The allocation of a shared resource within an organization

John O. Ledyard [a], Charles Noussair [b,*], David Porter [a]

[a] *California Institute of Technology, Pasadena, CA 91125, USA*
[b] *Department of Economics, Krannert School of Management, Purdue University, West Lafayette, IN 47907, USA*

Revised 1 June 1996

Abstract

Many resources such as supercomputers, legal advisors, and university classrooms are shared by many members of an organization. When the supply of shared resources is limited, conflict usually results between contending demanders. If these conflicts can be adequately resolved, then value is created for the organization. In this paper we use the methodology of applied mechanism design to examine alternative processes for the resolution of such conflicts for a particular class of scheduling problems. We construct a laboratory environment, within which we evaluate the outcomes of various allocation mechanisms. In particular, we are able to measure efficiency, the value attained by the resulting allocations as a percentage of the maximum possible value. Our choice of environment and parameters is guided by a specific application, the allocation of time on NASA's Deep Space Network, but the results also provide insights relevant to other scheduling and allocation applications. We find (1) experienced user committees using decision support algorithms produce reasonably efficient allocations in lower conflict situations but perform badly when there is a high level of conflict between demanders, (2) there is a mechanism, called the Adaptive User Selection Mechanism (AUSM), which charges users for time and yields high efficiencies in high conflict situations but, because of the prices paid, in which the net surplus available to the users is less than that resulting from the inefficient user committee (a reason why users may not appreciate 'market solutions' to organization problems) and (3) there is a modification of AUSM in which

* Corresponding author. Tel.: (+1) 317-494-4416; fax: (+1) 317-494-9658; e-mail: noussair@mgmt.purdue.edu.

tokens, or internal money, replaces real money, which results in highly efficient allocations without extracting any of the users' surplus. Although the distribution of surplus is still an issue, the significant increase in efficiency provides users with a strong incentive to replace inefficient user committees with the more efficient AUSM.

JEL classification: C92

Keywords: Scheduling; Auctions; Mechanism design

1. Introduction

Typically, shared resources are centrally financed and managed by an organization and used in various ways and intensities by diverse members of that organization. Examples of such resources include supercomputers, observatories, motor pools, legal advisors, lab facilities, etc.... [1] The methods and processes by which shared resources are allocated often evolve in a common pattern. We offer our stylized version here. In the beginning, the resource is proposed by potential users, designed by engineers, and financed and managed by bureaucrats. Many of these actors have a political stake in the performance and utility of the resources: thus, they want to keep control. The initial management structure is often a combination of user committees and bureaucratic responsibility. As long as desired use is not much larger than capacity, committee meetings and bureaucratic negotiation can usually overcome what little conflicts may arise in determining who gets what use of the shared facility. However, increases in demand or decreases in usable capacity can easily lead to congestion in the system and to contentious *non-negotiable* conflicts among users. If it is not possible, or is costly to expand the resources, the next step usually involves asking users and bureaucrats to streamline their decision processes. Decision Support Teams are often summoned to help facilitate the process. They provide computer programs (sometimes involving various voting mechanisms) which are intended to speed up the existing processes. Using valuation and resource requirement information provided by the users, the algorithms may heuristically solve very complex optimization problems in an attempt to compute good allocations. Unfortunately, these support programs tend to ignore the fact that users have incentives to bias the information they provide to the algorithm in order to achieve more favorable allocations. The new support systems work faster but congestion remains as much of a problem as before. Value is foregone.

---

[1] Another similar class of situations includes resources managed by the government for the private sector such as the electromagnetic spectrum managed by the FCC and airports managed by local governments. Virtually everything in this paper applies to these publicly managed assets. See for example Bykowsky et al. (1995) for related work on the FCC spectrum problem.

At this point, in desperation, economists are sometimes called for advice. Their usual response is 'auction it off' or 'price it at marginal cost', forgetting at least two facts that lead users to ignore the economists' advice. First, users who have been getting something for nothing will now have to pay to obtain an allocation. Thus, even if allocative efficiency improves and value increases, the users can be worse off because the bureaucracy may capture more in rents than the increase in the value of the allocations. So the users will use whatever political clout they have to block these 'economic solutions'. Second, even if the political resistance can be overcome, for example by the use of grandfathered allocations, [2] these scheduling problems can involve significant non-convexities which cause prices to be unable to function as coordinating devices. Thus even if the economists' advice is taken seriously, the results may be less than satisfactory and yield allocations no better for users than current processes are producing (see Banks et al., 1989).

In this paper we explore a different approach to finding solutions for congestion in shared resources: *applied mechanism design.* (See Ledyard (1993) for a summary of the approach.) We argue that new processes can be designed which improve upon congested user committees and other regulatory administrative procedures. The constraints on the design process, in addition to those familiar to economists such as incentive compatibility and budget balancing, include limits on the computational capability of any algorithm used and political constraints which must be satisfied if the mechanism is to be acceptable to the parties involved as a replacement to current procedures. This approach to the design of new mechanisms is ideally pursued by the combination of theorists, who can analyze some of the properties of an allocation mechanism, and experimental economists, who use the laboratory to test the performance and refine the procedures for demonstration and actual use. When theoretical knowledge is not sufficient to describe the behavior of individuals in a mechanism in the environment under study, the economist can still create new mechanisms, the choice guided by theoretical intuition and previous empirical evidence from other environments. The laboratory can then be used to test and improve the performance of the new mechanism in an environment based on the field application of interest. This approach to design is similar in spirit to the aircraft designer who uses wind tunnels to create new airplanes.

In this paper, we report the results from a set of experiments designed to provide a testbed and a demonstration of proof of concept. We based our experiments on a systematic study of a specific shared resource problem, the allocation of tracking time of NASA's worldwide Deep Space Network (DSN) of antennas. While one may or may not be interested in the DSN allocation problem,

---

[2] An interesting example of this can be found in Riker and Sened (1991) based on work of Grether et al. (1981).

the approach we take and the results of our performance tests have a wide range of application to scheduling and other allocation problems.

Our approach is straightforward to replicate. We first try to extract those elements of the situation that are most relevant to the mechanism design problem. For the DSN problem, we report this aspect of our research in Section 2 and Section 3 below. Using these historical and theoretical analyses, we then create a class of environments, our wind tunnel parameters, within which to test various organization designs. Next, we try to construct an experimentally testable version of the current process as a benchmark against which to test new designs. Then we create, being guided by relevant theory and experimental evidence, new mechanisms which we think will solve the allocation problem. Finally, we test the mechanisms in the environments. At this point we can eliminate designs with obviously poor performance and can fine tune the procedures (including stopping rules, information structures, etc.) for the designs that seem promising. The results of this experimental analysis are contained in Section 6. We conclude with a summary of lessons learned.

We turn now to the details.

2. A little background

The Deep Space Network is a worldwide network of antennas managed by the Jet Propulsion Laboratory (JPL) for NASA. The network is used to provide uplink and downlink communication with spacecraft in deep space. There are currently ten flight projects in deep space using the network, four ground based radio/radar astronomy projects and maintenance activities, three planned missions, and five pre-planning stage missions which will require use of the antennas. The antennas are located at three different locations, allowing for continuous tracking of a spacecraft, and there are three types of antennas (34 meter High Efficiency, 34 meter beam waveguide, and 70 meter) at each location. Each antenna may be directed to at most one user at a time. Demand varies greatly among users and across their lifetimes. For example, the Voyager and Pioneer missions are over 15 years old, have completed their prime missions, and are now in an extended mission mode at the outer reaches of our solar system; on the other hand, the Galileo mission to Jupiter and Ulysses solar polar mission are still en route to their prime targets. In addition, demands are interrelated since the goal of the organization is overall mission success. Therefore, scheduling in this environment is a very dynamic process.

Scheduling of the antenna time has been facilitated by the use of a decision support system and a series of user committees. There are currently the equivalent of 15 full time employees involved in constructing and updating time schedules, which tentatively assign each antenna beginning ten years in advance. For two years in advance, detailed minute by minute schedules are provided which can be

edited until the date scheduled. Heuristic look ahead algorithms help with the provision of initial schedules and aid long-term planning. However, despite the amount of resources used in the scheduling process, there remains widespread contention for resources and significant amounts of committee time used to resolve schedule conflicts. Recently, some effort has been made to commit to resource 'allocations' early so that projects can plan well in advance. However, there is still an oversubscription of resources and changes are commonplace.

3. A theoretical framework

The DSN is an example of a *shared resource*. The abstract structure of the allocation problem associated with shared resources is straightforward. There is a set, X, of resources to be allocated. Examples include time on an antenna, a computer memory, volume in the bay of a space shuttle, time with a consultant, the draw of power from a shared battery, or broadcasting over a particular spectrum band in a specific geographic area. There are a set of potential users of these resources, $I = \{1, \ldots, n\}$. A feasible allocation assigns a subset $X_i \subseteq X$ to each i so that the collection of sets $X_0, X_1, \ldots X_n$ is a partition [3] of X. In some cases there are natural obvious partitions, as for example with rooms in a dormitory, and in other cases there is not, as in time on an antenna. Users get utility or profits from each subset which we denote $U^i(X_i)$. [4] We will assume throughout that a user's total utility is $U^i(X_i) - y_i$ where y_i is the money paid to receive the use of X_i. An efficient feasible allocation $a = (X_1, \ldots, X_n)$ is one such that there is no other feasible allocation $d' = (X_1', \ldots, X_n')$ such that $\sum_{i=1}^{n} U^i(X_i') > \sum_{i=1}^{n} U^i(X_i)$. Thus, an efficient allocation solves

$$\max_{X_1, \ldots, X_N} \sum_i U^i(X_i) \tag{1}$$

subject to

$$\bigcup_{i=1}^{n} X_i \subseteq X \tag{2}$$

and

$$X_i \cap X_j = \varnothing \, \forall \, i, j \in I. \tag{3}$$

---

[3] $\bigcup_{i=0}^{N} X_i = X$ and $X_i \cap X_j = \varnothing$ for all $i, j \in I$. X_0 is the subset of unassigned resources.

[4] We think of $U^i(X_i)$ in the DSN problem as the scientific return to user i from receiving a subset of the resources. We suppose that the organization, JPL, is interested in maximizing the overall scientific return from the resources. Since each user wants to maximize his own return, he will generally want to acquire more resources than he would receive at the organization's optimum. We are assuming the U^i does not include any private, non-scientific benefits to user i, a reasonable assumption for the DSN example.

If the $U^i(\cdot)$ were known, this would be a complex non-linear computation which, in principle, could be solved or approximated. One might be tempted to simply turn it over to the operations research department. The only difficulties would be computational complexities. However, in many applications the users want to maximize their own interests and not those of the group. Therefore, getting accurate information on U^i becomes a problem. Standard bureaucratic processes ask for data on U^i but the distortions created by incentive compatibility constraints may actually lead the processes to choose allocations with $\sum_i U^i(x_i)$ significantly less than the maximum possible.

Setting up markets is an alternative often mentioned. One way to do this would be to organize a market for each $x \in X$, with a price $p(x) \in R$, so that i would pay $\int_{X_i} p(x)\mathrm{d}x$ for the set X_i. For problems like ours, if a market equilibrium exists, the prices would support the solution. That is, for all i and all X'_i, $U^i(X_i^\circ) - \int_{X_i^\circ} p(x)\mathrm{d}x \geq U^i(X'_i) - \int_{X'_i} p(x)\mathrm{d}x$ where X_i° is the optimal allocation.

However, if the demands for the items are highly complementary or if serious non-convexities exist, then a simple set of markets (one for each resource) is unlikely to solve the problem (see Koopmans and Beckman, 1957). In order to discover and support optimal allocations, the price adjustment process requires information about the complementarity of demands and about the extent of non-convexities (see Calsamiglia, 1977; Hurwicz, 1994; Jordan, 1987; Saari and Simon, 1978). In extreme cases one might require price information for all subsets of X, rather than only of x (see Papai, 1994).

When bureaucratic processes and simple markets will not work very well, some success has been realized using a special type of ascending bid auction. These auction mechanisms are relatively easy to understand and achieve fairly high efficiencies. They are called AUSM (Adaptive User Selection Mechanisms). The basic components of AUSM include: (i) a bid, (A^{ik}, b^i), where $A^k \subseteq X$ and $b \in R$, read as i bids b^i to receive A^{ik}, and (ii) a provisional allocation, $(X_0, X_1, \ldots, X_n)$, where X_i is composed of provisionally accepted bids and $X_i = \bigcup_k A^{ik}$, the union of all x, which i receives in the provisional allocation. The ascending bid nature of the mechanism comes from the rule which determines whether any new bid is to be included in the provisional allocation and which bids are then to be removed. This revision rule is simple: (A^{i*}, b^{i*}) is accepted if $b^{i*} > \sum_{\ell k \in z} b^{\ell k}$ where $z = \{\ell k \mid A^{\ell k} \cap A^{i*} \neq \varnothing\}$. That is i must agree to pay at least as much as those who will be displaced if i's bid is accepted. With this rule, the sum of the accepted bids $\sum_{\ell k} b^{\ell k}$ is always increasing. The continuous version of AUSM allows bids to be submitted at any time with a rule that specifies when bidding stops. This form of the process is computationally simple but it does leave open the possibility that users who only want small subsets may have difficulty displacing users who have successfully bid for a large subset. This problem has been addressed by adding a standby queue, described in Banks et al. (1989) and below in Section 4.2. The queue appears to work reasonably well but it is possible that another design would be better. That remains an open question.

To summarize, the generic problem facing the mechanism designer for a given environment X is to find a process which produces allocations (partitions) with high values of $\sum_i U^i(x_i)$. For our DSN problem, X is a collection of heterogeneous items called *slots* numbered $j = 1, \ldots, J$ which can be occupied by one and only one user. This is a one to many assignment problem. The relevant optimization problem for DSN is described in Section 4.3. We now turn to that specific example of a shared resource problem.

4. Our laboratory environment

4.1. Supply

The commodities to be allocated are called *slots*. Each slot is completely specified by a resource and a time. Let S denote the set of slots and $s_{mt} \in S$ denote the slot on resource m, at time t. There are M resources and T times, where $m = (1, \ldots, M)$ and $t = (1, \ldots, T)$. The slots can be represented by an $M \times T$ matrix as in Fig. 1. Let P_j denote package j, where $P_j \subset S$ and let $j = 1, \ldots, J$ where $J = 2^{MT} - 1$. Any collection of slots can be considered a package. In the experimental testbed we developed, $M = 2$ and $T = 10$ so that there are 20 slots to be allocated and $2^{20} - 1$ possible packages. This is shown in Fig. 1.

4.2. Demand

All subjects in the experiments are demanders. There are I demanders, denoted by $i = (1, \ldots, I)$, each of whom has a private valuation for each package. Let V_{ij} denote demander i's valuation for package j. The valuation indicates the monetary value demander i will receive if i obtains all of the slots in the package. V_{ij} is given in terms of an experimental currency, which is called 'francs'. At the end of each experimental session, subjects are paid an amount of U.S. dollars proportional to the total amount of francs they earn by obtaining packages of slots. In this

Resources	Time									
	1	2	3	4	5	6	7	8	9	10
a										
b										

Fig. 1. Commodity to be allocated (20 slots).

way, valuations are induced, so that subjects have a monetary incentive to maximize the total number of francs they earn. [5]

In choosing a set of demanders' types, or utility functions, we tried to capture the general qualities of the requirements of the DSN users, while keeping the types simple and the number of types small. We decided on the following four generic types of demand structures.

1. *Contiguous.* A contiguous demander values packages that are composed of slots at adjacent times on the same resource. Often, users of the DSN require long periods of continuous coverage during important phases of their mission; for example, when mapping a planet or repairing an antenna. In the experiment, contiguous demanders must obtain the same resource for *at least two times* in a row to receive any value. Thus, values for contiguous packages are superadditive.

2. *Periodic.* A periodic demander values packages that are composed of times separated by a fixed interval on the same resource. Some users of the DSN require contact with their spacecraft at regular time intervals. For example, Pioneer 10 must be contacted at least once every 36 hours to readjust its communications equipment in order to guarantee future contact. In our experimental design, periodic demanders need to use slots on the same resource spaced at five time slots apart or otherwise their value is zero. This is an example of nonconvexity in the demand structure.

3. *Array.* An array demander values packages that are composed of slots at the same time on multiple resources. Often, it may be desirable to point two antennae in the same direction in order to increase the data return rate. An example of this occurs during an encounter when all instruments want to be sending data using the 70 meter antenna and the 34 meter antennas at the same location simultaneously allow for a higher data transmission rate with lower error (data loss) rates. In our design, array demanders require two resources at the same time.

4. *Maintenance.* A maintenance demander values as many single slot packages on resources as he can acquire. This is a representation of the preferences of the DSN maintenance staff who are largely indifferent about the time the antenna is shut down for maintenance.

One feature of the information structure in the DSN environment is the common knowledge of the times of day at each location which can be of use to demanders (all users know when each project is on the same side of the earth as a particular antenna complex). These times are referred to as a spacecraft *viewperiod.* In the experiments, each demander was assigned one of two possible

---

[5] See Smith (1976) for more on the theory of induced valuation.

viewperiods, and the number of demanders with each viewperiod was known to all subjects. Subjects possessing viewperiods 1 and 2 had a zero valuation for any slot in times (columns) 1, 2, 6, or 7 (for viewperiod 1) or in times 4, 5, 9, and 10 (for viewperiod 2) respectively.

In each of our experimental sessions, there were six demanders. Demanders did not know the valuations of any other demanders, but they knew that there were a total of three subjects possessing each viewperiod. They also knew the set of possible types, but not the exact valuations which demanders of the various types would have. Each demander was assigned an identification number, one of the four types discussed above and one of the two possible viewperiods. Fig. 2 relates the identification numbers, the types, and the viewperiods.

4.3. Schedule

A *schedule* is an assignment of slots to demanders. A schedule is *feasible* if it satisfies

$$\sum_{i \in I} \sum_{j \in C_{mt}} X_{ij} \le 1; \forall m,t \tag{4}$$

and

$$X_{ij} \in \{0, 1\} \tag{5}$$

where $C_{mt} \subset J$ is the set of packages that contain slot s_{mt} and $X_{ij} = 1$ if demander i receives package j and zero otherwise. A slot can be included in at most one package that is allocated. Each package can be allocated to at most one demander. Also, a slot cannot be subdivided but must be assigned in its entirety. A schedule is efficient if it solves

$$\max \sum_{i \in I} \sum_{j \in J} V_{ij} X_{ij} \tag{6}$$

	User #					
	1	2	3	4	5	6
Type	C	C	P	P	A	M
View period	1	1	2	2	1	2

Fig. 2. Users (6).

C = contiguous, P = periodic, A = array, M = maintenance. Viewperiod 1 ⟹ usable slots ∈ columns 3,4,5,8,9,10; Viewperiod 2 ⟹ usable slots ∈ columns 1,2,3,6,7,8.

subject to Eqs. (4) and (5).[6] An efficient schedule achieves the highest possible total value to demanders of any feasible schedule. Define the *efficiency* of an allocation X^0 as

$$E = \frac{\sum_{i \in I} \sum_{j \in J} V_{ij} X_{ij}^0}{\overline{V}} * 100 \tag{7}$$

where E is the efficiency and $\overline{V}$ is the optimal value.

4.4. Parameters

Although the non-existence of market equilibrium prices for these slot allocation problems is of interest and important, that aspect has been studied in, for example, Banks et al. (1989) and Bykowsky et al. (1995). In the research reported in this paper, we were interested in finding an acceptable mechanism which was much better than the existing process. We therefore focused on environments which were reasonably well-behaved.

Our experiments were conducted using two sets of parameters (valuations) which differed in their level of 'contention'. The contention level gives us an indication as to how much conflict between users is present in the system. Our measure of contention (following Olson and Porter (1994)) is

$$\omega = \frac{\sum_j \sum_{s \in P_j} p_s}{\overline{V}} \tag{8}$$

where $\overline{V}$ is the value of the objective function evaluated at the solution to the maximization problem and p_s is the equilibrium price of slot s, as defined in Section 3. The levels of contention are 0.55 and 0.25 for our high and low contention conditions respectively.

It can be verified from inspection of Tables 1–4, which contain the valuations and the prices for all packages and slots for both the High and Low contention treatments that the prices are indeed competitive equilibrium prices for the parameters used in the experiment.

5. Allocation (scheduling) mechanisms

In this section we describe the general features of the three mechanisms whose performance we test in the environments of Section 3. The first mechanism we test

---

[6] This is a form of the problem known in integer programming as the knapsack problem. The idea is to choose items to take in a knapsack when one goes on a trip. Each object has value, but it also has weight. The problem is to maximize the total value of the items taken subject to the condition that the total weight is less than or equal to some constant. This type of problem is NP-complete. The efficient allocation problem is more complicated, with one constraint per slot and a number of packages (objects which could be put in the knapsack) that equals $2^{MT} - 1$.

Table 1
Demanders' valuations: High contention condition [a]

i	$s \in P_j$				V_{ij}	p_j
▶1	3a	4a	5a		2450	2900
1	3b	4b	5b		1450	1450
*1	8a	9a	10a		3150	2000
*1	8b	9b	10b		2150	1350
*2	4a	5a			3200	1800
*2	4b	5b			2000	1200
2	8a	9a	10a		2000	2000
2	8b	9b	10b		1000	1350
3	3a	8a			1600	1600
3	1b	6b			1600	1600
*3	2b	7b			3200	1250
3	3b	8b			1600	1600
*4	2a	7a			3000	2000
4	3a	8a			1500	1600
*4	1b	6b			1750	1600
4	2b	7b			1250	1250
4	3b	8b			750	1600
*5	3a	3b			3000	1350
5	4a	4b	5a	5b	3000	3000
5	8a	8b			1500	1850
5	9a	9b	10a	10b	1500	1500
*6	1a				750	0
6	2a				1000	1000
6	3a				500	1100
*6	6a				750	0
6	7a				1000	1000
6	8a				500	500
6	1b				750	850
6	2b				500	750
6	3b				250	250
6	6b				750	750
6	7b				500	500
6	8b				250	1350

[a] The asterisks mark the packages allocated at the optimum. The competitive equilibrium price of package j is given in the column labelled p_j. p_j is the sum of the competitive equilibrium prices of the slots comprising the package.

is designed to capture some of the basic features of the current process used by JPL in the allocation of the DSN. It combines a committee process, a decision support algorithm, and a bureaucratic appeals process. While we would not expect this mechanism to perform very well, we evaluate it to provide a benchmark against which to judge the performance of other mechanisms. In particular, since the research task was to find a better way to allocate shared resources like the DSN, we need to demonstrate that the new mechanisms proposed are indeed potentially better than the existing one.

Table 2
Demanders' valuations: Low contention condition[a]

i	$s \in P_j$				V_{ij}	p_j
► *1	9a	10a			4000	1000
1	9b	10b			850	850
2	9a	10a			1000	1000
*2	9b	10b			4000	850
*3	1a	6a			3250	1600
3	2a	7a			1000	1000
*3	2b	7b			2250	0
*4	2a	7a			3250	1000
4	3a	8a			1000	1000
*4	1b	6b			2750	1600
*5	4a	5a	4b	5b	3000	0
5	9a	10a	9b	10b	1000	1850
6	1a				800	800
*6	3a				1200	1000
6	6a				800	800
*6	8a				1000	0
6	1b				800	800
*6	3b				1000	0
6	6b				800	800
*6	8b				1000	0

[a] The asterisks mark the packages allocated at the optimum. The p_j indicate the competitive equilibrium prices of the packages.

Table 3
Competitive equilibrium prices for all slots: High contention

Resources	Time									
	1	2	3	4	5	6	7	8	9	10
a	0	1000	1100	1800	0	0	1000	500	1500	0
b	850	750	250	1200	0	750	500	1350	0	0

We test two alternatives to the current process. First, we auction off the various slots using AUSM, which was originally designed for a different allocation problem by Banks et al. (1989). It is an ascending bid auction which allows bids

Table 4
Competitive equilibrium prices for all slots: Low contention

Resources	Time									
	1	2	3	4	5	6	7	8	9	10
a	800	1000	1000	0	0	800	0	0	1000	0
b	800	0	0	0	0	800	0	0	850	0

for packages of slots as well as for individual slots. It also has a feature called a standby queue which is designed to help bidders overcome the 'threshold' problem common to these types of combinatoric problems. This mechanism is described in more detail below.

Although the AUSM with queue generally achieves high efficiencies, it also generally extracts significant surplus from the bidders. So what users have been getting inefficiently, but for free, may now be allocated efficiently but at a real monetary cost to them. The net benefit to the users, which we call users' surplus, can actually decline significantly when moving from a committee process to an auction even though efficiency improves. Proposing such a mechanism for the DSN problem is politically untenable. No mechanism which leaves current users significantly worse off will be adopted in an environment which values the users' opinions. [7] To see whether we can overcome this political constraint we test a third mechanism which we call AUSM with tokens. We use the same mechanism as above but we give all users an allocation of tokens with which to bid. [8] Our conjecture was that while this mechanism might not achieve the same levels of efficiency that an auction with real monetary transfers would, it would still leave users better off because it would allow and provide incentives for users to identify tradeoffs but would not require them to pay additional funds.

5.1. Negotiation (committee process)

In this section we represent the process currently used in the allocation [9] of DSN slots. We keep that representation in a form that can be implemented experimentally so that data on performance in a controlled setting can be acquired. The experiment is computerized, with each subject seated at his own terminal. The procedure occurs in three phases:

Phase 1: The Request Phase.   Demanders submit requests for packages of slots of the form (n, T, Q, R), where
- $n \in \{1, 2, \ldots, 20\}$ is the number of slots requested,
- $T \in \{$contiguous, periodic, array, maintenance$\}$ is the type of request,
- $Q \in \{1, 2\}$ is the quality of resource where a is an antenna (resource) of quality 1 and b is an antenna of quality 2, and
- $R \in \{0, 1, \ldots, 9\}$ equals the range of the slots requested (the range equals the number of the rightmost column (time) in the request minus the number of the leftmost column in the request).

There is a cost to each request $C(n, T, Q, R)$ which is decreasing in the range

---

[7] For more on mechanism design in which political constraints are important, see Ledyard (1993).

[8] This approach has sometimes been used in practice with some success (measured by users' satisfaction surveys). Examples include scheduling business school interviews, managing computer allocations, etc.

[9] For a detailed description of the process used at JPL see Olson and Porter (1991).

specified (flexibility) and increasing in resource quality and the number of slots requested. The function is

$$C(n,T,Q,R) = (3 - Q)(15 - R)^n \text{ francs}$$

where francs are the experimental currency. Although there are no direct costs assessed in the current procedure at DSN, there are important opportunity costs since large and inflexible requests lead to likelihood of conflict with additional committee time required and potential for appeals to upper management. The function C is intended to capture these costs.

Using the requests, a two-stage algorithm creates a schedule. The algorithm we used is the actual algorithm (RALPH) used by JPL. All requests are included in the schedule. The algorithm minimizes the 'average' level of contention for the resources and maximizes the 'average' level of resource utilization. In the first stage it assigns fractions of each request made to every possible combination of slots which would satisfy the request. In the second stage the requests are consolidated so as to spread out the excess demand for slots as evenly as possible across slots. In other words, in two stages, it finds a solution to the following:

$$\min_{X_{ij}} \sum_{s \in S} \left\{ \max \left\{ \sum_{i \in Ij \in C_{mt}} X_{ij}, 1 \right\} \right\}. \tag{9}$$

There are typically multiple solutions to Eq. (9). The algorithm selects the schedule based on the order in which the scheduler inputs the requests from a mission event priority guideline provided by management. Note that multiple demanders may be assigned to the same slot. Conflicts from the multiple assignments of a slot are then negotiated in a user committee.

Phase 2: The Negotiation Phase.   Once the algorithm has operated, there follows a negotiation phase. Here demanders may freely communicate and agree to drop or switch slots which they have been allocated. Although there is no direct restriction on the number of switches, all negotiations have to be completed in a predetermined time interval, which is known to all subjects. A demander's action in the negotiation phase may be either of the following:

1. A demander may agree to unilaterally give up her assignment to a slot. This may occur in the context of a binding agreement with other subjects who also agree to give up their assignments.
2. A demander may give his currently assigned slot to another demander. This could also form a part of a larger agreement with any number of other subjects.

    In our experiment, the experimenter manually alters the schedules for all subjects from his computer terminal to reflect the agreements concluded in the negotiation phase.

Phase 3: The Appeals Phase.   When agreement cannot be reached on the allocation of the schedule even after negotiation, the contending users can appeal to higher levels of management. In the experiment, any conflicts which are still

unresolved after the negotiation phase are settled through an appeals process which reduces the number of demanders in any slot to one. We sidestep the issue of the preferences of the higher level of management by conducting the appeals process anonymously on the computer and treating all of the subjects equally during the process. To capture the idea that larger requests are more difficult to appeal, we require that a separate appeal be made for each slot. In all of our experiments the appeals process treats the appeals of all contending users equally, although this may not be the case at JPL or other organizations. During the appeals phase, subjects are required to specify for every slot, to which they and at least one other demander are assigned, whether they wish to be involved in the appeal process. There is a per slot cost to demanders who appeal, which was equal to 100 francs in the experiment, again to represent the opportunity costs involved. Each demander appealing for a slot has an equal probability of receiving the slot in the final allocation. No demander may make an appeal for a slot which is not assigned to him after Phase 2.

At the end of Phase 3 an allocation is determined and subjects receive profits. Demanders' requests and appeals are unobservable to other subjects. Communication between subjects is permitted only in the negotiation phase.

5.2. Adaptive user selection mechanism (auction)

The adaptive user selection mechanism (AUSM), is a computerized ascending price auction in which demanders enter bids for packages of slots. All slots and all packages are auctioned continuously and simultaneously. (Also see Section 3.) A demander can enter an integer bid for any package. [10] Let $b_{ij} > 0$ denote a bid by demander i for package j. The process can be most easily understood using the concept of a standing bid. Let S_j be the standing bid for package j. At any time the collection of packages for which there is a standing bid will constitute a feasible allocation. A new bid, b'_{ij}, can become a standing bid if it is greater than the sum of all standing bids for all packages which contain slots which intersect with package j. In other words b'_{ij} is accepted and becomes a new standing bid s_j when

$$b'_{ij} > \sum_{k;\, P_k \cap P_j \neq \varnothing} S_k. \tag{10}$$

The previous standing bids S_k, where k is as defined in Eq. (10), are then displaced and are no longer considered standing bids. Bids can not be withdrawn once they are sent to the market. An important feature of this process is that the

---

[10] It is important that demanders need not enter bids for *all* of the packages. Thus, in general, bidders in AUSM submit fewer bids than the number of commodities, making AUSM less demanding than markets.

sum of all standing bids can only increase when a new bid is accepted. [11] When the auction closes, the demander in possession of the current standing bid for package *j* receives that package.

The auction was closed manually by the experimenter at random within a time interval known to subjects, although the exact time of closing was not known to them in advance. The auction was open for 6–8 minutes in the early periods of each experiment and 4–6 minutes in the late periods. The random ending was designed to prevent a sudden surge of bidding activity at the end of the market period which would occur if the exact time of closing were known to bidders. [12] Also, if at any time,no new bid was received for a 20 second time interval, the market was closed by the experimenter.

This mechanism as it stands has an undesirable feature that may lead to inefficient outcomes, because larger users can prevent smaller but higher-valued users from winning. Suppose, for example, that demander 1 has a valuation of 3000 francs for a package consisting of items 1a, 2a, and 3a, demander 2 has a valuation of 2000 francs for item 1a, and demander 3 has a valuation of 2000 for a package consisting of items 2a and 3a. In the highest-valued allocation, demander 2 receives item 1a and demander 3 receive 2a and 3a. However, if demander 1 bids 2001 francs for the package of 1a, 2a, and 3a, then neither demander 2 nor demander 3 can unilaterally displace 1's standing bid without bidding more than his individual value. Thus, the mechanism produces an inefficient outcome.

To overcome this threshold coordination problem, a feature called Standby Queue for Unifying Individual Demanders (SQUID) was added to the Adaptive User Selection Mechanism. [13] This queue allows individual demanders to jointly submit bids in order to displace standing bids for large packages. A public bulletin board is provided for bidders where they can place a bid that they would be willing to have win but which is not sufficiently large to become a standing bid by itself. In the above example, a bid to pay 1300 for slot 1a by demander 2 would be such a bid. Once placed on the standby queue, that bid can be used by others in concert with their own bid to displace a large bid such as the bid of 2001 for the package of slots 1a, 2a and 3a by demander 1. In the example, once demander 2 has placed a bid in the queue, demander 3 can then choose to combine a bid of his own with demander 2's offer on the standby queue. He might then bid 702 francs for slots 2a and 3a in combination with 2's offer of 1300 for 1a. Since they jointly have a sufficiently high bid to displace demander 1's bid they can do so.

---

[11] It is, of course, possible for the standing bids of any particular package to fluctuate over time.

[12] The experimenter drew closing times before the periods started using a random number generator. This allowed the experimenter to be sure that he was unaffected by the market activity during the period when deciding when to close the market.

[13] The idea is to provide a forum for communication since experimental evidence suggests that the best way to overcome the threshold problem is to allow subjects to communicate. See Ledyard (1995) for a survey of some of these experiments.

Demander 3 can unilaterally execute the joint bid and 2's component is binding on 2 once it is executed. Prior to execution, bids can be withdrawn from the standby queue. Any number of bids could be combined to form one joint bid, with the limitation that each bidder could have at most two bids on the standby queue at any time.

5.3. AUSM with tokens

The AUSM mechanism with the standby queue was also implemented as a mechanism where bids were made, not in terms of an experimental currency convertible to U.S. dollars, but rather using tokens which had no value to demanders and which served only as a medium of exchange. Each demander received a budget of tokens at the beginning of each period with which to bid. Any tokens remaining at the end of the period had no value to the demander and were forfeited before the start of the next period.

Since the initial distribution of tokens is an important aspect in the design of any bidding mechanism which uses tokens, we included two subconditions in the research design, which differed only in the amount of tokens with which subjects were endowed at the beginning of each period. In the Equal Endowment condition, each subject received an equal endowment of tokens each period. In the Competitive Endowment condition, the endowment of tokens available to each subject was proportional to the minimum number of francs required by each subject to achieve the competitive equilibrium allocation. [14] In each of the four experimental sessions of AUSM/tokens, each endowment was used for at least three market periods. Subjects were always informed if the endowments were going to change in the next period. Subjects knew their own endowment but not the endowment of other demanders. Subjects also knew that there were only two possible initial distributions. The initial allocations of tokens to subjects are given in Table 5.

Although some of the demanders in the Competitive Endowment condition (demander 5 in low contention and demander 6 in high contention) received endowments much lower than other demanders, they were nevertheless usually able to purchase units, since there existed packages for which only they had a positive valuation.

5.4. Sequential random efficiency

One metric by which to evaluate the performance of a mechanism in a particular environment is to compare the efficiency achieved to that generated by a

---

[14] If the minimum was less than 5 francs, the endowment given was raised to 5 francs. We recognize that in practice, the planner is unlikely to know the competitive equilibrium price vector, if indeed such a price vector exists. The Competitive Endowment treatment as used here should be viewed as a benchmark employed merely as an alternative to the Equal Endowment treatment. With two different endowments, we can gage the sensitivity of the efficiency of allocations and individual payoffs to the initial endowment of tokens.

Table 5
Initial endowment of tokens

Demander	Low contention					
	1	2	3	4	5	6
Equal endowment	1000	1000	1000	1000	1000	1000
Comp. endowment	1000	850	1600	1800	5	1000
	High contention					
Equal endowment	1000	1000	1000	1000	1000	1000
Comp. endowment	1825	1500	625	1800	675	5

naive computer algorithm. This protects against claims of superior performance by a mechanism when it is really only an 'easy' environment which allows high efficiencies. The algorithm we have chosen is a generalization of the sequential dictator algorithm analyzed in Olson and Porter (1994). The algorithm consists of the following steps.

1. Pick a user. Each user is picked with equal probability.
2. Allocate to that user his most valuable package which is feasible. A package is feasible if no subset of it has been allocated to any other demander.
3. Go to 1. if there exist slots which have not been allocated.
4. Compute $\sum_i \sum_j V_{ij} X_{ij}$ for that allocation, divide by $\bar{V}$.

The algorithm was run 10000 times for each set of parameters. The results of the first 2500 runs are depicted in Figs. 3 and 4. The algorithm generates an average efficiency of 0.8979 and 0.8261 for the low and high contention condi-
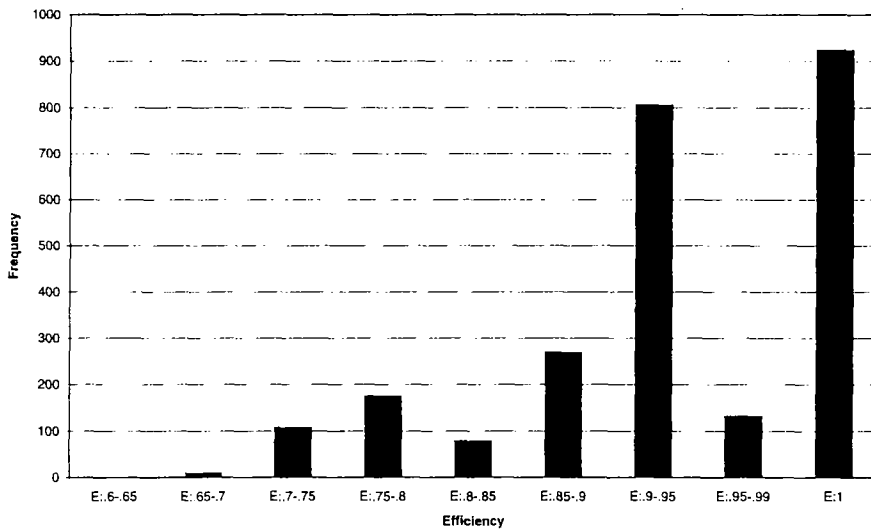

Fig. 3. Distribution of efficiency of random algorithm: 2500 trials – low contention.
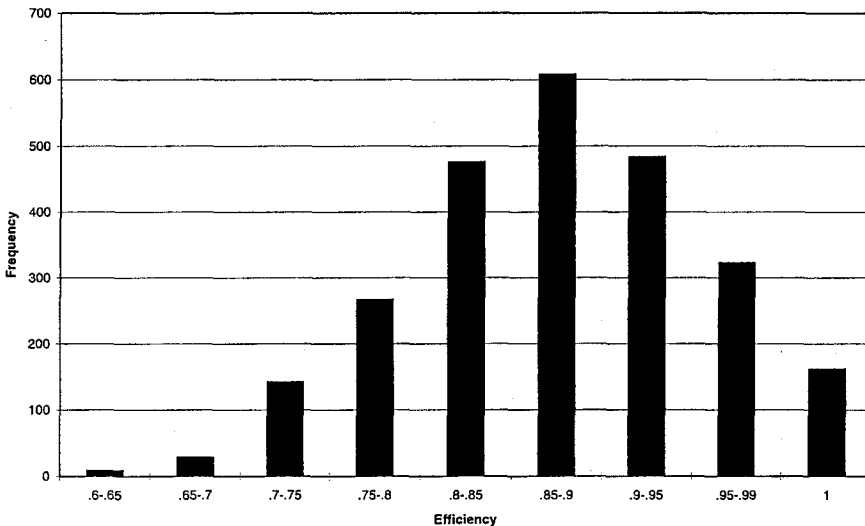
Fig. 4. Distribution of efficiency of random algorithm: 2500 trials – high contention.

tions respectively, with standard deviations of 0.0902 and 0.0876. The algorithm results in the optimal allocation 28.94 and 2.25 percent of the time in the low and high contention conditions, respectively. [15]

6. Results

All of the 12 experiments were conducted in April 1993 at the Laboratory for Experimental Economics and Political Science at the California Institute of Technology. Each experiment consisted of between 8 and 12 market periods. All subjects were undergraduates or pre-freshmen at the Institute. In Table 6, we display some design information concerning the experiments:

In the rows corresponding to AUSM/tokens in the column labelled Num. periods, the two numbers in parentheses represent the number of periods in the session where the Equal and the Competitive endowment conditions respectively were in effect. [16]

---

[15] The algorithm generates, on average, higher efficiencies under low contention for our parameters, although, in general, higher contention does not always imply lower average efficiency.

[16] The instructions used in the experiments can be found Ledyard et al. (1994).

6.1. The data

Our analysis of the effects on allocative efficiency and the cost to users of varying the level of contention and of replication of the market conditions is based on the estimation of regression equations. The following equations were estimated using the data from the committee and the AUSM/money processes:

$$y_i = \beta_0 + \beta_1 \text{period} + \beta_2 \text{hicon} + \epsilon, \quad i = 1, 2, \tag{11}$$

where $y_1 =$ efficiency (as defined in Eq. (7)) and $y_2 =$ cost to users. Users' surplus can then be computed as $y_1 - y_2$. The cost to users is the percentage of $\overline{V}$ paid out by subjects during the market period. In the committee process it is equal to the request costs plus the appeals costs. For the AUSM/money the cost to users is equal to the amount of cash paid out by subjects to obtain their allocation. There are no costs to subjects in the AUSM/tokens.

The variable 'period' is equal to the number of market periods that have elapsed in the experimental session including the current period. It can be interpreted as a variable which isolates the effect of the learning or experience of subjects on the value of the dependent variable. The dummy variable 'hicon' takes on a value of 1 in the high contention treatments and a value of 0 in the low contention treatments and therefore identifies the effect of variations in the contention level.

The following equation was estimated for the AUSM/tokens data:

$$e = \beta_0 + \beta_1 \text{period} + \beta_2 \text{hicon} + \beta_3 \text{compendow}, \tag{12}$$

where $e =$ efficiency and compendow $= 1$ if the Competitive Endowment condition is in effect and equals 0 otherwise. For the AUSM/tokens data, since all rents go to the demanders, market efficiency is exactly equal to users' surplus. The

Table 6
Information about experimental sessions

Exp. number.	Mechanism	H/L contention	Num. periods
1	Committee	H	10
2	Committee	H	9
3	AUSM	H	9
4	AUSM	H	11
5	AUSM/tokens	H	8 (3/5)
6	AUSM/tokens	H	11 (8/3)
7	Committee	L	8
8	Committee	L	10
9	AUSM	L	10
10	AUSM	L	10
11	AUSM/tokens	L	9 (6/3)
12	AUSM/tokens	L	12 (7/5)

Table 7
Allocative efficiency and cost to users of the three mechanisms (in percent) [a]

Dependent variable	One	Period	Hicon	Compendow	R^2	n
Efficiency – committee	80.434 (3.038)	1.546 (0.515)	−19.100 (2.520)		0.65	37
Efficiency – AUSM/Money	85.916 (3.398)	1.324 (0.486)	−3.096 (2.634)		0.20	40
Efficiency – AUSM/Tokens	80.317 (5.815)	1.035 (0.636)	6.216 (3.328)	0.400 (2.63)	0.14	40
User cost – committee	6.23 (1.36)	−0.702 (0.255)	5.94 (1.30)		0.44	37
User Cost – AUSM/Money	16.20 (4.45)	1.54 (0.731)	21.10 (3.65)		0.55	40

[a] Standard errors in parentheses.

variable 'compendow' identifies the effect on efficiency of varying the initial endowment of tokens. The results of the estimations of the five equations are given in Table 7. The standard errors of the estimates are given in parentheses.

Tables 8–10 contain the estimated efficiency and the estimated cost to users of each of the three allocation mechanisms in periods 1 and 10, as well as under the two contention levels. The numbers in Tables 8–10 are taken from the estimated coefficients in Table 7. L and H represent the Low and High contention treatments respectively. L1 and L10 denote the estimated value of the relevant variable in periods 1 and 10 respectively in the low contention condition. The estimates in Tables 8–10 are the values of the regression in the relevant period and treatment, and not the actual observed values of the variables. Thus, it is possible, as in the low contention period 10 estimate of the cost in the committee process, that the estimated value could be negative although the actual costs must always be greater than or equal to zero. The figures in parentheses *give the probability* that the efficiency or the users' surplus from the sequential dictator algorithm is less than or equal to the estimated surplus from the mechanism. So low numbers in

Table 8
Efficiency and user surplus: Committee process (in percent)

	L1	L10	H1	H10
Effic.	81.98 (16.44)	95.89 (71.06)	62.88 (1.21)	76.79 (25.77)
Cost	5.53	−0.79	11.47	5.15
Users' surplus	76.45 (11.84)	96.68 (71.06)	51.41 (0)	71.64 (10.1)

Table 9
Efficiency and user surplus: AUSM/Money (in percent)

	L1	L10	H1	H10
Effic.	87.24	99.16	84.14	96.06
	(26.54)	(71.06)	(54.75)	(95.09)
Cost	17.74	31.60	38.84	52.70
Users' surplus	69.50	67.56	45.30	43.36
	(1.81)	(1.47)	(0)	(0)

parentheses indicate poor performance by the mechanism relative to the sequential dictator (random) algorithm.

The efficiency generated by the committee process at the beginning of the experimental sessions is less than that generated by the random algorithm in both the high and the low contention environments. In the low contention condition, however, the efficiency of the committee process improves over time so that, on average, it is better than the random algorithm by period 10. However, in the high contention environment, the committee performs very poorly and, even by period 10, does not achieve efficiencies as high as the random algorithm.

The costs to users in the committee process decline with time as users apparently learn to avoid these bureaucratic frictions. Higher contention makes such avoidance more difficult. The most remarkable finding is that only in a low contention environment and only after a number of replications does the current committee process, with a decision support algorithm, outperform the simple sequential dictator algorithm.

From an efficiency point of view, AUSM/money performs better than the committee in both the high and the low contention conditions. The efficiency is also increasing over time. By the tenth period, AUSM significantly outperforms both the committee process and the sequential dictator algorithm. However, increases in efficiency are associated with increasing costs to the users so that user surplus remains unchanged. In fact, users' surplus is substantially less under AUSM than under the committee process or under the random algorithm described in Section 5.4.

Table 10
Efficiency and user surplus: AUSM/tokens/the Equal Endowment Condition (in percent)

	L1	L10	H1	H10
Effic.	81.35	90.67	87.56	96.88
	(16.44)	(57.25)	(72.71)	(95.09)
Cost	0	0	0	0
Users' surplus	81.35	90.67	87.56	96.88
	(16.44)	(57.25)	(72.71)	(95.09)

Under the AUSM/tokens, efficiency and, thus, users' surplus increases over time. The efficiency is higher under high than under low contention. Under high contention, the users' surplus was largest under the AUSM/tokens mechanism than any of the three others. Furthermore, as can be seen in Table 7, changing the token endowment had no effect on the efficiency levels recorded. This is some-what unexpected so it is reasonable to consider further the distributional effects of the different token endowments. A natural measure of this effect is the sample variance of payoffs among the users which can be defined as

$$S^2 = \sum_{i=1}^{I} \frac{(\pi_i - \bar{\pi})^2}{I-1} \tag{13}$$

where, S^2 is the sample variance, π_i is the profit of demander i, which equals the sum of his valuations (in terms of francs) for the packages he receives and $\bar{\pi}$ is the average profit of demanders. The variance is analysed by estimating the equation below. The estimates are reported in Table 11.

$$S^2 = \beta_0 + \beta_1 \text{period} + \beta_2 \text{hicon} + \beta_3 \text{compendow}. \tag{14}$$

One can see in Table 11 that the change in endowment also has no significant effect on the variance of the payoffs. The fact that the variances are not different, however, does not suggest that there were no distributional differences under the two endowments. For example, under low contention, demanders 3 and 4 were substantially worse off (payoffs were 15 and 19 percent lower respectively), while demander 6 was better off (by 19 percent) in the Equal Endowment condition than in the Competitive Endowment condition. Under high contention, demander 4 received considerably higher payoffs (by 30 percent) in the Competitive Endowment condition while demander 6 received considerably lower payoffs (by 37 percent). Tables 12 and 13 show the percentage of market periods in both subconditions in which each demander received each of the packages which had value to him. The columns labelled EE and CE indicate the percentage of the time that package P_j was allocated to demander i under the Equal Endowment and Competitive Endowment treatments respectively.

Table 11
The effect of the two endowments on the variance of payoffs across subjects [a]

Variable	Coefficient	Std. error	t-stat
one	3404260	788676	4.316
per	− 167333	112328	− 1.490
hicon	712256	600223	1.187
compendow	275415	648072	0.425

[a] $R^2 = 0.08$, $n = 40$.

Table 12
Allocated packages under AUSM/tokens under both endowments: High contention condition

i	$s \in P_j$				V_{ij}	E.E. [a]	C.E. [b]
1	3a	4a	5a		2450	0	0
1	3b	4b	5b		1450	0	0
•1	8a	9a	10a		3150	100	100
•1	8b	9b	10b		2150	91	100
•2	4a	5a			3200	91	100
•2	4b	5b			2000	91	100
2	8a	9a	10a		2000	0	0
2	8b	9b	10b		1000	9	0
3	3a	8a			1600	0	0
3	1b	6b			1600	9	25
•3	2b	7b			3200	64	50
3	3b	8b			1600	0	0
•4	2a	7a			3000	82	100
4	3a	8a			1500	0	0
•4	1b	6b			1750	55	75
4	2b	7b			1250	27	50
4	3b	8b			750	0	0
•5	3a	3b			3000	100	100
5	4a	4b	5a	5b	3000	9	0
5	8a	8b			1500	0	0
5	9a	9b	10a	10b	1500	0	0
•6	1a				750	100	100
6	2a				1000	18	0
6	3a				500	0	0
•6	6a				750	100	100
6	7a				1000	18	0
6	8a				500	0	0
6	1b				750	36	0
6	2b				500	9	0
6	3b				250	0	0
6	6b				750	36	0
6	7b				500	9	0
6	8b				250	0	0

[a] E.E. = % of periods in which this package was received in the Equal Endowment treatment.
[b] C.E. = % of periods in which this package was received in the Competitive Endowment treatment.

Under both endowments and both levels of contention, each package except for package (1b, 6b) under low contention in the Equal Endowment condition, was allocated efficiently (that is to the correct demander) at least 50 percent of the time. This suggests that efficiencies under both endowments are high and not much different from each other. Under both sets of endowments of tokens, the demanders with the highest valuations were generally able to bid high enough to price out competing demanders. Even the very small endowments of demander 5

Table 13
Allocated packages under AUSM/tokens under both endowments: Low contention condition

i	$s \in P_j$				V_{ij}	E.E. [a]	C.E. [b]
*1	9a	10a			4000	85	88
1	9b	10b			850	8	25
2	9a	10a			1000	8	12
*2	9b	10b			4000	85	75
*3	1a	6a			3250	85	100
3	2a	7a			1000	23	12
*3	2b	7b			2250	100	100
*4	2a	7a			3250	77	88
4	3a	8a			1000	15	25
*4	1b	6b			2750	38	50
*5	4a	5a	4b	5b	3000	100	100
5	9a	10a	9b	10b	1000	8	0
6	1a				800	15	37
*6	3a				1200	85	75
6	6a				800	15	0
*6	8a				1000	85	75
6	1b				800	62	50
*6	3b				1000	100	100
6	6b				800	62	50
*6	8b				1000	100	100

[a] E.E. = % of periods in which this package was received in the Equal Endowment treatment.
[b] C.E. = % of periods in which this package was received in the Competitive Endowment treatment.

under low contention and demander 6 under high contention were enough to allow the user to purchase the packages which they would receive in the competitive equilibrium, because there were no competing demanders for these packages.

Under high contention, demanders 1, 2 and 5, who were in the same viewperiod, usually received the same allocations across the two endowments. Demanders 3 and 4, who were in the same viewperiod as demander 6, sometimes purchased packages meant for each other. Demander 4 had a much larger endowment of tokens in the Competitive Endowment condition than demander 3 who in turn had a much larger endowment than demander 6. Not surprisingly, demander 4 received packages (1b, 6b), (2a, 7a) and (2b, 7b) more often in the Competitive Endowment condition when he had a larger endowment. The maintenance demander 6 received the same packages less often when she had the smaller endowment. Demander 3 was able to purchase (1b, 6b) more often in the Competitive Endowment condition, easily outbidding demander 6, who was never able to purchase (1b, 6b) in the Competitive Endowment condition but was able to purchase it 36 percent of the time in the Equal Endowment condition. The average allocative efficiency under the two endowments was 0.937 and 0.953 for the Equal and Competitive endowments respectively.

A similar pattern occurred under low contention. The contiguous demanders 1

and 2 usually divided packages (9a, 10a) and (9b, 10b) between them. Demanders 3 and 4 often received packages intended for the other, with 4 receiving more packages under the Competitive Endowment condition, when his token endowment was greater, usually at the expense of demander 6. Demander 5, with his very low endowment in the Competitive treatment, was able to obtain his most preferred package, but his budget was too low to obtain the other package. Demander 6 was substantially worse off in the Competitive Endowment condition since she was less able to compete in the bidding with demanders 3 and 4. Demander 3 was sometimes able to get the package consisting of (1a, 6a) and demander 4 received (3a, 8a) and (1b, 6b) more often. Efficiency was on average 0.87 and 0.902 in the Equal and Competitive Endowment treatments respectively.

7. Discussion

The effect of the different treatments on efficiency and user surplus is illustrated in Fig. 5. Under low contention, the Committee process provides more surplus to users as a whole than the AUSM/tokens which in turn provides more users' surplus than AUSM/money. This is consistent with the field observation that there is often little effort to replace committees with market-type systems of allocation when conflict between users is not too severe even though efficiency may be low. The committee process gives users more surplus than the two other mechanisms even though it does no better than the sequential dictator algorithm. Under high contention, however, AUSM/tokens yields the highest surplus to the users, followed in order by the sequential dictator algorithm, the committee and

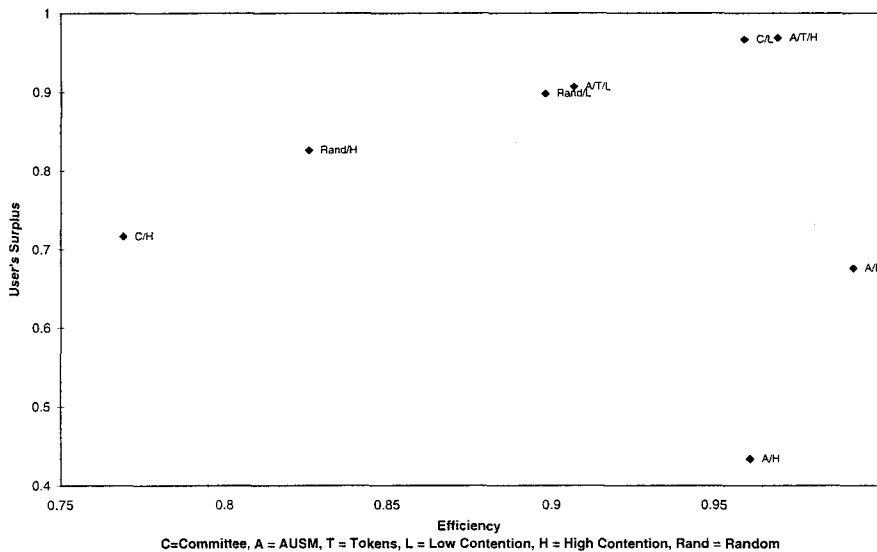C=Committee, A = AUSM, T = Tokens, L = Low Contention, H = High Contention, Rand = Random

Fig. 5. Efficiency and user surplus: All treatments – period 10.

AUSM/money. The committee clearly fails when the level of contention increases since the cost of negotiation and conflict resolution increases and the ability of the mechanism to allocate the schedule efficiently diminishes. AUSM/tokens behaves in the opposite way. The user surplus is higher when contention is high, since the efficiency is higher and user costs are zero. Under both high and low contention, the AUSM/money, with its extraction of rents from the users, generates lower users' surplus than the other mechanisms. When highly efficient auctions such as AUSM are proposed by economists to solve allocation problems, we should expect such recommendations to encounter resistance from thoughtful future bidders in the auction.

The relatively strong performance of the sequential dictator algorithm is not surprising. The sequential dictator algorithm has aspects which give it a good chance to enhance efficiency. One is the fact that it sidesteps the information revelation problem, by choosing for each subject his most preferred package. [17] Our implementation thus assumed perfect information but the resulting allocations were compared against those generated by the interaction of strategic agents in an incomplete information environment. The efficiency of the sequential dictator as we have measured it should therefore only be treated as a benchmark and this paper should not be considered a test of its performance as a mechanism.

From the standpoint of system efficiency, some of the mechanisms are superior to others. Under low contention, AUSM/money, the committee and the sequential dictator algorithm perform better than AUSM/tokens. Under high contention AUSM/money and AUSM/tokens generate more efficient allocations than the random algorithm, which in turn had more efficient outcomes than the committee. This reinforces the findings of Banks et al. (1989) that the AUSM is capable of generating very high allocative efficiencies.

There is evidence of improvement over time in the level of efficiency in all three of the mechanisms. The estimated efficiency level is higher in period ten than in period one in all six conditions. Cost declined in the committee as users seemed to learn to avoid conflicts over time. In AUSM/money, the revenue extracted from the users increased with replication of the auction, and rising revenue seemed to be related to increasing efficiency.

Although varying the distribution of tokens in AUSM/tokens had no effect on total payoffs or on the variance of payoffs for the particular parameters of our experiment, the distributions of payoffs across subjects was different, as demanders with lower endowments received lower payoffs. The robustness of this market efficiency finding should be interpreted with caution and does not suggest that the

---

[17] In the one-to-one assignment problem, such honest revelation is a dominant strategy. (See Olson, 1991; Olson and Porter, 1994.) This is no longer true in this fundamentally more complex many to one environment.

initial distribution of tokens can be ignored by planners in field applications. The result does suggest, however, that there is some room for flexibility in varying the initial distribution of tokens, without causing large fluctuations in allocative efficiency.

8. What have we learned?

At the beginning of this paper we presented a stylized story to illustrate why user committees might be able to survive as organizational devices to manage shared facilities even though it is widely recognized by both the users and the organization that significant value is being foregone. Our study provides new evidence supporting the foundations of this pessimistic view. However, our study, using the methodology of applied mechanism design, also establishes that there are viable alternatives to user committees that do not involve large losses in value to the users. This provides some evidence that significant increases in value, for both users and the organization, can be found and captured by using designed mechanisms to improve the management of shared facilities.

The basis for these somewhat sweeping statements can be found in three key findings of our study. We split the first finding focusing on the performance of user committees into two parts. (1a) *Experienced user committees using decision support algorithms produce reasonably efficient allocations in lower conflict situations.* The key facts supporting this observation are found in Table 8. By period 10 in the low contention environment, the user committees are achieving, on average, 96% of the maximum value with virtually no cost to the users. The probability that the user committee yields a higher value than the random algorithm is, however, only 71%, which is not good but is good enough to support the continued use of the committee. (1b) *Experienced user committees using decision support algorithms produce reasonably bad allocations in high conflict situations.* The key facts supporting this observation are also found in Table 8. By period 10 in the high contention environment the user committees are achieving on average, 77% of the maximum value with a cost to users of 5% of the maximum value. The basis for describing this performance as bad is that the probability that the user committee yields a higher value than the random algorithm is only 10% in the high contention environment. There is obviously significant value being foregone.

The second key finding concerns the viability of an 'economic' solution for the recovery of the value foregone. We also split this into two parts. (2a) *There is an economic process, called the Adaptive User Selection Mechanism (AUSM), in which users bid and pay dollars for time, which yields high efficiencies in high-conflict situations.* The key facts supporting this observation are found in Table 9. In the very first period and in the high contention environment, AUSM/Money is achieving 86% of the maximum possible value. By period 10 in

the high contention environment it is achieving, on average, 96% of the maximum possible value. The basis for calling this high is that the probability that the random algorithm would do better is only 5%. There is obviously significant value being created by this economic solution – about a 25% increase in value over the user committee and, using a better measure of improvement, a 365% increase in the probability of beating the random algorithm. (2b) *Because of the prices paid, the economic solution of AUSM / Money leaves users significantly worse off than user committees do.* The key facts supporting this observation are found in Tables 8 and 9. Looking at period 10 in the high contention environment, we see that the user committee leaves a users' surplus of 72% whereas, because of the payment of fees, AUSM leaves the users a surplus of only 43%. Neither is particularly desirable, since it is essentially a sure thing (almost a 100% probability) that the random algorithm would yield a higher surplus than AUSM. There should remain little mystery as to why user committees don't like market solutions especially in those high contention situations where they are most often espoused.

The third key finding concerns the existence of better mechanisms for both the users and the organization. (3) *There is a modification of AUSM in which tokens, or internal money, replace real money and which results in highly efficient allocations without extracting any of the users' surplus.* The key facts supporting this observation are found in Table 10. By period 10 in the high contention condition, with an equal allocation of tokens AUSM/Tokens is producing on average efficiency and users' surplus equal to 97% of the maximum possible. This is very high in the sense that there is only a 5% chance that the random allocation would produce a higher value. (A surprising and unexplained fact is that this is as high a gross value on average as AUSM/Money achieves.)

We conclude with several policy thoughts. From the point of view of a planner interested in achieving a high allocative efficiency, AUSM is clearly the best of the mechanisms considered here. The efficiencies generated by AUSM are, in our view, very high considering the complex structure of the environment. We are doubtful that other mechanisms could do better although it surely remains an open research challenge to find a better one.

From the point of view overall users' surplus, mechanisms can be found which surpass AUSM/money. When contention is low, it seems to be difficult to improve upon the current committee process. None of our alternative mechanisms were able to. This may be why extensive conflict is necessary before users are willing to discard committee systems. When contention is high we find that AUSM/Tokens generates very high surpluses for the users. The tokens seem to provide enough information to coordinate the allocation of slots efficiently, yet the buyers do not have to give up any surplus. This effect was surprisingly robust to changes in the initial endowment of tokens.When conflict is high, it will be revealed in complaints about the current committee/negotiation process. Under these circumstances, using a carefully designed auction with tokens may appeal to the users in a way that an auction using real money would not.

Acknowledgements

References

Banks, J., J. Ledyard and D. Porter, 1989, Allocating uncertain and unresponsive resources: An experimental approach, RAND Journal of Economics 20, 1–25.

Bykowsky, M., R. Cull and J. Ledyard, 1995, Mutually destructive bidding: The FCC auction design problem, Social science working paper no. 916 (California Institute of Technology, Pasadena, CA)

Calsamiglia, X., 1977, Decentralized resource allocation and increasing returns, Journal of Economic Theory 14, 263–283.

Grether, D., R.M. Isaac and C. Plott, 1981, The allocation of landing rights by unanimity among competitors, American Economic Review, Proceedings, 71, 166–171.

Hurwicz, L., 1994, Economic design, adjustment processes, mechanisms, and institutions, Economic Design 1, 1–14.

Jordan, J., 1987, The informational requirements of local stability in decentralized allocation mechanisms, In: T. Groves, R. Radner and S. Reiter, eds., Information, incentives and economic mechanisms (University of Minnesota Press, Minneapolis, MN) Ch. 5.

Koopmans, T. and M. Beckman, 1957, Assignment problems and the location of economic activities, Econometrica 25, 53–76.

Ledyard, J., 1993, The design of coordination mechanisms and organizational computing, Journal of Organizational Computing 3, 121–134.

Ledyard, J., 1995, Public goods: A survey of experimental research, In: A. Roth and J. Kagel, eds., Handbook of experimental economics (Princeton University Press, Princeton, NJ).

Ledyard, J., C. Noussair and D. Porter, 1994, The allocation of a shared resource within an organization, Working paper 1063 (Krannert School of Management, Purdue University, West Lafayette, IN).

Olson, M., 1991, Dominant and Nash strategy mechanisms for the assignment problem, Social science working paper no. 770 (California Institute of Technology, Pasadena, CA).

Olson, M. and D. Porter, 1991, A deep space resource allocation primer, Mimeo. (California Institute of Technology, Pasadena, CA).

Olson, M. and D. Porter, 1994, An experimental examination into the design of decentralized methods to solve the assignment problem with and without money, Economic Theory 4, 11–40.

Papai, S., 1994, Assignment by design, Mimeo. (California Institute of Technology, Pasadena, CA).

Riker, W. and I. Sened, 1991, A political theory of the origin of property rights: Airport slots, American Journal of Political Science 35, 951–969.

Saari, D. and C. Simon, 1978, Effective price mechanisms, Econometrica 46, 1097–1125.

Smith, V., 1976, Experimental economics: Induced value theory, American Economic Review 66, 274–279.