Optimal Energy Allocation and Admission Control for Communications Satellites

Alvin C. Fu, *Student Member, IEEE*, Eytan Modiano, *Senior Member, IEEE*, and John N. Tsitsiklis, *Fellow, IEEE*

***Abstract*—We address the issue of optimal energy allocation and admission control for communications satellites in earth orbit. Such 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.**

***Index Terms*—Communication, dynamic programming, resource allocation, satellite.**

I. Introduction

FOR MOST satellites, energy management is a critical issue, for the simple reason that energy efficiency 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 providing packet data services, demand for such 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 [12]. 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 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 circuit-switched and packet-based networks, while Shaft [14] looked at unconstrained allocation of power and gain to service communication satellite traffic. Recently, many researchers have examined the use of satellites to supplement terrestrial data networks [13], [15]. This work primarily focused on design and performance evaluation of such space networks, but little attention was paid to energy allocation issues. Perhaps the closest study to our current work is one by Ween *et al.* [16], who investigated resource allocation for low-earth-orbit satellites providing GSM cellular services. Resource allocation for satellite beams and path selection has been studied in [11], and the allocation of bandwidth was examined in [2].

Manuscript received May 15, 2002; revised November 6, 2002; approved by IEEE/ACM TRANSACTIONS ON NETWORKING Editor M. Zukerman. This work was supported by the Defense Advanced Research Projects Agency under the Next Generation Internet Initiative.

A. C. Fu is with the Laboratory for Information and Decision Systems, Massachusetts Institute of Technology, Cambridge, MA 02139-4307 USA (e-mail: alfu@mit.edu).

E. Modiano is with the Laboratory for Information and Decision Systems, Massachusetts Institute of Technology, Cambridge, MA 02139-4307 USA (e-mail: modiano@mit.edu).

J. N. Tsitsiklis is with the Massachusetts Institute of Technology, Cambridge, MA 02139-4307 USA (e-mail: jnt@mit.edu).

Digital Object Identifier 10.1109/TNET.2003.813041

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 [10]. 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 depends on 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 that affect the 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 a closed-form optimal solution for 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.

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.

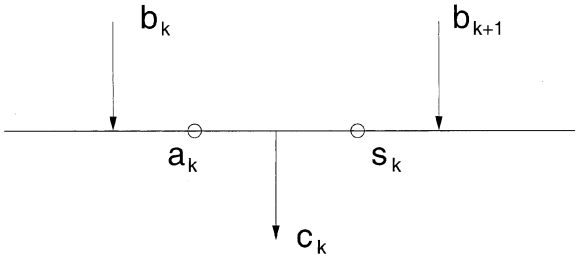

Fig. 1. Energy flow.

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 $u_k = a_k - c_k$ 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

$$\begin{aligned} s_k &= \min(a_k - c_k, E_{\max}) \\ &= \min(u_k, E_{\max}). \end{aligned}$$

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

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

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

$$\begin{aligned} a_{k+1} &= \min(u_k, E_{\max}) + b_{k+1} \\ &= s_k + b_{k+1}. \end{aligned}$$

Each unit of energy consumed provides the satellite operator with a reward r_k. The reward r_k is a nonnegative random variable with a probability distribution $p_{r_k}(\cdot)$ that varies with time. Although $p_{r_k}(\cdot)$ 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 nonnegative random variable with *a priori* known probability distribution $p_{d_k}(\cdot)$, 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 the total expected reward

$$E\left[\sum_{i=1}^{n} r_i c_i\right] \tag{2}$$

over a time horizon of n time steps, subject to demand and energy constraints.

Notice that implicit in (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 average power and energy than steady discharge, 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. However, 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. We also 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. Finally, 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 approach to the problem formulated in the previous section. As usual in dynamic programming, we introduce the value function $J_k(a_k, r_k, d_k)$. This function 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:

$$J_k(a_k, r_k, d_k) = \max_{0 \le c_k \le a_k} \{r_k \min(c_k, d_k) + \bar{J}_{k+1}(\min(a_k - c_k, E_{\max}) + b_{k+1})\} \quad (3)$$

where

$$\bar{J}_k(a_k) = E_{r,d}[J_k(a_k, r_k, d_k)].$$

The maximization is taken over consumed energy c_k and 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 is given by $a_{k+1} = \min(a_k - c_k, E_{\max}) + b_{k+1}$. The expected reward for having this much energy available is given by the expectation $E_{r,d}[J_{k+1}(a_{k+1}, r_{k+1}, d_{k+1})]$, 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 (3). 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 maximizing over the stored energy term $s_k = \min(a_k - c_k, E_{\max})$. Hence, for stage k

$$J_k(a_k, r_k, d_k) = \max_{0 \le s_k \le \min(a_k, E_{\max})} \{r_k \min(a_k - s_k, d_k) + \bar{J}_{k+1}(s_k + b_{k+1})\}. \quad (4)$$

Maximizing over the unused energy term $u_k = a_k - c_k$ gives rise to yet another useful formulation:

$$J_k(a_k, r_k, d_k) = \max_{\max(a_k - d_k, 0) \le u_k \le a_k} \{r_k(a_k - u_k) + \hat{J}_{k+1}(u_k)\} \quad (5)$$

where the term $\hat{J}_k(u)$ is defined as

$$\hat{J}_k(u) = E_{r,d}[J_k(\min(u, E_{\max}) + b_k, r_k, d_k)]. \quad (6)$$

For every formulation, the value function at the final stage, stage n, is given by

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

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 $E_{r,d}[J_k(a_k, r_k, d_k)]$ for every a_k, r_k, and d_k, and all k. In addition, it is necessary to optimize over c_k for each combination of a_k, r_k, and d_k. 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: The expected value function $\bar{J}_k(a_k)$ is concave in a_k as well, since it is a linear combination of concave functions.

Note that the value function can be shown to be concave in d_k and $E_{\max}$ as well.

The concavity properties of the expected value function $\bar{J}_{k+1}(a_{k+1})$ dictate 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. Properly balancing these two functions results in an optimal policy.

We now derive the form of an optimal policy. Let $\phi_k(r_k)$ be a value of u_k that maximizes the expression

$$r_k(a_k - u_k) + \hat{J}_{k+1}(u_k) \tag{7}$$

over all $u_k \geq 0$. In other words

$$\phi_k(r_k) = \arg\max_{u_k \geq 0} \hat{J}_{k+1}(u_k) - r_k u_k. \tag{8}$$

Theorem 2: The choice of

$$u_k = \left\{ \begin{array}{ll} \min(\phi_k(r_k), a_k), & \text{if } a_k < \phi_k(r_k) + d_k \\ a_k - d_k, & \text{if } \phi_k(r_k) + d_k \leq a_k \end{array} \right\}$$

attains the maximum in the right-hand side of (5).

In effect, $\phi_k(r_k)$ is a threshold beyond which the reward for consuming exceeds the reward for saving. It does not depend on the available energy a_k or the demand d_k, and is hence easy to compute.

The proof of Theorem 2 uses the following well-known lemma, which we state without proof.

Lemma 1: If $f(x)$ and $g(x)$ are concave in x, and $f(x)$ is increasing, then $f(g(x))$ is concave in x.

Proof of Theorem 2: It is an immediate consequence of Lemma 1 and Theorem 1 that $\hat{J}_{k+1}(u_k)$ is concave in u_k. Also, (7) is concave in u_k since it is a sum of concave functions. We also notice that the range $u_k \geq 0$ contains the range $\max(a_k - d_k, 0) \leq u_k \leq a_k$.

As a result, an optimizing value of u_k in the right-hand side of (5) is simply $\phi_k(r_k)$ projected on the interval $[\max(a_k - d_k, 0), a_k]$. The theorem follows.

Concavity is also critical in proving the following important property of the value function. □

Theorem 3: If b_k, d_k, and $E_{\max}$ are integer for all k, the value function $J_k(a_k, r_k, d_k)$ for fixed r_k and d_k will be piecewise linear in a_k, with corner points only at integer values of a_k. Furthermore, $\phi_k(r_k)$ can be chosen integer for every k and r_k.

Proof: Given in Appendix B. □

Corollary: If, in addition, the initial energy a_0 is also integer, then there exists an optimal policy under which u_k and a_k are integer for all k.

Proof: We use induction. By Theorem 3, $\phi_k(r_k)$ can be assumed to be integer. Then when a_k is integer, the choice of u_k given by Theorem 2 will be integer as well. As a result, a_{k+1} is also integer.

We have seen that the slope of the value function changes only at integer values of a_k when b_k, d_k, and $E_{\max}$ are integer. As a consequence, a numerical method need only consider integer values of a_k. Therefore, let us assume from now on, 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. □

B. Computation of the Value Function

The concavity of $\bar{J}_k(a_k)$ not only dictates the form of 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 (3). 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 when r_k is held fixed. It is then only necessary to weigh and sum over r_k to get the expectation over r_k and complete the calculation for $\bar{J}_k(a_k)$.

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

$$\begin{aligned} \bar{J}_k(a_k) &= E_{r,d}[J_k(a_k, r_k, d_k)] \\ &= \sum_{r_k=0}^{\infty} p_{r_k}(r_k) E_d[J_k(a_k, r_k, d_k) \,|\, r_k] \end{aligned}$$

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

Whenever $a_k < \phi_k(r_k)$, the optimal consumption is zero (see Theorem 2) and

$$E_d[J_k(a_k, r_k, d_k) \,|\, r_k] = \hat{J}_{k+1}(a_k)$$

where $\hat{J}_k(u)$ is defined in (6).

When $a_k \geq \phi_k(r_k)$, it is proven in [8] that

$$\begin{aligned} &E_d[J_k(a_k, r_k, d_k) \,|\, r_k] \\ &\quad = \sum_{d_k=0}^{a_k - \phi_k(r_k)} \Big\{ p_{d_k}(d_k) \cdot [r_k d_k + \hat{J}_{k+1}(a_k - d_k)] \Big\} \\ &\qquad + \left[\sum_{d_k > a_k - \phi_k(r_k)} p_{d_k}(d_k) \right] \\ &\qquad \cdot [r_k(a_k - \phi_k(r_k)) + \hat{J}_{k+1}(\phi_k(r_k))]. \end{aligned}$$

In our experience, this method often leads to 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 the optimal value function is frequently even faster than the one detailed above. The algorithm relies on the concavity of the value function and 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. The dynamic programming recursion is written in the form

$$J_k(a_k, r_k, d_k) = \max_{0 \leq c_k \leq a_k} \{ r_k \min(c_k, d_k) + \hat{J}_{k+1}(a_k - c_k) \}.$$

It can be shown (see [8]) that

$$\begin{aligned} &J_k(a_k, r_k, d_k) \\ &\quad = \hat{J}_{k+1}(a_k) + \sum_{c_k=1}^{\min(a_k, d_k)} \max(r_k - \hat{J}'_{k+1}(a_k - c_k), 0) \end{aligned} \tag{9}$$

where $\hat{J}'_{k+1}(x)$ is the first difference of $\hat{J}_{k+1}(x)$

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

Note that the term

$$\sum_{c_k=1}^{\min(a_k, d_k)} \max(r_k - \hat{J}'_{k+1}(a_k - c_k), 0)$$

is omitted if $\min(a_k, d_k) = 0$.

Equation (9) is a significant simplification of the earlier dynamic programming formulations from a numerical standpoint.

It replaces the maximization over c_k with the summation of a simple maximum of two quantities. Rather than optimize over all available energy, each incremental unit of energy is allocated by comparing r_k and $\hat{J}'_{k+1}(a_k - c_k)$.

To compute the expected value function, it is necessary to average (9) over r_k and d_k:

$$\bar{J}_k(a_k) = \sum_{d_k=0}^{\infty} p_{d_k}(d_k) \sum_{r_k=0}^{\infty} p_{r_k}(r_k) \left[\hat{J}_{k+1}(a_k) + \sum_{c_k=1}^{\min(a_k,d_k)} \max(r_k - \hat{J}'_{k+1}(a_k - c_k), 0) \right].$$

After applying a change in the order of summation, we have

$$\bar{J}_k(a_k) = \hat{J}_{k+1}(a_k) + \sum_{c_k=1}^{a_k} \left\{ \left[\sum_{d_k=c_k}^{\infty} p_{d_k}(d_k) \right] \cdot \left[\sum_{r_k=0}^{\infty} p_{r_k}(r_k) \max(r_k - \hat{J}'_{k+1}(a_k - c_k), 0) \right] \right\}.$$

This change in the summation order eliminates a minimization in the earlier expressions and eliminates the need to average over d_k. The remaining maximization can be eliminated by noticing that

$$\sum_{r_k=0}^{\infty} p_{r_k}(r_k) \max(r_k - \hat{J}'_{k+1}(a_k - c_k), 0) = \sum_{r_k=\lceil \hat{J}'_{k+1}(a_k - c_k) \rceil}^{\infty} p_{r_k}(r_k)(r_k - \hat{J}'_{k+1}(a_k - c_k))$$

where $\lceil \cdot \rceil$ is the ceiling operator. Thus

$$\bar{J}_k(a_k) = \hat{J}_{k+1}(a_k) + \sum_{c_k=1}^{a_k} \left\{ \left[\sum_{d_k=c_k}^{\infty} p_{d_k}(d_k) \right] \cdot \left[\sum_{r_k=\lceil \hat{J}'_{k+1}(a_k - c_k) \rceil}^{\infty} p_{r_k}(r_k)(r_k - \hat{J}'_{k+1}(a_k - c_k)) \right] \right\}. \tag{10}$$

An efficient computational method readily follows from this representation of $\bar{J}_k(a_k)$. Furthermore, note that if the distribution of r_k or d_k do not change with time, then quantities such as

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

and

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

only need to be computed once, resulting in further reduction in computation time.

C. Certainty Equivalent Policy

A certainty equivalent (CEQ) policy 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 computationally intensive and can be extremely slow. With a 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

$$\tilde{J}_k(a_k) = \max_{0 \le c_k \le a_k} \{E[r_k] \min(c_k, E[d_k]) + \tilde{J}_{k+1}(\min(a_k - c_k, E_{\max}) + b_{k+1})\} \tag{11}$$

and

$$\tilde{J}_n(a_n) = E[r_n] \min(a_n, E[d_n]).$$

Once the value functions $\tilde{J}_k(a_k)$ are available, a decision at time $k \le n-1$ is obtained by maximizing c_k in the expression

$$\max_{0 \le c_k \le a_k} \{r_k \min(c_k, d_k) + \tilde{J}_{k+1}(\min(a_k - c_k, E_{\max}) + b_{k+1})\}. \tag{12}$$

The decision at time n is set to $c_n = \min(a_n, d_n)$.

In the special case where rewards in each time step have the same expected value ($E[r_k] = E[r]$ for all k), the certainty equivalent value function and the resulting policy take on a particularly simple form.

Theorem 4: Assume that $E[r_k] = E[r]$ for all k. Then, the value function $\tilde{J}_k(a_k)$ for the underlying deterministic problem is of the form

$$\tilde{J}_k(a_k) = E[r] \min(a_k, \delta_k) + \gamma_k \tag{13}$$

where

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

and

$$\gamma_k = E[r] \min(b_{k+1}, \delta_{k+1}) + \gamma_{k+1}.$$

Proof: Given in Appendix C. □

Although the formal proof of the theorem is given in the Appendix, a more intuitive justification can be obtained by considering the underlying deterministic problem. Since the (expected) reward is the same at all times, an optimal policy is a greedy policy that consumes as much as possible at all times. Then $\tilde{J}_k(a_k)$ is equal to $E[r]$ times the total consumption (in the deterministic problem) over the entire horizon.

Given this fact, it is possible to infer the structure of the value function. Let $\gamma_k = \tilde{J}_k(0)$. As a_k increases from 0, as long as each additional unit of available energy can be consumed, now or in the future, 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. This happens when the current expected demand $E[d_k]$ has been exceeded, and saving the energy for future use is not possible because either the battery capacity or the future expected demand has been exceeded.

As seen by the preceding argument, the quantities γ_k and δ_k have an intuitive interpretation that results in recursive formulas for computing these constants. The total expected reward given

that the current available energy is zero is given by γ_k. The maximum available energy at time k that can be consumed immediately, or saved and consumed later, is given by δ_k. If the satellite has more available energy, the excess is wasted.

At stage n, it is clear that $\gamma_n = 0$ and $\delta_n = E[d_n]$. The formula for γ_k may be obtained using the fact that $\gamma_k = \tilde{J}_k(0)$:

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

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

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

The consumption policy for the special case where $E[r_k]$ is the same for all k is also relatively straightforward to describe. Expression (12) becomes

$$\begin{aligned}\max_{0\le c_k\le a_k} \{&r_k \min(c_k, d_k) + \gamma_{k+1}\\ &+ E[r]\cdot[\min(\min(a_k - c_k, E_{\max}) + b_{k+1}, \delta_{k+1})].\end{aligned}$$

If $r_k > E[r]$, the CEQ policy will consume as much as possible (up to d_k) and then save any remaining energy. If $r_k < E[r]$, the policy 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 where reward was uniformly distributed (see Section IV) the CEQ policy regularly obtained 80%–90% of the optimal reward.

It is possible, however, to construct examples where the performance of the CEQ policy is arbitrarily bad. For instance, consider the extreme case where demand and battery capacity are unlimited. Suppose there are four possible rewards that appear with equal probability and that are chosen from the set $\{0, R-\epsilon, R+\epsilon, 2R\}$, where $0 < \epsilon < R$. Clearly, the optimal policy is to wait until best reward appears to consume energy. The CEQ policy, on the other hand, will consume all available energy whenever the reward is above R. The difference between the reward obtained by the CEQ policy and an optimal policy can be made arbitrarily large simply by adjusting probabilities and rewards.

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

$$\begin{aligned}&J_k(a_k, r_k)\\ &= \max_{0\le c_k\le a_k} \{r_k c_k + E_r[J_{k+1}(\min(a_k - c_k, E_{\max}) + b_{k+1}, r_k)]\}.\end{aligned} \tag{14}$$

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

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

Theorem 5: An optimal consumption policy, for $1 \le k < n$, is given by the following.

If $r_k \ge \alpha_n^{k+1}$, then

$$c_k = a_k. \tag{15}$$

Otherwise

$$c_k = \max(a_k - \beta_j^{k+1}, 0) \tag{16}$$

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

Furthermore, the value function is given by

$$\begin{aligned}&J_k(a_k, r_k)\\ &\quad= \max\left(r_k, \alpha_n^{k+1}\right)\cdot\left[\min\left(\beta_n^{k+1}, a_k\right)\right]\\ &\qquad+\max\left(r_k, \alpha_{n-1}^{k+1}\right)\cdot\left[\min\left(\beta_{n-1}^{k+1}, a_k\right) - \min\left(\beta_n^{k+1}, a_k\right)\right]\\ &\qquad\vdots\\ &\qquad+\max\left(r_k, \alpha_{k+2}^{k+1}\right)\cdot\left[\min\left(\beta_{k+2}^{k+1}, a_k\right) - \min\left(\beta_{k+3}^{k+1}, a_k\right)\right]\\ &\qquad+\max\left(r_k, \alpha_{k+1}^{k+1}\right)\cdot\left[\min\left(\beta_{k+1}^{k+1}, a_k\right) - \min\left(\beta_{k+2}^{k+1}, a_k\right)\right]\\ &\qquad+r_k\cdot\left[a_k - \min\left(\beta_{k+1}^{k+1}, a_k\right)\right] + \omega\end{aligned} \tag{17}$$

where ω is a constant (the actual value of which does not affect the 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 distribution for the reward is known for each time.) Notice that for a given i, α_j^i is nondecreasing 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 nonincreasing 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.

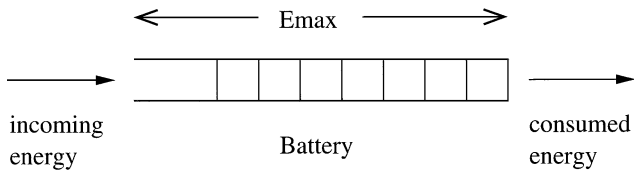

Fig. 2. Battery FIFO queue.

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 $\beta_j^{k+1} \leq E_{\max}$ so that in all instances the policy consumes any energy that cannot be saved in the battery.

This closed-form solution can be computed in time proportional to the number of stages n and the number of possible values for the rewards r_j.

Proof of Theorem 5: The theorem can be verified through tedious algebraic manipulation of (14) [8]. 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 $E_{\max}$ (see Fig. 2). 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 consume any number of energy packets in the queue to obtain r_k units of reward per unit energy. The task is to find the consumption 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 either consumed or discarded by the time $E_{\max}$ additional energy packets arrive after it. If the energy packet is not consumed, 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 any optimal policy, the energy packet must be consumed 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 nondecreasing.

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 consume 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 the 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 $j-1$ or less 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 consume energy packets according to FIFO ordering. We have thus obtained an optimal FIFO policy for consuming energy packets. Finally, note that because the energy packets are indistinguishable, an optimal FIFO policy is also an optimal policy in general.

The value function $J_k(a_k, r_k)$ given in (17) can be better understood by using the terminology developed in the proof and by looking at each individual line of the expression. Each line represents the total reward that can be obtained from all the energy with a certain expiration time. With the exception of the top and bottom lines, each line has the form

$$\max\left(r_k, \alpha_i^{k+1}\right) \cdot \left[\min\left(\beta_i^{k+1}, a_k\right) - \min\left(\beta_{i+1}^{k+1}, a_k\right)\right].$$

The $\max(r_k, \alpha_i^{k+1})$ term represents the expected reward for energy expiring at time i, and the $[\min(\beta_i^{k+1}, a_k) - \min(\beta_{i+1}^{k+1}, a_k)]$ term represents the amount of energy expiring at time i. The top and bottom lines can be similarly approached. For instance, the bottom line of the equation gives the total reward that can be obtained from energy expiring at time k. The reward per unit energy is given by r_k, and the amount of energy expiring at time k is the amount of available energy a_k that exceeds battery capacity $E_{\max}$. This amount of energy is given by

$$a_k - \min(E_{\max}, a_k)$$

or equivalently

$$a_k - \min\left(\beta_{k+1}^{k+1}, a_k\right).$$

Hence, the total reward from this energy is

$$r_k\left[a_k - \min\left(\beta_{k+1}^{k+1}, a_k\right)\right]$$

which is precisely the last line in (17). □

IV. Example: A Low Earth Orbit Satellite

Three procedures for allocating energy have been introduced: the optimal policy for the general case, the certainty equivalent policy, and the optimal policy 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 policy that expends as much energy as it can—$\min(a_k, d_k)$ units of energy—during each time step.

The objective is to maximize total reward obtained over a 24-h time period, which is divided into 15-min time slots. Although we do not do so in this example, it is possible to use much shorter time slots. In fact, it is possible to use our methodology to decide whether to accept or reject individual packets.

The hypothetical satellite has a 90-min 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 20 units of energy and receives 10 units of energy from its solar cells during each time slot it is in sunlight.

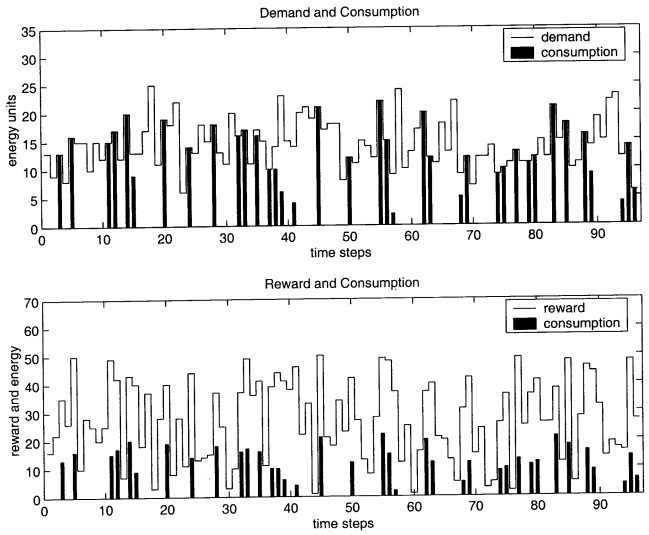

Fig. 3. Reward, consumption, and demand, $\lambda = 15$.

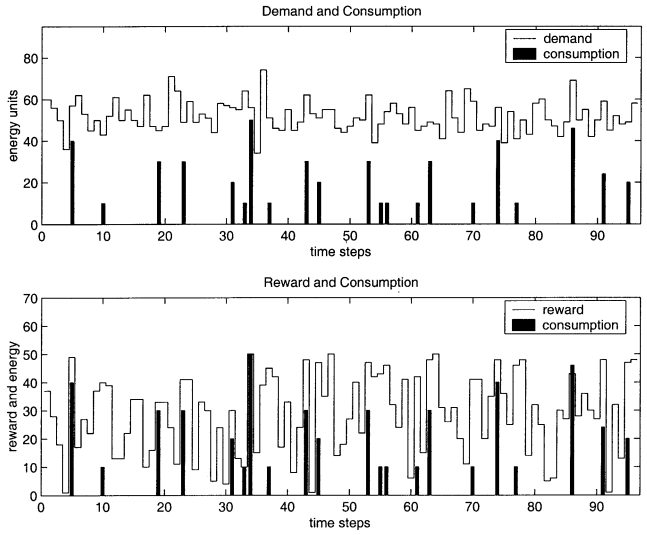

Fig. 5. Reward, consumption, and demand, $\lambda = 50$.

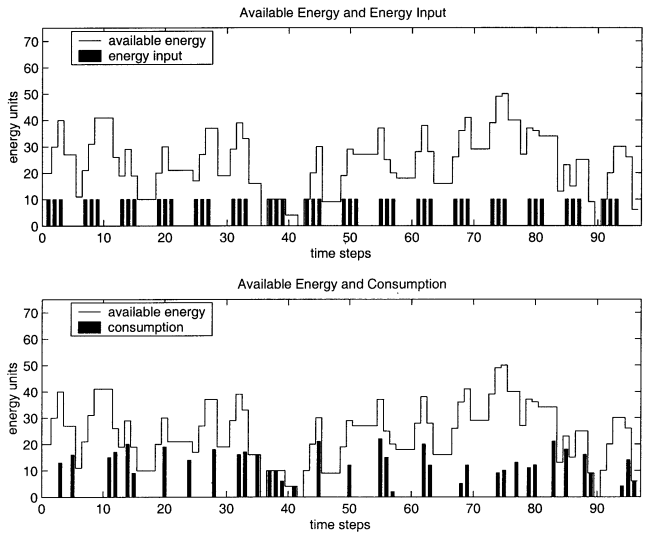

Fig. 4. Energy levels and consumption, $\lambda = 15$.

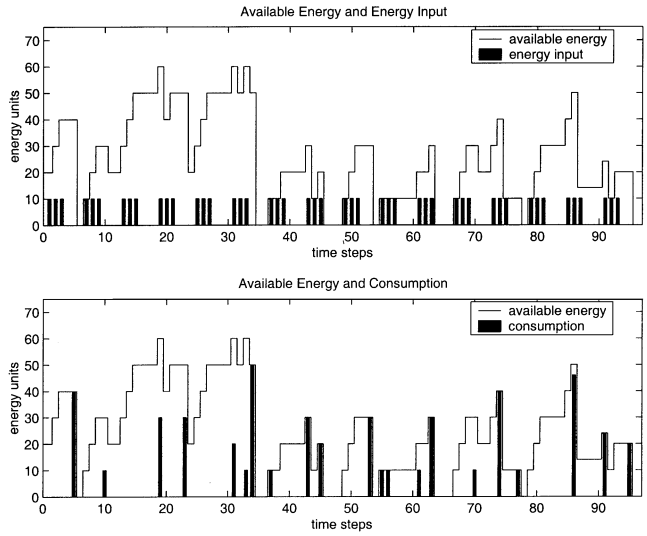

Fig. 6. Energy levels and consumption, $\lambda = 50$.

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.

Fig. 3 shows the reward and demand parameters for a single randomly generated scenario, along with the energy consumption as determined by the optimal policy. Demand was Poisson distributed with $\lambda = 15$, and the battery capacity was 50 energy units. The first plot shows the relationship between consumption and demand, and the second the relationship between reward and consumption. As might be expected, the optimal policy elected to consume only when reward was relatively high. Also, consumption at peak points was often equal to demand—in this particular scenario, the demand was generally lower than available energy. Thus if the policy elected to consume, it was usually constrained by demand, not available energy.

Fig. 4 shows the energy levels of the satellite in the same scenario. The first subplot shows the energy in the battery and the energy input from the solar panels. The oscillations in battery levels that result from periods of light and darkness are readily apparent. The second subplot shows the available energy (battery plus input energy) and consumption. In general, there is much more available energy than demand when the policy elects to consume. In such a situation, we would expect the unlimited demand policy to yield considerably poorer results than the optimal policy.

Fig. 5 shows the reward, demand, and energy consumption for another randomly generated scenario where demand was Poisson distributed with $\lambda = 50$. Fig. 6 shows the energy levels of the satellite in this scenario. Unlike the situation where $\lambda = 15$, the satellite seldom serves all of the available demand. In this case the satellite is energy constrained, not demand constrained. Under this circumstance, it is to be expected that the unlimited demand heuristic would perform well.

The value function (under an optimal policy) and the value functions underlying the unlimited demand and CEQ heuristics for time step 54 are shown in Fig. 7. This particular time step was chosen because the nature of the value functions is more visible

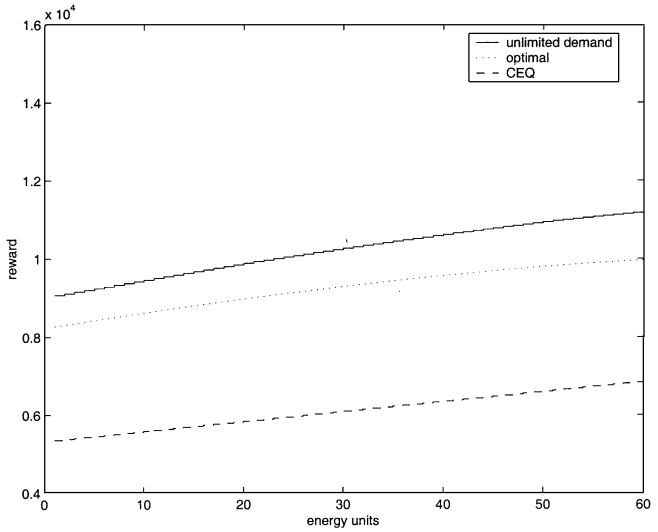

Fig. 7. Underlying value functions.

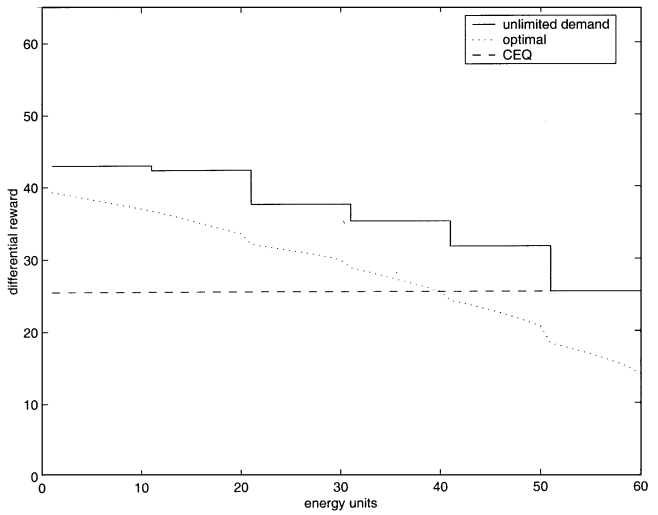

Fig. 8. First difference of underlying value functions.

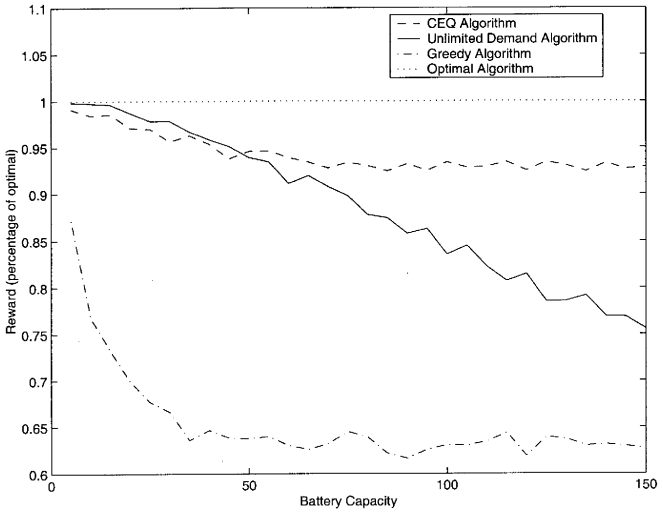

Fig. 9. Performance of policies as a function of battery capacity, $\lambda = 15$.

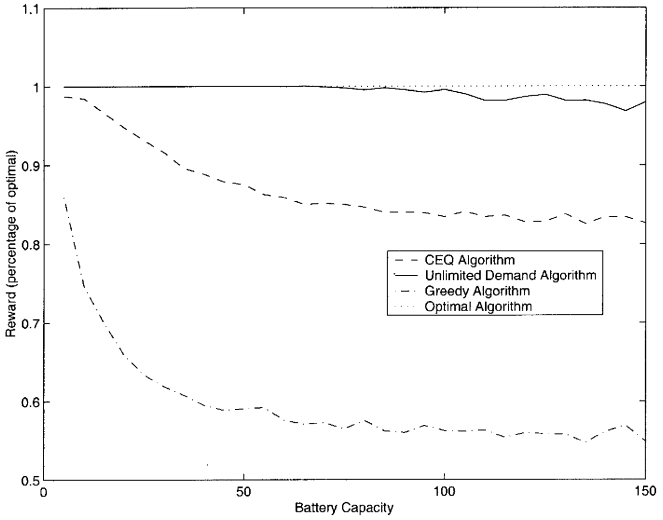

Fig. 10. Performance of policies as a function of battery capacity, $\lambda = 50$.

than at other time steps. Recall that the two heuristics generate an approximate value function and then generate a policy based on them. These are the "underlying" value functions that are plotted in the figures. As can be seen from the figures, all of these functions are concave in energy. Notice that the unlimited demand policy tends to overestimate the value of saving energy while the CEQ policy significantly underestimates the value of saving energy.

The first differences of the value functions are plotted in Fig. 8. The first difference gives the expected marginal reward for every extra energy unit as calculated by each policy.

The first difference of the underlying value function for the unlimited demand heuristic is always a staircase function. This structure results from the expiration times (explained earlier) that are imposed on incoming energy. If there is not much energy in the battery, incoming energy does not have to be spent for a long time. The policy can then wait for a time slot with high reward and accordingly, the expected value for an extra unit of energy is high. However, with a full battery, an extra unit of energy must be spent immediately, and hence the expected value is simply the expected value of the reward. The staircase structure results from the fact that new energy always comes in groups of ten units, and accordingly, the expiration time and marginal reward for energy changes every ten energy units.

The first difference of the optimal value function can be understood in the same framework. As can be seen from Fig. 8, this value function is always less than the one corresponding to the unlimited demand case. This reflects the possibility that insufficient demand is available and that energy cannot be spent before expiration. Hence, an extra unit of energy is always worth less when demand is limited.

The first difference of the underlying value function corresponding to the CEQ policy is simply a constant. As shown in Theorem 4, the value function is a piecewise linear function of available energy. In the range shown by the plot, however, the value function is completely linear. Since available energy is never outside the range shown by the plot, the underlying value function is effectively linear.

Figs. 9 and 10 show the total reward obtained by the various policies as battery capacity changes from 5 to 150, with $\lambda = 15$

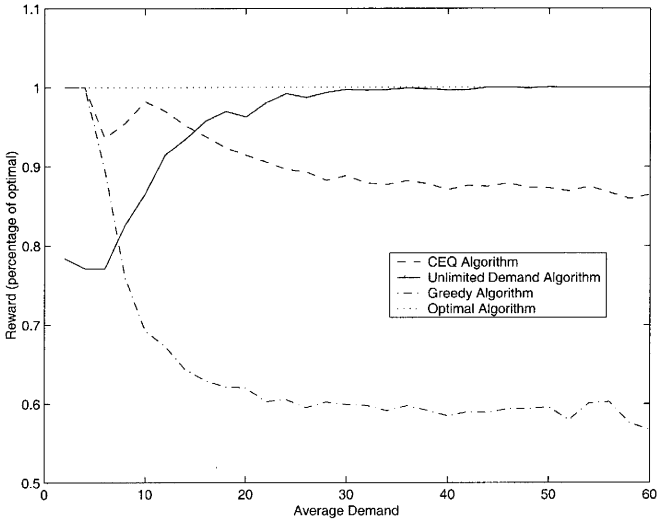

Fig. 11. Performance of policies as a function of λ (average demand).

and $\lambda = 50$, respectively. Fig. 11 shows the performance of the various policies 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-h horizon. The reward obtained by each policy is plotted as a fraction of the reward obtained by the optimal policy.

As can be seen from the figures, the three policies we have considered significantly outperform the greedy policy. The certainty equivalent heuristic always obtained at least 80% of the optimal reward, while the unlimited demand heuristic was always above 70%. Figs. 10 and 11 also show that the unlimited demand policy performed particularly well when the average demand was relatively high. Also notice from Figs. 9 and 10 that the performance of every suboptimal policy 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 policy. In contrast, when the battery capacity was small, all policies 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 policy deteriorating with increasing battery capacity and increasing demand, the total rewards obtained by the greedy policy 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 policy, which stores as little energy as possible. Similarly, the greedy policy 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 policy in the simulation was due mainly to the increased reward obtained by the other policies, which were able to exploit higher battery capacity and demand levels in making consumption decisions.

The computations were carried out on a Pentium III computer using Matlab 5.0. Computing underlying value functions and optimal policies for a typical data point from Fig. 11 required roughly 0.92 s when using the second method for calculating an optimal value function [see (10)]. The calculations for the unlimited demand approach required 0.51 s and those for the CEQ approach took 0.39 s. In contrast, the greedy policy required no precomputation, while a direct calculation of the optimal value function required about 26 min, 39 s.

V. Other Applications

The policies 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 approach 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 [9]. 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 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.

The equations that result for the first problem are almost identical to the satellite energy allocation problem. Throughput is analogous to reward in the satellite problem and the power limit is equivalent to demand. 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 significantly simplify calculations; nevertheless, the policies detailed above will be completely applicable.

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 policy: the optimal one, the unlimited demand policy, and the certainty equivalent policy. The three methods trade off computational complexity against performance and their behavior 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 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 methods as a satellite design tool rather than as an aid to operation. Because the computations run quickly on a computer, the effects of a reduction in battery capacity or an increase in average demand can be readily discerned. Another natural extension of our model would be to capture battery

charge/discharge effects, as discussed earlier. Finally, it would be interesting to study similar problems involving the acceptance or rejection of circuit-oriented connections, rather than offered 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

$$J_k(a_k,r_k,d_k)=\max_{0\le s_k\le \min(a_k,E_{\max})}\{r_k\min(a_k-s_k,d_k) + E_{r,d}[J_{k+1}(s_k+b_{k+1},r_{k+1},d_{k+1})]\} \quad (18)$$

and

$$J_n(a_n,r_n,d_n)=r_n\min(a_n,d_n). \quad (19)$$

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

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

$$f(\lambda y+\bar{\lambda}z)\ge\lambda f(y)+\bar{\lambda}f(z) \quad (20)$$

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

Lemma 2: If f and g are concave and $\alpha\ge 0$, then $f+g$ and αf are concave.

Proof: Follows from definition of concavity. □

Lemma 3: If $0\le\lambda\le 1$ and $\lambda+\bar{\lambda}=1$, then

$$\lambda\min(a,b)+\bar{\lambda}\min(c,b)\le\min(\lambda a+\bar{\lambda}c,b). \quad (21)$$

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 $J_n(a_n,r_n,d_n)$ is concave in a_n and the expected value function $E_{r,d}[J_n(a_{n-1}+b_n,r_n,d_n)]$ 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. Hence, $J_n(s_{n-1}+b_n,r_n,d_n)$ is concave in s_{n-1} as well, and by Lemma 2, the expectation $E_{r,d}[J_n(s_{n-1}+b_n,r_n,d_n)]$ is also concave in s_{n-1} since it is a weighted sum of concave functions.

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

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

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

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

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

Similarly

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

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 $\bar{\lambda}$

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

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

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

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

$$\lambda s_k^x+\bar{\lambda}s_k^y\le\lambda x+\bar{\lambda}y \quad (22)$$

and

$$\lambda s_k^x+\bar{\lambda}s_k^y\le\lambda E_{\max}+\bar{\lambda}E_{\max}. \quad (23)$$

Combining (22) and (23)

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

and

$$\begin{aligned}&\lambda J_k(x,r_k,d_k)+\bar{\lambda}J_k(y,r_k,d_k)\\&\quad\le\max_{0\le s_k\le\min(\lambda x+\bar{\lambda}y,E_{\max})}\{r_k\min(\lambda x+\bar{\lambda}y-s_k,d_k)\\&\qquad+E_{r,d}[J_{k+1}(s_k+b_{k+1},r_{k+1},d_{k+1})]\}\\&\quad=J_k(\lambda x+\bar{\lambda}y,r_k,d_k).\end{aligned} \quad (24)$$

This shows that $J_k(a_k,r_k,d_k)$ is concave in a_k. A direct application of Lemma 2 shows that $E_{r,d}[J_k(s_{k-1}+b_k,r_k,d_k)]$ is also concave in s_{k-1} and the induction is complete. □

B. Proof of Theorem 3: Piecewise Linearity of the Value Function

The objective is to show that the value function $J_k(a_k,r_k,d_k)$ is piecewise linear with corner points at the integers under the integrality assumptions of Theorem 3. We prove this by induction. At time n

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

Since we assume d_n to be integer, this function is clearly piecewise linear in a_n with corner points only at the integers.

Now assume that $J_{k+1}(a_{k+1}, r_{k+1}, d_{k+1})$ is piecewise linear in a_{k+1} with corner points at the integers. We show that $J_k(a_k, r_k, d_k)$ has the same property, using the formula for the value function given in (5).

It is clear that the term $\hat{J}_{k+1}(u)$ is also piecewise linear with corners at the integers. To see this, note that

$$\bar{J}_{k+1}(a_{k+1}) = E_{r,d}[J_{k+1}(a_{k+1}, r_{k+1}, d_{k+1})]$$

is a linear combination of functions with this property, and hence itself is piecewise linear with corners at the integers. Then, noting that

$$\hat{J}_{k+1}(u) = \bar{J}_{k+1}(\min(u, E_{\max}) + b_{k+1}))$$

we see that $\hat{J}_{k+1}(u)$ has the same property since $E_{\max}$ and b_k are assumed to be integer.

We have from (5) that

$$J_k(a_k, r_k, d_k) = \max_{\max(a_k - d_k, 0) \le u_k \le a_k} \{r_k(a_k - u_k) + \hat{J}_{k+1}(u_k)\}.$$

Theorem 2 provides the optimal values for u_k in the above expression. Substituting these values for the maximization, we obtain for $a_k < \phi_k(r_k)$

$$\begin{aligned} J_k(a_k, r_k, d_k) &= r_k(a_k - a_k) + \hat{J}_{k+1}(a_k) \\ &= \hat{J}_{k+1}(a_k) \end{aligned}$$

for $\phi_k(r_k) \le a_k \le \phi_k(r_k) + d_k$

$$J_k(a_k, r_k, d_k) = r_k(a_k - \phi_k(r_k)) + \hat{J}_{k+1}(\phi_k(r_k))$$

and for $\phi_k(r_k) + d_k < a_k$

$$\begin{aligned} J_k(a_k, r_k, d_k) &= r_k(a_k - (a_k - d_k)) + \hat{J}_{k+1}(a_k - d_k) \\ &= r_k d_k + \hat{J}_{k+1}(a_k - d_k). \end{aligned}$$

It is apparent that $J_k(a_k, r_k, d_k)$ is piecewise linear with corner points at the integers as long as $\phi_k(r_k)$ is integer. But $\phi_k(r_k)$ is a value of u_k that maximizes (7). This expression is concave and is also piecewise linear with corners at the integers. Thus, an integer maximizing value can always be found. Therefore, $J_k(a_k, r_k, d_k)$ is piecewise linear with corner points at the integers.

C. Proof of Theorem 4: Value Function for Certainty Equivalent Policy Under Fixed Average Reward

The underlying value function for the certainty equivalent policy is given by

$$\begin{aligned} \tilde{J}_k(a_k) = \max_{0 \le c_k \le a_k} \{&E[r]\min(c_k, E[d_k]) \\ &+ \tilde{J}_{k+1}(\min(a_k - c_k, E_{\max}) + b_{k+1})\} \end{aligned} \quad (25)$$

and

$$\tilde{J}_n(a_n) = E[r]\min(a_n, E[d_n]).$$

We seek to show by induction that (25) takes the form

$$\tilde{J}_k(a_k) = E[r]\min(a_k, \delta_k) + \gamma_k \quad (26)$$

where

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

and

$$\gamma_k = E[r]\min(b_{k+1}, \delta_{k+1}) + \gamma_{k+1}.$$

At time n, the underlying value function can obviously be written in this form, with $\gamma_n = 0$ and $\delta_n = E[d_n]$. Now, assume that (26) is true at time $k+1$. We show that it is true at time k as well.

First, by the CEQ assumption, future reward is the same at all times and equal to $E[r]$. It is also apparent that in the underlying value function (25), the reward for consuming at time k is also $E[r]$. Therefore, an optimal policy is a greedy policy that consumes as much as possible at all times, and $c_k = \min(a_k, E[d_k])$. Then the underlying value function can be written as

$$\begin{aligned} \tilde{J}_k(a_k) = &E[r]\min(a_k, E[d_k]) \\ &+ \tilde{J}_{k+1}(\min(a_k - \min(a_k, E[d_k]), E_{\max}) + b_{k+1})\}. \end{aligned}$$

Using (26) and substituting for $\tilde{J}_{k+1}(a_{k+1})$

$$\begin{aligned} \tilde{J}_k(a_k) = &E[r]\min(a_k, E[d_k]) \\ &+ E[r]\min(\min(a_k - \min(a_k, E[d_k]), E_{\max}) \\ &+ b_{k+1}, \delta_{k+1}) + \gamma_{k+1}. \end{aligned} \quad (27)$$

If $b_{k+1} > \delta_{k+1}$, then the term

$$\min(a_k - \min(a_k, E[d_k]), E_{\max}) + b_{k+1}$$

is always greater than δ_{k+1} and (27) simplifies to

$$\tilde{J}_k(a_k) = E[r]\min(a_k, E[d_k]) + E[r]\delta_{k+1} + \gamma_{k+1}.$$

If $b_{k+1} \le \delta_{k+1}$, then (27) can be written as

$$\begin{aligned} \tilde{J}_k(a_k) = &E[r]\min(a_k, E[d_k]) \\ &+ E[r]\min(\min(a_k - \min(a_k, E[d_k]) \\ &+ b_{k+1}, E_{\max} + b_{k+1}), \delta_{k+1}) + \gamma_{k+1} \end{aligned}$$

which can be reduced to

$$\begin{aligned} \tilde{J}_k(a_k) = E[r]\min(&a_k + b_{k+1}, a_k + E_{\max} \\ &+ b_{k+1}, E[d_k] + E_{\max} + b_{k+1}, a_k \\ &+ \delta_{k+1}, E[d_k] + \delta_{k+1}) + \gamma_{k+1}. \end{aligned}$$

Using the fact that $b_{k+1} \le \delta_{k+1}$ and $a_k + b_{k+1} \le a_k + b_{k+1} + E_{\max}$, we may eliminate several terms from the minimization above:

$$\begin{aligned} \tilde{J}_k(a_k) &= E[r]\min(a_k + b_{k+1}, E[d_k] + E_{\max} \\ &\quad + b_{k+1}, E[d_k] + \delta_{k+1}) + \gamma_{k+1} \\ &= E[r]\min(a_k, E[d_k] + \min(E_{\max}, \delta_{k+1} - b_{k+1})) \\ &\quad + E[r]b_{k+1} + \gamma_{k+1}. \end{aligned}$$

The value function is now in the desired form.

We have shown that the underlying value function at time k can be written as

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

where when $b_{k+1} > \delta_{k+1}$

$$\begin{aligned} \delta_k &= E[d_k] \\ \gamma_k &= E[r]\delta_{k+1} + \gamma_{k+1} \end{aligned}$$

and when $b_{k+1} \leq \delta_{k+1}$

$$\delta_k = E[d_k] + \min(E_{\max}, \delta_{k+1} - b_{k+1})$$
$$\gamma_k = E[r]b_{k+1} + \gamma_{k+1}.$$

The definitions of the constants may be consolidated by writing

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

and the induction is complete. □

REFERENCES

[1] J. M. Aein and O. S. Kosovych, "Satellite capacity allocation," *Proc. IEEE*, vol. 65, pp. 332–342, Mar. 1977.

[2] H. O. Awadalla, L. G. Cuthbert, and J. A. Schormans, "Predictive resource allocation for real time video traffic in broadband satellite networks," in *Proc. 5th Int. Conf. Broadband Communications*, Hong Kong, China, Nov. 1999, pp. 509–520.

[3] D. Berman, "The manned space station power system: An operational scheduler," M.S. thesis, Massachusetts Inst. Technol., Cambridge, 1986.

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

[5] C. F. Chiasserini and R. R. Rao, "Energy efficient battery management," in *Proc. IEEE INFOCOM*, vol. 2, Tel-Aviv, Israel, 2000, pp. 396–403.

[6] ——, "Improving battery performance by using traffic shaping techniques," *IEEE J. Select. Areas Commun.*, vol. 19, pp. 1385–1394, July 2001.

[7] R. C. Collette and B. L. Herdan, "Design problems of spacecraft for communication missions," *Proc. IEEE*, vol. 65, pp. 342–356, Mar. 1977.

[8] A. Fu, "Energy allocation and transmission scheduling for satellite and wireless networks," Ph.D. dissertation, Dept. Elect. Eng. Comput. Sci., Massachusetts Inst. Technol., Cambridge, 2002.

[9] A. Fu, E. Modiano, and J. Tsitsiklis, "Transmission scheduling over a fading channel with energy and deadline constraints," presented at the Conf. Information Sciences and Systems, Princeton, NJ, Mar. 2002.

[10] R. Kraus and T. Hendricks, "Satellite solar power analysis and sizing model," in *Proc. 26th Intersociety Energy Conversion Engineering Conf.*, vol. 1, La Grange Park, IL, 1991, pp. 195–200.

[11] I. Koutsopoulos and L. Tassiulas, "A unified framework for handover prediction and resource allocation in nongeostationary mobile satellite networks," in *Proc. IEEE 50th Vehicular Technology Conf.*, vol. 4, 1999, pp. 2106–2110.

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

[13] E. Papapetrou, I. Gragopoulos, and F. Pavlidou, "Performance evaluation of LEO satellite constellations with inter-satellite links under self-similar and poisson traffic," *Int. J. Satellite Commun.*, vol. 17, pp. 51–64, 1999.

[14] P. D. Shaft, "Unconstrained allocation of communication satellite traffic," in *Proc. IEEE Int. Conf. Commun.*, New York, 1977, pp. 26–30.

[15] M. Werner, J. Frings, F. Wauquiez, and G. Maral, "Topological design, routing, and capacity dimensioning for ISL networks in broadband LEO satellite systems," *Int. J. Satellite Commun.*, vol. 19, pp. 499–527, 2001.

[16] A. Ween *et al.*, "Dynamic resource allocation for multi-service packet based LEO satellite communications," in *Proc. IEEE GLOBECOM*, vol. 5, 1998, pp. 2954–2959.

Alvin C. Fu (S'02) received the B.S. degree in electrical engineering and the B.S. degree in biology, both in 1994 from Stanford University, Stanford, CA. He received the M.S. and Ph.D. degrees in electrical engineering and computer science from the Massachusetts Institute of Technology, Cambridge, in 1996 and 2003, respectively.

His M.S. thesis was on the topic of music recognition, and his current research is on satellite and wireless networks.

Eytan Modiano (S'90–M'93–SM'00) received the B.S. degree in electrical engineering and computer science from the University of Connecticut, Storrs, in 1986 and the M.S. and Ph.D. degrees in electrical engineering from the University of Maryland, College Park, in 1989 and 1992, respectively.

He was a Naval Research Laboratory Fellow between 1987 and 1992 and a National Research Council Postdoctoral Fellow during 1992–1993, while he was conducting research on security and performance issues in distributed network protocols. From 1993 and 1999, he was with the Communications Division at Lincoln Laboratory, Massachusetts Institute of Technology (M.I.T.), Cambridge, where he designed communication protocols for satellite, wireless, and optical networks, and was the Project Leader for Lincoln Laboratory's Next Generation Internet (NGI) project. Since 1999, he has been a Member of the Faculty of the Aeronautics and Astronautics Department and the Laboratory for Information and Decision Systems (LIDS) at M.I.T., where he conducts research on communication networks and protocols with emphasis on satellite and hybrid networks and high-speed networks.

John N. Tsitsiklis (F'99) received the B.S. degree in mathematics in 1980 and the B.S., M.S., and Ph.D. degrees in electrical engineering in 1980, 1981, and 1984, respectively, all from the Massachusetts Institute of Technology (M.I.T.), Cambridge.

He is currently a Professor of electrical engineering and computer science and a Codirector of the Operations Research Center at M.I.T. He has coauthored four books. His research interests are in the fields of systems, optimization, communications, control, and operations research. He is currently a member of the editorial board for the Springer-Verlag Lecture Notes in Control and Information Sciences series, and an Associate Editor of *Mathematics of Operations Research*.