# Deep Space Network Scheduling Using Multi-Objective Optimization With Uncertainty

Mark D. Johnston*
*Jet Propulsion Laboratory/California Institute of Technology, Pasadena, CA, 91109*

**We have developed a novel technique to incorporate uncertainty modeling within an evolutionary algorithm approach to multi-objective scheduling, with the goal of identifying a Pareto frontier (tradeoff curve) that recognizes the likelihood of events that can impact the schedule outcome. Our approach is particularly applicable to the generation of multi-objective optimized *robust* schedules, where objectives are assigned a service level, for example that we require an objective value to be ≥X with Y% confidence. We have demonstrated that such an approach can, for example, minimize scheduling on less reliable resources, based solely on a resource reliability model and not on any ad hoc heuristics. We have also investigated an alternative method of optimizing for robustness, in which we add to the set of objectives a failure risk objective to minimize. We compare the advantages and disadvantages of these two approaches. Future plans for further developing this technology include its application to space-based observatory scheduling problems.**

## I. Introduction

In the context of multi-mission scheduling of expensive shared systems such as communications resources, a critical challenge is that of exploring and managing tradeoffs among missions. For NASA's Deep Space Network (DSN), this will become more acute if the network architecture evolves to incorporate arrays of smaller antennas that can be grouped dynamically and allocated in highly flexible ways. The objective of the work reported here is to develop techniques to explicitly optimize the multiple simultaneous and competing objectives of individual mission users as well as the network system as a whole. For the former, objectives center around maximally satisfying communications needs in terms of link quality, quantity, timing, and other factors. For the latter, objectives are based on minimizing cost and maximizing network availability. Along with the problem of competing multiple objectives, the DSN array would be subject to significant additional sources of uncertainty that complicate planning and scheduling. Chief among these is the sensitivity of Ka-band antennas to atmospheric moisture levels, which implies that weather will impact advance planning in ways that make longer lead-time scheduling more difficult. Other sources of uncertainty include equipment failures and return-to-service times, and unanticipated disruptive spacecraft events.

In the following section (II) we first give an overview of the Deep Space Network (DSN) and its potential evolution to an array-based architecture — the proposed Deep Space Array Network (DSAN). We describe how this evolution would present both opportunities and challenges, and how uncertainty enters to complicate scheduling in ways not present in today's network. We then describe briefly our multi-objective approach to schedule optimization (III), and the evolutionary algorithm solution technique we are using. Next we discuss two approaches to incorporating uncertainty into the solution approach (IV), one based on explicitly modeling probability of failure as an objective to optimize (IV.A), the other based on a stochastic assessment of objective values (IV.B). We include an illustrative sample problem and show how it is solved using each technique. Finally we summarize our conclusions and describe some next steps (V).

## II. Overview of the Deep Space Network (DSN) and Array

The Deep Space Network is NASA's collection of assets for communicating with spacecraft beyond near-earth orbit. It currently comprises dozens of large antennas of diameters 26m, 34m, and 70m, distributed geographically over three complexes spaced sufficiently far apart in longitude to afford full sky coverage (Goldstone, California; Madrid, Spain; and Canberra, Australia). In addition to the antennas, the complexes contain a variety of supporting

* Principal Scientist, Planning and Execution Systems Section, MS 301-260.

electronics and computing facilities. Each complex operates around the clock to handle spacecraft data downlink, command uplink, and other services. All of NASA's deep space missions currently use the DSN, which is operating today at very near to capacity. Some portions of the sky are already substantially oversubscribed (e.g. Mars, where as of early 2008 there are two surface vehicles and three orbiters operating), and this situation is projected to become more severe in the future. Over the next 25 years it is expected that the number of missions will increase by a factor of three, that data rates and volumes will grow by a factor of 100, along with a significant increase in data link difficulty. In addition, plans for human exploration of the Moon, and eventually Mars, would place even greater demands on data rates and on the quality and reliability of communications links.

The most promising growth path for the DSN is a move to a proposed *array* architecture[1]. Antenna arrays are already coming into wide use for radio telescopes — examples include the Very Large Array[2] and the more recent Atacama Large Millimeter Array and Allen Telescope Array[3]. Arrays offer a number of advantages over expanding the current DSN collection of large antennas. First, arrays are made of smaller and much cheaper antennas, so there is a large economic advantage. Secondly, arrays offer the potential for much greater flexibility. Rather than dedicating a single antenna to a single mission for some time period, numerous subsets of an equivalent sensitivity array could be devoted to multiple missions, overlapping in time as the various spacecraft come into view. The number of array antennas could even vary within a single contact, in that fewer antennas would be required to achieve the same sensitivity when a spacecraft is near zenith rather than near the horizon, due to atmospheric signal attenuation. Thirdly, the implicit redundancy of the array elements means the network would be less vulnerable to single points of failure. However, along with the array's advantages come several new challenges:

- scheduling would become much more complex, in that time-varying resource subsets have to be identified and assigned to the missions that need them
- smaller antennas call for higher frequencies which are more affected by atmospheric moisture, with a corresponding greater risk of schedule upset[4]

Most missions need relatively long-term advance schedules to be in place so they can develop command timelines and build and validate their onboard sequences. Thus the uncertainty caused by weather adds a new element to the advance scheduling problem. Statistical models can be constructed of average rain effects, but deviations from the average can be very significant and cause major schedule changes. In addition, there is a large variation in sensitivity caused simply by the apparent elevation of the spacecraft at the antenna, due to the path length through the atmosphere. This has a dependency of the form $1/\sin\varepsilon$ where $\varepsilon$ is the spacecraft elevation.

There as been much effort invested in automating DSN operations and scheduling over the years[5]. An evolutionary algorithm for DSN scheduling has been described by Guillaume et al.[6], who considered a single-objective optimization problem, but with different sets of objectives. Scheduling for an array architecture is a relatively new development, discussed by Cheung[7] who used a genetic algorithm to optimize a single-objective formulation of the problem. Johnston[8,9] applied an evolutionary algorithm to a multi-objective formulation of the problem (described further below), as well as to the current DSN scheduling problem[10].

## III. Multi-Objective Optimization for an Array Architecture

From an optimization perspective, both the DSN and a potential array-based DSAN architecture are characterized by *multiple competing objectives* from the points of view of the various mission users. Each user is trying to obtain a commitment to scheduled contacts that provide their required communications services, based on requirements specific to that mission. These requirements come in many forms[11] and can be based on timing and gaps, coverage duration, specific antennas or antenna types, etc.

### A. Objectives and Constraints for Array Scheduling

The optimization objectives in this problem are most naturally formulated on a user-by-user basis. Each user's objectives can be viewed as quantifying a “degree of satisfaction” metric, where examples of factors that might contribute are provided in Table 1. From an overall system perspective, optimization is driven by satisfying the maximum number of users, as reflected in the satisfaction of their individual objectives.

Constraints in the DSN scheduling problem come from several sources. Mission constraints may be formulated in terms similar to objectives, the main difference being their importance. For example, during a mission-critical event, what might otherwise be a preference for communications coverage may be elevated to the highest level of importance, such that no schedule without coverage will be considered as feasible. System level constraints include those based on overall resource availability, for example, reflecting maintenance schedules and the planned introduction of new assets.

Constraints and objectives can play a complementary role in a practical scheduling problem, which we exploit in the solution approach described in the next section. For example, consider a problem which is overconstrained such that no solution exists. In this situation it is extremely useful to obtain some insight into what constraints must be relaxed, and by how much, in order to assess feasibility.

| Objective | Description |
|---|---|
| contact duration | min and max limits on duration, where a contact is the union of the coverage intervals of overlapping passes |
| contact gap duration | min and max limits on the sizes of any gaps between contacts |
| pass duration | min and max limits on individual pass duration |
| gap duration | min and max limits on the sizes of any gaps between individual passes |
| coverage fraction | fraction of some specified time interval with scheduled contact coverage (e.g. "1" means continuous coverage) |
| coverage level | number of distinct passes simultaneously providing coverage (e.g. "2" would mean simultaneous coverage at two different sites) |
| total gap duration | total gap in coverage over a specified interval |
| pass time shift | how much a pass has shifted in time from some baseline requested time |
| objective out of limit | extent to which an objective value exceeds a specified limit |

**Table 1. Example objectives relevant to contact scheduling problems.**

### B. Multi-Objective Optimization

The DSN scheduling problem is naturally *multi-objective* in that there is no single scalar that characterizes an optimal solution. The traditional approach to problems like this is to construct such a single objective, e.g. by taking some function of the individual user objectives. However, it is obvious that important information is lost when this is done. To avoid this, we have adopted a multi-objective optimization perspective, in which information about each objective is kept separate and is thus available to assess tradeoffs and sensitivity. Among the best current techniques for solving multi-objective optimization problems are *evolutionary algorithms*, in which a population of candidate solutions is developed and evolved[12,13,14].

We define a multi-objective optimization problem to minimize $M$ objectives subject to $K$ constraints:

minimize: $\{f_i(\vec{x})\},\ i = 1 \ldots M$

subject to: $\left(g_j(\vec{x})\right)^{\gamma} \le 0,\ j = 1 \ldots K$

Here $\vec{x}$ represents a vector in decision space of dimension $D$. A solution is called *Pareto optimal* when no improvement can be made to one objective that does not make worse at least one other objective. The set of Pareto optimal solutions is called the *Pareto frontier*. What we seek as a solution to the multi-objective optimization problem is a good approximation to the Pareto frontier. Two important characteristics of a good solution technique are convergence to the Pareto frontier, and diversity so as to sample the frontier as fully as possible.

### C. An Evolutionary Algorithm Solution Approach

We have adopted an evolutionary algorithm approach to the array scheduling problem as detailed in previous papers[8-10]. Here we briefly summarize the method, which is based on the Generalized Differential Evolution 3 algorithm of Kukkonen et al.[15] Among techniques developed to solve multi-objective optimization problems, evolutionary algorithms have become popular for a variety of reasons. They have been shown effective on a wide range of problems and are capable of dealing with objectives that are not mathematically well-behaved (e.g. discontinuous, non-differentiable). By maintaining a population of solutions they are capable of representing the entire Pareto frontier at any stage. They also lend themselves to parallelization, which is an important performance consideration for large problems. Here we concentrate on one particular variant called Generalized Differential Evolution 3, or GDE3[15]. This technique is based on Differential Evolution, a single objective evolutionary algorithm for real-valued decision spaces[16]. GDE3 makes use of concepts pioneered in the algorithm NSGA II[17], including:

- *non-dominated* sorting of the population into ranks, such that members of rank $n$ dominate members of all ranks $>n$, where rank 1 members constitute the non-dominated set, i.e. the current approximation to the Pareto frontier
- *crowding distance* is used as a secondary discriminator on members of the same rank: members in crowded regions of the population are scored lower, so the surviving members after selection have greater diversity. This helps prevent premature convergence of the population to a small portion of the Pareto frontier (see also[18])

- population members are compared with a *domination* or *constraint-domination* relation — the latter allows for domination comparisons even when constraints are violated

GDE3 operates as follows to evolve the population of size *N* from one generation to the next:

1. For each parent member of the population $\vec{x_i}$, select three distinct population members $\vec{x_{r_1}}$, $\vec{x_{r_2}}$, $\vec{x_{r_3}}$, all different and different from the parent

2. Calculate a trial vector $\vec{y_i} = \vec{x_{r_1}} + F \cdot (\vec{x_{r_2}} + \vec{x_{r_3}})$, where F is a scaling factor

3. Modify the trial vector by binary crossover with the parent with probability *CR* The result is compared with the parent as follows:

- in case of infeasible vectors, the trial vector is selected if it weakly dominates the parent vector in constraint violation space, otherwise the parent vector is selected
- in the case of feasible and infeasible vectors, the feasible vector is selected
- if both vectors are feasible, then the trial is selected if it weakly dominates the parent in objective space; if the parent dominates the trial, then the parent is selected; if neither dominates, then both are selected

The selected vectors may constitute a population of size >*N*, in which case the population size is reduced through the non-dominated sorting and crowding distance mechanism of NSGA II. In addition to high performance, it is also worth noting that one of the strengths of GDE3 is its natural treatment of multiple constraints: it is straightforward to change constraints into objectives when investigating overconstrained problems. This is especially valuable when constraints must be relaxed in order to find any feasible solutions. To apply this method to the array scheduling problem we must encode the allocation of a varying subset of antennas over some time period. We start by consider a collection of *U* users (i.e. missions and other users), over some scheduling time period $[T^s, T^e]$. Associated with each user is a set of view periods, each of which is a time interval during which some specific antenna is available for allocation to that mission, or in case of the array, when some number of array antennas at one site may be allocated. We denote the view periods by $[V^s_{up}, V^e_{up}]$, where $u = 1 \ldots U$ ranges over users, and $p = 1 \ldots P_u$ ranges over the set of view periods for each user. For array allocations, the minimum required time-varying antenna profile is given by $A^{req}_{up}(t \in V_{up})$, which may differ from one view period to another. Above the minimum required level, additional array antennas might be allocated, e.g. to improve signal strength in the face of uncertain weather: we denote the maximum additional allocation by $\Delta_{up}$. For single antenna allocations, the profile function is constant $A^{req}_{up}(t) = 1$, and $\Delta_{up} = 0$. For decision variables we selected a mechanism that preserves neighborhoods in general, so that a small perturbation in the value of the decision vector will result in a small change to the scheduled allocation. These are defined as follows (suppressing the *up* subscripts), as illustrated in Figure 1:

1. For each view period, define a triple of real-valued decision variables $\xi_1, \xi_2, \xi_3 \in [0,1]$

2. Calculate the start and end of the allocated portion of the view period as $t^s = V^s + \xi_1 (V^e - V^s)$ and $t^e = t^s + \xi_2 (V^e - t^s)$, respectively

3. Calculate the allocated antenna quantity (for array allocations) as $A(t) = A^{req}(t) + ceil(\xi_3 \Delta)$

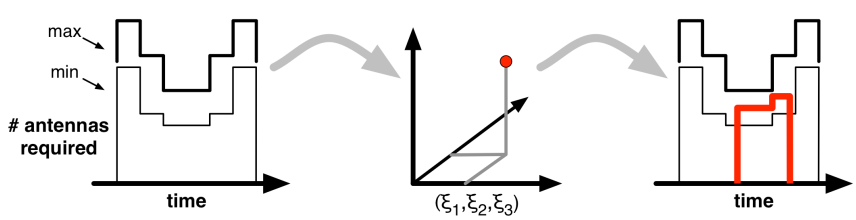


**Figure 1. Decision variables for antenna allocation.** *A real-valued triple is sufficient to specify the antenna allocation profile over each possible viewing interval.*

GDE3 is very effective at exploiting structure in the candidate population to help speed convergence to the Pareto frontier. Figure 2 illustrates an example that compares GDE3 with a trial vector generation method that randomly creates candidates within the decision variable bounding values. This problem represents two users competing for time over several days, with a maintenance outage on one day that implies that neither user can obtain the total requested time over the entire interval. The details are:

- two identical missions, user1 and user2, with periodic communications requirements

- each mission requires a constant allocation of 5 antennas for each link (constraint)
- each mission requires contacts that are at least 3 hours in duration (constraint), with a preferred duration of 12 hours (objective)
- gaps between contacts are limited to 18 hours (constraint)
- the scheduling interval is 4 days in duration
- both missions have the same 12 hour view period each day
- a single site with a fixed number of 10 antennas is available at all times except day 2, when only 5 are available

Were it not for the shortage of antennas on day 2, this problem would have a trivial solution where each mission could maximally achieve all of its objectives. As it is, there is contention on day 2 for the available antenna resources. When this problem is encoded as described above, it has the following characteristics:

- two objectives: the penalty functions for user1 and user2
- one constraint for each user, for minimum pass size
- one system level constraint, to enforce antenna allocation levels to not exceed the available quantity
- 16 decision variables, two for each user/viewperiod combination (antenna allocations are constant, so only two decisions variables per viewperiod are required)

Figure 2 (top) shows the evolution of the non-dominated set after various numbers of generations using the GDE3 algorithm: each plot shows the values of the user1 and user2 objectives for each member of the candidate population, on the $x$ and $y$ axes respectively. The run parameters were: time resolution = 1 hour, population size N=400, F=0.5, CR=0.1. The initial conditions were uniformly random within the bounds of the decision variable limits. Initially (Generation 0, Figure 2(a) top), the population is nearly all constraint violated (red squares), since a random decision variable vector is very likely to exceed either the resources on day 2, or the minimum gap constraint. After 50 generations (Figure 2(b) top) the population has evolved to consist entirely of feasible solutions (blue circles), and shifts to the lower left as increasingly smaller values of the penalty values are discovered. By generation 100 (Figure 2(c) top) the population has nearly evolved to the Pareto frontier for this problem: the non-dominated set is shown as green circles. By generation 150 the convergence is complete: values at the extreme represent solutions where the available 5 antennas are allocated entirely to the other mission. The remaining values along the frontier represent partial allocations to each mission. The gaps in the curve reflect the constraint that solutions with pass durations <3 hours are excluded as not feasible.

By contrast, Figure 2(a-d) bottom illustrates the same problem solved with everything identical except that a random trial vector generator was used instead of GDE3. After 50 generations (Figure 2(b) bottom), the population still shows a handful of candidates with constraint violations, and shows much slower convergence towards the Pareto frontier. After 150 generations (Figure 2(d) bottom) the candidates are still very spread out and far from the frontier located by GDE3.

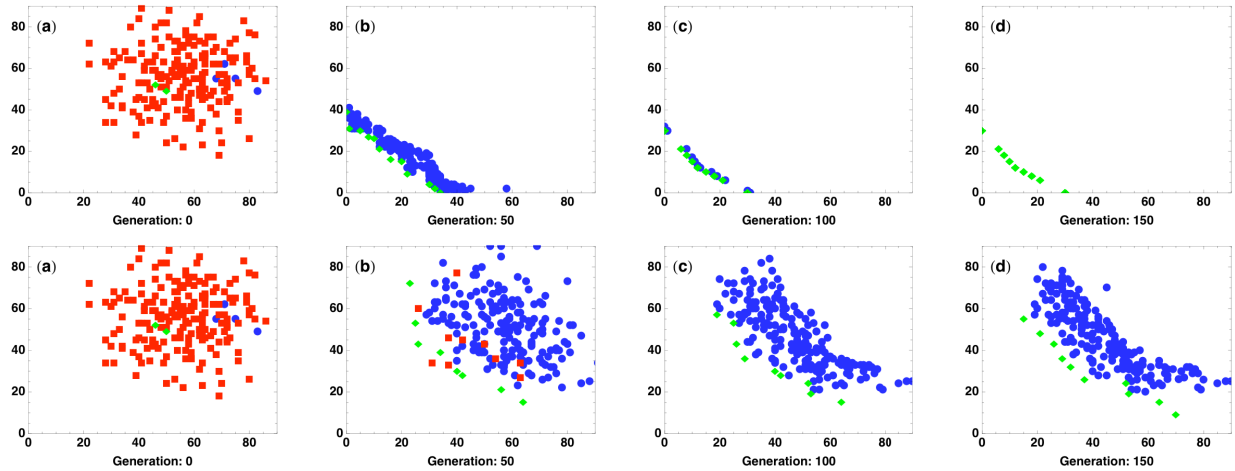


**Figure 2. Comparison of GDE3 and random trial vector generation.**

## IV. Scheduling With Uncertainty

In this section we describe two approaches to uncertainty modeling in the multi-objective array scheduling problem: the first is based on an explicit objective to represent the probability (or an indicator) of failure, to include in the set of multiple objectives that are to be optimized. The second exploits a statistical evaluation of objectives for

a given schedule. We show how each of the two techniques can be used to solve the same simple example problem, thus illustrating their advantages and disadvantages.

### A. Choosing Objectives to Improve Schedule Robustness

We first consider the explicit inclusion of an objective to reflect differing degrees of schedule risk. Such an objective can be optimized along with all the others in the problem, requiring no algorithmic changes. With this approach, the risk tradeoff curve can be constructed simply from the Pareto frontier that is built up as part of running the optimization algorithm. Note that while there is no reason to limit to a single risk-like objective, it may be more difficult to comprehend the results if there are many, so there is some advantage to attempting to combine risk factors in to a single objective. The following illustrative example shows how this works in the context of a simplified array example.

We consider a simple example problem based on uncertainty due to equipment reliability. Consider a two-antenna scheduling problem with the following choices: one antenna, A1, has a relatively short mean time to failure (MTTF) and to repair (MTTR), but if a failure occurs during a scheduled track, the entire track is lost. A second antenna, A2, has a much longer MTTF and MTTR, but the availability ratio MTTF/(MTTF+MTTR) is the *same* for both antennas. Based only on availability, the two antennas are equal in desirability. Assuming that failures are random, at any given time either antenna could be scheduled for use and would be expected to be available.

From a single user's perspective, an overall probability of schedule failure can be expressed as:

$$f(\vec{x}) = 1 - \prod_i p_i$$

where $p_i$ is the probability that the $i^{th}$ contact is successful, and the product is over all contacts in the schedule. Since we are keeping this quantity as a separate objective, it is not essential to normalize the probability or even to scale for commensurability with other objectives. If we assume an exponential failure distribution, then $p_i = \exp^{-d_i / MTTF_i}$, where $d_i$ and $MTTF_i$ are the duration and MTTF of the $i^{th}$ contact, respectively, and thus

$$f(\vec{x}) = 1 - \exp\left(-\sum_i \frac{d_i}{MTTF_i}\right)$$

Figure 3 illustrates an example problem based on this scenario, where there are two objectives: the *x*-axis corresponds to scheduled time meeting duration and gap preferences and constraints, while the *y*-axis corresponds to the schedule failure probability $f(\vec{x})$.

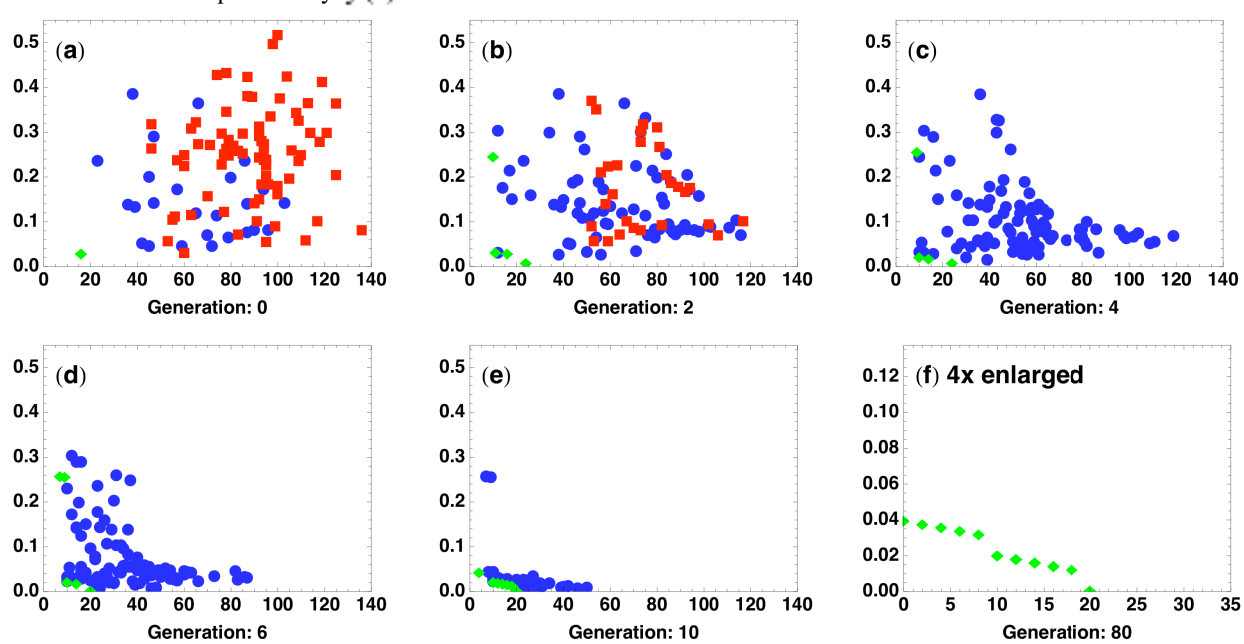


**Figure 3. Convergence to a Pareto frontier with failure risk as one of the objectives.**

In this example, once the population eliminates trial solutions with constraint violations (Figure 3(c)) it consists of passes scheduled on both A1 and A2. As the population evolves further, it settles into a Pareto frontier that reflects the tradeoff between contact duration and risk of failure, and places all scheduled passes onto A2. (Figure 3(f)). On this frontier, the highest user preference value ($x$=0) corresponds to the greatest failure probability: this is a consequence of the fact that higher user preference values (increasing $x$) correspond to longer passes, which are more likely to fail based on our exponential failure probability.

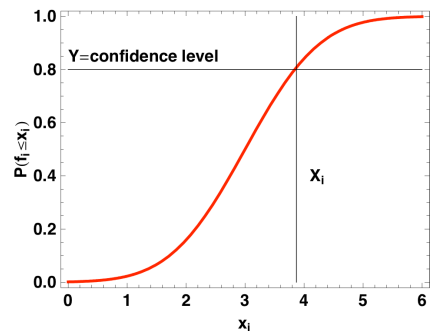


**Figure 4. Selecting a confidence level for sampling objective values.**

### B. Model-Based Uncertainty

An alternate approach to considering uncertainty that can impact the schedule is to focus on the evaluation of the objective functions, $f_i$. In the basic form of the algorithm, a candidate schedule represented by a decision variable vector $\vec{x}$ is evaluated to compute a vector of objective function values $\vec{f}(\vec{x})$. However, this "point" evaluation may depend on one or more parameters $\vec{\phi}$, e.g. through $\vec{f}(\vec{x};\vec{\phi})$, where $\vec{\phi}$ has non-deterministic values. Usually a mean or characteristic value $\hat{\vec{\phi}}$ is selected, but a better approach is to sample of the probability distribution of $\vec{\phi}$ and choose for $\vec{f}(\vec{x};\vec{\phi})$ a value reflecting a certain confidence level $Y_i$. Thus, for each objective function $f_i(\vec{x};\vec{\phi})$, we seek a value $X_i$ such that the probability $P[f_i(\vec{x};\vec{\phi}) \le X_i] = Y_i$, where we keep $\vec{x}$ fixed and vary $\vec{\phi}$ (see Figure 4). This essentially requires a Monte Carlo evaluation of each candidate schedule in the population — a potentially costly operation that depends on the details of how the parameters $\vec{\phi}$ affect the schedule evaluation.

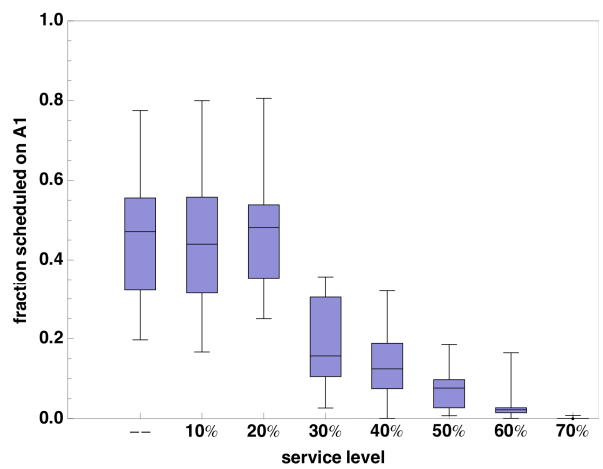


**Figure 5. Migration of scheduled contacts to a more reliable resource as service level increases.**

This technique can be illustrated on the same problem used above to show how objectives can be added to model risk. Here we do not add any additional objectives, but evaluate the (in this case, single) objective value using a Monte Carlo evaluation to simulate the failure of passes scheduled on antenna A1 or A2. The results are plotted in Figure 5 where the fraction of time scheduled on antenna A1 is shown as a function of service level $Y_i$. For low values of the service level, solutions are roughly equally distributed on the two antennas. As the service level increases, to around 25%, the scheduled passes show a pronounced shift to A2, and at higher service levels (>60%) they are essentially all scheduled only on A2.

An interesting feature of this approach is that it does not rely on ad hoc rules or heuristics. The robustness of the schedule follows directly from the selection of candidates that have better objective function values in the face of random variations that can influence them. This is in contrast to the alternative approach described above, which requires the explicit identification of factors that can lead to schedule breakage, and then a way to calculate the risk value as an objective.

One drawback of this approach is that is does not take into account any scheduling reaction that could be taken once a failure occurs. Since we are holding $\vec{x}$ fixed, the schedule remains unchanged when we evaluate the objective functions, even though corrective action could be taken during execution to lessen the severity of a failure. In this sense the method is overly pessimistic. On the other hand, a schedule that scores well using this technique is robust to variations in $\vec{\phi}$ that might otherwise cause a schedule to fail (provided the confidence level Y is high enough).

## V. Conclusion

We have described a multi-objective formulation of the proposed Deep Space Array Network communications scheduling problem, and two methods for incorporating sources of uncertainty.

The first method explicitly incorporates probability of failure into the multi-objective formulation by defining one of more explicit objectives that quantify this value or an indicator of it. The advantage of this approach is that it allows for probability of failure to directly and visibly trade off against other objectives. The main drawback is that explicit probability of failure can be difficult to formulate and to calculate, and increases the dimensionality of the objective space.

The second method makes use of a stochastic assessment of each member of the schedule population, by evaluating the multiple objective functions at a specified confidence level. The advantage of this method is that it does not rely on ad hoc heuristics or on a complete formulation of probable failure, but rather uses the distribution of objective function values to drive towards the Pareto frontier. The drawback is that this method can be computationally very costly, requiring a large number of Monte Carlo evaluations of the population schedules.

We plan to apply both techniques to larger scale problems, and to quantify the performance differences in several problems of practical interest and scale. We are also evaluating this technique on the problem of multi-objective scheduling of scientific observatories, in which the balancing of scientific objectives with operational requirements naturally suggests a multi-objective formulation.

## Acknowledgments

The research described in this paper was carried out at the Jet Propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration.

## References

1 Bagri, D. S. and J. Statman. (2004). “Preliminary Concept of Operations for the Deep Space Array-Based Network.” The Interplanetary Network Progress Report, 42-157, from http://tmo.jpl.nasa.gov/progress_report/42-157.

2 Thompson, A. R., B. G. Clark, C. M. Wade and P. J. Napier (1980). “The Very Large Array.” *The Astrophysical Journal Supplement Series* **44**: 151.

3 Webber, J. C., M. W. Pospieszalski, N. R. A. Obs and V. A. Charlottesville (2002). “Microwave instrumentation for radio astronomy.” *Microwave Theory and Techniques, IEEE Transactions on* **50**(3): 986-995.

[4] Larson, W. J. and J. R. Wertz, Eds. (1999). *Space Mission Analysis and Design*. Dordrecht, Kluwer Academic Publishers.

[5] Clement, B. J. and M. D. Johnston (2005). “The Deep Space Network Scheduling Problem”. IAAI 2005, Pittsburgh, PA, AAAI Press.

[6] Guillaume, A., S. Lee, Y. F. Wang, H. Zheng, R. Hovden, S. Chau, Y. W. Tung and R. J. Terrile (2007). "Deep Space Network Scheduling Using Evolutionary Computational Methods." *Aerospace Conference, 2007 IEEE*: 1-6.

[7] Cheung, K.-M., C. H. Lee and J. Ho (2005). “Problem Formulation for Optimal Array Modeling and Planning.” IEEEAC 438.

[8] Johnston, M. D. (2006). Multi-Objective Scheduling for NASA's Deep Space Network Array. International Workshop on Planning and Scheduling for Space (IWPSS-06). Baltimore, MD, Space Telescope Science Institute.

[9] Johnston, M. D. (2008). Multi-Objective Optimization With Uncertain Outcomes Applied to Space Network Scheduling. iSAIRAS 2008. Hollywood, CA.

[10] Johnston, M. D. (2008). "An Evolutionary Algorithm Approach to Multi-Objective Scheduling of Space Network Communications." *International Journal of Intelligent Automation and Soft Computing*: in press.

[11] Clement, B. J., M. D. Johnston, D. Tran and S. R. Schaffer (2008). Experience with a Constraint and Preference Language for DSN Communications Scheduling. iSAIRAS 2008. Los Angeles, CA.

[12] Bagchi, T. P. (1999). *Multiobjective Scheduling by Genetic Algorithms*. Boston, Kluwer.

[13] Deb, K. (2001). *Multi-Objective Optimization Using Evolutionary Algorithms*. New York, John Wiley & Sons.

[14] Abraham, A., L. Jain and R. Goldberg (2005). *Evolutionary Multiobjective Optimization*, Springer.

[15] Kukkonen, S. and J. Lampinen (2005). GDE3: The Third Evolution Step of Generalized Differential Evolution. The 2005 Congress on Evolutionary Computation.

[16] Price, K., R. Storn and J. Lampinen (2005). *Differential Evolution: A Practical Approach to Global Optimization*. Berlin, Springer-Verlag.

[17] Deb, K., A. Pratap, S. Agrawal and T. Meyarivan (2002). "A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II." *IEEE Transactions on Evolutionary Computation* **6**(2): 182-197.

[18] Kukkonen, S. and K. Deb (2006). Improved Pruning of Non-Dominated Solutions Based on Crowding Distance for Bi-Objective Optimization Problems. Proceedings of the 2006 Congress on Evolutionary Computation (CEC 2006). Vancouver, BC, Canada.