Predictive Assessment for Phased Array Antenna Scheduling

Randy Jensen[1], Richard Stottler[2], David Breeden[3], Bart Presnell[4], Kyle Mahan[5]
Stottler Henke Associates, Inc., San Mateo, CA 94404

and

Gary Scalzi[6]
Air Force Research Labs, Wright Patterson AFB, OH 45433

In contrast to traditional parabolic dish antennas which must be mechanically steered to point at satellites, phased array antennas operate by electronically activating subarrays in configurations that can be maneuvered across the surface of the antenna without any physical movement. This leads to many benefits including an increase in capacity from the fact that multiple active areas can be enabled simultaneously on the same phased array. Under this concept, a single antenna can support multiple simultaneous contacts, although there are still constraints specific to any implementation. The phased array is made up of subarrays with Transmit and Receive modules, which present their own specific limitations. In one implementation, an individual Transmit / Receive module can simultaneously support two Receive beams from two distinct satellites, but only one Transmit beam. As the active areas for separate beams move across the surface of the antenna, the conditions where they overlap may overload specific modules in the overlapping area. Thus when constructing automated logic for allocating supports to antennas, a predictive compatibility assessment mechanism is required to determine if a trial allocation will lead to conditions that violate the constraints of the antenna hardware. When two or more supports will be active on the same phased array, a predictive assessment must consider their active areas and their paths across the surface over time to identify conflicts and then determine if such conflicts can be remedied. A phased array antenna allows active areas to be shifted away from their optimal positions, as long as they are also increased in size to compensate. This paper describes a central piece of the assessment mechanism implemented in conjunction with an automated scheduling algorithm, which predicts whether beam conflicts will occur, and whether they can be deconflicted with changes in position and size. A series of small experiments was conducted to determine boundary conditions for the deconfliction algorithm, such as a threshold number of iterations for incrementally separating beams, and impacts from the positions of active areas such as proximity to an edge of the antenna. These experiments also explored how the deconfliction thresholds change with different combinations of larger and smaller active areas. This paper summarizes the results of these experiments, and also related elements of the algorithm such as the information preserved from the deconfliction process to inform the logic for trying alternative allocations when necessary. Finally, this paper also presents results from a larger experiment conducted to compare performance in two scenarios – a baseline scenario with only parabolic antennas, and an alternate scenario with phased array antennas substituted at several ground stations. Using a sample set of satellite support requests over a 24 hour period, we compare overall performance in the phased array scenario with the baseline, in terms of the ability to satisfy requests in each case via the automated scheduling algorithm.

[1] Group Manager, 951 Mariner's Island Blvd., Suite 360, San Mateo, CA 94404, AIAA Contributor.
[2] President, 951 Mariner's Island Blvd., Suite 360, San Mateo, CA 94404, AIAA Member.
[3] Software Engineer, 951 Mariner's Island Blvd., Suite 360, San Mateo, CA 94404, AIAA Contributor.
[4] Software Engineer, 951 Mariner's Island Blvd., Suite 360, San Mateo, CA 94404, AIAA Contributor.
[5] Software Engineer, 951 Mariner's Island Blvd., Suite 360, San Mateo, CA 94404, AIAA Contributor.
[6] Chief, AFRL/RYMDA, 2241 Avionics Circle, Bldg 620, Wright-Patterson AFB, OH 45433, AIAA Contributor.

Nomenclature

BA	=	Bottleneck Avoidance
GDPAA	=	Geodesic Dome Phased Array Antenna
LOS	=	Line of Sight
T/R	=	Transmit / Receive

I. Introduction

IN contrast to traditional parabolic dish antennas which must be mechanically steered to point at satellites, phased array antennas operate by electronically activating subarrays in configurations that can be maneuvered across the surface of the antenna without any physical movement. This leads to many benefits including an increase in capacity from the fact that multiple active areas can be enabled simultaneously on the same phased array. Under this concept, a single antenna can support multiple simultaneous contacts, although there are still constraints specific to any implementation. The phased array is made up of subarrays with Transmit and Receive (T/R) modules, which present their own specific limitations. In one working design of a Geodesic Dome Phased Array Antenna (GDPAA) architecture, an individual T/R module can simultaneously support two Receive beams from two distinct satellites, but only one Transmit beam. As the active areas for separate beams move across the surface of the antenna, the conditions where they overlap may overload specific modules in the overlapping area. Thus when constructing automated logic for allocating supports to antennas, a predictive compatibility assessment mechanism is required to determine if a trial allocation will lead to conditions that violate the constraints of the antenna hardware. When two or more supports will be active on the same phased array, a predictive assessment must consider their active areas and their paths across the surface over time to identify conflicts and then determine if such conflicts can be remedied.

A phased array antenna allows active areas to be shifted away from their optimal positions, as long as they are also increased in size to compensate. This is the primary method for deconflicting the conditions where beams overlap, and it involves a variety of variables and constraints that must be tuned in order to find a solution, if one exists. For example, there are limits on how far an active area can be shifted away from its optimal position, and there may also be limits on the direction, depending on the starting location on the phased array. The mechanism to automate this deconfliction process when conditions require it is implemented as a central piece of the predictive mechanism in conjunction with an automated scheduling algorithm. A series of small experiments was conducted to determine boundary conditions for the deconfliction algorithm, such as a threshold number of iterations for incrementally separating beams, and impacts from the positions of active areas such as proximity to an edge of the antenna. These experiments also explored how the deconfliction thresholds change with different combinations of larger and smaller active areas. This paper summarizes the results of these experiments, and also related elements of the algorithm such as the information preserved from the deconfliction process to inform the logic for trying alternative allocations when necessary.

Following the integration of the predictive assessment capability with an automated scheduling framework, an additional experiment was conducted to compare performance in two scenarios – a baseline scenario with only parabolic antennas, and an alternate scenario with phased array antennas substituted at several ground stations. Using a sample set of satellite support requests over a 24 hour period, we compare overall performance in the phased array scenario with the baseline, in terms of the ability to satisfy requests in each case via the automated scheduling algorithm. Despite the additional complexity of the phased array problem, performance was roughly comparable in both scenarios, with very close proportions of successfully scheduled contacts, and also similar processing times as well.

II. Background

In order to explore examples and discuss methodologies, we first introduce definitions for several terms.

Term	Description
Beam	An active line of sight (LOS) contact between a satellite and an antenna, with associated power requirements to accommodate the signal to or from the satellite.
Transmit/Receive (T/R) Module	An individual module on the surface of a phased array, with simultaneous capabilities to send and receive signals to and from a satellite.

Term	Description
Active Area	A bounded region on the surface of a phased array, with an associated shape and quantity of constituent T/R modules to support the communication requirements for a beam. Often the active area is a circular region mapped onto the surface of the phased array.
Optimal Position	The ideal placement of a beam active area on a phased array, based on the angle of visibility to the satellite. A beam's ideal placement when using a circular active area is centered on the optimal position. If the beam will be moved off of the optimal position to avoid an overlap, then its active area must increase in size. Also there is a limit how far a beam may be moved off of optimal.
Beam Path	A continuous route across the surface of the phased array, plotting the locations of a beam's active area throughout the duration of the contact.
Incompatibility	A condition involving two or more beams whose paths cannot be deconflicted, which means their active areas overlap in at least one instance, and cannot be separated far enough to eliminate the overlap.
Conflict	A condition involving two or more beams with an incompatibility that cannot be resolved. In implementations where a Receive module can accommodate two simultaneous contacts, a Receive incompatibility can be resolved by allowing the overlap, and allocating the beams to different ports on the phased array. In implementations where a Transmit module can only accommodate one contact, any incompatibility is equivalent to a conflict.

Figure 1 shows an example in both the Transmit and Receive contexts, shown simultaneously side-by-side, for two beams with overlapping active areas at the optimal positions.

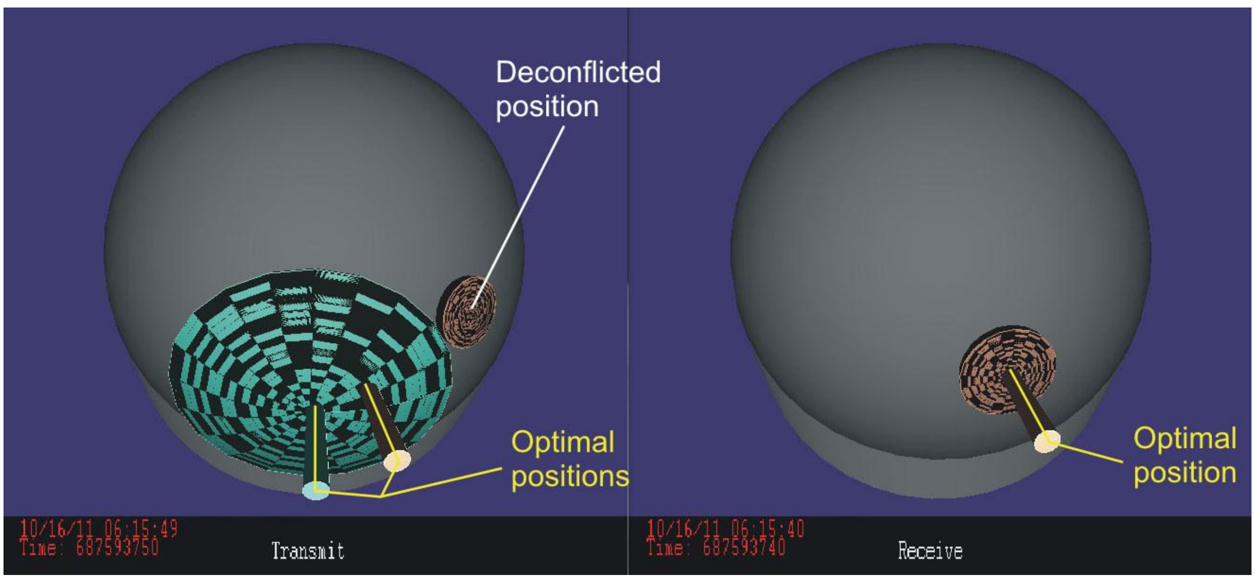

Figure 1. Determining if Transmit and Receive overlaps represent conflicts.

Although the beams overlap, they are not in conflict, because it is possible to find a solution in both contexts. In this visualization, circular areas represent the placement of active areas in the beam path solution. These placements may differ from the optimal positions, which are indicated by extruded poles representing the optimal centerpoints. Looking first at the Transmit case on the left, the beams must be on the same port, so their overlapping active areas must be resolved through deconfliction. A solution is found where one beam is shifted away from its optimal position, to eliminate the overlap with the larger active area of the second beam. This deconflicted position is shown above, on the left. Looking next at the Receive case on the right, in this context it is possible to accommodate overlaps by assigning beams to different ports. There are two available ports, so the view on the right shows the allocation to one of the two ports, for the Receive context. In this case, because the two beams are simply assigned to different ports, no further deconfliction is necessary. Each beam's active area can remain at its optimal position, as shown for one of the beams in the example. Thus it can be concluded that there is no conflict, for this allocation, at this moment in time.

The objective of the predictive assessment capability is to determine if such solutions can be found, or identify likely incompatibilities and conflicts. This provides a utility for an automated scheduling tool to consult in the process of creating potential assignments. For example, a sample use case may involve a network of ground stations, where some have multi-beam GDPAAs and others have traditional single-beam reflector antennas. In the network-level scheduling process, beams assigned to a phased array antenna go through several extra steps to determine feasibility with respect to active area incompatibilities. There are several different situations where a scheduling system needs to consult the phased array allocation and path planning logic, which roughly break down into two categories:

- **Automated scheduling of single tasks or groups of tasks.** Automated scheduling is performed in several stages, which are all initiated by a scheduler user. A typical use case involves successive forms of automated scheduling on a time-delimited set of support requests (also known as tasks), such as an 8 hour or 24 hour period. As part of this function, for scenarios involving GDPAAs, the network-level scheduling process relies on a predictive assessment capability to determine if an allocation will be successful, and ultimately to fully calculate a solution if so.
- **Manual scheduling of single tasks or small numbers of tasks.** Human schedulers use an interface which allows for individual supports to be manually repositioned, created, or deleted. For any of these activities, there is still an automated capability provided by the scheduling engine to validate the schedule. Once again in the case of GDPAA allocations it is necessary to consult the phased array logic to re-evaluate allocations resulting from a manual move.

In order to support the different contexts for network-level scheduling, the allocation and path planning utilities are structured to take place in two passes. The **initial pass** is a recurring procedure performed during automated network-level scheduling, with each beam to be scheduled on a phased array. This generates approximated predictions for incompatibilities between beams that may be allocated to the same phased array, and also attempts to find viable port allocations for both the transmit and receive operations, based on these predicted incompatibilities. Because of the frequency of calls to the initial pass procedure, there is a strong need to optimize its computational efficiency. This motivates an approach where the initial pass attempts to predict incompatibilities through an approximation of the beam path planning procedure, rather than calculating complete deconflicted beam paths for each candidate. This analysis is an approximation because it only considers the points of closest proximity between beams. It is much less computationally intensive to attempt deconfliction on a single point for each beam pair, as opposed to many samples along the optimal paths. Since this considers the worst cases of closest proximities, it is still likely to identify incompatibilities in most cases where they exist. As the initial pass is repeated with successive beams, it maintains persistent records of incompatibilities identified earlier, to short-circuit the logic for subsequent calls examining the same set of beams.

The **global pass** is a one-time final verification procedure performed at the conclusion of the network-level allocation process, when an initial schedule has been developed. The task of calculating final beam paths for all beams allocated to a phased array is performed at this stage. The scope of the global pass varies with the context in which network-level scheduling is being performed. In automated scheduling, the global pass is performed on the entire duration or grouping that is being scheduled. In manual scheduling, the global pass is performed on the minimum collection of supports that are contiguous or overlapping with the supports being scheduled. Essentially this can be considered the group of tasks surrounding a given task, and bounded by any condition where there is one or zero simultaneous supports on the same GDPAA.

III. Optimizing Deconfliction for Predictive Assessment

Both the initial pass and the global pass involve beam path planning. When the active areas for two or more beams overlap, the deconfliction process is a general method used to determine if the beams can be moved off their optimal positions to eliminate an overlap. The deconfliction process involves iterations of shifting beam positions apart in increments until a solution is found, or an iteration limit is reached, or a violation of constraints is reached. Figure 2 shows an example of four beams that have been successfully deconflicted.

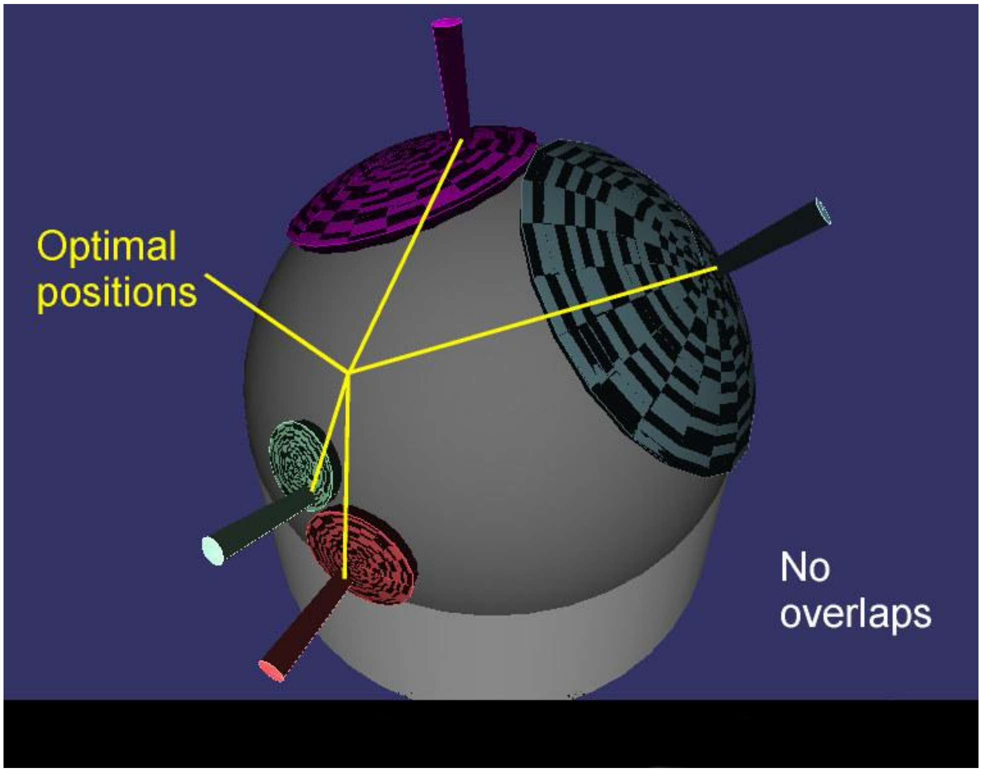

Figure 2. Four simultaneous beams with deconflicted paths. *All four beams in this example have active areas shifted away from their optimal positions for deconfliction.*

In this example, there are two pairs of beams that overlap when centered on their optimal positions. All active areas are shown in positions that are no longer centered on the optimal positions. This illustrates how pairs of beams are separated to eliminate overlaps, if this can be accomplished without violating other constraints.

There are several variables involved in iterative deconfliction, including the direction, the incremental distance to move, and the threshold limit on the number of iterations. The optimal values for these may vary with the nature of the beams involved. For example, a beam with a large active area may have less flexibility for relocation off of the optimal position, due to the constraints on the maximum offset angle. Likewise, a beam near the skirt of the GDPAA may have less flexibility due to its position and the boundary condition of useable area on the phased array.

In order to optimize the deconfliction procedure, experiments were conducted to determine thresholds that can yield minimal false negatives and also minimal excess iterations when a solution is unlikely to be found. A false negative would be a case where the deconfliction procedure fails to find a solution, when in fact the problem is solvable (i.e., two overlapping beams in question can be positioned such that they do not overlap and still do not violate path planning constraints). Excess iterations are a condition where deconfliction attempts continue beyond the threshold where a solution would be found, if there is one.

In our experiments, we generated test cases with n simultaneous beams allocated to a GDPAA (n = 2, 3, 4, 5, or 6). The analysis is a snapshot in time with n beams initially placed in their optimal positions at that instant. This represents an instant where the predictive assessment would be performed, notionally a worst case condition of closest proximity for the beams involved. Test cases are constructed with randomly generated beams from an even distribution of active area sizes and optimal positions. In order to match the conditions within the initial pass incompatibility prediction procedure where beams are incrementally added and tested during allocation, the test cases are constructed so that (n-1) beams are deconflictable. This may mean either they have no overlap, or overlaps exist but there is a deconfliction solution. So, assuming that a solution has been found for the (n-1) beams, the question is then, how many iterations are necessary to deconflict the nth beam when it is added? Seeking to look at statistical outcomes, 10,000 configurations of n beams were generated for each case (n = 2, 3, 4, 5, or 6). Then 50 deconfliction iterations were run with each configuration. Table 1 shows the thresholds for each case. The required number of iterations refers to the threshold or plateau condition, after which further iterations would fail to yield a solution if one had not been found. This table also shows the success rate achieved for the entire group of 10,000 configurations, after each went through (a maximum of) the threshold number of iterations.

Table 1. Deconfliction thresholds.

# beams	Required # of iterations	Success rate at threshold
2	5	95%
3	31	92%
4	34	81%
5	39	37%
6	40	12%

Although the current antenna hardware design does not support situations with 5 or 6 beams allocated to a GDPAA, we included these in our experimentation out of general interest. Naturally it is expected that the deconfliction success rate drops significantly with 5 or 6 beams. Looking at the difference in the number of required iterations for 2 beams versus 3 beams, it helps optimize the algorithm considerably to have pre-calculated values for the different thresholds. For example in the 2 beam case, if a solution has not been found after 5 iterations, then the algorithm can essentially save 26 iterations by halting any further attempts.

A secondary area of investigation with the data sets generated from this experiment was to explore the impact from the number of overlapping pairs in a set of beams, as well as the number of beams near the equator of the dome. Table 2 shows statistical data gathered from the same experiments, to identify how deconfliction success relates to the number of overlapping beam pairs involved.

Table 2. Impact on deconfliction from number of overlaps.

# beams	Success rate per number of overlapping pairs						
	0	1	2	3	4	5	6
3	100%	96%	97%	74%	-	-	-
4	100%	95%	95%	80%	75%	60%	13%

Once again, the results match expectations. When no beams overlap, deconfliction is obviously successful 100% of the time. The success rate begins to fall as the number of overlapping pairs increases, although not as dramatically as one might imagine. Only in the 4 beam case where all beams overlap (i.e., there are 6 overlapping pairs), does the success rate drop to a dramatically reduced level.

With regard to the proximity of beam optimal positions to the boundary condition at the dome equator, a surprising finding was that this does not seem to impact the success rates or the threshold for the number of iterations. In other words, situations where more beams are close to the equator are not necessarily harder to deconflict than cases with fewer beams near the equator. Thus there is no factor for this in the deconfliction thresholds used by our algorithm, although this may be an area to explore further with future experimentation.

IV. Integration with Scheduling Tool and Experimental Results

The network-level scheduling component is not the subject of this paper, but a brief description is helpful to establish the context for how beams are handled. As described earlier, there are use cases involving either automated or manual scheduling situations, and in either case, phased array allocations and paths must be developed when GDPAAs are involved.

For a discussion of network-level scheduling, it is important to be clear about terminology, particularly with respect to the discussion of conflicts. From a phased array perspective, a conflict refers to a condition where it is not possible to generate a beam path for a support that will have no incompatibilities with other beams on the same GDPAA. At the network level when allocating supports to antennas, a scheduling conflict refers to a condition where an allocation violates a scheduling constraint. A typical example of a scheduling conflict would be two supports allocated to the same parabolic antenna with overlapping times. More specific equipment constraints may lead to other forms of scheduling conflicts.

A. Network-Level Scheduling Algorithm Overview

Within the automated scheduling procedure, there are three phases: baseline scheduling, bottleneck scheduling, and scheduling with business rules. Baseline scheduling is universally performed with the very first load of a set of support requests. The objective in baseline scheduling is essentially to distribute requests into the schedule as tasks, with minimal up-front processing so that the result essentially mirrors the requests as closely as possible. Where constraints are given, they are followed in baseline scheduling. Where there is flexibility, some automated scheduling logic is applied to choose between available options. The result of baseline scheduling is a populated schedule, with schedule conflicts identified. By the same token, it is necessary to identify beam path conflicts for any beams initially assigned to GDPAAs as a result of baseline scheduling. Thus in baseline scheduling the initial pass incompatibility predictions are used for each beam assignment, and the global pass is used to verify the conclusions about incompatibility and/or beam path conflicts. However, the primary objective for these calls in baseline scheduling is to identify beam path conflicts, and not necessarily to mitigate them in the schedule. Therefore it is common to see beam path conflicts on GDPAA assignments after baseline scheduling.

The results of incompatibility checking during baseline scheduling are also cached for reuse during subsequent scheduling activities. This is an optimization that helps speed up processing time for the allocation and planning steps associated with any subsequent schedule changes.

In the next step, a Bottleneck Avoidance (BA) algorithm is applied to find and resolve many of the schedule conflicts that remain after baseline scheduling. BA is an artificial intelligence technique that is well-suited to satellite scheduling as it mimics the processes of human schedulers. [1,2] In the BA methodology, beam requests are organized into a precedence scheme based on resource contention, where the most heavily requested or loaded resources are treated as having the highest contention (or bottlenecks). BA then iterates through beam requests in order of contention, starting with the worst bottlenecks, and schedules them with the goal of reducing bottlenecks. In order to interoperate with this sequential process, phased array allocation logic must provide functions that can be consulted one by one with individual beams during network-level bottleneck scheduling.

In the typical use case, bottleneck scheduling is performed on groups of tasks such as an 8 hour or 24 hour period. In this kind of context, the phased array initial pass logic is used with each tentative assignment on a GDPAA. Supposing that a GDPAA is one of the possible allocations for a beam, then the initial pass incompatibility prediction is used to determine if that allocation will be viable. If not, then this allocation is removed from the candidates. However, if it is not viable but still the only candidate allocation, then the beam will be assigned to that GDPAA but it will be marked as an assignment with a beam path conflict. At the conclusion of this process with all bottlenecks having been scheduled, the phased array global pass is then run on the entire scope of the schedule being addressed.

The final step in the automated scheduling sequence involves a mechanism for applying business rules to solve some of the remaining schedule conflicts after bottleneck scheduling. In most cases, BA has resolved the easy schedule conflicts, those that can be solved via legal moves. So the purpose of business rules is to codify situations where some constraints can be bent or even broken in order to satisfy requests that are in conflict. These rules are captured from human schedulers, to identify how rules could be applied in what situation, and to what extent.

There are several ways to apply these strategies. The user has the option to apply business rules to a single task or to all conflicted tasks in the current time period. Rules are ordered such that those that cause the least damage are attempted first. Rather than applying all business rules to one task before moving on to the next task, an iterative algorithm is used to avoid highly damaging one task when a lower damage change to another task may have resolved the conflict. This traverses all conflicted tasks, applying the lowest damage business rule, then traverses a second time applying the two lowest damage business rules, until all conflicts are resolved or until all rules have been applied to all conflicted tasks.

In a manner akin to the other conditions for querying the phased array logic, the process of applying business rules must create port assignments and generate beam paths for any beam that will be allocated to a GDPAA in the course of the schedule changes under the business rules. Because the business rules result in more targeted changes, the global pass phased array routine is applied in a selective manner to the individual supports that are changed, and the associated intervals of contiguous beams that potentially impacted by their chained dependencies.

B. Experimental Data

We sought to use realistic data for testing out the integrated capability. A sample data set was constructed with a realistic distribution of satellites, orbital information, and support requests for a 24 hour period. There are two comparison conditions used for testing:

1. **Scenario1 – baseline with all single-beam reflector antennas.** This is the baseline condition, using a scenario where all antennas are single-beam parabolic reflector antennas, with no GDPAAs. In this scenario there are 19 total antennas, with a total capacity of 19 simultaneous supports.
2. **Scenario2 – GDPAA scenario with several multi-beam antennas substituted in the baseline.** For the second scenario, the only thing that changes is a modified network of ground stations, with GDPAAs substituted at three sites. The same set of satellite requests over a 24 hour period is used as the input, and scheduled with the modified ground stations. In this second scenario, the logic for calculating compatibility and beam paths is incorporated into the scheduling process. In this scenario there are 10 total antennas, with a total capacity of 15 simultaneous supports (these values differ because of the multi-beam capacity of the three GDPAAs).

Note that there is a net drop in total nominal capacity across the two scenarios above, from 19 to 15. Part of the experimental objective in this case was to determine if this drop in capacity led to a reduction in performance in terms of the percent of satellite requests successfully scheduled by the automated system, using a realistic data set. In more general terms, a high level objective of this research is to support investigations with a variety of experimental conditions involving a range of factors in the prospective utilization of phased array antennas, such as the potential placement of GDPAAs within a network of ground stations. This experiment represented one initial direction for specific investigation.

C. Experimental Results

The input data set for the experiments contains 516 satellite support requests over a 24 hour period. Table 3 below shows performance results in the two comparison scenarios, in terms of the proportion of supports successfully scheduled, and also processing time, after each stage of automated scheduling.

Table 3. Scheduling performance results in comparison scenarios.

	Scenario1 - baseline (19 antennas, capacity 19)		Scenario2 – with GDPAAs (10 antennas, capacity 15)	
	# tasks, percentage	processing time	# tasks, percentage	processing time
Resolved tasks with no conflict after baseline scheduling	330 / 516 64%	8 sec	319 / 516 62%	8 sec
Resolved tasks with no conflict after bottleneck scheduling	378 / 515 73%	15 sec	371 / 515 72%	16 sec
Resolved tasks with no conflict after applying business rules	388 / 515 75%	3 sec	390 / 515 76%	3 sec

As the results above show, the final outcome after applying business rules in the last step of automated scheduling is a slightly higher success rate in the GDPAA scenario (Scenario2), with 390 tasks scheduled without conflicts versus 388 in Scenario1. Considering the nominally lower capacity in Scenario2 (15 versus 19) and almost half the number of actual antennas (10 versus 19), this is a very encouraging result. Going into this experiment, an objective was to measure if a proportional performance threshold is reached in Scenario2. Given 15 / 19 the capacity in Scenario2, anything over a threshold of 306 tasks supported (= 15 / 19 * 388 tasks with no conflict in Scenario1) would have been a good result. In that light, the yield of a comparable support outcome (390 versus 388) far exceeded objective goals.

The processing times are also very promising, with little or no difference in the two scenarios, despite the addition of phased array allocation and beam path planning logic in Scenario2. Although processing times may vary with different computing hardware, these results were measured while running on a conventional Windows laptop.

There are several ways in which the results could be analyzed further, although it rapidly becomes difficult to isolate factors contributing to performance. For example, it may be of interest to investigate how many conflicts at each stage are network-level scheduling resource conflicts versus beam path conflicts local to a phased array. However, this statistic would have the potential to be somewhat misleading on its own. There may be instances where tasks are left in a condition with network-level conflicts because it is not possible to find an allocation on a GDPAA that doesn't cause a local-level conflict. And the converse is true as well. Essentially, a network-level conflict may exist because of the avoidance of a local-level conflict on a GDPAA, or vice versa. Thus a simple count of each kind of conflict doesn't necessarily indicate all the contributing factors that lead to that outcome.

In analyzing the outcomes with the experimental data, we did encounter some interesting examples. We found at least one instance where the algorithm used a phased array antenna to successfully resolve a conflict arising from resource contention on a single-beam parabolic antenna. In this example, a satellite was initially allocated to a site with parabolic antennas as a result of the initial baseline scheduling process. But this allocation occurs during a time period with a high degree of contention, leading to numerous conflicts for other tasks assigned to the same ground station at the same time. However, the satellite in question has a simultaneous visibility with one of the ground stations where all parabolic antennas were replaced by a single GDPAA. During the automated bottleneck scheduling process, an available window is discovered at the GDPAA site. The satellite's beam path is checked for incompatibilities with other beams assigned there, and finding none, it is successfully assigned there. This also relieves the bottleneck at the original station with single-beam antennas. There are also several examples in the output data where GDPAA sites are used to full capacity. This implies that the logic for deconflicting multiple simultaneous beams on a single GDPAA is effectively finding solutions that can be applied in the higher level network scheduling process, contributing to the overall solution for a larger time interval such as a 24 hour period.

V. Conclusion

The initial automated planning and allocation methods discussed in this paper serve two purposes. They establish the feasibility of the phased array concept in terms of the prospect for effectively making use of the increased capacity afforded by the platform. Second, and more specifically, they provide an experimentation base for exploring utilization concepts. With capabilities for allocating beams to GDPAAs and automatically developing beam paths, there are several directions for further research to build on these results. One of the promising features of phased array antennas is the potential to support dynamic requirements occurring in real-time. With electronic control comes the ability to adjust gain on demand, for situations such as a tumbling satellite. When such adjustments are necessary, it may also become important to find alternative allocations. For example, if a GDPAA has several simultaneous contacts, and one of them enters a problem state, then the others may need to be dynamically allocated elsewhere in near real-time, to allow for more subarray modules to be used for the problem contact. Automated technology becomes critical for this kind of dynamic operation. Another real-time need involves changes in the status of T/R modules on the surface of the phased array. These kinds of dynamic situations would be natural areas for future research.

Another area for further investigation involves additional refinement and optimization of the allocation algorithm. Further experiments and statistical analysis of the performance in different scenarios would help to refine the balance in the mitigation strategy for addressing beam path conflicts that are local to a phased array antenna, versus network-level resource conflicts.

References

[1]Stottler, R., "General Scheduling Service using Intelligent Resource Management Techniques," *Infotech@Aerospace 2010 Conference, American Institute of Aeronautics and Astronautics*, Atlanta, GA, 2010.

[2]Stottler, R., Mahan, K., and Jensen, R., "Bottleneck Avoidance Techniques for Automated Satellite Communication Scheduling," *Infotech@Aerospace 2011 Conference, American Institute of Aeronautics and Astronautics*, Reston, VA.