Paging and Registration in Cellular Networks: Jointly Optimal Policies and an Iterative Algorithm

February 18, 2007

Bruce Hajek, Kevin Mitzel, and Sichao Yang
Department of Electrical and Computer Engineering
University of Illinois at Urbana Champaign
1308 W. Main Street, Urbana, IL 61801

Abstract—This paper explores optimization of paging and registration policies in cellular networks. Motion is modeled as a discrete-time Markov process, and minimization of the discounted, infinite-horizon average cost is addressed. The structure of jointly optimal paging and registration policies is investigated through the use of dynamic programming for partially observed Markov processes. It is shown that there exist policies with a certain simple form that are jointly optimal, though the dynamic programming approach does not directly provide an efficient method to find the policies.

An iterative algorithm for policies with the simple form is proposed and investigated. The algorithm alternates between paging policy optimization and registration policy optimization. It finds a pair of individually optimal policies, but an example is given showing that the policies need not be jointly optimal. Majorization theory and Riesz’s rearrangement inequality are used to show that jointly optimal paging and registration policies are given for symmetric or Gaussian random walk models by the nearest-location-first paging policy and distance threshold registration policies.

Index Terms—Paging, registration, cellular networks, partially observed Markov processes, majorization, rearrangement theory

I. INTRODUCTION

The growing demand for personal communication services is increasing the need for efficient utilization of the limited resources available for wireless communication. In order to deliver service to a mobile station (MS), the cellular network must be able to track the MS as it roams. In this paper, the problem of minimizing the cost of tracking is discussed. Two basic operations involved in tracking the MS are paging and registration.

There is a tradeoff between the paging and registration costs. If the MS registers its location within the cellular network more often, the paging costs are reduced, but the registration costs are higher. The traditional approach to paging and registration in cellular systems uses registration areas which are groups of cells. An MS registers if and only if it changes registration area. Thus, when there is an incoming call directed to the MS, all the cells within its current registration area are paged. Another method uses reporting centers [3]. An MS registers only when it enters the cells of reporting centers, while every search for the MS is restricted to the vicinity of the reporting center to which it last reported.

Some dynamic registration schemes are examined in [4]: time-based, movement-based, and distance-based. These policies are threshold policies and the thresholds depend on the MS motion activities. In [14], dynamic programming is used to determine an optimal state-based registration policy. Work in [2] considers congestion among paging requests for multiple MSs, and considers overlapping registration regions. Basic paging policies can be classified as follows:

- **Serial Paging.** The cellular network pages the MS sequentially, one cell at a time.
- **Parallel Paging.** The cellular network pages the MS in a collection of cells simultaneously.

Serial paging policies have lower paging costs than parallel paging policies, but at the expense of larger delay. The method of parallel paging is to partition the cells in a service region into a series of indexed groups referred to as paging areas. When a call arrives for the MS, the cells in the first paging area are paged simultaneously in the first round and then, if the MS is not found in the first round of paging, all the cells in the second paging area are paged, and so on. Given disjoint paging areas, searching them in the order of decreasing probabilities minimizes the the expected number of searches [19]. This paging order is denoted as the maximum-likelihood serial paging order. An interesting topic of paging is to design the optimal paging areas within delay constraints [12, 19, 22]. However, in this paper, we consider only serial paging polices.

Each paper mentioned above assumes a certain class of paging or registration policy. Given one policy (paging policy or registration policy) and the parameters of an assumed
motion model, the counterpart policy (registration policy or paging policy, respectively) is found. For instance, the optimal paging policy is identified in [19] for a given registration policy. This is shown as the top branch of Figure 1. Conversely, an expanding “ping-pong” order paging policy suited to the given motion model is assumed in [14]. With this knowledge, dynamic programming is applied to solve for the optimal registration policy. This corresponds to the bottom branch of Figure 1.

Several studies have addressed minimizing the costs, considering the paging and registration policies together [1, 14, 19, 20]. In [20], a timer-based registration policy combined with maximum-likelihood serial paging is introduced. The minimum paging cost can be represented by the distribution of locations where the MS last reported. Then an optimal timer threshold is selected to minimize the total cost of registration and paging. By contrast, a movement-based registration policy is used in [1]. An improvement of [20] is given in [21] by assuming that the MS knows not only the current time, but also its own state and the conditional distribution of its state given the last report. This is a state-based registration policy and is aimed to minimize the total costs by running a greedy algorithm on the potential costs. Although the papers discuss the paging and registration policy together, they don’t consider jointly optimizing the policies.

The contributions of this paper are as follows. The structure of jointly optimal paging and registration policies is identified. It is shown that the conditional probability distribution of the states of an MS can be viewed as a controlled Markov process, controlled by both the paging and registration policies at each time. Dynamic programming is applied to show that the jointly optimal policies can be represented compactly by certain reduced complexity laws. An iterative algorithm producing a pair of RCLs is proposed based on closing the loop in Figure 1. The algorithm is a heuristic which merges the approaches in [14] and [19]. Several examples are given. The first example is an illustration of numerical computation of an individually optimal policy pair. The second example is a simple one illustrating that individually optimal policies are not necessarily jointly optimal. Finally, three more examples are given based on random walk models of motion: one-dimensional discrete state symmetric, multidimensional symmetric, and multidimensional Gaussian. Majorization theory and Riesz’s rearrangement inequality are used to show that jointly optimal paging and registration policies are given for these random walk models by the nearest-location-first paging policy and distance threshold registration policies.

The paper is organized as follows. Notation and cost functions are introduced in Section II. Jointly optimal policies are investigated in Section III. The iterative optimization formula for computing individually optimal policy pairs is developed in Section IV. The first two examples are given in Section V and the random walk examples are given in Section VI. Conclusions are given in Section VII.

1Earlier versions of this work appeared in [10, 11].
for \( t \geq 0 \),
\[
N_t = \sigma((I_{P_t}, N_s, I_{R_t} : 1 \leq s \leq t), \\
(X(s) : 1 \leq s \leq t \text{ and } I_{P_t \cup R_t} = 1))
\]

The initial state \( x_0 \) is treated as a constant, so even though it is known to the network it is not included in the definition of \( \mathcal{N}_t \). Note that the initial \( \sigma \)-algebra \( \mathcal{N}_0 \) is the trivial \( \sigma \)-algebra: \( \mathcal{N}_0 = \{\emptyset, \Omega\} \).

When the MS is to be paged, the cells are to be searched sequentially according to a permutation \( a \) of the cells. The associated paging order vector \( r = (r_j : j \in S) \) is such that for each state \( j \), \( r_j \) is the number of cells that must be paged until the cell for state \( j \) comes up, and the MS is reached. For example, suppose \( S = \{1, 2, 3, 4, 5, 6\} \) and \( C = \{c_1, c_2, c_3\} \) with \( c_1 = \{1, 2\} \), \( c_2 = \{3, 4\} \), and \( c_3 = \{5, 6\} \). Then if the cells are search according to the permutation \( a = (c_2, c_1, c_3) \), meaning to search cell \( c_2 \) first, \( c_1 \) second, and \( c_3 \) third, then the paging order vector is \( r = (2, 1, 2, 1, 3, 3) \). A paging policy \( u \) is a collection \( u = (u(t) : t \geq 1) \) such that for each \( t \geq 1 \), \( u(t) \) is an \( \mathcal{N}_{t-1} \) measurable random variable with values in the set of paging order vectors. Note that \( \mathcal{N}_t = I_{P_t}uX(t)(t) \).

C. Registration policy notation

Let \( \mathcal{M}_t \) denote the \( \sigma \)-algebra representing the information available to the MS by time \( t \), after the paging and registration decisions for time \( t \) have been made and carried out. Thus,
\[
\mathcal{M}_t = \sigma(X(s), I_{P_t}, N_s, I_{R_t} : 1 \leq s \leq t).
\]
The MS also knows the initial position \( x_0 \), which is treated as a constant. In practice an MS wouldn’t learn \( N_s \), the number of pages used to find the MS at time \( s \). While we assume such information is available to the MS, we will see that optimal policies need not make use of the information. With this definition, we have \( \mathcal{N}_t \subset \mathcal{M}_t \), meaning that the MS knows everything the network knows (and typically more).

When the MS has to decide whether to register at time \( t \), it already has the information \( \mathcal{M}_{t-1} \). In addition it knows \( X(t) \) and \( I_{P_t} \). If the MS is paged at time \( t \), then the network learns the state of the MS as a result, so there is no advantage for the MS to register at time \( t \). Thus, we assume without loss of generality that the MS does not register at time \( t \) if it is paged at time \( t \). This leads to the following definition.

A registration policy \( v \) is a collection \( v = (v(t) : t \geq 1) \) such that for each \( t \geq 1 \), \( v(t) \) is an \( \mathcal{M}_{t-1} \) measurable random vector with values in \([0, 1]^S\) with the following interpretation. Given the information \( \mathcal{M}_{t-1} \), if \( X(t) = l \) and if the MS is not paged at time \( t \), then the MS registers with probability \( v_l(t) \).

D. Cost function

Let \( \beta \) be a number with \( 0 < \beta < 1 \), called the discount factor. An interpretation of \( \beta \) is that \( 1/(1-\beta) \) is the rough time horizon of interest. Given a paging policy \( u \) and registration policy \( v \), the expected infinite horizon discounted cost \( C(u, v) \) is defined as
\[
C(u, v) = E \left[ \sum_{t=1}^{\infty} \beta^t \{P_{I_{P_t}N_t + R_{I_{R_t}}} \} \right]. 
\] (1)
The pair \((u, v)\) is jointly optimal if \( C(u, v) \leq C(u', v') \) for every paging policy \( u' \) and registration policy \( v' \).

III. JOINTLY OPTIMAL POLICIES

This section investigates the structure of jointly optimal policies by using the theory of dynamic programming for Markov control problems with partially observed states. While the structure results do not directly yield a computationally feasible solution, they shed light on the nature of the problem.

Intuitively, on one hand, the paging policies are selected based on the past of the registration policy, because the past of the registration policy influences the conditional distribution of the MS state. On the other hand, by the nature of dynamic programming, the optimal choice of registration policy at a given time depends on future costs, which are determined by the future of the registration policy. To break this cycle, we consider the problem entirely from the viewpoint of the network. In order that current decisions not depend on past actions, the state space is augmented by the conditional distribution of the state of the MS given the information available to the network.

A. Evolution of conditional distributions

For \( t \geq 0 \), let \( w(t) \) be the conditional probability distribution of \( X(t) \), given the observations available to the network up to time \( t \) (including the outcomes of a report at time \( t \), if there was any). That is, \( w_j(t) = P[X(t) = j | N_t] \) for \( j \in S \). Note that, with probability one, \( w(t) \) is a probability distribution on \( S \). Intuitively, the network can control the distribution valued process \( w(t) \) by dictating the registration policy of the MS. Since \( \mathcal{N}_0 \) is the trivial \( \sigma \)-algebra and \( X(0) = x_0 \), the initial conditional distribution is given by \( w(0) = \delta(x_0) \).

While the network may not know the recent past trajectory of the state process, it can still estimate the registration probabilities used by the MS. In particular, as shown in the next lemma, the estimate \( \hat{w}_j(t) \), defined by \( \hat{w}_j(t) = \frac{E[v_j(t)X(t) = j | N_{t-1}]}{P[X(t) = j | N_{t-1}]} \), plays a role in how the network can recursively update the \( w(t) \)’s. In more conventional notation, we have
\[
\hat{w}_j(t) = \frac{E[v_j(t)I(X(t) = j) | N_{t-1}]}{P[X(t) = j | N_{t-1}]}.
\]

Define a function \( \Phi \) as follows. Let \( w \) be a probability distribution on \( S \) and let \( d \in [0, 1]^S \). Let \( \Phi(w, d) \) denote the probability distribution on \( S \) defined by
\[
\Phi_t(w, d) = \frac{\sum_{j \in S} w_j P_{I_{P_j} + R_{I_{R_j}}} (1 - d_j)}{\sum_{j \in S} \sum_{j' \in S} w_{j'} P_{I_{P_{j'}}} (1 - d_{j'})}.
\]
\( \Phi(w, d) \) is undefined if the denominator in this definition is zero. The meaning of \( \Phi \) is that if at time \( t \) the network knows that \( X(t) \) has distribution \( w \), if no paging occurs at time \( t + 1 \), and if the MS registers at time \( t + 1 \) with probability \( d_{X(t+1)} \), then \( \Phi(w, d) \) is the conditional distribution of \( X(t+1) \) given...
no registration occurs at time $t+1$. This interpretation is made precise in the next lemma. The proof is in the appendix.

**Lemma 3.1:** The following holds, under the paging and registration policies $u$ and $v$:

$$
w(t+1) = \delta(X(t+1))I_{P_{t+1} \cup R_{t+1}} + \Phi(w(t), \hat{v}(t+1))I_{P_{t+1} \cap R_{t+1}^c}.
$$

(2)

**B. New state process**

For $t \geq 1$, let $\Theta(t) = (w(t), I_P, N_t, I_{R_t})$. Note that the $t$th term in the cost function is a function of $\Theta(t)$. Note also that $\Theta(t)$ is measurable with respect to $N_t$, so that the network can calculate $\Theta(t)$ at time $t$ (after possible paging and registration). Moreover, the first coordinate of $\Theta(t)$, namely $w(t)$, can be updated with increasing $t$ with the help of Lemma 3.1. The random process $(\Theta(t) : t \geq 0)$ can be viewed as a controlled Markov process, adapted to the family of $\sigma$-algebras $(N_t : t \geq 0)$ with controls $(u(t), \hat{v}(t) : t \geq 1)$. Note that $u(t+1)$ and $\hat{v}(t+1)$ are each $N_t$ measurable for each $t \geq 0$. The one-step transition probabilities for $(\Theta(t))$ are given as follows. (The variables $j$ and $l$ range over the set of states $S$.)

| $\Theta(t+1)$ | Probability |
|-------------|-------------|
| $(\delta(l), 1, u(t+1), 0)$ | $\lambda_p \sum_j w_j(t)p_{jl}$ |
| $(\delta(l), 0, 0, 1)$ | $(1 - \lambda_p) \sum_j w_j(t)p_{jl}\hat{v}(t+1)$ |
| $(\Phi(w(t), \hat{v}(t+1)), 0, 0, 0)$ | $(1 - \lambda_p) \sum_j w_j(t)p_{jl}(1 - \hat{v}(t+1))$ |

Observe that although the MS uses a registration policy $v$, the one-step transition probabilities for $\Theta$ depend only on $\hat{v}$. Moreover, $\hat{v}$ is itself a registration policy. Indeed, since $N_{t-1} \subset M_{t-1}$, $\hat{v}(t)$ is $M_{t-1}$ measurable, and it takes values in $[0, 1]^S$. If $\hat{v}$ were used instead of $v$ as a registration policy by the MS, the one-step transition probabilities for $\Theta$ would be unchanged. Thus, the policy $\hat{v}$ is adapted to the family of $\sigma$ algebras $(N_t : t \geq 0)$, and it yields the same cost as $v$. Therefore, without loss of generality, we can restrict attention to registration policies $\hat{v}$ that are adapted to $(N_t : t \geq 0)$.

Combining the observations summarized in this section, we arrive at the following proposition.

**Proposition 3.1:** The original joint optimization problem is equivalent to a Markov optimal control problem with state process $(\Theta(t) : t \geq 0)$ adapted to the family of $\sigma$-algebras $(N_t : t \geq 0)$, with controls $(u(t), \hat{v}(t) : t \geq 1)$.

**C. Dynamic programming equations**

Above it was assumed that $w(0) = \delta(x_0)$, where $x_0$ is the initial state of the MS, assumed known by the network. In order to apply the dynamic programming technique, in this section the initial distribution $w(0)$ is allowed to be any probability distribution on $S$. It is assumed that the network knows $w(0)$ at time zero, and that the initial state of the MS is random, with distribution $w(0)$. The evolution of the system as described in the previous section is well defined for an arbitrary initial distribution $w(0)$. Let $E_w$ denote conditional expectation in case the initial distribution $w(0)$ is taken to be $\omega$. The initial $\sigma$-algebra $N_0$ is still the trivial $\sigma$-algebra, because $w(0)$ is treated as a given constant.

Define the cost with $n$ steps to go as

$$U_n(w) = \min_{u, \hat{v}} \{ \beta \sum_{t=1}^{n} \beta^t \{ P_{I_{P_t}N_t} + R_{I_{R_t}} \} \}.$$

Next apply the backwards solution method of dynamic programming, by separating out the $t = 1$ term in the cost for $n + 1$ steps to go. This yields

$$U_{n+1}(w) = \min_{u, \hat{v}} \left[ \lambda_p P \sum_j w_j p_{jl} u(1) + (1 - \lambda_p) R \sum_j w_j p_{jl} \hat{v}(1) + E_w [ \sum_{t=2}^{n+1} \beta^{t-1} \{ P_{I_{P_t}N_t} + R_{I_{R_t}} \}] \right].$$

Note that $u(1)$ and $\hat{v}(1)$ are both measurable with respect to the trivial $\sigma$-algebra $N_0$. Therefore these controls are constants. Henceforth we write $d$ for the registration decision vector $\hat{v}(1)$. The vector $d$ ranges over the space $[0, 1]^S$.

The first sum in the expression for $U_{n+1}(w)$ involves the control policies only through the choice of the paging order vector $u(1)$. This sum is simply the mean number of single-cell pages required to find the MS given that the state of the MS has distribution given by the product $wP$, where $P$ is the matrix of state transition probabilities. It is well known that the optimal search order is to first search the cell with the largest probability, then search the cell with the second largest probability, and so on [19]. Ties can be broken arbitrarily.

The first sum in the expression for $U_{n+1}(w)$ can thus be replaced by $s(wP)$, where $s(q)$ denotes the mean number of single cell pages required to find the MS given that the state of the MS has distribution $q$ and the optimal paging policy is used. We remark that Massey [17] explored comparisons between $s(w)$, in the case of one state per cell, and the ordinary entropy, $H(w) = \sum_i w_i \log w_i$. The measure $s(w)$ was called the guessing entropy in [8], and work continues to compare it to other forms of entropy [9].

The dynamic programming equation thus becomes

$$U_{n+1}(w) = \beta \lambda_p P s(wP)\min_d \left[ \sum_j \sum_l w_j p_{jl} \left( \lambda_p U_n(\delta(l)) + (1 - \lambda_p) d(l)(R + U_n(\delta(l))) \right) \right] + \left( \sum_j \sum_l w_j p_{jl}(1 - d(l)) \right) U_n(\Phi(w, d)).$$

Formally we denote this equation as $U_{n+1} = T(U_n)$. By a standard argument for dynamic programming with discounted cost, $T$ has the following contraction property:

$$\sup_w |T(U) - T(U')| \leq \beta \sup_w |U - U'|$$

for any bounded, measurable functions $U$ and $U'$, defined on the space of all probability distributions $w$ on $S$. Consequently
there exists a unique $U_*$ such that $T(U_*) = U_*$, and $U_n \to U$ uniformly as $n \to \infty$. Moreover, $U_*$ is the minimum possible cost, and a jointly optimal pair of paging and registration policies is given by a pair $(\bar{f}, \bar{g})$ of state feedback controls, for the state process $(w(t))$. A jointly optimal control is given by $w(t) = \bar{f}(w(t - 1))$ and $v(t) = \bar{g}(w(t - 1))$, where $\bar{f}$ and $\bar{g}$ are determined as follows. For any probability distribution $w$ on $S$, $f(w)$ is the paging order vector for paging the cells in order of decreasing probability under distribution $wP$, and $g(w)$ is a value of $d$ that achieves the minimum in the right hand side of (3) with $U_n$ replaced by $U_*$. Then if there is no report at time $t + 1$, the conditional distribution $w(t)$ is updated simply by:

\[ w(t + 1) = \Phi(w(t), g(w(t))) \]  

Clearly under such stationary state feedback control laws $(\bar{f}, \bar{g})$, the process $(w(t) : t \geq 0)$ is a time-homogeneous Markov process. Note that the optimal mapping $\bar{f}$ does not depend on $\bar{g}$.

**Lemma 3.2:** The registration policy $\bar{g}$ can be taken to be $\{0, 1\}^S$ valued (rather than $[0, 1]^S$ valued) without loss of optimality.

**Proof:** It is first proved that $U_n$ is concave for any given $n \geq 0$. Suppose $w_1$ and $w_2$ are two probability distributions on $S$, suppose $0 < \eta < 1$ and suppose $w = \eta w_1 + (1 - \eta)w_2$. Then $U_n(w)$ can be viewed as the cost to go given the MS has distribution $w_1$ with probability $\eta$ and distribution $w_2$ with probability $1 - \eta$, and the network does not know which distribution is used. The sum $\eta U_n(w_1) + (1 - \eta)U_n(w_2)$ has a similar interpretation, except the network does know which distribution is used. Thus, the sum is less than or equal to $U_n(w)$, so that $U_n$ is concave. Therefore $U_*$ is also concave.

Given a function $H$ defined on the space of all probability distributions for $S$, let $\hat{H}$ be an extension of $H$ defined on the positive quadrant $R^S_+$ as follows. For any probability distribution $w$ and any constant $c \geq 0$, $H(cw) = cH(w)$. It is easy to show that if $H$ is concave then the extension $\hat{H}$ is also concave. With this notation, the dynamic programming equation for $U_*$ can be written as:

\[ U_*(w) = \beta \lambda_p P s(wP) \]

\[
\min_d \beta \left[ \sum_j \sum_l w_j p_{jl} (\lambda_p U_*(\delta(l))) \\
+ (1 - \lambda_p) d_l (R + U_*(\delta(l)))) \\
+ U_*(wP diag(1 - d)) \right],
\]

where $diag(1 - d)$ is the diagonal matrix with $i$th entry $1 - d_i$. The expression to be minimized over $d$ in this equation is a concave function of $d$, and hence the minimum of the function occurs at one of the extreme points of $[0, 1]^S$, which are just the binary vectors $\{0, 1\}^S$. The minimizing $d$ is $\bar{g}(w)$. This completes the proof of the lemma.

**D. Reduced complexity laws**

Given a pair of feedback controls $(\bar{f}, \bar{g})$, a more compact representation of the controls is possible. Indeed, suppose the controls are used, and suppose in addition that $X(0) = x_0$, where $x_0$ is an initial state known to the network. Given $t \geq 1$, define $k \geq 1$ and $i_0 \in S$ as follows. If there was a report before time $t$, let $t - k$ be the time of the last report before $t$. If there was no report before time $t$ let $k = t$. In either case, let $i_0 = X(t - k)$. Since the network knows $X(t - k)$ at time $t - k$ (after possible paging or registration), we have that $w(t - k) = \delta(i_0)$. Since there were no state updates during the times $t - k + 1, \ldots , t - 1$, it follows that $w(t - 1)$ is the result of applying the update (5) $k - 1$ times, beginning with $\delta(i_0)$. Hence, $w(t - 1)$ is a function of $i_0, k$. Moreover, since $w(t) = \bar{f}(w(t - 1))$ and $v(t) = \bar{g}(w(t - 1))$, it follows that both the paging order vector $u(t)$ and the registration decision vector $v(t)$ are determined by $i_0$ and $k$. Let $f$ and $g$ denote the mappings such that $u(t) = \bar{f}(i_0, k)$ and $v(t) = \bar{g}(i_0, k)$. Note that $f(i_0, k)$ is a paging order vector and $g(i_0, k) \in \{0, 1\}^S$ for each $i_0$. We call the mappings $f, g$ reduced complexity laws (RCLs). We have shown the following proposition.

**Proposition 3.2:** There is no loss in optimality for the original joint paging and registration problem to use policies based on RCLs.

Figure 2 shows an example of a registration policy represented by an RCL for a three-state Markov chain.
IV. ITERATIVE ALGORITHM FOR FINDING INDIVIDUALLY OPTIMAL POLICIES

A. Overview of iterative optimization formulation

While jointly optimal policies can be efficiently represented by RCLs \( f \) and \( g \), the dynamic programming method described for finding the optimal policies is far from computationally feasible, even for small state spaces, because functions of distributions on the state space must be considered. In this section we explore the following method for finding a pair of policies with a certain local optimality property. First it is shown how to find, for a given paging RCL \( f \), an optimal paging RCL \( g \). Then it is shown how to find, for a given registration RCL \( g \), an optimal paging RCL \( f \). Iterating between these two optimization problems produces a pair of RCLs \((f, g)\) such that for each RCL fixed, the other is optimal. Such pairs of RCLs are said to be individually optimal.

In this section we impose the constraint that an MS must register if \( k \geq k_{\text{max}} \), for some large integer constant \( k_{\text{max}} \). With this constraint, the sets of possible registration and paging RCLs are finite, and numerical computation is feasible for fairly large state spaces. The initial state \( x_0 \) is assumed to be known and we write \( C(f, g) \) for the averaged infinite horizon, discounted cost, for paging RCL \( f \) and registration RCL \( g \).

B. Optimal registration RCL for given paging RCL

Suppose a paging RCL \( f \) is fixed. In this subsection we address the problem of finding a registration RCL \( g \) that minimizes \( C(f, g) \) with respect to \( g \). Dynamic programming is again used, but here the viewpoint of the MS is taken. The states used for dynamic programming in this section are the augmented states of the form \((i_0, k, j)\), rather than the set of all probability distributions on \( S \).

Since time is implicitly included in the variable \( k \) in the augmented state, it is computationally more efficient to consider dynamic programming iterations based on cycles rather than on single time steps, where each cycle ends when there is a report. Let \( \tau_m \) be the time of the \( m \)th report. Replacing the infinite horizon by time horizon \( \tau_m \) reduces \( C(f, g) \) to

\[
E \left[ \sum_{t=1}^{\tau_m} \beta^t \{ P I_f \delta t + R I R_t \} \right].
\]

Letting \( m \to \infty \) in (6) yields \( C(f, g) \).

Then for each \((i_0, j, k)\), write \( V_m(i_0, k, j) \) for the cost-to-go for \( m \geq 1 \) update cycles:

\[
V_m(i_0, k, j) = \min_u E \left[ \sum_{t=1}^{\tau_m} \beta^t \{ P I_f \delta t + R I R_t \} \right],
\]

where the expectation \( E \) is taken assuming that (a) the paging RCL \( f \) is used for the paging policy, (b) at \( t = 0 \) the MS is in state \( j \), and (c) the last report occurred \( k \) time units earlier in state \( i_0 \). Also, define \( V_0(i_0, k, j) \equiv 0 \), because the cost is zero when there are no report cycles to go.

The dynamic programming optimality equations are given by

\[
V_m(i_0, k, j) = \beta \sum_{l \in S} p_{jl} [\lambda_p(P f(i_0, k + 1) + V_{m-1}(l, 0, l)) + (1 - \lambda_p) \min \{V_m(i_0, k + 1, l), \mathcal{R} + V_{m-1}(l, 0, l)\}] \tag{8}
\]

As mentioned earlier, registration is forced at relative time \( k = k_{\text{max}} + 1 \) for some large but fixed value \( k_{\text{max}} \). Therefore we set \( V_m(i_0, k_{\text{max}} + 1, l) = \infty \) and use (8) only for \( 0 \leq k \leq k_{\text{max}} \). These equations represent the basic dynamic programming optimality relations. For each possible next state, the MS chooses whichever action has lesser cost: either continuing the current registration cycle or registering for cost \( \mathcal{R} \).

Equation (8) can be used to compute the functions \( V_m \) sequentially in \( m \) as follows. The initial conditions are \( V_0 \equiv 0 \). Once \( V_{m-1} \) is computed, the values \( V_m(i_0, k, j) \) can be computed using (8), sequentially for \( k \) decreasing from \( k_{\text{max}} \) to 0. Formally we denote this computation as \( V_m = T(V_{m-1}) \).

The mapping \( T \) is a contraction with constant \( \beta \) in the sup norm, so that \( V_m \) converges uniformly to a function \( V^* \) satisfying the limiting form of (8):

\[
V^*_s(i_0, k, j) = \beta \sum_{l \in S} p_{jl} [\lambda_p(P f_s(i_0, k + 1) + V^*(l, 0, l)) + (1 - \lambda_p) \min \{V^*(i_0, k + 1, l), \mathcal{R} + V^*(l, 0, l)\}] \tag{9}
\]

for \( 0 \leq k \leq k_{\text{max}} \) and \( V^*(i_0, k_{\text{max}} + 1, l) \equiv \infty \). The corresponding optimal registration RCL \( g^*_s \) is given by

\[
g^*_s(i_0, k) = \begin{cases} 0, & \text{if } V_s(i_0, k + 1, l) \leq \mathcal{R} + V_s(l, 0, l) \\ 1, & \text{else}. \end{cases}
\]

for \( i_0 \in S \) and \( 1 \leq k \leq k_{\text{max}} \).

Thus, for a given paging RCL \( f \), we have identified how to compute a registration RCL \( g \) to minimize \( C(f, g) \).

C. Optimal paging RCL for given registration RCL

Suppose a registration RCL \( g \) is fixed. In this subsection we address the problem of finding a paging RCL \( f \) to minimize \( C(f, g) \). For \( i_0 \in S \) and \( 0 \leq k \leq k_{\text{max}} \), let \( w(i_0, k) \) denote the conditional probability distribution of the state of the MS, given that the most recent report occurred \( k \) time units earlier and the state at the time of the most recent report was \( i_0 \). Thus, \( w(i_0, 0) = \delta(i_0) \), and for larger \( k \) the \( w \)'s can be computed by the recursion:

\[
w(i_0, k + 1) = \Phi(w(i_0, k), g(i_0, k))
\]

The paging order vector \( f(i_0, k) \) is simply the one to be used when the MS must be paged \( k \) time units after the previous report. At such time the conditional distribution of the state of the MS given the observations of the base station is \( w(i_0, k - 1)P \). Thus, the probability the MS is located in cell \( c \), just before the paging begins is given by

\[
p(c|i_0, k) = \sum_{j \in S} \sum_{l \in c} w_j(i_0, k - 1)p_{jl}
\]

Finally, \( f(i_0, k) \) is the paging order vector for ordering the cells \( c \) according to decreasing values of the probabilities \( p(c|i_0, k) \).
D. Iterative optimization algorithm

In the previous subsections we described how to find an optimal g for given f and vice versa. This suggests an iterative method for finding an individually optimal pair \((f, g)\). The method works as follows. Fix an arbitrary registration RCL \(g^0\). Then execute the following steps.

- Find a paging RCL \(f^0\) to minimize \(C(f^0, g^0)\).
- Find a registration RCL \(g_1\) to minimize \(C(f^0, g_1)\).
- Find a paging RCL \(f^1\) to minimize \(C(f^1, g_1)\), and so on.

Then \(C(f^0, g^0) \geq C(f^0, g^1) \geq C(f^1, g_1) \geq C(f^1, g^2) \geq \cdots\)

Since there are only finitely many RCLs, it must be that for some integer \(d\), \(C(f^d, g^d) = C(f^d, g^{d+1})\). By construction, the paging RCL \(f^d\) is optimal given the registration RCL \(g^d\). Similarly, \(g^{d+1}\) is optimal given \(f^d\). However, since \(C(f^d, g^d) = C(f^d, g^{d+1})\), it follows that \(g^d\) is also optimal given the registration RCL \(f^d\). Therefore, \((f^d, g^d)\) is an individually optimal pair of RCLs.

V. EXAMPLES

Two examples are given in this section. Additional examples based on random walk models are in the next section.

A. Rectangular grid example

Consider a rectangular grid topology, such that each cell has four neighbors. The diagram to the left in Figure 3 shows the finite \(i_{\text{max}} \times j_{\text{max}}\) rectangular grid topology. To provide the full complement of four neighbors to cells on the edges of the grid, the region is wrapped into a torus. The torus can serve to move a cell to any of its four neighbors on the edges of the grid.

Each cell in Figure 3 is represented by the index pair \((i, j)\), where \(i = 0, 1, \ldots, i_{\text{max}} - 1\) is the index for the horizontal axis, and \(j = 0, 1, \ldots, j_{\text{max}} - 1\) is the index for the vertical axis.

For simplicity, we assume that there is only one state per cell, so we can take \(C = S\). For a numerical example, consider a \(15 \times 15\) torus grid with motion parameters \(p_{\text{stay}} = 0.4, p_u = p_d = p_l = 0.1, p_r = 0.3, \sigma_0 = (5, 5)\) and other parameters \(\lambda_p = 0.03, \mathcal{P} = 1, \mathcal{R} = 0.6, \beta = 0.9, \) and \(k_{\text{max}} = 200\). We numerically calculated an individually optimal pair \((f, g)\) of RCLs. A sample path of \(X\) and \(w\) generated using these controls is indicated in Figure 4. The figure shows for selected times \(t\) the state \(X(t)\), indicated by a small black square, and the conditional state distribution \(w(t)\), indicated as a moving bubble. The distribution \(w(t)\) collapses to a single unit mass point at \(t = 9\) due to a page and at \(t = 27\) due to a registration. Roughly speaking, the MS registers when it is not where the network expects it to be, given the last report received by the network. For instance, at time \(t = 26\) the MS is located at the tail edge of the bubble, so the network has low accuracy in guessing the MS location. One time unit later, at \(t = 27\), the MS finds itself so far from where the network thinks it should be that the MS registers.

B. Simple Example

The following is an example of a small network for which jointly optimal paging and registration policies can be computed. The example also affords a pair of individually optimal RCLs which are not jointly optimal. The space structure of the example is shown in Figure 5. \(S = \{0, 1, 2, 3, 4\}\) and \(C = \{c_0, c_1, c_2\}\) with \(c_0 = \{0\}, c_1 = \{1, 2\}, c_2 = \{3, 4\}\). From state 0, the MS transits to state 1 with probability 0.4 and to state 3 with probability 0.6. The other possible transitions shown in the figure have probability 1. The initial state is taken to be 0.
Similarly, let \((\delta, P)\) be a pair of mappings \((\delta, P)\) after entering state \(1\) and not being paged. The pair \((\delta, P)\) would cost \(P[A_1 = 1] + P[A_2 = 1\mid A_1 = 1] + P[A_3 = 1\mid A_2 = 1\mid A_1 = 1] + \ldots\). Thus, again the process \((w(t), \delta(t))\) takes values in a set of at most seven states, and the possible transitions are shown in Figure 6. The dynamic programming problem for jointly optimal policies thus reduces to a finite state problem. The optimal choice of the mapping \(\delta\) is given by \(\delta^*(w)\), which pages states in decreasing order of \(wP\). It remains to find the optimal registration policy mapping \(\delta\).

We claim that if \(t \mod 3 = 0\) or \(t \mod 3 = 2\), then it is optimal to not register at time \(t\). Indeed, if \(t \mod 3 = 0\) then the network already knows the MS is in state 0, so registration would cost \(R\) and provide no benefit. If \(t \mod 3 = 2\), then the network knows that the MS will be in state 0 at time \(t + 1\), which is the next time of a potential page. Thus, again the registration at time \(t\) would cost \(R\) and provide no benefit. This proves the claim.

Therefore, it remains to find the optimal registration vector \(v(t)\) to use when \(t \mod 3 = 1\). Such vector is deterministic, given by \(\delta(\delta(0))\). There are essentially only four possible choices for \(\delta(\delta(0))\), as indicated in Table I.

The cost for any pair \((\tilde{\delta}, \tilde{\gamma})\) is given by

\[
C(\tilde{\delta}, \tilde{\gamma}) = \frac{R\beta(1 - \lambda_p)P[A_1\mid P_{1\tilde{\gamma}}]}{1 - \beta^3} + \frac{\lambda_p P[1A\beta + \beta^2 + \beta^2(1 - \lambda_p)P[N_2 = 2\mid P_{1\tilde{\gamma}} \cap P_2] + \beta^3]}{1 - \beta^3}
\]

Consulting Table I we thus find that \((\tilde{\delta}^*, \tilde{\gamma}_A)\) is jointly optimal if \(R \geq \lambda_p P\beta\), and \((\tilde{\delta}^*, \tilde{\gamma}_B)\) is jointly optimal if \(R \leq \lambda_p P\beta\).

For the remainder of this example we consider policies given by RCLs. Under the assumption that \(0 < R \leq \lambda_p P\beta\), the pair of mappings \((\tilde{\delta}^*, \tilde{\gamma}_B)\) is equivalent to a pair of RCLs, which we denote by \((\delta_B, \gamma_B)\). Under \(\gamma_B\), the MS registers only after entering state 1 and not being paged. The pair \((\delta_B, \gamma_B)\) is jointly optimal, and hence it is also individually optimal. Similarly, let \((\delta_C, \gamma_C)\) be RCLs corresponding to the feedback mappings \((\tilde{\delta}^*, \tilde{\gamma}_C)\). In particular, an MS using registration RCL \(\gamma_C\) registers only after entering state 3 and not being paged.

**Proposition 5.1:** The pair of RCLs \((\delta_C, \gamma_C)\) is individually optimal, but not jointly optimal.

**Proof:** The paging RCL \(\delta_C\) is optimal for the registration RCL \(\gamma_C\) because for \(\gamma_C\) fixed, it is equivalent to the optimal feedback mapping \(\tilde{\delta}^*\). Suppose then that the MS uses the paging RCL \(\delta_C\). Note that if the MS does not report at time \(t = 1\), and if it is paged at time \(t = 2\), the network will page cell \(c_1\) first. Hence, if the MS enters state 3 at time \(t = 1\) and if it is not paged at \(t = 1\), then by registering for cost \(R\) it can avoid the two or more pages required at time \(t = 2\) in case of a page at \(t = 2\). Since \(R \leq \lambda_p P\beta\), it is optimal to have the MS register at \(t = 1\) in this situation. Thus \(\gamma_C\) is optimal for \(\delta_C\), so the pair is individually optimal. However, \(C(\delta_C, \gamma_C) > C(\delta_B, \gamma_B)\), so that \((\delta_C, \gamma_C)\) is not jointly optimal.

### VI. JOINTLY OPTIMAL POLICIES FOR SOME RANDOM WALK MODELS

The structure of jointly optimal paging and registration policies are identified in this section for three random walk models of motion. The first is a discrete state one-dimensional random walk, the second is for a symmetric random walk in \(\mathbb{R}^d\) for any \(d \geq 1\), and the third is for a Gaussian random walk in \(\mathbb{R}^d\) for any \(d \geq 1\).

#### A. Symmetric random walk in \(\mathbb{Z}\)

Suppose the motion of the MS is modeled by a discrete-time random walk on an infinite linear array of cells, such that the displacement of the walk at each step has some probability distribution \(b\). Equivalently, \((X(t) : t \geq 0)\) is a discrete time Markov process on \(\mathbb{Z}\) with one-step transition probability matrix \(P\) given by \(p_{ij} = b_{j-i}\). For any probability distribution \(w\), \(wP = w * b\). It is assumed that \(b_i\) is a nonincreasing function of \(|i|\), or in other words, \(b\) is symmetric about zero and unimodal. In the general form of our model, multiple states can correspond to the same cell, but for this example, each integer state \(i\) corresponds to a distinct cell in which the MS can be paged. So \(C = S = \mathbb{Z}\). It is assumed that the network knows the initial state \(x_0\).

Due to the translation invariance of \(P\) for this example, the update equations of the dynamic program are translation invariant, and therefore the paging and registration RCLs can also be taken to be translation invariant. Thus, we write the RCLs as \(f = (f(k) : k \geq 1)\) and \(g = (g(k) : k \geq 1)\). These RCLs give the control decisions if the last reported state is \(i_0 = 0\), and hence for other values of \(i_0\) by translation in space.

It turns out that for this example, the optimal paging policy is ping-pong type: cells are searched in an order of increasing distance from the cell in which the previous report occurred. The optimal registration policy is a distance threshold type: cells are searched in an order of increasing distance from the cell in which the previous report occurred. Specifically, only RCLs of the following form need to be considered. The actions of the policies do not depend on the time \(k\) elapsed since last report, so the argument \(k\) is suppressed. For the paging
policy we take the ping-pong policy, given by the RCL \( f^* = (0, 1, -1, 2, -2, 3, -3, \ldots) \). Thus, if the MS is to be paged and if it was last reported to be at state \( i_0 \), then the states are searched in the order \( i_0, i_0 + 1, i_0 - 1, i_0 + 2, i_0 - 2, \ldots \). The registration policy is given by the RCL \( g_r^* = I_{\{i \geq d_r \text{ or } i \leq -d_r\}} \) where the two distance thresholds \( d_1, d_r \geq 1 \) are such that either \( d_1 = d_r \) or \( d_1 = d_r - 1 \).

**Proposition 6.1:** There is a choice of the distance thresholds \( d_1, d_r \) such that the ping-pong paging policy given by \( f^* \) and the distance-threshold registration policy given by \( g^* \) are jointly optimal.

The related work of Madhow, Honig, and Steiglitz [15] finds the optimal registration policy assuming that the paging policy is fixed to be the ping-pong policy. Also, it is not difficult to show that for the distance threshold registration policy specified by \( g^* \), the optimal paging policy is the ping-pong paging policy. However, a pair of individually optimal RCLs may not be jointly optimal, as shown in the example of Section V-B.

The remainder of this section is devoted to the proof of Proposition 6.1. The following notation is standard in the theory of majorization [16]. Given \( x = (x_1, x_2, \ldots, x_n) \in \mathbb{R}^n \), let \( x_1 \) denote the nonincreasing rearrangement of \( x \). That is, \( x_1 = (x_1, x_2, \ldots, x_n) \), where the coordinates \( x_1, x_2, \ldots, x_n \) are in a nonincreasing order. Given two vectors \( x \) and \( y \), we say that \( y \) majorizes \( x \), denoted by \( x \preceq y \), if the following conditions hold:

\[
\sum_{i=1}^r x_{[i]} \leq \sum_{i=1}^r y_{[i]} \quad \text{for } 1 \leq r \leq n-1
\]

\[
\sum_{i=1}^n x_{[i]} = \sum_{i=1}^n y_{[i]}
\]

Write \( x \approx y \) to denote that both \( x \prec y \) and \( y \prec x \), meaning that \( y \) is a rearrangement of \( x \). The relation \( x \prec y \) can be defined in a similar fashion, in case \( x \) and \( y \) are nonnegative, summable functions defined on some countably infinite discrete set. In such case, \( x_{[i]} \) denotes the \( i^{th} \) coordinate, when the coordinates of \( x \) are listed in a nonincreasing order.

Given a probability distribution \( \mu \) on \( \mathbb{Z} \), let \( s(\mu) \) denote the mean number of states that must be searched to find the MS, given that the MS has distribution \( \mu \) and the optimal search order for \( \mu \) is used. The optimal search order is maximum likelihood search [19], under which states are searched in order of decreasing probability. Summation by parts yields

\[
s(\mu) = \sum_{i=1}^{\infty} i \mu_{[i]} = 1 + \sum_{r=1}^{\infty} (1 - \sum_{i=1}^{r} \mu_{[i]}),
\]

which immediately implies the following lemma.

**Lemma 6.1:** If \( \mu \prec \nu \), then \( s(\mu) \leq s(\nu) \).

A function or probability distribution \( \mu \) on \( \mathbb{Z} \) is said to be neat if \( \mu_0 \geq \mu_1 \geq \mu_{-1} \geq \mu_2 \geq \mu_{-2} \geq \ldots \).

**Lemma 6.2:** If \( \mu \) is a neat probability distribution, then the convolution \( \mu * b \) is neat.

**Proof:** For \( i \geq 0 \), let \( b^{(i)} \) denote the uniform probability distribution over the interval of integers \([-i, i]\). The conclusion is easy to verify in case \( b \) has the form \( b^{(i)} \) for some \( i \). In general, \( b \) is a convex combination of such \( b^{(i)} \)’s, and then \( \mu * b \) is a convex combination of the functions \( \mu * b^{(i)} \), using the same coefficients. Convex combinations of neat distributions are neat, so \( \mu * b \) is indeed neat.

**Lemma 6.3:** If \( \mu \) and \( \nu \) are probability distributions such that \( \mu \prec \nu \) and \( \nu \) is neat, then \( \mu * b \prec \nu * b \).

The proof of Lemma 6.3 is placed in the appendix because the proof is specific to the discrete state setting. Lemma 6.7 in the next subsection is similar, and its proof shows the connection to Riesz’s rearrangement inequality.

Let \( \mu \) be a probability distribution on \( \mathcal{Z} \) and let \( 0 \leq \lambda < 1 \). Let \( T(\mu, \lambda) \) be the set of probability distributions \( \nu \) on \( \mathcal{Z} \) such that \( (1 - \lambda)\nu \leq \mu \), pointwise. Intuitively, such a \( \nu \) is obtained from \( \mu \) by trimming away from \( \mu \) probability mass \( \lambda \) and renormalizing the remaining mass. The following lemma has an easy proof which is left to the reader. Roughly speaking, the lemma means that given \( \mu \) and \( \lambda \), the most maximal distribution in \( T(\mu, \lambda) \), in the majorization order, is obtained by trimming mass from the smallest \( \mu_{[i]} \)’s.

**Lemma 6.4:** (Optimality of minimum likelihood trimming)

There exists \( \nu \in T(\mu, \lambda) \) such that for some \( k \geq 1 \),

\[
(1 - \lambda)\mu_{[j]} = \begin{cases} 
\mu_{[j]} & \text{if } j < k \\
0 & \text{if } j \geq k.
\end{cases}
\]

Furthermore, for any other \( \nu' \in T(\mu, \lambda) \), \( \nu' \prec \nu \).

Let \( f \) and \( g \) be RCLs (possibly dependent on the elapsed time \( k \) since last report). The cost \( C(f, g) \) can be computed by considering the process only up until the first time \( \tau \) that a report occurs (i.e. one reporting cycle). Let \( \alpha(k) = P[\tau = k] = \alpha_p(k) + \alpha_r(k) \), where \( \alpha_p(k) \) is the probability \( \tau = k \) and the first report is a page, and \( \alpha_r(k) \) is the probability \( \tau = k \) and the first report is a registration. Also let \( w(k) \) denote the conditional distribution of the MS given that no report occurs up to time \( k \) for the pair of RCLs \( (f, g) \). Then

\[
C(f, g) = \frac{E[\sum_{j=1}^{\tau} \beta^j (PI_{P}, N_t + R I_{R_t})]}{1 - E[\beta^\tau]} = \frac{\sum_{k=1}^{\infty} \beta^k (\alpha_p(k)s(w(k-1) \ast b) + \alpha_r(k))}{1 - \sum_{k=1}^{\infty} \beta^k \alpha(k)}
\]

Note that the cost depends entirely on the \( \alpha_{[i]} \)’s and on the mean numbers of pages required, given by the terms \( s(w(k-1) \ast b) \).

**Lemma 6.5:** (Optimality of ping-pong paging \( f^* \)) There exists a registration RCL \( g^* \) so that \( C(f^*, g^*) \leq C(f, g) \).

**Proof:** Take the registration RCL \( g^* \) to be of distance threshold type with time varying thresholds and possibly with randomization at the left threshold if the thresholds are equal, or at the right threshold if the right threshold is one larger than the left threshold. More precisely, for fixed \( k \), all the values \( g_r^*(k) \) are binary except for possibly one value of \( l \), and \( 1 - g^*(k) \) is neat. Select the thresholds and randomization parameter so that the \( \alpha_{[i]} \), \( \alpha_p \)’s, and \( \alpha_r \)’s are the same for the pair \( (f^*, g^*) \) as for the originally given pair \( (f, g) \).

Let \( w(p(k) : k \geq 0) \) and \( \tau^* \) be defined for \( (f^*, g^*) \) just as \( w(k) : k \geq 0 \) and \( \tau \) are defined for \( (f, g) \). To complete the proof of the lemma it remains to show that \( s(w(p(k-1) \ast b) \leq s(w(k-1) \ast b) \) for \( k \geq 1 \). The sequences \( w \) and \( w^o \) are updated...
in similar ways, by Lemma 6.1,
\[
\begin{align*}
w(k) &= \Phi(w(k-1), g(k)) & 1 \leq k \leq \tau - 1 \\
w^o(k) &= \Phi(w^o(k-1), g^o(k)) & 1 \leq k \leq \tau - 1
\end{align*}
\]

By the definition of \(\Phi\), this means the distribution \(w(k)\) is obtained by first forming the convolution \(w(k-1) \ast b\), removing a fraction \(g(k)\) of the mass at each location \(l\), and renormalizing to obtain a probability distribution. The RLC \(g^o\) trims mass in a minimum likelihood fashion. Thus, Lemmas 6.2, 6.3 and 6.4 show by induction that for all \(k \geq 1\): \(w^o(k)\) is neat, \(w(k) < w^o(k)\), and \(w(k-1) \ast b < w^o(k-1) \ast b\). Thus by Lemma 6.1 \(s(w^o(k-1) \ast b) \leq s(w(k-1) \ast b)\), completing the proof of Lemma 6.5.

**Proof of Proposition 6.1.** In view of Lemma 6.5, it remains to show that if the ping-pong paging policy specified by \(f^*\) is used, then for some choice of fixed distance thresholds \(d_1\) and \(d_2\), the registration policy specified by \(g^*\) is optimal. This can be done by examining a dynamic program for the optimal registration policy, under the assumption that the RCL \(f^*\) is used. Let \(V_n(j)\) denote the mean discounted cost for \(n\) time steps to go, given that the mobile is located directed distance \(j\) from its last reported state. Then
\[
V_{n+1}(j) = \beta \sum_{l \in \mathbb{Z}} b_{j-l} \left[ \lambda_p(P f^*_l + V_n(l)) + (1 - \lambda_p) \min\{V_n(l), R + V_n(0)\} \right]
\]

By a contraposition property of these dynamic programming equations, the limit \(V_n = \lim_{n \to \infty} V_n\) exists. Argument by induction yields that the functions \(-V_n\) are neat, and hence that \(-V_n\) is neat. By the dynamic programming principle, an optimal registration policy is given by the RCL \(g^*\) specified by:
\[
g^*_l = \begin{cases} 1 & \text{if } V_n(l) \geq R + V_n(0) \\ 0 & \text{else} \end{cases}
\]

Since \(-V_n\) is neat, the optimal registration RCL \(g^*\) has the required threshold type. Proposition 6.1 is proved.

**B. Symmetric random walk in \(\mathbb{R}^d\)**

To extend Proposition 6.1 to more than one dimension, we consider a continuous state mobility model, with \(S = \mathbb{C} = \mathbb{R}^d\), for an integer \(d \geq 1\). Of course in practice we expect \(d \leq 3\). A function on \(\mathbb{R}^d\) is said to be symmetric nonincreasing if it can be expressed as \(\phi(|x|)\), for some nonincreasing function \(\phi\) on \(\mathbb{R}_+\), where \(|x|\) denotes the usual Euclidean norm of \(x\). Let \(x_0 \in \mathbb{R}^d\), and let \(b\) be a symmetric nonincreasing probability density function (pdf) on \(\mathbb{R}^d\). The location of the MS at time \(t\) is assumed to be given by \(X(t) = x_0 + \sum_{s=1}^{t} B_s\), where the initial state \(x_0\) is known to the network, and the random variables \(B_1, B_2, \ldots\) are independent, with each having pdf \(b\).

Let \(\mathcal{L}^d(A)\) denote the volume (i.e. Lebesgue measure) of a Borel set \(A \subset \mathbb{R}\). A paging order function \(V = \{r_x : x \in \mathbb{R}^d\}\) is a nonnegative function on \(\mathbb{R}^d\) such that \(\mathcal{L}^d(\{x : r_x \leq \gamma\}) = \gamma\) for all \(\gamma \geq 0\). Thus, as \(\gamma\) increases, the volume of the set \(\{x : r_x \leq \gamma\}\) increases at unit rate. Imagine the set \(\{x : r_x \leq \gamma\}\) increasing as \(\gamma\) increases, until the MS is in the set. If the MS is located at \(\bar{x}\) and is paged according to the paging order function \(r\), then \(r_{\bar{x}}\) denotes the volume of the set searched to find \(\bar{x}\). So the paging cost is \(P r_{\bar{x}}\), where \(P\) is the cost of paging per unit volume searched. An example of a paging order is increasing distance search, starting at \(x_o\), which corresponds to letting \(r_{\bar{x}}\) be the volume of a ball of radius \(|x - x_o|\) in \(\mathbb{R}^d\). As in the finite state model, assume the cost of a registration is \(R\).

Paging and registration policies \(u\) and \(v\) can be defined for this model just as they were for the finite state model, with paging order functions playing the role of paging order vectors. Thus, for each \(t \geq 0\), \(u(t) = (u_x(t) : x \in \mathbb{R}^d)\) is a paging order function, and \(v(t) = (v_x(t) : x \in \mathbb{R}^d)\) is a \([0,1]\)-valued function. In addition, translation invariant RCLs \(f\) and \(g\) can be defined as they were for the one-dimensional network model, and they determine policies \(u\) and \(v\) as follows. If the location of the most recent report was \(x_o\), then \(u_x(t) = f_{x-x_o}\) and \(v_x(t) = g_{x-x_o}.\) Let \(f^*\) be the RCL for increasing distance search paging; \(f_{x-x_o}\) is the volume of the radius \(|x - x_o|\) ball in \(\mathbb{R}^d\). Let \(g^*\) be the RCL for the distance threshold registration policy with some threshold \(\eta\); \(g_{x-x_o} = I_{|x| \geq \eta}\).

**Proposition 6.2:** There is a choice of the distance threshold \(\eta\) such that \(f^*\) and \(g^*\) are jointly optimal.

The proof of Proposition 6.1 can be used for the proof of Proposition 6.2 with symmetric nonincreasing functions on \(\mathbb{R}^d\) replacing neat probability distributions on \(\mathbb{Z}\). A suitable variation of Lemma 6.3 must be established, and we will show that this can be done by applying Riesz’s rearrangement inequality. To get started, we introduce some notation from the theory of rearrangements of functions (similar to the notation in [13].) If \(A\) is a Borel subset of \(\mathbb{R}^d\) with \(\mathcal{L}^d(A) < \infty\), then the symmetric rearrangement of \(A\), denoted by \(A^\sigma\), is the open ball in \(\mathbb{R}^d\) centered at \(0\) such that \(\mathcal{L}^d(A) = \mathcal{L}^d(A^\sigma)\). Given an integrable, nonnegative function \(h\) on \(\mathbb{R}^d\), its symmetric nonincreasing rearrangement, \(h^\sigma\), is defined by
\[
h^\sigma(x) = \int_0^\infty I_{\{h^\sigma > t\}} dt
\]

Let \(h_1 \ast h_2\) denote the convolution of functions \(h_1\) and \(h_2\), and let \((h_1, h_2) = \int_{\mathbb{R}^d} h_1 h_2 dx\). A proof of the following celebrated inequality is given in [13].

**Lemma 6.6:** F. Riesz’s rearrangement inequality[18] If \(h_1\), \(h_2\), and \(h_3\) are nonnegative functions on \(\mathbb{R}^d\), then \((h_1, h_2) \prec (h_3, h_3)\). Given two probability densities on \(\mathbb{R}^d\), \(\nu\) majorizes \(\mu\), written \(\mu \prec \nu\), if
\[
\int_{|x| \leq \rho} \mu^\sigma dx \leq \int_{|x| \leq \rho} \nu^\sigma dx \text{ for all } \rho > 0.
\]

Equivalently, \(\mu \prec \nu\) if, for any Borel set \(F \subset \mathbb{R}^d\), there is another Borel set \(F' \subset \mathbb{R}^d\) with \(\mathcal{L}^d(F) = \mathcal{L}^d(F')\), such that
\[
\int_F \mu dx \leq \int_F \nu dx.
\]

If \(\mu \prec \nu\), then \((\mu^\sigma, h) \leq (\nu^\sigma, h)\) for any symmetric nonincreasing function \(h\). (To see this, use the fact that such
Lemma 6.7: If $\mu$ and $\nu$ are probability densities such that $\mu \prec \nu$, and if $\nu$ is symmetric nonincreasing, then $\mu * b \prec \nu * b$.

Proof: Let $F$ be an arbitrary Borel subset of $\mathbb{R}^d$. Let $h_1 = \mu$, $h_2 = 1_F$, and $h_3 = b$. Then $h_1^t = \mu^t$, $h_2^t = 1_{F^t}$, $h_3^t = b$, and Riesz’s rearrangement inequality yields $(\mu * b)^t \leq (\nu^t * 1_{F^t} * b)$. Since $\mu \prec \nu = \nu^t$ and $1_{F^t} * b$ is symmetric nonincreasing, $(\mu^t, 1_{F^t} * b) \leq (\nu, 1_{F^t} * b)$. Combining yields $(\mu, 1_{F^t} * b) \leq (\nu, 1_{F^t} * b)$, or, equivalently by the symmetry of $b$, $(\mu * b, 1_{F^t}) \leq (\nu * b, 1_{F^t})$. That is,

$$\int_F \mu * b \, dx \leq \int_{F^t} \nu * b \, dx.$$ 

Since $F$ was an arbitrary Borel subset of $\mathbb{R}^d$ and $\mathcal{L}^d(F) = \mathcal{L}^d(F^t)$, $\mu * b \prec \nu * b$.

Proof of Proposition 6.2: Proposition 6.2 follows from Lemma 6.7 and the same arguments used to prove Proposition 6.2. The details are left to the reader.

C. Gaussian random walk in $\mathbb{R}^d$

Consider the following variation of the model of Section VII-B: Let $X(t) = x_0 + \sum_{s=1}^{t} B_s$, where the random variables $B_s$ are independent with a $d$-dimensional Gaussian density with mean vector $m$ and covariance matrix $\Sigma$. Given a vector $y$ let $|y|_\Sigma = \langle y, \Sigma^{-1} y \rangle^{1/2}$. Proposition 6.2 can be applied to the process with initial state $\Sigma^{-1/2} x_0$ and increments $B_i = \Sigma^{-1/2} (B_i - m)$. Suppose the time of the last report was $t_o$ and the location at that time was $x_o$, and suppose the MS just jumped to a new state at time $t$. Let $\bar{x}(t) = x_o + (t - t_o) m$. If the MS must be paged at time $t$, the optimal paging policy is to page according to expanding ellipses of the form $\{ x : |x - \bar{x}(t)|_\Sigma \leq \rho \}$. If the MS is not paged at time $t$, the optimal registration policy is for the MS to register if $|X(t) - \bar{x}(t)|_\Sigma \geq \eta$, for a suitable threshold $\eta$.

A continuous time version of this result can also be established, for which the motion of the MS is modeled as a $d$-dimensional Brownian motion with drift vector $m$ and infinitesimal covariance matrix $\Sigma$.

VII. Conclusions

There are many avenues for future research in the area of paging and registration. This paper shows how the joint paging and registration optimization problem can be formulated as a dynamic programming problem with partially observed states. In addition, an iterative method is proposed, involving dynamic programming with a finite state space, in order to find individually optimal pairs of RCLs. While an example shows that, in principle, the individually optimal pairs need not be jointly optimal, no bounds are given on how far from optimal the individually optimal pairs can be. Furthermore, even the problem of finding individually optimal RCLs may be computationally prohibitive, so it may be fruitful to apply approximation methods such as neurodynamic programming [7]. This becomes especially true if the model is extended to handle additional features of real world paging and registration problems, such as the use of parallel paging, overlapping registration regions, congestion and queuing of paging requests for different MSs, positive probabilities of missed pages, more complex motion models, estimation of motion models, and so on.

This paper shows that jointly optimal paging and registration policies for symmetric or Gaussian random walk models are given by nearest-location-first paging policies and distance threshold registration policies. It remains to be seen whether these policies are good ones, even if no longer optimal, when the assumptions of the model are violated. It also remains to be seen if jointly optimal policies can be identified for other subclasses of motion models.

We found that majorization theory, and, in particular, Riesz’s rearrangement inequality, are tools well suited for the study of a certain search algorithms with feedback. These tools may be more widely useful for addressing search or distributed sensing problems.

APPENDIX

APPENDIX A: ON $\sigma$-ALGEBRA NOTATION

Some basic definitions involving $\sigma$-algebras are collected in this appendix. In this paper the network only observes random variables with finite numbers of possible outcomes, so that emphasis is given to conditioning with respect to finite $\sigma$-algebras.

The collections of random variables considered in this paper are defined on some underlying probability space. A probability space is a triple $(\Omega, \mathcal{F}, P)$, such that $\Omega$ is the set of all possible outcomes, $\mathcal{F}$ is a $\sigma$-algebra of subsets of $\Omega$ (so $\emptyset \in \mathcal{F}$ and $\mathcal{F}$ is closed under complements and countable intersections) and $P$ is a probability measure, mapping each element of $\mathcal{F}$ to the interval $[0,1]$. The sets in $\mathcal{F}$ are called events. A random variable $X$ is a function on $\Omega$ which is $\mathcal{F}$ measurable, meaning that $\mathcal{F}$ contains all sets of the form $\{ \omega : X(\omega) \leq c \}$. In the remainder of this section, $\mathcal{N}$ denotes a $\sigma$-algebra that is a subset of $\mathcal{F}$. Intuitively, $\mathcal{N}$ models the information available from some measurement: one can think of $\mathcal{N}$ as the set of events that can be determined to be true or false by the measurement. A random variable $Y$ is said to be $\mathcal{N}$ measurable if $\mathcal{N}$ contains all sets of the form $\{ \omega : Y(\omega) \leq c \}$. Intuitively, $Y$ is $\mathcal{N}$ measurable if the information represented by $\mathcal{N}$ determines $Y$.

An atom $B$ of $\mathcal{N}$ is a set $B \in \mathcal{N}$ such that if $A \subset B$ and $A \in \mathcal{N}$ then either $A = \emptyset$ or $A = B$. Note that if $C \in \mathcal{N}$ and $B$ is an atom of $\mathcal{N}$, then either $B \subset C$ or $B \subset C^c$. If $\mathcal{N}$ is finite (has finite cardinality) then there is a finite set of atoms $B_1, \ldots, B_m$ in $\mathcal{N}$ such that each element of $\mathcal{N}$ is either $\emptyset$ or the union of one or more of the atoms.

Given a random variable $X$ with finite mean, one can define $E[X|\mathcal{N}]$ in a natural way. It is an $\mathcal{N}$ measurable random variable such that $E[X|\mathcal{N}] = E[E[X|\mathcal{N}]|\mathcal{N}]$ for any bounded, $\mathcal{N}$ measurable random variable $Z$. In particular, if $A$ is an atom in $\mathcal{N}$, then $E[X|\mathcal{N}]$ is equal to $E[X|A]/|P[A]|$ on the set $A$. (Any two versions of $E[X|\mathcal{N}]$ are equal with probability one.)

Given a random variable $Y$, we write $\sigma(Y)$ as the smallest $\sigma$-algebra containing all sets of the form $\{ \omega \in \Omega : Y(\omega) \leq c \}$.
The notation $E[X|Y]$ is equivalent to $E[X|\sigma(Y)]$. In case $Y$ is a random variable with a finite number of possible outcomes $\{y_1, \ldots, y_m\}$, the $\sigma$-algebra $\sigma(Y)$ is finite with atoms $B_i = \{\omega : Y(\omega) = y_i\}$, $1 \leq i \leq m$. Furthermore, given a random variable $X$ with finite mean, $E[X|Y]$ is the function on $\Omega$ which is equal to $E[X|B_i]$ on $B_i$ for $1 \leq i \leq m$.

APPENDIX B: PROOF OF LEMMA 3.1

Since all the random variables generating $N_{t+1}$ have only finitely many possible values, the $\sigma$-algebra $N_{t+1}$ is finite. Both sides of (2) are $N_{t+1}$ measurable, so both sides are constant on each atom of $N_{t+1}$. Thus, if $A$ denotes an atom of $N_{t+1}$, each side of (2) can be viewed as a function of $A$, and it must be shown that the equality holds for all such $A$. Below we shall write $w_l(t+1, A)$ for the value of $w_l(t+1)$ on the atom $A$.

Since $P_{t+1} \cup R_{t+1} \in N_{t+1}$, it follows that either $A \subset P_{t+1} \cup R_{t+1}$ or $A \subset P^c_{t+1} \cap R^c_{t+1}$. If $A \subset P_{t+1} \cup R_{t+1}$ then $X(t+1)$ is determined by $A$, and $w_l(t+1, A) = \delta(X(t+1))$, so that (2) holds on $A$. So for the remainder of the proof, assume that $A \subset P^c_{t+1} \cap R^c_{t+1}$.

It follows that $A$ can be expressed as $A = \hat{A} \cap P^c_{t+1} \cap R^c_{t+1}$ for some atom $\hat{A}$ of $N_t$. Thus for any state $l$,

$$w_l(t+1, A) = P[X(t+1) = l|A] = \frac{T_l}{\sum_{l' \in S} T_{l'}}$$

where, letting $w_l(t, \hat{A})$ denote the value of $w_l(t)$ on $\hat{A}$ and $v_l(t+1, \hat{A})$ denote the value of $\tilde{v}_l(t+1)$ on $\hat{A}$,

$$T_l = P[R^c_{t+1} \cap \{X(t+1) = l\} | A \cap P^c_{t+1}] = E[(1 - v_l(t+1))I_{\{X(t+1) = l\}} | A \cap P^c_{t+1}] = E[(1 - v_l(t+1))I_{\{X(t+1) = l\}} | \hat{A}] = P[X(t+1) = l|\hat{A}] (1 - \tilde{v}_l(t+1, \hat{A})) = \left(\sum_j w_j(t, \hat{A})q_{jl}\right) (1 - \tilde{v}_l(t+1, \hat{A})).$$

Therefore

$$w_l(t+1, A) = F_l(t+1, A),$$

for any atom $A$ of $N_{t+1}$ with $A \subset P^c_{t+1} \cap R^c_{t+1}$. Lemma 3.1 is proved.

APPENDIX C: PROOF OF LEMMA 6.3

Lemma 6.3 is proved following the statement and proof of three lemmas.

**Lemma 1.1:** Consider two monotone sequences of some finite length $n$: $a_1 \geq a_2 \geq \ldots \geq a_n = 0$ and $0 = b_1 \leq b_2 \leq \ldots \leq b_n$. Let $c_i = a_i + b_i$ for $1 \leq i \leq n$, let $d_i = a_i + b_{i+1}$ for $1 \leq i < n-1$, and let $d_n = 0$. Then $c < d$.

**Proof:** Note that $d_i \geq c_i$ and $d_i \geq c_{i+1}$ for $1 \leq i \leq n-1$, and the sum of the $c$’s is equal to the sum of the $d$’s. Therefore, for any subset $A$ of $\{1, 2, \ldots, n\}$, there is another subset $A'$ with $|A| = |A'|$ such that $\sum_{i \in A} c_i \leq \sum_{i \in A'} d_i$. That proves the lemma.

**Lemma 1.2:** Let $r$ and $L$ be positive integers. Consider the convolution $FW$ of two binary valued functions on $Z$, such that the support of $F$ has cardinality $r$, and the support of $G$ is a set of $L$ consecutive integers. Then the convolution is maximal in the majorization order, if the support of $F$ is a set of $r$ consecutive integers.

**Proof:** Suppose without loss of generality that $G = I_{\{0 \leq j \leq L-1\}}$. If the support of $F$ is not an interval of integers, let $j_{\max}$ be the largest integer in the support of $F$ and let $j_0$ be the smallest integer such that the support of $F$ contains the interval of integers $[j_0, j_{\max}]$. Then $F = F^{a} + F^{b}$, such that $F^{a} = 0$ for $i \geq j_0 - 1$ and the support of $F^{b}$ is the interval of integers $[j_0, j_{\max}]$. Let $F'$ be the new function defined by $F'_{j} = F^{a}_{j} + F^{b}_{j}$. The graph of $F'$ is obtained by sliding the rightmost portion of the graph of $F$ to the left unit one.

We claim that $F \cdot G < F' \cdot G$. To see this, note that $F \cdot G = F^{a} \cdot G + F^{b} \cdot G$. The idea of the proof is to focus on the interval of integers $I = [j_0 - 1, j_0 + r - 2]$ and appeal to Lemma 1.1. The function $F' \cdot G$ is nonincreasing on $I$, it takes value zero at the right endpoint of $I$, and it is also zero everywhere to the right of $I$. The function $F^{b} \cdot G$ is nondecreasing on $I$, it takes value zero at the left endpoint of $I$, and it is also zero everywhere to the left of $I$. The convolution $F' \cdot G$ is the same as $F \cdot G$ except the second function $F^{b} \cdot G$ is shifted one unit to the right. Lemma 1.1 thus implies that $F \cdot G < F' \cdot G$. This procedure can be repeated until $F$ is reduced to a function with support being a set of $r$ consecutive integers. The lemma is proved.

**Lemma 1.3:** Let $r \geq 1$ and consider the convolution $F \cdot b$ such that $F$ is a binary valued function on the integers with support of cardinality $r$. Then the convolution is maximal in the majorization order if the support of $F$ consists of $r$ consecutive integers.

**Proof:** For $i \geq 0$, let $b^{(i)}$ denote the uniform probability distribution on the interval $[-i, i]$, of $L = 2i + 1$ integers. The lemma is true if $b = b^{(i)}$ for some $i$ by Lemma (1.2). Let $F^*$ denote the unique neat binary valued function with support of cardinality $r$. Note that $F^* \cdot b^{(i)}$ is neat for all $i \geq 0$ because both $b^{(i)}$ and $F^*$ are neat. In general, $b$ can be written as $b = \sum_{i=0}^{\infty} \lambda_{i} b^{(i)}$ for some probability distribution $\lambda$ on $Z$. Therefore, for any binary $F$ with support of cardinality $r$,

$$b \cdot F = \sum_{i} \lambda_{i} (b^{(i)} \cdot F) \leq \sum_{i} \lambda_{i} (b^{(i)} \cdot F^{*}) = (b \cdot F^{*})^{\perp} \equiv b \cdot F^{*}.$$
to denote inner products.
\[
\sum_{i=1}^{r} (\mu \ast b)_{[i]} = \max_{F} (\mu \ast b, F) = \max_{F} (\mu, b \ast F)
\]
\[
\leq \max_{F} (\mu_{i}, (b \ast F)_{i})
\]
\[
\leq (\nu_{i}, (b \ast F^{*})_{i}) \quad (c)
\]
\[
= (\nu \ast b, F^{*}) \quad (d) = \sum_{i=1}^{r} (\nu \ast b)_{[i]}
\]
Here, (a) follows from the fact that rearranging each of two distributions in nonincreasing order increases their inner product, (b) follows from Lemma 1.3 and the monotonicity of \( \mu_{i} \), (c) follows from the fact that both \( \nu \) and \( b \ast F^{*} \) are neat, so their inner product is the same as the inner product of their rearranged probability distributions, and (d) follows from the fact that \( \nu \ast b \) is neat.

**ACKNOWLEDGMENT**

The authors are grateful for useful discussions with Rong-Rong Chen and Richard Sowers.

**REFERENCES**

[1] I. F. Akyildiz, J. S. M. Ho, and Y. B. Lin, “Movement-based location update and selective paging for PCS networks,” *IEEE/ACM Transactions on Networking*, vol. 4, pp. 629–638, August 1996.

[2] F. Anjum, L. Tassiulas, and M. Shayan, “Optimal paging for mobile location tracking,” *Advances in Performance Analysis*, vol. 3, pp. 153–178, March 2002.

[3] A. Bar-Noy and I. Kessler, “Tracking mobile users in wireless communication networks,” *Proc. IEEE Infocom*, pp. 1232–1239, 1993.

[4] A. Bar-Noy, I. Kessler, and M. Sidi, “Mobile users: To update or not to update?” *IEEE/ACM Transactions on Networking*, vol. 4, pp. 629–638, August 1996.

[5] D. P. Bertsekas, *Dynamic Programming and Optimal Control*, Vol. I, Athena Scientific, Belmont, MA, 1995.

[6] D. P. Bertsekas, *Dynamic Programming and Optimal Control*, Vol. II, Athena Scientific, Belmont, MA, 1995.

[7] D. P. Bertsekas and J. N. Tsitsiklis, *Neuro-Dynamic Programming*, Athena Scientific, Belmont, MA, 1996.

[8] C. Cachin, *Entropy Measures and Unconditional Security in Cryptography*, PhD thesis, Swiss Federal Institute of Technology Zürich, 1997.

[9] A. De Santis, A. G. Gaggia, and U. Vaccaro, “Bounds on entropy in a guessing game,” *IEEE Trans. Information Theory*, vol. 47, pp. 468–473, 2001.

[10] B. Hajek, “Jointly optimal paging and registration for a symmetric random walk,” *IEEE Information Theory Workshop*, Bangalore, India, pp. 20–23, October 2002.

[11] B. Hajek, K. Mützel, and S. Yang, “Paging and registration in cellular networks: jointly optimal policies and an iterative algorithm,” *IEEE INFOCOM 2003*, March 30–April 3, pp. 524–532, 2003.

[12] B. Krishnamachari, R.-H. Gau, S. B. Wicker, and Z. J. Haas, “Optimal sequential paging in cellular networks,” *ACM Wireless Networks*, vol. 10, no. 2, pp. 121–131, March 2004.

[13] E. H. Lieb and M. Loss, *Analysis*, second edition, American Mathematical Society, Providence, 2001.

[14] S. Madhavapeddy, K. Basu, and A. Roberts, “Adaptive paging algorithms for cellular systems,” *Proc. 45th IEEE Vehicular Technology Conference*, vol. 2, pp. 976–980, 1995.

[15] U. Madhow, M. L. Honig, and K. Steiglitz, “Optimization of wireless resources for personal communications mobility tracking,” *IEEE/ACM Transactions on Networking*, vol. 3, pp. 698–707, December 1995.

[16] A. W. Marshall and I. Olkin, *Inequalities: Theory of Majorization and Its Applications*, Academic Press, New York, 1979.

[17] J. L. Massey, “Guessing and entropy,” *Proc. 2004 IEEE International Symposium on Information Theory*, p. 204, Trondheim, Norway, 1994.

[18] F. Riesz, “Sur une inégalité intégrale,” *J. London Mathematical Society*, vol. 5, pp. 162–168, 1930.

[19] C. Rose and R. Yates, “Minimizing the average cost of paging under delay constraints,” *Wireless Networks*, vol. 1, pp. 211–219, 1995.

[20] C. Rose, “Minimizing the average cost of paging and registration: A timer-based method,” *Wireless Networks*, vol. 2, pp. 109–116, 1996.

[21] C. Rose, “State-based paging/registration: a greedy technique,” *IEEE Transactions on Vehicular Technology*, vol. 48, pp. 166–173, January 1999.

[22] W. Wang, I. F. Akyildiz, and G. L. Stuber, “Effective paging schemes with delay bounds as QoS constraints in wireless systems,” *Wireless Networks*, vol. 7, pp. 455–466, 2001.