Satellite Communication Scheduling, Optimization, and Deconfliction using Artificial Intelligence Techniques

Dick Stottler[1]
Stottler Henke Associates, Inc., San Mateo, CA., 94404

This paper describes algorithms for automatic scheduling and deconfliction of satellite communication requests on highly constrained antenna resources. In addition to initially honoring the constraints included in the communication requests, the system also resolves remaining conflicts by suggesting "bending the rules" to resolve the conflicts based on past precedents retrieved from a case base. A prototype implementation is described along with the results from executing it on realistic data sets. Future work is also presented.

I. Problem Description

Scheduling and optimizing satellite communication resources is an enormously complex task. There are many constraints to be considered. The satellite must have LOS and be in-range of the antenna that it communicates with during the entire window for which communication is being requested. Different stations have antennas with different capabilities and have different support equipment so different satellites will have different sets of ground stations they can communicate with. Most satellites have requirements to have a certain number of communication events per day with maximum and minimum allowed time separations between events. Especially for Low Earth Orbit (LEO) satellites, often the required communication event time period is the entire, or a very large fraction of, each calculated view period window. Additionally some amount of time is required between communication events to change the configuration, including repositioning the antenna. Communications resources are very expensive and have become very tight, making optimal use of those resources extremely important. There is a very strong goal of meeting ALL communications requests, regardless of priority.

To accelerate satellite communication scheduling and deconfliction and to handle increased complexity of new resources, there is a need for a fully automated satellite communication resource planner which would receive communication requests and generate an optimized/deconflicted schedule that assigned resources for specific time windows to best meet those requests, considering the applicable constraints. The scheduling, optimization, and deconfliction process could be fully automated or performed interactively with human involvement, as desired by the user.

Currently, a first pass schedule is generated automatically, which leaves conflicts to be resolved by the human scheduler. There are two main categories of conflicts - those that can be resolved while satisfying all the constraints in the requests, presumably by shuffling other resource assignments in allowable ways (including reducing communication events to their minimum allowed time periods from their preferred values) and those that cannot. Resolving the former is relatively straightforward but does involve the use of good heuristic algorithms to produce as good a result as possible. Resolving the latter requires actually deciding what requests should have their constraints relaxed somewhat. This is currently done by human schedulers based on past experience and precedent. An automated deconflictor must also follow past precedent by having a case base containing previous requests to the satellite communication network clients to relax constraints (called "suggestions"), along with whether the suggestion was accepted or denied. Then, when trying to resolve a current conflict, the system can first calculate which constraint relaxation suggestion would resolve the conflict and check the case base to determine if similar

[1] Job Title: President; Department Name: N/A; Street Address/Mail Stop: 951 Mariner's Island Blvd., STE. 360, San Mateo, CA. 94404; AIAA Member Grade: Regular New

suggestions have been made in the past and whether they were usually approved. If so, that suggestion can be automatically made by the system. E.G., if one solution to a conflict is to move a conflicting pass to another site with which it has visibility, even though the original request specified the pass had to be at the original site, then this should be suggested, if this satellite's program office has approved similar moves in the past. Similarly, if the solution to a conflict is to shorten the prepass time of the conflicting pass by 2 minutes and this type of request deviation was approved in the past, then it should be suggested now to resolve the current conflict.

II. Satellite Communication Scheduler Prototype Description

We developed a prototype satellite communication Scheduler/Deconflictor based on these concepts. The prototype is tasked with scheduling satellite communication requests with antennas. It takes the view periods, time windows and resource requirements for all communication requests and schedules them. After exhausting legal means to resolve conflicts, the prototype provides a capability for "bending the rules" according to past precedent, to generate a valid schedule when one is impossible within the given constraints.

A. Scheduling Overview

The steps in the prototype scheduling process are described below.

1. Initialization

1. The Global Scheduler loads communication request time windows, required antennas and view periods from files.

2. The Scheduler sets up the queue that will be used to run the scheduling loop. Communication requests are put onto the queue, along with a command to schedule each of them. Prepasses, though represented as distinct activities, are always tied to their requests. They are not added to the queue, but will be scheduled at the same time as their requests.

3. The Schedule Coordinator starts looping through the queue. It runs the scheduling process by retrieving each pass from the queue, in turn, and scheduling it. The loop continues until the queue is exhausted.

2. The Scheduling Loop

4. A communication request asks the Scheduler to schedule it.

5. The availability of each antenna that the communication event has designated for use is restricted according to the periods of visibility between the satellite and antenna.

6. The Scheduler searches the allowed time window for availability on the specified antennas. The scheduler checks that there is time on the antennas for the request's prepass, as well, though this doesn't need to be within the satellite's view period. If the Scheduler can't satisfy the communication event's requirements, it will attempt to resolve the conflict in the next step. If there is a valid slot, the pass and prepass are scheduled there.

7. If there was a conflict, the Scheduler will schedule the request such that it is assigned to an antenna that it requires at a time when it is visible and within its given time window, but may be double-booked with another request. The Scheduler then attempts to swap out the competing request and reschedule it and its prepass at a different allowed time or a different allowed antenna or site. During this reschedule, the competing request may itself swap out its other competitors, though it will not attempt to reschedule requests that have already been rescheduled, which could lead to an infinite loop. If this does not succeed, the two requests remain where they are, in a conflicted state.

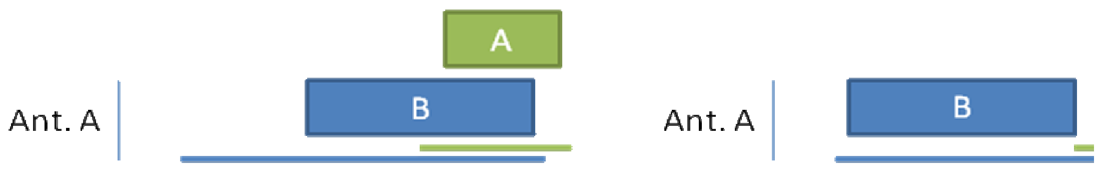


Figure 1: Swapping example (horizontal lines represent view periods) Left: Initially A and B cannot be scheduled because B is using the antenna for too much of A's view period. Right: Schedule A in its view period anyway and B rescheduled.

B. Bending the Rules

After scheduling, conflicts may exist where it was impossible to schedule all requests without some overallocation of antennas. To resolve these situations, the Scheduler can automatically resolve selected conflicts by "bending the rules" to loosen the constraints on a request. This can be done all at once or one at a time. In the latter case, the user initiates this process by selecting a conflict to be resolved. The conflict is sent to the Conflict Manager, which tries a number of solution methods, in order. These are:

1. Move from an antenna it can currently use to another at the same site
2. Move from a site it can currently use to another site
3. Shorten the duration of a prepass
4. Expand the valid time window of a request
5. Shorten the duration of a pass

When trying each solution, the Conflict Manager will consult the Case-Based Reasoning (CBR) module to confirm that a change that would resolve the conflict is likely to be approved. If the CBR module indicates that similar changes have been approved a high percentage of the time in the past, the Conflict Manager will commit the change and ask the user for approval.

C. Moving Antennae

The first solution method is to move a request from one antenna to another at the same site. To do this, the Conflict Manager first asks the CBR for cases from the past where requests were able to move from one of the antennae the satellite specifically requested to another at the same site. It then adds all the antennae that are returned to the antennae that the request could possibly use, and reschedules. By calling the rescheduling routine, some reshuffling, if appropriate to make room, can occur.

Moving from one site to another is similar; the Conflict Manager first asks the CBR module for sites that the request is likely able to move to from its current available sites. It then adds these sites to the antennae the request can use, and reschedules the request to one of these sites if it is visible there.

D. Shortening a Prepass

The next method attempted to resolve a conflict is to shorten the duration of prepasses involved in the conflict. When attempting to shorten a prepass, the Conflict Manager determines how much time is available before the prepass' associated request. It then asks the CBR module if similar cases were approved. If so, the prepass' duration is shortened and the request is rescheduled.

E. Expanding a Request's Time Window

To expand a request's time window, the Conflict Manager first asks the CBR module for cases from the past where the satellite involved had been approved for scheduling outside its time window. From these cases it expands the request's time window as far as it is likely to be approved, and then attempts rescheduling.

F. Shortening a Pass

Lastly, the Conflict Manager attempts shortening the passes involved in the conflict. When attempting to shorten a pass, the Conflict Manager searches for available time windows of any length where the pass would be able to be scheduled if its duration were shortened. The CBR module is asked if shortening the pass to fit the largest window found is likely to be acceptable. If so, the pass' duration is set accordingly and rescheduled.

III. Results and Future Work

The prototype was applied to representative sample scenarios. These were created from descriptions of ground station sites and various satellites. A realistic sample of about 400 requests was generated corresponding to a single 24 hour period and submitted to the prototype. The prototype first executed steps 1 though 7, described above, to generate a preliminary schedule with conflicts in less than 30 seconds on a standard Windows PC. A portion of such a schedule is shown below. The major resources, the antennas, are shown vertically along the left, time flows to the right horizontally and passes and prepasses are shown as rectangles. (Conflicts are shown by 1 activity involved in the conflict being outlined in red and the other appearing above its designated antenna in an extra blue row.)

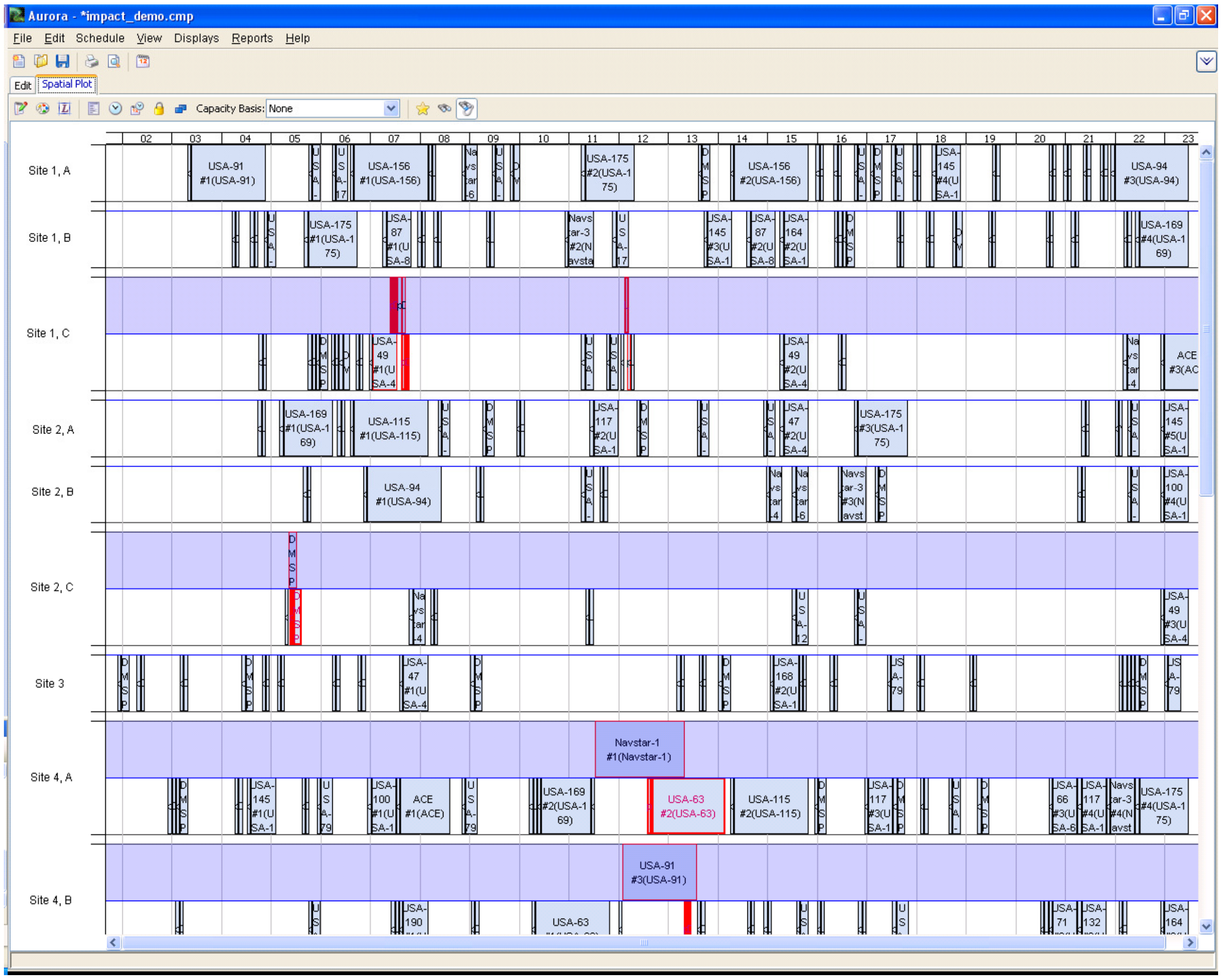


Figure 2: Screenshot of schedule results with conflicts shown in red.

The prototype was able to successfully schedule and deconflict the very large majority of requests. A typical scenario involving 400 requests would usually produce a preliminary schedule with 10 to 30 conflicts. Usually the automatic deconfliction procedure would resolve all of these. Occasionally 1 or 2 couldn't be resolved. This would decrease the scheduling/deconfliction workload of a human scheduler by one or two orders of magnitude.

Some example results from running the prototype are shown below. In Figure 3, a conflict is caused by too many passes requesting Site 2 at approximately the same time. The user selects one of the pass requests involved in the conflict and requests a suggested resolution. The system checks the case base and finds precedents for moving that constellation's requests from Site 2 to Site 4, when there is visibility at both at the same time. Figure 4 shows this step being taken.

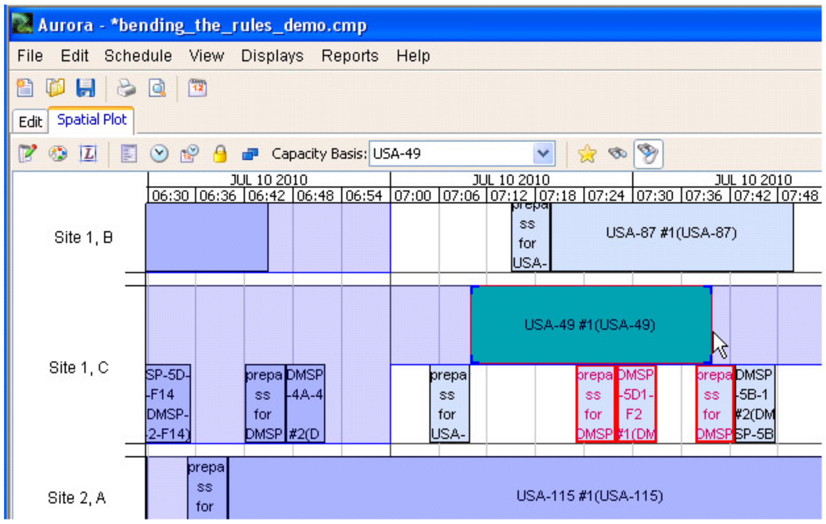
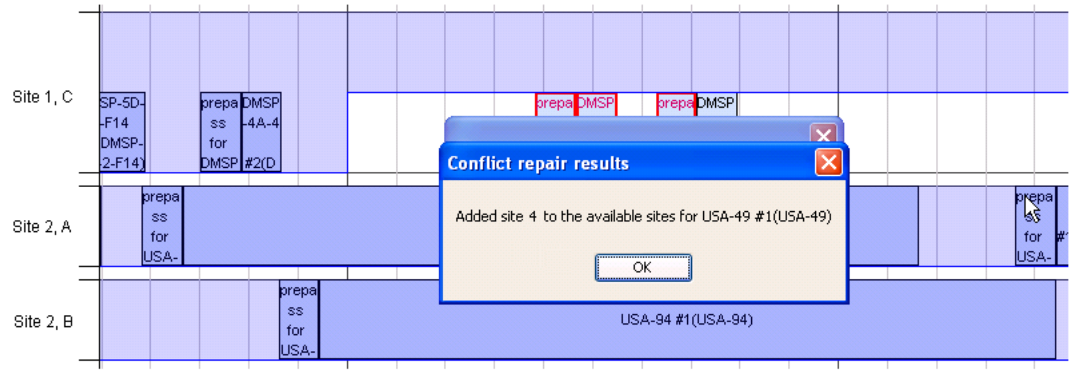


Figure 3: USA-49 Conflict with two DMSPs.



Added site 4 to the available sites for USA-49 #1(USA-49)

Figure 4: Conflict resolved by moving USA-49 to Site 4.

In another example, shown initially in Figure 5, the user requests resolution of a conflict caused by two passes requesting the same antenna at the same time. In Figure 6, resolution is not possible, because no applicable precedents are in the case base. For demonstration purposes, an applicable precedent case was added into the case base and resolution was again requested. This time, as shown in Figure 7, resolution was successful because of the new case, thus showing how the system could evolve over time.

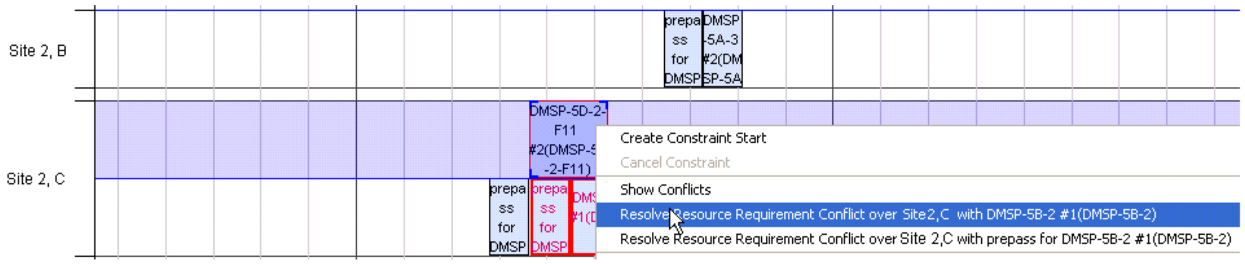
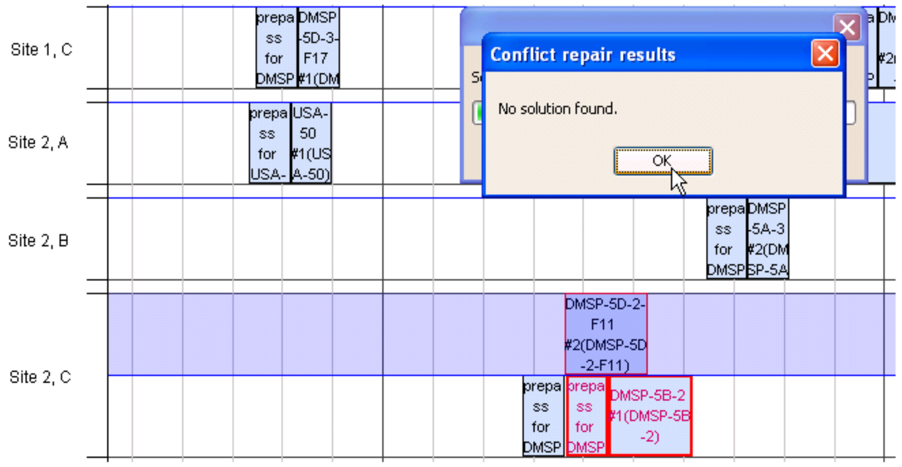

Site 2, B
Site 2, C
prepass for DMSP
DMSP-5A-3 #2(DMSP-5A
DMSP-5D-2-F11 #2(DMSP-5D-2-F11)
Create Constraint Start
Cancel Constraint
Show Conflicts
Resolve Resource Requirement Conflict over Site2,C with DMSP-5B-2 #1(DMSP-5B-2)
Resolve Resource Requirement Conflict over Site 2,C with prepass for DMSP-5B-2 #1(DMSP-5B-2)

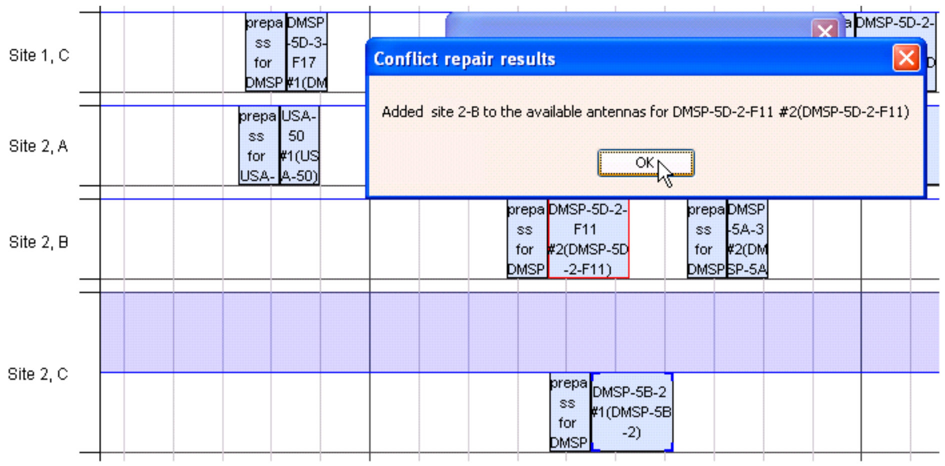
Figure 5: Attempt to Resolve Conflict.



Conflict repair results

No solution found.

Figure 6: No Solution Found with the Existing Case Base.



Conflict repair results

Added site 2-B to the available antennas for DMSP-5D-2-F11 #2(DMSP-5D-2-F11)

Figure 7: With a new case added to the Case Base, Conflict Resolved by switching Antennas.

It has been decided to continue this research and development and fund the next phase of the effort, development of a full-scale automatic scheduling and deconfliction system for testing and evaluation. The full-scale system has several additional issues to consider. One is seeding of the initial precedent case base which is aided by the fact that current satellite communication network schedule management processes offer a tremendous opportunity for developing a case-based conflict resolution suggestion capability. When schedulers, in order to deconflict communication passes, decide to suggest to satellite program offices that they deviate from the constraints listed with the satellite communication service request, they carefully annotate in a formal way the suggested changes, including identifying which other pass there is a conflict with, and the precise nature of the change and deviation. These suggestions are then either accepted with the initials of the individual at the program office or rejected, but in either case they are stored in the archived database. Discussions with the schedulers confirm that a year's worth of this data is far more than enough to populate the case base in a reasonable way and that at least a year's worth of data is readily available.

Another issue for the full-scale system is operating in concert with the current work flow. For example, the automatic satellite communications network scheduler/deconflictor may determine that moving a specific pass from one site to another will eliminate a particular conflict. By retrieving from the year's worth of past suggestions several suggestions made for the same constellation to move from the same first site to the same second site would be found. If a high percentage of the time, those suggestions were approved, the CBR system would return that fact and the automatic scheduler/deconflictor, depending on the mode it was in, would either suggest that to the user, or automatically make the same changes and notations that the human scheduler would have. To the actual human scheduler the effect would be identical to the current process of handing of the schedule from one human scheduler to another to work on for a little while and being handed it back, which is currently done quite often. Alternatively, even if there were no suggestions with the same second site, the fact that there were several suggestions in the case base for moving from the same first site would be almost as good a precedent. This how CBR systems work: they do not look for an exact match but simultaneously look for the best matches and make use of whatever is returned, even if it is not an exact copy.

Another issue relates to the fact that precedents are not exactly symmetric. Approval of the same or greater deviation was considered similar but denial of the same or lesser deviation was also considered similar. Consider, for example, a possible move to shorten a 10 minute prepass by 2 minutes. For this potential prepass shortening move, acceptances of past prepass shortening by 2 or more minutes would be considered similar and denials of prepass shortening of 2 or 1 minutes would be considered similar. So if there were 5 past cases where a 10 minute prepass for the same satellite constellation was shortened by 2 or 3 minutes, and only 2 cases where a 10 minute prepass shortening of 1 or 2 minutes was denied, then this suggestion would be made and, for example, a previous suggestion of a 4 minute shortening being denied would be ignored as would a previous suggestion of a 1 minute shortening being accepted. The original duration also factored into the similarity on a secondary level so that 10 minute prepasses were more similar than longer ones that were accepted and more similar than shorter ones being rejected (since longer beginning durations imply a smaller percent change and therefore a lesser deviation.)

IV. Conclusion

This effort showed that an automatic scheduling and deconfliction system for satellite communication network scheduling was feasible, practical, and able to run on realistically sized data sets in reasonable amount of time. Use of Case-Based Reasoning was important to mimic the human scheduler's use of past experience to resolve otherwise unsolvable conflicts. Full-scale development for testing and evaluation is commencing with additional considerations for usability, more refined case reasoning, and initial case seeding.