Admission Control for Wireless Networks

Cynara C. Wu Dimitri P. Bertsekas
cynara@alum.mit.edu bertsekas@lids.mit.edu

Laboratory for Information and Decision Systems
Massachusetts Institute of Technology
Cambridge, MA 02139, USA

Abstract

With the population of wireless subscribers increasing at a rapid rate, overloaded situations are likely to become an increasing problem. Admission control can be used to balance the goals of maximizing bandwidth utilization and ensuring sufficient resources for high priority events. In this paper, we formulate the admission control problem as a Markov decision problem. While dynamic programming can be used to solve such problems, the large size of the state space makes this impractical. We propose an approximate dynamic programming technique, which involves creating an approximation of the original model with a state space sufficiently small so that dynamic programming can be applied. Our results show that the method improves significantly on policies that are generally in use, in particular, the greedy policy and the reservation policy. Much of the computation required for our method can be done off-line, and the real-time computation required is easily distributed between the cells.

Keywords

Cellular networks, admission control, dynamic programming.

1 Introduction

Efficient resource utilization is a primary problem in cellular communications systems. Resource issues include determining with which users to establish connections, and assigning transmit power levels to connected users subject to acceptable signal quality. For most users, the inability to initiate a call is perceived as more tolerable than the unexpected termination of a call. As a result, admission control, in which users requesting a connections are not automatically admitted even if resources are available to handle the connections, may be necessary to ensure sufficient resources are available for handoffs and other higher priority events.

The most common method of dealing with handoffs is to reserve a number of "guard" channels exclusively for handoffs. New calls are admitted only if the number of available channels exceed the number of guard channels. Such systems were introduced in the mid-1980's ([HR86, PG85]). A great deal of work has been done on developing more complex admission control schemes. However, many studies make a number of unrealistic, simplifying assumptions such as modeling handoff arrivals as Poisson processes that are independent of new call arrivals ([AAN96],[Gue88], [TJ92]). Other studies use more realistic models that require extremely complex calculations ([AS96],[LAN97]). Several authors have noted that while the problem is best modeled as a continuous-time Markov chain, the size of the state space makes the problem difficult to solve [BWE95, SS97]. Chang and Geraniotis actually use dynamic programming to solve their Markov decision problem, but their method does not generalize well to larger problems [CG98]. All the above works have focused on systems in which channels are orthogonal and all users require the same resources. In certain systems such as CDMA systems, different users may require varying resources. In addition, there may be different classes of users with different resource requirements.

In this paper, we consider the problem of optimal admission control: given a particular configuration of users of various classes in various regions, determine whether or not to accept a new call request. We assume we have available an algorithm that can determine for any distribution of users of various classes in various regions whether there is a feasible power assignment satisfying the signal to noise requirements for all users, and if so, provides a unique power assignment for the distribution. Our goal is to formulate the problem as a Markov decision process and to provide a solution method that is general enough to be widely applicable and can be implemented in real-time.

In Sec. 2, we develop a model for a multiple access cellular communications system and formulate the admission control problem as a Markov decision process. While such processes can be solved by dynamic programming, the size of the problem makes this impractical. In Sec. 3, we consider an approximate dynamic programming solution. We then present computational results in Sec. 4.

2 The Admission Control Problem

In this section, we develop a model for the admission control problem for a two-dimensional system of cells with multiple user classes. We first provide a general description of the system which we consider and then formulate the problem as a Markov decision process.

2.1 General Description

We consider a two-dimensional system of M cells in which users of C various classes can establish connections. Each cell contains a single base station. The jth cell, for $j = 1, \ldots, M$, contains R_j regions. We assume each region of a cell is small enough so that the propagation effects on all signals transmitted from users in a particular region can be assumed to be equivalent. The total number of cell regions is $R = \sum_{j=1}^{M} R_j$, and we denote the cell regions as $r = 1, \ldots, R$. (See Fig. 1.)

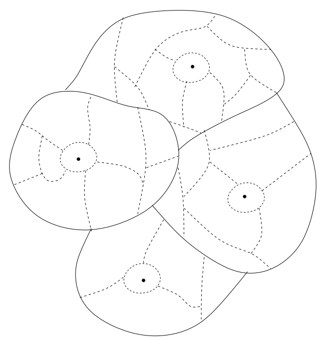


Figure 1: Model of a cellular system. A cell can be of any shape and is divided into regions that are represented by the dashed lines. The base station is represented by the small solid circle near the middle.

If a user in region r, for $r = 1, \ldots, R$, transmits a signal with a power level w, the signal strength received at the base station in cell j, for $j = 1, \ldots, M$, is $a_{rj}w$. The value a_{rj} is the amount that a signal transmitted from region r is attenuated by interference, multi-path, etc., by the time it is received by the base station in cell j. We assume that these values can be obtained either empirically or by analyzing appropriate propagation models and that they are available.

We assume that we have an algorithm that can determine whether any particular configuration of users in cell regions is feasible and if so, provides a unique assignment of power levels to each user. The system at any particular time is then characterized by the number of users of each class in each region. We represent this information by a matrix n of nonnegative integers:

$$n = \begin{bmatrix} n_{11} & \ldots & n_{1R} \\ & \vdots & \\ n_{C1} & \ldots & n_{CR} \end{bmatrix},$$

where n_{cr} is the number of users of class c in cell region r.

The arrival of connection requests in region $r = 1, \ldots, R$ by users of class $c = 1, \ldots, C$ is modeled as a Poisson process with rate λ_{cr}. When a request arrives, a decision has to be made regarding whether to admit the user or not. Given that an existing user of class c is currently in region r, the probability that he moves to region r' is independent of the regions the user has already visited and is given to be $q_{crr'}$. The probability that this user ends his connection in the current region r is $q_{crt} = 1 - \sum_{r'} q_{crr'} > 0$. If a user of class c does not move to any more regions, the length of the remaining connection time is exponentially distributed with rate μ_c. Otherwise, assuming the user moves from region r to region r', the length of time until this occurs is exponentially distributed with rate $\mu_{crr'}$.

While we assume we have some information regarding the cellular network such as the call request arrival rate to the various regions and the probability distribution of traffic movement, we do not assume knowledge of where a particular user is heading or a history of its locations; however, our methodology extends to cases where such information is available. In addition, we have assumed that a user's movement and remaining connection time is independent of his previous movement and the previous length of connection. As a result, the matrix n providing the number of users of each class in each cell region describes the "state" of the system. We assume that we can determine this information either through measurements or other means.

As requests for connections are accepted, existing connections are completed, and users move from one region to another, the state of the system changes. The changes depend on random elements such as user requests to establish connections, existing connections being completed, and movements of users from one region to another, as well as on decisions that need to be made, such as admission decisions. Costs or rewards can be associated with admission decisions, as well as with lengths of connection times. The admission control problem can therefore be viewed as a continuous-time Markov decision process. We describe how to do so in the next section.

2.2 Formulation of Problem as a Markov Decision Process

In an infinite horizon Markov decision process with a finite set of states, $\mathcal{S}$, the state evolves through time according to given transition probabilities $P_{ii'}$ that depend on a decision or control from a set U that depends on the current state. Suppose transitions occur at times $t_1, t_2, \ldots$. If the system is in state $x(t_k) = i$ after the kth transition and decision $u(t_k) \in U(i)$ is selected, then during the $(k+1)$st transition, the system moves to state $x(t_{k+1}) = i'$ with given probability $P_{ii'}(u(t_k))$. The interval between transitions is referred to as a "stage." During the kth stage, we incur a cost $g(x(t_k), u(t_k))\tau_k$, where g is a given function and $\tau_k = t_{k+1} - t_k$ is the length of the kth stage.

The goal is to minimize over all possible decisions at each stage the average cost per unit time. This cost function is of the form

$$\lim_{N \to \infty} \frac{1}{E\{t_N\}} E\left\{ \int_0^{t_N} g\big(x(t), u(t)\big) dt \right\}.$$

We assume the process is stationary; i.e., the transition probabilities, the set U of available controls, the cost functions g, etc, do not depend on the particular transition.

To formulate the admission control problem as a discrete-state infinite horizon Markov process, we first define the state as a combination of a "cell configuration" and the current event. We then provide the control space, the probability distribution of the interval between transitions, the transition probabilities, and the cost function.

2.2.1 The State Space

The state consists of the following two components:

1. A primary state component which describes the number of users of each class in each region and is denoted by n.

2. A random state component which describes the current random event and is denoted by ω.

The set $\mathcal{N}$ of all primary state components is the set of all n for which there is a feasible power assignment. We refer to an element of $\mathcal{N}$ as a feasible cell configuration.

The random state component describes the current random event and is of one of the following three types.

1. "Arrival": a connection request by a new user of any class in any region.

2. "Departure": a connection completion by an existing user of any class in any region.

3. "Movement": a movement by an existing user of any class from any region to another.

We denote a particular random state component by

$$\omega = \begin{bmatrix} \omega_{11} & \dots & \omega_{1R} \\ & \vdots & \\ \omega_{C1} & \dots & \omega_{CR} \end{bmatrix},$$

where ω_{cr} is equal to 1 if a user of class c in region r requests to make a connection, ω_{cr} is equal to -1 if an existing user of class c in region r ends his connection, $\omega_{c\bar{r}} = -1$ and $\omega_{c\hat{r}} = 1$ if an existing user of class c in region $\bar{r}$ moves to region $\hat{r}$, and ω_{cr} is equal to 0 otherwise. The set Ω of all possible events is then

$$\Omega = \Omega_a \cup \Omega_d \cup \Omega_m,$$

where Ω_a is the set of all events consisting of an arrival to the cell, Ω_d is the set of all events consisting of a departure from the cell, and Ω_m is the set of all events consisting of a single user moving from one region to another.

The state space $\mathcal{S}$ is composed of all possible combinations of primary state components and random state components and is given by

$$\mathcal{S} = \left\{ i = (n, \omega) \;\middle|\; n \in \mathcal{N}, \omega \in \Omega, \text{ and } \omega_{cr} \geq 0 \text{ if } \sum_{l=1}^{W_c} n_{crl} = 0 \right\},$$

where the last condition holds since departures and movements from region r by a user of class c can only occur if the number of users of class c in region r is positive.

2.2.2 The Control Space

For each state $i = (n, \omega) \in \mathcal{S}$, the set of available controls $U(i)$ depends on the random element as follows.

1. Arrival: $\omega \in \Omega_a$. If the random element corresponds to a connection request, the control consists of determining whether or not to admit the user. We have

   $$U(i) = \{u_a, u_b\},$$

   where u_a indicates the requesting connection is admitted and u_b indicates the requesting connection is blocked.

2. Departure: $\omega \in \Omega_d$. If the random element corresponds to an existing user completing his connection, there is no decision to be made.

3. Movement: $\omega \in \Omega_m$. If the random element corresponds to an existing user moving from one region to another, the control consists of determining whether or not to maintain the connection. We have

   $$U(i) = \{u_h, u_d\},$$

   where u_h indicates the attempted handoff is accommodated u_d indicates the attempted handoff is dropped.

There are essentially two types of controls: one in which the random event is accommodated and one in which the random event is not. Note that the control determines the primary state component n' for the next state i'. The available controls and the resulting primary state components are summarized in Table 1.

2.2.3 The State Transition Rates

For any cell configuration $n \in \mathcal{N}$, the time until the next random event occurs depends on the number of possible random events. The arrival of each of these events is a Poisson process with the rates provided in Table 2.

Random State Component		Available Controls		Next Primary State Component
Arrival	$\omega \in \Omega_a$	Admit User	$u = u_a$	$n' = n + \omega$
		Block User	$u = u_b$	$n' = n$
Departure	$\omega \in \Omega_d$	None		$n' = n + \omega$
Movement	$\omega \in \Omega_m$	Maintain Connection	$u = u_h$	$n' = n + \omega$
		Drop Connection	$u = u_d$	$n' = n + \omega^-$ $(\omega_{cr}^- = \min\{\omega_{cr}, 0\})$

Table 1: List of the possible random state components, the available controls, and the resulting cell configurations according to the selected control, given a state $i = (n, \omega)$.

Random State Component	Rate	Total Rate
Call request by user of class c in region r	λ_{cr}	$\sum_{c=1}^{C}\sum_{r=1}^{R}\lambda_{cr}$
Call completion by user of class c in reg r	$n_{cr}q_{crt}\mu_c$	$\sum_{c=1}^{C}\sum_{r=1}^{R} n_{cr}q_{crt}\mu_c$
Movement by user of class c from r to r'	$n_{cr}q_{crr'}\mu_{crr'}$	$\sum_{c=1}^{C}\sum_{r=1}^{R}\sum_{r'=1}^{R} n_{cr}q_{crr'}\mu_{crr'}$

Table 2: Possible next events for a particular configuration $n \in \mathcal{N}$ and the associated arrival rate. The second column indicates the rate for the particular random event described. The third column indicates the overall rate for all events of the type described.

The overall rate Λ_n at which events occur starting from a configuration n is the sum of the rates of all possible events and is given by

$$\Lambda_n = \sum_{c=1}^{C}\sum_{r=1}^{R}\left(\lambda_{cr} + n_{cr}\left[\sum_{r'=1}^{R} q_{crr'}\mu_{crr'} + q_{crt}\mu_c\right]\right).$$

For any state $i = (n, \omega) \in \mathcal{S}$, the control $u \in U(i)$ determines the cell configuration n' for the next state $i' = (n', \omega')$ as seen in Table 1. Assuming the control takes effect immediately, the transition rate from state i under control $u \in U(i)$ is then $\Lambda_{n'}$, or equivalently, Λ_u. We denote the expected value of the time from the transition to state i under control u to the transition to the next state as $\bar{\tau}_i(u)$:

$$\bar{\tau}_i(u) = \frac{1}{\Lambda_u}.$$

2.2.4 The State Transition Probabilities

Given a state $i = (n, \omega) \in \mathcal{S}$, the transition to the next state $i' = (n', \omega')$ consists of the following two factors.

1. The next cell configuration n' is determined by the control $u \in U(i)$ exercised for state i.

2. The probability distribution of the next random event ω' depends on the cell configuration resulting from the current state i and control u. Given the next reduced cell configuration n', the probability of a transition from $i = (n, \omega)$ to $i' = (n', \omega')$ is the rate at which the event occurs divided by the total transition rate:

$$P_{ii'}(u) = \begin{cases} \lambda_{cr}/\Lambda_{n'} & \text{if } \omega' \in \Omega_a,\ \omega'_{cr} = 1,\ c = 1, \ldots, C,\ r = 1, \ldots, R, \\ n'_{cr} q_{crt} \mu_c / \Lambda_{n'} & \text{if } \omega' \in \Omega_d,\ \omega'_{cr} = -1,\ c = 1, \ldots, C,\ r = 1, \ldots, R, \\ n'_{cr} q_{cr_1r_2} \mu_{cr_1r_2} / \Lambda_{n'} & \text{if } \omega' \in \Omega_m,\ \omega'_{cr_1} = 1,\ \omega'_{cr_2} = -1,\ c = 1, \ldots, C, \\ & \qquad r_1, r_2 = 1, \ldots, R. \end{cases} \tag{1}$$

We describe the state transition probabilities from a state $i = (n, \omega)$ to a state $i' = (n', \omega')$ in more detail below for the various possible values of ω.

1. If $\omega \in \Omega_a$, the possible controls are $u = u_a$, in which case $n' = n + \omega$, and $u = u_b$, in which case $n' = n$. The probability distribution for the next event ω' then depends on n' according to Eq. 1. The possible transitions are illustrated in Fig. 2 and described in the following table.

Control	Next Configuration	Next Event	Transition Probability from (n, ω) to (n', ω')
Admit User $u = u_a$	$n' = n + \omega$	$\omega' \in \Omega_a$	$\lambda_{cr}/\Lambda_{n'}$
		$\omega' \in \Omega_d$	$n'_{cr} q_{crt} \mu_c / \Lambda_{n'}$
		$\omega' \in \Omega_m$	$n'_{cr} q_{cr_1r_2} \mu_{cr_1r_2} / \Lambda_{n'}$
Block User $u = u_b$	$n' = n$	$\omega' \in \Omega_a$	$\lambda_{cr}/\Lambda_{n'}$
		$\omega' \in \Omega_d$	$n'_{cr} q_{crt} \mu_c / \Lambda_{n'}$
		$\omega' \in \Omega_m$	$n'_{cr} q_{cr_1r_2} \mu_{cr_1r_2} / \Lambda_{n'}$

2. If $\omega \in \Omega_d$, there is no decision to be made and $n' = n + \omega$. The possible transitions are described in the table below.

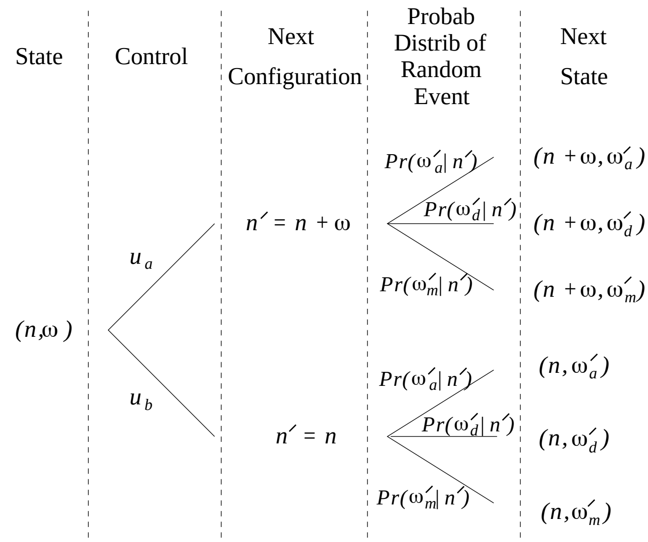


Figure 2: Illustration of the transition from state (n, ω) when there is a connection request.

Control	Next Configuration	Next Event	Transition Probability from (n, ω) to (n', ω')
None	$n' = n$	$\omega' \in \Omega_a$	$\lambda_{cr}/\Lambda_{n'}$
		$\omega' \in \Omega_d$	$n'_{cr} q_{crt} \mu_c / \Lambda_{n'}$
		$\omega' \in \Omega_m$	$n'_{cr} q_{cr_1 r_2} \mu_{cr_1 r_2} / \Lambda_{n'}$

3. If $\omega \in \Omega_m$, the possible controls are $u = u_h$, in which case $n' = n + \omega$, and $u = u_d$, in which case $n' = n + \omega^-$, where $\omega_{cr}^- = \min\{\omega_{cr}, 0\}$. The possible transitions are described in the table below.

Control	Next Configuration	Next Event	Transition Probability from (n, ω) to (n', ω')
Maintain Connection $u = u_h$	$n' = n + \omega$	$\omega' \in \Omega_a$	$\lambda_{cr}/\Lambda_{n'}$
		$\omega' \in \Omega_d$	$n'_{cr} q_{crt} \mu_c / \Lambda_{n'}$
		$\omega' \in \Omega_m$	$n'_{cr} q_{cr_1 r_2} \mu_{cr_1 r_2} / \Lambda_{n'}$
Drop Connection $u = u_d$	$n' = n + \omega^-$ ($\omega_{cr}^- = \min\{\omega_{cr}, 0\}$)	$\omega' \in \Omega_a$	$\lambda_{cr}/\Lambda_{n'}$
		$\omega' \in \Omega_d$	$n'_{cr} q_{crt} \mu_c / \Lambda_{n'}$
		$\omega' \in \Omega_m$	$n'_{cr} q_{cr_1 r_2} \mu_{cr_1 r_2} / \Lambda_{n'}$

2.2.5 The Cost Function

The goal is to select controls at each possible state that minimizes the expected average cost per unit time,

$$\lim_{N \to \infty} \frac{1}{E\{t_N\}} E\left\{ \int_0^{t_N} g\big(x(t), u(t)\big) dt \right\},$$

where g is a given function. We can rewrite this objective cost function as a sum of stage costs:

$$\lim_{N\to\infty} \frac{1}{E\{t_N\}} \sum_{k=1}^{N} E\{G_k\},$$

where

$$G_k = \int_{t_k}^{t_{k+1}} g\big(x(t_k), u(t_k)\big) dt$$

is the cost of the kth stage. We assume the function g has a component $\hat{g}$ that depends linearly on the length of time spent in a particular state and a component $\check{g}$ that does not depend on the length of time spent in a particular state. In this case, the cost of the kth stage is then

$$G_k = (t_{k+1} - t_k)\hat{g}\big(x(t_k), u(t_k)\big) + \check{g}\big(x(t_k), u(t_k)\big).$$

The first component can accommodate assigning rewards (or negative costs) for the amount of time users are connected, while the second component can accommodate assigning costs for not admitting a user.

2.3 Solving the Markov Decision Problem

In Sec. 2.2, we formulated the admission control problem as an average cost Markov decision problem. We provided the state space $\mathcal{S}$, the set of available controls $U(i)$ for each possible state $i \in \mathcal{S}$, and the state transition probabilities $P_{ii'}(u)$. The objective is to select for each state $x(t) \in \mathcal{S}$ resulting from the kth transition, decisions $u(x(t_k)) \in U(x(t_k))$ that minimize the average cost per stage. In this section, we discuss applying dynamic programming (DP) to solve the Markov decision problem.

Let ψ be any given stationary policy such that for any state $i \in \mathcal{S}$, $\psi(i) \in U(i)$ indicates the control provided by the policy at state i. We denote the average cost per unit time under policy ψ by v^{ψ} and the average cost per unit time under an optimal policy ψ^* by v^*. It can be shown that for the admission control problem, this value is independent of the initial state. As a result, states are compared using a "differential" cost. We denote the differential cost of starting in state i relative to some reference state under policy ψ as $h^{\psi}(i)$ and the differential cost under an optimal policy as $h^*(i)$.

The expected cost for a single stage corresponding to state i under an optimal policy is

$$v^*\bar{\tau}_i(u),$$

where $\bar{\tau}_i(u)$ is the expected length of the transition time corresponding to state i when control $u = \psi^*(i)$ is exercised. Similarly, the expected cost for a single stage corresponding to state i when control u is exercised, which we denote $G(i, u)$, is

$$G(i, u) = \hat{g}(i, u)\bar{\tau}_i(u) + \check{g}(i, u).$$

It can be shown that for the admission control problem, the vector h^* and scalar v^* satisfy the following system of equations:

$$h(i) = \min_{u \in U(i)} \left\{ G(i,u) - v\bar{\tau}_i(u) + \sum_{i' \in \mathcal{S}} P_{ii'}(u) h(i') \right\}, \qquad \text{for all } i \in \mathcal{S}, \tag{2}$$

known as Bellman's equation.

Given the optimal average cost per unit time v^* and the optimal differential cost function h^*, the optimal decision at state i, $\psi^*(i)$, is that which minimizes the immediate cost of the current stage minus the expected average cost for the stage plus the remaining expected differential cost based on the possible resulting states; i.e., we have

$$\psi^*(i) = \arg\min_{u \in U(i)} \left\{ G(i,u) - v^*\bar{\tau}_i(u) + \sum_{i' \in \mathcal{S}} P_{ii'}(u) h^*(i') \right\}, \qquad \text{for all } i \in \mathcal{S}.$$

For the admission control problem, the optimal policy for any state $i = (n, \omega) \in \mathcal{S}$, $\psi^*(i)$, is obtained by solving the following equation at each stage:

$$\psi^*(i) = \begin{cases} \arg\min_{u_a, u_b} \Big\{ G(i,u_a) - v^*\bar{\tau}_i(u_a) + \sum_{i' \in \mathcal{S}} P_{ii'}(u_a) h^*(i'), \\ \qquad G(i,u_b) - v^*\bar{\tau}_i(u_b) + \sum_{i' \in \mathcal{S}} P_{ii'}(u_b) h^*(i') \Big\}, & \text{if } \omega \in \Omega_a, \\ \arg\min_{u_h, u_d} \Big\{ G(i,u_h) - v^*\bar{\tau}_i(u_h) + \sum_{i' \in \mathcal{S}} P_{ii'}(u_h) h^*(i'), \\ \qquad G(i,u_d) - v^*\bar{\tau}_i(u_d) + \sum_{i' \in \mathcal{S}} P_{ii'}(u_d) h^*(i') \Big\}, & \text{if } \omega \in \Omega_m, \end{cases} \tag{3}$$

where the quantities G, $\bar{\tau}_i$, and $P_{ii'}$ are those provided in Sec. 2.2. Note that if $\omega \in \Omega_d$, there is no control to exercise.

There are a number of methods for solving Bellman's equation to obtain the values for v^* and h^*. Most of these are forms of the value iteration and policy iteration algorithms. The computation can be done off-line, i.e., before the real system starts operating. Once this computation is completed, the optimal decision at each call request can be determined by comparing the cost resulting from applying each of the available controls and selecting the control with the minimum value. A comprehensive treatment of DP can be found in [Ber95].

One major advantage of DP over others is that the computation required to incorporate any system details or an arbitrarily complex cost function is done off-line and therefore need not be real-time. Once the cost function is computed, the optimal policy is often determined quickly using the cost function with real-time information on the current state of the system. Furthermore, the optimal cost function can vary from cell to cell to account for variances such as arrival rates, and cell shapes and sizes.

Unfortunately, even off-line, the computation required to determine these values is overwhelming due to the large number of states. As a result, suboptimal methods for the solution must be used. In the next section, we discuss a technique to obtain an approximation for the optimal average cost and optimal differential cost function.

3 An Approximate Solution: Cell-by-Cell Decomposition with Feature Extraction

In Sec. 2.2, we formulated the admission control problem as a Markov decision process. Unfortunately, the size of the state space is too large for exact dynamic programming to be practical. In this section, we consider an approximate solution.

For convenience, we define

$$H^*(i,u) = G(i,u) - v^*\bar{\tau}_i(u) + \sum_{i' \in \mathcal{S}} P_{ii'}(u) h^*(i'), \quad i \in \mathcal{S}, \ \ u \in U(i).$$

If we were able to obtain the optimal average cost per stage v^* and the optimal differential cost vector h^*, the optimal policy, given by Eq. 3, would be equivalent to

$$\psi^*(i) = \arg \min_{u \in U(i)} H^*(i,u) = \begin{cases} \arg \min_{u_a, u_b} H^*(i,u), & \text{if } \omega \in \Omega_a, \\ \arg \min_{u_h, u_d} H^*(i,u), & \text{if } \omega \in \Omega_m, \end{cases} \qquad i \in \mathcal{S}. \tag{4}$$

We refer to $H^*(i,u)$ as the Q-factor of the state-control pair (i,u). It is the expected cost corresponding to exercising control u at state i and then proceeding to follow an optimal policy. Note that the optimal differential cost vector h^* satisfies the equation

$$h^* = \min_{u \in U(i)} H^*(i,u).$$

Instead of calculating the actual average cost per stage v^* and differential cost vector h^* and using Eq. 4 to determine the optimal decision at every state, we construct approximations of the Q-factors $H^*(i,u)$, to which we refer as $\tilde{H}(i,u)$. After determining such approximations, the control for every state is given by

$$\psi(i) = \arg \min_{u \in U(i)} \tilde{H}(i,u), \qquad \text{for all } i \in \mathcal{S}.$$

We consider cell-by-cell decomposition combined with feature extraction to formulate a new Markov decision problem that is an approximation of the original problem but has a significantly smaller state space so that dynamic programming can be applied. We apply relative value iteration to the new formulation to determine the optimal average cost per stage

$\tilde{v}$ and differential cost function $\tilde{h}$ of the approximation, thereby obtaining an approximation of $H^*(i,u)$ of the form

$$\tilde{H}(i,u) = G(i,u) - \tilde{v}\bar{\tau}_i(u) + \sum_{i' \in \mathcal{S}} P_{ii'}(u)\tilde{h}(i'), \qquad i \in \mathcal{S}, \quad u \in U(i). \tag{5}$$

In the approximate problem, we decompose the original model into individual cells and consider each cell independent of the others. For each cell j, we obtain an estimate of the average cost per stage $\tilde{v}_j$ and the differential cost vector $\tilde{h}_j$. Costs associated with a particular user are attributed to the cell in which the user is located when the cost is incurred. For instance, suppose there is a fixed cost associated with blocking a connection request by a user of class c. Each cell in which a user of class c is blocked is then attributed that blocking cost. The total average cost per stage and total differential cost is the sum of the costs associated with each cell:

$$\tilde{v} = \sum_{j=1}^{N} \tilde{v}_j, \qquad \text{and} \qquad \tilde{h}(i) = \sum_{j=1}^{N} \tilde{h}_j(i), \qquad i \in \mathcal{S}. \tag{6}$$

To obtain the approximations $\tilde{v}_j$ and $\tilde{h}_j$, we formulate a Markov decision process for each cell. The state of each process consists of "features" of the state of the original admission control problem. With an appropriate selection of features, each process is simple enough so that traditional dynamic programming techniques such as value iteration can be applied to obtain $\tilde{v}_j$ and $\tilde{h}_j$. Specifically, for each state $i \in \mathcal{S}$ and cell $j = 1, \ldots, N$, $f_j(i) = \big(f_{j1}(i), \ldots, f_{jK_j}(i)\big)$ is a mapping of the state i to a vector of features, where K_j is the number of features that are to be extracted for use in evaluating cell j. The differential cost function is then approximated by a sum over all cells of functions that depend on these features instead of the actual state:

$$\tilde{h}(i) = \sum_{j=1}^{N} \tilde{h}_j\big(f_j(i)\big).$$

This approximation is illustrated in Fig. 3. Ideally, the features that are "extracted" summarize important characteristics of the state which affect the cost, and they should incorporate the designer's prior knowledge or intuition about the problem and about the structure of the optimal controller.

We summarize the cell-by-cell decomposition with feature extraction procedure as follows.

1. Select suitable features for each cell.

2. Formulate a Markov decision process for each cell where the state consists of the features selected in Step 1. We refer to the formulated process as the "feature-based MDP" for the cell.

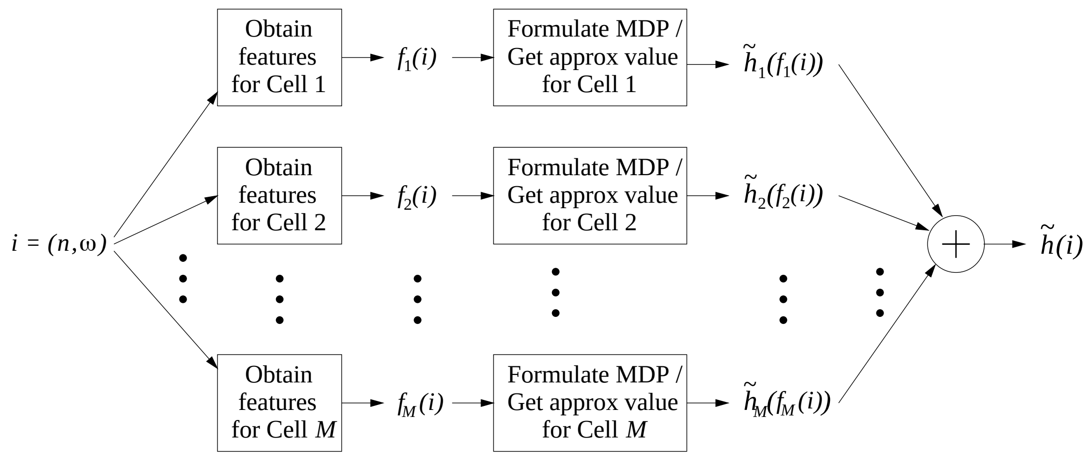


Figure 3: Cell-by-cell decomposition with feature extraction: For any state i, a set of features are extracted for each cell. The features are used as the state space for a Markov decision process associated with the cell. The costs resulting from solving the Markov decision process are approximations of the costs incurred by that cell.

3. Calculate the average costs per stage $\tilde{v}_j$ and the differential cost vectors $\tilde{h}_j$ for the Markov decision processes formulated for each cell in Step 2.

4. Approximate the overall average cost per stage and differential cost vector as a sum of the costs associated with the individual cells and determined in Step 3 (see Eq. 6). These values can be used in Eq. 5 to obtain $\tilde{H}(i,u)$, which is then used to make decisions at state $i \in \mathcal{S}$ according to

$$\psi(i) = \arg\min_{u \in U(i)} \tilde{H}(i,u), \qquad \text{for all } i \in \mathcal{S}.$$

Note that after using cell-by-cell decomposition and feature extraction to obtain an approximate problem, the features of a number of cells will not be affected by the current decision and therefore these cells need not be included in evaluating decisions. Specifically, given a state $i = (n, \omega)$, let $\mathcal{J}(i)$ be the set of cells whose features are modified by any control in $U(i)$. The decision at state $i \in \mathcal{S}$ is then given below:

$$\begin{aligned} \psi(i) &= \arg\min_{u \in U(i)} \tilde{H}(i,u) \\ &= \arg\min_{u \in U(i)} \left\{ G(i,u) - \bar{\tau}_i(u) \sum_{j \in \mathcal{J}(i)} \tilde{v}_j + \sum_{i' \in \mathcal{S}} \left[P_{ii'}(u) \sum_{j \in \mathcal{J}(i)} \tilde{h}_j(f_j(i')) \right] \right\}. \end{aligned}$$

If we do not include costs associated with cells whose features are not affected by the addition of a new user, the amount of calculation required to make each admission decision becomes independent of the total number of cells in the system. The total amount of calculation then grows linearly in the number of cells. Furthermore, since each admission decision depends on local information, the method is easily distributed so that each cell includes a controller responsible for making decisions for requests within its regions.

We describe the first three steps in detail in the following sections. In Sec. 3.1, we describe the process of selecting suitable features and provide several examples. In Sec. 3.2, we illustrate with an example the process of formulating the Markov decision process for a particular cell given a set of features. In Sec. 3.3, we describe the process of determining the values of $\tilde{v}_j$ and $\tilde{h}_j$ given a Markov decision process formulation for cell j.

3.1 Selecting Suitable Features

In this section, we describe the first step involved in the cell-by-cell decomposition with feature extraction procedure: the process of selecting features from the overall state for each cell. The features for a particular cell j are basically the vector of numbers to which a function f_j maps a given state. We describe issues to consider when selecting features and provide several examples.

To determine what features to extract as a representation of the state for a particular cell, we must consider what information is relevant in assessing the value of a state, and how the value changes upon the addition of a new user or upon the movement of an existing user. We must therefore consider the objective cost function. In selecting examples of features to use, we have made certain assumptions about the cost function. We assume there are costs associated with blocked calls and dropped calls, as well as possibly rewards for users that are connected. These costs and rewards may depend on the user class.

Under this assumption, the value of the state depends on the likelihood of future blocking and future dropping of calls of the various classes. The expected number of future blocks for new call attempts in a particular cell depends for the most part on the interference within the cell. The amount of interference depends on the number of calls of each class in the given cell and the number of calls in other cells that create interference to this cell. Note that the particular regions in which users in the given cell are located are not likely to be a significant factor. This may not necessarily be the case for users in other cells that generate interference within the given cell. Depending on the location of these other users and the base stations to which they are connected, the interference generated to the given cell can vary from being negligible to significant. For example, in Fig. 4, depending on propagation effects, a call in cell 2 near the border to cell 1 may generate a significant amount of interference at the base station in cell 1. It may also generate some interference at the base station in cell 3, although the amount is likely to be negligible.

The preceding considerations suggest two sets of features of state i relevant to determining costs incurred from blocked calls occurring in each cell j:

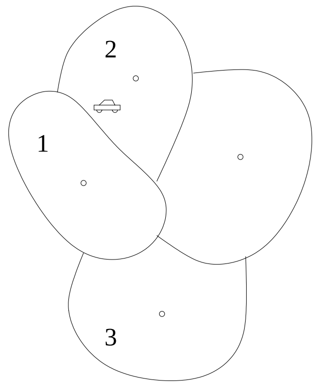


Figure 4: The user in cell 2 is likely to generate a fair amount of interference to the base station in cell 1 and therefore affect the ability to accept future connection requests. Its influence on future arrivals in cell 3, however, is likely to be significantly smaller.

1. $f_{j1c}(i)$: the number of users of class c in cell j,
2. $f_{j2c}(i)$: the number of users of class c in other cells whose signal levels at the base station in cell j are above some threshold.

The second set of features can further be subdivided into groups by creating various signal level threshold ranges to distinguish between users in other cells that generate a significant amount of interference and those that generate a moderate amount. Tradeoffs between the additional information that is obtained at the expense of larger state spaces should be considered in selecting the number of subgroups.

Another factor that may affect future blocks are the users within the cell that may move to another cell. For instance, users that move from periphery regions are more likely to move to another cell in the near future than users located in the center. In addition, traffic patterns may provide additional information. If cells are built around a highway, there will be cells where users can only move in a certain direction (see Fig. 5). Users that are likely to

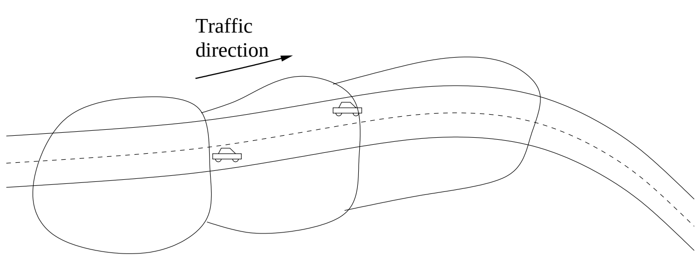


Figure 5: Assume traffic flows only left to right on the illustrated road. The user on the left is more likely to be connected to the base station in the center cell for a longer period of time than that on the right. In addition, the user on the right is likely to require its call be handed off in the near future.

leave the cell are more likely to generate less interference in the future. This factor suggests including the following set of features:

3. $f_{j3c}(i)$: number of users of class c in cell j that are positioned in regions in which the probability of moving to a region of another cell is greater than 0.

The expected number of future drops for users attempting to move to a particular cell depends on the number of existing users of each class positioned to move into this cell in addition to the interference currently present. For instance, the user on the right in Fig. 5 is likely to move to the cell on the right and may require its call be handed off. This suggests the following set of features:

4. $f_{j4c}(i)$: number of users of class c in other cells that are positioned in regions in which the probability of moving to cell j is greater than 0.

The set of features can be tailored to each cell. Once the set of features are selected for a particular cell, a Markov decision process is formulated whose state consists of the selected features. This feature-based MDP is not automatically determined by the chosen features. Instead, the transition probabilities and other characteristics are based on additional assumptions and approximations that are to some extent heuristic and reflect our understanding of the practical problem at hand. We illustrate this process in the next section.

3.2 Formulation Example

In this section, we illustrate with an example the second step involved in the cell-by-cell decomposition with feature extraction procedure: the process of formulating the feature-based MDP for a particular cell. This example is the basis of a portion of the computational results presented in Sec. 4. In this example, we extract from any state $i \in \mathcal{S}$ the following $2C$ features for each cell $j = 1, \ldots, N$:

1. $f_{jc}(i)$ = the number of users of class $c \in \{1, \ldots, C\}$ in all of the regions in cell j,

2. $f_{j(C+c)}(i)$ = the number of users of class $c \in \{1, \ldots, C\}$ in all regions r such that $wa_{rj} > T$, where r is a region not in cell j, w is the power level assigned to any user in region r, and T is an arbitrary constant.

The purpose of the first set of features is to indicate how much interference is generated by users within a particular cell. The purpose of the second set of features is to indicate how much interference is generated by users in other cells. For convenience, let $\mathcal{R}_j$ be the set of regions in cell j and let $\mathcal{R}^o_{jc}(i)$ be the set of regions r not in cell j such that given a state i, the available power control algorithm assigns a power level w to users of class c such that $wa_{rj} > T$. In this example, we assume that for any cell j, the set of regions not in j in which users generate sufficient interference to the base station in j is independent of the user class and of the actual power assignment (and consequently, the state). We therefore simplify notation by referring to this set by $\mathcal{R}^o_j$.

We now formulate the Markov decision process based on the selection of these features. Both the state space and the control space are analogous to those in the formulation of the admission control problem. In general, the transition rates and probabilities, and the cost function for this feature-based MDP can not be determined exactly. We provide examples of suitable approximations in our formulation.

The State Space The state space consists of the following two components:

1. A primary state component which describes the feature values of the state for cell j and is denoted by $\tilde{n} = (\tilde{n}_1, \ldots, \tilde{n}_{(2C)})$. In this example, the features are the number of users of each class in cell j and the number of users of each class in regions in $\mathcal{R}_j^o$. We also refer to this component as the feature configuration.

2. A random state component which describes the current random event and is denoted by $\omega = (\omega_1, \ldots, \omega_{(2C)})$.

Let $\tilde{\mathcal{N}}_j$ be the set of primary state components for cell j. We have

$$\tilde{\mathcal{N}}_j = \left\{ \tilde{n} \;\middle|\; \tilde{n}_c = \sum_{r \in \mathcal{R}_j} n_{cr}; \;\; \tilde{n}_{(C+c)} = \sum_{r \in \mathcal{R}_j^o} n_{cr}, \; c = 1, \ldots, C, \text{ for some } n \in \mathcal{N} \right\},$$

where $\mathcal{N}$ is the set of primary state components for the admission control problem.

We are only concerned with events that can change a particular cell configuration. These consist of arrivals to, departures from, and movements involving regions in $\mathcal{R}_j$ or $\mathcal{R}_j^o$. Movements of users from a region in $\mathcal{R}_j$ to another region in $\mathcal{R}_j$ or from a region in $\mathcal{R}_j^o$ to another region in $\mathcal{R}_j^o$ need not be considered an event. In addition, we can treat a movement by a user from a region in $\mathcal{R}_j$ or $\mathcal{R}_j^o$ to a region not in either $\mathcal{R}_j$ or $\mathcal{R}_j^o$ as a departure, and a movement by a user in a region not in either $\mathcal{R}_j$ or $\mathcal{R}_j^o$ to a region in $\mathcal{R}_j$ or $\mathcal{R}_j^o$ as an arrival.

The set of all random events for a particular cell j is given by

$$\Omega = \Omega_a \cup \Omega_{ao} \cup \Omega_d \cup \Omega_{do} \cup \Omega_{m1} \cup \Omega_{m2},$$

where Ω_a and Ω_{ao} are the sets of all events consisting of an arrival to cell j and to a region in $\mathcal{R}_j^o$, respectively, Ω_d and Ω_{do} are the sets of all events consisting of a departure from cell j and from a region in $\mathcal{R}_j^o$, respectively, and Ω_{m1} and Ω_{m2} are the sets of all events consisting of a user moving from a region in cell j to a region in $\mathcal{R}_j^o$ and vice versa, respectively.

The state space associated with cell j, which we denote $\mathcal{S}_j$, is composed of all possible combinations of primary state components and random state components and is given by

$$\mathcal{S}_j = \left\{ i = (\tilde{n}, \omega) \;\middle|\; \tilde{n} \in \tilde{\mathcal{N}}_j, \;\; \omega \in \Omega \text{ and } \omega_c \geq 0 \text{ if } \tilde{n}_c = 0 \right\},$$

where the last condition holds since departures and movements by a user of class c can only occur if the number of users of class c is positive.

The Control Space For any state $i = (\tilde{n}, \omega)$, the possible controls as well as the resulting feature configurations $\tilde{n}'$ are given in Table 3 according to the various possible values of ω. If the random state component is a departure, there is no choice of control and the event is accommodated. Otherwise, the control can either be to accommodate or not accommodate the event.

Random State Component		Available Controls		Next Primary State Component
Arrival	$\omega \in \Omega_a \cup \Omega_{ao}$	Admit User Block User	$u = u_a$ $u = u_b$	$\tilde{n}' = \tilde{n} + \omega$ $\tilde{n}' = \tilde{n}$
Departure	$\omega \in \Omega_d \cup \Omega_{do}$	None		$\tilde{n}' = \tilde{n} + \omega$
Movement	$\omega \in \Omega_{m1} \cup \Omega_{m2}$	Maintain Connection Drop Connection	$u = u_h$ $u = u_d$	$\tilde{n}' = \tilde{n} + \omega$ $\tilde{n}' = \tilde{n} + \omega^-$ $(\omega_c^- = \min\{\omega_c, 0\})$

Table 3: List of the possible random state components, the available controls, and the resulting configurations according to the selected control.

Probabilities of Moving In order to determine the transition rates and probabilities, we need to make certain approximations since information regarding the particular region a user is located is no longer available. For example, the probability that a user moves from a particular cell j to a region in $\mathcal{R}_j^o$ must be estimated since we do not know whether the user is in a region to which it is possible to move to another cell, and even if the user is in such a region, the probability may vary depending on the particular region. The actual choice of approximation should be determined by the actual system and should vary according to cell. One possibility is to take a weighted average of the probabilities of moving for each of the regions within a cell. If the weights depend on the arrival rate to the particular region, the approximate probability that a user of class c in cell j moves to a region in $\mathcal{R}_j^o$, which we denote $\tilde{q}_{jcm}$, is

$$\tilde{q}_{jcm} = \frac{\sum_{r \in \mathcal{R}_j} \left(\lambda_{cr} \cdot \sum_{r' \in \mathcal{R}_j^o} q_{crr'} \right)}{\sum_{r \in \mathcal{R}_j} \lambda_{cr}}.$$

We can similarly approximate the probability that a user of class c in a region in $\mathcal{R}_j^o$ moves to a region in cell j, which we denote $\tilde{q}_{jcm}^o$ by

$$\tilde{q}_{jcm}^o = \frac{\sum_{r \in \mathcal{R}_j^o} \left(\lambda_{cr} \cdot \sum_{r' \in \mathcal{R}_j} q_{crr'} \right)}{\sum_{r \in \mathcal{R}_j^o} \lambda_{cr}}.$$

Note that from the perspective of cell j, we can consider a movement by a user in a region in $\mathcal{R}_j^o$ to a region that is not in $\mathcal{R}_j^o$ nor in cell j as a departure or connection completion. We therefore include the probability of such movements as part of the probability that a user of class c in a region in $\mathcal{R}_j^o$ ends an existing connection, which we denote $\tilde{q}_{jct}^o$:

$$\tilde{q}_{jct}^o = \frac{\sum_{r \in \mathcal{R}_j^o} \lambda_{cr} \cdot \left(q_{crt} + \sum_{r' \notin \mathcal{R}_j \cup \mathcal{R}_j^o} q_{crr'} \right)}{\sum_{r \in \mathcal{R}_j^o} \lambda_{cr}}.$$

We also denote the total departure probability of a user of class c in cell j by $\tilde{q}_{jct}$:

$$\tilde{q}_{jct} = \frac{\sum_{r \in \mathcal{R}_j} \lambda_{cr} q_{crt}}{\sum_{r \in \mathcal{R}_j} \lambda_{cr}}.$$

Transition Rates Before presenting the transition rates and probabilities, we denote the total rate at which users of class c request a connection in each cell $j = 1, \ldots, N$ by $\tilde{\lambda}_{jc}$:

$$\tilde{\lambda}_{jc} = \sum_{r \in \mathcal{R}_j} \lambda_{cr},$$

and the total rate at which users of class c request a connection in a region not in cell j but which interferes with the base station in cell j by $\tilde{\lambda}_{jc}^o$:

$$\tilde{\lambda}_{jc}^o = \sum_{r \in \mathcal{R}_j^o} \lambda_{cr}.$$

We can make approximations to specifying the overall rates at which users move or complete their connections in a manner similar to that in making approximations to specifying the probabilities of movements or connection completions. For instance, to determine an approximation to the rate at which a user of class c in cell j moves to a region in $\mathcal{R}_j^o$, which we denote $\tilde{\mu}_{jcm}$, we can weight the various possible transition rates according to the arrival rate to individual regions in cell j, and then consider the probability of moving from any of these regions to a region in $\mathcal{R}_j^o$. Such a weighting results in the transition rate

$$\tilde{\mu}_{jcm} = \frac{\sum_{r \in \mathcal{R}_j} \left(\lambda_{cr} \cdot \sum_{r' \in \mathcal{R}_j^o} q_{crr'} \mu_{crr'} \right)}{\sum_{r \in \mathcal{R}_j} \left(\lambda_{cr} \cdot \sum_{r' \in \mathcal{R}_j^o} q_{crr'} \right)}.$$

An approximation to the rate at which a user of class c in $\mathcal{R}_j^o$ moves to a region in cell j can be done in a similar manner, resulting in the rate

$$\tilde{\mu}_{jcm}^o = \frac{\sum_{r \in \mathcal{R}_j^o} \left(\lambda_{cr} \cdot \sum_{r' \in \mathcal{R}_j} q_{crr'} \mu_{crr'} \right)}{\sum_{r \in \mathcal{R}_j^o} \left(\lambda_{cr} \cdot \sum_{r' \in \mathcal{R}_j} q_{crr'} \right)}.$$

The rate at which a user of class c in cell j ends an existing connection is independent of the particular region and is therefore also independent of the cell:

$$\tilde{\mu}_{ct} = \mu_c.$$

Since the probability that a user of class c in $\mathcal{R}_j^o$ ends an existing connection includes the probability that such a user moves to a region not in either $\mathcal{R}_j^o$ or in cell j, we can similarly weight such transitions to obtain an approximation to the rate at which a user of class c in $\mathcal{R}_j^o$ ends an existing connection:

$$\tilde{\mu}_{jct}^o = \frac{\sum_{r \in \mathcal{R}_j} \lambda_{cr} \cdot \left(q_{crt} \mu_c + \sum_{r' \notin \mathcal{R}_j \cup \mathcal{R}_j^o} q_{crr'} \mu_{crr'} \right)}{\sum_{r \in \mathcal{R}_j} \lambda_{cr} \cdot \left(q_{crt} + \sum_{r' \notin \mathcal{R}_j \cup \mathcal{R}_j^o} q_{crr'} \right)}.$$

Note that in the above approximation, the probabilities and rates of transitions for users in cell j and in $\mathcal{R}_j^o$ are weighted based on the arrival rates to the individual regions of cell j and of $\mathcal{R}_j^o$. Such a weighting assumes that the distribution of existing connections in the various regions is the same as the distribution of connection requests. Such an assumption would be appropriate if admission decisions were made independently of the particular region in which connection requests occurred. However, it is likely that controls which give preference to connection requests in certain regions result in better performance. As a result, a weighting that is based on arrival rates may not be appropriate. In many systems, however, the transition rates may not vary significantly from region to region and therefore such a weighting can still be very accurate.

Using the preceding approximations, for any feature configuration $\tilde{n} \in \tilde{\mathcal{N}}_j$ of cell j, the time until the next random event occurs for the Markov process associated with cell j is an exponential process with an approximate rate

$$\Lambda_{j\tilde{n}} = \sum_{c=1}^{C} \left(\tilde{\lambda}_{jc} + \tilde{\lambda}_{jc}^{o} + \tilde{n}_{jc}[\tilde{q}_{jcm}\tilde{\mu}_{jcm} + \tilde{q}_{jct}\tilde{\mu}_{jct}] + \tilde{n}_{j(c+C)}[\tilde{q}_{jcm}^{o}\tilde{\mu}_{jcm}^{o} + \tilde{q}_{jct}^{o}\tilde{\mu}_{jct}^{o}] \right).$$

Transition Probabilities As described in Sec. 2.2.4, the transition for any cell j from a state $i = (\tilde{n}, \omega) \in \mathcal{S}_j$ under control $u \in U(i)$ to a state $i' = (\tilde{n}', \omega') \in \mathcal{S}_j$ depends on two factors. The configuration for the next state is determined by the control exercised at state

Random State Component	Control		Next Configuration
Arrival $\omega \in \mathit{\Omega}_a \cup \mathit{\Omega}_{ao}$	Admit	$u = u_a$	$\tilde{n}' = \tilde{n} + \omega$
	Block	$u = u_b$	$\tilde{n}' = \tilde{n}$
Departure $\omega \in \mathit{\Omega}_d \cup \mathit{\Omega}_{do}$	None		$\tilde{n}' = \tilde{n}$
Movement $\omega \in \mathit{\Omega}_{m1} \cup \mathit{\Omega}_{m2}$	Maintain	$u = u_h$	$\tilde{n}' = \tilde{n} + \omega$
	Drop	$u = u_d$	$\tilde{n}' = \tilde{n} + \omega^-$

Table 4: List of the possible random events and the resulting configurations according to the exercised control.

i, as shown in Table 4, and the probability distribution of the random event for the next state depends on the next configuration, as shown in Table 5.

Next Event	Transition Probability from $(\tilde{n}, \omega)$ to $(\tilde{n}', \omega')$
$\omega' \in \mathit{\Omega}_a$	$\tilde{\lambda}_{jc}/\Lambda_{j\tilde{n}}$
$\omega' \in \mathit{\Omega}_{ao}$	$\tilde{\lambda}_{jc}^{o}/\Lambda_{j\tilde{n}}$
$\omega' \in \mathit{\Omega}_d$	$\tilde{n}_{jc}\tilde{q}_{jct}\tilde{\mu}_{ct}/\Lambda_{j\tilde{n}}$
$\omega' \in \mathit{\Omega}_{do}$	$\tilde{n}_{jc}\tilde{q}_{jct}^{o}\tilde{\mu}_{jct}^{o}/\Lambda_{j\tilde{n}}$
$\omega' \in \mathit{\Omega}_{m1}$	$\tilde{n}_{jc}\tilde{q}_{jcm}\tilde{\mu}_{jcm}/\Lambda_{j\tilde{n}}$
$\omega' \in \mathit{\Omega}_{m2}$	$\tilde{n}_{jc}\tilde{q}_{jcm}^{o}\tilde{\mu}_{jcm}^{o}/\Lambda_{j\tilde{n}}$

Table 5: List of the transition probabilities for the possible next random events.

The Cost Function Instead of associating costs with the current state and control, we need to reformulate costs so that they are associated with a particular cell and can be derived based on the features of the state that are selected. Essentially, we approximate the expected cost for a single stage corresponding to state i under control u by a sum of costs associated with each cell:

$$G(i,u) \approx \sum_{j=1}^{N} \tilde{G}_j(f_j(i),u),$$

where $\tilde{G}_j(f_j(i),u)$ contains the expected single stage costs corresponding to state i under control u and associated with cell j, and must be derived from the features from the state used to represent information relevant to cell j.

Determining how to decompose the costs according to the individual cells depends on the particular form of the cost function. Note that decomposing costs according to cells is straightforward if costs were only associated with blocking or accepting connection requests, and with dropping or maintaining existing connections. Then, in the case in which a connection is requested, costs could be associated with the cell in which the connection request is located, and in the case in which an existing user moves, costs could be associated with the cell to which the user is moving. Note that the features extracted from a state for any cell would need to contain the information necessary to collect the desired costs, as in the case of the current formulation example.

If costs depend on the entire state, e.g., if there were bonuses for maintaining a certain number of connections in the entire system, decomposing the costs would be much less straightforward unless the features selected included such information. As a result, selecting the features to represent the relevant information for a particular cell should take into consideration the form of the cost function.

3.3 Calculating $\tilde{v}$ and $\tilde{h}$.

In this section, we describe the third step involved in the cell-by-cell decomposition with feature extraction procedure: the process of determining the values of $\tilde{v}_j$ and $\tilde{h}_j$ given a feature-based MDP for each cell j. After these values are determined, the decision at each state is obtained by determining the cells affected by the decision, summing up the approximate costs associated with each of these cells, and selecting the control resulting in the smallest cost:

$$\psi(i) = \arg\min_{u\in U(i)} \tilde{H}(i,u) = \begin{cases} \arg\min\limits_{u_a,u_b} \left\{\tilde{H}(i,u_a), \tilde{H}(i,u_b)\right\}, & \text{if } \omega \in \Omega_a, \\ \arg\min\limits_{u_h,u_d} \left\{\tilde{H}(i,u_h), \tilde{H}(i,u_d)\right\}, & \text{if } \omega \in \Omega_m, \end{cases} \qquad i \in \mathcal{S}, \tag{7}$$

where

$$\tilde{H}(i,u) = G(i,u) - \bar{\tau}_i(u) \sum_{j\in\mathcal{J}(i)} \tilde{v}_j + \sum_{i'\in\mathcal{S}} \left[P_{ii'}(u) \sum_{j\in\mathcal{J}(i)} \tilde{h}_j(f_j(i')) \right].$$

Our method of determining the values for $\tilde{v}_j$ and $\tilde{h}_j$ is based on the discussion in [Ber95], §5.3. Before we consider the problem of determining for each cell j the values for $\tilde{v}_j$ and $\tilde{h}_j$, we consider the "auxiliary discrete-time average cost" version of the problem. Let $\mathcal{S}_j$ be the feature space associated with cell j and let γ be any scalar such that

$$0 < \gamma < \frac{\bar{\tau}_i(u)}{1 - \tilde{P}_{jii}(u)},$$

for all $i \in \mathcal{S}_j$ and $u \in U(i)$ with $\tilde{P}_{jii} < 1$. We define new transition probabilities for all i and $u \in U(i)$:

$$\hat{P}_{ii'}(u) = \begin{cases} \dfrac{\gamma \tilde{P}_{jii'}(u)}{\bar{\tau}_i(u)}, & \text{if } i' \neq i, \\ 1 - \dfrac{\gamma(1 - \tilde{P}_{jii}(u))}{\bar{\tau}_i(u)}, & \text{if } i' = i, \end{cases}$$

where the $\tilde{P}_{jii'}$ are the transition probabilities for the Markov process associated with cell j, and define

$$\hat{G}_j(i,u) = \frac{\tilde{G}_j(i,u)}{\bar{\tau}_i(u)}$$

as the expected cost per stage associated with cell j. If for any cell j, the scalar $\hat{v}_j$ and vector $\hat{h}_j$ satisfy

$$\hat{h}_j(i) = \min_{u \in U(i)} \left[\hat{G}_j(i,u) - \hat{v}_j + \sum_{i' \in \mathcal{S}_j} \hat{P}_{ii'}(u) \hat{h}_j(i') \right], \quad i \in \mathcal{S}_j,$$

then $\hat{v}_j$ and $\tilde{h}_j$ where

$$\tilde{h}_j(i) = \gamma \hat{h}_j(i), \quad i \in \mathcal{S}_j \tag{8}$$

satisfy

$$\tilde{h}_j(i) = \min_{u \in U(i)} \left[G_j(i,u) - \hat{v}_j \bar{\tau}_i(u) + \sum_{i' \in \mathcal{S}_j} \tilde{P}_{ii'}(u) \tilde{h}_j(i') \right], \quad i \in \mathcal{S}_j.$$

We now consider the problem of determining the values for $\hat{v}_j$ and $\hat{h}_j$. One method for calculating these values is a technique called relative value iteration. To simplify notation, we refer to $\hat{v}_j$ with v_j and $\hat{h}_j$ with h_j.

Let t be any fixed state, and initialize $h_j^0(i)$ to arbitrary scalars for all $i \in \mathcal{S}_j$. We run the following iterative algorithm

$$h_j^{k+1}(i) = (Th_j^k)(i) - (Th_j^k)(t), \quad i \in \mathcal{S}_j, \tag{9}$$

where

$$(Th_j^k)(i) = \min_{u \in U(i)} \left\{ \hat{G}_j(i,u) + \sum_{j \in \mathcal{S}} \hat{P}_{ii'}(u) h_j^k(i') \right\}.$$

Suppose at any particular stage, there are a total of m connected users regardless of class in the entire system. Due to capacity constraints, m must be bounded. There is then some $\epsilon > 0$ such that the probability that each of the next m stages involves some user completing his connection is greater than ϵ. As a result, for our Markov problem, there exists a positive integer m such that under any policy, the probability of reaching the "empty" state, the state in which there are no users connected, from any state in m stages is greater than some $\epsilon > 0$. It has been shown that for such a problem, the sequence $\{h_j^k\}$ generated by Eq. 9 converges to a vector h_j such that $(Th_j)(t)$ is equal to the optimal average cost per stage and h_j is an associated differential cost vector ([Ber95]). It also follows that

$$v_j^k = (Th_j^k)(t),$$

converges to the optimal average cost per stage.

Once the values for h_j and v_j are determined, we can obtain $\tilde{h}_j$ using Eq. 8, obtain $\tilde{v}_j$ using the relation $\tilde{v}_j = (T\tilde{h}_j)(t)$, and apply the decisions determined by Eq. 7. Equivalently, we could simply use the values obtained from h_j and v_j and apply the decision determined as follows for any state $i \in \mathcal{S}$:

$$\psi(i) = \begin{cases} \arg\min\limits_{u_a, u_b} \Bigg\{ \hat{G}(i,u_a) - \sum\limits_{j \in \mathcal{J}(i)} v_j + \sum\limits_{i' \in \mathcal{S}} \Bigg[\hat{P}_{ii'}(u_a) \sum\limits_{j \in \mathcal{J}(i)} h_j(f_j(i')) \Bigg], \\ \qquad \hat{G}(i,u_b) - \sum\limits_{j \in \mathcal{J}(i)} v_j + \sum\limits_{i' \in \mathcal{S}} \Bigg[\hat{P}_{ii'}(u_b) \sum\limits_{j \in \mathcal{J}(i)} h_j(f_j(i')) \Bigg] \Bigg\}, & \text{if } \omega \in \Omega_a, \\ \arg\min\limits_{u_a, u_b} \Bigg\{ \hat{G}(i,u_h) - \sum\limits_{j \in \mathcal{J}(i)} v_j + \sum\limits_{i' \in \mathcal{S}} \Bigg[\hat{P}_{ii'}(u_h) \sum\limits_{j \in \mathcal{J}(i)} h_j(f_j(i')) \Bigg], \\ \qquad \hat{G}(i,u_d) - \sum\limits_{j \in \mathcal{J}(i)} v_j + \sum\limits_{i' \in \mathcal{S}} \Bigg[\hat{P}_{ii'}(u_d) \sum\limits_{j \in \mathcal{J}(i)} h_j(f_j(i')) \Bigg] \Bigg\}, & \text{if } \omega \in \Omega_m, \end{cases} \tag{10}$$

where

$$\hat{G}(i,u) = \frac{G(i,u)}{\bar{\tau}_i(u)}.$$

Finally, note that since the term $\sum_{j \in \mathcal{J}(i)} v_j$ is independent of the control, it need not be calculated in making decisions and can be omitted from Eq. 10.

4 Computational Results

In this section, we describe the results of applying the approximation technique described in Sec. 3 to a simulated cellular system. We first describe the simulated system in Sec. 4.1. We then describe the specific details involved in implementing the approximation technique for this particular system, and present some computational results in Sec. 4.2.

4.1 Simulation Model

In actual systems, cells can vary significantly in shape, size, as well as in other factors. However, the approaches that we have considered do not require that cells be of any specific or uniform geometry. In fact, we only need to be able to determine the arrival rate of call requests to particular regions of the cell and the likelihood that users will move from one region to another. These factors are influenced by the shape and size, but our model includes the ability to specify these details by allowing the number of regions per cell be variable as well as by allowing arrival rates and movement probabilities be specified for individual regions. We therefore consider a system in which the geometry has been simplified in order to keep the implementation and collection of results straightforward. Also for the sake of simplicity, our system consists of a single user class.

The description of the simulated system is divided into the following four parts, with each description providing the set of parameters that need to be specified for each simulation.

1. description of the geometry of the system;
2. description of path gains for each cell region to each base station;
3. description of each user's behavior;
4. description of the objective function.

Geometry We consider a two-dimensional rectangular system in which cells are of hexagonal shape. Each cell is divided into two concentric zones, and the outer zone is further divided into six identical regions, as shown in Figure 6. The parameters that need to be specified to determine the geometry for a particular simulation are

1. M_v, the number of cells in the vertical direction,
2. M_h, the number of cells in the horizontal direction, and
3. E_i, the fraction of the area of a cell that comprises the inner region. (Note that E_o, the fraction of the area of a cell that comprises an outer region is $(1 - E_i)/6$.)

Path Gains We assume that the path attenuation gain from any part of an inner region of any cell to that cell's base station is A_1; i.e., if a user in an inner region of a cell transmits at a power level of w, the signal level received at the cell's base station is wA_1. We also assume that the path gain from a periphery region of any cell to that cell's base station is $A_2 < A_1$. We assume that the power assignment algorithm assigns power levels of W_1 and W_2 to users in an inner region of a cell and users in a periphery region, respectively,

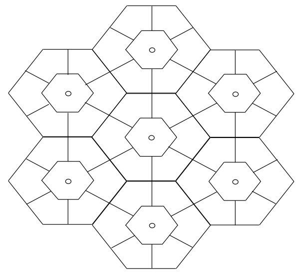


Figure 6: Cell Model: Each cell consists of an inner region and six identical periphery regions.

and that $W_1 A_1 = W_2 A_2$; i.e., we assume that power levels are assigned according to the inverted channel power control scheme so that the signal levels received by a base station is the same for all users that are connected to that base station. We also assume that mobiles within the inner region of a cell, where transmissions occur at relatively low powers, cause negligible interference in neighboring cells, while mobiles within the periphery regions, where transmissions occur at relatively high powers, may cause significant interference in the neighboring cells that border the particular outer region. We therefore assume that a signal from a user in another cell that is in a region that is adjacent to cell j is attenuated by a factor of $A_3 < A_2$ by the time it is received at the base station of cell j. Fig. 7 illustrates the various possibilities.

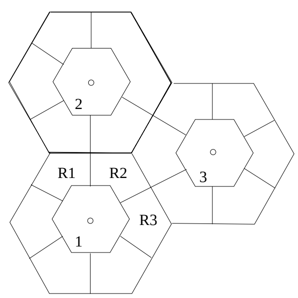


Figure 7: The path gain from the inner region of any cell to its own base station is A_1 and 0 elsewhere. The path gain from region $R1$ to cells 1 and 2 are A_2 and A_3, respectively, and 0 elsewhere. The path gain from $R3$ to cells 1 and 3 are A_2 and A_3, respectively, and 0 elsewhere. The path gain from $R2$ to cells 1, 2, and 3 are A_2, A_3, and A_3, respectively, and 0 elsewhere.

Given our assumptions, the signals received at any base station are of two levels: transmissions from any user within the cell are received at a signal level of $I_1 = W_1 A_1 = W_2 A_2$, and transmissions from any user in an adjacent region of a neighboring cell are received at a signal level of $I_2 = W_2 A_3$. The two signal levels essentially specify the three different path gains and we use them as the parameters for the simulated system. In addition, we assume that each user must satisfy the following signal to noise requirement:

$$\frac{I_1}{n_1 I_1 + n_2 I_2} > T,$$

where n_1 is the number of users within this cell, n_2 is the number of users in an adjacent region of a neighboring cell of j, and T is a given threshold. We assume that the receiver noise at each base station is negligible. The parameters that need to be specified are then

1. I_1, the signal level received at a base station from any user within the cell,
2. I_2, the signal level received at a base station from any user in an adjacent region of a neighboring cell, and
3. T, the signal to noise threshold ratio.

User Behavior The arrival rate of new connection requests to cell j is λ_j. The probability that the connection request is in any particular region of a cell is proportional to the size of the region. The directions of travel of users can be unrestricted, in which case a user can move to any region to which his current region is adjacent, or specified. One possible direction specification is to allow all users to only move to regions that are "above" the current region. The probability that a user moves from one cell region to another is q. If the user moves to another region, he is equally likely to move to any region to which he is allowed to move. We assume that the probability that a user moves more than once is zero. The lengths of time that a user that moves stays in its originating region and in its destination region are both exponentially distributed with mean μ. The lengths of a call of a user that is stationary are also exponentially distributed with mean μ.

The parameters that need to be specified to describe user arrivals, departures, and movements are as follows:

1. λ_j, the arrival rate of call requests to each cell,
2. μ, the rate at which a user moves to another region or completes a connection,
3. q, the probability that a user moves to another region, and
4. whether there are any restrictions on movement directions.

Objective Function We assign costs to blocking a connection request and to dropping a connection for a user that moves from one region to another. For any particular state i when control u is exercised, the cost per stage $G(i, u)$ is

$$G(i,u) = \begin{cases} B, & \text{if } \omega \in \Omega_a, u = u_b, \\ D, & \text{if } \omega \in \Omega_m, u = u_d, \\ 0, & \text{otherwise.} \end{cases}$$

The objective function which we wish to minimize is a linear combination of the expected average number of blocks and of drops per unit time:

$$\lim_{N\to\infty} \frac{1}{E(t_N)} \sum_{k=1}^{N} E\{G_k\},$$

where G_k is the cost of the kth stage.

The parameters that need to be specified to describe the cost function are as follows:

1. B, the cost of blocking a user, and
2. D, the cost of dropping a user.

We assume that the costs B and D are positive quantities and that the cost of dropping a call is worse than that of blocking a call: $D > B$.

4.2 Computational Results using Cell-by-Cell Decomposition

In this section, we describe our results of applying the approximate decomposition with feature extraction procedure to the simulated system presented in Sec. 4.1. Using various selections of features and decomposing the problem into individual cells, approximations were created with state spaces sufficiently small for dynamic programming to be applied as described in Sec. 3.3 to obtain the average cost per stage v_j and differential cost vector h_j for the feature-based MDP associated with each cell j.

In our simulations, we formulated feature-based MDPs for three sets of features. Table 6 indicates the features that were used for each set. Note that the formulation for the first feature set was presented in Sec. 3.2.

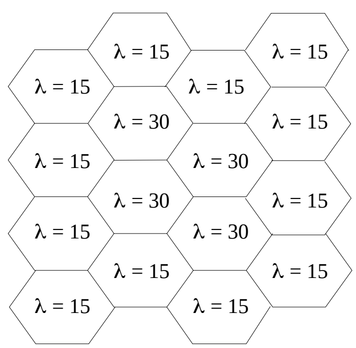
Feature Description	Feature Set		
	1	2	3
# existing connections in cell	x	x	x
# existing connections in adjacent regions of neighboring cells	x	x	x
# existing connections in cell that are positioned so that they can move to another cell		x	
# existing connections in adjacent regions of neighboring cells positioned so that they can move to given cell			x

Table 6: List of the features and those used in each feature set.

After the selection of a set of features, we formulated an appropriate feature-based MDP for each cell as illustrated in Sec. 3.2. We then applied relative value iteration on the resulting formulations to obtain the average costs per stage and differential cost vector for the approximate problems as described in Sec. 3.3. We have applied the policies resulting from these cost functions on various random sequences of a large number of simulated events. We refer to each of these sequences as a particular data set. Each data set consisted of simulated events over a period of 3000 time units.

The problems we considered consisted of a system of four cells by four cells with a single user class. The area of the inner region of a cell was set to $E_i = 1/4$ and the area of each periphery region was set to $E_o = 1/8$. Calls generated a unit of interference ($I_1 = 1$) at the base station of the cell in which the call was located. Calls in adjacent regions of neighboring cells generated a fraction 0.3 of a unit of interference ($I_2 = 0.3$). The total amount of interference that was acceptable was set to 50.15, resulting in a threshold of $T = 1/50.15$. Arrival rates varied from $\lambda = 15$ to 30 call requests per unit time. The call completion rate was set to $\mu = 1$ per unit time. The probability that a particular call moves varied from $q = .2$ to $q = .4$. In several of the runs, users that move were equally likely to move to any neighboring regions; in others, users that move were equally likely to move to any neighboring region that was *above* the region in which the call was initiated. The cost associated with a blocked call was $B = 0.1$, and the cost associated with a dropped call was $D = 1.0$.

For the first six data sets, the arrival rate of connection requests to each cell was the same and varied from 20 to 25 requests per unit time. In the seventh data set, the arrival rate of connection requests was 30 per unit time to the four inner cells and 15 per unit time to the twelve outer cells.



The specific details for each data set are provided in Table 7.

The number of value iterations we performed to obtain our DP policies varied from 40 to 500. We compared the results of applying the these policies on the above data sets to those when the greedy algorithm or a reservation policy was applied. Under a greedy algorithm, a user is admitted if there are sufficient resources to handle the user; under a reservation policy, a user is admitted if there are sufficient resources to handle a certain number of later handoffs in addition to the user. In our computations, the optimal number of later handoffs for which to reserve resources was always one. The greedy and reservation algorithms are

Data Set	Arrival Rate	Travel Direction	Prob of Movement	Total Req'd Connections	# Attempted Movements
1	25	Uniform	0.2	1197265	232824–234402
2	25	Upwards	0.2	1199966	224420–226087
3	20	Uniform	0.3	958454	285919–275728
4	20	Upwards	0.3	959784	275128–275728
5	25	Uniform	0.4	1198456	437022–444531
6	25	Upwards	0.4	1199540	413321–426210
7	15 or 30	Uniform	0.3	898742	254554–256559

Table 7: List of the simulation data sets and their parameters. Each simulation was run for 3000 time units. The expected length of a connection was 1 unit time.

generally used in current systems.

The results of applying our DP policies to data sets 1 and 2 are provided in Tables 8 and 9. The results for data sets 3 through 7 are similar. In each of the runs that were conducted,

Cell-by-Cell Decomposition Applied to Data Set 1							
Feature Set	# iters	# of blocks	% blocked	# of drops	B+10·D (Cost)	% Improvement over Greedy Pol	% Improvement over Reserv Pol
1	40	26386	2.20	428	30666	22.55	8.66
	100	26785	2.24	343	30215	23.69	10.00
	500	26991	2.25	311	30101	23.97	10.34
2	40	26389	2.20	429	30679	22.51	8.62
	100	26788	2.24	342	30208	23.70	10.01
	500	26986	2.25	310	30086	24.01	10.39
3	40	26380	2.20	437	30750	22.33	8.41
	100	26751	2.23	349	30241	23.62	9.92
	500	26848	2.24	340	30248	23.60	9.90
Greedy		25623	2.14	1397	39593	–	–
Reserv		33443	2.79	13	33573	15.20	–

Table 8: Results of applying cell-by-cell decomposition with feature extraction to Data Set 1.

applying dynamic programming to formulations resulting from cell-by-cell decomposition with feature extraction performed significantly better than the greedy policy and somewhat better than the reservation policy. The improvement in cost varied from twenty to nearly forty-five percent over the greedy policy. In a few cases when the number of value iterations

Cell-by-Cell Decomposition Applied to Data Set 2							
Feature Set	# iters	# of blocks	% blocked	# of drops	B+10·D (Cost)	% Improvement over Greedy Pol	% Improvement over Reserv Pol
1	40	28696	2.39	367	32366	26.36	8.29
	100	28995	2.42	297	31960	27.28	9.44
	500	29390	2.45	248	31870	27.49	9.69
2	40	28644	2.39	367	32314	26.48	8.44
	100	28989	2.42	298	31969	27.26	9.41
	500	29384	2.45	251	31894	27.43	9.63
3	40	28676	2.39	372	32396	26.29	8.20
	100	28814	2.40	347	32284	26.54	8.52
	500	28804	2.40	348	32284	26.54	8.52
Greedy		26670	2.22	1728	43950	–	–
Reserv		35051	2.92	24	35291	19.70	–

Table 9: Results of applying cell-by-cell decomposition with feature extraction to Data Set 2.

was small, the DP solution performed slightly worse than the reservation policy, but generally the DP solution resulted in a cost that improved upon the reservation policy from a negligible amount to about thirteen percent. Increasing the number of iterations generally improved performance, but not consistently nor substantially. When the utilization of the system is high, the DP solution outperformed the greedy policy to a greater extent, but outperformed the reservation policy to a lesser extent. In general, performance did not seem to depend on which of the three feature sets were used. In fact, the first feature set often outperformed the others in spite of the fact that its state space is the coarsest.

Table 10 summarizes the best results of applying cell-by-cell decomposition with feature extraction.

5 Summary

In this paper, we formulated the admission control problem as a Markov decision problem. One of the main advantages of such a formulation is that it can incorporate an arbitrary amount of detail necessary to describe real cellular systems. While dynamic programming can be used to solve such problems, the large size of the state space makes this impractical. We proposed an approximate dynamic programming technique, cell-by-cell decomposition with feature extraction, which involved creating an approximation of the original model with a state space sufficiently small so that dynamic programming could be applied. The real-time computation required by our proposed technique is small and is easily distributed between

Data Set Number	Feature Set 1 % Imp/ Greedy	Feature Set 1 % Imp/ Reserv	Feature Set 2 % Imp/ Greedy	Feature Set 2 % Imp/ Reserv	Feature Set 3 % Imp/ Greedy	Feature Set 3 % Imp/ Reserv
1	23.97	10.34	24.01	10.39	23.62	9.92
2	27.49	9.69	27.43	9.63	26.54	8.52
3	25.02	12.60	23.91	11.31	25.05	12.64
4	30.77	11.84	30.77	11.84	29.99	10.84
5	40.40	3.08	40.39	3.07	39.33	1.33
6	44.70	4.71	44.43	4.24	43.96	3.43
7	33.39	7.23	33.14	6.89	32.21	5.59

Table 10: Summary of the best results of applying cell-by-cell decomposition with feature extraction to the seven data sets. The cost improvement of the results for each of the feature sets over the results of applying the greedy and reservation policies are provided.

the cells. Our comparisons on a simulated systems with commonly used heuristic policies have shown significant improvements in performance for our methodology.

Acknowledgments

This research was supported by NSF under grant NCR-9622636.

References

[AAN96] P. Agrawal, D. K. Anvekar, and B. Narendran. Channel management policies for handovers in cellular networks. *Bell Labs Technical Journal*, pp. 97–110, Autumn 1996.

[AS96] M. Asawa and W. E. Stark. Optimal scheduling of handoffs in cellular networks. *IEEE/ACM Transactions on Networking*, 4:428–441, June 1996.

[Ber95] D. P. Bertsekas. *Dynamic Programming and Optimal Control, Vols I and II.* Athena Scientific, Belmont, MA, 1995.

[BWE95] C. Barnhart, J. Wieselthier, and A. Ephremides. Admission-control policies for multihop wireless networks. *Wireless Networks*, 1:373–387, 1995.

[CG98] Y. Chang and E. Geraniotis. Optimal policies for handoff and channel assignment in networks of leo satellites using cdma. *Wireless Networks*, 4:181–187, 1998.

[Gue88] R. Guerin. Queueing-blocking system with two arrival streams and guard channels. *IEEE Transactions on Communications*, 36(2):153–163, 1988.

[HR86] D. Hong and S. S. Rappaport. Traffic model and performance analysis for cellular mobile radio telephone systems with prioritized and nonprioritized handoff procedures. *IEEE Transactions on Vehicular Technology*, VT-35(3):77–92, Aug. 1986.

[LAN97] D. A. Levine, I. F. Akyildiz, and M. Naghshineh. A resource estimation and call admission algorithm for wireless multimedia networks using the shadow cluster concept. *IEEE/ACM Transactions on Networking*, 5:1–12, Feb. 1997.

[PG85] E. C. Posner and R. Guerin. Traffic policies in cellular radio that minimize blocking of handoff calls. *ITC 11.*, Kyoto, Japan 1985.

[SS97] M. Sidi and D. Starobinski. New call blocking versus handoff blocking in cellular networks. *Wireless Networks*, 3:15–27, 1997.

[TJ92] S. Tekinay and B. Jabbari. A measurement-based prioritization scheme for handovers in mobile cellular networks. *IEEE Journal on Selected Areas in Communications*, 10:1343–1350, 1992.