Appears in Proceedings of the 1997 IEEE Aerospace Conference, Aspen, CO, February 1997.

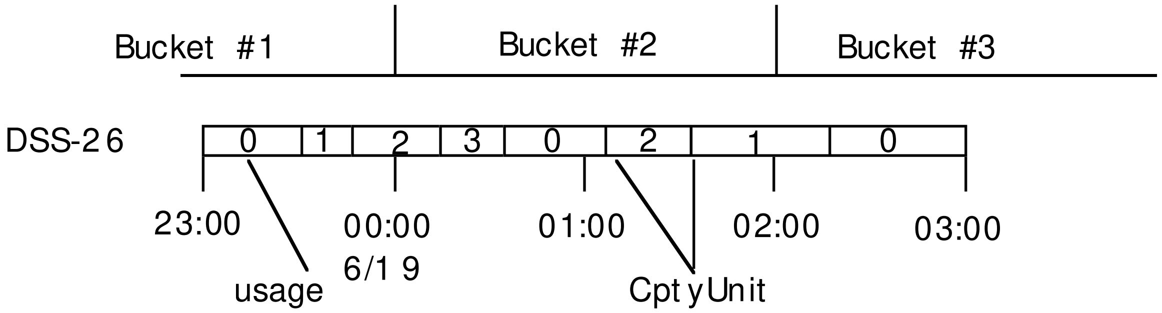

Figure 3: Capacity Timeline Resource Representation

5. INFERENCE ENGINE

The current version of DANS employs a priority-based inference strategy to satisfy the project requests. Future versions will incorporate other strategies to make the reasoning process more robust.

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, and they can only commit to one activity at any given time. On the other hand, there are many identical subsystem resources that support unexpected events. Subsequently, the scheduling of the subsystems is still important but less critical. Using a top-down approach, the inference engine can focus on conflict resolution of critical resources first. Then, it can reason at the next level of abstraction to resolve less-critical resource conflicts. The Priority-Based strategy exploits this representation to reason at both the antenna and the subsystem levels.

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 inferencing and subsystem inferencing flowcharts are shown in Figures 4 and 5.

At the DSN 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, where an orbit view is a time period during which a specific antenna can physically view the spacecraft. The result is a list of one or more valid intervals within the window. For each possible interval, the system tries to schedule the ACT into it. When conflict arises between the ACT and the current schedule, 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 is tracking the cost of each solution. The cost of a solution 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.

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 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.

Appears in Proceedings of the 1997 IEEE Aerospace Conference, Aspen, CO, February 1997.

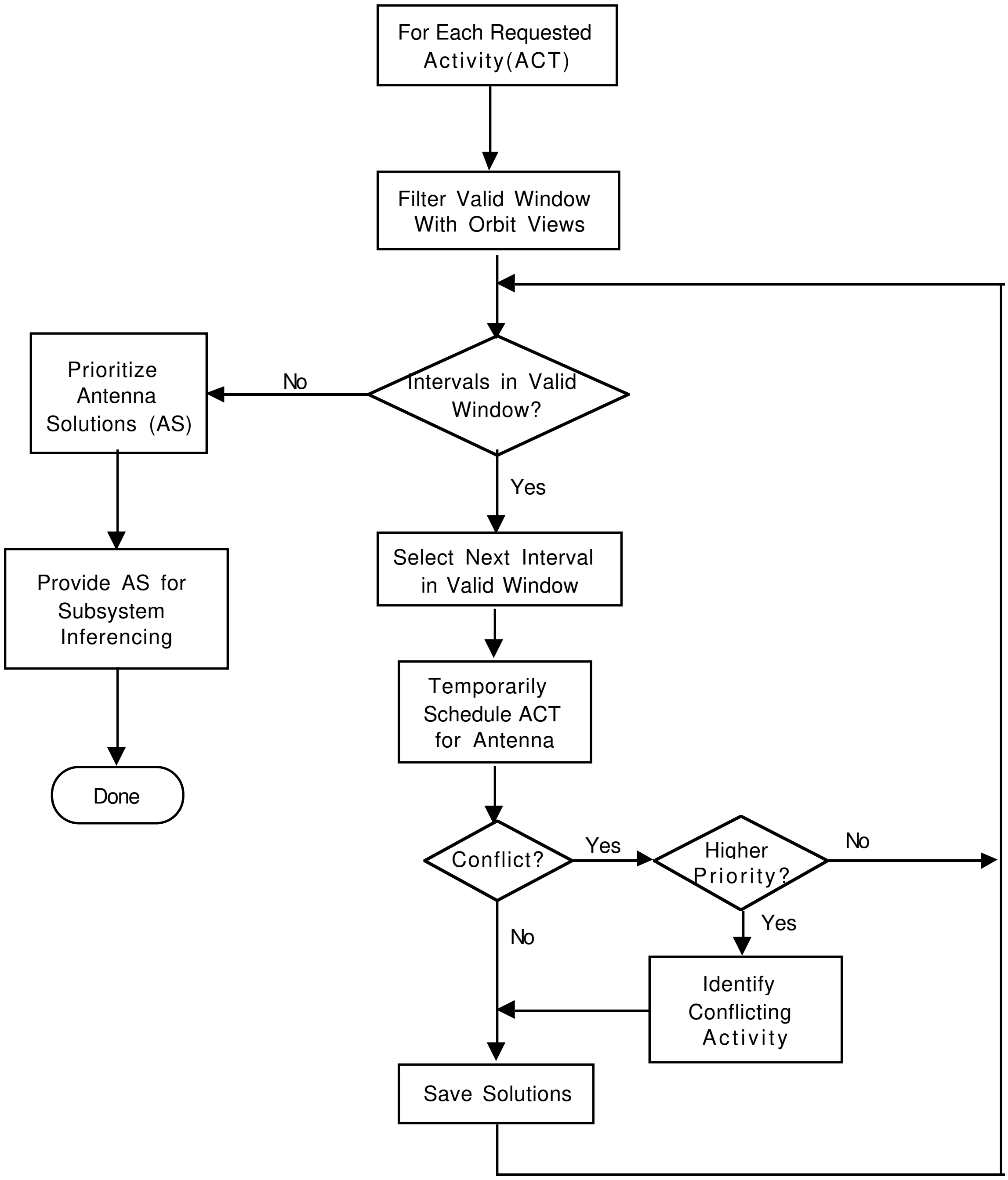

Figure 4: Priority-Based Antenna Inferencing Flowchart

Appears in Proceedings of the 1997 IEEE Aerospace Conference, Aspen, CO, February 1997.

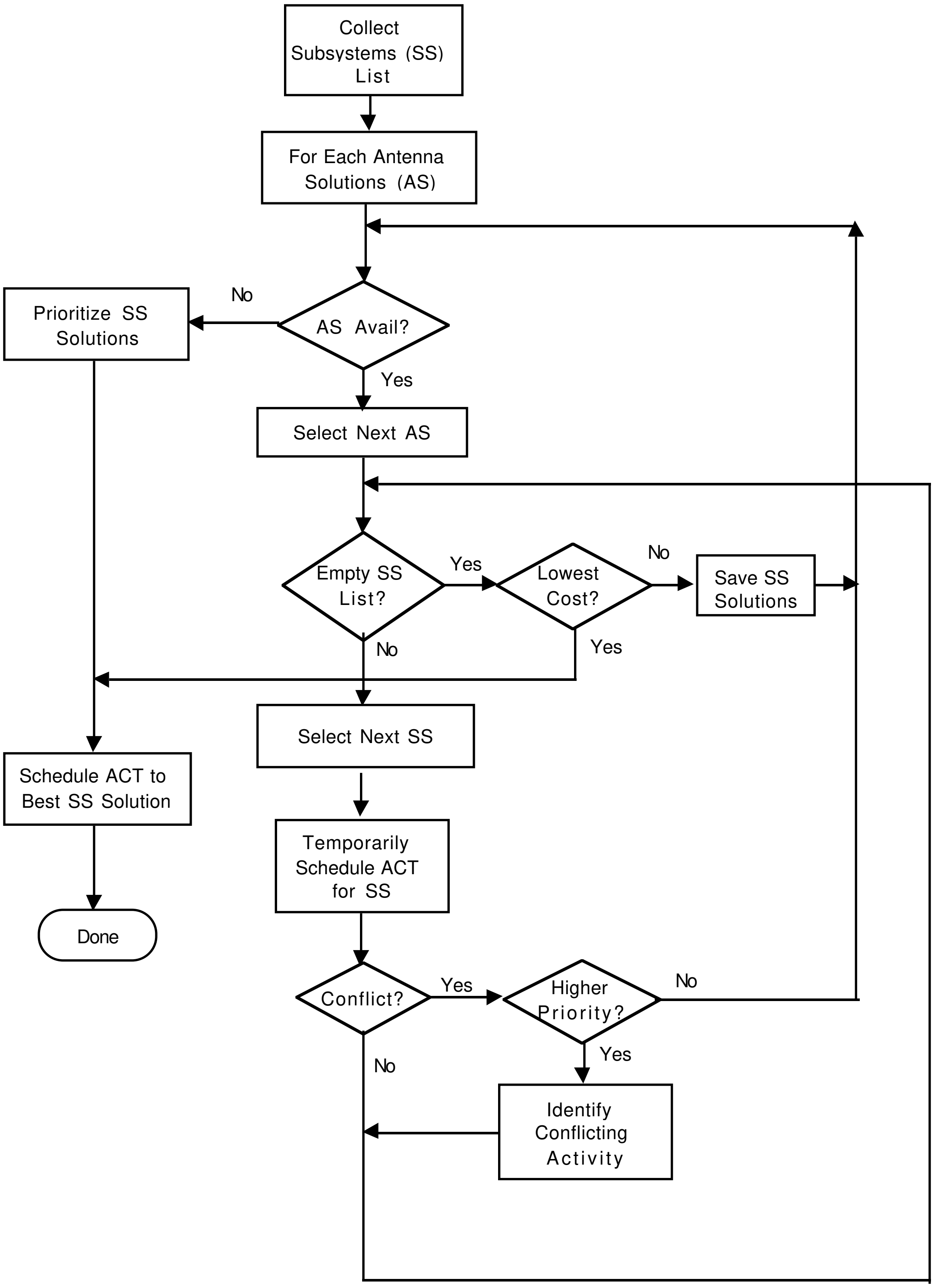

Figure 5: Priority-Based Subsystem Inferencing Flowchart

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.

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.

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

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.

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

Appears in Proceedings of the 1997 IEEE Aerospace Conference, Aspen, CO, February 1997.

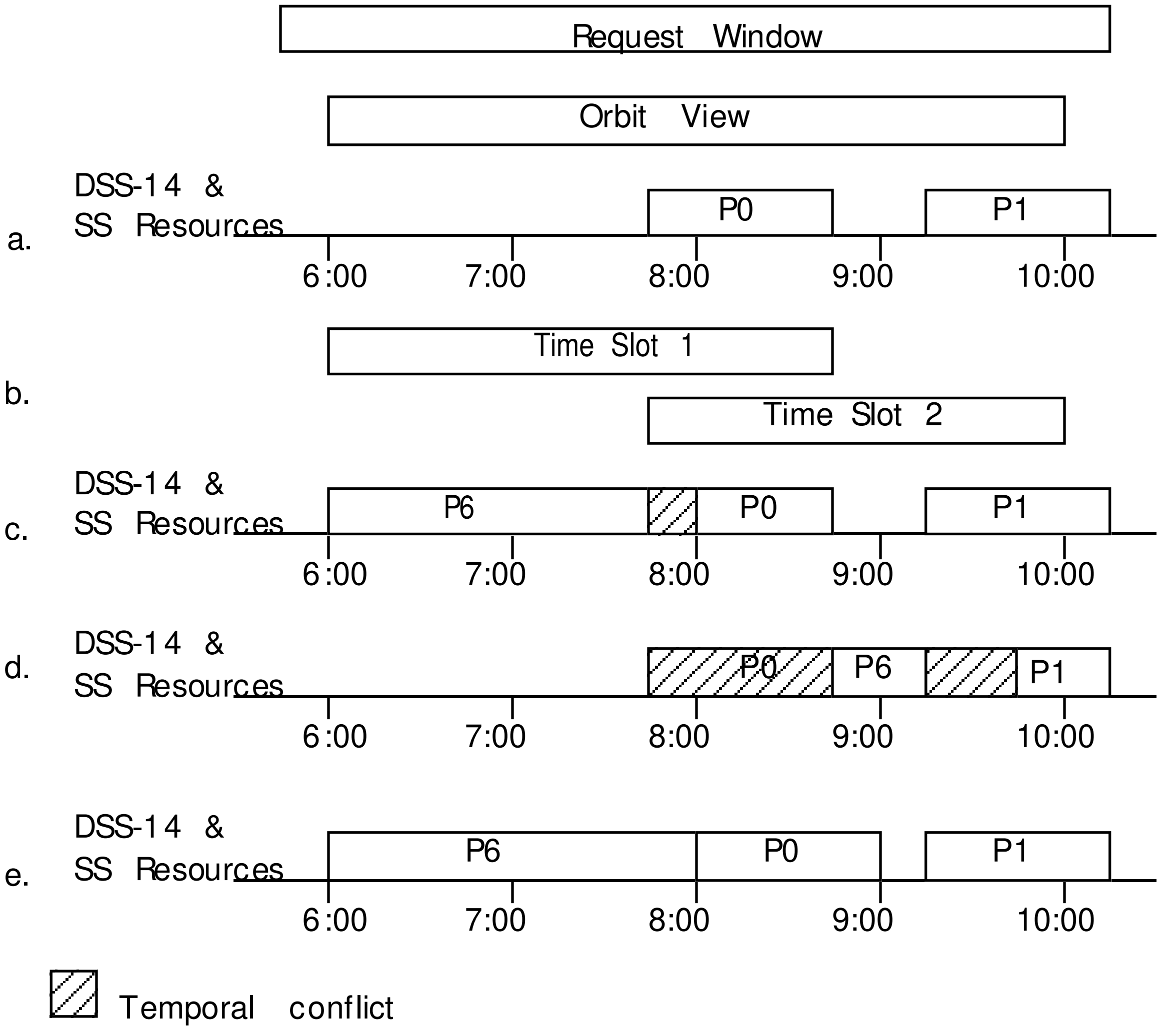

Figure 6: Priority-Based Scheduling Example; a) initial condition; b) valid time slots; c) hypothesis 1 causes P0 conflicts; d) hypothesis 2 causes P0 & P1 conflicts; e) final schedule after placing P6 at 6:00am and rescheduling P0 to 8:00am.

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 currently examining two solution methods. In both methods DANS first updates all 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 exhaustive search is intractable. For example, if an antenna unexpectedly 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: JPL Deep Space 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. These stations would use a resident workstation to operate the antenna and be capable of unattended normal operations [9].

However, such a network of LEO-T stations 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.

In this demonstration, we simulated five LEO-T sites, using actual candidate sights in order to force realistic satellite to ground station geometries.

Station	Location	Latitude	Longitude
JPL	Pasadena, CA	34.204819	118.173223
FAIR	Fairbanks, AK	64.978001	147.499239
GUAM	Guam, US Prot.	13.589282	144.868408
SPIT	Spitsbergen, Norway	80	10
KOUR	Kourou, Africa	5.252180	-52.805959

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.

Current efforts (see the discussion section below) involve extending the DANS system to more accurately represent tracking requests to reflect the true number of tracks needed and how priority depends on how globally the tracks have been allocated to a project.

9. DISCUSSIONS AND CONCLUSIONS

This paper has described the Demand-based Automated Network Scheduling System (DANS), an automated scheduling system being developed at the Jet Propulsion Laboratory (JPL) to schedule DSN resources. DANS uses localized search and priority-based pre-emption to perform priority-based rescheduling in response to changing network demand. In this techniques, 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) 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] T. Estlin, X. Wang, A. Govindjee, and S. Chien, "DPLAN Deep Space Network Antenna Operations Planner Programmers Guide Version 1.0," *JPL Technical Document D-13377*, Jet Propulsion Laboratory, California Institute of Technology, February 1996.

[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 Francisco, 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.

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.

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 Institute of Aeronautics & Astronautics, 1995.

[8] M. Zweben, B. Daun, E. Davis, & M. Deale, Scheduling and Rescheduling with Iterative Repair, in

Intelligent Scheduling, Morgan Kaufman, San Francisco, 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*, March 1996.

Steve Chien *is Technical Group Supervisor of the Artificial Intelligence Group of the Jet Propulsion Laboratory, California Institute of Technology where he leads efforts in research and development of automated planning and scheduling systems. He is also an adjunct assistant professor in the Department of Computer Science at the University of Southern California. He holds a B.S., M.S., and Ph.D. in Computer Science from the University of Illinois. His research interests are in the areas of: planning and scheduling, operations research, and machine learning.*

Ray Lam *is a member of the technical staff of the Artificial Intelligence Group of the Jet Propulsion Laboratory, California Institute of Technology where he participates on the research and developent of automated scheduling systems. He holds a B.S. degree in Electrical Engineering from University of California, Los Angeles; and a M.S. degree in Computer Engineering from University of Southern California. His research interests are in the areas of: scheduling, operations research, and neural networks.*

Quoc Vu *is a member of the technical staff of the Automation and Scheduling Technology Group of the Jet Propulsion Laboratory, California Institute of Technology where he participates on the research and development of automated scheduling systems. He holds a B.S. degree in Computer Science from California Polytechnic University, San Luis Obispo; and a M.S. degree in Computer Science from Asuza Pacific University, California. His research interests are in the areas of: planning and scheduling, operations research, and graphical user interface.*