Optimal Energy Allocation and Admission Control for Communications Satellites

Alvin Fu, Eytan Modiano, and John Tsitsiklis

Abstract— **We address the issue of optimal energy allocation and admission control for communications satellites in earth orbit. These satellites receive requests for transmission as they orbit the earth, but may not be able to serve them all, due to energy limitations. The objective is to choose which requests to serve so that the expected total reward is maximized. The special case of a single energy-constrained satellite is considered. Rewards and demands from users for transmission (energy) are random and known only at request time. Using a dynamic programming approach, an optimal policy is derived and is characterized in terms of thresholds. Furthermore, in the special case where demand for energy is unlimited, an optimal policy is obtained in closed form. Although motivated by satellite communications, our approach is general and can be used to solve a variety of resource allocation problems in wireless communications.**

Keywords— **Satellite, Communication, Resource Allocation, Dynamic Programming**

I. INTRODUCTION

FOR most satellites, energy management is a critical issue, for the simple reason that energy efficiency in a satellite directly translates into cost savings. A satellite with lower energy requirements requires a smaller energy source (solar panel, reactor, etc.) and a lighter battery pack, both of which translate into weight savings. The weight savings generally provide an economic benefit - a smaller launch vehicle might be selected, thus decreasing cost, or more maneuvering fuel could be carried, which would result in longer system life.

It is thus important to accurately anticipate energy input and storage requirements for satellites. To do so, one must model the operation of the satellite and its energy consumption. If appropriate, it may be necessary to determine a strategy for energy consumption.

For instance, a television broadcast satellite in geosynchronous orbit will enjoy continuous sunshine for its solar cells except for brief periods of eclipse, while demand for energy is relatively steady and unchanging [7]. With both input and output of energy relatively static, such a satellite may not require a sophisticated energy consumption strategy. On the other hand, a data communications satellite in medium or low earth orbit will experience prolonged periods of darkness and lack of energy input. At the same time, if the satellite is servicing a best-effort packet data network (such as the Internet), demand for services will often be bursty, and the satellite must choose amongst users to be served. In such a situation, the need for an energy consumption strategy is obvious.

Energy input for a data communications satellite in earth orbit generally consists of power from solar cells [10]. The quantity and timing of the input are known and can be determined well in advance. As for energy outflow, a major source of energy expenditure is often the power needed to transmit on the downlink connection back to earth. Receiving signals sent up from earth requires relatively little power in comparison, and sending signals to neighboring satellites (if the satellite is part of a constellation with satellite crosslinks) is generally not energy intensive. In the presence of multiple competing demands for downlink service, the optimization of energy consumption consists of deciding which users to serve.

The amount of service demanded by users is often a widely varying quantity. For instance, a satellite providing wireless phone service will likely experience much more demand when it is over New York than when it is over the North Pole. Furthermore, the energy required for servicing different users is usually not the same. Thunderstorms, for example, can severely attenuate the satellite signals. Users may differ in distance to the satellite, overhead atmospheric conditions, or even antenna size, all of which imply that the satellite must expend a different amount of energy to service each user. To complicate matters even further, different users or user classes may provide differing payments and rewards for service by a satellite.

There is little prior research on the topic of optimal allocation of satellite energy under limited power and finite energy storage conditions. In the 1970s, a study by Aein and Kosovych [1] investigated capacity allocation for satellites serving both switched and packet based networks, while Shaft [12] looked at unconstrained allocation of power and gain to service communication satellite traffic. Recently,

The authors are with the Laboratory for Information and Decision Systems, Massachusetts Institute of Technology (MIT), Cambridge, USA. E-mail: alfu@mit.edu, modiano@mit.edu, jnt@mit.edu. This research was supported by DARPA under the Next Generation Internet Initiative.

many researchers have examined the use of satellites to supplement terrestrial data networks [11] [13]. This work is most often focused on design and performance evaluation of such space networks, but there is little attention paid to energy allocation issues for satellites in such a network. Perhaps the closest study to our current work is one by Ween et.al., [14] who studied resource allocation for low-earth-orbit satellites providing GSM cellular services. Resource allocation for satellite beams and path selection has been studied, [9], as has the allocation of bandwidth [2].

Much work has been done on design and analysis of power systems for satellites. For instance, Kraus and Hendricks have developed a model for estimating satellite power system performance [8]. A study in 1986 examined operational scheduling for the (then) proposed manned space station [3], and centered on appropriately matching the many power sources to power sinks on the space station.

In general, current satellite operators follow heuristic rules about energy allocation. For example, a simple rule would be to serve all requests as long as sufficient energy is available. Such a “greedy” approach is clearly suboptimal if different users require different amounts of energy or provide different rewards for the same service.

This paper develops a method that allocates energy for a single satellite. As the satellite moves in its orbit, it encounters different users with different overhead atmospheric conditions, financial rewards, demand levels, and so forth. For each unit of energy expended, the satellite receives a certain amount of reward, which incorporates distances, atmospheric conditions, and financial considerations. The reward changes with each time step, and is assumed to be random and unknown until the actual time of service, although its probability distribution is known. The satellite may also face a limit on the amount of energy it can expend: there may be a physical power limit for its transmitter, or there may simply not be enough customer demand. The demand is again assumed to be random and not known until the time of service. At the same time, the parameters for available energy are largely known: the satellite has a battery whose size is known and finite, and receives energy from its solar cells according to a known schedule. The objective is to expend the energy (service the users) in a way that maximizes reward.

We present a method for optimizing energy consumption to maximize reward. In addition, we provide useful suboptimal heuristics for the general case based on certainty equivalent control and on a closed-form optimal solution to the special case where demand is unlimited. Finally, although originally motivated by a satellite energy allocation problem, our approach has a natural application to wireless networking, which we discuss in section V.

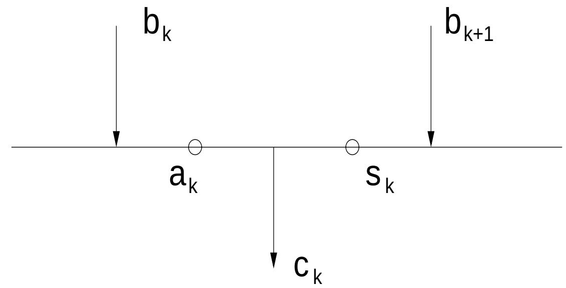

Fig. 1. Energy Flow

II. System Model

We consider a satellite system with slotted time, stochastic reward, stochastic demand, and a finite time horizon. The satellite receives energy in each time slot according to a fixed and known schedule and can store it in a battery of finite size. At the same time, it serves customers by expending energy. The reward obtained per unit energy changes randomly in each time step. The demand for energy during each time step is random as well. The objective is to find an optimal policy that maximizes expected reward by choosing how much (if any) of the demand to service at each time.

Denote the energy available for the satellite to spend at time slot k with the variable a_k. It is assumed that during any time slot, the satellite can spend the energy in its battery plus any incoming energy from the solar panels. Thus a_k consists of the energy in the battery plus the energy input for time slot k, denoted b_k.

The inputs b_k represent incoming energy from the solar panels or reactor. Because orbits and reactor performance are predictable, the energy inputs b_k are assumed to be known in advance. In this model, the satellite starts with energy a_0 and at each time $k > 0$ receives energy input b_k according to a predetermined and known schedule.

At each time slot k, the satellite operator may elect to consume an amount of energy c_k (up to a_k) in servicing users. Any unused energy $\hat{s}_k = a_k - c_k$ must be stored in the battery, which has a capacity of $E_{\max}$. Unused energy that cannot be stored is lost. Therefore, for any time slot, the energy in the satellite's battery consists of available energy from the previous stage minus consumption from the previous stage, subject to a battery capacity limit. The energy stored in the battery at time k for use in the next stage, which we define as s_k, is then given by the term $s_k = \min(E_{\max}, a_k - c_k)$.

As can be seen in figure 1, the energy available for use by the satellite at time $k+1$ is expressed as:

$$a_{k+1} = \min(E_{\max}, a_k - c_k) + b_{k+1} \tag{1}$$

Alternatively, a_{k+1} can be written in terms of stored energy s_k as

$$a_{k+1} = s_k + b_{k+1} \tag{2}$$

Each unit of energy consumed provides the satellite operator with a reward r_k. The reward r_k is a non-negative random variable with a probability distribution $p_{r_k}(r_k)$ that varies with time. Although $p_{r_k}(r_k)$ is known a priori, the actual value of r_k is not known until time k. Similarly, the user's demand for energy, d_k, is also a random variable with a priori known probability distribution $p_{d_k}(d_k)$, but the actual value of demand at time k is not known until time k. The random variables r_k and d_k, $k = 1, 2, \ldots, n$, are assumed independent.

The objective is to choose a consumption policy that maximizes total expected reward over a time horizon of n time steps. The total expected reward is given by

$$E\left[\sum_{i=1}^{n} r_i c_i\right] \tag{3}$$

where consumption is subject to demand and energy constraints.

Notice that implicit in equation (1) is the assumption that any incoming energy during a time slot can be consumed during that slot without being stored in the battery. This amounts to assuming that energy input and consumption rates are constant for the duration of a time slot, a realistic assumption for sufficiently small slot durations.

Furthermore, there is an inevitable energy loss associated with charging and discharging a battery, and the energy of a battery varies with its discharge rate. Although not currently captured, these battery effects can be incorporated into the model by proper adjustment of the reward structure. It is also known that the pulsed discharge of a battery yields significantly more energy than steady discharge at the same current, and Chiasserini and Rao [5] [6] have developed algorithms to exploit this property for data transmission. This property could be included in our formulation by the use of a model where reward probabilities are dependent on previous consumption and energy state. Due to the short duration of battery pulses, incorporating this effect would require the use of very short time slots (e.g. one second or less).

In the following sections, we formulate the energy allocation problem within the framework of dynamic programming [4]. Generating an optimal policy and a value function from the dynamic programming recursion can be computationally difficult. We prove concavity of the value function and thereby obtain some properties of an optimal policy. The concavity property is also the basis for two separate methods of calculating the value function and generating an optimal policy, both of which provide scalability and a significant decrease in computation time. Next, we analyze the certainty equivalent heuristic and show that it has a simple structure in the special case where the expected reward per energy unit is the same at each period. In addition, we derive an optimal policy for the special and limiting case where demand is unlimited. Last, we present a numerical example contrasting the performance of the three algorithms with a greedy algorithm and examine an alternative application in wireless networking.

III. Dynamic Programming Formulation

In this section, we present a dynamic programming algorithm for the problem formulated in the previous section. As usual in dynamic programming, we introduce the value function $J_k(a_k, r_k, d_k)$, which provides a measure of the desirability of the satellite having available energy level a_k at time k, given that current demand is d_k and current reward is r_k. The optimal value functions $J_k(a_k, r_k, d_k)$ for each stage k are related by the following dynamic programming recursion:

$$\begin{aligned} J_k(a_k, r_k, d_k) = \max_{0 \le c_k \le a_k} \{ & r_k \min(c_k, d_k) \\ & + E_{(r_{k+1}, d_{k+1})}[J_{k+1}(\min(a_k - c_k, E_{\max}) + b_{k+1}, \\ & \quad r_{k+1}, d_{k+1})]\} \end{aligned} \tag{4}$$

The two terms in the maximization represent the tradeoff in reward between consuming and saving energy. The $r_k \min(c_k, d_k)$ term represents the reward for consumption; the satellite receives r_k units of reward per unit of energy consumed, up to a maximum consumption of d_k. The expected value term represents the value of saving energy. As discussed earlier, the satellite's available energy in the next stage, a_{k+1}, is $a_{k+1} = \min(a_k - c_k, E_{\max}) + b_{k+1}$. The expected reward for having this much energy available is given by the expectation $E[J_{k+1}(a_{k+1}, r_{k+1}, d_{k+1})]$, which is taken over the distribution of d_{k+1} and r_{k+1}.

In order to maximize expected reward the satellite should choose the consumption c_k that maximizes the right-hand side in equation (4). Notice that any consumption beyond the demand d_k is wasted, as is any energy saved beyond $E_{\max}$.

An alternative expression for the value function can be obtained by using the stored energy term $s_k = \min(a_k -$

$c_k, E_{\max})$. Hence for stage k

$$J_k(a_k, r_k, d_k) = \max_{0 \le s_k \le \min(a_k, E_{\max})} \{r_k \min(a_k - s_k, d_k) + E_{(r_{k+1}, d_{k+1})}[J_{k+1}(s_k + b_{k+1}, r_{k+1}, d_{k+1})]\} \quad (5)$$

For both formulations, at the final stage, stage n, the value function is given by

$$J_n(a_n, r_n, d_n) = r_n \min(a_n, d_n) \quad (6)$$

This of course represents the reward for consuming the remaining energy in the satellite.

A. Concavity of the Value Function

The value function can be evaluated numerically; however, execution time can be slow. The major difficulty is computing the expectation $E_{(r_{k+1}, d_{k+1})}[J_{k+1}(s_k + b_{k+1}, r_{k+1}, d_{k+1})]$, for every a_k, s_k, and k. It is apparent that there is a five-stage loop here: the algorithm must consider all values of s_k, a_k, r_k, and d_k in each stage, and there are a total of n stages. Fortunately, the execution time can be considerably improved by taking advantage of some properties of the value function.

Theorem 1:
$J_k(a_k, r_k, d_k)$ is concave in a_k for any fixed r_k and d_k.

Proof:
Given in appendix -A.

Corollary:
Let us define $\overline{J}_k(a_k)$ as

$$\overline{J}_k(a_k) = E_{r,d}[J_k(a_k, r_k, d_k)] \quad (7)$$

Then $\overline{J}_k(a_k)$ is concave in a_k as well.

The concavity properties of the expected value function $\overline{J}_{k+1}(a_{k+1})$ dictates the nature of an optimizing consumption policy. In the dynamic programming recursion, the expected value function for time $k+1$ represents the expected reward for saving energy at time k. Since this function is concave, it translates into a decreasing marginal reward for saving energy. The marginal reward for consuming energy, on the other hand, is r_k and then zero after the demand limit is reached.

Let us assume from now on, and throughout the rest of the paper, that the variables a_k, s_k, c_k, d_k, $E_{\max}$, and b_k are all integer. This will allow us to consider computational methods for solving the problem of interest.

Let $\phi_k(r_k)$ be the smallest s_k in the range $0 \le s_k \le E_{\max}$ such that

$$\overline{J}_{k+1}(s_k + 1 + b_{k+1}) - \overline{J}_{k+1}(s_k + b_{k+1}) < r_k$$

and set $\phi_k(r_k) = E_{\max}$ if such an s_k does not exist.

Because of the concavity of $\overline{J}_k(a_k)$, an optimal policy can be obtained by setting s_k to be

$$\begin{array}{lll} \min(a_k, E_{\max}) & \text{for} & a_k \le \phi_k(r_k) \\ \min(\phi_k(r_k), E_{\max}) & \text{for} & \phi_k(r_k) < a_k \le \phi_k(r_k) + d_k \\ \min(a_k - d_k, E_{\max}) & \text{for} & \phi_k(r_k) + d_k < a_k \end{array}$$

In effect, $\phi_k(r_k)$ is a threshold beyond which the reward for consuming exceeds the reward for saving.

B. Computation of the Value Function

The concavity of $\overline{J}_k(a_k)$ not only dictates an optimal policy, but also can be exploited to quickly calculate the value function itself. Two different methods have been developed to do so. The first method is based on the fact that knowing $\phi_k(r_k)$ eliminates the need to maximize over consumption in equation (4). Moreover, $\phi_k(r_k)$ is independent of the demand and available energy. Because of this, the expectation of the value function over d_k becomes similar to a convolution while r_k is held fixed. It is only necessary to weigh and sum over r_k to get the expectation over r_k and complete the calculation for $\overline{J}_k(a_k)$.

Using this strategy, the expected value function can be expressed as

$$\overline{J}_k(a_k) = E_{r,d}[J_k(a_k, r_k, d_k)] = \sum_{r_k=0}^{\infty} p_r(r_k) E_d[J_k(a_k, r_k, d_k)|r_k] \quad (8)$$

where a_k, r_k, and d_k are taken as discrete and integer for the purposes of computation.

It can be shown that whenever $a_k < \phi_k(r_k)$,

$$E_d[J_k(a_k, r_k, d_k)|r_k] = E_{r,d}[J_k(a_k, r_k, d_k)] = \overline{J}_{k+1}(\min(a_k, E_{\max}) + b_{k+1}) \quad (9)$$

and when $a_k \ge \phi_k(r_k)$,

$$\begin{aligned} &E_d[J_k(a_k, r_k, d_k)|r_k] \\ &= \sum_{d_k=0}^{a_k - \phi(r_k)} p_{d_k}(d_k) \overline{J}_{k+1}(\min(a_k - d_k, E_{\max}) + b_{k+1}) \\ &\quad + [1 - \overline{F}_{d_k}(a_k - \phi_k(r_k))] \\ &\quad \cdot [r_k(a_k - \phi_k(r_k)) + \overline{J}_{k+1}(\min(\phi_k(r_k), E_{\max}))] \\ &\quad + r_k \overline{F}_{d_k}(a_k - \phi_k(r_k)) \end{aligned} \quad (10)$$

where $\overline{F}_{d_k}$ is defined as

$$\overline{F}_{d_k}(x) = \sum_{d_k=0}^{x} p_{d_k}(d_k)$$

In practice this method is frequently able to obtain a dramatic improvement in computation speed over the standard dynamic programming algorithm, in some cases over two orders of magnitude.

The second method of calculating expected value is frequently even faster than the one detailed above. The basis of this method is a change in the order of summation and the concavity properties explored earlier. The algorithm essentially chooses the maximum of either the expected marginal reward from saving or from consuming for each incremental unit of energy it is able to use.

It can be shown that for $a_k > 0$, the expected value function $\overline{J}_k(a_k)$ is equal to:

$$\overline{J}_k(a_k) = \hat{J}_{k+1}(a_k) + \sum_{c_k=1}^{a_k} F_{d_k}(c_k) G_k(a_k - c_k) \quad (11)$$

and for $a_k = 0$,

$$\overline{J}_k(0) = \hat{J}_{k+1}(0) \quad (12)$$

where $\hat{J}_{k+1}(a_k)$ is the expected future reward if the current consumption is set to zero. More precisely,

$$\begin{aligned} \hat{J}_{k+1}(a_k) = &E_{(r,d)}[J_{k+1}(\min(E_{\max}, a_k) \\ &+ b_{k+1}, r_{k+1}, d_{k+1})] \end{aligned} \quad (13)$$

Also, $F_{d_k}(x)$ and $G_k(x)$ are defined by

$$F_{d_k}(x) = \sum_{d_k=x}^{\infty} p_{d_k}(d_k)$$

$$G_k(x) = e_{r_k}(\lceil \hat{J}'_{k+1}(x) \rceil) - \hat{J}'_{k+1}(x) F_{r_k}(\lceil \hat{J}'_{k+1}(x) \rceil)$$

where $\lceil \cdot \rceil$ is the ceiling operator that rounds up, and

$$F_{r_k}(x) = \sum_{r_k=x}^{\infty} p_{r_k}(r_k)$$

$$e_{r_k}(x) = \sum_{r_k=x}^{\infty} r_k p_{r_k}(r_k)$$

and $\hat{J}'_{k+1}(x)$ is the first difference of $\hat{J}_{k+1}(x)$, defined by

$$\hat{J}'_{k+1}(x) = \hat{J}_{k+1}(x+1) - \hat{J}_{k+1}(x) \quad (14)$$

The above equations may appear complicated, but are relatively easy to evaluate numerically. Note that $F_{r_k}(x)$ and $e_{r_k}(x)$ do not change unless the probability distributions for r_k change with time. For problems with unchanging probability distributions, this algorithm is even faster than the first method detailed above. While both algorithms must loop over a_k and k, the first method must also loop over r_k and sum over d_k, while the second method only needs to sum over c_k.

C. *Certainty Equivalent Policies*

Certainty equivalent (CEQ) control is a heuristic policy that at each stage applies a decision that would have been optimal if the future rewards r_k and demands d_k were all deterministic and equal to their expectations $E[r_k]$ and $E[d_k]$, respectively. As seen above, dynamic programming requires taking expectations over random variables. This process is computation intensive and can be extremely slow. In the certainty equivalent heuristic, the decision at each stage is found by solving a much easier deterministic problem.

The dynamic programming recursion for the deterministic problem underlying the CEQ policy is given by

$$\begin{aligned} \overline{J}_k(a_k) = \max_{0 \le s_k \le \min(a_k, E_{\max})} &\{E[r_k] \min(a_k - s_k, E[d_k]) \\ &+ \overline{J}_{k+1}(s_k + b_{k+1})\} \end{aligned} \quad (15)$$

and

$$\overline{J}_n(a_n) = E[r_n] \min(a_n, E[d_n]) \quad (16)$$

Once the value functions $\overline{J}_k(a_k)$ are available, a decision at time $k < n-1$ is obtained by setting $c_k = a_k - s_k$, where s_k is the maximizing value in the expression

$$\begin{aligned} \max_{0 \le s_k \le \min(a_k, E_{\max})} &\{r_k \min(a_k - s_k, d_k) \\ &+ \overline{J}_{k+1}(s_k + b_{k+1})\} \end{aligned} \quad (17)$$

The decision at time n is set to $c_n = a_n$.

In the special case where rewards in each time step are independent and identically distributed (i.e., $p_{r_k}(r_k) = p_r(r)$ and $E[r_k] = E[r]$ is the same for all k), the certainty equivalent value function takes on a particularly simple form, and the resulting consumption policy is relatively easy to analyze:

<u>Theorem 2</u>:

Assume that $E[r_k]$ is the same for all k. Then, the value function $\overline{J}_k(a_k)$ for the underlying deterministic problem is of the form

$$\overline{J}_k(a_k) = E[r][\min(a_k, \delta_k) + \gamma_k] \quad (18)$$

where δ_k and γ_k are some constants and $\delta_k \ge E[d_k]$.

<u>Proof</u>:

Consider the underlying deterministic problem. Since the (expected) reward is the same at all times, an optimal policy is to consume as much as possible at all times, and $\overline{J}_k(a_k)$ is equal to $E[r]$ times the total consumption (in the deterministic problem) over the entire horizon. Let $\gamma_k =$

$\overline{J}_k(0)/E[r]$. As a_k increases from 0, each additional unit of available energy will be eventually consumed, and the total reward increases linearly. However, once a_k reaches a certain threshold value δ_k, any additional available energy will have to be wasted and will not result in any additional reward. The fact that $\delta_k \geq E[d_k]$ is immediate because any available energy up to $E[d_k]$ can be profitably consumed at time k and will not be wasted.

As seen by the preceding proof, the quantities γ_k and δ_k have an intuitive interpretation that results in recursive formulas for computing these constants. Indeed, assume that γ_{k+1} and δ_{k+1} have already been determined. We then have

$$\begin{aligned}\gamma_k =&\overline{J}_k(0)/E[r]\\ =&\overline{J}_{k+1}(b_{k+1})/E[r]\\ =&\min(b_{k+1},\delta_{k+1})+\gamma_{k+1}\end{aligned} \tag{19}$$

To determine δ_k, we need to determine the maximum possible available energy a_k that will not be wasted. The first $E[d_k]$ units are not wasted because they can be consumed immediately. Any further useful available energy cannot exceed $E_{\max}$, since this the most that can be conserved for future use. At the next time, the maximum useful available energy is δ_{k+1}. Since there will be a fresh supply of b_{k+1} units, any useful transfer from time k is limited to $\max(\delta_{k+1} - b_{k+1}, 0)$. Putting everything together, we obtain

$$\delta_k = E[d_k] + \min\{E_{\max}, \max(0, \delta_{k+1} - b_{k+1})\} \tag{20}$$

The CEQ policy is determined by using the special form of the value function in equation (17), to obtain

$$\begin{aligned}\max_{0\leq s_k\leq \min(a_k,E_{\max})}&\{r_k \min(a_k - s_k, d_k)\\ &+E[r][\min(s_k+b_{k+1},\delta_{k+1})+\gamma_{k+1}]\end{aligned} \tag{21}$$

If $r_k > E[r]$, the algorithm will consume as much as possible (up to d_k) and then save any remaining energy. If $r_k < E[r]$, the algorithm will save as much as possible, up to $\delta_{k+1} - b_{k+1}$ units of energy, and try to consume the rest. This policy appears to be a reasonable one, and in tests the CEQ algorithm regularly obtained 80% to 90% of the optimal reward.

D. Unlimited Demand Policy

When demand is unlimited one can obtain a closed-form expression for an optimal consumption policy, described by a simple threshold scheme. This formulation also applies to the case where demand is finite but is guaranteed to always exceed the available energy. This policy can be used as a heuristic to solve the general demand-limited case.

As before, the objective is to choose a consumption policy that maximizes total expected reward over n time steps. Since demand is unlimited, the dynamic programming recursion becomes:

$$\begin{aligned}J_k(a_k,r_k) = &\max_{0\leq c_k\leq a_k}\{r_k c_k\\ &+E[J_{k+1}(\min(a_k-c_k,E_{\max})+b_{k+1},r_k)]\}\end{aligned} \tag{22}$$

For $1 \leq i \leq j \leq n$, define the constants

$$\begin{aligned}\alpha_j^j &= E[r_j]\\ \alpha_j^i &= E[\max(r_i, \alpha_j^{i+1})]\\ \beta_j^j &= E_{\max}\\ \beta_j^i &= \max(\beta_j^{i+1} - b_i, 0)\end{aligned}$$

<u>Theorem 3</u>:

An optimal consumption policy, for $1 \leq k < n$, is given by the following: If $r_k \geq \alpha_n^{k+1}$, then

$$c_k = a_k \tag{23}$$

Otherwise,

$$c_k = \max(a_k - \beta_j^{k+1}, 0) \tag{24}$$

where j is the smallest j in the range in the range $k+1 \leq j \leq n$ such that $r_k < \alpha_j^{k+1}$.

Furthermore, the value function is given by

$$\begin{aligned}&J_k(a_k,r_k) =\\ &\quad[\max(r_k,\alpha_n^{k+1}) - \max(r_k,\alpha_{n-1}^{k+1})]\cdot[\min(\beta_n^{k+1},a_k)]\\ &\quad+[\max(r_k,\alpha_{n-1}^{k+1}) - \max(r_k,\alpha_{n-2}^{k+1})]\cdot[\min(\beta_{n-1}^{k+1},a_k)]\\ &\quad\quad\vdots\\ &\quad+[\max(r_k,\alpha_{k+2}^{k+1}) - \max(r_k,\alpha_{k+1}^{k+1})]\cdot[\min(\beta_{k+2}^{k+1},a_k)]\\ &\quad+[\max(r_k,\alpha_{k+1}^{k+1}) - r_k]\cdot[\min(\beta_{k+1}^{k+1},a_k)]\\ &\quad+r_k a_k+\omega\end{aligned} \tag{25}$$

where ω is a constant (the actual value of which does not affect policy).

The physical intuition behind the constants above is as follows: α_j^i represents the optimal expected reward in an optimal stopping problem in which there is a unit of energy that can be consumed at any time $i, i+1, \ldots j$ between stages i and j. (The reward r_k for any given time step is not known until the time step is reached, but the probability

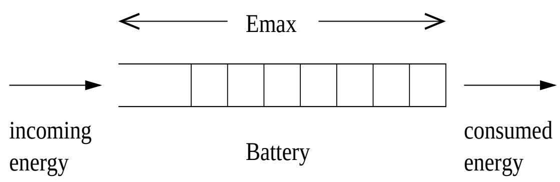

Fig. 2. Battery FIFO Queue

distribution for reward is known for each time.) Notice that for a given i, α_j^i is non-decreasing with j.

The constant β_j^{i+1} represents $E_{\max}$ less the incoming energy $b_{i+1}+\cdots+b_{j-1}$ between time $i+1$ and time $j-1$, as long as it does not become negative. Notice that β_j^{i+1} is non-increasing with j. It is interpreted as the amount of energy at time i that can be saved until time j, without overflowing the battery, in view of the future energy inputs $b_{i+1}, \ldots, b_{j-1}$.

The policy can be interpreted as follows: If the current reward r_k is greater than the expected reward for consuming at an optimally chosen time between time $k+1$ and time n, then the policy consumes all available energy immediately. In other words, if the expected reward for saving is less than the reward for consuming, the policy consumes.

If not, the policy finds the smallest time j such that current reward is less than the expected reward given that the user must consume between time $k+1$ and time j. The policy then consumes available energy less β_j^{k+1} (subject to the constraint that consumption cannot go below zero). Note that in all instances the policy consumes any energy that cannot be saved in the battery.

This closed form solution has an execution time dependent only on the number of stages n and the number of possible values for the rewards r_j.

<u>Proof of Theorem 3</u>:

The theorem can be verified through tedious algebraic manipulation of equation (22). However, there is another approach that is more intuitive. Notice that it is never optimal to save more energy than the battery capacity. Any amount of saved energy greater than the battery capacity is wasted, whereas one can always obtain some reward (however minimal) by consuming, since demand is unlimited.

With this observation in mind, let us consider the battery as a queue for energy packets with a capacity of $E_{\max}$. Assume without loss of generality that each energy packet is of size one. At each time k, b_k energy packets arrive, and the satellite can "service" any number of energy packets in the queue to obtain r_k units of reward per unit energy. The task is to find the service policy that generates the greatest expected reward.

Now consider the class of first-in-first-out (FIFO) policies for managing this queue. First, notice that any energy packet in the queue must be serviced or discarded as soon as $E_{\max}$ additional energy packets arrive after it. If the energy packet is not serviced, queue capacity is exceeded and the energy packet will be wasted.

Since the schedule for energy packet arrivals is known, each energy packet in this queue has an effective expiration time. The expiration time for each energy packet is the time at which a total of $E_{\max}$ additional energy packets arrive after it. Under an optimal policy, the energy packet must be serviced by this time. Note that as one moves from the head of the queue to the end of the queue, the time until expiration for each energy packet is non-decreasing.

Given these expiration times, an optimal FIFO policy simply picks the best time between the current time and the expiration time of the energy packet to service it. This involves solving an optimal stopping problem for each energy packet.

The solution to the optimal stopping problem is well known: For an energy packet with expiration time j, an optimal strategy is to compare current reward r_k with α_j^{k+1}. If $r_k < \alpha_j^{k+1}$ the satellite should save the energy packet; if not, it consumes the energy packet. If the satellite consumes an energy packet with expiration time j, it also will want to consume all energy packets with expiration times before j. At time k, the number of energy packets with expiration time before j is given by $\max(a_k - \beta_j^{k+1}, 0)$. This leads us to the optimal policy described above.

Since the time until expiration is shorter as one moves toward the head of the queue, the satellite will always service energy packets according to FIFO ordering. We have thus obtained an optimal FIFO policy for servicing energy packets. Finally, note that because the energy packets are indistinguishable, an optimal FIFO policy is also an optimal policy in general.

IV. Example: A Low Earth Orbit Satellite

Three procedures for allocating energy have been introduced: the optimal algorithm for the general case, the certainty equivalent method, and the optimal algorithm for the unlimited demand case, which can be used as a heuristic for the general case. We now apply these three procedures to a hypothetical satellite in low earth orbit and compare their performance to a simple greedy algorithm that expends as much energy as it can - $\min(a_k, d_k)$ units of energy - during each time step.

The objective of the algorithms is to maximize total reward obtained over a 24 hour time period, which is divided into 15 minute time slots. The hypothetical satellite has a

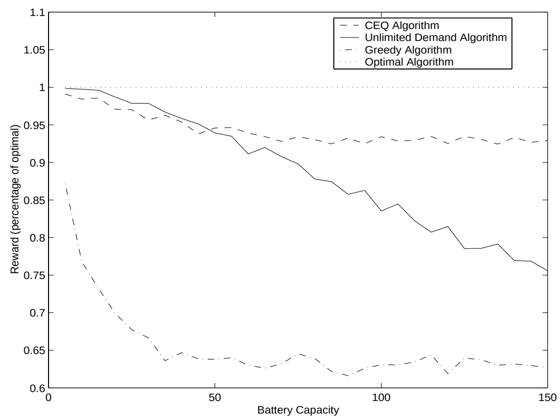

Fig. 3. Performance of Algorithms as a Function of Battery Capacity, $\lambda = 15$

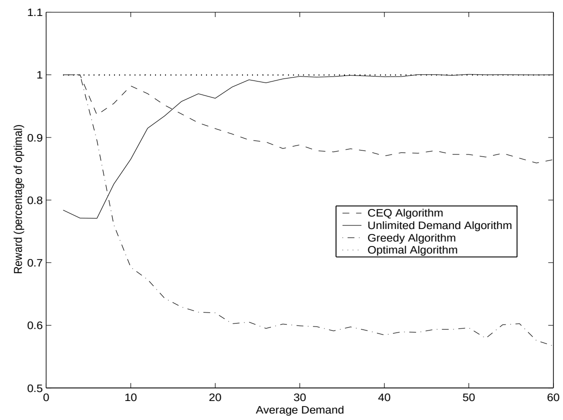

Fig. 5. Performance of Algorithms as a Function of λ (Average Demand)

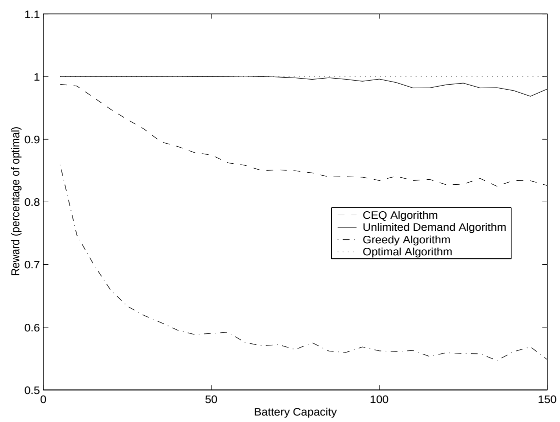

Fig. 4. Performance of Algorithms as a Function of Battery Capacity, $\lambda = 50$

90 minute orbital period, half of which is spent in sunlight, half in darkness. Accordingly, the satellite sees a pattern of three time slots with incoming energy, followed by three time slots without. The satellite starts with 10 units of energy and receives 10 units of energy from its solar cells during each time slot it is in sunlight.

At each time slot k, the satellite can expend up to d_k units of energy for r_k units of reward per unit energy. The demand d_k is Poisson distributed with parameter λ, and the reward r_k has a discrete uniform distribution between 1 and 50.

Figures 3 and 4 show the performance of the algorithms in garnering reward as battery capacity changes from 5 to 150, with $\lambda = 15$ and $\lambda = 50$ respectively. Figure 5 shows the performance of the policies resulting from each algorithm as λ changes from 2 to 60 and for a fixed battery capacity of 50 energy units. In each figure, every data point is the average performance observed in 50 simulations of a policy over the 24-hour horizon. The reward obtained by each policy is plotted as a fraction of the reward obtained by the policy resulting from the optimal algorithm.

As can be seen from the figures, the three algorithms introduced earlier significantly outperformed the greedy algorithm. The certainty equivalent heuristic always obtained at least 80% of the optimal reward, while the unlimited demand heuristic was always above 70%. Figure 5 also shows that the unlimited demand policy performed particularly well when the average demand was relatively high. Also notice from figures 3 and 4 that the performance of every suboptimal algorithm deteriorated as battery capacity increased. The explanation is that a larger battery leads to more choices as to when to consume energy, which the heuristics do not handle as well as the optimal algorithm. In contrast, when the battery capacity was small, all algorithms performed similarly, as the opportunity to save energy was limited by the battery capacity.

Note that while the plots show the relative performance of the greedy algorithm deteriorating with increasing battery capacity and increasing demand, the total rewards obtained by the greedy algorithm actually remained fairly constant. It is easy to see that increasing battery capacity would have little impact on the total reward obtained by the greedy algorithm, which stores as little energy as possible. Similarly, the greedy algorithm would not be able to take advantage of increased demand levels by saving energy for future, higher reward opportunities. Hence, the deteriorating relative performance of the greedy algorithm in the simulation was due mainly to the increased reward obtained by the other algorithms, which were able to exploit higher battery capacity and demand levels in making consumption decisions.

The algorithms were run on a Pentium III computer using Matlab 5.0. Computing value functions and policies for a typical data point from figure 5 required roughly .92 seconds when using the second method for calculating an optimal value function (equation (11)). The unlimited demand algorithm required .51 seconds and the CEQ algorithm .39 seconds. In contrast, the greedy algorithm required only .006 seconds, while a direct calculation of the optimal value function required about 26 minutes, 39 seconds.

V. Other Applications

The algorithms and analysis presented above are applicable in many situations where there is a stored resource that can be expended for a reward. For instance, the operator of a hydroelectric dam with a limited supply of water could use a similar algorithm to maximize revenue when faced with a fluctuating price for power.

One particularly interesting application is that of maximizing throughput in a fading channel given finite battery capacity. Assume that a mobile transmitter seeks to transmit over a fading channel where throughput per unit energy expended is not known until the time of transmission. The probability density of the throughput is independently distributed over time and known. We also impose a power limit on the transmitter and a deadline by which the transmission must take place.

This application gives rise to two problems that can be solved using the approach described in this paper. First, one may seek to maximize expected total throughput during a fixed time period and given a limited amount of energy. Second, one may seek to minimize the energy expected to be consumed given a fixed amount of data to send during a fixed time period.

For the first problem, if throughput is seen as a reward rate and power limit seen as demand, the resulting formulation is almost identical to the satellite energy allocation problem. There are only two places where the problems differ. First, energy inputs for the mobile transmitter are zero for all time. Second, in most cases power constraints will be static and known a priori. These two conditions will tend to significantly simplify calculations; nevertheless, the algorithms detailed above will be completely applicable. In particular, note that the unlimited power/demand algorithm degenerates to an optimal stopping problem.

The second problem can be solved with techniques similar to the ones used for the first problem; however, the problem is a minimization rather than a maximization, and some modification of our approach will be necessary.

VI. Conclusion

This paper developed a dynamic programming formulation for optimizing satellite energy allocation and presented three methods for efficiently obtaining a solution. The three methods trade off computational complexity with optimality and their performance and properties have been analyzed. The approach developed is general and can be used for other stored resource allocation problems, including throughput maximization for wireless communications.

There are a number of areas for further investigation. The algorithms and policies presented thus far are valid only for a single satellite. Additional work needs to be done on extending the results to a constellation of satellites. It would also be interesting to explore the use of these algorithms as a satellite design tool rather than as an aid to operation. Because the algorithms run quickly on a computer, it is easy to see the effects of a reduction in battery capacity or an increase in average demand. Another natural extension of our model would be to capture battery charge/discharge effects, as discussed earlier. Finally, it would be interesting to investigate the use of extremely short time steps: the algorithms could be used to decide whether to accept or reject individual packets.

Appendix

A. Proof of Theorem 1: Concavity of the Value Function

The dynamic programming equations for stochastic reward and stochastic demand energy allocation are given by

$$\begin{aligned} J_k(a_k, r_k, d_k) = &\max_{0 \le s_k \le \min(a_k, E_{\max})} \{ r_k \min(a_k - s_k, d_k) \\ &+ E_{(r_{k+1}, d_{k+1})}[J_{k+1}(s_k + b_{k+1}, r_{k+1}, d_{k+1})] \} \end{aligned} \tag{26}$$

and

$$J_n(a_n, r_n, d_n) = r_n \min(a_n, d_n) \tag{27}$$

We now show that $J_k(a_k, r_k, d_k)$ is concave with respect to a_k, for every r_k and d_k.

<u>Definition</u>:

A function $f : \Re \to \Re$ is concave if for $0 \le \lambda \le 1$ and $\lambda + \overline{\lambda} = 1$ we have

$$f(\lambda y + \overline{\lambda} z) \ge \lambda f(y) + \overline{\lambda} f(z) \tag{28}$$

for all $y, z \in \Re$.

<u>Lemma 1</u>:

If f and g are concave and $\alpha \ge 0$, then $f + g$ and αf are concave.

<u>Proof</u>:

Follows from definition of concavity.

Lemma 2:

If $0 \le \lambda \le 1$ and $\lambda + \overline{\lambda} = 1$, then

$$\lambda \min(a,b) + \overline{\lambda} \min(c,b) \le \min(\lambda a + \overline{\lambda} c, b) \tag{29}$$

Proof:

For fixed b, the function $\min(a,b)$ is a concave function of a and the result follows.

Theorem:

$J_k(a_k, r_k, d_k)$ is concave in a_k for any fixed r_k and d_k.

Proof:

We use induction. First, note that the value function $J_n(a_n, r_n)$ is concave in a_n and the expected value function $E[J_n(a_{n-1} + b_n, r_n)]$ is concave in a_{n-1}. Indeed, from the problem formulation, we see that

$$J_n(a_n, r_n, d_n) = r_n \min(a_n, d_n)$$

is a piecewise linear and concave function of a_n. $J_n(a_{n-1} + b_n, r_n, d_n)$ is concave in a_{n-1} as well, and by lemma 1, the expectation $E_{r,d}[J_n(a_{n-1} + b_n, r_n, d_n)]$ is also concave to a_{n-1} since it is a weighted sum of concave functions.

Now assume $E_{r,d}[J_{k+1}(a_k + b_{k+1}, r_{k+1}, d_{k+1})]$ is concave in a_k. We show that $J_k(a_k, r_k, d_k)$ is concave to a_k. To complete the induction, we also show that $E_{r,d}[J_k(a_{k-1} + b_k, r_k, d_k)]$ is concave in a_{k-1}.

Let us look at $J_k(x, r_k, d_k)$ and $J_k(y, r_k, d_k)$. We have

$$\begin{aligned} J_k(x,r_k,d_k) = &\max_{0 \le s_k \le \min(x, E\max)} \{r_k \min(x - s_k, d_k) \\ &+ E_{r,d}[J_{k+1}(s_k + b_{k+1}, r_{k+1}, d_{k+1})]\} \end{aligned}$$

There must be an optimizing value for s_k. Denote this by s_k^x. Then

$$\begin{aligned} J_k(x,r_k,d_k) = &r_k \min(x - s_k^x, d_k) \\ &+ E_{r,d}[J_{k+1}(s_k^x + b_{k+1}, r_{k+1}, d_{k+1})] \end{aligned}$$

Similarly,

$$\begin{aligned} J_k(y,r_k,d_k) = &r_k \min(y - s_k^y, d_k) \\ &+ E_{r,d}[J_{k+1}(s_k^y + b_{k+1}, r_{k+1}, d_{k+1})] \end{aligned}$$

where s_k^y is an optimizing value for s_k in the equation for $J_k(y, r_k, d_k)$. Combining the two equations and weighting by λ or $\overline{\lambda}$,

$$\begin{aligned} &\lambda J_k(x, r_k, d_k) + \overline{\lambda} J_k(y, r_k, d_k) \\ &\quad = \lambda\{r_k \min(x - s_k^x, d_k) \\ &\qquad + E_{r,d}[J_{k+1}(s_k^x + b_{k+1}, r_{k+1}, d_{k+1})]\} \\ &\qquad + \overline{\lambda}\{r_k \min(y - s_k^y, d_k) \\ &\qquad + E_{r,d}[J_{k+1}(s_k^y + b_{k+1}, r_{k+1}, d_{k+1})]\} \\ &\quad = r_k(\lambda \min(x - s_k^x, d_k) + \overline{\lambda} \min(y - s_k^y, d_k)) \\ &\qquad + \lambda E_{r,d}[J_{k+1}(s_k^x + b_{k+1}, r_{k+1}, d_{k+1})] \\ &\qquad + \overline{\lambda} E_{r,d}[J_{k+1}(s_k^y + b_{k+1}, r_{k+1}, d_{k+1})]\} \end{aligned}$$

The terms $\min(x - s_k^x, d_k)$ and $\min(y - s_k^y, d_k)$ are piecewise linear and concave. By the induction hypothesis, we also know that $E_{r,d}[J_{k+1}(s_k^x + b_{k+1}, r_{k+1}, d_{k+1})]$ and $E_{r,d}[J_{k+1}(s_k^y + b_{k+1}, r_{k+1}, d_{k+1})]$ are concave in s_k. Then

$$\begin{aligned} &\lambda J_k(x,r_k, d_k) + \overline{\lambda} J_k(y, r_k, d_k) \\ &\qquad \le r_k \min(\lambda x + \overline{\lambda} y - \lambda s_k^x - \overline{\lambda} s_k^y, d_k) \\ &\qquad + E_r[J_{k+1}(\lambda s_k^x + \overline{\lambda} s_k^y + b_{k+1}, r_{k+1}, d_{k+1})] \end{aligned}$$

Now examine the range of the maximization. Since $s_k^x \le \min(x, E_{\max})$ and $s_k^y \le \min(y, E_{\max})$,

$$\lambda s_k^x + \overline{\lambda} s_k^y \le \lambda x + \overline{\lambda} y$$

and

$$\lambda s_k^x + \overline{\lambda} s_k^y \le \lambda E_{\max} + \overline{\lambda} E_{\max}$$

Combining,

$$\lambda s_k^x + \overline{\lambda} s_k^y \le \min(\lambda x + \overline{\lambda} y, E_{\max})$$

and

$$\begin{aligned} &\lambda J_k(x, r_k, d_k) + \overline{\lambda} J_k(y, r_k, d_k) \\ &\quad \le \max_{0 \le s_k \le \min(\lambda x + \overline{\lambda} y, E_{\max})} \{r_k \min(\lambda x + \overline{\lambda} y - s_k, d_k) \\ &\qquad + E_{r,d}[J_{k+1}(s_k + b_{k+1}, r_{k+1}, d_{k+1})]\} \\ &\quad = J_k(\lambda x + \overline{\lambda} y, r_k, d_k) \end{aligned} \tag{30}$$

This shows that $J_k(a_k, r_k, d_k)$ is concave in a_k. A direct application of lemma 1 shows that $E_{r,d}[J_k(a_{k-1} + b_k, r_k, d_k)]$ is also concave in a_{k-1} and the induction is complete.

References

[1] J.M. Aein, O.S. Kosovych, "Satellite capacity allocation," *Proceedings of the IEEE*, vol. 65, no. 3, pp. 332-42, March 1977.

[2] H.O. Awadalla, L.G. Cuthbert, J.A. Schormans, "Predictive resource allocation for real time video traffic in broadband satellite networks," in *Proceedings of the Fifth International Conference*

on Broadband Communications. (Hong Kong, People's Republic of China, November 10-12, 1999).

[3] D. Berman, *The Manned Space Station Power System: An Operational Scheduler*. Master's thesis, Massachusetts Institute of Technology, 1986.

[4] D. Bertsekas, *Dynamic Programming and Optimal Control*. Belmont, MA: Athena Scientific, 1995.

[5] C.F. Chiasserini, R.R. Rao, "Energy efficient battery management," in *Infocom 2000 Proceedings*. (Tel-Aviv, Israel, 2000, vol. 2, pp. 396-403).

[6] C.F. Chiasserini, R.R. Rao, "Improving battery performance by using traffic shaping techniques," *IEEE Journal on Selected Areas in Communications*, vol. 19, no. 7, pp. 1385-1394, July 2001.

[7] R. C. Collette and B. L. Herdan, "Design problems of spacecraft for communication missions," *Proceedings of the IEEE*, vol. 65, no.3, pp. 342-356, March 1977.

[8] R. Kraus, T. Hendricks, "Satellite solar power analysis and sizing model," in *Proceedings of the 26th Intersociety Energy Conversion Engineering Conference*. (La Grange Park, IL, 1991, vol. 1, pp. 195-200).

[9] I. Koutsopoulos, L. Tassiulas, " A unified framework for handover prediction and resource allocation in non-geostationary mobile satellite networks," in *IEEE VTS 50th Vehicular Technology Conference.* (1999, vol. 4, pp. 2106-10).

[10] G. Maral, M. Bousquet, *Satellite Communications Systems*, 3rd ed. New York, NY: John Wiley and Sons, 1998.

[11] E. Papapetrou, I. Gragopoulos, F. Pavlidou, "Performance evaluation of LEO satellite constellations with inter-satellite links under self-similar and poisson traffic," *International Journal of Satellite Communications*, vol. 17, pp. 51-64, 1999.

[12] P.D. Shaft, "Unconstrained allocation of communication satellite traffic," in *1977 International Conference on Communications*. (New York, NY, 1977, part 3, pp. 26-30).

[13] M. Werner, J. Frings, et.al., "Topological design, routing, and capacity dimensioning for ISL networks in broadband LEO satellite systems," *International Journal of Satellite Communications*, to appear.

[14] A. Ween, et.al., "Dynamic resource allocation for multi-service packet based LEO satellite communications," in *IEEE GLOBECOM 1998*. (1998, vol. 5, pp. 2954-9).