Resource Re-scheduling for a Network of Communications Antennas

S. Chien, R. Lam, and Q. Vu
Jet Propulsion Laboratory, California Institute of Technology
4800 Oak Grove Drive, M/S 525-3660, Pasadena, CA 91109-8099
{steve.chien, raymond.lam, quoc.vu}@jpl.nasa.gov

Abstract

This paper describes the Demand Access Network Scheduler (DANS) system for automatically scheduling and rescheduling resources for a network of communications antennas. DANS accepts a baseline schedule and supports rescheduling of antenna and subsystem resources to satisfy tracking goals in the event of: changing track requests, equipment outages, and inclement weather.

1. Introduction.

The Deep Space Network (DSN) [2] was established in 1958 and since then it has evolved into the largest and most sensitive scientific telecommunications and radio navigation network in the world. The purpose of the DSN is to support unpiloted interplanetary spacecraft missions and support radio and radar astronomy observations in the exploration of the solar system and the universe. There are three deep space communications complexes, located in Canberra, Australia, Madrid, Spain, and Goldstone, California. Each DSN complex operates four deep space stations -- one 70-meter antenna, two 34-meter antennas, and one 26-meter antenna. The functions of the DSN are to receive telemetry signals from spacecraft, transmit commands that control the spacecraft operating modes, generate the radio navigation data used to locate and guide the spacecraft to its destination, and acquire flight radio science, radio and radar astronomy, very long baseline interferometry, and geodynamics measurements.

From its inception the DSN has been driven by the need to create increasingly more sensitive telecommunications devices and better techniques for navigation. The operation of the DSN communications complexes require a high level of manual interaction with the devices in the communications link with the spacecraft. In more recent times NASA has added some new drivers to the development of the DSN: (1) reduce the cost of operating the DSN, (2) improve the operability, reliability, and maintainability of the DSN, and (3) prepare for a new era of space exploration with the New Millennium program: support small, intelligent spacecraft requiring very few mission operations personnel[4].

This work was performed by the Jet Propulsion Laboratory, California Institute of Technology, under contract with the National Aeronautics and Space Administration.

This paper appears in the working notes of the 1997 NASA Workshop on Planning and Scheduling for Space.

This paper describes the DANS system for rescheduling and resource allocation for antenna and subsystem resources in the DSN. DANS works from an initial schedule and uses prioritized pre-emption and localized search to find antenna and other equipment resources required to support changes to schedule requirements which may be caused by a wide range of circumstances including: changing track requirements from the flight projects, equipment outages, and inclement weather.

This paper is organized in the following manner. We begin by characterizing the current mode of operations of the DSN. Then we begin the main body of the paper which describes the DANS scheduling system. We begin by describing the problem representation within the DANS system. We then describe the priority-based pre-emption scheduling algorithm. Next we provide an example of the algorithm performing rescheduling. Finally, we describe the various reasons why rescheduling occurs.

2. Overview

Each week, a complex matching process between spacecraft project communication service requests and NASA Deep Space Network (DSN) resources occurs. In this process, project requests and priorities are matched up with available resources in order to meet communications needs for earth-orbiting and deep space spacecraft. This scheduling process involves considerations of thousands of possible tracks, tens of projects, tens of antenna resources and considerations of hundreds of subsystem configurations. Once this initial schedule is produced (8 or more weeks before implementations), it undergoes continual modification due to changing project needs, equipment availability, and weather considerations. Responding to changing context and minimizing disruption while rescheduling is a key issue.

The high level resource allocation problem for the smaller DSN antennas (26M and smaller) is currently handled by the OMP[6] scheduler. OMP accepts generalized service requests from spacecraft projects of the form "we need three 4-hour tracks per week" and resolves conflicts using a priority request scheme to attempt to maximize satisfaction of high priority projects. OMP deals with schedules involving thousands of possible tracks and a final schedule involving hundreds of tracks.

More recently, there has been a desire to automate the scheduling of the larger (34 and 70 m.) antennas. This paper describes the Demand-based Automated Network Scheduling System (DANS), an automated scheduling system being developed at the Jet Propulsion Laboratory (JPL) to schedule DSN 34 and 70 meter antenna resources. DANS is an evolution of the highly successful OMP system.

DANS uses priority-driven, best-first, constraint -based search and iterative optimization techniques to perform priority-based rescheduling in response to changing network demand. DANS first considers the antenna allocation process, as antennas are the central focus of resource contention. After establishing a range of antenna options, DANS then considers allocation of the 5-13 subsystems per track out of the tens of shared subsystems at each antenna complex (signal processing Center (SPC)) used by each track. DANS uses constraint-driven, branch and bound, best first search to efficiently consider the many possible subsystems schedules. Like the GPSS [1, 8] and SPIKE [5] systems which schedule space shuttle ground processing and science observations for the Hubble Telescope, DANS uses iterative repair and local search to repair (improve) schedules. However DANS also uses temporal constraint network and SIMPLEX techniques to efficiently reason about metric time constraints and to optimize time sensitive utility functions. Other work on scheduling ground stations by GSFC [10,11] has focused more on schedule editing rather than automate scheduling.

The DSN domain contains many resources. In the existing configuration (as of July 1996), it consists of 11 SPCs, 45 antennas, and 161 subsystems located at different sites around the world. The majority of the antennas can be classified as 26, 34, and 70 meter antennas. The 26 meter antennas on average handles 600 activities per week. The 34 and 70 meter antennas performs over 200 activities per week. Additionally, this workload is expected to increased dramatically in the next several years.

3. Domain Characteristics

In addition to the basic antenna resource allocation problem, the DSN scheduling problem is further complicated by three factors: (1) context-dependent priority; (2) subsystem allocation; and (3) the possibility of reducing the length of the tracks. DSN track priorities are context dependent in that they are often contingent on the amount of tracking the project has received so far in the week. For example, a project might have priority 3 to get 5 tracks, priority 4 to get 7 tracks and priority 6 to get 9 tracks (where lower priorities represent more important tracks). This reflects that 5 tracks are necessary to maintain spacecraft health and get critical science data to ground stations; 7 tracks will allow a nominal amount of science data to be downlinked; and 9 tracks will allow for downlinking of all science data (e.g., beyond this level additional tracks have little utility). An important point is that specific tracks cannot be simply labeled with these priorities (e.g., the project is allowed to submit 5 tracks at priority 3, 2 at priority 4 and so on). Rather when considering adding, deleting, or moving tracks the scheduler must consider the overall priority of the project in the current allocation context.

In addition to allocating antennas, DSN scheduling involves allocating antenna subsystems which are shared by each Signal Processing Center (such as telemetry processors, transmitters and exciters). Allocating these complicates the scheduling problem because it adds to the number of resources being scheduled and certain subsystems may only be required for parts of the track.

Finally, the DSN scheduling problem is complicated by the fact that the track duration can be relaxed. For example, a project may request a 3 hour track but specify a minimum track time of 2 hours. When evaluating potential resource conflicts the scheduler must consider the option of shortening tracks to remove resource conflicts. Currently OMP and DANS use a linear weighting scheme in conjunction with a modified SIMPLEX algorithm to trim tracks in accordance with prioritizations.

DANS accepts two types of inputs: 1. an 8-week prior-to-operation schedule from the Resource Allocation and Planning (RAP) team[1], and 2. activity requests from each individual flight project. The 8-week schedule is only a baseline for creating a conflict-free schedule. Many scheduled activities at that time are tentative at best, and subject to revision due to changing project status. Also, the schedule is for the antenna resources only; DSN subsystem scheduling is not considered at all in the 8-week schedule.

The activity requests are used by the flight projects to add and delete activities on an existing schedule due to changing project requirements and/or resource availability. The DANS objective is to satisfy as many activity requests as possible while maintaining a conflict-free status with minimum disruption to the existing schedule. DANS is intended for use by the operation personnel to maintain and update the DSN schedule throughout each schedule.

Another issue is the placement of activities onto the schedule. The possible times for a spacecraft track are limited by spacecraft orbit views, which are the periods in which the spacecraft is visible from a ground station. Also, the range from the antenna to spacecraft dictates the quantity and types of antenna(s) required for each activity. Sometimes, an array of multiple antennas instead of a single one is required to communicate with the spacecraft. In addition, the uplink and downlink activities can occur on different antennas, and can be several hours apart imposing additional dependencies between activities.

There are two types of activities in the DSN domain: spacecraft activities and ground activities. Spacecraft activities are submitted by projects and used to interact with spacecraft. They are required to satisfy the domain constraints above. Ground activities represent hardware maintenance. Antenna time which is not occupied by spacecraft activities is used for ground activities such as non-regular maintenance requirements and testing, with maintenance having higher priority.

Each DSN spacecraft activity is divided into 3 steps: pre-calibration (precal), tracking, and post-calibration (postcal). The time periods for each step are specified in ranges of

[1] The RAP team uses its own forecasting and scheduling systems [3,7] to establish and revise this baseline schedule.

values. The time periods are unique for each activity type, and depend on the antenna type and subsystem usage. DANS models this dependency by either shrinking or shifting activities to maximize resource utilization as dictated by activity type, antenna type, and subsystem usage.

DANS is required to schedule two different kinds of DSN resources: antennas and subsystems. Antennas and subsystems are unit resources and as such can not be shared by more than one activity. Subsystem resources are hardware such as transmitters which are required to work with an antenna during communication. Normally, there are many pieces of hardware that support each antenna and DANS is required to generate a schedule which allocates all necessary subsystems.

4. Resource Representation

The DSN consists of many Signal Processing Centers (SPC) situated around the globe. Each SPC may contain one or more antennas and many associated subsystems. These subsystems are used in conjunction with the antennas to perform tracks. DANS represents these resources in a hierarchical manner. Each SPC contains one or more antennas, which are the children of the SPC. There are also many SPC subsystem resources residing also as children of the SPC. For each antenna in the hierarchy tree, there are many DSS subsystems associated with it.

All the DSN resources are represented as capacity timelines which model a resource's usage at any instant of time for the duration of a schedule. A timeline is a sequence of capacity units each of which represents a constant resource usage within a time period. Shown in Figure 3 is the timeline representation of the DSS-26 antenna resource.

CapacityTimelines are constantly being modified and updated during the inference process to reflect the state of the resource at that instant of time. For performance reasons, a time slice caching scheme is used to expedite the query process. The scheme divides the timeline into a number of buckets. For timeline operations, the system finds the bucket that contains the moment first, and then changes the appropriate value. For example, when the system looks for a CptyUnit which contains the 1:30 moment as shown in Figure 3, it first identifies Bucket #2 as the container which includes the 1:30 moment. Then it traverses down the timeline starting from the beginning of Bucket #2 until it locates the CptyUnit that contains the 1:30 moment.

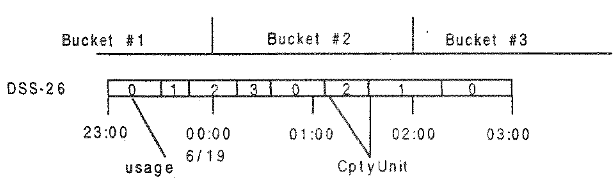

Figure 3: Capacity Timeline Resource Representation

5. Inference Engine

The current version of DANS employs a priority-based inference strategy to satisfy the project requests. The major bottleneck of the DSN domain is the antenna resources. This is due to the fact that there are limited numbers of antennas available while there are many identical subsystem resources that support unexpected events. In order to leverage this domain characteristic, DANS first focuses on antenna scheduling then resolves subsystem constraints.

The DANS scheduling process involves hypotheses generation, conflict identification, and conflict resolution. The process of scheduling a single activity consists of three major steps. First, the system generates an exhaustive list of solutions for each activity request at the antenna level. Second, this solutions list is then applied to the subsystem level to pick the best solution. Finally, the activity is placed onto the schedule. If the activity addition causes another activity deletion, the deleted activity(ies) will be rescheduled immediately. For ground subsystem maintenance activities which do not require antenna resources, the scheduling process will skip the first step(antenna scheduling) and start from the second step(subsystem scheduling). The antenna and subsystem inferencing flowcharts are shown in Figures 4 and 5.

At the antenna inferencing level, the system first selects an activity request (ACT) and extracts from the request the temporal window for the request (the time during which the project has requested a track). This window is then matched up with overlapping orbit views for the spacecraft (an orbit view is a time period during which a specific antenna can physically view the spacecraft), producing a list of valid intervals within the window. For each possible interval, the system tries to schedule the ACT. When a conflict, the system first tries to shift the ACT within the interval. If this action does not resolve the conflict and the conflicting activities have lower priority, the system identifies the conflicted activity(ies) for deletion.

While DANS is searching for possible ways to satisfy a request, it tracks the cost of each solution. The solution costs reflects the number of tracks which the new track displaces. Because scheduling further details of a track (more of the required subsystems) can only introduce more conflicts as a track becomes more completely scheduled its cost can stay the same or increase but never decrease.

DANS seeks to minimize the displaced tracks because the displaced tracks at best will be moved and at worst will not be accommodated in the final schedule. Either of these outcomes is bad, it is extremely disruptive to the projects adjust their operations in response to a moved track. Indeed, it is even worse if a track that a project was counting on is suddenly deleted. Thus, minimizing movements and deletions of tracks is key.

At the DSN subsystem inferencing level, DANS first identifies all the subsystems which are required to support the ACT. Then DANS selects the first available antenna solution and tries to schedule the ACT to each of the subsystems at the specified time slot.

If conflict exists, it will try to resolve it as described above. When a solution exists, the system calculates the completed solution cost for both the antenna(s) and subsystems, and compares them to the previous completed solution costs and next antenna solution. If the current solution cost is less than or equal to the other solution costs, the system will commit to this solution, and will schedule the ACT to this specified time slot. Otherwise, this solution will be saved for future reference.

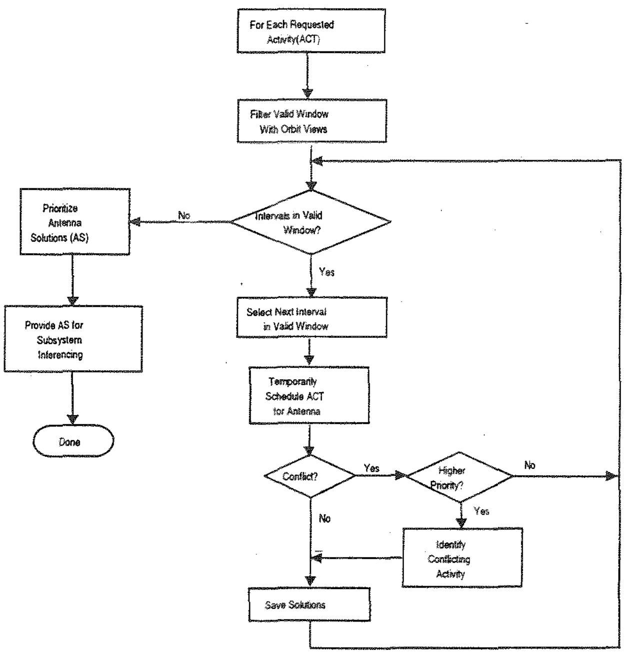

Figure 4: Priority-Based Antenna Inferencing Flowchart

After the system evaluates all the antenna solutions at the subsystem level, it will pick the best solution (i.e. the lowest cost solution to schedule the activity request). If this action requires deletion of other lower prioritized activities, these deleted activities will be submitted back to the schedule as a request immediately. DANS uses an equation to calculate each solution's cost. The equation is as follows:

$$\text{Solution Cost} = \frac{\text{NAD} * \text{activity priority}}{\text{NAD} - (\text{NAD} - 1) * 0.1}$$

where NAD = number of deletions required to schedule the current activity.

The cost is based on the activity priority. When there is no deletion required for scheduling the ACT, the cost is zero. When one deletion is required, the cost is equal to the activity priority. When there is more than one deletion required in order to schedule the ACT, the cost increases with the number of deletions.

6. Priority-Based Rescheduling: An Example

Consider the following example of the DANS scheduling algorithm. Initially, the DSS-14 antenna and its subsystems have committed their resources to two activities between 6:00am and 10:15am. Activity P0 has a valid window from 7:45am to 12:45pm, and occupies the 7:45am to 8:45am time slot. Activity P1 has a valid window from 9:15am to 12:45pm, and occupies the 9:15am to 10:15am time slot. Both P0 and P1 are DSN ground activities with priorities equal to 4. Activity P6 is a Galileo activity which requests a two hour duration between 5:45am and 10:15am on the

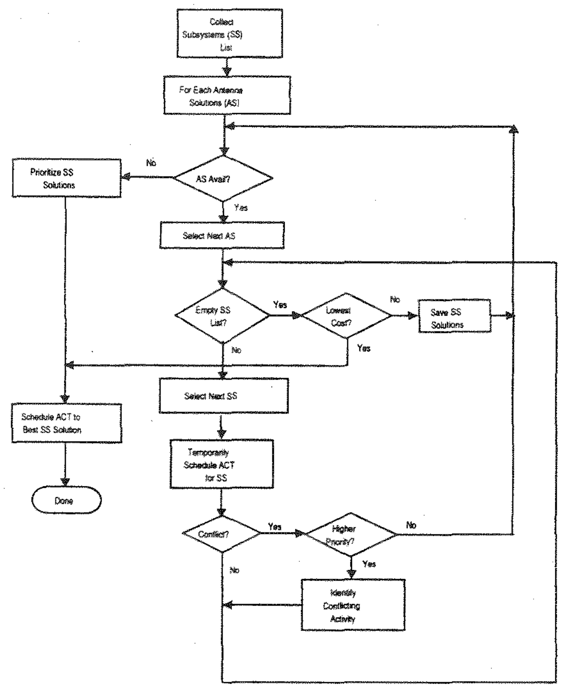

Figure 5: Priority-Based Subsystem Inferencing Flowchart

DSS-14 resources. P6 has a priority value of 3, which is higher than the priority of both P0 and P1. Subsequently, P6 can bump these two activities from the timelines when conflict arises. This information is shown in Table 1.

The DANS objective is to commit DSS-14 and subsystem resources to P6 activity and to maintain the conflict-free schedule with minimum disruption to the existing schedule. See Figure 6 for the scheduling sequences for this example.

For the Galileo P6 request, DANS first identifies all orbit views which are subsets of the P6 valid window. This enables DANS to filter out the invalid gaps and limits the search space. For this example, there is only one orbit view existing from 6:00am to 10:00am. Then the system turns its attention to the critical antenna resource to generate hypotheses. It traverses within the valid orbit view duration on the DSS-14 antenna timeline to identify time slots which can satisfy the 2 hour duration constraint. There are two valid time slots: Time Slot 1 from 6:00am to 8:45am; and

Time Slot 2 from 7:45am to 10:00am. DANS schedules P6 to both Time Slot 1 and Time Slot 2 to create two hypotheses. When P6 is place at Time Slot 1 for hypothesis 1 (HY1), it causes conflict with P0. Since P6 has higher priority than both P0 and P1, placement of P6 within this duration will delete activity P0 and the antenna solution cost becomes 4. When P6 is placed at Time Slot 2 for hypothesis 2 (HY2), it causes deletion of both P0 and P1, and the antenna solution cost is 4.21053. The system then sort all the hypotheses based on the antenna solution cost in ascending order. The result guides the inference process at the subsystem level without the necessity of performing an exhaustive search.

Activity	Project	Priority	Orbit View	Request Duration	Valid Window	Assignment
P0	DSN	4	N/A	60	7:45-12:45	7:45-8:45
P1	DSN	4	N/A	60	9:15-12:45	9:15-10:15
P6	GLLO	3	6:00-10:00	120	5:45-10:15	

Table 1: Example Activities Description

The system then continues the conflict identification at the subsystem level for both hypotheses. Activity P6 requires seven subsystem resources to accomplish the task. They are the LMC, SLE, TGC-A, MDA, NAR, RCV, and S-TWM. DANS identifies resource conflicts with P0 for the LMC, MDA, RCV subsystems for both hypotheses. The combined solution cost for the HY1 becomes 8.28571. This combined cost is then compared to the HY2 antenna cost, which is 4.21053. If the combined cost would have been less than the antenna cost value, the system would stop here and select the current hypothesis as the best solution. Since this is not the case, the system continues on the next hypothesis and calculates the combined cost for HY2 as 8.53485. Based on the result, DANS selects the first hypothesis as the solution since it has the lowest combined cost. It schedules P6 to the 6:00am to 8:00am duration and deletes P0 from the resource timelines. The system applies the same process to reschedule P0 activity. It identifies 3 time slots to generate hypotheses as shown in Figure 6. The system identifies the first time slot to schedule P0 between 8:00am and 9:15am with zero cost. It stops here having completed its task successfully to place the Galileo activity on the timeline without deleting any existing activities from the schedule.

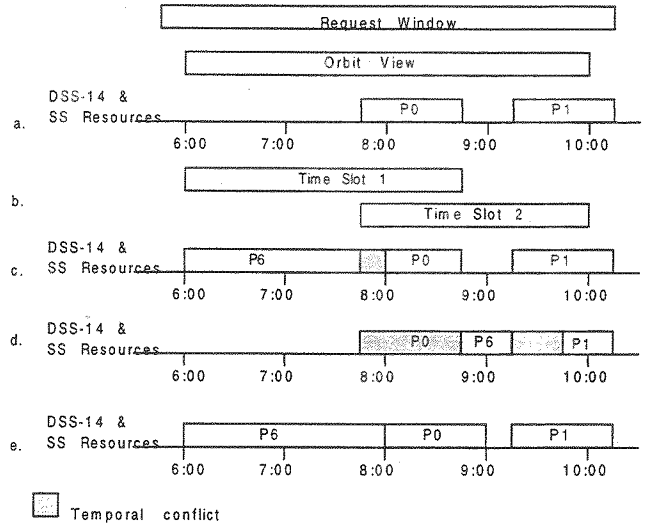

Figure 6: Priority-Based Scheduling Example; a) initial condition; b) valid time slots; c) hypothesis 1 causes P0 conflicts; d) hypothesis 2 causes P0 & P1 conflicts; e) final schedule after placing P6 at 6:00am and rescheduling P0 to 8:00am.

7. Rescheduling Context

Rescheduling DSN tracks is often necessary due to: equipment outages, last minute track requests, last minute changes to scheduled tracks, and atmospheric conditions impact on tracking capabilities. Rescheduling can occur in two ways: (1) it can be initiated top-down due to a change to a previously scheduled track or addition of another request; and (2) it can occur bottom-up in that equipment outages can occur or tracks can fail necessitating rescheduling. In the event of a new or modified request, DANS uses localized search to consider alternative methods for satisfying the new request (as previously described). This search uses as its bounding function a disruption cost measure which accounts for the overhead involved in moving already scheduled tracks and also a satisfaction measure accounting for what level of requests have been satisfied. Because we use branch and bound techniques DANS can guarantee that it will provide a reschedule optimal with respect to the combined disruption and satisfaction cost function.

In the event of a change in equipment availability, we are examining two solution methods. In both methods DANS first updates resource timelines to reflect the new resource level. Then, depending on the size of the change there are two options. First, if the change is localized DANS can perform branch and bound search (using the previously described cost function) to re-evaluate requests in light of the new equipment situation. However, if the change is too large in scope This exhaustive search is intractable. If an antenna goes down for a several day period the cascading effect on tracks can be quite great and thus rule out exhaustive search techniques. In these cases DANS can instead first performs prioritized pre-emption to remove low-priority tracks to remove conflicts (by removing the lowest priority tracks participating in each conflict) and then re-evaluate project requests. This approach requires far less search but can produce suboptimal results (with respect to the twin goals of minimizing disruption and maximizing request satisfaction).

8. DANS LEO-T Demonstration

The DANS rescheduling system is currently being considered for two DSN scheduling applications: Network wide Network Preparation and Planning (NPP) datasets and also scheduling a simulated network of automated terminals designed to service low earth orbiting antennas (LEO-T). In this section we describe the demonstration of DANS on the simulated network of LEO-T antennas.

As part of an investigation into alternative low-cost means of providing communications services, JPL has been investigating the feasibility of fielding a network of small (3-meter) highly automated antenna stations which would use a resident workstation to operate the antenna and be capable of unattended normal operations [9]. This network would need to be scheduled automatically in order to fully automate communications services to low earth orbiting spacecraft. Such a scheduler would accept as inputs a set of requests for tracks from the projects supported by the LEO-T network and allocate coverage slots to be provided by the LEO-T stations.

In a demonstration of the DANS scheduling system we have scheduled a simulated LEO-T network, allocating tracks in response to real projects currently being supported by the regular DSN 26 Meter subnetwork. we simulated five LEO-T sites, using actual candidate sights in order to force realistic satellite to ground station geometries. These sites were: Pasadena, CA, Fairbanks, Alaska, Guam, US Protectorate, Spitzbergen Norway, and Kourou Africa. We then used five existing supported low-earth orbiting projects as the simulated project users of the LEO-T network: sampex-1, solar-a, strv-1a, strv-1b, and topex-1. Using existing orbiting projects allows us easy access orbital pattern and request data.

For each of these projects, we then generated requests that each project be covered to the maximal extent possible (i.e. each project requested continuous coverage from all possible stations). This was implemented by submitting a tracking request for each possible spacecraft to ground station view. In some cases where view periods were quite long (e.g., for the STRV-1a and STRV-1b projects) we artificially segmented the each view period into 3 equal time tracking requests in order to allow the LEO-T stations greater flexibility in covering spacecraft.

In the LEO-T demonstration, DANS used a very simple incremental schedule construction algorithm which is shown below.

```
for each project p in P
  for request r in p
    attempt to place r
    if r causes a conflict
    then remove the lowest priority
    track participating in the conflict
```

As the algorithm shows, DANS simply schedules the requests by considering each request and strongly preferring the highest priority requests. Note that this simple scheduling algorithm does not reflect that projects do not really need all of the possible tracks and that a projects priority is generally a function of the number of tracks that it has received so far. Using this simple algorithm, the DANS scheduler was able to solve the problems very quickly, the total problem includes consideration of hundreds tracking activities and 5 resources (the LEO-T stations themselves). DANS was able to generate the schedule in tens of CPU seconds running on a Sparcstation 20 with 64MB RAM.

9. Conclusions

This paper has described DANS, an automated scheduling system being developed to schedule DSN resources. DANS uses localized search and priority-based pre-emption to perform priority-based rescheduling in response to changing network demand. DANS first considers the antenna allocations, then considers allocation of the 5-13 subsystems per track (out of the tens of shared subsystems at each antenna complex) used by each track. DANS uses localized priority-driven, best first search to efficiently consider the large set of possible subsystems schedules.

References

[1] M. Deale, M. Yvanovich, D. Schnitzius, D. Kautz, M. Carpenter, M. Zweben, G. Davis, and B. Daun, "The Space Shuttle Ground Processing System," in *Intelligent Scheduling*, Morgan Kaufman, San Francisco, 1994.

[2] Deep Space Network, *Jet Propulsion Laboratory Publication 400-517*, April 1994.

[3] G. Fox and C. Borden, "Low Earth Orbiter Resource Allocation and Cpacity Planning for the Deep Space Network Using LEO4CAST," *Telecommunications and Data Acquisition* 42-118, pp. 169-178, August 1994.

[4] R. W. Hill, Jr., S. A. Chien, K. V. Fayyad, C. Smyth, T. Santos, and R. Bevan, "Sequence of Events Driven Automation of the Deep Space Network," *Telecommunications and Data Acquisition* 42-124, October-December 1995.

[5] M. Johnston & S. Minton, "Analyzing a Heuristic Strategy for Constraint Satisfaction and Scheduling," in *Intelligent Scheduling*, Morgan Kaufman, San Franc., 1994.

[6] E. Kan, J. Rosas, and Q. Vu, "Operations Mission Planner - 26M Users Guide Modified 1.0", *JPL Technical Document D-10092*, April 1996.

[7] S. J. Loyola, "PC4CAST - A Tool for Deep Space Netowrk Load Forecasting and Capacity Planning," *Telecommunications and Data Acquisition*, 42-114, pp. 170-194, August 1993.

[8] M. Zweben, B. Daun, E. Davis, & M. Deale, Scheduling and Rescheduling with Iterative Repair, in *Intelligent Scheduling*, Morgan Kaufman, San Fran., 1994.

[9] N. Golshan, W. Rafferty, C. Ruggier, M. Wilhelm, B. Haggerty, M. Stockett, J. Cucchisi, and D. McWatters, Low Earth Orbiter Demonstration Terminal, *Telecommunications and Data Acquisition 42-125*, 1996.

[10] Weinstein, S. C. Whitney, D. Zoch, M. Tong, "Developing a Scheduling System in Ada-Problems and Lessons Learned." *Fourth Annual NASA Ada User's Symposium Proceedings*. Hampton,VA: NASA/Langley Research Center, 1992.

[11] Wilkinson, G. R. Monetleone, S. Weinstein, M. Mohler, D. Zoch, and M. Tong, "Achieving Reutilization of Scheduling Software through Abstraction and Generalization." *Computing in Aerospace 10 Proceedings*. Washington, DC: American Inst. Aero. & Astro., 1995.