Revenue Optimization via Call Admission Control and Pricing for Mobile Cellular Systems

Jian Ni and Sekhar Tatikonda
Department of Electrical Engineering, Yale University, New Haven, CT 06520, USA
Email: {jian.ni, sekhar.tatikonda}@yale.edu

Abstract—Service differentiation is now becoming a more and more desirable feature in mobile cellular systems making the case for pricing differentiation among the users. One of the main objectives of a mobile service provider is to maximize its revenue. In general there are two approaches to optimize the revenue rate of a base station. One is through call admission control (CAC), the other is through pricing. For most real systems it is prohibitively difficult to compute the optimal CAC policy or pricing scheme because of the ‘curse of dimensionality’. Therefore we focus on CAC policies that have simple structures and static pricing schemes in which the charging rates are independent of the system state. However, finding the optimal structured CAC policies or static pricing scheme is still combinatorial in nature. In this paper we show that our recently proposed iterative coordinate search algorithm provides an efficient and effective method to find the optimal or near-optimal structured CAC policies and static pricing schemes for mobile cellular systems.

Keywords—Mobile Cellular Systems, Revenue Optimization, Call Admission Control, Pricing.

I. INTRODUCTION

In mobile cellular systems, the service area is divided into cells. Each user (mobile subscriber/terminal) communicates via the base station (BS) in the cell that it is currently residing in. The BS has a limited number of channels (frequency bands, time slots, or codes). When a user initiates a call but the BS has no free channels, the call is blocked and the potential revenue is lost. One of the main objectives of a mobile service provider is to maximize its revenue.

Service differentiation is now becoming a more and more desirable feature in mobile cellular systems making the case for pricing differentiation among the users. For example, some users may want to pay more to get better quality of service. This requires the system to have multiple pricing levels. Once the charging rates for different user (call) classes are fixed, one way to optimize the revenue rate of the BS is through call admission control (CAC), i.e., more profitable users are given higher priority for channel access.

Traditionally, channel access priority is given to handoff calls so that they experience a lower call blocking probability than new calls. The reason is that from a user’s viewpoint, having a call abruptly terminated in the middle of a conversation is more annoying than being blocked occasionally on a new call attempt. One common method for giving priority to handoff calls is the so-called cutoff priority scheme [3], [5] (also known as guard channel [11]). Under a cutoff priority scheme, new calls and handoff calls are treated equally for channel access up to a predetermined channel utilization level, then only handoff calls are accepted (provided there are free channels) while new calls are simply blocked. It seems that a fraction of the total available channels of the BS is reserved for handoff calls. Later the scheme was generalized for wireless networks with multiple call classes [5]. The cutoff priority schemes are reservation based CAC policies [8].

For single-service mobile cellular systems, with the assumptions of independent Poisson call arrivals and exponential channel occupancy times, it is well known that the optimal CAC policy which maximizes the revenue rate of the BS is a reservation (based CAC) policy [6], [11]. For multiservice mobile cellular systems the optimal CAC policy in general does not have a simple structure and is prohibitively difficult to compute. The threshold (based CAC) policies were proposed for multiservice wireless networks [4], [7]. Under a threshold policy the stationary distribution of the system state has a product form and can be evaluated through efficient convolution algorithms [12], [13].

The reservation and threshold policies are called structured CAC policies [8]. Since the number of structured CAC policies grows exponentially with the number of call classes and the capacity of the BS, finding the optimal structured CAC policies is a complex combinatorial optimization problem. We have proposed an iterative coordinate search algorithm to find the optimal or near-optimal structured CAC policies. Through a number of numerical experiments we show that the search algorithm converges quickly and the returned structured CAC policies significantly increase the revenue rate of the BS when the traffic demands are heavy.

Another approach to optimize the revenue rate of the BS is through pricing [9]. In the CAC approach, we assume that the charging rates are predetermined and fixed by the mobile service provider, and the BS conducts CAC to maximize its revenue rate. In the pricing approach, however, we study the problem of how to choose the charging rates to maximize the revenue rate of the BS. Like CAC, the optimal (dynamic) pricing scheme is generally complicated and is prohibitively difficult to compute. The near-optimality of static pricing was established in a number of limiting regimes [9]. In addition, static pricing schemes also have some obvious advantages over dynamic pricing schemes: a static pricing scheme is easy to implement and evaluate; under a static pricing scheme charging rates are predictable to the users hence no complex real-time mechanism is required to communicate the charging rates to the users; and computing the optimal or near-optimal
II. REVENUE OPTIMIZATION VIA CALL ADMISSION CONTROL

Assume that the BS has \( C \) channels and there are \( K \) different classes of calls. Class-\( k \) calls arrive at the cell according to a Poisson process with rate \( \lambda_k \) and have mean channel occupancy time \( 1/\mu_k \). A class-\( k \) call, once accepted, occupies \( b_k \) channels and is charged at rate \( p_k \) per unit time when in progress. In this section we assume that the charging rates \( p_k \) are predetermined and fixed by the mobile service provider.

Let \( \gamma_k = p_k/b_k \) be the revenue coefficient of class-\( k \) calls, which can be interpreted as the revenue generated by one channel per unit time when the channel is used by a class-\( k \) call. Clearly classes with higher revenue coefficients are more profitable. Let \( B_k \) be the call blocking probability of class-\( k \) calls. The revenue rate \( g \) of the BS is defined (in the long term or at the steady state) as the average revenue collected from all ongoing calls supported by the BS per unit time. By Little’s Theorem \( g = \sum_{k=1}^{K} \lambda_k \mu_k (1 - B_k)/\mu_k \). The goal of CAC is to maximize the revenue rate \( g \).

A. Single-Service Mobile Cellular Systems

For single-service mobile cellular systems, calls of all classes have the same resource requirement and channel occupancy time distribution, i.e., \( b_k = 1 \) (without loss of generality) and \( \mu_k = \mu \) for all \( k \). We order the call classes by their revenue coefficients so that \( \gamma_1 \geq \gamma_2 \geq \cdots \geq \gamma_K \). Under a reservation policy, each call class \( k \) is associated with a reservation parameter \( r_k \). A class-\( k \) call is accepted upon arrival if and only if the BS has \( r_k \) or more free channels reserved (for other classes) after acceptance.

1) Performance Evaluation: For single-service mobile cellular systems, once a call is accepted, its class becomes irrelevant to the future evolution of the system state. It suffices to choose \( n \) as the system state when there are \( n \) calls being in progress in the cell. Let \( \rho_k = \lambda_k/\mu \) and \( \bar{\rho}_k = \sum_{j=1}^{K} \rho_j \) for \( 1 \leq k \leq K \). Let \( r_{K+1} = C \).

The system state under a reservation policy with parameters \( 0 \leq r_1 \leq r_2 \leq \cdots \leq r_K \leq C \) can be modelled as a one-dimensional birth-death process with the following steady-state probabilities:

\[
\pi_C(0) = G(C)^{-1} = \left( 1 + \sum_{k=1}^{K} \frac{\bar{\rho}_k^{n-C+r_k+1}}{n!} \prod_{i=0}^{K} \bar{\rho}_i^{r_i+1-r_i} \right)^{-1},
\]

for \( k = K, K-1, \ldots, 1, C - r_k + 1 \leq n \leq C - r_k \),

\[
\pi_C(n) = \frac{\bar{\rho}_k^{n-C+r_k+1}}{n!} \prod_{i=0}^{K} \bar{\rho}_i^{r_i+1-r_i}
\]

Note that the birth-death process is time-reversible so the insensitivity property [12] guarantees that (1)-(2) hold for arbitrary channel occupancy time distributions with finite means.

Under the reservation policy, the call blocking probability of class-\( k \) calls is \( B_k = \sum_{n=C-r_k}^{\infty} \pi_C(n) \) and then we can calculate the revenue rate of the BS. However, for large \( C \), direct calculation faces numerical overflow problems when calculating the normalization constant \( G(C) \). We have developed a set of fast recursive formulas that can evaluate the steady-state probabilities and the system performance under any reservation policy for systems with very large capacity and arbitrary number of call classes, with a computational complexity of \( O(C) \) [8].

2) Revenue Optimization: For single-service mobile cellular systems, with the assumptions of independent Poisson call arrivals and exponential channel occupancy times, the optimal (Markov, deterministic, and stationary) CAC policy is a reservation policy with ordered reservation parameters \( 0 = r_1^* \leq r_2^* \leq \cdots \leq r_K^* \leq C \) (see [1], [6], [8]). However, the theoretical results do not provide any practical method to determine the optimal reservation parameters.

Given the parameters \( C, K, \mu \), and \( (p_k, \lambda_k)_{k=1, \ldots, K} \), the revenue rate \( g \) is a function of the reservation vector \( r \in R^k = \{0, 1, \ldots, C\}^K \). Since \( |R| = (C+1)^K \), there exist \( O(C^K) \) different reservation policies. Finding the optimal reservation policy is a complex combinatorial optimization problem.

**Definition 1.** A reservation vector \( r = (r_1, r_2, \ldots, r_K) \in R^k \) is an ordered coordinate optimal point if it satisfies the following two properties:

\[
r_0 \Delta 0 = r_1 \leq r_2 \leq \cdots \leq r_K \leq r_{K+1} \Delta C \quad (3)
\]

\[
g(r_1, \ldots, r_k, \ldots, r_K) = g(r_1, \ldots, x, \ldots, r_K) \quad (4)
\]

for all \( x \) in \( [r_{k-1}, r_{k+1}] \), \( k = 1, \ldots, K \). A reservation policy with reservation vector \( r \) is an ordered coordinate optimal reservation policy if \( r \) is an ordered coordinate optimal point.

If the call arrival rates, the mean channel occupancy times, and the charging rates are finite, then for single-service mobile cellular systems the parameters of the optimal reservation policy constitute an ordered coordinate optimal point in \( R^k \). The following iterative coordinate search algorithm can be applied to find an ordered coordinate optimal reservation policy among all reservation policies.
\[ \eta = \eta(k)/\mu \] is the reservation rate of the simple complete sharing (CS) policy (under which calls of all classes are treated equally for channel access).

Let \( \eta_k = (\lambda_k/\mu)/C \) be the (normalized) traffic demand of class-\( k \) calls, and \( \eta = \sum_{k=1}^{K} \eta_k \) be the (normalized) total traffic demand of all classes. For any given \( \eta \), we consider the following four traffic distributions among different classes:

1. UNIF: \( \eta_k = 1/\eta \) for all \( k \).
2. H: \( \eta_k = 1/k \) for \( 1 \leq k \leq K/2 \), \( \eta_k = 1/k \) for \( k+1 \leq k \leq K \), i.e., higher-profit classes have higher traffic demands.
3. H: \( \eta_k = 1/k \) for \( 1 \leq k \leq K/2 \), \( \eta_k = 1/k \) for \( k+1 \leq k \leq K \), i.e., higher-profit classes have lower traffic demands.
4. RAND: \( \eta_k \) is chosen at random uniformly in \( [1/k, 1/k] \) for all \( k \).

For all cases we found that Algorithm 1 converges very fast, with the number of iterations on the order of \( K \). Even for \( K = 16 \) and \( C = 200 \), on average Algorithm 1 converges in a few seconds, running on a PC with Pentium III 933MHz CPU.

For \( K = 4 \), \( C = 20 \) (Table I), we can find the optimal reservation policy (which is also the optimal CAC policy) using the value iteration algorithm [8], [10]. It’s rather striking that for all the cases we studied the reservation policy returned by Algorithm 1 is exactly the optimal CAC policy.

For large \( K \) and \( C \), e.g., \( K = 16 \) and \( C = 200 \) (Table II), it is infeasible to find the optimal CAC policy using the value iteration algorithm or brute-force search. We compared the performance of the returned reservation policy with two heuristic reservation policies. We first determined the number \( \tilde{K} \) such that \( \sum_{k=1}^{K} \lambda_k/\mu \leq C < \sum_{k=1}^{\tilde{K}+1} \lambda_k/\mu \). Heuristic reservation policy 1 sets its parameters as follows:

\[ r_k = 0 \text{ for } 1 \leq k \leq \tilde{K}; \quad r_k = 5\%C \text{ for } \tilde{K} + 1 \leq k \leq K. \]

Heuristic reservation policy 2 sets its parameters as follows:

\[ r_k = 0 \text{ for } 1 \leq k \leq \tilde{K}; \quad r_k = 10\%C \text{ for } \tilde{K} + 1 \leq k \leq K. \]

For all cases the reservation policy returned by Algorithm 1 outperforms the two heuristic reservation policies, and significantly increases the revenue rate of the BS when the traffic demands are heavy.

### B. Multiservice Mobile Cellular Systems

Next generation mobile cellular systems are expected to evolve into multiservice systems that can support both narrowband and wideband multimedia services. In multiservice mobile cellular systems, calls of different classes may have different resource requirements and channel occupancy time distributions. Let \( n = (n_1, n_2, \ldots, n_K) \) denote the state of the system, where \( n_k \in \mathbb{Z}^+ \) is the number of class-\( k \) calls that are currently in the cell.

Under a threshold policy, each call class \( k \) is associated with a threshold parameter \( t_k \). A class-\( k \) call is accepted upon arrival if and only if there are sufficient free channels and the number of class-\( k \) calls in the cell does not exceed \( t_k \) after acceptance. Under a threshold policy with parameters

\[ \hat{r}_k = (\hat{r}_{k_1}, \hat{r}_{k_2}, \ldots, \hat{r}_{k_K}) = r_k - 1, \quad k_1, k_2, \ldots, k_K \text{ such that:} \]

\[ \hat{r}_k \text{ is chosen at random uniformly in } [1/k, 1/k] \text{ for all } k. \]

For all cases we found that Algorithm 1 converges very fast, with the number of iterations on the order of \( K \). Even for \( K = 16 \) and \( C = 200 \), on average Algorithm 1 converges in a few seconds, running on a PC with Pentium III 933MHz CPU.

For \( K = 4 \), \( C = 20 \) (Table I), we can find the optimal reservation policy (which is also the optimal CAC policy) using the value iteration algorithm [8], [10]. It’s rather striking that for all the cases we studied the reservation policy returned by Algorithm 1 is exactly the optimal CAC policy.

For large \( K \) and \( C \), e.g., \( K = 16 \) and \( C = 200 \) (Table II), it is infeasible to find the optimal CAC policy using the value iteration algorithm or brute-force search. We compared the performance of the returned reservation policy with two heuristic reservation policies. We first determined the number \( \tilde{K} \) such that \( \sum_{k=1}^{K} \lambda_k/\mu \leq C < \sum_{k=1}^{\tilde{K}+1} \lambda_k/\mu \). Heuristic reservation policy 1 sets its parameters as follows:

\[ r_k = 0 \text{ for } 1 \leq k \leq \tilde{K}; \quad r_k = 5\%C \text{ for } \tilde{K} + 1 \leq k \leq K. \]

Heuristic reservation policy 2 sets its parameters as follows:

\[ r_k = 0 \text{ for } 1 \leq k \leq \tilde{K}; \quad r_k = 10\%C \text{ for } \tilde{K} + 1 \leq k \leq K. \]

For all cases the reservation policy returned by Algorithm 1 outperforms the two heuristic reservation policies, and significantly increases the revenue rate of the BS when the traffic demands are heavy.
under any threshold policy with a computational complexity in [13] can be applied to evaluate the system performance solution algorithm with binary tree implementation proposed arbitrary channel occupancy time distributions. The convo-

optimal threshold policy [8].

1. For multiservice mobile cellular systems we define \( \eta_k = (\lambda_k b_k/\mu_k)/C \) and \( \eta = \sum_{k=1}^{K} \eta_k. \)

3) Numerical Experiments: We did a number of numerical experiments and the results are shown in Table III and Table IV. We found that Algorithm 2 converges very fast, with the number of iterations on the order of \( K. \) Even for \( K = 16 \) and \( C = 200, \) on average Algorithm 2 converges in 1-2 minutes, running on a PC with Pentium III 933MHz CPU.

For \( K = 4 \) and \( C = 20 \) (Table III), we are able to find the optimal threshold policy using brute-force search. It’s rather striking that for all the cases we studied the coordinate optimal threshold policy returned by Algorithm 2 is exactly the optimal threshold policy. For these cases we can also find the optimal CAC policy using the value iteration algorithm. We found that the optimal threshold policy yields a revenue rate sufficiently close to that of the optimal CAC policy.

For large \( K \) and \( C, \) e.g., \( K = 16 \) and \( C = 200 \) (Table IV), it is infeasible to find the optimal threshold policy using brute-force search. We compared the performance of the returned threshold policy with the heuristic RCS scheme proposed in [2]. For all cases the threshold policy returned by Algorithm 2 outperforms the RCS scheme, and significantly increases the revenue rate of the BS when the traffic demands are heavy.

### III. REVENUE OPTIMIZATION VIA PRICING

In the previous section, we assumed that the charging rates are predetermined and fixed by the mobile service provider, and we were trying to maximize the revenue rate of the BS via call admission control. In this section, we study the problem of how to choose the charging rates to maximize the revenue rate of the BS.

The model is similar as before: the BS has \( C \) channels and there are \( K \) different classes of calls; class-\( k \) calls arrive at the cell according to a Poisson process and have mean channel occupancy time \( 1/\mu_k; \) a class-\( k \) call, once accepted, occupies \( b_k \) channels and is charged at rate \( p_k \) per unit time when in progress. In addition, we assume that there is a known nonnegative demand function \( \lambda_k(p_k), \) which determines the
arrival rate of class-\(k\) calls as a function of the charging rate \(p_k\).

Let \(P_k\) be the set of potential charging rates for class-\(k\) calls. For example, \(P_k\) can be the set \(\{p_k : p_k = i\Delta p, i = 0, 1, \ldots, \frac{K}{\Delta p}\}\), where \(p_k\) is the maximum possible charging rate above which the demand \(\lambda_k(p_k)\) drops to zero, and \(\Delta p\) is the minimum charging unit (e.g., 1 cent). Let \(P_k = [P_k]\).

A pricing scheme is a rule that determines the current charging vector \(p = (p_1, p_2, \ldots, p_K)\) as a function of the current system state \(n = (n_1, n_2, \ldots, n_K)\). Let \(p_k(\lambda_k)\) be the inverse demand function (if it exists, e.g., if \(\lambda_k(p_k)\) is continuous and strictly decreasing in the range \(p_k \in [0, \bar{p}_k]\)). Let \(f_k(\lambda_k) = \lambda_k p_k(\lambda_k)/\mu_k\) be the instantaneous revenue rate function. Let \(\hat{g}^{ub}\) be the optimal value of the following nonlinear optimization problem:

\[
\begin{align*}
\text{maximize} & \quad \sum_{k=1}^{K} f_k(\lambda_k) \\
\text{subject to} & \quad \sum_{k=1}^{K} \lambda_k b_k(\lambda_k) / \mu_k \leq C.
\end{align*}
\]

It has been shown that if the functions \(f_k\) are concave, then \(\hat{g}^{ub}\) provides an upper bound for the revenue rate of the optimal pricing scheme [9].

### A. Optimal or Near-Optimal Static Pricing

A pricing scheme is static if the charging vector is fixed, i.e., it is independent of the system state. [9] shows that static pricing schemes are asymptotically optimal in a number of limiting regimes (e.g., the regime involves a system with a large number of small users). Static pricing schemes also have some obvious advantages over dynamic pricing schemes. Therefore, we focus on static pricing schemes.

We assume that the BS employs a certain CAC policy. For example, it can be a reservation policy, a threshold policy, or just the simple CS policy. Given the parameters \(C, K, b_k, \mu_k\), the demand functions \(\lambda_k(p_k)\) for \(1 \leq k \leq K\), and the specific CAC policy, the revenue rate \(g\) is a function of the charging vector. Since there exist \(|P| = \prod_{k=1}^{K} P_k\) different charging vectors, finding the optimal static pricing scheme is a difficult task for large \(K\).

**Definition 3.** A charging vector \(p = (p_1, p_2, \ldots, p_K) \in P\) is a coordinate optimal point if it satisfies the following coordinate optimal property:

\[
g(p_1, \ldots, p_{k-1}, \xi, \ldots, p_K) \geq g(p_1, \ldots, p_{k-1}, p_k, \ldots, p_K)
\]

for all \(\xi \in P_k, \ k = 1, \ldots, K\). A static pricing scheme with charging vector \(p\) is a coordinate optimal static pricing scheme if \(p\) is a coordinate optimal point.

If the call arrival rates, the mean channel occupancy times, and the potential charging rates are finite, then the charging vector of the optimal static pricing scheme is a coordinate optimal point in \(P\). We can apply the iterative coordinate search algorithm to find a coordinate optimal static pricing scheme among all static pricing schemes.

#### Algorithm 3

**Iterative Coordinate Search Algorithm for Static Pricing Schemes**

1. Choose an initial charging vector \(p^0\), e.g., \(p_k^0 = 0\) for all \(k\). Set \(m = 1\).
2. For \(k = 1, 2, \ldots, K\), search along the \(k\)-th coordinate set \(P_k\) in sequence to find \(p_k^m\) such that:

\[
p_k^m = \arg\max_{\xi \in P_k} g(p_1^m, \ldots, p_{k-1}^m, \xi, p_{k+1}^m, \ldots, p_K^m)
\]

(11)

(In a situation where more than one \(\xi \in P_k\) achieve the same maximum \(g\), the largest \(\xi\) is chosen to break the tie.)

3. If \(p_k^m = (p_1^m, p_2^m, \ldots, p_K^m) = p_k^{m-1}\), stop. Otherwise, increase \(m\) by 1 and go to step 2.

As before, Algorithm 3 converges in a finite number of iterations and the returned static pricing scheme is a coordinate optimal static pricing scheme.
B. Numerical Experiments

We did a number of numerical experiments and the results are shown in Table V and Table VI. Like in [9], we used linear demand functions of the form \( \lambda_k(p_k) = a_k - c_k p_k \). The BS employs the CS CAC policy under which calls of all classes are treated equally for channel access. For all cases, we calculated the upper bound for the optimal revenue rate by solving the nonlinear optimization problem (9).

We found that Algorithm 3 converges very fast, with the number of iterations on the order of \( K \). Even for \( K = 16 \) and \( C = 200 \), on average Algorithm 3 converges in a few seconds, running on a PC with Pentium III 933MHz CPU.

For \( K = 2 \) and \( C = 200 \) (Table V), we are able to find the optimal static pricing scheme using brute-force search. For most cases the static pricing scheme returned by Algorithm 3 is exactly the optimal static pricing scheme. In addition, the optimal static pricing scheme yields a revenue rate very close to the upper bound.

For large \( K \) and \( C \), e.g., \( K = 16 \) and \( C = 200 \) (Table VI), it is infeasible to find the optimal static pricing scheme using brute-force search. Nevertheless, the static pricing scheme returned by Algorithm 3 yields a revenue rate close to the upper bound.

IV. Conclusions

In this paper we studied two approaches to optimize the revenue rate for mobile cellular systems: one through call admission control, the other through pricing. We focused on CAC policies that have simple structures and static pricing schemes in which the charging rates are independent of the system state. We showed that our recently proposed iterative coordinate search algorithm provides an efficient and effective method to find the optimal or near-optimal structured CAC policies and static pricing schemes for mobile cellular systems.

Which approach should one employ to optimize the revenue rate, CAC or pricing? Our suggestion is both. First a service provider should have an estimation of the user demand functions based on some long-term statistics of observed user behaviors, and determine the corresponding optimal or near-optimal charging rates. Since the estimation may not be accurate and the user demands may vary over time, a BS can measure and estimate the user demands (call arrival rates) in real time, and then implement the optimal or near-optimal structured CAC policy. The service provider and the BS periodically update the charging rates and the CAC policy based on new statistics and estimations, in different time scales. For example, the charging rates may be updated in months or weeks, and the CAC policy may be updated daily.

REFERENCES