivity, low recording rates, or weak sources a notable amount of observations are removed. Extreme cases are schedules for the T2 campaign, which are using a 128 MBit/s recording mode and some stations with an SEFD of 18,000 (S-Band Oh), 10,000 (X-Band Oh), and 13,000 (S-Band Vm); see Section 3. Even then, only less than 5% of the observations are removed. When using a higher recording rate and more sensitive stations, this value decreases even further. However, using this technique helps integrating low sensitive antennas by removing only observations between low sensitive antennas, instead of removing the low sensitive antenna from the scan.

The score, $score_{n_{obs}}$, is defined through the ratio of observations, n_{obs} , at this scan and the maximum possible observations, $n_{obs_{max}}$, assuming a scan with all stations; see Equation (9):

$$score_{n_{obs}} = \frac{n_{obs}}{n_{obs_{max}}}. \quad (10)$$

2.1.4. Idle Time

The score for the idle time optimization criterion, $score_{idle}$, is calculated per station. Two parameters are used to calculate it: a parameter, the so-called idle time interval, $interval$, and the idle time per station, $idle$, before this scan. The idea of this optimization criterion is to give stations that have not been observing in while a higher weight.

The idle time before this scan $idle$ is defined through Equation (11). The parameter $current_time$ refers to the end time of the last observation of a station, which is the current time for this station viewed from the scheduler:

$$idle_{ista} = \max(current_time) - current_time_{ista}. \quad (11)$$

This idle time is divided by the $interval$ time. Therefore, stations that have not been observing for a duration of $interval$ gain a score of 1, while stations that have not been observing for a duration of $2 \cdot interval$ gain a score of 2 and so on. The longer the station has not been observing, the higher the score is and the more likely it is to pick a scan with this station again.

The total score is gained through the sum of all scores per participating station:

$$score_{idle} = \sum_{i_{sta}=0}^{n_{sta}} \frac{idle_{i_{sta}}}{interval}. \quad (12)$$

Therefore, the *idle* time is completely independent of the tested possible next scan. This is necessary because otherwise scans that begin later would gain a higher score due to more idle time, which is usually unwanted because it is in direct contradiction to the score based on the duration 2.1.1.

It is also noted here that unlike all other optimization criteria scores, which have a maximum score of usually 1, this score can grow without bounds. This means that the longer a station is not observing the higher the total score for scans with this station gets, no matter which weight factor is used. Therefore, this criterion is especially important for asymmetric station networks because it makes sure that all stations observe regularly.

2.1.5. Average Stations

The idea behind the average stations optimization criterion is to increase the weight of stations with a lower number of scheduled observations compared to the average number of scheduled observations per station. If a station has fewer observations than the average station, it gets a score, $score_{asta_{ista}}$, based on the difference between the average number of scheduled observations per station, $n_{obs_{mean}}$, and the number of scheduled observations for this station, $n_{obs_{ista}}$:

$$score_{asta_{ista}} = \begin{cases} 0 & n_{obs_{ista}} > n_{obs_{mean}} \\ \frac{n_{obs_{mean}} - n_{obs_{ista}}}{n_{obs_{mean}} - \min(n_{obs_{ista}})} & n_{obs_{ista}} < n_{obs_{mean}} \end{cases}. \quad (13)$$

Therefore, stations with the lowest number of scheduled observations get a score of 1, while a station with an average number of scheduled observations or more would get a score of 0. Linear interpolation is used in between.

The total score of the average station criteria, $score_{asta}$, is the sum of the station wise scores, $score_{asta_{ista}}$, multiplied by a scale factor. The scale factor is the fraction of the number of observations with this station, $n_{obs_{ista}}$, in the tested scan by the maximum number of theoretically possible observations per station, which is basically the number of stations in this session minus one:

$$score_{asta} = \sum_{i_{sta}=0}^{n_{sta}} \left(score_{asta_{ista}} \cdot \frac{n_{obs_{ista}}}{n_{sta_{max}} - 1} \right). \quad (14)$$

This makes sure that scans that so far exhibit poorly scheduled stations would get an increased weight, and that weight gets higher the more observations with this stations are scheduled in this scan.

2.1.6. Average Sources

The average sources criterion also uses Equation (13) to calculate the score for this source $score_{asrc}$ but with the number of observations per source instead of the number of observations per station. Equation (14) changes to

$$score_{asrc} = score_{asrc} \cdot \frac{n_{obs}}{n_{obs_{max}}} \quad (15)$$

where $n_{obs_{scan}}$ is the number of observations in this scan and $n_{obs_{max}}$ is the theoretical maximum number of observations assuming a scan with all stations, see Equation (9).

It is to note here that this optimization criterion will only work if a small source list is used and all of those sources should be observed during the schedule. Typically a geodetic source list contains between 300 to 700 sources to choose from, but only ≈ 80 of them are scheduled. This leads to a lot of sources with zero observations and the Equation (13) might not work as intended due to the low average number of observations.

2.1.7. Average Baselines

Finally, the average baselines criterion also uses Equation (13) but with the number of observations per baseline instead of the number of observations per station. The idea is to increase the weight of certain baselines in order to create scans with different subnetworks. Equation (14) changes to:

$$score_{abl} = \sum_{i_{bl}=0}^{n_{bl}} score_{abl_{i_{bl}}} \quad (16)$$

2.1.8. Low Elevation

Tropospheric delays are one of the biggest error sources in geodetic VLBI; see Petrachenko et al. (2012) and Boehm et al. (2006). To estimate tropospheric delays better, it is advantageous to have observations at low elevations (Schuh & Böhm 2013). Therefore, the low elevation optimization criterion can be used to prefer scans with stations observing at low elevations. The score $score_{el}$ is calculated per station $score_{el_{ista}}$ and is based on the elevation of the observation as well as on two parameters, el_{begin} and el_{full} :

$$score_{el_{ista}} = \begin{cases} 0 & el > el_{begin} \\ \frac{el_{begin} - el_{ista}}{el_{full} - el_{begin}} & el_{begin} > el_{ista} > el_{full} \\ 1 & el < el_{full} \end{cases}. \quad (17)$$

If the elevation is above the el_{begin} threshold, no additional score is gained. If the elevation is lower than the el_{full} threshold, a score of 1 is gained. Linear interpolation is used in between. An approach with two thresholds and a linear interpolation in between instead of a step function is used to smooth the effect

at the borders. If a step function is desired, it is possible to give the parameters el_{begin} and el_{full} the same value.

The total score $score_{el}$ is calculated via the sum of the individual scores per station for all participating stations normalized by the total number of station in this session $n_{sta_{max}}$. Due to the normalization the maximum achievable score is 1:

$$score_{el} = \sum_{i_{sta}=0}^{n_{sta}} score_{el_{i_{sta}}} \cdot \frac{1}{n_{sta_{max}}}. \quad (18)$$

2.1.9. Low Declination

The low declination optimization criterion, $score_{dec}$, is used to increase the score of scans to sources at lower declination. Similar to the low elevation criterion the low declination criterion is defined by two parameters, dec_{begin} and dec_{full} and by the declination of the source dec :

$$score_{dec_{src}} = \begin{cases} 0 & dec > dec_{begin} \\ \frac{dec_{begin} - dec}{dec_{full} - dec_{begin}} & dec_{begin} > dec_{src} > dec_{full} \\ 1 & dec < dec_{full} \end{cases} \quad (19)$$

The total score is the source based score multiplied by a scale factor, where n_{obs} is the number of observations in this scan, and $n_{obs_{max}}$ is the maximum possible number of observations assuming a scan with all stations; see Equation (9). Due to the scale factor, scans with a higher number of observations at low declination get an increased score.

$$score_{dec} = score_{dec_{src}} \cdot \frac{n_{obs}}{n_{obs_{max}}} \quad (20)$$

Sources at a lower declination will get an increased score and therefore more observations. This helps negate the fact that sources at the southern hemisphere have considerably fewer observations than sources at the northern hemisphere (Plank et al. 2017a).

2.1.10. Other

Besides optimization criteria, other factors can have an influence on the calculated score. All stations, sources, and baselines can get a different weight in the parameters, $weight_{sta}$, $weight_{src}$, and $weight_{bl}$, respectively. When using individual weight, Equation (1) expands to Equation (21):

$$score_{scan} = \sum_{i_{opt}=0}^{n_{opt}} (weight_{i_{opt}} \cdot score_{i_{opt}}) \cdot \prod_{i_{sta}=0}^{n_{sta}} (weight_{i_{sta}}) \cdot weight_{src} \cdot \prod_{i_{bl}=0}^{n_{bl}} (weight_{i_{bl}}). \quad (21)$$

By default, all weights are set to 1. It is noted here that the score will only be multiplied by $weight_{bl}$ if there is an actual observation at this baseline. As described in Section 2.1.3, VieSched++ does not necessarily schedule all possible observations between all participating stations.

Besides this, there is another optional multiplicative factor of the score. It is possible to use a custom sequence of scans to different source groups as used by the STAR scheduling mode (McCallum et al. 2017). When using this option, scores from scans to sources to the current target group are multiplied by a factor of 100 otherwise they are divided by a factor of 100.

2.2. Recursive Scan Selection

A scheduler should aim at creating an observing plan that is as efficient as possible, i.e., antenna idle time should be avoided as much as possible. There are three major factors why idle time occurs in a schedule. First, antennas have different slew speeds. If a scan is finished, the antennas begin to slew to the direction of the next scheduled source. By definition and due to the limitations of the .skd file format, the observation starts as soon as all antennas finished slewing. Therefore, fast slewing antennas have to wait for slower slewing antennas with the start of the observation. Another reason for idle time is the different sensitivities of antennas. In geodetic VLBI, the recording time is typically estimated based on the participating stations, the recording rate, the observed source flux density, and the S/N that should be achieved; see Gipson (2016) or Petrov et al. (2009). The recording differs for each baseline involved in a scan. Less sensitive antennas have to observe longer while more sensitive antennas finish earlier that leads to idle time. Third and other than that, extra idle time can occur if a station is not participating in a scan and is simply waiting for the next scan.

To minimize the idle time, additional scans are inserted in case multiple antennas are waiting for too long. Typically, it is checked if multiple stations finish a scan way earlier than other stations. If this is the case, additional scans that only consider those stations are introduced. This is called a fillin-mode (Gipson 2010). While this works well in certain cases, it has some drawbacks. It only considers idle time due to different observing times, and it always adds those additional scans in a sequential order with time which means that fillin-mode scans are always added after a scan is selected.

VieSched++ uses a different approach to minimize idle time. It is using a recursive scan selection. If the software is selecting and scheduling a scan, it checks if it is possible to squeeze in another (fillin-mode) scan between the previous and the upcoming scan. Using this approach, the idle time can be reduced significantly, and idle times due to all causes like different slew times and observing times are considered. Figure 3 shows a flowchart of the recursive scan selection

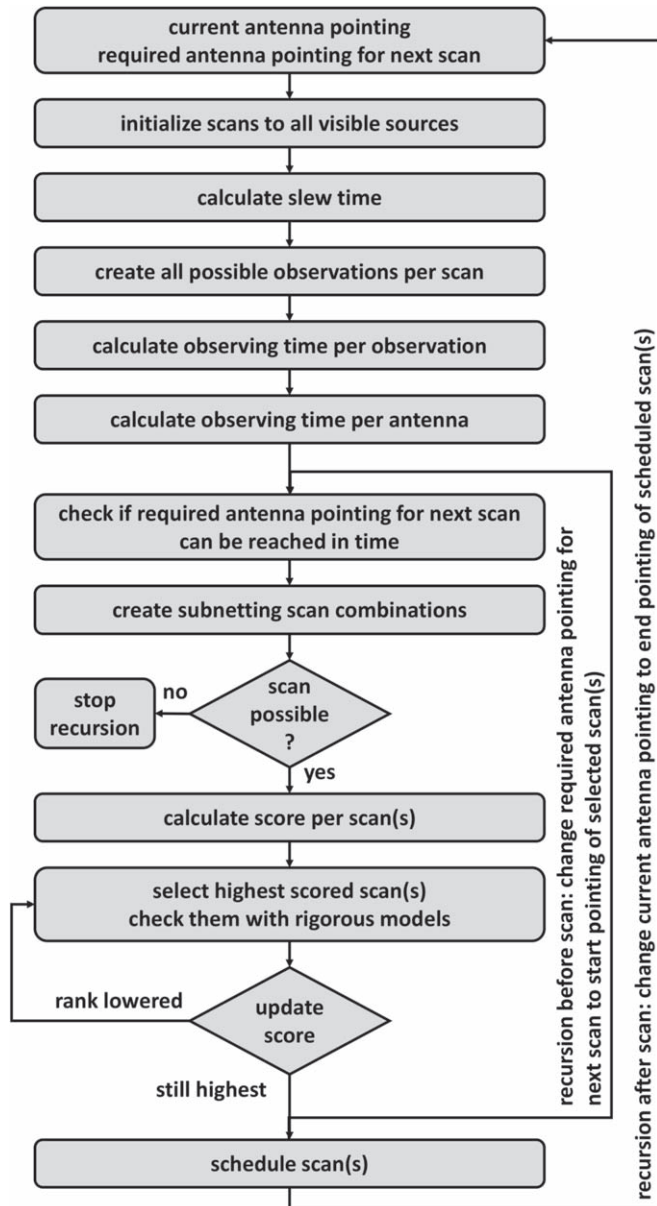


Figure 3. Recursive scan selection.

algorithm. An example of how the recursive scan selection works can be found in more details in Schartner et al. (2017).

The recursive scan algorithm opens up the chance to create schedules not necessarily in sequential time order. VieSched++ supports the creation of the schedule with up to four phases. The first phase is the so-called a priori scan selection. In the beginning, some a priori scans can be fixed at any point in time during the session. Typically, these are scans that are considered the most important scans for this session. In case the session is scheduled for a certain experiment, such as close observations to the Sun (Titov et al. 2018), the necessary scans

can be fixed a priori. In case of intensive sessions, scans that are close to the edges of the commonly visible sky can be fixed, because it is assumed that those scans have the highest impact on the result (Uunila et al. 2012). As discussed in Section 4 it is also planned to schedule satellite observations during this phase. After the a priori scan selection ends, the main phase starts where the scheduler fills the schedule starting with the beginning of the session; it is using the recursive scan selection to fill the gaps between scans. Afterward, a fillin-mode a posteriori can be used to further reduce the idle time. During this mode, the scheduler checks between two consecutive scans if it is possible to squeeze in another scan. While this may never happen during standard sessions where those times are already checked with the recursive scan selection in the main phase, it can lead to an improvement when scheduling stations in tagalong-mode (Schartner & Böhm 2019). The final phase covers the extension of the observing time in case of leftover idle time. If antennas are still idling but there is not enough time to squeeze in another scan, the software extends the observing time of the scans to increase the S/N. Altogether, this leads to a high number of observations with improved S/N, while reducing the idle time to a minimum. All of those phases besides the main phase are optional and can be turned off when running the software.

2.3. Schedule Optimization

Based on the parameters and the weighting of the optimization criteria used to create a schedule the quality of the results can vary greatly. In VieSched++, currently over 40 parameters can be used to adjust the schedule, as well as a combination of nine optimization criteria. Especially changing the weighting of the optimization criteria has a significant influence on the quality of the expected results. The weighting is controlled by the so-called weight factors. Unfortunately, it is necessary to adjust the weighting parameters based on the network, sources, and observing mode used during a session to achieve the highest quality.

VieSched++ comes with a feature called multi-scheduling that helps using a good set of parameters and weight factors (Schartner et al. 2017). Instead of creating only a single schedule, it is possible to create multiple schedules automatically; see Figure 4. To keep processing time, low multi-scheduling uses multi-core support. Each version of the schedule is created using different parameters and weight factors. It is then possible to compare all created versions based on metrics such as the total number of scans, the total number of observations, the number of observed sources, and many more. However, most of the time the user wants to create a schedule for a specific purpose; for example, to estimate EOPs or station coordinates. The results from the multi-scheduling feature can be used directly in the VieVS VLBI software to simulate artificial observations and estimate geodetic and

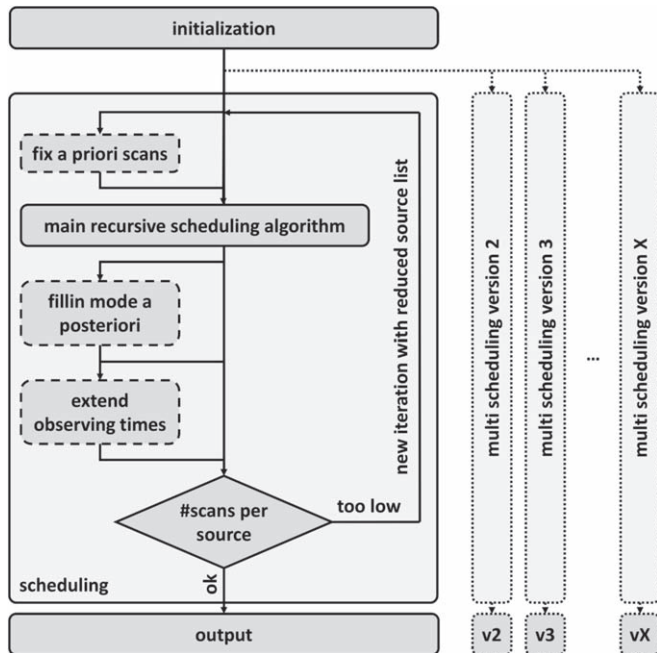


Figure 4. Flowchart of multi-scheduling algorithm and automatic source selection.

astrometric parameters and their accuracy. Furthermore, it is possible to use Monte Carlo simulations (Metropolis & Ulam 1949) in VieVS VLBI to get the repeatabilities of those parameters.

Another feature of VieSched++ is the automatic source selection. A typical geodetic VLBI source list contains several hundred sources. During a session, typically no more than one hundred of those sources are observed to have at least a couple of scans per source. This is a challenge for the scheduling software, because the schedule is created scan after scan and the software does not know in advance if a source will be observed multiple times. Several studies were made to manually adjust the source selection, especially for intensive sessions (Baver et al. 2012; Baver & Gipson 2013; Gipson & Baver 2016).

Sked has an algorithm that tries to preselect the best sources for the session and only creates a schedule using this sources. It creates a test schedule using a series of pseudo-scans spaced 10 minutes apart. The sources that are selected during this test schedule up to a certain threshold are considered the best sources for the schedule. More information about the algorithm can be found in Gipson (2016) under the *bestsource* command. However, there are two downsides with this approach. It is necessary to manually select the number of sources to be used and the source selection relies on a series of test scans. Additionally, the source selection happens before the real schedule is created, thus it is not adjusted if required for the real scheduling. Therefore, it is not guaranteed that the best sources

are really scheduled and that all sources get enough observations.

VieSched++ solves this issue by using an iterative source selection algorithm that runs a posteriori; see Figure 4. It is possible to set a minimum number of required scans per source, similar to sked’s “cull” command. VieSched++ will create a complete schedule and at the end check if the minimum number of required scans per source is reached in all cases. If this is not the case, those sources are rejected and VieSched++ creates another schedule using the reduced list of sources. These steps are repeated until the minimum number of required scans are reached for all sources. This makes sure that all sources are scheduled often enough and that the number of selected sources fits to the given session. There are several options available to fine-tune the automatic source selection and limit the number of iterations. The biggest difference is usually between the first iteration, which is using the full list, and the second iteration. The sked catalog source list of geodetic good source contains roughly 300 sources. Using a typical R1 network about 110 of those sources would be observed, but only around 70 of the sources would have the required three or more scans. This would mean that 230 sources would be removed between the first and the second iteration. To limit the number of rejected sources an option called gentle source reduction can be used for the first couple of iterations. If this option is selected, only every second source gets rejected from the schedule, which leads to a better schedule due to more freedom for the scheduling software at the first iterations.

2.4. Station-, Source-, and Baseline-based Parameters

VieSched++ is designed to be as flexible as possible. It is possible to select custom parameters for each station, source, and baseline. The idea is to be able to group similar objects and optimize the scheduling parameters. One use case would be to schedule calibration sources different than astrometric sources as necessary for scheduling astrometric sessions (Petrov et al. 2009). While the goal of calibration sources is to reach a high S/N, it might be necessary to optimize astrometric sources with respect to the UV plane for imaging. Another use case is the distinction between VGOS and legacy antennas. VGOS antennas are small and fast slewing antennas while most of the legacy antennas are bigger and slower. It is obvious that different parameters are required for both groups of stations (Schartner & Böhm 2019). An example can be to limit the slew distance for legacy antennas to increase the number of observations while a high slew distance might be ok for VGOS antennas.

These parameters are dependent on the network, the source list, and other factors, which makes it difficult to provide certain rules on how to use which parameters. The multi-scheduling feature can help to select reasonably good parameters; see Section 2.3.

Besides choosing customized parameters, it is also possible to change parameters over time. Using this feature, it is possible to divide the schedule into different phases as it is done in case of relativistic experiments (Titov et al. 2018). Every parameter can change at certain time points. This feature also allows for the introduction of antenna down times. In case of an antenna downtime, the user can either change the parameter “available” to “no”, or “tagalong” to “yes”. While with the former parameter there are simply no scans scheduled with this antenna, the latter option allows to schedule tagalong scans. More information about the tagalong-mode can be found in Gipson (2016). Those tagalong scans have no influence on the performance of the normal scans but the benefit is that the station can start observing again as soon as it is ready again. This approach was, for example, used during the CONT14 campaign (Behrend 2015).

2.5. Astrometric Optimization

VieSched++ supports the generation of astrometric schedules. Astrometric schedules typically contain two source lists: one for calibrator sources and one for astrometric sources (Petrov et al. 2009). This can be achieved in VieSched++ by creating two groups, one for calibrators and one for astrometric sources and adjusting the parameters for each group; see Section 2.4.

Astrometric sources can be scheduled to optimize the UV-coverage of these sources. Therefore, a target number of scans `target_scans` need to be set. VieSched++ calculates for the whole session duration how many minutes the source is visible `min_visible`. It is possible to set the minimum number of time between two scans to the same source `min_between` individually for each source based on those parameters:

$$\text{min_between}_{src} = \frac{\text{min_visible}_{src}}{\text{target_scans}_{src}}. \quad (22)$$

This leads to observations of astrometric sources at different hour angles and therefore optimizes the UV-coverage. It is possible to increase the weight of sources as soon as they are observed to make it more likely that these sources are observed again. Together with the previously discussed option to individually adjust the time between scans to the same source, this leads to a schedule optimized for astrometry and imaging.

This is not only a nice option for astrometry. It is also possible to select a small subset of sources during a geodetic schedule and schedule these sources in a way that they are optimized for imaging. This might be useful for VGOS schedules where it is necessary to image the sources for highest precision.

Additionally, for astrometry, it is possible to introduce calibrator blocks at certain time intervals. The idea of the calibrator block is to observe 3–5 tropospheric calibrators. Observations to tropospheric calibrators should occur at

different elevations. The purpose of including calibrators is to estimate the tropospheric delay during analysis and to link the positions of rarely observed astrometric sources to with those of frequently observed calibrators (Petrov et al. 2009).

3. Initial Tests

VieSched++ was used to generate observing plans for the International VLBI Service for Geodesy and Astrometry (IVS). The following IVS Schedules were created: AUA37, AUA40, AUA41, AUA44, AUA46, AUA47, as well as AUM01 to AUM10 and T2129. Additionally, nine intensive schedules between stations in Wettzell (Germany) and Santa Maria (Azores) were scheduled.

Comparing the results from session T2129 with previous schedules of the T2 campaign reveals significant improvements; see Table 1.

T2129 is the first T2 session that is scheduled with VieSched++. The goal of this session is to estimate station coordinates for the terrestrial reference frame. Creating a T2 schedule is a challenging task, because many stations that usually do not observe often for geodesy are participating. Some of these stations are located in very remote places like O’Higgins in Antarctica, while the majority of the stations are in Europe. It is a serious challenge to include O’Higgins in the overall network, especially because the sensitivity of O’Higgins is very low. Additionally, the network contains many different types of stations. Table 2 lists a selection of participating stations and some parameters.

However, besides all those challenges, VieSched++ manages to create a schedule with twice as many observations as the previous versions created with `sked`, although T2129 had the lowest number of participating stations. The average idle time was decreased from 30% per station to 8%, while the observing time increases from 48% to 67%. The bad performance of the previous T2 schedules created with `sked` is most likely due to a not optimally used scheduling setup for this experiments. To avoid this situation, the multi-scheduling feature is very helpful, because it does not rely on one single scheduling setup, but tries many different ones. It is also noteworthy that VieSched++ schedules the least number of scans while gaining the highest number of observations. This means that a high number of stations are participating in every scan which indicates a good integration of the network.

The T2129 schedule was created using the multi-scheduling tool; see Section 2.3. A total of 256 schedules were prepared using different optimization criteria and weight factors. All schedules were simulated 500 times using VieVS to estimate station coordinate repeatabilities. Based on these simulations, the most promising schedule was selected.

Besides these actually observed schedules, a large variety of arbitrary schedules were created and large-scale Monte Carlo simulations were performed to improve the algorithms used in

Table 1

Comparison with Previous T2 Schedules. Although T2129 has the Lowest Number of Stations, it has Twice the Number of Observations, and the Idle Time is Reduced Significantly

Session	Software	#Sta	#Scans	#Obs	$\frac{scans}{h}$	%Obs	%Idle
T2123	sked	19	617	6773	6	51	33
T2124	sked	17	733	7175	8	45	35
T2125	sked	17	1064	5528	8	54	27
T2126	sked	17	1075	6081	9	50	28
T2127	sked	17	627	6304	6	45	36
T2128	sked	18	803	5983	7	45	33
T2129	VieSched++	15	526	12713	10	67	8

Table 2

Overview about the Difference of Participating Stations. Although the Stations have Very different Parameters, VieSched++ Manages to Include Every Station Properly Into the Network, and Creates a Schedule with a High Number of Observations and Low Idle Time

Station	Diameter [m]	Alew Az [deg/min]	El [deg/min]	SEFD-X [Jy]	SEFD-S [Jy]
ISHIOKA	13.2	720	360	1950	1750
DSS65A	34.0	48	48	220	195
OHIGGINS	9	90	90	10000	18000
...					

VieSched++. The schedules were checked with the Matlab VieVS scheduling software and with sked to validate the results.

4. Plans for VieSched++ and Outlook

In recent years, there were some successful observations of satellites using VLBI (see Hellerschmied et al. 2016; Plank et al. 2017b; Sun et al. 2018). Those observations were scheduled using the Matlab VieVS scheduling software (Hellerschmied et al. 2017). It is planned to include the possibility for satellite observations into VieSched++ as well. The plan is to schedule satellite observations during the a priori scan selection; see Figure 4. Due to the high velocity of satellites, it is more difficult to select the proper cable wrap at the start. Scheduling satellite observations as a priori scans would remove this difficulty because the cable wrap can be chosen freely. Additionally, due to the velocity of satellites and the lower altitude the observing windows can become rather short. If the satellite scan is scheduled in a scan after scan algorithm and the observing windows is short, problems may arise because at a certain time, the scheduler has to switch from observing quasars to observing the satellite. If this time is too late, a scan can already last into the satellite observing window and therefore affect the possible observing duration. The same can happen if a scan stops early enough, but the antenna pointing is far from the direction to the satellite, and therefore the slew time lasts into the possible satellite observing duration.

On the other hand, if the time is too early, additional idle time is introduced in the schedule. By scheduling satellite observations as a priori scans the observing start can be set optimal because there are no restrictions like slew times or previous antenna pointings that have to be considered. It is the task of the recursive scheduling algorithm to take care of choosing the right sources before a satellite scan in order to get valid slew times and cable wraps. Since the recursive scheduling algorithm already knows when a satellite scan occurs and which cable wrap and antenna pointing are required it is way easier to optimize the schedule.

Another focus of future work is to automate the creation of full VLBI scheduling campaigns. The idea is to carry over the information from previous schedules like the number of scans and observations. This would simplify the creation of schedules for dedicated observing programs, like the SOAP program, where a certain list of sources have to be observed a certain amount of time.

Additionally, it is planned to add more optimization criteria. One idea is to calculate a score based on the volume spanned by the stations participating in a scan. This would help estimating EOP as well as other geodetic parameters (Nothnagel et al. 2017). For VGOS observations, it is assumed that a schedule based on optimizing the global sky coverage is preferable; see Sun et al. (2014) and Petrachenko et al. (2009). Therefore, an optimization criterion based on an uniform coverage of the celestial sphere over short time intervals will be implemented. Last, an optimization criterion based on impact factors might help improving the overall quality of the schedule, especially for intensives like Leek et al. (2015) and Nothnagel et al. (2015).

5. Conclusion


VieSched++ is a new VLBI scheduling software developed at Technische Universität Wien (TU Wien), and is part of the Vienna VLBI and Satellite Software (VieVS). It can create high-quality schedules for geodetic and astrometric VLBI. The software is freely available at <https://github.com/TUW-VieVS>. It is an improved version of the Matlab-based VLBI scheduler that was previously developed at TU Wien. While

some basic ideas are inspired by existing software solutions like sked, all algorithms were redesigned from scratch and great care was taken to provide sophisticated optimization criteria. A graphical user interface is available to simplify using the software and for analyzing and comparing schedules. Due to the parameterization and the recursive scan selection algorithm, the software is very flexible and can create high-quality schedules for different networks, tasks, and scientific goals. It supports the automatic creation of multiple schedules using different parameters that can be used directly in VieVS for simulations and analysis. This allows to create and select schedules based on simulations and reliable geodetic and astrometric results instead of scheduling based information like the number of observations and the total observing time. First results show a significant improvement for the quality of the schedules.

The authors acknowledge die Austrian Science Fund (FWF) for supporting this work within project VGOS Squared (P31625).

ORCID iDs

Matthias Schartner  <https://orcid.org/0000-0001-5855-7280>

Johannes Böhm  <https://orcid.org/0000-0002-1208-5473>

References

- Baver, K., & Gipson, J. 2013, in Proc. of the 21st Meeting of the European VLBI Group for Geodesy and Astronomy, ed. N. Zubko & M. Poutanen (Espoo, Finland: Reports of the Finnish Geodetic Institute), 205
- Baver, K., Gipson, J., Carter, M. S., & Kingham, K. 2012, in Seventh General Meeting (GM2012) of the International VLBI Service for Geodesy and Astrometry (IVS), ed. D. Behrend & K. D. Baver (NASA), 251
- Behrend, D. 2015, in Proc. of the 22nd European VLBI Group for Geodesy and Astrometry Working Meeting, ed. R. Haas & F. Colomer (Ponta Delgada, Azores), 145
- Board, I. S. 2019, IAU SOFA Software Collection, <http://www.iausofa.org>
- Boehm, J., Werl, B., & Schuh, H. 2006, *JGRB*, **111**, B02406
- Böhm, J., Böhm, S., Boisits, J., et al. 2018, *PASP*, **130**, 044503
- Bolotin, S., Baver, K., Gipson, J., Gordon, D., & MacMillan, D. 2014, in International VLBI Service for Geodesy and Astrometry 2014 General Meeting Proceedings: VGOS: The New VLBI Network, ed. D. Behrend, K. D. Baver, & K. L. Armstrong (Beijing: Science Press), 253
- Bourda, G., Charlot, P., & Biancale, R. 2007, in SF2A-2007: Proc. of the Annual meeting of the French Society of Astronomy and Astrophysics, ed. J. Bouvier, A. Chalabaev, & C. Charbonnel (Grenoble, France), 82
- Fey, A. L., Gordon, D., Jacobs, C. S., et al. 2015, *AJ*, **150**, 58
- Gipson, J. 2010, in Sixth International VLBI Service for Geodesy and Astronomy. Proceedings from the 2010 General Meeting, VLBI2010: From Vision to Reality, ed. D. Behrend & K. D. Baver (Tasmania, Australia), 77
- Gipson, J. 2016, Sked VLBI Scheduling Software (NVI, Inc. NASA/Goddard Space Flight Center)
- Gipson, J., & Baver, K. 2016, *JGeod*, **90**, 287
- Hellerschmied, A., Böhm, J., Kwak, Y., McCallum, J., & Plank, L. 2016, in EGU General Assembly 2016 (Vienna, Austria), EPSC2016-8895
- Hellerschmied, A., Böhm, J., Neidhardt, A., et al. 2017, in REFAG 2014, ed. T. van Dam (Cham: Springer International Publishing), 59, (doi:10.1007/1345_2015_183)
- Hjelle, G. A., Dähn, M., Fausk, I., Kirkvik, A.-S., & Mysen, E. 2017, in EGU 19th EGU General Assembly, EGU2017 (Vienna, Austria), 8486
- Hobiger, T., Otsubo, T., Sekido, M., et al. 2010, *EP&S*, **62**, 933
- Leek, J., Artz, T., & Nothnagel, A. 2015, *J Geod*, **89**, 911
- Ling, B. S. 2011, The Boost C++ Libraries (XML Press)
- Lovell, J., Plank, L., McCallum, J., Shabala, S., & Mayer, D. 2016, in International VLBI Service for Geodesy and Astrometry 2016 General Meeting Proc., New Horizons with VGOS, ed. D. Behrend, K. D. Baver, & K. L. Armstrong (NASA/CP-2016-219016), 92
- Mayer, D., Böhm, J., Lovell, J., et al. 2015, *VGI*, **2+3**, 162
- McCallum, L., Mayer, D., Le Bail, K., et al. 2017, *PASA*, **34**, e063
- Metropolis, N., & Ulam, S. 1949, *J. Am. Stat. Assoc.*, **44**, 335
- Nothnagel, A., Artz, T., Behrend, D., & Malkin, Z. 2017, *JGeod*, **91**, 711
- Nothnagel, A., Leek, J., Beier, M., Artz, T., & Ullrich, D. 2015, in Proc. of the 22nd European VLBI Group for Geodesy and Astrometry Working Meeting, ed. R. Haas & F. Colomer (Ponta Delgada, Azores), 185
- Petrachenko, B., Niell, A., Behrend, D., et al. 2009, in NASA/TM-2009-214180, 62
- Petrachenko, W. T., Niell, A. E., Corey, B. E., et al. 2012, in Int. Association of Geodesy Symp., **136**, 999
- Petrov, L., Gordon, D., Gipson, J., et al. 2009, *JGeod*, **83**, 859
- Plank, L., Hellerschmied, A., McCallum, J., Böhm, J., & Lovell, J. 2017a, *JGeod*, **91**, 867
- Plank, L., Lovell, J. E. J., McCallum, J. N., et al. 2017b, *JGeod*, **91**, 803
- Schartner, M., & Böhm, J. 2019, in Int. VLBI Service for Geodesy and Astrometry, 2018 General Meeting Proc., in press
- Schartner, M., Böhm, J., Mayer, D., McCallum, L., & Hellerschmied, A. 2017, in Proc. of the 23rd European VLBI Group for Geodesy and Astrometry Working Meeting, 113
- Schuh, H., & Böhm, J. 2013, Very Long Baseline Interferometry for Geodesy and Astrometry (Berlin, Heidelberg: Springer), 339
- Sovers, O. J., Fanselow, J. L., & Jacobs, C. S. 1998, *RvMP*, **70**, 1393
- Sun, L. 2013, Geowissenschaftliche Mitteilungen, 92, <https://resolver.obvsg.at/urn:nbn:at:at-ubtuw:3-92>
- Sun, J., Böhm, J., Nilsson, T., et al. 2014, *JGeod*, **88**, 449
- Sun, J., Tang, G., Shu, F., et al. 2018, *AdSpR*, **61**, 823
- Titov, O., Girdiuk, A., Lambert, S. B., et al. 2018, *A&A*, **618**, A8
- Titov, O., Tesmer, V., & Boehm, J. 2004, in International VLBI Service for Geodesy and Astrometry 2004 General Meeting Proceedings, ed. N. R. Vandenberg & K. D. Baver, 267
- Uunila, M., Nothnagel, A., & Leek, J. 2012, in Seventh General Meeting (GM2012) of the international VLBI Service for Geodesy and Astrometry (IVS), ed. D. Behrend & K. D. Baver (National Aeronautics and Space Administration), 395
- Vandenberg, N. 1997, sked's Catalogs Program Reference Manual (NVI, Inc. NASA/Goddard Space Flight Center)
- Vandenberg, N. 1999, sked: Interactive/Automatic Scheduling Program (Greenbelt, MD: Goddard Space Flight Center)
- Walker, R. 2018, The SCHED User Manual, Version 11.4, <http://www.aoc.nrao.edu/software/sched/index.html>